

---

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

---

Kennzahl: \_\_\_\_\_

Kennwort: \_\_\_\_\_

Arbeitsplatz-Nr.: \_\_\_\_\_

---

**Herbst  
2021**

**66116**

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen  
— Prüfungsaufgaben —**

---

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbanksysteme, Softwaretechnologie**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 20

---

Bitte wenden!

Thema Nr. 1  
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!  
Alle Lösungsschritte sind sorgfältig zu begründen!

**Teilaufgabe I: Datenbanksysteme**

**Aufgabe 1 (Datenmodellierung mit dem ER-Modell)**

[34 PUNKTE]

Wir betrachten eine Datenbank, die Informationen einer Klinik abbildet.

- Eine Klinik besitzt einen Namen, ein Baujahr, eine eindeutige ID und ist in einer Stadt ansässig.
  - Eine Klinik besteht aus mindestens zwei Stationen. Eine Station hat einen Namen, einen Stationsleiter sowie eine eindeutige Stationsnummer und ist genau einer Klinik zugeordnet.
  - Weiterhin sind einer Station mehrere Ärzte zugeordnet. Ein Arzt hat einen Namen, ein Fachgebiet und eine eindeutige Sozialversicherungsnummer (kurz: SVNr). Ein Arzt kann mehreren Stationen zugeordnet werden.
  - Die Klinik dient der Versorgung von Patienten. Patienten können dabei auf mehreren Stationen behandelt werden. Jedem Patienten wird ein zuständiger Arzt zugewiesen. Ein Arzt kann für mehrere Patienten verantwortlich sein. Auf einer Station können mehrere Patienten behandelt werden.
  - Ein Patient besitzt einen Namen, eine eindeutige Patientennummer und ein Geburtsdatum.
  - Es soll davon ausgegangen werden, dass ein Patient auf jeder seiner Stationen wegen jeweils einer Krankheit behandelt wird, welche als Zahlencode dargestellt wird.
- a) Modellieren Sie die oben beschriebene Situation möglichst vollständig in einem erweiterten ER-Diagramm. Kennzeichnen Sie die Schlüssel der Entity-Typen durch Unterstreichen und geben Sie für die Relationship-Typen jeweils Funktionalitäten in Chen-Notation oder in (MIN, MAX)-Notation an.
- b) Übertragen Sie Ihr ER-Modell in das relationale Datenmodell. Erstellen Sie dazu die Tabellen **Klinik**, **Patient** und **Behandelt** (Patient wird in Station behandelt) mit Hilfe von CREATE TABLE-Statements in SQL. Berücksichtigen Sie für die Tabellen **Patient** und **Behandelt** die Fremdschlüsselbeziehungen, und geben Sie für die Tabelle **Patient** explizit sinnvolle Update- und Delete-Modi über ON UPDATE bzw. ON DELETE an. Wählen Sie sinnvolle Datentypen.
- c) Was versteht man unter Redundanzen? Wann können diese nützlich sein?
- d) Nennen Sie jeweils einen Vor- und Nachteil eines relationalen Datenbanksystems.

Fortsetzung nächste Seite!

**Aufgabe 2 (SQL und relationale Algebra)****[32 PUNKTE]**

Wir betrachten eine Datenbank zur Speicherung von Flugdaten. Die Datenbank enthält die Tabellen Flughafen, Flugzeug und Flug, deren Inhalt im Folgenden in Auszügen angegeben ist:

Flughafen		
Kürzel	Name	Terminalanzahl
ATL	Atlanta	7
LAX	Los Angeles	6
FRA	Frankfurt	2
DFW	Dallas/Fort Worth	5
DXB	Dubai	3
CDG	Paris	3
LHR	London	5

Flugzeug	
Modell_Nr	Plätze
A380	555
A220	115
A330	406

Flug					
Flug_Nr	Modell_Nr	Startflughafen	Zielflughafen	Abhebezeit	Landezeit
3015	A380	LAX	DFW	08:22	13:32
3015	A380	DFW	ATL	16:33	19:32
3015	A380	ATL	CDG	21:36	11:45
1508	A220	DXB	FRA	07:12	12:44
1508	A220	FRA	LHR	14:50	16:16
1508	A220	LHR	CDG	18:50	21:14
2303	A220	DFW	FRA	08:52	16:34

Fortsetzung nächste Seite!

- a) Geben Sie das Ergebnis des folgenden relationalen Ausdrucks angewandt auf die oben dargestellten Auszüge der Datenbank an:

$$T = \Pi_{\text{Flug\_Nr, Dep, Arr}} \left( \sigma_{\text{Modell\_Nr} = 'A380'} \left( \rho_{(\text{Abhebezeit} \rightarrow \text{Dep, Landezeit} \rightarrow \text{Arr})} (\text{Flug}) \right) \right).$$

$\rho$  bezeichnet die Variablenumbenennung.

