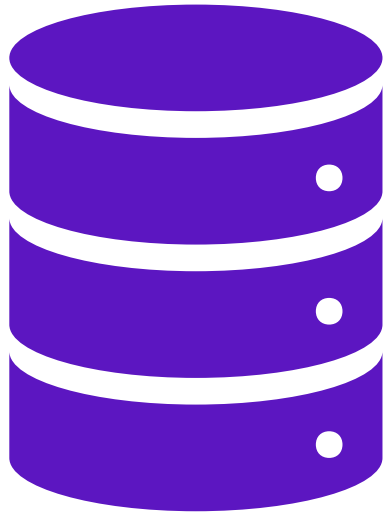


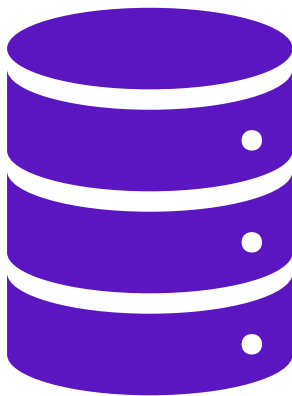
Datenbanken und SQL



Datenbanksystem = System zur Datenverwaltung

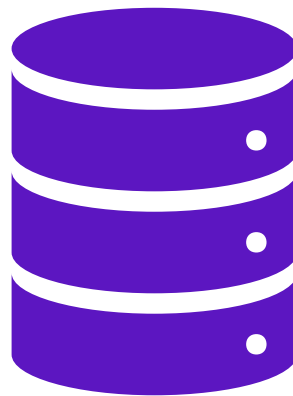
besteht aus Datenbank und der
Verwaltungssoftware
(Datenbankmanagementsystem = DBMS)

Relationale Datenbanken



Abfragesprache: SQL
(Structured Query Language)

Nicht-Relationale / NoSQL Datenbanken



Sammelbegriff für verschiedene Systeme
(NoSQL = Not only SQL)

- Dokumentenorientierte Datenbanken
- Graphdatenbanken
- Key-Value-Stores
- Spaltenorientierte Datenbanken
- ...

