

---

**Prüfungsteilnehmer****Prüfungstermin****Einzelprüfungsnummer**

---

**Kennzahl:** \_\_\_\_\_**Kennwort:** \_\_\_\_\_**Arbeitsplatz-Nr.:** \_\_\_\_\_**Herbst  
2019****46116**

---

**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: 14**

---

**Bitte wenden!**

## Thema Nr. 1

### Aufgabengruppe

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

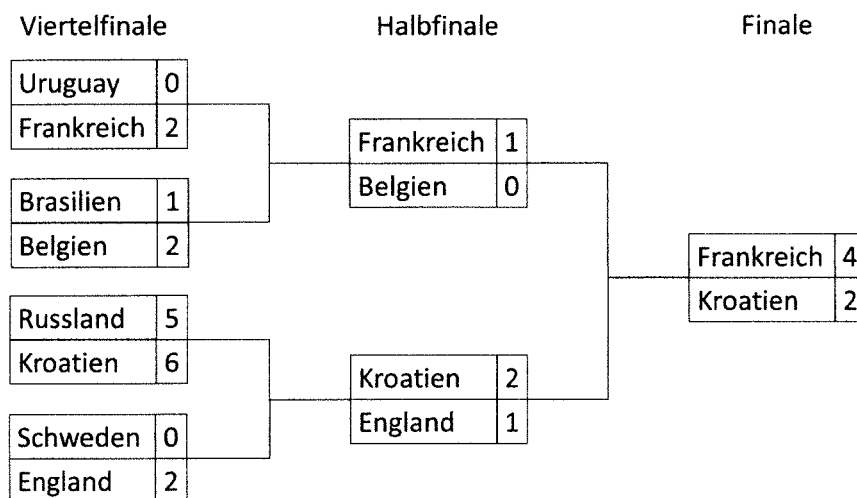
### Teilaufgabe I: Softwaretechnologie

#### Aufgabe 1 (Klassen- und Objektdiagramme)

(30 Punkte)

In einem Fußballturnier treten Mannschaften in Spielen gegeneinander an. An jedem Spiel sind zwei Mannschaften beteiligt. Jedes Spiel hat ein Ergebnis und genau einen Sieger. Das Turnier besteht aus mehreren Runden. In der ersten Runde treten alle Mannschaften an. Der Sieger erreicht jeweils die nächste Runde. Ein Spielplan legt fest, welche Mannschaften gegeneinander antreten. Das Turnier endet mit einem Endspiel (kein Spiel um den dritten Platz).

Beispiel: Endrunde der WM 2018 ab Viertelfinale



- a) Erstellen Sie ein UML-Klassendiagramm zur Modellierung eines Fußballturniers, an dem  $2^n$  Mannschaften teilnehmen können ( $n \geq 1$  beliebig). Geben Sie für Attribute jeweils den Namen und den Datentyp an. Geben Sie für Assoziationen den Namen und für jedes Ende den Rollennamen und die Multiplizität an. Nutzen Sie ggf. Vererbung, Komposition oder Aggregation. Verzichteten Sie auf Operationen und Sichtbarkeiten.
- b) Erstellen Sie für das Klassendiagramm aus a) und die Endrunde aus der Aufgabenstellung ein Objektdiagramm. Beschränken Sie sich dabei auf die Mannschaften und Spiele im Halbfinale und im Finale. Geben Sie Rollennamen nur an, wenn es notwendig ist, um die Enden eines Links (Instanz einer Assoziation) zu unterscheiden.

**Fortsetzung nächste Seite!**

**Aufgabe 2** (Entwurfsmuster)

(30 Punkte)

Ein Kurznachrichtendienst versendet Nachrichten, die aus einer Rubrik (Politik, Wirtschaft oder Sport) und einer Zeichenkette bestehen. Abonnenten können sich beim Kurznachrichtendienst registrieren. Der Kurznachrichtendienst versendet alle Nachrichten an alle Abonnenten. Die Abonnenten entscheiden darüber, wie sie die Nachricht verarbeiten. Ein Abonnent für Sportnachrichten gibt auf der Standardausgabe die Überschrift SPORT, gefolgt vom Nachrichtentext aus; Abonnenten für die übrigen Rubriken verhalten sich analog.

