	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2014/15
	Aufgabenzettel	4			
	STiNE-Gruppe 30	Cornelia Hofsäß, Aleksej Davletcurin, Sascha Marcel Hacker			
	Ausgabe	Mi. 15.10.2014	Abgabe	Do. 31.10.2014	

## 1 Relationenalgebra

a)

$$\pi_{\text{Jahresgehalt}}(\text{Job} \bowtie_{\text{JNR}=\text{Job}} \text{Bewerbung} \bowtie_{\text{Bewerber}=\text{PNR}} (\sigma_{\text{Geb} \geq '1980-01-01'}(\text{Person})))$$

b)

$$\pi_{\text{Titel}, \text{Jahresgehalt}}(\text{Job} \bowtie_{\text{JNR}=\text{Job}} \text{Bewerbung} \bowtie_{\text{Bewerber}=\text{PNR}} \text{Person} \bowtie_{\text{Heimat}=\text{LNR}} (\sigma_{\text{Name}='Schweiz'}(\text{Land})))$$

c)

$$\pi_{\text{Vorname}, \text{Nachname}}(\text{Personen} \bowtie (\pi_{\text{PNR}}(\text{Personen}) - \pi_{\text{PNR}}(\text{Bewerber})))$$

d)

Gebe das Geburtsdatum der Personen aus, deren Sachbearbeiter nach dem 31.12.1994 geboren worden sind.

## 2 Schemadefinition


a)

Listing 1: Ein Beispiel

```

1 create table gdb_gruppe067.Buch (
2   Titel varchar(30) primary key,
3   Erscheinungsjahr int not null,
4   Seitenzahl int check(0<Seitenzahl && Seitenzahl<4000),
5   Verlag varchar(30) not null
6 );
7
8 create table gdb_gruppe067.Person (
9   PID int primary key,
10  Vorname varchar(30) not null,
11  Nachname varchar(30) unique not null,
12  Lieblingsbuch varchar(30) not null,
13  CONSTRAINT fav_book FOREIGN KEY (Lieblingsbuch) REFERENCES Buch (Titel)
14 );
15

```

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>			WS 2014/15
	Aufgabenzettel	<b>4</b>			
	STiNE-Gruppe 30	<b>Cornelia Hofsäß, Aleksej Davletcurin, Sascha Marcel Hacker</b>			
	Ausgabe	<b>Mi. 15.10.2014</b>	Abgabe	<b>Do. 31.10.2014</b>	


```

16 create table gdb_gruppe067.Schreibt(
17   Autor int ,
18   Buch varchar(30),
19   constraint schreibt_key primary key (Autor,Buch),
20   constraint schreibt_autor foreign key (Autor) references Person(PID),
21   constraint schreibt_buch foreign key (Buch) references Buch(Titel)
22 );
23
24
25 create table gdb_gruppe067.Begutachtet(
26   Lektor int ,
27   Buch varchar(30),
28   constraint begutachtet_key primary key (Lektor,Buch),
29   constraint begutachtet_autor foreign key (Lektor) references Person(PID),
30   constraint begutachtet_buch foreign key (Buch) references Buch(Titel)
31 );

```

b)

Dadurch, dass die referentielle Integrität von Fremdschlüsseln nicht verzögert am Ende der Transaktion geprüft werden kann, können keine wechselseitigen Abhängigkeiten zwischen unterschiedlichen Tabellen aufgebaut werden. Wenn die Tabelle Buch noch das Attribut Editor enthalten würde, das auf Person.Pid zeigt, dann können beide Tabellen erst einmal nicht erstellt werden, da sie voneinander abhängig sind. Eine Lösung wäre den Fremdschlüssel mit Hilfe des Befehls alter table später einzufügen.