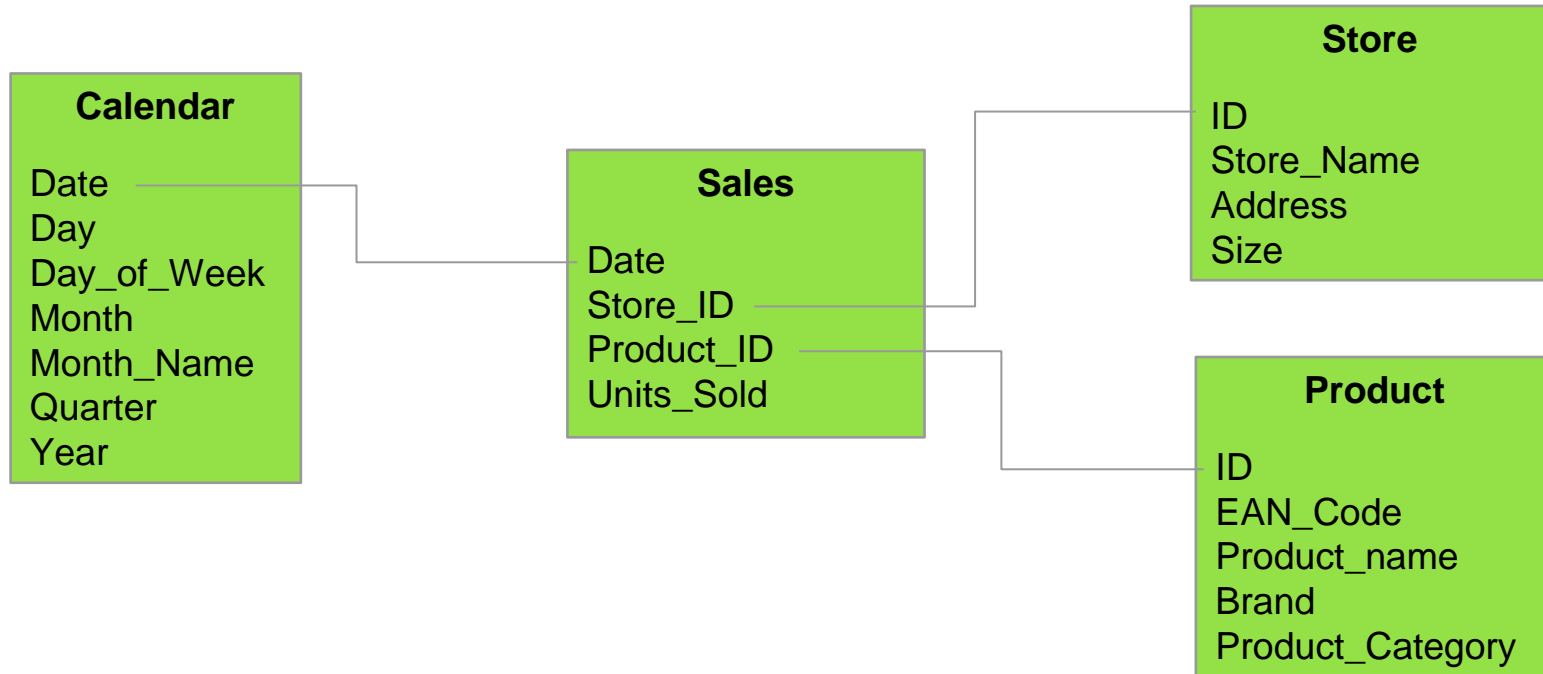
1

Relationale Datenbanken

- Die zentralen Objekte einer relationalen Datenbank sind Tabellen
- Mathematisch nennt man eine Tabelle eine **Relation**, Zeilen heißen **Tupel** und Spalten **Attribute**. Operationen auf der Datenbank (Filterung, Vereinigung, Modifikationen, ...) bilden die sogenannte **relationale Algebra**.
- Ein Datensatz (Zeile) muss eindeutig über Schlüssel (Keys) identifizierbar sein

Fall	Patient	Klinik	Aufnahmedatum	Entlassdatum	Diagnose_ID
1	1234	11	04.04.2022 09:34	06.04.2022 12:50	654
2	1235	11	04.04.2022 11:10	07.04.2022 09:00	687
3	1236	12	04.04.2022 08:40	10.04.2022 11:28	987
4	1237	12	04.04.2022 07:03	NULL	123

Relationale Datenbanken werden als ERD (entity relationship diagram) visualisiert



Es gibt viele SQL Datenbanksysteme, die ihre Vor- und Nachteile haben. Die TOP 10 laut des [DB-Engines Ranking](#) sind:

- Oracle Database
- MySQL
- Microsoft SQL Server
- **PostgreSQL**
- IBM Db2
- Microsoft Access
- **SQLite**
- MariaDB
- Snowflake
- Microsoft Azure SQL Database

Die 3 großen Cloud-Anbieter

Amazon AWS
Microsoft Azure
Google Cloud

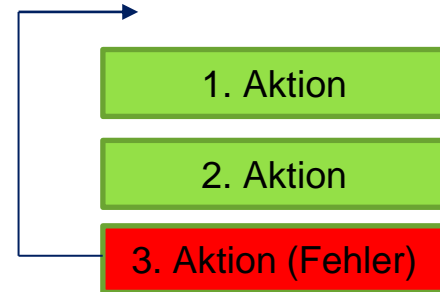
SQL ist die universelle Sprache für relationale Datenbanken. Es gibt jedoch verschiedene Dialekte!

- ACID (deutsch AKID) beschreibt gewünschte Eigenschaften von Transaktionen:
 - **A**tomarität/Abgeschlossenheit: Ganz oder gar nicht
 - **K**onsistenz: Transaktion hinterläßt konsistenten Datenbankzustand
 - **I**solation: Nebenläufige Transaktionen beeinflussen sich nicht (Sperrverfahren)
 - **D**auerhaftigkeit: dauerhafte Speicherung
- Alle relationalen DBMS sind ACID-konform
- wesentlicher Unterschied zu NoSQL

Atomarität/Abgeschlossenheit: Ganz oder gar nicht

Beispiel: Überweisung bei einer Bank.

