



Vorlesung Einführung in die Informatik: Systeme und Anwendungen im SS 2001

Übungsblatt 6

Organisatorisches:

Abgabetermin: Freitag 29. Juni 2001, 13:00 (strikt)

per Email an die Korrektoren gemäß dem Anfangsbuchstaben desjenigen, der die Email absendet.

A – D	funkc@dbs.informatik.uni-muenchen.de	Caroline Funk
E – J	martinj@dbs.informatik.uni-muenchen.de	Johanna Martin
K – R	reisslh@dbs.informatik.uni-muenchen.de	Harald Reissl
S – Z	wiedeman@dbs.informatik.uni-muenchen.de	Florian Wiedemann

BITTE BEACHTEN!

Die Email muß des Weiteren folgende Informationen enthalten:
Name, Vorname und Matrikelnummer aller Mitglieder des Teams, das die Aufgabe gemeinsam bearbeitet hat.

Schreiben Sie als Subject der Mail: Übungsblatt 6.

Bei Mehrfachabgaben wird die zeitlich erste gewertet und bei Zusendungen an mehrere Korrektoren wird die Abgabe gewertet, welche am schlechtesten bewertet wurde.

BESONDERS WICHTIG

Teilen sie bitte bei der Abgabe der Lösung den Korrektoren mit welche Fachrichtung Sie studieren.

Da je nach Studienrichtung verschiedene Prüfungsvorschriften existieren, müssen wir Sie bei der Klausur gemäß Ihrer Studienrichtung in die verschiedenen Räume verteilen.

Wenn uns Ihre Studienrichtung nicht bekannt ist, können wir Sie nicht zur Klausur zulassen.

Aufgabe 8 (16 Punkte)

Im folgenden wird wieder mit der Beispieldatenbank `beispiel1` gearbeitet.

Wenn Sie die Aufgabe mit dem Datenbanksystem PostgreSQL bearbeiten wollen, können Sie die in der Datei `blatt5.beispiel1.txt` enthaltenen Befehle zur Einrichtung dieser Datenbank benutzen. Sie benötigen diese Befehle außerdem als Ausgangspunkt zur Beantwortung von einigen der folgenden Fragen.

Im Anhang sind ähnliche Anfragen formuliert, an welchen man sich orientieren kann.

- (a) Finden Sie für jeden Auftrag die Auftragsnummer, die Kundennummer und den Namen des Kunden, der den Auftrag gegeben hat.
- (b) Wie (a), wobei nur die Aufträge, die von dem Kunden 'Betten Kaiser' gegeben wurden, ausgewählt werden sollen.
- (c) Finden Sie für jeden (jede) Angestellte(n), der (die) einen Auftrag bearbeitet hat, in dem der Artikel mit Nummer 301001 bestellt wurde, den Nachnamen des (der) Angestellten, und die in dem Auftrag bestellte Menge des Artikels 301001.

Bilden sie bei den folgenden Anfragen keine 'Joins' sondern geschachtelte Anfragen.

- (d) Finden Sie die Auftragsnummer und das Bestelldatum aller Aufträge, die von der Angestellten 'Winter' bearbeitet wurden.
- (e) Finden Sie die Auftragsnummer, das Bestelldatum und die Personalnummer aller Aufträge, die von Angestellten, die in 'HAMBURG' eingesetzt sind, bearbeitet wurden.
- (f) Finden Sie den Vornamen und den Nachnamen jedes (jeder) Angestellten, der (die) einen Auftrag für Kunden aus 'Stuttgart' bearbeitet hat.
- (g) Finden Sie den Vornamen und den Nachnamen aller Angestellten, die momentan keinen Auftrag bearbeitet haben.

Aufgabe 9 (11 Punkte)

Die Definitionen (`create table ...`) der Datenbank `beispiel1` sollen im Folgenden um Integritätsbedingungen erweitert werden.

- (a) Erweitern Sie die Definition der Tabelle `personal` so, dass die Spalte `einsatz` nur einen der Werte „MÜNCHEN,“ „HAMBURG,“ „FRANKFURT“ oder „KÖLN“ annehmen kann.
- (b) Erweitern Sie die Definition der Tabelle `inventar` so, dass nur positive Lagerbestände (d.h. `lagerbest > 0`) vorkommen können.
- (c) Erweitern Sie die Definition der Tabelle `kunde` um den Primärschlüssel `kund_nr` und die Definition der Tabelle `personal` aus (a) um den Primärschlüssel `persnr`.
- (d) Welche Schlüsselkandidaten gibt es für die Tabelle `verkauf`, wenn man davon ausgeht, dass jeder Angestellte nur einen Verkauf pro Kunde und Tag durchführen kann, dass ein Kunde aber bei verschiedenen Angestellten einkaufen kann.
- (e) Erweitern Sie die Definition der Tabelle `verkauf` um den Primärschlüssel `auftr_nr`
- (f) Erweitern Sie die Definition der Tabelle `inventar` aus (b) um den aus den Spalten `art_nr` und `lagerort` bestehenden Primärschlüssel.
- (g) Erweitern Sie die Definition der Tabelle `inventar` aus (f) um den aus den Spalten `art_bez` und `lagerort` bestehenden Alternativ-Schlüssel
- (h) Geben Sie die Fremdschlüssel für die Tabelle `verkauf` an.
- (i) Erweitern Sie die Definition der Tabelle `ausgang` so, dass `auftr_nr` ein Fremdschlüssel ist.

