1nf0rmat1ca

voortgezet onderwijs (havo/vwo)

SQL

Jacco Gnodde

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Onderwerpen:

inleiding
basiscommando's
functies
enkel-/meervoudige condities
sorteren
joins
subqueries
aggregeren
groeperen
tijdelijke tabellen
correlated subqueries
samenvoegqueries: union
muteren

Jacco Gnodde

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Onderwerpen:

inleiding
basiscommando's
functies
enkel-/meervoudige condities
sorteren
joins
subqueries
aggregeren
groeperen
tijdelijke tabellen
correlated subqueries
samenvoegqueries: union
muteren

Jacco Gnodde

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL – Tot nu toe…

Tot nu toe hebben we enkel gegevens opgehaald uit één enkele tabel.

In de meeste gevallen zal de benodigde informatie in verschillende tabellen zijn opgeslagen.

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
Gegevens in verschillende tabellen

Er zijn twee manieren om de gegevens uit verschillende tabellen te combineren:

1. door gebruik van cartesisch product: (impliciete en expliciete) joins
2. door gebruik van subqueries (dit wordt later besproken)

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
Cartesisch product

De meest voor de hand liggende manier om informatie uit meerdere tabellen te combineren is het cartesisch product.

Het cartesisch product van twee tabellen krijg je door elke rij uit de ene tabel te combineren met alle rijen uit de andere tabel.

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

Leden

lidnr	naam
1	Jan Punt
2	Bor de Wolf

x

Boeken

boeknr	auteur	titel	<lener>
1	Tolkien	De Hobbit	1
2	Reve	De Avonden	1
3	Mulisch	De Aanslag	2

```
SELECT *
FROM Leden, Boeken;
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

Leden

lidnr	naam
1	Jan Punt
2	Bor de Wolf

x

Boeken

boeknr	auteur	titel	<lener>
1	Tolkien	De Hobbit	1
2	Reve	De Avonden	1
3	Mulisch	De Aanslag	2

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

Leden

lidnr	naam
1	Jan Punt
2	Bor de Wolf

Boeken

boeknr	auteur	titel	<lener>
1	Tolkien	De Hobbit	1
2	Reve	De Avonden	1
3	Mulisch	De Aanslag	2

=

lidnr	naam	boeknr	auteur	titel	<lener>
1	Jan Punt	1	Tolkien	De Hobbit	1
1	Jan Punt	2	Reve	De Avonden	1
1	Jan Punt	3	Mulisch	De Aanslag	2

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

Leden

lidnr	naam
1	Jan Punt
2	Bor de Wolf

Boeken

boeknr	auteur	titel	<lener>
1	Tolkien	De Hobbit	1
2	Reve	De Avonden	1
3	Mulisch	De Aanslag	2

=

lidnr	naam	boeknr	auteur	titel	<lener>
1	Jan Punt	1	Tolkien	De Hobbit	1
1	Jan Punt	2	Reve	De Avonden	1
1	Jan Punt	3	Mulisch	De Aanslag	2

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

Leden

lidnr	naam
1	Jan Punt
2	Bor de Wolf

X

Boeken

boeknr	auteur	titel	<lener>
1	Tolkien	De Hobbit	1
2	Reve	De Avonden	1
3	Mulisch	De Aanslag	2

=

lidnr	naam	boeknr	auteur	titel	<lener>
1	Jan Punt	1	Tolkien	De Hobbit	1
1	Jan Punt	2	Reve	De Avonden	1
1	Jan Punt	3	Mulisch	De Aanslag	2
2	Bor de Wolf	1	Tolkien	De Hobbit	1
2	Bor de Wolf	2	Reve	De Avonden	1
2	Bor de Wolf	3	Mulisch	De Aanslag	2

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

lidnr	naam	boeknr	auteur	titel	<lener>
1	Jan Punt	1	Tolkien	De Hobbit	1
1	Jan Punt	2	Reve	De Avonden	1
1	Jan Punt	3	Mulisch	De Aanslag	2
2	Bor de Wolf	1	Tolkien	De Hobbit	1
2	Bor de Wolf	2	Reve	De Avonden	1
2	Bor de Wolf	3	Mulisch	De Aanslag	2

```
SELECT *
FROM Leden, Boeken;
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

lidnr	naam	boeknr	auteur	titel	<lener>
1	Jan Punt	1	Tolkien	De Hobbit	1
1	Jan Punt	2	Reve	De Avonden	1
1	Jan Punt	3	Mulisch	De Aanslag	2
2	Bor de Wolf	1	Tolkien	De Hobbit	1
2	Bor de Wolf	2	Reve	De Avonden	1
2	Bor de Wolf	3	Mulisch	De Aanslag	2

```
SELECT *
FROM Leden, Boeken;
WHERE lener=lidnr;
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

lidnr	naam	boeknr	auteur	titel	<lener>
1	Jan Punt	1	Tolkien	De Hobbit	1
1	Jan Punt	2	Reve	De Avonden	1
2	Bor de Wolf	3	Mulisch	De Aanslag	2

```
SELECT *
FROM Leden, Boeken
WHERE lener=lidnr;
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

Wat nou als de namen van de velden in de tabellen gelijk zijn aan elkaar?

