SQL

1 Introductie phpMyAdmin en SELECT

datasource

? Leerdoelen

- Je weet hoe je XAMPP en phpMyAdmin gebruikt.
- Je kunt een database importeren.
- Je weet wat SQL is.
- Je kunt gegevens ophalen met `SELECT` en `FROM`.

? Uitleg

In deze eerste opdracht werk je met een database met informatie over films. Je gebruikt de `SELECT` -instructie om gegevens op te halen uit de tabel `movies`.

Je voert verschillende queries uit om de inhoud van de tabel te verkennen.

Wat is SQL precies?

SQL staat voor **Structured Query Language**. Het is een programmeertaal die speciaal is ontworpen om te "praten" met databases. Een database is eigenlijk gewoon een heel goed georganiseerde verzameling van gegevens. Denk bijvoorbeeld aan de databases van:

- **Instagram:** met alle gebruikers, foto's, likes en reacties.
- **Fortnite:** met alle spelers, hun skins, V-Bucks en statistieken.
- **Een webshop:** met alle producten, prijzen en voorraad.

Met **SQL** kun je deze databases vragen stellen (queries) of opdrachten geven.

?Video - SQL introductie

https://www.youtube.com/watch?v=zpnHsWOy0RY

https://www.youtube.com/embed/zpnHsWOy0RY

Wat heb je nodig?

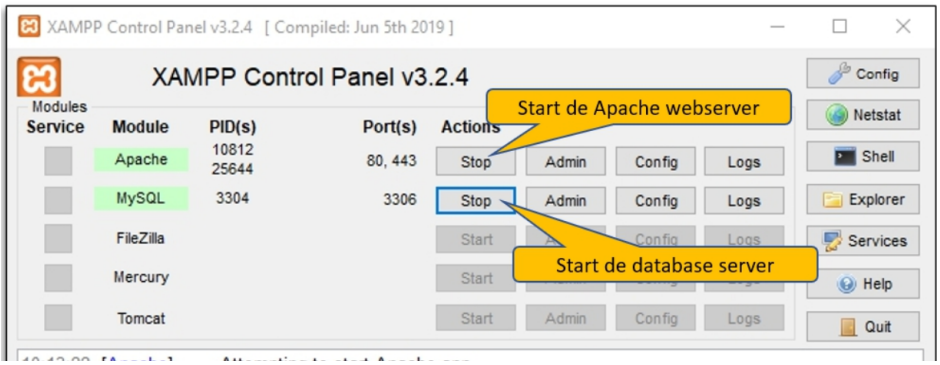
- XAMPP (Apache en MySQL moeten aan staan)
- De database `mdb_movies.sql`sql

XAMPP en phpMyAdmin

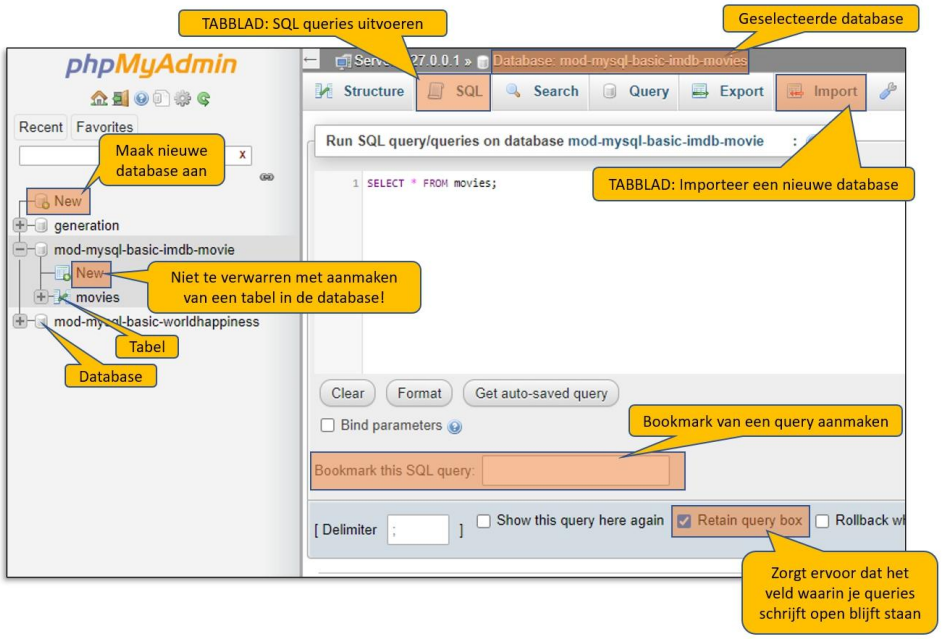
We gaan gebruik maken van phpMyAdmin.

