

Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 1 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; text-align: center;"> nicht abzugeben, dient dem Selbsttest </div>		



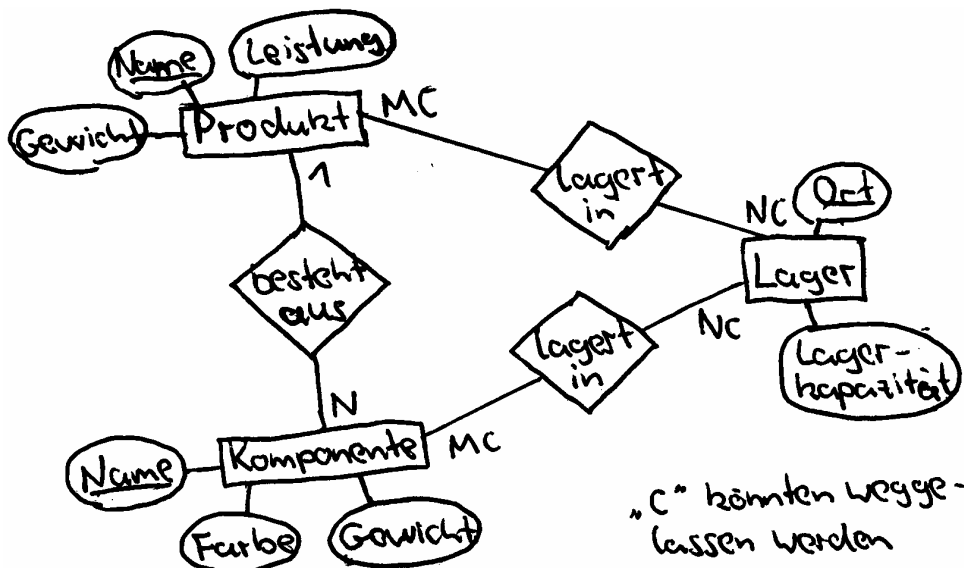
Hinweis: Es handelt sich um Beispielaufgaben, es ist also kein Schluss möglich, dass nicht genannte Themen nicht vorkommen oder dass genannte Themen nur in dieser Form abgefragt werden können. Insbesondere können sich Schwerpunkte verschieben. Es werden 45 der 90 Punkte zum Bestehen der Klausur benötigt. **(Letzte Korrekturänderungen in rot)**
Zugelassene Hilfsmittel: keine.

1) ER-Diagramm erstellen (5+3+3 = 11 Punkte)

Formulieren sie folgende Sachverhalte als ER-Diagramm (Entitäten, Relationen, Attribute, Schlüsselkandidat, Kardinalitäten der Relationen). Falls sie keine Attribute im Text identifizieren können, geben sie zumindest immer ein Schlüsselattribut an.

- Eine Firma stellt verschiedene Produkte her, die sich im Namen unterscheiden und jeweils ein Gewicht und eine Leistung haben. Jedes Produkt besteht aus mindestens zwei verschiedenen Komponenten, die einen eindeutigen Namen, eine Farbe und ein Gewicht haben. Jede Komponente wird nur in einem Produkt benutzt. Beliebige Komponenten können in beliebigen Lagern gelagert werden. Die Lager sind durch ihren eindeutigen Ort bekannt und haben eine eigene Lagerkapazität. Jedes der Produkte kann auch in mehreren Lagern aufbewahrt werden, wobei die Lager auch verschiedene Produkte aufnehmen können.
- In einer Bibliothek gibt es „Buchtitel“ und „Buchexemplare“. Für einen Buchtitel können mehrere Exemplare vorhanden sein, jedoch immer mindestens eins. Ausleiher leihen Buchexemplare. Ausleiher können Buchtitel vormerken.
- Jede Vorlesung wird von einem Professor gehalten. Ein Professor hält mehrere Vorlesungen. Ein Student besucht mehrere Vorlesungen. Eine Vorlesung wird von mehreren Studenten besucht, aber erst nach Semesterbeginn steht fest, von wem. Ein Professor empfiehlt für eine bestimmte Vorlesung ein Buch.

