

Übungsblatt „Foreign Keys“

Quellen:

Erklärungen zum Foreign Key:

<https://www.sqlitetutorial.net/sqlite-foreign-key/>

Grundlagen, SQLite – Tools und Übungsdatenbank (falls versäumt):

<https://www.sqlitetutorial.net/sqlite-getting-started/>

Installation von sqlite3 via sqlite-tools und Angabe in der PFAD - Variable

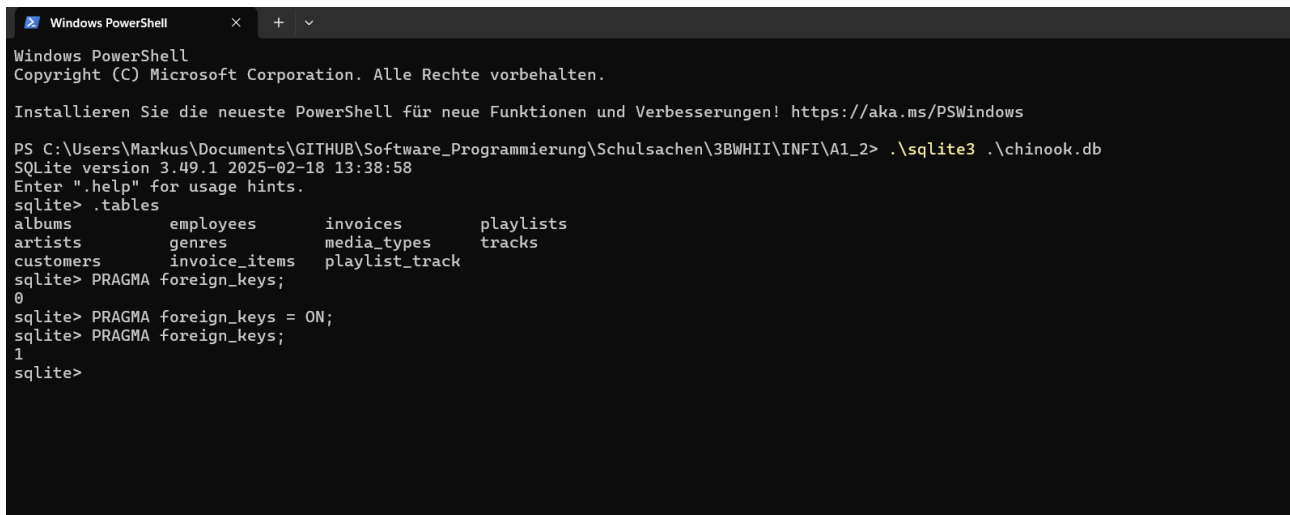
Bearbeitungshinweise: Führe die Übungsaufgabe mit deinem eigenen Computer/ Account durch und dokumentiere mit Screenshots [gesamter Bildschirm] in nachvollziehbarer und strukturierter Weise. Die gesamte Übungsaufgabe bezieht sich auf die Datenbank „chinook.db“ im unveränderten Originalzustand.

Die Herausforderung besteht darin, die Erklärungen zur Tabelle „suppliers“ und „supplier_groups“ im Tutorial zu verstehen und sinngemäß für die nachfolgenden Anwendungsbeispiele aus der Datenbank „chinook“ umzusetzen!

Die Abgabe der Dokumentation erfolgt auf moodle.

AUFGABE 1) Stelle fest, ob das PRAGMA „foreign_keys“ auf „ON“ ist. Falls nicht führe die Einstellung entsprechend durch. Dokumentiere das Ergebnis mittels Screenshot (immer kompletter Bildschirm!)

SCREENSHOTS:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Installieren Sie die neueste PowerShell für neue Funktionen und Verbesserungen! https://aka.ms/PSWindows

PS C:\Users\Markus\Documents\GITHUB\Software_Programmierung\Schulsachen\3BWHII\INFI\A1_2> .\sqlite3 .\chinook.db
SQLite version 3.49.1 2025-02-18 13:38:58
Enter ".help" for usage hints.
sqlite> .tables
albums          employees       invoices        playlists
artists         genres         media_types     tracks
customers       invoice_items  playlist_track
sqlite> PRAGMA foreign_keys;
0
sqlite> PRAGMA foreign_keys = ON;
sqlite> PRAGMA foreign_keys;
1
sqlite>
```

Abbildung 1: PRAGMA foreign_keys Einstellung

AUFGABE 2) Foreign Key Constraints

a) Gib die FOREIGN KEY CONSTRAINTS der Tabelle album aus.

SCREENSHOTS: Hinweis → CREATE STATEMENT KOPIEREN

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Installieren Sie die neueste PowerShell für neue Funktionen und Verbesserungen! https://aka.ms/PSWindows

PS C:\Users\Markus\Documents\GITHUB\Software_Programmierung\Schulsachen\3BWHII\INFI\A1_2> .\sqlite3 .\chinook.db
SQLite version 3.49.1 2025-02-18 13:38:58
Enter ".help" for usage hints.
sqlite> PRAGMA foreign_key_list('albums');
0|0|artists|ArtistId|ArtistId|NO ACTION|NO ACTION|NONE
sqlite> |
```

Abbildung 2: Foreign Key Constraints von "album"

b) Erläutere welche Tabelle „parent“ und welche Tabelle „child“ ist. Erkläre die Bedeutung von „NO ACTION“.

