Inhaltsverzeichnis

LS 4 – Info - Benötigte Programme

DB-Server → mariadb (Bestandteil von XAMPP)

SQL-Client → mysql.exe (Konsolen-Client; Bestandteil von XAMPP)

→ phpMyAdmin (Browser-Client; Bestandteil von XAMPP)

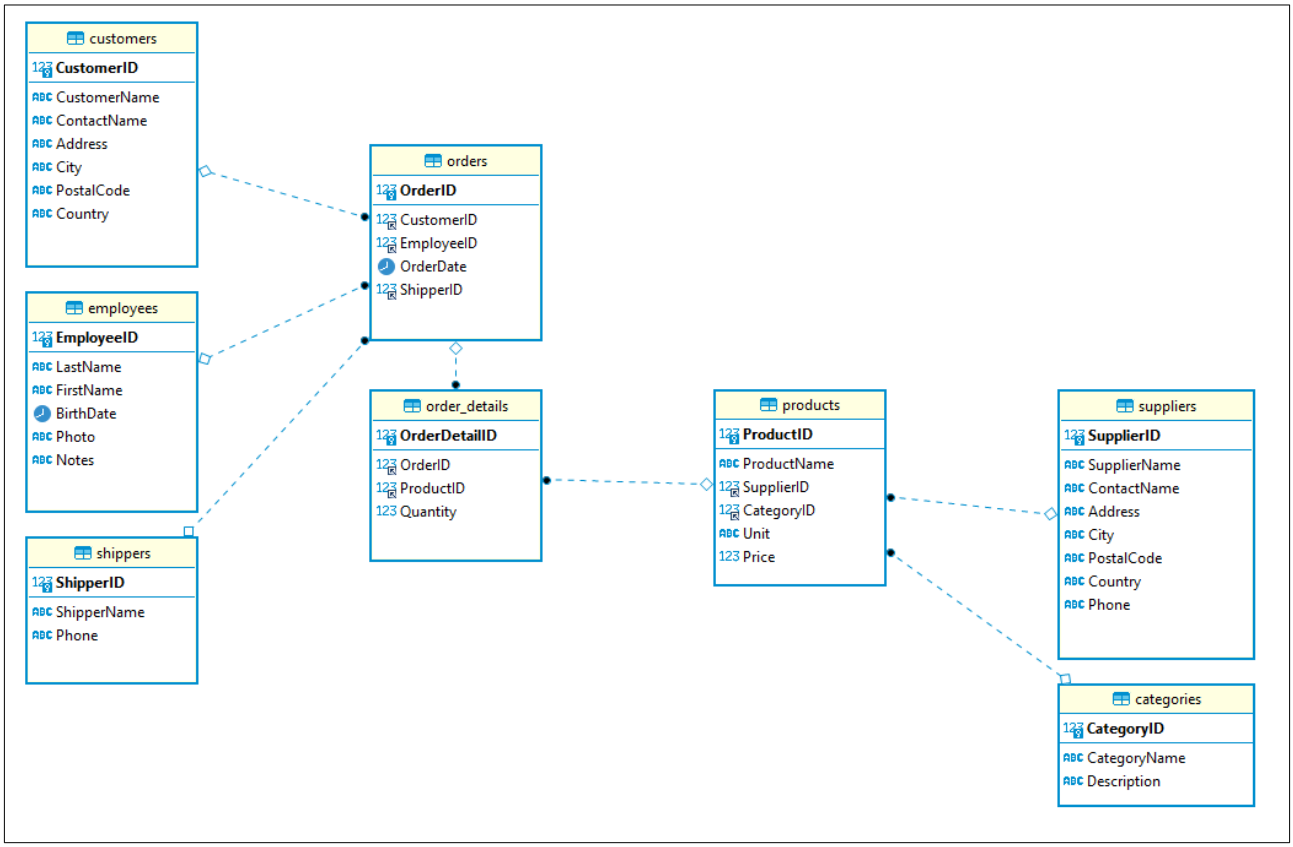
→ DBeaver (GUI-Client)

→ W3Schools (Online-Datenbank)

→ Visual Code + Extensions (Entwicklungsumgebung)

LS 4 – Info - Beispieldatenbank w3schools

`beispiel_ls4_01.sql`



LS 4 – Info – SQL

CRUD-Befehle

- **C**reate → INSERT
- **R**ead → SELECT
- **U**pdate → UPDATE
- **D**elete → DELETE

