



Kommunikation und Datenhaltung

Sprachen zur Datenbankdefinition



Überblick über den Datenhaltungsteil

- Einleitung
 - Motivation und Grundlagen
 - Architektur von Datenbanksystemen
- Datenbankabfragen
 - Relationenmodell und Relationenalgebra
 - Relationale Datenbanksprachen (SQL)
- Datenbankentwurf
 - ER- und EER-Modell
 - Abbildung von ER-Modellen auf das Relationenmodell
 - Relationaler Entwurf
 - Sprachen zur Datenbankdefinition
- Transaktionsverwaltung
- Anfrageoptimierung
- Datenbankanwendungsentwicklung



Datendefinitionssprachen

- Datendefinitionssprachen (Data Definition Language DDLs)
 - Werkzeuge zur Umsetzung eines logischen Entwurfs (DB-Schema) in ein konkretes DBS (konkretes Datenmodell!)
- DDLs existieren für verschiedene Datenmodelle
 - **SQL-DDL**
 - Teil der Standardsprache SQL für relationale Datenbanksysteme
 - **ODL** (Object Definition Language)
 - für objektorientierte Datenbanksysteme nach dem ODMG-Standard.

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



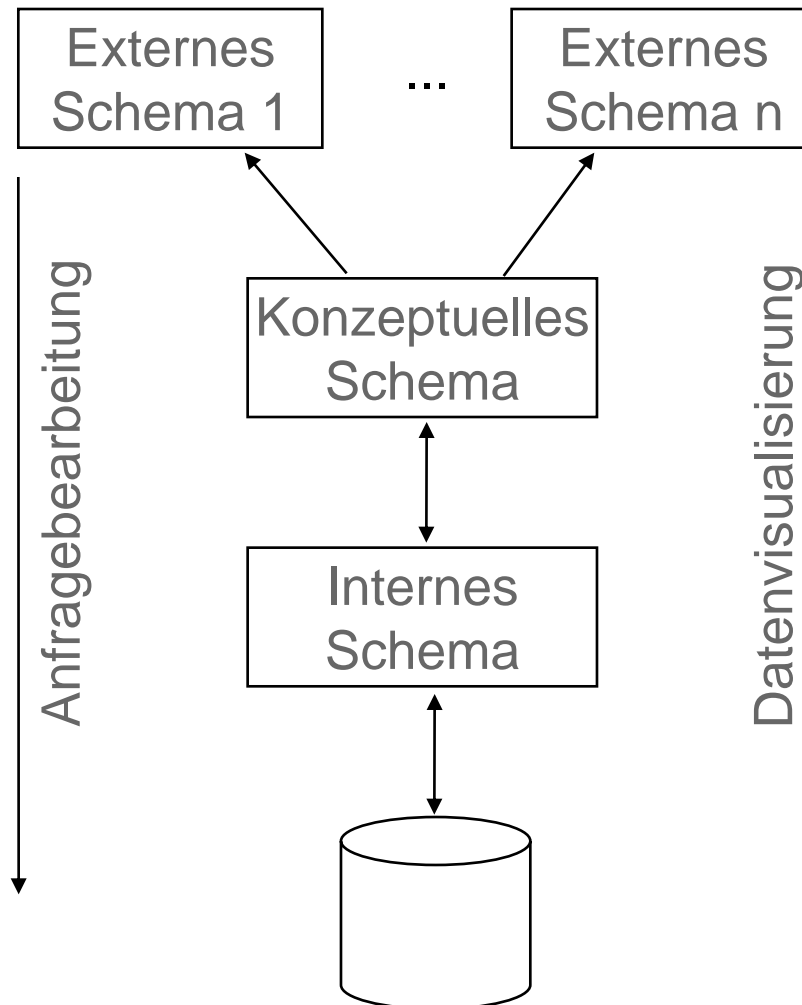
Erzeugung von Relationen

- Schwerpunkt dieser Vorlesung: SQL-DDL für relationale DBS
- Erzeugen von Relationen
 - Festlegung des Schemas
 - Bestimmung der Integritätsbedingungen

