

Datenbanken

- Eine **Datenbank** ist eine Sammlung von Daten, die aus der Sicht des Benutzers zusammen gehören.
- Ein **Datenbankverwaltungssystem** (DBMS) ist ein informatisches System zur Verwaltung einer Datenbank.
- In einer **relationalen Datenbank** werden die Daten in Form von zweidimensionalen **Tabellen** verwaltet.

Tabellen

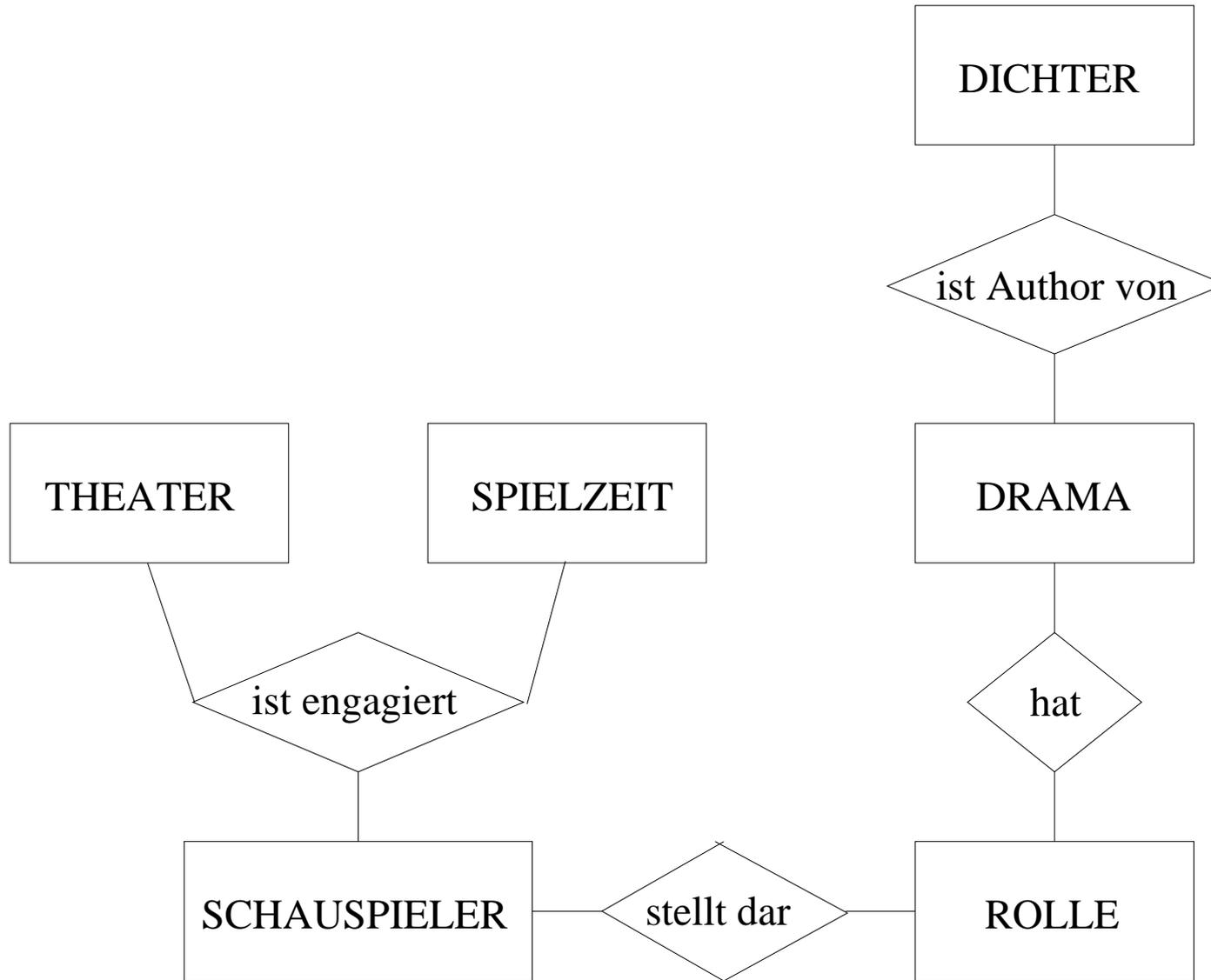
- Eine Zeile der Tabelle ist ein **Datensatz**.
- Eine Spalte ist ein **Feld**.
- Ein **Schlüssel** ist ein Feld, das zum Identifizieren und Verknüpfen von Datensätzen dient (Matrikelnummer).
- Ein **Attribut** ist ein Feld, welches vom Schlüssel abhängige Daten enthält (Name, Vorname, Geburtsdatum).

