

# **Datenbanksysteme**

## Sommersemester 2009

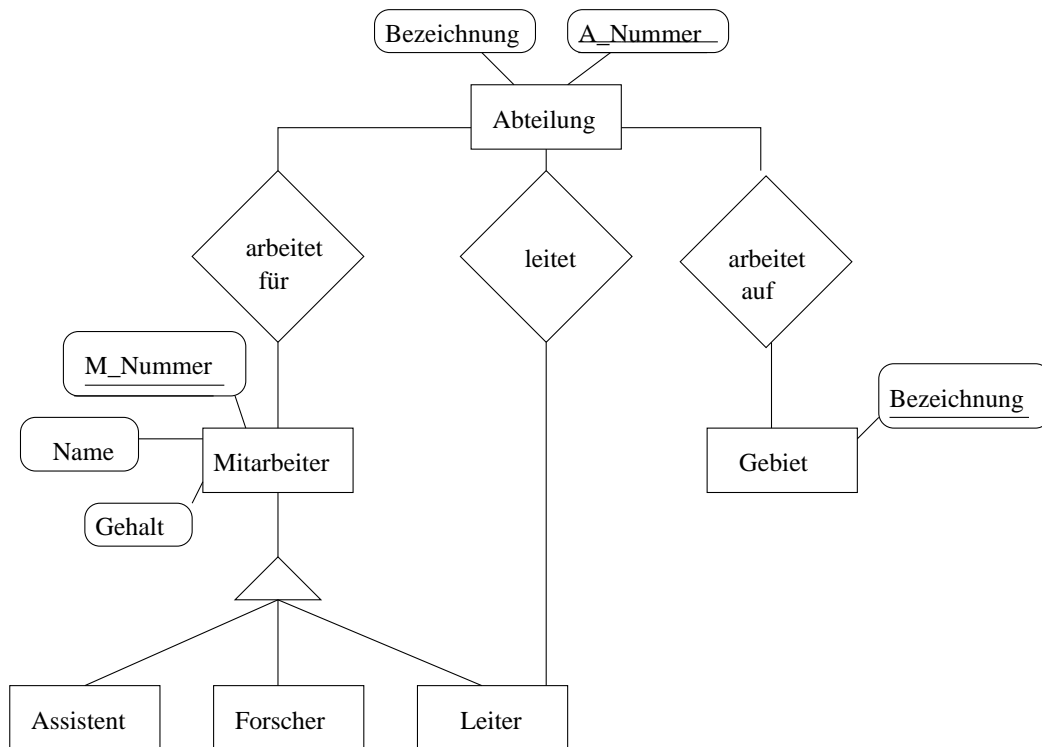
### **13. Übungsblatt (Probeklausur)**

#### **Hinweise:**

- Es ist sinnvoll, die Aufgaben zum Test zunächst in der in der Klausur vorgegebenen Zeit zu lösen (90 Minuten).
- Anschliessend sollten Sie die Aufgaben gründlicher bearbeiten und Ihre Lösungen angeben.
- Die angegebenen Punkte dienen nur Ihrer Orientierung und werden nicht für das Erlangen der Zugangsvoraussetzungen zur Klausur berücksichtigt.

## Aufgabe 1: ER-Modell

Gegeben sei das folgende ER-Modell, das ein Forschungslabor modelliert.



- Erweitern Sie das ER-Modell um den Aspekt, dass Forscher an einem oder mehreren Themen forschen. Diese Themen haben eine Bezeichnung und gehören zu einem bestimmten Gebiet. (4 Punkte)
- Geben Sie die Kardinalitäten der Beziehungen im ER-Modell an. Geben Sie außerdem den genauen Typ der Beziehung zwischen **Mitarbeiter**, **Assistent**, **Forscher** und **Leiter** an. (8 Punkte)
- Überführen Sie das Original-ER-Modell in ein relationales Modell. Geben Sie Schlüssel und Fremdschlüssel an. Falls weitere Einschränkungen erforderlich sind, um das relationale Modell äquivalent zum ER-Modell zu machen, geben Sie dieses in Textform an. (8 Punkte)

## Aufgabe 2: Anfragesprachen

Gegeben seien die folgenden Relationen, die Bestandteil einer Zugverbindungsdatenbank sind.

- Passagier(p\_id, name, adresse)
- Bahnhof(b\_id, name, stadt, gleise : int)
- Verbindung(v\_id, start → Bahnhof, ziel → Bahnhof, abfahrt: Zeit, ankunft: Zeit, zugkategorie)
- Zwischenstop(verbindung → Verbindung, stop → Bahnhof)
- Ticket(t\_id, passagier → Passagier, preis : int, von → Bahnhof, bis → Bahnhof, zugkategorie)

Unterstrichene Attribute markieren Schlüssel. Pfeile (→) markieren Fremdschlüssel gefolgt von der referenzierten Relation. Falls nicht anders angegeben, ist der Typ eines Attributs eine Zeichenkette. Der Attributtyp *zeit* bezeichnet eine Zeit bestehend aus Datum und Uhrzeit. Die Relation *zwischenstop* enthält alle Zwischenstops an weiteren Bahnhöfen einer Verbindung. Das Attribut *zugkategorie* gibt an, welche Züge für eine Verbindung eingesetzt werden bzw. für welche Verbindungsarten ein Ticket gültig ist. Ein Ticket ist für eine Verbindung gültig, wenn die Zugkategorie übereinstimmt und der mit *von* und der mit *bis* referenzierte Bahnhof Zwischenstops der Verbindung sind. *von* und *bis* können auch Start- bzw. Zielbahnhof der Verbindung sein.

