

# SQL opgaven

## Relationele model:

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

## Opgaven:

1. Selecteer de klanten die bij fabriek F1 of fabriek F4 een bestelling hebben geplaatst.
2. Selecteer de klanten die bij fabriek F1 en fabriek F4 een bestelling hebben geplaatst.
3. Selecteer de artikelen die door klant Smit zijn besteld, de fabriek waarbij en de datum waarop het artikel is besteld.
4. Selecteer de klanten die artikel A5 niet hebben besteld.
5. Selecteer de fabriek die artikel A2 èn artikel A3 niet produceren.
6. Selecteer de klanten die bij tenminste twee fabrieken een bestelling hebben geplaatst.

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

1. Selecteer de klanten die bij fabriek F1 of fabriek F4 een bestelling hebben geplaatst.

```
SELECT klant
FROM B
WHERE fab# = 'F1' OR fab# = 'F4';
```

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

2. Selecteer de klanten die bij fabriek F2 en fabriek F3 een bestelling hebben geplaatst.

```
SELECT klant
FROM B
WHERE fab# = 'F2'
AND klant IN
      (SELECT klant
       FROM B
       WHERE fab# = 'F3');
```

◀◀ niet simpelweg AND !!

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

3. Selecteer de artikelen die door klant Smit zijn besteld, de fabriek waarbij en de datum waarop het artikel is besteld.

```
SELECT art#, fab#, datum
FROM BA, B
WHERE klant = 'Smit'
AND BA.b# = B.b#;
```

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

4. Selecteer de klanten die artikel A5 niet hebben besteld.

```
SELECT klant
FROM B
WHERE klant NOT IN
      SELECT klant
      FROM B, BA
      WHERE B.b# = BA.b#
      AND art# = 'A5';
```

◀◀ NIET art# <> 'A5'

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

5. Selecteer de fabrieken die niet artikel A2 en ook niet artikel A3 produceren.

```
SELECT fab#
FROM F
WHERE fab# NOT IN
      (SELECT fab#
       FROM F
       WHERE art# = 'A2' OR art# = 'A3');
```

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

6. Selecteer de klanten die bij tenminste twee fabrieken een bestelling hebben geplaatst.

```
SELECT klant
FROM B AS BX
WHERE klant IN
      SELECT klant
      FROM B
      WHERE B.fab# <> BX.fab#;
```

# SQL opgaven

## Relationele model:

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

## Opgaven:

1. Selecteer de artikelen en het aantal verschillende prijzen waarvoor ze besteld zijn.
2. Selecteer de artikelen en de gemiddelde voorraad van de artikelen waarvan de voorraad kleiner is dan de minimumvoorraad.
3. Selecteer de artikelen die door slechts één fabriek worden geproduceerd.
4. Selecteer de artikelen waarvan twee of meer bestellingen van meer dan 200 eenheden zijn.
5. Selecteer de fabrieken waarvan de fabriek een voorraad heeft kleiner dan de totale bij de fabriek bestelde hoeveelheid.
6. Selecteer de klanten met de meeste bestellingen.



## Deze serie opgaven heeft betrekking op setfuncties in SQL

COUNT (ATTRIBUUT) : aantal verschillende attribuutwaarden

COUNT (\*) : aantal tupels

SUM (ATTRIBUUT) : som attribuutwaarden

AVG (ATTRIBUUT) : gemiddelde attribuutwaarde

MAX (ATTRIBUUT) : maximale attribuutwaarde

MIN (ATTRIBUUT) : minimale attribuutwaarde

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

1. Selecteer de artikelen en het aantal verschillende prijzen waarvoor ze besteld zijn.

```
SELECT      art#, COUNT(prijs)
FROM        BA
GROUP BY   art#;
```

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

2. Selecteer de artikelen en de gemiddelde voorraad van de artikelen waarvan de voorraad kleiner is dan de minimumvoorraad.

```
SELECT    art#, AVG(vrd)
FROM      F
WHERE     vrd < minvrd
GROUP BY  art#;
```

**SETFUNCTIE vereist GROUP BY clause !**

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

3. Selecteer de artikelen die door slechts één fabriek worden geproduceerd.

```
SELECT      art#  
FROM        F  
GROUP BY   art#  
HAVING     COUNT(*) = 1;    ◀◀ na HAVING altijd setfunctie
```

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

4. Selecteer de artikelen waarvan twee of meer bestellingen van meer dan 200 eenheden zijn.

```
SELECT      art#
FROM        BA
WHERE       hoev > 200
GROUP BY   art#
HAVING      COUNT(*) > 1;    ◀◀ na HAVING altijd setfunctie
```

# SQL oplossing

F ( <u>fab#</u> , <u>art#</u> , vrd, minvrd)	Fabrieken produceren artikelen
B ( <u>b#</u> , klant, fab#, datum)	Klanten bestellen bij fabrieken
BA ( <u>b#</u> , <u>art#</u> , hoev, prijs)	Diverse artikelen per bestelling

5. Selecteer de fabrieken en de artikelen waarvan de fabriek een voorraad heeft kleiner dan de totale bij de fabriek bestelde hoeveelheid.

```
SELECT fab#, art#  
FROM F  
WHERE vrd <  
      SELECT SUM(hoev)  
      FROM B, BA  
      WHERE B.b# = BA.b#  
      AND BA.art# = F.art#  
      AND B.fab# = F.fab#
```

