

LU02 - Daten mit SQL abfragen

Lernziele

Nach dieser Lerneinheit können Sie:

- **Einfache Abfragen:** mit dem SQL-Befehl *SELECT* alle Spalten oder nur bestimmte Spalten aus einer Tabelle anzeigen.
- **Sortieren von Daten:** mit *ORDER BY* die Ergebnisse sortieren (aufsteigend oder absteigend).
- **Filtern mit Bedingungen:** mit *WHERE* Daten filtern – sowohl nach Zahlen als auch nach Text, inklusive der richtigen Anführungszeichen bei Wörtern oder Texten.
- **Mehrere Bedingungen kombinieren:** Ich kann mit *AND*, *OR* und *NOT* mehrere Bedingungen verknüpfen.
- **Richtig starten:** Ich weiss, dass ich in WebStorm zuerst das richtige Schema auswählen muss, bevor ich Abfragen mache.

Einleitung: Was ist SQL?

SQL (Structured Query Language) ist die Sprache, mit der wir mit einer relationalen Datenbank «kommunizieren»: Wir schicken ihr eine Anfrage, sie schickt uns die passenden Daten zurück.

Mit SQL können wir:

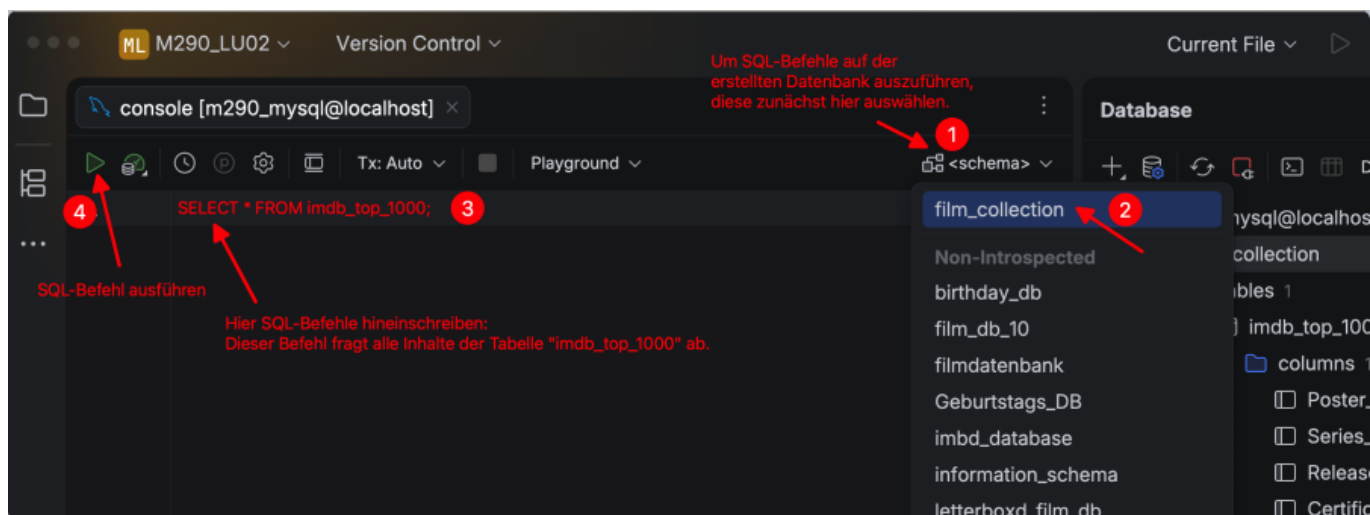
- Daten **auslesen** (abfragen)
- Daten **einfügen, ändern oder löschen**
- die **Struktur** einer Datenbank erstellen (z.B. neue Tabellen)

Die Syntax orientiert sich stark an der englischen Sprache und ist deshalb vergleichsweise leicht lesbar – auch ohne Programmiererfahrung.

Im Begriff **SQL** steckt das englische Wort *query* (= Abfrage). Mit dem Befehl *SELECT* können Daten abgefragt werden.

Bevor Sie loslegen: Schema auswählen

Bevor Sie in der Query-Console von WebStorm eine Abfrage ausführen können, müssen Sie zuerst das **Schema** (also Ihre Datenbank, z.B. ``film_collection``) auswählen. Sonst weiss MySQL nicht, in welcher Datenbank gesucht werden soll, und Sie erhalten eine Fehlermeldung.