- a) Implementieren Sie diese Anwendung in einer Programmiersprache Ihrer Wahl mit Hilfe des Beobachter-Musters; geben Sie diese Sprache an. Bei den Abonnenten beschränken Sie sich dabei auf die Rubrik Sport.
- b) Visualisieren Sie folgenden Ablauf mit Hilfe eines Sequenzdiagramms, wobei Sie annehmen, dass das Senden von Nachrichten von einem Testprogramm ausgelöst wird:
  1. Der Abonnent `politik` registriert sich.
  2. Der Abonnent `sport1` registriert sich.
  3. Das Testprogramm ruft die Versende-Operation des Kurznachrichtendienstes auf.
  4. Der Abonnent `sport1` deregistriert sich.
  5. Der Abonnent `sport2` registriert sich.
  6. Der Abonnent `wirtschaft` registriert sich
  7. Das Testprogramm ruft die Versende-Operation des Kurznachrichtendienstes auf.

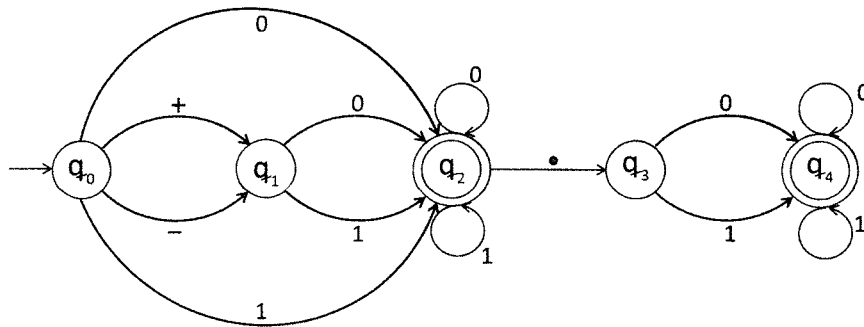
Hinweis: Verzichten Sie auf Aktivierungsbalken, Aufrufparameter und Rücksprünge. Verwenden Sie als Grundlage Ihre Lösung a). Setzen Sie das Testprogramm als gegeben voraus.

**Fortsetzung nächste Seite!**

**Aufgabe 3** (Zustandsbasiertes Testen)

(15 Punkte)

Gegeben sei der folgende endliche Automat zur Erkennung von Dualzahlen mit optionalem Vorzeichen und optionalem gebrochenen Anteil mit Dezimalpunkt:



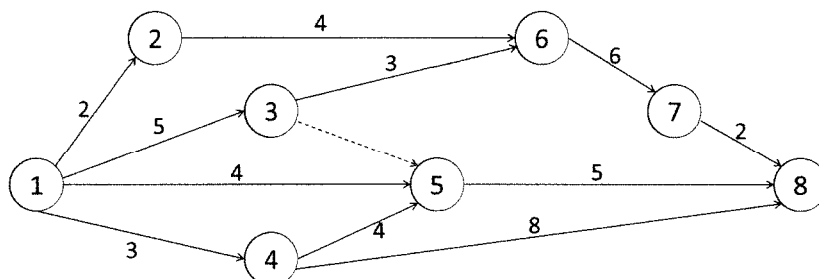
- Erläutern Sie die Begriffe Zustandsüberdeckung und Transitionsüberdeckung beim zustandsbasierten Testen.
- Geben Sie eine minimale Testmenge für die Zustandsüberdeckung an.
- Geben Sie eine minimale Testmenge für die Transitionsüberdeckung an.
- Welches der beiden Überdeckungskriterien ist schärfer? Begründen Sie Ihre Antwort.

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

**Aufgabe 4** (CPM-Netzwerke)

(15 Punkte)

Gegeben sei folgendes CPM-Netzwerk („Critical Path Method“):



- Berechnen Sie für jedes Ereignis den frühesten Zeitpunkt (Projektstart zum Zeitpunkt 0).
- Berechnen Sie für jedes Ereignis den spätesten Zeitpunkt (= frühester Zeitpunkt des letzten Ereignisses).
- Berechnen Sie für jedes Ereignis die Pufferzeit.
- Geben Sie einen kritischen Pfad für das Netzwerk an.

**Fortsetzung nächste Seite!**

**Teilaufgabe II: Datenbanken****Aufgabe 1** (Wissensfragen)

(5 Punkte)

Antworten Sie kurz und prägnant.