Anhang

(Wird in der Übungstunde am Freitag, den 22. Juni 2001 besprochen.)

Im folgenden wird wieder mit der Beispieldatenbank `beispiel1` gearbeitet.

Verknüpfungen (Joins):

- Finden Sie für jeden Auftrag die Auftragsnummer, die Personalnummer und den Nachnamen des (der) Angestellten, der (die) diesen Auftrag bearbeitet hat.

```
b ) select auftr_nr, personal.persnr, nachname
      from verkauf, personal
      where verkauf.persnr = personal.persnr
```

auftr_nr	persnr	nachname
20002	5	Hartinger
20003	1	Winter
20004	8	Sandner
20005	7	Moll
20006	12	Neumann
20007	12	Neumann
20008	1	Winter
20009	9	Starck
20010	11	Freudenfeld
20011	8	Sandner

- Wie zuvor, wobei nur die Aufträge, die von der Angestellten 'Winter' bearbeitet wurden, ausgewählt werden sollen.

```
select auftr_nr, personal.persnr, nachname
      from verkauf, personal
      where verkauf.persnr = personal.persnr
      and nachname = 'Winter'
```

auftr_nr	persnr	nachname
20003	1	Winter
20008	1	Winter

- Finden Sie für alle in der Tabelle `ausgang` gespeicherten Bestellungen des Artikels mit der Nummer 203333 die Artikelnummer, die bestellte Menge und alle Lagerbestände und Lagerorte, an denen eine ausreichende Stückzahl des Artikels 203333 vorhanden ist.

```
select ausgang.art_nr, menge, lagerbest, lagerort
       from ausgang, inventar
       where ausgang.art_nr = inventar.art_nr
       and lagerbest >= menge
       and ausgang.art_nr = 203333
```

art_nr	menge	lagerbest	lagerort
203333	3	4	FRANKFURT
203333	3	10	HAMBURG
203333	3	6	KOELN
203333	3	4	FRANKFURT
203333	3	10	HAMBURG
203333	3	6	KOELN
203333	2	4	FRANKFURT
203333	2	10	HAMBURG
203333	2	6	KOELN

- *Verknüpfung von 3 Tabellen:* Finden Sie für jeden (jede) Angestellte(n), der (die) einen Auftrag bearbeitet hat, den Nachnamen, die Auftragsnummer aller von dem (der) Angestellten bearbeiteten Aufträge und den Namen des Kunden, der den jeweiligen Auftrag erteilt hat.

```
select nachname, auftr_nr, kund_name
       from personal, verkauf, kunde
       where personal.persnr = verkauf.persnr
       and verkauf.kund_nr = kunde.kund_nr
```

nachname	auftr_nr	kund_name
Hartinger	20002	Ulrich & Co.
Winter	20003	Peter Hauser
Sandner	20004	Manfred Keller GmbH
Moll	20005	Stuhl Kaiser
Neumann	20006	Miriam Zechmeister KG
Neumann	20007	Betten Kaiser
Winter	20008	Stadtverwaltung Landshut
Starck	20009	Planmöbel GmbH
Freudenfeld	20010	Paula Peterson
Sandner	20011	Möller & Partner

- *Selbstverknüpfung*: Finden Sie für jeden (jede) Angestellte(n) den Nachnamen des (der) Angestellten und den Nachnamen des (der) Vorgesetzten.

```
select ang.nachname, vorg.nachname AS vorgesetzer
      from personal ang, personal vorg
      where ang.vorgesetzt = vorg.persnr
```

nachname	vorgesetzer
Winter	Sandner
Thomas	Roser
Scholl	Sandner
Hartinger	Sandner
Reiztig	Moll
Anger	Starck
Freudenfeld	Sandner
Neumann	Roser

Geschachtelte Anfragen (Unterabfragen):

- Finden Sie die Auftragsnummer und das Bestelldatum aller Aufträge des Kunden 'Stuhl Kaiser.'

```
select auftr_nr, bestelldat
      from verkauf
      where kund_nr =
          (select kund_nr
           from kunde
           where kund_name = 'Stuhl Kaiser')
```

auftr_nr	bestelldat
20005	24.07.88

- Finden Sie die Auftragsnummer, das Bestelldatum und die Kundennummer aller Aufträge, die von Kunden aus 'Stuttgart' erteilt wurden.

```
select auftr_nr, bestelldat, kund_nr
      from verkauf
      where kund_nr in
          ( select kund_nr
           from kunde
           where ort = 'Stuttgart')
```

auftr_nr	bestelldat	kund_nr
20003	23.07.88	EK8992
20004	23.07.88	EK5558
20011	24.07.88	GK4441

