Grundlagen des relationalen Datenmodells

Medieninformatik Bachelor Modul 9: Datenbanksysteme

Sie erinnern sich: Ihre Aufgaben





- Aufgabe 1 Anfragen & Modellierung"
 - Denken Sie mal darüber nach, welche Anfragen Sie an die AOL Daten stellen möchten. Bitte Sie bitte ein logisches und physisches Schema zur Beantwortung dieser Anfragen.
- Aufgabe 2 "SQL und Anfrageausführung"
 Bitte formulieren Sie für Ihre Analyseideen aus 1.) die SQL Anfragen. Sie verstehen auch Möglichkeiten der Anfrageausführung bzw. Optimierung.
- Aufgabe 3 "Datenintegration" Zur Ausführung der Ausführung fehlen Ihnen noch externe Daten, z.B. aus dem Internet Archive, DMOZ oder Freebase.org. Bitte ergänzen Sie Ihr Schema und die Datenbasis.
- Aufgabe 4 "Analyse, Erkenntnisgewinn und Wert" Stellen Sie in 5 Minuten die wichtigsten Erkenntnisse aus den Daten vor. Bewerten Sie den Erkenntnisgewinn, z.B. gegenüber Ihren Kommilitonen oder der Literatur! Welche Erkenntnisse hätten einen kommerziellen Wert?

Die Themen



- Was sind Datenbanken?
 - Motivation, Historie, Datenunabhängigkeit, Einsatzgebiete
- Datenbankentwurf im ER-Modell & Relationaler Datenbankentwurf
 - Entities, Relationships, Kardinalitäten, Diagramme
 - Relationales Modell, ER -> Relational, Normalformen, Transformationseigenschaften
- Relationale Algebra & SQL
 - Kriterien für Anfragesprachen, Operatoren, Transformationen
 - SQL DDL, SQL DML, SELECT ... FROM ... WHERE ...
- Datenintegration & Transaktionsverwaltung
 - JDBC, Cursor, ETL
 - Mehrbenutzerbetrieb, Serialisierbarkeit, Sperrprotokolle, Fehlerbehandlung, Isolationsebenen in SQL
- Ausblick
 - Map/Reduce, HDFS, Hive ...
 - Wert von Daten



Min, Max vs Chen Notation



Die (min,max)-Notation zählt die Ausprägung von Beziehungen, während die Chen Notationen Entitätstypausprägungen zählen.

(min,max) [Entity 1]	Chen-Notation	(min,max) [Entity 2]	
(0,1)	1:1	(0,1)	
(0,N)	1:N + Totalität	(0,1)	
(0,N)	1:N + Totalität	(1:1	
(0,N)	M:N	(0,N)	
(1,1)	Totalität + 1:1	(0,1)	
(1,N)	Totaliät + 1:N	(0,1)	
(1,1)	Totalität + 1:1 + Totalität	(1,1)	
(1,N)	Totalität+ 1:N + Totalität	(1,1)	
(1,N)	Totalität + M:N	(0,N)	
(1,N)	Totalität + M:N + Totalität	(1,N)	

http://de.wikipedia.org/wiki/Min-Max-Notation







- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Modell-inhärente Integritätsbedingungen
 - Entity Integrität (Primärschlüsselbedinung)
 - Referentielle Integrität (Fremdschlüsselbedingung)
 - Wahrung der referentiellen Integrität bei Datenmanipulation
- Exkurs: Konvertierung von Spezialisierung
- In der Übung: Anlegen von Tabellen in der Datenbank

Kapitel 2 des Lehrbuchs



Die Relation



- Konzeptuell ist eine Datenbank eine Menge von Tabellen.
 - Relation zwischen Werten der Attributdomänen.
 - Tabellen = Relationen

Titel	Jahr	Länge	Тур
Basic Instinct	1992	127	Farbe
Total Recall	1990	113	Farbe
Dead Man	1995	121	s/w

- Die Relation ist das einzige Konstrukt des relationalen Modells
 - Sehr einfach
 - Einfach in einer DB abzubilden (zwei-dimensional)
 - Relationen können nicht nur Entities sondern auch Relationships darstellen.
 - Entspricht oft unserer Vorstellung der Daten
 - Ist das abstrakte Modell hinter SQL.

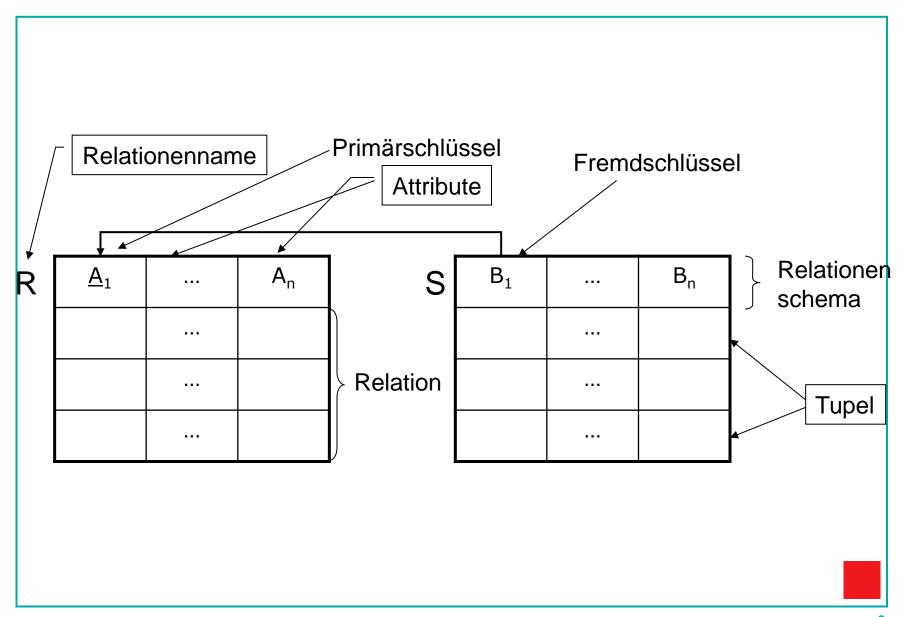
Elemente einer Relation



- Datenbankschema
 - Besteht aus einem oder mehreren Relationenschemata.
- Relationenschema
 - Weitere Einträge in der Tabelle: Die "Relation"
 - Besteht aus keinem oder mehr Tupeln.
- Eine Zeile der Tabelle: Tupel
 - Tupel bilden eine Menge (nicht eine Liste).
- Eine Spaltenüberschrift: Attribut
 - Attribute bilden eine geordnete Menge (Reihenfolge!).
- Ein Eintrag: Attributwert
 - Atomar
 - Stammt aus einer elementaren Domäne (Integer, String, ...)