- b) Geben Sie das Resultat der folgenden SQL-Anweisung angewandt auf die oben dargestellten Auszüge der Datenbank an:

```
SELECT Flug_Nr, COUNT(*) AS Stops
FROM Flug
GROUP BY Flug_Nr
HAVING Stops <> 1;
```

- c) Schreiben Sie eine SQL-Anweisung, die die Anzahl der Terminals jedes Flughafens um 1 verringert.
- d) Schreiben Sie eine SQL-Anweisung, die eine View erzeugt, die die Namen der Flughäfen enthält, die von einem Flugzeug angefliegen werden, dessen Modell.Nr mit *A3* beginnt. Angefliegen bedeutet hier starten oder landen.
- e) Schreiben Sie eine SQL-Anweisung, die den Flug mit der Flug\_Nr 2503 für die Modellnummer *A330* von *Atlanta* nach *Dubai* (Abflug 10:30, Ankunft 20:15 Uhr) einfügt.
- f) Schreiben Sie eine SQL-Anweisung, die alle Flüge mit Zielflughafen in Atlanta löscht.
- g) Schreiben Sie eine SQL-Anweisung, die die Namen und die Terminalanzahlen aller Flughäfen zurückliefert, deren Terminalanzahl größer ist als die durchschnittliche Terminalanzahl aller Flughäfen. Die Flughäfen sollen absteigend nach ihrer Terminalanzahl ausgegeben werden, bei Gleichheit alphabetisch sortiert.
- h) Schreiben Sie eine SQL-Anweisung, die die Modellnummern all jener Flugzeuge zurückliefert, die im Laufe eines Gesamtfluges denselben Flughafen mehrfach anfliegen. Geben Sie jedes Paar von Modellnummer und Flughafenkürzel nur einmal aus. Ein Gesamtflug setzt sich aus allen Teilflügen mit der gleichen Flug\_Nr zusammen.

Fortsetzung nächste Seite!

**Aufgabe 3 (Funktionale Abhängigkeiten und Zerlegungen)****[31 PUNKTE]**

Gegeben sei das Relationenschema  $R = (U, F)$  mit der Attributmenge  $U$  und der Menge  $F$  von funktionalen Abhängigkeiten:

$$U = \{A, B, C, D, E, G\},$$

$$F = \{ \{B\} \rightarrow \{E, G\}, \{B, G\} \rightarrow \{A, E\}, \{D\} \rightarrow \{A, C\},$$

$$\{A, C\} \rightarrow \{D, E, G\}, \{C, D\} \rightarrow \{E\} \}.$$

- Zeigen Sie, dass die funktionale Anhängigkeit  $\{C, D\} \rightarrow \{A, G\}$  bereits aus  $F$  folgt.
- Geben Sie alle Kandidaten-Schlüssel für das Relationenschema  $R$  sowie die nicht-primen Attribute an (mit Begründung).
- Ist  $R$  in 3NF, ist  $R$  in BCNF? Begründen Sie jeweils Ihre Antwort.
- Geben Sie eine Basis (kanonische Überdeckung)  $G$  von  $F$  an und führen Sie damit die 3NF-Synthese aus.

Fortsetzung nächste Seite!

**Aufgabe 4 (Transaktionen)****[23 PUNKTE]**

a) Gegeben sei der folgende Schedule.

$T_1$	$T_2$	$T_3$
		$read(Z)$
	$read(Y)$	
$read(X)$		
		$write(Z)$
$write(X)$		
		$write(X)$
	$read(Z)$	
	$write(X)$	
	$write(Z)$	

Alle Transaktionen sollen am Ende erfolgreich beendet werden (commit).

Prüfen Sie den Schedule auf konfliktbasierte Serialisierbarkeit. Geben Sie eine Begründung und alle möglichen Serialisierungsreihenfolgen an.

- b) Beschreiben Sie das Problem des *Lost Update*. Geben Sie ein Beispiel hierfür an.
- c) Beschreiben Sie das Zwei-Phasen-Sperrprotokoll. Benennen Sie hierfür die beiden Phasen und geben Sie ein Beispiel für einen Schedule an, der nicht dem Zwei-Phasen-Sperrprotokoll entspricht und welcher aus maximal zwei Transaktionen und zwei Variablen besteht.
- d) Welche Vorgehensweise in Verbindung mit dem Zwei-Phasen-Sperrprotokoll benötigt man, um Deadlocks zu vermeiden?

Fortsetzung nächste Seite!

## Teilaufgabe II: Softwaretechnologie

### Aufgabe 1 (UML-Klassendiagramm)

[30 PUNKTE]

Wir betrachten in dieser Aufgabe das Szenario *Online-Shop*. Erstellen Sie hierfür ein **UML-Klassendiagramm** entsprechend der folgenden Beschreibung. Setzen Sie insbesondere Klassen (mit Vererbungsbeziehungen), Assoziationen (mit Namen und Leserichtungen), Attribute, Operationen und Rollennamen ein. Verwenden Sie dabei Kompositionsbeziehungen an geeigneten Stellen und wählen Sie sinnvolle Multiplizitäten. Legen Sie zudem entsprechende Sichtbarkeiten fest; wählen Sie öffentliche Sichtbarkeit, wenn keine Sichtbarkeit explizit genannt wird. Setzen Sie für Attribute geeignete Datentypen.

Jeder Händler hat einen Namen und bietet ein Sortiment von mindestens zehn Artikeln an. Jeder Artikel hat eine spezifische Artikelnummer (Ganzzahl), welche für alle Artikel fortlaufend vergeben wird, beginnend bei 1 als Startwert. Jeder Artikel wird von genau einem Händler angeboten. Mittels einer Operation des Händlers kann für die gegebene Artikelnummer der Artikel gesucht werden. Zusätzlich werden für jeden Artikel eine textuelle Beschreibung, der Preis als Gleitpunktzahl sowie optional eine Bewertung als ganze Zahl zwischen 0 und 5 spezifiziert.

