

# SQL



## Onderwerpen:

inleiding

basiscommando's

functies

enkel-/meervoudige condities

sorteren

joins

subqueries

aggregeren

groeperen

tijdelijke tabellen

correlated subqueries

samenvoegqueries: union

mteren

## Onderwerpen:

inleiding

basiscommando's

functies

**enkel-/meervoudige condities**

sorteren

joins

subqueries

aggregeren

groeperen

tijdelijke tabellen

correlated subqueries

samenvoegqueries: union

mteren

Grootste complexiteit zit in WHERE-condities

Volledige WHERE-conditie bestaat uit één of meerdere condities

Voorbeelden van enkelvoudige condities:

- `plaats = 'Soest'`
- `datum = '2015-02-20'`
- `klant_code = 123`
- `basisprijs = 7.5`

Naast = kun je in WHERE ook gebruiken:

< : kleiner

> : groter

<= : kleiner of gelijk

>= : groter of gelijk

<> : ongelijk



!=

Voorbeelden:

datum < '2015-04-30'

klant\_code <> 123

basisprijs >= 8.5

plaats > 'B'

Bij het vergelijken met Strings wordt de ASCII-tabel gebruikt. Deze is afhankelijk van de taal-instelling van het RDBMS!

Met behulp van LIKE kan worden gecontroleerd of een tekstveld een bepaald deelpatroon bevat.

Hierbij wordt gebruik gemaakt van wildcards:

\_ → 1 teken

% → 0, 1 of meer tekens.

# SQL

## WHERE condities - LIKE

Voorbeeld:

```
SELECT *  
FROM Pizza  
WHERE UPPER(omschrijving)  
LIKE '%KAPPERTJES%';
```

pizza_code	naam	omschrijving	basisprijs
2	Napoletana	Tomaat, kaas, ansjovis, olijven, kappertjes en ore...	7.50
27	Salmone	Tomaat, kaas, zalm, uien, kappertjes en oregano	8.50
31	Pazza	Tomaat, kaas, uien, artisjokken, tonijn, olijven, ...	8.50
32	Inferno	Tomaat, kaas, ham, spaanse peper, kappertjes, sala...	8.50

## SQL WHERE condities - LIKE

Voorbeeld:

```
SELECT *  
FROM Pizza  
WHERE UPPER (naam)  
      LIKE '%HAWA_';
```

pizza_code	naam	omschrijving	basisprijs
8	Hawai	Tomaat, kaas, ham, annanas en oregano	7.50



## SQL WHERE condities - LIKE

Voorbeeld:

```
SELECT *  
FROM Pizza  
WHERE naam LIKE '%_%' ;
```

De pizzagegevens van alle pizza's waarvan de naam uit ten minste één teken bestaat.

De pizzagegevens van alle pizza's waarvan de naam uit ten minste één teken bestaat.

## SQL WHERE condities - LIKE

Om een speciaal teken (% of \_) als gewoon teken te gebruiken, moet het escape teken (#) worden gebruikt.

## SQL

### WHERE condities - LIKE

Om een speciaal teken (% of \_) als gewoon teken te gebruiken, moet het escape teken (#) worden gebruikt.

Voorbeeld: 

```
SELECT *
FROM Pizza
WHERE naam LIKE '%#_%';
```

De pizzagegevens van alle pizza's waarvan de naam een underscore (\_) bevat.



Klant (klant\_code, wachtwoord, naam, adres, postcode, plaats, telefoon)

Maak SELECT statements voor:

7. De namen van alle klanten die een wachtwoord hebben zonder hoofdletters.
8. De namen en de adressen van alle klanten die in een laan wonen.

## SQL

### WHERE condities - AND

Als er aan meerdere condities zijn waaraan moet worden voldaan, dan worden deze condities in de WHERE worden opgenomen, verbonden met AND.

Voorbeeld:

```
SELECT klant_code
FROM Klant
WHERE      plaats='Baarn' AND
          naam = 'Smit';
```

(de codes van de klanten met naam Smit uit Baarn)

## SQL

### WHERE condities - OR

Als er aan minstens één van meerdere condities moet worden voldaan, dan worden deze condities in de WHERE worden opgenomen, verbonden met OR.

Voorbeeld:

```
SELECT klant_code
FROM Klant
WHERE      plaats='Baarn' OR
           naam = 'Smit';
```

De klantcodes van iedereen die Smit heet en iedereen die in Baarn woont.

## SQL

### WHERE condities - OR

Als er aan minstens één van meerdere condities moet worden voldaan, dan worden deze condities in de WHERE worden opgenomen, verbonden met OR.

Voorbeeld:

```
SELECT klant_code
FROM Klant
WHERE      plaats='Baarn' OR
          naam = 'Smit';
```

De klantcodes van iedereen die Smit heet EN iedereen die in Baarn woont.



