

Databanken 3 Optimalisatie

Academiejaar 2018 - 2019

Jan Van Overveldt

Tom De Reys

KdG Karel de Grote
Hogeschool

Je bent reeds vertrouwd met een aantal optimalisatietechnieken in databanken. In de opleiding lag de focus lag tot nu toe op Oracle. Hoewel Oracle een zeer goede database technologie in handen heeft mogen we niet negeren dat er ook heel wat andere aanbieders zijn van databasetechnologie. Uiteraard zijn de optimalisatietechnieken die zij gebruiken grotendeels gelijkaardig. We herhalen daarom kort de belangrijkste technieken en plaatsen ze in de bredere context.

Database optimalisaties in Oracle RDBMS



Voorkennis

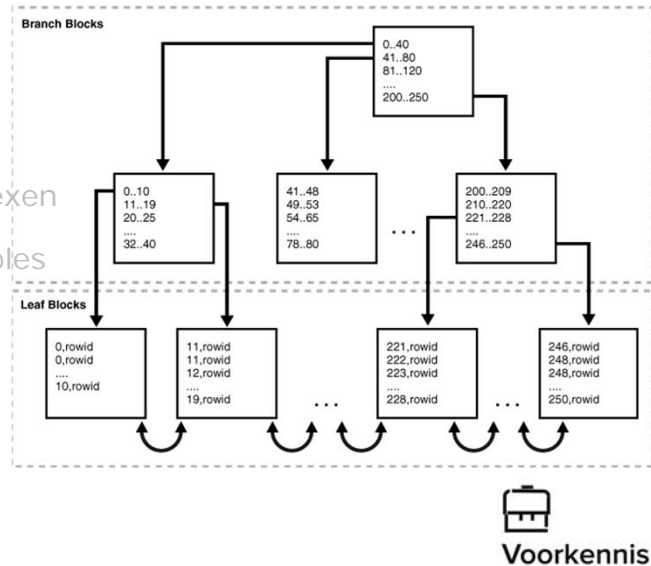
Inhoudstafel

1. Database optimalisaties in Oracle RDBMS
2. Basisconcepten MS SQL Server RDBMS
3. Database optimalisaties in MS SQL Server 2017 RDBMS

Even herhalen: Oracle performantie...

- Indexen:

- [B-Tree indexen](#)
- Bitmapindexen
- Function based indexen
- Index organized tables

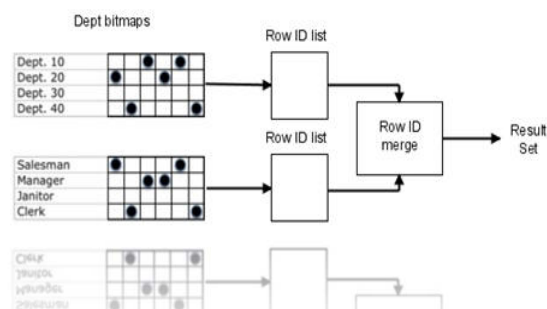


De meest gebruikte (en standaard) index is de **B-Tree index**. Een index is een datastructuur waarin je snel kan navigeren om de locatie van een bepaalde rij te vinden. Om zo snel mogelijk te ontdekken welke rijen voor een bepaald veld een bepaalde waarde bevatten wordt voor dat veld (het veld waarvoor een B-Tree index is aangemaakt) een boomstructuur opgezet waarbij de boom wordt opgedeeld in ranges van mogelijke waarden. In een beperkt aantal stappen kan de boomstructuur doorlopen worden voor de op te zoeken waarde. Voor elke waarde worden de verwijzing(en) naar de rijen met die betreffende waarde opgeslagen zodat het RDBMS snel te weten komt waar de relevante rijen zijn opgeslagen. Als je query echter veel rijen van een tabel moet teruggeven is een full table scan (het sequentieel doorlopen van de hele tabel) vaak sneller.

Even herhalen: Oracle performantie...