phpMyAdmin is eenonderdeel van XAMPP en wordt veel gebruikt om met datbases te werken. Je kan databases aanmaken, verwijderen, aanpassen en inzien.

Je kunt phpMyAdmin pas opstarten als je XAMPP goed draait; Apache en mySQL staan aan.



Start localhost/phpmyadmin



Stappen om de database te importeren:

1. Start Apache en MySQL via het XAMPP Control Panel.
2. Ga naar `http://localhost/phpmyadmin` in je browser.
3. Maak een nieuwe database aan met de naam `imdb_movies`.
4. Selecteer de database en gebruik het tabblad *Import* om het bestand `imdb_movies.sql` te importeren.

Kom je er niet uit: hier staat met plaatjes uitgelegd hoe je een database importeert.

?? Opdracht

1. Je hebt de database imdb_movies.sql geimporteerd
2. Voer de volgende drie SQL-query's uit op de tabel `movies`:
   - `SELECT * FROM movies;`
   - `SELECT movie FROM movies;`
   - `SELECT movie, rating FROM movies;`

   Heb je meer uitleg nodig over hoe je query maakt, hier staat een voorbeeld met plaatjes)
3. Bekijk het resultaat van elke query. Wat valt je op?

? Reflectie

- Welke kolommen lijken jou het nuttigst als je een lijst met filmaanbevelingen zou maken?
- Wat is het verschil tussen `SELECT *` en `SELECT kolomnaam`?

? Inleveren

1. Maak een screenshot waarbij je laat zien dat je een query hebt uitgevoerd op de database imdb_movies.
2. Beantwoord de refelctievragen en lever die in (txt of pdf).

2 WHERE en logica

? Leerdoelen

- Je kunt gegevens filteren met `WHERE`.
- Je begrijpt het gebruik van logische operatoren zoals `=`, `>`, `<`, `AND`, en `OR`.
- Je kunt AI gebruiken om een query te genereren en deze zelf controleren en verbeteren.

? Uitleg

Met `SELECT` haal je gegevens op. Met `WHERE` kun je die gegevens filteren. Bijvoorbeeld: alleen landen met een hoge geluksindex, of alleen landen uit Europa.

We gebruiken de database mod-mysql-basic-worldhappiness.sql. Deze bevat o.a. tabellen `jaar2015` en `jaar2016` met kolommen als `country`, `region`, `rank` en `score`.

Voorbeelden:

```
SELECT * FROM jaar2016 WHERE score > 7000;
SELECT country, region FROM jaar2015 WHERE region = "Western Europe";
SELECT country, score FROM jaar2016 WHERE region = "Western Europe" AND score > 7300;
```

?? Opdracht

1. Importeer de database mod-mysql-basic-worldhappiness.sql in phpMyAdmin en selecteer de database `worldhappiness`.
2. Voer de volgende query's uit en controleer het resultaat:
   - Selecteer alle landen uit de tabel `jaar2015`.
   - Selecteer alleen `country` en `score` uit `jaar2016`.
   - Selecteer alle regio's uit 2015.
   - Selecteer alle scores hoger dan 7200 in 2016.
   - Selecteer landen uit de regio "Western Europe" in 2015.
3. Gebruik ChatGPT om een extra query te genereren waarbij je gebruik maakt van `OR`. Maak hierbij gebruik van de database `worldhappiness`

? Reflectie

- Welke query vond je het lastigst en waarom?
- Welke filters heb je gebruikt? Noem minstens twee logische operatoren.
- Wat heeft AI (ChatGPT) goed gedaan, en wat moest je zelf aanpassen?