Jeder Kunde hat mindestens einen Vornamen und genau einen Nachnamen. Premium-Kunden sind spezielle Kunden, für welche eine privat sichtbare Bonuspunktzahl hinterlegt ist. Für jeden Kunden ist zudem eine nur in der Klasse selbst und in Unterklassen sichtbare Rechnungsadresse persistiert. Kunden tätigen Bestellungen, wobei jede Bestellung genau einem Kunden zugeordnet ist. Für jede Bestellung ist ihr Status sowie die privat sichtbare Lieferadresse hinterlegt. Daraus leitet sich für jeden Kunden die Anzahl der getätigten Käufe ab. Eine Bestellung umfasst mehrere Artikel, die jeweils mit einer Angabe der Anzahl referenziert werden. Jede Adresse enthält eine Straße, eine Hausnummer (zur Vereinfachung eine Ganzzahl), eine Postleitzahl sowie einen Ort; es werden also nur Inlandssendungen berücksichtigt.

Jeder Händler kann eine Menge an Paketdiensten beauftragen, wobei Paketdienste für mehrere Händler liefern dürfen. Jeder Paketdienst hat einen Namen. Je nach Status sind Bestellungen entweder dem jeweiligen Händler (Status *im Warenkorb* oder *gekauft*) oder dem Paketdienst (Status *versendet* oder *empfangen*) zur Verwaltung zugeordnet und von diesem existentiell abhängig.

Fortsetzung nächste Seite!

**Aufgabe 2 (White-Box-Testverfahren)****[30 PUNKTE]**

Ihnen ist folgende Methode (in Java) gegeben. Diese Methode prüft, ob die übergebene **Zeichenkette** aus mindestens zwei Zeichen besteht und mindestens eine Ziffer und mindestens einen kleingeschriebenen Buchstaben beinhaltet. Als Eingabe sind alle möglichen Zeichenketten erlaubt.

```
1 public static boolean checkPassword(String pw) {
2     boolean result = true;
3     if (pw == null || pw.length() < 2) {
4         result = false;
5     } else {
6         boolean numbersUsed = false;
7         boolean charactersUsed = false;
8         for (int i = 0; i < pw.length(); i++) {
9             char c = pw.charAt(i);
10            if (c >= '0' && c <= '9') {
11                numbersUsed = true;
12            } else if (c >= 'a' && c <= 'z') {
13                charactersUsed = true;
14            }
15        }
16        result = numbersUsed && charactersUsed;
17    }
18    return result;
19 }
```

- Geben Sie für die Methode einen **Kontrollflussgraphen** an, wobei Sie die Knoten mit den jeweiligen Zeilen des Programmcodes beschriften.
- Geben Sie eine **minimale Testmenge** an, die das Kriterium der *Zweigüberdeckung* (*Kantenüberdeckung*) erfüllt. Geben Sie für jeden Testfall den Kontrollfluss als Pfad in der Notation  $1 \rightarrow 2 \rightarrow \dots$  an. Welches Testkriterium wird dabei ebenfalls auf jeden Fall erfüllt?
- Geben Sie die Größe (Mächtigkeit) der minimalen Testmenge an, die benötigt wird, um das Kriterium der *Boundary-Interior-Pfadüberdeckung* zu erfüllen und erläutern Sie Ihre Antwort. Geben Sie darüber hinaus zwei Testfälle aus der Testmenge an, die verschiedene Pfade abdecken. Geben Sie für jeden Testfall den Kontrollfluss als Pfad in der Notation  $1 \rightarrow 2 \rightarrow \dots$  an.

**Hinweis:** Eine *Testmenge* ist *minimal*, wenn es keine Testmenge mit einer kleineren Zahl von Testfällen gibt. Die Minimalität muss nicht bewiesen werden.

Fortsetzung nächste Seite!



**Aufgabe 3 (Projektmanagement)****[30 PUNKTE]**

Gegeben sei folgende Tabelle zur groben Planung eines hypothetischen Softwareprojekts:

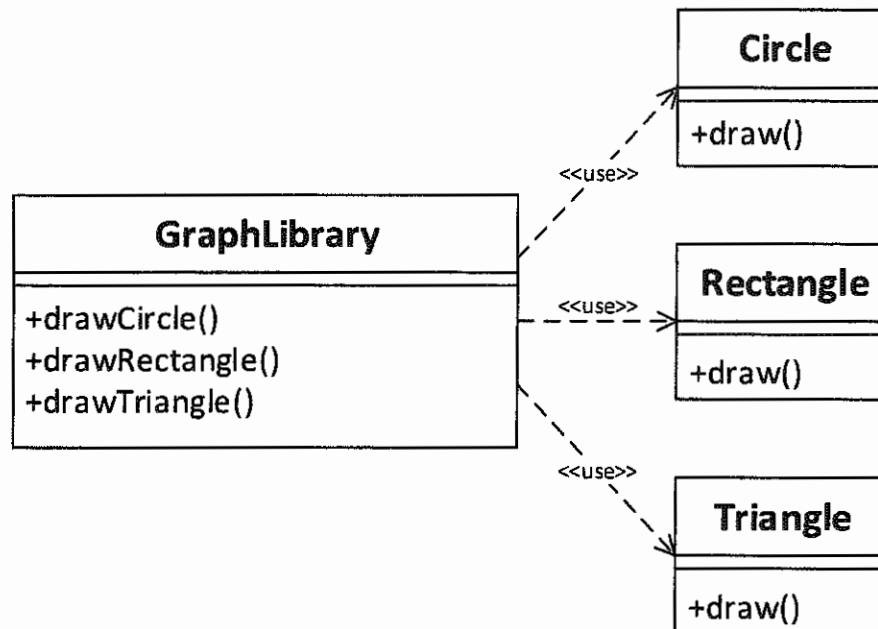
Aktivität	Dauer [Monate]	Einschränkungen
<i>Entwurf</i>	3	–
<i>Implementierung</i>	4	Startet frühestens nach Ende der Entwurfsphase.
<i>Testen</i>	5	Startet mit Beginn der Implementierung. Endet 1 Monat nach Ende der Implementierungsphase.