- a) Beschreiben Sie kurz den Zweck einer Datenbank, eines Datenbankmanagementsystems und eines Datenbankensystems.
- b) Nennen Sie zwei verschiedene Notationen, die man zur Darstellung von Mengenrelationen in der ER-Modellierung verwendet. Geben Sie zusätzlich jeweils die möglichen Ausprägungen der Notation an.

**Aufgabe 2** (ER-Modellierung)

(35 Punkte)

Entwerfen Sie ein ER-Diagramm, welches eine imaginäre Fernsehserie abbildet:

In dieser Serie gibt es Figuren mit nicht eindeutigen Namen, einem Sozialstatus und einem Wohlstand. Sie gehören entweder zum Volk, zu den Adeligen oder zu den Regierenden.

Ein Adelige oder ein Regierender ist zwangsläufig Teil einer Familie. Diese Familie ist eindeutig gekennzeichnet durch den Namen, und hat zusätzlich noch eine Anzahl an Mitgliedern. Es gibt weiterhin Ortschaften, welche einen eindeutigen Namen haben und entweder ein Dorf, eine Ruine, eine Festung oder ein Hafen sein können. Jede existierende Ortschaft ist genau einer Familie zugeordnet, wobei Familien bereits ausgestorben sein können und daher keine Ortschaften oder Figuren zugeordnet haben.

Weiterhin gibt es sieben Königreiche. Jedes Königreich hat einen gewissen Reichtum und hat zugeordnete Familien, wobei einem Königreich mindestens fünf Familien zugehören müssen. Weiterhin regiert eine Familie ein oder mehrere (bis zu allen 7) Königreiche.

Außerdem gibt es Armeen, die abhängig von einer Familie sind. Da eine Familie mehrere Armeen haben kann, sind diese zusätzlich durch ihren Rufnamen identifiziert. Dabei kann eine Familie eine Armee besitzen, muss aber nicht. Die Armeen haben verschiedene Größen und sind unterschiedlich stark.

Zuletzt gibt es besondere Figuren, die einen Drachen fliegen können. Dieser existiert nur in Beziehung mit seinem Besitzer und hat einen Namen.

- a) Modellieren Sie ein ER-Diagramm. Überlegen Sie sich, welche Entitäten und welche Arten von Beziehungen in diesem Beispiel relevant sind, bestimmen Sie Primärschlüssel und fügen Sie die beschriebenen Attribute hinzu. Falls kein eindeutiger Primärschlüssel gegeben ist, führen Sie einen Surrogatschlüssel ein.
- b) Geben Sie im Diagramm die Funktionalitäten in der Min-Max-Notation an!

**Fortsetzung nächste Seite!**

**Aufgabe 3** (SQL)

(30 Punkte)

Formulieren Sie folgende Anfragen in SQL gegen die angegebene Zauberer-Datenbank.

Schüler (Id, Name, Patronus, Haarfarbe, Aktiv, Gesamtnote)  
teil\_von (Id, Name,  
FK (Id) references Schüler(Id),  
FK (Name) references Haus(Name) )  
Haus (Name, Geist, Vertrauenslehrer, Wappen)  
Jahrgang (Schuljahr, Einschulung, Größe)  
gehört\_zu (Id, Einschulung  
FK (Id) references Schüler(Id),  
FK (Einschulung) references Jahrgang(Einschulung) )  
Quidditch (Haus, Captain, Tropäen,  
FK (Haus) references Haus(Name) )

- a) Welche Schüler (Name ist gesucht) hat eine "Hirschkuh" als Patronus?
- b) Wie viele Schüler gehören jeweils zu den verschiedenen Häusern?
- c) Geben Sie eine Liste der im Haus Gryffindor existierenden Patronie an.
- d) Gesucht sind die besten 5 Schüler aus dem 4. Schuljahr. Geben Sie den Namen, die Note und die Platzierung geordnet nach der Platzierung aus.
- e) Erstellen Sie eine neue Tabelle namens "Familie" mit folgendem Schema.

Familie (Name, Familiensymbol, Oberhaupt)

Alle Attribute müssen gesetzt werden.

- f) Ändern Sie die Tabelle Schüler so, dass jeder Schüler einer Familie angehören kann, aber nicht muss.
- g) Geben Sie die Namen der Schüler an, bei welchen das Familiensymbol nicht identisch zu irgendeinem Hauswappen eines Hauses ist.
- h) Löschen Sie alle Schüler aus Jahrgängen mit weniger als 20 Schülern.

