M O T I V A T I O N

Die Firma "5 Stars" bewertet und vermarktet privat vermietete Ferienwohnung in Deutschland. Dazu reisen Mitarbeiter zu den Wohnungen, erstellen ein Exposee und bewerten die Wohnungen. Für die anfallenden Inlandsflüge hält die Firma eine Tabelle bereit, in der Flüge verzeichnet sind, die jeden Tag durchgeführt werden.

Öffne die Datenbank "FluegeDB" aus dem Material.

A) Finde den billigsten Flug von Hannover nach Nürnberg. Benutze den Standardfilter und die Sortierung.

B) Finde den spätesten Flug von Düsseldorf nach Münster mithilfe des Standardfilters und der Sortierung.

C) Kann man die Anzahl aller Flüge ermitteln, die von "Quick Lift" von Bremen aus durchführt werden.

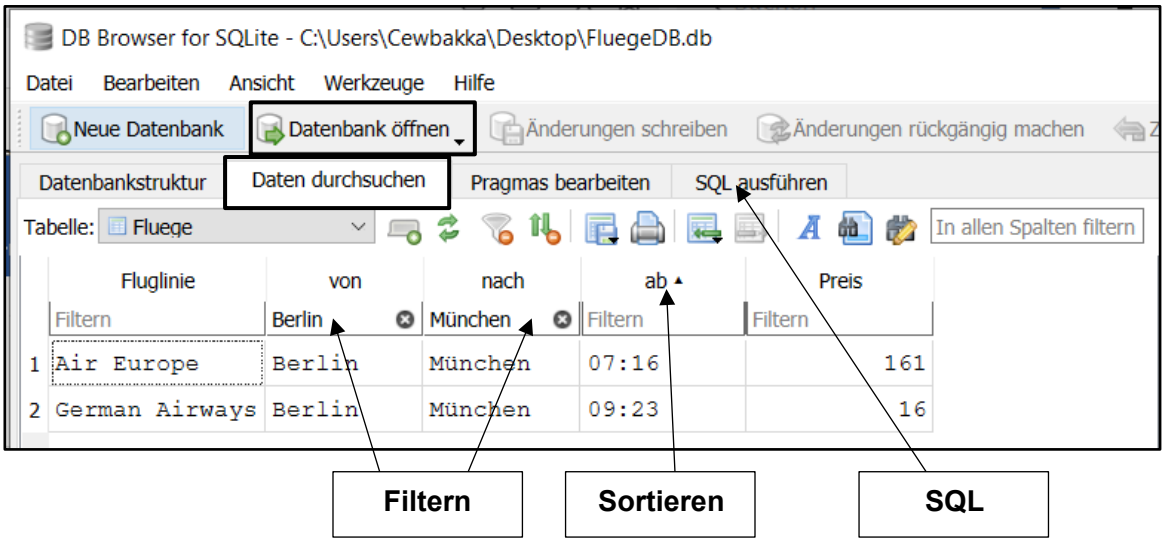
Ü B E R S I C H T

Hier wird der **DB Browser für SQLite-Datenbanken** benutzt, der unter **https://sqlitebrowser.org** downloadbar ist. Die Screenshots beziehen sich auf dieses Programm.

> *Ebenso gut funktioniert **SQLiteStudio**. Auch hier kann einfach gefiltert und sortiert werden, auch die Übersicht über alle Tabellen sowie die SQL-Abfrage funktioniert „genau so“ wie im hier benutzten DB Browser.*

Unter **„Datenbank öffnen“** kann eine bestehende SQLite-Datenbank geöffnet und der Inhalt einzelner Tabellen unter **„Daten durchsuchen“** angesehen werden.

Es gibt **Filter** und **Sortiermöglichkeiten**.