- Welcher essentielle Schritt muss der Entwurfsphase im klassischen Projektmanagement vorausgehen?
- Erstellen Sie ein **Gantt-Diagramm**, das die Aktivitäten und Abhängigkeiten des obigen Projektplans beschreibt.
- Erstellen Sie nun ein **CPM Netzwerk**, das die Aktivitäten und Abhängigkeiten des obigen Projektplans beschreibt. Fügen Sie dazu einen Startknoten ein, von dem der Beginn aller Aktivitäten abhängt, sowie einen Endknoten, der vom Ende aller oben aufgeführten Aktivitäten abhängt.
- Berechnen Sie anhand des CPM-Netzwerks für jedes Ereignis die **früheste Zeit**, die **späteste Zeit** und die **Pufferzeit**. Das Projekt startet zum Zeitpunkt 0.
- Geben Sie einen **kritischen Pfad** durch das CPM-Netzwerk an. Was folgern Sie für das Projektende für den Fall, dass sich eine Aktivität verzögert?
- Nennen Sie **drei** agile Projektorganisationsformen.

Fortsetzung nächste Seite!

**Aufgabe 4 (Entwurfsmuster)****[30 PUNKTE]**

Gegeben ist folgendes **UML-Klassendiagramm**, welches eine grafische Bibliothek darstellt.



Im Folgenden soll ein abgewandeltes Factory-Entwurfsmuster angewandt und ein neues UML-Klassendiagramm ausgehend von diesem Muster erstellt werden.

- Wenden Sie das Entwurfsmuster **Factory** an und erstellen Sie ein neues **UML-Klassendiagramm** ausgehend von diesem Muster. Erweitern Sie dazu auf Ihrem Bearbeitungsbogen das Klassendiagramm um die Klasse **Client**, welche ein Attribut **Name** besitzt und die Klasse **GraphLibrary** benutzt. Die Klasse **GraphLibrary** soll nur eine `drawShape()` Methode enthalten, der ein Figurtyp als **String** übergeben wird. Des Weiteren soll diese Klasse eine **ShapeFactory** enthalten. Die **ShapeFactory** soll in der Lage sein, über eine Methode die jeweiligen Figuren instanziierten zu können.
- Erstellen Sie ein **Sequenzdiagramm** für folgendes Szenario: Der **Client Bob** ruft die Methode zum Zeichnen einer Figur vom Typ **Circle** der Klasse **GraphLibrary** auf. Daraufhin wird in der **GraphLibrary** das entsprechende **Factory**-Objekt erzeugt. Auf der erzeugten **Factory** wird die Methode zum Erzeugen einer Figur vom Typ **Circle** aufgerufen. Letztendlich wird die Methode `draw()` der erzeugten Figur von der Klasse **GraphLibrary** aufgerufen.
- Geben Sie den **Quelltext** in einer objektorientierten Programmiersprache Ihrer Wahl für die Implementierung der Klassen **ShapeFactory** und **GraphLibrary** an. Dabei muss die in Teilaufgabe **b** beschriebene Funktionalität gewährleistet werden. Die Klasse **ShapeFactory** soll in der Lage sein, Figuren vom Typ **Circle**, **Rectangle** und **Triangle** zu erzeugen. Im Falle von anderen Typen wird **null** zurückgegeben. Geben Sie eine Fehlermeldung auf der Konsole aus, falls ein nicht unterstützter Figurtyp angegeben wird.

Thema Nr. 2  
(Aufabengruppe)

Es sind alle Aufgaben dieser Aufabengruppe zu bearbeiten!  
Alle Lösungsschritte sind sorgfältig zu begründen!

**Teilaufgabe I: Datenbanksysteme**

**Aufgabe 1 (Vermischte Fragen)**

[12 PUNKTE]

- a) Was versteht man unter referentieller Integrität? Erläutern Sie in ein bis zwei Sätzen.
- b) Was ist der Unterschied zwischen einem INNER JOIN und einem LEFT OUTER JOIN? Erläutern Sie in ca. zwei Sätzen.
- c) Welche Auswirkung hat die Verwendung von ON UPDATE CASCADE bei einem Fremdschlüsselattribut? Erläutern sie in ein bis zwei Sätzen.
- d) Warum kann eine abgeschlossene (*committed*) Transaktion nicht wieder zurückgesetzt werden? Begründen sie in ein bis zwei Sätzen.
- e) Was versteht man unter Datenunabhängigkeit? Erklären Sie in ein bis zwei Sätzen.
- f) Was sind Trigger und wofür werden diese verwendet? Erläutern Sie in ein bis zwei Sätzen.

