	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	4 (Lösungsvorschläge)			
	Gesamtpunktzahl	40			
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015	

Aufgabe 1: Relationenalgebra

[6 P.]

Gegeben seien die folgenden Relationenschemata:

Person(PNR, Vorname, Nachname, DOB, Lieblingsobst \rightarrow Obst.ONR)

Obst(ONR, Sorte, Entdecker \rightarrow Person.PNR)

Allergie(Person \rightarrow Person.PNR, Obst \rightarrow Obst.ONR, Symptom)

Benutzen Sie zur Lösung der folgenden Aufgaben ausschließlich die in der Vorlesung vorgestellten Operatoren der Relationenalgebra!

- a) Geben Sie einen Relationenalgebra-Ausdruck an, der zu dem unten angegebenen SQL-Ausdruck äquivalent ist. [2 P.]

```
SELECT DISTINCT o.Sorte
FROM   Personen p, Obst o
WHERE  o.Entdecker = p.PNR
AND    p.Vorname = 'Horst'
```

Lösungsvorschlag:

$$\pi_{\text{Sorte}}((\sigma_{\text{Vorname}='Horst'}(\text{Person})) \bowtie_{\text{PNR}=\text{Entdecker}} \text{Obst})$$

- b) Geben Sie einen Relationenalgebra-Ausdruck an, der die Vor- und Nachnamen aller Personen ausgibt, die eine Allergie haben, die mit dem Symptom „Halskratzen“ auftritt. [2 P.]


Lösungsvorschlag:

$$\pi_{\text{Vorname}, \text{Nachname}}(\text{Person} \bowtie_{\text{PNR}=\text{Person}} (\sigma_{\text{Symptom}='Halskratzen'}(\text{Allergie})))$$

- c) Geben Sie einen Relationenalgebra-Ausdruck an, der für jede Obstsorte die Sorte und den Nachnamen des jeweiligen Entdeckers listet, wenn die Obstsorte bei ihrem Entdecker einen Würgereiz auslöst. [2 P.]

Lösungsvorschlag:

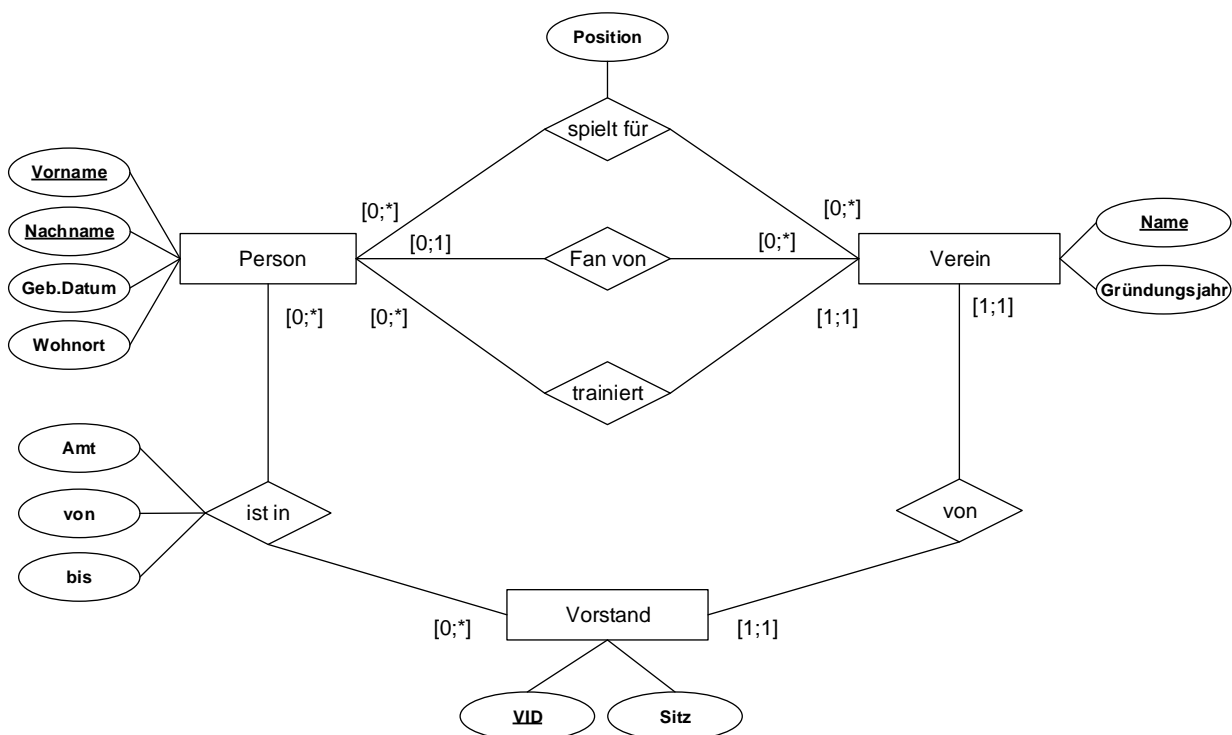
$$\pi_{\text{Sorte}, \text{Nachname}}(\text{Obst} \bowtie_{\text{Entdecker}=\text{Person} \wedge \text{ONR}=\text{Obst}} ((\sigma_{\text{Symptom}='Würgereiz'}(\text{Allergie})) \bowtie_{\text{Person}=\text{PNR}} \text{Person}))$$

