

SQL

Inleiding relationele databases

bd.thrijswijk.nl

DBMS

DataBase Management System

- hiërarchische databases.
- netwerk databases.
- relationele databases.
- semantische databases.
- object oriënted databases.

Op dit moment gebruiken bijna alle DBMS'en het relationele model. Deze worden RDBMS'en genoemd.

SQL Relationele databases

bestaan uit tabellen

Tabel: leverancier

LevNr	Naam	Kwaliteit	Plaats
1	Simon	20	Den Haag
2	Janssen	10	Utrecht
3	Bernard	30	Utrecht
4	Janssen	20	Den Haag
5	Bout	30	Leiden

- entiteitstype (tabelnaam)
- entiteit (rij)
- attribuut (kolom)
- attribuutwaarde (hokje)
- sleutel (key)
- primaire sleutel (primary key)

Sleutels

Voorbeeld

- entiteit STUDENT met attributen:
 - studentnummer,
 - naam,
 - woonplaats,
 - geboortedatum,
 - SOFI-nummer en
 - paspoortnummer

Wat zijn kandidaat sleutels?
Welke sleutel kies je als primary key?

SQL artikel

Tabel: artikel

ArtNr	Naam	Kleur	Gewicht
1	moer	rood	12
2	bout	groen	17
3	schroef	blauw	17
4	schroef	rood	14
5	veer	blauw	12
6	as	rood	19

SQL Foreign key

Tabel: bestelling

LevNr	ArtNr	Aantal
1	1	300
1	2	200
1	3	400
1	4	200
1	5	100
1	6	100
2	1	300
2	2	400
3	2	200
4	2	200
4	4	300
4	5	400

SQL

Structured Query Language

- SQL is used to communicate with a database.
- SQL is an ANSI standard
- common RDBMS's use SQL:
 - Oracle
 - Sybase
 - Microsoft SQL Server
 - Ingres
 - MySQL
- SQL contains:
 - DDL = Data Definition Language
 - DML = Data Manipulation Language.

SQL SELECT

The SELECT statement is used to query the database and retrieve selected data.

```
SELECT
column1[,column2,etc]
FROM tablename
[WHERE condition];
```

[] = optional

- you can use a * to select all columns.
- operators that can be used in condition: =, <>, <, <=, >, >=, LIKE, AND, OR, NOT, IN, BETWEEN
- the result is a table.

SQL Voorbeeld

```
SELECT Gewicht, Naam  
FROM artikel;
```

Gewicht	Naam
12	moer
17	bout
17	schroef
14	schroef
12	veer
19	as

SQL Voorbeeld

```
SELECT Naam, Gewicht  
FROM artikel  
WHERE Kleur='rood';
```

Naam	Gewicht
moer	12
schroef	14
as	19

```
SELECT *  
FROM leverancier  
WHERE Naam LIKE 'B%';
```

LevNr	Naam	Kwaliteit	Plaats
3	Bernard	30	Utrecht
5	Bout	30	Leiden

SQL Voorbeeld

```
SELECT *  
FROM leverancier  
WHERE Plaats IN ('Den Haag',  
'Leiden');
```

LevNr	Naam	Kwaliteit	Plaats
1	Simon	20	Den Haag
4	Janssen	20	Den Haag
5	Bout	30	Leiden

```
SELECT Naam  
FROM artikel  
WHERE Gewicht BETWEEN 14 AND 17;
```

Naam
bout
schroef
schroef

SQL Mathematical

- operators
 - +, -, *, /, %
- functions
 - ABS(x), SIGN(x), MOD(x, y),
FLOOR(x), CEIL(x),
POWER(x, y), ROUND(x),
ROUND(x, d), SQRT(x)

```
SELECT LevNr, LevNr+Kwaliteit  
FROM leverancier;
```

LevNr	LevNr + Kwaliteit
1	21
2	12
3	33
4	24
5	35

SQL

CREATE TABLE

The CREATE TABLE statement is used to create a new table.

```
CREATE TABLE tablename  
(column1 datatype [constraint],  
column2 datatype [constraint],  
column3 datatype [constraint]);
```

[] = optional

- belangrijkste datatypes:
CHARACTER(size),
VARCHAR(size), INTEGER, FLOAT,
DATE, MONEY, BOOLEAN, AUTOINC,
BLOB
- belangrijkste constraints:
UNIQUE, NOT NULL, PRIMARY KEY

T:Rijswijk

13

SQL

Voorbeeld

```
CREATE TABLE artikel  
(ArtNr AUTOINC PRIMARY KEY,  
Naam VARCHAR(25) NOT NULL,  
Kleur VARCHAR(15),  
Gewicht INTEGER);
```

Tabel: artikel

ArtNr	Naam	Kleur	Gewicht
-------	------	-------	---------

Zoals je ziet bevat de tabel na de CREATE TABLE instructie nog geen data. De tabel kan later met behulp van de instructie INSERT worden gevuld.

