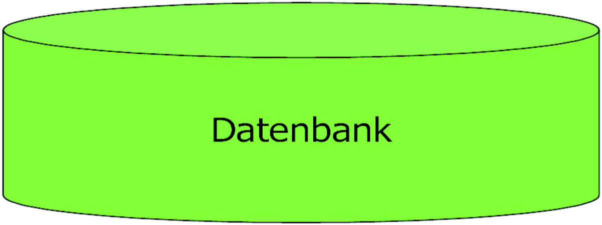
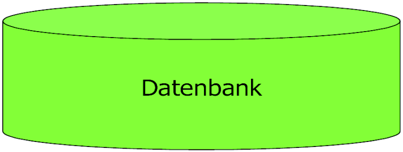
Chapter 1: Introduction



Wellcome!



A student database.

The represented information.

- Hans Eifrig is assigned Matrikelnummer 1223. His address is Seeweg 20. He is studying in the second semester.
- Lisa Lustig is assigned Matrikelnummer 3434. Her address is Bergstraße 11. She is studying in the fourth semester.
- Maria Gut is assigned Matrikelnummer 1234. Her address is Am Bächle 1. She is studying in the second semester.

Relation

Student

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

- We see a *table*, or on a more abstract level, *a relation*,
- name of a table/relation, columns/attributes, rows/tuples, key, primary key,
- structured data.

Structure and content

- Student(MatrId, Name, Address, Semester)

- Student

MatrId	Name	Address	Semester
1223	Hans Eifrig	Seeweg 20	2
3434	Lisa Lustig	Bergstraße 11	4
1234	Maria Gut	Am Bächle 1	2

Information analysis

Course K010 covering Databases is offered by Institute DBIS.
Its contents are the foundations of databases.

Course K011 covering Information Systems is offered by Institute DBIS.
It builds on Databases and its contents are the foundations of Information Systems.

Student Hans Eifrig passed Databases WS2003/04 and got grade 2.0.

Professor Lausen's PersonalId is 1234 and he belongs to DBIS.

Professor Lausen gives the course about databases in WS2003/04.

```
Student(MatrId, Name, Address, Semester)
Course(CourseId, Institute, Name, Content)
Registration(MatrId, CourseId, Semester, Grade)
Professor(PersId, Name, Institute)
Offering(PersId, CourseId, Semester)
```

Some tuples

Student

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

Course

<u>CourseId</u>	Institute	Name	Description
K010	DBIS	Databases	Foundations of Databases
K011	DBIS	Information Systems	Foundations of Information Systems

Registration

<u>MatrId</u>	<u>CourseId</u>	Semester	Grade
1223	K010	WS2003/2004	2.3
1234	K010	SS2004	1.0

Objects and relationships

- We understand the information to be represented in a database as a set of *objects* and *relationships* between them.
- Objects are represented by tuples. Objects can be distinguished by their key values.
- Relationships are represented by tuples; the involved objects are represented by the respective key values. We talk about *foreign keys*.

Using a database

- Application programs communicate with a database to *query*, *update*, *insert* and *delete* the state of the database.
- All these operations are based on the query language of the database (meaning SQL).
- Query expressions have a *set-oriented*, *declarative semantics*:
  - The result of a query is a set of tuples.
  - The query defines the *what* and not the algorithmically *how*.
- Given the *what*, an optimizer can try to improve the efficiency of the query evaluation.

What are the names of the 'DBIS'-professors?

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

Which students are registered for which courses?

```
SELECT S.Name, K.Name
FROM Student S, Registration B, Course K
WHERE S.MatrId = B.MatrId AND
B.KursId = K.KursId
```

Summary

- A relational database - or simply database - uses relations (tables) to represent the information required for a certain business, i.e. tasks of an enterprise, web portal, or even your personal life.
- We also say, a database represents a relevant state of its environment.
- We distinguish the *definition* of the structure - the type - of a relation from its concrete time-dependant *state* - the value.
- The *schema* of a relation refers to the type, the *instance* to a certain value, i.e. a set of tuples, respectively rows, if we think of a table.

Formalization

More on attributes

- Objects and relationships are represented by their properties. Properties give rise to the attributes of our relations.
- Let $X = \{A_1, \ldots, A_k\}$ be a (finite) set of attributes, $k \geq 1$.
- Each attribute $A \in X$ is assigned a non-empty *domain* $dom(A)$.
- $dom(X) = \cup_{A \in X} dom(A)$.

