

### **Lösung Aufgabe 1: Datenbank, DBMS, DBS**

**(7 Punkte)**

*integriert:* Die Daten sind entsprechend der natürlichen Zusammenhänge in der Anwendungswelt strukturiert - und nicht danach, wie einzelne Anwendungen die Daten benötigen. (0,5 Punkte)

*gemeinsame Basis:* Die Daten in der Datenbank können durch viele Benutzer genutzt werden, wobei sie dann unterschiedliche Sichten auf diese Daten haben werden. (0,5 Punkte)

*Datenbankmanagementsystem:* Das Datenbankmanagementsystem (DBMS) ist ein Softwaresystem, das es ermöglicht, eine Datenbank zu definieren, Daten zu speichern, zu verändern und zu löschen, so wie Anfragen an die Datenbank zu stellen. (1 Punkt)

*Datenbanksystem:* Datenbank und Datenbankmanagementsystem (DBMS) bilden zusammen ein Datenbanksystem. (1 Punkt)

*Datenunabhängigkeit:* (insgesamt 3 Punkte)

Datenunabhängigkeit bedeutet, dass Anwendungsprogramme von Änderungen auf der internen und der konzeptuellen Ebene unberührt bleiben. Man unterscheidet physische und logische Datenunabhängigkeit. (1 Punkt)

*Physische Datenunabhängigkeit* bedeutet Isolierung der Anwendungsprogramme vor Änderungen der physischen Datenorganisation. (1 Punkt)

*Logische Datenunabhängigkeit* bedeutet Isolierung der Anwendungsprogramme vor Änderungen des konzeptuellen Modells. (1 Punkt)

*Integrität der Daten.* Korrektheit und Vollständigkeit der abgespeicherten Daten (= beides zu messen an der Realität, über die die Datenbank Daten enthält). (1 Punkt)

## **Lösung Aufgabe 2: Probleme traditioneller Datenverwaltung (7 Punkte)**

*Redundanz:* (für die Nennung des Begriffs: 0,5 Punkte).

Da die Daten jeweils speziell für bestimmte Anwendungen entworfen werden, werden dieselben Daten in verschiedenen Dateien wieder auftauchen (z.B. Namen, Adressen). Redundanz führt zu Speicherverschwendung und zu erhöhten Verarbeitungskosten, vor allem bei Änderungen. Schlimmer jedoch ist es, daß diese Redundanz in der Regel nicht zentral kontrolliert wird, so daß Konsistenzprobleme auftreten. (2 Punkt).

*Inkonsistenz:* (für die Nennung des Begriffs: 0,5 Punkte).

Die Konsistenz der Daten (d.h. die logische Übereinstimmung der Datei-Inhalte) kann nur schwer gewährleistet werden, es ist also problematisch zu vermeiden, dass verschiedene Programme zum selben Zeitpunkt unterschiedliche Werte derselben Größe sehen können. (1 Punkt)

*Daten-Programm-Abhängigkeit:* (für die Nennung des Begriffs: 0,5 Punkte).

Ändert sich der Aufbau einer Datei oder ihrer Organisationsform, so müssen darauf basierende Programme geändert werden. (1 Punkt)

*Inflexibilität:* (für die Nennung des Begriffs: 0,5 Punkte).

Da die Daten nicht in ihrer Gesamtheit sondern nur anwendungsbezogen gesehen werden, ist es in vielen Fällen sehr kompliziert, neue Anwendungen oder Auswertungen vorhandener Daten zu realisieren. Dies gilt insbesondere für Auswertungen, die Daten aus verschiedenen Dateien benötigen würden. Die Organisation nach diesem konventionellen Vorgehen ist sehr wenig anpassungsfähig an die sich verändernden Anforderungen in einem Unternehmen. (1 Punkt)

**Lösung Aufgabe 3: Datenbankmanagementsystem****(9 Punkte)**

*eigenständige Datenmanipulationssprachen:* Eigenständige DML heißen auch *Abfragesprachen (Query Language)*. Dies sind einfache Programmiersprachen speziell zum Arbeiten mit der Datenbank. (0,5 Punkte)

