Geo-DBMS als standaard bouwsteen voor Rijkswaterstaat

Theo Tijssen, Wilko Quak & Peter van Oosterom

GISt Rapport No. 59

GIS TECHNOLOGY

TU Delft Delft University of Technology

Challenge the future

Geo-DBMS als standaard bouwsteen voor Rijkswaterstaat

Theo Tijssen, Wilko Quak & Peter van Oosterom

Oktober 2012

GISt Rapport No. 59

Samenvatting

Oracle Spatial fungeert binnen Rijkswaterstaat als standaard bouwsteen voor ruimtelijke toepassingen. Doel van dit onderzoek is vast te stellen of andere DBMS'en op een zodanig niveau zijn dat deze ook als standaard bouwsteen zouden kunnen fungeren. De ontwikkeling van open source software biedt mogelijk een alternatief en kan wellicht tot interessante kostenbesparingen leiden. Het onderzoek betreft de vergelijking van Oracle Spatial met PostgreSQL / PostGIS, SQL Server en MySQL en heeft in drie fasen plaatsgevonden: fase 1 bestond uit een globale vergelijking van alle relevante DBMS'en (op basis van beschikbare openbare documentatie, waarbij MySQL is afgevallen vanwege te beperkte ruimtelijke functionaliteit); fase 2 bestond uit analyse van referenties en ervaringen van andere gebruikers (ook op niet technische aspecten, helaas konden in Nederland geen geschikte SQL Server referenties gevonden worden); fase 3 bestond uit het daadwerkelijk testen van de overgebleven DBMS'en (PostGIS en SQL Server, resultaten met Oracle vergeleken). De testen zijn uitgevoerd met een representatieve dataset: het Nationaal Wegen Bestand (NWB). De bijlagen bij dit rapport bevatten details met de systeemconfiguratie, de data, de selecties, de testresultaten en de reacties van de betrokken leveranciers/bouwers. Hoewel de verschillende databases ook op ruimtelijk vlak elk hun eigen sterke punten hebben kan worden gesteld dat naast Oracle, PostGIS het meest volwassen is en als standaard bouwsteen kan worden gezien.

ISBN: 978-90-77029-33-6
ISSN: 1569-0245

Section GIS technology
OTB Research Institute for Housing, Urban and Mobility Studies
TU Delft
Jaffalaan 9, 2628 BX Delft, the Netherlands
Tel.: +31 (0)15 278 4548; Fax +31 (0)15-278 2745
Websites: http://www.otb.tudelft.nl
http://www.gdmc.nl

E-mail: t.p.m.tijssen@tudelft.nl, c.w.quak@tudelft.nl, p.j.m.vanoosterom@tudelft.nl

Inhoud

1 Globale vergelijking databases

De eerste fase van het onderzoek naar alternatieve DBMS bouwstenen voor ruimtelijke toepassingen betreft een algemene vergelijking van geboden functionaliteit voor 4 systemen: MySQL, PostGIS, Oracle en SQL Server. Er zal naar alle (ruimtelijke) facetten van de genoemde databases gekeken worden, alleen met relatief weinig diepgang in deze eerste fase. In volgende fases zal naar minder systemen en minder facetten gekeken worden, maar wel met meer diepgang.

In dit hoofdstuk zal eerst de globale vergelijking gepresenteerd en toegelicht worden (1.1), daarna volgen de conclusies en de onderbouwing van gemaakte keuzes (1.2).

1.1 Databases vergeleken

De manier waarop in dit onderzoek de databases op een globaal niveau zijn vergeleken is vooral geïnspireerd door de manier waarop dat eerder in GIM International door Chrit Lemmen gedaan is: ‘Product Survey on Geo-databases’ (http://www.gim-international.com/files/productsurvey_v_pdfdocument_14.pdf). Dit soort vergelijkingen wordt wel vaker uitgevoerd, een ander voorbeeld is http://www.bostongis.com/PrinterFriendly.aspx?content_name=sqlserver2008r2_oracle11gr2_postgis15_compare. De aspecten waarop de systemen vergeleken worden is in de loop van het onderzoek enigszins gewijzigd. Sommige aspecten uit de oorspronkelijke lijst bleken niet meer zo relevant, gezien de ICT ontwikkelingen. Andere aspecten zijn wat meer uitgesplitst om zo een iets gedetailleerder vergelijking mogelijk te maken. Tenslotte zijn er nog enige aspecten toegevoegd die specifiek voor Rijkswaterstaat interessant zijn.

Het resultaat van de vergelijking is in tabel 1 weergegeven. De informatie in deze tabel is hoofdzakelijk gebaseerd op vrij beschikbare documenten, met name de (online) manuals van de diverse systemen waar het de ‘geo-component’ betreft:

MySQL: http://dev.mysql.com/doc/refman/5.5/en/spatial-extensions.html
Oracle: http://docs.oracle.com/cd/E11882_01/appdev.112/e11830/toc.htm
PostGIS: http://postgis.refractions.net/documentation/manual-2.0/
SQL Server: http://msdn.microsoft.com/en-us/library/bb933790.aspx

Voor SQL Server was een nuttige aanvulling het document ‘New Spatial Features in SQL Server Code-Named "Denali"’ (http://social.technet.microsoft.com/wiki/contents/articles/4136.aspx).

Om iets te kunnen zeggen over de mate waarin men zich al dan niet aan de internationale OGC en ISO standaarden houdt zijn deze ook geraadpleegd. De belangrijkste standaard voor (vector data in) geo-databases is ‘OpenGIS Implementation Standard for Geographic information - Simple feature access’ (Part 1: Common architecture, Part 2: SQL option, Document 06-103r4). De hier genoemde standaard is de actuele 1.2.1 versie uit 2010. Opvallend is dat de DBMS documentatie vrijwel uitsluitend refereert aan de oudere 1999 versie ‘OpenGIS Simple Features Specification - For SQL, Revision 1.1’ (Document 99-049). Deels overlappend met de OGC standaard, maar voor een groot deel ook een uitbreiding, vormt de ISO ‘SQL/MM’ 13249-3 standaard: ‘Information technology - Database languages - SQL multimedia and application packages - Part 3: Spatial’.

Een eerste versie van de tabel met de globale vergelijking is voorgelegd aan de DBMS leveranciers. Dit heeft geleid tot een enkele aanpassing en enige opmerkingen bij de tabel die in dit rapport zijn verwerkt. Waar er een '?' staat in de tabel was het met de beschikbare informatie niet met volledige zekerheid mogelijk het betreffende aspect te beoordelen.

Product				
DBMS product name, version, release date	**MySQL Community Edition** Version 5.5.25, May 2012 (MySQL 5 released December 2009)	**Oracle Database 11g Enterprise Edition** Release 11.2.0.3.0, September 2011 (Oracle 11g R2 released September 2009)	**PostGIS** Version 2.0.1, June 2012 (PostGIS 2 released May 2012)	**SQL Server 2012**, April 2012
Standardization				
Adherence to OGC / ISO standards	Subset of OGC 'Simple feature access' implemented, also no metadata views	Supports OGC 'Simple feature access' types and functions and subset of ISO 'SQL/MM'	Implements OGC 'Simple feature access', with extensions, subset of ISO 'SQL/MM' supported	Implements OGC 'Simple feature access', with extensions, subset of ISO 'SQL/MM' supported
Features				
Vector data types	Only Geometry types, (Multi-) Point / Linestring / Polygon, GeometryCollection	Integrated spatial type (SDO_GEOMETRY) supporting Geometry and Geography reference systems, (Multi-) Point / Linestring / Polygon, GeometryCollection, Solid	Geometry and Geography types, (Multi-) Point / Linestring / Polygon, GeometryCollection	Geometry and Geography types, (Multi-) Point/ Linestring/ Polygon, GeometryCollection
Z coordinates, Measures	No?	Yes	Yes	Yes
3D support	No	3D Polygon, 3D PolyhedralSurface, 3D Solid, some length, area, volume and distance functions	3D Polygon, 3D PolyhedralSurface, some length, area and distance functions	No
Curves	No	Circle, circular arcs in 2D lines and polygons	Circular arcs in (2D?) lines and polygons	Circular arcs in (2D?) lines and polygons
Basic vector operations	Subset of operations implemented	Yes	Yes, following OGC standard and with extensions	Yes, following OGC standard and with extensions
Advanced vector operations	No	Yes	Yes, following OGC standard and with extensions	Yes, following OGC standard and with extensions
Spatial relationships	Subset of relationships based only on MBR (Minimal Bounding Rectangle)	Yes, 9-intersection model	Yes, 9-intersection model	Yes, 9-intersection model
Geometry validation	Limited (only IsSimple)	Yes (sometimes with Oracle-specific rules)	Yes	Yes
Spatial join algorithm, optimization	No	Yes, integrated in query optimizer	Yes, integrated in query optimizer	No?
Spatial indexing	2D R-tree, only on MyISAM tables	R-tree, up to 4 dimensions	R-tree, up to 4 dimensions	2D B-tree on 4 or 8 grid levels (in Hilbert space-filling curve order)

	MySQL Community Edition 5.5.25	**Oracle Database 11g R2 Enterprise Edition**	**PostGIS 2.0.1**	**SQL Server 2012**
Spatial data clustering	No, but partitioning on non-spatial attributes supported	No, but partitioning on non-spatial attributes supported	Yes	No?
Linear referencing	No	Yes	Yes	No
Network (node - link) topology	No	Yes	Yes	No
Planar (node - edge - face) topology	No	Yes	Yes	No
Point cloud, TIN	No	Point cloud and TIN supported	TIN supported	No
Raster data types	No	SDO_GEORASTER (metadata) and SDO_RASTER (cell values) data types	Integrated RASTER data type (comparable to vector data type)	No
Raster data operations	No	Storage (using tiling, compression, pyramids) and retrieval	Yes, integrated / compatible with vector operations	No
Spatial reference systems, transformations	Only SRID values, no transformations	Mix of EPSG and Oracle-specific reference systems, transformations supported	Yes, using EPSG and PROJ.4 library	Using EPSG reference systems, no transformations
Support for spatial/temporal modelling	Yes, versioning supported by InnoDB tables	Yes, via versioning	Yes, built-in support	No?
Supported by ArcSDE	No	Yes (SDO_GEOMETRY, ST_GEOMETRY, SDELOB, WKB_GEOMETRY)	Yes (PostgreSQL 8.4 / PostGIS 1.4 versions; GEOMETRY, ST_GEOMETRY)	Yes (SQL Server 2008 supported, 2012 in ArcSDE 10.1?; GEOMETRY, GEOGRAPHY, SDEBINARY, OGCWKB)
Data exchange				
Standard input/output formats	Well-Known Text, Well-Known Binary	SDO_GEOMETRY, Well-Known Text, Well-Known Binary, GML, KML	Well-Known Text, Well-Known Binary, GML, KML, GeoJSON	Well-Known Text, Well-Known Binary, GML
Other output formats	No	No	X3D (partial), SVG	No
Supported by FME	Yes	Yes	Yes	Yes
Supported by OGR	Yes	Yes	Yes	Yes
Platform				
Platforms	Source available; binaries available for many different Linux platforms, various Unix platforms, MS Windows, Apple Mac OS	Generic Linux version, various Unix platforms (AIX, HP-UX, Solaris), MS Windows	Source available; binaries available for many different Linux and Unix platforms, MS Windows, Apple Mac OS	Various MS Windows versions (Vista, Server 2008, 7)

Tabel 1. Globale vergelijking MySQL, Oracle, PostGIS en SQL Server als geo-database.

De meeste aspecten behoeven geen toelichting voor iemand met enige ervaring met betrekking tot geo-databases, maar bij enkele is dat wel nuttig. Het meest duidelijk is de tabel als er ‘Nee’ staat in een cel. Staat er ‘Ja’, eventueel gevolgd door een bepaalde detaillering, dan is de betreffende ‘feature’ op een zeker minimum niveau aanwezig in de database. Maar de functionaliteit en kwaliteit die dan geboden wordt kan nog aanzienlijk verschillen (dit geldt vooral voor de meer geavanceerde features). De ‘Basic vector operations’ zijn alle operaties die geometrie aanmaken, (een deel van) een geometrie opvragen, een geometrie wegschrijven en de ‘eenvoudiger’ bewerkingen en berekeningen (bijv. schalen, of lengte en oppervlak berekenen). De ‘Advanced vector operations’ betreffen o.a. het genereren van buffers, het berekenen van verschil (ST_Difference) of vereniging (ST_Union) van twee geometries, en het zoeken naar ‘dichtstbijzijnde’ objecten. Met ‘Linear referencing’ wordt bedoeld het koppelen van attributen of ‘events’ aan een punt op, of een deel van, een lijnobject. Een typisch voor beeld is het toekennen van huisnummers aan punten op/delen van een straat.

Wat betreft de ondersteuning van internationale standaarden, geen van de DBMS’en is voor de onderzochte versie formeel gecertificeerd op dit moment. Enkele oudere versies van bijv. Oracle zijn wel gecertificeerd. Dat klinkt wat vreemd omdat het ‘native’ en generieke Oracle geometrie type SDO_GEOMETRY is, hetgeen niet overeenkomt met de geometrie typen van de standaarden. Maar de formele eisen aan ‘compliant’ producten zijn zodanig ruim dat afwijkende datatypen en functie namen geen bezwaar zijn voor certificering. Om deze reden wordt de onderstening van de standaarden bij Oracle als ‘supports’ aangeduid, hetgeen een minder volledige ondersteuning van de standaard impliceert dan de ‘implements’ kwalificatie bij de andere databases. Dit komt tevens terug bij de ‘basic’ en ‘advanced vector operations’. Deze zijn in Oracle beschikbaar maar grotendeels met andere namen dan gespecificeerd in de standaard (alleen ‘yes’ in de tabel), de andere databases volgen de standaard meer volledig (‘following OGC standard’ in de tabel). Overigens kent Oracle ook een aantal meer met de internationale standaarden overeenkomende ‘wrapper’ ST_* typen (ST_POINT, ST_POLYGON, etc.) en ST_* functies (ST_DISTANCE, ST_INTERSECTS, etc.).

De drie ‘Supported by’ aspecten genoemd in de tabel nemen een aparte positie in, ze hebben betrekking op andere software en zijn niet een kenmerk van een DBMS. Ze zeggen wel iets over de bruikbaarheid van de diverse databases, de inzetbaarheid daarvan wordt groter naarmate ze beter ondersteund worden door andere software.

De totale tabel overziend is het niet zo moeilijk een zekere ordening in de onderzochte databases aan te brengen. Aan de ene kant staat MySQL dat op ruimtelijk gebied relatief weinig te bieden heeft. Aan de andere kant staan Oracle en PostGIS die duidelijk veel te bieden hebben. SQL Server staat hier tussen in, met een functionaliteit die voor een aanzienlijk deel van de ruimtelijke toepassingen voldoende lijkt.

1.2 Conclusies globale vergelijking databases

De eerste fase van het DBMS onderzoek, een globale vergelijking, moet de informatie opleveren in hoeverre de onderzochte systemen in aanmerking komen om als alternatieve DBMS bouwsteen voor ruimtelijke toepassingen te fungeren. Uit het

onderzoek blijkt dat PostgreSQL/PostGIS en, in iets mindere mate, SQL Server die potentie hebben. In vergelijking met Oracle, dat bekend is en als referentie fungeert, hebben ze zoveel te bieden dat het verder onderzoeken van de mogelijkheden in de volgende fasen zeker de moeite waard is.

MySQL heeft die potentie niet. In de overzichtstabel worden een heleboel ‘features’ genoemd, maar die zijn niet allemaal essentieel in een DBMS om een meerderheid van ruimtelijke toepassingen te ondersteunen. Maar ook essentiële features ontbreken in MySQL:

- Er zijn te weinig ruimtelijke basis operaties geïmplementeerd, bij gebruik als spatial DBMS zou Rijkswaterstaat snel tegen allerlei hiaten aan lopen.
- Ruimtelijke selectie, cruciaal in een geo-database, is alleen mogelijk op basis van ‘bounding boxes’. De applicatie wordt dan verantwoordelijk voor de selectie van de objecten die men echt nodig heeft. Dit is generieke functionaliteit die in de database hoort en niet in de toepassing.
- Ruimtelijke indexen, noodzakelijk voor de snelheid van veel ruimtelijke operaties, zijn alleen mogelijk bij gebruik van tabellen die niet ‘transaction-safe’ zijn. Kies je voor de index dan blijft er een database over die nauwelijks die naam verdient. Kies je voor een echte database dan heb je geen ruimtelijke index en zal de performance bij veel operaties niet acceptabel zijn.

Geen enkele van de ruimtelijke queries zoals getest in de laatste fase van dit onderzoek zou met MySQL uitgevoerd kunnen worden op de manier waarop dat met de andere databases wel kan. Op te merken is verder dat MySQL op ruimtelijk gebied stagneert, de huidige functionaliteit bestaat al vrij lang en er zijn geen indicaties dat dit snel zal wijzigen. PostGIS en Oracle daarentegen brengen uitbreidingen en verbeteringen van de geo-component met iedere nieuwe versie. SQL Server ontwikkelt zich wat langzamer, maar van de 2008 naar de 2012 versie zijn er significante aanpassingen doorgevoerd.

In overleg met de opdrachtgevers is besloten om MySQL, gezien het gebrek aan functionaliteit op ruimtelijk gebied, niet verder mee te nemen in de rest van het onderzoek.

2 Ervaringen gebruikers

Dit hoofdstuk beschrijft het gebruik van PostGIS DBMS binnen verschillende projecten bij de Nederlandse Overheid. De informatie is grotendeels gebaseerd op een interview met Erik van der Zee (Geodan) die als IT-architect is betrokken bij het PDOK (Publieke Dienstverlening op de Kaart) project. Veel van de inzichten in dit hoofdstuk zijn ook bevestigd in discussies met Thijs Brentjens die als lid van de Open Geo groep support levert op Open Source producten. Enkele van de projecten waaraan Thijs gewerkt heeft zijn:

- Database voor Hansje Brinker om gegevens ten behoeve van dijk monitoring op te slaan en te distribueren via webservices: http://www.hansjebrinker.com/nl/
- Ook heeft Thijs PostGIS ingezet bij gegevensanalyses voor het VPRO televisieprogramma 'Nederland van Boven'.

Een andere interessante bron van informatie over PostGIS projecten is de weblog van Paul Ramsey (PostGIS architect): http://blog.cleverelephant.ca/

In de volgende paragrafen worden achtereenvolgens behandeld: Waarom is voor PostGIS gekozen (2.1) Stabiliteit en support (2.2) Functionaliteit van PostGIS (2.3) Microsoft SQL Server (2.4).

2.1 Keuze voor PostGIS als DBMS

Twee belangrijke redenen binnen PDOK om voor PostGIS te kiezen waren:

- Het 'open source tenzij' beleid van de Nederlandse overheid (zoals bijv. beschreven in het NORA - Nederlandse Overheid Referentie Architectuur – document) heeft aanzienlijk bijgedragen.
- Daarnaast kent PDOK een groot aantal beoogde gebruikers en worden er flinke pieken in het gebruik van de PDOK diensten verwacht (extreme pieken kunnen ontstaan is crisissituaties, bijv. bij overheid.nl die door PDOK services bediend wordt). Om deze pieken op te kunnen vangen is een zeer schaalbare architectuur nodig. Commerciële aanbieders (Oracle) kunnen dit prima leveren, maar door de licentiestructuur (je betaalt per CPU waarop een server draait) is dit zeer kostbaar. Ook met PostGIS is een schaalbare structuur te bouwen. Bij open source software zoals PostGIS betaal je geen licentiekosten en betaal je dus niet meer bij het inzetten van meerdere CPU's. Overigens is niet alle software die gebruikt is bij PDOK open source.

Bij het Kadaster, waar PDOK in beheer is, werd van origine voornamelijk Oracle gebruikt. Introductie van PostGIS naast Oracle levert wel extra kosten op:

- Bij invoering van PostGIS was er nog vrijwel geen expertise binnen het Kadaster met PostGIS, ingebruikneming van PostGIS levert extra kosten op in de vorm van: inwerken in het nieuwe product, opleiding en inhuur externe support. Wanneer je support nodig hebt kun je niet rechtstreeks naar de producent van de software maar moet je die bij een andere partij inhuren. In Nederland zijn deze partijen wel te

vinden (bijvoorbeeld bij CapGemini of Geodan via raamovereenkomsten die het Kadaster heeft met deze organisaties).
- Logischerwijs zijn er ook extra kosten verbonden aan het in de lucht houden van twee verschillende systemen; sommige dingen moet je dan dubbel doen.
- Over andere aspecten van de TCO is het moeilijk uitspraken te doen. Voor PostGIS ondersteuning is een organisatie niet verplicht bij een bepaalde leverancier te zijn, wat concurrentie bevordert. Uitvoeren van software installatie en updates is goed te doen, helemaal op Linux servers waar updates uitvoeren via de standaard software management systemen mogelijk is en geen specialistische kennis vereist. Omdat PostgreSQL standaard SQL gebruikt is het eigen maken van de werking eenvoudig voor iemand met algemene database vaardigheden. Hetzelfde geldt voor PostGIS, omdat dit (OGC) standaarden implementeert. Zowel PostgreSQL als PostGIS zijn zeer goed gedocumenteerd. PostGIS gebruikt doorgaans, t.o.v. andere ruimtelijke databases, weinig hardware resources, zoals geheugen. Dit zijn allemaal aspecten die bijdragen aan een lage TCO.

2.2 Stabiliteit en support

In praktijk blijk er geen verschil tussen de stabiliteit van Oracle of PostGIS. Als PDOK uit de lucht is ligt dat in ieder geval niet aan het DBMS. Dat is stabiel, draaiend op een Red Hat Linux server.

Binnen projecten zie je overigens vaak dat er niet los voor een DBMS gekozen wordt maar voor een volledige software stack. Voorbeelden (De .NET stack van Microsoft, een Java stack met open source software, een Oracle stack, of een volledige ESRI stack). Vaak blijkt dat zelfs wanneer in verschillende software stacks open standaarden gebruikt worden, bepaalde combinaties van software uit verschillende stacks voor problemen te zorgen. Zo wordt ArcGIS Server bijv. wel op een Linux server ondersteund, maar bij updates worden de Windows versies altijd eerder uitgebracht. De installed base van ArcGIS Server op Linux is vaak veel kleiner, dus de community van gebruikers dan ook. Problemen worden daardoor vaak moeilijker oplosbaar. Sommige pakketten doorbreken dit patroon. Bijvoorbeeld FME werkt prima samen met de verschillende stacks, ook met PostGIS. Ook Oracle functioneert goed als DBMS alternatief in sommige andere stacks.

Binnen PDOK is geen rechtstreeks contact met de ontwikkelaars van PostGIS. Omdat alle issues door eigen experts konden worden opgelost is er nooit contact gezocht. [1] Bij GeoServer en GeoWebCache (andere onderdelen van de open source stack van PDOK) is wel goede communicatie met de ontwikkelaars.

Wat updates betreft, er komen regelmatig nieuwe updates uit voor PostGIS. Deze worden ook geïnstalleerd in de serveromgeving van PDOK en hebben nog niet tot backwards compatibiliteits issues geleid.

[1] Bij de TU Delft hebben we regelmatig contact met ontwikkelaars van PostGIS en Oracle en door beide partijen worden onze commentaren serieus genomen.

2.3 Functionaliteit

PostGIS ondersteunt het ‘simple feature model‘ en flink wat uitbreidingen daarop. Wanneer echter specifieke functionaliteit buiten simple features gewenst is verdient het aanbeveling voor implementatie te checken in hoeverre dit werkt. Bij functionaliteit die buiten het simple feature model valt, zoals bijv. cirkelbogen, puntenwolken, rasters of een nearest neighbor query, geniet het aanbeveling eerst te kijken of dit wel ondersteund wordt en daarnaast de volwassenheid van de module te testen.

Tot nu toe is de PDOK server nog niet tegen grenzen in de hoeveelheid data aangelopen. In Bijlage 1 staat een overzicht van de datasets die momenteel via WFS uit de PostGIS database worden geserveerd. Bij deze datasets zitten ook de DTB en NWB bestanden van Rijkswaterstaat. Binnen het PDOK programma, waar Rijkswaterstaat deel van uitmaakt, is dus al veel expertise over het gebruik van PostGIS voor Rijkswaterstaatgegevens voorhanden.

Momenteel verwerkt de PDOK server twee miljoen hits per maand. Die worden door drie identieke virtuele servers met load-balancing afgehandeld. In eerste instantie draaide in iedere virtuele server een met een eigen kopie van de PostGIS DBMS met alle bijbehorende data. Omdat deze configuratie resulteerde in hele grote virtuele servers is onlangs de DBMS server uit de virtuele omgeving gehaald, zodat niet iedere server zijn eigen kopie van de hele DBMS hoefde te hebben. PostGIS staat nu dus buiten de virtuele omgeving.[2]

Om de pieken in het gebruik van PDOK af te vlakken (en zo performance problemen te voorkomen) zijn er twee dienstenniveaus gedefinieerd[3]:

1. PDOK basis. Deze is voor overheden en heeft een gegarandeerde beschikbaarheid.
2. PDOK fair use. Deze is voor niet-overheden. In de voorwaarden staat dat bij meer dan 10K hits per dag de gebruiker kan worden afgesloten (IP blocking).

Binnen PDOK zijn tot nu toe nog geen problemen ondervonden van specifieke queries die door het DBMS niet opgelost kunnen worden. Het type queries dat PDOK doet op het DBMS is ook beperkt omdat alle queries door de WFS server worden gegenereerd. Op de WFS server is wel geconfigureerd dat het resultaat van een query maximaal 15K objecten mag bevatten.

De queries die PDOK doet zijn vrijwel allemaal read only (er worden geen inserts, updates en deletes gedaan op het DBMS). Updates van de datasets worden gedaan door een volledig nieuw kopie van de dataset in het DBMS te importeren. Incrementeel updates van bijvoorbeeld de data van de basisregistraties is wel in de planning.

[2] Hoe PostGIS precies geconfigureerd is staat beschreven in een detail ontwerp van de configuratie. Voor het verkrijgen van dit document is goedkeuring van de PDOK projectleider nodig. Hiervoor is vooralsnog geen verzoek ingediend.

[3] Zie: http://www.geonovum.nl/dossiers/pdok/dienstenniveau

2.4 Microsoft SQL Server

De 2012 release van SQL Server ziet er veelbelovend uit maar navraag bij Microsoft heeft geen gebruikers opgeleverd om te interviewen. Inmiddels is gebleken dat een oudere versie van SQL Server gebruikt is bij het 'Portaal Natuur en Landschap'[4] van het IPO. Er is wel contact gezocht met dit project maar nog niet gevonden. Uit dit deel van het onderzoek valt alleen te concluderen dat het in Nederland nog niet echt storm loopt met SQL Server toepassingen. Vooralsnog zijn onze eigen experimenten (zie elders in dit rapport) nog de beste bron van informatie over dit product.

2.5 Conclusies op basis van referentiebezoeken

Op basis van de referentiebezoeken kan geconcludeerd worden dat PostGIS binnen verschillende overheidsorganen gebruikt wordt en dat men daar zeker tevreden is over de functionaliteit en de kosten van dit DBMS. Het verdient wel aanbeveling dat wanneer niet standaard (‘simple feature’) functionaliteit van PostGIS gebruikt wordt eerst de volwassenheid van dat deel van het DBMS te controleren. Over de TCO van een product is het moeilijk kwantitatieve conclusies te trekken omdat kosten van zeer veel factoren afhangen. In ieder geval betaal je geen licentiekosten voor PostGIS en zijn er diverse bedrijven die support bieden op dit product. Een andere conclusie van de referentiebezoeken is dat de keuze voor de DBMS niet losstaand te maken is maar altijd samenhangt met de rest van de software stack.

Wat betreft de ruimtelijke functionaliteit van SQLServer kan alleen maar geconcludeerd worden dat het nog weinig gebruikt wordt binnen de Nederlandse geo overheid. Over de kwaliteit van dit product zegt dat overigens niets.

[4] http://www.portaalnatuurenlandschap.nl

3 Testen databases

Uit de eerste fase van het onderzoek is naar voren gekomen dat er nog drie geo-databases zijn waarvoor het interessant is deze wat meer in detail te bekijken en ze door concreet testen ook echt 'langs de meetstok' te leggen. Net als eerder fungeert Oracle hierbij als referentie, het wordt op ruime schaal gebruikt binnen Rijkswaterstaat en de voor-en nadelen van dit DBMS zijn bekend.

In de overzichtstabel met ruimtelijke functionaliteit uit fase 1 worden een heleboel aspecten genoemd. Gezien de beschikbare tijd kan slechts een beperkt deel daarvan echt worden getest. Maar wel zijn die operaties gekozen waarmee een belangrijk stuk van de vraag naar 'ruimtelijke ondersteuning' uit Rijkswaterstaat toepassingen kan worden gedekt. Het betreft vooral het opvragen van vector data, in combinatie met administratieve data, en enige relatief simpele berekeningen (en daarnaast ook nog enkele meer geavanceerde operaties). Het testen concentreert zich op de functionaliteit van de databases, maar enige aandacht voor 'performance' is er ook.

In dit hoofdstuk zal eerst de 'omgeving' waarin getest is, worden beschreven (3.1). Vervolgens zal worden ingegaan op het prepareren van de data die bij de tests gebruikt is (3.2), een aspect dat bij dit soort onderzoek altijd verrassend veel moeite en tijd kost. Tenslotte komen de eigenlijke tests, in de vorm van test queries, aan de orde en wat die opgeleverd hebben (3.3).

3.1 Hardware, installatie en configuratie databases

Beschikbare hardware, operating system

Het testen van de databases heeft plaatsgevonden in de testomgeving van Rijkswaterstaat. Voor het onderzoek stonden twee servers ter beschikking: een Windows machine met daarop de SQL Server database, en een Linux machine met daarop de Oracle en PostgreSQL/PostGIS databases. Qua hardware zijn beide machines identiek, de belangrijkste gegevens over de systemen zijn:

Server:
- HP ProLiant BL460c G7
- 24 GB intern geheugen
- 2 Intel Xeon L5640 processoren, 2.27 GHz (totaal 12 cores, 24 threads)

Disk storage:
- SAN, iSCSI links, effectief 1 Gbit per connectie
- Software RAID5 disk set bestaande uit 4 x 300 GB disks
- (Serial Attached SCSI, 10000 toeren/min)

Operating systems:
- Windows Server 2008 R2 SP1, 64 bit
- Red Hat Enterprise Linux 6.1, kernel release 2.6.32-131.0.15.el6.x86_64

Iedere database heeft een 'dedicated' RAID5 disk set ter beschikking. Vanuit de database gezien betekent dit dat alle data op een enkele, logische disk staat, hoewel die disk dan fysiek wel uit 4 aparte disk drives bestaat. Dit is niet helemaal ideaal voor een database, in principe is het beter de I/O naar tabellen, indexen, logs, tijdelijke werkruimte e.d. te spreiden over aparte disks (met onafhankelijke I/O

channels). Gezien de beperkte hoeveelheid data gebruikt in het onderzoek, en ook de geringe mate waarin data in de database weggeschreven wordt (alleen bij het laden en enigszins bij sommige queries) was de verwachting dat de I/O geen probleem zou vormen (later bleek dat Oracle in een bepaalde situatie toch problemen zou krijgen). Vóór het uitvoeren van de test queries is onderzocht met welke snelheid data gelezen/geschreven kan worden. Een ‘sustained data rate’ van 100 MB/sec bleek makkelijk haalbaar, werkend vanuit de database. Tijdens de test queries komen dergelijke data rates niet voor, bij de verschillende databases ‘piekte’ de I/O op 75-80 MB/sec.

Voor het werk dat gedaan moet worden is de testserver enigszins ‘overgedimensioneerd’. Het testen van de databases is het enige dat er op de betreffende systemen draait. Er is ruim voldoende memory beschikbaar zodat de database (voor zover de configuratie het toelaat) en het operating system in principe alle gebruikte data kunnen cachen. De 24 CPU’s worden door Oracle en PostGIS niet gebruikt (niet meer dan 1 tegelijk), wel echter door SQL Server. Met tools als ‘top’ en ‘iostat’ onder Linux, en de ‘Performance Monitor’ onder Windows, is het resource gebruik van de databases in de gaten gehouden.

Oracle

De Oracle DBMS software is geïnstalleerd door Rijkswaterstaat DBA’s op de voor hen gebruikelijke manier, inclusief een ‘kale’ database. De installatie is afgemaakt door de onderzoekers: er zijn REDO log files toegevoegd, en er is een tablespace van 20 GB aangemaakt, ruim voldoende voor de in dit onderzoek te gebruiken data.

Als belangrijkste opstartparameter is ingesteld ‘`memory_target = 1024M`’ (voor alle parameters, zie Bijlage B3.01). Dit betekent dat Oracle in totaal 1 GB aan memory toegewezen krijgt dat het DBMS dan verder zelf gaat toekennen aan de diverse ‘onderdelen’ (zoals de buffer cache, row cache, library cache, shared pool, java pool, sort, hash-join, etc.). Een dergelijke memory omvang betekent dat Oracle start met een buffer cache van ongeveer 250 MB, maar het systeem kan die cache vergroten als het dat nuttig vindt. In de praktijk betekent het dat de Oracle ‘image’ ruim 1 GB in omvang is (gealloceerd memory), maar dat daarvan 300-350 MB gebruikt wordt (resident memory) bij niet al te intensief gebruik, zoals de test queries in dit onderzoek.

PostgreSQL/PostGIS

De PostgreSQL/PostGIS software is door de onderzoekers geïnstalleerd. Op de testserver ging dit niet zo makkelijk als normaal onder Linux omdat er geen installatie server ter beschikking stond. Het gevolg is dat alle ‘prerequisites’ handmatig als individuele RPM pakketten geïnstalleerd moeten worden (RPM: een 'package manager' voor het installeren, updaten, verwijderen, verifiëren en bevragen van softwarepakketten). Voor PostGIS kan dat een hele lijst worden omdat er behoorlijk was ‘dependencies’ zijn. In de eerste plaats is PostGIS natuurlijk afhankelijk van PostgreSQL, het DBMS waarvan PostGIS een ruimtelijke extensie is. PostgreSQL zelf heeft overigens ook al (beperkte) support voor ruimtelijke data, wat PostGIS toevoegt is echter veel rijker aan functionaliteit dan wat er ‘native’ in PostgreSQL zit.

PostGIS op zijn beurt is afhankelijk van een drietal software pakketten voor bepaalde delen van de functionaliteit:

PROJ.4: coordinaat transformaties
GEOS (Geometry Engine - Open Source): o.a. de 'topologische' relaties tussen geometries
GDAL (Geospatial Data Abstraction Library): o.a. raster operaties

Met name GDAL heeft ook weer veel dependencies. Uiteindelijk zijn de volgende pakketten geïnstalleerd in de volgorde zoals aangegeven om op de testserver het PostgreSQL/PostGIS DBMS beschikbaar te krijgen:

```
cfitsio-3.240-3.el6.x86_64.rpm
libgeotiff-1.3.0-4.el6.x86_64.rpm
hdf5-1.8.5.patch1-7.el6.x86_64.rpm
netcdf-4.1.1-3.el6.2.x86_64.rpm
libtool-ltdl-2.2.6-15.5.el6.x86_64.rpm
unixODBC-2.2.14-11.el6.x86_64.rpm
ogdi-3.2.0-0.14.beta2.el6.x86_64.rpm
librx-1.5-14.el6.x86_64.rpm
xerces-c-3.0.1-0.20.1.el6.x86_64.rpm
libspatialite-2.4.0-0.6.RC4.el6.x86_64.rpm
libdap-3.11.0-1.el6.x86_64.rpm

proj-4.7.0-1.rhel6.x86_64.rpm
geos-3.3.5-1.rhel6.x86_64.rpm
gdal-1.8.1-1.el6.x86_64.rpm

postgresql91-9.1.4-3PGDG.rhel6.x86_64.rpm
postgresql91-docs-9.1.4-3PGDG.rhel6.x86_64.rpm
postgresql91-libs-9.1.4-3PGDG.rhel6.x86_64.rpm
postgresql91-server-9.1.4-3PGDG.rhel6.x86_64.rpm
postgis2_91-2.0.1-1.rhel6.x86_64.rpm
postgis2_91-docs-2.0.1-1.rhel6.x86_64.rpm
```

Na installatie van de software is de volgende stap het initialiseren van een PostgreSQL database omgeving, dat gebeurt met het '`initdb`' commando. Op dit moment wordt er een locatie vastgelegd waar de belangrijkste onderdelen van de database terecht komen (o.a. 'master' database, log files), In principe is de database nu klaar voor gebruik, maar in de meeste gevallen is nog enige verdere configuratie nodig. De toegang tot een PostgreSQL database wordt geregeld vanuit de '`pg_hba.conf`' configuratie file, in ieder geval moet 'locale' toegang tot PostgreSQL hierin geconfigureerd zijn. Alle opstartparameters voor de database staan in de '`postgresql.conf`' configuratie file. Vergelijkbaar met Oracle worden daar bijv. de parameters t.a.v. het gebruik van memory opgegeven. Voor de test database is o.a. gespecificeerd '`shared_buffers = 256MB`' (voor de complete configuratie file zie Bijlage B3.02). Naast dit type memory kunnen nog de omvang van verschillende typen 'work' memory, tijdelijke buffers en een '`max_stack_depth`' worden opgegeven, naast een heleboel andere parameters die de werking van de database beïnvloeden. De PostgreSQL 'image' als resultaat van de opgegeven parameters is ongeveer 500 MB in omvang, gebruikt memory tijdens queries zoals uitgevoerd in dit onderzoek is 300-350 MB. Na alle genoemde configuratie kan de PostgreSQL database ook echt worden gestart.

In de ‘master’ database die nu bestaat zijn de volgende stappen het creëren van ‘roles’ (users), en eventueel het definieren van tablespaces. In PostgreSQL is dat laatste een simpele operatie in vergelijking met Oracle: er wordt een subdirectory aangemaakt en naar die directory wordt een ‘symbolische link’ gelegd. Ondertussen zijn we nu zover dat er een nieuwe database aangemaakt kan worden, dit kan bijv. met het ‘`createdb`’ commando. En de laatste stap om tot een echte geo-database te komen is het ‘laden’ van de PostGIS extensie in de nieuw aangemaakte database. Dit kan met het SQL commando ‘`create extension postgis;`’ (en desgewenst ‘`create extension postgis_topology;`’). Er zijn andere methoden dan hier beschreven om PostGIS te laden in een PostgreSQL database, en dat proces kan ook geautomatiseerd worden.

SQL Server

De installatie van SQL Server op de Windows machine bestond uit het starten van de installatie en vervolgens op de juiste plaatsen in de menus op OK drukken. Het enige niet geheel standaard element van de installatie was het opgeven van de locaties waar de data terecht moet komen (voor de opstartparameters zie Bijlage B3.03). Een kleine vertraging werd veroorzaakt door het niet geactiveerd zijn van een specifieke .NET module op de testserver. Vanuit de standaard management tool (SQL Server Management Studio) is vervolgens met een enkele klik een nieuwe database aangemaakt.

FME

FME (Feature Manipulation Engine van Safe Software) wordt in dit onderzoek gebruikt om de benodigde data van de ene in de andere database te krijgen. De 2012 Desktop versie van FME is geïnstalleerd op de Windows machine. De diverse databases worden volledig ondersteund door FME. Alleen is er voor het gebruik van Oracle wel een speciale licentie nodig. In combinatie met PostGIS bleek FME de functienamen van een oudere versie van PostGIS te gebruiken (bij de overgang naar PostGIS 2.0 zijn een aantal functies van naam veranderd). Deze discrepantie is op te lossen door het draaien van een ‘legacy’ script in PostGIS waarmee de huidige functies ook met hun oude naam beschikbaar komen. De firewalls van de bij de test betrokken machines bleken bij standaard instellingen de communicatie tussen FME en de databases in de weg te zitten. De firewalls zijn zodanig aangepast dat netwerkverkeer tussen FME en de databases doorgelaten wordt.

3.2 Data preparatie

Keuze data

Voor het testen van de databases was het idee dat te doen met data die relevant is voor Rijkswaterstaat. Na enige discussie kwam naar voren dat potentieel interessant zouden zijn de datasets van het NWB (Nationaal Wegen Bestand) en die van Kerngis (die een droge en een natte variant kent). De NWB data was beschikbaar als een Oracle dump van de (Oracle) database. Kerngis werd aangeleverd als een set van 29 ArcGIS Personal File Geodatabases.

Na bestudering van deze datasets en verdere discussie is besloten de Kerngis data toch niet in het onderzoek te gebruiken. De reden is tweeledig. Kerngis bestaat uit een

database met daarop een ArcGIS applicatie. Door deze opzet is de data sterk verweven met de manier waarop ArcGIS werkt. De geleverde Geodatabases bevatten slechts een deel van de data, een ander deel zit in gerelateerde ArcGIS 'update' tabellen. Het zou lastig zijn en veel tijd kosten om een voor het database onderzoek bruikbare set Kerngis tabellen uit deze structuur te verkrijgen, een reden om de Kerngis data niet te gebruiken. Kerngis bevat meer tabellen met vlakken dan het NWB, maar omdat deze laatste ook wel vlakken bevat, naast punten en lijnen, kan er met alle belangrijke typen geometrie worden getest. Het weglaten van Kerngis staat niet het testen van de voor Rijkswaterstaat relevante ruimtelijke vragen in de weg.

Initieel data laden en valideren

De Oracle dump met NWB data bevat ruim 60 tabellen, deze hebben betrekking op wegen, spoorwegen en vaarwegen. Als eerste stap in het testproces is de dump geladen in de Oracle database (zie Bijlage B3.04). In het oorspronkelijke NWB is de data verdeeld over verschillende users, tablespaces, etc. Om het verdere testen wat eenvoudiger te maken is alles geladen onder een enkele user in dezelfde tablespace. Ook is alleen de tabel data als zodanig geladen, zaken als constraints, triggers, sequences etc. worden niet meegenomen. Wat dan overblijft is de 'zuivere' tabel, met NOT NULL 'constraints'. Voor het beheer en de bijhouding van het NWB zijn al die 'aanvullende' database mogelijkheden uiteraard van groot belang, bijv. voor het consistent houden van de data. Maar voor de database queries zoals uitgevoerd in het huidige onderzoek spelen ze geen rol en kunnen dus weggelaten worden.

Het is een verstandige gewoonte om de gegevens die men gaat gebruiken in een database te controleren op kwaliteit. De database zelf biedt hiervoor verschillende mechanismen, bijv. de al eerder genoemde constraints en triggers die men bij het laden of updaten van data kan toepassen. Een fundamentele keuze op dit punt is of je (zo veel mogelijk) controleert bij het opslaan van de data, zodat er 'nooit' ongeldige data in de database komt. Een andere keuze is (het meeste) accepteren wat aangeboden wordt en dan in een tweede fase controleren of de data in de database al dan niet correct is. Bij ruimtelijke data in databases is er meestal sprake van een mix van beide aanpakken: sommige aspecten van de datakwaliteit worden gecontroleerd tijdens het opslaan, andere aspecten moet men na opslag alsnog zelf controleren. De reden van deze mix is het feit dat sommige aspecten van validiteit bij ruimtelijke data behoorlijk lastig te controleren zijn. Bijv. de vraag of een 2D polygon correct is (om nog maar niet te spreken van 3D polygons of nog moeilijker 3D solids), is een behoorlijk rekenintensieve procedure. Als je alle aspecten van ruimtelijke data validiteit altijd bij laden zou controleren, zou het laadproces van die data ongewenst traag worden.

Het 'lastige' van verschillende databases, ook die gebruikt in het huidige onderzoek, is dat ze t.a.v. bovenstaande verschillende strategieën toepassen. Oracle is een voorbeeld van een database die bij het opslaan van ruimtelijke data relatief weinig controleert, PostGIS bijv. controleert meer, maar ook lang niet alles. Voor alle databases geldt dat er na laden van ruimtelijke data nog 'invalid' data kan bestaan, vóór je deze data gaat gebruiken moet het worden gecorrigeerd (of verwijderd).

Na het laden van de Oracle dump met NWB data is dus als eerste de Oracle validatie functie (SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT) op alle

kolommen met ruimtelijke data ‘losgelaten’ (zie bijlage B3.05). Hieruit bleek dat alle tabellen met punten en lijnen correct waren, maar dat er bij de tabellen met polygons diverse problemen waren (bij een tolerantie van 0.5 mm, de tolerantie die voor het NWB gehanteerd wordt). De aard van deze problemen was:

NULL geometrie
grens van polygon snijdt zichzelf
overlappende ringen in een polygon
dubbele punten in een geometrie
verkeerde orientatie van binnen- of buitenringen

Bij navraag bleek de eerste categorie, NULL geometrie, verklaarbaar door het opnemen van een ‘lege’ geometrie op de plekken waarvan men weet dat er in de nabije toekomst een nieuw object moet komen. De andere problemen zouden eigenlijk niet mogen voorkomen, maar zijn wellicht te verklaren door ‘oudere’ data (NWB bevat ook historie) die niet geheel correct is.

Bij gebruik van de data, en bij conversie naar andere databases, gaan de ongeldige polygons voor problemen zorgen. In overleg is daarom besloten alle records met problematische polygons (144 in totaal, de NULL geometrie niet meegerekend, dat waren er vele honderden) uit de NWB dataset te verwijderen (zie Bijlage B3.06). Tevens is voor alle geometrie objecten de SRS (Spatial Reference System) op 28992 gezet, dit is de EPSG code voor de huidige versie van het RD (Rijksdriehoeks-coördinaten; in de oorspronkelijke NWB tabellen staat deze overal op NULL).

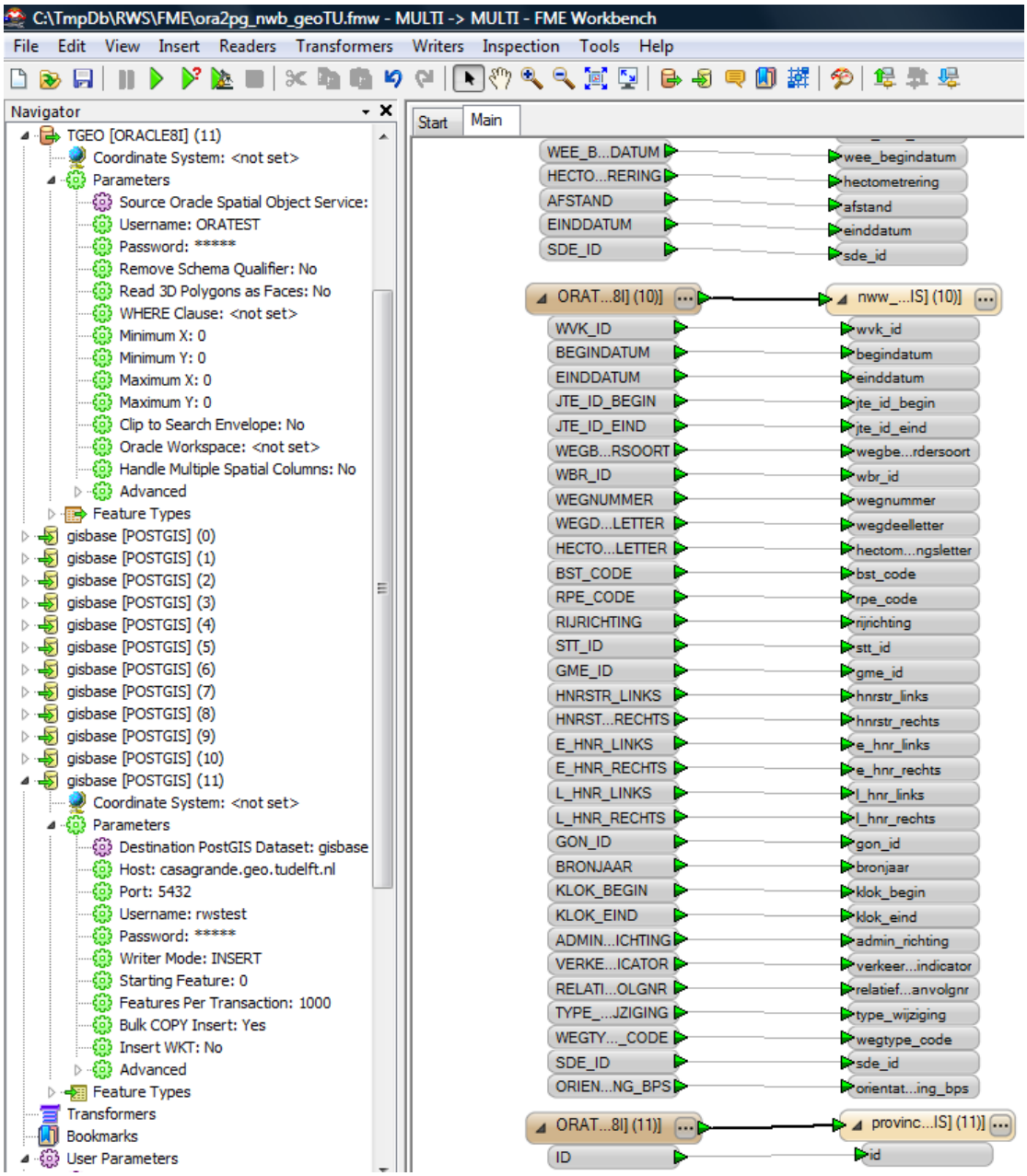
Na verwijdering van de incorrecte polygons was volgens Oracle alle ruimtelijke data in de NWB dataset correct. Bij de latere conversie naar de PostGIS database bleek echter dat dit niet precies hetzelfde betekent in verschillende databases. Tijdens de conversie ging PostGIS protesteren over het feit dat een aantal tabellen met 2D punten (alle geometrie in NWB is 2D) ook punten met X, Y én Z-coordinaten bevat (dus in één tabel met 2D geometrie zitten punten met X, Y coordinaten, én punten met X, Y, Z coördinaten). Klaarblijkelijk is dit een aspect van data validiteit die bij PostGIS tijdens het laden wordt gecontroleerd, in Oracle wordt het noch bij het laden, noch bij de validatie functies meegenomen (Oracle negeert domweg de Z-coordinaat bij 2D punten). Alvorens verder te werken met de data zijn in de initiële Oracle database alle Z-coördinaten verwijderd (op NULL gezet).

Conversie naar andere databases

Aangezien de bedoeling is met verschillende databases te testen moet op een of andere manier de data in deze databases geladen worden. Er bestaan in principe een heleboel manieren om gegevens van de ene naar de andere database te krijgen. Het is onder andere mogelijk de data weg te schrijven naar een ‘simpel’ formaat (bijv. ‘Comma Separated Values’ voor alfanumerieke attributen en ‘Well Known Text’ voor ruimtelijke attributen) en deze gegevens dan in de diverse databases in te lezen. Hier is niet voor gekozen omdat de inschatting was dat dit, gezien de beschikbare tijd voor het onderzoek, behoorlijk veel tijd zou vergen. Uiteindelijk is gekozen voor het gebruik van FME voor het converteren van de data tussen de databases. Dit is gebruikersvriendelijke, gespecialiseerde sofware voor het omzetten van talloze data

formaten, waaronder ook de drie te gebruiken databases. Tevens is het beschikbaar binnen Rijkswaterstaat.

In FME Workbench, een ‘grafische’ programmeeromgeving, zijn conversies gedefinieerd voor het omzetten van de data van de ene in de andere database: van Oracle -> PostGIS, van PostGIS -> Oracle, en van Oracle -> SQL Server. De tweede conversie, van PostGIS -> Oracle, lijkt misschien wat vreemd want de data was al vanaf het begin in Oracle beschikbaar. Deze conversie is toegevoegd om er voor te zorgen dat alle data waarmee getest wordt op dezelfde wijze geladen is, via FME. Dit om te voorkomen dat Oracle wellicht een ongemerkt voordeel zou hebben doordat de data op een Oracle-specifieke manier aangeleverd is.



Figuur 1 Een stukje FME Workbench voor de Oracle -> PostGIS conversie.

In dit onderzoek is FME een goed en efficient te gebruiken conversie tool gebleken, maar helemaal probleemloos was het gebruik toch niet. De diverse databases hebben vergelijkbare, maar soms in detail afwijkende datatypen. In niet alle gevallen gaat FME daar helemaal goed mee om. Zo ondersteunt FME niet het NUMERIC type van PostgreSQL, het equivalent van Oracle's NUMBER type. Bij een default conversie zet FME de NUMBERs van Oracle om naar FLOATs in PostgreSQL, dus van een exacte naar een inexacte representatie met potentieel verlies van informatie als gevolg. Ook met de datum/tijd typen is de conversie niet helemaal rechttoe rechtaan maar dat is niet direkt aan FME te verwijten. Het DATE type in Oracle kent 6 elementen: jaar, maand, dag, uur, minuut, seconde. In PostgreSQL bevat DATE alleen de eerste 3 elementen, hetgeen meer is wat je van DATE zou verwachten. Het datum/tijd type met 6 elementen heet TIMESTAMP in PostgreSQL (met als extra dat daar ook fracties van seconden in zitten, vergelijkbaar met het TIMESTAMP type dat ook Oracle kent).

