

# GDB Uebung 2, Gruppe 61

Arne Beer, MN 6489196

Oliver Heidmann, MN 6420331, Minh Nguyen, MN 6423136

12. Dezember 2013

1. (a)  $\pi_{\text{Sorte}}((\sigma_{\text{Vorname}=\text{"Horst"}}\text{Personen}) \bowtie_{\text{Entdecker=PNR}} \text{Obst})$   
(b)  $\pi_{\text{Vorname, Nachname}}((\sigma_{\text{Symptom}=\text{"Halskratzen"}}\text{Allergien}) \bowtie \text{Person})$   
(c)  $\pi_{\text{Sorte, Nachname}}(((\sigma_{\text{Symptom}=\text{"Wuergreiz"}}\text{Allergie}) \bowtie \text{Person}) \bowtie_{\text{Entdecker=PNR}} \text{Obst})$
2. (a) 

```
CREATE TABLE Rennfahrer(  
    RID int PRIMARY KEY,  
    Vorname varchar(50) NOT NULL,  
    Nachname varchar(50) NOT NULL,  
    Geburt varchar(10) NOT NULL,  
    Wohnort varchar(50),  
    CONSTRAINT Rennstall FOREIGN KEY (Rennstall) REFERENCES Rennstall (RSID)  
);  
CREATE TABLE Rennstall(  
    RSID int PRIMARY KEY,  
    Name varchar(50) NOT NULL  
    Teamchef varchar(50),  
    Budget int check(0<Budget AND Budget<500)  
);  
CREATE TABLE Rennort(  
    OID int PRIMARY KEY,  
    Name varchar(50) NOT NULL,  
    Strecke varchar(50) NOT NULL,  
);  
CREATE TABLE Platzierung(  
    CONSTRAINT RID FOREIGN KEY (Rennfahrer) REFERENCES Rennfahrer(PID),  
    CONSTRAINT OID FOREIGN KEY (Rennort) REFERENCES Rennort(OID),  
    Platz int NOT NULL,  
    CONSTRAINT pk_platz PRIMARY KEY (RID, OID),  
);
```

  
(b) Wenn eine Table auf eine andere Table referenziert, welche noch nicht initianilisiert ist, wird ein Fehler geworfen. Daher muss die Fremdschlüssel im Nachträglich hinzugefügt werden.  
(c) 

```
ALTER TABLE Rennstall(  
    ADD CONSTRAINT Star FOREIGN KEY(Rennfahrer) REFERENCES Rennfahrer(PID)  
);
```

  
(d) 

```
i. delete from Rennfahrer S  
    where S.Vorname Like "F%"  
  
ii. drop table *
```

3. (a) 

```
SELECT DISTINCT B.Sorter
FROM Person A, Obst B, Allergie C
WHERE A.Vorname = "Peter" AND A.Nachname = "Meyer"
AND C.PNR = A.PNR
AND C.ONR = B.ONR
ORDER BY B.Sorter DESC;
```
- (b) 

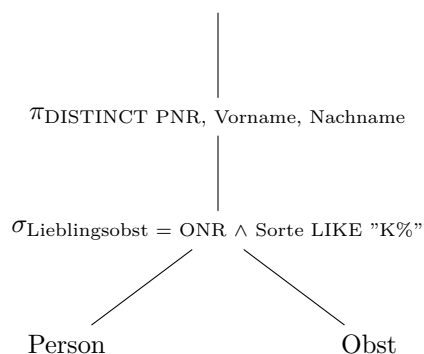
```
SELECT A.PNR, A.Nachname, COUNT (DISTINCT B.PNR)
FROM Person A, Allergie B
WHERE B.PNR = A.PNR;
```
- (c) 

```
SELECT A.PNR,
FROM Person A, Obst B
WHERE B.Entdecker = A.PNR AND COUNT(B.Sorter) > 6;
```
- (d) 

