

Hausarbeit zur Vorlesung Datenbanksysteme

Dieses Dokument enthält die Aufgabenstellungen, die in der Hausarbeit beantwortet werden sollen. Es sollen alle 5 Aufgaben bearbeitet werden. Insgesamt können 59 Punkte erreicht werden. Die Punktzahl ist bei jeder Aufgabe angegeben.

- **Beachten Sie die Formalien auf dem Hinweisblatt!**
- **Geben Sie Ihre Matrikelnummer auf der Hausarbeit an!**
- **Akzeptieren Sie die Eigenständigkeitserklärung auf Uni2Work!**

Aufgabe 1 Offene Fragen

(6 Punkte)

Erörtern Sie folgende Fragestellungen. Belegen Sie Ihre Aussagen mit Quellen und nutzen Sie Zitate richtig!

(Alle Teilaufgaben müssen in einer durchsuchbaren Form vorliegen. D.h. alle Wörter und Textbausteine müssen mit STRG+F auffindbar sein.)

- (a) Einige Datenbanken bieten die in der Vorlesung besprochenen SQL-Mengenoperationen INTERSECT und EXCEPT nicht an. Erläutern Sie Vor- und Nachteile für den Benutzer und für den Hersteller des Datenbanksystems. (Die Antwort sollte in etwa 200 Wörter umfassen.)

Geben Sie zusätzlich mit Hilfe einer kleinen Beispielrelation sowohl für INTERSECT als auch EXCEPT jeweils zwei SQL Anfragen an, welche unterschiedliche (in der Vorlesung besprochene) sprachliche Mittel von SQL nutzen, um die jeweilige Operation zu simulieren.

- (b) Gegeben sei die Relation Produkt(Name, Größe, Gewicht, Preis, Menge). Ihnen wurde von einem Kollegen folgende Aussage im Tupelkalkül mitgeteilt, die Sie ausführen sollen.

$Schema(a) = (Name : String)$

$\{a | (\exists b \in Produkt)(a.Name = b.name) \wedge (b.preis > 209) \wedge$

$(\forall c \in Produkt)((c.Größe > b.Größe \vee c.Gewicht > b.Gewicht) \Rightarrow (c.Preis > b.Preis \wedge c.Menge < b.Menge))\}$

- i. Geben Sie in Worten wieder, was mit Hilfe dieser Aussage gesucht wird.

- ii. Auf Sie wirkt diese Aussage außerdem zunächst sehr unverständlich. Formulieren Sie die Aussage daher so um, dass anschließend ein Ausdruck herauskommt, der ein äquivalentes Ergebnis liefert und die folgende Formel beinhaltet:

$$\left(\left(\neg(c.Größe > b.Größe) \vee \neg(c.Gewicht > b.Gewicht) \right) \wedge \left(\neg(c.Preis > b.Preis) \vee \neg(c.Menge < b.Menge) \right) \right)$$

Tragen Sie dazu für die \neg -Symbole ein Negationssymbol (\neg) ein falls erforderlich. Es dürfen durchaus auch alle weiteren Bestandteile der Ursprungs-Aussage angepasst werden. Geben Sie alle notwendigen Zwischenschritte an.

(Sie dürfen die Teilaufgabe (ii) auch handschriftlich bearbeiten und als Bild einfügen.)

Aufgabe 2 ER-Diagramm

(14 Punkte)

Um die Prozesse von Krankenhäusern besser planen zu können, wollen Sie eine entsprechende Datenbank anlegen. Diese soll einige Anforderungen erfüllen:

