
Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
---------------------------	-----------------------	-----------------------------

Kennzahl: _____

Kennwort: _____

**Frühjahr
2020**

46116

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Softwaretechnologie/Datenbanksysteme**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 13

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1 (Aktivitätsprogramm)

[15 PUNKTE]

In dieser Aufgabe wird der fiktive Messenger XYZ für Smartphones betrachtet, welcher auch mit einer Smartwatch interagieren kann, falls diese mit dem Handy verbunden ist. Wir gehen davon aus, dass maximal eine Smartwatch mit dem Handy verbunden ist. Erstellen Sie ein Aktivitätsdiagramm für den im Folgenden beschriebenen Umgang des Messenger XYZ mit neu ankommenden Nachrichten.

Erhält der Messenger XYZ auf dem Handy eine neue Nachricht, überprüft er zuerst, ob das Handy stumm geschaltet ist. Falls das Handy stumm geschaltet ist, zeigt der Messenger XYZ nur eine kurze Benachrichtigung über eine neue Nachricht auf dem Handy an. Ist das Handy hingegen nicht auf stumm geschaltet, so überprüft der Messenger XYZ, ob eine Smartwatch verbunden ist. Ist keine Smartwatch mit dem Handy verbunden, so wird die komplette Nachricht auf dem Handy angezeigt. Sollte eine Smartwatch verbunden sein, so überprüft der Messenger, ob auf dieser ebenfalls der Messenger XYZ installiert ist. Falls ja, so wird die komplette Nachricht auf der Smartwatch angezeigt. Falls nein, so wird die komplette Nachricht auf dem Handy angezeigt.

Aufgabe 2 (UML-Klassendiagramm)

[37 PUNKTE]

Modellieren Sie die wie folgend spezifizierte Autovermietung mit Hilfe eines UML-Klassendiagramms.

In dieser Aufgabe soll der Fuhrpark einer Autovermietung modelliert werden.

Die Autovermietung besitzt eine Reihe von Fuhrparks. Ein Fuhrpark ist einer Stadt zugeordnet und besitzt eine Reihe von benachbarten Fuhrparks, von denen Fahrzeuge geliehen werden können.

Es gibt zwei Arten von Fahrzeugen: PKWs und Kombis. Jedes Fahrzeug ist gekennzeichnet durch einen Modelltyp und eine interne ID. Für Kombis ist zusätzlich die Größe des Frachtraums relevant. Ein Fuhrpark besitzt jedes Fahrzeug in einer bestimmten Anzahl.

Jedes Fahrzeug hat mehrere Mietpreise: Einen Jahrespreis, für den das Fahrzeug für ein Jahr geliehen werden kann; alternativ gibt es zwei Arten von Tagespreisen, bei denen jeweils eine bestimmte Rate pro Tag gezahlt werden muss. Es gibt zusätzlich zum Standardpreis, bei dem die normale Rate gezahlt werden muss, noch einen Frühbucherpreis, bei dem der Preis sich aus einem prozentualen Rabatt und der gleichen Rate wie beim Standardpreis berechnet.

Hinweis: Achten Sie auf das Vorhandensein von beschrifteten Assoziationen, Attributen und Methoden und nutzen Sie Vererbungsstrukturen, wenn dies angemessen erscheint. Sichtbarkeiten müssen nicht modelliert werden.

Fortsetzung nächste Seite!

Aufgabe 3 (Design-Patterns)**[20 PUNKTE]****Adapter****3.1 Einsatz [2 Punkte]**

Unter welchen Umständen wird das Adaptermuster eingesetzt?

3.2 Typen [2 Punkte]

Nennen Sie die beiden Typen des Adaptermusters und zeichnen Sie jeweils das Klassendiagramm. Geben Sie dabei explizit an, wie die adaptierte Methode aufgerufen wird.

