Karlsruher Institut für Technologie

Datenbanksysteme

Kapitel 3: Datenbank-Definitionssprachen

Lehrstuhl für Systeme der Informationsverwaltung, Fakultät für Informatik

Photo by Wolf-Ulf Wulfrolf

Lernziele für heute …

- Kenntnis der Konzepte zur Definition von Datenbankschemata im relationalen Modell

- Fähigkeit zur Implementierung eines Datenbankschemas

Photo by Jonathan Cohen

Beispiel Realwelt: Flugwesen (1)

- Reale, anfassbare Gegenstände:
  - Flugzeuge mit Flugzeugtyp, Besatzung, Sitzplätzen, Laderäumen, Farbanstrichen;
  - Flughäfen mit Bezeichnung, Flugsteigen, Passagieraufkommen, Rollbahnen;
  - Passagiere mit Namen, Anschriften, Telefonnummern, Größe, Gewicht, Haarfarbe, Religionszugehörigkeit;
  - Flugscheine mit Vordrucken und Aufdrucken.
  - Gepäckstücke der Passagiere.
  - Zeitschriften mit Verlag und Auflage, die auf den Flügen ausgegeben werden.
  - …

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Beispiel Realwelt: Flugwesen (1)

- Reale, anfassbare Gegenstände:
  - Flugzeuge mit Flugzeugtyp, Besatzung, Sitzplätzen, Laderäumen, Farbanstrichen;
  - Flughäfen mit Bezeichnung, Flugsteigen, Passagieraufkommen, Rollbahnen;
  - Passagiere mit Namen, Anschriften, Telefonnummern, Größe, Gewicht, Haarfarbe, Religionszugehörigkeit;
  - Flugscheine mit Vordrucken und Aufdrucken.
  - Gepäckstücke der Passagiere.
  - Zeitschriften mit Verlag und Auflage, die auf den Flügen ausgegeben werden.
  - …

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Gewinnung der Konventionen (1)

- Beschränkte Anwendungswelt, die **Miniwelt** (bzw. **relevanter Weltausschnitt** oder **Diskursbereich**).
- Daten – **Modelle** (gedankliche Abstraktionen) der Miniwelt.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Gewinnung der Konventionen (2)

- **Datenbasis ist bedeutungstreu, wenn ihre Elemente Modelle einer gegebenen Miniwelt sind. (Datenbasiskonsistenz)**
- Datenbasiskonsistenz
  – schärfste Konsistenzforderung.

