

LU13a: Datenimport & -export (CSV)

Lernziele

Sie können ...

- CSV-Dateien mit dem Datenbank-Plugin von WebStorm in bestehende oder neue Tabellen importieren.
- Tabellen- oder Abfrageergebnisse als CSV-, Excel- oder JSON-Dateien exportieren.
- typische Stolpersteine beim CSV-Austausch (Trennzeichen, Encoding, Datum/Zahlen, Header-Zeile) erkennen und beheben.
- Vor- und Nachteile von CSV im Vergleich zu SQL-Skripten benennen.

Warum verwenden wir CSV-Dateien?

```
Employee,Position,Department,Age,City of Residence,Date of Hire,Annual Income,  
Alice Johnson,Software Engineer,IT,28,New York,05-15-20,$90000.00,  
Bob Davis,Marketing Specialist,Marketing,35,Los Angeles,09-20-18,$75000.00,  
Charlie Smith,Financial Analyst,Finance,30,Chicago,03-10-19,$85000.00,  
Diana Rodriguez,HR Manager,Human Resources,40,Miami,07-05-17,$95000.00,  
Eva Williams,Project Manager,Operations,32,San Francisco,01-12-16,$110000.00,
```

Ausschnitt einer CSV-Datei mit (fiktiven) Daten über Mitarbeiter:innen eines Unternehmens.

CSV¹⁾ sind einfache Textdateien (Dateiendung .csv). Die Daten werden zeilenweise gespeichert, die Spalten durch ein Trennzeichen (z. B. Komma oder Semikolon) getrennt.

Vorteile von CSV

- **Universell:** Fast jede Anwendung (z. B. Excel, Google Sheets, Datenbanken) kann CSV-Daten importieren und exportieren.
- **Einfach & schnell:** Ideal für Importe mit vielen Daten. Die Dateien können ohne zusätzliche Spezialsoftware angezeigt werden.

Eigenschaften von CSV

- **Trennzeichen:** Komma (,) oder Semikolon (;) konsistent verwenden.
- **Encoding:** Möglichst **UTF-8** wählen, damit Umlaute und Sonderzeichen korrekt dargestellt werden.
- **Header-Zeile:** Beim Import die erste Zeile als Spaltenkopf markieren (First row is

header).

- **Datentypen/Formate:** Beim Import Zahlen, Datumsformate (z. B. YYYY-MM-DD) und Dezimaltrennzeichen (Punkt vs. Komma) prüfen.
- **Verknüpfungen in der Datenbank:** CSV enthält keine Constraints²⁾. Diese werden erst im RDBMS definiert und gepflegt.

Voraussetzungen

- WebStorm mit aktiviertem **Database Tools and SQL**-Plugin
- Datenquelle (z. B. MySQL) ist verbunden (Login, Treiber eingerichtet)
- Beispiel-Dateien: Genres einer Filmdatenbank (Daten von letterboxd.com)
 - Den vollständigen Datensatz können Sie auf Kaggle herunterladen: [Link zum ganzen Datensatz](#)

Video: CSV-Daten importieren

Über die WebStorm-Benutzeroberfläche und via SQL-Befehl `LOAD DATA LOCAL INFILE`

[CSV-Import \(Video\)](#)

Hier eine sprachlich und didaktisch etwas gestraffte Version deines Abschnitts:

1) Varianten im Überblick

Daten können auf verschiedene Arten in eine Datenbank importiert bzw. aus ihr exportiert werden. In dieser Lerneinheit unterscheiden wir zwei typische Szenarien:

- **Reine Datensätze** (ohne Struktur): Import/Export über CSV-Dateien mit dem Datenbank-Plugin von WebStorm oder via SQL-Befehl `LOAD DATA LOCAL INFILE`.
- **Ganze Datenbanken** inkl. Struktur und Daten: Export/Import als Dump über SQL-Skripte.