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

SQL als Datendefinitionssprache

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

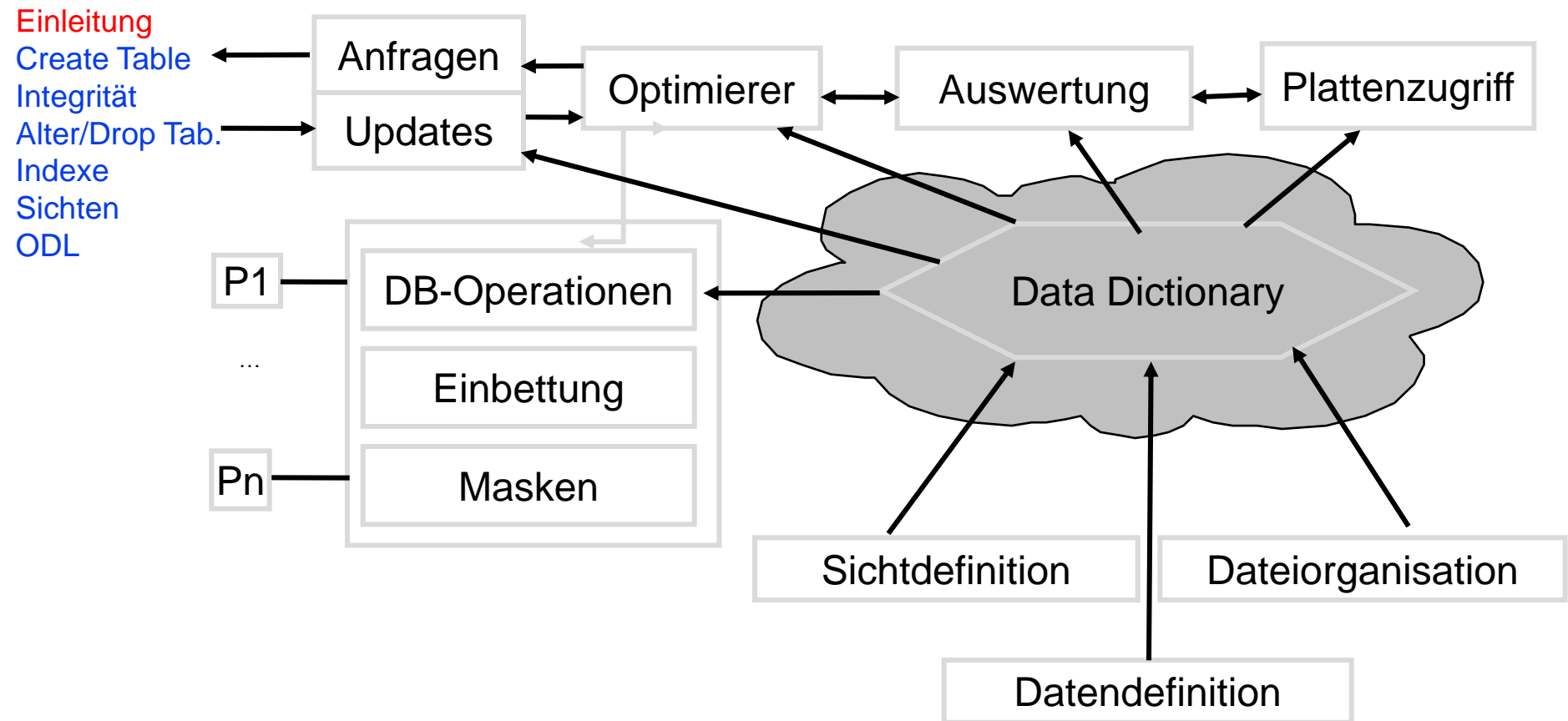


create view
drop view

create table create domain
drop table drop domain
alter table alter domain

create index
drop index
alter index

ANSI-SPARC-Architektur



Data Dictionary – Illustration

AUSLEIH	INV.NR	NAME

BUCH	INV.NR	NAME	TITEL	ISBN	AUTHOR

REL	RNR	NAME
	1	AUSLEIH
	2	BUCH

ATTR	RNR	NAME	TYP	SCHLÜSSEL
	1	INV.NR	int	true
	1	NAME	String	false
	2	INV.NR	int	true
	2	TITEL	String	false

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Anmerkungen (Wdh.)

DVDs	Name	Sprache	Ersch.Jahr
	Red Dragon	englisch	2002
	Amores	spanisch	2001
	Perros		
	Dark Star	NULL	NULL
	irgendwas	englisch	2002

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

- **NULL** als Attributwert grundsätzlich zulässig.
- Ebenso Duplikate, d. h. Relation ist **Multimenge**.
(Solange kein Konflikt mit Schlüsseldefinition.)



Agenda

- Einführung
- **Create Table**
- Integritätsbedingungen
- Alter / Drop Table
- Indexe
- Sichten
- ODL

Die Anweisung create table I

- Syntax:

```
create table relationenname (  
  spaltenname_1 wertebereich_1 [not null],  
  ...  
  spaltenname_k wertebereich_k [not null])
```
- Führt zu
 - Eintrag im Data Dictionary,
 - Erzeugen einer leeren Basisrelation in der Datenbank.

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Vordefinierte Datentypen (Wertebereiche)

- **integer**, int (4 bytes) und **smallint** (2 bytes)
- **float(p)** (oder auch kurz float),
 - p bestimmt die Genauigkeit (sowie Länge der internen Speicherung)
 - Beispiel SQL Server:
 - p zwischen 1 und 53
 - float(24) entspricht real (4 byte), float(53) entspricht double (8 byte)
- **decimal(p,q)** und **numeric(p,q)** (p Stellen insgesamt, davon q nach dem Komma)
- **character(n)** oder char(n), Strings fester Länge
- **varchar(n)** variable Länge, maximal n Zeichen
- **bit(n)** für Bitfolgen,
- **date**, **time** bzw. **timestamp** für Datums-, Zeit- und kombinierte Datums-Zeit-Angaben.