*nicht-eigenständigen Datenmanipulationssprachen:* Hierbei muß der Anwender eine übliche Programmiersprache (COBOL, PL/I, usw.) benutzen, lediglich für Zwecke des Zugriffs auf die Datenbank stehen ihm DML-Befehle zur Verfügung. (0,5 Punkte)

Die Abarbeitung des DML-Befehls geht wie folgt vor sich:

- (1) Übergabe des DML-Befehls an das DBMS. (0,5 Punkte)
- (2) Interpretation des DML-Befehls durch das DBMS; Ermittlung der zugehörigen konzeptuellen Beschreibung. (0,5 Punkte)
- (3) Ermittlung der Speicherungsstruktur und der möglichen Zugriffspfade auf den gesuchten Satz (internes Schema). (0,5 Punkte) Optimierung der Abfrage: Festlegung der Folge der Abarbeitungsschritte. (0,5 Punkte)
- (4) Das DBMS ermittelt die Datenseite (Page), auf der der gesuchte Satz gespeichert ist. (0,5 Punkte) Es prüft, ob sich diese Datenseite bereits im Systempuffer befindet. Falls nicht, muß auf die Datenbank zugegriffen werden; sonst weiter mit (8). (0,5 Punkte)
- (5) Auswahl einer Seite im Systempuffer, die durch die benötigte Seite überlagert werden kann. (0,5 Punkte) Falls die zu ersetzende Seite verändert wurde, muß sie in die Datenbank geschrieben werden. (0,5 Punkte)
- (6) Das DBMS ruft das Betriebssystem für zwei E/A-Vorgänge auf:
  - Schreiben der zu ersetzenden Seite in die Datenbank (entfällt gegebenenfalls) (0,5 Punkte)
  - Einlesen der gesuchten Seite. (0,5 Punkte)
- (7) Das Betriebssystem führt die physischen E/A-Aufträge durch und speichert die angeforderte Seite an der vorgegebenen Adresse im Systempuffer. (0,5 Punkte)
- (8) Das DBMS liest den gesuchten Satz aus dem Systempuffer, transformiert ihn in die durch das externe Schema definierte Form und überträgt ihn in den Arbeitsbereich (user work area - UWA) des Anwendungsprogrammes. In der UWA ist ein Speicherbereich für diesen Satztyp reserviert. (1,5 Punkt)
- (9) Das DBMS hinterlegt Status-Information über den Ausgang der Operation in einem speziellen Bereich der UWA. Diese Status-Information ist dem Anwendungsprogramm zugänglich. (0,5 Punkte)
- (10) Das Anwendungsprogramm verarbeitet den Satz. (0,5 Punkte)

**Lösung Aufgabe 4: Entity-Relationship-Modell****(12 Punkte)**

1. CREATE TABLE Kunde (KDNR INTEGER NOT NULL,  
Name CHAR(30),  
Anschrift CHAR(50),  
PRIMARY KEY (KDNR))
2. CREATE TABLE Ware (WNR INTEGER NOT NULL,  
Bezeichnung CHAR(30),  
Preis INTEGER,  
HNR INTEGER NOT NULL,  
PRIMARY KEY (WNR),  
FOREIGN KEY (HNR) REFERENCES Hersteller)
3. CREATE TABLE Warenhaus (WHNR INTEGER NOT NULL,  
Name CHAR(30),  
Anschrift CHAR(50),  
HNR INTEGER NOT NULL,  
PRIMARY KEY (WHNR),  
FOREIGN KEY (HNR) REFERENCES Hersteller)
4. CREATE TABLE Hersteller (HNR INTEGER NOT NULL,  
Name CHAR(30),  
Anschrift CHAR(50),  
PRIMARY KEY (HNR))
5. CREATE TABLE bezieht (KDNR INTEGER NOT NULL,  
WNR INTEGER NOT NULL,  
PRIMARY KEY (KDNR,WNR),  
FOREIGN KEY (KDNR) REFERENCES Kunde,  
FOREIGN KEY (WNR) REFERENCES Ware)
6. CREATE TABLE kauft\_bei (KDNR INTEGER NOT NULL,  
WHNR INTEGER NOT NULL,  
PRIMARY KEY (KDNR,WHNR),  
FOREIGN KEY (KDNR) REFERENCES Kunde,  
FOREIGN KEY (WHNR) REFERENCES Warenhaus)