Elemente einer Relation





Formal



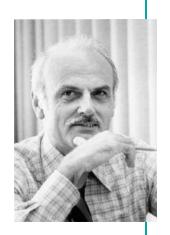
- Domänen D1, ..., Dn
- Relation $R \subseteq D1 \times ... \times Dn$
- Beispiel
 - Relationenschema: Filme(Titel, Jahr, Länge, Typ)
 - Tupel: (Star Wars, 1977, 124, farbig)

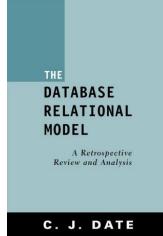


Edgar F. Codd



- http://en.wikipedia.org/wiki/Edgar_F._Codd
- Promotion an der University of Michigan Ann Arbor
- Entwicklung des Relationalen Modells bei IBM (Almaden)
- "A Relational Model of Data for Large Shared Data Banks" (1970)
- Artikelserie
- Literaturhinweis:
 - The Database Relational Model: A Retrospective Review and Analysis:
 - A Historical Account and Assessment of E. F. Codd's Contribution to the Field of Database Technology
 - Chris J. Date
 - ISBN: 0-201-61294-1 (9.99 EUR)





Codd – Grundregeln für Relationen



- Repräsentation als zweidimensionale Tabelle: Atomarität der Attributwerte
- Eindeutigkeit der Tupel (Datensätze, Zeilen) muss gegeben sein.
- Primärschlüssel (primary key, oder kurz PK) legt die minimale, als eindeutig zu betrachtende Attributkombination eines Tupels fest Varianten: einfacher, zusammengesetzter oder künstlicher PK
- Variabilität der Tupelposition
 (i.e. ob ein Datensatz einer Relation bspw. in Zeile 1 oder Zeile 5 abgebildet wird, sei gleich)
- Eindeutigkeit des Attributnamens ("Spaltenname")
- Variabilität der Attributposition
 (i.e. ob das Attribut einer Relation bspw. in Spalte 1 oder 2 abgebildet wird, sei gleich)
- Information für den Benutzer ausschließlich durch Werte ausgedrückt (i.e. es gibt bspw. keine Verweise/Zeiger/Referenzen sondern nur Wertevergleiche)
- Darstellung relationenübergreifender Information durch *Fremdschlüssel* (foreign key, oder kurz *FK*)
- Anwendung der *Normalformenlehre* für den Aufbau von (möglichst redundanzarmen) Relationenschemata







- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Modell-inhärente Integritätsbedingungen
 - Entity Integrität (Primärschlüsselbedinung)
 - Referentielle Integrität (Fremdschlüsselbedingung)
 - Wahrung der referentiellen Integrität bei Datenmanipulation
- Exkurs: Konvertierung von Spezialisierung
- In der Übung: Anlegen von Tabellen in der Datenbank

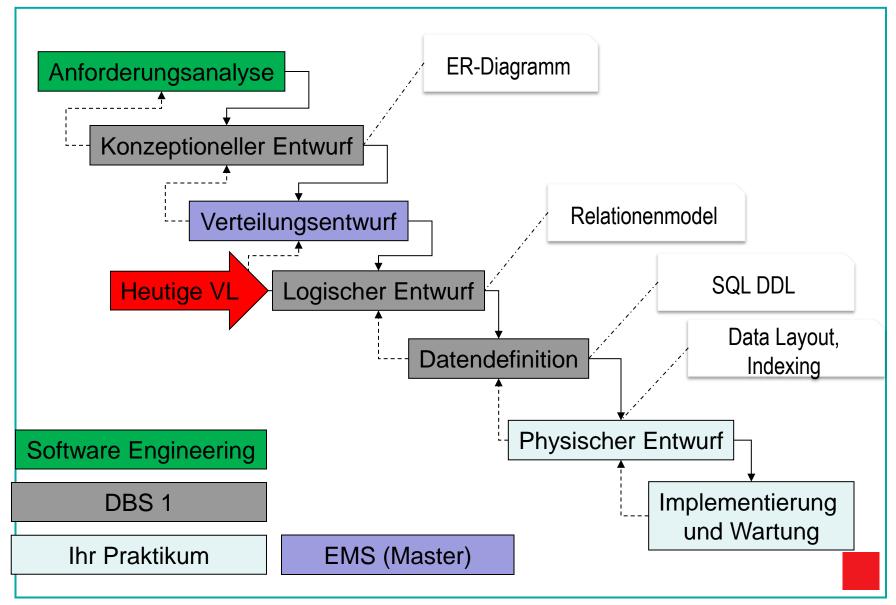
Kapitel 2 des Lehrbuchs



P

Phasenmodell für den Datenbankentwurf







Ziele der Abbildung:

Informationserhaltung

Möglichst komplette Information aus dem konzeptuellen Schema nutzbar machen.

Minimale Redundanz (mindestens 3NF)

Das konzeptuelle Schema sollte keine Redundanzen besitzen. Das logische Schema sollte redundanzfrei sein, mit Ausnahme der Schlüsselredundanzen. Es sollte mindestens die 3NF umsetzen.

Minimale Anzahl von Relationen

Es sollten keine unnötigen Relationen erzeugt werden, da dies das Abfrageverhalten negativ beeinflusst. Allerdings sollten i.d.R. keine Relationen "eingespart" werden, wenn dies die 3NF unterläuft.

Natürlichkeit, Verständlichkeit der Abbildung

Probleme der Abbildung:

Verlust an Semantik

Das relationale Datenmodell gehört nicht zu den semantischen Datenmodellen, wie ERM und UML. Es verfügt über weniger Konzepte. Ein semantisch "reicheres" wird auf ein "ärmeres" Modell abgebildet.

Abbildung nicht eindeutig reversibel

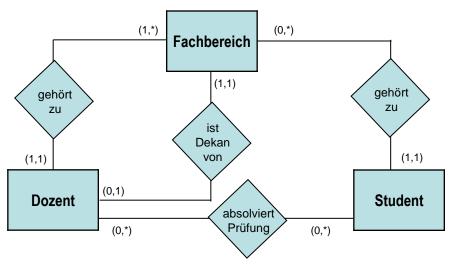
Verschiedene Konzepte des Ausgangsmodells werden auf ein Konzept des Zielmodells abgebildet.

Zeitpunkt der Abbildung (inkrementelles Arbeiten?)









Ausgangsschema:

- ERM-Beispiel: UoD Hochschule
- Modellelemente:
 - Attribut
 - Entitätentyp
 - Beziehungstyp
 - Generalisierung

Zielschema:

- RDM
- Modellelemente:
 - Attribut
 - Relation
 - Integritätsbedingung
 - Primärschlüssel
 - Fremdschlüssel



Grundalgorithmus



- Wandle jeden Entitytypen in eine Relation mit den gleichen Attributen um.
- Wandle jeden Relationshiptypen in eine Relation um, deren Attribute die zugehörigen Attribute und die Schlüsselattribute der beteiligten Entitytypen sind.
- Verfeinere den Entwurf
 - Zusammenlegung von Relationen
 - Normalisierung
- Ausnahmen
 - IST Relationships
 - Schwache Entitytypen



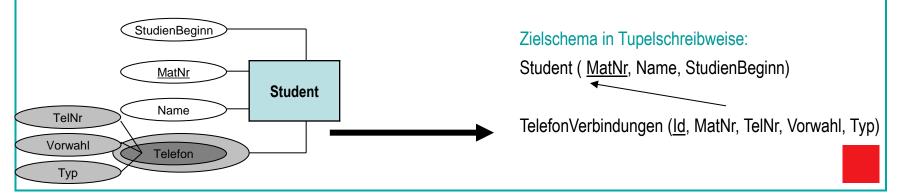


Transformation von Attributen:

 Elementare Attribute und (Standard-)Wertebereiche des ERM werden auf Attribute und Wertebereiche des RDM abgebildet.



