ingenieur
wissenschaften
htw saar

Digital Skills für Ingenieur*innen

3. Vorlesungstag

„Datenbanken“

ingenieur
wissenschaften
htw saar

Datenbanken

Allgemein – CSV/Excel vs. Datenbanken [1]

- Einfache Datenspeicherformate sind grundsätzlich gut, wenn
  - Sie keine schlüsselfertigen Datenbanklösungen besitzen.
  - Ihnen die Technologiekompetenzen fehlen.
  - nur eine Person den Datenspeicher gleichzeitig verwendet.
  - Sie den Datenspeicher und die Visualisierungen zügig benötigen.
  - fällt das Datenwachstum und die Datenkomplexität gering bis moderat aus.
  - ist die fachliche Kritikalität des Tools und seinen Daten gering.

[1]: Vgl. https://www.palladio-consulting.de/excel-vs-datenbank/

ingenieur
wissenschaften
htw saar

Datenbanken

Allgemein – CSV/Excel vs. Datenbanken [1]

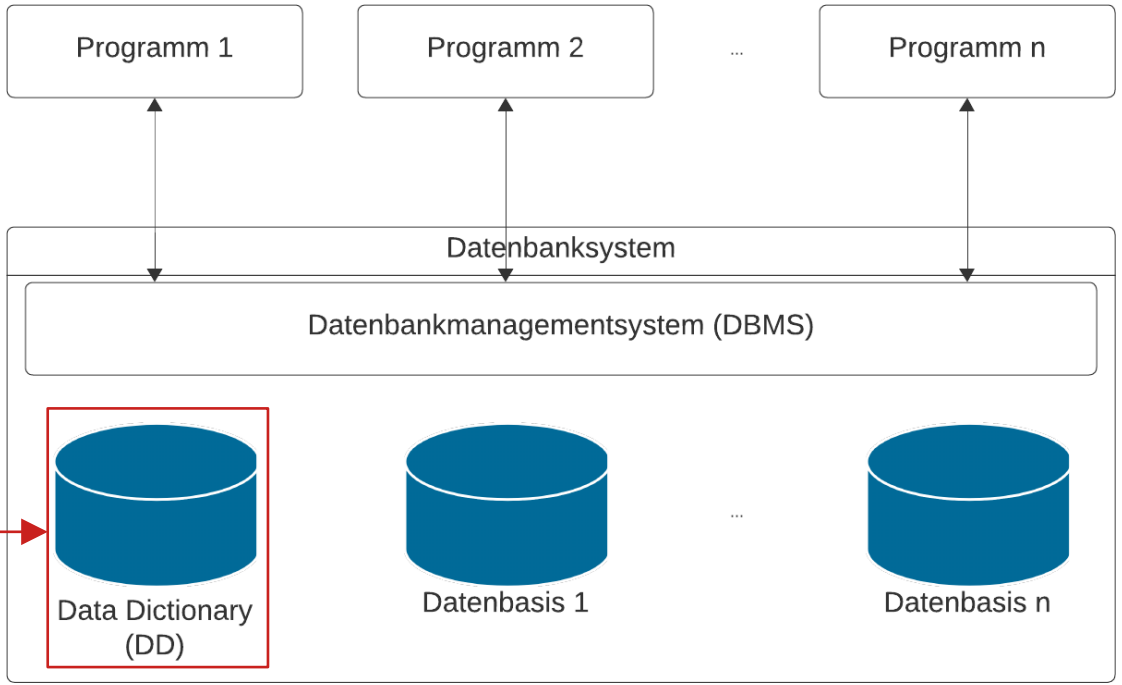
- Ein Datenbankmanagement System (DBMS) sollte gewählt werden, wenn
  - Sie die Daten vor unbefugten Lese- und Schreibzugriffen schützen möchten.
  - mehrere Anwender*innen gleichzeitig und verteilt auf die Daten zugreifen sollen.
  - komplexe Datenabfragen möglich sein sollen.
  - die Datenqualität einen hohen Stellenwert einnimmt.
  - die Lösung ihre Daten an weitere Systeme per Schnittstelle zur Verfügung stellt.