ERLÄUTERUNG:

Die parent – Tabelle ist artists. Albums child. No action verhindert löschen oder verändern von werten in parent tabelle

AUFGABE 3) Erläutere was passiert, wenn ein Eintrag aus der „parent“ Tabelle gelöscht wird, indem du versuchst, den Artist mit der ArtistId = „MEINE KATALOGNUMMER“ via SQL Statement zu löschen. Dokumentiere dein SQL Statement und das Ergebnis des „DELETE“ – Versuches via Screenshot

SCREENSHOTS:

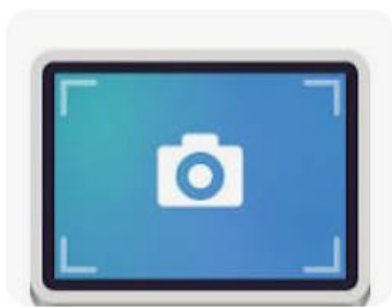


Abbildung 3: Löschversuch "artist"

AUFGABE 4) Erstelle zwei backup – Tables für artist und album mit der Bezeichnung artist_backup_name und album_backup_name und ändere anschließend die Einstellung für den child table „albums“ auf: „ON DELETE SET NULL“ .

Es wird für beide Tabellen eine Backup - Table erzeugt, damit die Daten durch den Löschversuch anschließend wieder hergestellt werden können.

Anschließend wird die albums Tabelle gelöscht und mit dem entsprechend angepassten CREATE Statement (inklusive ON DELETE SET NULL) neu angelegt. Die Daten können durch „INSERT INTO – SELECT FROM my_backup_table_albums“ übernommen werden. Nun kann der artist gelöscht werden und die Auswirkungen auf dessen Alben sollten sichtbar sein.

Hinweis zur Vorgehensweise:

```
CREATE TEMPORARY TABLE temp AS
SELECT
  id,
  parent_id,
  description
FROM child;

DROP TABLE child;

CREATE TABLE child (
  id INTEGER PRIMARY KEY,
  parent_id INTEGER,
  description TEXT,
  FOREIGN KEY(parent_id) REFERENCES parent(id));

INSERT INTO child
( id,
  parent_id,
  description)
SELECT
  id,
  parent_id,
  description
FROM temp;
```

Abbildung 4: Beispiel zur Vorgehensweise bei Änderungen am Table Schema (Stackoverflow)

AUFGABE 5) Lösche nun individuell jenen Eintrag im „parent“ Table, auf den du in deiner Playlist am ehesten verzichten kannst.

Dokumentiere das „DELETE“ Statement und das resultierende Ergebnis im „child“ Table. Dazu ist eine Abfrage erforderlich, aus der die NULL – Werte ersichtlich sind.

SCREENSHOTS:

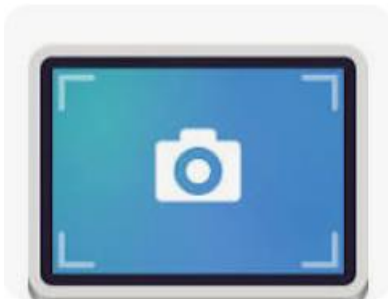


Abbildung 5: Auswirkung von ON DELETE SET NULL auf albums bei Löschen eines artist

Lösungshinweise:

Für Aufgabe 4 muss auch die Verbindung zwischen tracks und albums aufgehoben werden!

```
sqlite> .schema tracks
CREATE TABLE IF NOT EXISTS "tracks"
(
    [TrackId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    [Name] NVARCHAR(200) NOT NULL,
    [AlbumId] INTEGER,
    [MediaTypeId] INTEGER NOT NULL,
    [GenreId] INTEGER,
    [Composer] NVARCHAR(220),
    [Milliseconds] INTEGER NOT NULL,
    [Bytes] INTEGER,
    [UnitPrice] NUMERIC(10,2) NOT NULL,
    FOREIGN KEY ([AlbumId]) REFERENCES "albums" ([AlbumId])
        ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY ([GenreId]) REFERENCES "genres" ([GenreId])
        ON DELETE NO ACTION ON UPDATE NO ACTION,
    FOREIGN KEY ([MediaTypeId]) REFERENCES "media_types" ([MediaTypeId])
        ON DELETE NO ACTION ON UPDATE NO ACTION
);
CREATE INDEX [IFK_TrackAlbumId] ON "tracks" ([AlbumId]);
CREATE INDEX [IFK_TrackGenreId] ON "tracks" ([GenreId]);
CREATE INDEX [IFK_TrackMediaTypeId] ON "tracks" ([MediaTypeId]);
```

Dazu wird tracks in eine Backup Tabelle kopiert und gelöscht. Das Löschen von tracks ist nur möglich, wenn PRAGMA foreign_keys = OFF.

Anschließend wieder auf ON und es kann ein Artist gelöscht werden.

Z.B. Nr. 1 = AC/DC

Das Ergebnis kann wie folgt abgefragt werden:

```
sqlite> Drop TABLE tracks;
sqlite> .schema tracks_backup;
sqlite> .schema tracks_backup
CREATE TABLE tracks_backup(
    TrackId INT,
    Name TEXT,
    AlbumId INT,
    MediaTypeId INT,
    GenreId INT,
    Composer TEXT,
    Milliseconds INT,
    Bytes INT,
    UnitPrice NUM
);
sqlite> PRAGMA foreign_keys = ON
...> ;
sqlite> PRAGMA foreign_keys
...> ;
1
sqlite> DELETE FROM artists WHERE ArtistId = 1;
sqlite> SELECT * FROM albums WHERE ArtistId= 1;
sqlite> SELECT * FROM albums WHERE ArtistId ISNULL;
1|For Those About To Rock We Salute You|
4|Let There Be Rock|
sqlite> |
```