**Aufgabe 2 (ER-Modellierung)**

[28 PUNKTE]

Im Folgenden finden Sie die Beschreibung einer Fahrradvermietung. Erstellen Sie zu dieser Beschreibung ein erweitertes ER-Diagramm. Kennzeichnen Sie die Primärschlüssel durch Unterstreichen und geben Sie die Kardinalitäten in Chen-Notation (= Funktionalitäten) an. Kennzeichnen Sie auch die totale Teilnahme von Entity-Typen an Beziehungstypen.

Es gibt Fahrräder. Diese haben eine eindeutige Rahmennummer, eine Farbe sowie eine Modellbezeichnung. Außerdem gibt es Mieter, die über eine eindeutige Mieternummer identifiziert werden können. Ein Mieter kann entweder eine Person oder eine Firma sein. Jede Person hat einen Nachnamen und einen Vornamen, ein Geburtsdatum und eine Handynummer, mit der diese eindeutig identifiziert werden kann. Jede Firma hat einen eindeutigen Namen und einen Hauptsitz. Ein Mieter kann ein Fahrrad über eine bestimmte Zeit mieten, wobei die insgesamt fällige Höhe der zu leistenden Mietzahlungen gespeichert wird und genau ein Zahlungsmittel zum Einsatz kommt. Der Mieter kann sich für jedes Mietverhältnis zwischen der Zahlung per-Kreditkarte oder Lastschrift entscheiden. Für das Lastschriftverfahren wird die international eindeutige IBAN und der Name des Kontoinhabers erfasst, bei Zahlung mit Kreditkarte der Name des Karteninhabers, die eindeutige Kartenummer, das Ablaufdatum sowie der Kartenanbieter, der automatisch aus der Kartenummer hergeleitet werden kann.

Fortsetzung nächste Seite!

**Aufgabe 3 (Relationaler Entwurf)****[20 PUNKTE]**

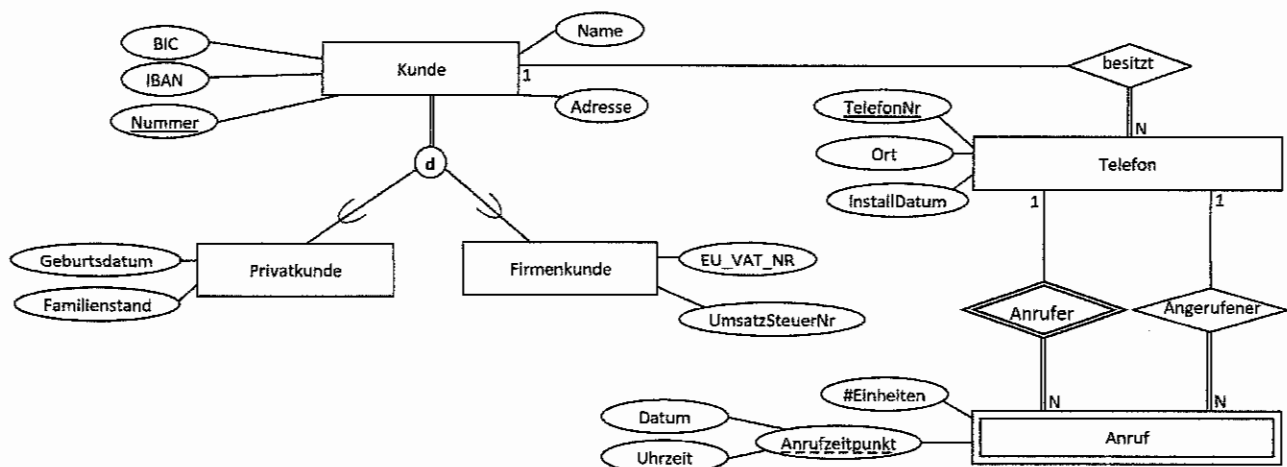
Entwerfen Sie zum untenstehenden ER-Diagramm ein verfeinertes Relationenschema mit möglichst wenigen Relationen.

Verwenden Sie dabei folgende Notation: Primärschlüssel werden durch Unterstreichen gekennzeichnet, Fremdschlüssel durch die Nennung der Relation, auf die sie verweisen, in eckigen Klammern hinter dem Fremdschlüsselattribut. Attribute zusammengesetzter Fremdschlüssel werden durch runde Klammern als zusammengehörig markiert. Wenn ein Attribut zur korrekten Abbildung des ER-Diagramms als UNIQUE oder NOT NULL ausgezeichnet werden muss, geben Sie dies an.

Beispiel:

```

Relation1 (Primärschlüssel, Attribut1, Attribut2,
  Fremdschlüsselattribut1[Relation1],
  (Fremdschlüssel2_Attribut1, Fremdschlüssel2_Attribut2)[Relation2]);
Attribut1 UNIQUE
Attribut2 NOT NULL
  
```



Fortsetzung nächste Seite!