- Jedes Krankenhaus hat einen eindeutigen Namen, eine Adresse und ein festgelegtes Budget.
- Ein Patient hat ebenfalls einen Namen, ein Geburtsdatum, ein Geschlecht sowie eine Körpergröße.
- Ein Krankenhaus kann mehrere Patienten aufnehmen. Hierbei soll gespeichert werden, an welchem Datum ein Patient in das Krankenhaus eingewiesen wurde und wann er wieder entlassen wird. Ein Patient kann im Laufe seines Lebens in mehrere Krankenhäuser eingewiesen werden.
- In einem Krankenhaus arbeiten außerdem mehrere Ärzte, wobei ein Arzt der Chefarzt eines Krankenhauses ist.
- Ein Arzt arbeitet immer nur für ein Krankenhaus und hat einen Namen, ein Gehalt sowie ein Fachgebiet.
- Ärzte können Behandlungen an Patienten durchführen. Diese Behandlungen finden in einem Operationsraum statt. Hierbei wird immer die Art der Behandlung gespeichert. Ein Patient kann durchaus mehrmals behandelt werden, wobei sich die beteiligten Ärzte und Operationsräume gleichen können, aber nicht müssen.
- Ein Krankenhaus besteht aus mehreren Räumen. Jeder Raum befindet sich natürlich in einem Krankenhaus und hat eine eindeutige Raumnummer sowie eine Raumgröße.
- Bei Operationsräumen wird zusätzlich gespeichert, wann diese zuletzt gewartet wurden.
- Außerdem gibt es noch Besprechungsräume. Diese weisen eine gewisse Anzahl an Sitzplätzen auf. Besprechungsräume können von einem oder mehreren Ärzten verwendet werden. Ein Arzt verwendet allerdings immer den gleichen Besprechungsraum. Damit es hierbei nicht zu Doppelbelegungen kommt, wird gespeichert, an welchem Wochentag ein Besprechungsraum einem Arzt zusteht.

Übertragen Sie die obigen Anforderungen in ein E/R-Diagramm indem Sie so wenig Entities und Attribute wie nötig verwenden. Benutzen Sie hierzu die in der Vorlesung vorgestellte Notation. Markieren Sie die Funktionalität jeder Beziehung. Die Schlüsselkandidaten müssen nicht markiert werden.

Verwenden Sie in Ihrem Diagramm mindestens eine sinnvolle ternäre Beziehung.

Hinweis: Entscheiden Sie sich bei der Angabe der Funktionalitäten für eine der Notationen. Andernfalls müssen wir die schlechtere Antwort werten.

(Sie dürfen diese Aufgabe auch handschriftlich bearbeiten und als Bild einfügen.)

Aufgabe 3 Relationale Algebra, SQL und Kalküle

(22 Punkte)

Gegeben sei das folgende Datenbankschema zur Organisation von Hackathons, zusammen mit einem Teilauszug der Ausprägungen. Dabei sind die Primärschlüssel unterstrichen. Fremdschlüssel-Attribute heißen wie die assoziierten Relationen. Das Attribut 'Leiter' ist ebenfalls ein Fremdschlüssel.

Die Relation *Teilnehmer* enthält allgemeine Informationen bezüglich Personen, die im System registriert sind. Teilnehmer können *Mitglied* in einer oder mehreren *Gruppen* sein und mit diesen an verschiedenen Hackathons teilnehmen. Jede Gruppe hat einen Teilnehmer, der diese leitet und durch das Attribut *Leiter* angegeben ist.

In der Relation *Hackathon* sind verschiedene Daten der unterschiedlichen Wettbewerbe gespeichert. Die Kapazität gibt die maximale Anzahl teilnehmender Gruppen an. Die Relation *Teilnahme* zeigt, welche Gruppen an welchen Hackathons teilnehmen. Zusätzlich wird gespeichert, wie viele Punkte eine Gruppe bei einem Wettbewerb erzielt hat. Mehr Punkte sind gleichbedeutend mit einem besseren Resultat in einem Wettbewerb. Zudem gibt die Relation *Preise* an, welche Artikel es für die jeweilige Platzierung bei den Hackathons gibt.

