

LU02b - Daten mit SQL abfragen

Daten abfragen

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

SELECT

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;
```

Abfragen filtern

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 seiner 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

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;
```

Logische Verknüpfungen

AND, OR, NOT

→ verbinden mehrere Bedingungen. Beispiel: 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 Sci-Fi-Filme von Christopher Nolan  
SELECT Series_Title, Director, Genre  
FROM imdb_top_1000  
WHERE Director = 'Christopher Nolan' AND Genre = 'Drama';
```

IN-Operator

IN, NOT IN

→ vereinfacht Abfragen mit vielen Bedingungen.

```
-- Filme von Nolan oder Coppola
SELECT Series_Title, Director
FROM imdb_top_1000
WHERE Director IN ('Christopher Nolan', 'Francis Ford
Coppola');
```

Das Gegenteil mit *NOT IN*:

```
-- Alle Filme, die NICHT von Nolan oder Coppola sind
SELECT Series_Title, Director
FROM imdb_top_1000
WHERE Director NOT IN ('Christopher Nolan', 'Francis Ford
Coppola');
```

Zusammenfassung

- **SELECT**

- bestimmt die Spalten

- **FROM**

- bestimmt die Tabelle

- **WHERE**

- filtert die Zeilen

- **ORDER BY**

- sortiert die Ergebnisse

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

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

Permanent link:
https://wiki.bzz.ch/modul/m290_guko/learningunits/lu02/theorie/b_sql_select

Last update: **2025/08/31 11:27**



Last

update:

2025/08/31 modul:m290_guko:learningunits:lu02:theorie:b_sql_select https://wiki.bzz.ch/modul/m290_guko/learningunits/lu02/theorie/b_sql_select

11:27