Bedeutungs-treue

SQL-DDL

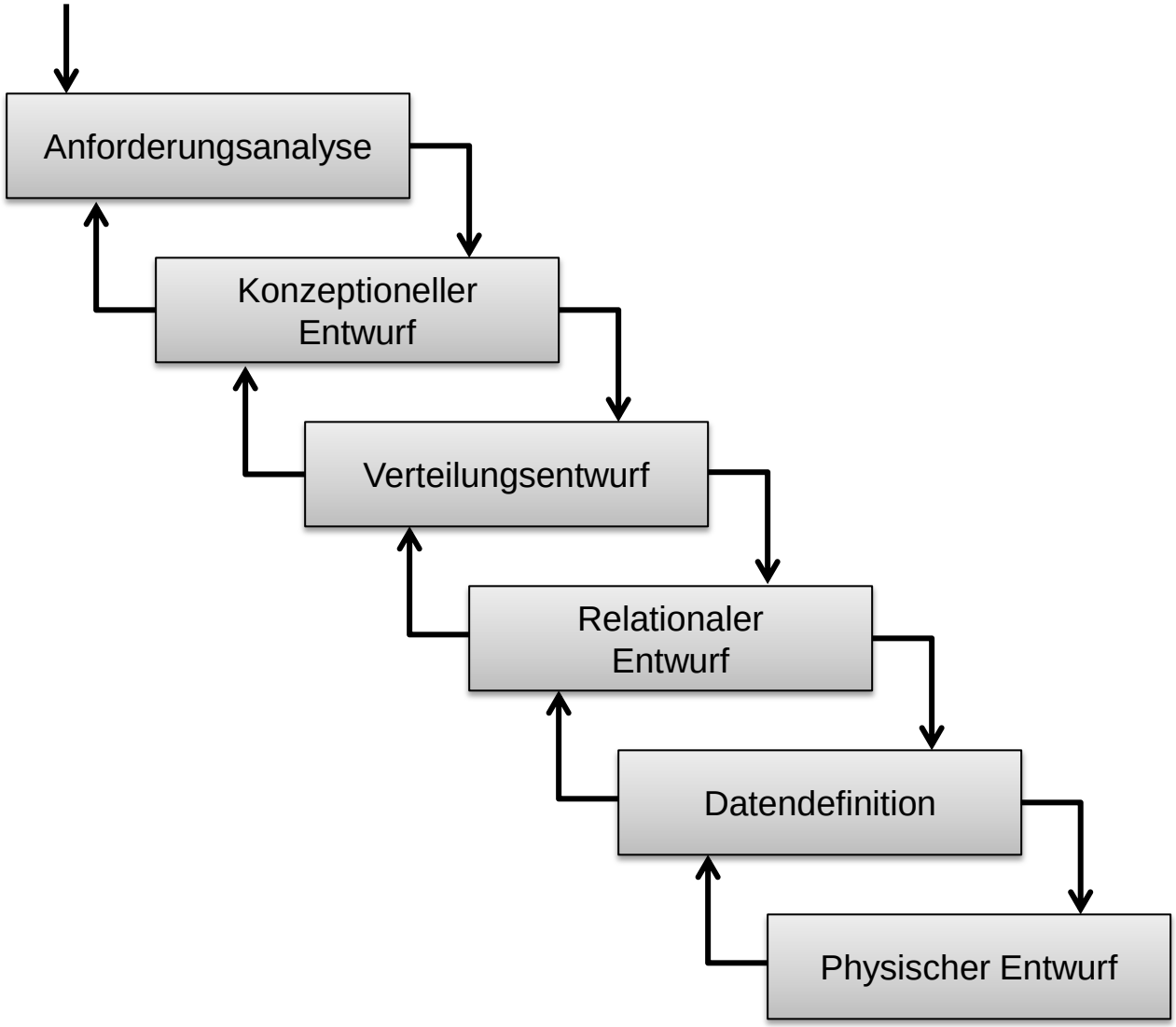
create table

Integritäts-bed.

alter/
drop table

Index

Phasenmodell des Datenbankentwurfs



Datenbasisschemata (1)

- Vorgehen bei der Modellierung
  (als erstem Schritt der Systemunterstützung):
  Wir versuchen, Ausschnitt der Wirklichkeit
  mit Schema zu beschreiben.
- Typen beschreiben die Struktur der Entitäten.

Bedeutungs-
treue

SQL-DDL

create table

Integritäts-
bed.

alter/
drop table

Index

Beispiel

Datenbasistyp Buchungsdatenbasis

Typ Flughafen ::= **datensatz** [
FlughCode: string, Stadt: string, Land: string,
Name: string, Zeitzone: int]

Typ Flugzeugtyp ::= **datensatz** [
FtypId: string, Name: string, First: int,
Business: int, Economy: int]

Typ Flug ::= **datensatz** [
FlugNr: string, von: string, nach: string,
FtypId: string, Wochentag: string,
Abflugszeit: time, Ankunftszeit: time,
Entfernung: int]

Typ Flughäfen ::= **menge** {Flughafen} **key** FlughCode

Typ Flugzeugtypen ::= **menge** {Flugzeugtyp} **key** FtypId

Typ Flüge ::= **menge** {Flug} **key** FlugNr

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/
drop table

Index

Weitere denkbare Konsistenzbedingungen

Datenbasistyp Buchungsdatenba

Typ Flughafen ::= **datensatz** [
FlughCode: string, Stadt: string, Land: string,
Name: string, Zeitzone: int]

Typ Flugzeugtyp ::= **datensatz** [
FtypId: string, Name: string, First: int,
Business: int, Economy: int]

Typ Flug ::= **datensatz** [
FlugNr: string, von: string, nach: string,
FtypId: string, Wochentag: string,
Abflugszeit: time, Ankunftszeit: time,
Entfernung: int]