- Indexen:

- B-Tree indexen
- Bitmapindexen
- Function based indexen
- Index organized tables



5

Bij **bitmapindexen** wordt per voorkomende waarde in een bepaald veld een zogenaamde bitmap bijgehouden. De bitmap geeft voor elke rij aan of het veld die waarde heeft voor die rij. Bij een beperkt aantal mogelijke waarden (vb. woonplaats) is dit type bitmap vaak veel sneller dan een b-tree index. De bitmapindex wordt typisch in een datawarehouse omgeving toegepast. We gaan hier later dieper op in.

Even herhalen: Oracle performantie...

- Indexen:

- B-Tree indexen
- Bitmapindexen
- Function based indexen
- Index organized tables

6

Function based indexen zijn indexen waarbij je een index aanmaakt die berekende gegevens bevat.

Stel dat je in een applicatie studenten moet kunnen opzoeken op basis van achternaam (waarop een b-tree index is geplaatst).

De volgende query zal die index gebruiken.

```
select * from t_students where achternaam = "Van Overveldt"
```

Je wil echter niet afhankelijk zijn van die hoofdletters. Dus de uitgevoerde query zal zijn:

```
select * from t_students where UPPER(achternaam) = "VAN OVERVELDT"
```

Omdat je nu in de where-clause UPPER(achternaam) gebruikt zal de b-tree index niet worden ingezet. Door een function based index te plaatsen op UPPER(achternaam) zal die index steeds gebruikt worden als je in de where-clause die exacte functie gebruikt.

Index organized tables zijn gelijkaardig aan B-Tree indexen met het verschil dat het hier niet om een afzonderlijk opgeslagen index gaat maar dat de hele tabel georganiseerd is in een B-tree die gebaseerd wordt op de primary key van het veld. Deze methodiek zorgt

voor aanzienlijke performance winsten als je vaak opzoeken doet op de primaire sleutel (vb. orderID)

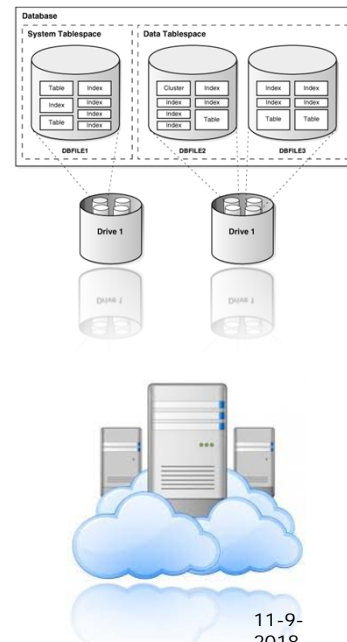
Even herhalen: Oracle performantie...



Voorkennis

-Tablespaces:

- Belangrijk in systemen waar de DBA goed op de hoogte is van de fysieke disks van de gebruikte volumes
- In de praktijk: Moeilijker voor flexibeler architecturen waar de volumes bijvoorbeeld in de cloud zitten.



Eén van de manieren om de performantie te verbeteren van een database is het gebruik van tablespaces te optimaliseren in functie van de database structuur.

Toch speelt dit de laatste jaren een minder grote rol. In flexibeler architecturen (lees public/private/hybrid clouds) heeft een DBA geen duidelijk zicht meer op de fysieke locaties van volumes. Het effect van het opzettelijk scheiden van tabellen en/of indexen op verschillende volumes is daarom onvoorspelbaar.

Even herhalen: Oracle performantie...



Voorkennis

-Execution plan optimalisatie

11-9-
2018

Elk RDBMS stelt voor een query een execution plan op. De optimizer van het RDBMS doet zijn best om de data zo efficiënt mogelijk op te halen. Toch kan het zijn dat je als DBA met extra 'HINTS' het plan kan verbeteren. Ook bij het gebruik van een ORM is het dus belangrijk dat je kritisch bekijkt welke queries het ORM lanceert en hier eventueel aanpassingen aan doet.

