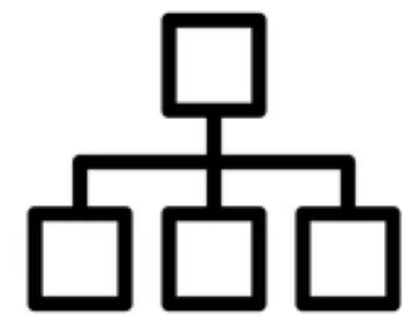


Modelling & Databases

Les 11:

SQL > DQL

SQL > DML



Onderwerpen les:

- SQL DQL intro
- Queries met SELECT en FROM
- WHERE
- DISTINCT
- LIKE
- ORDER BY & GROUP BY & LIMIT
- NULL
- SQL DM intro
- INSERT
- UPDATE
- DELETE

Leerdoelen

- ✓ Aan het einde van de dag kun je simpele queries schrijven met SQL
- ✓ Aan het einde van de dag weet je hoe je order by, group by en limit kunt toepassen
- ✓ Aan het einde van de dag kun je data toevoegen aan een database
- ✓ Aan het einde van de dag kan je data updaten en verwijderen

SQL DQL

Wat is DQL?

Data Query Language

DQL-instructies worden gebruikt voor het uitvoeren van queries op de records in de database.

Een DQL statement bevat de SELECT instructie. De resultaten worden gecompileerd in een tijdelijke tabel die wordt weergegeven of ontvangen door een ander programma.



SQL DQL

SELECT FROM

Met select from kun je alle gegevens van alle klanten tonen, of alleen de voornaam en email adres van alle klanten, of juist alleen de id en telefoonnummer van alle klanten etc.

SELECT

- met SELECT geef je aan welke columns / fields je wil ophalen uit een tabel.
- met een * asterisk geef je aan dat je alle columns wil ophalen.

FROM

- met FROM geef je aan uit welke table je de gegevens wil halen. Bijvoorbeeld de klant table of de product table.

SQL DQL

SELECT FROM

Voorbeelden

haal alle gegevens van alle klanten in de table klant

```
SELECT * FROM klant;
```

haal de naam en adres van alle klanten in de table klant

```
SELECT naam, adres FROM klant;
```

dit kan ook

```
SELECT  
naam,  
adres  
FROM klant;
```

| https://www.w3schools.com/sql/sql_select.asp

SQL DQL

WHERE

Wil je niet alle klanten of alle producten of iets anders uit een bepaalde table? Dan kun je deze filteren met WHERE.

WHERE gebruiken

- met WHERE geef je aan waar een column / field waarde aan moet voldoen
- de column of field moet bijvoorbeeld gelijk zijn aan een waarde. Bv is gelijk aan de string "Londen", of gelijk aan het getal 2
- Je kan hier ook relational operators gebruiken zoals >, <, >=, <=, !=, =
- Geavanceerdere opties gaan we in volgende lessen bespreken

SQL DQL

WHERE

Voorbeelden

haal alle gegevens van alle klanten met plaats Amersfoort in de tabel klant

```
SELECT * FROM klant WHERE plaats = 'Amersfoort';
```

haal naam en adviesprijs op van artikel waar de adviesprijs groter is dan 5

```
SELECT * FROM artikel WHERE adviesprijs > 5;
```

| https://www.w3schools.com/sql/sql_where.asp

SQL DQL

Opdracht

- 1 Zoek het klantnr van klant met de naam Staal uit Amsterdam
- 2 Hoeveel bestellingen heeft Staal gedaan?
- 3 Welke artikelen heeft Staal besteld op 2003-05-04?

SQL DQL

DISTINCT

Met de DISTINCT instructie worden alleen verschillende (distinct) waarden geretourneerd.

Een kolom bevat vaak veel dubbele waarden; en soms wil je alleen de verschillende (distinct) waarden ophalen.

Voorbeeld

haal alleen unieke plaatsen op vanuit de tabel klant

```
SELECT DISTINCT plaats FROM klant;
```

| https://www.w3schools.com/sql/sql_distinct.asp

SQL DQL

LIKE

%	0 of meer karakters
_	1 karakter

Met LIKE zoek je aan de hand van een patroon. Komt de column value overeen met mijn patroon?
Bijvoorbeeld begint de naam met een a? Of eindigt het met een nummer? Zit er een @ in? etc. In de fabriek db zoek je misschien een artikel naam met een naam die je niet meer helemaal weet, maar wel een bepaald woord in de naam.