**Fortsetzung nächste Seite!**

**Aufgabe 4** (Relationale Algebra)

(10 Punkte)

Gegeben ist das Datenbankenschema aus Aufgabe 3.

Übertragen Sie die folgenden Ausdrücke in die relationale Algebra. Beschreiben Sie diese Ausdrücke umgangssprachlich, bevor Sie die Ausdrücke umformen.

- a)  $\{s \mid s \in \text{Schüler} \wedge \neg \exists t \in \text{teil\_von} (t.\text{Id} = s.\text{Id})\}$
- b)  $\{s \mid s \in \text{Schüler} \wedge \exists t \in \text{teil\_von} (t.\text{Id} = s.\text{Id}) \wedge \exists h \in \text{Haus} (t.\text{Name} = h.\text{Name}) \wedge \exists q \in \text{Quidditch} (h.\text{Name} = q.\text{Haus} \wedge q.\text{Captain} = \text{"Harry Potter"})\}$

**Aufgabe 5** (Kanonische Überdeckung und Normalfall)

(10 Punkte)

Gegeben sei  $\mathcal{R} = (A, B, C, D, E, F, G, H)$  mit folgenden funktionalen Abhängigkeiten.

$F \rightarrow AH$   
 $E \rightarrow FG$   
 $AG \rightarrow BF$   
 $BC \rightarrow DH$

- a) Geben Sie die Hülle von E an.
- b) Geben Sie alle Schlüsselkandidaten an.
- c) Überführen Sie die Relation  $\mathcal{R}$  in die 3. NF, indem Sie den Synthesealgorithmus anwenden.

**Thema Nr. 2**  
Aufgabengruppe

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

**Teilaufgabe I: Softwaretechnologie**

(90 Punkte)

Ein Testfall besteht neben Informationen zur HW/SW-Konfiguration des Testsystems aus einer Eingabe und einer erwarteten Ausgabe oder der Charakterisierung einer erwarteten Ausgabe für das Programm. Für ein Programm, das die Quadratwurzel berechnet, sind mögliche Testfälle (4.0,2.0), (-2.3,error) oder für die Ausgabe  $x$  des Programms ( $0.04, 0.19 \leq x \leq 0.21$ ).

- a) [3 Punkte] Warum ist der erste Testfall problematisch und eine Lösung in der Art des *dritten Testfalls* vorzuziehen?
- b) [2+2 Punkte] Warum kann es allgemein sinnvoll sein, nicht konkrete Ausgabewerte als erwartete Ausgabe anzugeben, sondern stattdessen eine Menge möglicher Werte zu charakterisieren? Nennen Sie zwei Gründe, die unterschiedlich von der Begründung zu a) sind.

Wir vernachlässigen im Folgenden das Problem der Bestimmung der erwarteten Ausgabe, indem wir als erwartete Ausgabe konstant „nicht abgestürzt“ bzw. „keine Exception“ definieren. Bei der Testeingabeselektion wird der Eingaberaum üblicherweise in  $k$  Blöcke partitioniert und aus jedem Block dieselbe Anzahl  $n$  an Testdaten zufällig gleichverteilt ausgewählt. Eingaben können bzgl. einer hier angenommenen Spezifikation fehlerverursachend oder korrekt sein. Ein Ziel des Testens ist es, Tests mit möglichst vielen fehlerverursachenden Eingaben zu definieren.

Nun interessiert uns die Frage nach der Güte dieses sogenannten partitionsbasierten Testens, also der  $k \cdot n$  Testfälle. Nehmen wir an, dass das Gütemaß „Wahrscheinlichkeit, mindestens einen Fehler zu finden“ ist, also mindestens einen Testfall mit einer fehlerverursachenden Eingabe definiert zu haben.

- c) [14 Punkte] Wie groß ist dann i. A. die Güte einer Testsuite aus  $n \cdot k$  Testfällen für  $k$  Partitionen, wenn wir annehmen, dass die Testfälle unabhängig voneinander ausgewählt wurden, derselbe Test mehrfach ausgewählt werden kann und Block  $i$  der Partition  $c_i$  korrekte und  $w_i$  fehlerverursachende Eingaben enthält, so dass die Wahrscheinlichkeit eines Elements des Blocks  $i$ , fehlerverursachend zu sein, gleich der Fehlerrate  $r_i = w_i / (c_i + w_i)$  ist? Geben Sie die Güte der Testsuite als Wahrscheinlichkeitsfunktion in Abhängigkeit von  $r_i$ ,  $k$  und  $n$  an.

