RS ✓

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr 2013

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: **11**

Bitte wenden!

Thema Nr. 1

Teilaufgabe 1:

1. Sie sollen für einen Online-Shop für Musikinstrumente eine Software zur Verwaltung des Inventars implementieren. Zu jedem Artikel gehört eine Nummer, eine Art (Instrument, Noten, Zubehör, ...), ein Hersteller, ein Lieferant, ein Einzelpreis, sein Lagerbestand (Anzahl der Einheiten). Dazu kommt eine Zusatzinformation, etwa ob und in welchem Umfang ein Artikel vorbestellt wurde.
   Es soll möglich sein, den Lagerbestand aufzufüllen und zu reduzieren (zum Beispiel beim Verkauf). Außerdem soll es möglich sein, Aufträge an Lieferanten zu erstellen.
   Die beteiligten Personen können Kunden, Verkäufer oder Administratoren sein; der Einfachheit halber können Verkäufer und Administratoren nicht Kunden sein.
   Beteiligte Personen können sich beim System an- und wieder abmelden.
   Kunden können Artikel kaufen und bezahlen. Außerdem können Kunden den Bestand durchsuchen und erhalten dann für jeden geführten Artikel die Mitteilung, ob er verfügbar ist oder nicht. In letzterem Falle können sie ihn vorbestellen. Kunden können aber nicht Stückzahlen oder Lieferanten einsehen.
   Verkäufer können Artikel verkaufen, den Bestand durchsuchen, verändern und Aufträge an Lieferanten tätigen. Außerdem können sie Kunden neu aufnehmen oder löschen.
   Administratoren haben alle Befugnisse von Verkäufern und können außerdem Verkäufer und Administratoren aufnehmen oder löschen.

   a) Entwickeln Sie aus der gegebenen Anwendungsbeschreibung ein UML-Klassendiagramm!

   Alle Angaben aus dem Text sollen sich in Ihrem Modell wiederfinden; weitere sollten nur dann hinzugenommen werden, wenn es aufgrund der von Ihnen gewählten Modellierung erforderlich erscheint!
   Verwenden Sie Vererbung und Interfaces, wo es sinnvoll möglich ist und achten Sie auf eine angemessene Darstellung von Assoziationen und Aggregationen, wo erforderlich. Es geht hier nur um das UML-Klassendiagramm; insbesondere ist eine Modellierung als relationale Datenbank *nicht* verlangt und auch nicht im Sinne der Aufgabenstellung!

   b) Modellieren Sie den gesamten Anwendungsfall des Kaufens eines Artikels (Anmelden, Bestand anzeigen, Aussuchen, Kaufen, Abmelden) zunächst als UML-Anwendungsfall und dann als UML-Aktivitätsdiagramm. Achten Sie auch auf die Möglichkeit des Fehlschlagens einer Teilaktivität!

2. Erläutern Sie in jeweils ca. 3 Sätzen die folgenden Begriffe aus der modernen Softwaretechnologie: Pair Programming, Refactoring, agile Softwareentwicklung.

Fortsetzung nächste Seite!

Teilaufgabe 2:

I. Datenbanksysteme

1. Modellierung

Sie sollen ein System zur Verwaltung von Pferderennen entwerfen.
Gehen Sie dabei von folgendem Szenario aus:

- Unternehmen werden über ihre eindeutige Unternehmens-Id identifiziert. Sie haben eine Adresse und besitzen Rennställe.
- Der Name eines Rennstalls ist nur innerhalb eines Unternehmens eindeutig. Für jeden Rennstall wird das Gründungsdatum gespeichert.
- Pferde gehören immer zu einem Rennstall. Pferdenamen werden in einem Rennstall nur jeweils maximal einmal vergeben.
- Jockeys sind in einem Rennstall beschäftigt. Jeder Rennstall vergibt seine eigenen Personalnummern. Für jeden Jockey werden sein Vorname und Name gespeichert.
- Rennen haben ein Datum, ein Preisgeld und einen Namen, über den sie identifiziert werden.
- Unternehmen unterstützen Rennen finanziell mit einem bestimmten Betrag.
- Jockeys nehmen mit Pferden an Rennen teil. Im Rennen erreichen sie einen bestimmten Platz. Die Kombination aus Jockey und Pferd ist nicht fest, bei unterschiedlichen Rennen können Jockeys verschiedene Pferde reiten. Jockeys können auch mit Rennpferden von fremden Rennställen, die anderen Unternehmen gehören können, an Rennen teilnehmen.

a) Entwerfen Sie für das beschriebene Szenario ein ER-Modell.
Bestimmen Sie hierzu:

- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
- ein passendes ER-Diagramm,
- die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen, und
- die Funktionalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an. Achten Sie dabei insbesondere darauf, dass die Relationenschemata keine redundanten Attribute enthalten.

