

## Les S-02: Meer geavanceerde SQL-instructies

### 2.0 Overzicht les 1:

De **basisvorm** van een SQL query ziet er als volgt uit:

SELECT	(DISTINCT) <i>selecteer de velden</i>
FROM	<i>uit de tabel</i>
WHERE	<i>waar de volgende voorwaarde geldt ;</i>

Bij de WHERE voorwaarde is het **gegevenstype** van belang:

"....."	bij tekst
# ..... #	bij datum/tijd
niets	bij getallen

Er zijn verschillende **operatoren** die je bij de selectie van gegevens kunt gebruiken:

=	is gelijk aan
<	is kleiner dan
>	is groter dan
<=	is kleiner of gelijk aan
>=	is groter of gelijk aan
<>	is ongelijk aan

**Logische operatoren** gebruik je als je meerdere voorwaarden wilt stellen:

AND	als aan een voorwaarde EN een andere voorwaarde voldaan moet worden
OR	als aan een voorwaarde OF een andere voorwaarde voldaan moet worden
NOT	als aan een voorwaarde NIET voldaan moet worden

Ook kan je in SQL een aantal rekenkundige **functies** gebruiken:

COUNT(veldnaam)	telt het aantal rijen (records) waarin een bepaald veld verschillende waarden aanneemt
COUNT(*)	telt het aantal rijen (records) in een tabel
SUM(veldnaam)	telt de waarden van een bepaald veld in een tabel bij elkaar op
AVG(veldnaam)	geeft de gemiddelde waarde van een bepaald veld in een tabel
MAX(veldnaam)	geeft de maximale waarde van een bepaald veld in een tabel
MIN(veldnaam)	geeft de minimale waarde van een bepaald veld in een tabel

Met de GROUP BY instructie kan je gegevens per groep records selecteren

Met de HAVING instructie kan je voorwaarden aan een groep stellen.

## **2.1 Meerdere tabellen raadplegen**

Ook in deze lesbrief gebruiken we het voorbeeld van de volleybalvereniging.

Stel je voor dat je een tabel wilt maken van alle namen van de leden en de klasse waarin zij uitkomen. In dat geval moet je de gegevens van de tabellen met leden en teams combineren.

Je zou de volgende query uit kunnen proberen:

### **Voorbeeld 11**

Probeer de volgende query uit:

```
SELECT    voornaam, tussenvoegsel, achternaam, klasse
FROM      leden, teams;
```

Het resultaat is een tabel waarin ieder lid met iedere klasse wordt gecombineerd.

Dat komt doordat we in de query niet hebben aangegeven op welke manier de tabellen met leden en teams moeten worden samengevoegd. Dat doen we door in de WHERE instructie aan te geven dat een lid alleen met een klasse moet worden gecombineerd als deze klasse bij zijn/haar team hoort:

### **Voorbeeld 12**

Met de query:

```
SELECT    voornaam, tussenvoegsel, achternaam, klasse
FROM      leden, teams
WHERE     leden.teamnummer = teams.teamnummer;
```

selecteer je de namen van de spellers en de klasse waarin zij uitkomen. Omdat bij dit type query twee tabellen met elkaar worden “verbonden” noemen we deze een **inner join**.

Je kunt deze gegevens ook **sorteren** met de instructie ORDER BY:

### **Voorbeeld 13**

Met de query:

```
SELECT    voornaam, tussenvoegsel, achternaam, klasse
FROM      leden, teams
WHERE     leden.teamnummer = teams.teamnummer
ORDER BY  leden.teamnummer;
```

selecteer je de namen van de spelers en de klasse waarin zij uitkomen gesorteerd per team.

### **Voorbeeld 14**

Met de toevoeging DESC (descending = aflopend) of ASC (ascending = oplopend) kan je oplopend of aflopend sorteren:

```
SELECT    voornaam, tussenvoegsel, achternaam, klasse
FROM      leden, teams
WHERE     leden.teamnummer = teams.teamnummer
ORDER BY  leden.teamnummer ASC;
```

## **2.2 Werken met subqueries**

Er zijn situaties waarin je eerst een query nodig hebt om gegevens te selecteren die je vervolgens weer nodig hebt om verder op te selecteren. Je krijgt dan een query binnen een query. Een query binnen een query noemen we een **subquery**.