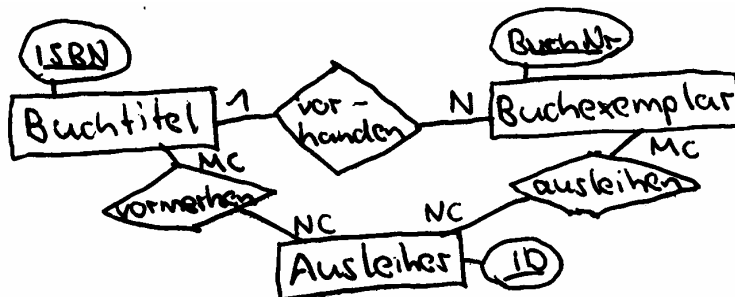
Zu a)



Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 2 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; text-align: center;"> nicht abzugeben, dient dem Selbsttest </div>		

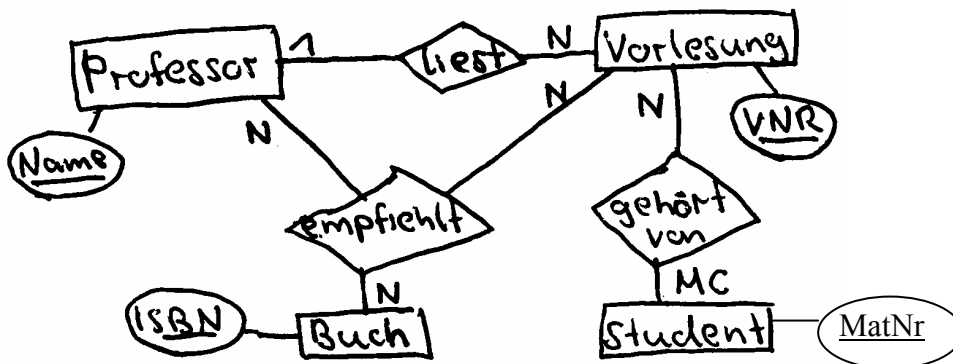


Zu b)



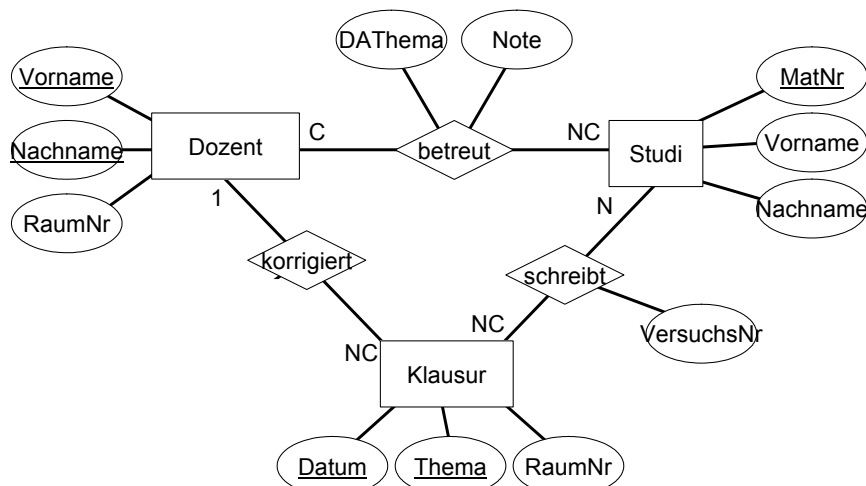
Zu c)

Anmerkung: Die Formulierung mit dem einen Buch ist unglücklich (wurde unbesehen aus einem Beispiel im Internet übernommen, also mangelnde Qualitätskontrolle beim Dozenten), deshalb sind auch andere Lösungen möglich mit zwei Beziehungen (Profssor-Buch, Vorlesung-Buch). Würde nach mehreren Büchern gefragt gäbe es nur die folgende Lösung (man beachte, dass die Angabe der Kardinalitäten bei empfiehlt schwierig sind, da ein Professor für verschiedene Veranstaltungen verschiedene Bücher und ein Buch zu verschiedenen Veranstaltungen empfohlen werden kann):



2) ER-Diagramm in Relationen (Tabellen) übersetzen (7 Punkte)

Gegeben sei das folgende ER-Diagramm:



Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 3 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; text-align: center;"> nicht abzugeben, dient dem Selbsttest </div>		



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.