De mogelijke conversie problemen zijn omzeild door in alle 3 de databases voorafgaand aan de conversie met FME expliciet de 'create table' statements uit te voeren met de voor die database correcte datatypen (zie Bijlagen B3.08, B3.10 en B3.12) . In FME bestaat de optie om het resultaat van de conversie naar een bestaande tabel weg te schrijven, in plaats van het aanmaken van de tabel door FME zelf. De aanpassingen in datatypen tussen de databases zien er als volgt uit:

Van Oracle -> PostgreSQL/PostGIS:
- varchar2 -> varchar
- number -> numeric
- date -> timestamp
- sdo_geometry -> geometry(GEOMETRY,28992)

Van Oracle -> SQL Server:
- varchar2 -> varchar
- number -> numeric
- date -> datetime
- sdo_geometry -> geometry

Het PostGIS geometrie type zoals boven aangegeven definieert 'generieke' geometrie met RD, code 28992, als SRS. Dit is gebruikt bij de NWB tabellen met vlakken. In dit geval is dat noodzakelijk omdat NWB vlakken een mix zijn van POLYGON en MULTIPOLYGON. In PostGIS zijn de tabellen met punten en lijnen resp. als geometry(POINT,28992) en als geometry(LINESTRING,28992) gedefinieerd.

Een concreet voorbeeld hoe het create table statement er uit ziet in de verschillende databases, in dit geval voor de GEB_ZONE_EFEMERIDEN tabel, is:

Oracle:

```
create table geb_zone_efemeriden
(
 zne_znt_code varchar2(2) not null,
 zne_code     varchar2(8) not null,
 begindatum   date        not null,
 einddatum    date,
```

```
 sde_id       number(38)  not null,
 polygoon     sdo_geometry
);
```

PostgreSQL/PostGIS:

```
create table geb_zone_efemeriden
(
 zne_znt_code varchar(2)  not null,
 zne_code     varchar(8)  not null,
 begindatum   timestamp   not null,
 einddatum    timestamp,
 sde_id       numeric(38) not null,
 polygoon     geometry(GEOMETRY,28992)
);
```

SQL Server:

```
create table geb_zone_efemeriden
(
 zne_znt_code varchar(2)  not null,
 zne_code     varchar(8)  not null,
 begindatum   datetime    not null,
 einddatum    datetime,
 sde_id       numeric(38) not null,
 polygoon     geometry
);
```

Beschrijving gebruikte data

De complete NWB dataset bevat meer dan 60 tabellen die tijdens de onderzoeksfase van de conversies van/naar Oracle en PostGIS allemaal geladen zijn omdat op dat moment nog niet bekend was welke vragen er in de testfase gesteld zouden gaan worden (en dus welke tabellen daarvoor nodig zijn). Vanwege de handmatige aanpassingen in FME van alle ‘incorrecte’ datatypen heeft dit ook relatief veel tijd gekost. Uiteindelijk bleek echter dat voor de test queries slechts een zeer beperkt aantal NWB tabellen gebruikt is, niet meer dan 4.

Binnen de NWB dataset vormen de wegen, qua omvang, het overgrote deel. Hoewel het niet het doel is van dit onderzoek om echt op ‘performance’ te testen, is het toch wel nuttig daar enige indruk van te krijgen. Daarvoor zijn tabellen met een zekere omvang nodig en de tabellen die betrekking hebben op vaarwegen en spoorwegen zijn zodanig klein (allemaal minder dan 10,000 records, behalve een enkele tabel met spoorhectopunten van 40,000 records) dat ze t.a.v. performance nauwelijks informatie zullen opleveren. Ook bevatten ze in essentie hetzelfde type data als de tabellen voor de wegen, dus ook op dit punt leveren ze geen aanvullende informatie. De meeste wegen tabellen bevatten uitsluitend alfanumerieke data, er is er één met punten (NWW_HECTOPUNTEN) en één met lijnen (NWW_WEGVAK_EFEMERIDEN). Er is over gediscussieerd of het nut had binnen dit onderzoek met punten te testen, uiteindelijk is er voor gekozen dat niet te doen. De aanname is dat punten voor een ruimtelijke database relatief makkelijke objecten zijn. Als er wordt getest met het opvragen van lijnen (en vlakken) dan kun je er van uit gaan dat het opvragen van punten minstens zo goed en snel gaat (en waarschijnlijk sneller).

Naast de verschillende soorten wegen zijn er binnen de NWB dataset ook allerlei gebiedsindelingen beschikbaar, bijv. gemeenten, waterschappen, diverse soorten Rijkswaterstaat eenheden, etc. Van deze gebieden zijn de vlakken beschikbaar, met name de gemeenten zijn gebruikt om bij de test queries ook iets met vlakken te doen. Om tevens wat grotere en complexere vlakken te hebben zijn de gemeenten geaggregeerd tot provincies, en tot een vlak met de grens van Nederland. Deze aggregatie is uitgevoerd op basis van een binnen Rijkswaterstaat beschikbaar Oracle PL/SQL script dat kleinere gebiedsindelingen kan aggregeren tot grotere eenheden (zie Bijlagen B3.14 en B3.15). PostgreSQL heeft een programmeertaal, PL/pgSQL, die veel lijkt op Oracle's PL/SQL en waarmee hetzelfde op een vergelijkbare manier bereikt kan worden. De vlakken met de provincies en Nederland zijn opgeslagen in een tabel (PROVINCIES) die normaliter geen deel uitmaakt van het NWB.

Uiteindelijk zijn er 5 tabellen gebruikt voor het uitvoeren van de test queries, 4 NWB tabellen en de 'afgeleide' PROVINCIES tabel. Deze tabellen, hier weergegeven in de SQL Server variant, zijn:

```
create table gga_gemeenten
(
 id                numeric(4)  not null,
 naam              varchar(24) not null,
 gme_id_in_vln     numeric(3),
 einddatum_dialoog datetime
);

create table gga_gemeente_efemeriden
(
 gme_id     numeric(4)  not null,
 begindatum datetime    not null,
 einddatum  datetime,
 pve_code   varchar(2)  not null,
 sde_id     numeric(38) not null,
 polygoon   geometry
);

create table gga_straten
(
 id             numeric(8)  not null,
 gme_id         numeric(4)  not null,
 wps_id         numeric(4)  not null,
 stt_type       varchar(1)  not null,
 naam           varchar(29) not null,
 naam_officieel varchar(43),
 naam_ptt       varchar(17),
 naamafkorting  varchar(5)
);

create table nww_wegvak_efemeriden
(
 wvk_id                numeric(10) not null,
 begindatum            datetime    not null,
 einddatum             datetime,
 jte_id_begin          numeric(10) not null,
 jte_id_eind           numeric(10) not null,
 wegbeheerdersoort     varchar(1)  not null,
 wbr_id                numeric(4),
 wegnummer             varchar(5),
```

```
 wegdeelletter          varchar(1),
 hectometreringsletter  varchar(1),
 bst_code               varchar(3),
 rpe_code               varchar(2),
 rijrichting            varchar(1),
 stt_id                 numeric(8),
 gme_id                 numeric(4),
 hnrstr_links           varchar(1),
 hnrstr_rechts          varchar(1),
 e_hnr_links            numeric(5),
 e_hnr_rechts           numeric(5),
 l_hnr_links            numeric(5),
 l_hnr_rechts           numeric(5),
 gon_id                 numeric(2)  not null,
 bronjaar               numeric(4),
 klok_begin             numeric(2)  not null,
 klok_eind              numeric(2)  not null,
 admin_richting         varchar(1),
 verkeersbaan_indicator varchar(1),
 relatief_baanvolgnr    numeric(1),
 type_wijziging         varchar(1),
 wegtype_code           varchar(2),
 sde_id                 numeric(38) not null,
 lijn                   geometry,
 orientatierichting_bps varchar(1)
);

create table provincies
(
 id       numeric(10) not null,
 polygoon geometry
);
```

Van de NWB tabellen is NWW_WEGVAK_EFEMERIDEN duidelijk de grootste, deze speelt een belangrijke rol bij de test queries.

		omvang in MB		
tabel	#rows	PostGIS	Oracle	SQL Server
gga_gemeenten	932	0.05	0.20	0.04
gga_gemeente_efemeriden	7723	40.00	80.00	83.00
gga_straten	355891	31.00	23.00	25.00
nww_wegvak_efemeriden	2923714	741.00	821.00	797.00
provincies	13	0.47	1.20	1.23

Tabel 2 Omvang van tabellen gebruikt bij de test queries (omvang > 10 MB afgerond op hele MB).

Het kan behoorlijk lastig zijn de correcte omvang van een tabel te bepalen. Voor zover de data ‘inline’ (in database ‘blocks’, ofwel ‘pages’) opgeslagen wordt is het geen probleem dit te achterhalen. Maar ruimtelijke data bestaat voor een deel uit ‘complexe’ objecten, bijv. een polygon met een groot aantal punten, die niet meer in een page passen. De betreffende geometrie komt dan terecht in een BLOB (Binary Large Object) die op een andere manier wordt opgeslagen dan ‘normale’ data. Deze BLOBS worden in grotere eenheden gealloceerd, de gealloceerde ruimte is dan zichtbaar maar niet hoeveel er werkelijk gebruikt wordt. Een geschatte of redelijk grof afgeronde grootte is dan wat rest om de omvang aan te geven. De getallen in

bovenstaande tabel moeten derhalve met een zekere marge (plus of min 20%) geïnterpreteerd worden.

Metadata, indexen en statistieken

Een prettige bijkomstigheid van het werken met ruimtelijke data in SQL Server en de huidige versie (2) van PostGIS is dat dit type data niet apart in de metadata geregistreerd behoeft te worden. Is een ruimtelijke kolom eenmaal gedefinieerd dan is deze ook automatisch bekend in de 'system catalog'. In Oracle is het nog steeds zo dat database gebruikers zelf moeten zorgen voor het 'registreren' van ruimtelijke kolommen in de USER_SDO_GEOM_METADATA view (zie Bijlage B3.07). Deze laatste wordt ook niet automatisch gesynchroniseerd met de tabel.

Indexen kunnen, zeker bij grotere tabellen, enorm helpen om de responstijd van queries terug te brengen. Teneinde de query optimizer de gelegenheid te bieden desgewenst indexen te gebruiken zijn voor alle bij de test queries betrokken tabellen één of meer indexen aangemaakt (zie Bijlagen B3.09, B3.11 en B3.13 voor de scripts waarin de indexen worden aangemaakt). Alle tabellen hebben in ieder geval een primary key gedefinieerd op de eerste of de eerste twee kolommen (dit laatste geldt voor de beide 'efemeriden' tabellen). Als voorbeeld de statements voor het toevoegen van een primary key op NWW_WEGVAK_EFEMERIDEN:

PostGIS:
```
alter table nww_wegvak_efemeriden add primary key
(wvk_id,begindatum);
```

Oracle:
```
alter table nww_wegvak_efemeriden add primary key
(wvk_id,begindatum);
```

SQL Server:
```
alter table nww_wegvak_efemeriden add constraint
nww_wegvak_efemeriden_pkey primary key clustered (wvk_id,begindatum);
```

In SQL Server moet er expliciet een 'clustered' primary key worden aangemaakt op een tabel met een ruimtelijke kolom als je daarna op die kolom een ruimtelijke index wilt aanmaken.

Op de kolommen die gebruikt worden in de WHERE clause van de test queries, voor zover geen onderdeel van de primary key, zijn normale indexen gezet. En uiteraard zijn er ruimtelijke indexen aangemaakt voor alle kolommen met ruimtelijke data, als voorbeeld de commando's voor opnieuw de NWW_WEGVAK_EFEMERIDEN tabel:

PostGIS:
```
create index nww_wegvak_efemeriden_sidx on nww_wegvak_efemeriden
using gist (lijn);
```

Oracle:
```
create index nww_wegvak_efemeriden_sidx on nww_wegvak_efemeriden
(lijn) indextype is mdsys.spatial_index parameters
('layer_gtype=line');
```

SQL Server:

```
create spatial index nww_wegvak_efemeriden_sidx on
nww_wegvak_efemeriden (lijn)
with (bounding_box=(-50000,250000,300000,650000));
```

In tabel 3 zijn de statistieken van enkele geselecteerde indexen op de NWW_WEGVAK_EFEMERIDEN tabel weergegeven om zo een (enigszins beperkte) vergelijking tussen de databases mogelijk te maken op dit punt.

	"elapsed" tijd in sec			omvang in MB		
	PostGIS	Oracle	SQL Server	PostGIS	Oracle	SQL Server
nww_wegvak_efemeriden						
primary key	23	34	12	113	80	769
stt_id	17	6	1	63	51	73
spatial index	34	225	126	148	254	644

Tabel 3. Benodigde tijd voor genereren en omvang van enkele indexen.

Meer nog dan bij de statistieken over de tabellen zijn bij het bovenstaande overzicht een aantal kwalificaties nodig om de informatie op de juiste manier te interpreteren. De weergegeven tijden zijn niet tot stand gekomen onder strikt gecontroleerde omstandigheden zoals dat bij de test queries wel het geval is. Hier weergegeven is het resultaat van de laatste ‘run’ van het aanmaken van de indexen. Mogelijk is dat de server op dat moment ook met andere taken bezig was, hoewel de invloed daarvan gezien de (over)capaciteit van het gebruikte systeem gering zal zijn.

Heel verschillend is wel de manier waarop de diverse databases de systeem resources gebruiken. SQL server ‘pakt’ alle beschikbare CPU’s (24 in de testserver) en bij het aanmaken van een index springt het CPU gebruik direkt naar 100 %. Aan memory wordt tussen 2 en 2.5 GB gebruikt (op basis van de standaard, ‘out-of-the-box’, configuratie zoals bij SQL Server toegepast). PostgreSQL en Oracle daarentegen gebruiken niet meer dan 1 CPU en daar komt het CPU gebruik voor het totale systeem dus niet boven de 4 %. De hoeveelheid memory die PostgreSQL en Oracle (kunnen) gebruiken wordt grotendeels bepaald door configuratie parameters die bij het opstarten van de database worden meegegeven. Bij de gebruikte parameters (zie par 3.1) resulteert dit voor beide databases in een actief (‘resident’) gebruik van tussen 300 en 350 MB aan memory bij het aanmaken van de indexen (waarbij het gealloceerde memory dan een factor 1.5, PostgreSQL, of een factor 3, Oracle, hoger ligt). Vanwege het parallel gebruik van de CPU’s is bij SQL Server de ‘CPU-tijd’ bij het aanmaken van de indexen een factor 10 tot 15 hoger dan de ‘elapsed’ tijd.

Bovenstaande wil niet zeggen dat andere databases niet kunnen profiteren van de aanwezigheid van meerdere CPU’s. Voor de standaard (‘produktie’) versie van PostgreSQL/PostGIS staat benutten van ‘parallel processing’ op de ‘to-do’ lijst, er zijn wel prototypes en gerelateerde produkten waarmee parallel gewerkt kan worden (zie http://blog.opengeo.org/2010/10/05/parallel-postgis/). In Oracle is de default instelling dat er serieel gewerkt wordt, maar met opties als

```
create index ... parallel ...
create table ... parallel ...
select /*+ parallel(...) */ ...
```

kan men parallel werken activeren. Overigens zijn er dan wel restricties aan wat er ‘geparallelliseerd’ kan worden. Sommige operaties op BLOBs en ‘user-defined datatypes’, zoals SDO_GEOMETRY in Oracle, zijn niet te paralleliseren, of alleen als ze in gepartitioneerde tabellen zitten.

Wat de omvang van de indexen betreft, de primary key op NWW_WEGVAK_EFEMERIDEN is een combinatie van een numerieke en een datetime kolom die in SQL Server onevenredig groot lijkt te zijn. Ook de ruimtelijke index van SQL Server is relatief groot, dat kan te maken hebben met het soort index van die database. Oracle en PostGIS gebruiken als spatial index een R-tree die ‘zelf-tunend’ is en in de meeste gevallen goed presteert. De hiërarchische grid index van SQL Server kan in principe een zeer effectieve toegangsmethode opleveren, alleen is het lastig de optimale setup daarvan te bepalen. In dit geval is de ‘auto_grid’ optie gebruikt met een (wellicht te) ruime bounding box waardoor de index onnodig groot en daardoor ook minder efficiënt wordt. Een betere index is ongetwijfeld te maken, maar daar is dan ook meer werk mee gemoeid.

De weergegeven omvang van de ruimtelijke index in Oracle is groter dan eigenlijk nodig is. In dit geval ‘schat’ de database de beschikbare versus de benodigde ruimte in de database blocks verkeerd in met als gevolg dat grote aantallen ‘chained rows’ ontstaan die leiden tot inefficiënt ruimtegebruik (en ook met nadelige gevolgen bij het gebruik van de index). Dit is te verhelpen met het correct instellen van de ongedocumenteerde ‘sdo_fanout’ parameter bij het aanmaken van de index, maar uiteraard kost dat moeite en tijd.

Als laatste stap in de voorbereiding op het eigenlijke testen is er voor gezorgd dat de statistieken, t.b.v. de query optimizer, in orde zijn. SQL Server genereert statistieken bij het aanmaken/vullen van tabellen, deze bleken nog up-to-date aangezien er verder ook niets aan de geladen tabellen veranderd is. Bij aanpassingen bepaalt de database automatisch wanneer het nodig is de statistieken te vernieuwen, je kunt dit desgewenst ook handmatig doen.

Bij de Oracle en PostgreSQL databases worden er enige statistieken gegenereerd bij het aanmaken/vullen van tabellen, maar om echt actuele statistieken voor de query optimizer (query planner) te hebben zijn er aanvullende acties nodig. Die aanvullende acties kan men overlaten aan achtergrond processen die beide databases hebben (mits geconfigureerd) om de statistieken up-to-date te houden. In PostgreSQL is dit ‘autovacuum’, in Oracle ‘auto optimizer stats collection’. Afhankelijk van de omvang van de database en de mate waarin tabellen wijzigen (en de configuratie van het automatisch verzamelen van statistieken) kan het enige/een behoorlijke tijd duren voor de statistieken zijn bijgewerkt. Om er zeker van te zijn dat de statistieken actueel zijn, is dit in beide databases met commando’s ‘geforceerd’. In PostgreSQL komt dit neer op het geven van een VACUUM statement met de FULL en ANALYZE opties. In Oracle is het gedaan. met de DBMS_STATS.GATHER_DATABASE_STATS procedure met opties die er voor zorgen dat van alle objecten in de database de actuele statistieken worden verzameld.

3.3 Opzet en resultaten test queries

Opzet queries

Nadat besloten was alleen de NWB dataset te gebruiken voor het testen is er in overleg een set van representatieve queries samengesteld. Representatief in de zin dat de meest voorkomende manieren van bevragen van het NWB daarin terug moet komen. Het meest gebruikt, dit geldt in het algemeen voor het gebruik van ruimtelijke databases, is de rechthoekige 'window' query: het opvragen van een ruimtelijke subset van de complete dataset op basis van een opgegeven rechthoek. Dit type query is bij de test queries met rechthoeken in verschillende groottes opgenomen. Daarnaast zijn er ook queries die objecten opvragen op basis van een opgegeven vlak (polygon), een type query dat 'moeilijker' wordt voor de database naarmate er meer punten nodig zijn voor de vastlegging van het vlak. De opgevraagde objecten zijn voornamelijk wegvakken, dus lijnen waar het de geometrie betreft.

Met de bovengenoemde soorten queries is reeds een groot deel van het feitelijke gebruik van het NWB gedekt. Maar om een idee te krijgen van de toepasbaarheid van alternatieve databases bij gebruik van andersoortige datasets en voor andere toepassingen zijn er aanvullende queries toegevoegd. Er worden bijv. ook vlakken opgevraagd op basis van een lijn, en vlakken op basis van een vlak. Naast opvragen zitten er bij de queries ook enkele basale geometrische berekeningen: bepalen van de lengte van lijnen en het oppervlak van vlakken. Verder bevatten vrijwel alle queries een combinatie van administratieve en ruimtelijke voorwaarden, omdat dit in de Rijkswaterstaat praktijk ook het meest voorkomt. De uitzondering hierop zijn de eerste twee 'basis' queries die alleen 'administratieve' (alfanumerieke) voorwaarden gebruiken. Om tenslotte ook nog iets meer geavanceerde ruimtelijke functionaliteit te testen is er een query opgenomen die selecteert op, een bepaald aantal, dichtstbijzijnde objecten ('nearest neighbour') en een query waarbij bufferzones rond vlakken worden berekend.

De meeste queries selecteren een subset van de totale NWB dataset. Om meer inzicht te krijgen in de aspecten waar de te vergelijken databases beter/minder presteren zijn deze queries in twee varianten uitgevoerd.
Als

```
select count(*) from ... where ...
```

en als

```
create table NEWTABLE as select ATTRIB1,... from ... where ...
```

(de syntax van de laatste in SQL Server is:

```
select ATTRIB1,... into NEWTABLE from ... where ...
```

).

De '`select count(*)`' variant test vooral het vermogen van de database om een fatsoenlijk query plan op te stellen. Waar het de alfanumerieke attributen betreft kan worden volstaan met het gebruiken van de index, er is een (zeer) beperkte hoeveelheid I/O nodig. Dit geldt in mindere mate voor de ruimtelijke voorwaarden. De opgenomen queries vragen altijd of het 'echte' ruimtelijke object interactie heeft met de 'query geometry', het gaat niet alleen om 'bounding box' overlap. De ruimtelijke index, bij de R-tree (dus PostGIS en Oracle), bevat alleen de bounding box, is de volledige geometrie nodig dan moet die opgehaald worden van disk (of uit de cache).

De ‘`create table as select`’ variant, uitgevoerd voor de eerste 7 queries, heeft (veel) meer I/O tot gevolg. In ieder geval moeten bij alle queries de data van de geselecteerde rows opgehaald worden, en het resultaat moet weggeschreven worden naar een tabel. Bij enkele queries, met name 1 en 5, is de resultaat-tabel van een behoorlijke omvang (bij query 1 voor Oracle 184 MB), dit is dan vanzelfsprekend terug te vinden in de responstijd.

Test queries

De 14 test queries, voornamelijk in de PostgreSQL/PostGIS syntax met de syntax van de andere databases waar relevant, zijn onderstaand weergegeven (zie ook Bijlagen B3.16, B3.21 en B3.26). In alle queries komen ‘variabelen’ voor die een bepaalde ‘peildatum’ of een ‘query geometrie’ voorstellen, de syntax voor parameters in de diverse databases is:

PostgreSQL: `:peildatum`
Oracle: `&peildatum`
SQL Server: `@peildatum`

```
--
-- 01 Basis, administratieve, query
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
;
--
-- 02 Met beperking op wegbeheerdersoort
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
--
-- En met ruimtelijke beperking
-- (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
;
```

Oracle: `...and sdo_anyinteract(wee.lijn, &small_query_window) = 'TRUE'`
SQL Server: `... and wee.lijn.STIntersects(@small_query_window) = 1`

```
--
-- 04 Medium window: 50 x 50 km, groot deel van Zeeland
--
```

```
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :medium_query_window)
;
--
-- 05 Large window: 120 x 120 km in midden van Nederland
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :large_query_window)
;
--
-- 06 Met ruimtelijke beperking
--    (polygon met grens van provincie Limburg: 7946 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :prov_limburg
    and  ST_Intersects(wee.lijn, p.polygoon)
;
```

Oracle: `... and sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE'`
SQL Server: `... and wee.lijn.STIntersects(p.polygoon) = 1`

```
--
-- 07 Met ruimtelijke beperking
--    (polygon met grens van gemeente Delft: 252 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
       gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is null)
    and  gme.naam = :gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
    and  ST_Intersects(wee.lijn, gee.polygoon)
;
--
-- 08 Met berekening van lengte van wegvakken
--
select wee.wvk_id, wee.begindatum, ST_Length(wee.lijn)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
    and  ST_Length(wee.lijn) > 100
  order by wee.wvk_id
;
```

Oracle: `... and sdo_geom.sdo_length(wee.lijn, &tol) > 100`
SQL Server: `... and wee.lijn.STLength() > 100`

```
--
-- 09 Selectie van gemeente polygons op basis van
--    provincie polygon (ZH: 2569 punten)
--
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme, provincies p
  where  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is NULL)
    and  p.id = :prov_zh
    and  ST_Intersects(gee.polygoon, p.polygoon)
;
--
-- 10 Selectie van gemeente polygons op basis van
--    lijn dwars door Nederland
--
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme
  where  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is NULL)
    and  ST_Intersects(gee.polygoon,ST_GeomFromText(:nl_line, :srid))
;
```

Query 11, het selecteren van dichtstbijzijnde lijnen, betreft een operatie die niet als zodanig in de OGC en ISO standaarden voorkomt. Het gevolg is dat deze query er in de verschillende databases behoorlijk anders uitziet, bij de overige queries is dat minder het geval.

In PostGIS is het de '`<#>`' operator die er voor zorgt dat de index gebruikt wordt om nabijgelegen objecten te zoeken, in Oracle is dat de `sdo_nn` 'operator'. In beide databases zorgt de combinatie met andere voorwaarden in de WHERE clause voor complicaties: je weet niet zeker of de ruimtelijke voorwaarde eerst getest wordt en dan de andere voorwaarden, of andersom. De query optimizer kiest voor het eerste, in de voorlopige resultset na deze eerste stap moeten dan wel voldoende rows zitten zodat na toepassing van de overige voorwaarden nog het gevraagde aantal rows, 100 in dit geval, overblijft. Door op te geven dat er in de eerste stap 500 (nabijgelegen) objecten geselecteerd moeten worden blijven er aan het eind voldoende over om aan alle voorwaarden te voldoen.

In SQL Server is het een specifieke combinatie van TOP en ORDER BY, samen met het gebruik van de STDistance method in de WHERE en (als eerste) in de ORDER BY clauses die maakt dat de database de ruimtelijke index kan gebruiken om nabijgelegen objecten te zoeken.

```
--
-- 11 Selecteren van :nn_count dichtstbijzijnde lijnen
--    (wegvakken) vanaf een punt (PostGIS uitvoering)
--
select wvk_id, naam, afstand from
 (select wee.wvk_id, stt.naam, ST_Distance(wee.lijn, :rd_origin) afstand,
         wee.lijn <#> :rd_origin bb_afstand
    from nww_wegvak_efemeriden wee, gga_straten stt
    where  wee.stt_id = stt.id
```

```
      and  wee.begindatum <= :peildatum
      and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    order by bb_afstand limit 500
 ) as foo
order by afstand, naam, wvk_id limit :nn_count
;
--
-- 11 Selecteren van &nn_count dichtstbijzijnde lijnen
--    (wegvakken) vanaf een punt (Oracle uitvoering)
--
select * from
 (select wee.wvk_id, stt.naam, sdo_nn_distance(1) afstand
    from nww_wegvak_efemeriden wee, gga_straten stt
    where  wee.stt_id = stt.id
      and  wee.begindatum <= &peildatum
      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
      and  sdo_nn(wee.lijn, &rd_origin, 'sdo_num_res=500', 1) = 'TRUE'
    order by afstand, naam, wvk_id)
where rownum <= &nn_count
;
--
-- 11 Selecteren van :nn_count dichtstbijzijnde lijnen
--    (wegvakken) vanaf een punt (SQl Server uitvoering)
--
select top(@nn_count) wee.wvk_id, stt.naam,
                      wee.lijn.STDistance(@rd_origin) afstand
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is null)
    and  wee.lijn.STDistance(@rd_origin) is not NULL
  order by wee.lijn.STDistance(@rd_origin), naam, wvk_id
;

--
-- 12 Genereren van buffer rond een aantal gemeenten:
--    Amsterdam, Arnhem, Eindhoven, Groningen, Rotterdam, Utrecht
--
create table pg_result12 as select gme.naam,
             ST_Buffer(gee.polygoon, :buf_dist) gem_buffer
  from gga_gemeenten gme, gga_gemeente_efemeriden gee
  where  gme.naam in (:gem_list)
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
;
select naam, ST_Area(gem_buffer) from pg_result12 order by naam;
```

Oracle: `sdo_geom.sdo_buffer(gee.polygoon,&buf_dist,&tol) gem_buffer`
en: `... sdo_geom.sdo_area(gem_buffer, &tol)`
SQL Server: `gee.polygoon.STBuffer(@buf_dist) gem_buffer`
en: `... gem_buffer.STArea() buffer_area`

```
--
-- 13 Met ruimtelijke beperking
--    (groot window waar Nederland geheel in past)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
```

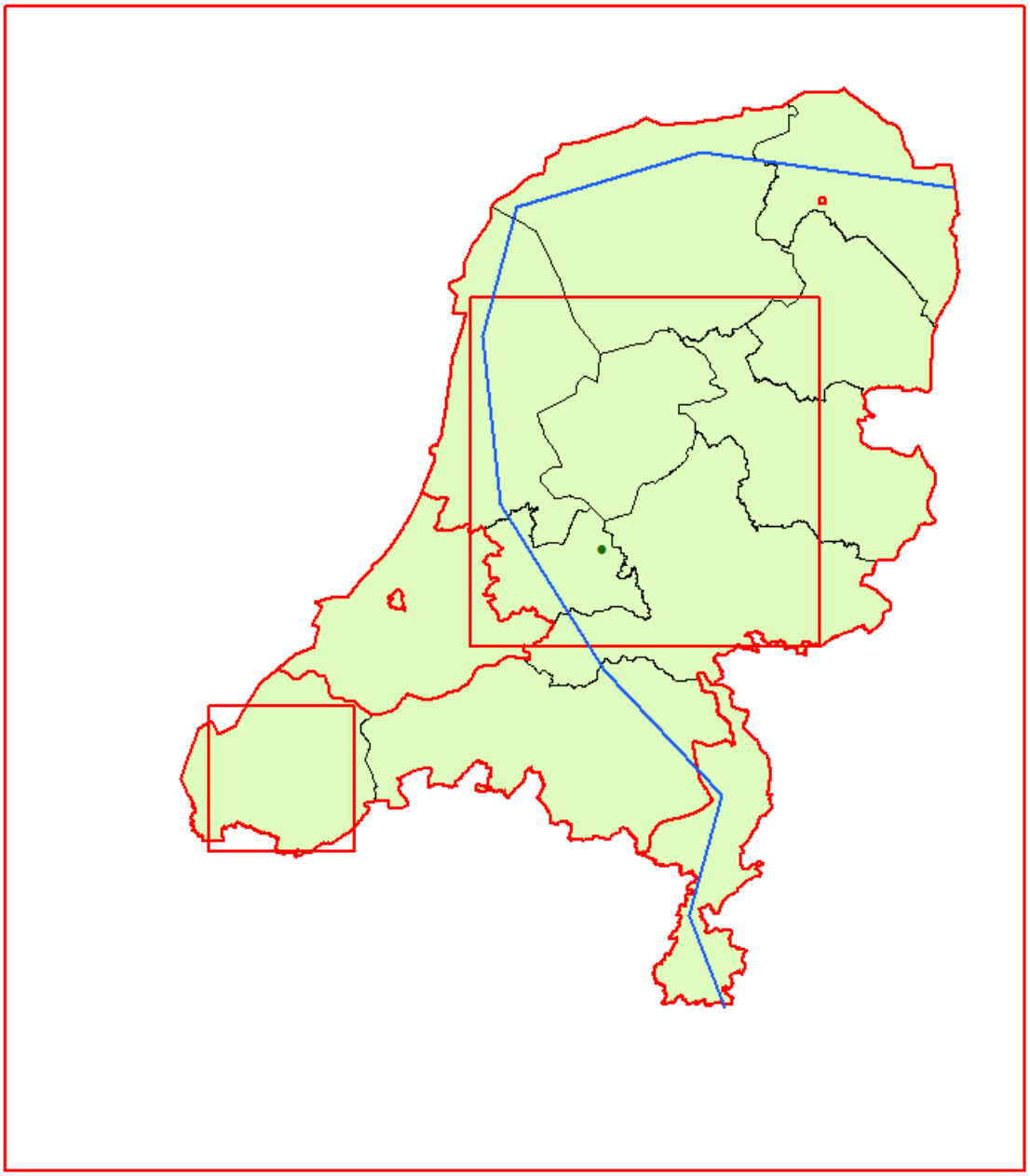
```
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :nl_query_window)
;
--
-- 14 Met ruimtelijke beperking
--    (polygon met grens van Nederland: 18631 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :nederland
    and  ST_Intersects(wee.lijn, p.polygoon)
;
```

Query 13 en 14 zijn ‘vreemde’ queries in de zin dat er ruimtelijke selecties uitgevoerd worden die niets weg selecteren, of andersom gezegd selecteren ze alles. Het resultaat is identiek aan dat van query 01 waar alleen alfanumerieke voorwaarden worden gebruikt. Alleen ‘weet’ de database van te voren niet dat de ruimtelijke selectie eigenlijk overbodig is en gaat het deze selectie dus op de gebruikelijke wijze bepalen. Query 13 is te zien als een opgeschaalde versie van de andere window queries (03, 04 en 05). Op dezelfde wijze is query 14 te beschouwen als een opgeschaalde variant van query 06 en 07, selectie op basis van een polygon. In het geval van query 14 is de polygon wat ‘complexer’, deze bevat de grens van Nederland vastgelegd met 18631 punten.

Van query 1 t/m 7 zijn er ook ‘`create table as select`’ varianten getest. Bij al deze queries worden dezelfde attributen opgevraagd: een alfanumeriek (WVK_ID) en een ruimtelijk attribuut (LIJN) uit de NWW_WEGVAK_EFEMERIDEN tabel, en een alfanumeriek attribuut (ID) uit de GGA_STRATEN tabel (zie Bijlagen B3.18, B3.23 en B3.28). Een enkele query als voorbeeld volstaat om de opzet van dit type queries duidelijk te maken:

```
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen
--
create table pg_result03 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
;
```

Een overzicht van alle geometries die bij de test queries gebruikt zijn voor ruimtelijke selectie is in figuur 2 te vinden.



Figuur 2 De query geometries gebruikt bij de test queries (rood: vlakken, blauw: lijn, donkergroen: punt).

Uitvoering queries

De 14 (select count) resp. 7 (create table as select) queries zijn bij elkaar gezet in twee scripts die vervolgens een aantal keer zijn ‘gerund’. In PostgreSQL wordt hiervoor ‘psql’ gebruikt, in Oracle ‘SQL*Plus’ en in SQL Server ‘sqlcmd’. In dergelijke scripts kunnen SQL commando’s gegeven worden, maar ook zijn er allerlei ‘extensies’ die extra mogelijkheden bieden. Een mogelijkheid waarvan gebruikt gemaakt is, is het definiëren van ‘variabelen’ waardoor de scripts makkelijker consistent te houden zijn en snel met verschillende parameters zijn uit te voeren. De extra mogelijkheden in de diverse omgevingen zijn ongeveer vergelijkbaar, maar de daarbij te gebruiken syntax varieert nogal, zoals te verwachten is.

SQL, voor zover echt gestandaardiseerd, is (vrijwel) gelijk voor de drie databases. De eerste twee (‘administratieve’) queries zijn in alle gevallen identiek. Hoewel er voor ‘spatial’ OGC en ISO standaarden zijn, blijkt dit in de implementatie van de verschillende DBMS leveranciers toch tot allerlei kleinere en grotere verschillen te leiden. Zo zijn bijv. de meeste ruimtelijke operaties in PostGIS en Oracle als ‘functies’ geïmplementeerd, terwijl dit in SQL Server als ‘method’ is, hetgeen resulteert in een andere syntax.

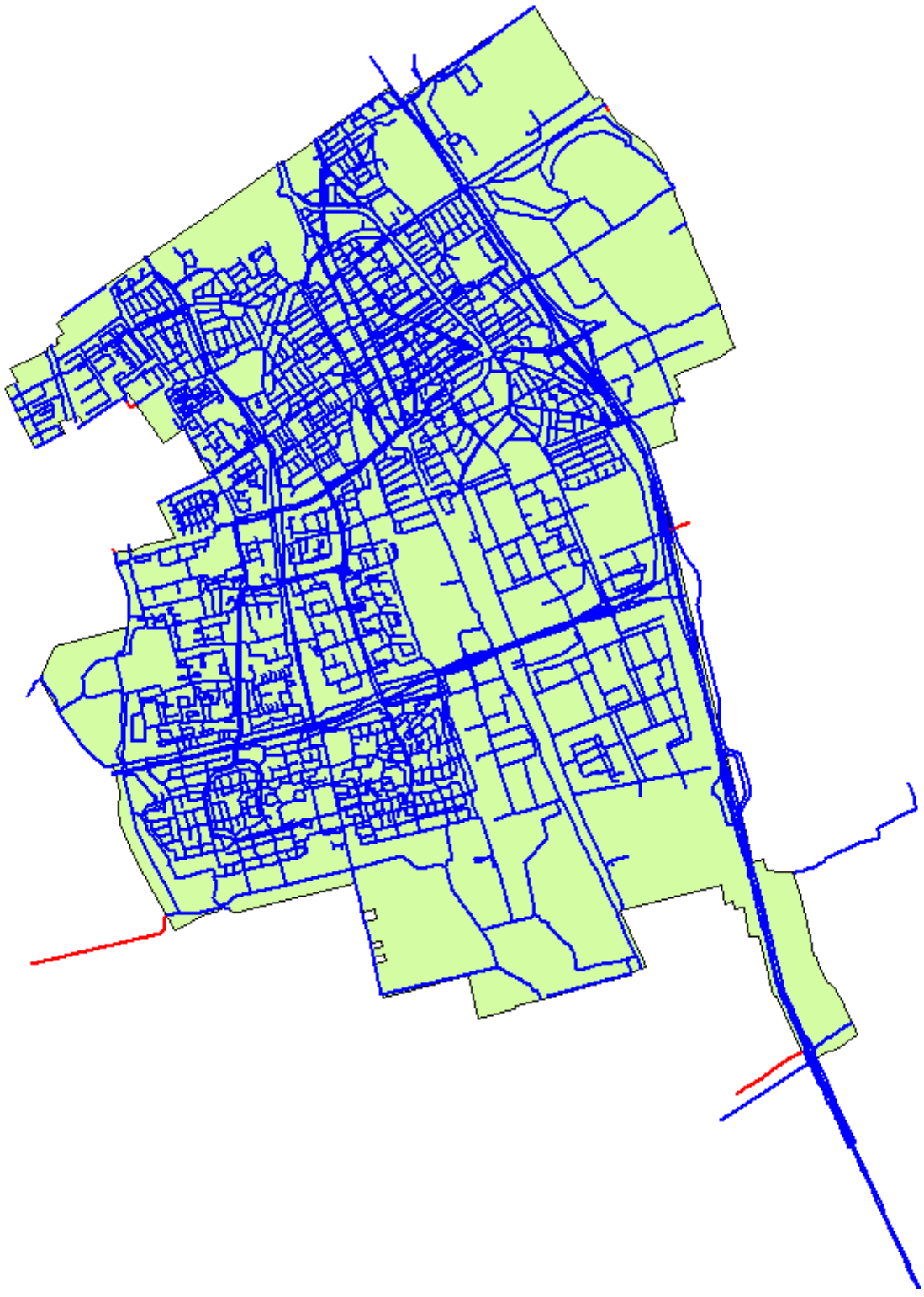
Ter bevordering van de consistentie in de responstijden is altijd voor het runnen van een script de betreffende database gestopt en opnieuw gestart. De server zelf is niet steeds opnieuw gestart. Dit kan betekenen dat op het niveau van het operating system bepaalde voor de queries benodigde data uit de cache komt, en dus niet fysiek van disk gelezen hoeft te worden. De verschillende queries in een script worden binnen een sessie, achter elkaar uitgevoerd. Latere queries zullen in toenemende mate profiteren van data, van indexen en van tabellen, die in de database cache aanwezig zijn. De weergegeven responstijden zijn ‘elapsed’ tijden, en het gebaseerd op 5 runs. Van die 5 runs zijn de snelste en langzaamste verwijderd, van de 3 overblijvende runs is de gemiddelde responstijd berekend. De spreiding in de responstijden van de runs waarmee het gemiddelde berekend is, is relatief klein bij queries die langer duren, bij de kort durende queries kan het oplopen tot 50% verschil (meet ruis) tussen de runs.

Resultaten queries

Het belangrijkste resultaat van de test queries is dat alle geteste databases deze correct uitvoeren. In tabel 5 (blz. 41) zijn de aantallen rows als resultaat van de test queries weergegeven (voor de complete logs met resultaten zie Bijlagen B3.17, B3.22 en B3.27). Een grafische weergave van een correct query resultaat is op de volgende pagina in figuur 3 te vinden.

Figuur 3 Selectie van wegvakken binnen een rechthoek van 2 x 2 km, query 03.

Bij vrijwel alle queries zijn de resultaten hetzelfde bij de verschillende databases. Dit geldt alleen niet bij queries 06 en 07, selectie van wegvakken op basis van een niet-rechthoekig vlak (provincie, resp. gemeente). Bij deze queries vindt Oracle enkele wegvakken meer dan de andere twee databases. De reden hiervoor is het feit dat Oracle bij alle ruimtelijke operaties en berekeningen een tolerantie hanteert, dat is niet het geval bij de andere twee. Voor het NWB is een tolerantie gespecificeerd van 0.5 mm. De extra wegvakken die Oracle selecteert liggen eigenlijk buiten de selectie polygon, maar wel op minder dan 0.5 mm afstand (zie ook figuur 4).



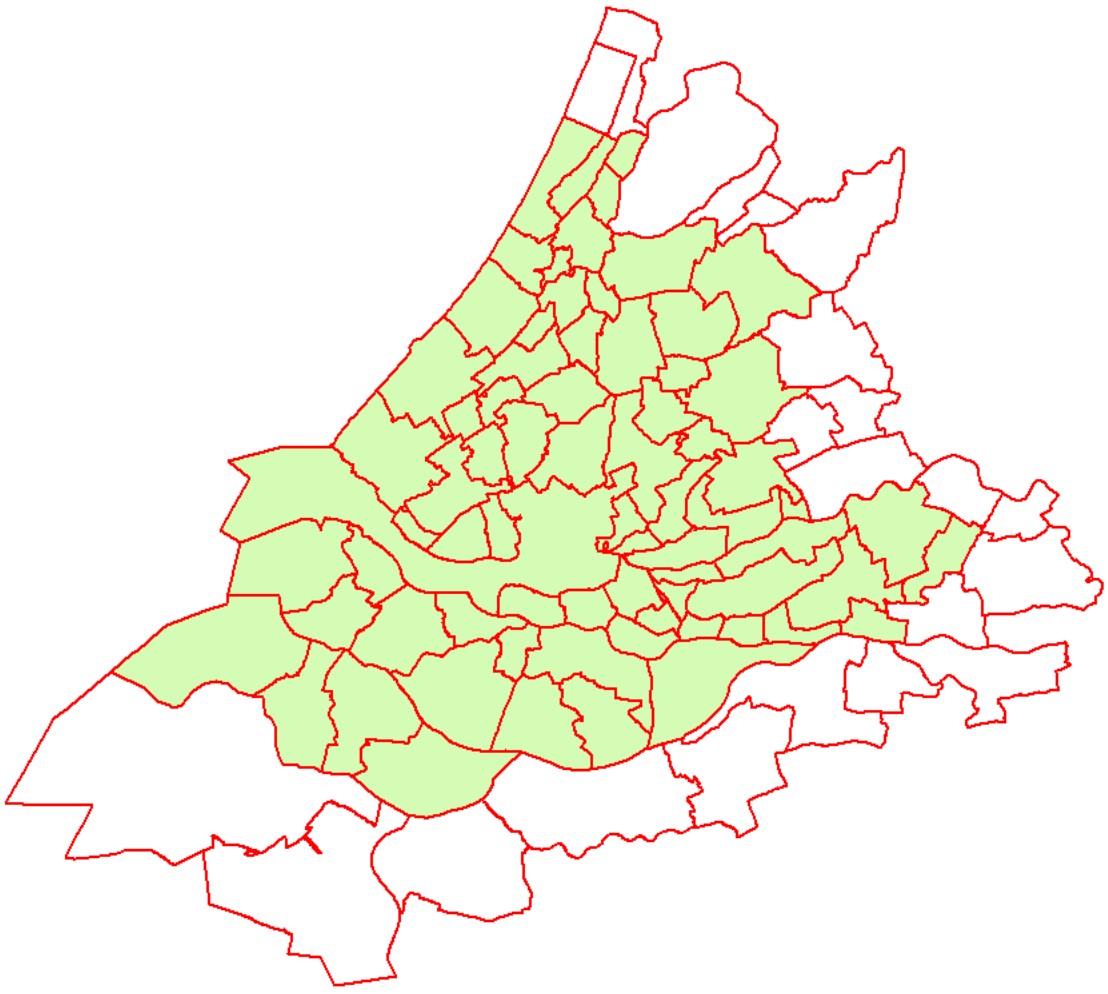
Figuur 4 Selectie van wegvakken binnen de gemeente Delft, query 07
(rood: extra geselecteerde wegvakken in Oracle t.o.v. PostGIS en SQL Server).

Dat er met name bij de gemeente als selectie polygon verschillen optreden, en niet bij de rechthoekige polygons, heeft te maken met het gegeven dat wegvakken aan de rand van een gemeente beginnen/eindigen op de gemeentegrens. De kans dat een wegvak dan net naast een gemeentegrens begint of eindigt is veel groter dan bij de ‘willekeurige’ horizontale of verticale lijnen van een rechthoek polygon.

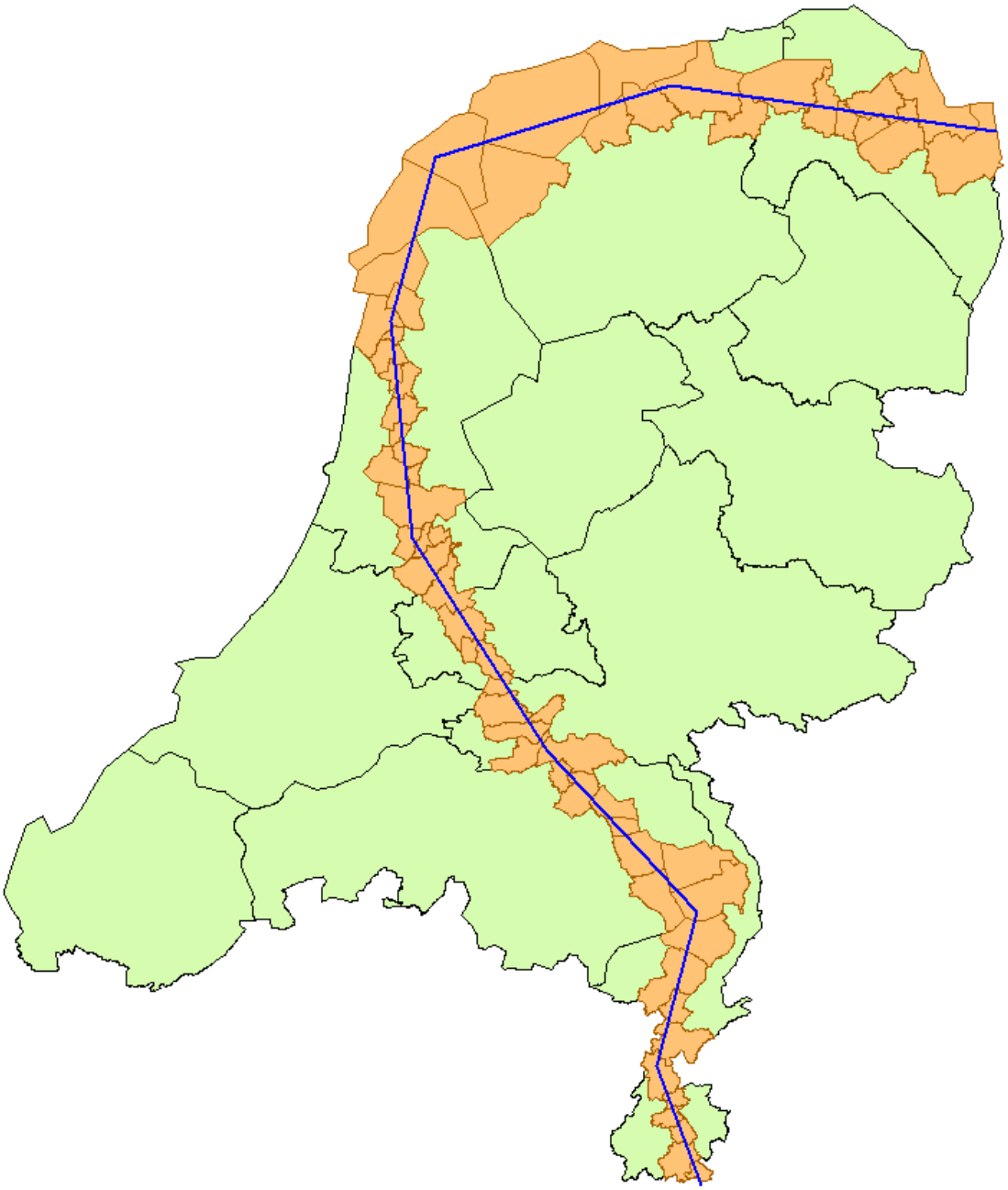
Het resultaat van query 08 met de berekening van de lengte van wegvakken is hetzelfde bij de geteste databases. Het PostgreSQL/PostGIS resultaat is (eerste 10 records weergegeven):

```
  wvk_id   |     begindatum      |     st_length
-----------+---------------------+--------------------
 459565016 | 2007-02-01 00:00:00 | 260.01201434742961
 459566001 | 2002-07-01 00:00:00 |  261.1438804829844
 459566006 | 2002-07-01 00:00:00 | 239.45385672381383
 459566010 | 2002-07-01 00:00:00 |  250.9034397929241
 459567032 | 2010-06-01 00:00:00 | 389.41504502649883
 460564008 | 2002-12-01 00:00:00 | 132.09466302617983
 460564009 | 2007-02-01 00:00:00 | 139.88994137535684
 460564026 | 2002-12-01 00:00:00 | 131.85598204101322
 460564032 | 2002-12-01 00:00:00 | 132.03408650799233
 460564034 | 2007-02-01 00:00:00 | 631.49006020520142
 . . .
```

De meeste test queries selecteren wegvakken omdat die voor het NWB het meest relevant zijn. Query 09 selecteert vlakken (gemeenten) op basis van een vlak (provincie Zuid Holland, figuur 5). Query 10 selecteert vlakken (gemeenten) op basis van een lijn (figuur 6).



Figuur 5 Geselecteerde gemeente polygons op basis van Zuid-Holland polygon, query 09.



Figuur 6 Geselecteerde gemeente polygons op basis van lijn door Nederland, query 10.