Aufgabe 4 (Implementierung)**[18 PUNKTE]**

Gegeben sei folgender Java Code:

```
1 // does the main stuff
2 public class Class{
3     static int a,b;
4     static final int ONE = 1;
5
6     /**
7      * Computes
8      *
9      * @param x number
10     * @return b result
11     */
12     public static long compute(int x) {
13         if (x <= ONE) return x;
14         else return compute(x-1) + compute(x-2);
15     }
16
17     /**
18      * Computes for all
19      *
20      * @param a lower
21      * @param b upper
22     */
23     public static void something(int a, int b) {
24         for (int i = a; i <= b; i++)
25             System.out.println(i + ":_" + compute(i));
26     }
27
28     /**
29      * Performs the task.
30     */
31     public static void main(String[] args) {
32         a = Integer.parseInt(args[0]);
33         b = Integer.parseInt(args[0]);
34         something(a, b)
35     }
36 }
```

Fortsetzung nächste Seite!

4.1 Funktion [2 Punkte]

Benennen Sie, was ausgegeben wird, wenn die `main()`-Funktion dieser Klasse aufgerufen wird.

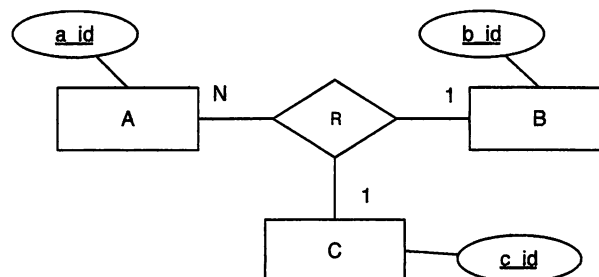
4.2 Verbesserungen [16 Punkte]

Benennen Sie *vier* Möglichkeiten, wie der obige Code hinsichtlich Stil und Lesbarkeit verbessert werden kann. Geben Sie jeweils an

- (i) warum dies im jetzigen Zustand suboptimal ist
- (ii) ein betroffenes Beispiel im Code
- (iii) wie dies behoben werden kann und
- (iv) wie die korrigierte Version aussieht.

Teilaufgabe II: Datenbanken**Aufgabe 1 (Grundwissen Datenbanksysteme)****[3+4+3+7 PUNKTE]**

1. Nennen Sie drei Gründe für den Einsatz eines Datenbanksystems. (**3 Stichworte**)
2. Nennen Sie die vier transaktionalen Garantien an Datenbanksysteme, die unter dem Namen ACID zusammengefasst werden. (**nur Stichpunkte, keine Erklärung nötig**)
3. Nennen Sie drei Vorteile, die man mit einem RAID-Verbund von Festplatten erzielen kann. (**nur Stichpunkte**)
- 4.a) Was ist der Unterschied zwischen einer Punktanfrage (Point Query) und einer Bereichsanfrage (Range Query)?
- b) Geben Sie je eine beispielhafte SQL Anfrage für eine Punkt- und eine Bereichsanfrage an. Benutzen Sie hierzu die Tabelle *elements* mit den Spalten *Key* und *Value*.
- c) Nennen Sie für beide Arten von Anfragen jeweils eine geeignete Indexstruktur, um die Zugriffe auf die Tabelle zu beschleunigen. Die beiden Indexstrukturen sollen möglichst effizient für die jeweilige Anfrageart sein. (**nur Namen nennen, keine Erklärung nötig**)

Aufgabe 2 (Relationale Modellierung)**[2+4+2 PUNKTE]**

Fortsetzung nächste Seite!

- a) Welche partiellen Funktionen gelten?
- b) Setzen Sie das ER-Modell in eine – oder falls nötig mehrere – Relation(en) um.
(Form: RelationName : Attribut 1, Attribut 2, ...).
Bestimmen Sie einen Schlüssel für die Beziehung R, so dass möglichst viele Einschränkungen aus dem ER-Modell auch in der Relation für die Beziehung modelliert werden.

