

IWGS2_SS20_Zettel4

July 7, 2020

1 Informatische Werkzeuge in den Geistes- und Sozialwissenschaften II

1.1 Hausaufgabe 4 (Einführung in SQL-Datenbanken)

Erschienen: 16.05.2020

Abgabe bis: 24.05.2020

Bitte laden Sie Ihre Notebooks bis 23:59 Uhr am Abgabetag in Ihrer Übungsgruppe bei [StudOn](#) hoch.

Wenn Ihnen einige der hier verwendeten Konzepte unbekannt sind oder Sie nicht wissen, wie Sie fortfahren sollen, können Sie die [Vorlesungsunterlagen](#) zu Rate ziehen oder jederzeit Fragen im [Forum](#) oder auf [Slack](#) stellen, im [Tutorium](#) nachfragen, oder Ihrem Tutor eine Mail schreiben.

1.2 Aufgabe 4.1 (SQL Basics, 30 Punkte)

Für die folgenden Aufgaben versetzen wir uns in die Situation, dass wir aus nicht näher genannten Gründen der nationalen Sicherheit, der nuklearen Abrüstung und dem Wohl der Wale im Atlantik gerne eine Datenbank von Nachahmungen beliebter Videospiele aufbauen möchten. Zunächst soll nur eine Tabelle darin enthalten sein, die ein paar Eckdaten für jedes Spiel enthält.

Aufgabe: Schreiben Sie ein Python-Programm dass sich zur Datenbank `copycats.sqlite` verbindet (wenn diese Datenbank noch nicht existiert, wird sie automatisch erstellt). Danach soll Ihr Programm in dieser Datenbank eine Tabelle `Games` erstellen und füllen, die der folgenden Tabelle entspricht:

Title	Publisher	Year	Genre
Critter Crossing: Latest Skylines	Yamauchi	2020	Social Sim
Weird Life	Triangle Xine	2015	Graphic Adventure
Group Castle 2	Faucet	2007	First Person Shooter

Benutzen Sie dazu die SQL-Befehle `CREATE TABLE <name> (<columns>);` und `INSERT INTO <table> VALUES (<record>);`. Verwenden Sie sinnvolle Datentypen für jede Zeile.

Anmerkung: Denken Sie daran, dass Sie nach Ihren INSERT-Befehlen `.commit()` auf Ihrer Datenbankverbindung aufrufen müssen, damit die Änderungen an der Datenbank auch dauerhaft bleiben.

Wenn die Tabellen in Ihrer Datenbank zwar existieren aber keinen Inhalt haben, haben Sie dies wahrscheinlich vergessen.

Anmerkung: Wahrscheinlich werden Sie Ihr Python-Programm mehrmals ausführen, während Sie es entwickeln. Wenn allerdings in einer Datenbank eine Tabelle mit dem gleichen Namen schon existiert (z.B. aus dem vorherigen Durchlauf), tritt ein Fehler auf. Um diesen zu umgehen, können Sie bevor Sie die Tabelle erstellen den SQL-Befehl `DROP TABLE IF EXISTS <name>` ausführen. Dieser Befehl löscht eine gegebenenfalls vorhandene Tabelle.

```
[1]: # Ihr Code hier! (Doppelklick zum Editieren, Shift + Enter um die Zelle ↵
      ↪ auszuführen)
```

Anmerkung: Sie können die Datenbank, die Ihr Code erstellt entweder hier auf JupyterLab mit der Konsole untersuchen (hilfreiche Befehle in der SQLite-Konsole, welche sie mit `sqlite3` starten können, sind z.B. `tables` oder `SELECT * FROM Games;`), oder sich die `.sqlite`-Datei auf Ihren eigenen Rechner downloaden und dort mittels eines [Database Browsers](#) inspizieren.

1.3 Aufgabe 4.2 (Ermittlungen, 40 Punkte)

Nachdem Sie die Datenbank in Aufgabe 1 erstellt haben bekommen Sie nun vom Hauptquartier Informationen über einige Verdächtige an ihrer Universität zugesendet, die angeblich jeweils eins der genannten Spiele unter der Hand verkaufen sollen. Ihre Datenbank soll nun so erweitert werden, dass auch diese Informationen sinnig widerspiegelt werden. Dabei ist es dem Hauptquartier *extrem* wichtig, dass keine Information unnötig doppelt gespeichert wird.