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2014/15
	Aufgabenzettel	4			
	STiNE-Gruppe 30	Cornelia Hofsäß, Aleksej Davletcurin, Sascha Marcel Hacker			
	Ausgabe	Mi. 15.10.2014	Abgabe	Do. 31.10.2014	

c)

```

insert into gdb_gruppe067.Buch(Titel, Erscheinungsjahr,Seitenzahl,Verlag)
values('Schall und Wahn',1929,304,'Diogenes'),
('Als ich im Sterben lag',1930,173,'Diogenes'),
('Hundert Jahre Einsamkeit',1967,480,'Fischer'),
('Der Fremde',1942,160,'rororo'),
('Krieg und Frieden',1869,1536,'Anaconda'),
('Anna Karenina',1878,991,'Anaconda'),
('Schuld und Sühne',1866,752,'Deutscher Taschenbuch Verlag'),
('Requiem für einen Traum',1978,316,'Rowohlt'),
('Der Talisman',1984,714,'Heyne')
;


insert into gdb_gruppe067.Person(PID, Vorname, Nachname,Lieblingsbuch)
values (1,'Leo','Tolstoi','Schuld und Sühne'),
(2,'Fjodor','Dostojewski','Krieg und Frieden'),
(3,'Hubert','Selby','Der Fremde'),
(4,'Albert','Camus','Schuld und Sühne'),
(5,'William','Faulkner','Schuld und Sühne'),
(6,'Stephen','King','Hundert Jahre Einsamkeit'),
(7,'Peter','Straub','Schall und Wahn'),
(8,'Gabriel','García Márquez','Requiem für einen Traum')
;

insert into gdb_gruppe067.Schreibt(Autor,Buch)
values
(2,'Schuld und Sühne'),
(3,'Requiem für einen Traum'),
(4,'Der Fremde'),
(5,'Schall und Wahn'),
(5,'Als ich im Sterben lag'),
(6,'Der Talisman'),
(7,'Der Talisman'),
(8,'Hundert Jahre Einsamkeit');

insert into gdb_gruppe067.Begutachtet(Lektor,Buch)
values (2,'Anna Karenina'),
(1,'Schuld und Sühne'),
(8,'Requiem für einen Traum'),
(6,'Requiem für einen Traum'),
(5,'Der Fremde'),
(4,'Als ich im Sterben lag'),
(2,'Krieg und Frieden'),
(7,'Hundert Jahre Einsamkeit');

```

d)

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2014/15
	Aufgabenzettel	4			
	STiNE-Gruppe 30	Cornelia Hofsäß, Aleksej Davletcurin, Sascha Marcel Hacker			
	Ausgabe	Mi. 15.10.2014	Abgabe	Do. 31.10.2014	

Listing 2: Ein Beispiel

```

1 drop table gdb_gruppe067.Begutachtet;
2 drop table gdb_gruppe067.Schreibt;
3 drop table gdb_gruppe067.Person;
4 drop table gdb_gruppe067.Buch;
5
6 select @peter:= PID
7 from gdb_gruppe067.Person
8 where Vorname = "Peter"
9 ;
10
11 delete from gdb_gruppe067.Begutachtet where Lektor = @peter;
12 delete from gdb_gruppe067.Schreibt where Autor = @peter;
13 delete from gdb_gruppe067.Person where PID = @peter;

```

### 3 SQL

#### 3.1 a)

Listing 3: Ein Beispiel

```

1 select count(Bewerbung.Sachbearbeiter), Person.Nachname, Person.PNR
2 from Person, Bewerbung
3 where Person.PNR=Bewerbung.Sachbearbeiter
4 group by(Bewerbung.Sachbearbeiter)

```


#### 3.2 b)

Listing 4: Ein Beispiel

```

1 select a.Sachbearbeiter
2 from Bewerbung a
3 where 2 < (select Count(*)
4 From Bewerbung b
5 Where a.Sachbearbeiter = b.Sachbearbeiter)
6 group by a.Sachbearbeiter

```

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b> WS 2014/15		
	Aufgabenzettel	<b>4</b>		
	STiNE-Gruppe 30	<b>Cornelia Hofsäß, Aleksej Davletcurin, Sascha Marcel Hacker</b>		
	Ausgabe	<b>Mi. 15.10.2014</b>	Abgabe	<b>Do. 31.10.2014</b>

### 3.3 c)

Listing 5: Ein Beispiel

```

1 select p1.Vorname
2 from Person p1, Bewerbung b, Person p2
3 where p1.PNR=b.Bewerber
4 and p2.PNR=b.Sachbearbeiter
5 and p2.Nachname=p1.Nachname;

```

### 3.4 d)

Listing 6: Ein Beispiel

```

1 select p.Vorname, p.Nachname, p.PNR
2 from Person p
3 where p.PNR not in
4 (select b.Sachbearbeiter from Bewerbung b);

```

## 4 Optimierung