
INFI – Protokoll

Datenbanken



"[Dieses Foto](#)" von Unbekannter Autor ist lizenziert gemäß [CC BY-SA-NC](#)

Ausgeführt von:

Michelle Lechner

Wien, am 16.09.2022

Inhaltsverzeichnis

1	Wiederholung / Einführung in SQL	3
1.1	SQL – Basics auf Khanacademy.org	3
1.1.1	Book List Challenge	3
1.1.2	Box Office	3
1.1.3	ToDo - List	4
1.1.4	Projekt: Design a store database	4
	Abbildungsverzeichnis	6

1 Wiederholung / Einführung in SQL

1.1 SQL – Basics auf Khanacademy.org

1.1.1 Book List Challenge

Challenge: Book list database

Step 2

Now, add three of your favorite books into the table.

Hint What's this?

INSERT INTO _ VALUES (-);

1 CREATE TABLE booklist
2 (id INTEGER PRIMARY KEY, name TEXT, rating INTEGER);
3 INSERT INTO booklist VALUES ("1", "Harry Potter", "9"), ("2",
4 "Saeculum", "7"), ("3", "Buch", "5");

DATABASE SCHEMA

booklist	3 rows
id (PK)	INTEGER
name	TEXT
rating	INTEGER

Congratulations!
You earned 1050 points!

Step 2/2 Spin-off

Abbildung 1 - Book List Challenge

1.1.2 Box Office

Challenge: Box office hits database

Filter recent movies

Now, add a second query after the first, that retrieves only the movies that were released in the year 2000 or later, not before. Sort the results so that the earlier movies are listed first. You should have 2 SELECT statements after this step.

Hint What's this?

SELECT ... WHERE -;

1 CREATE TABLE movies (id INTEGER PRIMARY KEY, name TEXT, release_year
2 INTEGER);
3 INSERT INTO movies VALUES (1, "Avatar", 2009);
4 INSERT INTO movies VALUES (2, "Titanic", 1997);
5 INSERT INTO movies VALUES (3, "Star Wars: Episode IV - A New Hope",
6 1977);
7 INSERT INTO movies VALUES (4, "Shrek 2", 2004);
8 INSERT INTO movies VALUES (5, "The Lion King", 1994);
9 INSERT INTO movies VALUES (6, "Disney's Up", 2009);
10
11 SELECT * FROM movies;
12 SELECT * FROM movies WHERE release_year >= 2000 ORDER BY release_year

QUERY RESULTS

id	name	release_year
1	Avatar	2009
2	Titanic	1997
3	Star Wars: Episode IV - A New Hope	1977
4	Shrek 2	2004
5	The Lion King	1994
6	Disney's Up	2009

Congratulations!
You earned 1500 points!

Step 2/2 Spin-off

Abbildung 2 - Box Office

1.1.3 ToDo - List

The screenshot shows the Khan Academy interface for a challenge titled "Challenge: TODO list database stats". The challenge is part of a series on SQL basics. The current step is Step 2, which asks the user to select the SUM of minutes for all items in the TODO list. The user has entered the following SQL code:

```
1 CREATE TABLE todo_list (id INTEGER PRIMARY KEY, item TEXT, minutes  
2 INTEGER);  
3 INSERT INTO todo_list VALUES (1, "Wash the dishes", 15);  
4 INSERT INTO todo_list VALUES (2, "vacuuming", 20);  
5 INSERT INTO todo_list VALUES (3, "Learn some stuff on KA", 30);  
6 INSERT INTO todo_list VALUES (4, "print homework", 5);  
7  
8 SELECT SUM (minutes) FROM todo_list  
9
```

The database schema for the `todo_list` table is shown as follows:

todo_list	4 rows
id (PK)	INTEGER
item	TEXT
minutes	INTEGER

The query results show the SUM (minutes) as 70. A congratulatory message states: "Congratulations! You earned 2100 points!". The interface includes a sidebar with navigation links, a search bar, and a progress indicator.

Abbildung 3 - ToDo-List

1.1.4 Projekt: Design a store database

The screenshot shows the Khan Academy interface for a project titled "Spin-off of 'Project: Design a store database'". The project is part of a series on SQL basics. The user is asked to create a store table with columns for id, name, lagerplatz, gewicht_kg, and preisEUR. The user has entered the following SQL code:

```
1 CREATE TABLE Store_Items (id INTEGER PRIMARY KEY, name TEXT,  
2 lagerplatz TEXT, gewicht_kg NUMERIC, preisEUR NUMERIC);  
3 INSERT INTO Store_Items VALUES  
4 ("1", "Kugelschreiber", "R4", 0.2, 1.99),  
5 ("2", "Block", "R9", 0.5, 1.50),  
6 ("3", "Farbstiftset", "R2", 0.6, 1.39),  
7 ("4", "Papierliniert", "R5", 0.9, 0.99),  
8 ("5", "Papierunliniert", "R5", 0.9, 1.50),  
9 ("6", "Papierblank", "R5", 0.9, 10.99),  
10 ("7", "Papierdiverbunt", "R5", 0.9, 14.59),  
11 ("8", "Bleistiftkatl", "R3", 0.1, 12.59),  
12 ("9", "Kreide", "R9", 0.6, 1.29),  
13 ("10", "Klarsichtfolien", "R7", 0.5, 1.19),  
14 ("11", "OrdnerA4", "R7", 1.0, 1.49),  
15 ("12", "Schnellhefter", "R8", 0.8, 14.28),  
16 ("13", "FinelinerSet", "R2", 0.6, 13.58),  
17 ("14", "Radiergummi", "R1", 0.5, 1.79),  
18 ("15", "Fühlfeder", "R4", 0.8, 3.29);  
19  
20 SELECT * FROM Store_Items  
21 ORDER BY preisEUR DESC;  
22  
23 SELECT AVG(preisEUR) FROM Store_Items;  
24 SELECT MIN(preisEUR) FROM Store_Items;
```

The database schema for the `Store_Items` table is shown as follows:

Store_Items	15 rows
id (PK)	INTEGER
name	TEXT
lagerplatz	TEXT
gewicht_kg	NUMERIC
preisEUR	NUMERIC

The query results show the following data:

id	name	lagerplatz	gewicht_kg	preisEUR
6	PapierBlank	R5	0.9	19.99
7	Papierdiverbunt	R5	0.9	14.5
12	Schnellhefter	R8	0.8	14.28

The interface includes a sidebar with navigation links, a search bar, and a progress indicator.

Abbildung 4 - Projekt: Design a store database - Screensh.1

Abbildung 5 - Projekt: Design a store database - Screensh.2

Abbildungsverzeichnis

Abbildung 1 - Book List Challenge	3
Abbildung 2 - Box Office	3
Abbildung 3 - ToDo-List	4
Abbildung 4 - Projekt: Design a store database - Screensh.1	4
Abbildung 5 - Projekt: Design a store database - Screensh.2	5