Abbuchung in der einen Tabelle, Gutschrift in einer anderen



Tritt ein Fehler auf, wird ein "Rollback" gemacht und der Zustand vor der Transaktion wiederhergestellt

Schlüsselwort COMMIT schließt eine Transaktion ab (je nach DBMS auch automatisch)

Konsistenz: Transaktion hinterläßt konsistenten Datenbankzustand

Integritätsbedingungen (z.B. eindeutiger Schlüssel, Feld nicht leer, ...) sind auch nach der Transaktion noch erfüllt

Isolation: Nebenläufige Transaktionen beeinflussen sich nicht (Sperrverfahren)

Liest eine Transaktion gerade von einer Tabelle, kann in diese nicht geschrieben werden

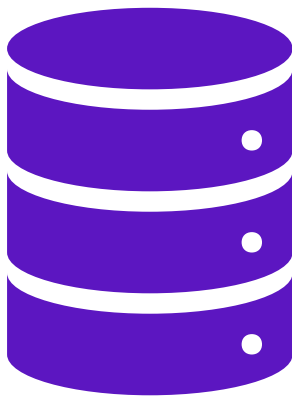
Dauerhaftigkeit: dauerhafte Speicherung

Garantierte Speicherung, auch bei Systemfehlern. Das wird durch einen Transaktionslog (d.h. Protokoll) realisiert.

2

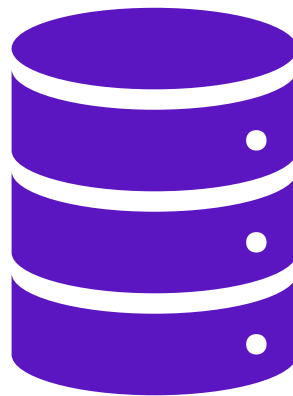
NoSQL Datenbanken

Relationale Datenbanken



Abfragesprache: SQL
(Structured Query Language)

Nicht-Relationale / NoSQL Datenbanken



Sammelbegriff für verschiedene Systeme
(NoSQL = Not only SQL)

- Dokumentenorientierte Datenbanken
- Graphdatenbanken
- Key-Value-Stores
- Spaltenorientierte Datenbanken
- ...

Big Data wird durch 3-6 Vs definiert

- **Volume:** Menge der Daten
- **Velocity:** Geschwindigkeit, mit der Daten erzeugt werden
- **Variety:** Vielfalt der Daten
- *Veracity: Echtheit*
- *Value: unternehmerischer Mehrwert*
- *Validity: Datenqualität*



Was sind Vor- und Nachteile von SQL-Datenbanken?

Was sind Vor- und Nachteile von SQL-Datenbanken?

- Starre Tabellenform
- Schreiben von Daten relativ "langsam"
- Skalierbarkeit
- Strukturen/Zusammenhänge nicht einfach abbildbar

NoSQL versucht, diese Einschränkungen aufzuheben, v.a. durch verteilte Systeme, hat dafür aber andere Einschränkungen

- Schlüssel-Wert-Paare wie Dictionaries in Python
- hohe Performanz und Skalierbarkeit, häufig in-memory
- besonders geeignet für Zwischenspeicher (Cache), Streaming, Message Broker
- z.B. für Warenkörbe in Online-Shops oder Session-Daten
- Redis ist eines der am häufigsten eingesetzten Key-Value-Stores

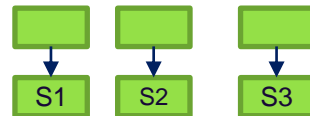


- Statt Zeilen in Tabellen besteht ein Eintrag einer dokumentenorientierten Datenbank aus einem Dokument
- Das Dokument kann strukturierte Informationen enthalten (z.B. JSON) oder auch eine Binärdatei sein (z.B. Video)
- MongoDB und CouchDB sind Datenbanken für JSON-ähnliche Dokumente und die weitest verbreitetsten NoSQL-Datenbank

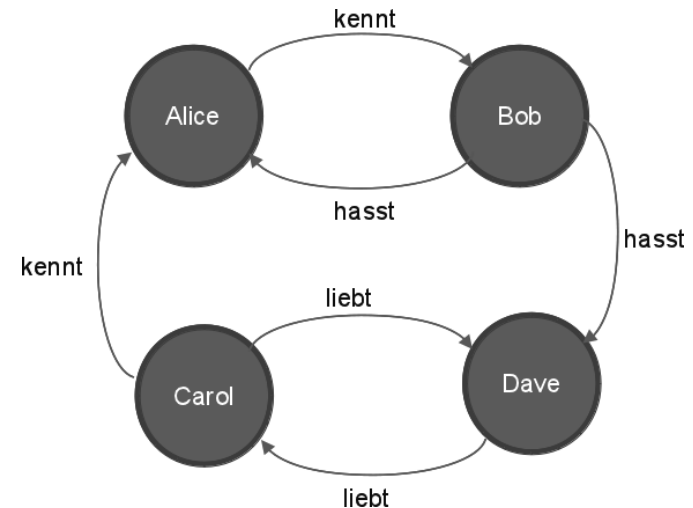
JSON-Datei

```
{
  "Mitarbeiter": [
    {
      "Vorname": "Silke",
      "Nachname": "Müller",
      "Alter": 30,
      "Adresse": {
        "Straße": "Heinealle 23a",
        "PLZ": "53177",
        "Ort": "Bonn",
        "aktiv": true
      },
      "Vorname": "Anna",
      "Nachname": "Djawi",
      "aktiv": false
    },
    {
      "Vorname": "Peter",
      "Nachname": "Schmitt",
      "aktiv": null
    }
  ]
}
```