WHERE name LIKE 'a%' – zoekt alle values die starten met een 'a'

WHERE name LIKE '%a' – zoekt alle values die eindigen met een 'a'

WHERE name LIKE '%o%' – zoekt alle values die een o in zich hebben in any position

WHERE name LIKE '_r%' – zoekt alle values die een r hebben in de 2e positie

WHERE name LIKE 'a__%' – zoekt alle values die een begint met 'a' en minstens 3 karakters heeft

WHERE name LIKE 'a%o' – zoekt alle values die starten met een 'a' en eindigen met een 'o'

SQL DQL

LIKE

Voorbeelden

haal alle gegevens van alle klanten op waar de naam begint met een S

```
SELECT * FROM klant WHERE naam LIKE 'S%';
```

haal alle adressen op van alle klanten op het woord 'straat' in het adres zit

```
SELECT * FROM klant WHERE straat LIKE '%straat%';
```

haal alle producten op met een naam waar de o op de 2e plek zit

```
SELECT * FROM artikel WHERE naam LIKE '_o%';
```

| https://www.w3schools.com/sql/sql_like.asp

SQL DQL

ORDER BY, GROUP BY & LIMIT

Soms is het handig om de gevonden records te ordenen, te groeperen of een max aantal records te tonen. In onze fabriek db wil je misschien alle bestellingen groeperen per fabriek of per klant. Misschien wil je alleen de laatste 3 bestellingen zien of de top 5 duurste artikelen.

ORDER BY

Ordenen kan met één of meerdere kolommen. Bijvoorbeeld bij personen eerst ordenen op achternaam en dan ordenen op voornaam. Dan komt Arend de Boer voor Bas de Boer. Bij het ordenen heb je de default waarde ASC ascending, of DESC descending.

haal naam en adviesprijs op van artikel waar de adviesprijs groter is dan 5

```
SELECT * FROM klant ORDER BY plaats, naam ASC;
```

SQL DQL

ORDER BY, GROUP BY & LIMIT

GROUP BY

Groeperen van items kan ook met één of meerdere kolommen. Misschien wil je bestellingen ophalen en groeperen per fabr of per klant. Daarbij kun je Aggregate Functions gebruiken zoals count, max, min sum etc. Hier leer je later meer over.

haal alle bestellingen op en groepeer deze per fabriek, bekijk dan het aantal klanten dat bij elke fabriek heeft besteld. Zet de fabrieken in descending order.

```
SELECT fabr, count(klantnr) FROM bestelling  
GROUP BY fabr  
ORDER BY fabr DESC;
```

SQL DQL

ORDER BY, GROUP BY & LIMIT

LIMIT & OFFSET

Misschien wil je niet alle gevonden records tonen. Bijvoorbeeld alleen de eerste 10, hiervoor gebruik je LIMIT. Indien je 10 resultaten wil 10, maar niet de eerste 3. Hiervoor gebruik je OFFSET.

Toon de 5 duurste artikelen

```
SELECT * FROM artikel  
ORDER BY adviesprijs DESC  
LIMIT 5;
```

SQL DQL

NULL

Een kolom/veld is NULL wanneer er geen waarde is ingevoerd.

Een kolom/veld met de waarde 0 of een spatie is niet hetzelfde als NULL

Het is niet mogelijk met comparison operators (=, <, >, != etc) een kolom/veld te testen op een NULL waarde. dit doe je met IS NULL of IS NOT NULL

Toon alle artikelen waar geen naam is ingevoerd

```
SELECT * FROM artikel  
WHERE naam IS NULL;
```


SQL DQL

Opdracht

- 1 Haal de 3 klanten op die met hun naam als eerst voorkomen in het alfabet
- 2 Hoeveel high light pennen zijn besteld? (gebruik sum)

SQL DML

Wat is DML?

Data Manipulation Language

DML-instructies worden gebruikt voor het toevoegen, verwijderen en wijzigen van gegevens in een database.

Een DML statemen bevat vaak een INSERT, UPDATE of DELETE instructie.