Aufgabe 3 (Relationales Modell)

[6+6 PUNKTE]

Gegeben ist folgendes Universitätsschema (eine beispielhafte Ausprägung hängt der Klausur an):

Studenten : {[MatrNr : integer, Name : string, Semester : integer]}

Vorlesungen : {[VorlNr : integer, Titel : string, SWS : integer, gelesenVon : integer]}

Professoren : {[PersNr : integer, Name : string, Rang : string, Raum : integer]}

Assistenten : {[PersNr : integer, Name : string, Fachgebiet : string, Boss : integer]}

hören : {[MatrNr : integer, VorlNr : integer]}

voraussetzen : {[Vorgänger : integer, Nachfolger : integer]}

prüfen : {[MatrNr : integer, VorlNr : integer, PersNr : integer, Note : integer]}

1. Geben Sie alle Vorlesungen an (VorlNr und Titel zurückgeben), die der Student Xenokrates gehört hat. Formulieren Sie die Anfrage in **relationaler Algebra**.
2. Finden Sie alle Studenten (nur Name zurückgeben), die in mindestens einer Vorlesung durchgefallen sind (Note > 4.0). Formulieren Sie die Anfrage im **Tupelkalkül**.

Aufgabe 4 (Relationale Entwurfstheorie)

[2+2+7+3 PUNKTE]

Betrachten Sie ein abstraktes Relationenschema $\mathcal{R} = \{M, N, V, T, P, PN\}$ mit den FDs

- $M \rightarrow M$
- $M \rightarrow N$
- $V \rightarrow T, P, PN$
- $P \rightarrow PN$

1. Bestimmen Sie alle Kandidatenschlüssel.
2. In welcher Normalform befindet sich die Relation? (Keine Begründung nötig)
3. Bestimmen Sie zu den angegebenen FDs die kanonische Überdeckung.
4. Falls nötig, überführen Sie die Relation verlustfrei und abhängigkeitsbewahrend in die dritte Normalform mithilfe des Synthesealgorithmus.

Verwenden Sie die folgenden Schritte:

Fortsetzung nächste Seite!

- a) Linksreduktion
- b) Rechtsreduktion
- c) Zusammenfassen von FDs
- d) Entfernung von leeren Klauseln auf der rechten Seite

Geben Sie die Zwischenergebnisse nach jedem Schritt an.

Aufgabe 5 (SQL)

[2+7+7+7 PUNKTE]

Gegeben ist das folgende Universitätsschema (eine beispielhafte Ausprägung hängt der Klausur an):

Studenten : {[MatrNr : integer, Name : string, Semester : integer]}
Vorlesungen : {[VorlNr : integer, Titel : string, SWS : integer, gelesenVon : integer]}
Professoren : {[PersNr : integer, Name : string, Rang : string, Raum : integer]}
Assistenten : {[PersNr : integer, Name : string, Fachgebiet : string, Boss : integer]}
hören : {[MatrNr : integer, VorlNr : integer]}
voraussetzen : {[Vorgänger : integer, Nachfolger : integer]}
prüfen : {[MatrNr : integer, VorlNr : integer, PersNr : integer, Note : integer]}

Geben Sie ein SQL-92 Statement an, welches jeweils die gestellte Frage beantwortet.

1. Wie viele Vorlesungen gibt es? (Ausgabe der Anfrage soll die Anzahl der Vorlesungen sein.)
2. Geben Sie die Namen der Professoren an, die Xenokrates aus Vorlesungen kennt.
3. Welche Vorlesungen werden von Studenten im Grundstudium (1.-4. Semester) gehört? Geben Sie die Titel dieser Vorlesungen an.
4. Finden Sie alle Studenten, die drei oder mehr Vorlesungen hören.

Aufgabe 6 (Baum)

[16 PUNKTE]

Fügen Sie in einen anfänglich leeren B-Baum mit $k = 2$ die Zahlen 1 bis 20 in aufsteigender Reihenfolge ein. Geben Sie den Zustand des Baums nach jedem Einfügen an, bei dem ein neuer Knoten hinzukommt.