Dozent: Vorname | Nachname | RaumNr
 Studi: MatNr | Vorname | Nachname
 Klausur: Datum | Thema | RaumNr | Vorname | Nachname
 Klausurversuch: Datum | Thema | MatNr | VersuchsNr
 Betreuung: MatNr | Vorname | Nachname | DAThema | Note

3) Relationen in Normalform (1+4+3 = 8 Punkte)

In einer Firma wird folgende Liste zur Verwaltung von Aufträgen genutzt:

AuftragsNr	Eingangsdatum	ProduktNr	ProduktName	Preis	Lieferant
1	11.02.2003	10	Reis	0.51	Meier
1	11.02.2003	12	Mais	0.53	Müller
2	11.02.2003	10	Reis	0.51	Meier
2	11.02.2003	11	Hirse	0.51	Schmidt

Dabei gelten folgende funktionale Abhängigkeiten:

- (1) {AuftragsNr} -> {Eingangsdatum}
- (2) {ProduktNr} -> {ProduktName, Preis, Lieferant}
- (3) {Produktname} -> {Preis, Lieferant}

Weitere funktionale Abhängigkeiten lassen sich aus den angegebenen Abhängigkeiten berechnen, weiterhin sind die Determinanten auf der linken Seite minimal.

- a) Geben sie einen möglichen Schlüsselkandidaten für die Tabelle an.
- b) Formen sie die Tabelle in Tabellen in 2NF und dann in 3NF um, die die gleiche Aussagekraft haben. Begründen sie kurz, warum ihre Tabellen in 2NF und in 3NF sind.
- c) Was würde sich bei ihren Tabellen ändern, wenn es mehrere Lieferanten für ein Produkt geben würde, also (3) durch {Produktname, Lieferant} -> {Preis} ersetzt würde?

Zu a)

{AuftragsNr, ProduktNr}

Zu b)

In 2NF: Auftrag: AuftragsNr | Eingangsdatum
 Produkt: ProduktNr | ProduktName | Preis | Lieferant
 Auftragsliste: AuftragsNr | ProduktNr
 Produkt nicht in 3NF, umgeformt:
 Produkt: ProduktNr | ProduktName
 Lieferant: ProduktName | Preis | Lieferant

Zu c)

Die Umformung in die 3NF ergäbe:

Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 4 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; text-align: center;"> nicht abzugeben, dient dem Selbsttest </div>		



Produkt: ProduktNr | ProduktName | Lieferant
Lieferant: ProduktName | Lieferant | Preis

4) Relationen in Normalform (4+1+2+4+2+2+2 = 17 Punkte)

Gegeben sei der folgende Ausschnitt aus einer Projektverwaltungstabelle:

PNr	TelNr	Projekt	PBudget	Rolle	VAbt
1	13	EAI	42	Entwickler	EBUIS
1	13	EAI	42	Entwickler	INFRA
1	13	ODB	15	Spezifizierer	INFRA
1	13	ODB	15	Spezifizierer	SERVICE
2	14	EAI	42	Spezifizierer	EBUIS
2	14	EAI	42	Spezifizierer	INFRA
2	14	WEB	15	Designer	INFRA

Für diese Tabelle gelten folgende Regeln

- I) Jeder Mitarbeiter wird eindeutig durch seine Personalnummer (PNr) identifiziert
- II) Jeder Mitarbeiter hat genau eine Telefonnummer (TelNr), jede Telefonnummer ist genau einem Mitarbeiter zugeordnet
- III) Jeder Mitarbeiter kann an mehreren Projekten (Projekt) mitarbeiten, in jedem Projekt können mehrere (und mindestens ein Mitarbeiter) mitarbeiten
- IV) Der Projektname (Projekt) ist innerhalb der betrachteten Firma eindeutig, d.h. es gibt keine zwei Projekte mit gleichem Namen
- V) Jedem Projekt ist ein Projektbudget (PBudget) zugeordnet
- VI) Jeder Mitarbeiter kann in jedem Projekt verschiedene Rollen annehmen
- VII) Für jedes Projekt gibt es mindestens eine, eventuell auch mehrere verantwortliche Abteilungen (VAbt)

- a) Geben sie vier volle funktionale Abhängigkeiten an, wobei keine der angegebenen aus den anderen abgeleitet werden können
- b) Warum gilt nicht $\{\text{Projekt, VAbt}\} \rightarrow \{\text{Rolle}\}$?
- c) Geben sie sämtliche Schlüsselkandidaten an
- d) Ist die Tabelle in zweiter Normalform? Wenn nicht, transformieren sie die Tabelle in Tabellen in zweiter Normalform
- e) Sind ihre Tabellen aus d) in dritter Normalform? Wenn nicht, transformieren sie die Tabellen in Tabellen in dritter Normalform
- f) Sind ihre Tabellen aus e) in Boyce-Cott-Normalform? Wenn nicht, transformieren sie die Tabellen in Tabellen in Boyce-Cott-Normalform
- g) Sind ihre Tabellen aus f) in vierter Normalform? Wenn nicht, transformieren sie die Tabellen in Tabellen in vierter Normalform

