

Datenbanken

Prof. Dr. Romeyke, TH Lübeck

Version 1.2



Agenda

Anwendung relationaler Datenbankkonzepte

Funktionsbereiche eines RDBMS

Datenbankschlüssel

Datenbankoperationen

Das SQL-Datenbankkonzept

SQL: Structured Query Language

Struktur: Datenbanken werden als zweidimensionale Tabellen *veranschaulicht* (die tatsächliche Datenorganisation aber nicht so)

Zeilen: Pro Zeile bildet sich ein Datensatz aus logisch zusammengehörenden Datenobjekten (z.B. pro Zeile ein Kunde mit den Datenobjekten Kundennummer, Name usw.)

Spalte: Pro Spalte findet sich eine Eigenschaft eines Datenobjekts, die in jeder Zeile (also bei jedem Objekt) unterschiedlich ausgeprägt sein kann

Mitarbeitertabelle

Name	Vorname	Geb.dat	Telefon	Email
Müller	Kurt	05.01.77	0231/5554987	kurt.mueller@unserefirma.de
Schneider	Daniela	16.02.80	0231/5554988	daniela.schneider@unserefirma.de
Meyer	Walter	02.07.63	0231/5553967	walter.meyer@unserefirma.de
Schmitz	Michael	25.08.59	0231/5556187	michael.schmitz@unserefirma.de
Wagner	Gaby	18.01.70	0231/5554787	gaby.wagner@unserefirma.de
Feyerabend	Werner	01.04.82	0231/5554997	werner.feyerabend@unserefirma.de
Langmann	Matthias	28.03.76	0231/5551927	matthias.langmann@unserefirma.de

Tabelle 1: Ein Beispiel für eine Mitarbeitertabelle

Begriffe der relationalen Datenbanksysteme¹

Relation: eine Tabelle zur Darstellung von Objekten

Tupel: Die Zeilen der Tabelle entsprechen je einem Tupel

Kardinalität: Anzahl der Tupel in einer Relation

Attribut: Die Spalten einer Tabelle entsprechen je einem Attribut

Grad: Der Grad einer Relation entspricht der Anzahl der Attribute

¹Vgl. Elmasri und Navathe, *Grundlagen von Datenbanksystemen*, S. 135ff.

Datenbankoperationen

- ▶ Alle Operationen eines Relationalen Datenbankmanagementsystems (RDBMS) arbeiten auf Tabellen. Sie generieren neue (Teil-) Tabellen oder löschen diese bzw. Teile davon.
- ▶ Allerdings sind die Tabellen nur spezielle Sichtweisen auf die in der Datenbank gespeicherten Daten.

Funktionsbereiche eines RDBMS (Meta-Ebene)

Datendefinition: Schnittstellen zur Definition der Datenstruktur (also der Tabellenspezifikation): Erstellen, Verändern, Löschen einer Tabelle oder einzelnen Attributen; Verwaltung der sogenannten Metadaten.

DDL: Häufig wird eine eigene *Sprache* verwendet (Data Definition Language, DDL)

Funktionsbereiche eines RDBMS (Daten-Ebene)

Datenmanipulation: Schnittstellen zum Erstellen, Verändern, Löschen von Tupeln oder einzelnen Attributwerten.

DML: Häufig wird eine eigene *Sprache* verwendet (Data Manipulation Language, DML)

Funktionsbereiche eines RDBMS (Betriebs-Ebene)

Manipulationsoptimierung: Ausführung der notwendigen Manipulationstransaktionen in der optimalen Reihenfolge.

Datensicherheit und Datenintegrität: Verhinderung von Inkonsistenzen durch Benutzeraktivitäten, Sicherung gegen Systemausfälle.

Mitarbeitertabelle

Name	Vorname	Geb.dat	Telefon	Email
Müller	Kurt	05.01.77	0231/5554987	kurt.mueller@unserefirma.de
Schneider	Daniela	16.02.80	0231/5554988	daniela.schneider@unserefirma.de
Meyer	Walter	02.07.63	0231/5553967	walter.meyer@unserefirma.de
Schmitz	Michael	25.08.59	0231/5556187	michael.schmitz@unserefirma.de
Wagner	Gaby	18.01.70	0231/5554787	gaby.wagner@unserefirma.de
Feyerabend	Werner	01.04.82	0231/5554997	werner.feyerabend@unserefirma.de
Langmann	Matthias	28.03.76	0231/5551927	matthias.langmann@unserefirma.de

Tabelle 2: Ein Beispiel für eine Mitarbeitertabelle

Grad der Tabelle?	
Kardinalität der Tabelle?	