Beispielausprägung

Professoren				Studenten		
PersNr	Name	Rang	Raum	MatrNr	Name	Semester
2125	Sokrates	C4	226	24002	Xenokrates	18
2126	Russel	C4	232	25403	Jonas	12
2127	Kopernikus	C3	310	26120	Fichte	10
2133	Popper	C3	52	26830	Aristoxenos	8
2134	Augustinus	C3	309	27550	Schopenhauer	6
2136	Curie	C4	36	28106	Carnap	3
2137	Kant	C4	7	29120	Theophrastos	2
				29555	Feuerbach	2

Vorlesungen				voraussetzen	
VorlNr	Titel	SWS	gelesen Von	Vorgänger	Nachfolger
5001	Grundzüge	4	2137	5001	5041
5041	Ethik	4	2125	5001	5043
5043	Erkenntnistheorie	3	2126	5001	5049
5049	Mäeutik	2	2125	5041	5216
4052	Logik	4	2125	5043	5052
5052	Wissenschaftstheorie	3	2126	5041	5052
5216	Bioethik	2	2126	5052	5259
5259	Der Wiener Kreis	2	2133		
5022	Glaube und Wissen	2	2134		
4630	Die 3 Kritiken	4	2137		

hören		Assistenten			
MatrNr	VorlNr	PersNr	Name	Fachgebiet	Boss
26120	5001	3002	Platon	Ideenlehre	2125
27550	5001	3003	Aristoteles	Syllogistik	2125
27550	4052	3004	Wittgenstein	Sprachtheorie	2126
28106	5041	3005	Rhetikus	Planetenbewegung	2127
28106	5052	3006	Newton	Keplersche Gesetze	2127
28106	5216	3007	Spinoza	Gott und Natur	2134
28106	5259				
29120	5001				
29120	5041				
29120	5049				
29555	5022				
25403	5022				
29555	5001				

prüfen			
MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2
25403	4630	2137	5

Abbildung 1: Beispielausprägung für eine Universitäts-Datenbank

Thema Nr. 2
(Aufabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1

[7 PUNKTE]

Bewerten Sie die Korrektheit der folgenden Aussagen und begründen Sie Ihre Antwort kurz.

1. Ein gut entworfenen und implementiertes System braucht kein Testing.
2. Nachdem eine Software ausgeliefert wurde, muss sie nicht mehr angepasst werden und ihr Lebenszyklus ist abgeschlossen.
3. Das Wasserfallmodell ist sinnvoller als das Spiralmodell.

Aufgabe 2

[10 PUNKTE]

1. In welchem der beiden folgenden Entwicklungsprozesse wird der Kunde stärker eingebunden: Im *Wasserfallmodell* oder bei *eXtreme Programming*? Begründen Sie Ihre Antwort.
2. Nennen und beschreiben Sie drei Vorteile einer inkrementellen Produktauslieferung.

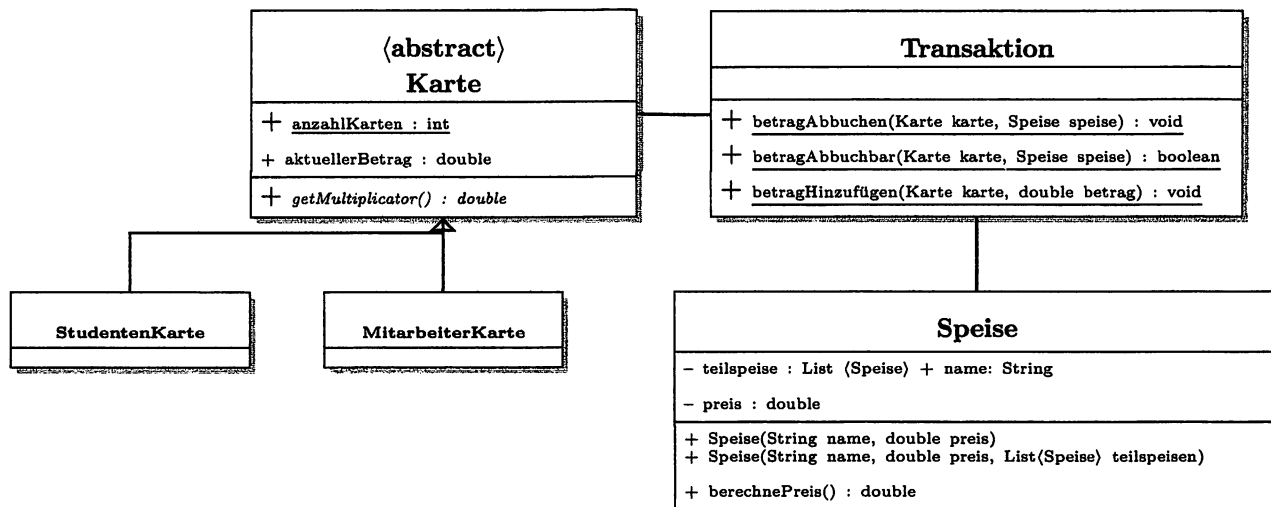
Aufgabe 3

[30 PUNKTE]