Leden

lidnr	naam
1	Jan Punt
2	Bor de Wolf

x

Boeken

boeknr	auteur	titel	<lidnr>
1	Tolkien	De Hobbit	1
2	Reve	De Avonden	1
3	Mulisch	De Aanslag	2

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL

Cartesisch product - (impliciete) JOIN

lidnr	naam	boeknr	auteur	titel	<lidnr>
1	Jan Punt	1	Tolkien	De Hobbit	1
1	Jan Punt	2	Reve	De Avonden	1
2	Bor de Wolf	3	Mulisch	De Aanslag	2

```
SELECT *
FROM Leden, Boeken
WHERE Leden.lidnr=Boeken.lidnr;
```

Het American National Standards Institute (ANSI), tot 1966 de American Standards Association, is een Amerikaanse non-profitorganisatie die het beheer voert over een aantal vrijwillige, Amerikaanse standaarden en normeringen.

De ANSI is in 1918 opgericht onder de naam American Engineering Standards Committee (AESC) als een samenwerkingsverband van een aantal Amerikaanse ingenieursorganisaties (onder andere de Institute of Electrical and Electronics Engineers (IEEE), destijds de AIEE) met als doel het invoeren en promoten van het gebruik van verschillende standaarden en normeringen binnen de Verenigde Staten (later ook internationaal) om zo de concurrentiepositie van Amerikaanse bedrijven te verstevigen. In 1928 veranderde zij haar naam in American Standards Association (ASA). Aan deze naam herinnert de ASA-standaard voor lichtgevoeligheid van film. Na een reorganisatie in 1966 ging de organisatie door als United States of America Standards Institute (USASI). In 1969 nam zij haar huidige naam aan.

Ongetwijfeld de bekendste standaard van de ANSI is de ANSI-tekenset voor computers. Deze tekenset is zo bekend dat er vaak verwarring ontstaat en veel mensen denken dat ANSI in feite deze tekenset is en niet beseffen dat er nog een heel instituut bij hoort dat zich ook met totaal andere onderwerpen bezighoudt.

(bron: Wikipedia)

Er zijn drie redenen om als het kan expliciete joins te gebruiken:

1. Het is sneller, omdat de database engine de join niet meer hoeft te bepalen (het is onduidelijk of dit veel scheelt).
2. Het is semantisch beter. Bij impliciete joins komen de join voorwaarden in je WHERE clause terecht, terwijl het feitelijk niets te maken heeft met welke rijen je selecteert. Met veel joins in een query kan dit zelfs ook nog behoorlijk oplopen waardoor je WHERE onnodig lang en onduidelijk wordt. Bij een expliciete JOIN staat alles direct bij de JOIN en is het dus veel duidelijker. Je WHERE blijft kort en als je ooit eens een join eruit wil halen dan kan dat vrij eenvoudig.
3. Met expliciete joins heb je meer controle over hoe je joins maakt. Het biedt bijvoorbeeld de mogelijkheid om de JOIN volgorde te bepalen. Bij het combineren van drie tabellen of meer kan dit de performance van een SQL statement behoorlijk beinvloeden.

SQL
JOIN conditie

De join conditie (ON) bestaat eruit dat we een veld uit de ene tabel gelijk stellen aan veld uit de andere tabel

(veelal moet de vreemde sleutel uit de ene tabel gelijk zijn aan de corresponderende primaire sleutel in de andere tabel).

Aanvullende selectiecriteria neem je op in de WHERE clausule.

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
INNER JOIN - voorbeeld

Alleen de titels van de boeken die geleend zijn door Jan Punt:

```
SELECT titel
FROM Leden
INNER JOIN Boeken
ON (Leden.lidnr=Boeken.lidnr) ...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
INNER JOIN - voorbeeld

Alleen de titels van de boeken die geleend zijn door Jan Punt:

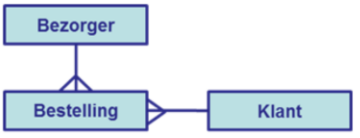
```
SELECT titel
FROM Leden
INNER JOIN Boeken
ON (Leden.lidnr=Boeken.lidnr)
WHERE (naam='Jan Punt');
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
drie tabellen combineren

Stel je wilt de naam zien van de klanten bij wie bezorger 'Ronald Marbus' een bestelling heeft afgegeven.