(je 2 Punkte)

**Aufgabe 5: SQL****21 Punkte**

- a) Ein Entity ist ein Begriff aus dem ER-Modell. Ein Entity ist ein Objekt der realen Welt, das für sich allein identifizierbar ist und für den betrachteten Ausschnitt der realen Welt von Relevanz ist. Ein Entity-Typ repräsentiert die Menge aller Entities, die die gleichen charakteristischen Eigenschaften besitzen. Ein Entity-Typ bezeichnet sowohl einen Typ, also eine Klassifizierung von Objekten, als auch die Menge der Objekte. (2 Punkte)
- b) Eine Tupelvariable kann genau ein Tupel einer Relation aufnehmen. Die Verwendung von Tupelvariablen ist erforderlich, wenn innerhalb einer Abfrage verschiedene Tupel derselben Relation betrachtet werden. (1 Punkt)
- c) Die Sicht eines Benutzers auf die Daten im Datenbanksystem heißt View. (1 Punkt)
- d) Eine im System physikalisch vorhandene Tabelle wird als Basistabelle bezeichnet (im Gegensatz zu einem View, der auch als abgeleitete Tabelle bezeichnet wird). (1 Punkt)
- e) (3 Punkte)

<b>B</b>	<b>A</b>
40	20
100	80

- f) 160 (2 Punkte)
- g) 99 (2 Punkte)
- h) `select distinct tupel1.a, tupel1.b, tupel1.c, tupel1.d from r2 tupel1, r2 tupel2`  
`where tupel1.a < tupel2.a+10 and tupel1.d > tupel2.d+10`  
 Das Ergebnis ist:

<b>14</b>	<b>176</b>	<b>196</b>	<b>208</b>
-----------	------------	------------	------------

(5 Punkte)

- j) `select b from r1`  
`where a > (select avg(a) from r1) and`  
`b > (select avg(b) from r1) and`  
`c > (select avg(c) from r1) and d > (select avg(d) from r1)`  
 Das Ergebnis ist 160. (4 Punkte)

**LösungAufgabe 6: Funktionale Abhängigkeiten, Normalformen (14 Punkte)**

Definitionen (4 Punkte)

*erste Normalform:* Eine Relation ist in erster Normalform, wenn die Werte der Wertebereiche jedes Attributs unteilbare Werte sind und nicht ihrerseits wieder aus Mengen oder Tupeln bestehen. (1 Punkt)

*zweite Normalform:* Eine Relation  $R$  ist in zweiter Normalform, wenn jedes Nichtschlüsselattribut  $A$  von  $R$  voll funktional abhängig von jedem Schlüssel  $X$  von  $R$  ist. (1 Punkt)

*dritte Normalform:* Ein Relationenschema  $R$  mit Fd-Menge  $F$  ist in dritter Normalform, wenn für alle  $X \twoheadrightarrow A$  aus  $F^+$  mit  $A$  nicht aus  $X$  gilt:  $X$  enthält einen Schlüssel für  $R$  oder  $A$  ist Schlüsselattribut. (1 Punkt)

*Boyce-Codd Normalform (BCNF):* Eine Relation  $R$  ist in Boyce-Codd Normalform, wenn für jede funktionale Abhängigkeit  $X \twoheadrightarrow A$  aus  $F^+$ ,  $A$  nicht aus  $X$ , gilt:  $X$  enthält einen Schlüssel für  $R$ . (1 Punkt)

a)

CDE, ACD, BCD

Lösung erfolgt über Closure von  $F$ , über voll funktional Abhängigkeit sowie über Schlüsseleigenschaften.