T:Rijswijk

© 2002 Harry Broeders

14

SQL

INSERT

The INSERT INTO statement is used to insert or add a row of data into the table.

```
INSERT INTO tablename  
[(first_column,...last_column)]  
VALUES  
(first_value,...last_value);
```

[] = optional

De uitkomst van een SELECT statement kan ook gebruikt worden als invoer voor een INSERT statement.

```
INSERT INTO tablename  
[(first_column,...last_column)]  
SELECT ... FROM ... WHERE ...
```

[] = optional

T:Rijswijk

15

SQL

Voorbeeld

Maak een tabel genaamd goedeLeveranciers met de namen en vestigingsplaatsen van alle leveranciers die een kwaliteit hebben >15.

```
CREATE TABLE goedeLeverancier  
(GoedeLevNr AUTOINC PRIMARY KEY,  
Naam VARCHAR(30),  
Plaats VARCHAR(30));
```

```
INSERT INTO goedeLeverancier  
(Naam, Vestigingsplaats)  
SELECT Naam, Plaats  
FROM leverancier  
WHERE Kwaliteit > 15;
```

Tabel: goedeLeverancier

GoedeLevNr	Naam	Vestigingsplaats
1	Simon	Den Haag
2	Bernard	Utrecht
3	Janssen	Den Haag
4	Bout	Leiden

T:Rijswijk

16

SQL UPDATE

The UPDATE statement is used to update or change records that match a specified criteria.

```
UPDATE tablename
SET column = newvalue
[,nextcolumn = newvalue2...]
WHERE condition;
```

[] = optional

- Integriteitsproblemen (er kan maar 1 tabel tegelijk geupdate worden).
- Oplossing: **transactions** (ondeelbare groep statements die òf allemaal òf geen van al uitgevoerd worden).

SQL Voorbeeld

```
UPDATE artikel
SET Gewicht = 25
WHERE Kleur = 'blauw';
```

```
UPDATE artikel
SET Kleur = 'groen', Gewicht = 15
WHERE Naam = 'moer'
AND Gewicht < 15;
```

Tabel: artikel

ArtNr	Naam	Kleur	Gewicht
1	moer	groen	15
2	bout	groen	17
3	schroef	blauw	25
4	schroef	rood	14
5	veer	blauw	25
6	as	rood	19

SQL DELETE

The DELETE statement is used to delete records or rows from the table.

```
DELETE FROM tablename
[WHERE condition];
```

[] = optional

- Integriteitsproblemen (er kunnen keys die in andere tabellen als foreign keys gebruikt zijn verwijderd worden).

SQL Voorbeeld

Verwijder alle leveranciers met een kwaliteit lager dan 15.

```
DELETE FROM leverancier
WHERE Kwaliteit < 15
```

Tabel: leverancier

LevNr	Naam	Kwaliteit	Plaats
1	Simon	20	Den Haag
3	Bernard	30	Utrecht
4	Janssen	20	Den Haag
5	Bout	30	Leiden

Leverancier met LevNr 2 is nu uit de database verwijderd maar er staan nog verwijzingen naar deze leverancier in de tabel bestelling!

DROP

The DROP TABLE statement is used to delete a table and all rows in the table.

```
DROP TABLE tablename;
```

SELECT

The SELECT statement is used to query the database and retrieve selected data that match the criteria that you specify.

```
SELECT [ALL|DISTINCT] column-list
FROM table-list
[WHERE condition]
[GROUP BY column-list]
[HAVING condition]
[ORDER BY column-list
[ASC|DESC]];
```

[] = optional

Voorbeeld

```
SELECT DISTINCT Gewicht
FROM artikel;
```

Gewicht
12
17
14
19

Aggregate Functions

- MIN
- MAX
- SUM
- AVG
- COUNT
- COUNT(*) returns the number of rows in a table

```
SELECT AVG(Gewicht)
FROM artikel
WHERE Kleur = 'rood';
```

AVERAGE OF Gewicht
15.00

SQL GROUP BY

The GROUP BY clause will gather all of the rows together that contain data in the specified column(s) and will allow aggregate functions to be performed on the one or more columns.

```
SELECT column-list  
FROM list-of-tables  
GROUP BY column-list;
```

[] = optional

SQL Voorbeeld

Geef het minimale aantal en het maximale aantal van de bestellingen voor elk artikel.

```
SELECT ArtNr,  
min(Aantal), max(Aantal)  
FROM bestelling  
GROUP BY ArtNr;
```

ArtNr	MIN OF Aantal	MAX OF Aantal
1	300	300
2	200	400
3	400	400
4	200	300
5	100	400
6	100	100