Typ Flughäfen ::= **menge** {Flughafen} **key** FlughCode

Typ Flugzeugtypen ::= **menge** {Flugzeugtyp} **key** FtypId

Typ Flüge ::= **menge** {Flug} **key** FlugNr

Land → *Zeitzone*

Wert unter *von* kommt als Wert unter *FlughCode* vor

(*von*,*nach*) → *Entfernung*

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Weitere denkbare Konsistenzbedingungen

Datenbasistyp Buchungsdatenbasis

Typ Flughafen ::= **datensatz** [
FlughCode: string, Stadt: string, Land: string,
Name: string, Zeitzone: int]

Typ Flugzeugtyp ::= **datensatz** [
FtypId: string, Name: string, First: int,
Business: int, Economy: int]

Typ Flug ::= **datensatz** [
FlugNr: string, von: string, nach: string,
FtypId: string, Wochentag: string,
Abflugszeit: time, Ankunftszeit: time,
Entfernung: int]

Typ Flughäfen ::= **menge** {Flughafen} **key** FlughCode

Typ Flugzeugtypen ::= **menge** {Flugzeugtyp} **key** FtypId

Typ Flüge ::= **menge** {Flug} **key** FlugNr

Welche weiteren Konsistenz-bedingun-gen sind sinnvoll?

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/
drop table

Index

Datenbasisschemata (3)

- Weitere Begrifflichkeit: *Schema-Konsistenz*.
- Einhaltung vorgegebener Konsistenzbedingungen Teil der Schema-Konsistenz.
- Dass Konsistenzbedingung vorgegeben ist, heißt nicht, dass sie der Wirklichkeit entspricht.
- System kann Schema-Konsistenz überprüfen, nicht aber Datenbasiskonsistenz, deshalb wichtig.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Konsistente Zustände

Legitime Zustände
durch Ausführen
der Operatoren
mit Interpretation
des Schemas
Schemakonsistenz

Idealfall: volle, aktuelle
Übereinstimmung von
Datenbasis und Miniwelt
Datenbasiskonsistenz

Ist interessant,
weil vom DBMS überprüfbar.

Konsistenz: Grad an Bedeutungstreue
einer Datenbasis

Bedeutungs-
treue

SQL-DDL

create table

Integritäts-
bed.

alter/
drop table

Index

Aspekte von SQL

- SQL – standardisierte Sprache für den Datenbank-Zugriff (relationale Datenbanken). Mehrere Aspekte:
  - Schemadefinition,
  - Datenmanipulation (Einfügen, Löschen, Ändern),
  - Anfragen.
- Gegenstand dieses Kapitels: Schemadefinition.
- SQL-DDL – Teil der Standardsprache. (DDL = Data Definition Language)

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Relationales Modell: SQL-DDL

SQL-DDL umfasst alle Klauseln von SQL, die mit Definition von

- Typen,
- Wertebereichen,
- Relationenschemata,
- Integritätsbedingungen zu tun haben.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Anmerkungen

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

Bedeutungs-
treue

SQL-DDL

create table

Integritäts-
bed.

alter/
drop table

Index

SQL als Definitionssprache (1)

- *Externe Ebene*
  - **CREATE VIEW**
  - **DROP VIEW**
- *Konzeptuelle Ebene*
  - **CREATE TABLE**
  - **ALTER TABLE**
  - **DROP TABLE**

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/
drop table

Index

SQL als Definitionssprache (2)

- *Konzeptuelle Ebene*
  - **CREATE DOMAIN**
  - **ALTER DOMAIN**
  - **DROP DOMAIN**
- *Interne Ebene*
  - **CREATE INDEX**
  - **ALTER INDEX**
  - **DROP INDEX**

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Data Dictionary – Illustration

Titel

TITLE ID	NAME	ART	GRÖSSE	KID
…	…			

Künstler

KID	NAME	LAND	JAHR
…	…	…	…

REL

R.NR	NAME
1	Titel
2	Künstler

ATTR

R.NR	NAME	TYP	SCHLÜSSEL
1	TITLE ID	INT	true
1	NAME	STRING	false
2	KID	INT	true
2	NAME	STRING	false
…	…	…	…