- Die Menge  $F^+$  aller Funktionaler Abhängigkeiten, die von  $F$  impliziert werden, heißt Closure von  $F$ .
- Für eine Fd-Menge  $F$  und eine funktionale Abhängigkeit  $X \twoheadrightarrow Y \in F^+$  heißt  $Y$  *voll funktional abhängig von  $X$* , genau dann wenn es keine echte Teilmenge  $X'$  von  $X$  gibt, so daß  $X' \twoheadrightarrow Y \in F^+$ .
- $X$  ist *Schlüssel* von  $\{A_1, \dots, A_n\}$  genau dann, wenn  $X \twoheadrightarrow \{A_1, \dots, A_n\} \in F^+$  und  $\{A_1, \dots, A_n\}$  ist voll funktional abhängig von  $X$ .

b)

$R$  ist in 3NF, da  $B$ ,  $E$  und  $A$  Teile der Schlüssel sind.

- Ein Relationenschema  $R$  mit Fd-Menge  $F$  ist in *dritter Normalform*, wenn für alle  $X \twoheadrightarrow A \in F^+$  mit  $A \notin X$  gilt:  $X$  enthält einen Schlüssel für  $R$  oder  $A$  ist Schlüsselattribut.

c)  $R$  ist nicht in BCNF, da weder  $A$ ,  $BS$  noch  $ED$  einen Schlüssel besitzen.

- Eine Relation  $R$  ist in *Boyce-Codd Normalform*, wenn für jede funktionale Abhängigkeit  $X \twoheadrightarrow A \in F^+$ ,  $A \notin X$  gilt:  $X$  enthält einen Schlüssel für  $R$ .

**Lösung Aufgabe 7: Relationenalgebra****(19 Punkte)**

a) In der *Relationenalgebra* spezifiziert man gewünschte Relationen durch Angabe einer Folge von Operationen, mit der die Relationen aufgebaut werden sollen. Der Benutzer wendet spezielle Operationen auf Relationen an, um seine gewünschte Relation zu konstruieren. (1,5 Punkte)

Im *Relationenkalkül* spezifiziert man hingegen gewünschte Relationen in deskriptiver Weise, d.h. ohne Angabe, welche Operationen zum Aufbau der Relation verwendet werden sollen. Mit Hilfe des Prädikatenkalküls wird die Menge der gewünschten Tupel beschrieben: es wird ein Prädikat (eine Bedingung) angegeben, das die Tupel erfüllen müssen. (1,5 Punkte)

b)

$R2[E=x]$	E	F	G
	x	10	b

(2 Punkte)

c)

$(R1[A,B] \mid B=G) \mid (R2[G])$	A	B	G
	1	c	c
	2	a	a
	3	a	a
	4	b	b

(2 Punkte)

d)

$(R1[B,C] \mid C=E) \mid R2[E] \mid B \mid B=G \mid R2$	B	E	F	G
	c	y	20	c
	b	x	10	b
	a	y	30	a

(2,5 Punkte)

e)

$((R1[D=F] \mid R2[A,B]) \mid B=G) \mid R2$	A	B	E	F	G
	1	c	y	20	c
	2	a	y	30	a
	3	a	y	30	a
	4	b	x	10	b

(2,5 Punkte)

a) (3 Punkte)

Range SCHIFF X  
{SCHIFF.S\_NAME, SCHIFF.TIEFGANG |  $\forall X: (X.TIEFGANG \geq$   
SCHIFF.TIEFGANG)}

b) (4 Punkte)

Range ROUTE R  
Range LADUNG L  
{SCHIFF.S\_NAME, SCHIFF.HEIMATHAFEN |  $\exists R: (R.S\_NAME = SCHIFF.S\_NAME$   
 $\wedge R.ZIELHAFEN = "HAMBURG" \wedge \exists L: (L.S\_NAME = R.S\_NAME \wedge$   
L.HEZEICHNUNG = "Kohle"))}