Hinweis: Die Formulierung von VI) ist unglücklich, man kann zwei Interpretationen machen, so dass zwei Lösungen möglich sind. Falls Sie die Übung wiederholen wollen, hier eine bessere Formulierung:

Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 5 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; text-align: center;"> nicht abzugeben, dient dem Selbsttest </div>		



VI) *Jeder Mitarbeiter kann in verschiedenen Projekten verschiedene Rollen annehmen. In jedem Projekt hat ein beteiligter Mitarbeiter genau eine Rolle.*

Da dieser Punkt in einer Klausur eventuell erst nach der Fragezeit entdeckt würde, müssten sie hier eine Annahme treffen. Beachten sie, wenn unter 10% der Leute eine Formulierung für missverständlich halten und eine Annahme treffen, die weder ich noch die restlichen 90% sehen, handelt es sich um eine falsche Annahme mit Konsequenzen bei Punktabzügen.

Version 1 (nicht intendierter, aber trotzdem herauslesbarer Ansatz):

Zu a)

$\{PNr\} \rightarrow \{TelNr\}$

$\{TelNr\} \rightarrow \{PNr\}$

$\{Projekt\} \rightarrow \{PBudget\}$

keine vierte volle funktionale Abhängigkeit (ein Indiz, dass man auf dem Holzweg ist).

Zu b)

Gegenbeispiel mit der 2. und 6. Zeile.

Zu c)

$\{PNr, Projekt, Rolle, VAbt\}$ und $\{TelNr, Projekt, Rolle, VAbt\}$

Zu d)

Nicht in 2NF, resultierende Tabellen:

Mitarbeiter: PNr | TelNr

Projekte: Projekt | PBudget

Projektmitarbeit: PNr | Projekt | Rolle | VAbt

Zu e) f), g):

Diese Normalformen sind alle erfüllt. Bei g) wirr konsequent davon ausgegangen, dass keine weiteren Annahmen gemacht werden, ansonsten wäre eine Analyse wie bei der zweiten Version möglich und man kann Verstöße gegen die 4 NF feststellen (Diese schwammige Bemerkung deutet an, dass eine systematische Untersuchung der 4 NF hier sehr aufwändig ist und eventuell weitere Randbedingungen benötigt.).

Version 2:

Zu a)

$\{PNr\} \rightarrow \{TelNr\}$

$\{TelNr\} \rightarrow \{PNr\}$

$\{Projekt\} \rightarrow \{PBudget\}$

$\{PNr, Projekt\} \rightarrow \{Rolle\}$

Zu b)

Gegenbeispiel mit der 2. und 6. Zeile.

Zu c)

Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 6 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> nicht abzugeben, dient dem Selbsttest </div>		



{PNr, Projekt, VAbt} und {TelNr, Projekt, VAbt}

Zu d)

Nicht in 2NF, resultierende Tabellen:

Mitarbeiter: PNr | TelNr

Projekte: Projekt | PBudget

Projektrolle: PNr | Projekt | Rolle

Projektmitarbeit: PNr | Projekt | VAbt

Zu e) f): Die Anforderungen von 3NF und BCNF sind alle erfüllt.

Zu g)

Die Tabelle Projektmitarbeit ist zu analysieren, für die anderen ist 4NF trivial.

Hier mal eine genaue Untersuchung, als Lösung reicht hier das Endergebnis.

Gilt PNr ->> VAbt ? Nein, dazu PNr=1, und Projekt=EAI VAbt aus {EBUIS, INFRA} und zu PNr=1 und Projekt=ODB VAbt aus {INFRA,SERVICE} gehört (Mengen bei VAbt müssten gleich sein).