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/
drop table

Index

Die Anweisung create table (1)

- 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`


Firefox
[redacted] / localhost / musicd...
https://[redacted]myadmin4/index.php?token=8b5d9d0f25045cb72588ec04e418c64d#PMAURL-5:db_sql.php?db=musicdb&table=&server=1&target=&token=8b5d9d0f25045cb72588ec04e418c64d
Google
Meistbesucht
Erste Schritte
Aktuelle Nachrichten
mufubox
Server: localhost » Datenbank: musicdb
Struktur
SQL
Suche
Abfrage
Exportieren
Importieren
Operationen
Routinen
Mehr
MySQL lieferte ein leeres Resultat zurück (d.h. null Datensätze). (Die Abfrage dauerte 0.0093 Sekunden)
create table Künstler (KID integer, NAME varchar(200), LAND varchar(50) not null, JAHR integer, primary key (KID))
[Inline] [Bearbeiten] [PHP-Code erzeugen]
SQL-Querybox anzeigen

Die Anweisung create table (2)

- **NOT NULL** schließt *Nullwerte* als Attributwerte aus.
- Anwendung kann es zwar auch sicherstellen,
  das entspricht jedoch nicht unserer Philosophie.
  (Insbesondere kritisch, wenn mehrere Anwendungen.)


Firefox
[redacted] / localhost / musicd...
https://[redacted]phpmyadmin4/index.php?token=8b5d9d0f25045cb72588ec04e418c64d#PMAURL-5:db_sql.php?db=musicdb&table=&server=1&target=&token=8b5d9d0f25045cb72588ec04e418c64d
Google
Meistbesucht
Erste Schritte
Aktuelle Nachrichten
mufubox
Server: localhost » Datenbank: musicdb
Struktur
SQL
Suche
Abfrage
Exportieren
Importieren
Operationen
Routinen
Mehr
Fehler
SQL-Befehl:
INSERT INTO Künstler VALUES (123, "Raul Seixas", NULL, 1945)
MySQL meldet:
#1048 - Column 'LAND' cannot be null
SQL-Befehl(e) in Datenbank musicdb ausführen:
1 INSERT INTO Künstler VALUES (123, "Raul Seixas", NULL, 1945)

Erlaubte Wertebereiche in create table (1)

- *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*,
  Illustration
- *bit(n)* oder *bit varying(n)* analog für Bitfolgen,
- *date*, *time* bzw. *timestamp* für Datums-, Zeit- und kombinierte Datums-Zeit-Angaben.

Erlaubte Wertebereiche in create table (2)

- Entscheidung zwischen *char(n)* und *varchar(n)*
  sollte streng genommen
  nicht dem Anwendungsentwickler obliegen,
  steht im Gegensatz zu physischer Datenunabhängigkeit.

Integritätsbedingungen (1)

SNUSER

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

- Schlüssel kann aus beliebig vielen Attributen bestehen. *Primattribute*.
- Entscheidung, welche Attribute Schlüssel bilden, ist anwendungsspezifisch.
- I. Allg. gibt es mehrere Schlüssel.

Bedeutungs-treue
SQL-DDL
create table
Integritäts-bed.
alter/ drop table
Index

Integritätsbedingungen (2)

- KID in Titel ist Fremdschlüssel.

Künstler

KID	NAME	LAND	JAHR
1012	Neil Young	Kanada	1945
1014	Rammstein	Deutschland	1994
1015	The Ramones	USA	1974
1016	Eric Fish	Deutschland	1969

Titel

TITLE ID	NAME	ART	GRÖSSE	KID
102	Neil Young – Heart of Gold	mp3	2.920kb	1012
103	Rammstein – Ich liebe Neil Young	wma	4.234kb	1014
104	Neil Young – Old Man	mp3	3.161kb	1012
105	Neil Young – Four Strong Winds	wma	5.125kb	1012

Beispiel

```
CREATE TABLE Titel(
    TITLEID integer NOT NULL,
    NAME varchar(200), …,
    KID integer,
    PRIMARY KEY(TITLEID),
    FOREIGN KEY(KID)
    REFERENCES Künstler(KID) )
```

einen oder mehrere

Relationen und Attribute, wo Fremdschlüssel als Schlüssel vorkommt.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Weitere Integritätsbedingungen