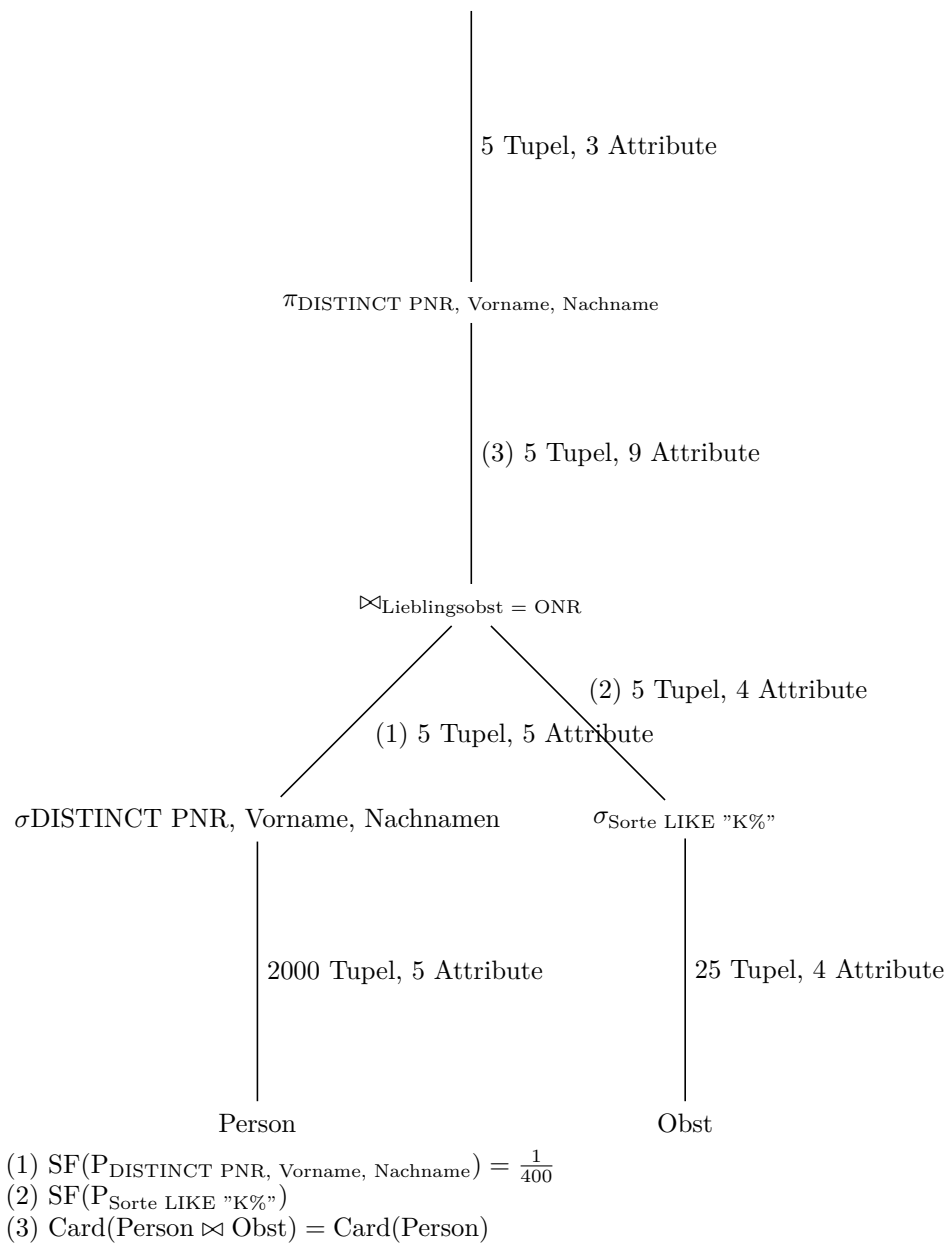
```
SELECT A.Vorname, A.Nachname
FROM Person A, Person B, Obst C
WHERE C.Entdecker = B.PNR
AND B.Vorname = A.Vorname;
```
- (e) 

```
SELECT A.Vorname, A.Nachname
FROM PERSON A
WHERE NOT EXISTS
(SELECT A.PNR
FROM OBST.B
WHERE A.PNR = B.Entdecker);
```

4.  $\text{Card}(\text{Person}) = 2000$ ,  
 $\text{Card}(\text{Obst}) = 25$ ,  
400 verschiedene Nachnamen,  
Sorter eindeutig,  
Obstsorten sind eindeutig und genau 5 beginnen mit dem Buchstaben K ,



Optimierter Baum:



Endergebnis ist optimaler, da die Selections frueh angewendet werden und join genutzt wird, wodurch die Zwischenergebnisse schon zu Beginn sehr klein werden.