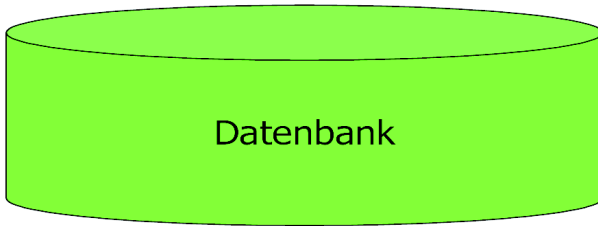


# Kapitel 1: Einführung



# Willkommen!



## Studierenden-Datenbank

- ▶ Hans Eifrig hat die Matrikelnummer 1223. Seine Adresse ist Seeweg 20. Er ist im zweiten Semester.
- ▶ Lisa Lustig hat die Matrikelnummer 3434. Ihre Adresse ist Bergstraße 11. Sie ist im vierten Semester.
- ▶ Maria Gut hat die Matrikelnummer 1234. Ihre Adresse ist Am Bächle 1. Sie ist im zweiten Semester.

# 1.1 Grundbegriffe

## relationale Datenbank

Student

<u>MatrNr</u>	Name	Adresse	Semester
1223	Hans Eifrig	Seeweg 20	2
3434	Lisa Lustig	Bergstraße 11	4
1234	Maria Gut	Am Bächle 1	2

- ▶ Tabellarische Darstellungen dieser Art sind die Grundstrukturen *relationaler Datenbanken*.
- ▶ Begriffe: *Relation*, *Relationsbezeichner*, *Attribut*, *Tupel*, *Schlüssel*, *Primärschlüssel*.

## Struktur und Inhalt

- ▶ Die abstrakte Struktur der Datenbank soll von ihrem Inhalt getrennt werden.

Student(MatrNr, Name, Adresse, Semester)

- ▶ Zustand einer Datenbank

Student

<u>MatrNr</u>	Name	Adresse	Semester
1223	Hans Eifrig	Seeweg 20	2
3434	Lisa Lustig	Bergstraße 11	4
1234	Maria Gut	Am Bächle 1	2

## Beispiel

Der Kurs K010 über Datenbanken wird vom Institut DBIS angeboten.  
Er behandelt die Grundlagen von Datenbanken.

Der Kurs K011 über Informationssysteme wird vom Institut DBIS angeboten.  
Er baut auf Datenbanken auf und behandelt Grundlagen von Informationssystemen.

Student Hans Eifrig hat Datenbanken im WS2003/04 mit der Note 2.0 bestanden.

Professor Lausen hat die Personalnummer 153062 und ist am Institut DBIS.

Professor Lausen hält die Vorlesung Datenbanken im WS2003/04.

```
Student(MatrNr, Name, Adresse, Semester)
Kurs(KursNr, Institut, Name, Beschreibung)
Belegung(MatrNr, KursNr, Semester, Note)
Professor(PersNr, Name, Institut)
Lehrangebot(PersNr, KursNr, Semester)
```

## Beispiel (Ausschnitt)

### Student

<u>MatrNr</u>	Name	Adresse	Semester
1223	Hans Eifrig	Seeweg 20	2
3434	Lisa Lustig	Bergstraße 11	4
1234	Maria Gut	Am Bächle 1	2

### Kurs

<u>KursNr</u>	Institut	Name	Beschreibung
K010	DBIS	Datenbanken	Grundlagen von Datenbanken
K011	DBIS	Informationssysteme	Grundlagen von Informationssystemen

### Belegung

<u>MatrNr</u>	<u>KursNr</u>	Semester	Note
1223	K010	WS2003/2004	2.3
1234	K010	SS2004	1.0

## Objekte und Beziehungen

- ▶ Objekte bilden die elementare Grundlage unserer Betrachtung.
- ▶ Beziehungen sind über Objekten oder anderen Beziehungen definiert; sie entstehen somit durch In-Bezug-Setzen von Objekten.
- ▶ Menge der als relevant betrachteten Objekte und Beziehungen: *Miniwelt*.
- ▶ Objektreferenzen in Beziehungen: *Fremdschlüssel*.

## Arbeiten mit einer Datenbanken

- ▶ Anwendungsprogramme kommunizieren mit einer Datenbank, indem sie Anfragen über den gespeicherten Zustand der Miniwelt stellen, bzw. diesen Zustand durch Ändern, Einfügen oder Löschen von Daten verändern.
- ▶ Ausdrücke einer Datenbankanfragesprache haben eine *mengenwertige, deklarative Semantik*.
  - ▶ Das Ergebnis einer Anfrage ist eine Menge von Tupeln.
  - ▶ Die Anfrage definiert, was für Zusammenhänge aus den Daten der Datenbank gebildet werden sollen, ohne dass die algorithmische Vorgehensweise hierzu spezifiziert werden muss.
- ▶ Anfrageoptimierer.
- ▶ Unabhängigkeit zwischen der Definition der Daten und ihrer Abspeicherung.