Neben Primär- und Fremdschlüssel:

- **DEFAULT** - Klausel: Defaultwerte für Attribute,
- **CREATE DOMAIN** - Anweisung benutzerdefinierte Wertebereiche,
- **CHECK** - Klausel weitere lokale Integritätsbedingungen innerhalb der zu definierenden Wertebereiche, Attribute und Relationenschemata.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

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 )
```

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Integritätsbedingungen mit check

```
CREATE TABLE Vorlesungen (
    V_Bezeichnung varchar(80) NOT NULL,
        PRIMARY KEY,
    SWS smallint, CHECK(SWS ≥ 0),
    Semester smallint, CHECK(
        Semester BETWEEM 1 AND 9),
    Studiengang Gebiete )
```

- Beliebig komplexe Bedingungen möglich, unter Verwendung von SQL.
- Jede Änderung zieht Ausführung einer solchen Anfrage nach sich. Kann beliebig teuer werden!
- Obwohl Teil des SQL-Standards, i. Allg. keine Systemunterstützung.

ALTER TABLE und DROP TABLE

- Syntax des **ALTER TABLE**-Kommandos:

```
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.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

alter table-Kommando

- Bei

  ADD spaltenname wertebereich

  auch Angabe von Default-Werten und **check**-Klauseln erlaubt:

  - ```
    ADD Budget decimal(8,2) DEFAULT 10000,
       CHECK(Budget >
          No_Planstellen * 1000)
    ```
  - ```
    ALETR TABLE bücher
    ADD CONSTRAINT kbkey PRIMARY KEY(ISBN)
    ```

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Integritätsbedingung – Illustration

- KID in Titel ist Fremdschlüssel.

Künstler

KID	NAME	LAND	JAHR
1012	Neil Young	Kanada	1945
1014	Rammstein	Deutschland	1994
1015	The Ramones	USA	1974
1016	Eric Fish	Deutschland	1969

Titel

TITLE ID	NAME	ART	GRÖSSE	KID
102	Neil Young – Heart of Gold	mp3	2.920kb	1012
103	Rammstein – Ich liebe Neil Young	wma	4.234kb	1014
104	Neil Young – Old Man	mp3	3.161kb	1012
105	Neil Young – Four Strong Winds	wma	5.125kb	1012

Was passiert, wenn Spalte KID aus Künstler entfernt wird?

alter- und drop-Klausel für Attribute

- 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**).

Die Anweisung drop table

- **DROP TABLE** basisrelationenname
  { **RESTRCIT** | **CASCADE**}
- **restrict** und **cascade** analog zum **drop** bei Attributen.

