Grundlagen: Datenbanken

1. Zentralübung - WS 16/17

Harald Lang, Linnea Passing

gdb@in.tum.de

11.1.2017

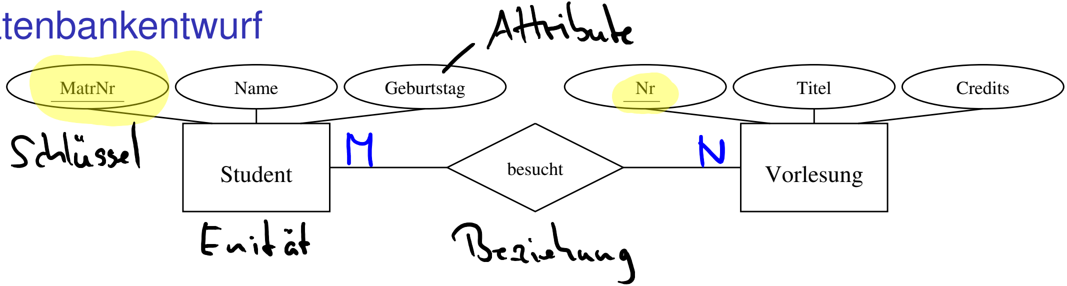
Prüfungstermin 01.03.2017, 10:30 Uhr

Anmeldung bis 15.01.2017, 23:59 Uhr

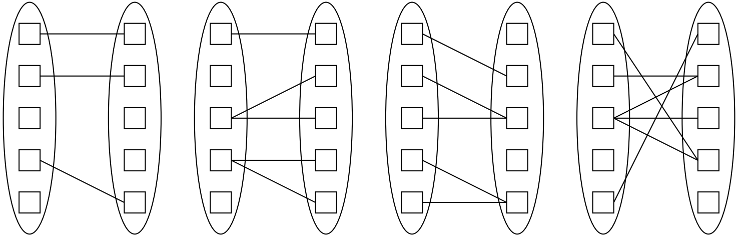
Diese Folien finden Sie online.

Die Mitschrift stellen wir im Anschluss online.

Datenbankentwurf



Funktionalitäten (Integritätsbedingungen)



1:1 1:N N:1 M:N

Das Relationale Modell

Definition

- Eine relationale Datenbank enthält eine Menge von Relationen
- Eine Relation R besteht aus zwei Bestandteilen:
  - Einer **Instanz** R: eine Tabelle mit Zeilen und Spalten; der *aktuelle Inhalt* der Relation (auch Ausprägung genannt)
  - Einem **Schema** $\mathcal{R}$: spezifiziert den *Namen der Relation* und die *Namen und Datentypen der Spalten*; legt die Struktur der Relation fest

Das Relationale Modell

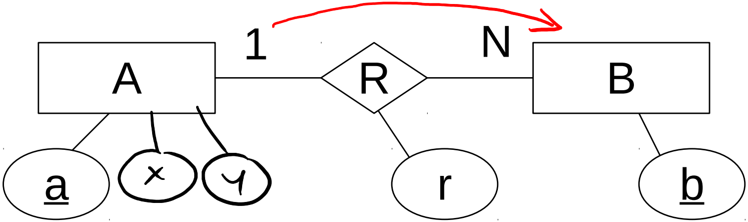
Beispielausprägung:

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	10
27550	Schopenhauer	6
...	...	...

Schema:

- 3 Attribute: MatrNr, Name, Semester
- das Schema assoziiert jedes Attribut mit einer Domäne (Wertebereich)
  - $D_{MatrNr} = \mathrm{dom}(MatrNr) = \mathrm{Integer} = [-2^{31}, 2^{31})$
  - ...
  - $Studenten \subseteq \mathrm{dom}(MatrNr) \times \mathrm{dom}(Name) \times \mathrm{dom}(Semester)$
  - $Studenten \subseteq integer \times string \times integer$
- Schreibweisen:
  - $Studenten : \{[MatrNr : \mathrm{int}, Name : \mathrm{string}, Semester : \mathrm{int}]\}$
  - $Studenten : \{[MatrNr, Name, Semester]\}$
  - $Studenten = \{MatrNr, Name, Semester\}$
  - $Studenten(MatrNr, Name, Semester)$

ER-Modell in Schema überführen und verfeinern



$A: \{\underline{a}, x, y\}$

$B: \{\underline{b}\}$

$R: \{a, \underline{b}, r\}$

Verfeinerung:

$A: \{\underline{a}, x, y\}$

$B: \{\underline{b}, r, a\}$

Relationale Algebra

Algebraische Operatoren:

Projektion	$\Pi_{A_1,\ldots,A_n}$
Selektion	σ_p
Kreuzprodukt	$\times$
Verbund (Join)	$\bowtie_\theta$, $⟕_\theta$, $⟖_\theta$, $⟗_\theta$, $\ltimes_\theta$, $\rtimes_\theta$, $\rhd_\theta$, $\lhd_\theta$
Mengenoperationen	$\cup$, $\cap$, $\setminus$
Division	$\div$ ALL - Quatifizierung
(Gruppierung/Aggregation)	$\Gamma_{A_1,\ldots,A_n\,;\,a_1:f_1,\ldots,a_m:f_m}$
Umbenennung	ρ_N, oder $\rho_{a_1\leftarrow b_1,\ldots,a_n\leftarrow b_n}$

Anmerkung: Natural-Join vs. allgemeiner Theta-Join

	Natural	Theta
Inner	$\bowtie$	$\bowtie_\theta$
Outer	$⟕$, $⟖$, $⟗$	$⟕_\theta$, $⟖_\theta$, $⟗_\theta$
Semi	$\ltimes$, $\rtimes$	$\ltimes_\theta$, $\rtimes_\theta$
Anti	$\rhd$, $\lhd$	$\rhd_\theta$, $\lhd_\theta$

- Natural
  - Implizite Gleichheitsbedingung auf gleichnamigen Attributen
  - Die gleichnamigen Attribute tauchen im Ergebnis nur einmal auf (inner und outer).
- Theta
  - Explizite (beliebige) Joinbedingung: θ.
  - Im Falle von Inner- und Outer-Join werden alle Attribute der beiden Eingaberelationen in das Ergebnis projiziert.

Übung: Relationale Algebra (1)

Finde Studenten (nur Namen ausgeben), die im gleichen Semester sind wie Feuerbach.

ρ_S

Studenten

Übung: Relationale Algebra (2)

Finde Studenten (nur MatrNr ausgeben), die alle Vorlesungen gehört haben.

Relationale Entwurfstheorie

Relationale Entwurftheorie

Funktionale Abhängigkeiten (kurz FDs, für functional dependencies):

- Seien α und β Attributmengen eines Schemas $\mathcal{R}$.
- Wenn auf $\mathcal{R}$ die FD $\alpha \rightarrow \beta$ definiert ist, dann sind nur solche Ausprägungen R zulässig, für die folgendes gilt:
  - Für alle Paare von Tupeln $r, t \in R$ mit $r.\alpha = t.\alpha$ muss auch gelten $r.\beta = t.\beta$.

Übung: Relationenausprägung vervollständigen

Gegen seien die folgende Relationenausprägung und die funktionalen Abhängigkeiten. Bestimmen Sie zunächst x und danach y, sodass die FDs gelten.

$$\begin{aligned} B &\rightarrow A \\ AC &\rightarrow D \end{aligned}$$

A	B	C	D
7	3	5	8
x	4	2	8
7	3	6	9
1	4	2	y

Funktionale Abhängigkeiten

Seien α,β,γ,$\delta \subseteq \mathcal{R}$

Axiome von Armstrong:

- *Reflexivität:*
  Falls $\beta \subseteq \alpha$, dann gilt immer $\alpha \to \beta$
- *Verstärkung*:
  Falls $\alpha \to \beta$ gilt, dann gilt auch $\alpha\gamma \to \beta\gamma$
- *Transitivität*:
  Falls $\alpha \to \beta$ und $\beta \to \gamma$ gelten, dann gilt auch $\alpha \to \gamma$

Mithilfe dieser Axiome können alle *geltenden* FDs hergeleitet werden.

Sei F eine FD-Menge. Dann ist F^+ die Menge aller geltenden FDs (Hülle von F)

Funktionale Abhängigkeiten

Nützliche und vereinfachende Regeln:

- *Vereinigungsregel*:
  Falls $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$ gelten, dann gilt auch $\alpha \rightarrow \beta\gamma$
- *Dekompositionsregel*:
  Falls $\alpha \rightarrow \beta\gamma$ gilt, dann gilt auch $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$
- *Pseudotransitivitätsregel*:
  Falls $\alpha \rightarrow \beta$ und $\gamma\beta \rightarrow \delta$ gelten, dann gilt auch $\gamma\alpha \rightarrow \delta$

Schlüssel

- Schlüssel identifizieren jedes Tupel einer Relation $\mathcal{R}$ eindeutig.
- Eine Attributmenge $\alpha \subseteq \mathcal{R}$ ist ein **Superschlüssel**, gdw. $\alpha \rightarrow \mathcal{R}$
- Ist α zudem noch *minimal*, ist es auch ein **Kandidatenschlüssel** (meist mit κ bezeichnet)
  - Es existiert also kein $\alpha' \subset \alpha$ für das gilt: $\alpha' \rightarrow \mathcal{R}$

- I.A. existieren mehrere Super- und Kandidatenschlüssel.
- Man muss sich bei der Realisierung für einen Kandidatenschlüssel entscheiden, dieser wird dann **Primärschlüssel** genannt.
- Der triviale Schlüssel $\alpha = \mathcal{R}$ existiert immer.