- Jedes Dokument kann anders aussehen, z.B. unterschiedliche Felder haben
- gleichzeitiges Schreiben auf mehrere Server
- Replikationen für bessere Verfügbarkeit
- z.B. MongoDB entscheidet sich für Konsistenz, während CouchDB die Verfügbarkeit bevorzugt



- Graphdatenbanken werden genutzt, um stark vernetzte Informationen darzustellen und abzuspeichern.
- Ein Graph besteht aus Knoten und Kanten
- [Neo4j](#) ist beliebteste Graphdatenbank



- spaltenorientierte Datenbanken speichern die Daten im Gegensatz zu relationalen Datenbanken nicht zeilenweise, sondern spaltenweise
- Performance-Vorteile bei Bearbeitung vieler Zeilen und weniger Spalten
- Komprimierung spaltenweise möglich
- Skalierbarkeit durch Verteilen der Spalten auf einzelne Server
- Apache Cassandra ist die populärste spaltenorientierte Datenbank

Patient	Klinik	Aufnahmedatum	Entlassdatum	Diagnose_ID
1234	11	04.04.2022 09:34	06.04.2022 12:50	654
1235	11	04.04.2022 11:10	07.04.2022 09:00	589
1236	12	04.04.2022 08:40	10.04.2022 11:28	987
1237	12	04.04.2022 07:03	NULL	123

```
{  
  Patient: [1234, 1235, 1236, 1237],  
  Klinik: [11, 11, 12, 12],  
  Aufnahmedatum: ["04.04.2022 09:34", "04.04.2022 11:10",  
                  "04.04.2022 08:40", "04.04.2022 07:03"],  
  Entlassdatum: ["06.04.2022 12:50", "07.04.2022 09:00",  
                 "10.04.2022 11:28", NULL],  
  Diagnose_ID: [654, 687, 987, 123]  
}
```

CAP-Theorem für verteilte Systeme: Consistency, Availability und Partition Tolerance nie alle drei gleichzeitig erfüllbar (nur zwei)

- **Consistency (Konsistenz):** Jeder Kunde sieht das gleiche, egal auf welchen Knoten er zugreift. D.h. in verteilten Systemen muss sichergestellt werden, dass nach Abschluss einer Transaktion alle Replikate aktualisiert werden (Achtung, es ist nicht die Konsistenz von ACID gemeint)
- **Availability (Verfügbarkeit):** Jede Anfrage an jeden Knoten wird beantwortet, gemeint ist hier eine akzeptable Antwortzeit
- **Partition tolerance (Partitionstoleranz):** System arbeitet weiter, auch wenn Verbindungen gestört sind, z.B. ein Server ausgefallen ist

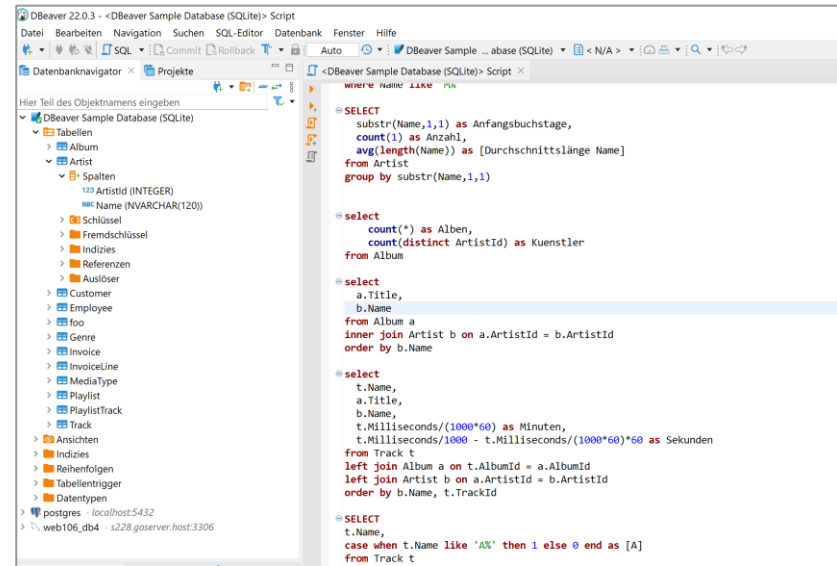
CAP-Theorem für verteilte Systeme: Consistency, Availability und Partition Tolerance nie alle drei gleichzeitig erfüllbar (nur zwei)