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Die Anweisung create table II

- **not null** schließt *Nullwerte* als Attributwerte in bestimmten Spalten aus.

- im DB-Kontext
 - *null* -> keine Information,
 - *null* \neq 0

- Beispiel:

```
create table Bücher (  
    ISBN char(10) not null,  
    Titel varchar(200),  
    Verlagsname varchar(30))
```



Anzeigen des Data Dictionary

- Beispiel Oracle:
 - `select * from user_tables;`
 - `select * from user_tab_columns;`
- Beispiel Mysql:
 - `show databases;`
 - `show tables;`
 - `desc <relationenname>;`

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Beispiel

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

```
mysql> create table buecher (  
-> ISBN char(10) not null,  
-> Titel varchar(200),  
-> Verlagsname varchar(30));  
Query OK, 0 rows affected (0.07 sec)  
  
mysql> desc buecher;  
+-----+-----+-----+-----+-----+-----+  
| Field          | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| ISBN           | char(10)      | NO   |     |         |       |  
| Titel          | varchar(200)  | YES  |     | NULL    |       |  
| Verlagsname    | varchar(30)   | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql> █
```

„not null“ - Constraint

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

```
mysql> create table buecher (  
  -> ISBN char(10) not null,  
  -> Titel varchar(200),  
  -> Verlagsname varchar(30));  
Query OK, 0 rows affected (0.07 sec)  
  
mysql> desc buecher;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| ISBN       | char(10)      | NO   |     |          |       |  
| Titel      | varchar(200)  | YES  |     | NULL    |       |  
| Verlagsname | varchar(30)   | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql> insert into buecher values (NULL, 'Datenbanksysteme', 'mitp');  
ERROR 1048 (23000): Column 'ISBN' cannot be null  
mysql> █
```

Überlegungen dazu

- Warum sinnvoll, Constraints (wie **not null**) in der Datenbank zu speichern?
 - „Anwendung kann es doch auch sicherstellen.“
- Datenintegrität!
 - Philosophie war, möglichst viel ans DBMS zu delegieren.
 - Mehrere Anwendungen, eine macht es nicht – mögliche Fehlerquelle.
- Außerdem Redundanzfreiheit (auf Code-Ebene)
 - Sonst müsste jede Anwendung dieses Constraint umsetzen

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Standardbelegungen

- Spaltendefinitionen können über die Klausel **default** `wert` zusätzlich mit Standardwerten versehen werden
- Wird anstelle eines Nullwerts beim Einfügen gesetzt, wenn ein Attribut nicht gesetzt ist

- Beispiel:

```
create table Bücher (  
    ISBN char(10) not null,  
    Titel varchar(200),  
    Verlagsname varchar(30) default 'Springer')
```



Definition eines Wertebereichs

- **create domain**-Anweisung:
Benutzerdefinierte Wertebereiche auf Basis vorhandener Datentypen
- Wird (in der Regel) verwendet, um Wertebereich eines Datentyps einzuschränken
- Besteht aus:
 - Name des Wertebereichs
 - Defaultwert
 - Wertebereichseinschränkung
(in Form einer Integritätsbedingung: “check-Klausel”;
s. Abschnitt “[Integrität](#)”)

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Definition eines Wertebereichs

```
create domain Gebiete varchar(20)  
    default 'Informatik'  
create table Vorlesungen (  
    V_Bezeichnung varchar(80) not null,  
    SWS smallint,  
    Semester smallint,  
    Studiengang Gebiete)  
create table Mitarbeiter (  
    PANr integer not null,  
    AngNr char(10) not null,  
    Fachbereich Gebiete,  
    Gehalt decimal(10,2),  
    Raum integer,  
    Einstellung date)
```

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Beispiel mit Wertebereichseinschränkung später



Agenda

- Einführung
- Create Table
- **Integritätsbedingungen**
- Alter / Drop Table
- Indexe
- Sichten
- ODL

Schlüssel I

- Zur **eindeutigen** Identifikation eines Tupels
- Kann aus beliebig vielen **Primattributen** bestehen
 - **Primattribut**: ein Element eines Schlüssels.
- Meist mehrere Schlüssel möglich
 - Gute Schlüssel: kleine Attributmenge mit Schlüsseleigenschaft
 - Entscheidung ist anwendungsspezifisch!

INV.NR	TITEL	ISBN	AUTOR
0007	Dr. No	3-324	Fleming
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-345	Vossen
4712	Datenbanken	3-345	Vossen
4717	PASCAL	3-989	Wirth

Schlüssel II

- Schlüssel bilden **funktionale Abhängigkeiten** auf das Relationenschema ab

- t ist ein Tupel aus der Relation R ,
- a und b sind Teilmengen der Attribute aus t
- $a \rightarrow b$ ist eine funktionale Abhängigkeit

es gilt: $\forall t_1, t_2 \in R: t_1(a) = t_2(a) \rightarrow t_1(b) = t_2(b)$

INV.NR	TITEL	ISBN	AUTOR
0007	Dr. No	3-324	Fleming
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-345	Vossen
4712	Datenbanken	3-345	Vossen
4717	PASCAL	3-989	Wirth

Was sind hier geeignete Schlüssel?

