

DML

Algemeen

- Strings in enkele quotes
- Namen (van tabellen etc.) niet in quotes!
- lijsten scheiden met komma's
- statements afsluiten met ;

- DML: Data Manipulation Language
- DDL: Data Definition Language

- Select?
- Create?
- Show?

Select

SELECT

[DISTINCT]

velden

FROM tabellen

WHERE voorwaarden

[GROUP BY {col_name | expr | position}]

[HAVING voorwaarden]

[ORDER BY {col_name | expr | position}]

[ASC | DESC], ...]

velden

- veldnaam
- als meerdere tabellen dezelfde veldnamen gebruiken:

tabelnaam.veldnaam

- voor output kun je een naam toevoegen

... as 'naam'

Tabellen

- lijst met tabelnamen
- eventueel 'afkortingen'

langetabelnaam a,

voorwaarden

- naam = 'arne'
- =, <, >, !=, <>, >=, <=
- is null
- is, and, or, not (voor booleans)
- expr LIKE pat [ESCAPE 'escape_char']
 - _ % \
- in
- subquery

Groeperingen

- group by groepeert
- having werkt op de groep
- having kan ook met aggregates werken

volgorde

SELECT

[DISTINCT]

6 velden

1 FROM tabellen

2 WHERE voorwaarden

3 [GROUP BY {col_name | expr | position}]

4 [HAVING voorwaarden]

5 [ORDER BY {col_name | expr | position}]

[ASC | DESC], ...]

select

- `Select * from tabelnaam;`
- `Select count(veld)...` => Aantal records met een waarde in veld
- `Select count(*)...` => aantal records in een relatie
- `Select sum(veld)...` => totaal van alle waarden in de kolom
- `Select avg(veld)...` => gemiddelde van alle waarden in de kolom
- `Select max(veld)...` => grootste van alle waarden in de kolom
- `Select min(veld)...` => kleinste van alle waarden in de kolom

Where

- ...where [not] voorwaarde [{and | or [not] voorwaarde}...]
- =, <, >, !=, <>, >=, <=, is

where

- `expressie [not] like expressie`
 - Wildcards `_` `%`
 - Escape `\`

`select naam from studenten where naam like 'A%';`

`select naam from studenten where naam like 'Ar_e';`

`select naam from studenten where achternaam like '%_';`

where

- Expressie [not] IN (expressie [{, expressie}...])

Select naam from studenten where wplaats in ('Groningen',
'Garrelsweer');

Group by ... having

- Groepeert records in het tussenresultaat (na de joins) op de aangegeven velden
- Levert 1 resultaat record per groep!
- In de select mogen alleen velden staan die in group by gebruikt worden (of functies)
- Having werkt op de group by en mag statistische functies bevatten

Group by ... having

Gemiddelde van cijfer per student die meer dan 1 examen gedaan heeft

```
Select stud_id, avg(cijfer)
  From examens
 Group by stud_id
 Having count(cijfer > 1);
```

insert

```
Insert into tabelnaam([kolomnaam{, kolomnaam}...])  
  values(expressie[ {, expressie} ...]);
```

- Elke kolomnaam matcht op precies een value
- Als je kolommen weglaat: vullen op volgorde van definitie

update

Update tabelnaam

Set kolomnaam = expressie[{, kolomnaam=expressie}...]

[where conditie];

update examens

set cijfer = 0

where stud_id = 'parn';

LET OP WHERE!!!

delete

Delete from tabel [where conditie];

- Verwijdert alleen rijen
- Truncate verwijdert rijen en reset indices etc.
- Drop verwijdert de tabel c.q. kolommen