Alle Attribute der Primär- und Fremdschlüssel sowie die Attribute 'Kapazität' und 'Platzierung' sind natürliche Zahlen größer 0. 'Punkte' ist eine Festkommazahl mit vier Vorkomma- und zwei Nachkommastellen größer-gleich 0. Datumsangaben werden als Zeichenkette der Länge 10 gespeichert. Die übrigen Attribute sind variable Zeichenketten mit einer maximalen Länge von 100. Aus Integritätsgründen dürfen die Attribute 'Artikel', 'Name', 'Nachname' sowie 'Gruppenname' niemals NULL sein. Außerdem sollen entsprechende Einträge in den Relationen *Preise* und *Teilnahme* automatisch gelöscht werden, wenn ein Hackathon abgesagt und daher aus der Relation *Hackathon* entfernt wird. Sollte ein Tupel aus der Relation *Teilnehmer* entfernt werden, sollen auch entsprechende Einträge in der Relation *Mitglied* gelöscht werden. Wird allerdings ein Tupel aus der Relation *Gruppe* gelöscht, sollen Referenzierungen auf den Wert NULL gesetzt werden.

Die genannten Integritätsbedingungen sind für die Bearbeitung der Aufgabe ausreichend. Vermeiden sie zusätzliche optionale Bedingungen.

(Alle Teilaufgaben (außer Relationale Algebra und Kalküle) müssen in einer durchsuchbaren Form vorliegen. D.h. alle Wörter und Textbausteine müssen mit STRG+F auffindbar sein.)

Teilnehmer

<u>TNr</u>	Vorname	Nachname	Geburtsdatum	Wohnort
1	Fritz	Müller	05.10.1991	München
2	Stefan	Schmidt	23.06.1995	München
3	Anke	Maier	01.03.2000	Berlin
4	Julia	Petersen	14.08.1996	Bochum
...

Gruppe

<u>GNr</u>	Gruppenname	Leiter
1	Mighty Ducks	1
2	MonStars	23
3	Starsheriffs	14
4	Care Bears	8
...

Hackathon

<u>HNr</u>	Name	Datum	Ort	Kapazität
1	Data Hacker	15.04.2020	Berlin	20
2	Securing the IoT	17.02.2020	Berlin	20
3	Fiction2Science	19.03.2020	Hannover	15
4	Transport Hackathon	24.06.2020	Menden	10
...

Mitglied

<u>Teilnehmer</u>	<u>Gruppe</u>
1	1
22	1
12	23
8	4
...	...

Teilnahme

<u>Gruppe</u>	<u>Hackathon</u>	Punkte
1	1	1000,00
1	5	45,50
5	1	150,15
46	2	75,90
...

Preise

<u>Hackathon</u>	Artikel	Gruppe	<u>Platzierung</u>
1	Tablet	1	1
1	Smartphone	46	2
1	Smartwatch	23	3
2	Power Bank	74	1
...

Teilnehmer(TNr, Vorname, Nachname, Geburtsdatum, Wohnort)

Gruppe(GNr, Gruppenname, Leiter[Teilnehmer])

Hackathon(HNr, Name, Datum, Ort, Kapazität)

Mitglied(Teilnehmer, Gruppe)

Teilnahme(Gruppe, Hackathon, Punkte)

Preise(Hackathon, Artikel, Gruppe, Platzierung)

- (a) Geben Sie die Anweisung in **SQL-DDL** an, die notwendig ist, um die Relation 'Mitglied' wie beschrieben zu erzeugen. Achten Sie dabei auf Fremdschlüsselbeziehungen und Integritätsbedingungen. Sie können davon ausgehen, dass die übrigen Relationen bereits existieren.
- (b) Nehmen Sie an, dass in den Relationen 'Teilnehmer', 'Gruppe' und 'Hackathon' jeweils 1000 Einträge enthalten sind und die entsprechenden Primärschlüssel von 1 bis 1000 durchnummeriert wurden. Erläutern Sie, ob es möglich ist, in der Relation 'Teilnehmer' die TNr 36 und 71 miteinander zu vertauschen. Sollte dies möglich sein, geben Sie die hierzu notwendigen SQL-Änderungsoperationen an. Andernfalls begründen Sie, wieso dies nicht zulässig ist. Sie können abermals davon ausgehen, dass alle Relationen bereits wie oben beschrieben existieren. Nutzen Sie die aus der Vorlesung bekannte Syntax.

Formulieren Sie die folgenden Anfragen in **SQL**. Verwenden Sie für die Anfragen nur aus der Vorlesung bekannte SQL-Befehle.