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Fremdschlüssel I

- Abbilden von Beziehungen zwischen zwei Relationen
- Beispiel: INV.NR in AUSLEIH ist **Fremdschlüssel**.

AUSLEIH	INV.NR	NAME
	4711	Meyer
	1201	Schulz
	0007	Müller
	4712	Meyer

BUCH	INV.NR	TITEL	ISBN	AUTOR
	0007	Dr. No	3-324	Fleming
	1201	Objektbanken	3-111	Heuer
	4711	Datenbanken	3-345	Vossen
	4712	Datenbanken	3-345	Ullman
	4717	PASCAL	3-989	Wirth

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Fremdschlüssel II

- Attribut, auf das der Fremdschlüssel verweist, muss Schlüssel der anderen Relation sein
- DBMS stellt sicher, dass zu jedem Fremdschlüssel ein Tupel in der verbundenen Relation gehört!
 - Vor allem bei Änderungsoperationen (insert, update, delete)

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

(Primär-) Schlüssel in SQL

- Definition von Schlüsseln

```
create table Bücher (  
    ISBN char(10),  
    Titel varchar(200),  
    Verlagsname varchar(30),  
    primary key (ISBN),  
    foreign key (Verlagsname)  
    references Verlage (Verlagsname)  
)
```

ein oder
mehrere
Attribute

primary key
impliziert
not null

- **primary key** kann wie not null auch direkt hinter dem Attribut angegeben werden
 - Falls Primärschlüssel nur aus 1 Attribut besteht
 - “Spaltenbedingungen” vs. “Tabellenbedingungen”

Fremdschlüssel in SQL

- Definition von Fremdschlüsselbeziehungen

```
create table Bücher (  
    ISBN char(10),  
    Titel varchar(200),  
    Verlagsname varchar(30),  
    primary key (ISBN),  
    foreign key (Verlagsname)  
    references Verlage (Verlagsname)
```

„Verlagsname“ in
Relation „Bücher“
verweist auf
„Verlagsname“ in
Relation „Verlage“

Attributname hier in
beiden Relationen
identisch, das muss
aber nicht so sein!



Fremdschlüssel in SQL

- Pro Tabelle ist ein Primärschlüssel zulässig, aber mehrere Fremdschlüssel
- Weitere Schlüssel können durch die **unique**-Klausel definiert werden (wird wie **primary key** benutzt)
- Wichtig: Reihenfolge der Tabellendefinition bei Fremdschlüsseln
 - Referenzierte Tabelle muss existieren
 - Alternativ: Nachträgliches Hinzufügen (Ändern der Tabellendefinition)

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Duplikate im Primärschlüssel

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

```
mysql> create table buecher (  
  -> ISBN char(10) primary key,  
  -> Titel varchar(200),  
  -> Verlagsname varchar(30));  
Query OK, 0 rows affected (0,08 sec)  
  
mysql>  
mysql>  
mysql> insert into buecher values (1, 'Datenbanksysteme', 'mitp');  
Query OK, 1 row affected (0,00 sec)  
  
mysql> insert into buecher values (1, 'Database Systeme', 'AddisonWesley');  
ERROR 1062 (23000): Duplicate entry '1' for key 1  
mysql>
```

Primärschlüssel ist NULL

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

```
mysql> desc buecher;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ISBN  | char(10) | NO   | PRI |          |       |
| Titel  | varchar(200) | YES |     | NULL    |       |
| Verlagsname | varchar(30) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> insert into buecher values (NULL, 'Database Systeme', 'AddisonWesley');
ERROR 1048 (23000): Column 'ISBN' cannot be null
mysql>
```

Fremdschlüssel nicht in der DB

- Einleitung
- Create Table
- Integrität**
- Alter/Drop Tab.
- Indexe
- Sichten
- ODL

```
mysql> select * from verlag;
+-----+-----+
| Name          | Ansprechpartner |
+-----+-----+
| AddisonWesley | Herr Schmitt    |
| Fachbuchverlag | Frau Mustermann |
| Pearson       | Herr Schmitt    |
+-----+-----+
3 rows in set (0.00 sec)

mysql> insert into buecher values (1, 'Datenbanksysteme', 'mitp');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`example/buecher`, CONSTRAINT `buecher_ibfk_1` FOREIGN KEY (`Verlagsname`) REFERENCES `verlag` (`Name`))
mysql>
```



Integritätsbedingungen mit check I

- **create domain** Gebiete *varchar(20)*
 default 'Informatik',
 check (
 value in ('Informatik',
 'Mathematik',
 'Elektrotechnik',
 'Linguistik')
)
)