Methode	Was ist das?	Vorteile	Nachteile
GUI in WebStorm	Import/Export von Daten über CSV-Dateien	Schnell, einfach, keine CLI ³⁾ nötig	Keine Struktur/Constraints, Datentypen und Formate müssen sorgfältig geprüft werden
SQL-Skript⁴⁾	Enthält sowohl Struktur (DDL) als auch Daten (INSERT-Befehle)	Reproduzierbar ⁵⁾ , portabel ⁶⁾	Mehr Aufwand beim Erstellen, bei sehr vielen Datensätzen u. U. langsamer

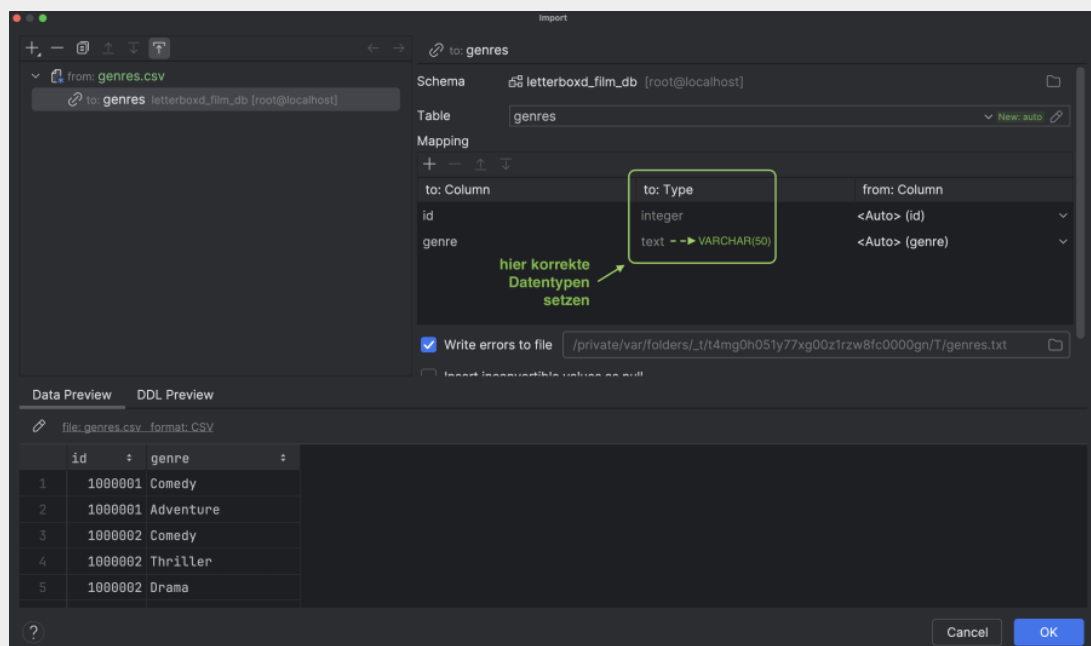
2) CSV importieren (WebStorm GUI)

Ziel: Eine CSV-Datei (z. B. genres.csv) in die Datenbank letterboxd_film_db laden (Datenbank vorher erstellen – siehe Video).

- Rechtsklick auf das gewünschte **Schema** → **Import/Export** → **Import Data from File(s)**.
- CSV-Datei auswählen → **Datentypen** prüfen und bei Bedarf anpassen (z. B. Spalte genre von TEXT zu VARCHAR(50) ändern).
- Vorschau kontrollieren → **Apply** / **OK**.

Screenshot: CSV-Daten importieren

CSV-Import-Dialog (Mapping)



3) CSV-Import per SQL

Wir laden die Datei genres.csv (z. B. aus einer letterboxd⁷⁾-Datenbank) in die Tabelle genres.