Hinweis: Machen Sie sich klar, dass die Wahrscheinlichkeit, keine inkorrekte Eingabe auszuwählen, das Komplement der Wahrscheinlichkeit ist, mindestens eine inkorrekte Eingabe auszuwählen und dass die Wahrscheinlichkeiten, in den einzelnen Blöcken voneinander zu finden, voneinander statistisch unabhängig sind.

**Fortsetzung nächste Seite!**



- d) [10 Punkte] Eine Alternative zum partitionsbasierten Testen ist das gleichverteilt zufällige Testen. Damit die Werte vergleichbar sind, nehmen Sie an, dass diese zufällige Testsuite aus  $N=k*n$  Testfällen und der Eingaberaum aus  $C=c_1+\dots+c_k$  korrekten und  $W=w_1+\dots+w_k$  fehlerverursachenden Eingaben besteht und die Wahrscheinlichkeit jeder Eingabe gleich der Fehlerrate  $R=W/(C+W)$  ist, fehlerverursachend zu sein. Wie groß ist die Güte dieser zufälligen Testsuite?

Partitionsbasiertes Testen ist i. A. aufwändiger als zufälliges Testen, weil die Partitionen errechnet und Testfälle entsprechend selektiert werden müssen. Deswegen sollte partitionsbasiertes Testen nur angewendet werden, wenn es gemäß unserem Gütemaß „besser“ als zufälliges Testen ist.

- e) [15 Punkte] Identifizieren Sie für  $C+W=100$ ,  $W=8$ ,  $n=1$ ,  $N=2$  und  $k=2$  drei Situationen, also Verteilungen von fehlerverursachenden Eingaben über die 100 Elemente des Eingaberaums, in denen partitionsbasiertes Testen einmal besser als zufälliges Testen, einmal schlechter als zufälliges Testen und einmal ebenso gut wie zufälliges Testen ist. Berechnen Sie die Güte der drei Testsuiten.

Aufgabe e) hat eine fundamentale Konsequenz: Da man i. A. nicht a-priori sagen kann, wo sich im Eingaberaum die fehlerverursachenden Eingaben befinden, kann man i. A. auch nicht voraussetzen, dass partitionsbasiertes Testen gemäß unserem Gütemaß immer besser als zufälliges Testen ist, was aus dieser theoretischen Perspektive das zufällige Testen als sinnvolle Alternative erscheinen lässt!

- f) [10 Punkte] Unter welchen Bedingungen wird partitionsbasiertes Testen i. A. besser sein als zufälliges Testen? Charakterisieren Sie abstrakt die dazu notwendige Verteilung der fehlerverursachenden Eingabedaten in den Blöcken der Partition.
- g) [7 Punkte] Wie sehen i. A. Testauswahlstrategien aus, die solche Partitionen induzieren? Begründen Sie Ihre Antwort.

Anforderungsbasiertes Testen bezeichnet eine Strategie, bei der Tests gemäß der Programmspezifikation ausgewählt werden: Pro in der Spezifikation beschriebener Funktionalität (use case) wird eine definierte Anzahl von Testfällen ausgewählt. Funktionalitäten bilden jeweils unterschiedliche Eingabemengen auf Klassen von Ausgaben ab und induzieren also eine Partition des Eingabedatenraums. Die vorherigen Teilaufgaben zeigen deswegen, dass anforderungsbasiertes Testen besser als, gleich gut wie oder schlechter als zufälliges Testen sein kann – und das führt zu dem überraschenden Resultat, dass man, wenn man das hier erarbeitete Modell zugrundelegt, aus Kostengründen eher zufällig als anforderungsbasiert testen sollte.

- h) [8 Punkte] Warum kann es sinnvoll sein, trotzdem anwendungsbasiert zu testen? Warum ist ein Großteil des Testens in der Praxis anforderungsbasiert?