Beispiele `beispiel_ls4_02.sql`

```
SELECT CustomerID, CustomerName
      FROM customers;

INSERT INTO customers(CustomerID, CustomerName, ContactName, Country)
      VALUES(101, "Bertas Bastelstube", "Berta Braun", "Germany");

UPDATE customers
      SET postalcode = "53945"
      WHERE ContactName = "Albert Ahrhaus";

DELETE FROM customers
      WHERE CustomerID = 100;
```

SQL-und Datenbank-Hintergrundwissen

SQL (Structured Query Language)

- SQL (Structured Query Language) ist eine Datenbanksprache zur Definition, Abfrage und Manipulation von Daten.
- SQL besitzt eine relativ einfache Syntax und ist an die englische Sprache angelehnt.
- SQL ist für fast alle heute gebräuchlichen DBMS verfügbar und damit quasi ein Standard
- ! SQL wird von verschiedenen DBMS verschieden interpretiert und unterstützt.

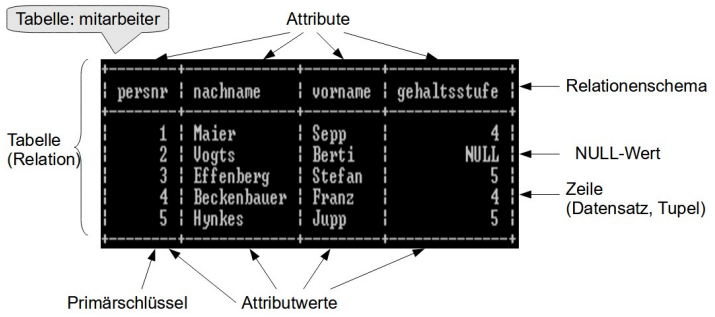
Bestandteile von SQL

- **Data Definition Language (DDL)**
  - CREATE DATABASE
  - CREATE TABLE
  - CREATE INDEX
  - DROP DATABASE
  - DROP TABLE
  - DROP INDEX
  - ALTER TABLE
- **Data Control Language (DCL)**
  - LOCK
  - UNLOCK
  - GRANT
  - REVOKE
- **Data Manipulation Language (DML)**
  - INSERT
  - LOAD
  - DELETE
  - UPDATE
  - SELECT
- **Administration & Utility Statements**
  - SET
  - SHOW
  - USE
  - OPTIMIZE
  - KILL

Funktionen
- ABS – Absolutwert
- BIN – Binärwert einer Dezimalzahl
- CURDATE – Aktuelles Datum
- CURTIME – Aktuelle Zeit
- DATABASE – Aktuelle Datenbank
- ISNULL – Test auf NULL-Wert
- div. mathematische Funktionen
- div. Datums- und Zeit-Funktionen
- div. Funktionen zur Zeichenkettenmanipulation

Aggregatfunktionen
- AVG – Mittelwert
- COUNT – Anzahl
- MAX – Maximum
- MIN – Minimum
- STD – Standardabweichung
- SUM - Summe

Grundbegriffe:
- **Daten** sind formalisierbare Informationen, die zur Verarbeitung mit einem Computer geeignet sind. Beispiele für Daten sind Nachname, Geburtsdatum, Artikelnummer, Artikelname, …
- Daten besitzen einen bestimmten **Datentypen**, der die Art der Daten beschreibt. Wichtige Datentypen sind: Zahl, Text, Datum/Uhrzeit, …
- Ein **(Record)** ist eine Zusammenfassung inhaltlich zusammengehöriger Daten. Die einzelnen Teile eines Datensatzes heißen **Datenfelder**. Beispiele: Alle zu einer Person gehörigen Daten kann man zum Datensatz Person zusammenfassen. Alle zu einem Artikel gehörenden Daten kann man zu einem Datensatz Artikel zusammenfassen.
- Eine **Datenbank** ist eine Sammlung aller zu einem Thema gehörender Daten. Eine Datenbank besteht typischerweise aus mehreren (vielen) **Tabellen** und Beziehungen zwischen diesen Tabellen. Eine Datenbank stellt immer nur einen Ausschnitt der Realität dar. Eine Datenbank zur Schulverwaltung kann Tabellen über Schüler, Klassen, Bildungsgänge, Noten, Lehrer u.s.w. enthalten. Eine Datenbank wird in einer oder mehreren Dateien gespeichert.
- Unter dem **Datenbankmanagementsystem** versteht man alle Programme, welche für die Erstellung und Pflege einer Datenbank notwendig sind.
- Ein **Datenbanksystem** besteht aus einer oder mehrerer Datenbanken und dem Datenbankverwaltungssystem.