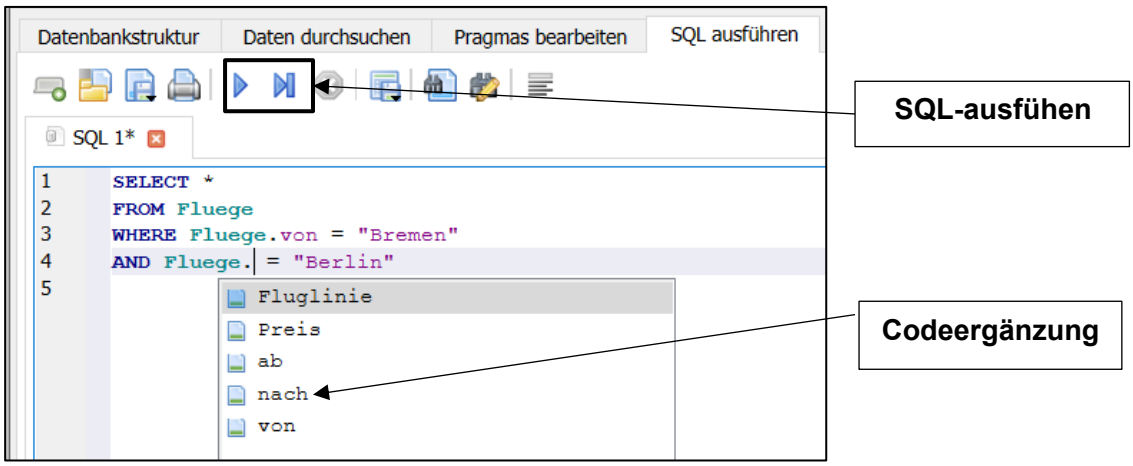


Schreibweise per Hand: von = „Berlin“ AND nach = „München“, ORDER BY von

Eine Tabellenzeile nennt sich **Datensatz**, der aus mehreren ***Datenfeldern*** besteht.

Jede Spalte wird durch ein **Attribut** beschrieben (Bsp.: Fluglinie oder von oder nach)

SQL-Abfragen können unter SQL-ausführen ausgeführt werden. Dabei besitzt der Datenbankbrowser eine sehr praktische **Codeergänzung**.



Eine **SQL-Anweisung** enthält immer
1.) Die **Tabellenspalten**, die später angezeigt werden,
2.) die **Tabellennamen**, aus denen ausgewählt wird,
3.) die **Abfragekriterien**.

SQL	*Beispiel 1*	*Beispiel 2*
SELECT Spalten	SELECT *	SELECT Fluege.Fluglinie
FROM Tabellen	FROM Fluege	FROM Fluege
WHERE Bedingung	WHERE Fluege.von = ‚Bremen‘	WHERE Fluege.Preis < 20
	AND Fluege.nach = 'Berlin‘	OR Fluege.Preis > 50

Beachte, dass hier **„Zeichenketten“ (Textvariablen)** in Hochkommas stehen, Tabellennamen und Spalten nicht. Integer ebenfalls nicht.

A N W E N D E N

Aufgabe 1 *(FluegeDB - Filter und Sortierung)*
Wende den **Standardfilter** an und schreibe jeweils die **Verknüpfungskriterien** des Filters auf, um alle entsprechenden Flüge anzeigen zu lassen. Ermittle auch durch Abzählen.

a) den billigsten Flug von Hannover nach Nürnberg,
b) den spätesten Flug von Düsseldorf nach Münster,
c) die Anzahl aller Flüge, die "Quick Lift" von Bremen aus durchführt,
d) die Anzahl der Flüge von Berlin aus nach 20:00 Uhr,
e) die Anzahl aller Flüge aller Städte, die das Nachtflugverbot (23:00-06:00) verletzen (→ werden Uhrzeiten korrekt verglichen?),

f) HA die Anzahl der Flüge unter 20 Euro,
g) HA die Anzahl der Flüg von Nürnberg aus unter 20 Euro.
h) HA Gibt es einen Flug von Leipzig nach Berlin nach 19:00 Uhr?

Aufgabe 2 *(FluegeDB - SQL)*
Erstelle zu den folgenden Fällen die **SQL-Anweisung**, um alle entsprechenden Flüge anzeigen zu lassen. Ermittle dann durch Abzählen. Die Abfragen sind absichtlich dieselben wie bei der vorherigen Aufgabe.

a) den billigsten Flug von Hannover nach Nürnberg,
b) den spätesten Flug von Düsseldorf nach Münster,
c) die Anzahl aller Flüge, die "Quick Lift" von Bremen aus durchführt,
d) die Anzahl der Flüge von Berlin aus nach 20:00 Uhr,
e) die Anzahl aller Flüge aller Städte, die das Nachflugverbot (23:00-06:00) verletzen (→ werden Uhrzeiten korrekt verglichen?),

f) HA die Anzahl der Flüge unter 20 Euro,
g) HA die Anzahl der Flüg von Nürnberg aus unter 20 Euro.
h) HA Gibt es einen Flug von Leipzig nach Berlin nach 19:00 Uhr?