- (c) Geben Sie die GNr und den Namen aller Gruppen an, die an einem Hackathon in Hannover mit einer Kapazität von über 15 teilgenommen haben. Geben Sie zusätzlich die Namen der entsprechenden Hackathons an.

- (d) Geben Sie das Datum an, an dem die meisten Hackathons stattgefunden haben. Nutzen Sie hierzu lediglich die aus der Vorlesung bekannte Syntax.
- (e) Bestimmen Sie für jede GNr die Summe der Punkte, die diese Gruppen insgesamt in allen Hackathons erreicht hat. Sollte eine Gruppe bisher noch keine Punkte erzielt haben, soll 0 ausgegeben werden.
- (f) Formulieren Sie die folgende Anfrage sowohl in Relationaler Algebra als auch im Tupelkalkül: Finden sie die GNr und den Namen aller Gruppen, die an den Hackathons mit den Nummern 17 und 94 teilgenommen haben.

Relationale Algebra: Es dürfen auch die abgeleiteten Operationen verwendet werden.

Tupelkalkül: Es dürfen die Erweiterungen aus den Übungen verwendet werden.

(Sie dürfen diese Teilaufgabe auch handschriftlich bearbeiten und als Bild einfügen.)

Aufgabe 4 Normalisierung

(10 Punkte)

(Alle Teilaufgaben müssen in einer durchsuchbaren Form vorliegen. D.h. alle Wörter und Textbausteine müssen mit STRG+F auffindbar sein.)

- (a) Gegeben sei das Relationenschema $S(A, B, C, D, E, F, G, H)$, sowie die Menge der zugehörigen nicht-trivialen funktionalen Abhängigkeiten \mathcal{F} :

- $G \rightarrow H$
- $F, G \rightarrow E, C$
- $F, G \rightarrow D$
- $B, A \rightarrow C$
- $E \rightarrow A, H$
- $F, G, A \rightarrow B$

Dabei ist $\{F, G\}$ der einzige Schlüsselkandidat.

Begründen Sie kurz, ob das Relationenschema S die Bedingungen der zweiten Normalform erfüllt.

- (b) Gegeben sei das Relationenschema $R(A, B, C, D, E, F, G, H, I)$, sowie die Menge der zugehörigen nicht-trivialen funktionalen Abhängigkeiten:

- $C, B \rightarrow A$
- $E \rightarrow C, F$
- $B, F \rightarrow I$
- $H \rightarrow C$
- $C, F \rightarrow B$
- $F \rightarrow D, G$
- $B, I \rightarrow E$

Bestimmen Sie alle Schlüsselkandidaten der Relation R und begründen Sie jeweils, warum es sich um einen Schlüsselkandidaten handelt. Begründen Sie auch, weshalb es keine weiteren Schlüsselkandidaten gibt.

Aufgabe 5 Transaktionen

(7 Punkte)

Welche der Anomalien

- Dirty Read
- Lost-Update
- Non-repeatable Read
- Phantomproblem

treten in den folgenden Schedules auf? Geben Sie für jede Anomalie an, ob diese mit Sicherheit auftritt und wenn ja bei welchen Objekten. Jede Anomalie kann entweder keinmal, einmal oder mehrmals je Schedule auftreten.

(Alle Teilaufgaben müssen in einer durchsuchbaren Form vorliegen. D.h. alle Wörter und Textbausteine müssen mit STRG+F auffindbar sein.)

(a) $S_1 = (r_2(w), r_2(v), r_1(x), r_2(y), w_2(y), w_2(v), w_1(x), r_1(w), r_1(x), w_2(y), w_2(w), r_2(y))$

(b) $S_2 = (w_1(w), r_2(v), r_1(y), w_1(v), r_1(v), w_2(v), w_1(y), r_2(w), r_2(y), r_1(v), w_1(y), w_2(w))$

(c) $S_3 = (r_2(y), r_2(x), w_1(x), r_1(x), w_1(y), r_2(w), w_2(x), r_1(x), r_2(y), w_2(y), w_2(w), w_2(w))$