[1]: Vgl. https://www.palladio-consulting.de/excel-vs-datenbank/

Datenbanken

Aufbau eines DBMS

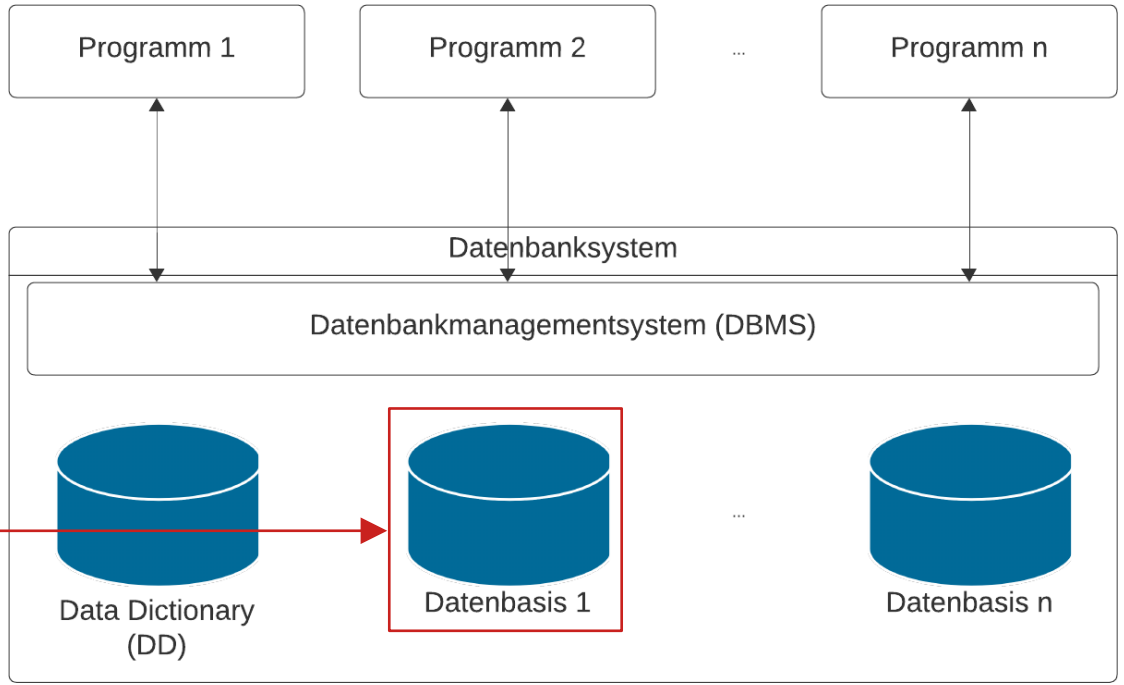
Data Dictionary oder auch Metadatenbank, kann als das Inhaltsverzeichnis des DBS bezeichnet werden. Enthält Namen der Tabellen, Spalten, Datentypen, etc.