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Integritätsbedingungen mit check II

```
create table Vorlesungen (  
    V_Bezeichnung varchar(80)  
        primary key,  
    SWS smallint,  
        check (SWS  $\geq$  0),  
    Semester smallint,  
        check (Semester between 1 and 9),  
    Studiengang Gebiete )
```

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Integritätsbedingungen mit check III

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

```
create table Buch_Versionen (  
  ISBN char(10),  
  Auflage smallint, check(Auflage > 0),  
  Jahr integer,  
    check (Jahr between 1800 and 2020),  
  Seiten integer, check(Seiten > 0),  
  Preis decimal(8,2), check(Preis ≤ 250),  
    primary key (ISBN, Auflage),  
    foreign key (ISBN)  
      references Bücher (ISBN),  
    check ((select sum(Preis) from  
Buch_Versionen) <  
      (select sum(Budget) from Lehrstühle))
```

→ Allgemeines check-Constraint, das sich nicht auf einzelne Attribute beschränkt



Agenda

- Einführung
- Create Table
- Integritätsbedingungen
- **Alter / Drop Table**
- Indexe
- Sichten
- ODL

Die Anweisung drop table

- Löschen von Tabellendefinitionen
- **drop table** relationenname
 { **restrict** | **cascade** }
- Beispiel: **drop table** buch
- Wirkung:
 - Löschen des Relationenschemas aus Katalog
 - Löschen der eigentlichen Relation aus Datenbank
- **cascade** (optional): Alle Sichten und Integritätsbedingungen zur Relation werden ebenfalls gelöscht
- **restrict** (Default): drop wird zurückgewiesen, falls Sichten oder Integritätsbedingen existieren

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Ändern von Tabellendefinitionen

- Mögliche Änderungen für Spalten
 - Hinzufügen
 - Ändern
 - Löschen
- Mögliche Änderungen für Integritätsbedingungen
 - Hinzufügen
 - Löschen

Ändern von Spalten

- Syntax :
`alter table relationenname modifikation`
- Mögliche Änderungen (modifikation) :
 - `add column` spaltendefinition
 - Spaltendefinition wie gehabt
 - `drop column` spaltenname {restrict | cascade}
 - Restrict und cascade wie beim Löschen von Tabellen
 - `alter column` spaltenname **set default** defaultwert
 - Umbenennungen oder die Änderung von Datentypen sind nicht möglich

alter table am Beispiel

```
alter table Lehrstühle  
    add column Budget decimal (8,2)
```

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

- Wirkung:
 - Änderung des Relationenschemas im Data Dictionary.
Im Beispiel wird dem Relationenschema `Lehrstühle` ein neues Attribut zugeordnet.
 - Erweiterung der existierenden Basisrelation um ein Attribut, das bei jedem existierenden Tupel mit **null** besetzt wird.
- Angabe von Default-Werten:

```
alter table Lehrstühle  
    alter column Budget set default 10000
```

alter table in der Praxis

- Beispiel:

```
alter table Lehrstühle
    add column Budget decimal(8,2) default 10000
```

- Oracle ergänzt Default-Wert automatisch in allen vorhandenen Tupeln.
- MSSQL:
 - Wenn neue Spalte NULL-Werte verbietet, Default-Wert zu existierenden Tupeln hinzufügen.
 - Ansonsten: Existierende Tupel werden nicht aktualisiert – es sei denn, explizite WITH VALUES Klausel:

```
alter table Lehrstühle
    add column Budget decimal(8,2) default 10000 with values
```

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Hinzufügen von Integritätsbedingungen

- Alle Integritätsbedingungen lassen sich über die add-Klausel nachträglich definieren

- Syntax :

```
alter table relationenname  
    add constraint Name constraint
```

- Beispiel:

```
alter table bücher  
    add constraint kbkey primary key(ISBN)  
  
alter table mitarbeiter  
    add constraint M_Eindeutig unique(Name, Geb_tag)
```

Name der Integritätsbedingung
(auch bei Definition in create table möglich)



Löschen von Integritätsbedingungen

- Löschen von Integritätsbedingungen ist über den Namen möglich (benutzer- oder systemdefiniert)
- Syntax :

```
alter table relationenname  
drop constraint Name
```



Agenda

- Einführung
- Create Table
- Integritätsbedingungen
- Alter / Drop Table
- **Indexe**
- Sichten
- ODL