Fortsetzung nächste Seite!

2. Anfragen

Zu einer Website, auf der Besucher im Browser Online-Spiele spielen können, liegt das folgende relationale Schema einer Datenbank vor:

$$\begin{aligned}\text{Team} &: \{[\underline{\text{TNr}}, \text{Name}, \text{Teamfarbe}]\}\\ \text{Spieler} &: \{[\underline{\text{SNr}}, \text{Name}, \text{Icon}, \text{TNr}, \text{EMail}]\}\\ \text{Minispiel} &: \{[\underline{\text{MNr}}, \text{Name}, \text{Kategorie}, \text{Schwierigkeit}]\}\\ \text{Wettkampf} &: \{[\underline{\text{WNr}}, \text{Sieger}, \text{Geschlagener}, \text{MNr}, \text{Dauer}]\}\end{aligned}$$

Auf der Website treten jeweils zwei Spieler gegeneinander in Minispielen an. In diesen ist es das Ziel, den gegnerischen Spieler in möglichst kurzer Zeit zu besiegen. Minispiele gibt es dabei in verschiedenen Schwierigkeitsstufen ('leicht', 'mittel', 'schwer', 'sehr schwer') und verschiedenen Kategorien ('Denkspiel', 'Geschicklichkeitsspiel', usw.). Die Attribute *Sieger* und *Geschlagener* sind jeweils Fremdschlüsselattribute, die auf das Attribut *SNr* der Relation *Spieler* verweisen. Beachten Sie, dass das *Dauer*-Attribut der Wettkampf-Relation die Dauer eines Wettkampfes in der Einheit Sekunden speichert.

a) Formulieren Sie geeignete Anfragen in relationaler Algebra für die folgenden Teilaufgaben:

   i. Geben Sie die Namen der Minispiele zurück, die zur Kategorie 'Denkspiele' zählen.

   ii. Geben Sie die Namen und E-Mail-Adressen aller Spieler zurück, die in einem Minispiel des Typs 'Geschicklichkeitsspiel' gewonnen haben.

b) Formulieren Sie geeignete SQL-Anfragen für die folgenden Teilaufgaben. Beachten Sie dabei, dass Ihre Ergebnisrelationen keine Duplikate enthalten sollen.

   i. Geben Sie jede Spielekategorie aus, für die ein Minispiel der Schwierigkeitsstufe 'sehr schwer' vorhanden ist.

   ii. Geben Sie die Wettkämpfe aus, deren Dauer unter der durchschnittlichen Dauer der Wettkämpfe liegt.

   iii. Geben Sie für jeden Spieler seine *SNr*, seinen Namen, die Anzahl seiner Siege, die durchschnittliche Dauer seiner siegreichen Wettkämpfe und die Anzahl der Teams, aus denen er bereits mindestens einen Spieler besiegt hat, zurück.

c) Verwenden Sie den relationalen Tupelkalkül, um die folgenden Anfragen zu formulieren:

   i. Finden Sie die Namen der Spieler des Teams 'Dream Team'.

   ii. Geben Sie die Namen der Minispiele zurück, bei denen bereits Spieler gegeneinander angetreten sind, deren Teams dieselbe Teamfarbe haben.

Hinweis: Die Anfragen im relationalen Tupelkalkül dürfen auch nicht sicher sein.

Fortsetzung nächste Seite!

3. Entwurfstheorie

Gegeben ist der folgende Ausschnitt einer relationalen Datenbank:

Vereine							
UserId	ClubId	Vorname	Name	Login	ClubName	Funktion	Vorstand
1	1	Max	Muster	maxm	IT Kicker	Trainer	false
1	2	Max	Muster	maxm	CC Nerds	2. Vorsitzende(r)	true
2	2	Marta	Maier	maier1	CC Nerds	1. Vorsitzende(r)	true
3	1	Tim	Peters	timpeters	IT Kicker	1. Vorsitzende(r)	true
4	1	Peter	Huber	timpeters	IT Kicker	Spieler(in)	false
5	3	Tina	Müller	tinam	IT Club	Kassenprüfer(in)	true
...	...	...	...	...	...	...	...

