

SQL

The image shows the letters 'SQL' in a large, bold, blue font with a slight shadow. Below the letters, there are three blue, cylindrical icons representing database tables or data storage, arranged in a slightly overlapping cluster.

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 (vervolg)
aggregeren
groeperen
tijdelijke tabellen
correlated subqueries
samenvoegqueries: union
muteren

Oefeningen (9)

Klant	(klant_code, wachtwoord, naam, adres, postcode, plaats, telefoon)
Pizza	(pizza_code, naam, omschrijving, basisprijs)
Bestelling	(bestel_code, datum, bestel_tijd, bezorg_tijd, < bezorger_code>, <klant_code>, korting)
Besteldepizza	(besteldepizza_code, <pizza_code>, <bestel_code>, <bodem_code>, <formaat_code>, aantal)

Maak queries (met subquery) voor:

23. Alle bestel_codes van de bestellingen van de klant met naam 'Wegewijs'.
24. Alle bestel_codes van bestellingen waarbij er op één van de pizza's zowel gehakt als tonijn zit.
25. De naam van de klant die ooit meer dan 5 euro korting heeft gekregen op een bestelling.

De oplossing van opgave 25 was dus:

```
SELECT naam
FROM Klant
WHERE klant_code=
      ( SELECT klant_code
        FROM Bestelling
        WHERE korting >= 5 );
```

Wat nou als meerdere klanten ooit minstens 5 euro korting hebben gekregen?

SQL Subquery

De oplossing van opgave 25 was dus:

```
SELECT naam  
FROM Klant  
WHERE klant_code=  
    ( SELECT klant_code  
      FROM Klant  
      WHERE korting >= 5 );
```

een verzameling klantcodes

Wat nou als meerdere klanten ooit minstens 5 euro korting hebben gekregen?

SQL Subquery

Je kan '=' alleen gebruiken als je zeker weet dat de subquery maar één regel oplevert

Als het meer regels (kunnen) zijn, moet je IN gebruiken

```
SELECT naam
FROM Klant
WHERE klant_code IN
    ( SELECT klant_code
      FROM Bestelling
      WHERE korting >= 5 );
```

Wat nou als een klant meerdere keren minstens 5 euro korting heeft gekregen?

DISTINCT?

```
SELECT naam  
FROM Klant  
WHERE klant_code IN  
    ( SELECT klant_code  
      FROM Bestelling  
      WHERE korting >= 5 );
```


Wat nou als een klant meerdere keren minstens 5 euro korting heeft gekregen?

~~DISCOUNT?~~

```
SELECT naam
FROM Klant
WHERE klant_code IN
  ( SELECT klant_code
    FROM Bestelling
    WHERE korting >= 5 );
```

Wat nou als een klant meerdere keren minstens 5 euro korting heeft gekregen?

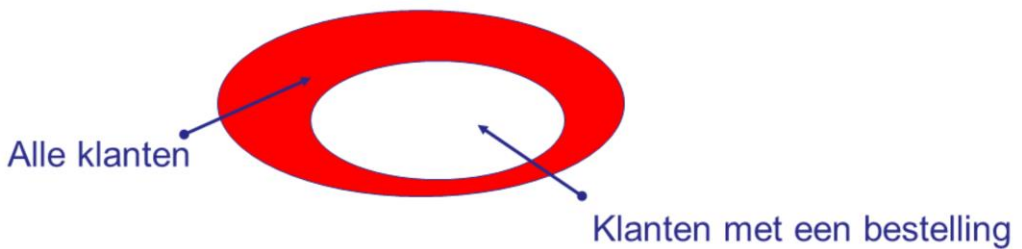
```
SELECT naam
FROM Klant
WHERE klant DISTINCT?
      ( SELECT klant_code
        FROM Bestelling
        WHERE korting >= 5 );
```

Wat nou als een klant meerdere keren minstens 5 euro korting heeft gekregen?

```
SELECT naam  
FROM Klant  
WHERE klant_ DISCOUNT?  
      ( SELECT klant_code  
        FROM Bestelling  
        WHERE korting >= 5 );
```

SQL Subquery – NOT IN

Geef alle klanten die nog nooit een bestelling hebben gedaan.



Dat zijn de klanten waarvan de gegevens wel in de tabel Klant (rode gebied) staan, maar waarvan de code niet voorkomt in de tabel Bestelling (het witte gebied).