 Liegen mehrwertige oder strukturierte Attribute des ERM vor, so wird für deren Abbildung eine Relation erzeugt. Sie enthält das mehrwertige Attribut und einen Fremdschlüssel.

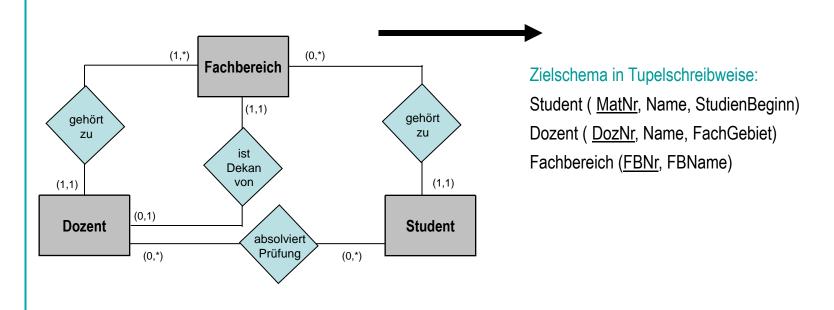






Transformation von Entitätentypen:

■ Ein Entitätentyp und seine elementaren Attribute werden auf eine Relation und deren Attribute abgebildet.

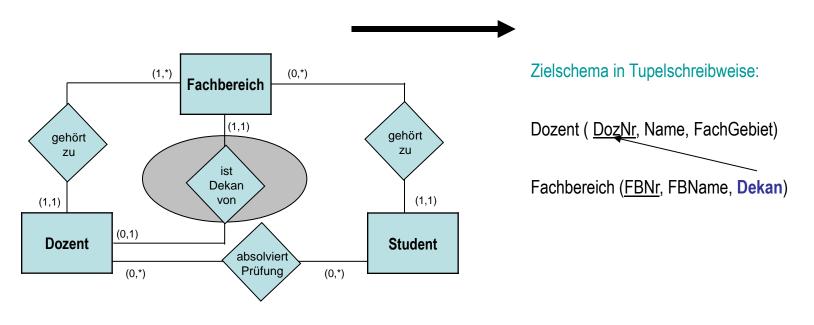






Transformation von Beziehungstypen (1):

■ Ein Beziehungstyp der Kardinalität 1:1 wird über einen Fremdschlüssel abgebildet: Der Primärschlüssel eines (beliebigen) Entitätentyps, der an der Beziehung teilnimmt, wird als Fremdschlüssel in der Relation des anderen Entitätentyps verwendet.

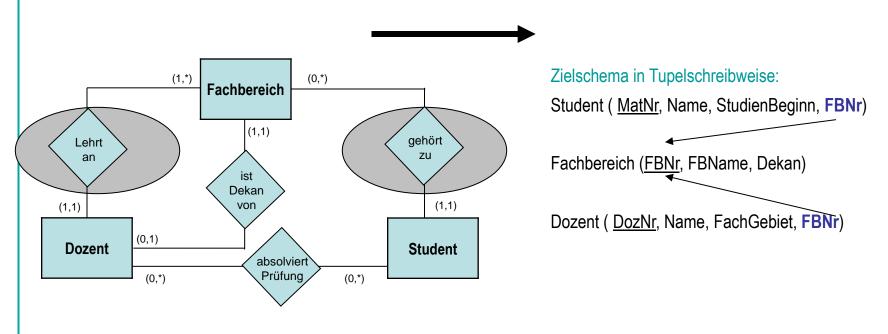






Transformation von Beziehungstypen (2):

■ Ein Beziehungstyp der Kardinalität 1:N wird über einen Fremdschlüssel abgebildet: Der Primärschlüssel des Entitätentyps der ,1er-Seite', der an der Beziehung teilnimmt, wird als Fremdschlüssel in der Relation des Entitätentyps der ,N-Seite' verwendet.

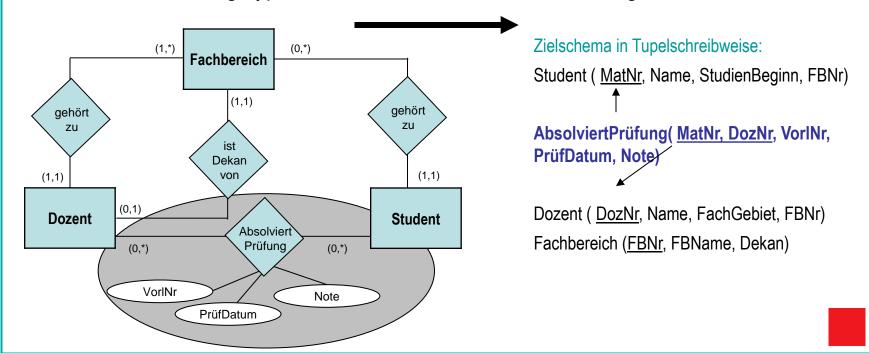






Transformation von Beziehungstypen (3):

■ Ein Beziehungstyp der Kardinalität M:N wird auf eine Relation abgebildet: Die Primärschlüssel der Entitätentypen, die an der Beziehung teilnehmen, werden als Fremdschlüssel in diese Relation übergeben und bilden i.d.R. deren zusammengesetzten Primärschlüssel. Attribute des Beziehungstyps werden ebenso dieser Relation zugeordnet.

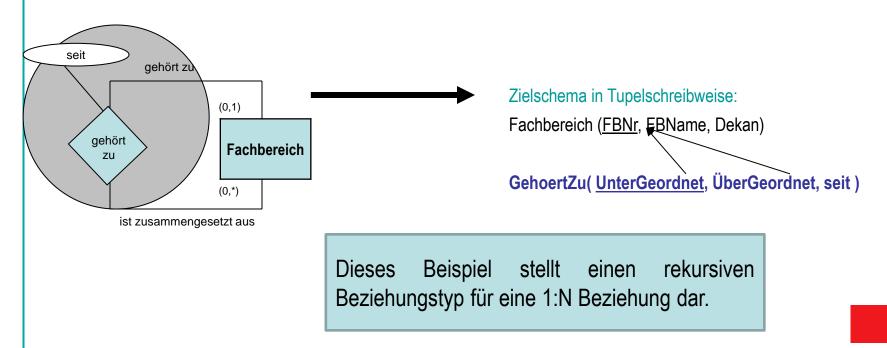






Transformation von Beziehungstypen (4):

Ein rekursiver Beziehungstyp wird auf eine Relation abgebildet: Der Primärschlüssel des Entitätentyps wird als Fremdschlüssel zweimal in diese Relation übergeben, wobei durch Umbenennung eindeutige Namen zu gewährleisten sind. Attribute des Beziehungstyps werden ebenso dieser Relation zugeordnet.