Eine andere Art des Testens ist das Grenzwerttesten. Dabei werden aus der Spezifikation für geordnete Datentypen relevante Intervalle extrahiert und als Grundlage für das Testen verwendet, in dem an den Rändern sowie aus der Mitte des Intervalls Tests ausgewählt werden. Eine Möglichkeit unter anderen ist, dass je ein Test genau links der linken/rechten Intervallgrenze, je ein Test genau auf der linken/rechten Intervallgrenze, je ein Test genau rechts der linken/rechten Intervallgrenze und ein Test an beliebiger anderer Stelle des Intervalls ausgewählt wird. Wenn also beispielsweise eine Spezifikation besagt, dass bei Temperaturen zwischen 4 und 80 Grad eine grüne Lampe leuchten soll und andernfalls eine rote, sind typische Grenzwerttests (3, rot), (4, grün), (5, grün), (42, grün), (79, grün), (80, grün), (81, rot). Hintergrund ist, dass empirisch Programmierer häufiger Fehler in „Randbereichen“ machen.

- i) [6 Punkte] Wenn wir also den Rändern von Intervallen von im Vergleich zu den sonstigen Eingaben erhöhten Fehlerwahrscheinlichkeiten ausgehen können, wie sieht dann allgemein eine partitionsbasierte Teststrategie mit 3 Blöcken und 3 Tests für ein Intervall  $[i..j]$  mit  $j > i+3$  aus? Können wir davon ausgehen, dass diese Teststrategie i. A. besser sein wird als zufälliges Testen?

100 % Pfadabdeckung eines deterministischen Programms bedeutet, dass alle Pfade von Startknoten zu Endknoten im Kontrollflussgraphen ausgeführt werden. Wir gehen im Folgenden davon aus, dass alle Pfade endlich sind, alle Schleifenzähler also eine uns vorab bekannte konstante obere Schranke besitzen.

- j) [2 Punkte] Kann man für jedes Programm eine solche obere Schranke für alle Schleifen berechnen, wenn es sie gibt, und umgekehrt herausfinden, ob jede Schleife über eine solche obere Schranke verfügt?
- k) [1+1 Punkte] Das Testkriterium Pfadabdeckung induziert offenbar eine Partition des Eingabedatenraums. Charakterisieren Sie umgangssprachlich diese Partition und diskutieren Sie anhand des oben entwickelten Modells, ob Sie einem Praktiker diese Form der Pfadabdeckung empfehlen würden.
- l) [3 Punkte] Wenn (endliche) Pfadabdeckung im obigen Sinn als Testselektionskriterium verwendet wird, kann damit die Korrektheit von Programmen bewiesen werden? Begründen Sie Ihre Antwort.

Das hier diskutierte Modell zur Bemessung der Qualität einer Testsuite basiert auf diversen Annahmen.

- m) [3+3 Punkte] Fassen Sie diese Annahmen zusammen und diskutieren Sie sowohl diese Annahmen als auch das gewählte Gütemaß. Wie könnte umgangssprachlich ein besseres Gütemaß aussehen?

**Teilaufgabe II: Datenbanken****Aufgabe 1** (ER-Modellierung)

(25 Punkte)

Erstellen Sie ein möglichst einfaches ER-Schema, das alle unten angegebenen Informationen enthält. Attribute von Entitäten und Beziehungen sind anzugeben, Schlüsselattribute durch Unterstreichen zu kennzeichnen. Verwenden Sie für die Angabe der Kardinalitäten von Beziehungen die Min-Max-Notation. Führen Sie Surrogatschlüssel nur dann ein, wenn es nötig ist, und modellieren Sie nur die im Text vorkommenden Elemente.

Zunächst gibt es Autos, die einen eindeutigen Namen, einen Typ sowie eine Liste an Ausstattungen besitzen. Autos werden aus Bauteilen zusammengesetzt. Diese besitzen eine ID sowie eine Beschreibung. Jedes Bauteil wird von genau einem Zulieferer geliefert. Zu jedem Zulieferer werden sein Name sowie seine Email-Adresse gespeichert.

Weiter gibt es Werke, die einen eindeutigen Namen sowie einen Standort besitzen. Jedes Werk besteht aus Hallen, welche werksintern eindeutig nummeriert sind. Zudem besitzt eine Halle noch eine gewisse Größe (in m<sup>2</sup>). Es gibt genau zwei Typen von Hallen: Produktionshallen und Ersatzteillager. In jeder Produktionshalle wird mindestens ein Auto hergestellt. Zu den Ersatzteillagern wird festgehalten welche Bauteile und wie viele sich dort befinden.