Geef alle klanten die nog nooit een bestelling hebben gedaan.

```
SELECT *  
FROM Klant  
WHERE klant_code NOT IN (  
    SELECT klant_code  
    FROM Bestelling  
);
```

SQL

meerder subqueries in één statement

Een geneste subquery



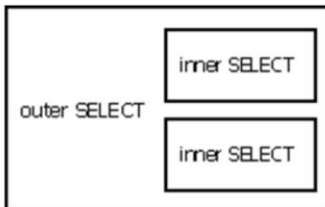
Wat doet deze query?

```
SELECT naam
FROM Klant
WHERE klant_code IN (
  SELECT klant_code
  FROM Bestelling
  WHERE bestel_code IN (
    SELECT bestel_code
    FROM BesteldePizza
    WHERE aantal >= 5 ) );
```

SQL

meerder subqueries in één statement

Twee subquery's naast elkaar



```
SELECT datum
FROM Bestelling
WHERE klant_code IN (
    SELECT klant_code
    FROM Klant
    WHERE plaats = 'Baarn') AND
bezorger_code = (
    SELECT bezorger_code
    FROM Bezorger
    WHERE naam = 'Afhalen');
```

Wat doet deze query?

- Je mag geen ORDER BY gebruiken
- Een subquery selecteert één kolom
- Moet rechts van de relationele operator staan

SQL

WHERE en subqueries: regels

- Je mag geen ORDER BY gebruiken
- Een subquery selecteert één kolom
- Moet rechts van de relationele operator staan

Dus geen:

```
SELECT naam
FROM Klant
WHERE ( SELECT klant_code
        FROM Bestelling
        WHERE bestel_code = 123
        ) = klant_code;
```

SQL

WHERE en subqueries: regels

- Je mag geen ORDER BY gebruiken
- Een subquery selecteert één kolom
- Moet rechts van de relationele operator staan

Maar:

```
SELECT naam
FROM Klant
WHERE klant_code =
      ( SELECT klant_code
        FROM Bestelling
        WHERE bestel_code = 123 );
```

SQL

joins en subqueries samen

De datum en de klantnaam van iedere bestelling zonder pizza Hawai.

(stap 1) De datum en de klantnaam van iedere bestelling:

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

(stap 2) De bestelcodes van alle bestellingen met een pizza Hawai:

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

(stap 3) De bestelcodes van alle bestellingen zonder pizza's Hawai:

```
SELECT bestel_code
FROM Bestelling
WHERE bestel_code NOT IN
    (
        bestelcodes van alle bestellingen
        met een pizza Hawai
    );
```

(stap 3) De bestelcodes van alle bestellingen zonder pizza's Hawai:

```
SELECT bestel_code
FROM Bestelling
WHERE bestel_code NOT IN
  ( SELECT bestel_code
    FROM BesteldePizza BP
    INNER JOIN Pizza P
    ON (BP.pizza_code=P.pizza_code)
    WHERE (naam='Pizza Hawai')
  );
```

SQL

joins en subqueries samen

(stap 4) De datum en de klantnaam zien van iedere bestelling zonder pizza Hawaii.

```
SELECT datum, naam  
FROM Klant K  
INNER JOIN Bestelling B  
ON (K.klant_code=B.klant_code)  
WHERE (bestel_code NOT IN (
```

```
) );
```


(stap 4) De datum en de klantnaam zien van iedere bestelling zonder pizza Hawai.

```
SELECT datum, naam
FROM Klant K
INNER JOIN Bestelling B
ON (K.klant_code=B.klant_code)
WHERE (bestel_code NOT IN (
        SELECT bestel_code
        FROM BesteldePizza BP
        INNER JOIN Pizza P
        ON (BP.pizza_code=P.pizza_code)
        WHERE (naam='Pizza Hawai') ) );
```



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)

Maak queries (met subquery) voor:

26. De namen van de klanten die dit jaar nog niets hebben besteld.
27. De namen van de bezorgers die zowel in januari als in februari een bestelling hebben 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)

Maak queries (met subquery) voor:

- 28a De codes van de bezorgers die in een andere maand dan maart 2015 hebben gewerkt.
 - 28b De codes van de bezorgers die uitsluitend in maart 2015 hebben gewerkt.
 - 28c De namen van alle klanten waar nog nooit een bestelling is afgegeven door een bezorger die uitsluitend in maart 2015 heeft gewerkt.
29. De datum van bestelling met de naam van de klant van alle bestellingen met uitsluitend grote of extra grote pizza's (maat_code 3 of 4).

Volgende les:

Aggregeren