LEHRERINFO ZUM VIDEO

Wir bauen eine Datenbank

FACH	STUFE	DAUER	VIDEO
Informatik	Klasse 7–9	13 min	youtube.com/@Erklaerbaer

01 Lernziele

Nach dem Video können die Schülerinnen und Schüler …

- eine Datenbank mit «DB Browser for SQLite» anlegen,
- Tabellen mit passenden Datentypen und Primärschlüssel planen und erstellen,
- Datensätze eintragen, ändern und löschen,
- einfache SQL-Abfragen schreiben: `SELECT`, `WHERE`, `ORDER BY`, `LIKE`, `COUNT`,
- zwei Tabellen über eine Nummer verbinden (Ausleihen),
- Datenschutz und Datensparsamkeit in der eigenen Datenbank umsetzen.

02 Voraussetzungen

Datentypen (Folge 3), Objekte mit Eigenschaften (Folge 10), Tabellen, Schlüssel und erste Abfragen (Folge 12), Datensparsamkeit (Folge 13). Software: «DB Browser for SQLite», kostenlos für Windows, macOS und Linux (sqlitebrowser.org); keine Anmeldung nötig.

03 Ablauf des Videos

ZEIT	KAPITEL	INHALT UND PRAXISBEISPIEL
0:00	Einstieg	Wer hat Momo? · Ident · Fahrplan
0:29	Anforderungen	erfassen, suchen, ausleihen, zurückgeben · Rollen · Bibliothek, Webshop, Verein
1:05	Das Werkzeug	DB Browser for SQLite · `mediathek.db` · drei Bereiche · Änderungen schreiben
1:46	Tabelle books	Spalten, Typen, Primärschlüssel, NOT NULL · `CREATE TABLE` · Merksatz
2:44	Daten eintragen	fünf Bücher per Maus · Anne Frank mit `INSERT`
3:38	Erste Abfragen	alle · Fantasy (3) · nach 2000 (2) · Merksatz PAUSE-FRAGE Michael Ende
4:33	Sortieren, suchen, zählen	`ORDER BY` · `LIKE '%Potter%'` · `COUNT(*)` · Spiele, Rezepte PAUSE-FRAGE vor 1950
5:35	Ändern und Löschen	UPDATE mit WHERE · die UPDATE-Falle · Backup · DELETE · erst SELECT · Merksatz
6:37	Ausleihen	`students`, `loans` · Wer hat Momo? → Mia · JOIN als Ausblick
7:41	Datenschutz	nur Vorname und Klasse · nicht teilen · Backup · alte Ausleihen löschen
8:21	Typische Fehler	vier Irrtümer (siehe 04)
9:15	Rückblick	Bausteine aus 3, 10, 11, 12 und 13 · Mediathek komplett
9:49	Zusammenfassung	Kernaussagen · zurück zum Einstieg
10:24	Quiz	4 Fragen · Endkarte Folge 17

04 Typische Fehlvorstellungen

- „`WHERE genre = Fantasy` geht auch“ – Text gehört in einfache Anführungszeichen, sonst hält SQL das Wort für einen Spaltennamen (Kapitel 5 und 10).
- „UPDATE und DELETE ändern nur die gemeinte Zeile“ – ohne WHERE trifft es alle Zeilen; vorher mit SELECT prüfen (Kapitel 7 und 10).
- „Zahlen kann man auch als Text speichern“ – als Text sortiert «95» hinter «320»; Zahlen als INTEGER (Kapitel 10).

- „Tabellen verbindet man über den Namen“ – Namen sind nicht eindeutig; verbunden wird über den Primärschlüssel (Kapitel 8 und 10).

05 Einsatz im Unterricht

Mitbauen: Das SQL-Skript aus dem Anhang im Reiter «SQL ausführen» einfügen – in zwei Minuten hat jede Gruppe dieselbe Datenbank.

Fragen-Duell: Gruppen stellen sich gegenseitig Fragen an die Mediathek und beantworten sie mit einer Abfrage.