Basisconcepten SQL Server

In dit opleidingsonderdeel werken we niet met Oracle RDBMS, dit in tegenstelling tot de andere databank gerelateerde vakken in deze opleiding. Er zijn twee belangrijke redenen om andere databank technologie te gebruiken. Veel technieken binnen RDBMS worden door verschillende vendors aangeboden. Vaak gebeurt dit onder een andere naam en met een iets andere implementatie. Het bekijken van alternatieve vendors stelt je in staat die verschillende technieken te veralgemenen. Als IT-profiel word je veronderstelt met enige zelfstandigheid nieuwe technologie te verkennen. Door het uitwerken van een databank met andere technologie kan je als student deze vaardigheden oefenen. We geven in dit hoofdstuk niet meer dan een korte introductie. Met deze informatie als startpunt is het de bedoeling dat jullie als DBA de optimale beslissingen nemen bij het opzetten van een database. Veel optimalisatietechnieken die beschikbaar zijn bij Oracle zijn ook beschikbaar bij MS SQL. Toch zijn ook veel zaken anders.

In de opdracht zijn jullie zelf verantwoordelijk voor de optimalisatie van de op te zetten database en dienen jullie gebruik te maken van de documentatie die beschikbaar is.

Inhoudstafel

- 1.Database optimalisaties in Oracle RDBMS
- 2.Basisconcepten MS SQL Server**
3. Database optimalisaties in MS SQL Server 2017 RDBMS

SQL Server positionering

Magic Quadrant for Operational Database Management Systems



Magic Quadrant for Data Management Solutions for Analytics (As of February 2017)




11-9-2018

SQL Server is één van de populairste databank vendors.

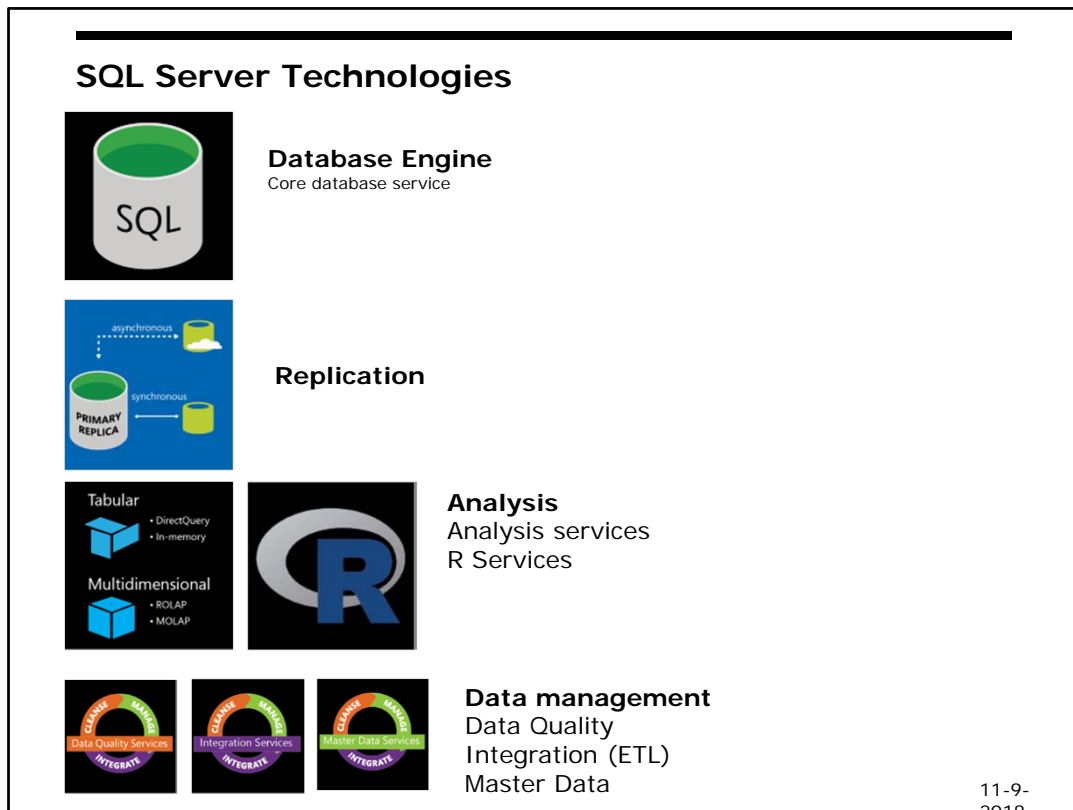
In de DB-Engines ranking komt SQL Server op de derde plaats na MySQL en Oracle.

