

RELATIONALE DATENBANKEN

Arbeitsauftrag – Tabellen Erstellen und Tupel Einfügen

Aufgabenstellung

Tabellen erstellen	2
SQLite Dokumentation	2
Aufgabe	2
Tabellen ändern	3
SQLite Dokumentation	3
Aufgabe	3
Tabellenattribute indizieren.....	3
SQLite Dokumentation	3
Aufgabe	3
Tabellen löschen.....	4
SQLite Dokumentation	4
Aufgabe	4
Tupel einfügen	4
SQLite Dokumentation	4
Aufgabe	4
Tupel updaten	5
SQLite Dokumentation	5
Aufgabe	5
Tupel löschen	5
SQLite Dokumentation	5
Aufgabe	5

Hinweise:

Einige der Aufgaben/Befehle dienen der Korrektur falls Ihnen ein Fehler beim Abarbeiten der Aufgaben unterläuft. Diese sind als optional gekennzeichnet. Falls sie einen dieser Befehle benutzen (eben weil Ihnen ein Fehler passiert ist) dokumentieren Sie die Verwendung. Beginnen Sie auf keinen Fall von vorne ;-)

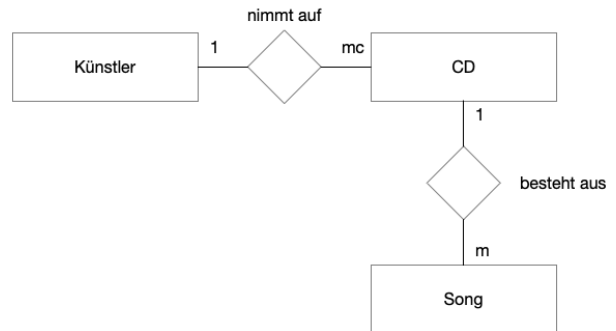
Ihre Abgabe soll alle ausgeführten SQL Befehle inkl. Screenshots der Ergebnisse beinhalten. Geben Sie ein PDF Dokument ab, dass Code und Screenshots den Aufgaben eindeutig zuordnet und Ihre Arbeit dokumentiert.

Tabellen erstellen

Das Fundament eines Datenbanksystems wird durch die Datenbasis gebildet. Diese wiederum besteht aus verschiedenen Tabellen mit unterschiedlichen Attributen.

Eine neue Tabelle wird mit dem Befehl „CREATE TABLE“ **erstellt**.

```
CREATE TABLE Tabellename (  
  Attributname_1 Datentyp [NOT NULL],  
  Attributname_2 Datentyp [NOT NULL],  
  Attributname_n Datentyp [NOT NULL] );
```



Wir haben bereits Tabellen für unsere CD-Sammlung erstellt:

```
CREATE TABLE "Kuenstler" (  
  "KId" INTEGER NOT NULL UNIQUE,  
  "Name" TEXT NOT NULL,  
  "Gruendungsjahr" INTEGER,  
  PRIMARY KEY("KId" AUTOINCREMENT)  
);  
  
CREATE TABLE "CD" (  
  "CDId" INTEGER NOT NULL UNIQUE,  
  "Albumtitel" TEXT NOT NULL,  
  "IId" INTEGER NOT NULL,  
  "Erscheinungsjahr" INTEGER,  
  PRIMARY KEY("CDId" AUTOINCREMENT),  
  FOREIGN KEY (IId) REFERENCES Kuenstler (KId) ON DELETE CASCADE ON UPDATE CASCADE  
);  
  
CREATE TABLE "Song" (  
  "CDId" INTEGER NOT NULL,  
  "Track" INTEGER NOT NULL,  
  "Titel" TEXT NOT NULL,  
  PRIMARY KEY("CDId","Track")  
  FOREIGN KEY (CDId) REFERENCES CD (CDId) ON DELETE CASCADE ON UPDATE CASCADE  
)
```

SQLite Dokumentation

Eine ausführliche Dokumentation zu CREATE TABLE in SQLite finden Sie hier:

https://www.sqlite.org/lang_createtable.html

Aufgabe

Erstellen Sie in SQLite die Tabellen für Künstler, CD und Song (siehe oben).