### Wie heißen die Professoren des Instituts 'DBIS'?

```
SELECT P.Name  
FROM Professor P  
WHERE P.Institut = 'DBIS'
```

### Welche Studierenden belegen welche Kurse?

```
SELECT S.Name, K.Name  
FROM Student S, Belegung B, Kurs K  
WHERE S.MatrNr = B.MatrNr AND  
B.KursNr = K.KursNr
```

## Globalität der Daten und Transaktionen

- ▶ Die Daten einer Datenbank sind *global*. Datenbanksysteme besitzen einen *Schutzmechanismus* der gewährleistet, dass jede Benutzergruppe nur zu denjenigen Daten zugreifen kann, zu den sie auch Zugriffsberechtigung hat.
- ▶ Eine Ausführung (Prozess) eines Anwendungsprogramms über einer Datenbank wird als *Transaktion* bezeichnet.
- ▶ Transaktionen transformieren einen gegebenen Datenbankzustand in einen neuen Datenbankzustand.

Datenbanken enthalten *Integritätsbedingungen* getrennt von den einzelnen Programmen und Daten, die die Daten in jedem Zustand der Datenbank erfüllen müssen.

- ▶ Es ist für Datenbanken typisch, dass unterschiedliche Transaktionen zeitlich verzahnt auf gemeinsame Daten zugreifen können; es dürfen keine inkorrekten Ergebnisse entstehen.
- ▶ Es muss gewährleistet sein, dass entweder alle Datenbankzugriffe einer Transaktion durchgeführt werden, oder keiner.

## ACID-Eigenschaften der Transaktionen

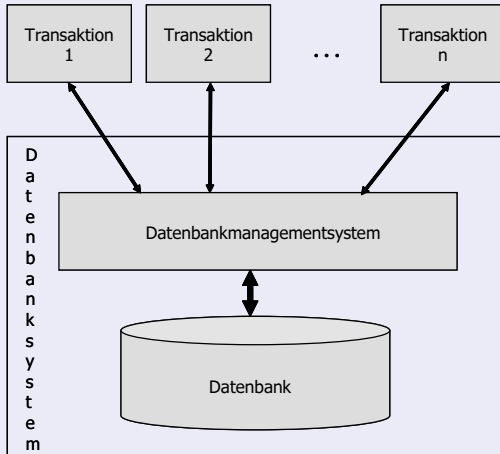
**Atomicity:** Bewirkte Änderungen eines Datenbankzustandes sind atomar und werden entweder vollständig oder gar nicht vorgenommen (*alles-oder-nichts-Prinzip*).

**Consistency:** Eine Transaktion bewirkt einen *konsistenten* Zustandsübergang in der Datenbank.

**Isolation:** *Simulierter* Einbenutzerbetrieb.

**Durability:** Wenn eine Transaktion erfolgreich ihr Ende erreicht hat, dann sind alle von ihr verursachten Änderungen dauerhaft (*persistent*).

# Basisarchitektur



## 1.2 Datenmodelle

### Struktur, Anfragen, Änderungen und Integrität

- ▶ Relationen werden unter Einsatz einer *Datendefinitionssprache* definiert.
- ▶ Zustandsübergänge werden mittels *Anfragesprachen*, bzw. allgemeiner, *Datenmanipulationsprachen* programmiert.
- ▶ *Integritätsbedingungen* werden mittels Anfrageausdrücken formuliert.

Alle diese Sprachen zusammen bezeichnet man als ein *Datenmodell*.

## Abstraktionen

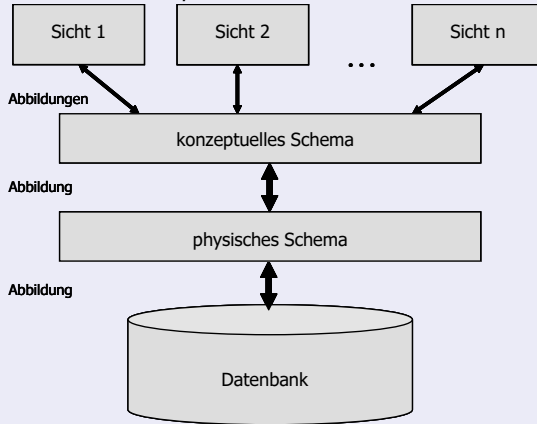
- ▶ Ein Datenmodell ermöglicht auf einer abstrakten Ebene Zugang zu einer Datenbank.
- ▶ Es werden typischerweise drei Abstraktionsebenen unterschieden: die *physikalische*, *konzeptuelle* und *externe Ebene*.