Daarnaast wordt de technologie ook geprezen door Gartner. In de Magic Quadrants hierboven kan je zien dat zowel de operationele als analytics functionaliteiten van Microsoft zeer hoog scoren ten opzichte van de concurrentie.

Hoewel je dus misschien geen fan bent van Microsoft technologie, valt niet te ontkennen dat we in dit opleidingsonderdeel aan de slag gaan met een degelijke vendor.

SQL Server What's in a name	
Oracle	MS SQL
Index organized tables	Clustered index: <ul style="list-style-type: none"> Tabel geordend opgeslagen Default geordend op Primary key Heap table = Non clustered table (dus niet geordend)
B-Tree index	Nonclustered index
Function Based Index	Index on computed column <ul style="list-style-type: none"> Stap 1: Maak bij create van table computed column Stap 2: Zet een index op de computed column
Tablespace	Filegroup
Schema	Database
Package	+ /- Schema
	

Gelukkig moet je bij het gebruik van een nieuw RDBMS niet alles van 0 af aan opnieuw leren. Buiten nuanceverschillen is de SQL taal van Microsoft gelijklopend met die van Oracle. Elke technologie heeft echter wel bijvoorbeeld zijn eigen datatypes en dat is zeker niet anders voor MS SQL Server. Vervelend is ook dat Microsoft de onhebbelijke gewoonte heeft de veel gebruikte terminologie naast zich neer te leggen en gelijkaardige concepten gewoon een andere naam te geven. Hierboven geven we een lijstje met concepten en hun respectievelijke naam voor Oracle en MS SQL. Merk vooral op dat Schema een totaal andere betekenis heeft in SQL Server dan bij Oracle.



De database engine is een service die op het OS draait en die instaat voor alle gebruikelijke taken die een database moet uitvoeren.

Replication services zorgen er voor dat je replicatie kan toepassen op de data. Hiermee wordt een kopie van databaseobjecten opgeslagen op andere servers om zo de availability en performance van de databank te verbeteren. Replicatie is in SQL Server op verschillende manieren mogelijk.

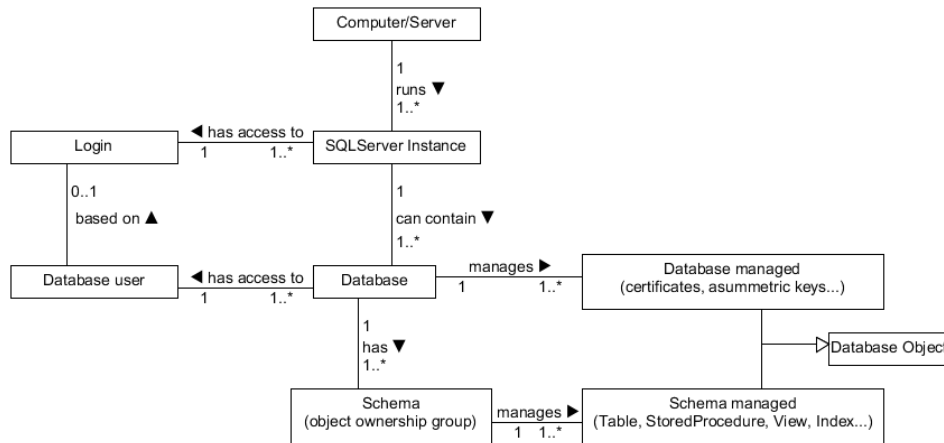
SQL Server biedt speciale services die data op een niet relationele manier opslaan om snelle analyses op data mogelijk te maken. Hier komen we in de volgende weken nog op terug.

SQL Server heeft ook een set van Data Management tools. Data Quality en Master Data vallen buiten de scope van dit vak maar we zullen Integration (ETL) wel behandelen. We maken hiervoor gebruik van een andere tool.

