



Tag 8

Einstieg DB

14. April 2023

Ablauf

- Rückblick siebter Tag
- Konsolidieren Geometrie-Übung
- Einstieg DB
- Installation
- SQL
- Projektarbeit:
Themenfindung und Gruppenbildung initiieren

Rückblick OOP & Git

Datenbanken

- Hier: Relationale Datenbanken
- Tabellen
- Schlüssel
- Normalformen
- Abfragen / Queries / Statements

Freie Datenbankserver

- **SQLite**: Klein, schnell, eigenständig (nicht geeignet für Server/Client)
- **MariaDB**: ex. MySQL, sehr weit verbreitet
- **PostgreSQL**: Am meisten Features
- Microsoft SQL Server Express:

Admin-Tools / Viewers

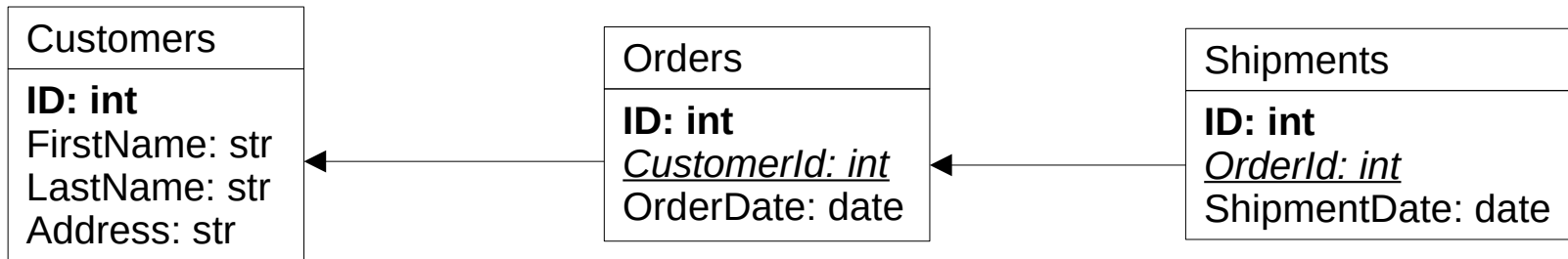
- **SQLite:** [DB Browser for SQLite](#)
- **MariaDB, PostgreSQL, SQLite:** [HeidiDB](#)

Installation

- Frei, wenn ihr eine Präferenz habt
- SQLite / DB Browser for SQLite sonst

Aufgabe: Erstellen und abfüllen

- DB ***small_shop_demo*** erstellen
- SQL-Script *create_*.sql* ausführen
- SQL-Script *fill_*.sql* ausführen



Übung SQL

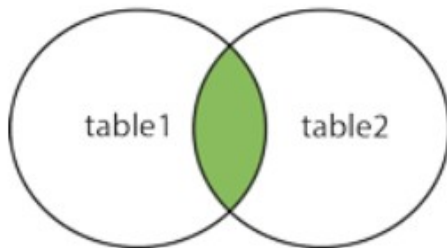
Beispiele:

- SELECT * FROM **Orders** WHERE **OrderDate** > '2022-01-31'
- INSERT INTO **Orders**(**CustomerId**, **OrderDate**) VALUES (-1, '2023-01-20 13:00:00')
- UPDATE **Orders** SET **OrderDate**='2099-01-20 13:00:00' WHERE **Id** = -1
- DELETE FROM **Orders** WHERE **Id** = -1
- Werte in Blau müssen/können angepasst werden
- **Ziel:** Referenzielle Integrität (→ Foreign Key constraints) erfahren und erklären können

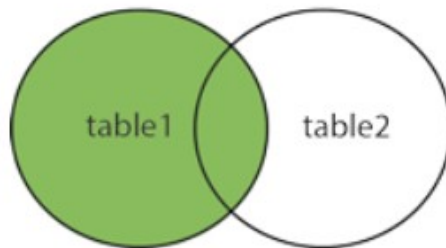
Join

- **INNER JOIN:** Einträge, die in beiden Tabellen vorkommen
 - z. B. Customers mit Orders
- **LEFT JOIN:** Einträge, die links vorkommen, und ggf. Match der rechten Tabelle
 - z. B. Customer mit und ohne Orders
- **RIGHT JOIN:** Einträge, die rechts vorkommen, und ggf. Match der linken Tabelle
 - Weniger relevant, kein Beispiel hier weil jeder Eintrag links ein Match rechts hat (→ referenzielle Integrität)
- **FULL OUTER JOIN**
 - Noch weniger relevant

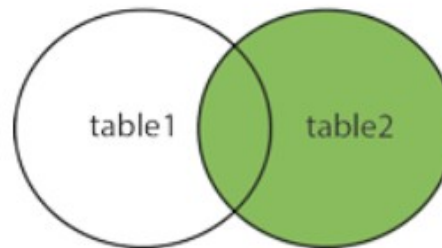
INNER JOIN



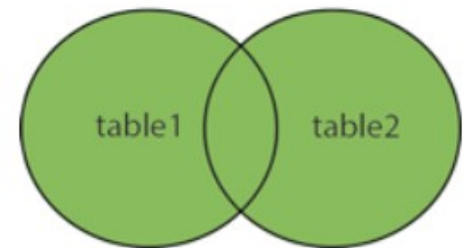
LEFT JOIN



RIGHT JOIN



FULL OUTER JOIN



Join

```
SELECT Orders.*, Customers.FirstName, Customers.LastName, Shipments.ShipmentDate, Shipments.Id  
FROM Orders
```

/ Join with Customers so we also have first- and last name of customer */*

```
INNER JOIN Customers ON Orders.CustomerId = Customers.Id
```

/ **Optionally** join with shipments.If it was already shipped, add date and id */*

```
LEFT JOIN Shipments ON Orders.Id = Shipments.OrderId
```

```
ORDER BY Customers.LastName
```

orders_and_customer.sql im sql-Ordner