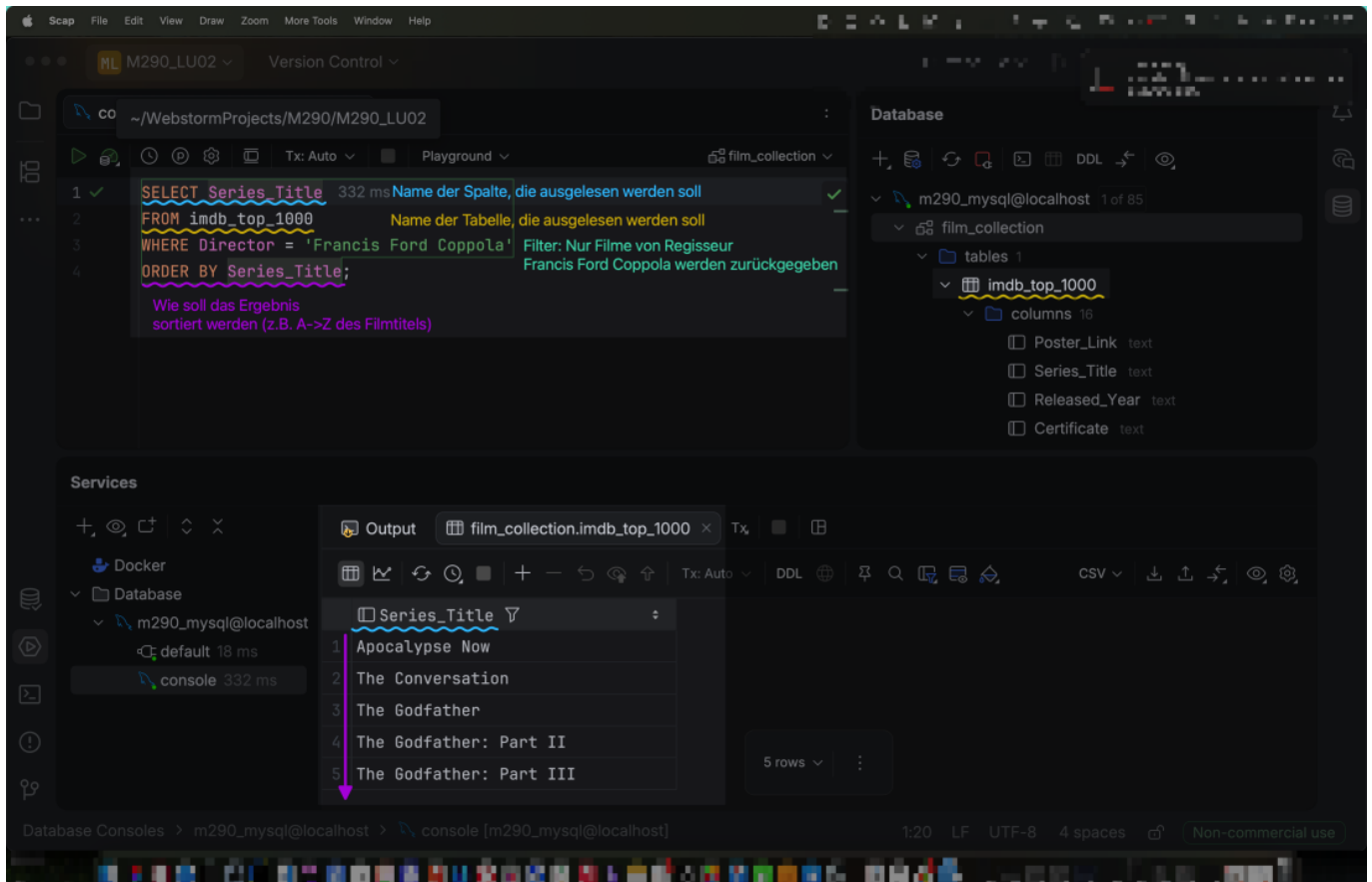
Daten abfragen

Das *SELECT*-Statement ist das am häufigsten verwendete SQL-Statement. Mit

```
SELECT ... FROM ... WHERE
```

können Daten in verschiedenen Varianten ausgegeben werden:

- alle Spalten
- bestimmte Spalten
- mit Filterbedingungen
- sortiert



Alle Spalten ausgeben

Die einfachste Abfrageform ist:

```
SELECT * FROM imdb_top_1000;
```

Die **FROM-Anweisung** sagt, in welcher Tabelle gesucht werden soll.

Dies gibt **alle Spalten** und **alle Zeilen** der Tabelle zurück.

Beispielauszug (die tatsächliche Tabelle hat noch mehr Spalten):

Poster	Filmtitel	Jahr	Genre	Rating	Kurzbeschreibung
	The Godfather	1972	Crime, Drama	9.2	An organized crime dynasty's aging patriarch transfers control of his clandestine empire to his reluctant son.
	Interstellar	2014	Adventure, Drama, Sci-Fi	8.6	A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival.
	The Green Mile	1999	Crime, Drama, Fantasy	8.6	The lives of guards on Death Row are affected by one of their charges: a black man accused of child murder and rape, yet who has a mysterious gift.

Poster	Filmtitel	Jahr	Genre	Rating	Kurzbeschreibung
	Whiplash	2014	Drama, Music	8.5	A promising young drummer enrolls at a cut-throat music conservatory where his dreams of greatness are mentored by a strict instructor.