Klant (**klant_code**, wachtwoord, naam, adres, postcode, plaats, telefoon)
Bezorger (**bezorger_code**, wachtwoord, naam, gebdatum, telefoon)
Bestelling(**bestel_code**, datum, bestel_tijd, bezorg_tijd, <bezorger_code>, <klant_code>, korting)



1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
drie tabellen combineren

Klant (**klant_code**, wachtwoord, naam, adres, postcode, plaats, telefoon)
Bezorger (**bezorger_code**, wachtwoord, naam, gebdatum, telefoon)
Bestelling(**bestel_code**, datum, bestel_tijd, bezorg_tijd, <bezorger_code>, <klant_code>, korting)

De klanten bij wie bezorger 'Ronald Marbus' een bestelling heeft afgegeven:

```
SELECT Klant.naam
FROM   Klant
       INNER JOIN Bestelling
       ON (Klant.klant_code = Bestelling.klant_code)
       INNER JOIN Bezorger
       ON (Bestelling.bezorger_code = Bezorger.bezorger_code)
WHERE (Bezorger.naam = 'Ronald Marbus');
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
drie tabellen combineren

Klant (**klant_code**, wachtwoord, naam, adres, postcode, plaats, telefoon)
Bezorger (**bezorger_code**, wachtwoord, naam, gebdatum, telefoon)
Bestelling(**bestel_code**, datum, bestel_tijd, bezorg_tijd, <bezorger_code>, <klant_code>, korting)

De klanten bij wie bezorger 'Ronald Marbus' een bestelling heeft afgegeven:
(verkorte notatie mbv aliassen)

```
SELECT K.naam
FROM   Klant K
       INNER JOIN Bestelling B
       ON (K.klant_code = B.klant_code)
       INNER JOIN Bezorger Bz
       ON (B.bezorger_code = Bz.bezorger_code)
WHERE (Bz.naam = 'Ronald Marbus');
```

SQL
veel tabellen combineren

Het combineren van nog 4, 5, 6, … tabellen gaat op exact dezelfde manier:

- Selecteer de velden die moeten worden getoond;
- Bepaal welke tabellen je hiervoor nodig hebt;
- Koppel de tabellen via een (INNER) JOIN / ON;
- Neem de selectiecriteria op in de WHERE-clausule.

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van de pizza's op de bestelling met bestelcode 755.

```
SELECT naam
FROM Pizza P
INNER JOIN BesteldePizza BP
ON (P.pizza_code=BP.pizza_code) ...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van de pizza's op de bestelling met bestelcode 755.

```
SELECT naam
FROM Pizza P
INNER JOIN BesteldePizza BP
ON (P.pizza_code=BP.pizza_code) ...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van de pizza's op de bestelling met bestelcode 755.

```
SELECT naam
FROM Pizza P
INNER JOIN BesteldePizza BP
ON (P.pizza_code=BP.pizza_code)
WHERE (bestel_code=755);
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van klanten uit Soest die ooit voor 6 uur 's avonds een bestelling hebben geplaatst.

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van klanten uit Soest die ooit voor 6 uur 's avonds een bestelling hebben geplaatst.

```
SELECT naam
FROM Klant K
INNER JOIN Bestelling B
ON (B.klant_code=K.klant_code)
...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van klanten uit Soest die ooit voor 6 uur 's avonds een bestelling hebben geplaatst.

```
SELECT naam
FROM Klant K
INNER JOIN Bestelling B
ON (B.klant_code=K.klant_code)
...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van klanten uit Soest die ooit voor 6 uur 's avonds een bestelling hebben geplaatst.

```
SELECT naam
FROM Klant K
INNER JOIN Bestelling B
ON (B.klant_code=K.klant_code)
WHERE (plaats='Soest') ...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van klanten uit Soest die ooit voor 6 uur 's avonds een bestelling hebben geplaatst.

```
SELECT naam
FROM Klant K
INNER JOIN Bestelling B
ON (B.klant_code=K.klant_code)
WHERE (plaats='Soest') ...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van klanten uit Soest die ooit voor 6 uur 's avonds een bestelling hebben geplaatst.

```
SELECT naam
FROM Klant K
INNER JOIN Bestelling B
ON (B.klant_code=K.klant_code)
WHERE (plaats='Soest') AND
      (HOUR(bestel_tijd)<18);
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van klanten uit Soest die ooit voor 6 uur 's avonds een bestelling hebben geplaatst.

