

Kapitel 2: Datenbank- definitionssprachen

Datenbank-Technologie – Vielfalt

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- Thema dieser Vorlesung (im wesentlichen):
Relationale Datenbanken –
zugrundeliegende Struktur sind Relationen.
- Es gibt auch andere Arten von Datenbanken,
z. B. objektorientierte Datenbanken.
- Objekte anstelle von Tupeln.

Erzeugung von Relationen

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- Vor dem Einfügen von Daten in Datenbank
müssen wir Relationen erzeugen.
- Erzeugung einer Relation
== Festlegung, welches Schema.

Aspekte von SQL

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- SQL – standardisierte Sprache
für den Datenbank-Zugriff
(relationale Datenbanken). Mehrere Aspekte:
 - ◆ Schemadefinition,
 - ◆ Datenmanipulation
(Einfügen, Löschen, Ändern),
 - ◆ Anfragen.
- Gegenstand dieses Kapitels: Schemadefinition.

Datenbankdefinitionssprachen

Einleitung

create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- **SQL-DDL**
Teil der Standardsprache
für relationale Datenbanksysteme: SQL.
(DDL = Data Definition Language)
- **ODL (Object Definition Language)**
für objektorientierte Datenbanksysteme
nach dem ODMG-Standard.

Relationales Modell: SQL-DDL

Einleitung

create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- SQL-DDL umfaßt alle Klauseln von SQL,
die mit Definition von
- Typen,
 - Wertebereichen,
 - Relationenschemata,
 - Integritätsbedingungen
zu tun haben.

Anforderungen an eine relationale DDL

Einleitung

create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- Nach Codd 1982 Sprachmittel zur Definition von
 1. Attributen,
 2. Wertebereichen,
 3. Relationenschemata,
 4. Primärschlüsseln,
 5. Fremdschlüsseln.
- Praxis SQL-89: Relationenschemata mit
 - ◆ Attributen und
 - ◆ Wertebereichen.
- Ab SQL-92 vollständig.
D. h. Schlüssel- und Fremdschlüsseldefinitionen
möglich.

Unterschiede zur relationalen Algebra

Einleitung

create table
Integritäts-
bed.
alter/
drop table
Index
ODL

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

SQL als Definitionssprache (1)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- *Externe Ebene*
 - ◆ **create view,**
 - ◆ **drop view;**
- *Konzeptuelle Ebene*
 - ◆ **create table,**
 - ◆ **alter table,**
 - ◆ **drop table;**

SQL als Definitionssprache (2)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- *Konzeptuelle Ebene (erst in SQL-92)*
 - ◆ **create domain,**
 - ◆ **alter domain,**
 - ◆ **drop domain;**
- *Interne Ebene*
 - ◆ **create index,**
 - ◆ **alter index,**
 - ◆ **drop index.**

Data Dictionary – Illustration

AUSLEIH	INV.NR	NAME
...

BUCH	INV.NR	TITEL	ISBN	AUTOR
...

REL	R.NR	NAME
	1	AUSLEIH
	2	BUCH

ATTR	R.NR	NAME	TYP	SCHLÜSSEL
	1	INV.NR	INT	true
	1	NAME	STRING	false
	2	INV.NR	INT	true
	2	TITEL	STRING	false

Z

Die Anweisung create table (1)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- **Syntax:**
create table basisrelationenname (
spaltenname_1 wertebereich_1 [**not null**],
...
spaltenname_k wertebereich_k [**not null**])
- **Führt zu**
 - ◆ Eintrag in Data Dictionary,
 - ◆ Vorbereitung der „leeren Basisrelation“ in der Datenbank.
- **Data Dictionary Zugriff mit Oracle beispielsweise:**
 - ◆ **select * from user_tables;**
 - ◆ **select * from user_tab_columns**

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

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

Abfrage starten

time	percent	rows	query
0.200258	98.95%	0	create table Bücher (ISBN char(10) not null, Titel varchar(200), VName varchar(30))
0.002127	1.05%	0	commit
0.202385			