Bestimmte Spalten ausgeben

Wenn nur bestimmte Daten benötigt werden, empfiehlt es sich, die Spalten **explizit** anzugeben. Die Spalten werden mit **Kommas (,)** getrennt:

```
SELECT Series_Title, Released_Year, IMDB_Rating
FROM imdb_top_1000;
```

Beispielauszug:

Filmtitel	Jahr	Bewertung
The Godfather	1972	9.2
Interstellar	2014	8.6
The Green Mile	1999	8.6
Whiplash	2014	8.5

Ausgabe sortieren

Mit *ORDER BY* können Ergebnisse sortiert werden.

```
-- Die Filme nach Erscheinungsjahr sortieren
SELECT Series_Title, Released_Year, IMDB_Rating
FROM imdb_top_1000
ORDER BY Released_Year ASC;
```

Mit *ASC* (engl. ascending = aufsteigend) oder *DESC* (engl. descending = absteigend) kann die Sortierung bestimmt werden.

Die besten Filme zuerst:

```
SELECT Series_Title, Released_Year, IMDB_Rating
FROM imdb_top_1000
ORDER BY IMDB_Rating DESC;
```

Daten filtern (WHERE)

Nicht immer sind alle Daten interessant – mit **WHERE** können Bedingungen definiert werden. Während die bisher behandelten *SELECT*- und *FROM*-Anweisungen in jedem *SELECT*-Befehl vorkommen müssen, ist die **WHERE-Anweisung optional**. Wegen ihrer wichtigen Bedeutung zur Einschränkung (Restriktion) einer Tabelle auf bestimmte Zeilen kommt die *WHERE*-Anweisung in der Praxis trotzdem in fast allen *SELECT*-Befehlen vor. Die Struktur einer Abfrage mit Bedingung lautet in Worten:

- **Was?** (welche **Spalten** werden ausgewählt → z.B. *SELECT Series_Title*)
- **Woher?** (aus welcher/n **Tabelle/n** → z.B. *FROM imdb_top_1000*)
- Unter **welcher/n Bedingung/en?** (welche **Feldwerte** → z.B. *WHERE IMDB_Rating > 8.5*)

```
-- Alle Filme mit Bewertung über 8.5
SELECT Series_Title, IMDB_Rating
FROM imdb_top_1000
WHERE IMDB_Rating > 8.5;
```

Ergebnis (Auszug):

Filmtitel	Bewertung
The Godfather	9.2
The Green Mile	8.6
Interstellar	8.6

Zahlen vs. Text in der WHERE-Klausel: Zahlenwerte (wie 8.5) schreibt man ohne Anführungszeichen. Text (sogenannte *Strings*) muss dagegen immer in **einfachen Anführungszeichen** ('text') stehen:

```
WHERE IMDB_Rating > 8.5          -- Zahl: keine
Anführungszeichen
WHERE Director = 'Christopher Nolan' -- Text: einfache
Anführungszeichen
```

Vergessen Sie die Anführungszeichen bei Text, meldet MySQL einen Fehler.

Operatoren

Im *WHERE*-Statement können verschiedene Operatoren eingesetzt werden, um die Auswahl von Datensätzen einzuschränken.

Vergleichsoperatoren

```
= , < , > , <= , >= , <>
```

→ werden vor allem bei Zahlen verwendet. Beispiel: nur Filme anzeigen, die **vor 1970** erschienen sind.

```
SELECT Series_Title, Released_Year
FROM imdb_top_1000
WHERE Released_Year < 1970;
```

Mehrere Bedingungen kombinieren (AND, OR, NOT)

AND, OR, NOT

→ verbinden mehrere Bedingungen.

Filme von Regisseuren *Christopher Nolan* **oder** *Francis Ford Coppola*:

```
SELECT Series_Title, Director
FROM imdb_top_1000
WHERE Director = 'Christopher Nolan' OR Director = 'Francis Ford Coppola';
```

Mit *AND* kann man Bedingungen kombinieren, z.B.:

```
-- Alle Drama-Filme von Christopher Nolan
SELECT Series_Title, Director, Genre
FROM imdb_top_1000
WHERE Director = 'Christopher Nolan' AND Genre = 'Drama';
```

Mit *NOT* kehrt man eine Bedingung um:

```
-- Alle Filme, die NICHT von Christopher Nolan sind
SELECT Series_Title, Director
FROM imdb_top_1000
WHERE NOT Director = 'Christopher Nolan';
```

Zusammenfassung

- **SELECT**
 - bestimmt die Spalten

- **FROM**

- bestimmt die Tabelle

- **WHERE**

- filtert die Zeilen (Text in 'Single Anführungszeichen', Zahlen ohne)

- **ORDER BY**

- sortiert die Ergebnisse

- **AND / OR / NOT**

- kombiniert mehrere Bedingungen

Mit diesen Befehlen sind die wichtigsten Grundlagen für Abfragen in relationalen Datenbanken gelegt.

 Guido Koch

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

https://wiki.bzz.ch/de/modul/m290_guko/learningunits/lu02/theorie/a_sql_basics

Last update: **2026/08/24 07:41**