Aufgabe: Erweitern Sie Ihren Python-Code aus der ersten Aufgabe so, dass die Tabelle `Games` nun einen Integer als Primary Key enthält, über den wir Einträge in dieser Tabelle referenzieren können:

GameID	Title	Publisher	Year	Genre
1	Critter Crossing: Latest Skylines	Yamauchi	2020	Social Sim
2	Weird Life	Triangle Xine	2015	Graphic Adventure
3	Group Castle 2	Faucet	2007	First Person Shooter

Erstellen Sie außerdem eine zweite Tabelle `Suspects`, die den Namen, das Geburtsdatum und die GameID der jeweiligen verdächtigten Person enthält. Die GameID soll hierbei ein Foreign Key sein, der den Primary Key der Tabelle `Games` referenziert.

Name	Birthday	GameID
Detlef Dreckamsteck	04.04.2000	1
Marcus Mafiosi	26.09.1999	3
Fatma Fantasiename	11.12.2004	2
Finn Ferbrecher	14.01.2001	2

Hinweis: In den letzten Aufgabenblättern haben wir oft betont, dass Sie den Code aus früheren Zellen weiterverwenden können und sich damit Code-Duplizierung sparen können. Da wir hier das

Layout der Tabelle aber ändern, empfehlen wir, einfach den Code von oben zu kopieren und dann entsprechend anzupassen.

```
[2]: # Ihr Code hier! (Doppelklick zum Editieren, Shift + Enter um die Zelle  
      ↪ auszuführen)
```

1.4 Aufgabe 4.3 (Feierabend, 30 Punkte)

Uuuuund geschafft! Ab jetzt sollen sich die Behörden um alles Weitere kümmern, um diesem Videospielring das Handwerk zu legen. Jetzt haben Sie sich den Feierabend wirklich redlich verdient – In dem Moment poppt ein einkommender Videoanruf auf Ihrem Computerbildschirm auf.

”Hallo? Hier ist Malte Museumsdirektor. Wir kennen uns noch von damals, über Ädgar Ägyptologe, wissen Sie noch?”

Noch bevor Sie ihm erklären können, dass Sie wirklich müde sind und gerade vermutlich die Wale im Atlantik gerettet haben, erzählt er auch schon. Er hätte vor ein paar Wochen Beatrice Beispielperson eingestellt und sie beauftragt, eine Bilddatenbank zu programmieren. Beatrice hätte lange Probleme gehabt, aber letzte Woche hat dann plötzlich alles funktioniert! Beatrice hätte berichtet, sie hätte ”ein ganz klein wenig Hilfe” von Ihnen bekommen.

Jetzt ist Beatrice im Urlaub und Malte braucht dringend Hilfe bei einem wichtigen Feature Request, der kommende Projekte deutlich vereinfachen soll. Ob Sie ihm helfen können, fragt er.

Aufgabe: In der letzten Woche haben wir den Inhalt von CSV-Dateien in eine Python-Liste gespeichert. Listen sind gut, aber wenn die Datenmengen groß werden, wird die Suche nach Datensätzen darin sehr ineffizient. Verwenden Sie den Code von letzter Woche, um die CSV-Datei *metadata.csv* in eine neue SQL Datenbank *image_database.sqlite* zu speichern. Das Vorgehen ist dabei größtenteils analog zu Teilaufgabe 4.1. Verbinden Sie sich zu einer neuen Datenbank und legen Sie eine Tabelle *Images* an. Die Spalten können Sie dabei aus dem CSV übernehmen. Ihr Programm sollte automatisch jede CSV-Datei mit diesem Spaltenformat lesen können, auch wenn die Datei deutlich mehr Zeilen als die bisherige *metadata.csv* hat.

Da das Programm in dieser Woche nicht an Bottle geknüpft ist, können Sie Ihre Lösung direkt hier im Notebook in der nächsten Zelle implementieren.

```
[3]: # Ihr Code hier! (Doppelklick zum Editieren, Shift + Enter um die Zelle  
      ↪ auszuführen)
```

```
[ ]:
```