Dokumentieren Sie ihre Arbeit mit den entsprechenden SQL Statements und Screenshots des Ergebnisses.

Tabellen ändern

Für den Fall, dass eine bestehende Tabelle noch mit weiteren Attributen ergänzt werden soll oder bestehende Attribute geändert werden müssen, kann dies mit dem Befehl „ALTER TABLE“ erreicht werden.

```
ALTER TABLE Tabellenname ADD  
( Neuer_Attributname Datentyp [NOT NULL], ... );
```

SQLite Dokumentation

Eine ausführliche Dokumentation zu ALTER TABLE in SQLite finden Sie hier:

https://www.sqlite.org/lang_altertable.html

Aufgabe

Falls Sie eine Tabelle ändern wollen/müssen, verwenden Sie den ALTER TABLE Befehl. Diese Aufgabe ist optional.

Dokumentieren Sie gegebenenfalls ihre Arbeit mit den entsprechenden SQL Statements und Screenshots des Ergebnisses.

Tabellenattribute indizieren

Die Indizierung von Tabellenattributen wird in folgenden Situationen eingesetzt:

- Ein Attribut oder eine Attributkombination darf nur eindeutige Werte annehmen können.
- Das Suchen nach Attributwerten soll beschleunigt werden

Situation a kommt meistens bei Identschlüsselattributen (diese müssen definitionsgemäß eindeutig sein) vor, oder aber bei Fremdschlüsselattributen, um ein 1-c-Beziehung zu definieren. Situation b kommt meist bei Fremdschlüsselattributen vor, um Abfragen zu beschleunigen. Doch auch bei Attributen, die für Sortierungen oder beim Suchen verwendet werden, bringt die Indizierung Geschwindigkeitsverbesserungen.

```
CREATE [UNIQUE] INDEX Indexname ON  
Tabellenname (Attribut1, Attribut2, ... );
```

Vor jedem Einfügen eines neuen Datensatzes bzw. Nachführen eines bestehenden Datensatzes überprüft das Datenbankverwaltungssystem automatisch, ob die Attributwertkombination noch eindeutig ist. Falls dies nicht der Fall ist, wird die laufende Transaktion mit einer Fehlermeldung abgebrochen. Der Namen des Index darf nicht mit dem Tabellennamen identisch sein.

Um einen Index wieder zu entfernen, wird ein SQL-Befehl verwendet:

```
DROP INDEX Indexname ON Tabellenname;
```

Wenn Tabelle indiziert wurde. Beim Suchen —> schneller, weil durgehasht wird. Nicht mehr nach Strings, etc. gefiltert wird.

SQLite Dokumentation

Eine ausführliche Dokumentation zu CREATE INDEX in SQLite finden Sie hier:

https://www.sqlite.org/lang_createindex.html

Aufgabe

Erstellen sie folgende Indizes:

- Eindeutiger Index CDId in der Tabelle CD für das Attribut CDId
- Eindeutiger Index KId in der Tabelle Kuenstler für das Attribut KId
- Eindeutiger Index TrackId in der Tabelle Song für die Attributkombination CDId und Track
- Aufsteigender Index ErscheinungsjahrIdx in der Tabelle CD für das Attribut Erscheinungsjahr

Dokumentieren Sie Ihre Arbeit mit den entsprechenden SQL Statements und Screenshots des Ergebnisses.

Tabellen löschen

Für das Löschen von Tabellen kommt folgender SQL-Befehl zum Einsatz.

```
DROP TABLE Tabellenname;
```

SQLite Dokumentation

Eine ausführliche Dokumentation zu DROP TABLE in SQLite finden Sie hier:

https://www.sqlite.org/lang_droptable.html

Aufgabe

Falls Sie eine Tabelle löschen wollen/müssen, verwenden Sie den DROP TABLE Befehl. Diese Aufgabe ist optional.