Gilt PNr ->> Projekt? Nein, aus 1,EBUIS schließt man auf {EAI}, aus 1,INFRA auf {EAI,ODB}

Gilt Projekt ->> VAbt? Ja, aus EAI,1 schließt man auf {EBUIS, INFRA} und aus EAI,2 auf die gleiche Menge. Andere Kombinationen aus Projekt und PNr bringen keine neuen Erkenntnisse, da es sonst keine zwei PNr zum gleichen Projekt gibt. Anschaulich: Die Projekte sind Abteilungen zugeordnet, diese Zuordnung ist unabhängig vom Mitarbeiter.

Gilt Projekt ->> Pnr? Ja, aus EAI,EBUIS schließt man auf {1,2} aus EAI,INFRA auf die gleiche Menge. Bei ODB,INFRA schließt man auf {1}, bei ODB, SERVICE auch auf {1}. Andere Kombinationen aus Projekt und VAbt bringen keine neuen Erkenntnisse, da es sonst keine zwei VAbt zum gleichen Projekt gibt. Anschaulich: Den Projekten sind Mitarbeiter zugeordnet, diese Zuordnung ist unabhängig von der für das Projekt verantwortlichen Abteilung.

Gilt VAbt ->> Pnr? Nein, aus INFRA,ODB schließt man auf {1}, aus INFRA,EAI auf {1,2}

Gilt VAbt ->>Projekt? Nein, aus INFRA,1 schließt man auf {EAI,ODB}, aus INFRA,2 auf {EAI,WEB}

Abteilungsverantwortung: Projekt | VAbt

Projektmitarbeit: Projekt | PNr

Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 7 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> nicht abzugeben, dient dem Selbsttest </div>		



5) SQL-Anfragen formulieren (2+3+3+3+3+3+3 = 20 Punkte)

Bon		
BonID	Kasse	Verkaeuer
1	1	Müller
2	2	Meier
3	2	Müller

Bonposition			
BPID	BonID	ProdID	Anzahl
1	1	1	2
2	1	2	1
3	1	3	5
4	2	1	3
5	2	4	4
6	3	1	2
7	3	3	3

Produkt		
ProdID	Pname	Preis
1	Wodka	5
2	Whisky	7
3	Cola	2
4	Wasser	1

Gegeben seien die oben gezeigten Tabellen für Supermarktbons.

Formulieren sie die folgenden Textzeilen jeweils als SQL-Anfragen.

- Geben sie die Namen aller Produkte (Pname) aus, die verkauft werden können.
- Geben sie die Namen aller Produkte (Pname) aus, die auf dem Bon mit der BonID 1 verkauft werden.
- Geben sie eine Liste aller Einzelverkäufe (Anzahl, Pname) aus, die Müller durchgeführt hat.
- Geben sie eine Liste aller Produkte (Pname) aus, die auf mindestens zwei verschiedenen Bons verkauft werden.
- Geben sie die Namen aller Produkte (Pname) aus, die auf dem Bon mit der BonID 1, aber nicht auf dem Bon mit der BonID 2 stehen.
- Geben sie für jedes Produkt aus, wie häufig es insgesamt (Pname, Gesamtzahl) verkauft wurde.
- Geben sie für jeden Bon die Gesamtsumme aller Verkäufe (BonID, Gesamtsumme) aus.

--a)

```
SELECT Produkt.Pname
FROM Produkt;
```

--b)

```
SELECT Produkt.Pname
FROM Bonposition, Produkt
WHERE Bonposition.ProdID=Produkt.ProdID
AND Bonposition.BonID=1;
```

--c)

```
SELECT Bonposition.Anzahl, Produkt.Pname
FROM Bon, Bonposition, Produkt
WHERE Bonposition.BonID=Bon.BonID
AND Bonposition.ProdID=Produkt.ProdID
AND Bon.Verkaeuer='Müller';
```


Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 8 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> nicht abzugeben, dient dem Selbsttest </div>		



--d)

```
SELECT DISTINCT Produkt.Pname
FROM Bonposition b1, Bonposition b2, Produkt
WHERE b1.BonID<b2.BonID
      AND b1.ProdID=b2.ProdID
      AND b1.ProdID=Produkt.ProdID;
```

--e) (schöne Alternative über Mengendifferenzen)

```
SELECT Produkt.Pname
FROM Produkt
WHERE Produkt.ProdID IN (SELECT Bonposition.ProdID
                        FROM Bonposition
                        WHERE Bonposition.BonID=1)
      AND NOT(Produkt.ProdID IN (SELECT Bonposition.ProdID
                                FROM Bonposition
                                WHERE Bonposition.BonID=2));
```