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	4 (Lösungsvorschläge)			
	Gesamtpunktzahl	40			
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015	

Aufgabe 2: Schemadefinition


[12 P.]

Geben Sie die SQL-DDL-Anweisungen an, die notwendig sind, um das DB-Schema für das nachfolgend dargestellte Entity-Relationship-Diagramm zu erstellen. Wählen Sie dabei geeignete SQL-Standard-Datentypen. Beachten Sie, dass die Kardinalitätsrestriktionen durch geeignete Constraints exakt abzubilden sind. Weiterhin ist bei 1:1-Beziehungen die Symmetrie sicherzustellen (Tipp: Fremdschlüssel in beiden Relationen). Testen Sie die SQL-Ausdrücke auf der Übungsdatenbank.



Weiterhin gelten folgende Integritätsbedingungen:

- **IB1:** Der Sitz eines Vorstandes ist eindeutig.
- **IB2:** Gründungsjahr, Geburtsdatum und das Datum, bis zu welchem eine Person ein Amt eines Vorstandes belegt, sind optional. Alle anderen Attribute sind verpflichtend anzugeben.
- **IB3:** Das Geburtsdatum einer Person muss (sofern angegeben) kleiner als das aktuelle Datum (CURRENT_DATE) sein.

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	4 (Lösungsvorschläge)			
	Gesamtpunktzahl	40			
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015	