In der Datenbank sollen folgende funktionale Abhängigkeiten gelten:

$UserId \rightarrow Vorname, Name, Login$
$Login \rightarrow Vorname, Name, UserId$
$UserId, ClubId, Name \rightarrow Funktion, ClubName$
$Login, ClubId, Name \rightarrow Funktion, ClubName$
$ClubId \rightarrow ClubName$
$Funktion \rightarrow Vorstand$

a) Geben Sie sämtliche Schlüsselkandidaten an.

b) Welcher Normalform genügt die Relation *Vereine*? Begründen Sie Ihre Antwort ausführlich.

c) Verwenden Sie den Synthesealgorithmus, um die Relation in die dritte Normalform zu überführen.

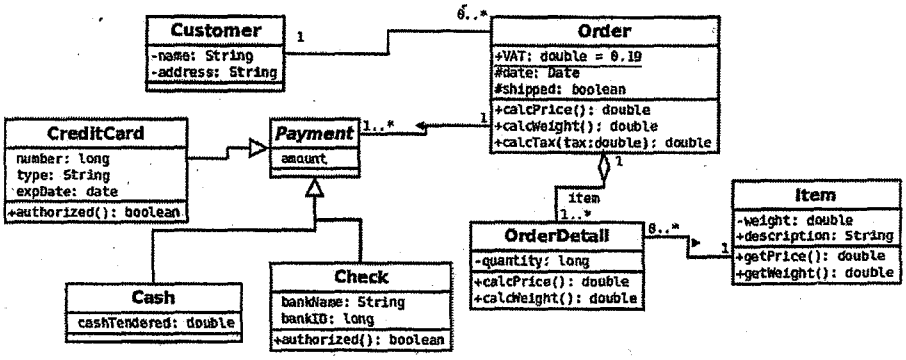
d) Befinden sich die neuen Relationen in Boyce-Codd-Normalform? Begründen Sie Ihre Antwort.

Thema Nr. 2

Teilaufgabe 1:

Aufgabe 1:

Gegeben sei folgendes Klassendiagramm:



a) Implementieren Sie die Klassen „Order", „Payment", „Check", „OrderDetail" und „Item" in einer geeigneten objektorientierten Sprache Ihrer Wahl. Beachten Sie dabei insbesondere Sichtbarkeiten, Klassen- vs. Instanzzugehörigkeiten und Schnittstellen bzw. abstrakte Klassen.

b) Erstellen Sie ein Sequenzdiagramm (mit konkreten Methodenaufrufen und Nummerierung der Nachrichten) für folgendes Szenario:
   i. „Erika Mustermann" aus „Rathausstraße 1, 10178 Berlin" wird als neue Kundin angelegt.
   ii. Frau Mustermann bestellt heute 1kg Gurken und 2kg Kartoffeln.
   iii. Sie bezahlt mit ihrer Visa-Karte, die im August 2014 abläuft und die Nummer „1234 567891 23456" hat – die Karte erweist sich bei der Prüfung als gültig.
   iv. Am Schluss möchte sie noch wissen, wie viel ihre Bestellung kostet – dabei wird der Anteil der Mehrwertsteuer extra ausgewiesen.

Fortsetzung nächste Seite!

Aufgabe 2:

Rekonstruieren Sie aus dem folgenden Java-Code das zugehörige UML-Klassendiagramm. Dabei sind folgende Randbedingungen zu beachten:

- Sichtbarkeiten von Feldern (Attributen u. Klassenvariablen), Methoden und Assoziationsrollen sind entspr. anzugeben – die der Klassen sind hingegen irrelevant.
- Methodenrümpfe und Initialwerte von Feldern können, müssen aber nicht angegeben werden.
- Abstrakte Klassen und Methoden sind mit einem ★ hinter dem Namen deutlich zu kennzeichnen.
- Assoziationen zwischen den gegebenen Klassen sind zusammen mit Rollennamen, Richtung der Navigierbarkeit und Multiplizitäten (graphisch im Diagramm) anzugeben – eine Unterscheidung in Aggregation und Komposition ist hingegen nicht erforderlich.

```
public interface A {
	public static int aa = 4711;
	public double ab = 0.815;

	public void ac();
}

abstract class B implements A {
	private static void ba() { }

	public abstract String bb(int x);
}

class C extends B {
	public A ca;
	private D cb;

	@Override
	public void ac() { }

	@Override
	public String bb(int x) {bb(""); return "";}

	private boolean bb(String y) {cb.ac(); return true;}
}

class D extends B {
	private C[] da;

	public void ac() { }

	public String bb(int x) {db(0, 0); return da.toString();}

	private void db(long a, int b) {}
}
```

Fortsetzung nächste Seite!