- a) Formulieren Sie die folgenden Anfragen in SQL. Ihre Anfragen sollten keine Duplikate liefern, aber ohne unnötige 'DISTINCT'-Anweisungen auskommen.
- (i) Geben Sie den Start- und Zielbahnhof (Name und Stadt) der Verbindung mit der *v\_id* v003 aus. (2 Punkte)
  - (ii) Wieviel Geld hat jeder einzelne Passagier für Tickets ausgegeben. (3 Punkte)
  - (iii) Geben Sie alle Verbindungen (Start-, Zielbahnhofs-id, Abfahrt und Ankunft) aus, die an Bahnhöfen mit dem Namen *Hamburg* halten. (3 Punkte)
- b) Geben Sie einen Ausdruck der Relationen Algebra an, der die Anfrage aus a)i) äquivalent stellt. (3 Punkte)
- c) Übersetzen Sie den folgenden Ausdruck der Relationen Algebra in eine äquivalente SQL-Anfrage. (3 Punkte)
- $$\pi_{preis}(Ticket) - \pi_{a.preis}(Ticket \bowtie_{a.preis > b.preis} Ticket)$$

d) Geben Sie einen Ausdruck des Tupelkalküls an, der die Anfrage aus a)iii) äquivalent stellt. (3 Punkte)

e) Übersetzen Sie den folgenden Ausdruck des Tupelkalküls in eine äquivalente SQL-Anfrage. (3 Punkte)

$$\begin{aligned} & \{i^{(1)} \mid (\exists v)(\exists t)(\exists s1)(\exists s2) \\ & \wedge (\text{Verbindung}(v) \wedge \text{Ticket}(t) \wedge \text{Zwischenstop}(s1) \wedge \text{Zwischenstop}(s2) \\ & \wedge v[1] = i[1] \wedge v[1] = s1[1] \wedge v[1] = s2[1] \wedge t[1] = 't001' \\ & \wedge t[6] = v[6] \wedge (t[4] = v[2] \vee t[4] = s1[2]) \\ & \wedge (t[5] = v[3] \vee t[5] = s2[2])\} \end{aligned}$$

### Aufgabe 3: Funktionale Abhängigkeiten

Gegeben sei die Relation  $R(A, B, C, D, E, F, G)$  mit den folgenden funktionalen Abhängigkeiten  $F$ :

$$\begin{aligned}A &\rightarrow B, & A &\rightarrow F, \\F &\rightarrow C, & DA &\rightarrow B, \\FA &\rightarrow C, & BCA &\rightarrow GD\end{aligned}$$

- Bestimmen Sie alle zu  $F$  und  $R$  gehörigen Schlüssel. (2 Punkte)
- Bestimmen Sie zu  $F$  eine minimale Menge  $F'$  von Abhängigkeiten. (6 Punkte)
- Zeigen Sie, ob die funktionale Abhängigkeit  $AFE \rightarrow ED$  in  $F^+$  ist. Benutzen Sie dabei nur die Armstrong-Axiome (Reflexivität, Erweiterung, Transitivität) oder ein Gegenbeispiel auf  $F'$ . (4 Punkte)
- Zeigen Sie (wie oben), ob die funktionale Abhängigkeit  $F \rightarrow A$  in  $F^+$  ist. (2 Punkte)
- Erhält die folgende Dekomposition die Abhängigkeiten aus  $F$ ?

$$R_1(ABCEF)$$

$$R_2(ACD)$$

$$R_3(ACG)$$

Begründen Sie Ihre Antwort.

(6 Punkte)

#### Aufgabe 4: Serialisierbarkeit von Transaktionen

Im Folgenden sind Schedules für je 4 Transaktionen T1, T2, T3 und T4 gegeben. Überprüfen Sie anhand eines Präzedenzgraphen, ob die Schedules serialisierbar sind und geben Sie ggf. einen äquivalenten seriellen Schedule an.

a) 1. Schedule

	T1	T2	T3	T4
(1)			LOCK X	
(2)			LOCK Y	
(3)			UNLOCK X	
(4)		LOCK X		
(5)			UNLOCK Y	
(6)	LOCK Y			
(7)		UNLOCK X		
(8)	LOCK Z			
(9)	LOCK X			
(10)				LOCK W
(11)	UNLOCK X			
(12)	UNLOCK Y			
(13)	UNLOCK Z			
(14)				UNLOCK W
(15)				LOCK Z
(16)				UNLOCK Z
(17)			LOCK Z	
(18)			UNLOCK Z	

(10 Punkte)

b) 2. Schedule

	T1	T2	T3	T4
(1)		LOCK A		
(2)		LOCK B		
(3)		UNLOCK A		
(4)			LOCK A	
(5)		UNLOCK B		
(6)	LOCK B			
(7)			UNLOCK A	
(8)	LOCK A			
(9)	LOCK C			
(10)	UNLOCK A			
(11)				LOCK A
(12)				UNLOCK A
(13)	UNLOCK B			
(14)	UNLOCK C			

(10 Punkte)

---

**Abgabetermin:** Dienstag, den 14. Juli 2009 bis 10 Uhr im Institut für Informationssysteme