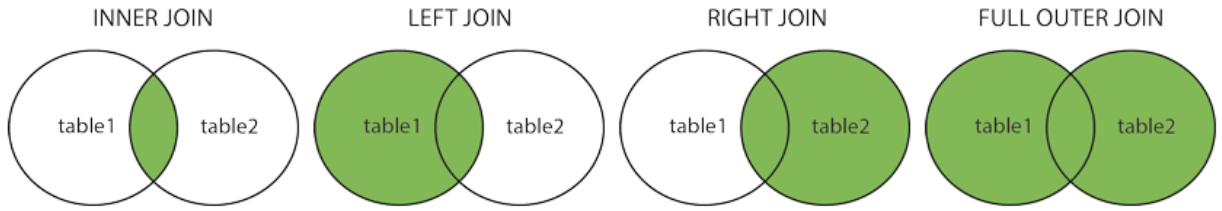
- Jedes Attribut hat einen Typen (Datentypen).
- Steht der Wert eines Attributs noch nicht fest, so wird **NULL-Wert** benutzt. Dieser ist keine mathematische 0.
- Es gibt keine zwei Zeilen, in denen alle Attribute gleich sind.
- Ein **Schlüssel** ist ein Attribut oder eine Kombination mehrerer Attribute, die zur

Identifizierung eines Datensatzes genutzt werden können.
- Unter **Primärschlüssel** (**Primary Key**) versteht man den Schlüssel, der primär zur Identifikation genutzt wird.
- Neben dem Primärschlüssel kann es noch weitere Sekundärschlüssel zum schnelleren Zugriff geben.
- Ein **Fremdschlüssel** (**Foreign Key**) wird zur Identifizierung von Datensätzen in anderen Tabellen genutzt.

JOINS

www.w3schools.com

- `(INNER) JOIN` : Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN` : Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN` : Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN` : Returns all records when there is a match in either left or right table

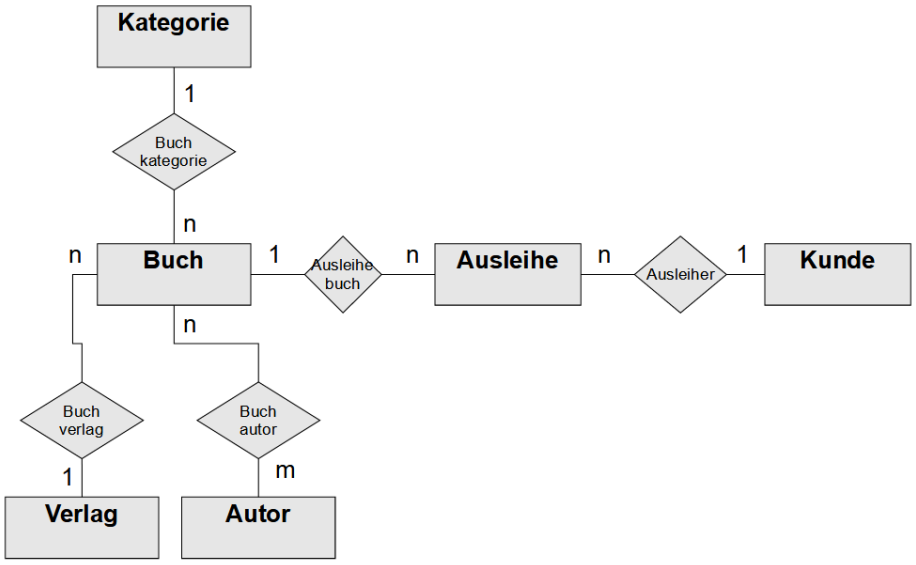


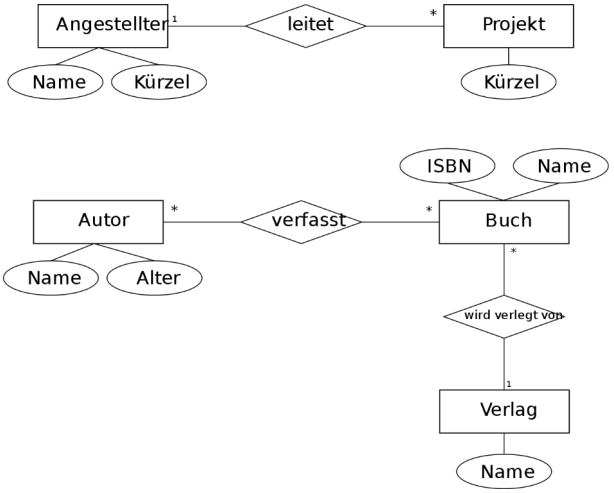
beispiel_ls4_03.sql

```
SELECT
    customers.customername,
    orders.orderdate
FROM
    customers
    INNER JOIN orders USING (CustomerID);

SELECT
    customers.customername,
    orders.orderdate
FROM
    customers
    LEFT JOIN orders USING (CustomerID);
```