SQL DML

INSERT INTO

Ondertussen weet je hoe een database, tables en columns/fields aanmaakt. De volgende stap is de database vullen met data/records. Met INSERT INTO geef je aan welke table en welke columns/fields je gaat vullen met data. Vervolgens geef je de values door. Hier kun je 1 maar ook meerdere records meegeven. Bij het invoeren van records geef je geen value voor een surrogaat primary key door. Deze wordt automatisch gegenereerd bij het aanmaken van de record.

Voeg 3 artikelen toe

```
INSERT INTO artikel (artnr, naam, adviesprijs)
VALUES
(201, 'Cartman Bubblehead', 12.50),
(202, 'Bruce Lee Bubblehead', 12.50),
(203, 'Elon Bubblehead', 12.50);
```

| https://www.w3schools.com/sql/sql_insert.asp

SQL DML

UPDATE

Nu er records in onze database staan kan het zijn dat deze in de loop van de tijd gewijzigd gaan worden. Denk aan de prijs van producten, een klant zijn adres wijzigen etc.

UPDATE	- geef aan in welke table je iets wil wijzigen.
SET	- geef aan de nieuwe values en de bijbehorende columns/fields
WHERE	- geef aan welke records je wil wijzigen dit kunnen 1 of meerdere records zijn die voldoen aan deze WHERE zoekopdracht

Verander het adres van mevrouw Staal klantnr 121 naar Atoomweg 1 in Utrecht

```
UPDATE klant  
SET adres = 'Atoomweg 1'  
WHERE klantnr = 121;
```

SQL DML

DELETE

Naast het wijzigen van de records is het ook mogelijk om records te verwijderen.

DELETE FROM – geef aan uit welke table je iets wil verwijderen

WHERE – geef aan welke records je wil verwijderen dit kunnen 1 of meerdere records zijn die voldoen aan deze WHERE zoekopdracht

Voeg een klant toe met klantnr 999 en verwijder deze.

```
DELETE FROM klant  
WHERE klantnr = 999;
```

SQL DQL

Opdracht

In deze casus voeg je data toe aan een lege database voor Bouwmarkt de Plank.

1 Maak een database in pgAdmin. 'bouwmarkt'

2 Maak de tabellen aan met het bestand 'bouwmarkt db'. Zie github of teams

Casus:

In de bouwmarkt kun je alleen winkelen met een pasje. Dit kan een anonieme pas zijn of een pas op naam (NAW gegevens).

Bekijk het sql script 'bouwmarkt db' goed voordat je data toevoegd aan de database.

SQL DQL

Opdracht

Voeg de volgende data toe:

Klanten:

Clara vraagt een pas aan maar wil anoniem blijven zij krijgt klantnummer 432210.

Kees vraagt een pas aan en vult zijn NAW gegevens in: Kees, Kerkstraat 12, Groningen met klantnummer 412345.

Pieter vraagt een pas aan met NAW: Pieter, Bergpad 5, Maastricht met klantnummer 543214

Filialen:

Filiaalnr 12, Amsterdam aan de Buitenweg 1

Filiaalnr 24, Utrecht aan de Atoomweg 41

Filiaalnr 36, Groningen aan de Burgerstraat 90

SQL DQL

Opdracht

Voeg de volgende data toe:

Producten:

Product 1, plank, 6.99
Product 2, spijkers, 2.69
Product 3, schroeven, 3.35
Product 4, hamer, 2.99
Product 5, boormachine, 24.49

Transactie en aankopen:

Op 1 November 2022 om 15:45 doet Clara (anonieme klant) een transactie in het Filiaal in Amsterdam. Zij heeft gekocht 5 planken, 1 spijkers en 1 hamer.

SQL DQL

Opdracht

Voeg de volgende data toe:

Transactie en aankopen:

Op 3 December 2022 om 12:25 doet Pieter een transactie in het filiaal in Utrecht. Hij koopt 12 planken 2 x schroeven, en 1 boormachine.

Op 6 December 2022 om 10:30 doet Kees een transactie in het filiaal in Groningen. Hij koopt 2 planken.

Kees komt thuis en komt erachter dat hij geen schroeven heeft. Hij doet weer een transactie op dezelfde dag om 12:30 en koopt 1 setje schroeven.