Ein Beispiel: Tabelle DICHTER

<u>AUTOR</u>	GEBURTSORT	GEBURTSJAHR
Schiller	Marbach	1759
Goethe	Frankfurt (Main)	1749
Calderón	Madrid	1600
Shakespeare	Stratford	1564
von Kleist	Frankfurt (Oder)	1777

Die unterstrichenen Felder bilden den Schlüssel

Entity-Relationship Diagramme



Relationales Schema

- THEATER (TH): TNAME, ORT, SPARTE
- SPIELZEIT (SZ): SAISON, JAHR, DAUER
- SCHAUSPIELER (SP): PNR, NAME, WOHNORT
- ENGAGIERT (ENG): PNR, TNAME, SAISON, JAHR
- ROLLE (RO): FIGUR, TITEL, ROLLEN-TYP
- DARSTELLER (DA):
PNR, FIGUR, AUFF-JAHR, AUFF-ORT, TNAME
- DRAMA (DR): TITEL, AUTOR, URAUFF-JAHR,
URAUFF-ORT
- DICHTER (DI): AUTOR, GEBURTSORT, GEBURTSJAHR

Anfragen

- Q1: Liste alle Dramen (mit TITEL, AUTOR, U-JAHR) auf, die nach 1800 uraufgeführt wurden
- Q2: Finde alle Schauspieler (NAME, WOHNORT), die einmal im 'Faust' mitgespielt haben
- Q3: Finde alle Schauspieler (NAME, WOHNORT), die bei in Weimar uraufgeführten Dramen an ihrem Wohnort als 'Held' mitgespielt haben.

Probleme

- Formale aber anschauliche Sprache für die Beschreibung von Anfragen definieren
- Prozedur für die Anfragensauswertung entwickeln
- Heuristiken für die Optimierung der Auswertung entwickeln

Standard Query Language (SQL)

Grundbaustein:

```
SELECT  AUTOR  
FROM    DICHTER  
WHERE   GEBURTSORT = 'Madrid'
```

SQL-Query für Anfrage Q3

```
SELECT  S.NAME, S.WOHNORT
        FROM  SCHAUPIELER S, DARSTELLER D, ROLLE R,
              DRAMA A
WHERE   S.PNR = D.PNR
        AND  D.FIGUR = R.FIGUR
        AND  R.TITEL = A.TITEL
        AND  A.URAUFF-ORT = 'Weimar'
        AND  R.ROLLENTYP = 'Held'
        AND  D.AUFF-ORT = S.WOHNORT
```

Beziehung zur Prädikatenlogik

- Tabelle \longrightarrow Prädikatensymbol mit Stelligkeit = Anzahl der Tabellenfelder

DICHTER \longrightarrow $Dichter(autor, geb_ort, geb_jahr)$

- Datensätze \longrightarrow Struktur \mathcal{A} für die Prädikatensymbole

$$Dichter^{\mathcal{A}} = \{ (Schiller, Marbach, 1759), \\ \dots, \\ (vonKleist, Frankfurt(Oder), 1777) \}$$

- SQL-Anfrage \longrightarrow Formel mit freien Variablen
(Notation: $F(x_1, \dots, x_n)$)

```
SELECT  AUTOR  
FROM    DICHTER  
WHERE   GEBURTSORT = 'Madrid'
```