◀◀ volgorde is

◀◀ van belang

◀◀ relatie met

◀◀ buitenblok

## SQL oplossing

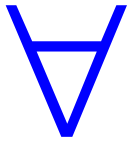
6. Selecteer de klanten met de meeste bestellingen.  
(COUNT en MAX gebruiken).

```
CREATE VIEW HULP (naam, aantal) AS
  SELECT      klant, COUNT(b#)
  FROM        B
  GROUP BY   klant;
```

```
SELECT naam
FROM   HULP
WHERE  aantal =
      SELECT MAX(aantal)
      FROM   HULP;
```

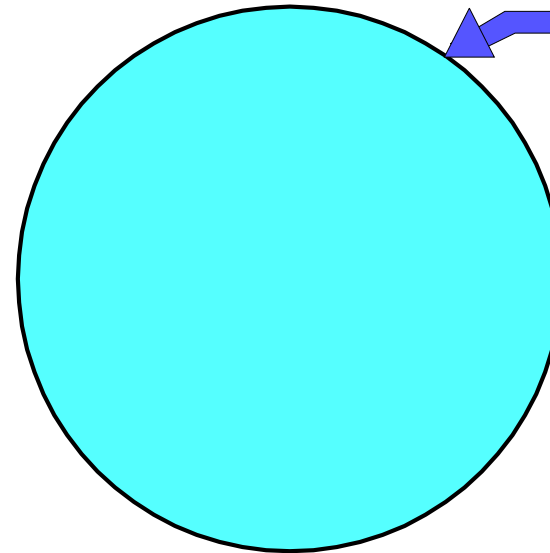
- VIEW:**
- is altijd nodig bij vraag met meer dan 1 functie !
  - na afloop verwijderen met de opdracht **DROP HULP;**

L( <u>L#</u> , Lnaam, Wpl)	leverancier
A( <u>A#</u> , Anaam, Kleur)	artikel
LA( <u>L#</u> , <u>A#</u> , Hoev)	leverantie



Geef leveranciersnamen van leveranciers die tenminste de artikelen leveren die leverancier L1 ook levert.

A# van L

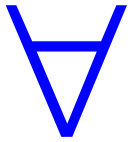


"Er bestaat geen A# van L1  
dat niet voorkomt bij L"

```
SELECT Lnaam
FROM L
WHERE NOT EXISTS (SELECT A#
                  FROM LA
                  WHERE L# = 'L1'
                  AND A# NOT IN (SELECT A#
                                FROM LA
                                WHERE L# = L.L#))
```



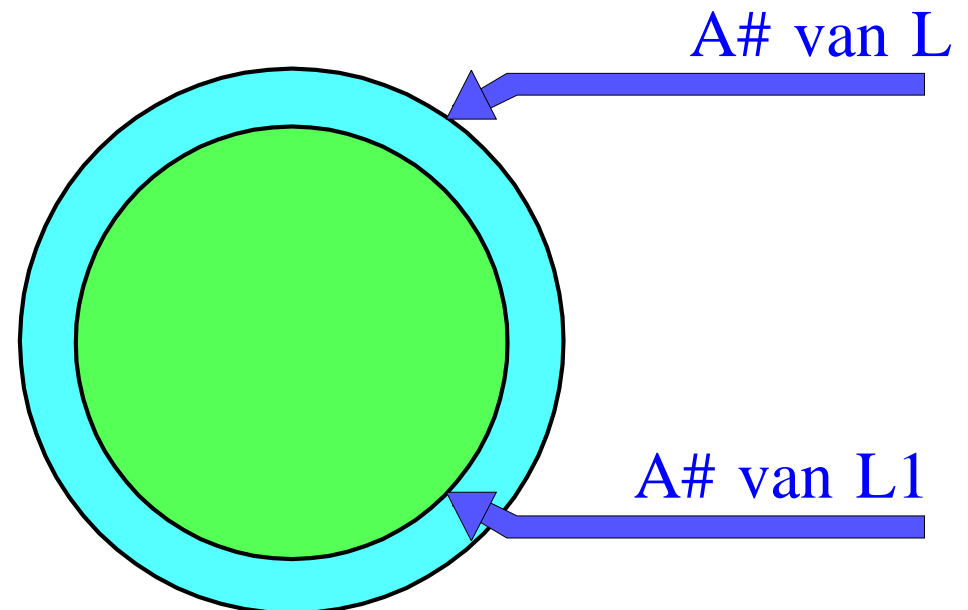
L(L#, Lnaam, Wpl) leverancier  
A(A#, Anaam, Kleur) artikel  
LA(L#, A#, Hoev) leverantie



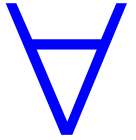
Geef leveranciersnamen van leveranciers die tenminste de artikelen leveren die leverancier L1 ook levert.

"Er bestaat geen A# van L1 dat niet voorkomt bij L"

```
SELECT Lnaam
FROM L
WHERE NOT EXISTS (SELECT A#
                  FROM LA
                  WHERE L# = 'L1'
                  AND A# NOT IN (SELECT A#
                                FROM LA
                                WHERE L# = L.L#))
```



L(L#, Lnaam, Wpl) leverancier  
A(A#, Anaam, Kleur) artikel  
LA(L#, A#, Hoev) leverantie



Geef leveranciersnamen van leveranciers die alle artikelen leveren.

"Er bestaat geen A# (van A!)  
dat niet voorkomt bij L"

```
SELECT Lnaam
FROM L
WHERE NOT EXISTS SELECT A#
                  FROM A
                  WHERE A# NOT IN SELECT A#
                                 FROM LA
                                 WHERE L# = L.L#
```

