

Datenbanken AbschlussveranstaltungNormalformen1. Normalform

→ atomare Werte

2. Normalform

→ 1. Normalform gilt

→ jedes Attribut muss vom gesamten Schlüssel abhängig3. Normalform

→ 1. + 2. Normalform liegen vor

→ keine funktionale Abhängigkeit zwischen Nichtschlüsselattributen

⇒ wenn alle Daten weiterhin erhalten bleiben sollen, muss die Tabelle in weitere Tabellen aufgespalten werden

• in Klausur sind nur Tabellen mit Spaltennamen vorgegeben, kein Tabelleninhalt

Relationen - Übung

R1 (A, B, C, D, E)

 $C \rightarrow A$ 

R2 (A, C, F)

 $(BC) \rightarrow E$  $(A, C) \rightarrow F$  $F \rightarrow A$  $A \rightarrow D$  $A \rightarrow (B, E)$ 

1. Bestimmen Sie alle Schlüsselkandidaten der beiden Relationen R1 und R2.

R1: C, da ist mit C alle anderen Attribute direkt oder indirekt ermitteln kann.

R2: C, A wenn nur hinsichtlich Normalform argumentiert wird, Schlüssel ist dann allerdings nicht mehr minimal, sonst nur C

2. Ermitteln Sie, welche Normalform die beiden Relationen genügen.

R1: Ausgegangen werden kann davon, dass die 1. besteht, 2. genügt nicht, da vom Schlüsselkandidaten (hier C) nicht direkt auf alle Nichtschlüsselattribute geschlossen werden kann. 3. kann nicht gelten, da Attribute nicht direkt von Nichtschlüsselattributen abhängen.

R2: für A, C dann 1. + 2. 3. gegeben, wenn C dann nur 1. da F auch von A und somit einem Nichtschlüsselattribut abhängig, 3. nicht gegeben da F auch von A abhängt.

Übung Normalform

Bestellungen						
BestellNr	KundenNr	WunschLiefDatum		BestellPosNr	ArtikelNr	Anzahl
1	K001	1.1.19	1.1.19	1	a01	10
1	K001	1.1.19	2.1.19	2	a02	20
1	K001	1.1.19	3.1.19	3	a03	30

*zwei Interpretationsmöglichkeiten*

Kunden					
KundenNr	Nachname	Bundesland	Ort	PLZ	Strasse

→ von PLZ kann auf das Bundesland geschlossen werden (⇒ nicht 3.NF)

⇒ aufspalten in 2 Tabellen

1. PLZ - Verzeichnis

PLZ	Bundesland	Ort

2. Kunde

KundenNr	Nachname	PLZ	Straße

Anomalien

→ bei Update, Insert, Delete muss auf besondere Fehlersituationen geachtet werden

## Group By

gruppiert wird, wenn jeder Wert in den aufgeführten Spalten identisch ist

sofern eine Gruppierung vorliegt, muss jeder Ausdruck in SELECT aggregiert oder gruppiert werden.

## Subqueries

### Korrelation

nichtkorreliert: wenn Unterabfrage von der oberen nicht abhängig ist

korreliert: Unterabfrage ist von oberer abhängig.

nicht korreliert:

- Auswertung der Unterabfrage
- Einsetzen in Oberabfrage

korreliert:

- teilweise Auswertung in der Oberabfrage
- Zeilenweises Einsetzen in der Unterabfrage
- Einsetzen in die Oberabfrage

## Join

- join ohne Bedingung ist ein Kreuzprodukt

### Left Outer join

- füllt rechts mit Null-Werten auf

### Right Outer join

- füllt links mit Null-Werten auf

Wenn mit (left) outer join angefangen wird sollte bei jedem weiteren join auch mit Outer gearbeitet werden, sonst werden Zeilen mit Null wieder herausgelöscht.

## Typische Fehler

- Sei Exists: Beliebige Anzahl an Zeilen
- Sei Vergleich: Eine Zeile, Spalte
- bei Skalarer Wert: " ( feste Erwartungswerte)
- bei in: viele Zeilen, eine Spalte

## Constraints

28.01.19

### Integrity Constraints

#### Entity Integrity

Primärschlüssel dürfen nicht Null sein

#### Referential Integrity

Fremdschlüssel sind entweder undefiniert oder es gibt einen entsprechenden Eintrag mit ihrem Primärschlüssel in einer anderen Tabelle

### Referential Constraints

Aktionen die bei Änderungen an den Daten durchgeführt werden sollen.

⇒ z.B. on delete constraints

#### Constraints Namen

Vergabe von Namen, zum späteren Löschen, (de-)aktivieren oder auch Fehlermeldungen, wenn ein Constraint fehlschlägt.