Übung: Schlüsseleigenschaft von Attributmengen ermitteln

- Ob ein gegebenes α ein Schlüssel ist, kann mithilfe der Armstrong Axiome ermittelt werden (i.A. zu aufwendig!)
- Besser: Die **Attributhülle** $AH(\alpha)$ bestimmen.

- Beispiel: $\mathcal{R} = \{\ A\ ,\ B\ ,\ C\ ,\ D\ \}$, mit $F_{\mathcal{R}} = \{AB \to CD, B \to C, D \to B\}$

$AH(\{D\})$:

$AH(\{A, D\})$:

$AH(\{A, B, D\})$:

Normalformen: 1NF $\supset$ 2NF $\supset$ 3NF $\supset$ BCNF $\supset$ 4NF

- **1. NF**: Attribute haben nur atomare Werte, sind also nicht mengenwertig.
- **2. NF**: Jedes Nichtschlüsselattribut (NSA) ist voll funktional abhängig von jedem Kandidatenschlüssel.
  - linke Seite minimal
  - β hängt **voll funktional** von α ab ($\alpha \xrightarrow{\bullet} \beta$), gdw. $\alpha \to \beta$ und es existiert kein $\alpha' \subset \alpha$, so dass $\alpha' \to \beta$ gilt.
- **3. NF**: Frei von transitiven Abhängigkeiten (*in denen NSAe über andere NSAe vom Schlüssel abhängen*).
  - für alle geltenden nicht-trivialen FDs $\alpha \to \beta$ gilt entweder
    - α ist ein Superschlüssel, oder
    - jedes Attribut in β ist in einem Kandidatenschlüssel enthalten
- **BCNF**: Die linken Seiten (α) aller geltenden nicht-trivalen FDs sind Superschlüssel.
- **4. NF**: Die linken Seiten (α) aller geltenden nicht-trivalen MVDs sind Superschlüssel.

Mehrwertige Abhängigkeiten

multivalued dependencies (MVDs)

"Halb-formal":

- Seien α und β disjunkte Teilmengen von $\mathcal{R}$
- und $\gamma = (\mathcal{R} \backslash \alpha) \backslash \beta$
- dann ist β mehrwertig abhängig von α ($\alpha \twoheadrightarrow \beta$), wenn in jeder gültigen Ausprägung von $\mathcal{R}$ gilt:
- Bei zwei Tupeln mit gleichem α-Wert kann man die β-Werte vertauschen, und die resultierenden Tupel müssen auch in der Relation enthalten sein.

Wichtige Eigenschaften:

- Jede FD ist auch eine MVD (gilt i.A. nicht umgekehrt)
- wenn $\alpha \twoheadrightarrow \beta$, dann gilt auch $\alpha \twoheadrightarrow \gamma$ (Komplementregel)
- $\alpha \twoheadrightarrow \beta$ ist trivial, wenn $\beta \subseteq \alpha$ ODER $\alpha \cup \beta = \mathcal{R}$ (also $\gamma = \emptyset$)

Beispiel: Mehrwertige Abhängigkeiten

Beispiel: $R = \{\text{ProfessorIn}, \text{Vorlesung}, \text{AssistentIn}\}$

ProfessorIn	Vorlesung	AssistentIn
K	GDB	Linnea
K	Tx Systems	Linnea

Übung: Höchste NF bestimmen

$$\mathcal{R} : \{[\ A\,, B\,, C\,, D\,, E\ ]\}$$

$$\begin{aligned} A &\rightarrow BCDE \\ AB &\rightarrow C \end{aligned}$$

- ○ 1. NF
- ○ 2. NF
- ○ 3. NF
- ○ BCNF
- ○ 4. NF
- ○ keine der angegebenen

Übung: Höchste NF bestimmen (2)

