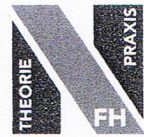


Thema:	Datenbanken (Wiederholungsklausur)		
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 2 von 6
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	02.06.2005	Bearbeitungszeit:	120 Minuten
Matrikelnummer:		Name:	



### 1) ER-Diagramm erstellen (9 Punkte)

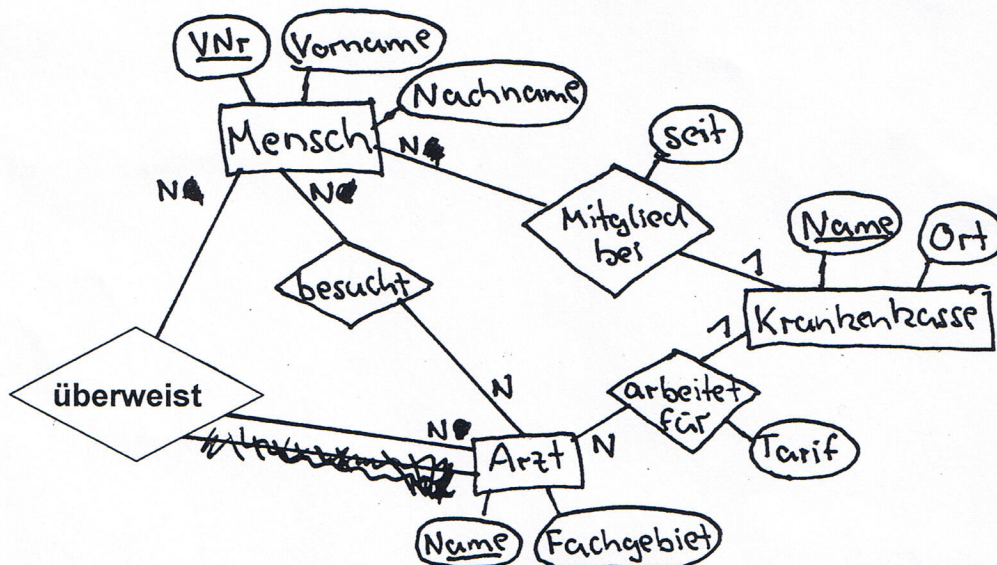
Formulieren sie folgenden Sachverhalt als ER-Diagramm (Entitäten, Relationen, Attribute, Schlüsselkandidat, Kardinalitäten (auch Stelligkeiten genannt) der Relationen).

Jeder Mensch wird durch seine Personalausweisnummer (PNr) identifiziert und hat einen Vor- und einen Nachnamen. Beliebige viele Menschen können Mitglieder von beliebig vielen Vereinen sein, die auch beliebig viele Mitglieder aufnehmen können. Für jede Mitgliedschaft ist das Jahr des Eintritts in den Verein bekannt. Vereine sind durch ihren Namen eindeutig und haben eine Adresse. Jeder Verein hat einen Vorstand, der aus einem oder mehreren Menschen besteht. Jeder Mensch kann maximal zum Vorstand eines Vereins gehören, wobei die Länge der Amtszeit bekannt ist.

Jeder Mensch kann für maximal eine Firma arbeiten, dabei ist der Eintrittstermin in die Firma bekannt. Jede Firma beschäftigt mindestens einen Menschen. Jede Firma sponsort beliebig viele Vereine, wobei jeder Verein maximal eine Firma als Sponsor hat.

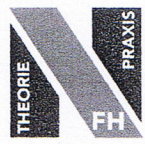
### 2) ER-Diagramm in Relationen (Tabellen) übersetzen (10+2 = 12 Punkte)

Gegeben sei das folgende ER-Diagramm:



- Leiten sie aus dem Diagramm Tabellen ab, die in den Spalten (insofern die Werte der im Diagramm genannten Attribute nicht NULL sind) keine NULL-Werte enthalten. Markieren sie einen Schlüsselkandidaten in jeder Tabelle. Vermeiden sie die Ableitung überflüssiger Einzeltabellen.
- Schauen sie sich das Attribut „seit“ an. Könnte man diese Information an andere Elemente des Diagramms verschieben, ohne dass sich die Aussage des Diagramms ändert?