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
select * from Bücher
```

Zuvor wurde eingegeben:
INSERT INTO Bücher
VALUES ('12345', 'Datenbanken', 'MITP');
commit

Abfrage starten

```
select * from Bücher
```

ISBN	TITEL	VName
12345	Datenbanken	MITP

time	percent	rows	query
0.001728	100%	1	select * from Bücher
0.001728			

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
select * from Bücher
```

Zuvor wurde eingegeben:
INSERT INTO Bücher
VALUES ('54321', 'Tante Julia', NULL);
commit

Abfrage starten

```
select * from Bücher
```

ISBN	TITEL	VName
12345	Datenbanken	MITP
54321	Tante Julia	

time	percent	rows	query
0.001733	100%	2	select * from Bücher
0.001733			

Die Anweisung create table (2)

Einleitung
 create table
 Integritäts-
 bed.
 alter/
 drop table
 Index
 ODL

- **not null** schließt *Nullwerte* als Attributwerte in bestimmten Spalten aus.
- **create table** Bücher (ISBN *char(10) not null*, Titel *varchar(200)*, Verlagsname *varchar(30)*)
- Primärschlüssel sollte nullwertfrei sein.
- Warum sinnvoll? „Anwendung kann es doch auch sicherstellen.“ – Ja, aber:
 - ◆ Philosophie: Möglichst viel an DBMS delegieren.
 - ◆ Mehrere Anwendungen, eine macht es nicht – mögliche Fehlerquelle.

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
INSERT INTO Bücher VALUES (NULL, 'Der Oberst hat niemand ...',
'Suhrkamp');
commit
```

Abfrage starten

Warning: OCIStmtExecute: ORA-01400: Einfügen von NULL in ("VLDKE"."BÜCHER"."ISBN") nicht möglich in /home/iti_ex/ralfduck/public_html/test.php on line 27

time	percent	rows	query
0.004051	59.53%	0	INSERT INTO Bücher VALUES (NULL, 'Der Oberst hat niemand ...', 'Suhrkamp')
0.002754	40.47%	0	commit
0.006805			

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
select * from Verlage
```

Abfrage starten

Zuvor wurde eingegeben:

```
create table Verlage (Vname varchar(30),
Stadt varchar(30), primary key (Vname))
INSERT INTO Verlage VALUES ('Volksstimme', 'Magdeburg');
INSERT INTO Verlage VALUES ('Fischer', 'Frankfurt');
INSERT INTO Verlage VALUES ('Suhrkamp', 'Frankfurt');
INSERT INTO Verlage VALUES ('Springer', 'Heidelberg');
```

VNAME	STADT
Volksstimme	Magdeburg
Fischer	Frankfurt
Suhrkamp	Frankfurt
Springer	Heidelberg

time	percent	rows	query
0.00303	100%	4	select * from Verlage
0.00303			

Erlaubte Wertebereiche in create table

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- *integer* (oder auch *integer4*, *int*), *smallint* (oder auch *integer2*),
- *float(p)* (oder auch kurz *float*),
- *decimal(p,q)* und *numeric(p,q)* mit jeweils *q* Nachkommastellen,
- *character(n)* (oder kurz *char(n)*, bei *n = 1* auch *char*) für Strings fester Länge *n*,
- *character varying(n)* (oder kurz *varchar(n)*) für Strings variabler Länge bis zur Maximallänge *n*,
- *bit(n)* oder *bit varying(n)* analog für Bitfolgen,
- *date*, *time* bzw. *timestamp* für Datums-, Zeit- und kombinierte Datums-Zeit-Angaben.

Integritätsbedingungen (1)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

VORNAME	NAME	STRASSE	ALTER	BERUF
Erik	Buchmann	Breiter Weg	26	Informatiker
Gunter	Saake	Waldweg	43	Professor
Klemens	Böhm	Nordstrasse	28	Professor
Peter	Oehm	Jahnstrasse	31	Ingenieur
Jan	Oberle	Flotowstrasse	33	Ingenieur
Andreas	Müller	Neckarstrasse	28	Informatiker
Ralf	Duckstein	Goethestrasse	25	Informatiker

- Schlüssel kann aus beliebig vielen Attributen bestehen. Primattribute.
- Entscheidung, welche Attribute Schlüssel bilden, ist anwendungsspezifisch.
- I. a. gibt es mehrere Schlüssel.
- **Primattribut**: Element eines Schlüssels.

