

Ex5_LuisZüttel_GionRubitschung_D1P

March 20, 2024

1 Aufgabe 1 (Preliminaries)

This task requires a working installation of SQLite. The way to install it depends on your platform, but it is usually very easy. For Ubuntu 20.04 it's just `sudo apt install sqlite3`

```
[ ]: %load_ext sql
```

2 Aufgabe 2 (University Schema)

Laden Sie die gegebene SQLite-Datenbank `university.db`

1. Schauen Sie sich das Schema an:

```
`.tables`
```

```
`.schema --indent`
```

2. Schauen Sie sich die Instanz an (die Daten in jeder Tabelle):

```
select * from <table-name>;
```

3. Vergleichen Sie mit dem in der Vorlesung dazu gegebenen Schema. Verstehen Sie die Bedeutung der Datensätze in allen diesen Tabellen, die Primary Keys und die Foreign-Key-Beziehungen zwischen den Tabellen. Beantworten Sie insbesondere folgende Fragen:

- (a) Wie viele `instructors` kann ein `student` als `advisor` haben? Warum?

Entweder kein oder ein Instruktor. Die `advisor` Tabelle ist eine Zwischentabelle die als eine N zu M relation fungiert, was heisst jeder Student hat entweder genau einen oder keinen Instruktor.

- (b) Wieviele `students` kann ein `instructor` als `advisor` haben? Warum?

```
CREATE TABLE advisor(  
  s_ID  varchar(5),  
  i_ID  varchar(5),  
  primary key(s_ID),  
  foreign key(i_ID) references instructor(ID)    on delete set null,  
  foreign key(s_ID) references student(ID)       on delete cascade  
);
```

Der `instructor` kann der `advisor` für beliebig viele `students` sein. Weil der Primary Key sich nur auf die `students.ID` bezieht und unique sein muss. Diesselbe `instructor.ID` kann mehrfach aufkommen in Datensätzen als `advisor.i_ID`

(c) Warum gibt es keine Foreign-Key-Beziehung von `section` auf `time_slot`?

Das Attribut `time_slot_id` ist in beiden Tabellen `section` und `time_slot` enthalten, jedoch ist es kein Primary Key in der `time_slot` Tabelle, demnach kann es auch keine Foreign-Key-Beziehung geben.

3 Aufgabe 3 (Abfragen mit `publications.db`)

Es ist das umseitige Schema gegeben. Die zugehörige SQLite-Datenbank ist in `publications.db`. Geben Sie SQL-Queries für alle folgenden Aufgaben an. Testen Sie die Queries mit dieser Datenbank.

```
[ ]: %sql sqlite:///publications.db
```

Connecting to 'sqlite:///publications.db'

1. Finden Sie alle Autoren, die in San Francisco wohnen.

```
[ ]: %%sql

select au_lname as lastname, au_fname as firstname from authors where city =
↳ 'San Francisco';
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+-----+
| lastname | firstname |
+-----+-----+
| Locksley | Chastity  |
+-----+-----+
```

2. Wieviele Titel beginnen mit 'S'?

```
[ ]: %%sql

select count(*) as titles_starting_with_s from titles where title like 'S%';
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+
| titles_starting_with_s |
+-----+
| 4                       |
+-----+
```

3. Bestimmen Sie den durchschnittlichen Preis eines Titels.

```
[ ]: %%sql

select avg(price) as average_price from titles;
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+
| average_price |
+-----+
|    14.76625   |
+-----+
```

4. Geben Sie das Datum aller Verkäufe des Ladens 'Bookbeat' an.

```
[ ]: %%sql

select sales.date from sales
  join stores on sales.stor_id = stores.stor_id
where stores.stor_name = 'Bookbeat';
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+
|          date          |
+-----+
| 1986-07-13 00:00:00 |
| 1989-05-23 00:00:00 |
| 1988-01-13 00:00:00 |
| 1991-03-20 00:00:00 |
| 1991-03-20 00:00:00 |
| 1991-03-20 00:00:00 |
+-----+
```

5. Geben Sie alle Titel aus, die im Laden 'Bookbeat' verkauft wurden.

```
[ ]: %%sql

select distinct titles.title from titles
  join salesdetail on titles.title_id = salesdetail.title_id
  join stores on salesdetail.stor_id = stores.stor_id
where stores.stor_name = 'Bookbeat';
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+
|                                title                                |
+-----+
|      Prolonged Data Deprivation: Four Case Studies      |
| Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean |
|                      Life Without Fear                      |
|                  The Gourmet Microwave                     |
|               Straight Talk About Computers                 |
|           You Can Combat Computer Stress!                   |
|      The Busy Executive's Database Guide                    |
|                      But Is It User Friendly?              |
+-----+
```

Emotional Security: A New Algorithm
Fifty Years in Buckingham Palace Kitchens

Truncated to displaylimit of 10.