Thema:	Datenbanken (Wiederholungsklausur)			
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 3 von 6	
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03	
Datum:	02.06.2005	Bearbeitungszeit:	120 Minuten	
Matrikelnummer:		Name:		

### 3) Relationen in Normalform (2+2+2+5 = 11 Punkte)

Gegeben sei folgende Beispieltabelle zur Verwaltung von Gruppenleitern in Vereinen (die angegebenen Abkürzungen können in der Aufgabe genutzt werden):

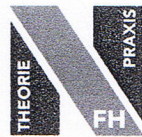
Vorname (Vn)	Nachname(Nn)	GruppenID(GID)	Gruppe(Gr)	Verein(Ver)	Vereinspräsident (VP)
Heinz	Meier	S04	Meisen	Spatz04	Assel
Heinz	Müller	S04	Meisen	Spatz04	Assel
Jörg	Assel	S01	Stare	Spatz04	Assel
Jörg	Meier	S02	Finken	Spatz04	Assel
Verena	Müller	A02	Finken	Adler05	Hannes
Jutta	Meier	A04	Meisen	Adler05	Hannes

Für die Tabelleninhalte gelten folgende Regeln

- i) Aus dem Vor- und Nachname zusammen kann man eindeutig auf die GruppenID schließen.
  - ii) Aus der GruppenID kann man eindeutig auf die Gruppe und den Verein schließen.
  - iii) Aus dem Verein kann man eindeutig auf den Vereinspräsident schließen.
  - iv) Aus dem Vereinspräsidenten kann man eindeutig auf den Verein schließen.
- a) Schreiben sie die Regeln als funktionale Abhängigkeiten auf.
  - b) Geben sie die Menge der Schlüsselattribute und die Menge der Nichtschlüsselattribute an.
  - c) Geben sie alle Schlüsselkandidaten an.
  - d) Überführen Sie (wenn nötig) die Tabelle (und die daraus resultierenden Tabellen) schrittweise in die 1NF, 2NF, 3NF und BCNF. Nehmen sie dabei nur die unbedingt nach dem vorgestellten Verfahren notwendigen Veränderungen vor. Markieren sie die Schlüsselkandidaten der Tabellen.



Thema:	Datenbanken (Wiederholungsklausur)		
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 4 von 6
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	02.06.2005	Bearbeitungszeit:	120 Minuten
Matrikelnummer:		Name:	



4) SQL-Anfragen formulieren (3+3+3+3+3+3+3+3 = 24 Punkte)

Gegeben seien folgende Tabellen, die mit Beispielinhalten gefüllt sind: (die jeweiligen Schlüssel sind markiert, die Angaben in der Tabelle Kino bedeuten z.B., dass es im Gloria 3 Säle mit jeweils 200 Plätzen gibt):

Film			Vorfuehrung		Kino		
<u>FID</u>	Titel	Laenge	<u>Film</u>	<u>Kino</u>	<u>Name</u>	Plaetze	Saele
1	Die Nase	90	1	Gloria	Gloria	200	3
2	Die Hand	85	2	Gloria	Apollo	300	2
3	Der Arm	120	3	Gloria			
4	Das Bein	75	2	Apollo			
			4	Apollo			

Formulieren sie die folgenden Textzeilen jeweils als SQL-Anfragen.