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

Integritätsbedingungen (2)

- 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	3-125	James Bond
	1201	Objektbanken	3-111	Heuer
	4711	Datenbanken	3-765	Vossen
	4712	Datenbanken	3-891	Ullman
	4717	PASCAL	3-999	Wirth

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

SQL-89 Level 2 mit IEF

- Zweite Stufe der SQL-89-Norm sieht Zusatz *IEF (Integrity Enhancement Feature)* vor.
- Definition von Schlüsseln und Fremdschlüsseln.

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

Beispiel Tabellendefinition mit IEF

```
create table Bücher (  
  ISBN char(10) not null,  
  Titel varchar(200),  
  Verlagsname varchar(30),  
  primary key (ISBN),  
  foreign key (Verlagsname)  
  references Verlage (Verlagsname) )
```

einen oder
mehrere

Relationen und Attribute,
wo Fremdschlüssel
als Schlüssel vorkommt.

Attribute sind hier namensgleich,
das muß aber nicht so sein!

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

create table in SQL-92

- **create table** Bücher (
 ISBN char(10),
 Titel varchar(200),
 Verlagsname varchar(30),
 primary key (ISBN),
 foreign key (Verlagsname)
 references Verlage (Verlagsname)
)
- **not null** implizit durch die **primary key**-Klausel.

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
INSERT INTO Bücher2 VALUES ('34567', 'Wer hat Palomino ...', 'Suhrkamp');
commit
```

Abfrage starten

Warning: OCISStmtExecute: ORA-00001: Verstoß gegen Eindeutigkeit, Regel (VLDKE.SYS_C009083) in /home/iti_ex/ralfdock/public_html/test.php on line 27

time	percent	rows	query
0.010268	88.61%	0	INSERT INTO Bücher2 VALUES ('34567', 'Wer hat Palomino ...', 'Suhrkamp')
0.00132	11.39%	0	commit
0.011588			

Zuvor wurde eingegeben:

```
create table Bücher2 (ISBN char(10) not null, Titel varchar(200),
Verlagsname varchar(30), primary key (ISBN), foreign key (Verlagsname)
references Verlage (Verlagsname) )
INSERT INTO Bücher2 VALUES ('12345', 'Datenbanken', 'Springer');
INSERT INTO Bücher2 VALUES ('23456', 'Weltalmanach', 'Fischer');
INSERT INTO Bücher2 VALUES ('34567', 'Der Oberst hat niemand ...',
'Suhrkamp');
```

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
INSERT INTO Bücher2 VALUES (NULL, 'Datenbanken', 'MITP');
commit
```

Abfrage starten

Warning: OCISStmtExecute: ORA-01400: Einfügen von NULL in ("VLDKE"."BÜCHER2"."ISBN") nicht möglich in /home/iti_ex/ralfdock/public_html/test.php on line 27

time	percent	rows	query
0.004281	64.46%	0	INSERT INTO Bücher2 VALUES (NULL, 'Datenbanken', 'MITP')
0.00236	35.54%	0	commit
0.006641			

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

```
INSERT INTO Bücher2 VALUES ('45678', 'Datenbanken', 'MITP');
commit
```

Abfrage starten

Warning: OCISStmtExecute: ORA-02291: Verstoß gegen Constraint (VLDKE.SYS_C009084). Übergeordn. Schlüssel nicht gefunden in /home/iti_ex/ralfdock/public_html/test.php on line 27

time	percent	rows	query
0.011096	89.35%	0	INSERT INTO Bücher2 VALUES ('45678', 'Datenbanken', 'MITP')
0.001322	10.65%	0	commit
0.012418			

Verlag 'MITP' nicht in Datenbank.