Dokumentieren Sie gegebenenfalls ihre Arbeit mit den entsprechenden SQL Statements und Screenshots des Ergebnisses.

Tupel einfügen

Das Einfügen eines neuen Datensatzes in eine Tabelle erfolgt mit dem Befehl „INSERT“. Mit der Attributliste wird angegeben, in welche Attribute eines neuen Datensatzes Attributwerte eingegeben werden sollen. Die Reihenfolge der Attributwerte muss mit der Reihenfolge der Attributnamen übereinstimmen. Falls keine Attributliste angegeben wird, muss die Reihenfolge der Attributwerte der Reihenfolge der Attribute bei der Tabellendefinition entsprechen. Alle „NOT NULL“-Attribute müssen Attributwerte annehmen.

```
INSERT INTO Tabellenname [ ( Attribut1, Attribut2 ... ) ]  
VALUES ( Attributwert1, Attributwert2, ... );
```

SQLite Dokumentation

Eine ausführliche Dokumentation zu INSERT INTO in SQLite finden Sie hier:

https://www.sqlite.org/lang_insert.html

Aufgabe

Fügen Sie folgende Tupel ein:

- Tabelle Künstler
 - Anastacia (1999)
 - Pink Floyd (1965)
- Tabelle CD
 - „Not that kind“ von Anastacia; erschienen 2000
 - „Wish you were here“ von Pink Floyd; erschienen 1975
 - „Free of nature“ von Anastacia; erschienen 2001
- Tabelle Song
 - Song Nr 1 - „Not that kind“ auf dem Album „Not that kind“
 - Song Nr 2 - „I’m outta love“ auf dem Album „Not that kind“
 - Song Nr 3 - „Cowboys and Kisses“ auf dem Album „Not that kind“
 - Song Nr 1 - „Shin on your craxxy diamond“ auf dem Album „Wish you were here“
 - Song Nr 1 - „Paid my dues“ auf dem Album „Free of nature“

Dokumentieren Sie gegebenenfalls ihre Arbeit mit den entsprechenden SQL Statements und Screenshots des Ergebnisses.

Tupel updaten

```
UPDATE Tabellenname  
SET Attribut1 = Ausdruck1, Attribut2 = Ausdruck2, ...  
[ WHERE Bedingung für Update ];
```

Es müssen der Tabellenname und die zu ändernden Attribute angegeben werden. Mit „Ausdruck“ ist ein Attributwert oder eine Berechnung gemeint. Mit dem Bedingungsteil (WHERE) kann angegeben werden, welche Datensätze in der Tabelle geändert werden sollen. Falls keine Bedingung angegeben wird, werden alle Datensätze der Tabelle nachgeführt.

SQLite Dokumentation

Eine ausführliche Dokumentation zu UPDATE in SQLite finden Sie hier:

https://www.sqlite.org/lang_update.html

Aufgabe

In der Tabelle Song ist Ihnen ein Fehler unterlaufen. Der Song Nr 1 auf „Wish you were here“ heißt natürlich „Shine on your crazy diamond“. Ändern Sie das Tupel entsprechend!

Dokumentieren Sie ihre Arbeit mit den entsprechenden SQL Statements und Screenshots des Ergebnisses.

Tupel löschen

Das Löschen von Datensätzen geschieht mit dem Befehl „DELETE“.

```
DELETE FROM Tabellenname  
[ WHERE Bedingung für Delete ];
```

SQLite Dokumentation

Eine ausführliche Dokumentation zu DELETE in SQLite finden Sie hier:

https://www.sqlite.org/lang_delete.html

Aufgabe

Falls Sie eine Tupel löschen wollen/müssen, verwenden Sie den DELETE Befehl. Diese Aufgabe ist optional.

Dokumentieren Sie gegebenenfalls ihre Arbeit mit den entsprechenden SQL Statements und Screenshots des Ergebnisses.