Index – Motivation

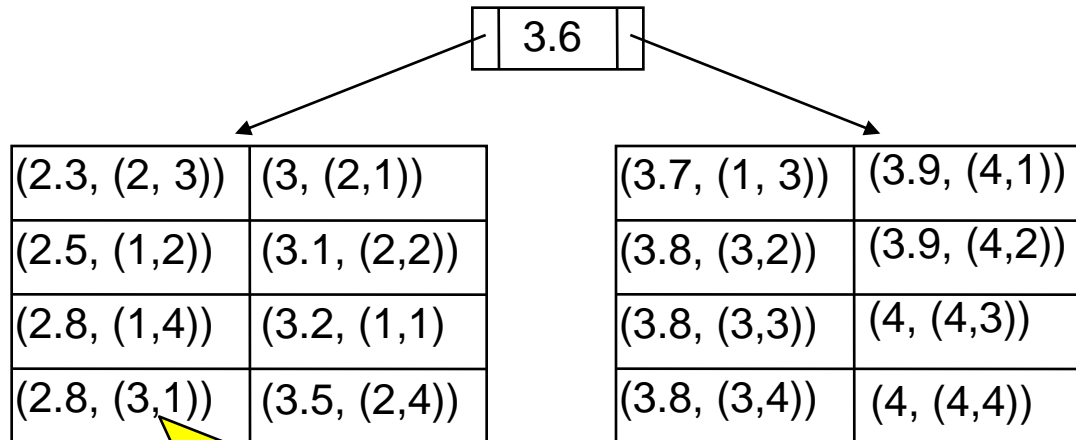
- Seitenweise Anordnung der Daten.
 - Seiten – Einheiten des Zugriffs.
 - Problem: auf welcher Seite liegen die gesuchten Daten?
- Daten müssen im Hauptspeicher vorliegen, damit Selektion etc. durchgeführt werden kann.
 - Laden einer Seite in den Hauptspeicher ist teuer → **Zugriffslücke**.

Tom, 20, 3.2, EE	Mary, 24, 3, ECE	Lam, 22, 2.8, ME	Chris, 22, 3.9, CS
Chang, 18, 2.5, CS	James, 24, 3.1, ME	Kathy, 18, 3.8, LS	Vera, 17, 3.9, EE
Bob, 21, 3.7, CS	Chad, 28, 2.3, LS	Kane, 19, 3.8, ME	Louis, 32, 4, LS
Pat, 19, 2.8, EE	Leila, 20, 3.5, LS	Martha, 29, 3.8, CS	Shideh, 16, 4, CS

Index – Illustration

- Student(name, age, gpa, major)
- Non-clustered secondary B⁺-tree für gpa.

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Indexwert **Seite, Position**

GPA: grade point average, MAJOR: Hauptfach

Tom, 20, 3.2, EE	Mary, 24, 3, ECE	Lam, 22, 2.8, ME	Chris, 22, 3.9, CS
Chang, 18, 2.5, CS	James, 24, 3.1, ME	Kathy, 18, 3.8, LS	Vera, 17, 3.9, EE
Bob, 21, 3.7, CS	Chad, 28, 2.3, LS	Kane, 19, 3.8, ME	Louis, 32, 4, LS
Pat, 19, 2.8, EE	Leila, 20, 3.5, LS	Martha, 29, 3.8, CS	Shideh, 16, 4, CS

Indexe – Beispiel

- Anfrage
select name **from** Student **where** gpa = 4.0
- Anfrage liefert Ergebnis,
unabhängig davon, ob jener Index existiert
oder nicht – physische Datenunabhängigkeit.
- Wenn Index existiert
 - erkennt DBMS das
und nutzt ihn für die Anfrage-Evaluierung,
 - erhebliche Beschleunigung.

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Index – Erläuterungen I

- Index für mehrere Attribute möglich.
- Index für (gpa, name) nicht dasselbe wie für (name, gpa).

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Index – Illustration II

- Index für (gpa, name):

3.6

((2.3, Chad), (2, 3))	((3, Mary), (2, 1))
((2.5, Chang), (1, 2))	((3.1, James), (2, 2))
((2.8, Lam), (3, 1))	((3.2, Tom), (1, 1))
((2.8, Pat), (1, 4))	((3.5, Leila), (2, 4))

((3.7, Bob), (1, 3))	((3.9, Chris), (4, 1))
((3.8, Kane), (3, 3))	((3.9, Vera), (4, 2))
((3.8, Kathy), (3, 2))	((4, Louis), (4, 3))
((3.8, Martha), (3, 4))	((4, Shideh), (4, 4))

- Suche nach gpa sowie nach gpa und name.
- Suche nach name.