Zur Modellierung der Mensa ist bereits eine UML-Modellierung vorhanden, die unten gegeben ist. Fertigen Sie dazu eine Implementierung in einer objektorientierten Programmiersprache, beispielsweise Java, an. Beachten Sie bei der Implementierung folgende Vorgaben:

1. Jeder Kunde der Mensa besitzt eine Mensakarte (kurz Karte).
2. Jede Karte besitzt ihren aktuellen Kontostand und es ist möglich, mittels Transaktionen Beträge abzubuchen oder hinzuzufügen. Außerdem kann so auch überprüft werden, ob ein Betrag überhaupt abbuchbar ist.
3. Es wird zwischen Studentenkarten und Mitarbeiterkarten unterschieden, wobei Mitarbeiter das 1,2-Fache der Studentenpreise zahlen.
4. Die Mensa überwacht die Anzahl der ausgegebenen Karten.
5. Eine Speise kann zusätzlich weitere Teilspeisen (z. B. mehrere Beilagen) enthalten. Der Preis einer Speise berechnet sich aus dem Preis der Speise und der Summe der Preise der zusätzlichen Teilspeisen. Hinweis: Eine Beilage kann aber auch als einzelne Speise verkauft werden.

Fortsetzung nächste Seite!

**Aufgabe 4**

[21 PUNKTE]

Entwerfen Sie ein UML-Klassendiagramm zur folgenden Anforderungsbeschreibung. Verwenden Sie bei der Modellierung des Systems ein passendes Design-Pattern. Nennen Sie das verwendete Design-Pattern.

Es soll ein Sensornetzwerk modelliert werden, in das einzelne Sensoren, aber auch Sensorstationen eingebunden werden können. Für eine Sensorstation soll im System der Name, die Position und die Liste aller zur Station gehörigen Sensoren gespeichert werden, die mittels Get-/Set-Methoden abgerufen und gesetzt werden können. Über die Sensorstation können auch einzelne Sensoren abgerufen oder hinzugefügt werden. Ein Sensor wiederum besitzt einen Namen, seine Position und die Kenntnis, in welcher Station er angebracht ist (mit jeweils dazugehörigen Get-/Set-Methoden).

Für die Speicherung der von den Sensoren erhobenen Daten gibt es zwei Arten von Loggern, die die eingehenden Daten der Sensorstation oder Sensoren in eine Datei bzw. auf die Konsole ausgeben können. Die Logger können sich bei einzelnen Sensorstationen und Sensoren registrieren oder auch diese Registrierung wieder entfernen.

Hinweise:

- Die Position einer Station oder eines Sensors kann als String gespeichert werden.
- Achten Sie bei der Modellierung auf sinnvolle Kapselung, Vererbungsbeziehungen und Relationen zwischen den verschiedenen Klassen.

Fortsetzung nächste Seite!

Aufgabe 5**[11 PUNKTE]**

1. Ein Softwaresystem wurde unter der Verwendung des MVC-Patterns erstellt. Zur Vereinfachung heißen die relevanten Komponenten in der Aufgabenstellung: **Model**, **View** und **Controller**. Erstellen Sie ein UML-Sequenzdiagramm, das folgendes Szenario abbildet:
 - Ein Nutzer füllt das Registrierungsformular der Anwendung aus, um sich im System zu registrieren.
 - Diese Eingaben werden direkt im **View** validiert, bevor sie an den **Controller** übergeben werden.
 - Der **Controller** gibt die Informationen an das **Model** weiter, das dafür sorgt, dass ein neues Nutzerobjekt erstellt wird, die eingegebenen Daten darin gespeichert werden sowie das Nutzerobjekt in der **Datenbank** gespeichert wird.