Teilaufgabe 2:

Aufgabe 1: E/R-Modellierung, SQL-DDL

Für die Verwaltung eines Zoos soll folgendes System entworfen werden:

- Die zentrale Entität eines Zoos ist ein Gehege. Ein Gehege hat eine eindeutige Nummer und einen Namen und kann eingeteilt werden in die Unterkategorien offenes Gehege, Streichelgehege oder Aquarium.
- Zusätzliche Attribute für spezielle Gehege sind Größe in Quadratmetern (offenes Gehege), Anzahl der Futterautomaten und maximal eingelassene Anzahl von Besuchern (Streichelgehege) sowie die Anzahl von Räumen und Becken (Aquarium). Für offene Gehege soll zusätzlich modelliert werden können, ob sich darin ein Gebäude für die Tiere befindet.
- In jedem Gehege des Zoos lebt mindestens eine Tierart. Eine Tierart ist eindeutig durch eine Bezeichnung beschrieben und hat zudem eine lateinische Bezeichnung. Eine Tierart lebt in genau einem Gehege des Zoos.
- Jede Tierart gehört genau einer bestimmten Klasse von Lebewesen an (z.B. Säugetiere). Diese Klasse besitzt ebenso eine eindeutige Bezeichnung sowie einen lateinischen Namen.
- Für eine später mögliche Neugestaltung der Gehege werden Informationen darüber benötigt, welche Tierarten sich untereinander vertragen.
- Für jede im Zoo lebende Tierart gibt es mindestens ein Tier, welches eine eindeutige Nummer hat. Zusätzlich sollen für jedes Tier der Name und das Geburtsdatum aufgezeichnet werden.
- Jede Tierart im Zoo wird durch mindestens einen Tierpfleger betreut. Ein Tierpfleger hat eine eindeutige Mitarbeiternummer sowie einen Namen und einen Vornamen. Jeder Tierpfleger betreut nur genau eine Tierart.
- Im Zoo werden wöchentlich für einige Tierarten öffentliche Fütterungen angeboten. Dies erfolgt für mindestens eine Tierart durch einen oder mehrere Tierpfleger. Zusätzliche Infos sind hier der Wochentag und die Uhrzeit.

1. Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank.
2. Setzen Sie das gegebene E/R-Diagramm in ein entsprechendes relationales Datenbankschema um. Geben Sie die resultierenden Relationenschemata in folgender Schreibweise an:

   `Relation (Attribut1, Attribut2, ..., AttributN)`

   Identifizieren Sie für jede Relation einen Primärschlüssel und unterstreichen Sie diesen. Achten Sie auf eine geeignete Modellierung der Beziehungen (Relationships). Nummerische Attribute sollen geeigneten Integritätsbedingungen unterliegen. Geben Sie außerdem weitere sinnvolle Constraints an.
3. Geben Sie die Anweisungen in SQL-DDL an, die notwendig sind, um die Relationen aus Teilaufgabe (2) in einer relationalen Datenbank zu erzeugen. Kennzeichnen Sie dabei die Primär- und Fremdschlüssel der Relationen.

Fortsetzung nächste Seite!

Aufgabe 2: Relationale Algebra

Gegeben sei das folgende relationale Schema mitsamt Beispieldaten für eine Datenbank von Musikalben. Die Primärschlüssel-Attribute sind jeweils unterstrichen, Fremdschlüssel sind überstrichen.

"Interpret":

IID	Name	Vorname	Künstlername	Geburtsstadt
I1	Germanotta	Stefani	Lady Gaga	New York
I2	Vetter	Jan	Farin Urlaub	Berlin
I3	Würdig	Paul	Sido	Berlin
I4	Köhler	Horst	Guildo Horn	Trier

"Komponist":

KID	Name	Vorname
K1	Raab	Stefan
K2	Khayat	Nadir
K3	Beardie	Roe
K4	Vetter	Jan

"Album":

AID	Albumtitel	Erscheinungsdatum
A1	The Fame	19.07.2008
A2	Maske	10.07.2004
A3	Endlich Urlaub!	01.07.2001

"Lied":

TID	Komponist	Interpret	Album	Titel	BesteChartplazierung
T1	K1	I4	null	Guildo hat euch lieb	5
T2	K2	I1	A1	Pokerface	1
T3	K2	I1	A1	Paparazzi	2
T4	K3	I3	A2	Fuffies im Club	18
T5	K4	I2	A3	Jeden Tag Sonntag	58

Formulieren Sie die folgenden Anfragen auf das gegebene Schema in relationaler Algebra:

1. Finden Sie die Namen aller Interpreten, welche in Berlin geboren sind.
2. Suchen Sie alle Komponisten, die Lieder für "Lady Gaga" geschrieben haben.
3. Finden Sie die Titel aller Alben mit mindestens einem Lied, dessen Interpret und Komponist den gleichen Vornamen und Nachnamen haben.
4. Finden Sie alle Komponisten, welche einen Titel komponiert haben, der vor dem 01.01.2000 auf einem Album erschienen ist.