Zu jedem Mitarbeiter werden eine eindeutige ID, sein Vor- und Nachname, die Adresse (Straße, PLZ, Ort), das Gehalt sowie das Geschlecht gespeichert. Jeder Mitarbeiter besitzt genau einen Vorgesetzten, der selbst Mitarbeiter ist.

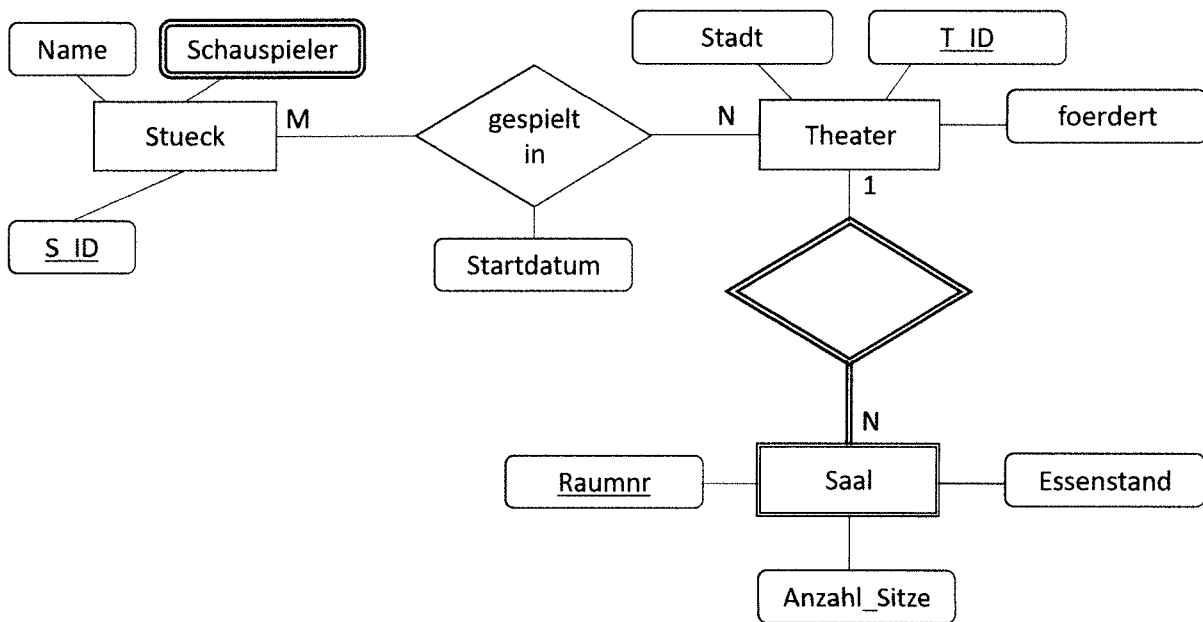
Mitarbeiter werden unter anderem in Reinigungskräfte, Werksarbeiter und Ingenieure unterteilt. Zu letzteren wird zusätzlich der Hochschulabschluss gespeichert. Ingenieure sind genau einem Werk zugeordnet, Werksarbeiter einer Halle. Eine Reinigungskraft reinigt mindestens eine Halle. Jede Halle muss regelmäßig gereinigt werden. Weiter sind Ingenieure Projekten zugeteilt. Zudem wird zu jedem Projekt genau ein Ingenieur als Projektleiter festgehalten.

**Fortsetzung nächste Seite!**

**Aufgabe 2** (Relationaler Entwurf und Normalisierung)

(22 Punkte)

Gegeben ist folgendes unsaubere ER-Modell. Das Attribut „foerdert“ gibt an, ob die Stadt die Theater in ihrer Stadt fördert, d. h. eine Stadt fördert jeweils alle oder kein Theater dieser Stadt.



- [7 Punkte] Transformieren Sie das ER-Diagramm in ein relationales Modell. Geben Sie die Herkunft der Fremdschlüssel an und markieren Sie die Primärschlüssel durch Unterstreichen.
- [4 Punkte] Geben Sie jeweils die Definition für die erste und zweite Normalform an.
- [7 Punkte] Welche Normalform erfüllen Ihre in Teilaufgabe a) erstellten Relationen jeweils? Stellen Sie ihre Ergebnisse in folgender Form dar und begründen Sie Ihre Antwort.