Datenbanken

Aufbau eines DBMS

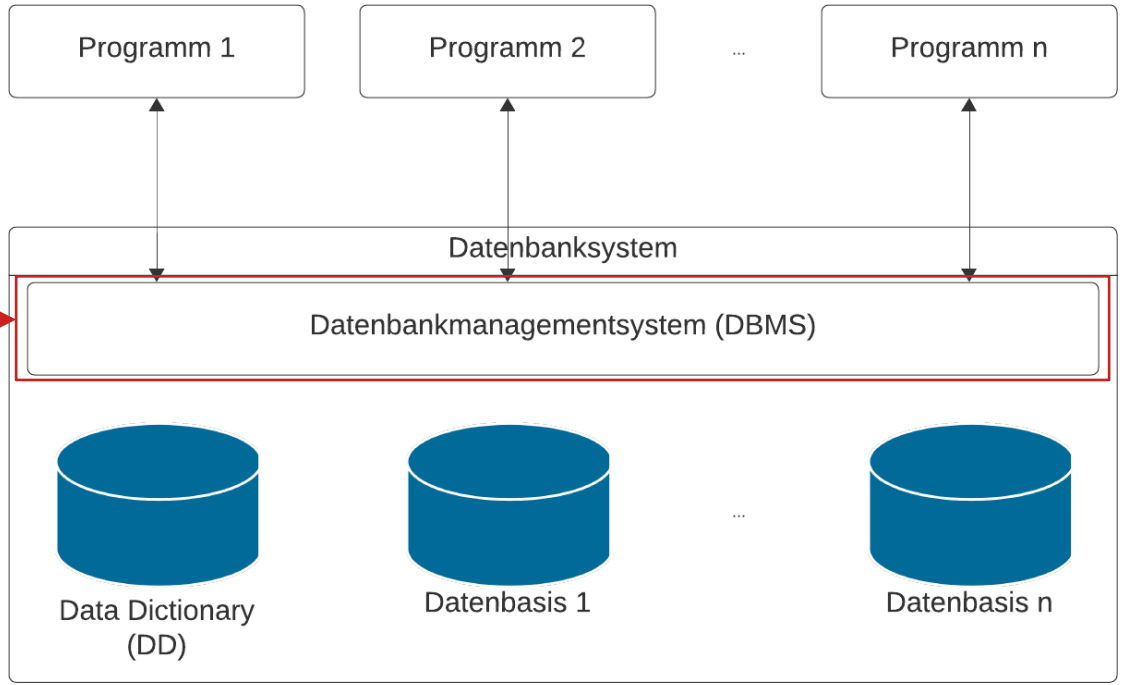
Unter der **Datenbasis** versteht man die eigentlichen, persistent gespeicherten Anwendungsdaten, kurz Daten, des Datenbanksystems (DBS)



Datenbanken

Aufbau eines DBMS

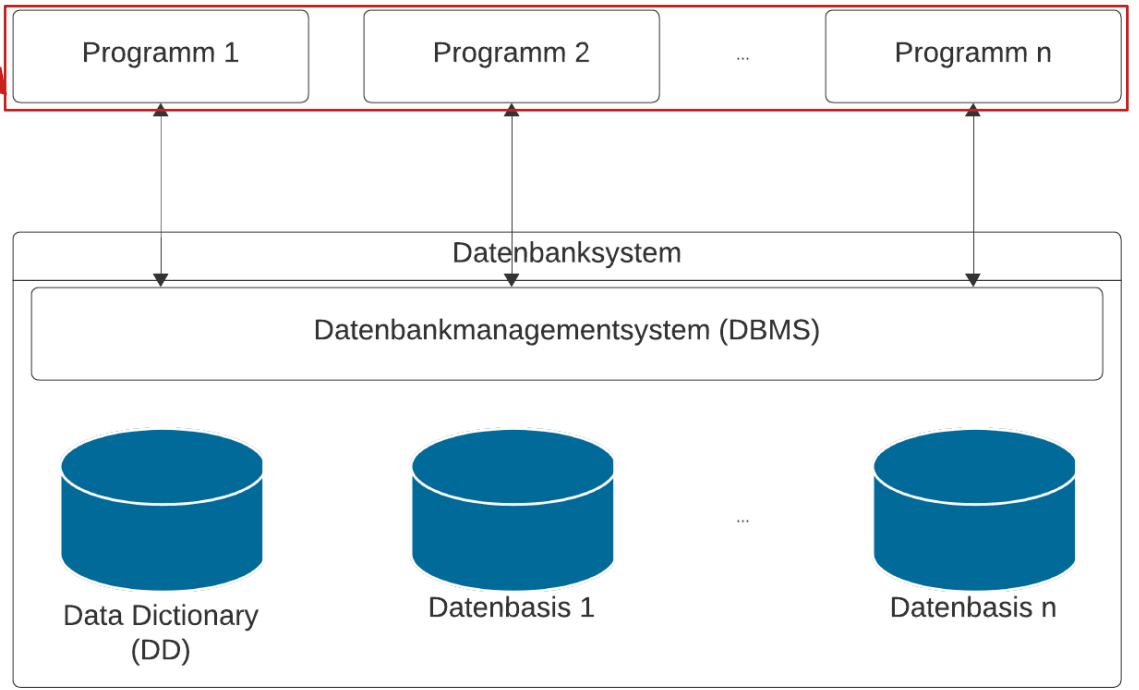
Das **DBMS** stellt Funktionen zum Einfügen, Ändern, Löschen und Wiederfinden der Daten zur Verfügung , ebenso wie die Funktionalität zur Datensicherheit und Datenschutz.