SQL Server Concepten



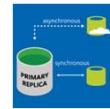
Database Engine: RDBMS Concepten



11-9-
2018

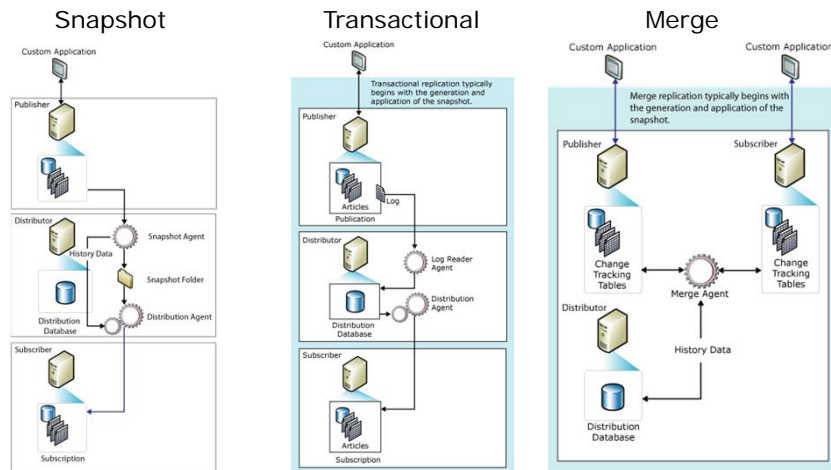
Bovenstaande model geeft een overzicht van (een deel van) de concepten die relevant zijn voor het relationele database management systeem. Meer uitleg kan je vinden op <https://docs.microsoft.com/en-us/sql/relational-databases/databases/databases>

SQL Server Concepten



Repliatie

- Kopie(s) van database naar verschillende servers.
- 3 Types



11-9-2018

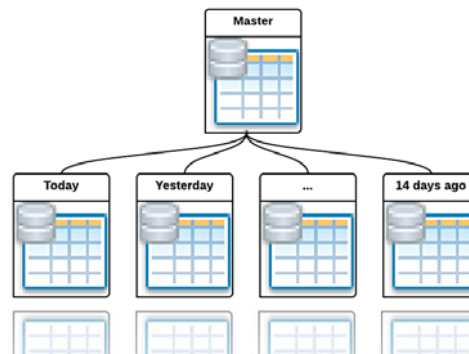
Repliatie wordt gebruikt om datasets te distribueren naar verschillende servers. Voor SQL Server betekent dit dat je databases gaat synchroniseren tussen databases. Er zijn verschillende redenen om dit te doen en optimalisatie is er hier één van

- Snapshot replicatie:
 - Bij snapshot replicatie worden er simpelweg snapshots (kopies) genomen van de publisher database.
 - De publisher database ontvangt alle INSERT/UPDATE/DELETE statements.
 - De distributor neemt periodiek een snapshot en die worden naar de subscribers gestuurd.
 - Subscribers databases zijn hierdoor niet steeds up to date
- Transactional replicatie:
 - Start met een snapshot
 - Vanaf dan gaat de distributor elke gelogde transactie van de publisher opslagen (M.a.w. INSERT/UPDATE/DELETE statements)
 - De subscribers voeren deze transacties vervolgens zelf ook uit.
 - Subscribers worden hierdoor snel up to date gebracht.
- Merge replicatie:
 - Ook op de Subscriber kunnen wijzigingen worden doorgevoerd waardoor synchronisatie in twee richtingen dient te gebeuren.
 - Conflict afhandeling is nodig als twee databases afzonderlijk wijzigingen op dezelfde data uitvoeren.

MS SQLServer partitioning

Partitionering:

- Splitst een tabel in verschillende subtabellen
- op basis van een bepaalde voorwaarde (vb. mensen jonger dan 20, mensen ouder dan 20)
- Als in de where die voorwaarde verwerkt zit (vb leeftijd = 18) dan zal enkel de relevante subtabel worden aangesproken.



11-9-
2018

Partitionering is een optimalisatietechniek die ook bij Oracle bestaat. Het komt er op neer dat een tabel opgeplitst wordt in subtabellen op basis van een bepaalde voorwaarde (vb. mensen jonger dan 20, mensen ouder dan 20). Als in de where die voorwaarde verwerkt zit (vb leeftijd = 18) dan zal enkel de relevante subtabel worden aangesproken. Aangezien deze subtabel aanzienlijk kleiner zal de query ook sneller gaan.