? Inleveren

- Lever de een screenshot van de zelf gemaakte query (opdracht, stap 3) in, laat zien dat die werkt en leg uit in eigen woorden hoe die werkt.

3 Aggregatiefuncties

? Leerdoelen

- Je kunt gebruik maken van SQL-functies zoals `COUNT()`, `AVG()`, `SUM()`, `MIN()` en `MAX()`.
- Je kunt kolommen hernoemen met `AS` (alias).
- Je begrijpt het verschil tussen `SELECT` van rijen en het samenvatten van gegevens.

? Uitleg

Aggregatiefuncties worden gebruikt om samenvattingen te maken van gegevens in een kolom. Ze voeren een berekening uit op meerdere rijen tegelijk in plaats van één rij.

Belangrijkste functies:

Functie	Doel	Voorbeeld
`COUNT()`	Telt hoeveel rijen er zijn	`SELECT COUNT(*) FROM players;`
`AVG()`	Geeft het gemiddelde van een kolom met getallen	`SELECT AVG(wage) FROM players;`
`SUM()`	Telt alle waarden in een kolom bij elkaar op	`SELECT SUM(value) FROM players;`
`MIN()`	Laat de kleinste waarde zien	`SELECT MIN(age) FROM players;`
`MAX()`	Laat de grootste waarde zien	`SELECT MAX(value) FROM players;`

Alias gebruiken met `AS`

Je kunt je resultaatkolom een duidelijke naam geven met het sleutelwoord `AS`.

```
SELECT AVG(wage) AS gemiddeld_loon FROM players;
```

Bonus: afronden met `ROUND()`

```
SELECT ROUND(AVG(wage)) AS gemiddeld_loon_afgerond FROM players;
```

?? Opdracht

Gebruik de database mod-mysql-basic-fifa2018.sql

1. Voer de volgende queries uit in phpMyAdmin:
   - Toon het gemiddelde loon van alle spelers bij Ajax.
   - Toon de totale waarde van spelers onder de 20 jaar.
   - Toon het hoogste loon van een speler bij FC Utrecht.
   - Toon het aantal spelers uit Nederland.
   - Toon het gemiddelde loon van alle Braziliaanse spelers, afgerond op hele euro's.
2. Gebruik bij elke query een duidelijke **alias** via `AS`.

? Reflectie

- Wat is het voordeel van een samenvattende query (zoals `AVG()`) in plaats van het handmatig bekijken van individuele rijen?
- Welke query vond je het lastigst en waarom?

? Inleveren

- Lever een .txt-bestand in met alle 5 query's én de antwoorden.

Vergeet de aliasen niet!

4 DELETE en veiligheid

? Leerdoelen

- Je begrijpt het doel van een `DELETE`-statement.
- Je kunt een `DELETE`-statement schrijven met een `WHERE`-clausule.
- Je weet waarom een `WHERE`-clausule cruciaal is bij verwijderen van gegevens.

? Uitleg

Met SQL kun je niet alleen gegevens opvragen, maar ook verwijderen. Dat doe je met het `DELETE`-statement. Hierbij is het essentieel dat je altijd een `WHERE`-clausule gebruikt. Als je dat niet doet, verwijder je álle rijen in de tabel!

Voorbeeld:

```
DELETE FROM players WHERE name = "K. Huntelaar";
```

Deze query verwijdert alleen de speler met die naam.

Fout voorbeeld (NIET DOEN!):

```
DELETE FROM players;
```

Deze query verwijdert **alle spelers** uit de tabel. Dit is onherstelbaar.

