



Topology structures: de volgende stap in spatial information management

Door ir. Friso Penninga

Oracle10g Topology getest

Oracle Spatial mag zich verheugen in een groeiende belangstelling. Steeds meer gebruikers kiezen voor het beheer van hun ruimtelijke gegevens in de database, zeker sinds met het beschikbaar komen van ruimtelijke indexeringsmethoden een goede query performance gegarandeerd is.

Tot nu toe was het echter alleen mogelijk om ruimtelijke data in geometrische vorm op te slaan. Hierin schuilt echter een gevaar: stel bijvoorbeeld dat twee aangrenzende percelen beide als polygoon zijn opgeslagen. De geometrie van de gezamenlijke grens in nu dubbel opgeslagen, zowel in de polygoon van perceel A als in die van perceel B. In deze redundante dataopslag is het risico van het ontstaan van data inconsistenties zeer groot, bijvoorbeeld als de gezamenlijke grens alleen in polygoon A wordt gewijzigd en niet ook in polygoon B.

Oplossing

De oplossing ligt in het gebruik van topologie: sla de grens slechts eenmaal op en voeg de informatie toe welke percelen links en rechts liggen. Een topologische datastructuur richt zich op het opslaan van de onderlinge relaties van ruimtelijke objecten, waarmee het mogelijk wordt om databestanden te valideren en zo consistent te houden.

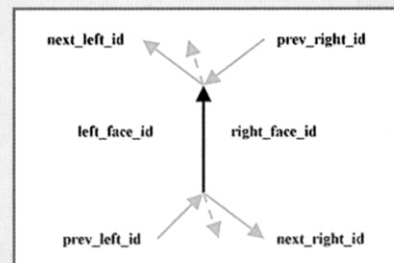
Naast de hierboven geschetste structuur (de zogenaamde 'vlakkenpartitie', waarbij de ruimte geheel wordt opgedeeld in niet overlappende vlakken) is er een tweede topologische structuur die nu door Oracle wordt ondersteund: het lineaire netwerk (denk aan een model voor wegen of leidingen netwerk). Naast het managen (consistent houden) van deze structuur in het DBMS heeft topologische structuur een ander belangrijk voordeel: bepaalde vragen of analyses kunnen nu efficiënt in het DBMS uit-

gevoerd worden. Denk hierbij aan vragen als 'Wat is de snelste route van knooppunt A naar B' of 'Geef me de buurpercelen van perceel X'. Met de huidige productierelease van Oracle 10g is topologisch databaseer nu voor het eerst zonder aanvullende software mogelijk in een database omgeving. Reden genoeg voor een aantal verkennende tests.

Achtergronden topologie

De basiscomponenten in een tweedimensionale topologie zijn de nodes, edges en faces (we beperken ons hier even tot de zogenaamde vlakkenpartitie). In het meest eenvoudige datamodel wordt een face begrensd door een aantal edges en een edge wordt opgespannen door twee nodes. Door deze verwijzingen vast te leggen en alleen van de nodes de geometrie (coördinaten) op te slaan, is het mogelijk om ook de geometrie van de edges en daarmee weer de faces te reconstrueren, zonder dat geometrische data redundant wordt opgeslagen.

In de praktijk wordt er vaak voor gekozen om de geometrie van de edges op te slaan, om zo te kunnen werken met edges die opgebouwd zijn uit verschillende lijnstukken. Nodes worden nu alleen gebruikt op plaatsen waar meerdere edges bij elkaar komen. Een veelgebruikte edge-gerichte datastructuur is de winged edge structuur (zie figuur 1), die naast verwijzingen naar de faces links en rechts ook vier verwijzingen bevat naar andere edges, namelijk de edges die voor en na de edge liggen en deel uitmaken van de linker- en rechterface. Merk op dat edges gericht zijn (beginnode en eindnode) om zo de begrippen links en rechts betekenis te geven. Als nu van bijvoorbeeld een face één edge gegeven is, dan kan door steeds de volgende edge te pakken de hele buitengrens afgelopen worden om zo de geometrie van deze grens te reconstrueren.



Figuur 1. De Winged Edge datastructuur zoals Oracle die gebruikt

In een topologische datastructuur kunnen objecten ook op basis van hun onderlinge relaties bevraagd worden, wat met name voor queries als 'geef me alle percelen die grenzen aan dit perceel' zou moeten zorgen voor een betere queryperformance in vergelijking met de geometrische variant van deze query. Doel van het testen van Oracle10g Topology was om de geboden functionaliteit te verkennen en te bekijken of met name de neighbourqueries sneller kunnen worden uitgevoerd in een topologische datastructuur dan op een geometrische datastructuur. Hiervoor is gebruik gemaakt van data van het Kadaster, omdat het Kadaster de perceelsgrenzen nu al opslaat in een winged edge datastructuur en er daarmee grote datasets compleet met topologische verwijzingen beschikbaar waren.

