

Relationale Datenbanken

- Speicherung von (vielen) strukturierten Daten
- Schnelles Finden gewünschter Daten
- Verknüpfen von Daten

- Relationale Datenbanken speichern Daten in Form von Tabellen
 - Mathematisch kann eine solche Tabelle auch als Relation aufgefasst werden (daher der Name)
- In jeder Zeile einer solchen Tabelle befindet sich ein **Datensatz**
- Jede Spalte steht für ein **Datenfeld** (synonym: **Attribut** oder **Selektor**)
 - In der Praxis verfügt ein Datenfeld auch über einen **Datentyp** (bspw. Ganzzahl, Text, Datum, ...)

- Tabelle **Schüler**:

Vorname	Name	Geburtstag	Geschlecht	Klasse
Hans	Wurst	12.03.2016	m	5
Lisa	Lustig	07.07.2010	w	11
Wolfi	Wiederholer	23.05.2011	m	8

- Interessiert man sich nur für den Aufbau einer Relation ohne den konkreten Inhalt, so beschreibt man nur das **Relationenschema**
 - Hier:
Schüler(Vorname, Name, Geburtstag, Geschlecht, Klasse)

- Zur eindeutigen Identifikation jedes Datensatzes einer Tabelle muss ein **Schlüssel** festgelegt werden
 - Einzelnes Datenfeld oder Kombination von Datenfeldern
- Begrifflich wird unterschieden:

Superschlüssel Menge von Attributen, die eine eindeutige Identifizierung jedes Datensatzes erlaubt (also für keine zwei Datensätze identische Werte annimmt)

Schlüsselkandidat Superschlüssel mit minimaler Anzahl von Attributen

Primärschlüssel Der als Schlüssel gewählte Schlüsselkandidat

- Bei der Darstellung als Relationenschema werden die zum Primärschlüssel gehörenden Datenfelder unterstrichen

- Das DBMS ist die Software, die es erlaubt, mit der Datenbank zu “Interagieren”
 - Aufsetzen und Konfigurieren der Datenbank
 - Einfügen, Löschen und Abfragen von Daten
 - Mehrbenutzerzugriff
 - Transaktionen
 - ...
- Der Zugriff auf die Daten in der Datenbank erfolgt in der Regel nur über die Schnittstelle des DBMS
- Zur Abfrage und Änderung von Daten stellt ein DBMS in der Regel eine **Abfragesprache** zur Verfügung

- Standardisierte Datenbanksprache für
 - Definition von Datenstrukturen
 - Einfügen, Ändern und Löschen von Datensätzen
 - Abfragen

- Grundlegender Aufbau:

SELECT <Datenfeld(er)> **FROM** <Tabelle(n)> **WHERE** <Bedingung(en)>

- Bezieht sich die Abfrage auf mehrere Datenfelder oder Tabellen, so werden diese jeweils durch Komma getrennt
 - Zum Verknüpfen von Bedingungen stehen die logischen Operatoren **AND, OR, NOT** zur Verfügung
 - SQL-Schlüsselwörter können groß oder klein geschrieben werden, bei Tabellennamen und Datenfeldern muss die Schreibweise jedoch exakt mit den tatsächlichen Namen übereinstimmen
- Beispiel

SELECT Name, Geburtstag **FROM** Schüler **WHERE** Vorname **LIKE** 'Wolf%'

- Der Zusatz **ORDER BY** kann benutzt werden, um die Reihenfolge der Ausgabe festzulegen
- Mittels **ASC** und **DESC** kann auf- oder absteigend sortiert werden
- Beispiel:

```
SELECT Vorname, Name FROM Schüler ORDER BY Geburtstag DESC
```

Aggregatfunktionen

- Aggregatfunktionen berechnen aus einer Menge von Werten ein einzelnes Ergebnis
- Die gebräuchlichsten sind:

MIN() / **MAX()** Berechnung von Minimum/Maximum

COUNT() Bestimmen der Anzahl

SUM() Berechnen der Summe

AVG() Berechnen des Durchschnitts

- Beispiele:

```
SELECT COUNT(*) FROM Schüler
```

```
SELECT MAX(Klasse) FROM Schüler
```

Aufgabe 1

- Öffne die Datei “geographie.odt” mit LibreOffice Base
- Ermittle folgende Daten mittels geeigneter SQL-Abfragen:
 - Name, Regierungschef und Hauptstadt aller Bundesländer
 - Alle Bundesländer mit 4 Stimmen im Bundesrat
 - Alle Bundesländer, die ein Stadtstaat sind
 - Name und Fläche aller Bundesländer mit einer Fläche von mehr als 20 000km², aufsteigend sortiert nach Einwohnerzahl
 - Alle Städte mit einer Einwohnerzahl zwischen 300 000 und 600 000 (jeweils einschließlich)
 - Alle Flüsse mit einer Länge von mindestens 200km, die in Rhein oder Donau münden