-- f)

```
SELECT Produkt.Pname, SUM(Bonposition.Anzahl)
FROM Produkt, Bonposition
WHERE Produkt.ProdID = Bonposition.ProdID
GROUP BY Produkt.Pname;
```

-- g)

```
SELECT Bon.BonID, SUM(Bonposition.Anzahl*Produkt.Preis)
FROM Bon, Bonposition, Produkt
WHERE Bon.BonID=Bonposition.BonID
      AND Produkt.ProdID=Bonposition.ProdID
GROUP BY Bon.BonID;
```

6) Tabelle in SQL definieren (6 Punkte)

Geben sie den SQL-Befehl zur Erzeugung der Tabelle Bonposition aus der Aufgabe (5) in SQL an (nur **CREATE**, kein **INSERT**). Neben den dort „sichtbaren“ Randbedingungen, sollen folgende Bedingungen aufgenommen werden:

- Kein Eintrag darf leer sein.
- Die Anzahl muss größer-gleich eins sein.
- Die BPID darf nie 13 sein.
- Wenn die ProdID 3 ist, muss die Anzahl größer als 2 sein.

```
CREATE TABLE Bonposition(
  BPID NUMBER,
  BonID NUMBER NOT NULL,
  ProdID NUMBER NOT NULL,
  Anzahl NUMBER NOT NULL,
  PRIMARY KEY (BPID) ,
```


Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 9 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> nicht abzugeben, dient dem Selbsttest </div>		



```

FOREIGN KEY (BonID) REFERENCES Bon(BonID) ,
FOREIGN KEY (ProdID) REFERENCES Produkt(ProdID) ,
CONSTRAINT CAnzahl CHECK (Anzahl>=1) ,
CONSTRAINT CBPID CHECK (BPID<>13) ,
CONSTRAINT CProdID CHECK (NOT(ProdID=3) OR Anzahl>2)
);

```

7) Auswertung von DB-Anfragen (2+2+3 = 7 Punkte)

Gegeben seien folgende SQL-Anfragen an die Datenbank aus Aufgabe (5).

- a)

```

SELECT DISTINCT b1.Kasse
FROM Bon b1, Bon b2
WHERE b1.BonID<b2.BonID
      AND b1.Kasse=b2.Kasse;

```
- b)

```

SELECT Bon.Kasse, SUM(Bonposition.Anzahl)
FROM Bon, Bonposition
WHERE Bon.BonID=Bonposition.BonID
GROUP BY Bon.Kasse;

```
- c)

```

SELECT Produkt.Pname
FROM Produkt
WHERE NOT EXISTS(
    SELECT *
    FROM Bon
    WHERE NOT(Bon.BonID IN
        (SELECT Bonposition.BonID
         FROM Bonposition
         WHERE Bonposition.ProdID=Produkt.ProdID)) );

```

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

```

-- a) An welchen Kassen wurden mindestens zwei (verschiedene)
-- Bons erstellt
KASSE
-----
      2

```

```

-- b) Wie viele Produkte wurden an den jeweiligen Kassen
verkauft
KASSE  SUM(BONPOSITION.ANZAHL)
-----
      1                8
      2               12

```

Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 10 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> nicht abzugeben, dient dem Selbsttest </div>		



-- c) Welches Produkt wurde auf allen Bons verkauft

PNAME

Wodka

8) Einige Fragen zu Datenbanken (3+3+2+2+2+2 = 14 Punkte)

- Beschreiben Sie kurz die drei grundlegenden Architektur-Ebenen eines Datenbanksystems (nach ANSI/X3/SPARC).
- Nennen sie drei zentrale Funktionalitäten eines Datenbank-Managementsystems und geben sie drei praxisnahe Beispiele, wie diese Funktionalitäten genutzt werden.
- Wie kann man in einer Tabelle für eine bestimmte Spalte zählen, wie oft der Wert NULL in dieser Spalte steht?
- Gehen sie davon aus, dass alle FOREIGN KEYS für die Tabellen in der Aufgabe (5) ohne ON DELETE CASCADE definiert sind. Was passiert, wenn man den Bon mit der BonID 1 löschen will? Was passiert, wenn man die letzte in der Tabelle Bonposition angegebene Zeile löschen will?
- Gehen sie davon aus, dass alle FOREIGN KEYS für die Tabellen in der Aufgabe (5) mit ON DELETE CASCADE definiert sind. Was passiert, wenn man den Bon mit der BonID 1 löschen will? Was passiert, wenn man die letzte in der Tabelle Bonposition angegebene Zeile löschen will?
- Was passiert, wenn für die Tabellen aus der Aufgabe (4) ein DROP TABLE Bonposition ausgeführt wird? Was passiert, wenn für die Tabellen aus der Aufgabe (5) ein DROP TABLE Bonposition CASCADE CONSTRAINTS ausgeführt wird?