**Aufgabe 4 (SQL)****[32 PUNKTE]**

Gegeben sind folgende Relationen:

**Spieler**(SpielerID, Vorname, Nachname, Adresse, Gehalt,  
MannschaftsID[Mannschaft])

**Mannschaft**(MannschaftsID, Bezeichnung, Ort, Kapitae[n][Spieler], Torwart[Spieler])  
Bezeichnung UNIQUE NOT NULL  
Kapitae[n] NOT NULL  
Torwart NOT NULL

Fremdschlüssel werden durch die Nennung der Relation, auf die sie verweisen, in eckigen Klammer dargestellt. Verwenden Sie im Folgenden nur Standard-SQL und keine produktspezifischen Erweiterungen. Sie dürfen bei Bedarf Views anlegen. Geben Sie einen Datensatz nicht mehrfach aus.

- Schreiben Sie eine SQL-Anweisung, die die Tabelle *Spieler* anlegt. Gehen Sie davon aus, dass die Tabelle *Mannschaft* bereits existiert.
- Schreiben Sie eine SQL-Anweisung, die Vor- und Nachnamen der Spieler aller Mannschaften aus München ausgibt, absteigend sortiert nach SpielerID.
- Schreiben Sie eine SQL-Anweisung, die Vor- und Nachnamen sowie das Gehalt von Spielern ausgibt, die mehr verdienen als der Kapitän ihrer Mannschaft. Sortieren Sie die Ausgabe absteigend nach dem Gehalt.
- Schreiben Sie eine SQL-Anweisung, die das Gehalt von allen Spielern aus der Mannschaft mit der ID 42 um 15% erhöht.
- Schreiben Sie eine SQL-Anweisung, welche die Vor- und Nachnamen und das Gehalt der fünf schlechtbezahltesten Spieler ausgibt, die keine Torwarte sind. Standardkonforme Sprachkonstrukte, die eine Beschränkung der Ausgabe bewirken, sind erlaubt.
- Schreiben Sie eine SQL-Anweisung, die für jede Mannschaft die Spieler ermittelt, die am meisten verdienen. Geben Sie die Vor- und Nachnamen der Spieler sowie die Bezeichnung der jeweiligen Mannschaft aus.

**Aufgabe 5 (Relationale Algebra)****[10 PUNKTE]**

Formulieren Sie basierend auf den in der Aufgabe 4 gegebenen Relationen die geforderten Anfragen in der relationalen Algebra.

- Formulieren Sie eine Anfrage, welche die Vor- und Nachnamen der Spieler ausgibt, die in einer Mannschaft aus Erlangen spielen.
- Formulieren Sie eine Anfrage, welche die Vor- und Nachnamen der Spieler ausgibt, die keine Mannschaft haben.

Fortsetzung nächste Seite!

**Aufgabe 6 (Normalformen)****[8 PUNKTE]**

Gegeben ist die Relation

**Prüfungen**(Matrikelnummer, Fach, StudName, GebDatum, Note)

mit den Schlüsselkandidaten {Matrikelnummer, Fach} und {StudName, Fach, GebDatum}. Es gelten außerdem die funktionalen Abhängigkeiten

$$\begin{aligned}\{\text{MatrNr}\} &\rightarrow \{\text{StudName}, \text{GebDatum}\} \\ \{\text{StudName}, \text{GebDatum}\} &\rightarrow \{\text{MatNr}\}\end{aligned}$$

Alle Attributwerte sind atomar.

Geben Sie die höchste Normalform an, die die Relation **Prüfungen** erfüllt. Zeigen Sie, dass alle Bedingungen für diese Normalform erfüllt sind und dass mindestens eine Bedingung der nächsthöheren Normalform verletzt ist. Beziehen Sie sich bei der Begründung auf die gegebene Relation und nennen Sie nicht nur die allgemeinen Definitionen der Normalformen.

**Aufgabe 7 (Optimierung)****[10 PUNKTE]**

- a) Erläutern Sie den Unterschied zwischen logischer (= algebraischer) und physischer Optimierung in zwei bis drei Sätzen.
- b) Übertragen Sie folgendes SQL-Statement in einen *nicht optimierten* algebraischen Ausdruck oder einen Anfrage-/Operatorgraph.
  - 1 SELECT Student.Name, Student.Matrikelnummer
  - 2 FROM Student, Pruefungsergebnis
  - 3 WHERE Student.Matrikelnummer = Pruefungsergebnis.Student
  - 4 AND Pruefungsergebnis.Note = 1.0;
- c) Nennen Sie zwei Möglichkeiten, den algebraischen Ausdruck oder einen Anfrage-/Operatorgraphen aus der vorhergehenden Teilaufgabe logisch (= algebraisch) zu optimieren. Beziehen Sie sich auf konkrete Stellen und Operatoren des von Ihnen aufgestellten algebraischen Ausdrucks oder Graphen.

Fortsetzung nächste Seite!

**Teilaufgabe II: Softwaretechnologie****Aufgabe 1 (Entwicklungsprozesse)****[25 PUNKTE]**

- a) Nennen Sie zwei Vorgehensmodelle zur strukturierten Durchführung eines Softwareprojekts und beschreiben Sie diese kurz. Geben Sie außerdem je Modell sowohl einen Vor- und Nachteil an.
- b) Nennen Sie drei Qualitätsmerkmale eines Softwareprodukts und beschreiben Sie diese kurz.
- c) Nennen Sie einen Vorteil von Reviews und Inspections gegenüber Tests. Geben Sie außerdem eine Phase im Softwarelebenszyklus an, in der Reviews und Inspections besser geeignet sind als Tests.
- d) Nennen Sie die drei wesentlichen Testphasen und die korrespondierenden Entwurfs- bzw. Spezifikationsphasen des V-Modells.
- e) Geben Sie zwei zentrale Aufgaben des Projektmanagements während der Projektabwicklung an.
- f) Während der Anforderungsanalyse können die einzelnen Anforderungen in unterschiedliche Kategorien eingeteilt werden. Stellen Sie den Unterschied der beiden Anforderungskategorien *Funktionale Anforderungen* und *Nichtfunktionale Anforderungen* dar.
- g) Elementarer Bestandteil eines Sprints in SCRUM ist die Durchführung von Daily Scrums. Nennen Sie die drei zentralen Fragen während der täglichen Besprechungen und beschreiben Sie deren Zweck mit einem Satz.