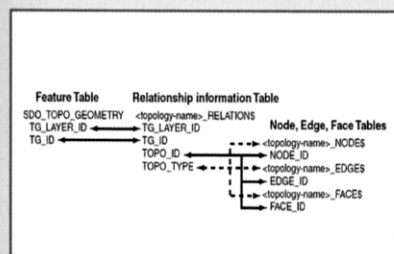
Oracle Topology datamodel

De topologische elementen nodes, edges en faces zijn de bouwstenen waarmee elk object uit de werkelijkheid kan worden opgebouwd. Deze objecten worden meestal als features aangeduid (ook in dit artikel), maar Oracle hanteert de term topology geometries. Elke feature wordt opgeslagen als een verzameling van topologische elementen, zo kan het object perceel in theorie bijvoorbeeld opgebouwd zijn uit meerdere faces (exclaves).

Om vast te kunnen leggen om welk type feature het gaat, wordt voor elke feature een referentie vastgelegd naar

zijn topology geometry (feature) layer. Zo'n layer is een verzameling features van een bepaald type (zelfde soort of klasse), bijvoorbeeld layers 'bomen', 'percelen' of 'huizen' (hier bevinden zich ook de thematische attributen van het feature). In de layer-definitie wordt ook vastgelegd of de objecten in die layer uit punten, lijnen, vlakken of een combinatie daarvan worden samengesteld. Het uiteindelijke datamodel is weergegeven in figuur 2.

Alhoewel verder niet onderzocht in de tests is het ook mogelijk om de geschiedenis van alle update bij te houden in de history tabel.



Figuur 2: Oracle Topology datamodel

Laden data

De procedure voor het aanmaken en vullen van een topologische datastructuur bestaat uit zes stappen:

- Aanmaken lege topologie. De gebruiker geeft een naam voor de topologie op en lege tabellen voor nodes, edges, faces en de history worden gecreëerd.
- Laden data in node, edge en face tabellen.
- Aanmaken feature tabellen. Voor elk type objecten wordt een aparte tabel gecreëerd.
- Koppelen feature tabellen aan topologie. Door het definiëren van feature layers worden de relaties tussen de feature tabellen enerzijds en de nodes, edges en faces anderzijds opgeslagen in de relationship tabel.
- Initialiseren metadata en creëren indexen. De create metadata procedure creëert indexen op de node, edge, face, history en relationship tabellen.
- Definiëren features m.b.v. nodes, edges en faces. Voor elke feature wordt vastgelegd uit welke nodes, edges en faces deze objecten zijn samengesteld. Bovendien worden

in deze stap ook de eventuele thematische attributen geladen.

Het aanmaken en vullen van de topologische datastructuur is het enige moment waarop de gebruiker direct op de node, edge en face tabellen werkt. Alle edit-operaties horen uitgevoerd te worden op de features of door middel van de speciale update commando's. Hiermee wordt een onderscheid gemaakt dat doet denken aan een front- en een backoffice. De gebruiker ziet alleen zijn features en een aantal commando's, terwijl achter de schermen dit vertaald wordt in bewerkingen op nodes, edges, faces en hun onderlinge relaties.

De edit-operaties worden uitgevoerd in een Java programma op een topo_map object, d.w.z. een in de cache ingeladen (deel van de) topologie. Alvorens de wijzigingen daadwerkelijk worden uitgevoerd, wordt deze topo_map eerst gevalideerd om zo de valide toestand van de topologie te waarborgen. Dit ontwerp is erg robuust en volstaat voor de gemiddelde gebruiker waarschijnlijk prima, maar expert users zullen vaker direct op de node, edge en face tabellen willen kunnen werken met hun zelf ontwikkelde toepassingen.

Een duidelijk nadeel van Oracle Topology dat verband houdt met de laadprocedure is de wijze van validatie. De huidige validatieprocedure is zodanig geprogrammeerd dat deze stopt bij het aantreffen van de eerste topologische fout. Dit maakt de procedure zeer robuust, maar maakt het, mede gezien de snelheid van het validatieproces, vrijwel onmogelijk om een dataset volledig te corrigeren. Daarvoor zou juist een complete lijst van topologische fouten gewenst zijn, ook al zullen hierin fouten voorkomen die mede veroorzaakt worden door een eerdere fout. Nu moet de data eigenlijk al valide zijn voordat men deze in de topologie gaat inladen.

Functionaliteit en performance

Er is getest met kadastrale bestanden van verschillende omvang. Eerst is getest met een zeer kleine kadastrale gemeente, waarvan een detail van de topologie is weergegeven in figuur 3. Later is ook getest met grotere be-

standen, waarvan de grootste topologie (kadastrale data van Zuid-Holland) bestond uit 1.001.134 faces, 3.363.761 edges en 2.401.586 nodes. Hoewel het Kadaster de grootste aandacht besteedt aan het geometrisch en topologisch correct houden van haar data, blijken er toch nog enkele fouten



Figuur 3. Detail van een kadastrale topologie

in de winged-edge verwijzingen van de ruim 3 miljoen edges te zitten. Opgemerkt moet worden dat de 6 verwijzingen per edge in de winged-edge structuur op dit moment niet naar de afnemers van het Kadaster worden geleverd, maar puur zijn bedoeld voor intern gebruik. Doordat Oracle per validatiepoging slechts één fout retourneert, is het ondoenlijk om de datasets te valideren. Hierdoor zijn de tests voor grote datasets met niet-valide topologieën uitgevoerd.