Stel je wilt de namen van de spelers selecteren die meer dan 120 euro contributie betalen. We zitten dan met het probleem dat de informatie uit twee tabellen gehaald moet worden. In de tabel “contributie” staat bij welke lidmaatschapscategorieën je meer dan 120 euro betaalt en in de tabel “leden” staat welke leden in die categorieën zitten.

### **Voorbeeld 15**

De query :

```
SELECT    lidmaatschapscategorie
FROM      contributie
WHERE     contributie.totalecontributie > 120;
```

selecteert uit de tabel “contributie” de lidmaatschapscategorieën met een contributie groter dan 120 euro. Als je deze query los uitvoert levert dat de categorieën A, B, C en S op.

### **Voorbeeld 16**

Met de query:

```
SELECT    voornaam, tussenvoegsel, achternaam
FROM      leden
WHERE     lidmaatschapscategorie IN
( SELECT  lidmaatschapscategorie
  FROM    contributie
  WHERE   contributie.totalecontributie > 120 );
```

selecteer je uit de tabel “leden” de namen van de spelers die in de categorieën A, B, C en S spelen en dus meer dan 120 euro contributie betalen.

We noemen de query die de namen selecteert de hoofdquery en de query die de lidmaatschapscategorieën selecteert de subquery.

## **OPDRACHT**

### **Opdracht 2.1**

Vertaal de bovenstaande query in een query met een inner join.

### **Opdracht 2.2**

Selecteer met behulp van een subquery de namen van de leden die geen coach hebben

## **2.3 Oefentoets**

### **Opgave 1**

Schrijf een query die alle dames uit het derde damesteam selecteert die voor 1980 geboren zijn.

### **Opgave 2**

Schrijf een query die de totaal te betalen contributie berekent.

### **Opgave 3**

Schrijf een query die per team berekent hoeveel leden er in het team zitten.

### **Opgave 4**

Schrijf een query die de namen selecteert van de leden die training krijgen van “E. Berends”.

### **Opgave 5**

Schrijf een query die alle namen van de teamgenoten van “David Liesman” selecteert.

**ANTWOORDEN****Opdracht 2.1**

```
SELECT    voornaam, tussenvoegsel, achternaam
FROM      leden, contributie
WHERE      leden.lidmaatschapscategorie=contributie.lidmaatschapscategorie
            AND contributie.totalecontributie > 120;
```

**Opdracht 2.2**

```
SELECT    voornaam, tussenvoegsel, achternaam
FROM      leden
WHERE      teamnummer IN
            (SELECT    teamnummer
             FROM      teams
             WHERE      teams.coach="geen");
```

**ANTWOORDEN OEFENTOETS****Opdracht 1**

```
SELECT    voornaam, tussenvoegsel, achternaam
FROM      leden
WHERE      teamnummer="D3" AND geboortedatum < #01/01/1980#;
```

**Opdracht 2**

```
SELECT    SUM(totalecontributie)
FROM      leden, contributie
WHERE      leden.lidmaatschapscategorie = contributie.lidmaatschapscategorie;
```

**Opdracht 3**

```
SELECT    teamnummer, count(*)
FROM      leden
GROUP BY teamnummer;
```

**Opdracht 4**

```
SELECT    voornaam, tussenvoegsel, achternaam
FROM      leden
WHERE      teamnummer IN
            ( SELECT    teamnummer
             FROM      teams
             WHERE      trainer = "E. Berends" );
```

**Opdracht 5**

```
SELECT    voornaam, tussenvoegsel, achternaam
FROM      leden
WHERE      teamnummer IN
            ( SELECT    teamnummer
             FROM      leden
             WHERE      voornaam = "David" AND achternaam = "Liesman");
```