Tom, 20, 3.2, EE
Chang, 18, 2.5, CS
Bob, 21, 3.7, CS
Pat, 19, 2.8, EE

Mary, 24, 3, ECE
James, 24, 3.1, ME
Chad, 28, 2.3, LS
Leila, 20, 3.5, LS

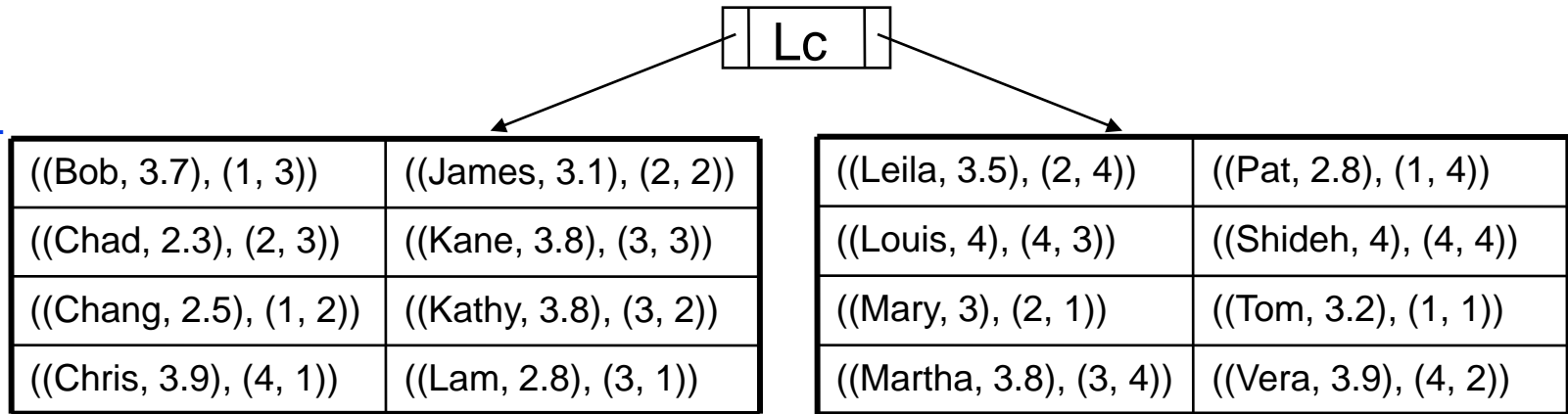
Lam, 22, 2.8, ME
Kathy, 18, 3.8, LS
Kane, 19, 3.8, ME
Martha, 29, 3.8, CS

Chris, 22, 3.9, CS
Vera, 17, 3.9, EE
Louis, 32, 4, LS
Shideh, 16, 4, CS

Index – Illustration III

- Index für (name, gpa):

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Tom, 20, 3.2, EE
Chang, 18, 2.5, CS
Bob, 21, 3.7, CS
Pat, 19, 2.8, EE

Mary, 24, 3, ECE
James, 24, 3.1, ME
Chad, 28, 2.3, LS
Leila, 20, 3.5, LS

Lam, 22, 2.8, ME
Kathy, 18, 3.8, LS
Kane, 19, 3.8, ME
Martha, 29, 3.8, CS

Chris, 22, 3.9, CS
Vera, 17, 3.9, EE
Louis, 32, 4, LS
Shideh, 16, 4, CS



Index – Erläuterungen II

- Man kann Index nachträglich anlegen; man kann Index wieder löschen, ohne die Daten selbst zu löschen.
- Index ist Bestandteil der physischen Ebene. Index-Definition ist Bestandteil des internen Schemas.

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Die Anweisung create index I

- SQL-89: Bestandteil der Norm – seit SQL-92 aber nicht mehr! (Anweisungen der internen Ebene gestrichen)

```
create [unique] index indexname  
      on relationenname (  
      spaltenname_1 ordnung_1, ...,  
      spaltenname_k ordnung_k)
```

- Ordnung: **ascending** oder **descending**
- Beispiel:
CREATE INDEX typ **ON** auto (hersteller,
modell, baujahr)
 - Reihenfolge der Attribute → Sortierreihenfolge.
 - Index typ hilft uns bei Suche nach Herstellern,
aber nur bedingt bei Suche gemäß Baujahr.

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Die Anweisung create index II

- **CREATE UNIQUE INDEX** `typ ON auto`
(`hersteller, modell, baujahr`)
 - ‚unique‘ – keine identischen Schlüssel im Index, d.h. keine zwei Tupel mit identischem Hersteller, Modell und Baujahr erlaubt
 - Zusätzlicher Constraint für Basisrelation (d.h., System überwacht zusätzlich Einhaltung dieser Bedingung)
 - Wie erreicht man dies ohne einen Index dafür zu nutzen?
 - Zusammenhang Schlüssel – Unique – Unique-Index!

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Simulierte Schlüsselbedingung

```
create table Bücher (  
    ISBN char(10) not null,  
    Titel varchar(200),  
    Verlagsname varchar(30))  
  
create unique index Buchindex  
    on Bücher  
    (ISBN asc)
```

- Was passiert, wenn ich Index nachträglich anlege, das entsprechende Attribut **unique**-Eigenschaft aber nicht erfüllt?
- Was heißt überhaupt 'nicht erfüllt'?
- **drop index** Buchindex

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Agenda

- Einführung
- Create Table
- Integritätsbedingungen
- Alter / Drop Table
- Indexe
- **Sichten**
- ODL



Sprachen und Sichten I

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

- Sichten erlauben die Strukturierung einer DB, angepasst an bestimmte Anwendungen
 - ‚Virtuelle‘ Datenbestände – 3 Ebenen Architektur
- Beispiel: Zugriffsschutz
- Hintergrund:
 - SQL als Standard-Anfragesprache.
 - Anfrageergebnis auch wieder Tabelle
→ Anfragen auf Anfrageergebnisse möglich!