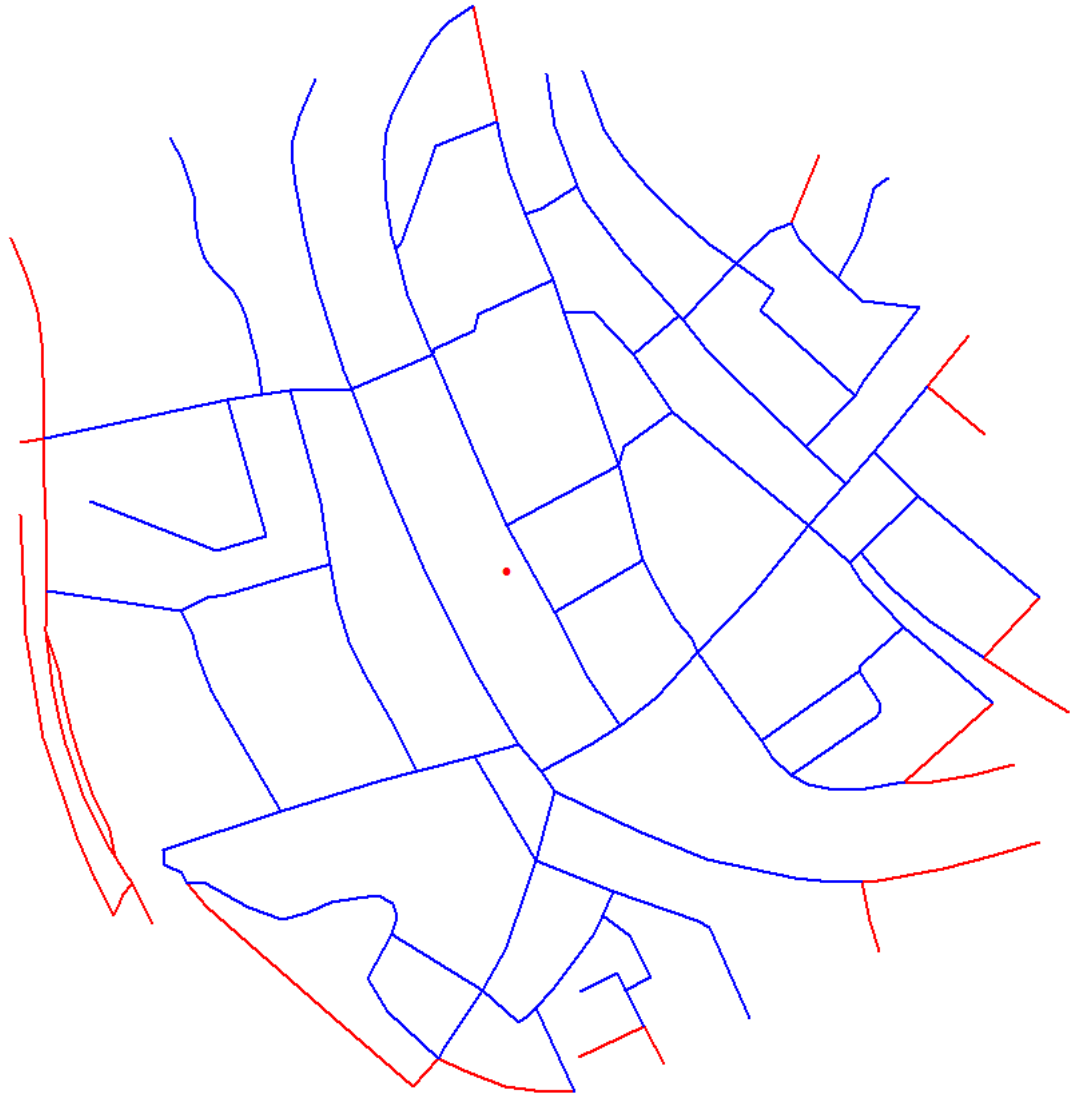
In alle gevallen waar ruimtelijke objecten geselecteerd worden gebeurt dit bij de test queries met de ST_Intersects functie/method (of in Oracle: sdo_anyinteract). Een reden hiervoor is dat dit een van de weinige functies is die gebruik maakt van een ruimtelijke index. In alle drie de databases zijn er nog veel meer ruimtelijke ‘operatoren’, maar de meerderheid daarvan werkt alleen op ‘losse’ ruimtelijke objecten, zonder gebruik van een index. ST_Intersects vindt alle objecten die ‘NOT DISJOINT’ zijn, dus ook alle objecten die (aan de ‘buitenkant’) raken zoals in figuur 5 te zien is. In de geteste databases kan men een “Dimensionally Extended 9 Intersection Model” (DE-9IM) ‘mask’ specificeren (via de ST_Relate functie, of sdo_relate in Oracle) waarbij preciezer kan worden aangegeven naar welke ruimtelijke relatie men op zoek is. Zo kan men bijv. objecten die slechts raken, aan de buitenkant, uitsluiten. Maar ST_Relate gebruikt niet (SQL Server) of niet automatisch

(PostGIS) een ruimtelijke index. Toegepast op een tabel zal ST_Relate dus qua snelheid niet in verhouding staan tot het gebruik van ST_Intersects.

Het zoeken naar de dichtstbijzijnde wegvakken vanaf het punt dat de ‘oude’ oorsprong van het RD in Amersfoort voorstelt blijkt goed mogelijk (query 11). In Oracle zien de eerste records er als volgt uit (grafische weergave resultaat in figuur 7):

WVK_ID	NAAM	AFSTAND
310326069	Krankeledenstraat	12.141073279477
309326070	Breestraat	25
310326003	Lieve Vrouwekerkhof	25
310325057	Krankeledenstraat	32.015621187164
310325020	Lieve Vrouwekerkhof	32.015621187164
309326018	Westsingel	36.380020071675
310326027	Lieve Vrouwestraat	69.50027233238
310326026	Lieve Vrouwestraat	70.349129347846
310326028	Lieve Vrouwestraat	79.75274673637
310326004	Zwanenhalssteeg	79.75274673637
309326088	Hellestraat	87.182058275235

. . .

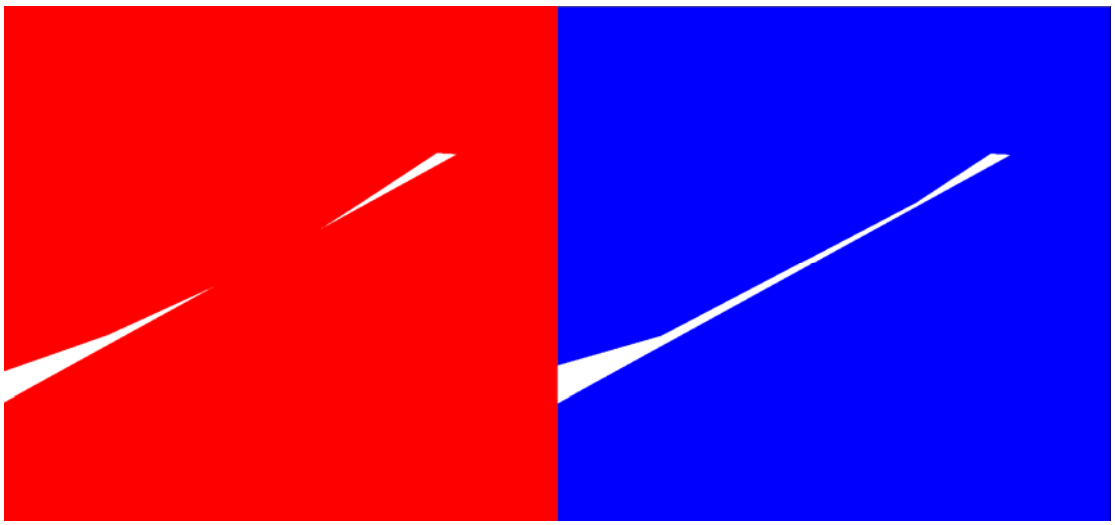


Figuur 7 Oracle resultaat voor selectie van 100 dichtstbijzijnde wegvakken vanaf punt RD 155000,463000, query 11 (in rood: wegvakken die niet bij de eerste 100 horen).

Op een globaal niveau is het resultaat van query 12, het genereren van een buffer rond 6 gemeenten, goed vergelijkbaar bij de drie databases. Een voorbeeld hiervan, de buffer rond Amsterdam, is weergegeven in figuur 8.

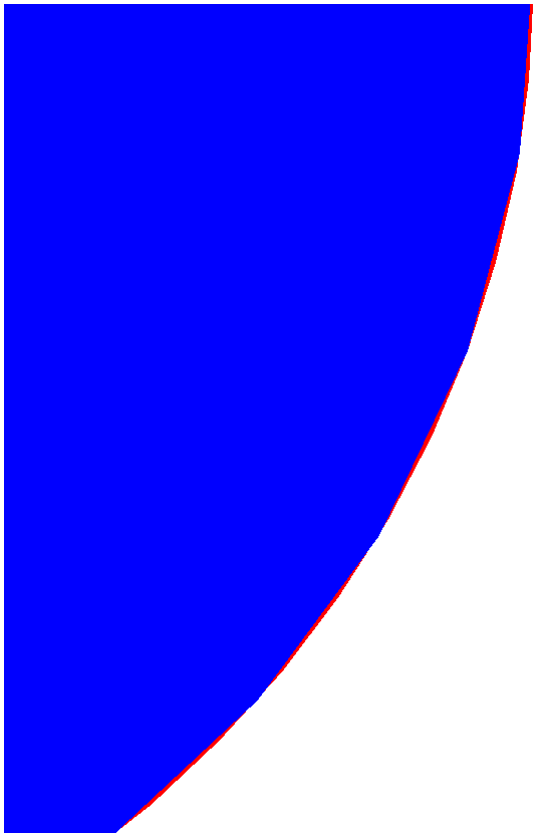
Figuur 8 Buffer van 500 m (rood) rond gemeente Amsterdam (zwart), query 12.

Meer in detail beschouwd zijn er allerlei kleine, en in een enkel geval ook wel wat grotere, verschillen waar te nemen. Een illustratie is te zien in figuur 9 waar er in Oracle een ‘gat’ onstaat in de buffer polygon (waar de buffers van Amsterdam en Amsterdam-Zuidoost elkaar raken) is dat bij PostGIS niet het geval



Figuur 9 Detail: verschil tussen Oracle buffer (rood) en PostGIS buffer (blauw).

Bij nadere beschouwing blijken de gegenereerde buffers ook enigszins verschillende typen geometrie te bevatten. In de Oracle buffers zitten ‘echte’ cirkelbogen, bij PostGIS en SQL Server bestaan de buffers alleen uit rechte lijnstukken waarbij de cirkelbogen benaderd zijn door een ‘stroked arc’ (zie figuur 10). Deze resultaten zijn bereikt met standaard instellingen bij het genereren van de buffers, in alle databases kan met toleranties en andere parameters het aanmaken van buffers nader gestuurd worden.



Figuur 10 Verschil tussen buffer met cirkelbogen (Oracle, rood, op achtergrond) en buffer met benaderde, ‘gelineariseerde’ cirkelbogen (PostGIS, blauw, voorgrond).

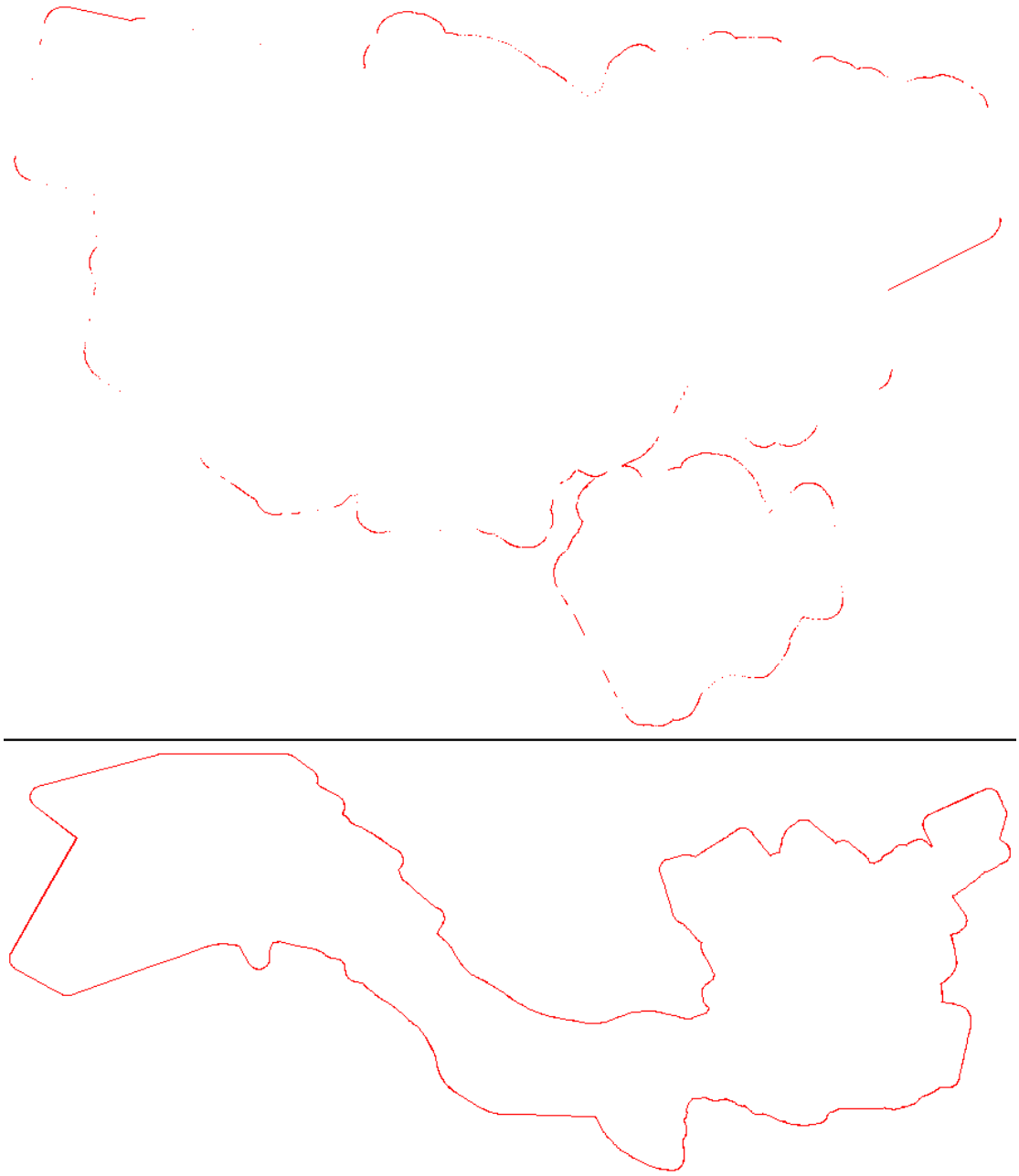
Een verdere indicatie van de verschillen geeft het oppervlak van de buffers als deze berekend en vergeleken worden (tabel 4).

Gemeente	Oracle	Ora - PG	Ora - SS
Amsterdam	267569031.9	52653.8	6349.7
Arnhem	131287150.7	30122.9	3182.6
Eindhoven	112722121.6	43104.6	4403.0
Groningen	106408385.3	35948.2	3945.4
Rotterdam	394407244.6	812228.0	757696.3
Utrecht	128207252.0	41332.7	4779.0

Tabel 4 Oppervlak in m^2 en verschil in oppervlak van de gegenereerde buffers, query 12.

In alle gevallen maakt Oracle de grootste buffers, PostGIS maakt de kleinste en SQL Server zit er tussen in, vrij dicht in de buurt van Oracle. Een deel van de verschillen is

het gevolg van het toepassen van cirkelbogen bij Oracle, maar de verschillen zouden veel kleiner zijn als dat het enige was. ‘Vreemd’ groot is het verschil bij Rotterdam waar Oracle om onduidelijke redenen een buffer maakt die royaal groter is dan bij de andere twee databases. Dit is ook visueel te zien in figuur 11. Weergegeven zijn daar de geometrische verschillen (Difference of SymDifference operatie) tussen de Oracle en PostGIS buffers. Het resultaat voor Amsterdam is wat je normaliter verwacht bij twee vlakken die relatief veel op elkaar lijken: een ‘GEOMETRYCOLLECTION’ bestaande uit punten, lijnen en vlakken (deze laatste in de vorm van ‘sliver polygons’).



Figuur 11 Geometrisch verschil in buffers tussen Oracle en PostGIS, Amsterdam (boven), Rotterdam (onder).

Voor Rotterdam is het resultaat een ‘simpele’ polygon met een groot gat. De strook die dan overblijft is ongeveer 5 m breed in het geval van Oracle - SQL Server, bij Oracle - PostGIS varieert de breedte van de strook tussen 4 m en 9 m.

Query responstijden

De responstijden in tabel 5 overziend valt het niet mee deze eenduidig te interpreteren. De gegevens zijn zo eerlijk en neutraal als mogelijk is verzameld, zoals eerder aangegeven. Toch is een DBMS een zodanig complex stuk software dat, afhankelijk van de configuratie of van ongemerkte interactie tussen hard- en software, onvoorziene effecten kunnen optreden. Het doel van dit onderzoek was nooit een echte benchmark uit te voeren, maar er wordt nu wel iets met performance gedaan en dat is per definitie ‘gevoelig’ (en moet dus goed worden onderbouwd).

Qry	Description	PG	Ora	SS	#PG/SS	#Ora	PG-cr	Ora-cr	SS-cr
01	basis, admin query	3.20	1.34	2.80	948481	948481	5.97	6.61	10.20
02	admin subset	1.13	1.00	0.94	42607	42607	1.86	1.59	2.05
03	klein window	0.02	0.22	8.94	524	524	0.04	0.31	5.01
04	middelgr. window	0.51	0.52	4.83	31835	31835	0.61	0.92	4.85
05	groot window	2.43	3.96	5.14	326461	326461	3.54	5.76	11.34
06	Limburg polygon	44.05	2.42	164.97	72901	72903	44.01	2.51	161.22
07	Delft polygon	0.19	0.27	131.05	4189	4195	0.19	0.37	135.74
08	lengte van lijnen	0.01	0.50	6.83	210	210			
09	ZH poly - gem poly	0.06	0.30	1.54	94	94			
10	NL line - gem	0.08	0.71	0.21	62	62			
11	nearest neighbors	0.02	0.12	8.02	100	100			
12	6 gem buffers	0.21	56.46	2.23	6	6			
13	NL window	5.96	10.37	6.11	948481	948481			
14	NL polygon	892.17	36.68	540.58	948481	948481			

PG, Ora, SS: responstijd in sec. van de ‘select count’ queries
PG-cr, Ora-cr, SS-cr: responstijd in sec. van de ‘create table as select’ queries
Groen: snelste van de 3 geteste databases
Rood: langzaamste van de 3 geteste databases
#PG/SS: aantal rows in resultaat bij PostgreSQL/PostGIS en SQL Server
#Ora: aantal rows in resultaat bij Oracle

Tabel 5 Query responstijden en aantallen rows als resultaat van de 14 resp. 7 test queries.

Met al deze ‘reserves’ in het achterhoofd kan worden vastgesteld dat geen van de geteste databases op alle aspecten het snelste of langzaamste is. PostGIS is de snelste bij een meerderheid van de queries, en doet het zeker erg goed op de belangrijke ‘window’ queries. Ook bij ‘berekeningen’ met de geometrie, zoals de lengte van lijnen of het genereren van buffers, is PostGIS relatief snel. Nogal onverwacht, gezien de ervaringen van de onderzoekers bij eerdere vergelijkingen van databases, is de matige performance van PostGIS bij selecties op basis van een polygon als deze laatste (wat) meer punten bevat, zoals bij query 06 en 14. Oracle scoort relatief goed op dat type query, maar heeft bijv. veel tijd nodig voor het genereren van buffers.

SQL Server springt er nergens in positieve zin uit waar het om de snelheid van selecties of operaties met geometrie gaat. Hierbij speelt wel mee dat bij de onderzoekers de kennis van en ervaring met deze database duidelijk minder is dan voor Oracle en PostGIS het geval is. Geen van de databases is voor dit onderzoek in hoge mate getuned, maar SQL Server is volledig out-of-the-box gebruikt. Dus potentieel is daar met beter configureren of meer tunen het meest te winnen.

Het resource gebruik tijdens de queries komt overeen met dat tijdens het aanmaken van de indexen. SQL Server 'pakt' van de beschikbare resources zoveel als nodig en/of mogelijk is: alle 24 CPU's van de testserver worden ingezet en 2 tot 2.5 GB memory. PostGIS en Oracle gebruiken bij de toegepaste configuratie niet meer dan 1 CPU en tussen 300 MB en 350 MB aan memory.

De 'create table as select' queries vertonen dezelfde trends als de 'select count' queries (zie Bijlagen B3.19, B3.24 en B3.29). In de gevallen dat er veel data gelezen of geschreven moet worden neemt het aandeel van de tijd voor I/O zichtbaar toe in de responstijd. Maar queries die eerder sterk 'CPU-bound' waren voor een database zijn dat nog steeds.

Opvallend is verder dat Oracle in eerste instantie grote moeite had met query 01. Van de 'create table as select' queries vergt deze verreweg de meeste I/O. Het is de eerste query die uitgevoerd wordt na het starten van de database, dus alle benodigde data moet gelezen worden. Het is ook de query met de grootste 'resultset', er moet meer weggeschreven worden dan bij de andere 'create table as select' queries. Bij de gebruikte configuratie, alle database data staat op dezelfde disk set, kreeg Oracle hier sterk last van 'disk contention'. Dit leidde tot een responstijd van 10 minuten i.p.v. de 6.6 sec. die nu als responstijd in de tabel staat. Deze laatste waarde is de responstijd bij Oracle na het oplossen van de 'disk contention' (nl. door het wegschrijven van het resultaat naar een andere disk). Vanuit de database gezien is er geen volledige controle over waar precies de database 'blocks' terechtkomen op disk. Dit vanwege het operating system en de RAID controller en software die tussen de database en de fysieke disks staan. Het zou kunnen dat de data bij Oracle door een samenloop van omstandigheden op 'onhandige' plekken terechtgekomen is. Maar een feit is dat, onder gelijke condities, Oracle wel en PostgreSQL/PostGIS en SQL Server geen last hadden van dit probleem.

3.4 Conclusies testen databases

De tests met de drie databases bevestigen voor een groot deel de algemene indruk die met de globale vergelijking in fase 1 van dit onderzoek ontstaan was. Er kan redelijk eenvoudig ruimtelijke data in deze databases opgeslagen worden, hoewel dit type data in veel gevallen wel iets meer inspanning vraagt dan 'gewone' alfanumerieke data. Eenmaal geladen kunnen er ruimtelijke indexen gemaakt worden. En vervolgens kan de ruimtelijke data meestal behoorlijk efficiënt opgevraagd worden. Ook berekeningen en bewerkingen met vector data, voor zover getest, verlopen goed.

Van Oracle was dit reeds bekend en ook al lang het geval, het is dus niet vreemd dat het voor ruimtelijke toepassingen binnen Rijkswaterstaat een standaard bouwsteen is (afgezien van de kosten).

De potentie van SQL Server wordt gestaafd door de bevindingen bij de tests, maar dit DBMS zit als geo-database nog niet op het niveau van Oracle of PostGIS. Er is minder functionaliteit beschikbaar, hoewel die hiaten niet in het deel zitten dat bij dit onderzoek is getest. Ook de performance is vaak de minste van de drie geteste databases, terwijl het gebruik van resources groot is. Een betere configuratie en tuning zouden kunnen helpen.

PostGIS, gecombineerd met de 'onderliggende' PostgreSQL database, heeft aangetoond dat het op veel aspecten net zo goed, of zelfs beter is dan Oracle. Het betreft dan dat deel van de ruimtelijke functionaliteit dat getest is in dit onderzoek en waarmee een aanzienlijk deel van Rijkswaterstaat toepassingen kan worden ondersteund. Het totaal van ruimtelijke functionaliteit overziend zijn er deelgebieden waar Oracle verder is (o.a. 3D, topologie, point clouds), maar bij PostGIS wordt er gewerkt aan verbeteringen op deze aspecten.

Qua performance zijn er, soms vrij grote, verschillen tussen de databases. Bij het bepaalde typen queries is PostGIS sneller, bij andere is dat Oracle. Een 'overall' winnaar is niet echt aan te geven, maar PostGIS werkt zeker efficiënt bij de meest gebruikte queries. Als database is PostgreSQL/PostGIS 'lean and mean', het doet wat het moet doen zonder allerlei overbodige rompslomp. Oracle kan veel en het heeft heel veel parameters en opties waarmee je kunt sturen. Maar dat sturen is ook eerder nodig dan bij PostgreSQL/PostGIS. Out-of-the-box kom je verder met PostgreSQL/PostGIS. Ook gebruikt deze laatste minder resources. Of andersom gezegd: met dezelfde resources kun je meer gebruikers bedienen.

Een eerste reactie op de bevindingen van dit onderzoek door de verschillende DBMS leveranciers is in Bijlage 2 te vinden.

4 Conclusies onderzoek

In dit slothoofdstuk zullen eerst de belangrijkste conclusies worden weergegeven in sectie 4.1. Vervolgens zullen in sectie 4.2 aanvullende ruimtelijke database functionaliteit en (mogelijke) testen kort worden beschreven, welke buiten de scope van het huidige onderzoek vallen (maar mogelijk in de toekomst wel relevant zijn).

4.1 Conclusies

In dit project hebben we in drie fases gekeken naar de geschiktheid van DBMS systemen als standaard bouwsteen voor ruimtelijke toepassingen binnen Rijkswaterstaat. In de eerste fase hebben we een globale vergelijking uitgevoerd van verschillende DBMSen: MySQL, PostGIS, SQL Server en Oracle. Op basis van een algemene analyse (hoofdstuk 1) is geconcludeerd dat de functionaliteit van MySQL onvoldoende is en in de vervolgfases niet meer meegenomen hoeft te worden.

De resterende DBMS'en (PostGIS, SQL Server en Oracle) zijn op twee manieren verder aan de tand gevoeld: door te vragen naar ervaringen bij gebruikers (hoofdstuk 2), en door het doen van concrete tests met de drie databases (hoofdstuk 3).

Omdat Rijkswaterstaat al bekend is met Oracle zijn alleen referentiebezoeken gepland bij gebruikers van PostGIS en SQL Server. Hierbij hebben we geprobeerd organisaties te bezoeken die vergelijkbaar zijn met Rijkswaterstaat. Voor PostGIS zijn we op bezoek geweest bij het PDOK project (waar Rijkswaterstaat ook partner in is). In dit project is o.a. voor PostGIS gekozen omdat de licentiekosten voor een schaalbare server architectuur, zoals nodig voor PDOK, met Oracle aardig op kunnen lopen, terwijl dezelfde functionaliteit met PostGIS is te leveren zonder licentiekosten. Voor SQL Server hebben we geen referentiecontact binnen de Nederlandse overheid kunnen vinden, de enige conclusie is derhalve dat het product (voor ruimtelijke toepassingen) niet veel gebruikt wordt.

In de functionele, en voor een klein deel performance, testen, is een stukje NWB in de drie database systemen geladen en een aantal voor Rijkswaterstaat relevante vragen daarop losgelaten. De geteste systemen blijken weinig problemen te hebben met het laden van de data en kunnen de testvragen binnen redelijke tijd beantwoorden. Er is niet één systeem aan te wijzen dat altijd het beste scoort.

Algemeen valt te concluderen dat PostGIS voor ruimtelijke toepassingen een geschikte kandidaat is om als standaard bouwsteen binnen Rijkswaterstaat op te nemen. Op functionaliteit en performance scoort dit DBMS in de tests niet minder dan de andere systemen. Wel is het verstandig voordat voor een specifieke ruimtelijke applicatie PostGIS (of SQL Server) daadwerkelijk wordt ingezet, na te gaan of de oplossing daadwerkelijk voldoende functionaliteit biedt en goed samengaat met de overige software componenten in de oplossing. Het is in de praktijk niet altijd zo dat het ene DBMS in een architectuur zomaar kan worden ingeruild voor het andere. Er blijken in de software stack toch altijd interoperabiliteits-issues te zijn. We zijn hier in dit onderzoek niet verder op ingegaan. Wel zien we dat het open source product PostGIS vaker wordt gebruikt in combinatie met andere open source producten (zoals GeoServer of OpenLayers).

Wanneer exact PostGIS in te zetten (in plaats van Oracle) is daarom afhankelijk van meerdere factoren. PostGIS wordt relatief vaak ingezet bij het massaal ontsluiten van geo-informatie via webservices. Er zijn echter ook situaties waarin er met verschillende andere systemen gecommuniceerd moet worden, die wel Oracle ondersteunen, maar PostGIS (nog) in mindere mate. Verder zijn er ook verschillende situaties waarin PostGIS de benodigde functionaliteit (nog) niet heeft en men dus bij Oracle uitkomt.

4.2 Overige functionaliteit en testen

Databases vergelijken en testen is een nagenoeg oneindige activiteit, dit komt o.a.. door de enorme variatie in functionaliteit (soorten data en gebruik) en de continue stroom van nieuwe releases. Om nog maar niet te spreken over de invloed van de onderliggende hardware componenten en operating systemen (en de variëteiten). Dit geldt voor ruimtelijke databases nog meer dat voor ‘gewone’ databases.

Oracle heeft zijn eigen SQL/spatial dialect (SDO_Geometry type en bijbehorende functies), en gedurende het onderzoek was dit de omgeving waarin getest is. Echter het is ook interessant om te onderzoeken in hoeverre Oracle goed overweg kan met de standaard SQL/SFS data typen en functies (ST), zowel wat functionaliteit als wat performance betreft. Dit is belangrijk om te weten in verband met. vervangen van het ene DBMS door het andere.

Wat betreft aard van de ruimtelijke data is in fase 2 en 3 van dit onderzoek vooral gekeken naar 2D vector data (simple features), maar natuurlijk zijn voor Rijkswaterstaat ook andere soorten ruimtelijke data relevant: 3D vector data, topologische structuren (planaire partities of lineaire netwerken), raster data (beelden), TINs (hoogtemodellen), puntenwolken (laser data of multi-beam echo data), etc. Ook is maar een beperkte set van de functionaliteit getest en is er veel meer dat niet getest is (denk bijv. aan een belangrijk onderwerp als coördinaat transformaties) dan dat er wel getest is. Bij het opzetten van eventuele toekomstige testen is het belangrijk om de juiste data sets te gebruiken en ook representatief gebruik van te voren vast te stellen (soorten queries, intensiteit en type van de updates, de te ondersteunen analyses).

Gedurende het onderzoek, kwam ook de vraag naar boven of de verschillende databases goed omgaan met de meerdere CPUs die beschikbaar zijn. Het testen hiervan was oorspronkelijk niet opgenomen en is dan ook slechts beperkt meegenomen. Dit is echter wel een zeer relevant onderwerp, zowel bij laden (massale dataset) als bij uitvoeren queries (veel gebruikers gelijktijdig kunnen bedienen), zou parallellisme voordelen kunnen bieden.

Literatuur en contactinformatie

Manuals en technische documentatie

MySQL 5.5 Reference Manual, Spatial Extensions
http://dev.mysql.com/doc/refman/5.5/en/spatial-extensions.html

Oracle Spatial Developer's Guide, 11g Release 2
http://docs.oracle.com/cd/E11882_01/appdev.112/e11830/toc.htm

PostGIS 2.0.2SVN Manual
http://postgis.refractions.net/documentation/manual-2.0/

PostgreSQL 9.1 Documentation
http://www.postgresql.org/docs/9.1/interactive/index.html

SQL Server 2012, Spatial Data
http://msdn.microsoft.com/en-us/library/bb933790.aspx

Ed Katibah, Milan Stojic
New Spatial Features in SQL Server Code-Named "Denali"
http://social.technet.microsoft.com/wiki/contents/articles/4136.aspx

Geo-DBMS vergelijkingen

Chrit Lemmen
Product Survey on Geo-databases, GIM International, May 2007
http://www.gim-international.com/files/productsurvey_v_pdfdocument_14.pdf

BostonGIS
Compare SQL Server 2008 R2, Oracle 11G R2, PostgreSQL/PostGIS 1.5 Spatial Features
http://www.bostongis.com/PrinterFriendly.aspx?content_name=sqlserver2008r2_oracle11gr2_postgis15_compare

Standaarden

Open GIS Consortium, Inc.
OpenGIS Simple Features Specification - For SQL, Revision 1.1 (Document 99-049)

Open Geospatial Consortium Inc.
OpenGIS® Implementation Standard for Geographic information - Simple feature access Part 1: Common architecture, Part 2: SQL option (Document 06-103r4)

International Organization for Standardization
International Standard ISO/IEC 13249-3; Information technology - Database languages - SQL multimedia and application packages - Part 3: Spatial (Document ISO/IEC 13249-3:2006(E))

Overige literatuur

Maarten Vermeij, Wilko Quak, Martin Kersten and Niels Nes, MonetDB, a novel spatial columnstore DBMS. In: Inge Netterberg and Serena Coetzee (Eds.); Academic Proceedings of the 2008 Free and Open Source for Geospatial (FOSS4G) Conference, OSGeo, pp. 193-199.

GISt Report No. 21, S. Zlatanova, T.P.M. Tijssen, P.J.M. van Oosterom and C.W. Quak, Research on usability of Oracle Spatial within the RWS organisation, (AGI-GAG-2003-21), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2003, 74 p.

GISt Report No. 23, T.P.M Tijssen, M.E. de Vries and P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle SDO_Geometry version 9i and version 10g Beta 2 (in the context of ArcGIS 8.3), Delft 2003, 20 p. (Confidential).

P.J.M. van Oosterom, C.W. Quak and T.P.M. Tijssen, Testing current DBMS products with real spatial data. In: E.M. Fendel, K. Jones, R. Laurini and M. Rumor (Eds.); Proceedings of UDMS '02, 23rd Urban Data Management Symposium, Prague, 2002, pp. VII.1-VII.18.

GISt Report No. 7, T.P.M. Tijssen, C.W. Quak and P.J.M. van Oosterom, Spatial DBMS testing with data from the Cadastre and TNO NITG, Delft 2001, 119 p.

Contactinformatie

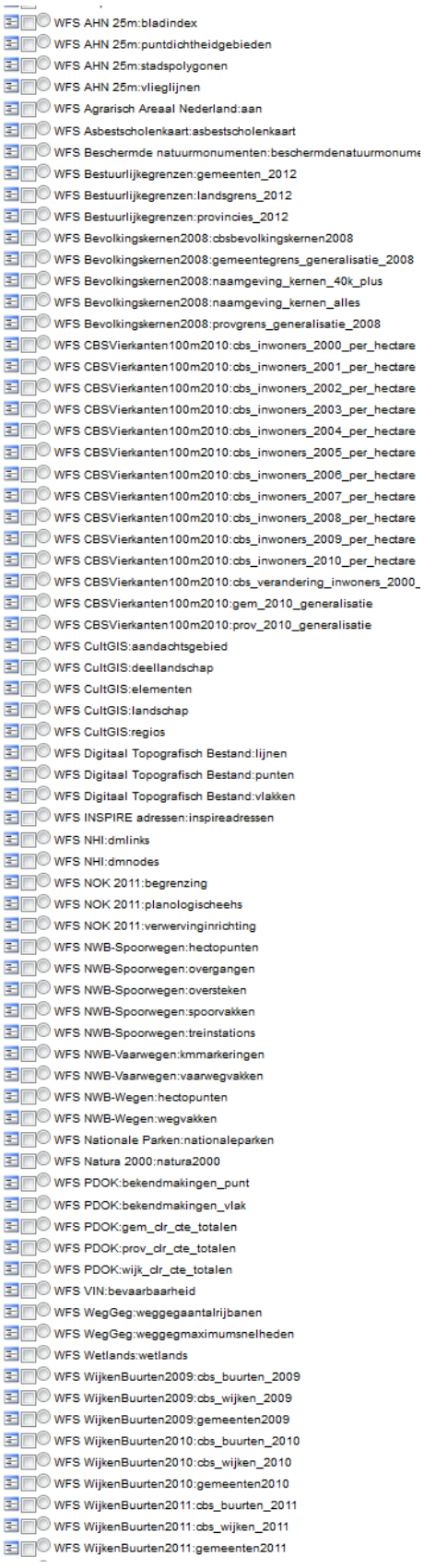
Oracle:
Siva Ravada [siva.ravada@oracle.com]
Oracle, Director of Development
Hans Viemann [hans.viehmann@oracle.com]
Oracle, Product Manager EMEA

In relatie tot PostGIS:
Thijs Brentjens [thijs@brentjensgeoict.nl]
Brentjens Geo-ICT, Geospatial Software Developer and Consultant
Paul Ramsey [pramsey@cleverelephant.ca]
Clever Elephant, Geospatial Architect
Erik van der Zee [erik.van.der.zee@geodan.nl]
Geodan, IT-architect

Microsoft SQL Server:
Martin Diepeveen [Martin.Diepeveen@microsoft.com]
Microsoft, Public Sector, Account Technology Strategist
Paul van Wingerden [paulvanw@microsoft.com]
Microsoft, technical evangelist

Bijlage 1 Overzicht PDOK kaartlagen

Lijst van alle WFS kaartlagen die door de PDOK server op te vragen zijn via:
http://geodata.nationaalgeoregister.nl



Bijlage 2 Reacties DBMS leveranciers

Op basis van de conceptrapportage is aan Oracle, PostGIS en Microsoft om een reactie gevraagd op de vergelijking van de drie geo-databases en de verdere bevindingen van het onderzoek. De ontvangen reacties zijn hier weergegeven.

Oracle

In een telefonische reactie geeft Hans Viehmann (hans.viehmann@oracle.com) aan niet in alle opzichten gelukkig te zijn met de manier waarop het onderzoek is uitgevoerd. Oracle vindt dat met name de test opzet met een ‘koude’ database, een enkele user op een verder lege server, etc. niet zo veel zegt over de meer realistische situaties in de meeste productieomgevingen waar dat allemaal niet het geval is. Volgens Oracle is hun database vooral goed in grootschalige multi-user omgevingen en bij gedistribueerde databases. Voor omgevingen waar kosten de bepalende factor zijn (en functionaliteit op basis niveau volstaat) werkt men aan ‘all-in-one’ pakketten met (SUN, nu Oracle) hardware, een operating system, DBMS software én data die in vergelijking met de bestaande Oracle omgevingen zeer gunstig geprijsd zal zijn. Verder verwacht men van Oracle versie 12, die momenteel in Beta is, significante verbeteringen in performance bij de meeste ruimtelijke operaties.

PostGIS

In een e-mail reactie schrijft Paul Ramsey (pramsey@cleverelephant.ca):

"I am also surprised about your results for the larger polygon queries, since those are pretty well optimized pieces of code (some of the most carefully polished in the project). It makes me wonder if something is going wrong further up the chain and you're getting a bad query plan driving things. If you run ANALYZE on your database to ensure all the stats are up-to-date do things work better? What does the EXPLAIN ANALYZE on the query look like?"

De ANALYZE op de database is voor de zekerheid opnieuw uitgevoerd maar dat maakte geen verschil. Daarna zijn de query plannen gegenereerd (zie Bijlagen B3.20 en B3.25). Deze geven een eerste indicatie wat er niet optimaal gaat in bepaalde gevallen. In een reactie daarop schrijft Paul:

“The problem *seems* to be in the statistics, in that we're grossly underestimating the number of rows the spatial join will return. Would it be too much to ask trying a run under PostGIS 1.5.x? I'm worried we might have introduced a regression in the statistics calculations in 2.0.”

Het proces van nagaan wat er gebeurt in andere versies van de PostgreSQL/PostGIS database loopt, de resultaten daarvan zijn nog niet bekend.

SQL Server

Paul van Wingerden (paulvanw@microsoft.com) schrijft in een e-mail:

“Ik heb (kort) naar de eerste resultaten gekeken en ik ben verbaasd over de verschillen in performance zoals die in de tabellen hieronder staan. Het verbaast me nog meer omdat dit niet in lijn is met wat ik van collega’s bij Microsoft in de US hoor die bij het bouwen van de spatial engine betrokken zijn, nog met mijn eigen ervaringen met bijvoorbeeld het laden en querien van bijvoorbeeld de BAG.

Om inhoudelijk naar deze case te kijken zou ik heel graag de onderliggende data willen hebben om de queries te kunnen analyzeren.
- Dit mag een backup van de database zijn, of de originele onderliggende data met het database schema. Ik ben zelf in staat de data te laden.
- Ik zal deze dan delen met bovengenoemde collega’s in de US.

De komende week (t/m 20 oktober) ben ik op vakantie. Ik zou heel graag in de week na mijn vakantie in detail naar de resultaten willen kijken om te ontdekken wat hieraan te verbeteren valt. Hiervoor zal ik ook mijn contacten bij Microsoft corp inschakelen. Ik heb hen reeds een mail gestuurd om dit onder de aandacht te brengen zodat er tijd voor vrijgemaakt kan worden. We kunnen hierover contact hebben per mail/telefoon, maar ik wil ook volgaarne bij jullie langskomen in Delft.

Ik heb in de cc mijn collega Jan Depping (jande@microsoft.com). Hij is op de hoogte van deze zaak en kan jullie te woord staan als jullie in de tussentijd eventuele vragen of opmerkingen hebben."

Bijlage 3 Scripts en log files

In deze bijlage zijn opgenomen de belangrijkste scripts zoals gebruikt bij het testen van de databases en een aantal configuratie- en log files.

B3.01 Oracle configuratie parameters

Output van ‘`select name, value from v$system_parameter;`’ voor de Oracle database gebruikt bij het onderzoek.

```
NAME                                 VALUE
------------------------------------ ------------------------------------------------------
lock_name_space
processes                            150
sessions                             264
timed_statistics                     TRUE
timed_os_statistics                  0
resource_limit                       FALSE
license_max_sessions                 0
license_sessions_warning             0
cpu_count                            24
instance_groups
event
sga_max_size                         1073741824
use_large_pages                      TRUE
pre_page_sga                         FALSE
shared_memory_address                0
hi_shared_memory_address             0
use_indirect_data_buffers            FALSE
lock_sga                             FALSE
processor_group_name
shared_pool_size                     0
large_pool_size                      0
java_pool_size                       0
streams_pool_size                    0
shared_pool_reserved_size            20342374
java_soft_sessionspace_limit         0
java_max_sessionspace_size           0
spfile                               /ora/u01/app/oracle/product/ee11.2.0/dbs/spfileTGEO.ora
instance_type                        RDBMS
nls_language                         AMERICAN
nls_territory                        AMERICA
nls_sort
nls_date_language

NAME                                 VALUE
------------------------------------ ------------------------------------------------------
nls_date_format
nls_currency
nls_numeric_characters
nls_iso_currency
nls_calendar
nls_time_format
nls_timestamp_format
nls_time_tz_format
nls_timestamp_tz_format
nls_dual_currency
nls_comp                             BINARY
nls_length_semantics                 CHAR
nls_nchar_conv_excp                  FALSE
fileio_network_adapters
filesystemio_options                 none
clonedb                              FALSE
disk_asynch_io                       TRUE
tape_asynch_io                       TRUE
dbwr_io_slaves                       0
backup_tape_io_slaves                TRUE
resource_manager_cpu_allocation      24
resource_manager_plan
cluster_interconnects
file_mapping                         FALSE
gcs_server_processes                 0
active_instance_count
sga_target                           0
memory_target                        1073741824
memory_max_target                    1073741824
control_files                        /ora/u02/oradata/TGEO/disk01/TGEO_control01.ctl,
                                     /ora/u02/oradata/TGEO/disk02/TGEO_control02.ctl
```

```
NAME                                 VALUE
------------------------------------ ------------------------------------------------------------
db_file_name_convert
log_file_name_convert
control_file_record_keep_time        7
db_block_buffers                     0
db_block_checksum                    TYPICAL
db_ultra_safe                        OFF
db_block_size                        8192
db_cache_size                        0
db_2k_cache_size                     0
db_4k_cache_size                     0
db_8k_cache_size                     0
db_16k_cache_size                    0
db_32k_cache_size                    0
db_keep_cache_size                   0
db_recycle_cache_size                0
db_writer_processes                  3
buffer_pool_keep
buffer_pool_recycle
db_flash_cache_file
db_flash_cache_size                  0
db_cache_advice                      ON
compatible                           11.2.0
log_archive_dest_1
log_archive_dest_2
log_archive_dest_3
log_archive_dest_4
log_archive_dest_5
log_archive_dest_6
log_archive_dest_7
log_archive_dest_8
log_archive_dest_9
log_archive_dest_10

NAME                                 VALUE
------------------------------------ ------------------------------------------------------------
log_archive_dest_11
log_archive_dest_12
log_archive_dest_13
log_archive_dest_14
log_archive_dest_15
log_archive_dest_16
log_archive_dest_17
log_archive_dest_18
log_archive_dest_19
log_archive_dest_20
log_archive_dest_21
log_archive_dest_22
log_archive_dest_23
log_archive_dest_24
log_archive_dest_25
log_archive_dest_26
log_archive_dest_27
log_archive_dest_28
log_archive_dest_29
log_archive_dest_30
log_archive_dest_31
log_archive_dest_state_1             enable
log_archive_dest_state_2             enable
log_archive_dest_state_3             enable
log_archive_dest_state_4             enable
log_archive_dest_state_5             enable
log_archive_dest_state_6             enable
log_archive_dest_state_7             enable
log_archive_dest_state_8             enable
log_archive_dest_state_9             enable
log_archive_dest_state_10            enable
log_archive_dest_state_11            enable

NAME                                 VALUE
------------------------------------ ------------------------------------------------------------
log_archive_dest_state_12            enable
log_archive_dest_state_13            enable
log_archive_dest_state_14            enable
log_archive_dest_state_15            enable
```

```
log_archive_dest_state_16              enable
log_archive_dest_state_17              enable
log_archive_dest_state_18              enable
log_archive_dest_state_19              enable
log_archive_dest_state_20              enable
log_archive_dest_state_21              enable
log_archive_dest_state_22              enable
log_archive_dest_state_23              enable
log_archive_dest_state_24              enable
log_archive_dest_state_25              enable
log_archive_dest_state_26              enable
log_archive_dest_state_27              enable
log_archive_dest_state_28              enable
log_archive_dest_state_29              enable
log_archive_dest_state_30              enable
log_archive_dest_state_31              enable
log_archive_start                      FALSE
log_archive_dest                       /ora/u02/oraarch/TGEO
log_archive_duplex_dest
log_archive_min_succeed_dest           1
standby_archive_dest                   ?/dbs/arch
fal_client
fal_server
log_archive_trace                      0
log_archive_config
log_archive_local_first                TRUE
log_archive_format                     TGEO_arch_%S_%t_%r.arc
redo_transport_user

NAME                                   VALUE
-------------------------------------- ----------------------------------------------------------
log_archive_max_processes              4
log_buffer                             9338880
log_checkpoint_interval                0
log_checkpoint_timeout                 1800
archive_lag_target                     0
db_files                               200
db_file_multiblock_read_count          8
read_only_open_delayed                 FALSE
cluster_database                       FALSE
parallel_server                        FALSE
parallel_server_instances              1
cluster_database_instances             1
db_create_file_dest
db_create_online_log_dest_1
db_create_online_log_dest_2
db_create_online_log_dest_3
db_create_online_log_dest_4
db_create_online_log_dest_5
db_recovery_file_dest
db_recovery_file_dest_size             0
standby_file_management                MANUAL
db_unrecoverable_scn_tracking          TRUE
thread                                 0
fast_start_io_target                   0
fast_start_mttr_target                 0
log_checkpoints_to_alert               FALSE
db_lost_write_protect                  NONE
recovery_parallelism                   0
db_flashback_retention_target          1440
dml_locks                              1160
replication_dependency_tracking        TRUE
transactions                           290

NAME                                   VALUE
-------------------------------------- ----------------------------------------------------------
transactions_per_rollback_segment      5
rollback_segments
undo_management                        AUTO
undo_tablespace                        UNDO
undo_retention                         10800
fast_start_parallel_rollback           LOW
resumable_timeout                      0
instance_number                        0
db_block_checking                      FALSE
recyclebin                             OFF
```

```
db_securefile                        PERMITTED
create_stored_outlines
serial_reuse                         disable
ldap_directory_access                NONE
ldap_directory_sysauth               no
os_roles                             FALSE
rdbms_server_dn
max_enabled_roles                    150
remote_os_authent                    FALSE
remote_os_roles                      FALSE
sec_case_sensitive_logon             FALSE
O7_DICTIONARY_ACCESSIBILITY          FALSE
remote_login_passwordfile            EXCLUSIVE
license_max_users                    0
audit_sys_operations                 FALSE
global_context_pool_size
db_domain                            RWS.NL
global_names                         FALSE
distributed_lock_timeout             60
commit_point_strength                1
global_txn_processes                 1
instance_name                        TGEO

NAME                                 VALUE
------------------------------------ -----------------------------------------------------
service_names                        TGEO.RWS.NL
dispatchers
shared_servers                       0
max_shared_servers
max_dispatchers
circuits
shared_server_sessions
local_listener
remote_listener
listener_networks
cursor_space_for_time                FALSE
session_cached_cursors               50
remote_dependencies_mode             TIMESTAMP
utl_file_dir
smtp_out_server
plsql_v2_compatibility               FALSE
plsql_warnings                       DISABLE:ALL
plsql_code_type                      NATIVE
plsql_debug                          FALSE
plsql_optimize_level                 3
plsql_ccflags
plscope_settings                     IDENTIFIERS:NONE
permit_92_wrap_format                TRUE
java_jit_enabled                     TRUE
job_queue_processes                  16
parallel_min_percent                 0
create_bitmap_area_size              8388608
bitmap_merge_area_size               1048576
cursor_sharing                       EXACT
result_cache_mode                    MANUAL
parallel_min_servers                 0
parallel_max_servers                 135

NAME                                 VALUE
------------------------------------ -----------------------------------------------------
parallel_instance_group
parallel_execution_message_size      16384
hash_area_size                       131072
result_cache_max_size                2686976
result_cache_max_result              5
result_cache_remote_expiration       0
shadow_core_dump                     partial
background_core_dump                 partial
background_dump_dest                 /ora/u02/oradata/diag/rdbms/tgeo/TGEO/trace
user_dump_dest                       /ora/u02/oradata/diag/rdbms/tgeo/TGEO/trace
core_dump_dest                       /ora/u02/oradata/diag/rdbms/tgeo/TGEO/cdump
audit_file_dest                      /ora/u02/oradata/admin/TGEO/audit
audit_syslog_level
object_cache_optimal_size            102400
object_cache_max_size_percent        10
session_max_open_files               10
```

```
open_links                           4
open_links_per_instance              4
commit_write
commit_wait
commit_logging
optimizer_features_enable            11.2.0.3
fixed_date
audit_trail                          NONE
sort_area_size                       65536
sort_area_retained_size              0
cell_offload_processing              TRUE
cell_offload_decryption              TRUE
cell_offload_parameters
cell_offload_compaction              ADAPTIVE
cell_offload_plan_display            AUTO
db_name                              TGEO

NAME                                 VALUE
------------------------------------ -----------------------------------------------------
db_unique_name                       TGEO
open_cursors                         1024
ifile
sql_trace                            FALSE
os_authent_prefix                    ops$
optimizer_mode                       ALL_ROWS
sql92_security                       FALSE
blank_trimming                       FALSE
star_transformation_enabled          FALSE
parallel_degree_policy               MANUAL
parallel_adaptive_multi_user         TRUE
parallel_threads_per_cpu             2
parallel_automatic_tuning            FALSE
parallel_io_cap_enabled              FALSE
optimizer_index_cost_adj             100
optimizer_index_caching              0
query_rewrite_enabled                TRUE
query_rewrite_integrity              TRUSTED
pga_aggregate_target                 0
workarea_size_policy                 AUTO
optimizer_dynamic_sampling           2
statistics_level                     TYPICAL
cursor_bind_capture_destination      memory+disk
skip_unusable_indexes                TRUE
optimizer_secure_view_merging        TRUE
ddl_lock_timeout                     0
deferred_segment_creation            TRUE
optimizer_use_pending_statistics     FALSE
optimizer_capture_sql_plan_baselines FALSE
optimizer_use_sql_plan_baselines     TRUE
parallel_min_time_threshold          AUTO
parallel_degree_limit                CPU

NAME                                 VALUE
------------------------------------ -----------------------------------------------------
parallel_force_local                 FALSE
optimizer_use_invisible_indexes      FALSE
dst_upgrade_insert_conv              TRUE
parallel_servers_target              384
sec_protocol_error_trace_action      TRACE
sec_protocol_error_further_action    CONTINUE
sec_max_failed_login_attempts        10
sec_return_server_release_banner     FALSE
enable_ddl_logging                   FALSE
client_result_cache_size             0
client_result_cache_lag              3000
aq_tm_processes                      1
hs_autoregister                      TRUE
xml_db_events                        enable
dg_broker_start                      FALSE
dg_broker_config_file1               /ora/u01/app/oracle/product/ee11.2.0/dbs/dr1TGEO.dat
dg_broker_config_file2               /ora/u01/app/oracle/product/ee11.2.0/dbs/dr2TGEO.dat
olap_page_pool_size                  0
asm_diskstring
asm_preferred_read_failure_groups
asm_diskgroups
asm_power_limit                      1
```

```
control_management_pack_access         DIAGNOSTIC+TUNING
awr_snapshot_time_offset               0
sqltune_category                       DEFAULT
diagnostic_dest                        /ora/u02/oradata
tracefile_identifier
max_dump_file_size                     unlimited
     trace_enabled                          TRUE
```