Lösungsvorschlag:

```

CREATE TABLE Person(
  Vorname varchar(50) NOT NULL,
  Nachname varchar(50) NOT NULL,
  Geburtsdatum date CHECK(Geburtsdatum < CURRENT_DATE),
  Wohnort varchar(50) NOT NULL,
  Lieblingsverein varchar(50),
  CONSTRAINT pk_person PRIMARY KEY (Vorname, Nachname)
);


CREATE TABLE Verein(
  Name varchar(50) PRIMARY KEY NOT NULL,
  Gruendungsjahr date,
  TrainerInVorname varchar(50) NOT NULL,
  TrainerInNachname varchar(50) NOT NULL,
  Vorstand int UNIQUE NOT NULL,
  CONSTRAINT fk_verein_trainer FOREIGN KEY (TrainerInVorname,TrainerInNachname)
  REFERENCES Person (Vorname,Nachname)
);

CREATE TABLE spielt_fuer(
  Vorname varchar(50) NOT NULL,
  Nachname varchar(50) NOT NULL,
  Verein varchar(50) NOT NULL,
  Position varchar(50) NOT NULL,
  CONSTRAINT pk_spielfuer PRIMARY KEY (Vorname, Nachname, Verein),
  CONSTRAINT fk_spielfuer_pers FOREIGN KEY (Vorname, Nachname)
  REFERENCES Person (Vorname, Nachname),
  CONSTRAINT fk_spielfuer_verein FOREIGN KEY (Verein) REFERENCES Verein(Name)
);

CREATE TABLE Vorstand(
  VID int PRIMARY KEY NOT NULL,
  Sitz varchar(50) UNIQUE NOT NULL,
  CONSTRAINT fk_vorstand_verein FOREIGN KEY (VID) REFERENCES Verein (Vorstand)
);

CREATE TABLE ist_in_Vorstand(
  Vorname varchar(50) NOT NULL,
  Nachname varchar(50) NOT NULL,
  Vorstand int NOT NULL,
  Amt varchar(50) NOT NULL,

```

	Lehrveranstaltung	Grundlagen von Datenbanken WS 2015/16		
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015

```

Von date NOT NULL,
Bis date,
CONSTRAINT pk_istinvorstand PRIMARY KEY (Vorname, Nachname, Vorstand),
CONSTRAINT fk_istinvorstand_pers FOREIGN KEY (Vorname, Nachname)
REFERENCES Person (Vorname, Nachname),
CONSTRAINT fk_istinvorstand_verein FOREIGN KEY (Vorstand) REFERENCES Vorstand(VID)
);

```

```


ALTER TABLE Person
ADD CONSTRAINT fk_person_lv FOREIGN KEY (Lieblingsverein) REFERENCES Verein(Name)
INITIALLY IMMEDIATE DEFERRABLE;

```

```

ALTER TABLE Verein
ADD CONSTRAINT fk_verein_vorstand FOREIGN KEY (Vorstand) REFERENCES Vorstand (VID)
INITIALLY IMMEDIATE DEFERRABLE;

```

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	4 (Lösungsvorschläge)			
	Gesamtpunktzahl	40			
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015	

Aufgabe 3: Datenmanipulation mit SQL

[17 P.]

Gegeben sei das Formel 1 Datenbankschema aus dem letzten Übungsblatt:

<u>RennfahrerIn</u>	<u>RID</u>	Vorname	Nachname	Geburt	Wohnort	<u>Rennstall</u>
	4	Sebastian	Vettel	1987-07-03	Kemmental (Schweiz)	2
	6	Fernando	Alonso	1981-07-29	Lugano (Schweiz)	5
	8	Marc	Webber	1976-08-27	Aston Clinton (UK)	2
	9	Lewis	Hamilton	1985-01-07	Genf (Schweiz)	31
	20	Jenson	Button	1980-01-19	Monte Carlo (Monaco)	31
	21	Felipe	Massa	1982-04-25	São Paulo (Brasilien)	5
	44	Brendon	Hartley	1989-11-10	Palmerston North (Neuseeland)	2


Rennstall → Rennstall.RSID

<u>Rennstall</u>	<u>RSID</u>	Name	TeamchefIn	Budget
	2	Red Bull	Christian Horner	120
	5	Ferrari	Stefano Domenicali	220
	31	McLaren	Martin Whitmarsh	220

<u>Rennort</u>	<u>OID</u>	Name	Strecke
	4	Brasilien GP	Autodromo Jose Carlos Pace, Interlagos
	15	Abu Dhabi GP	Yas Marina Circuit
	21	Großbritannien GP	Silverstone Grand Prix Circuit

<u>Platzierung</u>	<u>RID</u>	<u>OID</u>	Platz
	8	4	2
	4	15	1
	20	15	3
	4	4	1
	6	4	3
	8	15	8
	6	21	14
	9	15	2
	9	4	4
	21	15	10
	20	4	5
	21	4	15
	6	15	7