$$\mathcal{R} : \{[\ A\,,\, B\,,\, C\,,\, D\,,\, E\ ]\}$$

$$\begin{aligned} A &\rightarrow BCDE \\ B &\rightarrow C \end{aligned}$$

- ○ 1. NF
- ○ 2. NF
- ○ 3. NF
- ○ BCNF
- ○ 4. NF
- ○ keine der angegebenen

Schema in 3. NF überführen

Synthesealgorithmus

- Eingabe:
  - **Kanonische Überdeckung** $\mathcal{F}_c$
    - Linksreduktion
    - Rechtsreduktion
    - FDs der Form $\alpha \rightarrow \emptyset$ entfernen (sofern vorhanden)
    - FDs mit gleicher linke Seite zusammenfassen
- Algorithmus:
  1. Für jede FD $\alpha \rightarrow \beta$ in $\mathcal{F}_c$ forme ein Unterschema $\mathcal{R}_\alpha = \alpha \cup \beta$, ordne $\mathcal{R}_\alpha$ die FDs $\mathcal{F}_\alpha := \{\alpha' \rightarrow \beta' \in \mathcal{F}_c \mid \alpha' \cup \beta' \subseteq \mathcal{R}_\alpha\}$ zu
  2. Füge ein Schema $\mathcal{R}_\kappa$ mit einem Kandidatenschlüssel hinzu
  3. Eliminiere redundante Schemata, d.h. falls $\mathcal{R}_i \subseteq \mathcal{R}_j$, verwerfe $\mathcal{R}_i$
- Ausgabe:
  - Eine Zerlegung des unsprünglichen Schemas, wo alle Schemata in 3.NF sind.
  - Die Zerlegung ist **abhängigkeitsbewahrend** und **verlustfrei**.

Übung: Synthesealgorithmus

$$\mathcal{R} : \{[\ A\,,\, B\,,\, C\,,\, D\,,\, E\,,\, F\ ]\}$$

$$\begin{aligned} B &\rightarrow ACDEF \\ EF &\rightarrow BC \\ A &\rightarrow D \end{aligned}$$

Schema in BCNF überführen

BCNF-Dekompositionsalgorithmus (nicht abhängigkeitsbewahrend)

- Starte mit $Z = \{\mathcal{R}\}$
- Solange es noch ein $\mathcal{R}_i \in Z$ gibt, das nicht in BCNF ist:
  - Finde eine FD $(\alpha \rightarrow \beta) \in F^+$ mit
    - $\alpha \cup \beta \subseteq \mathcal{R}_i$ (FD muss in $\mathcal{R}_i$ gelten)
    - $\alpha \cap \beta = \emptyset$ (linke und rechte Seite sind disjunkt)
    - $\alpha \rightarrow \mathcal{R}_i \notin F^+$ (linke Seite ist kein Superschlüssel)
  - Zerlege $\mathcal{R}_i$ in $\mathcal{R}_{i.1} := \alpha \cup \beta$ und $\mathcal{R}_{i.2} := \mathcal{R}_i - \beta$
  - Entferne $\mathcal{R}_i$ aus Z und füge $\mathcal{R}_{i.1}$ und $\mathcal{R}_{i.2}$ ein, also $Z := (Z - \{\mathcal{R}_i\}) \cup \{\mathcal{R}_{i.1}\} \cup \{\mathcal{R}_{i.2}\}$

Schema in 4.NF überführen

4NF-Dekompositionsalgorithmus (nicht abhängigkeitsbewahrend)

- Starte mit $Z = \{\mathcal{R}\}$
- Solange es noch ein $\mathcal{R}_i \in Z$ gibt, das nicht in 4NF ist:
  - Finde eine MVD $\alpha \twoheadrightarrow \beta \in \mathcal{F}^+$ mit
    - $\alpha \cup \beta \subset \mathcal{R}_i$ (FD muss in $\mathcal{R}_i$ gelten)
    - $\alpha \cap \beta = \emptyset$ (linke und rechte Seite sind disjunkt)
    - $\alpha \rightarrow \mathcal{R}_i \notin \mathcal{F}^+$ (linke Seite ist kein Superschlüssel)
  - Zerlege $\mathcal{R}_i$ in $\mathcal{R}_{i.1} := \alpha \cup \beta$ und $\mathcal{R}_{i.2} := \mathcal{R}_i - \beta$
  - Entferne $\mathcal{R}_i$ aus Z und füge $\mathcal{R}_{i.1}$ und $\mathcal{R}_{i.2}$ ein, also $Z := (Z - \{\mathcal{R}_i\}) \cup \{\mathcal{R}_{i.1}\} \cup \{\mathcal{R}_{i.2}\}$

Übung: BCNF-Dekompositionsalgorithmus

$$\mathcal{R} = \{\, A,\, B,\, C,\, D,\, E,\, F\},\, F_{\mathcal{R}} = \{B \rightarrow AD,\, DEF \rightarrow B,\, C \rightarrow AE\}$$