Datenbanken

Aufbau eines DBMS [2]

Die Programme können über die Programmierschnittstellen, API genannt über das DBMS mit den Tabellen der Datenbanken interagieren.



Datenbanken

Datenbanksprache

- Die Sprache mit denen relationale Datenbanken „sprechen“, nennt sich Structured Query Language (SQL)

- SQL-Befehle lassen sich in fünf Kategorien unterteilen

  - Befehle zur Abfrage und Aufbereitung der gesuchten Informationen – Data Query Language (DQL)
  - Befehle zur Datenmanipulation (Ändern, Einfügen, Löschen von Datensätzen) – Data Manipulation Language (DML)
  - Befehle zur Definition des Datenbankschemas (Erzeugen, Ändern, Löschen von Datenbanktabellen, Definition von Primärschlüsseln und Fremdschlüsseln) – Data Definition Language (DDL)

Datenbanken

Datenbanksprache

- Befehle für die Rechteverwaltung – Data Control Language (DCL)
- Befehle für die Transaktionscontrolle – Transaction Control Language (TCL)

- Die Sprache basiert auf der relationalen Algebra, ihre Syntax ist relativ einfach aufgebaut und semantisch an die englische Umgangssprache angelehnt.

- Wird durch ISO und ANSI standardisiert.

Datenbanken

Entwicklung eines Datenbankschemas

- Bei der Entwicklung eines Datenbankschemas geht man oft in drei Beschreibungsschichten vor:
  - Fachkonzept: Strukturierte Darstellung der Geschäftsprozesse mittels Beschreibungsmodellen, die für die Fachseite verständlich sind.
  - Datenverarbeitungskonzept: Umsetzung des Fachkonzeptes in DV-nahe Beschreibungsmodelle (z.B. Entity-Relationship Modell, Relationen)
  - Implementierungsebene: DV-technische Realisierung der beschriebenen Prozessteile (z.B. Programmcode)

Datenbanken

Fachkonzept

- „(Semi-)formale, implementierungsunabhängige Beschreibung einer betriebswirtschaftlichen Konzeption“ [4]
- oft in Form eines Lasten- und Pflichtenhefts
  - Fachliche Beschreibung, keine technische Beschreibung
  - (Funktionale) Anforderungen aus Anwendersicht eines IT-Systems

    Bsp.: Die Datenbank muss die Möglichkeiten bieten, Mitarbeiter*innendaten zu pflegen (anlegen, ändern, löschen) und nach Mitarbeitern zu suchen.

Datenbanken

Modell

- Eine vereinfachte Abbildung eines Ausschnitts aus der Wirklichkeit (Realität) im Hinblick auf einem bestimmten Zweck.
- Merkmale eines Modells
  - Reduzierung der Komplexität
  - Abstraktion, Verallgemeinerung
  - Formalisierte Beschreibung, d.h. spezielle Darstellung und Notation

Datenbanken

Entity-Relationship Modell

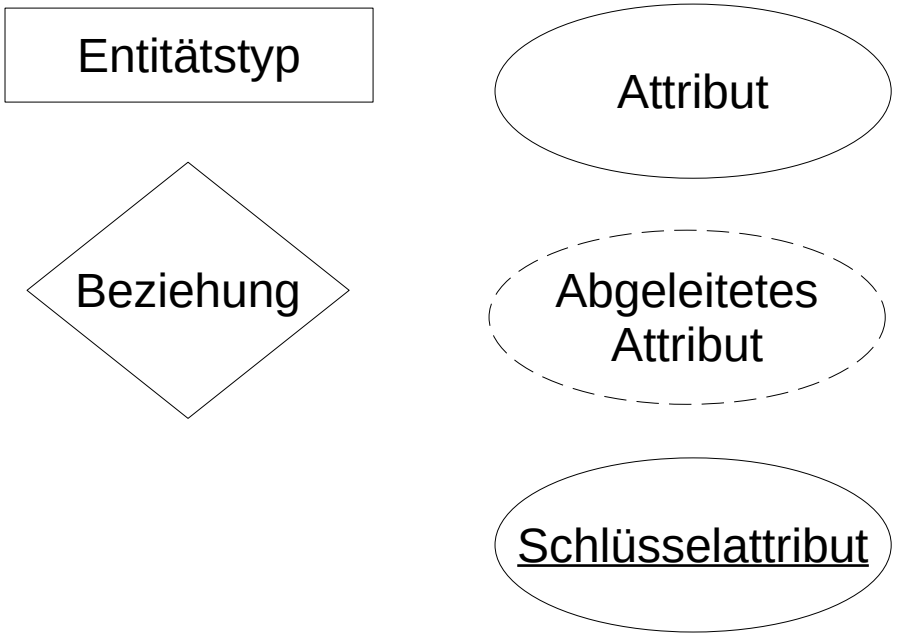
- Peter Chen
  - * 03.01.1947
  - Taiwanesischer Informatiker
- 1976 als universelles Modell entwickelt
  - nicht auf (relationale) Datenbanken beschränkt