Zu a)

Die drei Architektur-Ebenen sind:

- (1) Externes Modell: Dies ist die Benutzersicht auf die Daten.
- (2) Konzeptionelles Modell: Dies ist das logische Datenmodell.
- (3) Internes Modell: Dies definiert die Speicherstruktur der Daten.

Zu b) [ausführlich, drei Punkte wären zu nennen, weitere auf den VL-Folien]

- Integration
 - einheitliche Verwaltung aller Daten
 - Möglichkeit zur redundanzfreien Datenhaltung, z.B. gibt es einen eindeutigen Definitionsort für Matrikelnummern
- Operationen
 - Speichern, Suchen, Ändern, Löschen
 - Daten können von einer Applikation gespeichert werden. Terminiert die Applikation und wird sie erneut aufgerufen, hat sie die Möglichkeit, die vorher abgespeicherten Daten zu lesen und zu verändern.
- Benutzersichten
 - Unterschiedliche Anwendungen benötigen unterschiedliche Sichten
 - z.B. nur bestimmte Teildaten, z.B. bestimmte Übersichten

Thema:	Datenbanken, Selbsttest für das zweite Semester		
Dozent:	Kleuker	Seitennummer:	Seite 11 von 11
Studiengang:	Wirtschaftsinformatik	Jahrgang:	I03
Datum:	31.05.2004	Bearbeitungszeit:	75 Minuten
Matrikelnummer:	<div style="border: 1px solid black; padding: 5px; text-align: center;"> nicht abzugeben, dient dem Selbsttest </div>		



- Bei der Projektverwaltung kann die Mitarbeiternummer hilfreich sein, es muss aber kein Zugriff auf das eventuell in der gleichen Tabelle wie die Nummer gespeicherten Gehalt möglich sein
- Zugriffskontrolle
 - welcher Nutzer darf auf welche Daten in welcher Form zugreifen
 - Grundrechte zur Tabelleneinrichtung werden von Datenbankadministratoren und Projektadministratoren wahrgenommen
- Konsistenzüberwachung (Integritätssicherung)
 - Garantie der Korrektheit bei Datenbankänderungen
 - z.B. abhängige Daten werden mit verändert, z.B. nach einer Namensänderung wird weiterhin überall der korrekte Kundenname genutzt

Zu c)

```
SELECT COUNT(*)
FROM <Tabelle>
WHERE <Tabelle.Spalte> IS NULL;
```

Zu d)

Wenn man den Bon mit der BonID 1 löschen möchte, wird eine Fehlermeldung ausgegeben, da dieser Datensatz noch in weiteren Tabellen benutzt wird. Wenn die letzte Zeile in der Tabelle Bonposition gelöscht wird, wird dies ohne Fehlermeldung getan, da dieser Datensatz zwar auf Daten aus anderen Tabellen zugreift, aber nicht für diese benötigt wird.

Zu e)

Wenn die Tabellen mit ON DELETE CASCADE definiert sind, können die Datensätze ohne Probleme gelöscht werden. Wenn der Bon mit der BonID 1 gelöscht wird, werden auch alle mit diesem Datensatz zusammenhängenden Datensätze in anderen Tabellen gelöscht. Wenn die letzte Zeile in der Tabelle Bonposition gelöscht wird, wird dies wiederum ohne Fehlermeldung getan.

Zu f)

Wenn für die Tabellen aus Aufgabe (4) ein DROP TABLE Bonposition ausgeführt wird, würde eine Fehlermeldung ausgegeben, da diese Tabelle gar nicht existiert. Wenn für Aufgabe (5) ein DROP TABLE Bonposition CASCADE CONSTRAINTS ausgeführt wird, wird diese Tabelle immer gelöscht. Bei anderen Tabellen, die auf diese Tabelle in Fremdschlüsseln zugreifen, wird diese Integritätsbedingung gelöscht.