Z

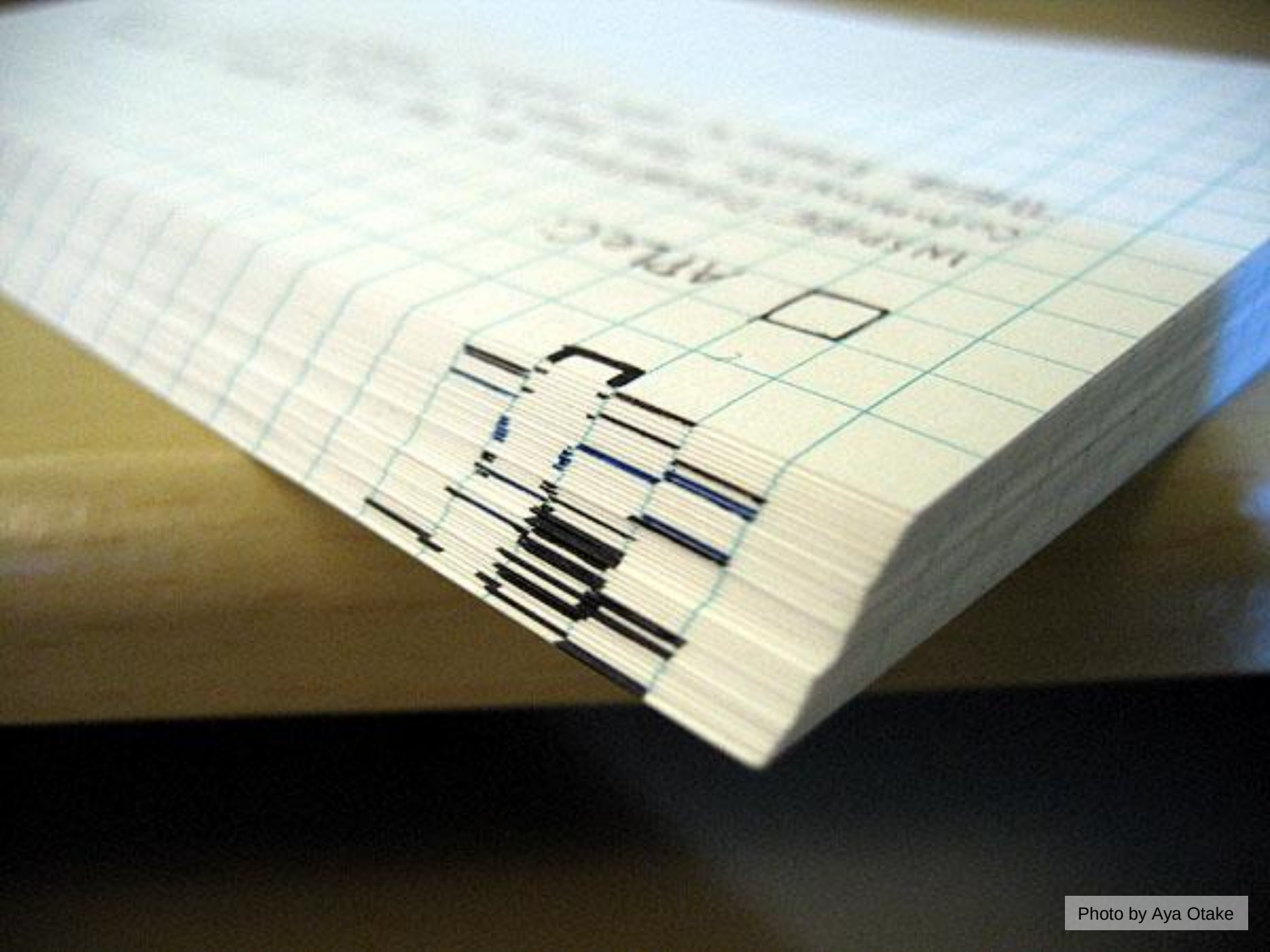