$Antw(autor) = \exists geb_jahr : Dichter(autor, 'Madrid', geb_jahr)$

- Antwort \rightarrow Menge der Autoren Au , für die $\mathcal{A}^{Au}(Antw) = 1$
mit $\mathcal{A}^{Au}(autor) = Au$

SQL-Query für Anfrage Q3 (vereinfacht)

```
SELECT  S.NAME, S.WOHNORT
FROM    SCHAUSPIELER S, DARSTELLER D, ROLLE R,
WHERE   S.PNR = D.PNR
        AND D.FIGUR = R.FIGUR
        AND R.ROLLENTYP = 'Held'
        AND D.AUFF-ORT = S.WOHNORT
```

$Antw(name, w_ort) = \exists pnr, figur, a_jahr, tname, titel :$
 $Schau(pnr, name, w_ort) \wedge$
 $Darst(pnr, figur, a_jahr, w_ort, tname) \wedge$
 $Rolle(figur, titel, 'Held')$

Geschachtelte Anfragen

- Finde die Schauspieler (NAME), die in Stuttgart den 'Faust' gespielt haben

```
SELECT  S.NAME
FROM    SCHAUPIELER S
WHERE   ('Stuttgart', 'Faust') IN
        (SELECT  D.AUFF-ORT, D.FIGUR
         FROM    DARSTELLER D
         WHERE   D.PNR = S.PNR)
```

- Formel für die innere Anfrage:

$$\begin{aligned} \textit{Antw1}(pnr) = \exists a_jahr, tname : \\ \textit{Darst}(pnr, \textit{'Faust'}, a_jahr, \textit{'Stuttgart'}, tname) \end{aligned}$$

- Formel für die Anfrage:

$$\begin{aligned} \textit{Antw}(name) = \exists pnr, w_ort : \\ \textit{Schau}(pnr, name, w_ort) \wedge \textit{Antw1}(pnr) \end{aligned}$$

Quantifizierte Anfragen

- Finde die Schauspieler (NAME), die mindestens einmal gespielt haben

```
SELECT  S.NAME
FROM    SCHAUSPIELER S
WHERE   EXISTS
        SELECT  *
        FROM    DARSTELLER D
        WHERE   D.PNR = S.PNR
```

- Formel für die innere Anfrage:

$$\begin{aligned} \textit{Antw1}(pnr) = \exists a_jahr, titel, a_jahr, a_ort, tname : \\ \textit{Darst}(pnr, titel, a_jahr, a_ort, tname) \end{aligned}$$

- Formel für die Anfrage:

$$\begin{aligned} \textit{Antw}(name) = \exists pnr, w_ort : \\ \textit{Schau}(pnr, name, w_ort) \wedge \textit{Antw1}(pnr) \end{aligned}$$

Quantifizierte Anfragen II

- Finde die Schauspieler (NAME), die *alle* Helden gespielt haben

$$\begin{aligned} \text{Antw}(\text{name}) = & \left(\exists \text{pnr}, \text{w_ort} : \right. \\ & \text{Schau}(\text{pnr}, \text{name}, \text{w_ort}) \wedge \\ & \left(\forall \text{figur}, \text{titel} : \right. \\ & \left. \left(\text{Rolle}(\text{figur}, \text{titel}, \text{'Held'}) \right. \right. \\ & \quad \longrightarrow \\ & \left. \left. \left. \exists \text{a_jahr}, \text{a_ort}, \text{tname} : \right. \right. \right. \\ & \left. \left. \left. \text{Darst}(\text{pnr}, \text{figur}, \text{a_jahr}, \text{a_ort}, \text{tname}) \right) \right) \right) \end{aligned}$$

SQL Anfrage

```
SELECT S.NAME
FROM SCHAUSPIELER S
WHERE NOT EXISTS
    (SELECT *
    FROM ROLLE R
    WHERE NOT EXISTS
        (SELECT *
        FROM DARSTELLER D
        WHERE D.PNR = S.PNR
        AND D.FIGUR = R.FIGUR
```

