

Studienleistung 3

David Halbhuber

Aufgabe: Erste Schritte mit SQLite in JavaScript

In dieser Aufgabe sollen Sie die ersten Schritte im Umgang mit SQLite und JavaScript umsetzen und demonstrieren. Für die Bearbeitung der Aufgabe stellen wir Ihnen ein Projekt via Grips zur Verfügung das bereits alle nötigen Elemente enthält. Das Projekt beinhaltet bereits:

- Eine eingebundene SQLite Datenbank
- Eine Node.Js Umgebung die eine Express App bereitstellt
- Eine Möglichkeit Queries an die eingebundene Datenbank zu senden

Beginnen Sie mit der Entwicklung in der './resources/js/index.js' Datei. In der 'index.js' initiiert die asynchrone Funktion 'initDatabaseConnection' die Verbindung zur Datenbank. Das Datenbankobjekt wird dann in der Variable 'db' gesichert. Durch ein Aufruf der 'runQuery' Funktion können Queries an die Datenbank gesendet werden. Queries werden als String in der Variable 'query' deklariert und dann an 'runQuery' übergeben. Das Ergebnis der Query, falls es eines gibt, wird in der 'response' Variable gesichert.

Beginnen Sie mit den folgenden Aufgaben nach dem Sie sich mit der Arbeitsumgebung vertraut gemacht haben. Gehen Sie dabei schrittweise vor und erstellen für jede Teilaufgabe ein neues Paar aus Query und Response. Orientierten Sie sich bei der Erstellung an dem mitgelieferten Beispiel.

1. Erstellen Sie eine neue Tabelle (CREATE) mit dem Namen 'Persons' in der Datenbank die, die folgenden Informationen abbilden soll: PersonenID, Nachname, Vorname, Adresse, Wohnort, AlterInJahren .
 - 1.1 Achten Sie bei der Erstellung auf die Verwendung von korrekten Datentypen (char, varchar, binary, tinyblob,...).
 - 1.2 Begründen Sie einem kurzen Kommentar die Wahl Ihrer Datentypen.
2. Erstellen Sie nun einen neuen Eintrag (INSERT) in der Persons Tabelle mit einem fiktiven Person die jünger als 50 Jahre ist.
3. Erstellen Sie eine Query (SELECT) um den gesamten Inhalt Persons Tabelle abzufragen. Geben Sie das Ergebnis der Abfrage in der Konsole aus.

4. Erstellen Sie nun eine eigene Funktion zum Erstellen von neuen Einträgen in die Persons Tabelle
 - 4.1 Definieren Sie die Datenbank, sowie den zu erstellenden Inhalt als Übergabeparameter.
 - 4.2 Erstellen Sie innerhalb der Funktion die SQL-Query aus den Übergabeparametern zusammen.
 - 4.3 Definieren Sie die Anfrage an die Datenbank innerhalb der Funktion.
5. Nutzen Sie nun die Funktion zum Erstellen von neuen Einträge um zwei weitere fiktive Personen die beide älter als 50 Jahre sind, in Ihrer Datenbank zu hinterlegen. Sollten Sie bei Schritt 4. auf Probleme gestoßen sein, erstellen Sie die Benutzer wie in Schritt 2.
6. Aktualisieren Sie nun den Wohnort aller Personen die älter als 50 Jahre sind auf "Dresden" (UPDATE).
7. Erstellen Sie nun eine zweite Tabelle die alle Wohnorte speichern soll. Normalisieren Sie die erstellen Tabellen nun soweit wie möglich.
8. Erstellen zuletzt eine komplexere Abfrage, die alle Personen die in Dresden wohnen in alphabetischer Reihenfolge erhält (SELECT). Geben Sie das Ergebnis der Abfrage aus. Achten Sie darauf das die Abfrage über die Wohnort Tabelle selektiert und nicht über die WohnortID in der Wohnort Tabelle.

Beachten Sie, dass die Anwendung Node.js verwendet. Analog zur App die wir in der Übung entwickelt haben, muss das Projekt daher via 'node index.js' und das Terminal oder dem integrierten NPM Skript gestartet werden.

Die folgenden Links sollen Sie dabei unterstützen die Funktionalität zu implementieren:

- [Hier](#) erfahren Sie mehr über SQL
- [Hier](#) erfahren Sie mehr über SQLite

Abgabekriterien:

Laden Sie Ihre Lösung bis spätestens 26.7.2022 (23:59 Uhr) als zip-komprimierten Ordner auf GRIPS hoch. Benennen Sie die einzelnen Dateien pro Aufgabe sinnvoll und verwenden Sie geeignete Formate:

- Aufgabe: Das gesamte Projekt (HTML, JS, CSS)

Der Name der Zip-Datei ergibt sich aus dem Präfix „SL_WT_SS22“, der Nr. der Studienleistung, ihrem Vor- und Nachnamen jeweils getrennt durch _ .

Beispiel: **SL_WT_SS22_3_Max_Mustermann.zip**