B3.02 PostgreSQL configuratie parameters

De PostgreSQL configuratie file '`postgresql.conf`', gebruikt in het onderzoek bij de PostgreSQL/PostGIS database, is onderstaand integraal opgenomen..

```
# 02 -----------------------------  23-08-2012  TT
# PostgreSQL 9.1 configuration file  /root/postgis/pgdata/postgresql.conf
# --------------------------------
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pg_ctl reload".  Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes        Time units:  ms  = milliseconds
#                MB = megabytes                     s   = seconds
#                GB = gigabytes                     min = minutes
#                                                   h   = hours
#                                                   d   = days

#------------------------------------------------------------------------------
# FILE LOCATIONS
#------------------------------------------------------------------------------

# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

#data_directory = 'ConfigDir'      # use data in another directory
                                   # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'# host-based authentication file
                                   # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf'    # ident configuration file
                                   # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = '(none)'      # write an extra PID file
                                   # (change requires restart)

#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'             # what IP address(es) to listen on;
                                   # comma-separated list of addresses;
                                   # defaults to 'localhost', '*' = all
                                   # (change requires restart)
port = 5432                        # (change requires restart)
max_connections = 128              # note: increasing max_connections costs
                                   # ~400 bytes of shared memory per
                                   # connection slot, plus lock space (see
                                   # max_locks_per_transaction).  You might
                                   # also need to raise shared_buffers to
                                   # support more connections.
superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directory = ''        # (change requires restart)
```

```
#unix_socket_group = ''            # (change requires restart)
#unix_socket_permissions = 0777    # begin with 0 to use octal notation
                                   # (change requires restart)
#bonjour_name = ''                 # defaults to the computer name
                                   # (change requires restart)

# - Security & Authentication -

authentication_timeout = 60s       # 1s-600s
ssl = off                          # (change requires restart)
ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers
                                   # (change requires restart)
ssl_renegotiation_limit = 1GB      # amount of data between renegotiations
password_encryption = on
db_user_namespace = off

# Kerberos and GSSAPI
#krb_server_keyfile = ''
#krb_srvname = 'postgres'          # (Kerberos only)
#krb_caseins_users = off

# - TCP Keepalives -               # see "man 7 tcp" for details

#tcp_keepalives_idle = 0           # TCP_KEEPIDLE, in seconds; 0 selects the system default
#tcp_keepalives_interval = 0       # TCP_KEEPINTVL, in seconds; 0 selects the system default
#tcp_keepalives_count = 0          # TCP_KEEPCNT; 0 selects the system default

#------------------------------------------------------------------------------
# RESOURCE USAGE (except WAL)
#------------------------------------------------------------------------------

# - Memory -

shared_buffers = 256MB             # min 128 KB or max_connections * 16 KB
temp_buffers = 16MB                # min 800 KB

max_prepared_transactions = 0      # zero disables the feature
                                   # (change requires restart)
# Note:  Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.

work_mem = 16MB                    # min 64 KB
maintenance_work_mem = 64MB        # min 1024 KB
max_stack_depth = 8MB              # min 100 KB

# - Kernel Resource Usage -

max_files_per_process = 1000       # min 25 (change requires restart)
#shared_preload_libraries = ''     # (change requires restart)

# - Cost-Based Vacuum Delay -

vacuum_cost_delay = 50ms           # 0-1000 milliseconds
vacuum_cost_page_hit = 1           # 0-10000 credits
vacuum_cost_page_miss = 10         # 0-10000 credits
vacuum_cost_page_dirty = 20        # 0-10000 credits
vacuum_cost_limit = 5000           # 0-10000 credits

# - Background writer -

bgwriter_delay = 200ms             # 10-10000 ms between rounds
bgwriter_lru_maxpages = 128        # 0-1000 buffers max written/round
bgwriter_lru_multiplier = 1.5      # 0-10.0 multipler on buffers scanned/round

# - Asynchronous Behavior -

effective_io_concurrency = 4       # 1-1000, 0 disables prefetching

#------------------------------------------------------------------------------
# WRITE AHEAD LOG
#------------------------------------------------------------------------------

# - Settings -
```

```
wal_level = minimal                # minimal, archive, or hot_standby
                                   # (change requires restart)

fsync = on                         # turns forced synchronization on or off
synchronous_commit = off           # synchronization level; on, off, or local

wal_sync_method = open_datasync    # the default is the first option supported
                                   # by the operating system: open_datasync,
                                   # fdatasync, fsync, fsync_writethrough, open_sync

full_page_writes = off             # recover from partial page writes
wal_buffers = -1                   # min 32kB, -1 sets based on shared_buffers
                                   # (change requires restart)
wal_writer_delay = 100ms           # 1-10000 milliseconds

commit_delay = 0                   # range 0-100000, in microseconds
#commit_siblings = 5               # range 1-1000

# - Checkpoints -

checkpoint_segments = 20           # in logfile segments, min 1, 16 MB each
checkpoint_timeout = 5min          # range 30s - 1h
checkpoint_completion_target = 0.5 # checkpoint target duration, 0.0 - 1.0
checkpoint_warning = 30            # in seconds, 0 is off

# - Archiving -

archive_mode = off                 # allows archiving to be done (change requires restart)
#archive_command = ''              # command to use to archive a logfile segment
#archive_timeout = 0               # force a logfile segment switch after this
                                   # number of seconds; 0 is off

#------------------------------------------------------------------------------
# REPLICATION
#------------------------------------------------------------------------------

# - Master Server -

# These settings are ignored on a standby server

#max_wal_senders = 0               # max number of walsender processes
                                   # (change requires restart)
#wal_sender_delay = 1s             # walsender cycle time, 1-10000 milliseconds
#wal_keep_segments = 0             # in logfile segments, 16MB each; 0 disables
#vacuum_defer_cleanup_age = 0      # number of xacts by which cleanup is delayed
#replication_timeout = 60s         # in milliseconds; 0 disables
#synchronous_standby_names = ''    # standby servers that provide sync rep
                                   # comma-separated list of application_name
                                   # from standby(s); '*' = all

# - Standby Servers -

# These settings are ignored on a master server

#hot_standby = off                 # "on" allows queries during recovery
                                   # (change requires restart)
#max_standby_archive_delay = 30s   # max delay before canceling queries
                                   # when reading WAL from archive;
                                   # -1 allows indefinite delay
#max_standby_streaming_delay = 30s # max delay before canceling queries
                                   # when reading streaming WAL;
                                   # -1 allows indefinite delay
#wal_receiver_status_interval = 10s# send replies at least this often
                                   # 0 disables
#hot_standby_feedback = off        # send info from standby to prevent
                                   # query conflicts

#------------------------------------------------------------------------------
# QUERY TUNING
#------------------------------------------------------------------------------

# - Planner Method Configuration -

#enable_bitmapscan = on
#enable_hashagg = on
#enable_hashjoin = on
```

```
#enable_indexscan = on
#enable_material = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on

# - Planner Cost Constants -

#seq_page_cost = 1.0                # measured on an arbitrary scale
#random_page_cost = 4.0             # same scale as above
#cpu_tuple_cost = 0.01              # same scale as above
#cpu_index_tuple_cost = 0.005       # same scale as above
#cpu_operator_cost = 0.0025         # same scale as above
#effective_cache_size = 128MB

# - Genetic Query Optimizer -

#geqo = on
#geqo_threshold = 12
#geqo_effort = 5                    # range 1-10
#geqo_pool_size = 0                 # selects default based on effort
#geqo_generations = 0               # selects default based on effort
#geqo_selection_bias = 2.0          # range 1.5-2.0
#geqo_seed = 0.0                    # range 0.0-1.0

# - Other Planner Options -

#default_statistics_target = 100   # range 1-10000
#constraint_exclusion = partition  # on, off, or partition
#cursor_tuple_fraction = 0.1       # range 0.0-1.0
#from_collapse_limit = 8
#join_collapse_limit = 8           # 1 disables collapsing of explicit JOIN clauses

#------------------------------------------------------------------------------
# ERROR REPORTING AND LOGGING
#------------------------------------------------------------------------------

silent_mode = on                   # Run server silently. DO NOT USE without syslog
                                   # or logging_collector (change requires restart)
# - Where to Log -

log_destination = 'stderr'         # Valid values are combinations of stderr, csvlog,
                                   # syslog, and eventlog, depending on platform.
                                   # csvlog requires logging_collector to be on.
                                   # (change requires restart)

# This is used when logging to stderr:
logging_collector = on             # Enable capturing of stderr and csvlog into log files.
                                   # Required to be on for csvlogs. (change requires restart)

# These are only used if logging_collector is on:
log_directory = '/var/log/pglog'   # Directory where log files are written
                                   # Can be absolute or relative to PGDATA
log_filename = 'postgres_%Y_%U.log'# Log file name pattern, can include strftime() escapes
                                   # (%U = week number)
log_truncate_on_rotation = off     # If on, any existing log file of the same name as the new
                                   # log file will be truncated rather than appended to. But
                                   # such truncation only occurs on time-driven rotation, not
                                   # on restarts or size-driven rotation. Default is off,
                                   # meaning append to existing files in all cases.
log_rotation_age = 7d              # Automatic rotation of logfiles will happen after that
time. 0 disables.
log_rotation_size = 0              # Automatic rotation of logfiles will happen after so many
                                   # kilobytes of log output, 0 to disable.

# These are relevant when logging to syslog:
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'

# - When to Log -

client_min_messages = notice       # Values, in order of decreasing detail:
                                   #   debug5
                                   #   debug4
```

```
                                #   debug3
                                #   debug2
                                #   debug1
                                #   log
                                # * notice
                                #   warning
                                #   error

log_min_messages = error        # Values, in order of decreasing detail:
                                #   debug5
                                #   debug4
                                #   debug3
                                #   debug2
                                #   debug1
                                #   info
                                #   notice
                                #   warning
                                # * error
                                #   log
                                #   fatal
                                #   panic

log_error_verbosity = default   # terse, default, or verbose messages

log_min_error_statement = fatal # values in order of decreasing detail:
                                #   debug5
                                #   debug4
                                #   debug3
                                #   debug2
                                #   debug1
                                #   info
                                #   notice
                                #   warning
                                #   error
                                #   log
                                # * fatal
                                #   panic (effectively off)

log_min_duration_statement = -1 # -1 is disabled, 0 logs all statements and their
                                # durations, > 0 logs only statements running
                                # at least this number of milliseconds

# - What to Log -

debug_print_parse = off
debug_print_rewritten = off
debug_print_plan = off
debug_pretty_print = off

log_lock_waits = on             # log lock waits >= deadlock_timeout
log_checkpoints = on
log_connections = off
log_disconnections = off

log_temp_files = 67108864       # log temporary files equal or larger
                                # than the specified size in kilobytes;
                                # -1 disables, 0 logs all temp files
log_duration = off
log_line_prefix = '%t '         # special values (e.g. '<%u%%%d> '):
                                #   %a = application name
                                #   %u = user name
                                #   %d = database name
                                #   %r = remote host and port
                                #   %h = remote host
                                #   %p = process ID
                                #   %t = timestamp without milliseconds
                                #   %m = timestamp with milliseconds
                                #   %i = command tag
                                #   %e = SQL state
                                #   %c = session ID
                                #   %l = session line number
                                #   %s = session start timestamp
                                #   %v = virtual transaction ID
                                #   %x = transaction ID (0 if none)
                                #   %q = stop here in non-session processes
                                #   %% = '%'
```

```
log_statement = 'none'             # none, ddl, mod, all
log_hostname = off
#log_timezone = '(defaults to server environment setting)'

#------------------------------------------------------------------------------
# RUNTIME STATISTICS
#------------------------------------------------------------------------------

# - Query/Index Statistics Collector -

track_activities = on
track_counts = on                  # must be on for autovacuum
track_functions = none             # none, pl, all
track_activity_query_size = 1024   # (change requires restart)
update_process_title = on
#stats_temp_directory = 'pg_stat_tmp'

# - Statistics Monitoring -

#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off
#log_statement_stats = off

#------------------------------------------------------------------------------
# AUTOVACUUM PARAMETERS
#------------------------------------------------------------------------------

autovacuum = on                    # enable autovacuum subprocess?
log_autovacuum_min_duration = 10000 # -1 disables, 0 logs all actions and their durations,
                                   # > 0 logs only actions running at least that time (msec)
autovacuum_max_workers = 3         # max number of autovacuum subprocesses
                                   # (change requires restart)
autovacuum_naptime = 5min          # time between autovacuum runs
autovacuum_vacuum_threshold = 500  # min # of tuple updates before vacuum
autovacuum_analyze_threshold = 500 # min # of tuple updates before analyze
autovacuum_vacuum_scale_factor = 0.1       # fraction of table size before vacuum
autovacuum_analyze_scale_factor = 0.1      # fraction of table size before analyze
autovacuum_freeze_max_age = 200000000      # maximum XID age before forced vacuum
                                   # (change requires restart)
autovacuum_vacuum_cost_delay = -1  # default vacuum cost delay for autovacuum,
                                   # -1 means use vacuum_cost_delay
autovacuum_vacuum_cost_limit = -1  # default vacuum cost limit for autovacuum,
                                   # -1 means use vacuum_cost_limit

#------------------------------------------------------------------------------
# CLIENT CONNECTION DEFAULTS
#------------------------------------------------------------------------------

# - Statement Behavior -

search_path = '"$user",public'     # schema names
default_tablespace = 'users'       # a tablespace name, '' uses the default
temp_tablespaces = 'temp'          # a list of tablespace names,
                                   # '' uses only default tablespace
check_function_bodies = on
default_transaction_isolation = 'read committed'
default_transaction_read_only = off
default_transaction_deferrable = off
session_replication_role = 'origin'
statement_timeout = 0              # in milliseconds, 0 is disabled
vacuum_freeze_min_age = 100000000
vacuum_freeze_table_age = 150000000

bytea_output = 'hex'               # hex, escape
xmlbinary = 'base64'
xmloption = 'content'

# - Locale and Formatting -

datestyle = 'iso, mdy'
intervalstyle = 'postgres'
#timezone = '(defaults to server environment setting)'
timezone_abbreviations = 'Default' # Select the set of available time zone
                                   # abbreviations, currently:
```

```
                                #   Default
                                #   Australia
                                #   India
                                # You can create your own file in share/timezonesets/
extra_float_digits = 2          # min -15, max 2
#client_encoding = sql_ascii    # actually, defaults to database encoding

# These settings are initialized by initdb -- they might be changed
lc_messages = 'C'               # locale for system error message strings
lc_monetary = 'C'               # locale for monetary formatting
lc_numeric = 'C'                # locale for number formatting
lc_time = 'C'                   # locale for time formatting

# default configuration for text search
default_text_search_config = 'pg_catalog.english'

# - Other Defaults -

#dynamic_library_path = '$libdir'
#local_preload_libraries = ''

#------------------------------------------------------------------------------
# LOCK MANAGEMENT
#------------------------------------------------------------------------------

deadlock_timeout = 1s
max_locks_per_transaction = 64     # min 10
                                # (change requires restart)
# note: each lock table slot uses ~270 bytes of shared memory, and there are
# max_locks_per_transaction * (max_connections + max_prepared_transactions)
# lock table slots.
max_pred_locks_per_transaction = 64 # min 10
                                # (change requires restart)

#------------------------------------------------------------------------------
# VERSION/PLATFORM COMPATIBILITY
#------------------------------------------------------------------------------

# - Previous PostgreSQL Versions -

#array_nulls = on
#backslash_quote = safe_encoding   # on, off, or safe_encoding
#default_with_oids = off
#escape_string_warning = on
#lo_compat_privileges = off
#quote_all_identifiers = off
#sql_inheritance = on
#standard_conforming_strings = on
#synchronize_seqscans = on

# - Other Platforms and Clients -

#transform_null_equals = off

#------------------------------------------------------------------------------
# ERROR HANDLING
#------------------------------------------------------------------------------

exit_on_error = off             # terminate session on any error?
restart_after_crash = on        # reinitialize after backend crash?

#------------------------------------------------------------------------------
# CUSTOMIZED OPTIONS
#------------------------------------------------------------------------------

#custom_variable_classes = ''      # list of custom variable class names

## EOF postgresql.conf
```

B3.03 SQL Server configuratie parameters

Output van '`EXEC sp_configure;`' voor de SQL Server test database.

name	minimum	maximum	config_value	run_value
access check cache bucket count	0	65536	0	0
access check cache quota	0	2147483647	0	0
Ad Hoc Distributed Queries	0	1	0	0
affinity I/O mask	-2147483648	2147483647	0	0
affinity mask	-2147483648	2147483647	0	0
affinity64 I/O mask	-2147483648	2147483647	0	0
affinity64 mask	-2147483648	2147483647	0	0
Agent XPs	0	1	0	0
allow updates	0	1	0	0
backup compression default	0	1	0	0
blocked process threshold (s)	0	86400	0	0
c2 audit mode	0	1	0	0
clr enabled	0	1	0	0
common criteria compliance enabled	0	1	0	0
contained database authentication	0	1	0	0
cost threshold for parallelism	0	32767	5	5
cross db ownership chaining	0	1	0	0
cursor threshold	-1	2147483647	-1	-1
Database Mail XPs	0	1	0	0
default full-text language	0	2147483647	1033	1033
default language	0	9999	0	0
default trace enabled	0	1	1	1
disallow results from triggers	0	1	0	0
EKM provider enabled	0	1	0	0
filestream access level	0	2	0	0
fill factor (%)	0	100	0	0
ft crawl bandwidth (max)	0	32767	100	100
ft crawl bandwidth (min)	0	32767	0	0
ft notify bandwidth (max)	0	32767	100	100
ft notify bandwidth (min)	0	32767	0	0
index create memory (KB)	704	2147483647	0	0
in-doubt xact resolution	0	2	0	0
lightweight pooling	0	1	0	0
locks	5000	2147483647	0	0
max degree of parallelism	0	32767	0	0
max full-text crawl range	0	256	4	4
max server memory (MB)	128	2147483647	2147483647	2147483647
max text repl size (B)	-1	2147483647	65536	65536
max worker threads	128	65535	0	0
media retention	0	365	0	0
min memory per query (KB)	512	2147483647	1024	1024
min server memory (MB)	0	2147483647	0	16
nested triggers	0	1	1	1
network packet size (B)	512	32767	4096	4096
Ole Automation Procedures	0	1	0	0
open objects	0	2147483647	0	0
optimize for ad hoc workloads	0	1	0	0
PH timeout (s)	1	3600	60	60
precompute rank	0	1	0	0
priority boost	0	1	0	0
query governor cost limit	0	2147483647	0	0
query wait (s)	-1	2147483647	-1	-1
recovery interval (min)	0	32767	0	0
remote access	0	1	1	1
remote admin connections	0	1	0	0
remote login timeout (s)	0	2147483647	10	10
remote proc trans	0	1	0	0
remote query timeout (s)	0	2147483647	600	600
Replication XPs	0	1	0	0
scan for startup procs	0	1	0	0
server trigger recursion	0	1	1	1
set working set size	0	1	0	0
show advanced options	0	1	1	1
SMO and DMO XPs	0	1	1	1
transform noise words	0	1	0	0
two digit year cutoff	1753	9999	2049	2049
user connections	0	32767	0	0
user options	0	32767	0	0
xp_cmdshell	0	1	0	0

B3.04 Import initiële Oracle dump met NWB data

Log file van de Oracle ‘imp’ commando’s gebruikt om de dump met NWB data in te lezen in een Oracle database.

```
Script started on Thu 23 Aug 2012 03:48:19 PM CEST

+ imp system@TGEO fromuser=NWA touser=ORATEST file=bnso1.dmp parfile=imp.par
log=nwa.log
'tables=(GGA_GEMEENTEN,GGA_GEMEENTE_EFEMERIDEN,GGA_PROVINCIES,GGA_REF_CODES,GGA_WOONPL
AATSEN,NWA_DISTRICTEN,NWA_REGIONALE_DIENSTEN,NWW_GEOMETRIEBRONNEN)'

Import: Release 11.2.0.3.0 - Production on Thu Aug 23 15:48:29 2012

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

Export file created by EXPORT:V09.02.00 via conventional path

Warning: the objects were exported by BNP, not by you

import done in US7ASCII character set and AL16UTF16 NCHAR character set
import server uses AL32UTF8 character set (possible charset conversion)
. importing NWA's objects into ORATEST
. . importing table                "GGA_GEMEENTEN"        932 rows imported
. . importing table      "GGA_GEMEENTE_EFEMERIDEN"       8029 rows imported
. . importing table               "GGA_PROVINCIES"         12 rows imported
. . importing table                "GGA_REF_CODES"         35 rows imported
. . importing table             "GGA_WOONPLAATSEN"       2651 rows imported
. . importing table               "NWA_DISTRICTEN"         19 rows imported
. . importing table       "NWA_REGIONALE_DIENSTEN"          9 rows imported
. . importing table         "NWW_GEOMETRIEBRONNEN"         12 rows imported
About to enable constraints...
Import terminated successfully without warnings.
+ imp system@TGEO fromuser=NWG touser=ORATEST file=bnso1.dmp parfile=imp.par
log=nwg.log
'tables=(GEB_STRUCTUREN,GEB_STRUCTUURELEMENTEN,GEB_ZONES,GEB_ZONESOORTEN,GEB_ZONE_EFEM
ERIDEN,GEB_ZONE_ZONE_RELATIES,NWG_STRUCTUREN,NWG_STRUCTUURELEMENTEN,NWG_ZONES,NWG_ZONE
SOORTEN,NWG_ZONE_EFN,NWG_ZONE_ZONE_RELATIES)'

Import: Release 11.2.0.3.0 - Production on Thu Aug 23 15:49:08 2012

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

Export file created by EXPORT:V09.02.00 via conventional path

Warning: the objects were exported by BNP, not by you

import done in US7ASCII character set and AL16UTF16 NCHAR character set
import server uses AL32UTF8 character set (possible charset conversion)
. importing NWG's objects into ORATEST
. . importing table               "GEB_STRUCTUREN"         14 rows imported
. . importing table       "GEB_STRUCTUURELEMENTEN"         40 rows imported
. . importing table                   "GEB_ZONES"       1447 rows imported
. . importing table             "GEB_ZONESOORTEN"         34 rows imported
. . importing table          "GEB_ZONE_EFEMERIDEN"       4100 rows imported
. . importing table       "GEB_ZONE_ZONE_RELATIES"       2091 rows imported
. . importing table               "NWG_STRUCTUREN"         15 rows imported
. . importing table       "NWG_STRUCTUURELEMENTEN"         43 rows imported
. . importing table                   "NWG_ZONES"       1580 rows imported
. . importing table             "NWG_ZONESOORTEN"         37 rows imported
. . importing table                "NWG_ZONE_EFN"       5724 rows imported
. . importing table       "NWG_ZONE_ZONE_RELATIES"       3892 rows imported
About to enable constraints...
Import terminated successfully without warnings.
```

```
+ imp system@TGEO fromuser=NWK touser=ORATEST file=bnso1.dmp parfile=imp.par
log=nwk.log
'tables=(NWK_SPOORWEGOVERGANGEN,NWK_SPOORWEGOVERGANG_EFN,NWK_SVK_WVK_KRUISINGEN)'

Import: Release 11.2.0.3.0 - Production on Thu Aug 23 15:50:13 2012

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

Export file created by EXPORT:V09.02.00 via conventional path

Warning: the objects were exported by BNP, not by you

import done in US7ASCII character set and AL16UTF16 NCHAR character set
import server uses AL32UTF8 character set (possible charset conversion)
. importing NWK's objects into ORATEST
. . importing table       "NWK_SPOORWEGOVERGANGEN"       4418 rows imported
. . importing table     "NWK_SPOORWEGOVERGANG_EFN"       4937 rows imported
. . importing table       "NWK_SVK_WVK_KRUISINGEN"       8720 rows imported
About to enable constraints...
Import terminated successfully without warnings.
+ imp system@TGEO fromuser=NWS touser=ORATEST file=bnso1.dmp parfile=imp.par
log=nws.log
'tables=(NWS_SPOORBAANVAKKEN,NWS_SPOORHECTOPUNT_EFEMERIDEN,NWS_SPOORJUNCTIES,NWS_SPOOR
JUNCTIE_EFEMERIDEN,NWS_SPOOROVERSTEEK_EFEMERIDEN,NWS_SPOORVAKKEN,NWS_SVK_EFEMERIDEN,NW
S_SVK_SMT_EFEMERIDEN,NWS_TREINSTATIONS,NWS_TREINSTATION_EFEMERIDEN,NWS_TSN_SVK_EFEMERI
DEN)'

Import: Release 11.2.0.3.0 - Production on Thu Aug 23 15:51:09 2012

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

Export file created by EXPORT:V09.02.00 via conventional path

Warning: the objects were exported by BNP, not by you

import done in US7ASCII character set and AL16UTF16 NCHAR character set
import server uses AL32UTF8 character set (possible charset conversion)
. importing NWS's objects into ORATEST
. . importing table          "NWS_SPOORBAANVAKKEN"        577 rows imported
. . importing table "NWS_SPOORHECTOPUNT_EFEMERIDEN"     40331 rows imported
. . importing table            "NWS_SPOORJUNCTIES"       1369 rows imported
. . importing table  "NWS_SPOORJUNCTIE_EFEMERIDEN"       1386 rows imported
. . importing table "NWS_SPOOROVERSTEEK_EFEMERIDEN"      1356 rows imported
. . importing table              "NWS_SPOORVAKKEN"       1675 rows imported
. . importing table           "NWS_SVK_EFEMERIDEN"       3102 rows imported
. . importing table       "NWS_SVK_SMT_EFEMERIDEN"       1996 rows imported
. . importing table            "NWS_TREINSTATIONS"        432 rows imported
. . importing table  "NWS_TREINSTATION_EFEMERIDEN"        473 rows imported
. . importing table       "NWS_TSN_SVK_EFEMERIDEN"        610 rows imported
About to enable constraints...
Import terminated successfully without warnings.
+ imp system@TGEO fromuser=NWV touser=ORATEST file=bnso1.dmp parfile=imp.par
log=nwv.log
'tables=(NWV_GEBRUIKERS,NWV_KM_MARKERING_EFEMERIDEN,NWV_MARKERINGSOORTEN,NWV_MEETLATTE
N,NWV_VAARROUTES,NWV_VAARROUTE_EFEMERIDEN,NWV_VAARWEGEN,NWV_VAARWEGJUNCTIES,NWV_VAARWE
GJUNCTIE_EFEMERIDEN,NWV_VAARWEGVAKKEN,NWV_VAARWEGVAK_EFEMERIDEN,NWV_VAARWEG_EFEMERIDEN
)'

Import: Release 11.2.0.3.0 - Production on Thu Aug 23 15:51:41 2012

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

Export file created by EXPORT:V09.02.00 via conventional path
```

```
Warning: the objects were exported by BNP, not by you

import done in US7ASCII character set and AL16UTF16 NCHAR character set
import server uses AL32UTF8 character set (possible charset conversion)
. importing NWV's objects into ORATEST
. . importing table                "NWV_GEBRUIKERS"         10 rows imported
. . importing table  "NWV_KM_MARKERING_EFEMERIDEN"       7116 rows imported
. . importing table          "NWV_MARKERINGSOORTEN"          3 rows imported
. . importing table                "NWV_MEETLATTEN"      11784 rows imported
. . importing table                "NWV_VAARROUTES"       1191 rows imported
. . importing table     "NWV_VAARROUTE_EFEMERIDEN"       1293 rows imported
. . importing table                 "NWV_VAARWEGEN"       2465 rows imported
. . importing table           "NWV_VAARWEGJUNCTIES"       3374 rows imported
. . importing table "NWV_VAARWEGJUNCTIE_EFEMERIDEN"      4329 rows imported
. . importing table             "NWV_VAARWEGVAKKEN"       4051 rows imported
. . importing table    "NWV_VAARWEGVAK_EFEMERIDEN"       6489 rows imported
. . importing table       "NWV_VAARWEG_EFEMERIDEN"       2610 rows imported
About to enable constraints...
Import terminated successfully without warnings.
+ imp system@TGEO fromuser=NWW touser=ORATEST file=bnso1.dmp parfile=imp.par
log=nww.log
'tables=(GEB_SO_KRUISPUNTEN,GEB_SO_WEGVAKKEN,GGA_STRATEN,NWW_BAANSOORTEN,NWW_BAANSUBSO
ORTEN,NWW_HECTOPUNTEN,NWW_HECTO_INTERVALLEN,NWW_JUNCTIES,NWW_RELATIEVE_POSITIES,NWW_RO
UTES,NWW_VERKEERSBAAN_EFEMERIDEN,NWW_VERKEERSBANEN,NWW_WEGBEHEERDERS,NWW_WEGVAKKEN,NWW
_WEGVAK_EFEMERIDEN,NWW_WEGVAK_IN_ROUTES,NWW_WEGVAK_RELATIES)'

Import: Release 11.2.0.3.0 - Production on Thu Aug 23 15:52:11 2012

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

Export file created by EXPORT:V09.02.00 via conventional path

Warning: the objects were exported by BNP, not by you

import done in US7ASCII character set and AL16UTF16 NCHAR character set
import server uses AL32UTF8 character set (possible charset conversion)
. importing NWW's objects into ORATEST
. . importing table           "GEB_SO_KRUISPUNTEN"      29619 rows imported
. . importing table             "GEB_SO_WEGVAKKEN"      79347 rows imported
. . importing table                  "GGA_STRATEN"     355891 rows imported
. . importing table             "NWW_BAANSOORTEN"         10 rows imported
. . importing table           "NWW_BAANSUBSOORTEN"         23 rows imported
. . importing table             "NWW_HECTOPUNTEN"    1679143 rows imported
. . importing table        "NWW_HECTO_INTERVALLEN"     363561 rows imported
. . importing table                 "NWW_JUNCTIES"     840841 rows imported
. . importing table       "NWW_RELATIEVE_POSITIES"          6 rows imported
. . importing table                   "NWW_ROUTES"        674 rows imported
. . importing table  "NWW_VERKEERSBAAN_EFEMERIDEN"      11359 rows imported
. . importing table            "NWW_VERKEERSBANEN"       9716 rows imported
. . importing table            "NWW_WEGBEHEERDERS"        980 rows imported
. . importing table                "NWW_WEGVAKKEN"    1372377 rows imported
. . importing table        "NWW_WEGVAK_EFEMERIDEN"    2923714 rows imported
. . importing table         "NWW_WEGVAK_IN_ROUTES"     161210 rows imported
. . importing table          "NWW_WEGVAK_RELATIES"     198869 rows imported
About to enable constraints...
Import terminated successfully without warnings.
+ exit

Script done on Thu 23 Aug 2012 04:11:38 PM CEST
```

B3.05 Oracle log van fouten in de NWB geometrie

Output van de Oracle ‘sdo_geom.validate_geometry_with_context’ validatie functie toegepast op de NWB dataset (alleen tabellen met fouten weergegeven).

```
SQL> select SDE_ID, status from (select SDE_ID, sdo_geom.validate_geometry_with_context(POLYGOON,
  2    (select diminfo from user_sdo_geom_metadata where table_name='GEB_ZONE_EFEMERIDEN')) status
  3  from GEB_ZONE_EFEMERIDEN) where status <> 'TRUE';

    SDE_ID STATUS
---------- --------------------------------------------------------------------------------
      4228 13349 [Element <1>] [Ring <1>][Edge <14>][Edge <13>]
      4241 13349 [Element <1>] [Ring <1>][Edge <56>][Edge <57>]
      4254 13349 [Element <1>] [Ring <1>][Edge <121>][Edge <122>]
      1696 13349 [Element <2>] [Ring <1>][Edge <2>][Edge <1>]
      4349 13349 [Element <1>] [Ring <1>][Edge <423>][Edge <421>]
      4344 13349 [Element <1>] [Ring <1>][Edge <847>][Edge <845>]
      1869 13349 [Element <2>] [Ring <1>][Edge <2>][Edge <1>]
      1872 13351 [Element <1>] [Ring <1>][Edge <1>] [Element <2>] [Ring <1>][Edge <4>]
      4196 13349 [Element <1>] [Ring <1>][Edge <16>][Edge <14>]
      4210 13349 [Element <1>] [Ring <1>][Edge <170>][Edge <171>]
      1922 13351 [Element <1>] [Ring <1>][Edge <472>] [Element <2>] [Ring <1>][Edge <2>]
      1938 13351 [Element <1>] [Ring <1>][Edge <443>] [Element <2>] [Ring <1>][Edge <4>]
      1940 13349 [Element <8>] [Ring <1>][Edge <1>][Edge <3>]
      2878 13349 [Element <1>] [Ring <1>][Edge <48>][Edge <53>]
      2946 13349 [Element <1>] [Ring <1>][Edge <169>][Edge <170>]
      4214 13349 [Element <1>] [Ring <1>][Edge <189>][Edge <187>]
      2774 13349 [Element <2>] [Ring <1>][Edge <2>][Edge <4>]
      4240 13349 [Element <1>] [Ring <1>][Edge <123>][Edge <124>]
      3681 13349 [Element <1>] [Ring <1>][Edge <382>][Edge <384>]
      3687 13349 [Element <1>] [Ring <1>][Edge <90>][Edge <88>]
      3692 13349 [Element <1>] [Ring <1>][Edge <1>][Edge <3>]
      3694 13349 [Element <1>] [Ring <1>][Edge <10>][Edge <8>]
      3705 13349 [Element <1>] [Ring <1>][Edge <1>][Edge <210>]
      3706 13349 [Element <1>] [Ring <1>][Edge <4>][Edge <2>]
      4294 13349 [Element <1>] [Ring <1>][Edge <17>][Edge <16>]
      3708 13349 [Element <1>] [Ring <1>][Edge <254>][Edge <255>]
      3709 13349 [Element <1>] [Ring <1>][Edge <44>][Edge <45>]
      4202 13349 [Element <1>] [Ring <1>][Edge <44>][Edge <45>]
      4183 13349 [Element <1>] [Ring <1>][Edge <250>][Edge <251>]

29 rows selected.

Elapsed: 00:00:03.21
SQL> select count(*),g.POLYGOON.SDO_GTYPE from GEB_ZONE_EFEMERIDEN g group by g.POLYGOON.SDO_GTYPE
order by g.POLYGOON.SDO_GTYPE;

  COUNT(*) POLYGOON.SDO_GTYPE
---------- ------------------
      3562               2003
        56               2007

SQL> select GME_ID, status from (select GME_ID, sdo_geom.validate_geometry_with_context(POLYGOON,
  2    (select diminfo from user_sdo_geom_metadata where table_name='GGA_GEMEENTE_EFEMERIDEN'))
status
  3  from GGA_GEMEENTE_EFEMERIDEN) where status <> 'TRUE';

    GME_ID STATUS
---------- --------------------------------------------------------------------------------
       211 13349 [Element <1>] [Ring <1>][Edge <48>][Edge <50>]

Elapsed: 00:00:05.05
SQL> select count(*),g.POLYGOON.SDO_GTYPE from GGA_GEMEENTE_EFEMERIDEN g group by
g.POLYGOON.SDO_GTYPE order by g.POLYGOON.SDO_GTYPE;

  COUNT(*) POLYGOON.SDO_GTYPE
---------- ------------------
      7699               2003
        25               2007

SQL> select ID, status from (select ID, sdo_geom.validate_geometry_with_context(POLYGOON,
  2    (select diminfo from user_sdo_geom_metadata where table_name='NWG_ZONE_EFN')) status
  3  from NWG_ZONE_EFN) where status <> 'TRUE';
```

ID	STATUS
1938	13351 [Element <1>] [Ring <1>][Edge <443>] [Element <2>] [Ring <1>][Edge <4>]
1940	13349 [Element <8>] [Ring <1>][Edge <1>][Edge <3>]
2774	13349 [Element <2>] [Ring <1>][Edge <2>][Edge <4>]
2878	13349 [Element <1>] [Ring <1>][Edge <48>][Edge <53>]
6608	13367 [Element <1>] [Ring <1>]
6609	13367 [Element <1>] [Ring <1>]
6610	13367 [Element <1>] [Ring <1>]
6611	13367 [Element <1>] [Ring <1>]
6612	13367 [Element <1>] [Ring <1>]
6613	13367 [Element <1>] [Ring <1>]
6590	13367 [Element <1>] [Ring <1>]
6615	13367 [Element <1>] [Ring <1>]
6616	13367 [Element <1>] [Ring <1>]
6567	13367 [Element <1>] [Ring <1>]
6569	13367 [Element <1>] [Ring <1>]
6570	13367 [Element <1>] [Ring <1>]
6571	13367 [Element <1>] [Ring <1>]
6572	13367 [Element <1>] [Ring <1>]
6573	13367 [Element <1>] [Ring <1>]
6574	13367 [Element <1>] [Ring <1>]
6576	13367 [Element <1>] [Ring <1>]
6577	13367 [Element <1>] [Ring <1>]
6578	13367 [Element <1>] [Ring <1>]
6579	13367 [Element <1>] [Ring <1>]
6580	13367 [Element <1>] [Ring <1>]
6581	13367 [Element <1>] [Ring <1>]
6582	13367 [Element <1>] [Ring <1>]
6585	13367 [Element <1>] [Ring <1>]
6586	13367 [Element <1>] [Ring <1>]
6587	13367 [Element <1>] [Ring <1>]
6588	13367 [Element <1>] [Ring <1>]
6589	13367 [Element <1>] [Ring <1>]
6685	13367 [Element <1>] [Ring <1>]
6700	13367 [Element <1>] [Ring <1>]
6699	13367 [Element <1>] [Ring <1>]
6669	13367 [Element <1>] [Ring <1>]
6645	13367 [Element <1>] [Ring <1>]
6603	13367 [Element <1>] [Ring <1>]
6643	13367 [Element <1>] [Ring <1>]
6644	13367 [Element <1>] [Ring <1>]
6651	13367 [Element <1>] [Ring <1>]
6652	13367 [Element <1>] [Ring <1>]
6653	13367 [Element <1>] [Ring <1>]
6655	13367 [Element <1>] [Ring <1>]
6663	13367 [Element <1>] [Ring <1>]
6606	13367 [Element <1>] [Ring <1>]
6614	13367 [Element <1>] [Ring <1>]
6618	13367 [Element <1>] [Ring <1>]
6619	13367 [Element <1>] [Ring <1>]
6620	13367 [Element <1>] [Ring <1>]
6621	13367 [Element <1>] [Ring <1>]
6617	13367 [Element <1>] [Ring <1>]
6623	13367 [Element <1>] [Ring <1>]
6628	13367 [Element <1>] [Ring <1>]
6629	13367 [Element <1>] [Ring <1>]
6624	13367 [Element <1>] [Ring <1>]
6625	13367 [Element <1>] [Ring <1>]
6630	13367 [Element <1>] [Ring <1>]
6631	13367 [Element <1>] [Ring <1>]
6632	13367 [Element <1>] [Ring <1>]
6633	13367 [Element <1>] [Ring <1>]
6634	13367 [Element <1>] [Ring <1>]
6635	13367 [Element <1>] [Ring <1>]
6637	13367 [Element <1>] [Ring <1>]
6638	13367 [Element <1>] [Ring <1>]
6639	13367 [Element <1>] [Ring <1>]
6640	13367 [Element <1>] [Ring <1>]
6641	13367 [Element <1>] [Ring <1>]
6642	13367 [Element <1>] [Ring <1>]
6636	13367 [Element <1>] [Ring <1>]
6554	13367 [Element <1>] [Ring <1>]
6555	13367 [Element <1>] [Ring <1>]
6560	13367 [Element <1>] [Ring <1>]
6561	13367 [Element <1>] [Ring <1>]

```
      6562 13367 [Element <1>] [Ring <1>]
      6563 13367 [Element <1>] [Ring <1>]
      6564 13367 [Element <1>] [Ring <1>]
      6565 13367 [Element <1>] [Ring <1>]
      6566 13367 [Element <1>] [Ring <1>]
      6568 13367 [Element <1>] [Ring <1>]
      6575 13367 [Element <1>] [Ring <1>]
      6583 13367 [Element <1>] [Ring <1>]
      6584 13367 [Element <1>] [Ring <1>]
      6660 13367 [Element <1>] [Ring <1>]
      6661 13367 [Element <1>] [Ring <1>]
      6662 13367 [Element <1>] [Ring <1>]
      6667 13367 [Element <1>] [Ring <1>]
      6668 13367 [Element <1>] [Ring <1>]
      6671 13367 [Element <1>] [Ring <1>]
      6604 13367 [Element <1>] [Ring <1>]
      6605 13367 [Element <1>] [Ring <1>]
      6607 13367 [Element <1>] [Ring <1>]
      9336 13348 [Element <1>] [Ring <1>]
      9324 13356 [Element <1>] [Coordinate <41>][Ring <1>]
      9325 13367 [Element <1>] [Ring <1>]
      9335 13356 [Element <1>] [Coordinate <372>][Ring <1>]
      9333 13367 [Element <1>] [Ring <1>]
      9327 13367 [Element <1>] [Ring <1>]
      9334 13348 [Element <1>] [Ring <1>]
      9326 13348 [Element <1>] [Ring <1>]
      9329 13348 [Element <1>] [Ring <1>]
      9328 13348 [Element <1>] [Ring <1>]
      9332 13367 [Element <1>] [Ring <1>]
      9331 13348 [Element <1>] [Ring <1>]
      9340 13367 [Element <1>] [Ring <1>]
      9338 13367 [Element <1>] [Ring <1>]
      9339 13367 [Element <1>] [Ring <1>]
      9337 13367 [Element <1>] [Ring <1>]
      9323 13367 [Element <1>] [Ring <1>]
      9322 13367 [Element <1>] [Ring <1>]
      9330 13367 [Element <1>] [Ring <1>]
      9813 13367 [Element <1>] [Ring <1>]
      9814 13367 [Element <1>] [Ring <1>]
      9812 13367 [Element <1>] [Ring <1>]

114 rows selected.

Elapsed: 00:00:04.46
SQL> select count(*),g.POLYGOON.SDO_GTYPE from NWG_ZONE_EFN g group by g.POLYGOON.SDO_GTYPE order by
g.POLYGOON.SDO_GTYPE;

  COUNT(*) POLYGOON.SDO_GTYPE
---------- -----------------
      5614              2003
        43              2007

13348, 00000, "polygon boundary is not closed"
// *Cause: The boundary of a polygon does not close.
// *Action: Alter the coordinate values or the definition of the
//          SDO_GTYPE or SDO_ETYPE attribute of the geometry.

13349, 00000, "polygon boundary crosses itself"
// *Cause: The boundary of a polygon intersects itself.
// *Action: Correct the geometric definition of the object.

13351, 00000, "two or more rings of a complex polygon overlap"
// *Cause: The inner or outer rings of a complex polygon
//          overlap.
// *Action: All rings of a complex polygon must be disjoint.
//          Correct the geometric definition of the object.

13356, 00000, "adjacent points in a geometry are redundant"
// *Cause: There are repeated points in the sequence of coordinates.
// *Action: Remove the redundant point.

13367, 00000, "wrong orientation for interior/exterior rings"
// *Cause: In an Oracle Spatial geometry, the exterior and/or interior rings are not
//         oriented correctly.
// *Action: Be sure that the exterior rings are oriented counterclockwise and the
     //          interior rings are oriented clockwise.
```

B3.06 Oracle script voor verwijderen incorrecte geometrie

PL/SQL script voor het verwijderen van geometries uit een tabel die als ‘invalid’ worden aangemerkt door de Oracle ‘`sdo_geom.validate_geometry_with_context`’ validatie functie.

```
-- delete_invgeom.sql  13-08-2012  TT
-- Script to delete records from table containing invalid geometries

spool delete_invgeom.log
set lines 120
set pages 500
set trim on
set trimspool on
set serveroutput on
set verify off
set timing on
set echo on

-- Tables with geometry errors:
-- GEB_ZONE_EFEMERIDEN  GGA_GEMEENTE_EFEMERIDEN  NWG_ZONE_EFN

define BASETABLE = "NWG_ZONE_EFN"
define GEOM_ATTR = "POLYGOON"

DECLARE
  n        pls_integer := 0;
  status   varchar2(200);
  dim_info sdo_dim_array;
BEGIN
  dbms_output.put_line('      ');
  select diminfo into dim_info from USER_SDO_GEOM_METADATA
    where table_name = '&&BASETABLE' and column_name = '&&GEOM_ATTR';

  for rec in
    (select rowid, &&GEOM_ATTR from &&BASETABLE)
  loop
    status := sdo_geom.validate_geometry_with_context(rec.&&GEOM_ATTR, dim_info);
    if status <> 'TRUE' then
      n := n + 1;
      dbms_output.put_line('Invalid '||n||': '||status);
      delete from &&BASETABLE where rowid = rec.rowid;
    end if;
  end loop;
  commit;

  dbms_output.put_line('      ');
  dbms_output.put_line('Rows deleted: '||n||' (from table &&BASETABLE, geometry column
&&GEOM_ATTR)');
END;
/

spool off
exit;
```

B3.07 Oracle script voor configureren ruimtelijke metadata

Script voor het correct instellen van de SRS (naar RD: 28992), het opgeven van de ruimtelijke Oracle metadata (in USER_SDO_GEOM_METADATA) en het goed zetten van de Z-coordinaat (naar NULL) van 2D punten.

```
-- meta.sql  16-08-2012  TT
-- Set SRID of spatial columns, set Z to NULL (for 2D points), add spatial metadata
-- for Oracle test data

spool meta.log
set lines 130
set pages 500
set trim on
set trimspool on
set serveroutput on
set verify off
set timing on
set echo on

define SRID = "28992"
define TOL  = "0.0005"
define XMIN = "-50000"
define XMAX = "300000"
define YMIN = "250000"
define YMAX = "650000"

update GEB_ZONE_EFEMERIDEN g           set g.POLYGOON.sdo_srid = &&SRID;
update GGA_GEMEENTE_EFEMERIDEN g       set g.POLYGOON.sdo_srid = &&SRID;
update NWG_ZONE_EFN g                  set g.POLYGOON.sdo_srid = &&SRID;
update NWS_SPOORHECTOPUNT_EFEMERIDEN g set g.PUNT.sdo_srid = &&SRID, g.PUNT.sdo_point.z = NULL;
update NWS_SPOOROVERSTEEK_EFEMERIDEN g set g.PUNT.sdo_srid = &&SRID, g.PUNT.sdo_point.z = NULL;
update NWS_SVK_EFEMERIDEN g            set g.LIJN.sdo_srid = &&SRID;
update NWS_TREINSTATION_EFEMERIDEN g   set g.PUNT.sdo_srid = &&SRID, g.PUNT.sdo_point.z = NULL;
update NWV_KM_MARKERING_EFEMERIDEN g   set g.PUNT.sdo_srid = &&SRID, g.PUNT.sdo_point.z = NULL;
update NWV_VAARWEGVAK_EFEMERIDEN g     set g.LIJN.sdo_srid = &&SRID;
update NWW_HECTOPUNTEN g               set g.PUNT.sdo_srid = &&SRID, g.PUNT.sdo_point.z = NULL;
update NWW_WEGVAK_EFEMERIDEN g         set g.LIJN.sdo_srid = &&SRID;
commit;

delete from USER_SDO_GEOM_METADATA where table_name = 'GEB_ZONE_EFEMERIDEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'GGA_GEMEENTE_EFEMERIDEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWG_ZONE_EFN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWS_SPOORHECTOPUNT_EFEMERIDEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWS_SPOOROVERSTEEK_EFEMERIDEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWS_SVK_EFEMERIDEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWS_TREINSTATION_EFEMERIDEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWV_KM_MARKERING_EFEMERIDEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWV_VAARWEGVAK_EFEMERIDEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWW_HECTOPUNTEN';
delete from USER_SDO_GEOM_METADATA where table_name = 'NWW_WEGVAK_EFEMERIDEN';

insert into USER_SDO_GEOM_METADATA values ('GEB_ZONE_EFEMERIDEN','POLYGOON',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('GGA_GEMEENTE_EFEMERIDEN','POLYGOON',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('NWG_ZONE_EFN','POLYGOON',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('NWS_SPOORHECTOPUNT_EFEMERIDEN','PUNT',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('NWS_SPOOROVERSTEEK_EFEMERIDEN','PUNT',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('NWS_SVK_EFEMERIDEN','LIJN',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('NWS_TREINSTATION_EFEMERIDEN','PUNT',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
```

```
insert into USER_SDO_GEOM_METADATA values ('NWV_KM_MARKERING_EFEMERIDEN','PUNT',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('NWV_VAARWEGVAK_EFEMERIDEN','LIJN',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('NWW_HECTOPUNTEN','PUNT',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
insert into USER_SDO_GEOM_METADATA values ('NWW_WEGVAK_EFEMERIDEN','LIJN',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
commit;

spool off
exit;
```

