

SQL Querys

Les 1

Tabelnaam

Kolomnaam/Attribuut

Record = data in kolommen

FROM = Aangeven uit welke tabel je wilt zoeken

SELECT = Er wordt iets uit een tabel geselecteerd

Query-voorbeeld: SELECT first_name, last_name FROM actor;

* = Alle gegevens van tabel selecteren/aantonen

Query-voorbeeld: SELECT * FROM actor

DISTINCT = Kolomnaam/attribuut wordt uit tabel geselecteerd en is uniek als resultaat (haalt alle dubbelingen eruit)

Query-voorbeeld: SELECT DISTINCT district FROM address;

AS = Kolomnaam/attribuut wordt uit tabel geselecteerd en krijgt bepaald label

Query-voorbeeld: SELECT DISTINCT district AS provincie FROM address;

WHERE = Kolomnaam/Attribuut wordt uit tabel geselecteerd waarvoor geldt dat het gelijk is aan iets

Query-voorbeeld: SELECT * FROM actor WHERE first_name = "Jeff";

ORDER BY = Kolomnaam/attribuut wordt gesorteerd: ASC = oplopend / DESC = aflopend

Query-voorbeeld: SELECT * FROM actor WHERE first_name = "Jeff" ORDER BY last_name DESC/ASC

Les 2

Tabelnaam

Kolomnaam/Attribuut

Record = data in kolommen

Veel voorkomende vergelijkingen

> Groter dan

< Kleiner dan

>= Groter of gelijk aan

<= Kleiner of gelijk aan

< > Ongelijk aan

Query-voorbeelden: SELECT * FROM actor WHERE actor_id <= 5;

SELECT * FROM actor WHERE actor_id < > 5;

SELECT * FROM actor WHERE actor_id >= 5;

SELECT * FROM actor WHERE actor_id < 5;

SELECT * FROM actor WHERE actor_id > 5;

LIKE en % (jokerteken) = Naar een speciale waarde kijken

Query-voorbeeld: Voorbeeld: SELECT * FROM actor WHERE first_name LIKE 'A%';

NOT = iets niet willen zien (bijv. IS NOT NULL, NOT LIKE, NOT BETWEEN)

Query-voorbeeld: SELECT * FROM actor WHERE first_name NOT LIKE 'A%';

AND, OR = Meerdere filters toepassen

Query-voorbeelden: SELECT * FROM actor WHERE first_name NOT LIKE "A%" AND actor_id <= 7;

SELECT * FROM actor WHERE first_name NOT LIKE "A%" OR actor_id <= 7;

BETWEEN (BETWEEN & AND) = Controleert of een attribuut/kolomnaam tussen getallen zit

Query-voorbeeld: SELECT * FROM actor WHERE actor_id BETWEEN 5 AND 10;

Les 3

Tabelnaam

Kolomnaam/Attribuut

Record = data in kolommen

AVERAGE/**AVG** = **Berekent gemiddelde uit** van alle records/data onder een kolomnaam/attribuut.

Query-voorbeeld: **SELECT AVG (length) FROM film;**

NULL = **Laat lege waardes zien** in een kolomnaam/attribuut.

Query-voorbeelden: **SELECT * FROM staff WHERE password IS NULL;**

SELECT * FROM staff WHERE password IS NOT NULL;

UPDATE = **Gegevens in een tabel wijzigen**

Query-voorbeeld: **UPDATE actor SET first_name = "Jan" WHERE actor_id = 5;**

INSERT (INTO) = **Nieuwe rijen toevoegen**

Query-voorbeeld: **INSERT INTO actor (first_name, last_name, last_update) VALUES ("Albert", "Heijn", "2023-1-21 11:16:37")**

(AI = Auto Increment/Primary Key/ID: Als kolomnaam/attribuut dit heeft dan wordt dit kolom automatisch gevuld.)

DELETE = **Verwijdert rijen**

Query-voorbeeld: **DELETE FROM actor WHERE actor_id = 11;**

Een berekende waarde vergelijken met een andere waarde

Je kan door gebruik te maken van haakjes (...) in je WHERE een nieuw SELECT-statement maken.

Query-voorbeeld: **SELECT AVG prijs FROM artikelen WHERE AVG prijs < (SELECT naam FROM prijs WHERE schoen = "Nike");**

Les 4

Tabelnaam

Kolomnaam/Attribuut

Record = data in kolommen

Gegevens uit meerdere tabellen

Kolomnaam/attribuut altijd **voor de tabelnaam zetten**, SQL weet dan uit welke tabel het kolomnaam/attribuut komt

Voorbeeld: `SELECT city.city FROM country.country`

Simpele **INNER JOIN** = **Koppeling tussen 2 tabellen** / Op basis van de overeenkomende sleutels (**Primary Key/Foreign Key**)

Query-voorbeeld: `SELECT city.city, country.country`

`FROM city INNER JOIN country ON city.country_id = country.country_id;`

Complexere **INNER JOIN** = Gegevens die verbonden zijn met een tussentabel met elkaar verbinden

Voorbeeldvraag: Welke acteur speelt in welk film? (Laat zien: acteur voornaam, achternaam, titel film waarin hij speelt)

Query-voorbeeld: `SELECT actor.first_name, actor.last_name, film.title`

`FROM actor INNER JOIN film_actor ON actor.actor_id = film_actor.actor_id`

`INNER JOIN film ON film_actor.film_id = film.film_id;`

(=) Hetzelfde/gelijk aan

Rechten en users: Het aanmaken van users en het toekennen van rechten

- Query: `CREATE USER 'username'@'%' IDENTIFIED BY "PASSWORD";`
Query-voorbeeld: `CREATE USER 'Johan'@'%' IDENTIFIED BY "welkom";`
- Query: `GRANT ALL PRIVILEGES ON databasenaam.tabelnaam TO 'username'@'%' WITH GRANT OPTION;`
- Query-voorbeeld: `GRANT ALL PRIVILEGES ON sakila.actor TO 'Johan'@'%' WITH GRANT OPTION;`
- Query: `FLUSH PRIVILEGES;`