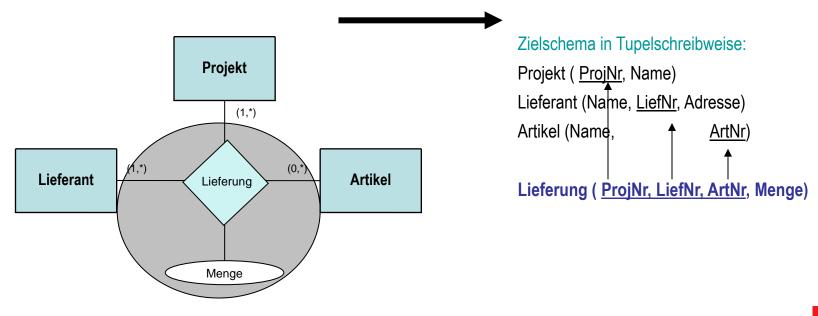






Transformation von Beziehungstypen (5):

Ein n-ärer Beziehungstyp wird auf eine Relation abgebildet: Die Primärschlüssel der Entitätentypen, die an der Beziehung teilnehmen, werden als Fremdschlüssel in diese Relation übergeben und bilden i.d.R. deren zusammengesetzten Primärschlüssel. Attribute des Beziehungstyps werden ebenso dieser Relation zugeordnet.





Create your Own Exam: Relationaler Entwurf



- Bitte erstellen Sie eine Multiple Choice Aufgabe zum Thema Relationaler Entwurf
 - Formulieren Sie eine Frage und 3 Antworten (A, B, C)
 - Davon sollte mindestens eine Antwort richtig und mindestens eine Antwort falsch sein
- Geben Sie die Aufgabe an Ihren rechten Nachbarn. Diskutieren Sie gemeinsam und markieren Sie die richtigen Lösungen
- Geben Sie am Ende der Vorlesung Ihre Aufgabe bei mir ab

5 min



"Just a darn minute! — Yesterday you said that X equals two!







- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Modell-inhärente Integritätsbedingungen
 - Entity Integrität (Primärschlüsselbedinung)
 - Referentielle Integrität (Fremdschlüsselbedingung)
 - Wahrung der referentiellen Integrität bei Datenmanipulation
- Exkurs: Konvertierung von Spezialisierung
- In der Übung: Anlegen von Tabellen in der Datenbank

Kapitel 2 des Lehrbuchs





Ausgangsdefinitionen (1):

- Was versteht man unter einem Primärschlüssel (PK)?
 - Eine identifizierende Attributkombination für ein Relationenschema
 - Jedes Tupel einer Relation wird einzig und allein über diese (sinnvolle und minimale) Attributkombination eindeutig identifiziert
 - Alle nicht in dieser festgelegten Attributkombination eingeschlossenen Attribute eines Relationenschemas sind für die eindeutige Identifikation eines Tupels unerheblich.

Dozent					
<u>DozNr</u>	Name	Vorname	Aufenthaltsort	Gebäude	
123	Blankenburg	Marco	D139	D	demselben Primärschlüssel-Wert (123)
123	Blankenburg	Marco	B131	В	existieren und daher nicht mehr eindeutig sind.

 Bei einem künstlichen PK werden Attribute in die Relation aufgenommen, die mit der abstrakten Beschreibung eines Anwendungsobjektes nichts zu tun haben, sondern "nur" Verwaltungsinformationen sind.

Anmerkung: Manchmal existieren solche ein Objekt eindeutig identifizierenden Verwaltungsinformationen aber auch schon in der Realwelt und können als einfacher PK benutzt werden (z.B. Matrikelnummer eines Studenten, ISBN-Nummer eines Buches, Kennzeichen eines Landes)



Ausgangsdefinitionen (2):

- Was versteht man unter einem Fremdschlüssel (FK)?
 - Attribut, das in Bezug auf den Primärschlüssel einer anderen Relation (oder auch derselben) definiert ist. Attribut darf <u>nur</u> die Werte der aktuell zur Verfügung stehenden Primärschlüsselmenge annehmen.
 - Beispiel:
 - Dozent(<u>DozNr</u>, Name, Vorname, Aufenthaltsort, <u>Gebäude</u>)
 - Gebäude(<u>Kennzeichen</u>, Name, Postadresse)

Г	١	^	7	۵	n	ŧ

<u>DozNr</u>	Name	Vorname	Aufenthaltsort	Gebäude
123	Blankenburg	Marco	B131	В
124	Sauer	Petra	D139	D

Erst durch Hinzufügen des Tupels mit dem Kennzeichen **D** in der Relation 'Gebäude' erhält dieser Buchstabe eine Semantik und darf in der Relation 'Dozent' als **Fremdschlüssel** verwendet werden.

Gebäude

Kennz.	Name	Post-Adresse
А	Haus Beuth	Lütticher Strasse 38, 13353 Berlin
В	Haus Gauß	Limburger Straße 20, 13353 Berlin
С	Haus Grashof	Luxemburger Straße 10, 13353 Berlin
D	Haus Bauwesen	Luxemburger Straße 9, 13353 Berlin





Primärschlüsselbedingung (Entity-Integrität):

- Forderung: Die Werte der Primärschlüsselattribute sind eindeutig und nicht "NULL" (also besitzen einen Wert)
- Die SQL-Syntax (DDL) zum Formulieren der Bedingung beim Anlegen eines Datenbankobjektes (also einer Tabelle) lautet allgemein wie folgt:

```
CREATE TABLE <tabellenname>
(<spalte1> <datentyp> [[CONSTRAINT <name der bedingung>] PRIMARY KEY],
...,
[[CONSTRAINT <name der bedingung>] PRIMARY KEY (<spalte1>,...)] );
```

Beispiel (als separate Integritätsbedingung):

```
CREATE TABLE Dozent
( DozNr CHAR(3),
...,
PRIMARY KEY (DozNr) );
```

Beispiel (als Erweiterung der Spaltendefinition):

```
CREATE TABLE Dozent
( DozNr CHAR(3) PRIMARY KEY,
...);
```





Fremdschlüsselbedingung (Referentielle Integrität):

- Forderung: Zu jedem Wert eines Fremdschlüsselattributs einer Relation R2 muss ein Wert des Primärschlüssels in Relation R1 vorhanden sein
- Die SQL-Syntax (DDL) zum Formulieren der Bedingung beim Anlegen eines Datenbankobjektes (also einer Tabelle) lautet allgemein wie folgt:

```
CREATE TABLE <tabellenname>
(<spalte1> <datentyp> [[CONSTRAINT <name der bedingung>] REFERENCES <tabellenname> [<spalte>]],
...,
[[CONSTRAINT <name der bedingung>] FOREIGN KEY (<spalte1>,...) REFERENCES <tabellenname> [<spalte>]]);
```

Beispiel (als separate Integritätsbedingung):

```
CREATE TABLE Prüfung
( DozentNr CHAR(3), StudentNr CHAR(6), ...,
PRIMARY KEY (DozentNr, StudentNr),
FOREIGN KEY (DozentNr) REFERENCES Dozent(DozNr),
FOREIGN KEY (StudentNr) REFERENCES Student(MatrNr) );
```

Beispiel (als Erweiterung der Spaltendefinition):

```
CREATE TABLE Prüfung
( DozentNr CHAR(3) REFERENCES Dozent(DozNr),
StudentNr CHAR(6) REFERENCES Student(MatrNr),
...,
PRIMARY KEY (DozentNr, StudentNr));
```





Regeln in Bezug auf Fremdschlüssel:

- FK und PK sind auf **gleichem** Wertebereich definiert (d.h. bspw: lst PK vom Typ CHAR(3) so muss der FK auch vom Typ CHAR(3) sein)
- Wertebereich für FK ist explizit anzugeben
 (d.h. leitet sich nicht implizit aus dem Datentyp und Wertebereich des referenzierten Pimärschlüssels ab)
- FK und PK gestatten die (logische) Verknüpfung von Relationen
- FK kann auch Teil des PK der Relation sein (siehe bspw. Relation "Prüfung" mit DozentNr und StudentNr)
- FK können Nichtschlüsselattribute (NSA) sein. In diesem Fall dürfen/können FK "NULL"-Werte aufweisen
- FK kann zusammengesetzt sein (wenn referenzierter PK zusammengesetzt ist)
- Relation kann mehrere FK besitzen, die die gleiche oder verschiedene Relationen referenzieren. Beispiel:

```
CREATE TABLE Mitarbeiter

( PersonalNr INTEGER PRIMARY KEY,
 Name CHAR(20),
 Vorname CHAR(20),
 Vorgesetzter INTEGER REFERENCES Mitarbeiter(PersonalNr),
 Assistent INTEGER REFERENCES Mitarbeiter(PersonalNr));
```





- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Modell-inhärente Integritätsbedingungen
 - Entity Integrität (Primärschlüsselbedinung)
 - Referentielle Integrität (Fremdschlüsselbedingung)
 - Wahrung der referentiellen Integrität bei Datenmanipulation
- Konvertierung von Spezialisierung
- Normalformenlehre

Kapitel 2 des Lehrbuchs





Wahrung der Referentiellen Integrität bei Datenmanipulation (1):

Kritische Datenmanipulationsoperationen für PK-FK Beziehungen:

Verschiedene Aktionen möglich

- Löschen (DELETE) eines Tupels in der referenzierten Relation
 Beispiel: Ein Dozent oder Student soll gelöscht werden, zu dem noch Prüfungsdaten vorhanden sind.
- Aktualisieren (UPDATE) eines PK Attributs in der referenzierten Relation Beispiel: DozNr des Dozenten Blankenburg in Relation "Dozent" soll von 123 zu 500 geändert werden und es gibt bereits Prüfungsdaten zu diesem Dozenten.

Verbot

- Aktualisieren (UPDATE) des FK Attributs in der referenzierenden Relation Beispiel: DozNr des Dozenten Blankenburg in Relation "Prüfung" soll von 123 zu 500 geändert werden.
- Einfügen (INSERT) eines Tupels in die referenzierende Relation
 Beispiel: Es soll ein Prüfungsdatum mit einem Prüfer (Dozenten) eingetragen werden, der noch nicht in der Relation "Dozent" existiert und dessen PK daher nicht referenziert werden kann.





Wahrung der Referentiellen Integrität bei Datenmanipulation (2):

- Der SQL-92 Standard (99, 2003, 2006) spezifiziert referentielle Aktionen, die ein DBMS für jeden FK bei einer DELETE oder UPDATE Operation ausüben kann:
 - ON DELETE (NO ACTION | RESTRICT | CASCADE | SET DEFAULT | SET NULL)
 - ON UPDATE (NO ACTION | RESTRICT | CASCADE | SET DEFAULT | SET NULL)
- Die referentielle Aktion kann bei der Spezifikation des Fremdschlüssels angegeben werden (muss jedoch durchs DBMS unterstützt werden):

```
CREATE TABLE <tabellenname>
  (<spalte1> <datentyp> [[CONSTRAINT <name der bedingung>] REFERENCES <tabellenname>
[<spalte>]
  [(ON UPDATE | ON DELETE) <name der aktion>], ...);
```



Wahrung der Referentiellen Integrität bei Datenmanipulation (3):

- Wird keine Aktion spezifiziert, so ist "NO ACTION" das Standardverhalten
- NO ACTION / RESTRICT: Operation ist beschränkt und wird nur ausgeführt, sofern (noch) keine referenzierenden Datensätze (i.e. FK-Werte) vorhanden sind. Ansonsten wird die Operation durch das DBMS zurückgewiesen.
 - No action: optimistisches Verfahren, ggf. Rollback erforderlich
 - Restrict: pessimistisches Verfahren
- CASCADE: Kaskadierende (i.e. weiterreichende) Operation
 - Beispiel: Wird Dozent gelöscht, werden auch alle Prüfungsdaten gelöscht
 - Beispiel: Wird die DozNr eines Dozenten von '123' auf '500' geändert, so werden auch automatisch alle Prüfungen, die Dozent '123' referenzierten auf '500' gesetzt.
- SET DEFAULT: Fremdschlüsselattribut wird auf den eingestellten Standardwert gesetzt
 - Beispiel: Wird ein Dozent gelöscht, werden alle Prüfungen für deren Betreuung er vorgesehen war (also, die ihn referenzierten) auf Dozent '123' eingestellt, wenn dies der Default-Wert ist
- SET NULL: Fremdschlüsselattribut wird auf NULL gesetzt (jedoch nur, wenn FK nicht gleichzeitig Teil eines PK ist, so wie bei der Relation 'Prüfung')





Beispiele zur Wahrung der Referentiellen Integrität (1):

1.) Was passiert mit den Prüfungsdaten, wenn ein Student exmatrikuliert wird?

```
CREATE TABLE Prüfung

(
    DozentNr CHAR(3),
    StudentNr CHAR(6),
    ...,
    PRIMARY KEY (DozentNr, StudentNr),
    FOREIGN KEY (DozentNr) REFERENCES Dozent(DozNr),
    FOREIGN KEY (StudentNr) REFERENCES Student(MatrNr) ON DELETE CASCADE
);
```

2.) Welche Veränderung zieht die Änderung der Matrikelnummer eines Studenten nach sich?

```
CREATE TABLE Prüfung
(
    DozentNr CHAR(3),
    StudentNr CHAR(6),
    ...,
    PRIMARY KEY (DozentNr, StudentNr),
    FOREIGN KEY (DozentNr) REFERENCES Dozent(DozNr),
    FOREIGN KEY (StudentNr) REFERENCES Student(MatrNr) ON UPDATE CASCADE
);
```





Beispiele zur Wahrung der Referentiellen Integrität (2):

3.) Was passiert mit den Prüfungsdaten, wenn ein Professor emeritiert wird?

```
CREATE TABLE Prüfung

(
DozentNr CHAR(3),
StudentNr CHAR(6),
...,
PRIMARY KEY (DozentNr, StudentNr),
FOREIGN KEY (DozentNr) REFERENCES Dozent(DozNr) ON DELETE SET NULL,
FOREIGN KEY (StudentNr) REFERENCES Student(MatrNr)
);
```

4.) Was passiert mit den Prüfungsdaten, wenn ein Professor emeritiert wird?

```
CREATE TABLE Prüfung

(
    DozentNr CHAR(3) DEFAULT '123',
    StudentNr CHAR(6),
    ...,
    PRIMARY KEY (DozentNr, StudentNr),
    FOREIGN KEY (DozentNr) REFERENCES Dozent(DozNr) ON DELETE SET DEFAULT,
    FOREIGN KEY (StudentNr) REFERENCES Student(MatrNr)
);
```





- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Modell-inhärente Integritätsbedingungen
 - Entity Integrität (Primärschlüsselbedinung)
 - Referentielle Integrität (Fremdschlüsselbedingung)
 - Wahrung der referentiellen Integrität bei Datenmanipulation
- Exkurs: Konvertierung von Spezialisierung
- In der Übung: Anlegen von Tabellen in der Datenbank