Gehen Sie davon aus, dass alle Aktionen erfolgreich durchgeführt werden können.

Aufgabe 6**[11 PUNKTE]**

1. Erläutern Sie die Unterschiede zwischen zentraler und dezentraler Versionsverwaltung und beschreiben Sie beide Arten der Versionsverwaltung.
2. Was versteht man unter Black-Box-Testing und White-Box-Testing? Beschreiben Sie beide Testing-Arten und gehen Sie auf die Unterschiede ein. Berücksichtigen Sie hierbei, dass die Tests ohne Zugabe weiterer Informationen eingesetzt werden.
3. Erklären Sie den Unterschied zwischen Verifikation und Validierung.

Teilaufgabe II: Datenbanken**Aufgabe 1 (ER-Modellierung)****[15+15 PUNKTE]**

***Hinweis:** Bei Wahl dieser Aufgabe wird Wissen über das erweiterte Entity-Relationship-Modell (beispielsweise schwache Entity-Typen, Vererbung) sowie die Verfeinerung eines relationalen Schemas vorausgesetzt.*

Gegeben seien folgende Informationen:

- Eine Messe findet in einem Messezentrum statt. Dabei wird eine Messe über die Kombination aus Jahr und Name identifiziert, während ein Messezentrum über den Ort identifiziert wird und zudem einen Namen hat.
- Ein Messezentrum besteht aus Gebäuden. Diese Gebäude haben eine Nummer, die nur jeweils im Zusammenhang mit dem Messezentrum eindeutig ist. Weiter hat ein Gebäude Räume, die über eine Raumnummer auch nur innerhalb des Gebäudes eindeutig identifizierbar sind. Zudem soll für einen Raum gespeichert werden, wie viele Personen darin Platz finden.
- Räume lassen sich unter anderem in Büros und Ausstellungshallen unterscheiden, wobei ein Raum nicht beides sein kann. Für eine Ausstellungshalle wird zudem ihre Fläche in Quadratmetern gespeichert.

Fortsetzung nächste Seite!

- Angestellte arbeiten für eine Firma und werden über eine Personalnummer identifiziert. Eine Firma hat eine eindeutige FirmenID und zudem eine Adresse.
 - Auf einer Messe sind Angestellte eingeteilt. Außerdem kann eine Messe einen Hauptraum haben.
 - Eine Firma stellt innerhalb einer Ausstellungshalle für eine Messe aus. Dabei wird auch die Standnummer gespeichert.
- a) Erstellen Sie für das oben gegebene Szenario ein geeignetes ER-Diagramm. Verwenden Sie dabei – wenn angebracht – das Prinzip der Spezialisierung. Kennzeichnen Sie die Primärschlüssel der Entity-Typen, totale Teilnahmen (existenzabhängige Beziehungen) und schwache Entity-Typen. Zeichnen Sie die Funktionalitäten der Relationship-Typen in das Diagramm ein.
- b) Überführen Sie Ihr in Aufgabe a) erstelltes Modell in ein verfeinertes relationales Schema. Kennzeichnen Sie die Schlüssel durch Unterstreichen. Datentypen müssen nicht angegeben werden. Die einzelnen Schritte müssen angegeben werden.

Aufgabe 2 (SQL)

[5+6+4+7+8+5 PUNKTE]

Gegeben sei der folgende Ausschnitt des Schemas eines Schulverwaltungssystems:

<pre>Familie : {[<u>ID</u> : INTEGER, Name : VARCHAR(255)]}</pre>	<pre>Bildungseinrichtung : {[<u>ID</u> : INTEGER, Name : VARCHAR(255), Ort : VARCHAR(255), Typ: VARCHAR(255)]}</pre>
<pre>besucht : {[<u>KindID</u> : INTEGER, <u>BildungseinrichtungID</u> : INTEGER, <u>von</u> : DATE, <u>bis</u> : DATE]}</pre>	<pre>Kind : {[<u>ID</u> : INTEGER, Name : VARCHAR(255), gehört_zu : INTEGER]}</pre>