Fortsetzung nächste Seite!

**Aufgabe 2 (Projektmanagement)****[19 PUNKTE]**

Betrachten Sie die folgende Tabelle zum Projektmanagement:

Arbeitspaket	Dauer (Tage)	kann erst beginnen nach..
A1	5	
A2	10	A1, A6
A3	5	A6
A4	5	A2, A3
A5	5	A4, A7
A6	10	
A7	25	
A8	5	A5

- a) Erstellen Sie ein Gantt-Diagramm, das die in der Tabelle rechte Spalte angegebenen Abhängigkeiten berücksichtigt.  
Das Diagramm muss nicht maßstabsgetreu sein, jedoch jede Information aus der gegebenen Tabelle enthalten.
- b) Wie lange dauert das Projekt mindestens?
- c) Geben Sie alle kritischen Pfade an.
- d) Nehmen Sie an, dass jedes Arbeitspaket maximal von einer Person gleichzeitig bearbeitet werden kann. Geben Sie an, welche Konsequenzen es hätte, wenn das Projekt von weniger als drei Personen bearbeitet wird.
- e) Identifizieren Sie, welche Arbeitspakete länger dauern könnten, damit insgesamt alle Arbeitspakete auf kritischen Pfaden liegen. Geben Sie die Anzahl der Puffer-Tage für diese Arbeitspakete an.
- f) Bewerten Sie folgende Aussage eines Projektmanagers:  
„Wir können das Projekt nicht durchführen, weil alle Arbeitspakete auf einem kritischen Pfad liegen.“

Fortsetzung nächste Seite!



**Aufgabe 3 (Use-Case-Diagramm)****[20 PUNKTE]**

Im Folgenden ist eine Systembeschreibung für eine „Kontaktdatenerfassung“ angegeben.

Universitäten und viele weitere Organisationen sind verpflichtet, die Kontaktdaten und Anwesenheitszeiten ihrer Veranstaltungsgäste zu erfassen und auf Anforderung dem Gesundheitsamt zu übermitteln. Um „Zettelwirtschaft“ zu vermeiden, wollen wir hierfür eine elektronische Lösung basierend auf einer Webseite zur Verfügung stellen, die dieses Problem zentral für viele Einrichtungen löst.

Beim Besuch einer Veranstaltung werden die Gäste auf eine bestimmte Internetadresse hingewiesen. Dort können Sie ohne gesonderte Anmeldung oder Registrierung den „Check-in“ durchführen, also ihre Kontaktdaten eintragen. Falls dabei unplausible Daten (wie z.B. Name „Hans123“) eingetragen werden, soll eine Fehlermeldung angezeigt werden. Um den Aufwand für Gruppen zu reduzieren, sollen Gäste dabei außerdem auch etwaige Begleitpersonen eintragen können. Beim Check-in wird eine PIN angezeigt, mit der die Gäste beim Verlassen der Veranstaltung dann den „Check-out“ durchführen können. Mit dieser PIN können sich wiederkehrende Gäste beim nächsten Besuch außerdem als Alternative zum Standard-Check-in wieder einchecken und sparen sich so die Eingabe der Kontaktdaten. Begleitpersonen müssen dabei aber trotzdem neu eingetragen werden.

Der erste Kontakt mit dem System für Inhaber einer Einrichtung ist eine Registrierung mit Benutzername und Passwort. Danach stellt das System für angemeldete Benutzer eine Reihe von Konfigurationsmöglichkeiten zur Verfügung (z.B. Hinzufügen des eigenen Logos etc.). Während einer Veranstaltung sollen Inhaber für Gäste ohne Smartphone den Check-in (ohne PIN) durchführen können. Und nach einer Veranstaltung müssen Inhaber alle verbleibenden Gäste austragen können, falls diese den Checkout vergessen haben. Und natürlich muss es möglich sein, die Kontaktdaten aller Besucher eines Zeitraums für das Gesundheitsamt abzufragen. Eine Fehlermeldung soll auch bei jeder Verwendung ungültiger PINs oder Anmeldedaten angezeigt werden. Um den gesetzlichen Anforderungen zu genügen, muss es jederzeit für alle möglich sein, die Datenschutzhinweise einzusehen.

- a) Erstellen Sie ein Use-Case-Diagramm für das beschriebene System, in dem die Akteure und Anwendungsfälle inkl. möglicher Generalisierungen und Beziehungen eingetragen sind. Achten Sie auf mögliche <<includes>>- und <<extends>>- Beziehungen und Bedingungen für Anwendungsfälle.
- b) Identifizieren Sie außer den modellierten Akteure zwei weitere Stakeholder und geben Sie je eine mögliche Anforderung des Stakeholders an und für welchen Anwendungsfall diese relevant ist.