LS 4 – Info – Datenbankentwurf

Entity-Relationship-Diagramme





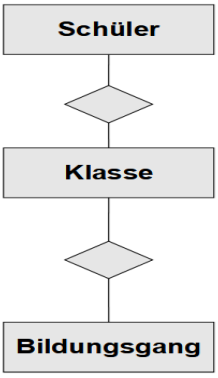
Logischer Datenbankentwurf

- Aufgabe des logische Datenbankentwurfs ist die sorgfältige Planung der Datenbank.
- Der logische Datenbankentwurf konzentriert sich auf die zu realisierende Aufgabe und verzichtet auf technische Aspekte.
- Der logische Datenbankentwurf ist (möglichst) unabhängig von dem zu nutzenden Datenbanksystem.
- Eine Darstellungsmethode für logische Datenmodelle sind Entity-Relationship-Modelle

bzw. Entity-Relationship-Diagramme

Entitiy(-typen)

- Entitiytypen werden als Rechtecke dargestellt.
- Entitytypen besitzen einen Namen.
- Entitytypen entsprechen vereinfacht den späteren Datentabellen

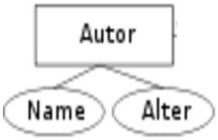


Relationship(-typen)

- Relationshiptypen werden als Rauten dargestellt.
- Relationshiptypen beschreiben Beziehungen zwischen Entitytypen.
- Relationshiptypen werden in der späteren Datenbank zu zusätzlichen Datenfeldern bzw. zu eigenen Datentabellen.
- Relationshiptypen können zum besseren Verständnis mit eigenen Namen versehen werden.

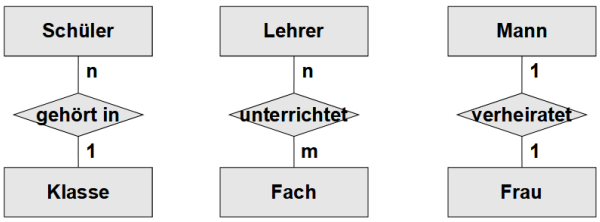
Attribute

- Entitytypen bestehen aus Attributen, die den späteren Datenfeldern entsprechen.
- Attribute können in dem Diagramm als Kreise oder Ellipsen dargestellt werden. Oft werden sie aber nur textuell dargestellt und nicht im Diagramm, um dieses übersichtlicher zu halten.



Kardinalität

- Relationships werden durch Kardinalitäten näher beschrieben. Die Kardinalität gibt an, wie viele Entities jeweils miteinander in Beziehung stehen. Die wichtigsten Beziehungen sind
  - 1:1 Beziehung
  - 1:n Beziehung
  - n:m Beziehung



Datenbank-Schemata

- Kompakte textuelle Darstellung einer Tabelle und enthaltener Attribute

```
CREATE TABLE t1_demo (
    t1_ID   SERIAL,
    t1_zeit DATETIME DEFAULT NOW(),
    t1_wert DECIMAL(9,2),
    PRIMARY KEY (t1_ID)
    );
```

⇔ t1_demo = (t1_ID, t1_zeit, t1_wert)

SQL-Befehle um Datenbanken und Tabellen anzulegen(Data Definition Language (DDL))

- CREATE DATABASE
- CREATE TABLE
- CREATE INDEX
- DROP DATABASE
- DROP TABLE
- DROP INDEX
- ALTER TABLE

Normalisierung

Normalisierung

- Sinn und Zweck der Normalisierung ist die Vermeidung von Doppelspeicherung von Informationen und das Erreichen von sinnvoll strukturierten Daten
- Optimierung des Datenmodells

Normalformen

1. Normalform
2. Normalform
3. Normalform

Boyce-Codd Normalform
4. Normalform

Erste Normalform

- Eine Tabelle befindet sich in der **ersten Normalform** (**1NF**), wenn die Wertebereiche der Attribute atomar sind.