B3.08 Oracle ‘create table’ script

```
-- create_tables.sql  16-08-2012  TT
-- Script to create Oracle tables used in testing

spool create_tables.log
set lines 80
set pages 500
set trim on
set trimspool on
set echo on

drop table geb_so_kruispunten purge;
drop table geb_so_wegvakken purge;
drop table geb_structuren purge;
drop table geb_structuurelementen purge;
drop table geb_zones purge;
drop table geb_zonesoorten purge;
drop table geb_zone_efemeriden purge;
drop table geb_zone_zone_relaties purge;
drop table gga_gemeenten purge;
drop table gga_gemeente_efemeriden purge;
drop table gga_provincies purge;
drop table gga_ref_codes purge;
drop table gga_straten purge;
drop table gga_woonplaatsen purge;
drop table nwa_districten purge;
drop table nwa_regionale_diensten purge;
drop table nwg_structuren purge;
drop table nwg_structuurelementen purge;
drop table nwg_zones purge;
drop table nwg_zonesoorten purge;
drop table nwg_zone_efn purge;
drop table nwg_zone_zone_relaties purge;
drop table nwk_spoorwegovergangen purge;
drop table nwk_spoorwegovergang_efn purge;
drop table nwk_svk_wvk_kruisingen purge;
drop table nws_spoorbaanvakken purge;
drop table nws_spoorhectopunt_efemeriden purge;
drop table nws_spoorjuncties purge;
drop table nws_spoorjunctie_efemeriden purge;
drop table nws_spooroversteek_efemeriden purge;
drop table nws_spoorvakken purge;
drop table nws_svk_efemeriden purge;
drop table nws_svk_smt_efemeriden purge;
drop table nws_treinstations purge;
drop table nws_treinstation_efemeriden purge;
drop table nws_tsn_svk_efemeriden purge;
drop table nwv_gebruikers purge;
drop table nwv_km_markering_efemeriden purge;
drop table nwv_markeringsoorten purge;
drop table nwv_meetlatten purge;
drop table nwv_vaarroutes purge;
drop table nwv_vaarroute_efemeriden purge;
drop table nwv_vaarwegen purge;
drop table nwv_vaarwegjuncties purge;
drop table nwv_vaarwegjunctie_efemeriden purge;
drop table nwv_vaarwegvakken purge;
drop table nwv_vaarwegvak_efemeriden purge;
drop table nwv_vaarweg_efemeriden purge;
drop table nww_baansoorten purge;
drop table nww_baansubsoorten purge;
drop table nww_geometriebronnen purge;
drop table nww_hectopunten purge;
drop table nww_hecto_intervallen purge;
drop table nww_juncties purge;
drop table nww_relatieve_posities purge;
drop table nww_routes purge;
drop table nww_verkeersbaan_efemeriden purge;
drop table nww_verkeersbanen purge;
drop table nww_wegbeheerders purge;
drop table nww_wegvakken purge;
drop table nww_wegvak_efemeriden purge;
drop table nww_wegvak_in_routes purge;
drop table nww_wegvak_relaties purge;
```

```
create table geb_so_kruispunten
(
 id number(7) not null
);

create table geb_so_wegvakken
(
 id number(7) not null
);

create table geb_structuren
(
 id           number(2)    not null,
 naam         varchar2(20),
 omschrijving varchar2(40) not null
);

create table geb_structuurelementen
(
 str_id   number(2)   not null,
 znt_code varchar2(2) not null,
 niveau   number(1)   not null
);

create table geb_zones
(
 znt_code varchar2(2)  not null,
 code     varchar2(8)  not null,
 naam     varchar2(40) not null
);

create table geb_zonesoorten
(
 code               varchar2(2)  not null,
 naam               varchar2(20),
 omschrijving       varchar2(40) not null,
 type               number(1)    not null,
 overlap_indicator  varchar2(1)  not null,
 volgorde_indicator varchar2(1)  not null
);

create table geb_zone_efemeriden
(
 zne_znt_code varchar2(2) not null,
 zne_code     varchar2(8) not null,
 begindatum   date        not null,
 einddatum    date,
 sde_id       number(38)  not null,
 polygoon     sdo_geometry
);

create table geb_zone_zone_relaties
(
 zne_znt_code     varchar2(2) not null,
 zne_code         varchar2(8) not null,
 zne_znt_code_sub varchar2(2) not null,
 zne_code_sub     varchar2(8) not null,
 volgnummer       number(3)
);

create table gga_gemeenten
(
 id                number(4)    not null,
 naam              varchar2(24) not null,
 gme_id_in_vln     number(3),
 einddatum_dialoog date
);

create table gga_gemeente_efemeriden
(
 gme_id     number(4)   not null,
 begindatum date        not null,
 einddatum  date,
 pve_code   varchar2(2) not null,
 sde_id     number(38)  not null,
```

```
 polygoon   sdo_geometry
);

create table gga_provincies
(
 code varchar2(2)  not null,
 id   number(2)    not null,
 naam varchar2(24) not null
);

create table gga_ref_codes
(
 rv_low_value    varchar2(240) not null,
 rv_high_value   varchar2(240),
 rv_abbreviation varchar2(240),
 rv_domain       varchar2(100) not null,
 rv_meaning      varchar2(240),
 rv_type         varchar2(10)
);

create table gga_straten
(
 id             number(8)    not null,
 gme_id         number(4)    not null,
 wps_id         number(4)    not null,
 stt_type       varchar2(1)  not null,
 naam           varchar2(29) not null,
 naam_officieel varchar2(43),
 naam_ptt       varchar2(17),
 naamafkorting  varchar2(5)
);

create table gga_woonplaatsen
(
 id            number(4)    not null,
 naam_nen      varchar2(24) not null,
 naam_ptt      varchar2(18),
 naamafkorting varchar2(4)
);

create table nwa_districten
(
 id         number(38)   not null,
 nummer     number(3)    not null,
 rdt_code   varchar2(2)  not null,
 naam       varchar2(40) not null,
 begindatum date         not null,
 einddatum  date
);

create table nwa_regionale_diensten
(
 code       varchar2(2)  not null,
 naam       varchar2(40) not null,
 begindatum date         not null,
 einddatum  date
);

create table nwg_structuren
(
 id           number(38)   not null,
 omschrijving varchar2(40) not null
);

create table nwg_structuurelementen
(
 str_id   number(38)  not null,
 znt_code varchar2(2) not null
);

create table nwg_zones
(
 znt_code varchar2(2)  not null,
 code     varchar2(8)  not null,
 naam     varchar2(40) not null
);
```

```
create table nwg_zonesoorten
(
 code            varchar2(2)  not null,
 omschrijving    varchar2(40) not null,
 ind_overlap     varchar2(1)  not null,
 ind_volgorde    varchar2(1)  not null,
 ind_aangrenzend varchar2(1)  not null,
 niveau          number(1)    not null
);

create table nwg_zone_efn
(
 id           number(38)  not null,
 zne_znt_code varchar2(2) not null,
 zne_code     varchar2(8) not null,
 begindatum   date not null,
 einddatum    date,
 polygoon     sdo_geometry
);

create table nwg_zone_zone_relaties
(
 zne_znt_code     varchar2(2) not null,
 zne_code         varchar2(8) not null,
 zne_znt_code_sub varchar2(2) not null,
 zne_code_sub     varchar2(8) not null,
 begindatum       date not null,
 einddatum        date,
 volgnummer       number(3)
);

create table nwk_spoorwegovergangen
(
 id number(38) not null
);

create table nwk_spoorwegovergang_efn
(
 id             number(38)  not null,
 sog_id         number(38)  not null,
 begindatum     date        not null,
 kruisingswijze varchar2(1) not null,
 einddatum      date
);

create table nwk_svk_wvk_kruisingen
(
 id         number(38) not null,
 svk_id     number(38) not null,
 wvk_id     number(38) not null,
 sog_id     number(38) not null,
 begindatum date       not null,
 einddatum  date
);

create table nws_spoorbaanvakken
(
 id   number(8)    not null,
 naam varchar2(60) not null
);

create table nws_spoorhectopunt_efemeriden
(
 svk_id         number(8)   not null,
 hectometrering number(4,1) not null,
 begindatum     date        not null,
 afstand        number(9,3) not null,
 einddatum      date,
 sde_id         number(38)  not null,
 punt           sdo_geometry
);

create table nws_spoorjuncties
(
 id number(8) not null
```

```
);

create table nws_spoorjunctie_efemeriden
(
 sje_id      number(8) not null,
 begindatum  date      not null,
 xcoordinaat number(9,3),
 ycoordinaat number(9,3),
 einddatum   date
);

create table nws_spooroversteek_efemeriden
(
 id            number(8)   not null,
 begindatum    date        not null,
 svk_id        number(8)   not null,
 oversteektype varchar2(1) not null,
 afstand       number(9,3) not null,
 einddatum     date,
 sde_id        number(38)  not null,
 punt          sdo_geometry
);

create table nws_spoorvakken
(
 id           number(8) not null,
 sje_id_begin number(8) not null,
 sje_id_eind  number(8) not null
);

create table nws_svk_efemeriden
(
 svk_id                number(8)   not null,
 begindatum            date        not null,
 gon_id                number(2)   not null,
 sbk_id                number(8),
 spooraantalklasse     varchar2(1) not null,
 vervoerssubjectklasse varchar2(1) not null,
 einddatum             date,
 sde_id                number(38)  not null,
 bronjaar              number(4),
 lijn                  sdo_geometry
);

create table nws_svk_smt_efemeriden
(
 svk_id             number(8)   not null,
 beginafstand       number(9,3) not null,
 begindatum         date        not null,
 eindafstand        number(9,3) not null,
 beginkilometrering number(6,3) not null,
 eindkilometrering  number(6,3) not null,
 einddatum          date
);

create table nws_treinstations
(
 id number(8) not null
);

create table nws_treinstation_efemeriden
(
 tsn_id      number(8)    not null,
 begindatum  date         not null,
 naam        varchar2(60) not null,
 xcoordinaat number(9,3)  not null,
 ycoordinaat number(9,3)  not null,
 sde_id      number(38)   not null,
 punt        sdo_geometry,
 einddatum   date
);

create table nws_tsn_svk_efemeriden
(
 tsn_id     number(8) not null,
 svk_id     number(8) not null,
```

```
 begindatum date      not null,
 einddatum  date
);

create table nwv_gebruikers
(
 naam       varchar2(30) not null,
 dt_netwerk date,
 vin_rol    varchar2(1),
 omschrijv  varchar2(50),
 in_dis_vwl varchar2(1)  not null,
 in_dis_kmg varchar2(1)  not null,
 in_ext_vin varchar2(1)  not null
);

create table nwv_km_markering_efemeriden
(
 vwk_id     number(10)  not null,
 begdtm     date        not null,
 afstand    number(5)   not null,
 pos_tov_as varchar2(1) not null,
 gtlwaarde  number(5,1) not null,
 ltrwaarde  varchar2(1),
 enddtm     date,
 mst_code   varchar2(1) not null,
 sde_id     number(38)  not null,
 punt       sdo_geometry
);

create table nwv_markeringsoorten
(
 code varchar2(1)  not null,
 naam varchar2(25) not null
);

create table nwv_meetlatten
(
 vwk_id number(10)  not null,
 begdtm date        not null,
 enddtm date,
 begkm  number(6,3) not null,
 endkm  number(6,3) not null
);

create table nwv_vaarroutes
(
 id   number(10)  not null,
 code varchar2(6) not null
);

create table nwv_vaarroute_efemeriden
(
 vrt_id    number(10)    not null,
 begdtm    date          not null,
 enddtm    date,
 vrtnaam   varchar2(240) not null,
 waternaam varchar2(240)
);

create table nwv_vaarwegen
(
 vrt_id  number(10)   not null,
 nr       number(3)   not null,
 richting varchar2(1) not null
);

create table nwv_vaarwegjuncties
(
 id number(10) not null
);

create table nwv_vaarwegjunctie_efemeriden
(
 vwj_id number(10) not null,
 begdtm date       not null,
 enddtm date,
```

```
 xcoord number(6)  not null,
 ycoord number(6)  not null
);

create table nwv_vaarwegvakken
(
 id         number(10) not null,
 vwj_id_beg number(10) not null,
 vwj_id_end number(10) not null
);

create table nwv_vaarwegvak_efemeriden
(
 vwk_id    number(10)  not null,
 begdtm    date        not null,
 enddtm    date,
 vrt_id    number(10)  not null,
 vwg_nr    number(3)   not null,
 richting  varchar2(1) not null,
 vaktype   varchar2(1) not null,
 vaklengte number(6,3) not null,
 gon_id    number(2)   not null,
 bronjaar  number(4),
 sde_id    number(38)  not null,
 lijn      sdo_geometry
);

create table nwv_vaarweg_efemeriden
(
 vrt_id  number(10)    not null,
 vwg_nr  number(3)     not null,
 begdtm  date          not null,
 enddtm  date,
 vwgnaam varchar2(240) not null
);

create table nww_baansoorten
(
 code varchar2(2)  not null,
 naam varchar2(30) not null
);

create table nww_baansubsoorten
(
 code     varchar2(3)  not null,
 naam     varchar2(60) not null,
 brt_code varchar2(2)  not null
);

create table nww_geometriebronnen
(
 id   number(2)    not null,
 naam varchar2(60) not null
);

create table nww_hectopunten
(
 wee_wvk_id     number(10) not null,
 wee_begindatum date       not null,
 hectometrering number(4)  not null,
 afstand        number(5)  not null,
 einddatum      date,
 sde_id         number(38) not null,
 punt           sdo_geometry
);

create table nww_hecto_intervallen
(
 wvk_id              number(10)  not null,
 begindatum          date        not null,
 beginafstand        number(5)   not null,
 eindafstand         number(5)   not null,
 beginkilometrering  number(6,3) not null,
 eindkilometrering   number(6,3) not null,
 baanpositie_tov_wol varchar2(1)
);
```

```
create table nww_juncties
(
 id              number(10) not null,
 sot_id          number(7),
 so_mutatiedatum date
);

create table nww_relatieve_posities
(
 code varchar2(2)  not null,
 naam varchar2(60) not null
);

create table nww_routes
(
 id          number(5)   not null,
 routeletter varchar2(1) not null,
 routenummer number(3)   not null
);

create table nww_verkeersbaan_efemeriden
(
 einddatum  date,
 vkn_id     number(7)    not null,
 begindatum date         not null,
 herkomst   varchar2(20) not null,
 bestemming varchar2(20) not null
);

create table nww_verkeersbanen
(
 id number(7) not null
);

create table nww_wegbeheerders
(
 id    number(4)    not null,
 soort varchar2(1)  not null,
 code  varchar2(4)  not null,
 naam  varchar2(35) not null
);

create table nww_wegvakken
(
 id                    number(10) not null,
 sot_id                number(7),
 jte_id_eind           number(10) not null,
 sok_id                number(7),
 so_mutatiedatum       date,
 gedwongen_rijrichting varchar2(1),
 vkn_id_geheel         number(7),
 vkn_id_links          number(7),
 vkn_id_rechts         number(7),
 jte_id_begin          number(10) not null
);

create table nww_wegvak_efemeriden
(
 wvk_id                 number(10)  not null,
 begindatum             date        not null,
 einddatum              date,
 jte_id_begin           number(10)  not null,
 jte_id_eind            number(10)  not null,
 wegbeheerdersoort      varchar2(1) not null,
 wbr_id                 number(4),
 wegnummer              varchar2(5),
 wegdeelletter          varchar2(1),
 hectometreringsletter  varchar2(1),
 bst_code               varchar2(3),
 rpe_code               varchar2(2),
 rijrichting            varchar2(1),
 stt_id                 number(8),
 gme_id                 number(4),
 hnrstr_links           varchar2(1),
 hnrstr_rechts          varchar2(1),
```

```
 e_hnr_links            number(5),
 e_hnr_rechts           number(5),
 l_hnr_links            number(5),
 l_hnr_rechts           number(5),
 gon_id                 number(2)   not null,
 bronjaar               number(4),
 klok_begin             number(2)   not null,
 klok_eind              number(2)   not null,
 admin_richting         varchar2(1),
 verkeersbaan_indicator varchar2(1),
 relatief_baanvolgnr    number(1),
 type_wijziging         varchar2(1),
 wegtype_code           varchar2(2),
 sde_id                 number(38)  not null,
 lijn                   sdo_geometry,
 orientatierichting_bps varchar2(1)
);

create table nww_wegvak_in_routes
(
 wee_wvk_id     number(9) not null,
 wee_begindatum date      not null,
 rte_id         number(5) not null
);

create table nww_wegvak_relaties
(
 wvk_id_opvolger   number(10) not null,
 wvk_id_voorganger number(10) not null
);

spool off
exit;
```

B3.09 Oracle ‘create index’ script

```
-- create_indexes.sql  23-08-2012  TT
-- Create Oracle indexes for test data tables

spool create_indexes.log
set lines 130
set pages 500
set trim on
set trimspool on
set serveroutput on
set verify off
set timing on
set echo on

alter table GGA_GEMEENTEN           drop primary key;
alter table GGA_GEMEENTE_EFEMERIDEN drop primary key;
alter table GGA_STRATEN             drop primary key;
alter table NWG_ZONE_EFN            drop primary key;
alter table NWW_WEGVAK_EFEMERIDEN   drop primary key;

drop index GGA_GEMEENTEN_NAAM_IDX;
drop index GGA_GEM_EFN_BEGDAT_IDX;
drop index GGA_GEM_EFN_ENDDAT_IDX;
drop index NWG_ZONE_EFN_ZNE_ZNT_IDX;
drop index NWG_ZONE_EFN_ZNE_CODE_IDX;
drop index NWG_ZONE_EFN_BEGDAT_IDX;
drop index NWG_ZONE_EFN_ENDDAT_IDX;
drop index NWG_ZZREL_ZNE_ZNT_COD_IDX;
drop index NWG_ZZREL_ZNE_CODE_IDX;
drop index NWG_ZZREL_ZZ_CODE_SUB_IDX;
drop index NWG_ZZREL_ZNE_COD_SUB_IDX;
drop index NWG_ZZREL_BEGDAT_IDX;
drop index NWG_ZZREL_ENDDAT_IDX;
drop index NWW_WEGVAK_EFN_BEGDAT_IDX;
drop index NWW_WEGVAK_EFN_ENDDAT_IDX;
drop index NWW_WEGVAK_EFN_WEGBEH_IDX;
drop index NWW_WEGVAK_EFN_STT_ID_IDX;

drop index GEB_ZONE_EFEMERIDEN_SIDX     force;
drop index GGA_GEMEENTE_EFEMERIDEN_SIDX force;
drop index NWG_ZONE_EFN_SIDX            force;
drop index NWS_SPOORHECTOPUNT_EFN_SIDX  force;
drop index NWS_SPOOROVERSTEEK_EFN_SIDX  force;
drop index NWS_SVK_EFEMERIDEN_SIDX      force;
drop index NWS_TREINSTATION_EFN_SIDX    force;
drop index NWV_KM_MARKERING_EFN_SIDX    force;
drop index NWV_VAARWEGVAK_EFN_SIDX      force;
drop index NWW_HECTOPUNTEN_SIDX         force;
drop index NWW_WEGVAK_EFEMERIDEN_SIDX   force;
drop index PROVINCIES_SIDX              force;

-- -------------------------------------------------

alter table GGA_GEMEENTEN           add primary key (ID)                using index tablespace INDX
enable;
alter table GGA_GEMEENTE_EFEMERIDEN add primary key (GME_ID,BEGINDATUM) using index tablespace INDX
enable;
alter table GGA_STRATEN             add primary key (ID)                using index tablespace INDX
enable;
alter table NWG_ZONE_EFN            add primary key (ID)                using index tablespace INDX
enable;
alter table NWW_WEGVAK_EFEMERIDEN   add primary key (WVK_ID,BEGINDATUM) using index tablespace INDX
enable;

create index GGA_GEMEENTEN_NAAM_IDX    on GGA_GEMEENTEN           (NAAM)              tablespace
INDX;
create index GGA_GEM_EFN_BEGDAT_IDX    on GGA_GEMEENTE_EFEMERIDEN (BEGINDATUM)        tablespace
INDX;
create index GGA_GEM_EFN_ENDDAT_IDX    on GGA_GEMEENTE_EFEMERIDEN (EINDDATUM)         tablespace
INDX;
create index NWG_ZONE_EFN_ZNE_ZNT_IDX  on NWG_ZONE_EFN            (ZNE_ZNT_CODE)      tablespace
INDX;
create index NWG_ZONE_EFN_ZNE_CODE_IDX on NWG_ZONE_EFN            (ZNE_CODE)          tablespace
INDX;
```

```
create index NWG_ZONE_EFN_BEGDAT_IDX   on NWG_ZONE_EFN            (BEGINDATUM)        tablespace
INDX;
create index NWG_ZONE_EFN_ENDDAT_IDX   on NWG_ZONE_EFN            (EINDDATUM)         tablespace
INDX;
create index NWG_ZZREL_ZNE_ZNT_COD_IDX on NWG_ZONE_ZONE_RELATIES  (ZNE_ZNT_CODE)      tablespace
INDX;
create index NWG_ZZREL_ZNE_CODE_IDX    on NWG_ZONE_ZONE_RELATIES  (ZNE_CODE)          tablespace
INDX;
create index NWG_ZZREL_ZZ_CODE_SUB_IDX on NWG_ZONE_ZONE_RELATIES  (ZNE_ZNT_CODE_SUB)  tablespace
INDX;
create index NWG_ZZREL_ZNE_COD_SUB_IDX on NWG_ZONE_ZONE_RELATIES  (ZNE_CODE_SUB)      tablespace
INDX;
create index NWG_ZZREL_BEGDAT_IDX      on NWG_ZONE_ZONE_RELATIES  (BEGINDATUM)        tablespace
INDX;
create index NWG_ZZREL_ENDDAT_IDX      on NWG_ZONE_ZONE_RELATIES  (EINDDATUM)         tablespace
INDX;
create index NWW_WEGVAK_EFN_BEGDAT_IDX on NWW_WEGVAK_EFEMERIDEN   (BEGINDATUM)        tablespace
INDX;
create index NWW_WEGVAK_EFN_ENDDAT_IDX on NWW_WEGVAK_EFEMERIDEN   (EINDDATUM)         tablespace
INDX;
create index NWW_WEGVAK_EFN_WEGBEH_IDX on NWW_WEGVAK_EFEMERIDEN   (WEGBEHEERDERSOORT) tablespace
INDX;
create index NWW_WEGVAK_EFN_STT_ID_IDX on NWW_WEGVAK_EFEMERIDEN   (STT_ID)            tablespace
INDX;

create index GEB_ZONE_EFEMERIDEN_SIDX on GEB_ZONE_EFEMERIDEN (POLYGOON)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX sdo_indx_dims=2');

create index GGA_GEMEENTE_EFEMERIDEN_SIDX on GGA_GEMEENTE_EFEMERIDEN (POLYGOON)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX sdo_indx_dims=2');

create index NWG_ZONE_EFN_SIDX on NWG_ZONE_EFN (POLYGOON)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX sdo_indx_dims=2');

create index NWS_SPOORHECTOPUNT_EFN_SIDX on NWS_SPOORHECTOPUNT_EFEMERIDEN (PUNT)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX layer_gtype=point sdo_indx_dims=2');

create index NWS_SPOOROVERSTEEK_EFN_SIDX on NWS_SPOOROVERSTEEK_EFEMERIDEN (PUNT)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX layer_gtype=point sdo_indx_dims=2');

create index NWS_SVK_EFEMERIDEN_SIDX on NWS_SVK_EFEMERIDEN (LIJN)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX layer_gtype=line sdo_indx_dims=2');

create index NWS_TREINSTATION_EFN_SIDX on NWS_TREINSTATION_EFEMERIDEN (PUNT)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX layer_gtype=point sdo_indx_dims=2');

create index NWV_KM_MARKERING_EFN_SIDX on NWV_KM_MARKERING_EFEMERIDEN (PUNT)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX layer_gtype=point sdo_indx_dims=2');

create index NWV_VAARWEGVAK_EFN_SIDX on NWV_VAARWEGVAK_EFEMERIDEN (LIJN)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX layer_gtype=line sdo_indx_dims=2');

create index NWW_HECTOPUNTEN_SIDX on NWW_HECTOPUNTEN (PUNT)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX layer_gtype=point sdo_indx_dims=2');

create index NWW_WEGVAK_EFEMERIDEN_SIDX on NWW_WEGVAK_EFEMERIDEN (LIJN)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX layer_gtype=line sdo_indx_dims=2');

create index PROVINCIES_SIDX on PROVINCIES (POLYGOON)
  indextype is mdsys.spatial_index parameters ('tablespace=INDX sdo_indx_dims=2');

spool off
exit;
```

B3.10 PostgreSQL/PostGIS ‘create table’ script

```
-- create_tables.sql  23-08-2012  TT
-- Script to create PostgreSQL/PostGIS tables used in testing
-- Compared to Oracle:
-- varchar2->varchar, number->numeric, date->timestamp,
--   sdo_geometry->geometry(GEOMETRY,28992)

drop table geb_so_kruispunten;
drop table geb_so_wegvakken;
drop table geb_structuren;
drop table geb_structuurelementen;
drop table geb_zones;
drop table geb_zonesoorten;
drop table geb_zone_efemeriden;
drop table geb_zone_zone_relaties;
drop table gga_gemeenten;
drop table gga_gemeente_efemeriden;
drop table gga_provincies;
drop table gga_ref_codes;
drop table gga_straten;
drop table gga_woonplaatsen;
drop table nwa_districten;
drop table nwa_regionale_diensten;
drop table nwg_structuren;
drop table nwg_structuurelementen;
drop table nwg_zones;
drop table nwg_zonesoorten;
drop table nwg_zone_efn;
drop table nwg_zone_zone_relaties;
drop table nwk_spoorwegovergangen;
drop table nwk_spoorwegovergang_efn;
drop table nwk_svk_wvk_kruisingen;
drop table nws_spoorbaanvakken;
drop table nws_spoorhectopunt_efemeriden;
drop table nws_spoorjuncties;
drop table nws_spoorjunctie_efemeriden;
drop table nws_spooroversteek_efemeriden;
drop table nws_spoorvakken;
drop table nws_svk_efemeriden;
drop table nws_svk_smt_efemeriden;
drop table nws_treinstations;
drop table nws_treinstation_efemeriden;
drop table nws_tsn_svk_efemeriden;
drop table nwv_gebruikers;
drop table nwv_km_markering_efemeriden;
drop table nwv_markeringsoorten;
drop table nwv_meetlatten;
drop table nwv_vaarroutes;
drop table nwv_vaarroute_efemeriden;
drop table nwv_vaarwegen;
drop table nwv_vaarwegjuncties;
drop table nwv_vaarwegjunctie_efemeriden;
drop table nwv_vaarwegvakken;
drop table nwv_vaarwegvak_efemeriden;
drop table nwv_vaarweg_efemeriden;
drop table nww_baansoorten;
drop table nww_baansubsoorten;
drop table nww_geometriebronnen;
drop table nww_hectopunten;
drop table nww_hecto_intervallen;
drop table nww_juncties;
drop table nww_relatieve_posities;
drop table nww_routes;
drop table nww_verkeersbaan_efemeriden;
drop table nww_verkeersbanen;
drop table nww_wegbeheerders;
drop table nww_wegvakken;
drop table nww_wegvak_efemeriden;
drop table nww_wegvak_in_routes;
drop table nww_wegvak_relaties;
drop table provincies;

-- ------------------------------------------------------------
```

```
create table geb_so_kruispunten
(
 id numeric(7) not null
);

create table geb_so_wegvakken
(
 id numeric(7) not null
);

create table geb_structuren
(
 id           numeric(2)  not null,
 naam         varchar(20),
 omschrijving varchar(40) not null
);

create table geb_structuurelementen
(
 str_id   numeric(2) not null,
 znt_code varchar(2) not null,
 niveau   numeric(1) not null
);

create table geb_zones
(
 znt_code varchar(2)  not null,
 code     varchar(8)  not null,
 naam     varchar(40) not null
);

create table geb_zonesoorten
(
 code               varchar(2)  not null,
 naam               varchar(20),
 omschrijving       varchar(40) not null,
 type               numeric(1)  not null,
 overlap_indicator  varchar(1)  not null,
 volgorde_indicator varchar(1)  not null
);

create table geb_zone_efemeriden
(
 zne_znt_code varchar(2)  not null,
 zne_code     varchar(8)  not null,
 begindatum   timestamp   not null,
 einddatum    timestamp,
 sde_id       numeric(38) not null,
 polygoon     geometry(GEOMETRY,28992)
);

create table geb_zone_zone_relaties
(
 zne_znt_code     varchar(2) not null,
 zne_code         varchar(8) not null,
 zne_znt_code_sub varchar(2) not null,
 zne_code_sub     varchar(8) not null,
 volgnummer       numeric(3)
);

create table gga_gemeenten
(
 id                numeric(4)  not null,
 naam              varchar(24) not null,
 gme_id_in_vln     numeric(3),
 einddatum_dialoog timestamp
);

create table gga_gemeente_efemeriden
(
 gme_id     numeric(4)  not null,
 begindatum timestamp   not null,
 einddatum  timestamp,
 pve_code   varchar(2)  not null,
 sde_id     numeric(38) not null,
 polygoon   geometry(GEOMETRY,28992)
```

```
);

create table gga_provincies
(
 code varchar(2)  not null,
 id   numeric(2)  not null,
 naam varchar(24) not null
);

create table gga_ref_codes
(
 rv_low_value    varchar(240) not null,
 rv_high_value   varchar(240),
 rv_abbreviation varchar(240),
 rv_domain       varchar(100) not null,
 rv_meaning      varchar(240),
 rv_type         varchar(10)
);

create table gga_straten
(
 id             numeric(8)  not null,
 gme_id         numeric(4)  not null,
 wps_id         numeric(4)  not null,
 stt_type       varchar(1)  not null,
 naam           varchar(29) not null,
 naam_officieel varchar(43),
 naam_ptt       varchar(17),
 naamafkorting  varchar(5)
);

create table gga_woonplaatsen
(
 id            numeric(4)  not null,
 naam_nen      varchar(24) not null,
 naam_ptt      varchar(18),
 naamafkorting varchar(4)
);

create table nwa_districten
(
 id         numeric(38) not null,
 nummer     numeric(3)  not null,
 rdt_code   varchar(2)  not null,
 naam       varchar(40) not null,
 begindatum timestamp   not null,
 einddatum  timestamp
);

create table nwa_regionale_diensten
(
 code       varchar(2)  not null,
 naam       varchar(40) not null,
 begindatum timestamp   not null,
 einddatum  timestamp
);

create table nwg_structuren
(
 id           numeric(38) not null,
 omschrijving varchar(40) not null
);

create table nwg_structuurelementen
(
 str_id   numeric(38) not null,
 znt_code varchar(2)  not null
);

create table nwg_zones
(
 znt_code varchar(2)  not null,
 code     varchar(8)  not null,
 naam     varchar(40) not null
);
```

```
create table nwg_zonesoorten
(
 code            varchar(2)  not null,
 omschrijving    varchar(40) not null,
 ind_overlap     varchar(1)  not null,
 ind_volgorde    varchar(1)  not null,
 ind_aangrenzend varchar(1)  not null,
 niveau          numeric(1)  not null
);

create table nwg_zone_efn
(
 id           numeric(38) not null,
 zne_znt_code varchar(2)  not null,
 zne_code     varchar(8)  not null,
 begindatum   timestamp   not null,
 einddatum    timestamp,
 polygoon     geometry(GEOMETRY,28992)
);

create table nwg_zone_zone_relaties
(
 zne_znt_code     varchar(2) not null,
 zne_code         varchar(8) not null,
 zne_znt_code_sub varchar(2) not null,
 zne_code_sub     varchar(8) not null,
 begindatum       timestamp  not null,
 einddatum        timestamp,
 volgnummer       numeric(3)
);

create table nwk_spoorwegovergangen
(
 id numeric(38) not null
);

create table nwk_spoorwegovergang_efn
(
 id             numeric(38) not null,
 sog_id         numeric(38) not null,
 begindatum     timestamp   not null,
 kruisingswijze varchar(1)  not null,
 einddatum      timestamp
);

create table nwk_svk_wvk_kruisingen
(
 id         numeric(38) not null,
 svk_id     numeric(38) not null,
 wvk_id     numeric(38) not null,
 sog_id     numeric(38) not null,
 begindatum timestamp   not null,
 einddatum  timestamp
);

create table nws_spoorbaanvakken
(
 id   numeric(8)  not null,
 naam varchar(60) not null
);

create table nws_spoorhectopunt_efemeriden
(
 svk_id         numeric(8)   not null,
 hectometrering numeric(4,1) not null,
 begindatum     timestamp    not null,
 afstand        numeric(9,3) not null,
 einddatum      timestamp,
 sde_id         numeric(38)  not null,
 punt           geometry(POINT,28992)
);

create table nws_spoorjuncties
(
 id numeric(8) not null
);
```

```
create table nws_spoorjunctie_efemeriden
(
 sje_id      numeric(8) not null,
 begindatum  timestamp  not null,
 xcoordinaat numeric(9,3),
 ycoordinaat numeric(9,3),
 einddatum   timestamp
);

create table nws_spooroversteek_efemeriden
(
 id            numeric(8)   not null,
 begindatum    timestamp    not null,
 svk_id        numeric(8)   not null,
 oversteektype varchar(1)   not null,
 afstand       numeric(9,3) not null,
 einddatum     timestamp,
 sde_id        numeric(38)  not null,
 punt          geometry(POINT,28992)
);

create table nws_spoorvakken
(
 id           numeric(8) not null,
 sje_id_begin numeric(8) not null,
 sje_id_eind  numeric(8) not null
);

create table nws_svk_efemeriden
(
 svk_id                numeric(8)  not null,
 begindatum            timestamp   not null,
 gon_id                numeric(2)  not null,
 sbk_id                numeric(8),
 spooraantalklasse     varchar(1)  not null,
 vervoerssubjectklasse varchar(1)  not null,
 einddatum             timestamp,
 sde_id                numeric(38) not null,
 bronjaar              numeric(4),
 lijn                  geometry(LINESTRING,28992)
);

create table nws_svk_smt_efemeriden
(
 svk_id             numeric(8)   not null,
 beginafstand       numeric(9,3) not null,
 begindatum         timestamp    not null,
 eindafstand        numeric(9,3) not null,
 beginkilometrering numeric(6,3) not null,
 eindkilometrering  numeric(6,3) not null,
 einddatum          timestamp
);

create table nws_treinstations
(
 id numeric(8) not null
);

create table nws_treinstation_efemeriden
(
 tsn_id      numeric(8)   not null,
 begindatum  timestamp    not null,
 naam        varchar(60)  not null,
 xcoordinaat numeric(9,3) not null,
 ycoordinaat numeric(9,3) not null,
 sde_id      numeric(38)  not null,
 punt        geometry(POINT,28992),
 einddatum   timestamp
);

create table nws_tsn_svk_efemeriden
(
 tsn_id     numeric(8) not null,
 svk_id     numeric(8) not null,
 begindatum timestamp  not null,
```

```
 einddatum  timestamp
);

create table nwv_gebruikers
(
 naam       varchar(30) not null,
 dt_netwerk timestamp,
 vin_rol    varchar(1),
 omschrijv  varchar(50),
 in_dis_vwl varchar(1)  not null,
 in_dis_kmg varchar(1)  not null,
 in_ext_vin varchar(1)  not null
);

create table nwv_km_markering_efemeriden
(
 vwk_id     numeric(10)  not null,
 begdtm     timestamp    not null,
 afstand    numeric(5)   not null,
 pos_tov_as varchar(1)   not null,
 gtlwaarde  numeric(5,1) not null,
 ltrwaarde  varchar(1),
 enddtm     timestamp,
 mst_code   varchar(1)   not null,
 sde_id     numeric(38)  not null,
 punt       geometry(POINT,28992)
);

create table nwv_markeringsoorten
(
 code varchar(1)  not null,
 naam varchar(25) not null
);

create table nwv_meetlatten
(
 vwk_id numeric(10)  not null,
 begdtm timestamp    not null,
 enddtm timestamp,
 begkm  numeric(6,3) not null,
 endkm  numeric(6,3) not null
);

create table nwv_vaarroutes
(
 id   numeric(10) not null,
 code varchar(6)  not null
);

create table nwv_vaarroute_efemeriden
(
 vrt_id    numeric(10)  not null,
 begdtm    timestamp    not null,
 enddtm    timestamp,
 vrtnaam   varchar(240) not null,
 waternaam varchar(240)
);

create table nwv_vaarwegen
(
 vrt_id  numeric(10) not null,
 nr       numeric(3) not null,
 richting varchar(1) not null
);

create table nwv_vaarwegjuncties
(
 id numeric(10) not null
);

create table nwv_vaarwegjunctie_efemeriden
(
 vwj_id numeric(10) not null,
 begdtm timestamp   not null,
 enddtm timestamp,
 xcoord numeric(6)  not null,
```

```
 ycoord numeric(6)  not null
);

create table nwv_vaarwegvakken
(
 id         numeric(10) not null,
 vwj_id_beg numeric(10) not null,
 vwj_id_end numeric(10) not null
);

create table nwv_vaarwegvak_efemeriden
(
 vwk_id    numeric(10)  not null,
 begdtm    timestamp    not null,
 enddtm    timestamp,
 vrt_id    numeric(10)  not null,
 vwg_nr    numeric(3)   not null,
 richting  varchar(1)   not null,
 vaktype   varchar(1)   not null,
 vaklengte numeric(6,3) not null,
 gon_id    numeric(2)   not null,
 bronjaar  numeric(4),
 sde_id    numeric(38)  not null,
 lijn      geometry(LINESTRING,28992)
);

create table nwv_vaarweg_efemeriden
(
 vrt_id  numeric(10)  not null,
 vwg_nr  numeric(3)   not null,
 begdtm  timestamp    not null,
 enddtm  timestamp,
 vwgnaam varchar(240) not null
);

create table nww_baansoorten
(
 code varchar(2)  not null,
 naam varchar(30) not null
);

create table nww_baansubsoorten
(
 code     varchar(3)  not null,
 naam     varchar(60) not null,
 brt_code varchar(2)  not null
);

create table nww_geometriebronnen
(
 id   numeric(2)  not null,
 naam varchar(60) not null
);

create table nww_hectopunten
(
 wee_wvk_id     numeric(10) not null,
 wee_begindatum timestamp   not null,
 hectometrering numeric(4)  not null,
 afstand        numeric(5)  not null,
 einddatum      timestamp,
 sde_id         numeric(38) not null,
 punt           geometry(POINT,28992)
);

create table nww_hecto_intervallen
(
 wvk_id              numeric(10)  not null,
 begindatum          timestamp    not null,
 beginafstand        numeric(5)   not null,
 eindafstand         numeric(5)   not null,
 beginkilometrering  numeric(6,3) not null,
 eindkilometrering   numeric(6,3) not null,
 baanpositie_tov_wol varchar(1)
);
```

```
create table nww_juncties
(
 id              numeric(10) not null,
 sot_id          numeric(7),
 so_mutatiedatum timestamp
);

create table nww_relatieve_posities
(
 code varchar(2)  not null,
 naam varchar(60) not null
);

create table nww_routes
(
 id          numeric(5) not null,
 routeletter varchar(1) not null,
 routenummer numeric(3) not null
);

create table nww_verkeersbaan_efemeriden
(
 einddatum  timestamp,
 vkn_id     numeric(7)  not null,
 begindatum timestamp   not null,
 herkomst   varchar(20) not null,
 bestemming varchar(20) not null
);

create table nww_verkeersbanen
(
 id numeric(7) not null
);

create table nww_wegbeheerders
(
 id    numeric(4)  not null,
 soort varchar(1)  not null,
 code  varchar(4)  not null,
 naam  varchar(35) not null
);

create table nww_wegvakken
(
 id                    numeric(10) not null,
 sot_id                numeric(7),
 jte_id_eind           numeric(10) not null,
 sok_id                numeric(7),
 so_mutatiedatum       timestamp,
 gedwongen_rijrichting varchar(1),
 vkn_id_geheel         numeric(7),
 vkn_id_links          numeric(7),
 vkn_id_rechts         numeric(7),
 jte_id_begin          numeric(10) not null
);

create table nww_wegvak_efemeriden
(
 wvk_id                 numeric(10) not null,
 begindatum             timestamp   not null,
 einddatum              timestamp,
 jte_id_begin           numeric(10) not null,
 jte_id_eind            numeric(10) not null,
 wegbeheerdersoort      varchar(1)  not null,
 wbr_id                 numeric(4),
 wegnummer              varchar(5),
 wegdeelletter          varchar(1),
 hectometreringsletter  varchar(1),
 bst_code               varchar(3),
 rpe_code               varchar(2),
 rijrichting            varchar(1),
 stt_id                 numeric(8),
 gme_id                 numeric(4),
 hnrstr_links           varchar(1),
 hnrstr_rechts          varchar(1),
 e_hnr_links            numeric(5),
```

```
 e_hnr_rechts           numeric(5),
 l_hnr_links            numeric(5),
 l_hnr_rechts           numeric(5),
 gon_id                 numeric(2)  not null,
 bronjaar               numeric(4),
 klok_begin             numeric(2)  not null,
 klok_eind              numeric(2)  not null,
 admin_richting         varchar(1),
 verkeersbaan_indicator varchar(1),
 relatief_baanvolgnr    numeric(1),
 type_wijziging         varchar(1),
 wegtype_code           varchar(2),
 sde_id                 numeric(38) not null,
 lijn                   geometry(LINESTRING,28992),
 orientatierichting_bps varchar(1)
);

create table nww_wegvak_in_routes
(
 wee_wvk_id     numeric(9) not null,
 wee_begindatum timestamp  not null,
 rte_id         numeric(5) not null
);

create table nww_wegvak_relaties
(
 wvk_id_opvolger   numeric(10) not null,
 wvk_id_voorganger numeric(10) not null
);

create table provincies
(
 id       numeric(10) not null,
 polygoon geometry(GEOMETRY,28992)
);
```

B3.11 PostgreSQL/PostGIS ‘create index’ script

```
-- create_indexes.sql  23-08-2012  TT
-- Create PostgreSQL/PostGIS indexes for test data tables

alter table gga_gemeenten           drop constraint gga_gemeenten_pkey;
alter table gga_gemeente_efemeriden drop constraint gga_gemeente_efemeriden_pkey;
alter table gga_straten             drop constraint gga_straten_pkey;
alter table nwg_zone_efn            drop constraint nwg_zone_efn_pkey;
alter table nww_wegvak_efemeriden   drop constraint nww_wegvak_efemeriden_pkey;
alter table provincies              drop constraint provincies_pkey;

drop index gga_gemeenten_naam_idx;
drop index gga_gem_efn_begdat_idx;
drop index gga_gem_efn_enddat_idx;
drop index nwg_zone_efn_zne_znt_idx;
drop index nwg_zone_efn_zne_code_idx;
drop index nwg_zone_efn_begdat_idx;
drop index nwg_zone_efn_enddat_idx;
drop index nwg_zzrel_zne_znt_cod_idx;
drop index nwg_zzrel_zne_code_idx;
drop index nwg_zzrel_zz_code_sub_idx;
drop index nwg_zzrel_zne_cod_sub_idx;
drop index nwg_zzrel_begdat_idx;
drop index nwg_zzrel_enddat_idx;
drop index nww_wegvak_efn_begdat_idx;
drop index nww_wegvak_efn_enddat_idx;
drop index nww_wegvak_efn_wegbeh_idx;
drop index nww_wegvak_efn_stt_id_idx;

drop index geb_zone_efemeriden_sidx;
drop index gga_gemeente_efemeriden_sidx;
drop index nwg_zone_efn_sidx;
drop index nws_spoorhectopunt_efn_sidx;
drop index nws_spooroversteek_efn_sidx;
drop index nws_svk_efemeriden_sidx;
drop index nws_treinstation_efn_sidx;
drop index nwv_km_markering_efn_sidx;
drop index nwv_vaarwegvak_efn_sidx;
drop index nww_hectopunten_sidx;
drop index nww_wegvak_efemeriden_sidx;
drop index provincies_sidx;

-- ------------------------------------------------------------------------------------------
\timing on

alter table gga_gemeenten           add primary key (id)                using index tablespace
indx;
alter table gga_gemeente_efemeriden add primary key (gme_id,begindatum) using index tablespace
indx;
alter table gga_straten             add primary key (id)                using index tablespace
indx;
alter table nwg_zone_efn            add primary key (id)                using index tablespace
indx;
alter table nww_wegvak_efemeriden   add primary key (wvk_id,begindatum) using index tablespace
indx;
alter table provincies              add primary key (id)                using index tablespace
indx;

create index gga_gemeenten_naam_idx    on gga_gemeenten           (naam)              tablespace
indx;
create index gga_gem_efn_begdat_idx    on gga_gemeente_efemeriden (begindatum)        tablespace
indx;
create index gga_gem_efn_enddat_idx    on gga_gemeente_efemeriden (einddatum)         tablespace
indx;
create index nwg_zone_efn_zne_znt_idx  on nwg_zone_efn            (zne_znt_code)      tablespace
indx;
create index nwg_zone_efn_zne_code_idx on nwg_zone_efn            (zne_code)          tablespace
indx;
create index nwg_zone_efn_begdat_idx   on nwg_zone_efn            (begindatum)        tablespace
indx;
create index nwg_zone_efn_enddat_idx   on nwg_zone_efn            (einddatum)         tablespace
indx;
create index nwg_zzrel_zne_znt_cod_idx on nwg_zone_zone_relaties  (zne_znt_code)      tablespace
indx;
```

```
create index nwg_zzrel_zne_code_idx    on nwg_zone_zone_relaties  (zne_code)          tablespace
indx;
create index nwg_zzrel_zz_code_sub_idx on nwg_zone_zone_relaties  (zne_znt_code_sub)  tablespace
indx;
create index nwg_zzrel_zne_cod_sub_idx on nwg_zone_zone_relaties  (zne_code_sub)      tablespace
indx;
create index nwg_zzrel_begdat_idx      on nwg_zone_zone_relaties  (begindatum)        tablespace
indx;
create index nwg_zzrel_enddat_idx      on nwg_zone_zone_relaties  (einddatum)         tablespace
indx;
create index nww_wegvak_efn_begdat_idx on nww_wegvak_efemeriden   (begindatum)        tablespace
indx;
create index nww_wegvak_efn_enddat_idx on nww_wegvak_efemeriden   (einddatum)         tablespace
indx;
create index nww_wegvak_efn_wegbeh_idx on nww_wegvak_efemeriden   (wegbeheerdersoort) tablespace
indx;
create index nww_wegvak_efn_stt_id_idx on nww_wegvak_efemeriden   (stt_id)            tablespace
indx;

create index geb_zone_efemeriden_sidx     on geb_zone_efemeriden           using gist (polygoon)
tablespace indx;
create index gga_gemeente_efemeriden_sidx on gga_gemeente_efemeriden       using gist (polygoon)
tablespace indx;
create index nwg_zone_efn_sidx            on nwg_zone_efn                  using gist (polygoon)
tablespace indx;
create index nws_spoorhectopunt_efn_sidx  on nws_spoorhectopunt_efemeriden using gist (punt)
tablespace indx;
create index nws_spooroversteek_efn_sidx  on nws_spooroversteek_efemeriden using gist (punt)
tablespace indx;
create index nws_svk_efemeriden_sidx      on nws_svk_efemeriden            using gist (lijn)
tablespace indx;
create index nws_treinstation_efn_sidx    on nws_treinstation_efemeriden   using gist (punt)
tablespace indx;
create index nwv_km_markering_efn_sidx    on nwv_km_markering_efemeriden   using gist (punt)
tablespace indx;
create index nwv_vaarwegvak_efn_sidx      on nwv_vaarwegvak_efemeriden     using gist (lijn)
tablespace indx;
create index nww_hectopunten_sidx         on nww_hectopunten               using gist (punt)
tablespace indx;
create index nww_wegvak_efemeriden_sidx   on nww_wegvak_efemeriden         using gist (lijn)
tablespace indx;
create index provincies_sidx              on provincies                    using gist (polygoon)
tablespace indx;
```

B3.12 SQL Server ‘create table’ script

```
-- Script to create SQL Server tables used in testing
-- Compared to Oracle:
-- varchar2->varchar, number->numeric, date->datetime, sdo_geometry->geometry

drop table gga_gemeenten;
drop table gga_gemeente_efemeriden;
drop table gga_straten;
drop table nww_wegvak_efemeriden;
drop table provincies;

-- ------------------------------------------------------------

create table gga_gemeenten
(
 id                numeric(4)  not null,
 naam              varchar(24) not null,
 gme_id_in_vln     numeric(3),
 einddatum_dialoog datetime
);

create table gga_gemeente_efemeriden
(
 gme_id     numeric(4)  not null,
 begindatum datetime    not null,
 einddatum  datetime,
 pve_code   varchar(2)  not null,
 sde_id     numeric(38) not null,
 polygoon   geometry
);

create table gga_straten
(
 id             numeric(8)  not null,
 gme_id         numeric(4)  not null,
 wps_id         numeric(4)  not null,
 stt_type       varchar(1)  not null,
 naam           varchar(29) not null,
 naam_officieel varchar(43),
 naam_ptt       varchar(17),
 naamafkorting  varchar(5)
);

create table nww_wegvak_efemeriden
(
 wvk_id                 numeric(10) not null,
 begindatum             datetime    not null,
 einddatum              datetime,
 jte_id_begin           numeric(10) not null,
 jte_id_eind            numeric(10) not null,
 wegbeheerdersoort      varchar(1)  not null,
 wbr_id                 numeric(4),
 wegnummer              varchar(5),
 wegdeelletter          varchar(1),
 hectometreringsletter  varchar(1),
 bst_code               varchar(3),
 rpe_code               varchar(2),
 rijrichting            varchar(1),
 stt_id                 numeric(8),
 gme_id                 numeric(4),
 hnrstr_links           varchar(1),
 hnrstr_rechts          varchar(1),
 e_hnr_links            numeric(5),
 e_hnr_rechts           numeric(5),
 l_hnr_links            numeric(5),
 l_hnr_rechts           numeric(5),
 gon_id                 numeric(2)  not null,
 bronjaar               numeric(4),
 klok_begin             numeric(2)  not null,
 klok_eind              numeric(2)  not null,
 admin_richting         varchar(1),
 verkeersbaan_indicator varchar(1),
 relatief_baanvolgnr    numeric(1),
 type_wijziging         varchar(1),
```

```
 wegtype_code           varchar(2),
 sde_id                 numeric(38) not null,
 lijn                   geometry,
 orientatierichting_bps varchar(1)
);

create table provincies
(
 id       numeric(10) not null,
 polygoon geometry
);
```

B3.13 SQL Server ‘create index’ script

```
-- create_indexes.sql  12-10-2012  TT
-- Create SQl Server indexes for test data tables

drop index gga_gemeente_efemeriden_sidx on gga_gemeente_efemeriden;
drop index nww_wegvak_efemeriden_sidx   on nww_wegvak_efemeriden;
drop index provincies_sidx              on provincies;

alter table gga_gemeenten           drop constraint gga_gemeenten_pkey;
alter table gga_gemeente_efemeriden drop constraint gga_gemeente_efemeriden_pkey;
alter table gga_straten             drop constraint gga_straten_pkey;
alter table nww_wegvak_efemeriden   drop constraint nww_wegvak_efemeriden_pkey;
alter table provincies              drop constraint provincies_pkey;

drop index gga_gemeenten_naam_idx    on gga_gemeenten;
drop index gga_gem_efn_begdat_idx    on gga_gemeente_efemeriden;
drop index gga_gem_efn_enddat_idx    on gga_gemeente_efemeriden;
drop index nww_wegvak_efn_begdat_idx on nww_wegvak_efemeriden;
drop index nww_wegvak_efn_enddat_idx on nww_wegvak_efemeriden;
drop index nww_wegvak_efn_wegbeh_idx on nww_wegvak_efemeriden;
drop index nww_wegvak_efn_stt_id_idx on nww_wegvak_efemeriden;

-- ------------------------------------------------------------------------------------
SET STATISTICS TIME ON
--SET STATISTICS IO ON

alter table gga_gemeenten           add constraint gga_gemeenten_pkey           primary key
clustered (id);
alter table gga_gemeente_efemeriden add constraint gga_gemeente_efemeriden_pkey primary key
clustered (gme_id,begindatum);
alter table gga_straten             add constraint gga_straten_pkey             primary key
clustered (id);
alter table nww_wegvak_efemeriden   add constraint nww_wegvak_efemeriden_pkey   primary key
clustered (wvk_id,begindatum);
alter table provincies              add constraint provincies_pkey              primary key
clustered (id);

create index gga_gemeenten_naam_idx    on gga_gemeenten           (naam);
create index gga_gem_efn_begdat_idx    on gga_gemeente_efemeriden (begindatum);
create index gga_gem_efn_enddat_idx    on gga_gemeente_efemeriden (einddatum);
create index nww_wegvak_efn_begdat_idx on nww_wegvak_efemeriden   (begindatum);
create index nww_wegvak_efn_enddat_idx on nww_wegvak_efemeriden   (einddatum);
create index nww_wegvak_efn_wegbeh_idx on nww_wegvak_efemeriden   (wegbeheerdersoort);
create index nww_wegvak_efn_stt_id_idx on nww_wegvak_efemeriden   (stt_id);

create spatial index gga_gemeente_efemeriden_sidx on gga_gemeente_efemeriden (polygoon)
 with ( bounding_box = ( -50000, 250000, 300000, 650000 ) );
create spatial index nww_wegvak_efemeriden_sidx   on nww_wegvak_efemeriden   (lijn)
 with ( bounding_box = ( -50000, 250000, 300000, 650000 ) );
create spatial index provincies_sidx              on provincies              (polygoon)
 with ( bounding_box = ( -50000, 250000, 300000, 650000 ) );
GO
```

B3.14 Oracle functies voor het aggregeren van gebiedsindelingen

PL/SQL functies voor het aggregeren van zone efemeriden tot een groter geheel.

```
-- create_functions.sql  21-08-2012
--
-- Beide procedures aanmaken in de database
-- aanroepen met bv ophalen_zee_geometry('12','01',sysdate) geeft de actuele polygoon
-- van de eerste provincie (groningen dacht ik)

spool create_functions.log
set pagesize 500
set linesize 100
set trim on
set trimspool on
set verify off
set timing off
set echo on

create or replace FUNCTION BEPAAL_MAX_ORDINATE
(
  P_GEOMETRIE IN MDSYS.SDO_GEOMETRY
)
  RETURN NUMBER
IS
  l_max number := 0;
begin
  for i_teller in p_geometrie.sdo_ordinates.first..p_geometrie.sdo_ordinates.last
  loop
      if l_max < p_geometrie.sdo_ordinates(i_teller)
      then
        l_max := p_geometrie.sdo_ordinates(i_teller);
      end if;
  end loop;
  return(l_max);
end;
/
show errors;

/* Bepaal de geometry van een zone-efemeride */

create or replace FUNCTION OPHALEN_ZEE_GEOMETRY
(
  P_ZNE_ZNT_CODE IN NWG_ZONE_EFN.ZNE_ZNT_CODE%TYPE,
  P_ZNE_CODE     IN NWG_ZONE_EFN.ZNE_CODE%TYPE,
  P_PEILDATUM    IN DATE
)
  RETURN MDSYS.SDO_GEOMETRY
IS
  l_geometry      mdsys.sdo_geometry;
  l_geometry_tot  mdsys.sdo_geometry;
  l_gevonden      boolean;

  cursor c_zee (b_zne_code     nwg_zone_efn.zne_code%type
  ,             b_zne_znt_code nwg_zone_efn.zne_znt_code%type
  ,             b_peildatum    date) is
    select zee.zne_znt_code
    ,      zee.zne_code
    from   nwg_zone_efn zee
    where (zee.zne_znt_code,zee.zne_code) in
       (select distinct zze.zne_znt_code_sub, zze.zne_code_sub
        from nwg_zone_zone_relaties zze
        where zze.begindatum <= b_peildatum
        and  (zze.einddatum is null or zze.einddatum >= b_peildatum)
        connect by prior  zze.zne_code_sub     = zze.zne_code
        and        prior  zze.zne_znt_code_sub = zze.zne_znt_code
        and        prior  zze.begindatum <= b_peildatum
        and       (prior  zze.einddatum  is null
        or         prior  zze.einddatum  >= b_peildatum)
        start with       zze.zne_code         = b_zne_code
        and              zze.zne_znt_code     = b_zne_znt_code)
    and  zee.begindatum <= b_peildatum
    and (zee.einddatum is null or zee.einddatum >= b_peildatum)
    order by bepaal_max_ordinate(zee.polygoon) desc
```

```
  ;

  cursor c_zee2 (b_zne_code nwg_zone_efn.zne_code%type
  ,              b_zne_znt_code nwg_zone_efn.zne_znt_code%type
  ,              b_peildatum    date) is
    select zee.polygoon
    from   nwg_zone_efn zee
    where  zee.zne_znt_code =  b_zne_znt_code
    and    zee.zne_code     =  b_zne_code
    and    zee.begindatum   <= b_peildatum
    and   (zee.einddatum    is null or zee.einddatum >= b_peildatum)
    and    zee.polygoon     is not null
  ;
  r_zee2 c_zee2%rowtype;
begin
  /* Ophalen van alle onderliggende zones via zone-zone-relaties */
  for r_zee in c_zee (p_zne_code
  ,                   p_zne_znt_code
  ,                   p_peildatum)
  loop
    /* Ophalen polygoon van de zone */
      open c_zee2 (r_zee.zne_code
    ,              r_zee.zne_znt_code
    ,              p_peildatum);
    fetch c_zee2 into r_zee2;
    close c_zee2;
    if l_geometry is null
    then
      l_geometry := r_zee2.polygoon;
    else
      l_geometry := sdo_geom.sdo_union(l_geometry, r_zee2.polygoon, 0.0005);
    end if;
    /*
    ** Als de polygoon (l_geometry) groter wordt dan pakweg 5000 punten gaat
    ** het steeds langer duren voordat een nieuwe polygoon (r_zee2.polygoon) is toegevoegd
    ** Daarom wordt de polygoon (l_geometry) 'leeggemaakt' zodra er 5000 punten in zitten
    ** Deze punten worden dan toegevoegd aan de totaal-polygoon (l_geometry_tot)
    ** Hoewel dit toevoegen ook niet echt snel gaat, weegt het op tegen de meertijden
    ** die nodig zouden zijn indien het toevoegen stug werd doorgezet.
    */
    if l_geometry.sdo_ordinates.count >= 10000
    then
      if l_geometry_tot is null
      then
        l_geometry_tot := l_geometry;
      else
        l_geometry_tot := sdo_geom.sdo_union(l_geometry_tot, l_geometry, 0.0005);
      end if;
      l_geometry := null;
    end if;
  end loop;

  /* Voeg l_geometry en l_geometry_tot samen in l_geometry_tot */
  if l_geometry is not null
  then
    if l_geometry_tot is null
    then
      l_geometry_tot := l_geometry;
    else
      l_geometry_tot := sdo_geom.sdo_union(l_geometry_tot, l_geometry, 0.0005);
    end if;
  end if;

  /*
  ** geen polygoon gevonden, waarschijnlijk niet opgedeeld
  ** probeer de polygoon direct op te halen uit zone-efemeriden
  */
  if l_geometry_tot is null
  then
      open c_zee2 (p_zne_code
    ,              p_zne_znt_code
    ,              p_peildatum);
    fetch c_zee2 into l_geometry_tot;
    close c_zee2;
  end if;
```

```
  return(l_geometry_tot);
exception
  when others
  then
    if c_zee%isopen
    then
      close c_zee;
    end if;
    if c_zee2%isopen
    then
      close c_zee2;
    end if;
    raise;
end;
/
show errors;
spool off
exit;
```

