

# LU06 - Vertiefte SQL-Abfragen

In **LU02** haben Sie gelernt, wie Sie mit

- *SELECT* Spalten auswählen
- *FROM* die Tabelle bestimmen
- *WHERE* Bedingungen setzen
- *ORDER BY* die Ergebnisse sortieren

Daten aus einer Tabelle abfragen können.

In dieser Lektion gehen wir einen Schritt weiter: Wir schauen uns an, **wie Bedingungen in WHERE noch mächtiger kombiniert werden können**, wie Sie mit *LIMIT* die Ergebnismenge einschränken und mit *DISTINCT* doppelte Werte vermeiden können.

## Vergleichsoperatoren

Die **einfachsten Bedingungen** vergleichen Werte mit den Operatoren:

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

```
SELECT Series_Title, Released_Year
FROM imdb_top_1000
WHERE Series_Title = 'Interstellar';
```

Beachten Sie: Wir haben hier nach einem Wert vom Typ **Text** gesucht. Texte sind **alphanumerische Zeichen** und müssen immer in **Anführungszeichen** stehen:

- „Interstellar“ oder 'Interstellar'
- Ohne Anführungszeichen versteht MySQL das Wort als **Namen einer Tabelle oder Spalte**.

Beispiele:

```
-- Filme vor 1970
SELECT Series_Title, Released_Year
FROM imdb_top_1000
WHERE Released_Year < 1970;
```

```
-- Filme mit Bewertung über 9.0
SELECT Series_Title, IMDB_Rating
FROM imdb_top_1000
WHERE IMDB_Rating > 9.0;
```

## BETWEEN

Prüfen, ob ein Wert zwischen zwei Grenzen liegt.

```
-- Filme aus den 90ern
SELECT Series_Title, Released_Year
FROM imdb_top_1000
WHERE Released_Year BETWEEN 1990 AND 1999;
```

Achtung: Der Operator *BETWEEN* schliesst die beiden Grenzen **ein** (also inkl. 1990 und 1999). Die folgenden beiden Bedingungen sind also **gleichwertig**:

```
Released_Year BETWEEN 1990 AND 1999
```

```
Released_Year >= 1990 AND Released_Year <= 1999
```

## IN

Prüfen, ob ein Wert in einer **Liste** vorkommt.

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

## LIKE

Mit *LIKE* können Sie nach **Textmustern** suchen.

- % = beliebig viele Zeichen
- \_ = genau ein Zeichen

```
-- Alle Filme, die mit "The" beginnen
SELECT Series_Title
FROM imdb_top_1000
WHERE Series_Title LIKE 'The%';
```

```
-- Alle Filme, die "Star" im Titel enthalten
SELECT Series_Title
FROM imdb_top_1000
```

```
WHERE Series_Title LIKE '%Star%';
```

## IS NULL / IS NOT NULL

Der Wert **NULL** (sprich: *null*, englisch ausgesprochen) bedeutet in SQL: → Es ist **kein Wert vorhanden**.

Das heisst, dass in dem entsprechenden Feld nichts in der Tabelle eingetragen ist. Das ist etwas anderes als **0** (Zahl) oder ein leerer Text "".

Beispiel in unserer Filmdatenbank: Manche Filme haben **keinen Metascore**<sup>1)</sup>.

```
-- Filme ohne Metascore
SELECT Series_Title, Meta_score
FROM imdb_top_1000
WHERE Meta_score IS NULL;
```

```
-- Filme mit Metascore
SELECT Series_Title, Meta_score
FROM imdb_top_1000
WHERE Meta_score IS NOT NULL;
```

## Logische Verknüpfungen (AND / OR / NOT)

Mehrere Bedingungen können kombiniert werden:

```
-- Alle Sci-Fi-Filme von Christopher Nolan
SELECT Series_Title, Genre, Director
FROM imdb_top_1000
WHERE Director = 'Christopher Nolan'
AND Genre LIKE '%Sci-Fi%';
```

```
-- Filme von Nolan oder Coppola
SELECT Series_Title, Director
FROM imdb_top_1000
WHERE Director = 'Christopher Nolan'
OR Director = 'Sofia Coppola';
```

```
-- Alle Filme, die KEINE Komödie sind
SELECT Series_Title, Genre
FROM imdb_top_1000
WHERE NOT Genre LIKE '%Comedy%';
```

## DISTINCT

Mit *DISTINCT* werden **doppelte Werte entfernt**. Beispiel: In der Tabelle gibt es viele „Certificates“ (Jugendfrei, ab 16 Jahren etc.) mehrfach.

```
-- Alle verschiedenen Genres
SELECT DISTINCT Certificate
FROM imdb_top_1000;
```

Ohne *DISTINCT* würde jedes Certificate so oft erscheinen, wie es in der Tabelle vorkommt.

## LIMIT

Oft ist es nicht sinnvoll, **alle Zeilen** auszugeben. Mit *LIMIT* können Sie die Anzahl der Zeilen einschränken.

```
-- Die 5 bestbewerteten Filme
SELECT Series_Title, IMDB_Rating
FROM imdb_top_1000
ORDER BY IMDB_Rating DESC
LIMIT 5;
```

```
-- Die ersten 10 Filme in der Tabelle
SELECT Series_Title, Released_Year
FROM imdb_top_1000
LIMIT 10;
```

## Zusammenfassung

Sie können WHERE-Bedingungen mit vielen Operatoren kombinieren:

| Operator                 | Beispiel                                                   |
|--------------------------|------------------------------------------------------------|
| = , < , > , <= , >= , <> | <i>IMDB_Rating</i> > 9.0                                   |
| BETWEEN ... AND ...      | <i>Released_Year</i> BETWEEN 1990 AND 1999                 |
| IN (...)                 | <i>Director</i> IN ('Nolan','Coppola')                     |
| LIKE                     | <i>Series_Title</i> LIKE 'The%'                            |
| IS NULL                  | <i>Meta_score</i> IS NULL                                  |
| IS NOT NULL              | <i>Meta_score</i> IS NOT NULL                              |
| AND / OR / NOT           | <i>Director</i> = 'Nolan' AND <i>Genre</i> LIKE '%Sci-Fi%' |
| DISTINCT                 | <i>SELECT DISTINCT Genre</i> keine Doppelten Ergebnisse    |
| LIMIT                    | <i>LIMIT 5</i> für die ersten 5 Ergebnisse                 |

<sup>1)</sup>

Der Metascore ist ein Wert, den die Plattform Metacritic vergibt. Er basiert auf Kritiken von professionellen Filmkritiker:innen (z. B. aus Zeitungen, Magazinen, Online-Portalen). Alle Kritiken werden in eine numerische Skala von 0 bis 100 umgerechnet. Anschliessend bildet Metacritic daraus einen gewichteten Durchschnittswert.

From:

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

Permanent link:

[https://wiki.bzz.ch/de/modul/m290\\_guko/learningunits/lu06/theorie/a\\_where\\_bedingungen](https://wiki.bzz.ch/de/modul/m290_guko/learningunits/lu06/theorie/a_where_bedingungen)Last update: **2026/08/13 16:06**