Tabelle 3: Eigenschaften der Relation

Primärschlüssel einer Relation

Ein **Primärschlüssel** ist eine Menge von Attributen, die für jedes Tupel der Relation grundsätzlich eindeutig ist.

Jedes Objekt (jeder Datensatz) ist damit eindeutig identifizierbar.

Anforderungen an Primärschlüssel

- Abstraktheit:** Ein Primärschlüssel sollte nie Informationen aus dem Anwendungsbereich (also beispielsweise eine von der Fachabteilung vergebene Bezeichnung) enthalten: Jede Änderung des als Primärschlüssel verwendeten Attributs hat ggf. unabsehbare Querwirkungen auf andere Tabellen. Entscheidet sich z.B. die Fachabteilung, den bisher eindeutigen Attributwert in zwei „Versionen“ weiterzuführen, ist keine Eindeutigkeit mehr gegeben.
- Vergabe:** Ein Primärschlüssel sollte deshalb immer aus einem zusätzlichen Schlüsselfeld bestehen, dessen jeweiliger Attributwert vom RDBMS bei der Entstehung eines Tupels vergeben wird.

Mitarbeitertabelle mit Primärschlüssel

ID	Name	Vorname	Geb.dat	Telefon	Email
1	Müller	Kurt	05.01.77	0231/5554987	kurt.mueller@unserefirma.de
2	Schneider	Daniela	16.02.80	0231/5554988	daniela.schneider@unserefirma.de
3	Meyer	Walter	02.07.63	0231/5553967	walter.meyer@unserefirma.de
4	Schmitz	Michael	25.08.59	0231/5556187	michael.schmitz@unserefirma.de
5	Wagner	Gaby	18.01.70	0231/5554787	gaby.wagner@unserefirma.de
6	Feyerabend	Werner	01.04.82	0231/5554997	werner.feyerabend@unserefirma.de
7	Langmann	Matthias	28.03.76	0231/5551927	matthias.langmann@unserefirma.de
8	Wagner	Gaby	18.01.70	0231/5554787	gaby2.wagner@unserefirma.de

Tabelle 4: Ein Beispiel für eine Mitarbeitertabelle mit Primärschlüssel

Auch bei Hinzukommen eines namensgleichen Mitarbeiters ist eine eindeutige Identifizierung gegeben.

Metadaten der Relation *Mitarbeiter*

Attributbezeichnung	Attributtyp
ID	Integer (not null, primary key)
Name	Char
Vorname	Char
Geb.dat	Date
Telefon	Char
Email	Char

Tabelle 5: Metadaten

Fremdschlüssel

Wird ein Primärschlüssel einer anderen Relation als Bezug verwendet, so spricht man von einem Fremdschlüssel. In der Mitarbeitertabelle gibt es keinen Fremdschlüssel. Anders ist das in der Dienstwagentabelle, in der das Attribut Mitarbeiter-ID als Fremdschlüssel auftaucht. Es muß in dieser Tabelle also nicht unbedingt eindeutig sein (es könnte z.B. ein Mitarbeiter mit zwei Dienstwagen ausgestattet sein)

ID	Kennzeichen	Farbe	Fahrzeugtyp-ID	Mitarbeiter-ID
1	DO-WB 421	elfenbein	14	1
2	DO-WB 422	elfenbein	14	3
3	DO-WB 423	elfenbein	14	5
4	DO-WB 424	elfenbein	14	7

Tabelle 6: Dienstwagen

Anlegen einer Relation mit der Data Definition Language in SQL

```
CREATE TABLE 'Mitarbeiter' (  
  'ID' int(11) NOT NULL AUTO_INCREMENT,  
  'Personalnummer' varchar(10) NOT NULL,  
  'Name' varchar(30) NOT NULL,  
  'Vorname' varchar(30) NOT NULL,  
  'Geburtsdatum' date NOT NULL,  
  'Telefon' varchar(30) DEFAULT NULL,  
  'Email' varchar(50) DEFAULT NULL,  
  PRIMARY KEY ('ID')  
)
```


Abfragebeispiele mit der Data Manipulation Language in SQL: **Selektion**

Auswahl von Tupeln (Zeilen) einer Relation

```
SELECT *  
FROM Mitarbeiter  
WHERE Vorname = 'Michael'  
ORDER BY Mitarbeiter.Name ASC
```

ID	Name	Vorname	Geb.dat	Telefon	Email
4	Schmitz	Michael	25.08.59	0231/5556187	michael.schmitz@unserefirma.de
22	Wagners	Michael	18.01.72	0231/5555687	michael.wagners@unserefirma.de

Tabelle 7: Ein Beispiel für eine Selektion

Abfragebeispiele² mit der Data Manipulation Language in SQL: **Projektion**

Auswahl von Attributen (Spalten) einer Relation. Reduktion der Zahl der Tupel möglich (Entfernung redundanter Tupel)

```
SELECT DISTINCT Vorname  
FROM 'Mitarbeiter'  
ORDER BY Vorname
```

Vorname
Angelina
Aydin
Bernhard
Christian
Christina
Frantisek
...