- **CP Datenbank:** Wenn eine Partition (Trennung eines Knotennetzwerks von einem anderen) passiert, dann werden die nicht-konsistenten Knoten abgeschaltet, d.h. sie sind dann nicht mehr verfügbar
- **AP Datenbank:** Wenn eine Partition passiert, dann bleiben alle Knoten verfügbar. Allerdings ist ggf. nicht an allen Knoten der aktuelle Stand abrufbar
- **CA Datenbank:** keinen praktischen Nutzen, da Partitionsfehler in Netzwerken unvermeidlich sind

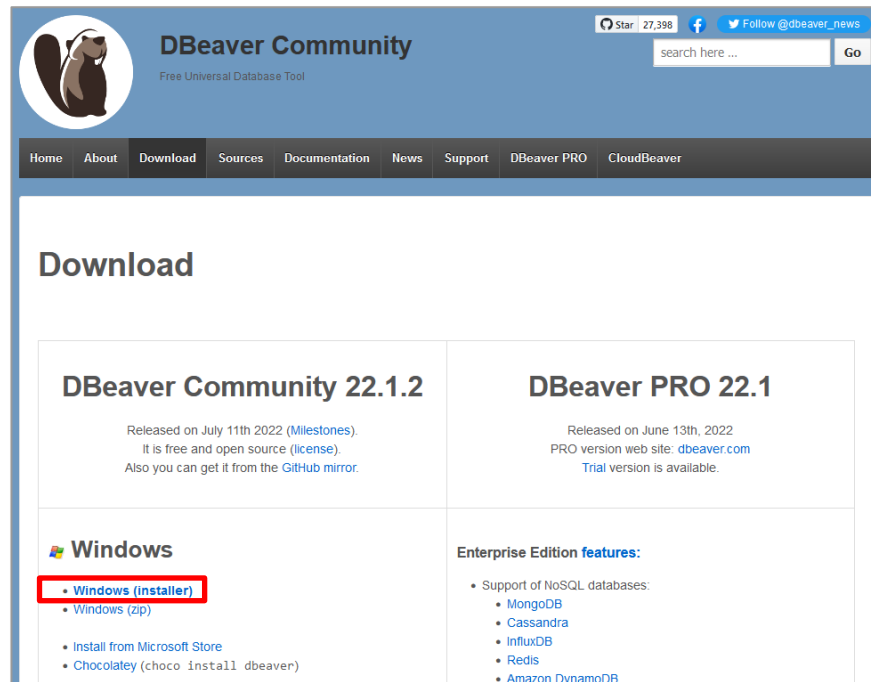
3

DBeaver

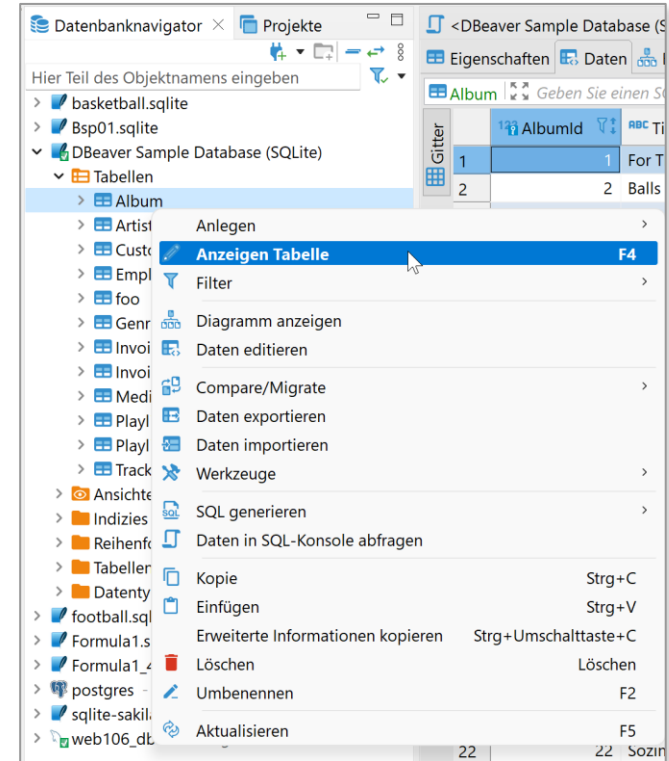
- Jede Datenbank bringt eine Möglichkeit mit, mit dieser zu kommunizieren. Bei vielen ist es allerdings nur eine Kommandozeile
- DBeaver ist ein Editor, der verschiedene Datenbank-Systeme gleichzeitig anbinden kann und eine komfortable Oberfläche bietet
- DBeaver ist kostenlos und Open-Source