Fortsetzung nächste Seite!

**Aufgabe 4 (Objektorientierte Analyse)****[15 PUNKTE]**

Betrachten Sie das folgende Szenario:

Eine Plattform für Online-Bestellung ist an mehrere Unternehmen angebunden.

**Bestellvorgang:** Die Nutzer des Systems können zuerst aus einer Liste von lokalen Unternehmen auswählen. Ist das Unternehmen ein Restaurant, können die Nutzer Speisen auswählen. Nach der Auswahl von Speisen können die Nutzer zusätzlich als Option einen gewünschten Liefertermin auswählen. Ist das Unternehmen ein Einzelhandel, können die Nutzer aus einer Liste von Artikeln auswählen. Die Auswahl eines Liefertermins bzw. eines Artikels beendet den Bestellvorgang.

**Kommunikation:** Wenn ein Nutzer einen Artikel auswählt, fragt die Plattform die Verfügbarkeit beim jeweiligen Unternehmen an. Erst wenn das Unternehmen bestätigt, dass der ausgewählte Artikel vorrätig ist, wird er der Bestellung des Nutzers hinzugefügt. Währenddessen kann der Nutzer aber weitere Artikel auswählen. Wenn der Nutzer die Bestellung absendet, sendet die Plattform die Bestellung an das Unternehmen weiter und der Nutzer erhält danach direkt eine Bestätigung.

- (a) Geben Sie **fünf** geeignete Klassen für das obige Szenario an. Nennen Sie dabei keine Klassen, welche durch Basisdatentypen wie Integer oder String abgedeckt werden können.
- (b) Erstellen Sie ein Aktivitätsdiagramm für den oben beschriebenen „Bestellvorgang“ (aus Sicht der **Nutzer**). Der Vorgang „Kommunikation“ muss dabei nicht berücksichtigt werden.
- (c) Beachten Sie folgenden Anwendungsfall:  
Ein Nutzer wählt einen Hut aus. Der Artikel ist vorrätig und der Nutzer sendet seine Bestellung ab. Das Diagramm soll mindestens je einen Akteur für Nutzer (Browser), Plattform (Webserver) und Unternehmen (Kleidungsladen) vorsehen.

Erstellen Sie ein Sequenzdiagramm für den Vorgang „Kommunikation“ für das gegebene System. Beginnen Sie mit dem Anklicken des Artikels „Hut“ durch den Benutzer.

Fortsetzung nächste Seite!

**Aufgabe 5 (Design-Pattern)****[15 PUNKTE]**

- (a) Nennen und erläutern Sie kurz **zwei** Vorteile von Design-Pattern im Allgemeinen.
- (b) Erklären Sie zu jeder Komponente des „Model-View-Controller“-Architekturmusters die zugrundeliegende Aufgabe.
- (c) Benennen Sie die zentrale Problemstellung und zwei der damit einhergehenden Probleme, die durch den Einsatz des „Model-View-Controller“-Patterns vermieden werden sollen.
- (d) Zeigen Sie mittels einer Skizze den Zusammenhang zwischen den verschiedenen Komponenten des „Model-View-Controller“-Architekturmusters. Geben Sie dabei an, welche Informationen zwischen den Komponenten ausgetauscht werden und in welcher Reihenfolge dies erfolgt. Als Startpunkt können Sie hierfür von einer Interaktion des Benutzers mit der Benutzeroberfläche ausgehen.
- (e) Welches Design-Pattern wird als fundamentaler Bestandteil einer MVC-Architektur eingesetzt, um das Modell und die View konsistent zu halten? Wie heißen die beiden zentralen Komponenten dieses Patterns?

Fortsetzung nächste Seite!

**Aufgabe 6 (Softwarequalitätssicherung)****[26 PUNKTE]**

Gegeben sei folgende Methode (in Java), welche den Index des größten Elements im Array zurückgibt:

```
1  public int max(int[] arr) {  
2      int left = 0;  
3      int right = arr.length - 1;  
4      while (left < right) {  
5          if (arr[left] < arr[right]) {  
6              left++;  
7          } else if (arr[left] > arr[right]) {  
8              right--;  
9          }  
10     }  
11     return right;  
12 }
```

- (a) Zeichnen Sie einen Kontrollflussgraphen obiger Methode. Beschriften Sie dabei die Knoten mit den Zeilennummern des Programmcodes.
- (b) Geben Sie maximal drei Testfälle für die gegebene Methode an, um vollständige Anweisungsüberdeckung zu erreichen.
- (c) Welche Testfälle sind notwendig, um die Testfälle der bisherigen Anweisungsüberdeckung so zu erweitern, dass eine 100 %-ige Zweigüberdeckung erfüllt ist? Begründen Sie Ihre Entscheidung.
- (d) Beschreiben Sie zwei allgemeine Vorteile von Zweigüberdeckung gegenüber der Anweisungsüberdeckung.
- (e) Beschreiben Sie kurz den Ansatz des Test-Driven-Developments und nennen Sie zwei Vorteile davon.
- (f) Nennen Sie zwei Vorteile von Unit-Tests und erläutern Sie kurz, welchen Einfluss diese Vorteile auf die Wartung eines größeren Softwaresystems haben.
- (g) Definieren Sie kurz den Begriff „Verifikation“ im Rahmen der Softwareentwicklung und geben Sie eine Möglichkeit an, wie diese umgesetzt werden kann.