

Name:

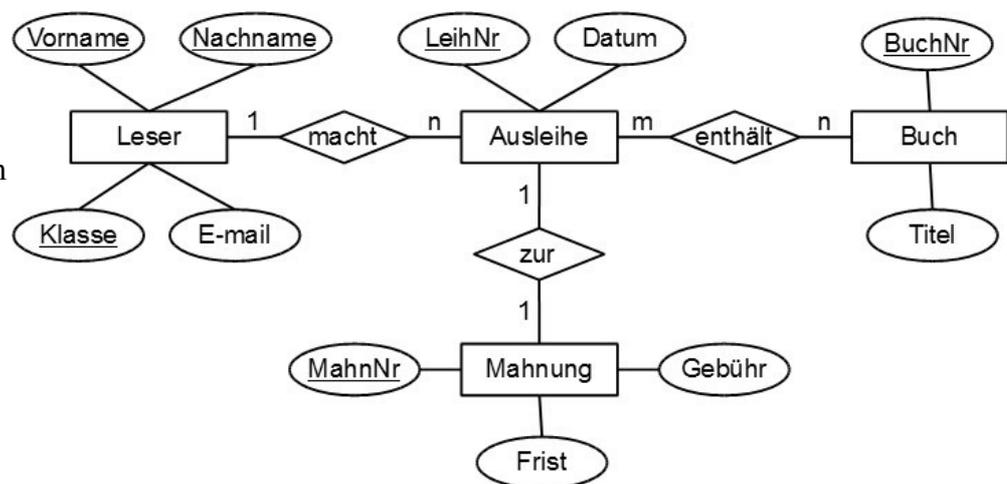
Aufgabe 1

Erstellen Sie ein ER-Diagramm zu folgender Miniwelt:

Ein Sportverein ist in Abteilungen (Leichtathletik, Turnen, Fußball, ...) aufgeteilt. Jede Abteilung bietet mehrere Sportgruppen an (z.B. bietet die Abteilung Fußball eine A-Jugend, B-Jugend, Altherrengruppe, ... an). Jede Sportgruppe wird von einem Trainer geleitet. Jede Abteilung hat einen Vorsitzenden. Trainer und Vorsitzende sind Mitglieder des Vereins. Jedes Mitglied wird über eine Mitgliedsnummer und Name, Adresse, Telefonnummer und E-mail-Adresse erfasst. Jedes Mitglied kann an beliebig vielen Sportgruppen unterschiedlicher Abteilungen teilnehmen.

Aufgabe 2

Erstellen Sie unter Anwendung der Transformationsregeln zu dem rechts dargestellten ER-Diagramm einer Schülerbücherei das passende relationale Datenbankschema.



Aufgabe 3

In einer Unternehmens-Datenbank ist eine Relation geplant, die Informationen über die Mitarbeiter speichern soll. Das Relationsschema soll wie folgt aussehen:

Mitarbeiter(Vorname, Nachname, Abteilung, Adresse, Gehaltsstufe, Gehalt)

Zwei beispielhafte Tupel der Relation könnten wie folgt aussehen:

<u>Vorname</u>	<u>Nachname</u>	Abteilung	Adresse	Gehaltsstufe	Gehalt
Arnold	Ahorn	Architektur	Alte Allee 2, 50773 Köln	7	3200
Berti	Buchsbaum	Bauleitung	Bahnhofstr. 5, 51140 Köln	10	4000

Das angegebene Gehalt ergibt sich dabei direkt aus der Gehaltsstufe. Es enthält keine weiteren Boni oder Alterszuschläge. Da es auch Mitarbeiter gibt, die von weiter her anreisen müssen, soll die Möglichkeit bestehen, die Mitarbeiter, die direkt in Köln wohnen, auszufiltern.

- Erklären Sie, inwiefern das angegebene Relationsschema nicht der dritten Normalform entspricht.
- Überführen Sie das Datenbankschema (soweit es angegeben ist) in die dritte Normalform und erläutern Sie alle Änderungen, die dazu notwendig sind.

Aufgabe 4

Eine andere Datenbank für die Gehaltsstruktur von Managern eines Unternehmens, die Grundgehälter und Boni für besondere Zusatzleistungen bekommen können, wird mit dem folgenden Datenbankschema verwirklicht:

Mitarbeiter (MitarbeiterID, Vorname, Nachname, Grundgehalt, Kontonummer, ↑Bankleitzahl)

Bank (Bankleitzahl, Name)

Bonus (BonusID, Leistung, Hoehe)

bekommt (↑MitarbeiterID, ↑BonusID)

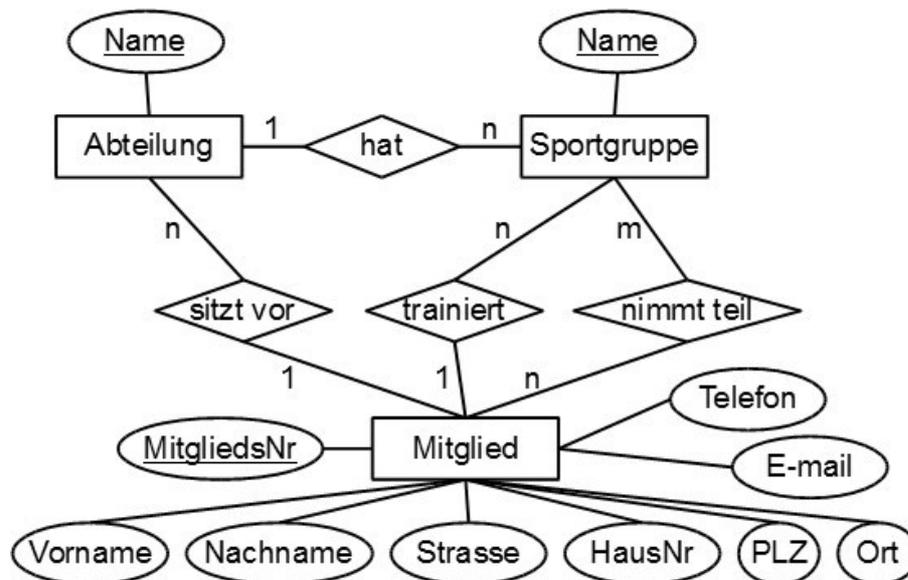
Formulieren Sie SQL-Anweisungen zu folgenden Anfragen:

- a) Gesucht sind alle Mitarbeiter, bei denen als Grundgehalt mehr als 4000 eingetragen ist. Das Ergebnis soll aus dem Vornamen, Nachnamen und dem Grundgehalt bestehen und nach dem Grundgehalt absteigend sortiert sein.
- b) Gesucht sind die Nachnamen aller Mitarbeiter, die mehr als einen Bonus bekommen.
- c) Gesucht sind die Vor- und Nachnamen aller Mitarbeiter und die Leistungen, für die sie Boni bekommen. Das Ergebnis soll alphabetisch nach Nachnamen und bei gleichem Nachnamen nach den Vornamen sortiert sein und auch die Mitarbeiter enthalten, die keine Boni bekommen.

Viel Erfolg!

Erwartungshorizont Beispielklausur Informatik 01.01.2010

Aufgabe 1



Aufgabe 2

Leser (Vorname, Nachname, Klasse, E-mail)

Ausleihe (LeihNr, Datum, ↑Vorname, ↑Name, ↑Klasse, ↑MahnNr)

Buch (BuchNr, Titel)

Mahnung (MahnNr, Frist, Gebühr)

enthält (↑LeihNr, ↑BuchNr)

Aufgabe 3

a) Die dritte Normalform umfasst auch die erste und zweite, deswegen müssen alle drei Normalformen überprüft werden. Laut erster Normalform müssen alle Attribute atomare, also nicht weiter aufteilbare Werte enthalten. Dies ist bei dem Attribut **Adresse** nicht der Fall, insofern ist die erste Normalform verletzt. Laut zweiter Normalform darf kein Attribut direkt von einem Teil des Schlüssels abhängig sein. Hier besteht der Primärschlüssel aus mehr als einem Attribut, nämlich **Vorname** und **Nachname**. Auch die zweite Normalform könnte also theoretisch verletzt sein. Allerdings gibt es kein anderes Attribut, welches direkt nur vom Vornamen oder nur vom Nachnamen abhängt. Insofern ist die zweite Normalform nicht verletzt. Laut dritter Normalform darf kein Attribut direkt von einem oder mehreren anderen Nichtschlüsselattributen abhängig sein. Dies ist hier bei dem Gehalt der Fall, welches direkt von der Gehaltsstufe abhängt. Insofern ist die dritte Normalform verletzt.

b) Die erste Normalform kann erfüllt werden, indem einfach das nicht-atomare Attribut in mehrere Attribute aufgeteilt wird. Aus dem Attribut **Adresse** werden die Attribute **Strasse**, **Hausnr**, **Plz** und **Ort** gemacht. Um die dritte Normalform zu erfüllen, muss das abhängige Attribut zusammen mit dem Nichtschlüsselattribut, von dem es abhängt, als Primärschlüssel in eine neue Relation ausgelagert werden. Das abhängige Attribut verschwindet dadurch aus der alten Relation und das Attribut, von dem es abhängt, wird dadurch in der alten Relation zu einem Fremdschlüsselattribut. Hier wird also eine neue Relation mit dem Relationsschema **Gehalt** (Gehaltsstufe, **Hoehe**) erstellt, das Attribut **Gehalt** aus der Relation **Mitarbeiter** gelöscht und in der Relation **Mitarbeiter** das Attribut **Gehaltsstufe** zu einem Fremdschlüssel gemacht.

Das Ergebnis der Normalisierung sieht dann wie folgt aus:

Mitarbeiter (Vorname, Nachname, **Abteilung**, **Strasse**, **Hausnr**, **Plz**, **Ort**, ↑**Gehaltsstufe**)

Gehalt (Gehaltsstufe, **Hoehe**)

Aufgabe 4

- a) `SELECT Vorname, Nachname, Grundgehalt
FROM Mitarbeiter
WHERE Grundgehalt > 4000
ORDER BY Grundgehalt DESC`
- b) `SELECT Nachname
FROM Mitarbeiter INNER JOIN bekommt
ON Mitarbeiter.MitarbeiterID = bekommt.MitarbeiterID
WHERE bekommt.MitarbeiterID IN
 (SELECT MitarbeiterID
 FROM bekommt
 GROUP BY MitarbeiterID
 HAVING COUNT(MitarbeiterID) > 1)`
- c) `SELECT Vorname, Nachname, Leistung
FROM Mitarbeiter
LEFT JOIN bekommt
ON Mitarbeiter.MitarbeiterID = bekommt.MitarbeiterID
LEFT JOIN Bonus
ON bekommt.BonusID = Bonus.BonusID
ORDER BY Nachname, Vorname`