Tabelle 8: Ein Beispiel für eine Projektion (Auszug)

²Vgl. Heuer, Saake und Sattler, *Datenbanken kompakt*, S. 24.

Abfragebeispiele mit der Data Manipulation Language in SQL: **Join**

- ▶ Kombination zweier (oder mehrerer) Relationen über gemeinsame (Schlüssel-) Attribute.
- ▶ Dieser Join erzeugt eine neue Relation, in der jeweils ein Tupel angelegt wird, wenn in beiden Ausgangsrelationen je ein Tupel mit gleichen Werten in den gemeinsamen Attributen vorhanden ist.
- ▶ Verknüpfung der Dienstwagentabelle (Folie 15) und der Mitarbeitertabelle (Folie 13)

Join: Ein Beispiel mit zwei Relationen (Auszug)

```
SELECT Mitarbeiter.Vorname,  
       Mitarbeiter.Name, Dienstwagen.Kennzeichen  
FROM Mitarbeiter, Dienstwagen  
WHERE Mitarbeiter.ID = Dienstwagen.Mitarbeiter_ID
```

Vorname	Name	Kennzeichen
Kurt	Müller	DO-WB 421
Walter	Meyer	DO-WB 422
Gaby	Wagner	DO-WB 423
Matthias	Langmann	DO-WB 424
...		

Tabelle 9: Ein Beispiel für einen Join zweier Relationen in der klassischen Formulierung

Inner Join: gleichwertiger Ausdruck

```
SELECT Mitarbeiter.Vorname,  
       Mitarbeiter.Name, Dienstwagen.Kennzeichen  
FROM Mitarbeiter  
INNER JOIN Dienstwagen ON Mitarbeiter.ID  
                        = Dienstwagen.Mitarbeiter_ID
```

Vorname	Name	Kennzeichen
Kurt	Müller	DO-WB 421
Walter	Meyer	DO-WB 422
Gaby	Wagner	DO-WB 423
Matthias	Langmann	DO-WB 424
...		

Tabelle 10: Ein Beispiel für einen Join zweier Relationen in der JOIN-Formulierung

Unterschied INNER JOIN, OUTER JOIN (hier LEFT)

```
SELECT Mitarbeiter.Vorname,  
       Mitarbeiter.Name, Dienstwagen.Kennzeichen  
FROM Mitarbeiter LEFT JOIN Dienstwagen ON  
Mitarbeiter.ID = Dienstwagen.Mitarbeiter_ID
```

Vorname	Name	Kennzeichen
Kurt	Müller	DO-WB 421
Daniela	Schneider	NULL
Walter	Meyer	DO-WB 422
...		

Tabelle 11: Ein Beispiel für einen OUTER Join

³Siehe auch: https://de.wikibooks.org/wiki/Einf%C3%BChrung_in_SQL:_Arbeiten_mit_JOIN

Risiko beim Formulieren einer Join-Operation

- ▶ Die Datenbank vergleicht alle Tupel aller einbezogenen Relationen miteinander
- ▶ In der Formulierung des Joins muß dafür gesorgt werden, dass nur *logisch sinnvolle* Kombinationen berücksichtigt werden (im vorhergehenden Beispiel ist das die Einschränkung *WHERE Mitarbeiter.ID = Dienstwagen.Mitarbeiter-ID*).
- ▶ Fehlt eine derartige Einschränkung, wird das *Kreuzprodukt*⁴ beider Relationen berechnet.

⁴Vgl. Schubert, *Datenbanken*, S. 97.

Literaturhinweise



Elmasri, Ramez und Shamkant B. Navathe. *Grundlagen von Datenbanksystemen. Ausgabe Grundstudium*. 3. Auflage. München Boston: Pearson Studium, 2007.



Heuer, Andreas, Gunter Saake und Kai-Uwe Sattler. *Datenbanken kompakt. Entwurf von Datenbanken, Einführung in SQL*. 1. Auflage. Landsberg: mitp, 2001.



Schubert, Matthias. *Datenbanken. Theorie, Entwurf und Programmierung relationaler Datenbanken*. 1. Auflage. Stuttgart, Leipzig, Wiesbaden: Teubner, 2004.