```
SELECT naam
FROM Klant K
INNER JOIN Bestelling B
ON (B.klant_code=K.klant_code)
WHERE (plaats='Soest') AND
      (HOUR(bestel_tijd)<18);
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van klanten uit Soest die ooit voor 6 uur 's avonds een bestelling hebben geplaatst.

```
SELECT DISTINCT naam
FROM Klant K
INNER JOIN Bestelling B
ON (B.klant_code=K.klant_code)
WHERE (plaats='Soest') AND
      (HOUR(bestel_tijd)<18);
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van alle pizza's die in januari 2015 minstens één keer door Peter van Tol zijn besteld:

```
SELECT P.naam
FROM Pizza P
INNER JOIN BesteldePizza BP ON (P.pizza_code=BP.pizza_code)
INNER JOIN Bestelling B ON (BP.bestel_code=B.bestel_code)
INNER JOIN Klant K ON (B.klant_code=K.klant_code)
...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van alle pizza's die in januari 2015 minstens één keer door Peter van Tol zijn besteld:

```
SELECT P.naam
FROM Pizza P
INNER JOIN BesteldePizza BP ON (P.pizza_code=BP.pizza_code)
INNER JOIN Bestelling B ON (BP.bestel_code=B.bestel_code)
INNER JOIN Klant K ON (B.klant_code=K.klant_code)
...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van alle pizza's die in januari 2015 minstens één keer door Peter van Tol zijn besteld:

```
SELECT P.naam
FROM Pizza P
INNER JOIN BesteldePizza BP ON (P.pizza_code=BP.pizza_code)
INNER JOIN Bestelling B ON (BP.bestel_code=B.bestel_code)
INNER JOIN Klant K ON (B.klant_code=K.klant_code)
WHERE (K.naam='Peter van Tol') ...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van alle pizza's die in januari 2015 minstens één keer door Peter van Tol zijn besteld:

```
SELECT P.naam
FROM Pizza P
INNER JOIN BesteldePizza BP ON (P.pizza_code=BP.pizza_code)
INNER JOIN Bestelling B ON (BP.bestel_code=B.bestel_code)
INNER JOIN Klant K ON (B.klant_code=K.klant_code)
WHERE (K.naam='Peter van Tol') ...
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van alle pizza's die in januari 2015 minstens één keer door Peter van Tol zijn besteld:

```
SELECT P.naam
FROM Pizza P
INNER JOIN BesteldePizza BP ON (P.pizza_code=BP.pizza_code)
INNER JOIN Bestelling B ON (BP.bestel_code=B.bestel_code)
INNER JOIN Klant K ON (B.klant_code=K.klant_code)
WHERE (K.naam='Peter van Tol') AND
      (MONTH(datum)=1) AND
      (YEAR(datum)=2015);
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van alle pizza's die in januari 2015 minstens één keer door Peter van Tol zijn besteld:

```
SELECT P.naam
FROM Pizza P
INNER JOIN BesteldePizza BP ON (P.pizza_code=BP.pizza_code)
INNER JOIN Bestelling B ON (BP.bestel_code=B.bestel_code)
INNER JOIN Klant K ON (B.klant_code=K.klant_code)
WHERE (K.naam='Peter van Tol') AND
      (MONTH(datum)=1) AND
      (YEAR(datum)=2015);
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

SQL
voorbeelden

De namen van alle pizza's die in januari 2015 minstens één keer door Peter van Tol zijn besteld:

```
SELECT DISTINCT P.naam
FROM Pizza P
INNER JOIN BesteldePizza BP ON (P.pizza_code=BP.pizza_code)
INNER JOIN Bestelling B ON (BP.bestel_code=B.bestel_code)
INNER JOIN Klant K ON (B.klant_code=K.klant_code)
WHERE (K.naam='Peter van Tol') AND
      (MONTH(datum)=1) AND
      (YEAR(datum)=2015);
```

1nf0rmat1ca
voortgezet onderwijs (havo/vwo)

Oefeningen (7)

Maak SELECT statements waarmee wordt getoond:

16. De datum van iedere bestelling met de naam van de klant die de bestelling heeft geplaatst;
17. De namen van de pizza's die ooit besteld zijn;
18. De naam van de bezorgers die in Baarn een bestelling hebben afgegeven;
19. De naam en de uiteindelijke prijs van iedere bestelde pizza;
20. De naam van iedere pizza die op 15 januari 2015 is besteld met de naam van de klant die de pizza heeft besteld.