## SQL

### WHERE condities - NOT

NOT kan worden gebruikt wanneer een regel in het resultaat moet worden opgenomen wanneer er niet aan een conditie wordt voldaan.

Voorbeeld:

```
SELECT klant_code  
FROM Klant  
WHERE NOT (plaats = 'Soest');
```

= *De codes van alle klanten die NIET in Soest wonen*



## SQL

### WHERE condities - NOT

NOT kan worden gebruikt wanneer een regel in het resultaat moet worden opgenomen wanneer er niet aan een conditie wordt voldaan.

De NOT geldt alleen voor de conditie die er direct achter staat.

Als de NOT betrekking heeft op meerdere condities, dan moeten deze condities tussen haakjes worden geplaatst, direct achter de NOT.

## SQL

### WHERE condities - NOT

Voorbeeld:

```
SELECT naam  
FROM Pizza  
WHERE NOT
```

=

*De namen van de pizza's  
zonder vis of prei*

```
    ((omschrijving LIKE '%vis%') OR  
     (omschrijving LIKE '%prei%'));
```

Als de NOT betrekking heeft op meerdere condities, dan moeten deze condities tussen haakjes worden geplaatst, direct achter de NOT.

## SQL

### WHERE condities - NOT

Voorbeeld:

```
SELECT naam  
FROM Pizza  
WHERE NOT
```

```
((omschrijving LIKE '%vis%') OR  
(omschrijving LIKE '%prei%'));
```

=

*De namen van de pizza's  
zonder vis of prei*

```
A WHERE NOT(omschrijving LIKE '%vis%') AND  
NOT(omschrijving LIKE '%prei%');
```

Moeten deze condities tussen haakjes worden geplaatst,  
direct achter de NOT.

## SQL WHERE condities (volgorde)

Volgorde van de operatoren

→ belangrijk voor uitvoering van WHERE!

1. NOT
2. AND
3. OR

Volgorde aanpasbaar met haakjes (...)

## SQL WHERE condities (volgorde)

Volgorde van de operatoren: 1-NOT, 2-AND, 3-OR

```
SELECT klant_code  
FROM Klant  
WHERE naam='Jansen' OR  
       naam='De Groot' AND  
       plaats='Soest';
```

naam	plaats	
De Groot	Soest	✓
Pieterse	Soest	✗
Jansen	Soest	✓
De Groot	Baarn	✗
Pieterse	Baarn	✗
Jansen	Baarn	✓

Zie paragraaf 8.5 in Het SQL Leerboek

## SQL WHERE condities (volgorde)

Volgorde van de operatoren: 1-NOT, 2-AND, 3-OR

```
SELECT klant_code  
FROM Klant  
WHERE (naam='Jansen' OR  
       naam='De Groot') AND  
       plaats='Soest';
```

naam	plaats	
De Groot	Soest	✓
Pieterse	Soest	✗
Jansen	Soest	✓
De Groot	Baarn	✗
Pieterse	Baarn	✗
Jansen	Baarn	✗



Zie paragraaf 8.5 in Het SQL Leerboek



Wat betekenen volgende statements?

9. 

```
SELECT * FROM Klant
WHERE NOT naam <> 'Smit';
```
10. 

```
SELECT * FROM Klant
WHERE naam <> 'Smit' OR plaats <> 'Soest';
```
11. 

```
SELECT * FROM Klant
WHERE LENGTH(naam) > -1;
```
12. 

```
SELECT * FROM Klant
WHERE klant_code = 104 AND klant_code = 106;
```
13. 

```
SELECT * FROM Klant
WHERE klant_code >= 104 OR klant_code < 104;
```

Als je meerdere waardes voor een kolom hebt, kan je ook IN gebruiken

Voorbeeld:

```
SELECT naam  
FROM Klant  
WHERE plaats IN ('Soest', 'Baarn');
```

=  
**WHERE plaats='Soest' OR plaats='Baarn';**

TODO: voorbeeld op whiteboard



Als je meerdere waardes voor een kolom hebt, welke binnen een bepaald bereik liggen, kan je BETWEEN x AND y gebruiken

Voorbeeld:

```
SELECT naam
FROM Bezorger
WHERE YEAR (CURDATE ( ) ) - YEAR (gebdatum)
      BETWEEN 15 AND 20;
```

= ***Alle bezorgers die dit jaar minstens 15 en hoogstens 20 jaar oud zijn (inclusief!)***



Bestelling (bestel\_code, datum, bestel\_tijd, bezorg\_tijd,  
<bezorger\_code>, <klant\_code>, korting)

Maak SELECT statements voor:

14. De codes van de bestellingen die tussen 15-01-2015 en 15-02-2015 zijn geplaatst door één van de volgende klanten: 130, 197, 359 en 390.

**Volgende les:**  
**Sorteren van gegevens**