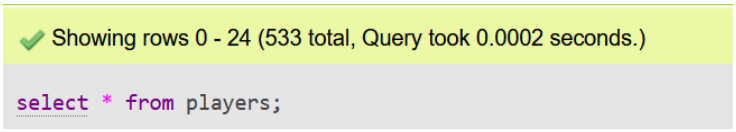
?? Opdracht

Gebruik de database mod-mysql-basic-fifa2018.sql (zoals in de vorige opdracht).

1. Voer de volgende opdrachten uit met een `DELETE`-statement:
   - Verwijder de speler "David Silva".
   - Verwijder alle spelers van de club "Willem II".
   - Verwijder alle Braziliaanse spelers die ouder zijn dan 34.
   - Verwijder alle spelers waarvan het loon (=wage) hoger is dan 200000.
2. Gebruik daarna de volgende query om te controleren hoeveel spelers er nog zijn:

   ```
   SELECT COUNT(*) FROM players;
   ```

   Het antwoord dat hier uit moet komen als je alles goed hebt gedaan is **533**

> TIP: maak eerst een select en als je de juiste resultaten terug krijgt vervang dan de **select *** in **delete**

Mocht het niet goed gaan dan kun je de database altijd opnieuw importeren en opnieuw beginnen.

? Reflectie

- Wat had er fout kunnen gaan als je de `WHERE`-clausule was vergeten?
- Waarom is het handig om altijd eerst een `SELECT` met dezelfde `WHERE`-clausule te doen voordat je `DELETE` uitvoert?

? Inleveren

- Voeg een screenshot toe van de output van je `SELECT COUNT(*)`-query.
- antwoord op de reflectievragen (txt of pdf)

5 Introductie tot JOINs

? Leerdoelen

- Je begrijpt het nut van tabellen koppelen met een `JOIN`.
- Je weet wat een `PRIMARY KEY` en `FOREIGN KEY` zijn.
- Je kunt een eenvoudige `INNER JOIN` uitvoeren.

? Uitleg

In een echte database worden gegevens vaak verdeeld over meerdere tabellen. Je gebruikt een **JOIN** om die tabellen aan elkaar te koppelen. Zo kun je bijvoorbeeld zien in welke klas een student zit en wie zijn studieloopbaanbegeleider is.

Wat is een `JOIN`?

Een JOIN combineert rijen uit twee tabellen op basis van een kolom die ze gemeenschappelijk hebben. Dit is vaak een id-veld zoals `klas_id`.

Voorbeeld:

Stel: we hebben twee tabellen:

```
studenten
+----+----------+-----------+----------+
| id | voornaam | achternaam| klas_id  |
+----+----------+-----------+----------+
| 1  | Fatima   | Bakker    | 101      |
| 2  | Noah     | de Vries  | 101      |
| 3  | Aziz     | Bouali    | 102      |


klassen
+------+----------+------------------+
| id   | klas_naam| studie_coach     |
+------+----------+------------------+
| 101  | SD1A     | meneer Willems   |
| 102  | SD1B     | mevrouw Jansen   |
```

Vraag: "*wie is de studiecoach van Fatima*?"
Antwoord: "*Meneer Willems*".
Klopt dat?

Met een JOIN kun je voor elke student zien in welke klas hij/zij zit én wie zijn of haar studiecoach is:

```
SELECT studenten.voornaam, studenten.achternaam, klassen.klas_naam, klassen.studie_coach
FROM studenten
INNER JOIN klassen ON studenten.klas_id = klassen.id;
```

De tabel **studenten** wordt verbonden met **klassen** waarbij de **primary key** (id) van **klassen** wordt verbonden met de **foreign key** (klas_id) van **studenten**

?? Opdracht

Maak een nieuwe database aan

- Open phpMyAdmin en maak een database aan met de naam `join_oefening`.