Kapitel 2 des Lehrbuchs



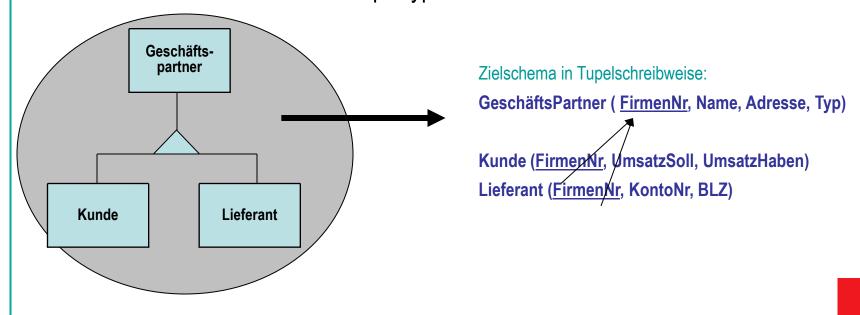


Schematransformation



Transformation von Generalisierungshierarchien (1):

Eine Generalisierungshierarchie wird derart abgebildet, dass für den Supertyp und alle Subtypen eigene Relationen angelegt werden. Die Relation des Supertyps enthält generalisierbare Attribute, einen Primärschlüssel und evtl. ein Typkennzeichen. Die Relationen der Subtypen beinhalten die speziellen Attribute und erben den Primärschlüssel des Supertyps.



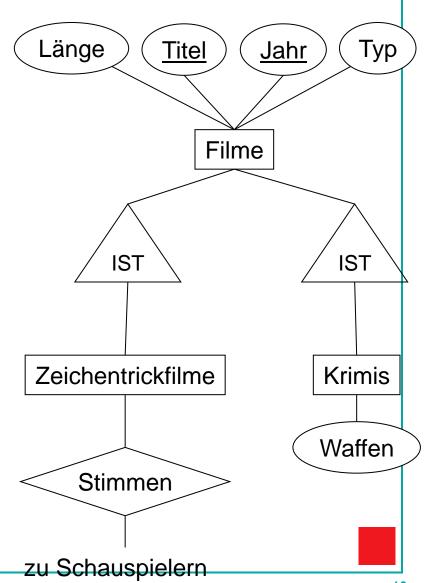


Wdh. Spezialisierung



Annahmen

- Es gibt ein Wurzel-Entitytyp der IST-Hierarchie.
- Der Wurzel-Entitytyp hat einen Schlüssel, der alle Entities der gesamten Hierarchie identifiziert.
- Ein Entity kann aus mehreren Komponenten der Hierarchie bestehen.





Drei Strategien



Im ER-Stil

■ Für jeden Entitytypen E der Hierarchie erzeuge eine Relation mit den Schlüsselattributen des Wurzel-Entitytypen und den Attributen von E.

Objekt-orientierter Stil

- Ein Entity gehört zu genau einer Klasse.
- Für jeden möglichen Teilbaum der Hierarchie, der auch die Wurzel enthält, erzeuge eine Relation mit allen Attributen der beteiligten Entitytypen.

Mit Null-Werten

 Erzeuge eine einzige Relation für die gesamte Hierarchie. Ein Entity wird durch ein Tupel repräsentiert mit Null-Werten für Attribute, die der Entity nicht besitzt.

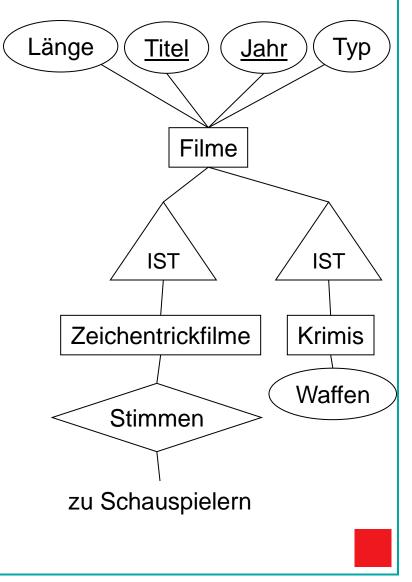






Anmerkungen

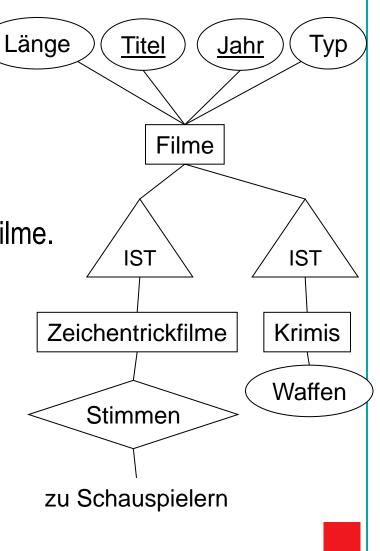
- Die IST-Relationship erhält keine Relation.
- Geerbte Schlüsselattribute werden für weitere Beziehungen verwendet.
- Es gibt vier verschiedene Filmsorten.
- Jeder Film hat ein Tupel in der Relation Filme.
- Ein konkreter Film (z.B. Roger Rabbit) kann Tupel in allen drei Relationen haben.



ER-Stil – Feinheiten



- Filme(Titel, Jahr, Länge, Typ)
- Krimis(Titel, Jahr, Waffen)
- Zeichentrickfilme(Titel, Jahr)
- Stimmen(Titel, Jahr, Name)
- Zeichentrickfilm ist Teilmenge von Filme.
 - Kann man es weglassen?
- Stumme Zeichentrickfilme!

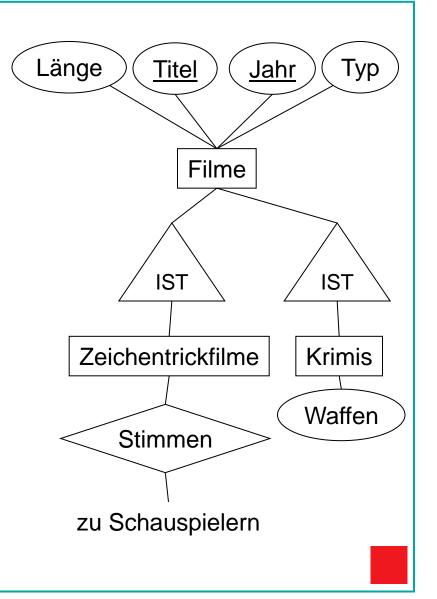




Objekt-orientierter Stil



- Erzeuge Relation für jeden Teilbaum.
- Diese Relation repräsentiert die Entities, die genau diese Komponenten der Hierarchie besitzen.
 - Objekte gehören zu genau einer Klasse.
- Vier Teilbäume
 - Nur Filme
 - Filme und Zeichentrickfilme
 - Filme und Krimis
 - Filme und Zeichentrickfilme und Krimis