Geben Sie das Ergebnis (bezüglich der Beispieldaten) der folgenden Ausdrücke der relationalen Algebra als Tabellen an:

5. $\pi_{KID,AID}(Komponist \times Album)$
6. $\pi_{Album,Titel,BesteChartplazierung}(Lied) \underset{Lied.Album=Album.AID}{\bowtie} \pi_{AID,Erscheinungsdatum}(Album)$

Fortsetzung nächste Seite!

Aufgabe 3: SQL

Gegeben sei das relationale Datenbank-Schema aus Aufgabe 2.

"Interpret":

IID	Name	Vorname	Künstlername	Geburtsstadt
I1	Germanotta	Stefani	Lady Gaga	New York
I2	Vetter	Jan	Farin Urlaub	Berlin
I3	Würdig	Paul	Sido	Berlin
I4	Köhler	Horst	Guildo Horn	Trier

"Komponist":

KID	Name	Vorname
K1	Raab	Stefan
K2	Khayat	Nadir
K3	Beardie	Roe
K4	Vetter	Jan

"Album":

AID	Albumtitel	Erscheinungsdatum
A1	The Fame	19.07.2008
A2	Maske	10.07.2004
A3	Endlich Urlaub!	01.07.2001

"Lied":

TID	Komponist	Interpret	Album	Titel	BesteChartplazierung
T1	K1	I4	null	Guildo hat euch lieb	5
T2	K2	I1	A1	Pokerface	1
T3	K2	I1	A1	Paparazzi	2
T4	K3	I3	A2	Fuffies im Club	18
T5	K4	I2	A3	Jeden Tag Sonntag	58

Formulieren Sie die folgenden Anfragen in SQL:

1. Finden Sie die Titel aller Lieder, welche irgendwann unter den Top 10 der Charts waren.
2. Geben Sie die Titel aller Lieder und, falls sie auf einem Album erschienen sind, den entsprechenden Albumtitel aus.
3. Finden Sie die Künstlernamen aller Interpreten, welche mehr als 10 Titel in den Top 10 der Charts hatten.
4. Finden Sie die ID des Komponisten, der für die meisten verschiedenen Künstler Lieder geschrieben hat.

Wie sieht die Ergebnisrelation zu folgenden Anfragen auf den Beispieldaten aus?

5. `select Komponist, Min(BesteChartplazierung) from Lied group by Komponist;`
6. `Select * from Interpret where IID not in (select Interpret from Lied, Album where Album = AID and Erscheinungsdatum > '01.06.2002');`

Fortsetzung nächste Seite!

Aufgabe 4: Entwurfstheorie

Gegeben sei die nachfolgende relationale Datenbank mit unterstrichenen Schlüsselattributen. Sie enthält Informationen über Zugverbindungen. Relation „Zugverbindungen“:

Zugtyp	Start	Start Gleis	Ziel	ZielGleis	Bordbistro	Halt	HaltGleis
ICE	Nürnberg	9	München	22	ja	Ingolstadt	4
IC	Hamburg	7	Berlin	1	ja	Berlin-Spandau	5
RE	Nürnberg	14	München	26	nein	Ingolstadt	2
IC	Hamburg	7	Berlin	1	ja	Wittenberge	3
ICE	München	22	Essen	4	ja	Frankfurt	7
RE	Nürnberg	14	München	26	nein	Pfaffenhofen	1
ICE	Hamburg	14	München	15	ja	Hannover	4
ICE	Hamburg	8	Berlin	11	ja	Berlin-Spandau	5
TGV	Stuttgart	9	Paris	3	ja	Karlsruhe	6

Sowie die folgenden funktionalen Abhängigkeiten:

FD1: Zugtyp —> Bordbistro
FD2: Zugtyp, Start, Ziel —> StartGleis, ZielGleis
FD3: Zugtyp, Start, Ziel, Halt —> HaltGleis

1. Beschreiben Sie kurz, welche Redundanzen in der Datenbank vorhanden sind und welche Anomalien auftreten können. Geben Sie ein Beispiel für jede Anomalie an.
2. Welcher Normalform genügt das Relationenschema dieser Datenbank? Begründen Sie Ihre Entscheidung!
3. Überführen Sie das Relationenschema der Datenbank in die dritte Normalform und geben Sie die mit obigen Daten gefüllten Relationen an.
4. Erläutern Sie kurz, welchen Nachteil Normalisierung allgemein für die Anfragebearbeitung haben kann.