Abbildung 1: Peter Chen.png

- Diese graphische Notation dient der Datenmodellierung. Es können Beziehungen zwischen Objekten (Entitäten) abgebildet werden.

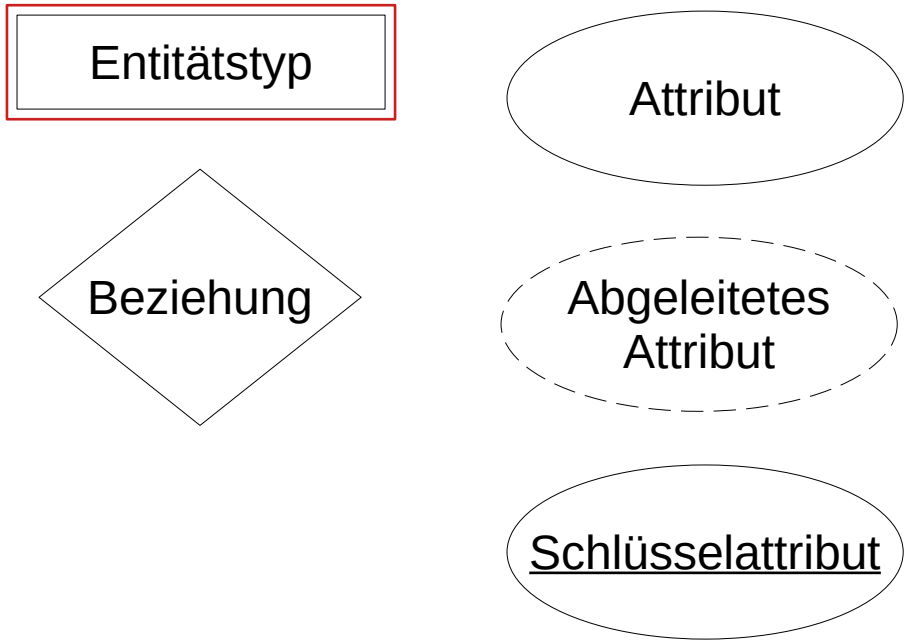
Datenbanken

Entity-Relationship Modell - Notation



Datenbanken

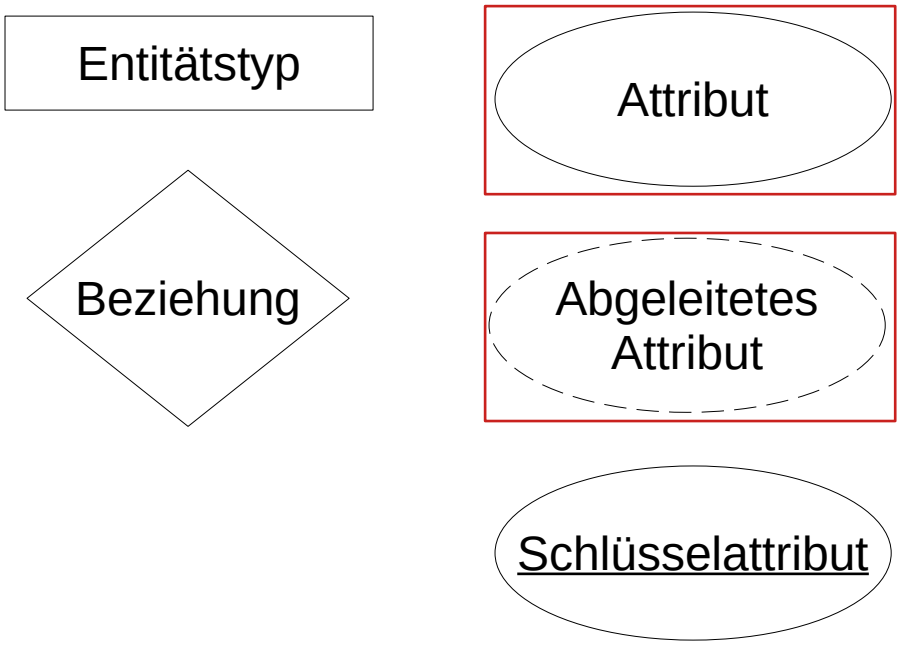
Entity-Relationship Modell - Notation



- Ein Entitätstyp ist eine abstrakte Beschreibung einer Menge von Entitäten mit gleichen Attributen.