Definitionen und Notationen

- Wir schreiben \mathbf{x} für $\{x_1, \dots, x_n\}$
 $\exists \mathbf{x}$ für $\exists x_1 \dots \exists x_n$
- Eine **Relation** ist eine Formel mit freien Variablen.
- Die **Stelleigkeit** einer Relation ist die Anzahl der freien Variablen.
- $R(\mathbf{x})$ bezeichnet eine Relation mit freien Variablen \mathbf{x} .
Wenn die freien Variablen einer Relation klar aus dem Kontext sind schreiben wir R statt $R(\mathbf{x})$.
- Eine **Bedingung** B ist eine aussagenlogische Kombination von Formeln der Form $x = a$.

Relationenalgebra

- Eine Formel $R(\mathbf{x})$ der **Relationenalgebra** hat die folgende Gestalt:

$$Tab(\mathbf{x})$$

$$\sigma_{B(\mathbf{x}')} (R) = R(\mathbf{x}) \wedge B(\mathbf{x}') \quad \text{mit } \mathbf{x}' \subseteq \mathbf{x}$$

$$\pi_{\mathbf{x}'} (R) = \exists \mathbf{x}'' R(\mathbf{x}) \quad \text{mit } \mathbf{x}' \subseteq \mathbf{x}, \mathbf{x}'' = \mathbf{x} \setminus \mathbf{x}'$$

$$(R_1 \cup R_2) = R_1(\mathbf{x}) \vee R_2(\mathbf{x})$$

$$(R_1 - R_2) = R_1(\mathbf{x}) \wedge \neg R_2(\mathbf{x})$$

$$(R_1 \times R_2) = R_1(\mathbf{x}) \wedge R_2(\mathbf{y})$$

$$(R_1 \bowtie_{i=j} R_2) = \exists z \ R_1(x_1, \dots, x_{i-1}, z, x_{i+1}, x_n) \wedge \\ R_2(y_1, \dots, y_{j-1}, z, y_{j+1}, y_m)$$

SQL → Relationenalgebra

```
SELECT  AUTOR  
FROM    DICHTER  
WHERE   GEBURTSORT = 'Madrid'
```

$$Antw(autor) = \pi_{autor}(\sigma_{geb_ort='Madrid'}(Dichter))$$

Auswertungsprozedur und Optimierungen

- Berechne die Relationen 'bottom-up'
- Nutze Äquivalenzregeln aus um die Auswertung zu Beschleunigen. (Triviale) Beispiele:

$$\sigma_{B_1}(\sigma_{B_2}(R)) \equiv \sigma_{B_2}(\sigma_{B_1}(R))$$

$$\pi_{\mathbf{x}}(R) \equiv \pi_{\mathbf{x}}(\pi_{\mathbf{y}}(R)) \quad \text{wenn } \mathbf{x} \subseteq \mathbf{y}$$

$$\pi_{\mathbf{x}}(\sigma_{B(\mathbf{y})}(R)) \equiv \sigma_{B(\mathbf{y})}(\pi_{\mathbf{x}}(R)) \quad \text{wenn } \mathbf{x} \supseteq \mathbf{y}$$

$$\pi_{\mathbf{x} \cup \mathbf{y}}(R \bowtie_{i=j} S) \equiv \pi_{\mathbf{x}}(R) \bowtie_{i=j} \pi_{\mathbf{y}}(S) \quad \begin{array}{l} \text{wenn } x_i \notin \mathbf{x} \\ \text{und } y_j \notin \mathbf{y} \end{array}$$

$$\sigma_{B(\mathbf{x})}(R \cup S) \equiv \sigma_{B(\mathbf{x})}(R) \cup \sigma_{B(\mathbf{x})}(S)$$

$$\pi_{\mathbf{x}}(R \cup S) \equiv \pi_{\mathbf{x}}(R) \cup \pi_{\mathbf{x}}(S)$$