Wie verwacht dat een topologische opslag door gebrek aan redundante opslag minder geheugenruimte vereist dan een op polygonen gebaseerd model, komt bedrogen uit. De tabellen (nodes, edges, faces, relations, history en feature tabel) van de Zuid-Holland topologie zijn samen 1.080 MB groot. Hierbij komt nog de ruimte voor de vele indexen die nodig zijn om ondanks alle verwijzingen toch een acceptabele performance te waarborgen. De 12 indexen nemen samen nog eens 1191 MB in, wat het totaal op 2.271 MB brengt.

Ter vergelijking: wanneer dezelfde percelen als polygonen worden opgeslagen kost dit slechts 311 MB. De spatial index neemt nog eens 110 MB in gebruik, wat het totaal op 421 MB brengt; ruim vijf keer minder dan de topologische variant. Hierbij moet

Topology structures: de volgende stap in spatial information management

worden opgemerkt dat deze groot-schalige data set het 'worst-case' scenario is voor een topologische structuur, daar de meeste edges slechts heel weinig punten bevatten en er dus weinig wordt uitgespaard door deze geometrie maar één keer op te slaan. Daartegenover staat dat het aantal wijzigingen per edge vast staat (en onafhankelijk is van het aantal tussenpunten). Voor midden- en kleinschalige data sets zullen de verhoudingen anders liggen (en zal de topologische structuur wellicht zelfs minder ruimte vergen; denk hierbij aan gedetailleerde gemeentekaart van Nederland of aan een gedetailleerde landenkaart van de wereld).

In ruil voor dit geheugengebruik maakt Oracle de verwachting dat neighbour queries sneller zijn in de topologische datastructuur dan in de puur geometrische variant, ruimschoots waar: tijdsbesparingen van meer dan 50% worden makkelijk gehaald. Geometrische queries (geef

alle percelen in een bepaalde rechtehoek, welke percelen snijden met deze lijn) zijn echter gemiddeld 2 tot 3 keer langzamer, alhoewel uitschieters tot 20 keer trager niet ongebruikelijk zijn. Tijdens onze tests bleek Oracle Topology (nog) niet echt stabiel, sommige topologies werden opeens onbruikbaar (queries gaven onjuiste antwoorden), zonder dat in log- of tracefiles aanwijzingen voor de oorzaak werden gevonden. De enige oplossing bleek het opnieuw aanmaken en laden van alle tabellen en indexen. Het is niet onwaarschijnlijk dat deze problemen in verband staan met het feit dat de test-dataset geen valide topologie was, maar feit blijft dat Oracle daar klaarblijkelijk wel erg gevoelig voor is. Helaas is het niet gelukt om dit corrupt worden van topologieën te herhalen, waardoor het gissen blijft naar de oorzaak.

Conclusie

Met Oracle10g Topology is het uiteindelijk mogelijk om data in een topologische structuur in de database op te

slaan. De belangrijkste winst die hiermee geboekt kan worden in het beter managen van gestructureerde geografische data sets, zoals bodemkaart, eigendomspercelen, topografische kaart, landgebruikkaart, etc. (maar ook leidingen en wegnetten, hoewel niet getest). Dit zorgt dat de kwaliteit (consistentie) van de betreffende data gegarandeerd kan worden, onafhankelijk van welke edit omgeving wordt gebruikt. Aangezien de performance van topologische queries goed is, is dit product veelbelovend. Helaas bleek het product niet echt stabiel, maar dat wordt vermoedelijk veroorzaakt door het werken met niet valide datasets.

Men kan stellen dat alleen het valide houden van data een taak is voor de database, maar het blijft in de praktijk wel een gemis. Aanvullende softwarepakketten blijven nu toch nodig.

Ir. Friso Penninga is werkzaam bij de sectie GIS Technologie, OTB, Technische Universiteit Delft.



Ook u kunt lid worden van de OGH

en ontvangt automatisch OGH Visie d.m.v. nevenstaand aanmeldingsformulier. Niet leden kunnen zich gratis abonneren via de website www.ogh.nl

**Kopieer, vul in en fax naar OGH secretariaat
030 - 696 23 78 of meld je aan via www.ogh.nl**

Naam organisatie

Postadres

Postcode/plaats

Contactpersoon

Telefoonnummer

Datum Interesse lidmaatschap? ja/nee

☐ A 1 verzendadres/3 personen toegang/contributie € 200,-

☐ B 3 verzendadressen/3 personen toegang/contributie € 340,-

☐ C 6 verzendadressen/6 personen toegang/contributie € 450,-

Geen interesse lidmaatschap, wel interesse (gratis) OGH Visie ja/nee

OGH leden ontvangen 15% korting op specials van Oracle University via
www.oracle.com/nl/education/specials