- Eine Entität ist ein eindeutig identifizierbares Element des Entitätstyps.
- Beispiele: Ein Fahrzeug, ein Konto, eine Person, ein Zustand.

Datenbanken

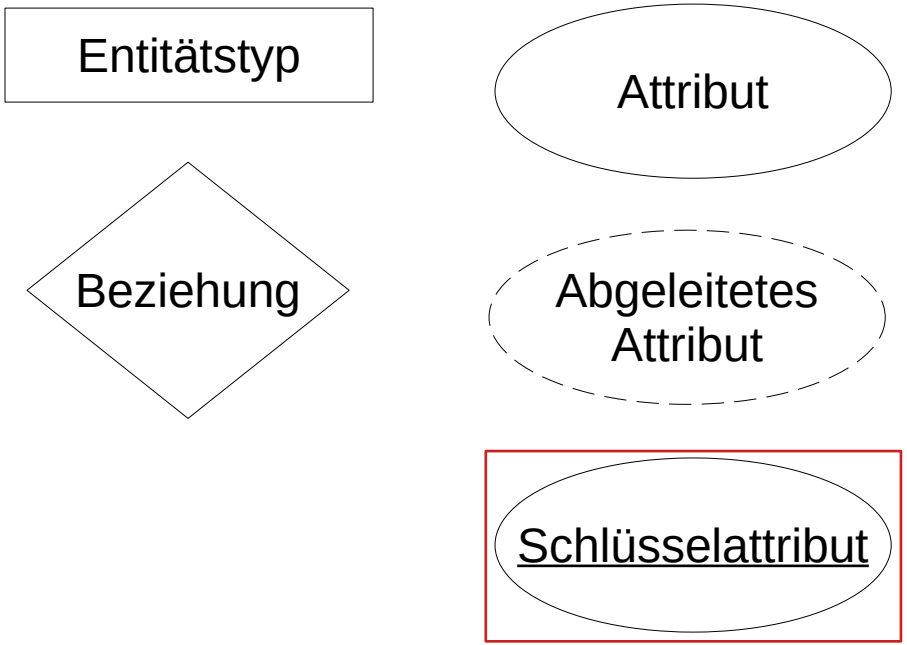
Entity-Relationship Modell - Notation



- Ein Attribut beschreibt eine Eigenschaft eines Entitätstyps, besitzt einen eindeutigen Namen und ist zeitinvariant.
- Attribute können einfach, zusammengesetzt oder abgeleitet sein.