Tuple

- The properties of the objects and relationships are grouped to form *tuples*.
- A *tuple* μ over X is a mapping

$$\mu : X \longrightarrow dom(X),$$

where $(\forall A \in X)\mu(A) \in dom(A)$.
- $\mathrm{Tup}(X)$ is defined as the set of all tuples over X.

Example

$\mu_1 =$ $\{$`MatrId` $\rightarrow$ 1223, `Name` $\rightarrow$ Hans Eifrig,
`Address` $\rightarrow$ Seeweg 20, `Semester` $\rightarrow$ 2$\}$

$\mu_2 =$ $\{$`MatrId` $\rightarrow$ 3434, `Name` $\rightarrow$ Lisa Lustig,
`Address` $\rightarrow$ Bergstraße 11, `Semester` $\rightarrow$ 4$\}$

$\mu_3 =$ $\{$`MatrId` $\rightarrow$ 1234, `Name` $\rightarrow$ Maria Gut,
`Address` $\rightarrow$ Am Bächle 1, `Semester` $\rightarrow$ 2$\}$

Tuples: mappings vs. vectors

$$\mu_1 = \{\texttt{MatrId} \rightarrow 1223,\ \texttt{Name} \rightarrow \text{Hans Eifrig},\ \texttt{Address} \rightarrow \text{Seeweg 20},\ \texttt{Semester} \rightarrow 2\}$$

$$\mu' = \{\texttt{MatrId} \rightarrow 1223,\ \texttt{Address} \rightarrow \text{Seeweg 20},\ \texttt{Semester} \rightarrow 2,\ \texttt{Name} \rightarrow \text{Hans Eifrig}\}$$

```
(1223,Hans Eifrig,Seeweg 20,2)
(1223,Seeweg 20,2,Hans Eifrig)
```

what kind of equality would you expect for tuples?

Relation

- A *relation* r over X is a *finite* set $r \subseteq \mathrm{Tup}(X)$.
- The set of all relations over X is denoted $\mathrm{Rel}(X)$.
- $r \in \mathrm{Rel}(X)$ is called an *instance* over X.
- Let R be a *relation name*.
  A *(relation) schema* of R is given as $R(X)$, where X a set of attributes, also called *format* of the schema.
  Instead of writing $R(\{A_1, \ldots, A_k\})$ we may also write $R(A_1, \ldots, A_k)$. k is called the *arity* of R.
  We may also write:

$$R(A_1 : dom(A_1), \ldots, A_k : dom(A_k))$$

Key

- For each schema $R(X)$ we distinguish a set of attributes K we call a *key* of R, $K \subseteq X$.
- Once a key is defined, in every instance r of R for every pair of tuples there holds: if both tuples agree on the attributes forming the key, they have to agree on all their attributes.
- In general, for a schema there may exist several keys.

Database

- A *(relational) database schema* $\mathcal{R}$ is given as a set of relation schemata,

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

  resp. $\mathcal{R} = \{R_1, \ldots, R_m\}$.

- An *Instance* $\mathcal{I}$ of a database schema $\mathcal{R} = \{R_1, \ldots, R_m\}$ is given as a set of finite relations, $\mathcal{I} = \{r_1, \ldots, r_m\}$, where r_i instance of R_i, $1 \leq i \leq m$.
  We may also write

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

Queries

- For any instance $\mathcal{I}$, a query Q defines a relation $Q(\mathcal{I})$, we call the *answer* to Q.
- A query formally is given as a mapping (transformation) from a database instance to a relation instance.
- Analogously to above, we may also write $\mathcal{I}(Q)$ to denote the answer to a query Q with respect to an instance $\mathcal{I}$.

Null value

- We may introduce a *null value*, whenever we want to express, that for some attribute the value is not known.
- The problem with nulls is that there exist several different possible interpretations: *value exists, however currently not known*, *value currently does not exist, however will exist in the future*, *value exits, however is unknown in principal*, and *attribute is not applicable*.

Example

Student

MatrId	Name	Address	Semester
1223	Hans Eifrig	null	2
3434	Lisa Lustig	Bergstraße 11	4
1234	Maria Gut	Am Bächle 1	null