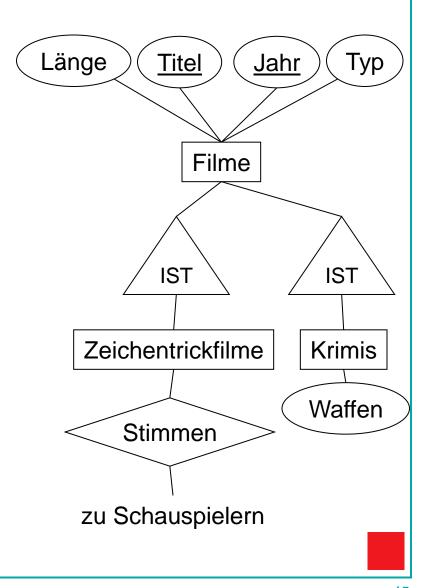




Objekt-orientierter Stil – Feinheiten



- Filme(Titel, Jahr, Länge, Typ)
- FilmeZ(Titel, Jahr, Länge, Typ)
- FilmeK(Titel, Jahr, Länge, Typ, Waffen)
- FilmeZK(Titel, Jahr, Länge, Typ, Waffen)
- Kann man Filme und FilmeZ kombinieren?
- Kann man FilmeK und FilmeZK kombinieren?
- Wie viele Relationen für Stimmen benötigt man?
- Stimmen(Titel, Jahr, Name)



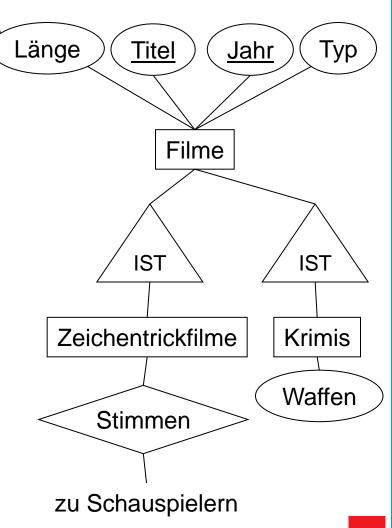
Mit Nullwerten



Eine einzige Relation mit allen Attributen,

Filme(Titel, Jahr, Länge, Typ, Waffen)

 Nicht-Krimis haben NULL-Wert als Attributwert für Waffen.





Schematransformation



Transformation von Generalisierungshierarchien - Vergleich:

- zwei Relationen (jeweils Supertyp und ein Subtyp)
 (Horizontale Partitionierung nach Saake/Sattler/Heuer)
- eine Relation
 (Typisierte Partitionierung (unbedingt mit Typkennzeichen!))

	Anfragen auf Supertyp	Anfragen auf Subtypen
Standardvariante	Effizient	Ineffizienter Zugriff auf vollständige Objekte (Join von Super- und Subtyp nötig)
Horizontale Partitionierung	Ineffizient	Effizient
Typisierte Partitionierung	Effizient, viele Nullwerte!!	Effizient, unflexibel



Sie erinnern sich: Ihre Aufgaben





Aufgabe 1 Anfragen & Modellierung"

Denken Sie mal darüber nach, welche Anfragen Sie an die AOL Daten stellen möchten. Bitte Sie bitte ein logisches und physisches Schema zur Beantwortung dieser Anfragen.

- Aufgabe 2 "SQL und Anfrageausführung"
 Bitte formulieren Sie für Ihre Analyseideen aus 1.) die SQL Anfragen. Sie verstehen auch Möglichkeiten der Anfrageausführung bzw. Optimierung.
- Aufgabe 3 "Datenintegration" Zur Ausführung der Ausführung fehlen Ihnen noch externe Daten, z.B. aus dem Internet Archive, DMOZ oder Freebase.org. Bitte ergänzen Sie Ihr Schema und die Datenbasis.
- Aufgabe 4 "Analyse, Erkenntnisgewinn und Wert" Stellen Sie in 5 Minuten die wichtigsten Erkenntnisse aus den Daten vor. Bewerten Sie den Erkenntnisgewinn, z.B. gegenüber Ihren Kommilitonen oder der Literatur! Welche Erkenntnisse hätten einen kommerziellen Wert?

Die Themen



- Was sind Datenbanken?
 - Motivation, Historie, Datenunabhängigkeit, Einsatzgebiete
- Datenbankentwurf im ER-Modell & Relationaler Datenbankentwurf
 - Entities, Relationships, Kardinalitäten, Diagramme
 - Relationales Modell, ER -> Relational, Normalformen, Transformationseigenschaften
- Relationale Algebra & SQL
 - Kriterien für Anfragesprachen, Operatoren, Transformationen
 - SQL DDL, SQL DML, SELECT ... FROM ... WHERE ...
- Datenintegration & Transaktionsverwaltung
 - JDBC, Cursor, ETL
 - Mehrbenutzerbetrieb, Serialisierbarkeit, Sperrprotokolle, Fehlerbehandlung, Isolationsebenen in SQL
- Ausblick
 - Map/Reduce, HDFS, Hive ...
 - Wert von Daten



Zusammenfassung und Ausblick



- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata

In der nächsten Veranstaltung:

- Integritätsbedingungen
- Normalformen und funktionale Abhängigkeiten



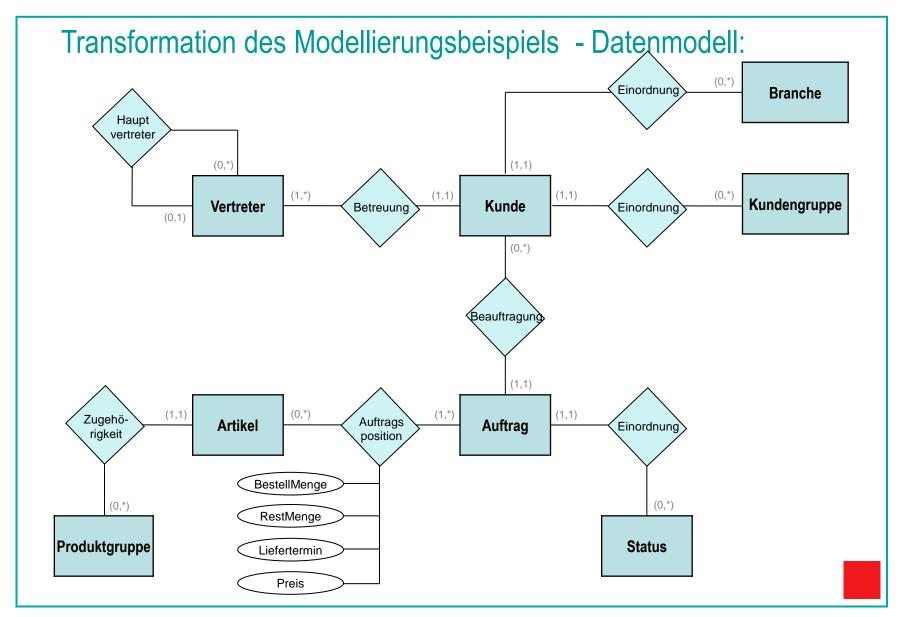


ÜBUNBGSAUFGABE



Schematransformation





Aufgabe



- Bilden sie das ER-Modell auf Tabellen ab.
- Legen sie diese Tabellen in der Datenbank an. Definieren Sie sinnvolle Datentypen der Attribute.
- Spezifizieren sie Primär- und Fremdschlüssel.