- Geben sie die Titel aller Filme aus, die länger als 85 Minuten sind.
- Geben sie die Namen aller Kinos aus, in denen der Film „Die Hand“ läuft.
- Geben Sie zu jedem Kino die Gesamtzahl aller zur Verfügung stehenden Plätze aus (Ausgabe: Kinoname, Gesamtplatzzahl).
- Geben sie die Titel der Filme aus, die in mindestens zwei Kinos laufen.
- Geben sie die Namen aller Kinos aus, in denen „Die Nase“ nicht läuft.
- Geben Sie zu jedem Kino die Länge des längsten Films aus, der in diesem Kino läuft (Ausgabe: Kinoname, maximale Länge).
- Geben sie zu jedem Filmtitel die maximale Anzahl von Zuschauern aus, die den Film gleichzeitig sehen können (Ausgabe: Filmtitel, maximale Besucherzahl; gehen sie davon aus, dass alle Vorführungen gleichzeitig und nur einmal am Tag stattfinden, ein Film in einem Kino nicht in mehreren Sälen läuft).
- Geben sie für jeden Film und jedes Kino an, ob dieser Film in diesem Kino läuft oder nicht (Ausgabe: Filmtitel, Kino, Anzahl). Dabei soll eine Anzahl>0 ausgegeben werden, wenn der Film in dem Kino läuft und sonst soll Anzahl=0 ausgegeben werden.

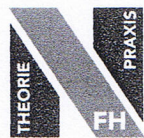
5) Tabelle in SQL definieren (6 Punkte)

Geben sie den SQL-Befehl zur Erzeugung der Tabelle Vorfuehrung aus der vorherigen Aufgabe in SQL an (nur **CREATE**, keine **INSERT**). Neben den dort „sichtbaren“ Randbedingungen, sollen folgende Bedingungen aufgenommen werden:

- im Gloria darf nie ein Film mit der Nummer 13 laufen
- es dürfen keine Filmnummern (Spalte Film) zwischen 500 und 600 (jeweils einschließlich) vergeben werden



Thema:	Datenbanken (Wiederholungsklausur)		
Dozent:	Prof. Dr. Stephan Kleuker	Seitennummer:	Seite 5 von 6
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	02.06.2005	Bearbeitungszeit:	120 Minuten
Matrikelnummer:		Name:	



Film		
FID	Titel	Laenge
1	Die Nase	90
2	Die Hand	85
3	Der Arm	120
4	Das Bein	75

Vorfuehrung	
Film	Kino
1	Gloria
2	Gloria
3	Gloria
2	Apollo
4	Apollo

Kino		
Name	Plaetze	Saele
Gloria	200	3
Apollo	300	2

6) Auswertung von DB-Anfragen (3+3+3 = 9 Punkte)

Gegeben seien folgende SQL-Anfragen an die Datenbank aus der Aufgabe 4.

- a) **SELECT \***  
**FROM Kino K1, Kino K2;**
- b) **SELECT F2.Titel**  
**FROM Film F1, Film F2**  
**WHERE F1.Titel='Die Hand'**  
**AND F2.Laenge > F1.Laenge;**
- c) **SELECT Kino.Name**  
**FROM Kino**  
**WHERE NOT EXISTS (**  
    **SELECT \***  
    **FROM Vorfuehrung, Film**  
    **WHERE Kino.Name=Vorfuehrung.Kino**  
        **AND Vorfuehrung.Film=Film.FID**  
        **AND Film.Laenge > 90);**

Geben sie jeweils die zugehörige Ausgabe an und beschreiben sie informell, was der Zweck der Anfrage ist.

7) Trigger (7 Punkte)

Schreiben sie einen Trigger der beim Einfügen eines neuen Films überprüft, ob es bereits einen Film gleichen Namens gibt. Ist dies der Fall, soll eine Warnung mit den Daten des Films bzw. der Filme gleichen Namens ausgegeben werden. Wird zum Beispiel folgendes INSERT-Statement ausgeführt

**INSERT INTO Film VALUES(5, 'Die Hand', 90)**

sollen folgende Meldung ausgegeben werden:

**Achtung, es gibt folgende(n) Film(e) gleichen Titels:**  
**Nr.2 Die Hand 85 Minuten**