Relation	1NF	2NF	3NF	BCNF
Tabelle1				
Tabelle2				
...				

- [4 Punkte] Transformieren Sie, sofern nötig, Ihr Schema in die dritte Normalform. Geben Sie alle hierfür notwendigen Zwischenschritte an.

**Fortsetzung nächste Seite!**

**Aufgabe 3.1** (SQL-Abfragen)

(20 Punkte)

Gegeben sind folgende Relationen aus einem Verwaltungssystem für die jährlichen Formel 1-Rennen:

```
Strecke(Strecken_ID, Streckenname, Land, Länge)
Fahrer(Fahrer_ID, Fahrername, Nation, Rennstall)
Rennen(Strecken_ID, Jahr, Wetter)
      FK (Strecken_ID) referenziert Strecke(Strecken_ID)
Rennteilnahme(Fahrer_ID, Strecken_ID, Jahr, Rundenbestzeit, Gesamtzeit, disquali-
fiziert)
      FK (Fahrer_ID) referenziert Fahrer(Fahrer_ID)
      FK (Strecken_ID, Jahr) referenziert Rennen(Strecken_ID, Jahr)
```

Der Einfachheit halber wird angenommen, dass Fahrer den Rennstall nicht wechseln können. Das Attribut „disqualifiziert“ kann die Ausprägungen „ja“ und „nein“ haben.

Formulieren Sie folgende Abfragen in SQL. Vermeiden Sie nach Möglichkeit übermäßige Nutzung von Joins und Views.

- [3 Punkte] Geben Sie die Anzahl der Fahrer aus, die 2017 am Rennen in „Monaco“ teilgenommen haben.
- [4 Punkte] Geben Sie für jeden Fahrer, der in den Jahren 2005-2017 mindestens einmal disqualifiziert wurde, seinen Namen, seine ID sowie die Anzahl seiner Disqualifikationen in diesem Zeitraum an. Ordnen Sie die Ausgabe absteigend nach der Anzahl der Disqualifikationen.
- [4 Punkte] Gesucht sind alle Länder aus denen noch nie ein Fahrer disqualifiziert wurde.
- [9 Punkte] Schreiben Sie eine SQL-Anweisung, die aus der Tabelle Strecke alle Strecken löscht, bei denen das Wetter öfter „regnerisch“ als „sonnig“ war.

**Aufgabe 3.2** (Tabellen mit SQL erstellen)

(5 Punkte)

Erstellen Sie in SQL eine Tabelle Person und setzen Sie die folgenden Constraints. Wählen Sie geeignete Datentypen.

```
Person(SSN, Name, Vorname, Geschlecht, Geburtsdatum, Gehalt)
```

Constraints:

- SSN ist der Primärschlüssel der Relation
- Gehalt muss zwischen 2000 und 3000 liegen
- Geschlecht darf nur die Ausprägungen 'm', 'w' und 'd' besitzen

**Fortsetzung nächste Seite!**

**Aufgabe 4** (Vermischte Fragen)

(18 Punkte)

Beantworten Sie die folgenden Fragen und begründen oder erläutern Sie Ihre Antworten.

- a) [2 Punkte] Nennen Sie den Unterschied zwischen einer Spezialisierungshierarchie und einem Spezialisierungsbaum.
- b) [2 Punkte] Sind Chen- und min-max-Notation äquivalent?
- c) [2 Punkte] Gibt es Sachverhalte, die in der ER-Modellierung nicht dargestellt, in SQL allerdings umgesetzt werden können? Falls ja, geben Sie konkrete Beispiele an, falls nein, begründen Sie ausführlich!
- d) [3 Punkte] Was versteht man unter Unionkompatibilität? Nennen Sie drei SQL-Operatoren, welche Unionkompatibilität voraussetzen.
- e) [3 Punkte] Nennen Sie die drei Teilsprachen von SQL und beschreiben Sie kurz deren Aufgaben.
- f) [2 Punkte] Erklären Sie den Begriff „Datenbankindex“ und nennen Sie zwei häufige Arten.
- g) [4 Punkte] Nennen Sie zwei Möglichkeiten, die Laufzeit einer SQL-Anfrage zu verbessern. Illustrieren Sie diese an je einem kurzen Beispiel.