B3.15 Oracle script voor aanmaken geaggregeerde gebiedsindelingen

Script voor het genereren van polygons met provincies en landsgrens, op basis van gemeente polygons, voor gebruik bij de GeoDBMS vergelijking.

```
spool get_zones.log
set pagesize 500
set linesize 100
set trim on
set trimspool on
set timing on
col status format a80
set echo on

define SRID = "28992"
define TOL  = "0.0005"
define XMIN = "-50000"
define XMAX = "300000"
define YMIN = "250000"
define YMAX = "650000"

drop table PROVINCIES;
create table PROVINCIES
(
  id       number(10) not NULL,
  polygoon sdo_geometry
);

-- ophalen_zee_geometry params:
--
--   1: soort zone (bijv. Ned[02], prov[12], gemeente[99], waterschap[10], etc.)
--   2: gebiedscode (prov Groningen[01], prov Zuid Holland[08], Nederland[TOTNL])

insert into PROVINCIES values ( 0, ophalen_zee_geometry('02','TOTNL',sysdate));

insert into PROVINCIES values ( 1, ophalen_zee_geometry('12','01',sysdate));
insert into PROVINCIES values ( 2, ophalen_zee_geometry('12','02',sysdate));
insert into PROVINCIES values ( 3, ophalen_zee_geometry('12','03',sysdate));
insert into PROVINCIES values ( 4, ophalen_zee_geometry('12','04',sysdate));
insert into PROVINCIES values ( 5, ophalen_zee_geometry('12','05',sysdate));
insert into PROVINCIES values ( 6, ophalen_zee_geometry('12','06',sysdate));
insert into PROVINCIES values ( 7, ophalen_zee_geometry('12','07',sysdate));
insert into PROVINCIES values ( 8, ophalen_zee_geometry('12','08',sysdate));
insert into PROVINCIES values ( 9, ophalen_zee_geometry('12','09',sysdate));
insert into PROVINCIES values (10, ophalen_zee_geometry('12','10',sysdate));
insert into PROVINCIES values (11, ophalen_zee_geometry('12','11',sysdate));
insert into PROVINCIES values (12, ophalen_zee_geometry('12','12',sysdate));

alter table PROVINCIES add primary key (ID) using index tablespace INDX enable;

delete from USER_SDO_GEOM_METADATA where table_name = 'PROVINCIES';
insert into USER_SDO_GEOM_METADATA values ('PROVINCIES','POLYGOON',
  sdo_dim_array(sdo_dim_element('X', &&XMIN, &&XMAX, &&TOL),
                sdo_dim_element('Y', &&YMIN, &&YMAX, &&TOL)), &&SRID);
commit;

select count(*), p.polygoon.sdo_gtype from PROVINCIES p group by p.polygoon.sdo_gtype
order by p.polygoon.sdo_gtype;

select id, status from (select id, sdo_geom.validate_geometry_with_context(POLYGOON,
  (select diminfo from user_sdo_geom_metadata where table_name='PROVINCIES')) status
from PROVINCIES) where status <> 'TRUE';

spool off
exit;
```

B3.16 Oracle script met ‘count’ queries

```
-- count_ora_queries.sql  02-10-2012  TT
--spool count_ora_queries.log

set pagesize 500
set linesize 100
set trim on
set trimspool on
set verify off
set numwidth 15
set echo on

define peildatum = "to_date('2011-07-30 12:00:00','YYYY-MM-DD HH24:MI:SS')"

define    nederland = "0"
define      prov_zh = "8"
define prov_limburg = "11"
define     nn_count = "100"
define     buf_dist = "500"
define     gem_list =
"'Amsterdam','Rotterdam','Utrecht','Eindhoven','Groningen','Arnhem'"
define    gem_delft = "'Delft'"

define  tol = "0.0005"
define srid = "28992"
define xmin = "-50000"
define xmax = "300000"
define ymin = "250000"
define ymax = "650000"

-- Small window: 2 x 2 km in Gemeente Groningen
define small_window = "230000,582000, 232000,584000"

-- Medium window: 50 x 50 km, groot deel van Zeeland
define medium_window = "20000,360000, 70000,410000"

-- Large window: 120 x 120 km in midden van Nederland
define large_window = "110000,430000, 230000,550000"

-- NL window: groot window waar Nederland geheel in past
define nl_window = "&xmin,&ymin, &xmax,&ymax"

-- Lijn dwars door Nederland
define nl_line =
"197000,306000,185000,338000,196000,379000,156000,422000,120000,479000,114000,537000,12
6000,581000,189000,600000,276000,588000"

define  small_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&small_w
indow))"
define medium_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&medium_
window))"
define  large_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&large_w
indow))"
define     nl_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&nl_wind
ow))"
define       nl_query_line =
"sdo_geometry(2002,&srid,NULL,sdo_elem_info_array(1,2,1),sdo_ordinate_array(&nl_line))"
define           rd_origin =
"sdo_geometry(2001,&srid,sdo_point_type(155000,463000,NULL),NULL,NULL)"

-- ------------------------------------------------------------------------
set timing on

-- 01 Basis, administratieve, query

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
```

```
;

-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;

-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)

-- 03 Small window: 2 x 2 km in Gemeente Groningen

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  sdo_anyinteract(wee.lijn, &small_query_window) = 'TRUE'
;

-- 04 Medium window: 50 x 50 km, groot deel van Zeeland

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  sdo_anyinteract(wee.lijn, &medium_query_window) = 'TRUE'
;

-- 05 Large window: 120 x 120 km in midden van Nederland

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  sdo_anyinteract(wee.lijn, &large_query_window) = 'TRUE'
;

-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  p.id = &prov_limburg
    and  sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE'
;

-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is null)
    and  gme.naam = &gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= &peildatum
    and (gee.einddatum  >= &peildatum or gee.einddatum is null)
    and  sdo_anyinteract(wee.lijn, gee.polygoon) = 'TRUE'
;

-- 08 Met berekening van lengte van de lijn

select wee.wvk_id, wee.begindatum, sdo_geom.sdo_length(wee.lijn, &tol)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
```

```
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  sdo_anyinteract(wee.lijn, &small_query_window) = 'TRUE'
    and  sdo_geom.sdo_length(wee.lijn, &tol) > 100
  order by wee.wvk_id
;

-- 09 Selectie van gemeente polygons op basis van provincie polygon (ZH: 2569 punten)

select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme, provincies p
  where  gee.gme_id = gme.id
    and  gee.begindatum <= &peildatum
    and (gee.einddatum  >= &peildatum or gee.einddatum is NULL)
    and  p.id = &prov_zh
    and  sdo_anyinteract(gee.polygoon, p.polygoon) = 'TRUE'
;

-- 10 Selectie van gemeente polygons op basis van lijn dwars door Nederland

select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme
  where  gee.gme_id = gme.id
    and  gee.begindatum <= &peildatum
    and (gee.einddatum  >= &peildatum or gee.einddatum is NULL)
    and  sdo_anyinteract(gee.polygoon, &nl_query_line) = 'TRUE'
;

-- 11 Selecteren van &nn_count dichtsbijzijnde lijnen (wegvakken) vanaf een punt

select * from
 (select wee.wvk_id, stt.naam, sdo_nn_distance(1) afstand
    from nww_wegvak_efemeriden wee, gga_straten stt
    where  wee.stt_id = stt.id
      and  wee.begindatum <= &peildatum
      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
      and  sdo_nn(wee.lijn, &rd_origin, 'sdo_num_res=500', 1) = 'TRUE'
    order by afstand, naam, wvk_id)
where rownum <= &nn_count
;

-- 12 Genereren van buffer rond een aantal gemeenten

create table ora_result12 as select gme.naam, sdo_geom.sdo_buffer(gee.polygoon,
&buf_dist, &tol) gem_buffer
  from gga_gemeenten gme, gga_gemeente_efemeriden gee
  where  gme.naam in (&gem_list)
    and  gee.gme_id = gme.id
    and  gee.begindatum <= &peildatum
    and (gee.einddatum  >= &peildatum or gee.einddatum is null)
;
select naam, sdo_geom.sdo_area(gem_buffer, &tol) from ora_result12 order by naam;

-- 13 Met ruimtelijke beperking (groot window waar Nederland geheel in past)

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  sdo_anyinteract(wee.lijn, &nl_query_window) = 'TRUE'
;

-- 14 Met ruimtelijke beperking (polygon met grens van Nederland: 18631 punten)

select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  p.id = &nederland
    and  sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE'
;

drop table ora_result12;
--spool off
exit;
```

B3.17 Oracle log voor ‘count’ queries

```
SQL*Plus: Release 11.2.0.3.0 Production on Fri Oct 12 11:29:15 2012
Copyright (c) 1982, 2011, Oracle.  All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

TGEO ORATEST SQL>
TGEO ORATEST SQL> define peildatum = "to_date('2011-07-30 12:00:00','YYYY-MM-DD
HH24:MI:SS')"
TGEO ORATEST SQL>
TGEO ORATEST SQL> define    nederland = "0"
TGEO ORATEST SQL> define      prov_zh = "8"
TGEO ORATEST SQL> define prov_limburg = "11"
TGEO ORATEST SQL> define     nn_count = "100"
TGEO ORATEST SQL> define     buf_dist = "500"
TGEO ORATEST SQL> define     gem_list =
"'Amsterdam','Rotterdam','Utrecht','Eindhoven','Groningen','Arnhem'"
TGEO ORATEST SQL> define    gem_delft = "'Delft'"
TGEO ORATEST SQL>
TGEO ORATEST SQL> define  tol = "0.0005"
TGEO ORATEST SQL> define srid = "28992"
TGEO ORATEST SQL> define xmin = "-50000"
TGEO ORATEST SQL> define xmax = "300000"
TGEO ORATEST SQL> define ymin = "250000"
TGEO ORATEST SQL> define ymax = "650000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- Small window: 2 x 2 km in Gemeente Groningen
TGEO ORATEST SQL> define small_window = "230000,582000, 232000,584000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- Medium window: 50 x 50 km, groot deel van Zeeland
TGEO ORATEST SQL> define medium_window = "20000,360000, 70000,410000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- Large window: 120 x 120 km in midden van Nederland
TGEO ORATEST SQL> define large_window = "110000,430000, 230000,550000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- NL window: groot window waar Nederland geheel in past
TGEO ORATEST SQL> define nl_window = "&xmin,&ymin, &xmax,&ymax"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- Lijn dwars door Nederland
TGEO ORATEST SQL> define nl_line =
"197000,306000,185000,338000,196000,379000,156000,422000,120000,479000,114000,537000,1
26000,581000,189000,600000,276000,588000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> define  small_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&small_
window))"
TGEO ORATEST SQL> define medium_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&medium
_window))"
TGEO ORATEST SQL> define  large_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&large_
window))"
TGEO ORATEST SQL> define     nl_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&nl_win
dow))"
TGEO ORATEST SQL> define       nl_query_line =
"sdo_geometry(2002,&srid,NULL,sdo_elem_info_array(1,2,1),sdo_ordinate_array(&nl_line))
"
TGEO ORATEST SQL> define           rd_origin =
"sdo_geometry(2001,&srid,sdo_point_type(155000,463000,NULL),NULL,NULL)"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- ------------------------------------------------------------------
----------
TGEO ORATEST SQL> set timing on
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 01 Basis, administratieve, query
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6  ;
```

```
       COUNT(*)
===============
         948481

1 row selected.

Elapsed: 00:00:01.31
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 02 Met beperking op wegbeheerdersoort, bv and
wee.wegbeheerdersoort in ('R','P')
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  wee.wegbeheerdersoort in ('R','P')
  7  ;

       COUNT(*)
===============
          42607

1 row selected.

Elapsed: 00:00:00.91
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- En/of met ruimtelijke beperking (rechthoekig window, in
verchillende groottes)
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 03 Small window: 2 x 2 km in Gemeente Groningen
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &small_query_window) = 'TRUE'
  7  ;

       COUNT(*)
===============
            524

1 row selected.

Elapsed: 00:00:00.07
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 04 Medium window: 50 x 50 km, groot deel van Zeeland
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &medium_query_window) = 'TRUE'
  7  ;

       COUNT(*)
===============
          31835

1 row selected.

Elapsed: 00:00:00.66
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 05 Large window: 120 x 120 km in midden van Nederland
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &large_query_window) = 'TRUE'
  7  ;
```

```
       COUNT(*)
===============
         326461

1 row selected.

Elapsed: 00:00:03.89
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 06 Met ruimtelijke beperking (polygon met grens van provincie
Limburg: 7946 punten)
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  p.id = &prov_limburg
  7      and  sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE'
  8  ;

       COUNT(*)
===============
          72903

1 row selected.

Elapsed: 00:00:01.95
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 07 Met ruimtelijke beperking (polygon met grens van gemeente
Delft: 252 punten)
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is null)
  6      and  gme.naam = &gem_delft
  7      and  gee.gme_id = gme.id
  8      and  gee.begindatum <= &peildatum
  9      and (gee.einddatum  >= &peildatum or gee.einddatum is null)
 10      and  sdo_anyinteract(wee.lijn, gee.polygoon) = 'TRUE'
 11  ;

       COUNT(*)
===============
           4195

1 row selected.

Elapsed: 00:00:00.20
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 08 Met berekening van lengte van de lijn
TGEO ORATEST SQL>
TGEO ORATEST SQL> select wee.wvk_id, wee.begindatum, sdo_geom.sdo_length(wee.lijn,
&tol)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &small_query_window) = 'TRUE'
  7      and  sdo_geom.sdo_length(wee.lijn, &tol) > 100
  8    order by wee.wvk_id
  9  ;

         WVK_ID BEGINDATU SDO_GEOM.SDO_LENGTH(WEE.LIJN,0.0005)
=============== ========= ====================================
      459565016 01-FEB-07                     260.01201434743
      459566001 01-JUL-02                     261.14388048298
      459566006 01-JUL-02                     239.45385672381
      459566010 01-JUL-02                     250.90343979292
      459567032 01-JUN-10                      389.4150450265
      460564008 01-DEC-02                     132.09466302618
      460564009 01-FEB-07                     139.88994137536
      460564026 01-DEC-02                     131.85598204101
```

460564032	01-DEC-02	132.03408650799
460564034	01-FEB-07	631.4900602052
460564045	01-FEB-07	199.51820287316
460564049	01-FEB-07	218.83130002385
460565004	01-DEC-02	162.24098552215
460565006	01-DEC-02	132.06059215375
460565008	01-DEC-02	203.1228264545
460565009	01-MAY-02	128.72451204025
460565011	01-MAY-02	153.93239918567
460565025	01-DEC-02	132.06059215375
460565026	01-DEC-02	132.45754036672
460565028	01-DEC-02	132.06059215375
460566001	01-MAY-02	301.47824500617
460566018	01-FEB-07	276.38760827824
460566020	01-FEB-07	138.28180844524
460566021	01-FEB-07	662.18745127522
460566022	01-JUL-02	131.39060574433
460566024	01-JUL-02	131.68192374204
460566026	01-JUL-02	137.20684262884
460567001	01-JUN-10	296.37205519022
460567003	01-JUN-10	266.47380560377
460567004	01-DEC-10	450.93389800818
460568020	01-DEC-10	119.22709667219
460568021	01-JUN-10	443.6015395944
461563003	01-SEP-02	265.53738003744
461564014	01-FEB-07	137.20329553767
461564029	01-FEB-07	116.55752435602
461564031	01-FEB-07	116.86862554593
461564036	01-FEB-07	114.79490924249
461564044	01-FEB-08	115.64902106377
461565004	14-JUN-97	154.97080516665
461565006	01-FEB-07	152.13427143461
461565009	14-JUN-97	153.08023006426
461565021	01-FEB-07	242.55105578345
461565022	01-FEB-07	244.42774551571
461565024	01-FEB-06	307.52596519745
461565025	01-FEB-06	118.25596102095
461565028	01-APR-06	111.15764933191
461565029	01-APR-06	154.00656427573
461565030	01-FEB-06	136.03126461955
461565031	01-FEB-07	116.84233573506
461565033	01-FEB-07	179.14122979023
461565034	01-FEB-07	146.08220726817
461566002	14-JUN-97	151.52887513606
461566003	14-JUN-97	147.70578864757
461566013	14-JUN-97	151.53248760374
461566015	14-JUN-97	151.53777859439
461566020	14-JUN-97	125.22779244241
461566021	14-JUN-97	250.86367997806
461566022	14-JUN-97	246.67792767088
461566024	14-JUN-97	185.86411156578
461566026	14-JUN-97	104.69479452198
461566028	27-NOV-97	146.45477117527
461566029	27-NOV-97	103.74020609067
461566033	01-MAY-01	236.11230015523
461567012	01-JUN-10	115.68090957699
461567016	01-JUN-10	104.84359565351
461567018	01-JUN-10	220.49022308419
461567021	01-JUN-10	166.47092420991
461567023	01-JUN-10	221.71733456163
461567024	01-JUN-10	125.6947528848
461567030	01-JUN-10	706.42122944798
461567033	01-JUN-10	122.27255647939
461567034	01-JUN-10	189.18381048642
461567035	01-JUN-10	299.69233239782
461568002	01-JUN-10	341.18773495565
461568018	01-FEB-05	387.11427817877
461568019	01-FEB-05	390.59328291741
461568022	01-JUN-10	640.48263154415
462564002	14-JUN-97	246.69664537137
462564016	14-JUN-97	131.24460134199
462564019	14-JUN-97	248.30783727357
462564023	01-FEB-07	331.10658740892
462564024	01-SEP-02	214.21997221915
462564026	14-JUN-97	117.64352935882
462564033	01-FEB-07	105.13881692794

462564058	01-FEB-07	119.28415344816
462564075	01-FEB-07	105.25087918871
462564083	01-FEB-07	167.35884309549
462564084	01-FEB-07	117.13889117195
462565001	01-SEP-02	106.25441167312
462565004	14-JUN-97	377.28463799152
462565005	14-JUN-97	125
462565006	14-JUN-97	126.17844506888
462565008	14-JUN-97	152.73840332767
462565025	14-JUN-97	150.84137302157
462565027	14-JUN-97	141.68274418573
462565034	14-JUN-97	127.14038467017
462565036	14-JUN-97	130.54501139454
462565041	01-SEP-07	351.72716699169
462565046	01-JUL-02	154.33081351435
462565047	01-SEP-07	161.17940922598
462565050	01-SEP-07	157.4533096793
462565054	01-SEP-07	160.28144534077
462565058	01-SEP-07	158.37874286665
462565061	01-SEP-07	167.34790738398
462566001	14-JUN-97	151.82348847311
462566002	14-JUN-97	157.19096666157
462566003	14-JUN-97	160.73289064073
462566007	14-JUN-97	150.58093060889
462566008	14-JUN-97	153.68833275902
462566010	14-JUN-97	101.80824194944
462566011	14-JUN-97	164.12495239908
462566019	14-JUN-97	152.15361647809
462566020	14-JUN-97	247.07262436538
462566021	14-JUN-97	116.71281466844
462566022	01-APR-04	120.44336488957
462566023	01-APR-04	465.58920625833
462566025	14-JUN-97	547.95424064068
462566026	01-APR-04	458.90942564516
462567011	14-JUN-97	117.95338062133
462567012	14-JUN-97	123.96773773849
462567019	01-JUN-02	123.97214562984
462567026	01-JUN-10	277.15643204613
462567030	01-JUN-10	289.51173245849
462567036	01-JUN-10	179.02342624594
462567037	11-NOV-97	328.80253391059
462567040	01-APR-04	366.39935486408
462567041	01-JUN-10	107.66764074057
462567044	01-NOV-08	183.17211599734
462567046	01-APR-04	162.83768534609
462567048	01-JUN-10	314.19355440936
462567051	01-FEB-07	128.95476137736
462567053	01-JUN-10	238.92144860853
462567056	01-JUN-10	425.62850351606
462567057	01-JUN-10	109.54123458059
462568013	01-FEB-07	527.11656330305
462569010	01-MAR-07	1305.6445483639
463564002	01-FEB-07	248.6343838108
463564003	14-JUN-97	251.71889884915
463564005	01-FEB-07	127.12985487288
463564008	14-JUN-97	127.7575829452
463564016	01-FEB-08	335.52745799262
463564017	01-FEB-08	330.68841870967
463564019	01-FEB-08	803.13886088791
463564022	01-FEB-08	801.39284928404
463565004	14-JUN-97	122.71104269788
463565005	14-JUN-97	112.89494246473
463565006	14-JUN-97	240.65241206738
463565012	14-JUN-97	104.04326023342
463565013	14-JUN-97	122.06555615734
463565014	14-JUN-97	122.06555615734
463565016	14-JUN-97	288.83626142219
463565017	14-JUN-97	221.36345062798
463565020	14-JUN-97	128.40009484283
463565021	14-JUN-97	124.27791436937
463565022	14-JUN-97	125.87615557576
463565023	14-JUN-97	127.12985487288
463565025	14-JUN-97	121.43310915891
463565026	14-JUN-97	115.78995520092
463565029	01-APR-04	502.38170825187
463565032	01-APR-04	498.55925104279

```
      463565034 01-AUG-09                    757.02104472756
      463566003 14-JUN-97                    117.95338062133
      463566005 14-JUN-97                    121.75795661886
      463566009 14-JUN-97                     124.9199743836
      463566010 11-NOV-97                    114.79468667961
      463566012 01-MAR-04                    117.95353589407
      463566015 14-JUN-97                     120.1665510864
      463566016 14-JUN-97                    122.71104269788
      463566018 14-JUN-97                    117.95338062133
      463566019 14-JUN-97                     116.0560209554
      463566021 14-JUN-97                    122.06555615734
      463566023 14-JUN-97                    122.06555615734
      463566025 14-JUN-97                    126.49110640674
      463566026 14-JUN-97                    125.22779244241
      463566027 14-JUN-97                    118.90332207302
      463566028 14-JUN-97                    121.11918837509
      463566029 14-JUN-97                    117.64352935882
      463566030 14-JUN-97                    117.95338062133
      463566031 14-JUN-97                    118.29740147204
      463566032 14-JUN-97                    115.43396380615
      463566033 14-JUN-97                    124.27791436937
      463566034 14-JUN-97                    176.17095069254
      463566036 14-JUN-97                    118.27087553578
      463566042 01-JUN-02                    134.98688457929
      463567019 14-JUN-97                    123.97132262304
      463567020 14-JUN-97                    124.29023838605
      463567022 14-JUN-97                    124.60476436662
      463567025 14-JUN-97                    118.60555539069
      463567026 14-JUN-97                    117.32433677631
      463567030 14-JUN-97                    124.27791436937
      463567031 14-JUN-97                    126.17844506888
      463567033 14-JUN-97                    120.17621590187
      463567034 14-JUN-97                    116.69190203266
      463567042 14-JUN-97                    147.05441169853
      463567044 14-JUN-97                    149.58085266252
      463567047 14-JUN-97                    138.19189556555
      463567048 14-JUN-97                    138.51353724456
      463567049 14-JUN-97                    129.40131679469
      463567050 01-JUN-01                    299.64520305767
      463567052 01-JAN-00                    255.94607355175
      463567053 01-JUN-01                    300.90227765271
      463567055 01-JAN-00                    252.53650891012
      463567057 14-JUN-97                    338.14677166246
      463567058 01-JUN-10                     561.4232382295
      463567061 14-JUN-97                    106.57613530839
      463567064 14-JUN-97                    117.64352935882
      463567068 01-APR-04                    561.41826099433
      463567069 01-FEB-07                     292.6441350678
      463568034 15-APR-98                    123.16655390162
      464564075 01-AUG-09                    776.62307283525

210 rows selected.

Elapsed: 00:00:00.35
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 09 Selectie van gemeente polygons op basis van provincie polygon
(ZH: 2569 punten)
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from gga_gemeente_efemeriden gee, gga_gemeenten gme, provincies p
  3    where  gee.gme_id = gme.id
  4      and  gee.begindatum <= &peildatum
  5      and (gee.einddatum  >= &peildatum or gee.einddatum is NULL)
  6      and  p.id = &prov_zh
  7      and  sdo_anyinteract(gee.polygoon, p.polygoon) = 'TRUE'
  8  ;

       COUNT(*)
===============
             94

1 row selected.

Elapsed: 00:00:00.18
TGEO ORATEST SQL>
```

```
TGEO ORATEST SQL> -- 10 Selectie van gemeente polygons op basis van lijn dwars door
Nederland
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from gga_gemeente_efemeriden gee, gga_gemeenten gme
  3    where  gee.gme_id = gme.id
  4      and  gee.begindatum <= &peildatum
  5      and (gee.einddatum  >= &peildatum or gee.einddatum is NULL)
  6      and  sdo_anyinteract(gee.polygoon, &nl_query_line) = 'TRUE'
  7  ;

       COUNT(*)
===============
             62

1 row selected.

Elapsed: 00:00:00.51
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 11 Selecteren van &nn_count dichtsbijzijnde lijnen (wegvakken)
vanaf een punt
TGEO ORATEST SQL>
TGEO ORATEST SQL> select * from
  2   (select wee.wvk_id, stt.naam, sdo_nn_distance(1) afstand
  3      from nww_wegvak_efemeriden wee, gga_straten stt
  4      where  wee.stt_id = stt.id
  5        and  wee.begindatum <= &peildatum
  6        and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  7        and  sdo_nn(wee.lijn, &rd_origin, 'sdo_num_res=500', 1) = 'TRUE'
  8      order by afstand, naam, wvk_id)
  9  where rownum <= &nn_count
 10  ;

         WVK_ID NAAM                           AFSTAND
=============== ============================== ===============
      310326069 Krankeledenstraat             12.141073279477
      309326070 Breestraat                                 25
      310326003 Lieve Vrouwekerkhof                        25
      310325057 Krankeledenstraat             32.015621187164
      310325020 Lieve Vrouwekerkhof           32.015621187164
      309326018 Westsingel                    36.380020071675
      310326027 Lieve Vrouwestraat             69.50027233238
      310326026 Lieve Vrouwestraat            70.349129347846
      310326028 Lieve Vrouwestraat             79.75274673637
      310326004 Zwanenhalssteeg                79.75274673637
      309326088 Hellestraat                   87.182058275235
      309325137 Varkensmarkt                  87.885195044445
      310325027 Varkensmarkt                  87.885195044445
      309326104 Hellestraat                   90.578494406787
      309325135 Sint Jorisstraat              90.578494406787
      309325138 Utrechtsestraat               95.108072475494
      309325136 Varkensmarkt                  95.108072475494
      310325051 Langestraat                    96.79797340735
      310325055 Langestraat                   97.077524115406
      310325022 Varkensmarkt                  102.59142264342
      310325056 Langestraat                   106.11018692851
      310325038 Scherbierstraat               106.11018692851
      309325110 Utrechtsestraat               111.00130152842
      310325083 Varkensmarkt                  113.74813746609
      310325021 Zuidsingel                    113.74813746609
      309326069 Bollebruggang                 117.95942228499
      309326105 Breestraat                    118.29231769225
      310326181 Paternosterstraat             118.29231769225
      310326060 Langegracht                   118.91896247866
      310326156 Langegracht                   118.91896247866
      309326089 Molenstraat                    123.3862219618
      309326023 Westsingel                     123.3862219618
      309326149 Hellestraat                   124.25463201426
      309326160 Torenstraat                   124.25463201426
      310326157 Langegracht                   129.49517365524
      310326155 Marktgang                     129.49517365524
      310326063 Lieve Vrouwestraat            136.51598652538
      309326087 Molenstraat                   145.45446022725
      309325120 Arnhemsestraat                 147.7633242723
      310325061 Koestraat                      147.7633242723
      310326191 Langegracht                   152.18433025775
```

```
      310326162 Krommestraat                 153.60339186359
      309326086 Molenstraat                  155.20631430454
      309326009 Stadhuisplein                155.20631430454
      310325050 Mooierstraat                 156.38128936993
      310325039 Scherbierstraat              156.38128936993
      310326072 Kortegracht                   156.6447011552
      310326054 Langestraat                   156.6447011552
      310326039 Achter het Oude Stadhuis     158.30109639547
      310326158 Krommestraat                 158.30109639547
      310326159 Krommestraat                  158.9607126368
      309325133 Riddergang                   160.69743148544
      309325134 Sint Jorisplein              162.26008708616
      310326161 Krommestraat                 166.47529820666
      310326160 Peperstraat                  166.47529820666
      309325111 Utrechtsestraat              167.72451737894
      309325114 Sint Jorisstraat             167.73307710764
      309326152 Molenstraat                  168.43396332094
      310325108 Koesteeg                     172.02906731131
      310325123 Koestraat                    172.02906731131
      309326107 Breestraat                   175.45304336205
      309326182 Kromme Elleboogsteeg         175.45304336205
      310326075 Achter de Heilige Geest      175.84981601925
      310326071 Kortegracht                  175.84981601925
      309325123 Ad Arnhemse Poortwal         177.58395100907
      310325046 Muurhuizen                   178.94964919219
      310325122 Rozemarijnsteeg              178.94964919219
      310326055 Langestraat                   179.9562448208
      310326076 Achter de Heilige Geest      181.45893347256
      310326194 Kortegracht                  181.45893347256
      310325107 Haagplein                    181.55791835389
      310325116 Koesteeg                     181.55791835389
      310326190 Kerkgang                     184.14741817359
      309326169 Langegracht                  184.14741817359
      310326164 Hof                          187.16928505498
      310325049 Mooierstraat                 187.24008230077
      310325121 Schelvissteegje              187.24008230077
      309325122 Ad Arnhemse Poortwal         193.48761960654
      309325119 Slijkpoortsteeg              193.48761960654
      310326085 Hof                          197.38140305509
      310326025 Vijver                       197.38140305509
      310326056 Langestraat                  198.50809346723
      310326030 Valkestraat                  198.50809346723
      310326163 Hof                          200.60210346107
      310326154 Krommestraat                 201.75531537483
      310325058 Kortegracht                  205.17207303139
      309325124 Ad Arnhemse Poortwal         211.18390966172
      310325115 Haagplein                    212.12303236093
      309325121 Arnhemsestraat               213.57414542495
      309325139 Koesteeg                     213.57414542495
      310326193 Valkestraat                  214.72663375789
      309325009 Ad Arnhemse Poortwal         219.03516370206
      310325114 Haagplein                    221.66884690909
      309326181 Kromme Elleboogsteeg         221.77759583195
      310325117 Koesteeg                     222.67484727735
      310325124 Koesteeg                     222.67484727735
      309325018 Stadsring                    225.47599817719
      310326179 Hof                          226.49559820227
      310326178 Hof                          227.29738406986
      310326177 Windsteeg                    227.29738406986

100 rows selected.

Elapsed: 00:00:00.11
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 12 Genereren van buffer rond een aantal gemeenten
TGEO ORATEST SQL>
TGEO ORATEST SQL> create table ora_result12 as select gme.naam,
sdo_geom.sdo_buffer(gee.polygoon, &buf_dist, &tol) gem_buffer
  2    from gga_gemeenten gme, gga_gemeente_efemeriden gee
  3    where  gme.naam in (&gem_list)
  4      and  gee.gme_id = gme.id
  5      and  gee.begindatum <= &peildatum
  6      and (gee.einddatum  >= &peildatum or gee.einddatum is null)
  7  ;

Table created.
```

```
Elapsed: 00:00:54.97
TGEO ORATEST SQL> select naam, sdo_geom.sdo_area(gem_buffer, &tol) from ora_result12
order by naam;

NAAM                     SDO_GEOM.SDO_AREA(GEM_BUFFER,0.0005)
======================== ====================================
Amsterdam                                        267569031.86765
Arnhem                                           131287150.67225
Eindhoven                                         112722121.5507
Groningen                                        106408385.27189
Rotterdam                                        394407244.56048
Utrecht                                          128207251.96278

6 rows selected.

Elapsed: 00:00:00.02
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 13 Met ruimtelijke beperking (groot window waar Nederland geheel
in past)
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &nl_query_window) = 'TRUE'
  7  ;

       COUNT(*)
===============
         948481

1 row selected.

Elapsed: 00:00:09.75
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 14 Met ruimtelijke beperking (polygon met grens van Nederland:
18631 punten)
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  p.id = &nederland
  7      and  sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE'
  8  ;

       COUNT(*)
===============
         948481

1 row selected.

Elapsed: 00:00:37.18
TGEO ORATEST SQL>
TGEO ORATEST SQL> drop table ora_result12;

Table dropped.

Elapsed: 00:00:00.04
TGEO ORATEST SQL> --spool off
TGEO ORATEST SQL> exit;
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

B3.18 Oracle script met ‘create’ queries

```
-- create_ora_queries.sql  06-09-2012  TT
--spool create_ora_queries.log

set pagesize 500
set linesize 100
set trim on
set trimspool on
set verify off
set serveroutput on
set echo on

define peildatum = "to_date('2011-07-30 12:00:00','YYYY-MM-DD HH24:MI:SS')"

define    nederland = "0"
define      prov_zh = "8"
define prov_limburg = "11"
define    gem_delft = "'Delft'"

define srid = "28992"

-- Small window: 2 x 2 km in Gemeente Groningen
define small_window = "230000,582000, 232000,584000"

-- Medium window: 50 x 50 km, groot deel van Zeeland
define medium_window = "20000,360000, 70000,410000"

-- Large window: 120 x 120 km in midden van Nederland
define large_window = "110000,430000, 230000,550000"

define  small_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&small_
window))"
define medium_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&medium
_window))"
define  large_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&large_
window))"

-- ------------------------------------------------------------------------
set timing on

-- 01 Basis, administratieve, query

create table ora_result01 tablespace RESULT as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
;

-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')

create table ora_result02 tablespace RESULT as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;

-- En/of met ruimtelijke beperking (simpel window, in verchillende groottes)

-- 03 Small window: 2 x 2 km in Gemeente Groningen

create table ora_result03 tablespace RESULT as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  sdo_anyinteract(wee.lijn, &small_query_window) = 'TRUE'
;
```

```
-- 04 Medium window: 50 x 50 km, groot deel van Zeeland

create table ora_result04 tablespace RESULT as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  sdo_anyinteract(wee.lijn, &medium_query_window) = 'TRUE'
;

-- 05 Large window: 120 x 120 km in midden van Nederland

create table ora_result05 tablespace RESULT as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  sdo_anyinteract(wee.lijn, &large_query_window) = 'TRUE'
;

-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)

create table ora_result06 tablespace RESULT as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
    and  p.id = &prov_limburg
    and  sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE'
;

-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)

create table ora_result07 tablespace RESULT as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= &peildatum
    and (wee.einddatum  >= &peildatum or wee.einddatum is null)
    and  gme.naam = &gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= &peildatum
    and (gee.einddatum  >= &peildatum or gee.einddatum is null)
    and  sdo_anyinteract(wee.lijn, gee.polygoon) = 'TRUE'
;
set timing off

select count(*) from ora_result01;
select count(*) from ora_result02;
select count(*) from ora_result03;
select count(*) from ora_result04;
select count(*) from ora_result05;
select count(*) from ora_result06;
select count(*) from ora_result07;

drop table ora_result01;
drop table ora_result02;
drop table ora_result03;
drop table ora_result04;
drop table ora_result05;
drop table ora_result06;
drop table ora_result07;

--spool off
exit;
```

B3.19 Oracle log voor ‘create’ queries

```
SQL*Plus: Release 11.2.0.3.0 Production on Thu Sep 6 16:53:39 2012
Copyright (c) 1982, 2011, Oracle.  All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

TGEO ORATEST SQL>
TGEO ORATEST SQL> define peildatum = "to_date('2011-07-30 12:00:00','YYYY-MM-DD
HH24:MI:SS')"
TGEO ORATEST SQL>
TGEO ORATEST SQL> define    nederland = "0"
TGEO ORATEST SQL> define      prov_zh = "8"
TGEO ORATEST SQL> define prov_limburg = "11"
TGEO ORATEST SQL> define    gem_delft = "'Delft'"
TGEO ORATEST SQL>
TGEO ORATEST SQL> define srid = "28992"
TGEO ORATEST SQL> define xmin = "-50000"
TGEO ORATEST SQL> define xmax = "300000"
TGEO ORATEST SQL> define ymin = "250000"
TGEO ORATEST SQL> define ymax = "650000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- Small window: 2 x 2 km in Gemeente Groningen
TGEO ORATEST SQL> define small_window = "230000,582000, 232000,584000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- Medium window: 50 x 50 km, groot deel van Zeeland
TGEO ORATEST SQL> define medium_window = "20000,360000, 70000,410000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- Large window: 120 x 120 km in midden van Nederland
TGEO ORATEST SQL> define large_window = "110000,430000, 230000,550000"
TGEO ORATEST SQL>
TGEO ORATEST SQL> define  small_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&small_
window))"
TGEO ORATEST SQL> define medium_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&medium
_window))"
TGEO ORATEST SQL> define  large_query_window =
"sdo_geometry(2003,&srid,NULL,sdo_elem_info_array(1,1003,3),sdo_ordinate_array(&large_
window))"
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- ------------------------------------------------------------------
----------
TGEO ORATEST SQL> set timing on
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 01 Basis, administratieve, query
TGEO ORATEST SQL>
TGEO ORATEST SQL> create table ora_result01 tablespace RESULT as select wee.wvk_id,
wee.lijn, stt.id
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6  ;

Table created.

Elapsed: 00:00:06.76
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 02 Met beperking op wegbeheerdersoort, bv and
wee.wegbeheerdersoort in ('R','P')
TGEO ORATEST SQL>
TGEO ORATEST SQL> create table ora_result02 tablespace RESULT as select wee.wvk_id,
wee.lijn, stt.id
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  wee.wegbeheerdersoort in ('R','P')
  7  ;

Table created.

Elapsed: 00:00:01.60
```

```
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- En/of met ruimtelijke beperking (simpel window, in verchillende
groottes)
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 03 Small window: 2 x 2 km in Gemeente Groningen
TGEO ORATEST SQL>
TGEO ORATEST SQL> create table ora_result03 tablespace RESULT as select wee.wvk_id,
wee.lijn, stt.id
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &small_query_window) = 'TRUE'
  7  ;

Table created.

Elapsed: 00:00:00.31
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 04 Medium window: 50 x 50 km, groot deel van Zeeland
TGEO ORATEST SQL>
TGEO ORATEST SQL> create table ora_result04 tablespace RESULT as select wee.wvk_id,
wee.lijn, stt.id
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &medium_query_window) = 'TRUE'
  7  ;

Table created.

Elapsed: 00:00:01.01
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 05 Large window: 120 x 120 km in midden van Nederland
TGEO ORATEST SQL>
TGEO ORATEST SQL> create table ora_result05 tablespace RESULT as select wee.wvk_id,
wee.lijn, stt.id
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &large_query_window) = 'TRUE'
  7  ;

Table created.

Elapsed: 00:00:05.81
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 06 Met ruimtelijke beperking (polygon met grens van provincie
Limburg: 7946 punten)
TGEO ORATEST SQL>
TGEO ORATEST SQL> create table ora_result06 tablespace RESULT as select wee.wvk_id,
wee.lijn, stt.id
  2    from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  p.id = &prov_limburg
  7      and  sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE'
  8  ;

Table created.

Elapsed: 00:00:02.52
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 07 Met ruimtelijke beperking (polygon met grens van gemeente
Delft: 252 punten)
TGEO ORATEST SQL>
TGEO ORATEST SQL> create table ora_result07 tablespace RESULT as select wee.wvk_id,
wee.lijn, stt.id
  2    from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is null)
```

```
  6      and  gme.naam = &gem_delft
  7      and  gee.gme_id = gme.id
  8      and  gee.begindatum <= &peildatum
  9      and (gee.einddatum  >= &peildatum or gee.einddatum is null)
 10      and  sdo_anyinteract(wee.lijn, gee.polygoon) = 'TRUE'
 11  ;

Table created.

Elapsed: 00:00:00.49
TGEO ORATEST SQL> set timing off
TGEO ORATEST SQL>
TGEO ORATEST SQL> select count(*) from ora_result01;

 COUNT(*)
=========
   948481

1 row selected.

TGEO ORATEST SQL> select count(*) from ora_result02;

 COUNT(*)
=========
    42607

1 row selected.

TGEO ORATEST SQL> select count(*) from ora_result03;

 COUNT(*)
=========
      524

1 row selected.

TGEO ORATEST SQL> select count(*) from ora_result04;

 COUNT(*)
=========
    31835

1 row selected.

TGEO ORATEST SQL> select count(*) from ora_result05;

 COUNT(*)
=========
   326461

1 row selected.

TGEO ORATEST SQL> select count(*) from ora_result06;

 COUNT(*)
=========
    72903

1 row selected.

TGEO ORATEST SQL> select count(*) from ora_result07;

 COUNT(*)
=========
     4195

1 row selected.

TGEO ORATEST SQL>
TGEO ORATEST SQL> drop table ora_result01;

Table dropped.

TGEO ORATEST SQL> drop table ora_result02;

Table dropped.
```

```
TGEO ORATEST SQL> drop table ora_result03;

Table dropped.

TGEO ORATEST SQL> drop table ora_result04;

Table dropped.

TGEO ORATEST SQL> drop table ora_result05;

Table dropped.

TGEO ORATEST SQL> drop table ora_result06;

Table dropped.

TGEO ORATEST SQL> drop table ora_result07;

Table dropped.

TGEO ORATEST SQL>
TGEO ORATEST SQL> --spool off
TGEO ORATEST SQL> exit;
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

B3.20 Oracle log met query plannen

```
SQL*Plus: Release 11.2.0.3.0 Production on Fri Oct 12 14:23:59 2012
Copyright (c) 1982, 2011, Oracle.  All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

TGEO ORATEST SQL> -- And/of with spatial selection (rectangular window, in different sizes)
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 03 Small window: 2 x 2 km in municipality Groningen
TGEO ORATEST SQL> set autotrace on
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &small_query_window) = 'TRUE';

       COUNT(*)
===============
           524

Elapsed: 00:00:00.08

Execution Plan
==========================================================
Plan hash value: 486044591

-------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                        | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                             |     1 |   263 |       |  7209   (1)| 00:01:27 |
|   1 |  SORT AGGREGATE               |                             |     1 |   263 |       |            |          |
|*  2 |   HASH JOIN                   |                             |  9489 |  2437K|  2504K|  7209   (1)| 00:01:27 |
|*  3 |    TABLE ACCESS BY INDEX ROWID| NWW_WEGVAK_EFEMERIDEN       |  9489 |  2390K|       |  6590   (1)| 00:01:20 |
|*  4 |     DOMAIN INDEX              | NWW_WEGVAK_EFEMERIDEN_SIDX  |       |       |       |     0   (0)| 00:00:01 |
|   5 |    INDEX FAST FULL SCAN       | SYS_C006716                 |   355K|  1737K|       |   206   (2)| 00:00:03 |
-------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("WEE"."STT_ID"="STT"."ID")
   3 - filter(("WEE"."EINDDATUM" IS NULL OR "WEE"."EINDDATUM">=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd
              hh24:mi:ss')) AND "WEE"."BEGINDATUM"<=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd hh24:mi:ss'))
   4 - access("MDSYS"."SDO_ANYINTERACT"("WEE"."LIJN","MDSYS"."SDO_GEOMETRY"(2003,28992,NULL,"SDO_ELEM_INFO_A
              RRAY"(1,1003,3),"SDO_ORDINATE_ARRAY"(230000,582000,232000,584000)))='TRUE')

Statistics
==========================================================
        535  recursive calls
          0  db block gets
       2693  consistent gets
        105  physical reads
          0  redo size
        527  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          1  sorts (memory)
          0  sorts (disk)
          1  rows processed

TGEO ORATEST SQL> -- 05 Large window: 120 x 120 km in the middle of the Netherlands
TGEO ORATEST SQL> set autotrace on
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &large_query_window) = 'TRUE';

       COUNT(*)
===============
         326461
```

```
Elapsed: 00:00:03.83

Execution Plan
==========================================================
Plan hash value: 486044591

-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                        | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                             |     1 |   263 |       |  7209   (1)| 00:01:27 |
|   1 |  SORT AGGREGATE               |                             |     1 |   263 |       |            |          |
|*  2 |   HASH JOIN                   |                             |  9489 |  2437K|  2504K|  7209   (1)| 00:01:27 |
|*  3 |    TABLE ACCESS BY INDEX ROWID| NWW_WEGVAK_EFEMERIDEN       |  9489 |  2390K|       |  6590   (1)| 00:01:20 |
|*  4 |     DOMAIN INDEX              | NWW_WEGVAK_EFEMERIDEN_SIDX  |       |       |       |     0   (0)| 00:00:01 |
|   5 |    INDEX FAST FULL SCAN       | SYS_C006716                 |   355K|  1737K|       |   206   (2)| 00:00:03 |
-----------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("WEE"."STT_ID"="STT"."ID")
   3 - filter(("WEE"."EINDDATUM" IS NULL OR "WEE"."EINDDATUM">=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd
              hh24:mi:ss')) AND "WEE"."BEGINDATUM"<=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd hh24:mi:ss'))
   4 - access("MDSYS"."SDO_ANYINTERACT"("WEE"."LIJN","MDSYS"."SDO_GEOMETRY"(2003,28992,NULL,"SDO_ELEM_INFO_A
              RRAY"(1,1003,3),"SDO_ORDINATE_ARRAY"(110000,430000,230000,550000)))='TRUE')

Statistics
==========================================================
      80265  recursive calls
          0  db block gets
     987271  consistent gets
      57559  physical reads
          0  redo size
        528  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          1  sorts (memory)
          0  sorts (disk)
          1  rows processed

TGEO ORATEST SQL> set autotrace off
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 06 With spatial selection (polygon with boundary of 'Limburg' province: 7946 points)
TGEO ORATEST SQL> set autotrace on
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  p.id = &prov_limburg
  7      and  sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE';

       COUNT(*)
===============
          72903

Elapsed: 00:00:01.94

Execution Plan
==========================================================
Plan hash value: 4199729836

-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                        | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                             |     1 |   387 |       |  7267   (1)| 00:01:28 |
|   1 |  SORT AGGREGATE               |                             |     1 |   387 |       |            |          |
|*  2 |   HASH JOIN                   |                             |  9489 |  3586K|  3656K|  7267   (1)| 00:01:28 |
|   3 |    NESTED LOOPS               |                             |  9489 |  3539K|       |  6592   (1)| 00:01:20 |
|   4 |     TABLE ACCESS BY INDEX ROWID| PROVINCIES                  |     1 |   124 |       |     1   (0)| 00:00:01 |
|*  5 |      INDEX UNIQUE SCAN        | SYS_C006713                 |     1 |       |       |     0   (0)| 00:00:01 |
|*  6 |     TABLE ACCESS BY INDEX ROWID| NWW_WEGVAK_EFEMERIDEN       |  9489 |  2390K|       |  6592   (1)| 00:01:20 |
|*  7 |      DOMAIN INDEX             | NWW_WEGVAK_EFEMERIDEN_SIDX  |       |       |       |     0   (0)| 00:00:01 |
|   8 |    INDEX FAST FULL SCAN       | SYS_C006716                 |   355K|  1737K|       |   206   (2)| 00:00:03 |
-----------------------------------------------------------------------------------------------------------------------
```

```
Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("WEE"."STT_ID"="STT"."ID")
   5 - access("P"."ID"=11)
   6 - filter(("WEE"."EINDDATUM" IS NULL OR "WEE"."EINDDATUM">=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd
              hh24:mi:ss')) AND "WEE"."BEGINDATUM"<=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd hh24:mi:ss'))
   7 - access("MDSYS"."SDO_ANYINTERACT"("WEE"."LIJN","P"."POLYGOON")='TRUE')

Statistics
==========================================================
      20557  recursive calls
          0  db block gets
     246185  consistent gets
      12484  physical reads
          0  redo size
        528  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          1  sorts (memory)
          0  sorts (disk)
          1  rows processed

TGEO ORATEST SQL> set autotrace off
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 07 With spatial selection (polygon with boundary of Delft municipality: 252 points)
TGEO ORATEST SQL> set autotrace on
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme, gga_gemeente_efemeriden gee
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is null)
  6      and  gme.naam = &gem_delft
  7      and  gee.gme_id = gme.id
  8      and  gee.begindatum <= &peildatum
  9      and (gee.einddatum  >= &peildatum or gee.einddatum is null)
 10      and  sdo_anyinteract(wee.lijn, gee.polygoon) = 'TRUE'
 11  ;
       COUNT(*)
===============
           4195
Elapsed: 00:00:00.20

Execution Plan
==========================================================
Plan hash value: 927064452

-------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                       |Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                            |    1 |  1002 |       | 14398   (1)| 00:02:53 |
|   1 |  SORT AGGREGATE               |                            |    1 |  1002 |       |            |          |
|*  2 |   HASH JOIN                   |                            |14489 |    13M|  5912K| 14398   (1)| 00:02:53 |
|   3 |    INDEX FAST FULL SCAN       | SYS_C006716                |  355K|  1737K|       |   206   (2)| 00:00:03 |
|   4 |    NESTED LOOPS               |                            |14489 |    13M|       | 13209   (1)| 00:02:39 |
|   5 |     NESTED LOOPS              |                            |    2 |  1478 |       |    15   (0)| 00:00:01 |
|   6 |      TABLE ACCESS BY INDEX ROWID| GGA_GEMEENTEN              |    1 |    15 |       |     2   (0)| 00:00:01 |
|*  7 |       INDEX RANGE SCAN        | GGA_GEMEENTEN_NAAM_IDX     |    1 |       |       |     1   (0)| 00:00:01 |
|*  8 |      TABLE ACCESS BY INDEX ROWID| GGA_GEMEENTE_EFEMERIDEN    |    2 |  1448 |       |    13   (0)| 00:00:01 |
|*  9 |       INDEX RANGE SCAN        | SYS_C006715                |   12 |       |       |     1   (0)| 00:00:01 |
|* 10 |     TABLE ACCESS BY INDEX ROWID | NWW_WEGVAK_EFEMERIDEN      | 9489 |  2390K|       | 13209   (1)| 00:02:39 |
|* 11 |      DOMAIN INDEX             | NWW_WEGVAK_EFEMERIDEN_SIDX |      |       |       |     0   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("WEE"."STT_ID"="STT"."ID")
   7 - access("GME"."NAAM"='Delft')
   8 - filter("GEE"."EINDDATUM" IS NULL OR "GEE"."EINDDATUM">=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd
              hh24:mi:ss'))
   9 - access("GEE"."GME_ID"="GME"."ID" AND "GEE"."BEGINDATUM"<=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd
              hh24:mi:ss'))
  10 - filter(("WEE"."EINDDATUM" IS NULL OR "WEE"."EINDDATUM">=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd
              hh24:mi:ss')) AND "WEE"."BEGINDATUM"<=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd hh24:mi:ss'))
  11 - access("MDSYS"."SDO_ANYINTERACT"("WEE"."LIJN","GEE"."POLYGOON")='TRUE')
```

```
Statistics
==========================================================
       3267  recursive calls
          0  db block gets
      14795  consistent gets
       1081  physical reads
          0  redo size
        527  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          1  sorts (memory)
          0  sorts (disk)
          1  rows processed

TGEO ORATEST SQL> -- 13 With spatial selection (window that includes the whole dataset)
TGEO ORATEST SQL> set autotrace on
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt
  3    where  wee.stt_id = stt.id
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  sdo_anyinteract(wee.lijn, &nl_query_window) = 'TRUE'
  7  ;

       COUNT(*)
===============
         948481

Elapsed: 00:00:10.29

Execution Plan
==========================================================
Plan hash value: 486044591

-------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                        | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                             |     1 |   263 |       |  7209   (1)| 00:01:27 |
|   1 |  SORT AGGREGATE               |                             |     1 |   263 |       |            |          |
|*  2 |   HASH JOIN                   |                             |  9489 |  2437K|  2504K|  7209   (1)| 00:01:27 |
|*  3 |    TABLE ACCESS BY INDEX ROWID| NWW_WEGVAK_EFEMERIDEN       |  9489 |  2390K|       |  6590   (1)| 00:01:20 |
|*  4 |     DOMAIN INDEX              | NWW_WEGVAK_EFEMERIDEN_SIDX  |       |       |       |     0   (0)| 00:00:01 |
|   5 |    INDEX FAST FULL SCAN       | SYS_C006716                 |   355K|  1737K|       |   206   (2)| 00:00:03 |
-------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("WEE"."STT_ID"="STT"."ID")
   3 - filter(("WEE"."EINDDATUM" IS NULL OR "WEE"."EINDDATUM">=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd
              hh24:mi:ss')) AND "WEE"."BEGINDATUM"<=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd hh24:mi:ss'))
   4 - access("MDSYS"."SDO_ANYINTERACT"("WEE"."LIJN","MDSYS"."SDO_GEOMETRY"(2003,28992,NULL,"SDO_ELEM_INFO_A
              RRAY"(1,1003,3),"SDO_ORDINATE_ARRAY"((-50000),250000,300000,650000)))='TRUE')

Statistics
==========================================================
     197899  recursive calls
          0  db block gets
    2691588  consistent gets
     170725  physical reads
          0  redo size
        528  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          1  sorts (memory)
          0  sorts (disk)
          1  rows processed

TGEO ORATEST SQL> set autotrace off
TGEO ORATEST SQL>
TGEO ORATEST SQL> -- 14 With spatial selection (boundary polygon of the Netherlands: 18631 points)
TGEO ORATEST SQL> set autotrace on
TGEO ORATEST SQL> select count(*)
  2    from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  3    where  wee.stt_id = stt.id
```

```
  4      and  wee.begindatum <= &peildatum
  5      and (wee.einddatum  >= &peildatum or wee.einddatum is NULL)
  6      and  p.id = &nederland
  7      and  sdo_anyinteract(wee.lijn, p.polygoon) = 'TRUE'
  8  ;

       COUNT(*)
===============
         948481

Elapsed: 00:00:37.58

Execution Plan
==========================================================
Plan hash value: 4199729836

-------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name                        | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                             |     1 |   387 |       |  7267   (1)| 00:01:28 |
|   1 |  SORT AGGREGATE               |                             |     1 |   387 |       |            |          |
|*  2 |   HASH JOIN                   |                             |  9489 |  3586K|  3656K|  7267   (1)| 00:01:28 |
|   3 |    NESTED LOOPS               |                             |  9489 |  3539K|       |  6592   (1)| 00:01:20 |
|   4 |     TABLE ACCESS BY INDEX ROWID| PROVINCIES                  |     1 |   124 |       |     1   (0)| 00:00:01 |
|*  5 |      INDEX UNIQUE SCAN        | SYS_C006713                 |     1 |       |       |     0   (0)| 00:00:01 |
|*  6 |     TABLE ACCESS BY INDEX ROWID| NWW_WEGVAK_EFEMERIDEN       |  9489 |  2390K|       |  6592   (1)| 00:01:20 |
|*  7 |      DOMAIN INDEX             | NWW_WEGVAK_EFEMERIDEN_SIDX  |       |       |       |     0   (0)| 00:00:01 |
|   8 |    INDEX FAST FULL SCAN       | SYS_C006716                 |   355K|  1737K|       |   206   (2)| 00:00:03 |
-------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("WEE"."STT_ID"="STT"."ID")
   5 - access("P"."ID"=0)
   6 - filter(("WEE"."EINDDATUM" IS NULL OR "WEE"."EINDDATUM">=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd
              hh24:mi:ss')) AND "WEE"."BEGINDATUM"<=TO_DATE(' 2011-07-30 12:00:00', 'syyyy-mm-dd hh24:mi:ss'))
   7 - access("MDSYS"."SDO_ANYINTERACT"("WEE"."LIJN","P"."POLYGOON")='TRUE')


Statistics
==========================================================
     198879  recursive calls
          0  db block gets
    2693392  consistent gets
     172791  physical reads
          0  redo size
        528  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          1  sorts (memory)
          0  sorts (disk)
          1  rows processed

TGEO ORATEST SQL> set autotrace off
```