- Die Anbindung der meisten Datenbanken erfolgt mittels JDBC



- Herunterladen der aktuellen Version von der Webseite dbeaver.io
- Installation
- Erstellen der Beispieldatenbank (Chinook, SQLite)
- Auswahl einer Tabelle, Rechtsklick, Anzeigen Tabelle

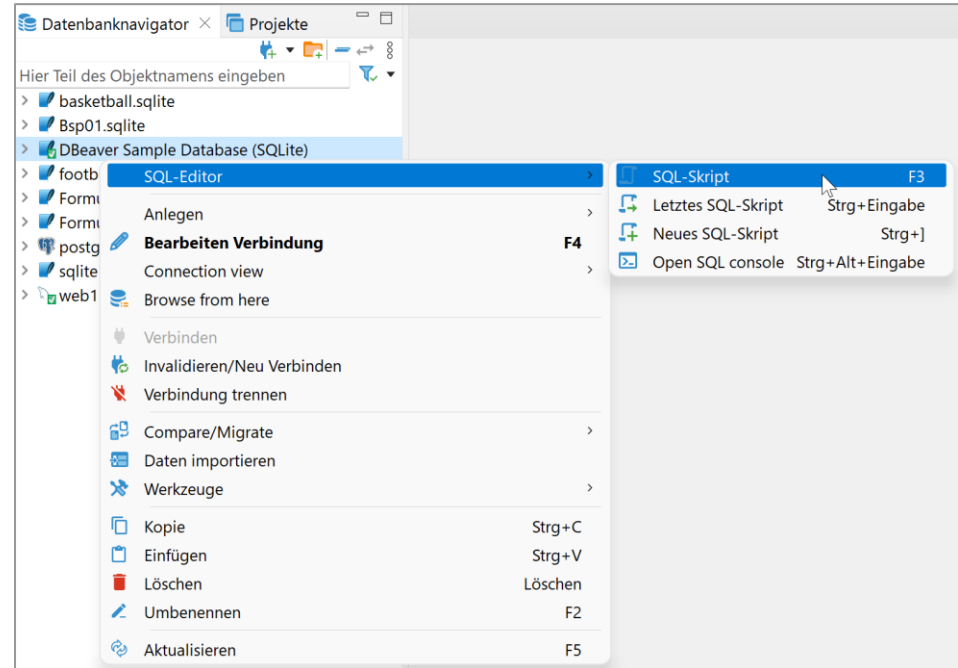


- Auswahl einer Tabelle, Rechtsklick, Anzeigen Tabelle (F4)
- Anzeige der Eigenschaften, Daten und ER Diagram
- Anzeige des gesamten ER Diagramms:
Rechtsklick auf Tabellen, Anzeige Tabellen



- Erstellen eines SQL-Skripts:
 - F3 bzw. Rechtsklick auf Datenbank, SQL-Editor, SQL-Skript
- Alternativ über Menü (SQL Editor > SQL-Skript)
- Alternativ: Datei > Neu > Allgemein > Datei
- Markierten Code mit Strg + Enter ausführen

```
select * from album
```



- SQLite-Datei liegt auf dem PC
- Neue Verbindung: SQLite
- Dateipfad angeben
- Verbindung kann danach umbenannt werden