Erweiterungen in SQL-92

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

Neben Primär- und Fremdschlüssel in SQL-92:

- **default-Klausel:** Defaultwerte für Attribute,
- **create domain-Anweisung** benutzerdefinierte Wertebereiche,
- **check-Klausel** weitere lokale Integritätsbedingungen innerhalb der zu definierenden Wertebereiche, Attribute und Relationenschemata.

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
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 )
```

Integritätsbedingungen mit check (1)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

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

Integritätsbedingungen mit check (2)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

```
• create table Vorlesungen (  
  V_Bezeichnung varchar(80) not null,  
    primary key,  
  SWS smallint, check(SWS ≥ 0),  
  Semester smallint, check(  
    Semester between 1 and 9),  
  Studiengang Gebiete )
```

Integritätsbedingungen mit check (3)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
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))
```

Integritätsbedingungen mit check (4)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

```
create table Buch_Versionen (  
ISBN char(10),  
Auflage smallint, check(Auflage > 0),  
Seiten integer, check(Seiten > 0),  
Preis decimal(8,2),  
Autor char(20),  
check (Autor in ('Klemens', 'Erik','Ralf')))
```

alter table und drop table

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- Syntax des **alter table**-Kommandos in SQL-89:
alter table basisrelationenname
add spaltenname wertebereich
alter table Lehrstühle
add Budget *decimal(8,2)*
- 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.

alter table-Kommando in SQL-2

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- Statt
add spaltenname wertebereich
auch Angabe von Default-Werten
und **check**-Klauseln erlaubt:
 - ◆ **add** Budget *decimal(8,2)* **default** 10000,
check (Budget > No_Planstellen * 1000)
 - ◆ **alter table** bücher4
add constraint kbkey **primary key**(ISBN)

alter- und drop-Klausel für Attribute (1)

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

- Die Klausel
alter spaltenname default_änderung
nur Änderung der Defaultwerte,
nicht Änderung von Datentypen.

Integritätsbedingung – Illustration

- 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	3-125	James Bond
	1201	Objektbanken	3-111	Heuer
	4711	Datenbanken	3-765	Vossen
	4712	Datenbanken	3-891	Ullman
	4717	PASCAL	3-999	Wirth

alter- und drop-Klausel für Attribute (2)

- Die Klausel
`drop spaltenname { restrict | cascade }`
erlaubt Löschen von Attributen, falls
 - ◆ keine Sichten und Integritätsbedingungen mit Hilfe dieses Attributs definiert wurden (im Fall **restrict**),
 - ◆ oder mit gleichzeitiger Löschung dieser Sichten und Integritätsbedingungen (im Fall **cascade**).
- Beispiel:
`alter table buch_versionen
drop column auflage`

Die Anweisung drop table

- `drop table basisrelationenname { restrict | cascade }`
- **restrict** und **cascade** analog zum **drop** bei Attributen.

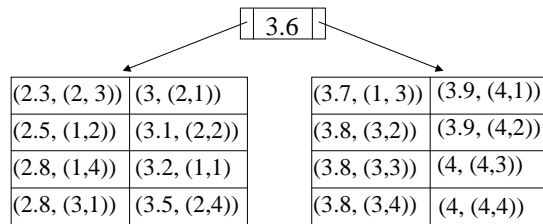
Index – Illustration (1)

Erläuterung:

- Seitenweise Anordnung der Daten.
- Daten müssen im Hauptspeicher vorliegen, damit Selektion etc. durchgeführt werden kann.
- Seiten – Einheiten des Zugriffs.
- Laden einer Seite in den Hauptspeicher ist teuer, *Zugriffslücke*.

Index – Illustration (2)

- Student(name, age, gpa, major); t(Student) = 16.
- Non-clustered primary B+-tree für Attribut gpa.



