

Databases cheatsheet

HO7

Werken met 1 tabel

```
SELECT [Gewenste kollomen]
FROM [Tabel waar kollomen inzitten];
```

Selectie

```
SELECT [Gewenste kollomen]
FROM [Tabel waar kollomen inzitten]
WHERE [Selectievoorwaarde bv. afdNr = 'D11'];
```

Vergelijkingsoperatoren

```
SELECT [Gewenste kollomen]
FROM [Tabel waar kollomen inzitten]
WHERE [Selectievoorwaarde bv. code = 56 , code >= 56];
```

Gebruik van LIKE of NOT LIKE

```
SELECT [Gewenste kollomen]
FROM [Tabel waar kollomen inzitten]
WHERE [Selectievoorwaarde] LIKE ['%P'];
```

Wildcards

naam beginnend met P: 'P%'

naam eindigend op C: '%C'

naam beginnend met ABC: 'ABC%'

naam eindigend met XYZ: 'XYZ%'

naam beginnend met A en eindigend op C: 'A%C'

naam die ergens SON bevat: '%SON%'

AND, OR en NOT

AND

```
SELECT [Gewenste kolumnen]
FROM [Tabel waar kolumnen inzitten]
WHERE [Selectievoorwaarde1] AND [Selectievoorwaarde2];
```

Zowel selectievoorwaarde1 en Selectievoorwaarde2 moeten gelden om gekozen te worden uit de tabel.

OR

```
SELECT [Gewenste kolumnen]
FROM [Tabel waar kolumnen inzitten]
WHERE [Selectievoorwaarde1] OR [Selectievoorwaarde2];
```

Een van de twee selectievoorwaarden moet gelden om gekozen te worden uit de tabel.

Vermijd het gebruik van de NOT operator

BETWEEN

```
SELECT [Gewenste kolumnen]
FROM [Tabel waar kolumnen inzitten]
WHERE [Selectievoorwaarde1] BETWEEN ['waarde1'] AND ['waarde2'];
```

IN

Hierdoor kan men de inhoud van een veld vergelijken met een lijst van waarden.

```
SELECT [Gewenste kolumnen]
FROM [Tabel waar kolumnen inzitten]
WHERE [Selectievoorwaarde1] IN (['waarde1'] , ['waarde2']);
```

Gebruik van NULL waarde

```
SELECT [Gewenste kolumnen]
FROM [Tabel waar kolumnen inzitten]
WHERE [Selectievoorwaarde1] IS NULL;
```

Combinatie voorbeeld

```
SELECT [Gewenste kolumnen]
FROM [Tabel waar kolumnen inzitten]
WHERE [Selectievoorwaarde1] = [waarde]
      AND [Selectievoorwaarde2] IN (waarde, waarde)
      AND [Selectievoorwaarde3] BETWEEN [waarde1] AND [waarde2]
      AND [Selectievoorwaarde4] IS NULL;
```

Formateren van het resultaat

Gebruik van ORDER BY

ALGEMENE VORM

```
SELECT *
FROM [Tabel waar kolumnen inzitten]
ORDER BY [Kolom-Specificatie] ASC/DESC;
```

Vb.

```
SELECT [Gewenste kolumnen]
FROM [Tabel waar kolumnen inzitten]
WHERE [Selectievoorwaarde1] IS NULL
ORDER BY [Selectievoorwaarden] DESC;
```

Gebruik van DISTINCT/ALL

```
SELECT = SELECT ALL
```

```
SELECT DISTINCT
```

Distinct gaat alle dubbele waarden overslaan\

BELANGRIJK VOORBEELD

```
SELECT FNaam
FROM Werknemer
WHERE FNaam = 'De Bruyn';
```

Elke keer dat 'De Bruyn' in de kolom staat wordt getoond voornaam maakt niet uit.

```
SELECT DISTINCT FNaam
FROM Werknemer
WHERE FNaam = 'De Bruyn';
```

Nu kom je 1 keer 'De Bruyn' uit ook al zijn er 2 werknemers met achternaam 'De Bruyn' en andere voornaam.

Oplossing

```
SELECT DISTINCT FNaam , VNaam
FROM Werknemer
WHERE FNaam = 'De Bruyn';
```

Gebruik van Aliassen

```
SELECT DISTINCT [Kolom alias] AS ['KolomNaam']
FROM [Tabel waar kolommen inzitten];
```

Enkele handige functies

Datum en tijd functies

```
CURDATE();
```

Huidige datum JJJ-MM-DD.

```
CURTIME();
```

Huidige tijd UU-MM-SS.

```
DATEDIFF(expr1, expr2)
```

Geeft verschil expr1 - expr2 in aantal dagen.

```
DATE_FORMAT(Datum, Format);
```

```
SELECT DATE_FORMAT('2017-10-15' , '%W %d %M %Y')
```

```
DAY();
```

Geeft de dag van de maand weer. getal tussen 1-31.

```
NOW()
```

Geeft huidige datum en tijd weer in formaat YYYY-MM-DD HH:MM:SS

Control flow functies

```
SELECT *  
CASE  
    WHEN [Selectievoorwaade1] IS NULL  
        THEN 'Result'  
    WHEN [Selectievoorwaade1] = 'value'  
        THEN 'Result'  
    ELSE 'result'  
END AS []  
FROM []  
ORDER BY []
```

```
IF(expr1, expr2,expr3)
```

Geeft expr2 indien expr1 waar is en verschillend van NULL zoniet is het resultaat expr3

```
IFNULL
```

```
NULLIF
```