Maak deze twee tabellen aan

Maak een database en voer importeer deze database

```
CREATE TABLE studenten (
  id INT AUTO_INCREMENT PRIMARY KEY,
  voornaam VARCHAR(50),
  achternaam VARCHAR(50),
  klas_id INT
);

CREATE TABLE klassen (
  id INT AUTO_INCREMENT PRIMARY KEY,
  klas_naam VARCHAR(50),
  aantal_leerlingen INT,
  studie_coach VARCHAR(100)
);
INSERT INTO klassen (klas_naam, aantal_leerlingen, studie_coach) VALUES
('SD1A', 24, 'meneer Willems'),
('SD1B', 22, 'mevrouw Jansen'),
('SD1C', 25, 'meneer El Idrissi');

INSERT INTO studenten (voornaam, achternaam, klas_id) VALUES
('Fatima', 'Bakker', 1),
('Noah', 'de Vries', 1),
```

```
('Aziz', 'Bouali', 2),
('Eva', 'Peeters', 2),
('Liam', 'Meijer', 1),
('Sophie', 'van der Laan', 2),
('Daan', 'Mulder', 3),
('Aisha', 'Khan', 3),
('Javi','Alvarez',4);
```

Schrijf een JOIN-query

1. Voer deze query uit:

   ```
   SELECT studenten.voornaam, studenten.achternaam, klassen.klas_naam,
   klassen.studie_coach
   FROM studenten
   INNER JOIN klassen ON studenten.klas_id = klassen.id;
   ```

2. Pas de query aan zodat je alleen studenten uit klas `SD1A` toont.

AI - ChatGPT

Als we alle studenten laten zien, dan zien we 8 studenten. In de database staan 9 studenten en als we goed kijken dan wordt (de laatste) student "Javi Alvarez" niet getoond. Hoe komt dat?

- Vraag aan ChatGPT hoe dit zit (tip: het heeft met de soort join te maken)
- Vraag aan ChatGPT om een JOIN-query te schrijven die alle studenten toont, dus ook Javi Alvarez
- Test of de AI-query werkt. Verbeter hem indien nodig.

? Reflectie

- Leg uit waarom Javi Alvarez eerst niet werd getoond
- Wat heb je gedaag om Javi Alvarez wel te tonen, leg uit.
- Beschrijf het verschil tussen de twee soorten joins die je hebt gebruikt.

? Inleveren

- Lever je reflectie in als .txt of .pdf, beschrijf hierin goed wat je hebt gedaan en beantwoord de refelctievragen in eigen woorden.
- Lever de Chat geschiedenis in: wat heb je precies gevraagd en wat was het antwoord (txt of pdf bestand).

6 AI en eigen query's

? Leerdoelen

- Je kunt zelfstandig SQL-vragen formuleren en uitvoeren.
- Je kunt AI gebruiken om een query te genereren, verbeteren en controleren.
- Je begrijpt hoe je informatie uit meerdere tabellen combineert.

? Uitleg

De database die we in deze opdracht gaat gebruiken heeft twee tabellen:

- `student` – bevat gegevens van studenten (zoals naam en inschrijfdatum)
- `progress` – bevat studieresultaten gekoppeld aan studenten

Verzin zelf een logische naam voro deze database en maak een lege database aan.

Importeer daarna de data met dit bestand: student-progress.sql

De tabellen zijn gekoppeld via `student_id`. Je kunt ze combineren met een `JOIN`.

Voorbeeld JOIN:

```
SELECT student.first_name, student.last_name, progress.subject_name, progress.grade_percentage
FROM student
JOIN progress ON student.student_id = progress.student_id;
```

?? Opdracht A

Gebruik AI / ChatGPT

1. Toon alle studenten met hun volledige naam en inschrijfdatum.

2. Toon alle vakken (`subject_name`) en behaalde percentages (`grade_percentage`) van student met voornaam "Aaron".
3. Toon per van start_jaar 2024 het hoogste cijfer dat is behaald voor C++. Gebruik `MAX()`
4. Toon alle studenten die in blok "276c8c" zitten én een cijfer boven de 90% hebben behaald.
   (er zouden 7 studenten moeten worden getoond)

? AI-opdracht B

Laat ChatGPT een query voor je schrijven die de volgende vraag beantwoordt (of verzin zelf een goede vraag):

- *Welk vak heeft gemiddeld over alle jaren de hoogste score?*

Test de gegenereerde query in phpMyAdmin. Als de query niet werkt, probeer dan samen met ChatGPT te achterhalen wat er fout gaat. Corrigeer en leg uit wat je hebt aangepast.