(PK)

SchuelerID	Name	Anschrift
1	Nobert Maier	52152 Simmerath, Hauptstraße 99

(PK)

SchuelerID	Nachname	Vorname	PLZ	Ort	Strasse	Hausnummer
1	Maier	Nobert	52152	Simmerath	Hauptstraße	99

(PK)

SchuelerID	Nachname	Vorname	Noten
1	Maier	Nobert	Deutsch 5, Englisch 6, Mathematik 2

(PK)

SchuelerID	Nachname	Vorname
1	Maier	Nobert

(PK) (FK)

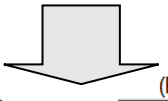
NotenID	SchuelerID	Fach	Note
1	1	Deutsch	5
2	1	Englisch	6
3	1	Mathematik	2

Zweite Normalform

- Eine Tabelle befindet sich genau dann in der **zweiten Normalform** (**2NF**), wenn sie sich in der ersten Normalform befindet und jedes Nichtschlüsselattribut von dem gesamten Primärschlüssel abhängig ist.

(PK) (PK)

LehrerID	Fach	Nachname	Kursanzahl
1	Mathematik	Maier	4
1	Informatik	Maier	3
2	Mathematik	Müller	2
2	Englisch	Müller	1



(PK)

LehrerID	Nachname
1	Maier
2	Müller

(PK) (FK)

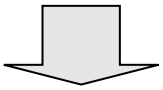
ID	LehrerID	Fach	Kursanzahl
1	1	Mathematik	4
2	1	Informatik	3
3	2	Mathematik	2
4	2	Englisch	1

Dritte Normalform

- Eine Tabelle befindet sich genau dann in der **dritten Normalform** (**3NF**), wenn sie sich in der zweiten Normalform befindet und kein Nichtschlüsselattribut transitiv von einem anderen Attribut abhängt.

(PK)

KundenID	PLZ	Ort
1	52152	Simmerath
2	52152	Simmerath
3	52156	Monschau
4	52156	Monschau



(PK)

KundenID	PLZ
1	52152
2	52152
3	52156
4	52156

(PK)

PLZ	Ort
52152	Simmerath
52156	Monschau

LS 4 – Info – Datenbankdokumentation

Beispiel Kunde-Bestellung

a) SQL-Skript

beispiel_ls4_04db.sql

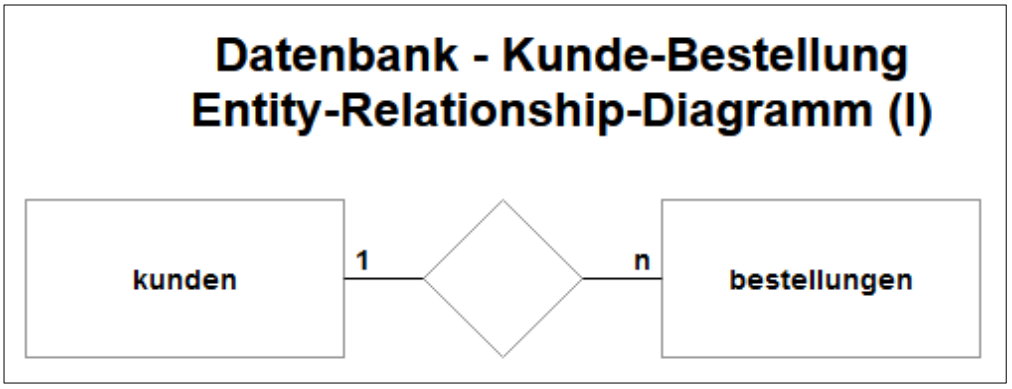
```
DROP DATABASE IF EXISTS beispiel_ls4_joins;

CREATE DATABASE beispiel_ls4_joins DEFAULT CHARACTER SET utf8;

USE beispiel_ls4_joins;

/* Tabellen anlegen */
CREATE TABLE kunden (
    KundenID int(11) NOT NULL AUTO_INCREMENT,
    KundenName varchar(255) DEFAULT NULL,
    PRIMARY KEY (KundenID)
);

CREATE TABLE bestellungen (
    BestellungsID int(11) NOT NULL AUTO_INCREMENT,
    KundenID int(11) DEFAULT NULL,
    Bestellungsdatum date DEFAULT NULL,
    PRIMARY KEY (BestellungsID)
);
```

b) Relationen-Schemata

kunden = (KundenID, KundenName)

bestellungen = (BestellungsID, KundenID, Bestellungsdatum)

c) Entity-Relationship-Diagramm ohne Attribute



d) **Entity-Relationship-Diagramm mit Attributen**