Als er voor partitionering gekozen wordt moet goed worden nagedacht over de partitiesleutel. De opsplitsing moet zo zijn vastgelegd zodat vaak gebruikte queries effectief maar één subpartitie moeten aanroepen.

Inhoudstafel

- 1.Database optimalisaties in Oracle RDBMS
- 2.Basisconcepten MS SQL Server
- 3. Database optimalisaties in MS SQL Server 2016 RDBMS**

Optimalisatie in SQL Server 2017 Documentatie

- Startpunt

Belangrijkste hoofdstukken:

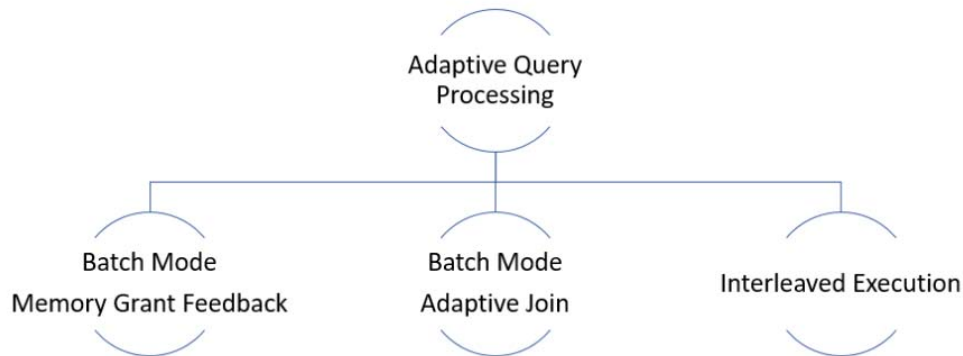
- SQL Server Technical Documentation
- Relational databases
- Partitions
- Replication
- Performance

11-9-
2018

Voor de projectopdracht zullen jullie de documentatie van MS SQL Server moeten doorspitten. In de opdracht staan een aantal voorwaarden beschreven waar de database aan moet voldoen. Op basis van deze voorwaarden is het aan jullie om de meest optimale database op te stellen. In de opdrachtbeschrijving vind je hierover meer verduidelijking.

Nieuwigheden in SQL Server 2017

Adaptive query processing



Bekijk hier zeker de volgende video in de link, dit geeft een voorbeeld over de 3 concepten:

<https://channel9.msdn.com/Shows/Data-Exposed/SQL-Server-2017-Adaptive-QP?ocid=player>

19

Dit is de start van een groep functionaliteiten die ervoor zorgen dat de database sommige zaken automatisch wil aanpakken. Het is de bedoeling dat deze functionaliteiten nog worden uitgebreid in de toekomst. Momenteel zijn er 3 delen.

- Batch Mode Memory Grant Feedback: Bij queries die meerdere keren worden uitgevoerd wordt er gezien naar het geheugengebruik en wordt dit zonodig aangepast indien er een te grote afwijking is tussen het geschatte aantal rijen en de actuele rijen.
 - Excessive Grant: Het aantal geschatte rijen is veel hoger dan het werkelijk aantal rijen, de geheugenbuffer die voorzien wordt is veel te groot, dit zorgt voor het beperken van parallele processen en zorgt voor performantieverlies omdat er telkens bij het uitvoeren van de query een te grote hoeveelheid geheugen moet worden klaargezet.
 - Spill event: Het aantal geschatte rijen is te laag, hierdoor wordt er te weinig geheugen vrijgezet waardoor de data op Disk moet worden opgeslagen in plaats van in het geheugen, dit zorgt voor serieus performantieverlies.
- Batch Mode Adaptive Join:
 - Hier gaat de software afhankelijk van het aantal rijen beslissen om vanaf een bepaalde cost een hash join te gebruiken in plaats van een nested loop.
- Interleaved Execution:
 - Dit gaat over optimalisatie van **multi-statement table valued functions (MSTVFs)** Dit ligt buiten scope van deze cursus. Hier gaat men tijdens het uitvoeren van een query de geschatte cardinaliteit(cost) aanpassen aan de

actuele kost van de subquery.

Thank you.

KdG Karel de Grote
Hogeschool

11/09/2018