Voraussetzungen

- Verbindung in WebStorm ist aktiv.
- In den Data-Source-Eigenschaften ist **AllowLoadLocalInfile** auf **TRUE** gesetzt (im Video gezeigt).
- Der Pfad zur CSV-Datei ist korrekt (angepasst an Betriebssystem und Benutzerkonto).

Tabelle anlegen

```
CREATE TABLE IF NOT EXISTS genres (  
  id INT NOT NULL,  
  genre VARCHAR(50) NOT NULL  
);
```

Import mit LOAD DATA LOCAL INFILE Anstatt via der Webstorm-Oberfläche können wir die Daten auch via SQL-Befehl importieren:

```
LOAD DATA LOCAL INFILE '/Pfad/zu/genres.csv'  
INTO TABLE genres  
FIELDS TERMINATED BY ','  
ENCLOSED BY '"'  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

Hinweise

- **Header-Zeile:** Durch IGNORE 1 ROWS wird die Kopfzeile übersprungen.
- **Trennzeichen & Encoding:** Für dieses CSV wird ein Komma als Separator und UTF-8 als Zeichensatz verwendet.
- **Windows:** Falls nötig LINES TERMINATED BY '\r\n' verwenden.
- **Mehrfach-Genres:** Ein Genre kann mehrfach vorkommen (z. B. Drama). Das ist hier ein „Fehler“ des Datensatzes – in der Praxis würde man nun hierfür eine Zwischentabelle zwischen der Tabelle movies und genres (movie_genre) erstellen.

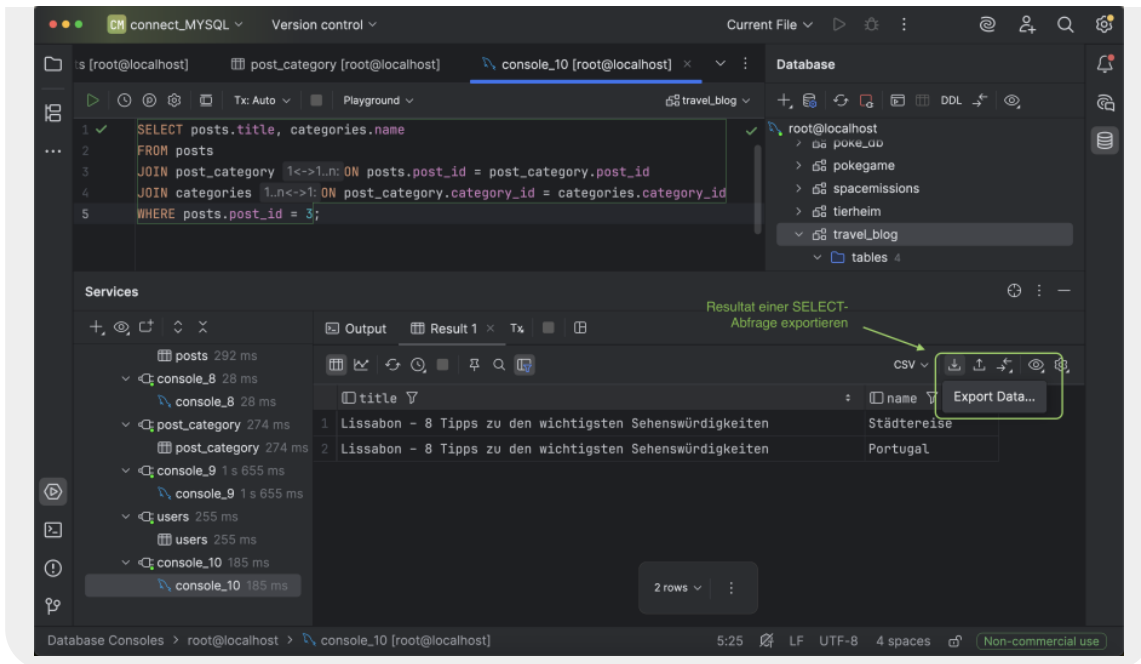
4) CSV/Excel/JSON exportieren (WebStorm GUI)

Ziel: Tabelleninhalte oder ein Abfrageergebnis aus WebStorm exportieren.

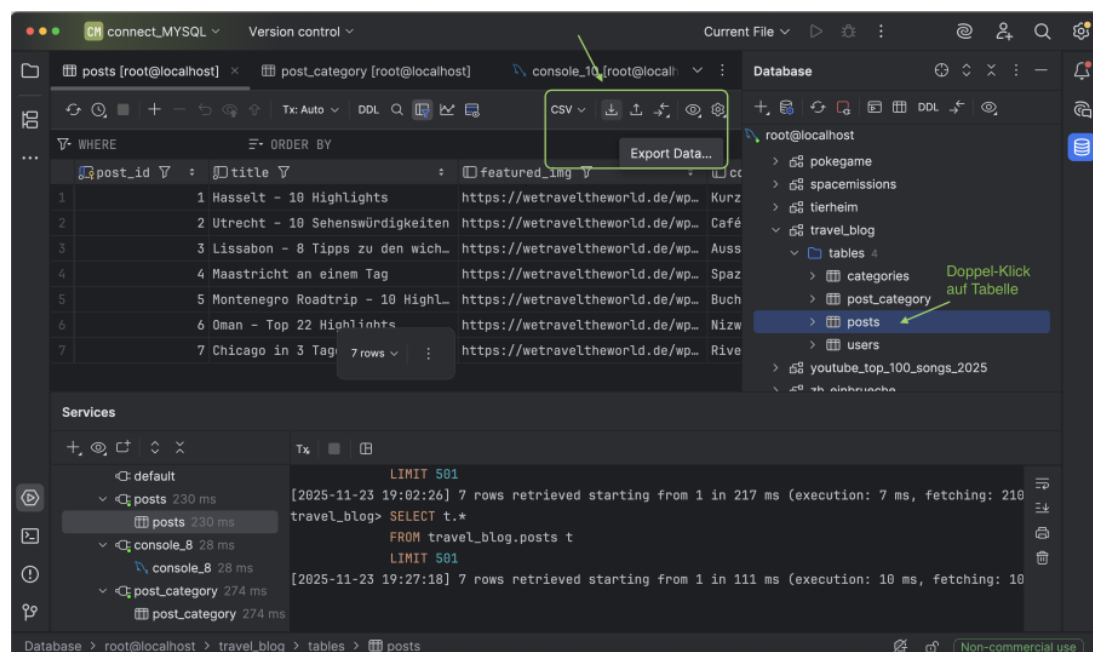
- Tabelle öffnen **oder** ein SELECT-Resultat anzeigen lassen.
- In der Toolbar: **Export Data** → gewünschtes Format wählen (CSV, Excel, JSON, Markdown ...).
- Wichtige Optionen:
 - **Add column header** (Spaltennamen in der ersten Zeile mitexportieren)
 - **Transpose** (optional, dreht Zeilen/Spalten)
 - **Output file** wählen → **Export to File** ausführen.

Screenshot: CSV-Daten exportieren

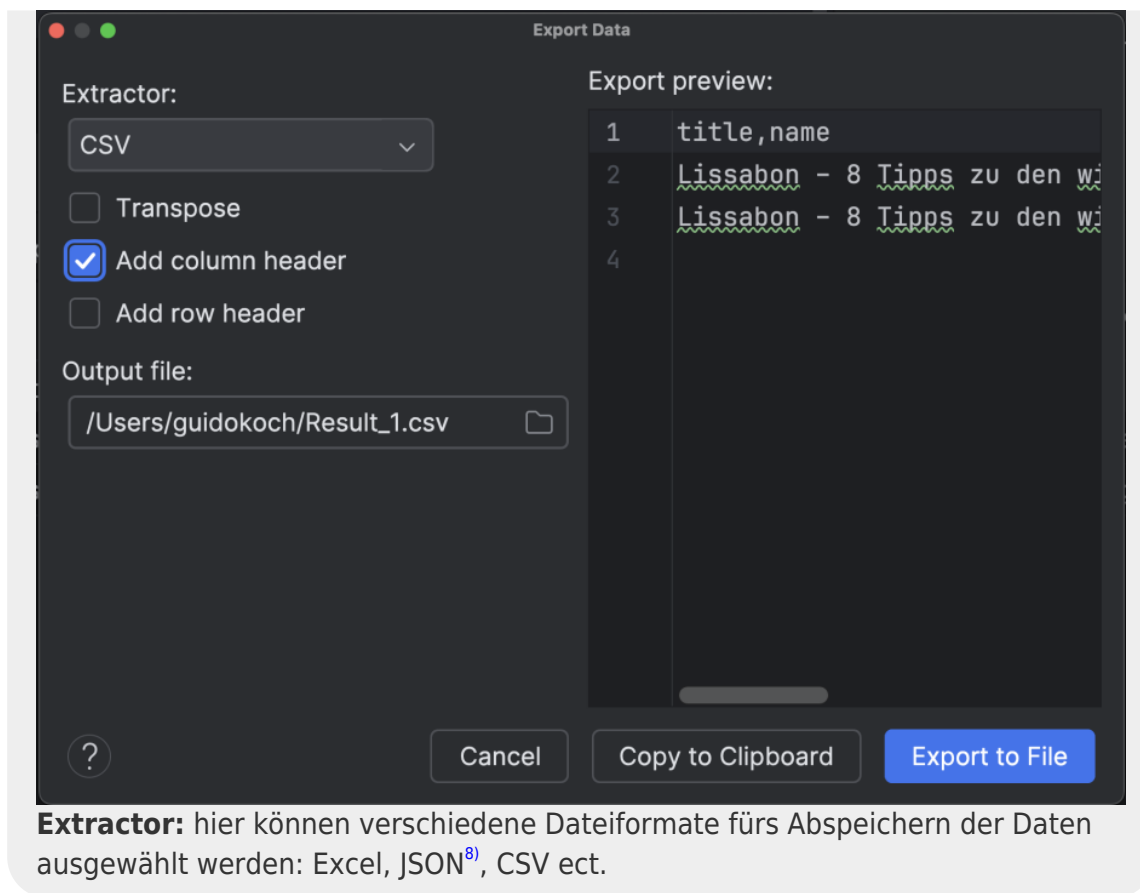
Export von Resultaten einer SELECT-Abfrage



Export einer ganzen Tabelle



Optionen für den CSV-Export



5) Fazit

- CSV eignet sich gut für Testdaten und schnelle Massenimporte.
- Achten Sie auf **Trennzeichen, Encoding und Datentypen**, um typische Fehler zu vermeiden.
- Für Backups und Migration ganzer Datenbanken ist ein Export als **SQL-Skript** (Dump mit Struktur und Daten) meist besser geeignet → nächste Seite.

¹⁾

CSV = Comma Separated Values

²⁾

Integritätsregeln wie Primär-/Fremdschlüssel

³⁾

CLI = Command Line Interface

⁴⁾

Textdatei mit SQL-Befehlen; meist Endung .sql

⁵⁾

bei erneutem Ausführen gleiches Ergebnis

⁶⁾

leicht auf andere Server/PCs übertragbar

⁷⁾

Film-Community-Plattform

⁸⁾

JSON = JavaScript Object Notation; textbasiertes, leichtgewichtiges Datenformat zur Darstellung strukturierter Daten, z. B. für den Datenaustausch zwischen Web-APIs und Anwendungen.

From:

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

Permanent link:

https://wiki.bzz.ch/modul/m290_guko/learningunits/lu13/theorie/a_data_import

Last update: **2025/11/23 20:04**