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

Klemens Böhm

IWM: Datenbankdefinitionssprachen – 41

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

Index – Erläuterungen

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

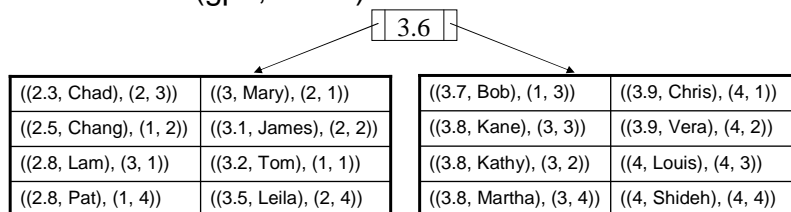
Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

Klemens Böhm

IWM: Datenbankdefinitionssprachen – 42

Index – Illustration (3)

- Index für (gpa, name):



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

- Suche nach gpa sowie nach gpa und name.
- Suche nach name.
(Ausgefeiltes DBMS würde es implementieren.)

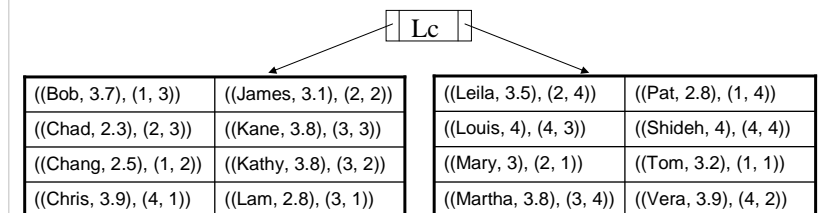
Klemens Böhm

IWM: Datenbankdefinitionssprachen – 43

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

Index – Illustration (4)

- Index für (name, gpa):



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

Klemens Böhm

IWM: Datenbankdefinitionssprachen – 44

Einleitung
create table
Integritäts-
bed.
alter/
drop table
Index
ODL

Indices und physische Datenunabhängigkeit – Beispiel (1)

- 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.

Indices und physische Datenunabhängigkeit – Beispiel (2)

- Anfrage
select name **from** Student
where gpa > 4.2 **and** age=27
- Ohne Index: Möglicherweise effizienter,
erst das age-Prädikat auszuwerten; unklar.
- Mit Index: I. d. R. erst gpa-Prädikat auswerten.
- Datenbank findet in beiden Fällen
überlegene Alternative –
physische Datenunabhängigkeit.

Die Anweisung create index

- SQL-89: Bestandteil der Norm
create [**unique**] **index** indexname
 on basisrelationenname (
 spaltenname_1 ordnung_1,
 ...,
 spaltenname_k ordnung_k
)
)
- ‚unique‘ – Schlüsselbedingung.
- Beispiel:
CREATE [UNIQUE] INDEX typ **ON** auto
(hersteller, modell, baujahr)
- Reihenfolge der Spaltennamen –
Sortierreihenfolge.
Index typ hilft uns bei Suche nach Herstellern,
aber nur bedingt bei Suche gemäß Baujahr.

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?

## Objektorientiertes Modell: ODL

Einleitung  
create table  
Integritäts-  
bed.  
alter/  
drop table  
Index  
ODL

```
interface Student : Person (
 extent Studenten, key matrnrl
 attribute char matrnrl[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);
}
```

## Erläuterung ODL-Beispiel (1)

Einleitung  
create table  
Integritäts-  
bed.  
alter/  
drop table  
Index  
ODL

Schnittstelle zum Objekttyp `Personen` 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 `matrnrl`),
- *Attribute* mit Datentypen und Namen.

## Erläuterung ODL-Beispiel (2)

Einleitung  
create table  
Integritäts-  
bed.  
alter/  
drop table  
Index  
ODL

- *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`.
- *Methoden* mit ihrer Schnittstelle und einer spezifizierten Ausnahmebehandlung, die im Fehlerfall ausgelöst wird, etwa bei Verletzung von Integritätsbedingungen.

## Mögliche Prüfungsfragen

- Was ist ein (Datenbank-)Schema?
- Was ist das Data Dictionary?
- Welche Integritätsbedingungen kennen Sie?
- Warum sollte man sich die Mühe machen, Integritätsbedingungen als Teil des Datenbank-Schemas zu formulieren?
- Sind Integritätsbedingungen Bestandteil des internen oder des konzeptuellen Schemas? Begründen Sie Ihre Antwort.
- Was ist ein Index? Wieso sind Indices Bestandteil des internen und nicht des konzeptuellen Schemas?