B3.21 PostGIS script met ‘count’ queries

```
-- count_pg_queries.sql 02-10-2012  TT

\set peildatum 'to_timestamp(\'2011-07-30 12:00:00\',\'YYYY-MM-DD HH24:MI:SS\')'

\set    nederland 0
\set      prov_zh 8
\set prov_limburg 11
\set     nn_count 100
\set     buf_dist 500
\set     gem_list
'\'Amsterdam\',\'Rotterdam\',\'Utrecht\',\'Eindhoven\',\'Groningen\',\'Arnhem\''
\set    gem_delft '\'Delft\''

\set srid 28992
\set xmin -50000
\set xmax 300000
\set ymin 250000
\set ymax 650000

-- Small window: 2 x 2 km in Gemeente Groningen
\set small_window 230000,582000, 232000,584000

-- Medium window: 50 x 50 km, groot deel van Zeeland
\set medium_window 20000,360000, 70000,410000

-- Large window: 120 x 120 km in midden van Nederland
\set large_window 110000,430000, 230000,550000

-- NL window: groot window waar Nederland geheel in past
\set nl_window :xmin,:ymin, :xmax,:ymax

-- Lijn dwars door Nederland
\set nl_line '\'LINESTRING(197000 306000,185000 338000,196000 379000,156000
422000,120000 479000,114000 537000,126000 581000,189000 600000,276000 588000)\''

\set  small_query_window ST_MakeEnvelope(:small_window, :srid)
\set medium_query_window ST_MakeEnvelope(:medium_window, :srid)
\set  large_query_window ST_MakeEnvelope(:large_window, :srid)
\set     nl_query_window ST_MakeEnvelope(:nl_window, :srid)
\set           rd_origin ST_SetSRID(ST_MakePoint(155000,463000), :srid)

-- ------------------------------------------------------------------------
\timing on
--
-- 01 Basis, administratieve, query
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
;
--
-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
--
-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
```

```
;
--
-- 04 Medium window: 50 x 50 km, groot deel van Zeeland
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :medium_query_window)
;
--
-- 05 Large window: 120 x 120 km in midden van Nederland
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :large_query_window)
;
--
-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :prov_limburg
    and  ST_Intersects(wee.lijn, p.polygoon)
;
--
-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is null)
    and  gme.naam = :gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
    and  ST_Intersects(wee.lijn, gee.polygoon)
;
--
-- 08 Met berekening van lengte van de lijn
--
select wee.wvk_id, wee.begindatum, ST_Length(wee.lijn)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
    and  ST_Length(wee.lijn) > 100
  order by wee.wvk_id
;
--
-- 09 Selectie van gemeente polygons op basis van provincie polygon (ZH: 2569 punten)
--
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme, provincies p
  where  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is NULL)
    and  p.id = :prov_zh
    and  ST_Intersects(gee.polygoon, p.polygoon)
;
--
-- 10 Selectie van gemeente polygons op basis van lijn dwars door Nederland
--
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme
  where  gee.gme_id = gme.id
```

```
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is NULL)
    and  ST_Intersects(gee.polygoon, ST_GeomFromText(:nl_line, :srid))
;
--
-- 11 Selecteren van :nn_count dichtsbijzijnde lijnen (wegvakken) vanaf een punt
--
select wvk_id, naam, afstand from
 (select wee.wvk_id, stt.naam, ST_Distance(wee.lijn, :rd_origin) afstand,
      wee.lijn <#> :rd_origin bb_afstand
    from nww_wegvak_efemeriden wee, gga_straten stt
    where  wee.stt_id = stt.id
      and  wee.begindatum <= :peildatum
      and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    order by bb_afstand limit 500
 ) as foo
order by afstand, naam, wvk_id limit :nn_count
;
--
-- 12 Genereren van buffer rond een aantal gemeenten
--
create table pg_result12 as select gme.naam, ST_Buffer(gee.polygoon, :buf_dist)
gem_buffer
  from gga_gemeenten gme, gga_gemeente_efemeriden gee
  where  gme.naam in (:gem_list)
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
;
select naam, ST_Area(gem_buffer) from pg_result12 order by naam;
--
-- 13 Met ruimtelijke beperking (groot window waar Nederland geheel in past)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :nl_query_window)
;
--
-- 14 Met ruimtelijke beperking (polygon met grens van Nederland: 18631 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :nederland
    and  ST_Intersects(wee.lijn, p.polygoon)
;
\timing off
drop table pg_result12;
```

B3.22 PostGIS log voor ‘count’ queries

```
-- count_pg_queries.sql 02-10-2012  TT
\set peildatum 'to_timestamp(\'2011-07-30 12:00:00\',\'YYYY-MM-DD HH24:MI:SS\')'
\set    nederland 0
\set      prov_zh 8
\set prov_limburg 11
\set     nn_count 100
\set     buf_dist 500
\set     gem_list
'\'Amsterdam\',\'Rotterdam\',\'Utrecht\',\'Eindhoven\',\'Groningen\',\'Arnhem\''
\set    gem_delft '\'Delft\''
\set srid 28992
\set xmin -50000
\set xmax 300000
\set ymin 250000
\set ymax 650000
-- Small window: 2 x 2 km in Gemeente Groningen
\set small_window 230000,582000, 232000,584000
-- Medium window: 50 x 50 km, groot deel van Zeeland
\set medium_window 20000,360000, 70000,410000
-- Large window: 120 x 120 km in midden van Nederland
\set large_window 110000,430000, 230000,550000
-- NL window: groot window waar Nederland geheel in past
\set nl_window :xmin,:ymin, :xmax,:ymax
-- Lijn dwars door Nederland
\set nl_line '\'LINESTRING(197000 306000,185000 338000,196000 379000,156000
422000,120000 479000,114000 537000,126000 581000,189000 600000,276000 588000)\''
\set  small_query_window ST_MakeEnvelope(:small_window, :srid)
\set medium_query_window ST_MakeEnvelope(:medium_window, :srid)
\set  large_query_window ST_MakeEnvelope(:large_window, :srid)
\set     nl_query_window ST_MakeEnvelope(:nl_window, :srid)
\set           rd_origin ST_SetSRID(ST_MakePoint(155000,463000), :srid)
-- ------------------------------------------------------------------------
\timing on
Timing is on.
--
-- 01 Basis, administratieve, query
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
;
 count
--------
 948481
(1 row)

Time: 2832.097 ms
--
-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
 count
-------
 42607
(1 row)

Time: 1827.196 ms
--
-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
```

```
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
;
 count
-------
   524
(1 row)

Time: 30.273 ms
--
-- 04 Medium window: 50 x 50 km, groot deel van Zeeland
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :medium_query_window)
;
 count
-------
 31835
(1 row)

Time: 647.984 ms
--
-- 05 Large window: 120 x 120 km in midden van Nederland
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :large_query_window)
;
 count
--------
 326461
(1 row)

Time: 2119.434 ms
--
-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :prov_limburg
    and  ST_Intersects(wee.lijn, p.polygoon)
;
 count
-------
 72901
(1 row)

Time: 44553.556 ms
--
-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is null)
    and  gme.naam = :gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
    and  ST_Intersects(wee.lijn, gee.polygoon)
;
 count
-------
```

```
  4189
(1 row)

Time: 263.921 ms
--
-- 08 Met berekening van lengte van de lijn
--
select wee.wvk_id, wee.begindatum, ST_Length(wee.lijn)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
    and  ST_Length(wee.lijn) > 100
  order by wee.wvk_id
;
  wvk_id   |     begindatum      |     st_length
-----------+---------------------+--------------------
 459565016 | 2007-02-01 00:00:00 | 260.01201434742961
 459566001 | 2002-07-01 00:00:00 |  261.1438804829844
 459566006 | 2002-07-01 00:00:00 | 239.45385672381383
 459566010 | 2002-07-01 00:00:00 |  250.9034397929241
 459567032 | 2010-06-01 00:00:00 | 389.41504502649883
 460564008 | 2002-12-01 00:00:00 | 132.09466302617983
 460564009 | 2007-02-01 00:00:00 | 139.88994137535684
 460564026 | 2002-12-01 00:00:00 | 131.85598204101322
 460564032 | 2002-12-01 00:00:00 | 132.03408650799233
 460564034 | 2007-02-01 00:00:00 | 631.49006020520142
 460564045 | 2007-02-01 00:00:00 | 199.51820287315655
 460564049 | 2007-02-01 00:00:00 | 218.83130002385303
 460565004 | 2002-12-01 00:00:00 |  162.2409855221523
 460565006 | 2002-12-01 00:00:00 | 132.06059215375342
 460565008 | 2002-12-01 00:00:00 | 203.12282645449886
 460565009 | 2002-05-01 00:00:00 | 128.72451204024819
 460565011 | 2002-05-01 00:00:00 | 153.93239918566536
 460565025 | 2002-12-01 00:00:00 | 132.06059215375342
 460565026 | 2002-12-01 00:00:00 | 132.45754036671525
 460565028 | 2002-12-01 00:00:00 | 132.06059215375342
 460566001 | 2002-05-01 00:00:00 | 301.47824500617082
 460566018 | 2007-02-01 00:00:00 | 276.38760827824387
 460566020 | 2007-02-01 00:00:00 | 138.28180844524314
 460566021 | 2007-02-01 00:00:00 | 662.18745127521606
 460566022 | 2002-07-01 00:00:00 | 131.39060574432605
 460566024 | 2002-07-01 00:00:00 | 131.68192374203522
 460566026 | 2002-07-01 00:00:00 | 137.20684262883572
 460567001 | 2010-06-01 00:00:00 | 296.37205519022081
 460567003 | 2010-06-01 00:00:00 | 266.47380560376587
 460567004 | 2010-12-01 00:00:00 | 450.93389800818227
 460568020 | 2010-12-01 00:00:00 | 119.22709667219263
 460568021 | 2010-06-01 00:00:00 | 443.60153959439998
 461563003 | 2002-09-01 00:00:00 | 265.53738003743706
 461564014 | 2007-02-01 00:00:00 | 137.20329553767178
 461564029 | 2007-02-01 00:00:00 | 116.55752435602365
 461564031 | 2007-02-01 00:00:00 | 116.86862554592786
 461564036 | 2007-02-01 00:00:00 | 114.79490924249252
 461564044 | 2008-02-01 00:00:00 | 115.64902106376735
 461565004 | 1997-06-14 00:00:00 | 154.97080516665449
 461565006 | 2007-02-01 00:00:00 | 152.13427143460498
 461565009 | 1997-06-14 00:00:00 | 153.08023006426032
 461565021 | 2007-02-01 00:00:00 | 242.55105578344649
 461565022 | 2007-02-01 00:00:00 | 244.42774551570594
 461565024 | 2006-02-01 00:00:00 | 307.52596519744657
 461565025 | 2006-02-01 00:00:00 | 118.25596102094647
 461565028 | 2006-04-01 00:00:00 | 111.15764933191394
 461565029 | 2006-04-01 00:00:00 | 154.00656427573202
 461565030 | 2006-02-01 00:00:00 | 136.03126461955188
 461565031 | 2007-02-01 00:00:00 | 116.84233573505895
 461565033 | 2007-02-01 00:00:00 | 179.14122979022545
 461565034 | 2007-02-01 00:00:00 |  146.0822072681685
 461566002 | 1997-06-14 00:00:00 | 151.52887513606112
 461566003 | 1997-06-14 00:00:00 | 147.70578864756791
 461566013 | 1997-06-14 00:00:00 | 151.53248760373998
 461566015 | 1997-06-14 00:00:00 | 151.53777859439026
 461566020 | 1997-06-14 00:00:00 | 125.22779244241272
 461566021 | 1997-06-14 00:00:00 | 250.86367997805803
 461566022 | 1997-06-14 00:00:00 | 246.67792767088019
```

461566024 | 1997-06-14 00:00:00 | 185.86411156578026
461566026 | 1997-06-14 00:00:00 | 104.69479452198185
461566028 | 1997-11-27 00:00:00 | 146.45477117526761
461566029 | 1997-11-27 00:00:00 | 103.74020609066537
461566033 | 2001-05-01 00:00:00 | 236.11230015522932
461567012 | 2010-06-01 00:00:00 | 115.68090957698688
461567016 | 2010-06-01 00:00:00 | 104.84359565351119
461567018 | 2010-06-01 00:00:00 | 220.49022308418731
461567021 | 2010-06-01 00:00:00 | 166.47092420990842
461567023 | 2010-06-01 00:00:00 | 221.71733456163315
461567024 | 2010-06-01 00:00:00 | 125.69475288479886
461567030 | 2010-06-01 00:00:00 | 706.42122944797563
461567033 | 2010-06-01 00:00:00 | 122.2725564793851
461567034 | 2010-06-01 00:00:00 | 189.18381048642377
461567035 | 2010-06-01 00:00:00 | 299.69233239781687
461568002 | 2010-06-01 00:00:00 | 341.18773495564596
461568018 | 2005-02-01 00:00:00 | 387.11427817876455
461568019 | 2005-02-01 00:00:00 | 390.59328291741417
461568022 | 2010-06-01 00:00:00 | 640.48263154415201
462564002 | 1997-06-14 00:00:00 | 246.69664537136674
462564016 | 1997-06-14 00:00:00 | 131.24460134199052
462564019 | 1997-06-14 00:00:00 | 248.30783727356797
462564023 | 2007-02-01 00:00:00 | 331.10658740891603
462564024 | 2002-09-01 00:00:00 | 214.2199722191543
462564026 | 1997-06-14 00:00:00 | 117.64352935882194
462564033 | 2007-02-01 00:00:00 | 105.13881692794212
462564058 | 2007-02-01 00:00:00 | 119.28415344816119
462564075 | 2007-02-01 00:00:00 | 105.25087918870744
462564083 | 2007-02-01 00:00:00 | 167.35884309549121
462564084 | 2007-02-01 00:00:00 | 117.1388911719468
462565001 | 2002-09-01 00:00:00 | 106.25441167311595
462565004 | 1997-06-14 00:00:00 | 377.28463799152422
462565005 | 1997-06-14 00:00:00 | 125
462565006 | 1997-06-14 00:00:00 | 126.17844506887855
462565008 | 1997-06-14 00:00:00 | 152.73840332767023
462565025 | 1997-06-14 00:00:00 | 150.84137302156893
462565027 | 1997-06-14 00:00:00 | 141.6827441857335
462565034 | 1997-06-14 00:00:00 | 127.14038467016485
462565036 | 1997-06-14 00:00:00 | 130.5450113945378
462565041 | 2007-09-01 00:00:00 | 351.72716699168973
462565046 | 2002-07-01 00:00:00 | 154.33081351434652
462565047 | 2007-09-01 00:00:00 | 161.1794092259837
462565050 | 2007-09-01 00:00:00 | 157.45330967929496
462565054 | 2007-09-01 00:00:00 | 160.28144534076904
462565058 | 2007-09-01 00:00:00 | 158.37874286665124
462565061 | 2007-09-01 00:00:00 | 167.34790738397888
462566001 | 1997-06-14 00:00:00 | 151.82348847311019
462566002 | 1997-06-14 00:00:00 | 157.19096666157378
462566003 | 1997-06-14 00:00:00 | 160.73289064072762
462566007 | 1997-06-14 00:00:00 | 150.58093060888939
462566008 | 1997-06-14 00:00:00 | 153.68833275902082
462566010 | 1997-06-14 00:00:00 | 101.8082419494436
462566011 | 1997-06-14 00:00:00 | 164.12495239907915
462566019 | 1997-06-14 00:00:00 | 152.15361647808524
462566020 | 1997-06-14 00:00:00 | 247.07262436538144
462566021 | 1997-06-14 00:00:00 | 116.71281466844178
462566022 | 2004-04-01 00:00:00 | 120.44336488956669
462566023 | 2004-04-01 00:00:00 | 465.58920625832667
462566025 | 1997-06-14 00:00:00 | 547.95424064067936
462566026 | 2004-04-01 00:00:00 | 458.90942564516308
462567011 | 1997-06-14 00:00:00 | 117.95338062132852
462567012 | 1997-06-14 00:00:00 | 123.96773773849388
462567019 | 2002-06-01 00:00:00 | 123.97214562983966
462567026 | 2010-06-01 00:00:00 | 277.15643204613093
462567030 | 2010-06-01 00:00:00 | 289.51173245849134
462567036 | 2010-06-01 00:00:00 | 179.02342624594286
462567037 | 1997-11-11 00:00:00 | 328.80253391058926
462567040 | 2004-04-01 00:00:00 | 366.39935486407956
462567041 | 2010-06-01 00:00:00 | 107.66764074056512
462567044 | 2008-11-01 00:00:00 | 183.17211599733557
462567046 | 2004-04-01 00:00:00 | 162.83768534608879
462567048 | 2010-06-01 00:00:00 | 314.19355440935652
462567051 | 2007-02-01 00:00:00 | 128.95476137736378
462567053 | 2010-06-01 00:00:00 | 238.92144860853415
462567056 | 2010-06-01 00:00:00 | 425.62850351606033
462567057 | 2010-06-01 00:00:00 | 109.54123458058822

462568013 | 2007-02-01 00:00:00 | 527.11656330305027
462569010 | 2007-03-01 00:00:00 | 1305.6445483638511
463564002 | 2007-02-01 00:00:00 | 248.63438381080084
463564003 | 1997-06-14 00:00:00 | 251.71889884914788
463564005 | 2007-02-01 00:00:00 | 127.12985487288185
463564008 | 1997-06-14 00:00:00 | 127.75758294520134
463564016 | 2008-02-01 00:00:00 | 335.52745799261885
463564017 | 2008-02-01 00:00:00 | 330.68841870967401
463564019 | 2008-02-01 00:00:00 | 803.13886088790525
463564022 | 2008-02-01 00:00:00 | 801.39284928403674
463565004 | 1997-06-14 00:00:00 | 122.71104269787622
463565005 | 1997-06-14 00:00:00 | 112.89494246473379
463565006 | 1997-06-14 00:00:00 | 240.65241206738071
463565012 | 1997-06-14 00:00:00 | 104.04326023342406
463565013 | 1997-06-14 00:00:00 | 122.06555615733703
463565014 | 1997-06-14 00:00:00 | 122.06555615733703
463565016 | 1997-06-14 00:00:00 | 288.83626142218492
463565017 | 1997-06-14 00:00:00 | 221.36345062797457
463565020 | 1997-06-14 00:00:00 | 128.40009484282902
463565021 | 1997-06-14 00:00:00 | 124.27791436936813
463565022 | 1997-06-14 00:00:00 | 125.87615557576433
463565023 | 1997-06-14 00:00:00 | 127.12985487288185
463565025 | 1997-06-14 00:00:00 | 121.43310915891102
463565026 | 1997-06-14 00:00:00 | 115.78995520092192
463565029 | 2004-04-01 00:00:00 | 502.3817082518708
463565032 | 2004-04-01 00:00:00 | 498.55925104279265
463565034 | 2009-08-01 00:00:00 | 757.02104472756264
463566003 | 1997-06-14 00:00:00 | 117.95338062132852
463566005 | 1997-06-14 00:00:00 | 121.75795661885921
463566009 | 1997-06-14 00:00:00 | 124.91997438360288
463566010 | 1997-11-11 00:00:00 | 114.7946866796083
463566012 | 2004-03-01 00:00:00 | 117.95353589407043
463566015 | 1997-06-14 00:00:00 | 120.16655108639841
463566016 | 1997-06-14 00:00:00 | 122.71104269787622
463566018 | 1997-06-14 00:00:00 | 117.95338062132852
463566019 | 1997-06-14 00:00:00 | 116.05602095539895
463566021 | 1997-06-14 00:00:00 | 122.06555615733703
463566023 | 1997-06-14 00:00:00 | 122.06555615733703
463566025 | 1997-06-14 00:00:00 | 126.49110640673517
463566026 | 1997-06-14 00:00:00 | 125.22779244241272
463566027 | 1997-06-14 00:00:00 | 118.90332207301863
463566028 | 1997-06-14 00:00:00 | 121.1191883750877
463566029 | 1997-06-14 00:00:00 | 117.64352935882194
463566030 | 1997-06-14 00:00:00 | 117.95338062132852
463566031 | 1997-06-14 00:00:00 | 118.29740147203719
463566032 | 1997-06-14 00:00:00 | 115.43396380615195
463566033 | 1997-06-14 00:00:00 | 124.27791436936813
463566034 | 1997-06-14 00:00:00 | 176.17095069253918
463566036 | 1997-06-14 00:00:00 | 118.27087553578015
463566042 | 2002-06-01 00:00:00 | 134.98688457928733
463567019 | 1997-06-14 00:00:00 | 123.97132262304241
463567020 | 1997-06-14 00:00:00 | 124.29023838605403
463567022 | 1997-06-14 00:00:00 | 124.60476436661851
463567025 | 1997-06-14 00:00:00 | 118.60555539069249
463567026 | 1997-06-14 00:00:00 | 117.32433677630571
463567030 | 1997-06-14 00:00:00 | 124.27791436936813
463567031 | 1997-06-14 00:00:00 | 126.17844506887855
463567033 | 1997-06-14 00:00:00 | 120.17621590186533
463567034 | 1997-06-14 00:00:00 | 116.69190203266035
463567042 | 1997-06-14 00:00:00 | 147.05441169852742
463567044 | 1997-06-14 00:00:00 | 149.58085266251646
463567047 | 1997-06-14 00:00:00 | 138.19189556555045
463567048 | 1997-06-14 00:00:00 | 138.51353724455961
463567049 | 1997-06-14 00:00:00 | 129.40131679468701
463567050 | 2001-06-01 00:00:00 | 299.64520305767439
463567052 | 2000-01-01 00:00:00 | 255.94607355174458
463567053 | 2001-06-01 00:00:00 | 300.90227765271408
463567055 | 2000-01-01 00:00:00 | 252.5365089101162
463567057 | 1997-06-14 00:00:00 | 338.14677166245764
463567058 | 2010-06-01 00:00:00 | 561.42323822950402
463567061 | 1997-06-14 00:00:00 | 106.57613530838952
463567064 | 1997-06-14 00:00:00 | 117.64352935882194
463567068 | 2004-04-01 00:00:00 | 561.41826099433433
463567069 | 2007-02-01 00:00:00 | 292.64413506779641
463568034 | 1998-04-15 00:00:00 | 123.16655390161731
464564075 | 2009-08-01 00:00:00 | 776.62307283524467

```
(210 rows)

Time: 14.167 ms
--
-- 09 Selectie van gemeente polygons op basis van provincie polygon (ZH: 2569 punten)
--
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme, provincies p
  where  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is NULL)
    and  p.id = :prov_zh
    and  ST_Intersects(gee.polygoon, p.polygoon)
;
 count
-------
    94
(1 row)

Time: 66.464 ms
--
-- 10 Selectie van gemeente polygons op basis van lijn dwars door Nederland
--
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme
  where  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is NULL)
    and  ST_Intersects(gee.polygoon, ST_GeomFromText(:nl_line, :srid))
;
 count
-------
    62
(1 row)

Time: 110.476 ms
--
-- 11 Selecteren van :nn_count dichtsbijzijnde lijnen (wegvakken) vanaf een punt
--
select wvk_id, naam, afstand from
 (select wee.wvk_id, stt.naam, ST_Distance(wee.lijn, :rd_origin) afstand,
      wee.lijn <#> :rd_origin bb_afstand
    from nww_wegvak_efemeriden wee, gga_straten stt
    where  wee.stt_id = stt.id
      and  wee.begindatum <= :peildatum
      and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    order by bb_afstand limit 500
 ) as foo
order by afstand, naam, wvk_id limit :nn_count
;
  wvk_id   |           naam           |      afstand
-----------+--------------------------+--------------------
 310326069 | Krankeledenstraat        |  12.14107327947711
 309326070 | Breestraat               |                 25
 310326003 | Lieve Vrouwekerkhof      |                 25
 310325057 | Krankeledenstraat        | 32.015621187164243
 310325020 | Lieve Vrouwekerkhof      | 32.015621187164243
 309326018 | Westsingel               | 36.380020071675162
 310326027 | Lieve Vrouwestraat       |  69.50027233237995
 310326026 | Lieve Vrouwestraat       |  70.34912934784623
 310326028 | Lieve Vrouwestraat       | 79.752746736369986
 310326004 | Zwanenhalssteeg          | 79.752746736369986
 309326088 | Hellestraat              | 87.182058275234837
 309325137 | Varkensmarkt             | 87.885195044445084
 310325027 | Varkensmarkt             | 87.885195044445084
 309326104 | Hellestraat              | 90.578494406786504
 309325135 | Sint Jorisstraat         | 90.578494406786504
 309325138 | Utrechtsestraat          | 95.108072475493998
 309325136 | Varkensmarkt             | 95.108072475493998
 310325051 | Langestraat              | 96.797973407349687
 310325055 | Langestraat              |  97.07752411540578
 310325022 | Varkensmarkt             | 102.59142264341595
 310325056 | Langestraat              | 106.11018692850723
 310325038 | Scherbierstraat          | 106.11018692850723
 309325110 | Utrechtsestraat          | 111.00130152841513
 310325083 | Varkensmarkt             | 113.74813746608973
```

310325021	Zuidsingel	113.74813746608973
309326069	Bollebruggang	117.9594222849846
309326105	Breestraat	118.29231769225188
310326181	Paternosterstraat	118.29231769225188
310326060	Langegracht	118.91896247866168
310326156	Langegracht	118.91896247866168
309326089	Molenstraat	123.38622196179519
309326023	Westsingel	123.38622196179519
309326149	Hellestraat	124.25463201426064
309326160	Torenstraat	124.25463201426064
310326157	Langegracht	129.49517365523704
310326155	Marktgang	129.49517365523704
310326063	Lieve Vrouwestraat	136.51598652537584
309326087	Molenstraat	145.45446022724775
309325120	Arnhemsestraat	147.76332427229701
310325061	Koestraat	147.76332427229701
310326191	Langegracht	152.18433025774991
310326162	Krommestraat	153.60339186359332
309326086	Molenstraat	155.20631430454111
309326009	Stadhuisplein	155.20631430454111
310325050	Mooierstraat	156.38128936992797
310325039	Scherbierstraat	156.38128936992797
310326072	Kortegracht	156.64470115520422
310326054	Langestraat	156.64470115520422
310326039	Achter het Oude Stadhuis	158.30109639546666
310326158	Krommestraat	158.30109639546666
310326159	Krommestraat	158.96071263680309
309325133	Riddergang	160.69743148543682
309325134	Sint Jorisplein	162.26008708616311
310326161	Krommestraat	166.47529820666207
310326160	Peperstraat	166.47529820666207
309325111	Utrechtsestraat	167.72451737894286
309325114	Sint Jorisstraat	167.73307710764087
309326152	Molenstraat	168.43396332094071
310325108	Koesteeg	172.02906731131225
310325123	Koestraat	172.02906731131225
309326107	Breestraat	175.45304336205376
309326182	Kromme Elleboogsteeg	175.45304336205376
310326075	Achter de Heilige Geest	175.84981601924542
310326071	Kortegracht	175.84981601924542
309325123	Ad Arnhemse Poortwal	177.583951009066
310325046	Muurhuizen	178.94964919219112
310325122	Rozemarijnsteeg	178.94964919219112
310326055	Langestraat	179.95624482080009
310326076	Achter de Heilige Geest	181.45893347256126
310326194	Kortegracht	181.45893347256126
310325107	Haagplein	181.55791835388874
310325116	Koesteeg	181.55791835388874
310326190	Kerkgang	184.14741817359379
309326169	Langegracht	184.14741817359379
310326164	Hof	187.16928505497563
310325049	Mooierstraat	187.24008230076771
310325121	Schelvissteegje	187.24008230076771
309325122	Ad Arnhemse Poortwal	193.48761960654116
309325119	Slijkpoortsteeg	193.48761960654116
310326085	Hof	197.38140305509299
310326025	Vijver	197.38140305509299
310326056	Langestraat	198.50809346723361
310326030	Valkestraat	198.50809346723361
310326163	Hof	200.60210346106746
310326154	Krommestraat	201.75531537482928
310325058	Kortegracht	205.17207303139134
309325124	Ad Arnhemse Poortwal	211.18390966171933
310325115	Haagplein	212.12303236093175
309325121	Arnhemsestraat	213.57414542494473
309325139	Koesteeg	213.57414542494473
310326193	Valkestraat	214.72663375789415
309325009	Ad Arnhemse Poortwal	219.03516370206268
310325114	Haagplein	221.66884690909447
309326181	Kromme Elleboogsteeg	221.77759583195109
310325117	Koesteeg	222.67484727735135
310325124	Koesteeg	222.67484727735135
309325018	Stadsring	225.47599817719083
310326179	Hof	226.49559820227412
310326178	Hof	227.29738406986348
310326177	Windsteeg	227.29738406986348

```
(100 rows)

Time: 20.949 ms
--
-- 12 Genereren van buffer rond een aantal gemeenten
--
create table pg_result12 as select gme.naam, ST_Buffer(gee.polygoon, :buf_dist)
gem_buffer
  from gga_gemeenten gme, gga_gemeente_efemeriden gee
  where  gme.naam in (:gem_list)
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
;
SELECT 6
Time: 274.908 ms
select naam, ST_Area(gem_buffer) from pg_result12 order by naam;
   naam    |      st_area
-----------+--------------------
 Amsterdam | 267516378.07174951
 Arnhem    | 131257027.76113045
 Eindhoven | 112679016.92049798
 Groningen |  106372437.0229324
 Rotterdam |  393595016.5863139
 Utrecht   | 128165919.25595875
(6 rows)

Time: 0.889 ms
--
-- 13 Met ruimtelijke beperking (groot window waar Nederland geheel in past)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :nl_query_window)
;
 count
--------
 948481
(1 row)

Time: 5874.287 ms
--
-- 14 Met ruimtelijke beperking (polygon met grens van Nederland: 18631 punten)
--
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :nederland
    and  ST_Intersects(wee.lijn, p.polygoon)
;
 count
--------
 948481
(1 row)

Time: 884975.428 ms
\timing off
Timing is off.
drop table pg_result12;
DROP TABLE
```

B3.23 PostGIS script met ‘create’ queries

```
-- create_pg_queries.sql  06-09-2012  TT

\set peildatum 'to_timestamp(\'2011-07-30 12:00:00\',\'YYYY-MM-DD HH24:MI:SS\')'

\set    nederland 0
\set      prov_zh 8
\set prov_limburg 11
\set    gem_delft '\'Delft\''

\set srid 28992

-- Small window: 2 x 2 km in Gemeente Groningen
\set small_window 230000,582000, 232000,584000

-- Medium window: 50 x 50 km, groot deel van Zeeland
\set medium_window 20000,360000, 70000,410000

-- Large window: 120 x 120 km in midden van Nederland
\set large_window 110000,430000, 230000,550000

\set  small_query_window ST_MakeEnvelope(:small_window, :srid)
\set medium_query_window ST_MakeEnvelope(:medium_window, :srid)
\set  large_query_window ST_MakeEnvelope(:large_window, :srid)

-- ------------------------------------------------------------------------
\timing on
--
-- 01 Basis, administratieve, query
--
create table pg_result01 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
;
--
-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')
--
create table pg_result02 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
--
-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen
--
create table pg_result03 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
;
--
-- 04 Medium window: 50 x 50 km, groot deel van Zeeland
--
create table pg_result04 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :medium_query_window)
;
--
-- 05 Large window: 120 x 120 km in midden van Nederland
--
create table pg_result05 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
```

```
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :large_query_window)
;
--
-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)
--
create table pg_result06 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :prov_limburg
    and  ST_Intersects(wee.lijn, p.polygoon)
;
--
-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)
--
create table pg_result07 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is null)
    and  gme.naam = :gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
    and  ST_Intersects(wee.lijn, gee.polygoon)
;
--
\timing off
drop table pg_result01;
drop table pg_result02;
drop table pg_result03;
drop table pg_result04;
drop table pg_result05;
drop table pg_result06;
drop table pg_result07;
```

B3.24 PostGIS log voor ‘create’ queries

```
-- create_pg_queries.sql  06-09-2012  TT
\set peildatum 'to_timestamp(\'2011-07-30 12:00:00\',\'YYYY-MM-DD HH24:MI:SS\')'
\set    nederland 0
\set      prov_zh 8
\set prov_limburg 11
\set    gem_delft '\'Delft\''
\set srid 28992
\set xmin -50000
\set xmax 300000
\set ymin 250000
\set ymax 650000
-- Small window: 2 x 2 km in Gemeente Groningen
\set small_window 230000,582000, 232000,584000
-- Medium window: 50 x 50 km, groot deel van Zeeland
\set medium_window 20000,360000, 70000,410000
-- Large window: 120 x 120 km in midden van Nederland
\set large_window 110000,430000, 230000,550000
\set  small_query_window ST_MakeEnvelope(:small_window, :srid)
\set medium_query_window ST_MakeEnvelope(:medium_window, :srid)
\set  large_query_window ST_MakeEnvelope(:large_window, :srid)
-- ------------------------------------------------------------------------
\timing on
Timing is on.
--
-- 01 Basis, administratieve, query
--
create table pg_result01 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
;
SELECT 948481
Time: 6117.284 ms
--
-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')
--
create table pg_result02 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
SELECT 42607
Time: 2085.041 ms
--
-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen
--
create table pg_result03 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
;
SELECT 524
Time: 46.450 ms
--
-- 04 Medium window: 50 x 50 km, groot deel van Zeeland
--
create table pg_result04 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :medium_query_window)
;
SELECT 31835
Time: 639.770 ms
--
```

```
-- 05 Large window: 120 x 120 km in midden van Nederland
--
create table pg_result05 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :large_query_window)
;
SELECT 326461
Time: 3816.671 ms
--
-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)
--
create table pg_result06 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :prov_limburg
    and  ST_Intersects(wee.lijn, p.polygoon)
;
SELECT 72901
Time: 44044.320 ms
--
-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)
--
create table pg_result07 as select wee.wvk_id, wee.lijn, stt.id
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is null)
    and  gme.naam = :gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
    and  ST_Intersects(wee.lijn, gee.polygoon)
;
SELECT 4189
Time: 186.441 ms
--
\timing off
Timing is off.
drop table pg_result01;
DROP TABLE
drop table pg_result02;
DROP TABLE
drop table pg_result03;
DROP TABLE
drop table pg_result04;
DROP TABLE
drop table pg_result05;
DROP TABLE
drop table pg_result06;
DROP TABLE
drop table pg_result07;
DROP TABLE
```

B3.25 PostGIS log met query plannen

```
-- And/of with spatial selection (rectangular window, in different sizes)
--
-- 03 Small window: 2 x 2 km in municipality Groningen
--
explain (analyze, buffers)
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :small_query_window)
;

QUERY PLAN
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------
 Aggregate  (cost=7747.25..7747.26 rows=1 width=0) (actual time=14.549..14.549 rows=1 loops=1)
   Buffers: shared hit=2151 read=176
   ->  Nested Loop  (cost=57.31..7746.81 rows=176 width=0) (actual time=2.990..14.413 rows=524 loops=1)
         Buffers: shared hit=2151 read=176
         ->  Bitmap Heap Scan on nww_wegvak_efemeriden wee  (cost=57.31..6350.64 rows=176 width=6) (actual
time=2.956..9.886 rows=524 loops=1)
               Recheck Cond: (lijn &&
'0103000020407100000100000005000000000000080130C4100000000E0C2214100000000080130C410000000080D2214100000000000520C
410000000080D2214100000000000520C4100000000E0C2214100000000080130C4100000000E0C22141'::geometry)
               Filter: ((begindatum <= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text))
AND ((einddatum >= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text)) OR (einddatum IS
NULL)) AND _st_intersects(lijn,
'0103000020407100000100000005000000000000080130C4100000000E0C2214100000000080130C410000000080D2214100000000000520C
410000000080D2214100000000000520C4100000000E0C2214100000000080130C4100000000E0C22141'::geometry))
               Buffers: shared hit=108 read=123
               ->  Bitmap Index Scan on nww_wegvak_efemeriden_sidx  (cost=0.00..57.26 rows=1634 width=0) (actual
time=1.255..1.255 rows=1303 loops=1)
                     Index Cond: (lijn &&
'0103000020407100000100000005000000000000080130C4100000000E0C2214100000000080130C410000000080D2214100000000000520C
410000000080D2214100000000000520C4100000000E0C2214100000000080130C4100000000E0C22141'::geometry)
                     Buffers: shared read=41
         ->  Index Scan using gga_straten_pkey on gga_straten stt  (cost=0.00..7.92 rows=1 width=6) (actual
time=0.007..0.007 rows=1 loops=524)
               Index Cond: (id = wee.stt_id)
               Buffers: shared hit=2043 read=53
 Total runtime: 14.793 ms
(15 rows)

Time: 48.215 ms
--
-- 05 Large window: 120 x 120 km in the middle of the Netherlands
--
explain (analyze, buffers)
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :large_query_window)
;

QUERY PLAN
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
-------------------------
 Aggregate  (cost=255783.50..255783.51 rows=1 width=0) (actual time=3107.239..3107.239 rows=1 loops=1)
   Buffers: shared hit=9165 read=41599
   ->  Hash Join  (cost=64924.02..255497.15 rows=114538 width=0) (actual time=742.332..3054.000 rows=326461
loops=1)
         Hash Cond: (wee.stt_id = stt.id)
         Buffers: shared hit=9165 read=41599
         ->  Bitmap Heap Scan on nww_wegvak_efemeriden wee  (cost=52981.48..241407.02 rows=114538 width=6)
(actual time=491.278..2384.221 rows=326461 loops=1)
```

```
               Recheck Cond: (((einddatum >= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD
HH24:MI:SS'::text)) OR (einddatum IS NULL)) AND (lijn &&
'0103000020407100000100000005000000000000000000DBFA4000000000C03E1A410000000000DBFA4000000000E0C820410000000080130C
4100000000E0C8204100000000080130C4100000000C03E1A410000000000DBFA4000000000C03E1A41'::geometry))
               Filter: ((begindatum <= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text))
AND _st_intersects(lijn,
'0103000020407100000100000005000000000000000000DBFA4000000000C03E1A410000000000DBFA4000000000E0C820410000000080130C
4100000000E0C8204100000000080130C4100000000C03E1A410000000000DBFA4000000000C03E1A41'::geometry))
               Buffers: shared hit=5230 read=41599
               ->  BitmapAnd  (cost=52981.48..52981.48 rows=343613 width=0) (actual time=479.271..479.271 rows=0
loops=1)
                     Buffers: shared hit=5191 read=7250
                     ->  BitmapOr  (cost=17560.75..17560.75 rows=946977 width=0) (actual time=136.815..136.815
rows=0 loops=1)
                           Buffers: shared hit=5190
                           ->  Bitmap Index Scan on nww_wegvak_efn_enddat_idx  (cost=0.00..4.58 rows=1 width=0)
(actual time=24.064..24.064 rows=0 loops=1)
                                 Index Cond: (einddatum >= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD
HH24:MI:SS'::text))
                                 Buffers: shared hit=2595
                           ->  Bitmap Index Scan on nww_wegvak_efn_enddat_idx  (cost=0.00..17498.90 rows=946977
width=0) (actual time=112.747..112.747 rows=948481 loops=1)
                                 Index Cond: (einddatum IS NULL)
                                 Buffers: shared hit=2595
                     ->  Bitmap Index Scan on nww_wegvak_efemeriden_sidx  (cost=0.00..35391.84 rows=1060645
width=0) (actual time=325.116..325.116 rows=1053213 loops=1)
                           Index Cond: (lijn &&
'0103000020407100000100000005000000000000000000DBFA4000000000C03E1A410000000000DBFA4000000000E0C820410000000080130C
4100000000E0C8204100000000080130C4100000000C03E1A410000000000DBFA4000000000C03E1A41'::geometry)

                           Buffers: shared hit=1 read=7250
         ->  Hash  (cost=7493.91..7493.91 rows=355891 width=6) (actual time=250.791..250.791 rows=355891 loops=1)
               Buckets: 65536  Batches: 1  Memory Usage: 13536kB
               Buffers: shared hit=3935
               ->  Seq Scan on gga_straten stt  (cost=0.00..7493.91 rows=355891 width=6) (actual
time=0.010..92.354 rows=355891 loops=1)
                     Buffers: shared hit=3935
 Total runtime: 3107.320 ms
(28 rows)

Time: 3110.420 ms
--
-- 06 With spatial selection (polygon with boundary of 'Limburg' province: 7946 points)
--
explain (analyze, buffers)
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :prov_limburg
    and  ST_Intersects(wee.lijn, p.polygoon)
;

QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
-----------
 Aggregate  (cost=13042.40..13042.41 rows=1 width=0) (actual time=43476.122..43476.123 rows=1 loops=1)
   Buffers: shared hit=830902 read=25009
   ->  Hash Join  (cost=11942.55..12921.00 rows=48563 width=0) (actual time=163.542..43448.228 rows=72901
loops=1)
         Hash Cond: (wee.stt_id = stt.id)
         Buffers: shared hit=830902 read=25009
         ->  Nested Loop  (cost=0.00..67.89 rows=48563 width=6) (actual time=4.380..43115.361 rows=72901 loops=1)
               Join Filter: _st_intersects(wee.lijn, p.polygoon)
               Buffers: shared hit=830865 read=21111
               ->  Seq Scan on provincies p  (cost=0.00..1.16 rows=1 width=355945) (actual time=0.014..0.017
rows=1 loops=1)
                     Filter: (id = 11::numeric)
                     Buffers: shared read=1
               ->  Index Scan using nww_wegvak_efemeriden_sidx on nww_wegvak_efemeriden wee  (cost=0.00..65.42
rows=5 width=483) (actual time=0.196..1255.155 rows=105452 loops=1)
                     Index Cond: (lijn && p.polygoon)
```

```
                     Filter: ((begindatum <= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD
HH24:MI:SS'::text)) AND ((einddatum >= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text))
OR (einddatum IS NULL)))
                     Buffers: shared hit=92688 read=21101
          ->  Hash  (cost=7493.91..7493.91 rows=355891 width=6) (actual time=159.110..159.110 rows=355891 loops=1)
                Buckets: 65536  Batches: 1  Memory Usage: 13536kB
                Buffers: shared hit=37 read=3898
                ->  Seq Scan on gga_straten stt  (cost=0.00..7493.91 rows=355891 width=6) (actual
time=0.007..64.998 rows=355891 loops=1)
                     Buffers: shared hit=37 read=3898
 Total runtime: 43477.399 ms
(21 rows)

Time: 43479.670 ms
--
-- 07 With spatial selection (polygon with boundary of Delft municipality: 252 points)
--
explain (analyze, buffers)
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme, gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is null)
    and  gme.naam = :gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= :peildatum
    and (gee.einddatum  >= :peildatum or gee.einddatum is null)
    and  ST_Intersects(wee.lijn, gee.polygoon)
;

QUERY PLAN
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
-------------------------------------------
 Aggregate  (cost=358.22..358.23 rows=1 width=0) (actual time=158.345..158.345 rows=1 loops=1)
   Buffers: shared hit=19812 read=1800
   ->  Nested Loop  (cost=20.28..358.12 rows=37 width=0) (actual time=1.667..157.598 rows=4189 loops=1)
         Buffers: shared hit=19812 read=1800
         ->  Nested Loop  (cost=20.28..97.36 rows=37 width=6) (actual time=1.648..136.001 rows=4189 loops=1)
               Join Filter: _st_intersects(wee.lijn, gee.polygoon)
               Buffers: shared hit=3107 read=1724
               ->  Nested Loop  (cost=20.28..32.58 rows=1 width=38167) (actual time=0.156..0.159 rows=1 loops=1)
                     Buffers: shared hit=5 read=11
                     ->  Index Scan using gga_gemeenten_naam_idx on gga_gemeenten gme  (cost=0.00..8.27 rows=1
width=5) (actual time=0.029..0.030 rows=1 loops=1)
                           Index Cond: ((naam)::text = 'Delft'::text)
                           Buffers: shared read=4
                     ->  Bitmap Heap Scan on gga_gemeente_efemeriden gee  (cost=20.28..24.30 rows=1 width=38172)
(actual time=0.124..0.125 rows=1 loops=1)
                           Recheck Cond: ((gme_id = gme.id) AND (begindatum <= to_timestamp('2011-07-30
12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text)) AND ((einddatum >= to_timestamp('2011-07-30 12:00:00'::text,
'YYYY-MM-DD HH24:MI:SS'::text)) OR (einddatum IS NULL)))
                           Buffers: shared hit=5 read=7
                           ->  BitmapAnd  (cost=20.28..20.28 rows=1 width=0) (actual time=0.108..0.108 rows=0
loops=1)
                                 Buffers: shared hit=5 read=6
                                 ->  Bitmap Index Scan on gga_gemeente_efemeriden_pkey  (cost=0.00..4.37 rows=12
width=0) (actual time=0.023..0.023 rows=16 loops=1)
                                       Index Cond: ((gme_id = gme.id) AND (begindatum <= to_timestamp('2011-07-30
12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text)))
                                       Buffers: shared hit=1 read=1
                                 ->  BitmapOr  (cost=15.65..15.65 rows=418 width=0) (actual time=0.078..0.078
rows=0 loops=1)
                                       Buffers: shared hit=4 read=5
                                       ->  Bitmap Index Scan on gga_gem_efn_enddat_idx  (cost=0.00..4.26 rows=1
width=0) (actual time=0.028..0.028 rows=0 loops=1)
                                             Index Cond: (einddatum >= to_timestamp('2011-07-30 12:00:00'::text,
'YYYY-MM-DD HH24:MI:SS'::text))
                                             Buffers: shared read=5
                                       ->  Bitmap Index Scan on gga_gem_efn_enddat_idx  (cost=0.00..11.39
rows=418 width=0) (actual time=0.049..0.049 rows=418 loops=1)
                                             Index Cond: (einddatum IS NULL)
                                             Buffers: shared hit=4
               ->  Index Scan using nww_wegvak_efemeriden_sidx on nww_wegvak_efemeriden wee  (cost=0.00..63.47
rows=5 width=483) (actual time=0.093..44.568 rows=5276 loops=1)
                     Index Cond: (lijn && gee.polygoon)
```

```
                    Filter: ((begindatum <= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD
HH24:MI:SS'::text)) AND ((einddatum >= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text))
OR (einddatum IS NULL)))
                    Buffers: shared hit=3102 read=1713
         ->  Index Scan using gga_straten_pkey on gga_straten stt  (cost=0.00..7.04 rows=1 width=6) (actual
time=0.004..0.004 rows=1 loops=4189)
               Index Cond: (id = wee.stt_id)
               Buffers: shared hit=16705 read=76
 Total runtime: 158.461 ms
(36 rows)

Time: 161.333 ms
--
-- 13 With spatial selection (window that includes the whole dataset)
--
explain (analyze, buffers)
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  ST_Intersects(wee.lijn, :nl_query_window)
;

QUERY PLAN
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
---
 Aggregate  (cost=389153.84..389153.85 rows=1 width=0) (actual time=5632.352..5632.352 rows=1 loops=1)
   Buffers: shared hit=8741 read=64733
   ->  Hash Join  (cost=29603.86..388364.69 rows=315659 width=0) (actual time=265.613..5507.568 rows=948481
loops=1)
         Hash Cond: (wee.stt_id = stt.id)
         Buffers: shared hit=8741 read=64733
         ->  Bitmap Heap Scan on nww_wegvak_efemeriden wee  (cost=17661.31..370503.54 rows=315659 width=6)
(actual time=109.696..4361.212 rows=948481 loops=1)
               Recheck Cond: ((einddatum >= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD
HH24:MI:SS'::text)) OR (einddatum IS NULL))
               Filter: ((lijn &&
'0103000020407100000100000005000000000000000006AE8C00000000080840E4100000000006AE8C00000000020D6234100000000804F12
410000000020D6234100000000804F1241000000000080840E4100000000006AE8C00000000080840E41'::geometry) AND (begindatum <=
to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text)) AND _st_intersects(lijn,
'0103000020407100000100000005000000000000000006AE8C00000000080840E4100000000006AE8C00000000020D6234100000000804F12
410000000020D6234100000000804F1241000000000080840E4100000000006AE8C00000000080840E41'::geometry))
               Buffers: shared hit=8565 read=60974
               ->  BitmapOr  (cost=17661.31..17661.31 rows=946977 width=0) (actual time=93.710..93.710 rows=0
loops=1)
                     Buffers: shared hit=2598 read=2592
                     ->  Bitmap Index Scan on nww_wegvak_efn_enddat_idx  (cost=0.00..4.58 rows=1 width=0) (actual
time=20.540..20.540 rows=0 loops=1)
                           Index Cond: (einddatum >= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD
HH24:MI:SS'::text))
                           Buffers: shared hit=3 read=2592
                     ->  Bitmap Index Scan on nww_wegvak_efn_enddat_idx  (cost=0.00..17498.90 rows=946977
width=0) (actual time=73.167..73.167 rows=948481 loops=1)
                           Index Cond: (einddatum IS NULL)
                           Buffers: shared hit=2595
         ->  Hash  (cost=7493.91..7493.91 rows=355891 width=6) (actual time=155.869..155.869 rows=355891 loops=1)
               Buckets: 65536  Batches: 1  Memory Usage: 13536kB
               Buffers: shared hit=176 read=3759
               ->  Seq Scan on gga_straten stt  (cost=0.00..7493.91 rows=355891 width=6) (actual
time=0.006..65.252 rows=355891 loops=1)
                     Buffers: shared hit=176 read=3759
 Total runtime: 5633.751 ms
(23 rows)

Time: 5634.835 ms
--
-- 14 With spatial selection (boundary polygon of the Netherlands: 18631 points)
--
explain (analyze, buffers)
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
```

```
  where  wee.stt_id = stt.id
    and  wee.begindatum <= :peildatum
    and (wee.einddatum  >= :peildatum or wee.einddatum is NULL)
    and  p.id = :nederland
    and  ST_Intersects(wee.lijn, p.polygoon)
;

QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
-----------
 Aggregate  (cost=13042.40..13042.41 rows=1 width=0) (actual time=889074.276..889074.276 rows=1 loops=1)
   Buffers: shared hit=15835119 read=227085
   ->  Hash Join  (cost=11942.55..12921.00 rows=48563 width=0) (actual time=168.997..888712.180 rows=948481
loops=1)
         Hash Cond: (wee.stt_id = stt.id)
         Buffers: shared hit=15835119 read=227085
         ->  Nested Loop  (cost=0.00..67.89 rows=48563 width=6) (actual time=10.099..886304.587 rows=948481
loops=1)
               Join Filter: _st_intersects(wee.lijn, p.polygoon)
               Buffers: shared hit=15835117 read=223152
               ->  Seq Scan on provincies p  (cost=0.00..1.16 rows=1 width=355945) (actual time=0.011..0.016
rows=1 loops=1)
                     Filter: (id = 0::numeric)
                     Buffers: shared read=1
               ->  Index Scan using nww_wegvak_efemeriden_sidx on nww_wegvak_efemeriden wee  (cost=0.00..65.42
rows=5 width=483) (actual time=0.160..10711.350 rows=948481 loops=1)
                     Index Cond: (lijn && p.polygoon)
                     Filter: ((begindatum <= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD
HH24:MI:SS'::text)) AND ((einddatum >= to_timestamp('2011-07-30 12:00:00'::text, 'YYYY-MM-DD HH24:MI:SS'::text))
OR (einddatum IS NULL)))
                     Buffers: shared hit=659421 read=223135
         ->  Hash  (cost=7493.91..7493.91 rows=355891 width=6) (actual time=158.843..158.843 rows=355891 loops=1)
               Buckets: 65536  Batches: 1  Memory Usage: 13536kB
               Buffers: shared hit=2 read=3933
               ->  Seq Scan on gga_straten stt  (cost=0.00..7493.91 rows=355891 width=6) (actual
time=0.013..64.982 rows=355891 loops=1)
                     Buffers: shared hit=2 read=3933
 Total runtime: 889077.191 ms
(21 rows)

             Time: 889079.183 ms
```

