

LU07b - SQL-DML: Einführung & Daten einfügen

Einführung

In LU05 haben wir die **DDL** (Data Definition Language) kennengelernt, mit der Tabellen angelegt werden.

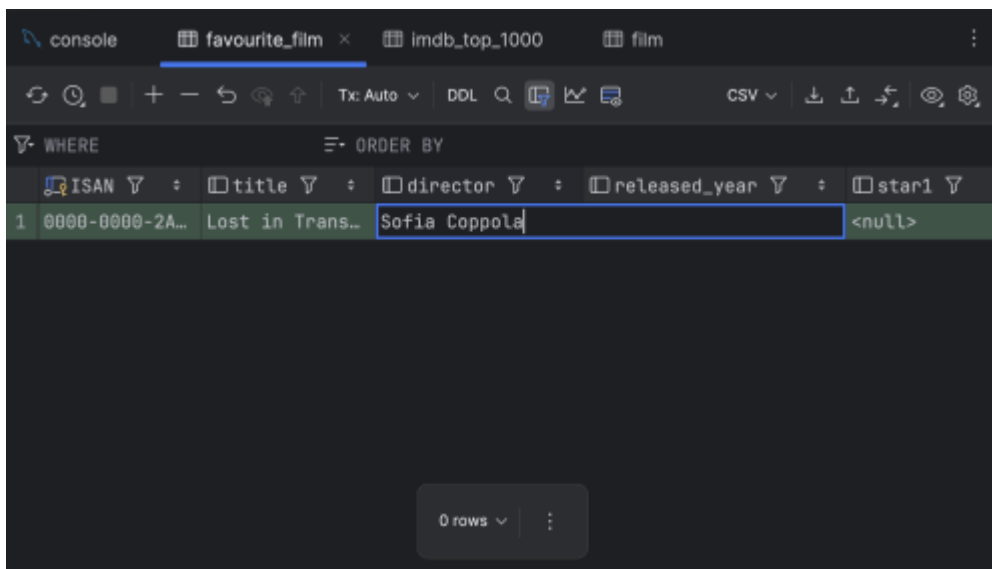
Folgender SQL-Code erzeugt eine **leere** Tabelle *favourite_film* mit den Spalten *film_id*¹⁾, *title*²⁾, *director*³⁾, *released_year*⁴⁾, *star1*⁵⁾:

```
CREATE TABLE favourite_film
(
  film_id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(100),
  director VARCHAR(50),
  released_year YEAR,
  star1 VARCHAR(50)
);
```

Wichtig: *film_id* ist der **Primärschlüssel** → jeder Datensatz erhält automatisch eine eindeutige ID.

DML = Data Manipulation Language: INSERT, UPDATE, DELETE. (Read erfolgt mit SELECT und gehört streng genommen zur **DQL**, wird im Alltag bei CRUD⁶⁾ aber mitgemeint.)

Bisher haben wir neue Datensätze über die **Webstorm-Oberfläche** eingefügt (*Edit Data* > „+“):



Das funktioniert, ist aber mühsam und fehleranfällig, wenn viele Daten eingetragen oder geändert werden müssen. Darum machen wir das ab jetzt **programmatisch mit SQL**.

Die CRUD-Operationen aus der vorherigen Lektion setzen wir dabei direkt mit den Befehlen der **DML**

(Data Manipulation Language) um: INSERT entspricht **Create**, SELECT entspricht **Read**, UPDATE entspricht **Update** und DELETE entspricht **Delete**.

INSERT Statement

Der Befehl INSERT fügt neue Datensätze in eine Tabelle ein. Dabei muss angegeben werden, **in welche Spalten** die Werte eingefügt werden.

Syntax (Einzelzeile)

```
INSERT INTO tabellenname (spalte1, spalte2, spalte3)
VALUES (wert1, wert2, wert3);
```

Syntax (mehrere Zeilen auf einmal)

```
INSERT INTO tabellenname (spalte1, spalte2, spalte3)
VALUES (wert1a, wert2a, wert3a),
       (wert1b, wert2b, wert3b);
```

Syntax (INSERT ... SELECT)

```
INSERT INTO zieltabelle (spalte1, spalte2, spalte3)
SELECT quelle_spalte1, quelle_spalte2, quelle_spalte3
FROM quelle
WHERE bedingung;
```

—

Variante A: Spalten explizit angeben

```
INSERT INTO favourite_film (title, director, released_year,
star1)
VALUES ('Lost in Translation', 'Sofia Coppola', 2003,
'Scarlett Johansson');
```

film_id	title	director	released_year	star1
1	Lost in Translation	Sofia Coppola	2003	Scarlett Johansson

Variante B: Mehrere Zeilen auf einmal

```
INSERT INTO favourite_film (title, director, released_year,
star1)
```

VALUES

```
( 'Inception', 'Christopher Nolan', 2010, 'Leonardo DiCaprio' ),
( 'Arrival', 'Denis Villeneuve', 2016, 'Amy Adams' );
```

film_id	title	director	released_year	star1
1	Lost in Translation	Sofia Coppola	2003	Scarlett Johansson
2	Inception	Christopher Nolan	2010	Leonardo DiCaprio
3	Arrival	Denis Villeneuve	2016	Amy Adams

Variante C: INSERT ... SELECT (Daten aus einer anderen Tabelle übernehmen) Mit INSERT ... SELECT können Datensätze direkt aus einer anderen Tabelle kopiert werden. Das ist z. B. praktisch, wenn Daten aus *imdb_top_1000* übernommen werden sollen.

```
INSERT INTO favourite_film (title, director, released_year,
star1)
SELECT Series_Title, Director, Released_Year, Star1
FROM imdb_top_1000
WHERE Released_Year >= 2000;
```

Hinweis: Mit WHERE können gezielt nur bestimmte Datensätze übernommen werden (z. B. ab Jahr 2000).

1)

Automatisch hochzählende ID, eindeutiger Primärschlüssel

2)

Filmtitel

3)

Regisseur:in

4)

Veröffentlichungsjahr, vierstellig

5)

Hauptdarsteller:in

6)

Create, Read, Update, Delete Operationen

From:

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

Permanent link:

https://wiki.bzz.ch/de/modul/m290_guko/learningunits/lu07/theorie/b_sql_dml

Last update: **2026/08/13 16:06**