Sicher üben: Vor der UPDATE-Falle eine Kopie der Datei anlegen lassen; danach gemeinsam wiederherstellen.

Eigene Sammlung: Als Projekt eine Tabelle für Spiele, Rezepte oder Sammelkarten planen und füllen – nur mit erfundenen oder unbedenklichen Daten.

Hinweis: Alle Abfrage-Ergebnisse sind mit SQLite nachgeprüft. Die Oberfläche des DB Browser ist im Video vereinfacht nachgebaut; Reiter heissen in der deutschen Version «Datenbankstruktur», «Daten durchsuchen» und «SQL ausführen». `AUTOINCREMENT` verhindert, dass eine gelöschte Nummer erneut vergeben wird; darum bekommt das doppelte Momo die Nummer 7. Personendaten im Unterricht nur erfunden oder mit Einverständnis verwenden.

06 Quellen

- DB Browser for SQLite – sqlitebrowser.org
- SQLite-Dokumentation: `CREATE TABLE`, `AUTOINCREMENT`, Datentypen – sqlite.org/docs.html
- Revidiertes Datenschutzgesetz (revDSG, Schweiz, seit 1.9.2023) · DSGVO Art. 5 (Datenminimierung)
- Lehrplan 21, Modul „Medien und Informatik“, MI.2.1 Datenstrukturen – lehrplan21.ch

NAME _____ KLASSE _____ DATUM _____

Wir bauen eine Datenbank

1 Merksätze ergänzen ●○○

Erst die Tabelle _____ – Spalten, Typen, _____ –, dann Daten eintragen. · Vor jedem UPDATE oder DELETE das _____ prüfen, am besten zuerst mit _____.

2 Tabelle planen ●○○

Eine Spielesammlung soll Titel, Plattform, Altersfreigabe und Erscheinungsjahr speichern. **Plane** die Tabelle `games`: Spaltennamen, Datentyp, Primärschlüssel, Pflichtfelder.

3 Abfragen schreiben ●●○

Schreibe für die Tabelle `books`: a) alle Bücher nach 1990 · b) alle Titel, sortiert nach Titel · c) die Anzahl der Fantasy-Bücher · d) alle Bücher, deren Titel «der» enthält.

4 Was kommt heraus? ●●○

Wie viele Zeilen liefern diese Abfragen bei unseren sechs Büchern? a) `SELECT * FROM books WHERE genre = 'Fantasy' AND year < 1980;` b) `SELECT * FROM books WHERE year BETWEEN 1940 AND 2000;`

5 Fehler finden ●●○

a) `SELECT * FROM books WHERE author = Michael Ende;` b) `DELETE FROM books;` – es sollte nur Buch 6 gelöscht werden. **Erkläre** den Fehler und **verbessere**.

6 Wer hat was? ●●●

In `loans` steht (student_id, book_id): (3, 2), (3, 1), (1, 5), (4, 3). **Finde** mit zwei Abfragen nacheinander heraus, wer Harry Potter (id 5) ausgeliehen hat, und **begründe**, warum `loans` keine Namen speichert.

Lösungen

1 Merksätze

planen, Schlüssel · WHERE, SELECT

2 Tabelle planen

z. B. `id INTEGER PRIMARY KEY AUTOINCREMENT`, `title TEXT NOT NULL`, `platform TEXT`, `age INTEGER` (Altersfreigabe als Zahl, damit man vergleichen kann), `year INTEGER`.

3 Abfragen

a) `SELECT * FROM books WHERE year > 1990;` (Harry Potter, Panem, Tschick) · b) `SELECT title FROM books ORDER BY title;` · c) `SELECT COUNT(*) FROM books WHERE genre = 'Fantasy';` → 3 · d) `SELECT * FROM books WHERE title LIKE '%der%';` (Der Hobbit, Harry Potter und der Stein der Weisen, Das Tagebuch der Anne Frank – LIKE unterscheidet in SQLite bei ASCII-Buchstaben nicht zwischen Gross und Klein).