Die Tabelle *Kind* enthält Informationen über Kinder, wobei die Spalte *gehört_zu* auf die Familie referenziert. *Bildungseinrichtung* beschreibt verschiedene Einrichtungen anhand ihrer ID, des Namens, des Orts und des Typs der Einrichtung. Die Tabelle *besucht* gibt an, welche Kinder zu welchem Zeitpunkt welche Einrichtung besuchen oder besucht haben. *Familie* enthält Informationen über die Familien bezüglich deren ID und Namen.

Beachten Sie bei der Formulierung der SQL-Anweisungen, dass die Ergebnisrelationen keine Duplikate enthalten dürfen. Sie dürfen geeignete Views definieren.

Fortsetzung nächste Seite!

1. Schreiben Sie eine SQL-Anweisung, welche alle Einträge in der Tabelle *Bildungseinrichtung* löscht, die von keinem Kind besucht werden (weder aktuell noch früher).
2. Schreiben Sie eine SQL-Anweisung, welche die Namen aller Familien bestimmt, bei denen mindestens ein Kind eine Bildungseinrichtung vom Typ 'Gymnasium' besucht hat oder besucht.
3. Schreiben Sie die SQL-Anweisung aus Teilaufgabe b) um, sodass nun die Namen aller Familien bestimmt werden, bei denen mindestens zwei Kinder eine Bildungseinrichtung vom Typ 'Gymnasium' besucht haben oder besuchen.
4. Schreiben Sie eine SQL-Anweisung, welche die IDs und Namen aller Kinder bestimmt, die vor dem Besuch eines Gymnasiums auch eine Mittelschule besucht haben.
5. Bestimmen Sie für die folgenden SQL-Anweisungen die **minimale** und **maximale** Anzahl an Tupeln im Ergebnis. Beziehen Sie sich dabei auf die Größe der einzelnen Tabellen.

Verwenden Sie für die Lösung folgende Notation:

— Familie — beschreibt die Größe der Tabelle *Familie*

a)

```
SELECT *  
FROM Kind  
EXCEPT  
SELECT k.*  
FROM Kind k  
LEFT OUTER JOIN besucht b ON k.id = b.kindid
```

b)

```
SELECT bi.ort  
FROM Bildungseinrichtung bi, besucht b, Kind k  
GROUP BY bi.ort
```

6. Beschreiben Sie den Effekt der folgenden SQL-Query in natürlicher Sprache:

```
SELECT k.gehört_zu, COUNT(DISTINCT bi.id)  
FROM besucht b, Kind k, Bildungseinrichtung bi  
WHERE k.id = b.kindid  
AND b.bildungseinrichtungid = bi.id  
GROUP BY k.gehört_zu
```

Aufgabe 3 (Entwurfstheorie)**[6+12+7 PUNKTE]**

Gegeben sei folgendes relationales Schema R in erster Normalform:

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

und die Zerlegung $\rho = \{R_1, R_2\}$ von R mit $R_1 = \{A, C, E\}$ und $R_2 = \{B, C, D, E\}$.

Für R gelte folgende Menge FD funktionaler Abhängigkeiten:

$FD = \{$

$$\begin{aligned} ACD &\rightarrow B, \\ CE &\rightarrow AD, \\ CD &\rightarrow BE, \\ ABE &\rightarrow E \end{aligned}$$

$\}$

1. Bestimmen Sie alle Kandidatenschlüssel/Schlüsselkandidaten von R mit FD .
Begründen Sie Ihre Antwort. Begründen Sie zudem, warum es keine weiteren Kandidatenschlüssel/Schlüsselkandidaten gibt.
Hinweis: Die Angabe von Attributmengen, die keine Kandidatenschlüssel sind, führt zu Abzügen.
2. Prüfen Sie, ob R mit FD in
 - BCNF,
 - 3NF oder
 - 2NF ist.
3. Zeigen oder widerlegen Sie, dass die Zerlegung ρ von R verlustfrei bzgl. FD ist.