Aufgabe 3 *(FluegeDB)*
Ermittle die folgenden Fragestellungen mithilfe von SQL-Anweisungen und schreibe die SQL-Anweisungen auf. Benutze dabei: **ORDER BY** *Kriterium* **ASC** / **ORDER BY** *Kriterium* **DESC** (aufsteigend / absteigend).

```
SELECT *
FROM Fluege
ORDER BY Fluege.Preis ASC
```

Lies dann das Ergebnis aus der Tabelle ab.

a) Ermittle die Stadt mit dem frühesten Start,
b) die Airline mit dem teuersten Flug,
c) die Airline mit dem billigsten Flug der in Berlin startet,
d) Ermittle den billigsten Flug von Berlin nach München.

e) HA Ermittle die Stadt mit dem spätesten Start,
f) HA die Airline mit dem billigsten Flug,
g) HA die Airline mit dem billigsten Flug der in Bremen startet,
h) HA ermittle den billigsten Flug von Dresden nach Bremen

Aufgabe 4 *(FluegeDB)*
a) Überprüfe mit einer einzigen SQL-Abfrage, ob es Verbindungen "Berlin - München" mit Hin- und Rückflug an einem Tag gibt.
b) Überprüfe mit einer einzigen SQL-Abfrage, ob es möglich ist, am selben Tag von Bremen nach Köln/Bonn und weiter nach Dresden zu fliegen.
c) Überprüfe mit einer einzigen SQL-Abfrage, ob es möglich ist, am selben Tag von Berlin nach München und wieder zurück mit derselben Airline zu fliegen.

d) HA Überprüfe mit einer einzigen SQL-Abfrage, ob es möglich ist, am selben Tag von Berlin nach Köln/Bonn und weiter nach Münster zu fliegen.
e) HA Überprüfe mit einer einzigen SQL-Abfrage, ob es möglich ist, am selben Tag von Stuttgart nach Bremen und wieder zurück mit derselben Airline zu fliegen.

Aufgabe 5 *(StreamingDatenDB)*
Öffne "StreamingDatenDB" und sieh die Daten in der Tabelle an. Die Anzahl der Datensätze kann mit dem Zusatz

```
SELECT COUNT(*)
FROM Nutzerdaten
WHERE Nutzerdaten.Alter > 30
```

ermittelt werden. Erstelle jeweils eine SQL-Anweisung:

a) der Anzahl der "Premium"-Abonnenten,
b) der Anzahl der "Mini"- oder "Test"-Abonnenten,
c) der Anzahl der Abonnenten mit 50 Jahren oder darüber,
d) der Anzahl der Abonnenten über 30 Jahren, die ein "Test"-Abo haben,

e) HA Anzahl der Abonnenten, die zwischen 30 und 50 sind und ein XL Abo haben,
f) HA die Abonnementform mit den meisten Teilnehmern an. Schreibe für jedes Abo eine neue SQL-Abfrage.

Aufgabe 6 *(StreamingDatenDB)*
AVG, MIN und MAX berechnen den Durchschnitt, das Minimum und das Maximum.

```
SELECT AVG (Fluege.Preis)
FROM Fluege

SELECT MIN (Fluege.Preis)
FROM Fluege

SELECT MAX (Fluege.Preis)
FROM Fluege
```

a) Berechne das Durchschnittsalter eines Premiumabonnenten,
b) Berechne das geringste Alter aller Basic-Abonnenten,
c) Finde das Alter des ältesten Pro-Abonnenten.

d) HA Berechne die Durchschnittsrestlaufzeit der Test-Abos.
e) HA Berechne die geringste Restlaufzeit der Premium-Abos.
f) HA Berechne die größte Restlaufzeit der Basic-Abos.
g) HA Berechne das Durchschnittsalter der Kunden des XL- oder Premium-Abos.