? AI-opdracht C

Kan je samen met AI de volgende output maken. In deze tabel staan per jaar de gemiddelden per vak en in de laatste kolom staat het gemiddelde over alle jaren.

←T→		subject_name 1	gemiddeld_2022	gemiddeld_2023	gemiddeld_2024	gemiddeld_totaal
☐	Edit Copy Delete	Art History	82.1	79.4	79.1	79.0
☐	Edit Copy Delete	Biology	74.3	79.5	77.0	76.8
☐	Edit Copy Delete	C++	80.9	76.4	78.0	77.8
☐	Edit Copy Delete	Calculus	77.7	78.1	74.8	77.9
☐	Edit Copy Delete	Chemistry	77.4	75.0	83.0	77.0
☐	Edit Copy Delete	Databases	69.3	78.2	80.3	76.8
☐	Edit Copy Delete	English Literature	75.7	75.6	78.8	78.0
☐	Edit Copy Delete	Geography	77.7	78.7	79.8	79.4

? Reflectie

- Welke query vond je het moeilijkst om te maken en waarom?
- Beschrijf welk SQL commando je door de AI opdrachten hebt bijgeleerd en beschrijf in eigne woorden wat dit commando doet.?

? Inleveren

- Lever een .txt-bestand in met alle 7 query’s uit deze opdracht (Opdracht A, B en CI).
- Voeg een screenshot toe van het resultaat van de moeilijkste query.
- Lever je reflectie in als .txt of .pdf.

? SQL Terugblik en Samenvatting

? Leerdoelen

- Je kunt de belangrijkste SQL-onderdelen die je hebt geleerd opsommen en uitleggen.
- Je kunt per onderdeel een voorbeeldquery schrijven.
- Je herkent waar je zelf nog onzekerheden of fouten maakt.

?? Opdracht

Maak een overzicht waarin je de belangrijkste SQL-onderdelen samenvat die je dit blok hebt geleerd.

1. Voor elk SQL-onderdeel of keyword geef je:
   - De naam (bijv. `SELECT`)
   - Een korte beschrijving in je eigen woorden
   - Een kort voorbeeld (één regel SQL is genoeg)
2. Werk dit netjes uit in een tabel of lijst. Gebruik minstens 10 begrippen, zoals:
   - `SELECT`, `FROM`, `WHERE`, `JOIN`, `AVG()`, `GROUP BY`, `ORDER BY`, `DELETE`, `AS`, `COUNT()`
3. Gebruik de termen table, row en column (table, rij en colum).
4. Je mag AI gebruiken om voorbeelden te controleren, maar de uitleg moet in jouw eigen woorden zijn.

? Voorbeeld (fragment)

SQL-onderdeel	Omschrijving (in eigen woorden)	Voorbeeldquery
`SELECT`	Gebruik je om aan te geven welke **kolommen** je wilt zien.	`SELECT naam FROM studenten;`

SQL-onderdeel	Omschrijving (in eigen woorden)	Voorbeeldquery
`WHERE`	Gebruik je om **rijen** te filteren op een voorwaarde.	`SELECT * FROM progress WHERE grade_percentage > 80;`

? Inleveren

- Werk de opdracht uit, maak een PDF en lever die in.

Fatima bestaat niet -> Aaron

Revision #27
Created 2025-06-11 07:19:26 UTC by Max
Updated 2026-07-10 09:05:08 UTC by Max