Statische Functies

```
AVG
```

Gemiddelde

```
SUM  
MIN, MAX
```

```
COUNT(*) : Telt Totaal aantal rijen  
COUNT(KolomNaam) : Telt aantal velden in kolom  
COUNT(DISTINCT KolomNaam)
```

Gebruik van GROUP BY

Vb. Geef het maximum, het minimum en het gemiddelde salaris van alle werknemers per afdeling.

```
SELECT Afd, MAX(Salaris), MIN(Salaris), AVG(Salaris)  
FROM Werknemer  
GROUP BY Afd
```

Werken met meerdere Tabellen

JOIN van verschillende tabellen

INNER JOIN van verschillende tabellen

2 manieren

```
SELECT [Kol1], [kol2], [Kol3]
FROM [Tabel1] JOIN [Tabel2] ON [Tabel1.Kol2] = [Tabel2.Kol3]
```

De ON Clause specificeert dan een bepaalde relatie tussen de rijen die moeten samengevoegd worden

```
SELECT [KOL1], [Kol2]
FROM [Tabel1] , [Tabel2]
WHERE [Tabel1].[Kol1] = [Tabel2].[Kol2];
```

Join van een tabel met **Zichzelf**

```
SELECT X.[KOL1], Y.[Kol2]
FROM [Tabel1]X JOIN [Tabel1]Y ON X.[KOL1] = Y.[KOL1]
WHERE X.[KOL2] >= Y.[KOL2]+5;
```

OUTER JOIN

Er zijn 3 types van outer join

```
LEFT (OUTER) JOIN
```

Retourneert alle rijen van de eerst genoemde tabel in de FROM clause

```
RIGHT (OUTER) JOIN
```

Retourneert alle rijen van de tweede tabel in de FROM clause

```
FULL JOIN
```

Retourneert ook rijen uit de eerste en tweede tabel die geen corresponderende entry hebben in andere tabel. De resultset bevat steeds alle rijden van de eerste tabel en alle rijen van de tweede tabel.

```
SELECT afdnr, afdNaam , count(FNaam)
FROM afdeling left outer join werknemer ON afd = afdnr
GROUP BY afdnr, afdNaam
ORDER BY afdnr;
```

CROSS JOIN

```
SELECT [Kollomen]
FROM [Tabel] CROSS JOIN [tabel2] ON [kol1] = [kol2]
```

Een CROSS JOIN is vergelijkbaar met het cartesisch product. Het aantal rijen in de resultaattabel is gelijk aan het aantal rijen in de eerste tabel maal het aantal rijen in de tweede tabel.

Deze join wordt in de praktijk zelden gebruikt, maar kan handig zijn om testdata te genereren

UNION

Union combineer 2 of meer tabellen tot 1.

```
SELECT [kol1], [kol2]
FROM [Tabel1]
UNION
SELECT [kol1], [kol2]
FROM [Tabel2]
```

HO9

Wijzigen van de inhoud van tabellen

INSERT

1 Rij

```
INSERT INTO [tabel]
VALUES('nieuwe waarde', 'nieuwe waarde', 'nieuwe waarde');
```


- Ofwel vermeld je in de lijst van kolomnamen, enkel de kolomnamen, die een waarde zullen krijgen. De niet vermelde kolomnamen krijgen dan automatisch NULL-waarden toegewezen.
- Je vermeldt geen lijst van kolomnamen en gebruikt het NULL-sleutelwoord voor de onbekende velden

NR	Vnaam	Init	Fnaam	Afd	Tel	Indienst	Gesl	Salaris
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

Meerdere rijen

```
CREATE TABLE [New Name]
[kol1] CHAR NOT NULL,
VNaam VARCHAR(12) NOT NULL,
(2)Init CHAR(1) NOT NULL,
(3)FNaam VARCHAR(15) NOT NULL,
(4)Afd CHAR(3) NOT NULL,
(5)Tel CHAR(4),
Indienst INT,
Gesl CHAR(1),
Salaris DECIMAL(8,2);
```

Vul new table nu op met data van 2 bedienden

```
INSERT INTO AFDD41(Nr, VNaam, Init, FNaam, Afd, Indienst, Gesl)
SELECT Nr, VNaam, Init, FNaam, Afd, Indienst, Gesl
FROM Werknemer
WHERE NR LIKE '4' OR Nr IN (SELECT ManNr FROM Afdeling WHERE AfdNr = 'D41');
```

UPDATE

```
UPDATE [Table]
SET [kol1]= 'value', [Kol2]= 'value'
WHERE [kol3]='value'
```

Indien er geen WHERE vermeld wordt, zullen alle rijen in de opgegeven tabel gewijzigd worden!!!

DELETE

```
DELETE
FROM [Table]
```

```
WHERE [kol] = 'value'
```

DELETE met Subselect

```
DELETE  
FROM [Table]  
WHERE [Kol] IN (SELECT [Kol] FROM [Table])
```

HO10

Aanmaken, wijzigen en verwijderen van tabellen

Creatie van een Table

```
CREATE TABLE AFDD41(  
  NR CHAR(3) NOT NULL,  
  VNaam VARCHAR(12) NOT NULL,  
  Init CHAR(1) NOT NULL,  
  FNaam VARCHAR(15) NOT NULL,  
  Afd CHAR(3) NOT NULL,  
  Tel CHAR(4) NULL,  
  InDienst INT NULL,  
  Gesl CHAR(1) NULL,  
  Salaris DECIMAL(8,2) NULL;
```

Wijzigen van de tabelStructuur

```
ALTER TABLE tabelnaam  
ADD (Kolomnaam type[, kolomnaam type])
```

DROP : Verwijderen van een volledige tabel

```
DROP TABLE [Tabelnaam];
```