## Ebenen

- ▶ Auf der *physikalischen Ebene* wird festgelegt, wo und wie die Daten auf den Speichermedien physisch abgelegt werden und welche Hilfsstrukturen zur Effizienzsteigerung zur Verfügung gestellt werden sollen.  
Dies ergibt die Grundlage für die Implementierung der Datentypen der konzeptuellen Ebene.
- ▶ Die Datentypen der *konzeptuellen Ebene* werden unabhängig von den Details einer physischen Implementierung definiert.  
Sie definieren das zugrunde liegende globale Gesamtmodell der Miniwelt.  
Dies ergibt die Grundlage für die Anwendungsprogrammierung.
- ▶ Die Datentypen der *externen Ebene* werden speziell für die Belange einzelner Benutzergruppen abgefasst.  
Sie erlauben *Sichten* auf die konzeptuelle Ebene und sind die Basis der Anwendungsprogrammierung.

# Schemata<sup>1</sup>

*physikalische Schema, konzeptuelle Schema, externe Schemata.*



<sup>1</sup>Architekturvorschlag nach ANSI/SPARC 1978 *American National Standards Institut, Standards Planning And Requirements Committee.*



## Datenunabhängigkeit

- ▶ *Physische Datenunabhängigkeit.*  
Das DBMS bildet Operationen der konzeptuellen Ebene automatisch in Operationen des Dateisystems ab. Änderungen der physikalischen Repräsentation bewirken keine Änderungen in den Anwendungsprogrammen.
- ▶ *Logische Datenunabhängigkeit.*  
Das DBMS bildet Operationen der externen Ebene automatisch in Operationen der konzeptuellen Ebene ab. Änderungen der konzeptuellen Repräsentation bewirken keine Änderungen in den Anwendungsprogrammen und umgekehrt.

## 1.3 Das relationale Datenmodell

### Überblick

- ▶ Relationale Datenbanken repräsentieren Zustände einer Miniwelt durch Relationen.
- ▶ Wir unterscheiden die *Definition* der Relationen von den konkreten zeitabhängigen *Inhalten* (Zuständen), den Tupeln der Relationen.
- ▶ Wir reden von einem *Schema* einer Relation, wenn wir die Definition meinen, und von einer *Instanz* einer Relation, wenn wir eine entsprechende Menge von Tupeln meinen.
- ▶ Analogie zu der Unterscheidung zwischen der Typdefinition einer Variablen und ihrem Wert.

## Attribute

- ▶ Objekte und Beziehungen werden durch ihre Eigenschaften repräsentiert. Eigenschaften werden im relationalen Modell als *Attribute* bezeichnet.
- ▶ Sei  $X = \{A_1, \dots, A_k\}$  eine endliche *Attributmenge*, wobei  $k \geq 1$ .
- ▶ Jedes Attribut  $A \in X$  besitzt einen nicht-leeren *Wertebereich*  $dom(A)$ .
- ▶ Die Vereinigung aller Wertebereiche ergibt sich dann zu  $dom(X) = \cup_{A \in X} dom(A)$ .

## Tupel

- ▶ Die Eigenschaften der Objekte und Beziehungen werden zu *Tupeln* zusammengefasst.
- ▶ Ein *Tupel*  $\mu$  über  $X$  ist eine Abbildung

$$\mu : X \longrightarrow \text{dom}(X),$$

wobei  $(\forall A \in X)\mu(A) \in \text{dom}(A)$ .

- ▶ Sei  $\text{Tup}(X)$  im folgenden die Menge aller Tupel über  $X$ .

### Beispiel

$$\begin{aligned}\mu_1 &= \{\text{MatrNr} \rightarrow 1223, \text{Name} \rightarrow \text{Hans Eifrig}, \\ &\quad \text{Adresse} \rightarrow \text{Seeweg 20}, \text{Semester} \rightarrow 2\} \\ \mu_2 &= \{\text{MatrNr} \rightarrow 3434, \text{Name} \rightarrow \text{Lisa Lustig}, \\ &\quad \text{Adresse} \rightarrow \text{Bergstraße 11}, \text{Semester} \rightarrow 4\} \\ \mu_3 &= \{\text{MatrNr} \rightarrow 1234, \text{Name} \rightarrow \text{Maria Gut}, \\ &\quad \text{Adresse} \rightarrow \text{Am Bächle 1}, \text{Semester} \rightarrow 2\}\end{aligned}$$