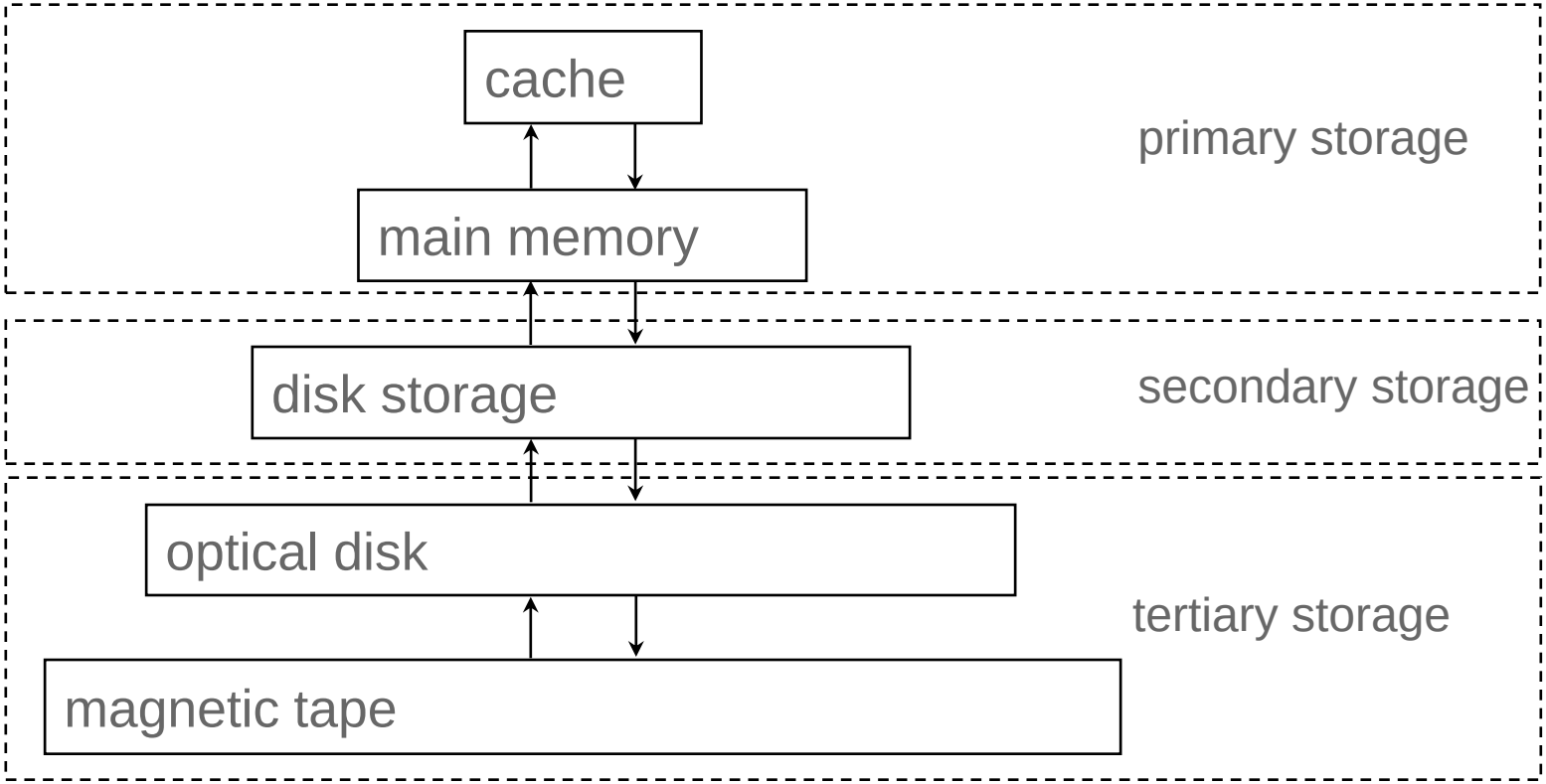