Sprachen und Sichten II

- Definition von Benutzersichten:
 - Datenbankabfragen (Queries) können unter einem „Sichtnamen“ als „virtuelle“ Tabelle in DB gespeichert werden.
 - Sicht = Query (+ Name)

create view Name as {beliebige SQL-Anfrage}

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Sprachen und Sichten III

- Zugrundeliegende Relation:
MGA (Mitarbeiter, Gehalt, Abteilung)

MGA	Mitarbeiter	Gehalt	Abteilung
	Buchmann	15	DKE
	Duckstein	15	DKE
	Saake	25	DB

- **create view** vielverdiener **as**
select Mitarbeiter, Gehalt
from MGA **where** Gehalt > 20

vielverdiener	Mitarbeiter	Gehalt
	Saake	25

- View dann verwendbar wie ‚normale‘ Relation
select * **from** vielverdiener

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Sprachen und Sichten IV

- Sichten wird aus DB abgeleitet \Rightarrow kann diese also nur einschränken (Daten ausblenden)
- Unterschiedliche Benutzer sehen unterschiedlichen Ausschnitt der Datenbank.
- Beispiel – Universität
(Studierende, Dozenten, Prüfungsamt, Dekanat etc.)
- Vorteile:
 - Zugriffsschutz,
 - Übersichtlichkeit,
 - Vereinfachung von Anfragen

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Sprachen und Sichten V

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

- Zugrundeliegende Relation:
MGA(Mitarbeiter, Gehalt, Abteilung)
- **create view** vielverdiener **as**
select Mitarbeiter, Gehalt
from MGA **where** Gehalt > 20
- Zugrundeliegende Relation ändert sich, z. B.:
ESD(Emp, Salary, Department)
- Sicht anpassen:
create view vielverdiener **as**
select Emp **as** Mitarbeiter,
Salary **as** Gehalt
from ESD **where** Salary > 20
- Anfrage auf View bleibt unverändert!
- Wie heißt diese Eigenschaft?



Agenda

- Einführung
- Create Table
- Integritätsbedingungen
- Alter / Drop Table
- Indexe
- Sichten
- **ODL**



ODL (Object Definition Language)

- ODL Teil des ODMG-Standards (Object Data Management Group)
 - Schemadefinition im Objektmodell
 - OQL (Object Query Language)
- Hintergrund der Betrachtung:
 - Bisher immer nur Relationales Modell
 - Es gibt aber ganze Reihe weiterer Modelle
 - Mächtiger als RM – erlauben native Umsetzung von (E)ER / UML Konzepten, die im RM nicht umsetzbar sind!
 - OO-Modell ist wichtig – aktuelle „relationale“ Systeme sind objektrelational!

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL



Objektorientiertes Modell: ODL

```
interface Student : Person (  
    extent Studenten, key matrnr)  
    attribute char matrnr[6];  
    attribute string studienfach;  
    attribute set<struct<float note, string fach>>  
                                                zeugnis;  
  
    relationship Person mutter inverse Person::kind;  
    relationship Person vater inverse Person::kind;  
  
    float durchschnittsnote ()  
        raises (keine_note);  
    void exmatrikulation (in string art)  
        raises (buecher_ausgeliehen);  
}
```

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Erläuterung ODL-Beispiel I

Schnittstelle zum Objekttyp `Student` beschreibt

- *Typhierarchie*: Angabe der Obertypen hinter dem Typnamen (hier: Obertyp `Person`),
- *Extension*, in der die aktuell erzeugten Objekte vom Typ `Student` gesammelt werden sollen (hier: Extension mit dem Namen `Studenten`),
- *Schlüssel* des Objekttyps, eine Auswahl der Attribute, die zur eindeutigen Identifizierung der Objekte unabhängig von der Objektidentität verwendet werden können (hier: nur das Attribut `matrnr`),
- *Attribute* mit Datentypen und Namen.

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL

Erläuterung ODL-Beispiel II

- *Beziehungen* zu anderen Klassen mit dem Wortsymbol **relationship**
 - auch inverse Beziehungen,
 - ermöglichen Wahl zwischen 1:1-, 1:n, und n:m-Kardinalitäten,
 - hier: zwei 1:n-Beziehungen `Vater` und `Mutter` zwischen `Studenten` und `Personen`, da nur die Rückrichtung einen `Set`-Typ enthält: `set<Person> kind` (im Bsp. nicht dargestellt)
- *Methoden* mit ihrer Schnittstelle und einer spezifizierten Ausnahmebehandlung, die im Fehlerfall ausgelöst wird, etwa bei Verletzung von Integritätsbedingungen.
- Implementierungsdetails werden in die Klasse (*class*) verschoben

Einleitung
Create Table
Integrität
Alter/Drop Tab.
Indexe
Sichten
ODL