4 Was kommt heraus?

a) 2 (Der Hobbit 1937, Momo 1973) · b) 3 (Anne Frank 1947, Momo 1973, Harry Potter 1997)

5 Fehler finden

a) Text fehlt in Anführungszeichen: `WHERE author = 'Michael Ende'` · b) Ohne WHERE werden alle Bücher gelöscht: `DELETE FROM books WHERE id = 6;` – vorher mit `SELECT * FROM books WHERE id = 6;` prüfen.

6 Wer hat was?

`SELECT student_id FROM loans WHERE book_id = 5;` → 1 · `SELECT name, class FROM students WHERE id = 1;` → Noah, 8b. Namen würden doppelt gespeichert und sind nicht eindeutig; über die Nummer bleibt jede Person einmal gespeichert, und eine Änderung (z. B. Klassenwechsel) betrifft nur eine Zelle.

Q Quiz im Video

1 B (SELECT) · 2 C (id) · 3 B (alle Bücher werden gelöscht) · 4 A (über die Nummer book_id)

Anhang: mediathek.sql

Im DB Browser for SQLite: «Neue Datenbank» → `mediathek.db` speichern → Reiter «SQL ausführen» → Skript einfügen → ausführen → «Änderungen schreiben». Mit SQLite 3 nachgeprüft.

```
-- Klassen-Mediathek – Serie Informatik, Folge 16 (Grundlagen)
-- Im DB Browser for SQLite: Neue Datenbank mediathek.db → Reiter «SQL ausführen» → alles einfügen → ausführen → Änderungen schreiben

CREATE TABLE books (
    id      INTEGER PRIMARY KEY AUTOINCREMENT,
    title   TEXT NOT NULL,
    author  TEXT,
    year    INTEGER,
    genre   TEXT
);

CREATE TABLE students (
    id      INTEGER PRIMARY KEY AUTOINCREMENT,
    name    TEXT NOT NULL,      -- nur der Vorname (Datensparsamkeit)
    class   TEXT NOT NULL
);

CREATE TABLE loans (
    id          INTEGER PRIMARY KEY AUTOINCREMENT,
    student_id  INTEGER NOT NULL,   -- Nummer aus students
    book_id     INTEGER NOT NULL,   -- Nummer aus books
    date        TEXT NOT NULL       -- Datum als Text, z. B. '2026-03-12'
);

INSERT INTO books (title, author, year, genre) VALUES
    ('Der Hobbit', 'J. R. R. Tolkien', 1937, 'Fantasy'),
    ('Momo', 'Michael Ende', 1973, 'Fantasy'),
    ('Tschick', 'Wolfgang Herrndorf', 2010, 'Roman'),
    ('Die Tribute von Panem', 'Suzanne Collins', 2008, 'Science-Fiction'),
    ('Harry Potter und der Stein der Weisen', 'J. K. Rowling', 1997, 'Fantasy'),
    ('Das Tagebuch der Anne Frank', 'Anne Frank', 1947, 'Biografie');

INSERT INTO students (name, class) VALUES
    ('Noah', '8b'), ('Lea', '8a'), ('Mia', '8a'), ('Elias', '8b');

INSERT INTO loans (student_id, book_id, date) VALUES
    (3, 2, '2026-03-12'), (3, 1, '2026-03-19'), (1, 5, '2026-03-20'), (4, 3, '2026-03-21');

-- Abfragen aus dem Video
SELECT * FROM books;
SELECT * FROM books WHERE genre = 'Fantasy';
SELECT * FROM books WHERE year > 2000;
SELECT * FROM books ORDER BY year;
SELECT * FROM books WHERE title LIKE '%Potter%';
SELECT COUNT(*) FROM books;
SELECT * FROM books WHERE author = 'Michael Ende';
SELECT * FROM books WHERE year < 1950;

-- Wer hat Momo? (Momo hat die id 2)
SELECT student_id FROM loans WHERE book_id = 2;
SELECT name, class FROM students WHERE id = 3;
```