Photo by Aya Otake

Speicherhierarchie (1)

Primär-, Sekundär- und Tertiärspeicher



Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/
drop table

Index

Speicherhierarchie (2)

- Speicherhierarchie: Server zur Datenverwaltung verfügt über mehrere Speicher.
- Primärspeicher
  – kann direkt von CPU zugegriffen werden.
- Programme residieren im main memory, werden zur Ausführung in Cache/Register geladen.
- Dabei sind Daten im Sekundärspeicher dauerhaft.
- Daten werden zur Verarbeitung in Primärspeicher kopiert. (Zugriffslücke.)
- Von oben nach unten: Speicher wird größer, Zugriff darauf aber langsamer.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

Speicherhierarchie (3)

- Tertiärspeicher:
  Jukebox o. ä. zur Verwaltung der Disks, Tapes.
  Ein wesentlicher Kostenfaktor
  für Zugriff auf Tertiärspeicher:
  Anzahl der (Disk-)Wechsel.

- Gilt für lokale Rechner ebenso wie für Cloud-Rechner.

Bedeutungs-
treue

SQL-DDL

create table

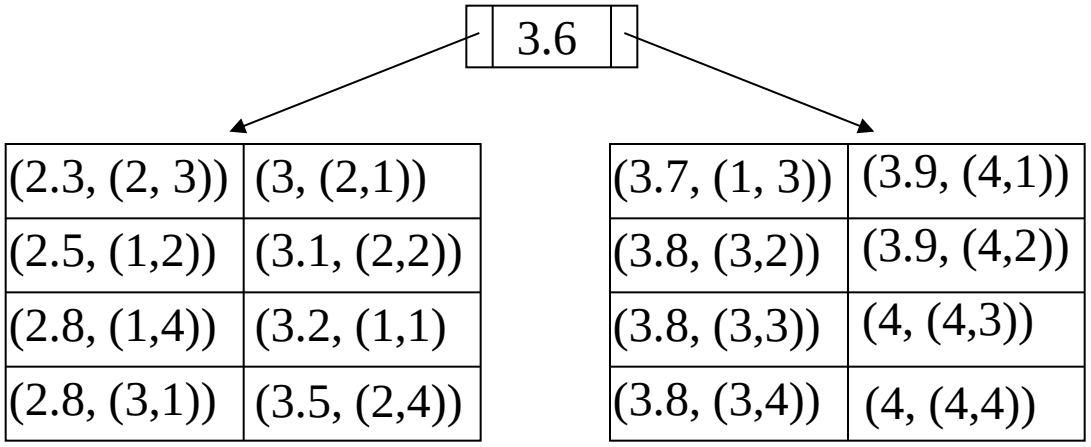
Integritäts-
bed.

alter/
drop table

Index

Index – Illustration (1)

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



Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

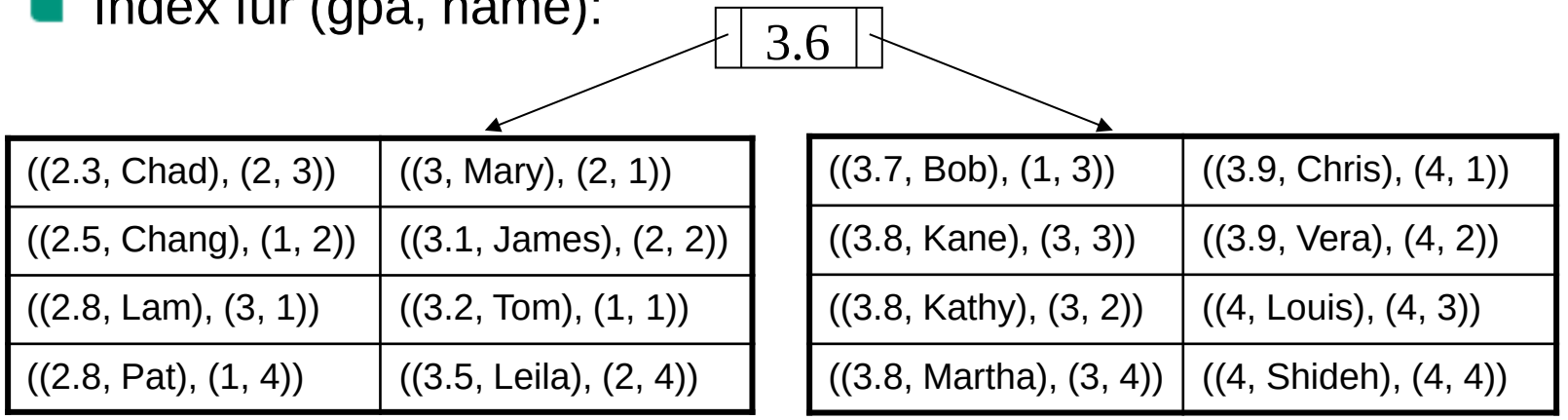
alter/ drop table

Index

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 (2)

- 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

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/
drop table

Index

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

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.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/
drop table

Index

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.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

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.

Bedeutungs-treue

SQL-DDL

create table

Integritäts-bed.

alter/ drop table

Index

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?
- **unique** – auch wieder Beispiel, wo Trennung zwischen logischer und interner Ebene aufgebrochen wird.

Prüfungsfragen, beispielhaft (1)

- Erläutern Sie anhand des Anwendungsbeispiels 'Universitätsbibliothek' <oder irgendetwas anderes>, warum man die Menge der zulässigen Zustände einschränken will.
- Erläutern Sie: Schema-Konsistenz, Datenbasis-Konsistenz.

Prüfungsfragen, beispielhaft (2)

- Was ist ein (Datenbank-)Schema?
- Was ist das Data Dictionary?
- 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.
- Wieso sind Indices Bestandteil des internen
  und nicht des konzeptuellen Schemas?
- Geben Sie Beispiele für Datenbank-Features, anhand derer sich zeigt, dass Datenbank-Systeme physische Datenunabhängigkeit nicht vollständig realisieren.