Schematransformation



Transformation des Modellierungsbeispiels - Zielschema:





- Das Relationale Modell
- Von ER-Diagrammen zu Relationenschemata
- Modell-inhärente Integritätsbedingungen
 - Entity Integrität (Primärschlüsselbedinung)
 - Referentielle Integrität (Fremdschlüsselbedingung)
 - Wahrung der referentiellen Integrität bei Datenmanipulation
 - Verzögerte Bedinungsprüfung bei referentiellen Zyklen
- Exkurs: Konvertierung von Spezialisierung
- In der Übung: Anlegen von Tabellen in der Datenbank

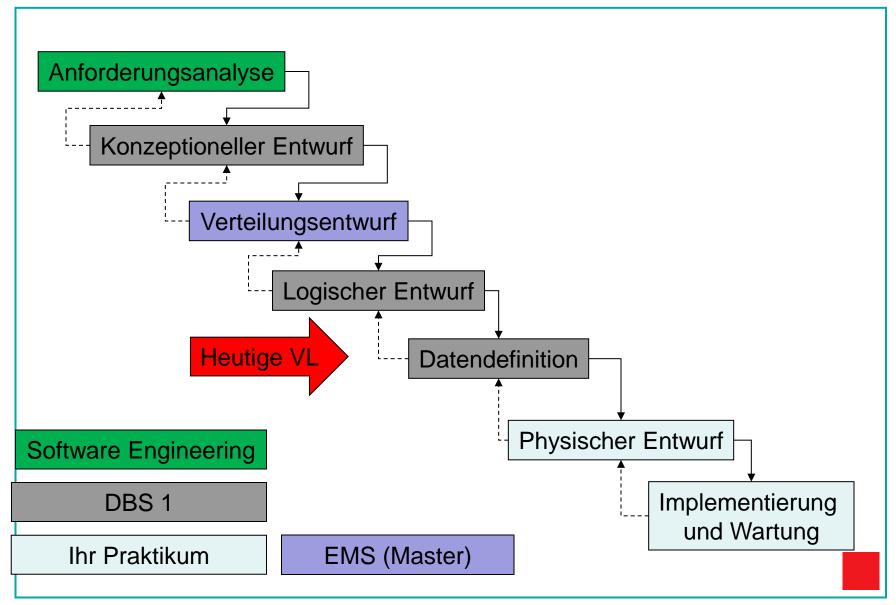
Kapitel 2 des Lehrbuchs





Phasenmodell für den Datenbankentwurf







SQL Statements zum Anlegen der Tabellen



Syntax zum Anlegen der Tabellen

Beispiele

```
CREATE TABLE Person
(id INTEGER NOT NULL,
name VARCHAR(20),
birthdate DATE,
PRIMARY KEY(id));
```



SQL Syntax für Foreign Keys



Fremdschlüssel werden über die Reference-Klausel angegeben

```
CREATE TABLE Employee

(id INTEGER NOT NULL,

name VARCHAR(20),

department INTEGER,

PRIMARY KEY(id),

FOREIGN KEY(department) REFERENCES Department(id));
```

 Eine Spalte (oder mehrere) werden als Referenz auf andere existierende Spalten gesetzt.

```
CREATE TABLE ExampleTable
( Key1 INTEGER,
   Key2 INTEGER,
   ...
PRIMARY KEY(Key1),
FOREIGN KEY(Key1, Key2) REFERENCES TabX(Key1, key2);
```



SQL Syntax für Foreign Keys



- Was passiert, wenn das Tuple, auf das ein Foreign Key zeigt gelöscht wird?
- Per Default: Es wird verhindert!
- Anderes Verhalten kann durch zusätzlichen Optionen spezifiziert werden,
 z.B. dass darauf referenzierende Tuple ebenfalls gelöscht werden.
 - → Dazu die Option "ON DELETE CASCADE" benutzen

```
CREATE TABLE example1
  (id INTEGER NOT NULL,
   name VARCHAR(20),
   department INTEGER,
   PRIMARY KEY(id),
   FOREIGN KEY(department) references DEPARTMENT(id)
   ON DELETE CASCADE);
```

SQL Statements zum Verändern von Tabellen



- SQL Statements zum Verändern von Tabellen fangen immer mit "ALTER TABLE" an
- Mögliche Veränderungen:
 - Column hinzufügen: "ALTER TABLE ADD <column> <type> [BEFORE <other-column>]"
 - Column löschen: "ALTER TABLE DROP <column>"
 - Datentyp ändern: "ALTER TABLE ALTER <column> SET DATA TYPE <new-type>"
 - Foreign Key hinzufügen "ALTER TABLE ADD FOREIGN KEY (<columns>) REFERENCES (<columns>)"



DDL vs. DML



- Bei SQL unterscheidet man zwischen zwei Klassen von Statements:
- DDL = Data Definition Language

Statements, die Objekte in der Datenbank (Tabellen, Indexe, etc.) anlegen oder modifizieren. Verändern nicht die eigentlichen Nutzdaten sondern die Meta-Daten.

DML = Data Manipulation Language

Alles, was mit gegebenen Tabellen arbeitet und Tupel einfügt, ändert, löscht, liest...

(SELECT, INSERT, UPDATE, DELETE, ...)
Arbeiten mit den Nutzdaten.

DCL (Data Control Language); TCL (Transaction Control)





Datentyp	Beschreibung
INTEGER / INT	Ganzzahl mit 32 Bit Wertebereich (vgl. int)
SMALLINTEGER	Kleine Ganzzahl mit 16 Bit Wertebereich (vgl. short)
BIGINT	Große Ganzzahl mit 64 Bit Wertebereich
REAL	Fließkommazahl mit 4 Byte
DOUBLE	Fließkommazahl mit 8 Byte
FLOAT(121)	Fließkomma Zahl, 4 Bytes
FLOAT(2253)	Fließkomma Zahl, 8 Bytes
DECIMAL(m, n)	Dezimalbruch der Länge der Länge m (m/2)+1 bytes
BOOLEAN	TRUE FALSE (vorher per CHAR(1))





Datentyp	Beschreibung
DATE	4 Bytes Datum, Bsp.: 2011-04-29
TIME	3 Bytes, Bsp.: 13:00:00
TIMESTAMP	Zeitstempel, Bsp.: 2011-04-29-12.00.00.000000
CHAR (n)	String der Länge <i>n.</i> Verbraucht immer <i>n</i> Bytes.
VARCHAR (n)	2 bis n+2 Bytes
GRAPHIC (n)	2n Bytes (double-byte graphic String)
VARGRAPHIC (n)	2 bis 2n+2 Bytes (varying-length graphic String)





Beschreibung		
Binary Large Object mit bis zu 2 ³¹ -1 Byte, Zuweisung oder Vergleich mit Werten anderer Datentypen		
Textdatenobjekt mit bis zu 2 ³¹ -1 Byte an Einbyte-Zeichendaten		
Double-Byte Textdatenobjekt mit bis zu 2 ³¹ -2 Byte Doppelbyte-Zeichendaten		
XML-Dokument (korrekt formatiert), Bis zu 2 GB		
Bis zu 40 Bytes (zur Identifizierung von Zeilen in einem DB2- Subsystem)		



- INSERT ermöglicht das Einfügen von Werten in eine Tabelle.
- Beim Einfügen prüft das RDBMS die Integritätsbedinungen und die Datentypen.

```
INSERT INTO Person (id, name, birthdate) VALUES
(001, "John Doe", '2003/05/03')
```

- http://docs.oracle.com/cd/E11882_01/server.112/e26088/state ments_9014.htm
- http://www.datenbank-sql.de/insert.htm