### Tupel als Abbildungen versus Vektoren

$$\mu_1 = \{\text{MatrNr} \rightarrow 1223, \text{Name} \rightarrow \text{Hans Eifrig}, \\ \text{Adresse} \rightarrow \text{Seeweg 20}, \text{Semester} \rightarrow 2\}$$
$$\mu' = \{\text{MatrNr} \rightarrow 1223, \text{Adresse} \rightarrow \text{Seeweg 20}, \\ \text{Semester} \rightarrow 2, \text{Name} \rightarrow \text{Hans Eifrig}\}$$

(1223,Hans Eifrig,Seeweg 20,2)

(1223,Seeweg 20,2,Hans Eifrig)

Diskutiere bzgl. Gleichheit!

## Relation

- ▶ Eine *Relation*  $r$  über  $X$  ist eine *endliche* Menge  $r \subseteq \text{Tup}(X)$ .
- ▶  $\text{Rel}(X)$  ist die Menge aller Relationen über  $X$ .
- ▶  $r \in \text{Rel}(X)$  ist eine *Instanz* zu  $X$ .
- ▶ Sei  $R$  ein *Relationsbezeichner*.

Ein (*Relations*)-*Schema* zu  $R$  hat die Form  $R(X)$ .  $X$  ist hier eine endliche Attributmenge, das so genannte *Format* des Schemas. Anstatt  $R(\{A_1, \dots, A_k\})$  schreiben wir auch  $R(A_1, \dots, A_k)$ .  $k$  ist die *Stelligkeit* des Relationsbezeichners.

Auch:

$$R(A_1 : \text{dom}(A_1), \dots, A_k : \text{dom}(A_k))$$

## Schlüssel

- ▶ Üblicherweise wird zu jedem Relationsschema ein *Schlüssel* festgelegt.
- ▶ Sei  $R(X)$  ein Relationsschema. Ein Schlüssel  $K$  zu  $R$  ist eine Teilmenge des Formates,  $K \subseteq X$ .
- ▶ Ist ein Schlüssel für ein Schema festgelegt, dann sind nur noch solche Instanzen zu diesem Schema zulässig, in denen keine zwei unterschiedlichen Tupel in sämtlichen Attributen eines Schlüssels gleich sind.
- ▶ Im Allgemeinen können zu einem Schema mehrere Schlüssel existieren.
- ▶ Schlüssel kennzeichnen wir durch Unterstreichung.

## Datenbank

- ▶ Ein (*relationales*) *Datenbank-Schema*  $\mathcal{R}$  ist gegeben durch eine Menge von (Relations-) Schemata,

$$\mathcal{R} = \{R_1(X_1), \dots, R_m(X_m)\},$$

bzw.  $\mathcal{R} = \{R_1, \dots, R_m\}$ .

- ▶ Eine *Instanz*  $\mathcal{I}$  zu einem relationalen Datenbankschema  $\mathcal{R} = \{R_1, \dots, R_m\}$  ist eine Menge von endlichen Relationen  $\mathcal{I} = \{r_1, \dots, r_m\}$ , wobei  $r_i$  Instanz zu  $R_i$  für  $1 \leq i \leq m$ .  
Häufig betrachten wir Instanzen  $\mathcal{I}$  auch als Abbildung:

$$\mathcal{I}(R_i) = r_i, 1 \leq i \leq m.$$



## Anfragen

- ▶ Eine Anfrage definiert zu einer gegebenen Instanz einer Datenbank eine neue Relation, die *Antwort* auf die Anfrage.
- ▶ Eine Anfrage ist eine Transformation, die angewendet auf eine Datenbank-Instanz  $\mathcal{I}$  eine aus den Tupeln in  $\mathcal{I}$  gebildete Relation definiert.
- ▶ Sei  $Q$  eine Anfrage.  $\mathcal{I}(Q)$  ist die Antwort auf  $Q$  bezüglich der Instanz  $\mathcal{I}$ .

## Nullwerte

- ▶ Ist zu einem Attribut kein Wert bekannt, so kann ein *Nullwert* null verwendet werden.
- ▶ Als mögliche Interpretationen eines Nullwertes können wir unterscheiden: *Wert existiert, jedoch zur Zeit unbekannt*, *Wert existiert erst in der Zukunft*, *Wert prinzipiell unbekannt*, oder auch *Attribut nicht anwendbar*.

### Beispiel

Student

<u>MatrNr</u>	Name	Adresse	Semester
1223	Hans Eifrig	null	2
3434	Lisa Lustig	Bergstraße 11	4
1234	Maria Gut	Am Bächle 1	null