

Gruppe B

Bitte tragen Sie **SOFORT** Namen und Matrikelnr. ein und legen Sie Ihren Studentenausweis bereit.

ÜBUNGSTEST AUS DATENBANKSYSTEME			27. 1. 2000
Kennnr.	Matrikelnr.	Familienname	Vorname
<input type="radio"/> Ja, ich habe Interesse, nächstes Jahr Tutor zu sein. Adresse/Telefon/email:			Habe die Übungsbeispiele im Studienjahr abgegeben.

Arbeitszeit: 90 Minuten. Die Aufgaben sind auf den Angabeblättern zu lösen.

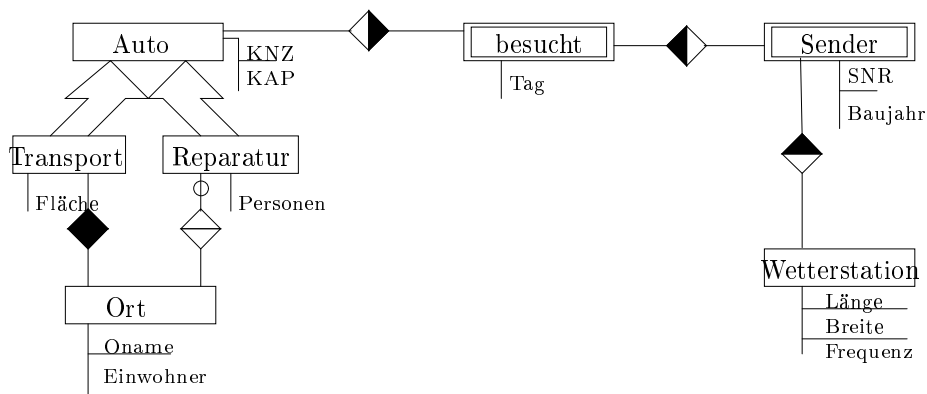
Aufgabe 1:

(6)

Gegeben ist folgendes EER-Diagramm der österreichischen Metereologieorganisation.

Leiten Sie aus dem EER-Diagramm die Relationen der Datenbank in 3.Normalform so ab, daß sie verbundtreu und abhängigkeits-treu sind. Führen Sie keine zusätzlichen Attributnamen ein, außer ein Attribut kommt mehrfach in einer Relation vor.

(**Hinweis:** Es werden nicht notwendigerweise alle Lücken gefüllt.)



()
()
()
()
()
()
()
()
()

Aufgabe 2:

(6)

Für die Verwaltung der Knochensammlung des Naturhistorischen Museums ist ein EER-Diagramm zu entwerfen. Verwenden Sie nur die vorgegebenen Attributnamen. (Diese sind nur bei ihrer jeweils ersten Erwähnung angeführt.)

1. Von jedem Sachbearbeiter des Museums wird die Sozialversicherungsnummer (SVNR) und der Name (NAME) gespeichert.
2. Jeder Knochen hat eine eindeutige Seriennummer (SNR), eine Bezeichnung (BEZEICHNUNG) und ein Datum (DATUM), an dem er gefunden wurde. Wurde der Knochen bereits eindeutig klassifiziert, dann muss die Tierart (ART), von der er abstammt und der Sachbearbeiter, der die Klassifizierung durchgeführt hat, gespeichert werden. Es können verschiedene Klassifizierungen ausgesprochen werden. Knochensplitter werden wie Knochen behandelt, nur dass zusätzlich noch die Länge (L) und Breite (B) gespeichert wird.
3. Skelette bestehen aus vielen Knochen. Jedes Skelett hat eine eindeutige Nummer (NR) und ist von einem bestimmten Tier (TIER). Für jeden Knochen ist angegeben, mit welchen anderen Knochen er bei der Konstruktion des Skeletts zusammengeklebt wurde und welche Sachbearbeiter für die Korrektheit der Konstruktion verantwortlich waren.

Aufgabe 3:

(12)

Als EDV Angestellter eines Versandhauses sollen Sie ein Programm zur Verwaltung von Kunden, Produkten und Verkäufen erstellen. Folgende Relationen haben Sie identifiziert:

<code>produkt(pnr, bez, preis)</code>	<code>kunde(knr, name, adresse, rabatt)</code>
<code>bestellung(bnr, pnr, knr, anz, dat)</code>	<code>rechnung(rnr, legdat, bezdat, bnr)</code>

Von den Produkten ist eine eindeutige Nummer (pnr), die Bezeichnung, sowie der Preis bekannt. Weiters sind die Kunden mit einer eindeutigen Kundennummer, sowie dem Namen, der Adresse und dem gewährten Rabatt gespeichert. Bestellungen haben eine eindeutige Nummer und sind einem Kunden und einem Produkt zugeordnet, die Anzahl der bestellten Produkte sowie das Bestelldatum sind ebenfalls in der Datenbank abgelegt. Zu jeder Bestellung gibt es eine Rechnung mit einer eindeutigen Rechnungsnummer (rnr) sowie dem Datum der Rechnungslegung (legdat). Wenn die Rechnung bezahlt wurde, wird das Datum in bezdat gespeichert, ansonsten ist bezdat gleich NULL.

Vervollständigen Sie die Funktionen `updateDB` und `printErr`. Die Verbindung zur Datenbank wird automatisch aufgebaut und muss von Ihnen nicht programmiert werden. Beachten Sie, dass Ihnen aus Kompatibilitätsgründen zu älteren Oracle Versionen nur die Datentypen `int` und `VARCHAR` zur Verfügung stehen.

1. Geben Sie Namen und Adresse jener Kunden aus, die im Jahr 1999 Produkte im Wert von mehr als 3.000,- ATS bestellt haben. Geben Sie weiters die Summe der bestellten Produkte aus. Erhöhen Sie danach den Rabatt dieser Kunden um jeweils 3%. [5]
2. Geben Sie Namen und Adresse all jener Kunden aus, die nach dem im Parameter `param` übergebenen Datum keine Bestellung mehr getätigt haben. Löschen Sie alle Bestellungen und Rechnungen dieser Kunden, und setzen Sie den Rabatt dieser Kunden auf 0. Erzeugen Sie anschließend eine Relation `schlechtekunden`, in die Sie ALLE Kunden mit keinem Rabatt eintragen. [7]

```
EXEC SQL INCLUDE SQLCA;
updateDB(char *param)
{
    EXEC SQL BEGIN DECLARE SECTION;
        VARCHAR sdat[20];

EXEC SQL END DECLARE SECTION;

EXEC SQL WHENEVER NOT FOUND
EXEC SQL WHENEVER SQLWARNING
EXEC SQL WHENEVER SQLERROR
strcpy(sdat.arr, param);
```

```

}
int printErr()
{

}

int main(int argc, char **argv)
{
    char *dat = "12-31-98";
    updateDB(dat);
}

```

Aufgabe 4:

(6)

Formulieren Sie die folgende SQL-Query! Die Definition der Relationen entnehmen Sie bitte der Aufgabe 3. Für die Statistik ist man interessiert, welche Kunden 1999 einen durchschnittlichen Umsatz pro bezahlter Rechnung haben, der den durchschnittlichen Umsatz ihrer offenen Rechnungen unterschreitet. Geben Sie den Namen, die Adresse des Kunden und die Anzahl der Rechnungen aus. Der Kundenrabatt ist nicht zu berücksichtigen.

Gesamtpunkte: 30