Datenbanken

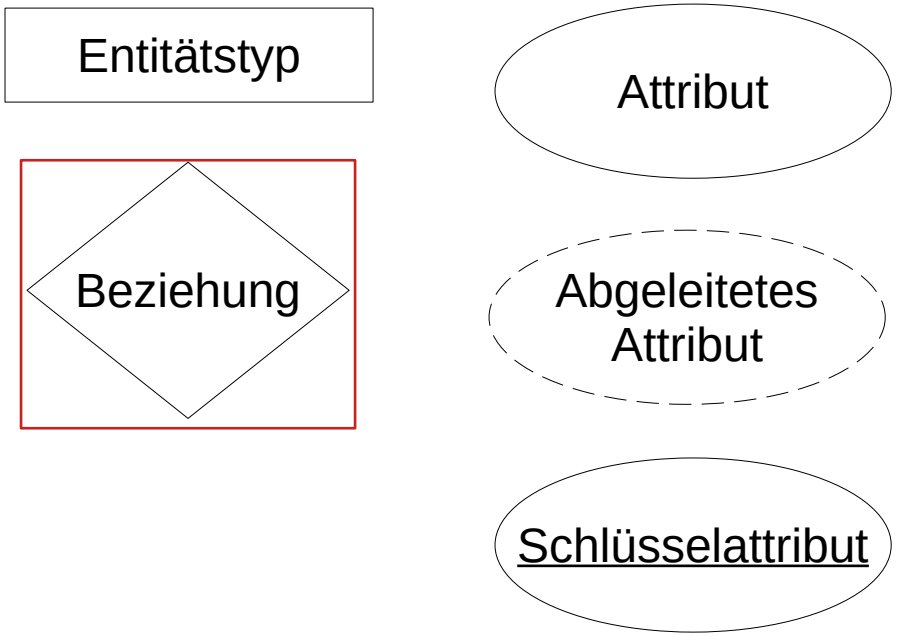
Entity-Relationship Modell - Notation



- Schlüsselattribut (Relationship):
  Ein Schlüssel ist ein Attribut oder eine Kombination mehrerer Attribute eines Entitätstyps, um jede Entität eindeutig zu identifizieren.
- Beispiel: Matrikelnummer bei Student*innen, Personalnummer bei Mitarbeiter*innen oder Geburtstag + Vorname + Nachname (zusammengesetzter Schlüssel)

Datenbanken

Entity-Relationship Modell - Notation



- Beziehung (Relationship): Verknüpfung/Zusammenhang zwischen zwei oder mehreren Entitäten. Beziehungen selbst können auch Attribute besitzen (z.B. *„Karl Meier ist in Abteilung.“* → seit wann?)
- Beispiel: Angestellter Müller leitet Projekt 4711.

Datenbanken

Entity-Relationship Modell – Kardinalitäten

- Die gebräuchlichsten Beziehungen werden im Hinblick auf ihre Kardinalität in ihrer Grundform wie folgt eingeteilt
  - 1:1 ([0 oder 1] zu [1 oder 0]): Jede Entität aus der ersten Entitätsmenge kann mit höchstens einer Entität aus der zweiten Entitätsmenge in Beziehung stehen, und umgekehrt.
  - Beispiel: Jeder Raum ist Klassenraum für höchstens eine Klasse. Jede Klasse hat höchstens einen Raum als Klassenraum.

Datenbanken

Entity-Relationship Modell – Kardinalitäten

- Die gebräuchlichsten Beziehungen werden im Hinblick auf ihre Kardinalität in ihrer Grundform wie folgt eingeteilt
  - 1:n: Jede Entität aus der ersten Entitätsmenge kann mit beliebig vielen Entitäten aus der zweiten Entitätsmenge in Beziehung stehen. Jede Entität aus der zweiten Entitätsmenge kann mit höchstens einer Entität aus der ersten Entitätsmenge in Beziehung stehen.
  - Beispiel: Jede*r Schüler*in geht in höchstens eine Klasse. Jede Klasse besteht aus mehreren Schüler*innen.

Datenbanken

Entity-Relationship Modell – Kardinalitäten

- Die gebräuchlichsten Beziehungen werden im Hinblick auf ihre Kardinalität in ihrer Grundform wie folgt eingeteilt
  - m:n: Jede Entität aus der ersten Entitätsmenge kann mit beliebig vielen Entitäten aus der zweiten Entitätsmenge in Beziehung stehen, und umgekehrt.
  - Beispiel: Jede*r Lehrer*in unterrichtet mehrere Klassen. Jede Klasse wird von mehreren Lehrer*innen unterrichtet.

Literatur

[1]: Vgl. https://www.palladio-consulting.de/excel-vs-datenbank/
[2]: Vgl. Chen, P. (1976): The Entity-Relationship Model - Toward a Unified View of Data, ACM Transactions on Database Systems Vol. 1. No. 1, p. 9 – 36.