B3.26 SQL Server script met ‘count’ queries

```
-- count_ss_queries.sql  03-10-2012  TT

DECLARE @peildatum datetime = '2011-07-30T12:00:00';

DECLARE    @nederland numeric(10) = 0;
DECLARE      @prov_zh numeric(10) = 8;
DECLARE @prov_limburg numeric(10) = 11;
DECLARE     @nn_count     integer = 100;
DECLARE     @buf_dist       float = 500;
DECLARE    @gem_delft varchar(10) = 'Delft';

-- Small window: 2 x 2 km in Gemeente Groningen
DECLARE @small_query_window geometry = geometry::STLineFromText('LINESTRING (230000
582000,232000 584000)',28992).STEnvelope();

-- Medium window: 50 x 50 km, groot deel van Zeeland
DECLARE @medium_query_window geometry = geometry::STLineFromText('LINESTRING (20000
360000,70000 410000)',28992).STEnvelope();

-- Large window: 120 x 120 km in midden van Nederland
DECLARE @large_query_window geometry = geometry::STLineFromText('LINESTRING (110000
430000,230000 550000)',28992).STEnvelope();

-- NL window: groot window waar Nederland geheel in past
DECLARE @nl_query_window geometry = geometry::STLineFromText('LINESTRING (-50000
250000,300000 650000)',28992).STEnvelope();

-- Lijn dwars door Nederland
DECLARE @nl_line geometry = geometry::STLineFromText('LINESTRING (197000 306000,185000
338000,196000 379000,156000 422000,120000 479000,114000 537000,126000 581000,189000
600000,276000 588000)',28992);

-- Punt op (oude) oorsprong RD
DECLARE @rd_origin geometry = geometry::STPointFromText('POINT (155000
463000)',28992);

-- ------------------------------------------------------------------------

-- 01 Basis, administratieve, query

PRINT 'Query 01';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
;
SET STATISTICS TIME OFF

-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')

PRINT 'Query 02';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
SET STATISTICS TIME OFF

-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen

PRINT 'Query 03';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
```

```
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@small_query_window) = 1
;
SET STATISTICS TIME OFF

-- 04 Medium window: 50 x 50 km, groot deel van Zeeland

PRINT 'Query 04';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@medium_query_window) = 1
;
SET STATISTICS TIME OFF

-- 05 Large window: 120 x 120 km in midden van Nederland

PRINT 'Query 05';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@large_query_window) = 1
;
SET STATISTICS TIME OFF

-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)

PRINT 'Query 06';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  p.id = @prov_limburg
    and  wee.lijn.STIntersects(p.polygoon) = 1
;
SET STATISTICS TIME OFF

-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)

PRINT 'Query 07';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is null)
    and  gme.naam = @gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is null)
    and  wee.lijn.STIntersects(gee.polygoon) = 1
;
SET STATISTICS TIME OFF

-- 08 Met berekening van lengte van de lijn

PRINT 'Query 08';
SET STATISTICS TIME ON
select wee.wvk_id, wee.begindatum, wee.lijn.STLength() wegvak_lengte
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@small_query_window) = 1
    and  wee.lijn.STLength() > 100
  order by wee.wvk_id
;
```

```
SET STATISTICS TIME OFF

-- 09 Selectie van gemeente polygons op basis van provincie polygon (ZH: 2569 punten)

PRINT 'Query 09';
SET STATISTICS TIME ON
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme, provincies p
  where  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is NULL)
    and  p.id = @prov_zh
    and  gee.polygoon.STIntersects(p.polygoon) = 1
;
SET STATISTICS TIME OFF

-- 10 Selectie van gemeente polygons op basis van lijn dwars door Nederland

PRINT 'Query 10';
SET STATISTICS TIME ON
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme
  where  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is NULL)
    and  gee.polygoon.STIntersects(@nl_line) = 1
;
SET STATISTICS TIME OFF

-- 11 Selecteren van @nn_count dichtsbijzijnde lijnen (wegvakken) vanaf een punt

PRINT 'Query 11';
SET STATISTICS TIME ON
select top(@nn_count) wee.wvk_id, stt.naam, wee.lijn.STDistance(@rd_origin) afstand
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is null)
    and  wee.lijn.STDistance(@rd_origin) is not NULL
  order by wee.lijn.STDistance(@rd_origin), naam, wvk_id
;
SET STATISTICS TIME OFF

-- 12 Genereren van buffer rond een aantal gemeenten

PRINT 'Query 12';
SET STATISTICS TIME ON
select gme.naam, gee.polygoon.STBuffer(@buf_dist) gem_buffer into ss_result12
  from gga_gemeenten gme, gga_gemeente_efemeriden gee
  where  gme.naam in
('Amsterdam','Rotterdam','Utrecht','Eindhoven','Groningen','Arnhem')
    and  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is null)
;
select naam, gem_buffer.STArea() buffer_area from ss_result12 order by naam;
SET STATISTICS TIME OFF

-- 13 Met ruimtelijke beperking (groot window waar Nederland geheel in past)

PRINT 'Query 13';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@nl_query_window) = 1
;
SET STATISTICS TIME OFF

-- 14 Met ruimtelijke beperking (polygon met grens van Nederland: 18631 punten)

PRINT 'Query 14';
SET STATISTICS TIME ON
select count(*)
```

```
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  p.id = @nederland
    and  wee.lijn.STIntersects(p.polygoon) = 1
;
SET STATISTICS TIME OFF
drop table ss_result12;
PRINT 'Finished';
GO
```

B3.27 SQL Server log voor ‘count’ queries

Output van ‘`select name, value from v$system_parameter;`’ voor de Oracle database gebruikt bij het onderzoek.

```
-- count_ss_queries.sql 14-09-2012  TT

DECLARE @peildatum datetime = '2011-07-30T12:00:00';

DECLARE    @nederland numeric(10) = 0;
DECLARE      @prov_zh numeric(10) = 8;
DECLARE @prov_limburg numeric(10) = 11;
DECLARE     @nn_count numeric(10) = 100;
DECLARE     @buf_dist       float = 500;
DECLARE    @gem_delft varchar(10) = 'Delft';

-- Small window: 2 x 2 km in Gemeente Groningen
DECLARE @small_query_window geometry = geometry::STLineFromText('LINESTRING (230000
582000,232000 584000)',28992).STEnvelope();

-- Medium window: 50 x 50 km, groot deel van Zeeland
DECLARE @medium_query_window geometry = geometry::STLineFromText('LINESTRING (20000
360000,70000 410000)',28992).STEnvelope();

-- Large window: 120 x 120 km in midden van Nederland
DECLARE @large_query_window geometry = geometry::STLineFromText('LINESTRING (110000
430000,230000 550000)',28992).STEnvelope();

-- NL window: groot window waar Nederland geheel in past
DECLARE @nl_query_window geometry = geometry::STLineFromText('LINESTRING (-50000
250000,300000 650000)',28992).STEnvelope();

-- Lijn dwars door Nederland
DECLARE @nl_line geometry = geometry::STLineFromText('LINESTRING (197000 306000,185000
338000,196000 379000,156000 422000,120000 479000,114000 537000,126000 581000,189000
600000,276000 588000)',28992);

-- Punt op (oude) oorsprong RD
DECLARE @rd_origin geometry = geometry::STPointFromText('POINT (155000
463000)',28992);

-- ------------------------------------------------------------------------

-- 01 Basis, administratieve, query

PRINT 'Query 01';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
;
SET STATISTICS TIME OFF

-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')

PRINT 'Query 02';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
SET STATISTICS TIME OFF

-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen

PRINT 'Query 03';
SET STATISTICS TIME ON
```

```
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@small_query_window) = 1
;
SET STATISTICS TIME OFF

-- 04 Medium window: 50 x 50 km, groot deel van Zeeland

PRINT 'Query 04';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@medium_query_window) = 1
;
SET STATISTICS TIME OFF

-- 05 Large window: 120 x 120 km in midden van Nederland

PRINT 'Query 05';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@large_query_window) = 1
;
SET STATISTICS TIME OFF

-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)

PRINT 'Query 06';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  p.id = @prov_limburg
    and  wee.lijn.STIntersects(p.polygoon) = 1
;
SET STATISTICS TIME OFF

-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)

PRINT 'Query 07';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is null)
    and  gme.naam = @gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is null)
    and  wee.lijn.STIntersects(gee.polygoon) = 1
;
SET STATISTICS TIME OFF

-- 08 Met berekening van lengte van de lijn

PRINT 'Query 08';
SET STATISTICS TIME ON
select wee.wvk_id, wee.begindatum, wee.lijn.STLength() wegvak_lengte
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
```

```
    and  wee.lijn.STIntersects(@small_query_window) = 1
    and  wee.lijn.STLength() > 100
  order by wee.wvk_id
;
SET STATISTICS TIME OFF

-- 09 Selectie van gemeente polygons op basis van provincie polygon (ZH: 2569 punten)

PRINT 'Query 09';
SET STATISTICS TIME ON
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme, provincies p
  where  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is NULL)
    and  p.id = @prov_zh
    and  gee.polygoon.STIntersects(p.polygoon) = 1
;
SET STATISTICS TIME OFF

-- 10 Selectie van gemeente polygons op basis van lijn dwars door Nederland

PRINT 'Query 10';
SET STATISTICS TIME ON
select count(*)
  from gga_gemeente_efemeriden gee, gga_gemeenten gme
  where  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is NULL)
    and  gee.polygoon.STIntersects(@nl_line) = 1
;
SET STATISTICS TIME OFF

-- 11 Selecteren van @nn_count dichtsbijzijnde lijnen (wegvakken) vanaf een punt

PRINT 'Query 11';
SET STATISTICS TIME ON
select top(@nn_count) wee.wvk_id, stt.naam, wee.lijn.STDistance(@rd_origin) afstand
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is null)
    and  wee.lijn.STDistance(@rd_origin) is not NULL
  order by wee.lijn.STDistance(@rd_origin), naam, wvk_id
;
SET STATISTICS TIME OFF

-- 12 Genereren van buffer rond een aantal gemeenten

PRINT 'Query 12';
SET STATISTICS TIME ON
select gme.naam, gee.polygoon.STBuffer(@buf_dist) gem_buffer into ss_result12
  from gga_gemeenten gme, gga_gemeente_efemeriden gee
  where  gme.naam in
('Amsterdam','Rotterdam','Utrecht','Eindhoven','Groningen','Arnhem')
    and  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is null)
;
select naam, gem_buffer.STArea() buffer_area from ss_result12 order by naam;
SET STATISTICS TIME OFF

-- 13 Met ruimtelijke beperking (groot window waar Nederland geheel in past)

PRINT 'Query 13';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@nl_query_window) = 1
;
SET STATISTICS TIME OFF

-- 14 Met ruimtelijke beperking (polygon met grens van Nederland: 18631 punten)
```

```
PRINT 'Query 14';
SET STATISTICS TIME ON
select count(*)
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  p.id = @nederland
    and  wee.lijn.STIntersects(p.polygoon) = 1
;
SET STATISTICS TIME OFF
--drop table ss_result11;
drop table ss_result12;
PRINT 'Finished';

Query 01

-
948481

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 7586 ms,  elapsed time = 2610 ms.
Query 02

-
42607

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 4542 ms,  elapsed time = 1052 ms.
Query 03

-
524

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 74662 ms,  elapsed time = 9032 ms.
Query 04

-
31835

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 88044 ms,  elapsed time = 4794 ms.
Query 05

-
326461

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 95223 ms,  elapsed time = 4955 ms.
Query 06

-
72901

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 3532845 ms,  elapsed time = 163211 ms.
Query 07

-
4189

(1 rows affected)
```

```
 SQL Server Execution Times:
   CPU time = 59107 ms,  elapsed time = 130384 ms.
Query 08
wvk_id begindatum wegvak_lengte
------ ---------- -------------
459565016 2007-02-01 00:00:00.000 260.01201434742961
459566001 2002-07-01 00:00:00.000 261.1438804829844
459566006 2002-07-01 00:00:00.000 239.45385672381383
459566010 2002-07-01 00:00:00.000 250.9034397929241
459567032 2010-06-01 00:00:00.000 389.41504502649883
460564008 2002-12-01 00:00:00.000 132.09466302617983
460564009 2007-02-01 00:00:00.000 139.88994137535684
460564026 2002-12-01 00:00:00.000 131.85598204101322
460564032 2002-12-01 00:00:00.000 132.03408650799233
460564034 2007-02-01 00:00:00.000 631.49006020520142
460564045 2007-02-01 00:00:00.000 199.51820287315655
460564049 2007-02-01 00:00:00.000 218.83130002385303
460565004 2002-12-01 00:00:00.000 162.2409855221523
460565006 2002-12-01 00:00:00.000 132.06059215375342
460565008 2002-12-01 00:00:00.000 203.12282645449886
460565009 2002-05-01 00:00:00.000 128.72451204024819
460565011 2002-05-01 00:00:00.000 153.93239918566536
460565025 2002-12-01 00:00:00.000 132.06059215375342
460565026 2002-12-01 00:00:00.000 132.45754036671525
460565028 2002-12-01 00:00:00.000 132.06059215375342
460566001 2002-05-01 00:00:00.000 301.47824500617082
460566018 2007-02-01 00:00:00.000 276.38760827824387
460566020 2007-02-01 00:00:00.000 138.28180844524314
460566021 2007-02-01 00:00:00.000 662.18745127521606
460566022 2002-07-01 00:00:00.000 131.39060574432605
460566024 2002-07-01 00:00:00.000 131.68192374203522
460566026 2002-07-01 00:00:00.000 137.20684262883572
460567001 2010-06-01 00:00:00.000 296.37205519022081
460567003 2010-06-01 00:00:00.000 266.47380560376587
460567004 2010-12-01 00:00:00.000 450.93389800818227
460568020 2010-12-01 00:00:00.000 119.22709667219263
460568021 2010-06-01 00:00:00.000 443.60153959439998
461563003 2002-09-01 00:00:00.000 265.53738003743706
461564014 2007-02-01 00:00:00.000 137.20329553767178
461564029 2007-02-01 00:00:00.000 116.55752435602365
461564031 2007-02-01 00:00:00.000 116.86862554592786
461564036 2007-02-01 00:00:00.000 114.79490924249252
461564044 2008-02-01 00:00:00.000 115.64902106376735
461565004 1997-06-14 00:00:00.000 154.97080516665449
461565006 2007-02-01 00:00:00.000 152.13427143460498
461565009 1997-06-14 00:00:00.000 153.08023006426032
461565021 2007-02-01 00:00:00.000 242.55105578344649
461565022 2007-02-01 00:00:00.000 244.42774551570594
461565024 2006-02-01 00:00:00.000 307.52596519744657
461565025 2006-02-01 00:00:00.000 118.25596102094647
461565028 2006-04-01 00:00:00.000 111.15764933191394
461565029 2006-04-01 00:00:00.000 154.00656427573202
461565030 2006-02-01 00:00:00.000 136.03126461955188
461565031 2007-02-01 00:00:00.000 116.84233573505895
461565033 2007-02-01 00:00:00.000 179.14122979022545
461565034 2007-02-01 00:00:00.000 146.0822072681685
461566002 1997-06-14 00:00:00.000 151.52887513606112
461566003 1997-06-14 00:00:00.000 147.70578864756791
461566013 1997-06-14 00:00:00.000 151.53248760373998
461566015 1997-06-14 00:00:00.000 151.53777859439026
461566020 1997-06-14 00:00:00.000 125.22779244241272
461566021 1997-06-14 00:00:00.000 250.86367997805803
461566022 1997-06-14 00:00:00.000 246.67792767088019
461566024 1997-06-14 00:00:00.000 185.86411156578026
461566026 1997-06-14 00:00:00.000 104.69479452198185
461566028 1997-11-27 00:00:00.000 146.45477117526761
461566029 1997-11-27 00:00:00.000 103.74020609066537
461566033 2001-05-01 00:00:00.000 236.11230015522932
461567012 2010-06-01 00:00:00.000 115.68090957698688
461567016 2010-06-01 00:00:00.000 104.84359565351119
461567018 2010-06-01 00:00:00.000 220.49022308418731
461567021 2010-06-01 00:00:00.000 166.47092420990842
461567023 2010-06-01 00:00:00.000 221.71733456163315
461567024 2010-06-01 00:00:00.000 125.69475288479886
461567030 2010-06-01 00:00:00.000 706.42122944797563
```

```
461567033 2010-06-01 00:00:00.000 122.2725564793851
461567034 2010-06-01 00:00:00.000 189.18381048642377
461567035 2010-06-01 00:00:00.000 299.69233239781687
461568002 2010-06-01 00:00:00.000 341.18773495564596
461568018 2005-02-01 00:00:00.000 387.11427817876455
461568019 2005-02-01 00:00:00.000 390.59328291741417
461568022 2010-06-01 00:00:00.000 640.48263154415201
462564002 1997-06-14 00:00:00.000 246.69664537136674
462564016 1997-06-14 00:00:00.000 131.24460134199052
462564019 1997-06-14 00:00:00.000 248.30783727356797
462564023 2007-02-01 00:00:00.000 331.10658740891603
462564024 2002-09-01 00:00:00.000 214.2199722191543
462564026 1997-06-14 00:00:00.000 117.64352935882194
462564033 2007-02-01 00:00:00.000 105.13881692794212
462564058 2007-02-01 00:00:00.000 119.28415344816119
462564075 2007-02-01 00:00:00.000 105.25087918870744
462564083 2007-02-01 00:00:00.000 167.35884309549121
462564084 2007-02-01 00:00:00.000 117.1388911719468
462565001 2002-09-01 00:00:00.000 106.25441167311595
462565004 1997-06-14 00:00:00.000 377.28463799152422
462565005 1997-06-14 00:00:00.000 125.0
462565006 1997-06-14 00:00:00.000 126.17844506887855
462565008 1997-06-14 00:00:00.000 152.73840332767023
462565025 1997-06-14 00:00:00.000 150.84137302156893
462565027 1997-06-14 00:00:00.000 141.6827441857335
462565034 1997-06-14 00:00:00.000 127.14038467016485
462565036 1997-06-14 00:00:00.000 130.5450113945378
462565041 2007-09-01 00:00:00.000 351.72716699168973
462565046 2002-07-01 00:00:00.000 154.33081351434652
462565047 2007-09-01 00:00:00.000 161.1794092259837
462565050 2007-09-01 00:00:00.000 157.45330967929496
462565054 2007-09-01 00:00:00.000 160.28144534076904
462565058 2007-09-01 00:00:00.000 158.37874286665124
462565061 2007-09-01 00:00:00.000 167.34790738397888
462566001 1997-06-14 00:00:00.000 151.82348847311019
462566002 1997-06-14 00:00:00.000 157.19096666157378
462566003 1997-06-14 00:00:00.000 160.73289064072762
462566007 1997-06-14 00:00:00.000 150.58093060888939
462566008 1997-06-14 00:00:00.000 153.68833275902082
462566010 1997-06-14 00:00:00.000 101.8082419494436
462566011 1997-06-14 00:00:00.000 164.12495239907915
462566019 1997-06-14 00:00:00.000 152.15361647808524
462566020 1997-06-14 00:00:00.000 247.07262436538144
462566021 1997-06-14 00:00:00.000 116.71281466844178
462566022 2004-04-01 00:00:00.000 120.44336488956669
462566023 2004-04-01 00:00:00.000 465.58920625832667
462566025 1997-06-14 00:00:00.000 547.95424064067936
462566026 2004-04-01 00:00:00.000 458.90942564516308
462567011 1997-06-14 00:00:00.000 117.95338062132852
462567012 1997-06-14 00:00:00.000 123.96773773849388
462567019 2002-06-01 00:00:00.000 123.97214562983966
462567026 2010-06-01 00:00:00.000 277.15643204613093
462567030 2010-06-01 00:00:00.000 289.51173245849134
462567036 2010-06-01 00:00:00.000 179.02342624594286
462567037 1997-11-11 00:00:00.000 328.80253391058926
462567040 2004-04-01 00:00:00.000 366.39935486407956
462567041 2010-06-01 00:00:00.000 107.66764074056512
462567044 2008-11-01 00:00:00.000 183.17211599733557
462567046 2004-04-01 00:00:00.000 162.83768534608879
462567048 2010-06-01 00:00:00.000 314.19355440935652
462567051 2007-02-01 00:00:00.000 128.95476137736378
462567053 2010-06-01 00:00:00.000 238.92144860853415
462567056 2010-06-01 00:00:00.000 425.62850351606033
462567057 2010-06-01 00:00:00.000 109.54123458058822
462568013 2007-02-01 00:00:00.000 527.11656330305027
462569010 2007-03-01 00:00:00.000 1305.6445483638511
463564002 2007-02-01 00:00:00.000 248.63438381080084
463564003 1997-06-14 00:00:00.000 251.71889884914788
463564005 2007-02-01 00:00:00.000 127.12985487288185
463564008 1997-06-14 00:00:00.000 127.75758294520134
463564016 2008-02-01 00:00:00.000 335.52745799261885
463564017 2008-02-01 00:00:00.000 330.68841870967401
463564019 2008-02-01 00:00:00.000 803.13886088790525
463564022 2008-02-01 00:00:00.000 801.39284928403674
463565004 1997-06-14 00:00:00.000 122.71104269787622
463565005 1997-06-14 00:00:00.000 112.89494246473379
```

```
463565006 1997-06-14 00:00:00.000 240.65241206738071
463565012 1997-06-14 00:00:00.000 104.04326023342406
463565013 1997-06-14 00:00:00.000 122.06555615733703
463565014 1997-06-14 00:00:00.000 122.06555615733703
463565016 1997-06-14 00:00:00.000 288.83626142218492
463565017 1997-06-14 00:00:00.000 221.36345062797457
463565020 1997-06-14 00:00:00.000 128.40009484282902
463565021 1997-06-14 00:00:00.000 124.27791436936813
463565022 1997-06-14 00:00:00.000 125.87615557576433
463565023 1997-06-14 00:00:00.000 127.12985487288185
463565025 1997-06-14 00:00:00.000 121.43310915891102
463565026 1997-06-14 00:00:00.000 115.78995520092192
463565029 2004-04-01 00:00:00.000 502.3817082518708
463565032 2004-04-01 00:00:00.000 498.55925104279265
463565034 2009-08-01 00:00:00.000 757.02104472756264
463566003 1997-06-14 00:00:00.000 117.95338062132852
463566005 1997-06-14 00:00:00.000 121.75795661885921
463566009 1997-06-14 00:00:00.000 124.91997438360288
463566010 1997-11-11 00:00:00.000 114.7946866796083
463566012 2004-03-01 00:00:00.000 117.95353589407043
463566015 1997-06-14 00:00:00.000 120.16655108639841
463566016 1997-06-14 00:00:00.000 122.71104269787622
463566018 1997-06-14 00:00:00.000 117.95338062132852
463566019 1997-06-14 00:00:00.000 116.05602095539895
463566021 1997-06-14 00:00:00.000 122.06555615733703
463566023 1997-06-14 00:00:00.000 122.06555615733703
463566025 1997-06-14 00:00:00.000 126.49110640673517
463566026 1997-06-14 00:00:00.000 125.22779244241272
463566027 1997-06-14 00:00:00.000 118.90332207301863
463566028 1997-06-14 00:00:00.000 121.1191883750877
463566029 1997-06-14 00:00:00.000 117.64352935882194
463566030 1997-06-14 00:00:00.000 117.95338062132852
463566031 1997-06-14 00:00:00.000 118.29740147203719
463566032 1997-06-14 00:00:00.000 115.43396380615195
463566033 1997-06-14 00:00:00.000 124.27791436936813
463566034 1997-06-14 00:00:00.000 176.17095069253918
463566036 1997-06-14 00:00:00.000 118.27087553578015
463566042 2002-06-01 00:00:00.000 134.98688457928733
463567019 1997-06-14 00:00:00.000 123.97132262304241
463567020 1997-06-14 00:00:00.000 124.29023838605403
463567022 1997-06-14 00:00:00.000 124.60476436661851
463567025 1997-06-14 00:00:00.000 118.60555539069249
463567026 1997-06-14 00:00:00.000 117.32433677630571
463567030 1997-06-14 00:00:00.000 124.27791436936813
463567031 1997-06-14 00:00:00.000 126.17844506887855
463567033 1997-06-14 00:00:00.000 120.17621590186533
463567034 1997-06-14 00:00:00.000 116.69190203266035
463567042 1997-06-14 00:00:00.000 147.05441169852742
463567044 1997-06-14 00:00:00.000 149.58085266251646
463567047 1997-06-14 00:00:00.000 138.19189556555045
463567048 1997-06-14 00:00:00.000 138.51353724455961
463567049 1997-06-14 00:00:00.000 129.40131679468701
463567050 2001-06-01 00:00:00.000 299.64520305767439
463567052 2000-01-01 00:00:00.000 255.94607355174458
463567053 2001-06-01 00:00:00.000 300.90227765271408
463567055 2000-01-01 00:00:00.000 252.5365089101162
463567057 1997-06-14 00:00:00.000 338.14677166245764
463567058 2010-06-01 00:00:00.000 561.42323822950402
463567061 1997-06-14 00:00:00.000 106.57613530838952
463567064 1997-06-14 00:00:00.000 117.64352935882194
463567068 2004-04-01 00:00:00.000 561.41826099433433
463567069 2007-02-01 00:00:00.000 292.64413506779641
463568034 1998-04-15 00:00:00.000 123.16655390161731
464564075 2009-08-01 00:00:00.000 776.62307283524467

(210 rows affected)

 SQL Server Execution Times:
   CPU time = 138813 ms,  elapsed time = 6791 ms.
Query 09

-
94

(1 rows affected)
```

```
 SQL Server Execution Times:
   CPU time = 390 ms,  elapsed time = 1580 ms.
Query 10

-
62

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 140 ms,  elapsed time = 214 ms.
Query 11
wvk_id naam afstand
------ ---- -------
310326069 Krankeledenstraat 12.14107327947711
309326070 Breestraat 25.0
310326003 Lieve Vrouwekerkhof 25.0
310325057 Krankeledenstraat 32.015621187164243
310325020 Lieve Vrouwekerkhof 32.015621187164243
309326018 Westsingel 36.380020071675638
310326027 Lieve Vrouwestraat 69.500272332394644
310326026 Lieve Vrouwestraat 70.34912934784623
310326028 Lieve Vrouwestraat 79.752746736369986
310326004 Zwanenhalssteeg 79.752746736369986
309326088 Hellestraat 87.182058275234837
309325137 Varkensmarkt 87.885195044445084
310325027 Varkensmarkt 87.885195044445084
309326104 Hellestraat 90.578494406786504
309325135 Sint Jorisstraat 90.578494406786504
309325138 Utrechtsestraat 95.108072475493998
309325136 Varkensmarkt 95.108072475493998
310325051 Langestraat 96.797973407349687
310325055 Langestraat 97.077524115359765
310325022 Varkensmarkt 102.59142264341595
310325056 Langestraat 106.11018692850723
310325038 Scherbierstraat 106.11018692850723
309325110 Utrechtsestraat 111.00130152841513
310325083 Varkensmarkt 113.74813746608973
310325021 Zuidsingel 113.74813746608973
309326069 Bollebruggang 117.9594222849846
309326105 Breestraat 118.29231769225188
310326181 Paternosterstraat 118.29231769225188
310326060 Langegracht 118.91896247866168
310326156 Langegracht 118.91896247866168
309326089 Molenstraat 123.38622196179519
309326023 Westsingel 123.38622196179519
309326149 Hellestraat 124.25463201426064
309326160 Torenstraat 124.25463201426064
310326157 Langegracht 129.49517365523704
310326155 Marktgang 129.49517365523704
310326063 Lieve Vrouwestraat 136.51598652537584
309326087 Molenstraat 145.45446022724775
309325120 Arnhemsestraat 147.76332427229701
310325061 Koestraat 147.76332427229701
310326191 Langegracht 152.18433025774991
310326162 Krommestraat 153.60339186357274
309326086 Molenstraat 155.20631430454111
309326009 Stadhuisplein 155.20631430454111
310325050 Mooierstraat 156.38128936992797
310325039 Scherbierstraat 156.38128936992797
310326072 Kortegracht 156.64470115520422
310326054 Langestraat 156.64470115520422
310326039 Achter het Oude Stadhuis 158.30109639546666
310326158 Krommestraat 158.30109639546666
310326159 Krommestraat 158.96071263680309
309325133 Riddergang 160.69743148546561
309325134 Sint Jorisplein 162.26008708616311
310326161 Krommestraat 166.47529820666207
310326160 Peperstraat 166.47529820666207
309325111 Utrechtsestraat 167.72451737894286
309325114 Sint Jorisstraat 167.73307710764087
309326152 Molenstraat 168.43396332094071
310325108 Koesteeg 172.02906731131225
310325123 Koestraat 172.02906731131225
309326107 Breestraat 175.45304336205376
309326182 Kromme Elleboogsteeg 175.45304336205376
```

```
310326075 Achter de Heilige Geest 175.84981601924542
310326071 Kortegracht 175.84981601924542
309325123 Ad Arnhemse Poortwal 177.583951009066
310325046 Muurhuizen 178.94964919219112
310325122 Rozemarijnsteeg 178.94964919219112
310326055 Langestraat 179.95624482080009
310326076 Achter de Heilige Geest 181.45893347256126
310326194 Kortegracht 181.45893347256126
310325107 Haagplein 181.55791835388874
310325116 Koesteeg 181.55791835388874
310326190 Kerkgang 184.14741817359379
309326169 Langegracht 184.14741817359379
310326164 Hof 187.16928505497563
310325049 Mooierstraat 187.24008230076771
310325121 Schelvissteegje 187.24008230076771
309325122 Ad Arnhemse Poortwal 193.48761960654116
309325119 Slijkpoortsteeg 193.48761960654116
310326085 Hof 197.38140305509299
310326025 Vijver 197.38140305509299
310326056 Langestraat 198.50809346723361
310326030 Valkestraat 198.50809346723361
310326163 Hof 200.60210346106746
310326154 Krommestraat 201.75531537482928
310325058 Kortegracht 205.17207303139134
309325124 Ad Arnhemse Poortwal 211.18390966171933
310325115 Haagplein 212.12303236093175
309325121 Arnhemsestraat 213.57414542494473
309325139 Koesteeg 213.57414542494473
310326193 Valkestraat 214.72663375789415
309325009 Ad Arnhemse Poortwal 219.03516370206268
310325114 Haagplein 221.66884690909447
309326181 Kromme Elleboogsteeg 221.77759583195109
310325117 Koesteeg 222.67484727735135
310325124 Koesteeg 222.67484727735135
309325018 Stadsring 225.47599817719083
310326179 Hof 226.49559820227412
310326178 Hof 227.29738406986348
310326177 Windsteeg 227.29738406986348

(100 rows affected)

 SQL Server Execution Times:
   CPU time = 177513 ms,  elapsed time = 110114 ms.
Query 12

 SQL Server Execution Times:
   CPU time = 2091 ms,  elapsed time = 2198 ms.

(6 rows affected)
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 8 ms.
naam buffer_area
---- -----------
Amsterdam 267562682.17025757
Arnhem 131283968.05675507
Eindhoven 112717718.56883621
Groningen 106404439.88105011
Rotterdam 393649548.26155281
Utrecht 128202472.91809082

(6 rows affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 8 ms.
Query 13

-
948481

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 108997 ms,  elapsed time = 6123 ms.
Query 14

-
```

```
948481

(1 rows affected)

 SQL Server Execution Times:
   CPU time = 11563506 ms,  elapsed time = 545386 ms.
Finished
```

B3.28 SQL Server script met ‘create’ queries

```
-- create_ss_queries.sql 14-09-2012  TT

DECLARE @peildatum datetime = '2011-07-30T12:00:00';

DECLARE    @nederland numeric(10) = 0;
DECLARE      @prov_zh numeric(10) = 8;
DECLARE @prov_limburg numeric(10) = 11;
DECLARE    @gem_delft varchar(10) = 'Delft';

-- Small window: 2 x 2 km in Gemeente Groningen
DECLARE @small_query_window geometry = geometry::STLineFromText('LINESTRING (230000
582000,232000 584000)',28992).STEnvelope();

-- Medium window: 50 x 50 km, groot deel van Zeeland
DECLARE @medium_query_window geometry = geometry::STLineFromText('LINESTRING (20000
360000,70000 410000)',28992).STEnvelope();

-- Large window: 120 x 120 km in midden van Nederland
DECLARE @large_query_window geometry = geometry::STLineFromText('LINESTRING (110000
430000,230000 550000)',28992).STEnvelope();

-- ------------------------------------------------------------------------
drop table ss_result07;

-- 01 Basis, administratieve, query

PRINT 'Query 01';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result01
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
;
SET STATISTICS TIME OFF

-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')

PRINT 'Query 02';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result02
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
SET STATISTICS TIME OFF

-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen

PRINT 'Query 03';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result03
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@small_query_window) = 1
;
SET STATISTICS TIME OFF

-- 04 Medium window: 50 x 50 km, groot deel van Zeeland

PRINT 'Query 04';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result04
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
```

```
    and  wee.lijn.STIntersects(@medium_query_window) = 1
;
SET STATISTICS TIME OFF

-- 05 Large window: 120 x 120 km in midden van Nederland

PRINT 'Query 05';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result05
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@large_query_window) = 1
;
SET STATISTICS TIME OFF

-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)

PRINT 'Query 06';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result06
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  p.id = @prov_limburg
    and  wee.lijn.STIntersects(p.polygoon) = 1
;
SET STATISTICS TIME OFF

-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)

PRINT 'Query 07';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result07
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is null)
    and  gme.naam = @gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is null)
    and  wee.lijn.STIntersects(gee.polygoon) = 1
;
SET STATISTICS TIME OFF
drop table ss_result01;
drop table ss_result02;
drop table ss_result03;
drop table ss_result04;
drop table ss_result05;
drop table ss_result06;
drop table ss_result07;
PRINT 'Finished';
GO
```

B3.29 SQL Server log voor ‘create’ queries

```
-- create_ss_queries.sql 14-09-2012  TT

DECLARE @peildatum datetime = '2011-07-30T12:00:00';

DECLARE    @nederland numeric(10) = 0;
DECLARE      @prov_zh numeric(10) = 8;
DECLARE @prov_limburg numeric(10) = 11;
DECLARE    @gem_delft varchar(10) = 'Delft';

-- Small window: 2 x 2 km in Gemeente Groningen
DECLARE @small_query_window geometry = geometry::STLineFromText('LINESTRING (230000
582000,232000 584000)',28992).STEnvelope();

-- Medium window: 50 x 50 km, groot deel van Zeeland
DECLARE @medium_query_window geometry = geometry::STLineFromText('LINESTRING (20000
360000,70000 410000)',28992).STEnvelope();

-- Large window: 120 x 120 km in midden van Nederland
DECLARE @large_query_window geometry = geometry::STLineFromText('LINESTRING (110000
430000,230000 550000)',28992).STEnvelope();

-- ------------------------------------------------------------------------

-- 01 Basis, administratieve, query

PRINT 'Query 01';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result01
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
;
SET STATISTICS TIME OFF

-- 02 Met beperking op wegbeheerdersoort, bv and wee.wegbeheerdersoort in ('R','P')

PRINT 'Query 02';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result02
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.wegbeheerdersoort in ('R','P')
;
SET STATISTICS TIME OFF

-- En/of met ruimtelijke beperking (rechthoekig window, in verchillende groottes)
--
-- 03 Small window: 2 x 2 km in Gemeente Groningen

PRINT 'Query 03';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result03
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@small_query_window) = 1
;
SET STATISTICS TIME OFF

-- 04 Medium window: 50 x 50 km, groot deel van Zeeland

PRINT 'Query 04';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result04
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@medium_query_window) = 1
```

```
;
SET STATISTICS TIME OFF

-- 05 Large window: 120 x 120 km in midden van Nederland

PRINT 'Query 05';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result05
  from nww_wegvak_efemeriden wee, gga_straten stt
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  wee.lijn.STIntersects(@large_query_window) = 1
;
SET STATISTICS TIME OFF

-- 06 Met ruimtelijke beperking (polygon met grens van provincie Limburg: 7946 punten)

PRINT 'Query 06';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result06
  from nww_wegvak_efemeriden wee, gga_straten stt, provincies p
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is NULL)
    and  p.id = @prov_limburg
    and  wee.lijn.STIntersects(p.polygoon) = 1
;
SET STATISTICS TIME OFF

-- 07 Met ruimtelijke beperking (polygon met grens van gemeente Delft: 252 punten)

PRINT 'Query 07';
SET STATISTICS TIME ON
select wee.wvk_id, wee.lijn, stt.id into ss_result07
  from nww_wegvak_efemeriden wee, gga_straten stt, gga_gemeenten gme,
gga_gemeente_efemeriden gee
  where  wee.stt_id = stt.id
    and  wee.begindatum <= @peildatum
    and (wee.einddatum  >= @peildatum or wee.einddatum is null)
    and  gme.naam = @gem_delft
    and  gee.gme_id = gme.id
    and  gee.begindatum <= @peildatum
    and (gee.einddatum  >= @peildatum or gee.einddatum is null)
    and  wee.lijn.STIntersects(gee.polygoon) = 1
;
SET STATISTICS TIME OFF

drop table ss_result01;
drop table ss_result02;
drop table ss_result03;
drop table ss_result04;
drop table ss_result05;
drop table ss_result06;
drop table ss_result07;
PRINT 'Finished';

Query 01

 SQL Server Execution Times:
   CPU time = 6193 ms,  elapsed time = 10567 ms.

(948481 rows affected)
Query 02

 SQL Server Execution Times:
   CPU time = 9449 ms,  elapsed time = 2137 ms.

(42607 rows affected)
Query 03

 SQL Server Execution Times:
   CPU time = 90168 ms,  elapsed time = 5022 ms.

(524 rows affected)
Query 04
```

```
 SQL Server Execution Times:
   CPU time = 91556 ms,  elapsed time = 5022 ms.

(31835 rows affected)
Query 05

 SQL Server Execution Times:
   CPU time = 124349 ms,  elapsed time = 10935 ms.

(326461 rows affected)
Query 06

 SQL Server Execution Times:
   CPU time = 2965937 ms,  elapsed time = 158320 ms.

(72901 rows affected)
Query 07

 SQL Server Execution Times:
   CPU time = 51902 ms,  elapsed time = 140334 ms.

(4189 rows affected)
Finished
```

Reports published before in this series

1. GISt Report No. 1, Oosterom, P.J. van, Research issues in integrated querying of geometric and thematic cadastral information (1), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
2. GISt Report No. 2, Stoter, J.E., Considerations for a 3D Cadastre, Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 30.p.
3. GISt Report No. 3, Fendel, E.M. en A.B. Smits (eds.), Java GIS Seminar, Opening GDMC, Delft 15 November 2000, Delft University of Technology, GISt. No. 3, 25 p.p.
4. GISt Report No. 4, Oosterom, P.J.M. van, Research issues in integrated querying of geometric and thematic cadastral information (2), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
5. GISt Report No. 5, Oosterom, P.J.M. van, C.W. Quak, J.E. Stoter, T.P.M. Tijssen en M.E. de Vries, Objectgerichtheid TOP10vector: Achtergrond en commentaar op de gebruikersspecificaties en het conceptuele gegevensmodel, Rapport aan Topografische Dienst Nederland, E.M. Fendel (eds.), Delft University of Technology, Delft 2000, 18 p.p.
6. GISt Report No. 6, Quak, C.W., An implementation of a classification algorithm for houses, Rapport aan Concernstaf Kadaster, Delft 2001, 13.p.
7. GISt Report No. 7, Tijssen, T.P.M., C.W. Quak and P.J.M. van Oosterom, Spatial DBMS testing with data from the Cadastre and TNO NITG, Delft 2001, 119 p.
8. GISt Report No. 8, Vries, M.E. de en E. Verbree, Internet GIS met ArcIMS, Delft 2001, 38 p.
9. GISt Report No. 9, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2001, 132 p.
10. GISt Report No. 10, Stoter, J.E., Nauwkeurig bepalen van grondverzet op basis van CAD ontgravingsprofielen en GIS, een haalbaarheidsstudie, Rapport aan de Bouwdienst van Rijkswaterstaat, Delft 2001, 23 p.
11. GISt Report No. 11, Geo DBMS, De basis van GIS-toepassingen, KvAG/AGGN Themamiddag, 14 november 2001, J. Flim (eds.), Delft 2001, 37 p.
12. GISt Report No. 12, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The second GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2002, Part 1, Main text, 63 p. and Part 2, Appendices B and C, 85 p.
13. GISt Report No. 13, Vries, M.E. de, T.P.M. Tijssen en P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle spatial and in Oracle/ArcSDE compressed binary format, Delft 2002, .72 p. (Confidential)
14. GISt Report No. 14, Stoter, J.E., 3D Cadastre, Progress Report, Report to Concernstaf Kadaster, Delft 2002, 16 p.

15. GISt Report No. 15, Zlatanova, S., Research Project on the Usability of Oracle Spatial within the RWS Organisation, Detailed Project Plan (MD-NR. 3215), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2002, 13 p.
16. GISt Report No. 16, Verbree, E., Driedimensionale Topografische Terreinmodellering op basis van Tetraëder Netwerken: Top10-3D, Report aan Topografische Dienst Nederland, Delft 2002, 15 p.
17. GISt Report No. 17, Zlatanova, S. Augmented Reality Technology, Report to SURFnet bv, Delft 2002, 72 p.
18. GISt Report No. 18, Vries, M.E. de, Ontsluiting van Geo-informatie via netwerken, Plan van aanpak, Delft 2002, 17p.
19. GISt Report No. 19, Tijssen, T.P.M., Testing Informix DBMS with spatial data from the cadastre, Delft 2002, 62 p.
20. GISt Report No. 20, Oosterom, P.J.M. van, Vision for the next decade of GIS technology, A research agenda for the TU Delft the Netherlands, Delft 2003, 55 p.
21. GISt Report No. 21, Zlatanova, S., T.P.M. Tijssen, P.J.M. van Oosterom and C.W. Quak, Research on usability of Oracle Spatial within the RWS organisation, (AGI-GAG-2003-21), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2003, 74 p.
22. GISt Report No. 22, Verbree, E., Kartografische hoogtevoorstelling TOP10vector, Report aan Topografische Dienst Nederland, Delft 2003, 28 p.
23. GISt Report No. 23, Tijssen, T.P.M., M.E. de Vries and P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle SDO_Geometry version 9i and version 10g Beta 2 (in the context of ArcGIS 8.3), Delft 2003, 20 p. (Confidential)
24. GISt Report No. 24, Stoter, J.E., 3D aspects of property transactions: Comparison of registration of 3D properties in the Netherlands and Denmark, Report on the short-term scientific mission in the CIST – G9 framework at the Department of Development and Planning, Center of 3D geo-information, Aalborg, Denmark, Delft 2003, 22 p.
25. GISt Report No. 25, Verbree, E., Comparison Gridding with ArcGIS 8.2 versus CPS/3, Report to Shell International Exploration and Production B.V., Delft 2004, 14 p. (confidential).
26. GISt Report No. 26, Penninga, F., Oracle 10g Topology, Testing Oracle 10g Topology with cadastral data, Delft 2004, 48 p.
27. GISt Report No. 27, Penninga, F., 3D Topography, Realization of a three dimensional topographic terrain representation in a feature-based integrated TIN/TEN model, Delft 2004, 27 p.
28. GISt Report No. 28, Penninga, F., Kartografische hoogtevoorstelling binnen TOP10NL, Inventarisatie mogelijkheden op basis van TOP10NL uitgebreid met een Digitaal Hoogtemodel, Delft 2004, 29 p.
29. GISt Report No. 29, Verbree, E. en S.Zlatanova, 3D-Modeling with respect to boundary representations within geo-DBMS, Delft 2004, 30 p.
30. GISt Report No. 30, Penninga, F., Introductie van de 3e dimensie in de TOP10NL; Voorstel voor een onderzoekstraject naar het stapsgewijs introduceren van 3D data in de TOP10NL, Delft 2005, 25 p.
31. GISt Report No. 31, P. van Asperen, M. Grothe, S. Zlatanova, M. de Vries, T. Tijssen, P. van Oosterom and A. Kabamba, Specificatie datamodel Beheerkaart Nat, RWS-AGI report/GIST Report, Delft, 2005, 130 p.
32. GISt Report No. 32, E.M. Fendel, Looking back at Gi4DM, Delft 2005, 22 p.

33. GISt Report No. 33, P. van Oosterom, T. Tijssen and F. Penninga, Topology Storage and the Use in the context of consistent data management, Delft 2005, 35 p.
34. GISt Report No. 34, E. Verbree en F. Penninga, RGI 3D Topo - DP 1-1, Inventarisatie huidige toegankelijkheid, gebruik en mogelijke toepassingen 3D topografische informatie en systemen, 3D Topo Report No. RGI-011-01/GISt Report No. 34, Delft 2005, 29 p.
35. GISt Report No. 35, E. Verbree, F. Penninga en S. Zlatanova, Datamodellering en datastructurering voor 3D topografie, 3D Topo Report No. RGI-011-02/GISt Report No. 35, Delft 2005, 44 p.
36. GISt Report No. 36, W. Looijen, M. Uitentuis en P. Bange, RGI-026: LBS-24-7, Tussenrapportage DP-1: Gebruikerswensen LBS onder redactie van E. Verbree en E. Fendel, RGI LBS-026-01/GISt Rapport No. 36, Delft 2005, 21 p.
37. GISt Report No. 37, C. van Strien, W. Looijen, P. Bange, A. Wilcsinszky, J. Steenbruggen en E. Verbree, RGI-026: LBS-24-7, Tussenrapportage DP-2: Inventarisatie geo-informatie en -services onder redactie van E. Verbree en E. Fendel, RGI LBS-026-02/GISt Rapport No. 37, Delft 2005, 21 p.
38. GISt Report No. 38, E. Verbree, S. Zlatanova en E. Wisse, RGI-026: LBS-24-7, Tussenrapportage DP-3: Specifieke wensen en eisen op het gebied van plaatsbepaling, privacy en beeldvorming, onder redactie van E. Verbree en E. Fendel, RGI LBS-026-03/GISt Rapport No. 38, Delft 2005, 15 p.
39. GISt Report No. 39, E. Verbree, E. Fendel, M. Uitentuis, P. Bange, W. Looijen, C. van Strien, E. Wisse en A. Wilcsinszky en E. Verbree, RGI-026: LBS-24-7, Eindrapportage DP-4: Workshop 28-07-2005 Geo-informatie voor politie, brandweer en hulpverlening ter plaatse, RGI LBS-026-04/GISt Rapport No. 39, Delft 2005, 18 p.
40. GISt Report No. 40, P.J.M. van Oosterom, F. Penninga and M.E. de Vries, Trendrapport GIS, GISt Report No. 40 / RWS Report AGI-2005-GAB-01, Delft, 2005, 48 p.
41. GISt Report No. 41, R. Thompson, Proof of Assertions in the Investigation of the Regular Polytope, GISt Report No. 41 / NRM-ISS090, Delft, 2005, 44 p.
42. GISt Report No. 42, F. Penninga and P. van Oosterom, Kabel- en leidingnetwerken in de kadastrale registratie (in Dutch) GISt Report No. 42, Delft, 2006, 38 p.
43. GISt Report No. 43, F. Penninga and P.J.M. van Oosterom, Editing Features in a TEN-based DBMS approach for 3D Topographic Data Modelling, Technical Report, Delft, 2006, 21 p.
44. GISt Report No. 44, M.E. de Vries, Open source clients voor UMN MapServer: PHP/Mapscript, JavaScript, Flash of Google (in Dutch), Delft, 2007, 13 p.
45. GISt Report No. 45, W. Tegtmeier, Harmonization of geo-information related to the lifecycle of civil engineering objects – with focus on uncertainty and quality of surveyed data and derived real world representations, Delft, 2007, 40 p.
46. GISt Report No. 46, W. Xu, Geo-information and formal semantics for disaster management, Delft, 2007, 31 p.
47. GISt Report No. 47, E. Verbree and E.M. Fendel, GIS technology – Trend Report, Delft, 2007, 30 p.
48. GISt Report No. 48, B.M. Meijers, Variable-Scale Geo-Information, Delft, 2008, 30 p.

49. GISt Report No. 48, Maja Bitenc, Kajsa Dahlberg, Fatih Doner, Bas van Goort, Kai Lin,Yi Yin, Xiaoyu Yuan and Sisi Zlatanova, Utilty Registration, Delft, 2008, 35 p.
50. GISt Report No 50, T.P.M. Tijssen en S. Zlatanova, Oracle Spatial 11g en ArcGIS 9.2 voor het beheer van puntenwolken (Confidential), Delft, 2008, 16 p.
51. GISt Report No. 51, S. Zlatanova, Geo-information for Crisis Management, Delft, 2008, 24 p.
52. GISt Report No. 52, P.J.M. van Oosterom, INSPIRE activiteiten in het jaar 2008 (partly in Dutch), Delft, 2009, 142 p.
53. GISt Report No. 53, P.J.M. van Oosterom with input of and feedback by Rod Thompson and Steve Huch (Department of Environment and Resource Management, Queensland Government), Delft, 2010, 60 p.
54. GISt Report No. 54, A. Dilo and S. Zlatanova, Data modeling for emergency response, Delft, 2010, 74 p.
55. GISt Report No. 55, Liu Liu, 3D indoor " door-to-door" navigation approach to support first responders in emergency response – PhD Research Proposal, Delft, 2011, 47 p.
56. GISt Report No. 56, Md. Nazmul Alam, Shadow effect on 3D City Modelling for Photovoltaic Cells – PhD Proposal, Delft, 2011, 39 p.
57. GIST Report No. 57, G.A.K. Arroyo Ohori, Realising the Foundations of a Higher Dimensional GIS: A Study of Higher Dimensional Data Models, Data Structures and Operations – PhD Research Proposal, Delft, 2011, 68 p.
58. GISt Report No. 58, Zhiyong Wang, Integrating Spatio-Temporal Data into Agent-Based Simulation for Emergency Navigation Support – PhD Research Proposal, Delft, 2012, 49 p.


ZUIDPOLDER
VAN DELFGAUW
MOLENSLOOT
POLDER