6. Finden Sie Titel und Preis des teuersten Buches.

```
[ ]: %%sql

select min(price) as cheapest, max(price) as most_expensive from titles;
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+-----+
| cheapest | most_expensive |
+-----+-----+
|    2.99  |    22.95      |
+-----+-----+
```

7. Finden Sie sämtliche Bücher, die teurer als das billigste Psychologie-Buch sind.

```
[ ]: %%sql

select title from titles
where price > (
    select min(price) as cheapest from titles
    where type like 'psychology%'
);
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+-----+
| title |
+-----+-----+
| Secrets of Silicon Valley |
| The Busy Executive's Database Guide |
| Emotional Security: A New Algorithm |
| Prolonged Data Deprivation: Four Case Studies |
| Cooking with Computers: Surreptitious Balance Sheets |
| Silicon Valley Gastronomic Treats |
| Sushi, Anyone? |
| Fifty Years in Buckingham Palace Kitchens |
| But Is It User Friendly? |
| Is Anger the Enemy? |
+-----+-----+
```

Truncated to displaylimit of 10.

8. Finden Sie die Autoren, die in einem Staat wohnen, in dem es keinen der erfassten Läden gibt.

```
[ ]: %%sql

with states_without_stores(state) as (
    with authors_states(state) as (select distinct(a.state) from authors as a)
    select au_st.state from stores as stor
    right join authors_states as au_st on stor.state = au_st.state
    where stor.state is null
)
select a.au_fname as first_name, a.au_lname as last_name, a.state
from authors as a, states_without_stores
where a.state = states_without_stores.state;
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+-----+-----+
| first_name | last_name | state |
+-----+-----+-----+
| Albert    | Ringer   | UT    |
| Anne      | Ringer   | UT    |
| Michel    | DeFrance | IN    |
| Sylvia    | Panteley | MD    |
| Morningstar | Greene   | TN    |
| Innes     | del Castillo | MI  |
| Meander   | Smith    | KS    |
+-----+-----+-----+
```

9. Geben Sie die Städte an, in denen es sowohl Autoren wie auch Verleger gibt.

```
[ ]: %%sql

with publisher_cities(city) as (
    select distinct(city) from publishers
),
authors_cities(city) as (select distinct(a.city) from authors as a)
select authors_cities.city from authors_cities
join publisher_cities on authors_cities.city = publisher_cities.city;
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+
| city |
+-----+
| Berkeley |
+-----+
```

10. Bestimmen Sie alle Bücher, die den gleichen Typ besitzen wie das Buch 'Net Etiquette'.

```
[ ]: %%sql
```

```
select title from titles
where type like (
    select t.type from titles as t
    where t.title like 'Net Etiquette'
);
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+
|          title          |
+-----+
| Secrets of Silicon Valley |
| But Is It User Friendly? |
|      Net Etiquette      |
+-----+
```

11. Geben Sie einen SQL Ausdruck an, der die Büchertypen zusammen mit der Anzahl Bücher jedes Typs ausgibt, von denen es mehr als 2 verschiedene Bücher gibt.

```
[ ]: %%sql

select t.type, count(t.type) as number_of_books
from titles as t
group by t.type;
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+-----+
|    type    | number_of_books |
+-----+-----+
| UNDECIDED  |          1      |
| business   |          4      |
| mod_cook   |          2      |
| popular_comp |          3      |
| psychology |          5      |
| trad_cook  |          3      |
+-----+-----+
```

12. Geben Sie einen SQL Ausdruck an, der die Büchertypen zusammen mit der Anzahl Bücher jedes Typs ausgibt, von denen es mehr als 2 verschiedene Bücher gibt.

```
[ ]: %%sql

select t.type, count(t.type) as number_of_books
from titles as t
group by t.type
having count(t.type) > 2;
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+-----+
|      type      | number_of_books |
+-----+-----+
| business       |          4       |
| popular_comp   |          3       |
| psychology     |          5       |
| trad_cook      |          3       |
+-----+-----+
```

13. Geben Sie einen SQL Ausdruck an, der die Anzahl Autoren pro Staat auflistet, wobei die Ausgabe nach Anzahl Autoren sortiert sein soll.

```
[ ]: %%sql

select a.state, count(a.au_id) as number_of_authors
from authors as a
group by a.state
order by number_of_authors desc;
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+-----+
| state | number_of_authors |
+-----+-----+
| CA    |          15       |
| UT    |           2       |
| TN    |           1       |
| OR    |           1       |
| MI    |           1       |
| MD    |           1       |
| KS    |           1       |
| IN    |           1       |
+-----+-----+
```

14. Bestimmen Sie alle Publisher, welche weniger Bücher herausgegeben haben als der Durchschnitt.

```
[ ]: %%sql

with pub_with_num_books(pub_id, pub_name, number_of_books) as (
    select p.pub_id, pub_name, count(t.title_id) as number_of_books
    from publishers as p
    join titles as t on p.pub_id = t.pub_id
    group by p.pub_id
)
select pub_name from pub_with_num_books
where number_of_books > (
    select avg(number_of_books) from pub_with_num_books
);
```

Running query in 'sqlite:///publications.db'

```
[ ]: +-----+
      |      pub_name      |
      +-----+
      | Binnet & Hardley |
      +-----+
```