SQL HAVING

The GROUP BY clause allows you to specify conditions on the rows for each group. The HAVING clause should follow the GROUP BY clause.

```
SELECT column-list  
FROM list-of-tables  
GROUP BY column-list  
HAVING condition;
```

[] = optional

SQL Voorbeeld

Geef het gemiddelde aantal per leverancier als het gemiddelde aantal groter is dan 250.

```
SELECT LevNr, avg(Aantal)  
FROM bestelling  
GROUP BY LevNr  
HAVING avg(Aantal) >= 250;
```

LevNr	AVERAGE OF Aantal
2	350.00
4	300.00

SQL ORDER BY

The GROUP BY clause allow you to display the results of your query in a sorted order (either ascending order or descending order) based on the columns that you specify to order by.

```
SELECT column-list  
FROM list-of-tables  
ORDER BY column-list [ASC|DESC];
```

[] = optional

SQL Voorbeeld

Geef alle bestellingen waarvan het aantal groter is dan 250 gesorteerd op aantal en daarna op artikelnummer.

```
SELECT *  
FROM bestelling  
WHERE Aantal > 250  
ORDER BY Aantal DESC, ArtNr;
```

LevNr	ArtNr	Aantal
2	2	400
1	3	400
4	5	400
1	1	300
2	1	300
4	4	300

SQL Table joins

Joins allow you to link data from two or more tables together into a single query result - from one single SELECT statement.

Joins can be recognized in a SQL SELECT statement if it has more than one table after the FROM keyword.

```
SELECT leverancier.Naam,  
       leverancier.Kwaliteit,  
       leverancier.Plaats,  
       artikel.Naam, artikel.Kleur,  
       artikel.Gewicht, bestelling.Aantal  
FROM bestelling, leverancier,  
       artikel  
WHERE bestelling.LevNr =  
       leverancier.LevNr AND  
       bestelling.ArtNr = artikel.ArtNr;
```

SQL Self joins

Ook is het mogelijk om gegevens te verkrijgen uit de combinatie van een tabel met zichzelf. Er zijn dan alias tabelnamen nodig.

SQL Voorbeeld

Geef alle mogelijke combinaties van 2 artikelen met dezelfde kleur.

```
SELECT x.Naam, y.Naam
FROM artikel x, artikel y
WHERE x.Kleur = y.Kleur
AND x.ArtNr < y.ArtNr;
```

Naam	Naam_1
schroef	veer
moer	schroef
moer	as
schroef	as

Waarom is de voorwaarde `x.ArtNr < y.ArtNr` nodig?

SQL Geneste selecties.

Voor sommige vragen is het nodig om geneste selecties toe te passen. Dit kun je doen door in de voorwaarde van de WHERE clause van een SELECT statement een ander SELECT statement op te nemen. Dit kan op 3 manieren:

- Door middel van de IN of NOT IN operators.
- Door middel van de EXISTS of NOT EXISTS operator.
- Door middel van een vergelijkingsoperator (`=`, `<>`, `<`, `>`, `<=` of `>=`).

SQL Voorbeeld

Geef het ArtNr, de Naam en de Kleur van bestellingen waarvan het Aantal > 300 is.

```
SELECT ArtNr, Naam, Kleur
FROM artikel
WHERE ArtNr IN (
    SELECT ArtNr
    FROM bestelling
    WHERE Aantal > 300
);
```

ArtNr	Naam	Kleur
2	bout	groen
3	schroef	blauw
5	veer	blauw

SQL Voorbeeld

Geef het ArtNr, de Naam en de Kleur van bestellingen waarvan het Aantal > 300 is.

```
SELECT ArtNr, Naam, Kleur
FROM artikel
WHERE EXISTS (
    SELECT *
    FROM bestelling
    WHERE artikel.ArtNr =
    ArtNr AND Aantal > 300
);
```

ArtNr	Naam	Kleur
2	bout	groen
3	schroef	blauw
5	veer	blauw

SQL Voorbeeld

Geef de nummers en de namen van de leveranciers waarbij ten minste 1 bestelling loopt voor een rood artikel.

```
SELECT LevNr, Naam
FROM leverancier
WHERE LevNr IN (
  SELECT LevNr
  FROM bestelling
  WHERE ArtNr IN (
    SELECT ArtNr
    FROM artikel
    WHERE Kleur = 'rood'
  )
);
```

LevNr	Naam
1	Simon
2	Janssen
4	Janssen

SQL Voorbeeld

Geef de nummers van de leveranciers die een kwaliteit hebben die hoger is dan de huidige gemiddelde kwaliteit van de leveranciers.

```
SELECT LevNr
FROM leverancier
WHERE Kwaliteit > (
  SELECT AVG(Kwaliteit)
  FROM leverancier
);
```

LevNr
3
5