- Finden Sie die Namen aller Kunden aus 'Frankfurt', die in einem Auftrag den Artikel mit Nummer 301001 bestellt haben.

```
select kund_name
  from kunde
  where ort = 'Frankfurt' and
        kund_nr in
        ( select kund_nr
          from verkauf
          where auftr_nr in
            ( select auftr_nr
              from ausgang
              where art_nr= 301001 ))
```

kund_name

Planmoebel GmbH

- Finden Sie die Namen aller Kunden, die momentan keinen Auftrag erteilt haben.

```
select kund_name
  from kunde
  where kund_nr not in
        (select kund_nr
         from verkauf )
```

kund_name

Stadtverwaltung Köln
Planungsbüro Bertoldt
Freiling & Partner
Einrichtungshaus Roeder
Systemmöbel Wisst & Co.

Integritätsbedingungen:

Die Datenbank biblio kann durch folgende Definitionen erzeugt werden:

```
create table buecher(buch_nr char(4), titel char(30), autor char(30));
create table kunden(kunden_nr char(4), name char(30), ort char(30));
create table verleih(verleih_nr int, kunden_nr char(4),
                    buch_nr char(4), rueckgabe char(8));
```

Diese Definitionen sollen um Integritätsbedingungen erweitert werden.

- Erweitern Sie die Definition der Tabelle verleih so, dass die Spalte verleih_nr nur positive Werte annehmen kann.

```
create table verleih(verleih_nr int,
                    kunden_nr char(4), buch_nr char(4), rueckgabe char(8),
                    check ( verleih_nr > 0));
```

-- Test

```
insert into buecher values ('b042', 'The SQL Standard', 'C.J.Date');
insert into kunden values ('k123', 'Klaus Mueller', 'MUENCHEN');
insert into verleih values (-1, 'k123', 'b042', '03.02.99');
```

- Erweitern Sie die Definition der Tabelle kunden so, dass die Spalte ort nur einen der Werte „MUENCHEN“ oder „AUGSBURG“ annehmen kann.

```
create table kunden (kunden_nr char(4), name char(30), ort char(30),
                    check ( ort in ('MUENCHEN', 'AUGSBURG')));
```

- Erweitern Sie die Definition der Tabelle buecher um den Primärschlüssel buch_nr und die Definition der Tabelle kunden um den Primärschlüssel kunden_nr.

```
create table buecher(buch_nr char(4) primary key,
                    titel char(30), autor char(30));
create table kunden (kunden_nr char(4) primary key,
                    name char(30), ort char(30),
                    check ( ort in ('MUENCHEN', 'AUGSBURG')));
```

-- Test

```
insert into buecher values ('b042', 'An Introduction to Database Systems',
                          'C.J.Date');
```

- Welche Schlüsselkandidaten gibt es für die Tabelle verleih?

```
-- Abhängig von Bedingungen,
-- möglich: {verleih_nr}, {kunden_nr, buch_nr, rueckgabe} oder
--          {verleih_nr}, {kunden_nr, buch_nr}
```


- Erweitern Sie die Tabelle `verleih` aus (a) um den aus `kunden_nr` und `buch_nr` bestehenden Primärschlüssel.

```
create table verleih(verleih_nr int,
                    kunden_nr char(4), buch_nr char(4), rueckgabe char(8),
                    primary key (kunden_nr, buch_nr),
                    check (verleih_nr > 0));
```

- Erweitern Sie die Tabelle `verleih` aus (a) um den Primärschlüssel `verleih_nr` und den aus `kunden_nr` und `buch_nr` bestehenden Alternativ-Schlüssel.

```
create table verleih(verleih_nr int primary key,
                    kunden_nr char(4), buch_nr char(4), rueckgabe char(8),
                    unique (kunden_nr, buch_nr),
                    check (verleih_nr > 0));
```

-- alternativ

```
create table verleih(verleih_nr int,
                    kunden_nr char(4), buch_nr char(4), rueckgabe char(8),
                    primary key (verleih_nr),
                    unique (kunden_nr, buch_nr),
                    check (verleih_nr > 0));
```

eigentlich besser:

=====

```
create table verleih(verleih_nr int
                    kunden_nr char(4) NOT NULL,
                    buch_nr char(4) NOT NULL,
                    rueckgabe char(8),
                    primary key (verleih_nr),
                    unique (kunden_nr, buch_nr),
                    check (verleih_nr > 0));
```

- Geben Sie die Fremdschlüssel für die Tabelle `verleih` aus (f) an.

-- kunden_nr, buch_nr

- Erweitern Sie die Tabelle `verleih` aus (f) um die Fremdschlüssel `kunden_nr` und `buch_nr`.

```
create table verleih(verleih_nr int primary key,
                    kunden_nr char(4), buch_nr char(4), rueckgabe char(8),
                    unique (kunden_nr, buch_nr),
                    foreign key (kunden_nr) references kunden (kunden_nr),
                    foreign key (buch_nr) references buecher (buch_nr),
                    check (verleih_nr > 0));
```