RID → RennfahrerIn.RID, OID → Rennorte.OID

	Lehrveranstaltung	Grundlagen von Datenbanken WS 2015/16		
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015

- a) Formulieren Sie entsprechende SQL-Anweisungen für die in den nachfolgenden Teilaufgaben angeführten natürlichsprachlich formulierten Mengenbeschreibungen. **Verwenden Sie den in der Vorlesung verwendeten SQL-Standard.** Das SQL-Schlüsselwort JOIN darf dabei aber nur zur Spezifizierung eines äusseren Verbundes verwendet werden. Testen Sie die SQL-Ausdrücke auf der Übungsdatenbank. **(9 Punkte)**

Hinweis: Die zum Testen benötigten Schema- und Instanzdaten sollten noch von dem letzten Übungsblatt vorhanden sein und können notfalls mit dem auf der Veranstaltungsseite bereitgestellten SQL-Skript erneut erstellt bzw. eingefügt werden.

- i) Zu jedem Rennstall dessen Namen und die Gesamtzahl der zu diesem Rennstall gehörenden FahrerInnen. (1 Punkt)

Lösungsvorschlag:

```
SELECT Name, COUNT(*) AS Anzahl_FahrerInnen
FROM RennfahrerIn, Rennstall
WHERE RSID = Rennstall
GROUP BY RSID, Name;
```

- ii) Vor- und Nachnamen der RennfahrerInnen, die keine Platzierung erlangt haben. (1 Punkt)

Lösungsvorschlag:

```
SELECT R.Vorname, R.Nachname
FROM RennfahrerIn R
WHERE R.RID NOT IN(SELECT P.RID
FROM Platzierung P);
```

- iii) Zu jeder RennfahrerIn, die jemals eine Platzierung erlangt hat, deren Nachnamen und deren bester erreichter Platz, nach Platzierung aufsteigend sortiert. (1 Punkt)


Lösungsvorschlag:

```
SELECT R.Nachname, MIN(P.Platz) AS Bester_Platz
FROM Platzierung P, RennfahrerIn R
WHERE P.RID = R.RID
GROUP BY R.RID, R.Nachname
ORDER BY Bester_Platz ASC;
```

- iv) Zu jeder RennfahrerIn (egal, ob jemals platziert oder nicht), deren Nachnamen und deren bester erreichter Platz, nach Nachname aufsteigend sortiert. (2 Punkte)

Lösungsvorschlag:

```
SELECT R.Nachname, MIN(P.Platz) AS Bester_Platz
FROM RennfahrerIn R LEFT OUTER JOIN Platzierung P ON P.RID = R.RID
```

	Lehrveranstaltung	Grundlagen von Datenbanken WS 2015/16		
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015

```
GROUP BY R.RID, R.Nachname
ORDER BY R.Nachname ASC;
```

- v) Alle Informationen zu RennfahrerInnen, die zwischen dem 01.01.1980 und dem 01.01.1985 geboren wurden und bei deren Nachname der zweite Buchstabe ein 'a' ist. (1 Punkt)

Lösungsvorschlag:

```
SELECT *
FROM RennfahrerIn
WHERE Geburt BETWEEN '1980-01-01' AND '1985-01-01'
AND Nachname LIKE '_a%';
```

- vi) Die TeamchefInnen, aus deren Rennställen RennfahrerInnen eine Platzierung an Rennorten hatten, an denen mehr als 5 Plätze vergeben wurden. (3 Punkte)

Lösungsvorschlag:


```
SELECT DISTINCT RS.TeamchefIn
FROM RennfahrerIn R, Rennstall RS, Platzierung P1
WHERE R.Rennstall = RS.RSID
AND R.RID = P1.RID
AND P1.OID IN(
    SELECT P2.OID
    FROM Platzierung P2
    GROUP BY P2.OID
    HAVING COUNT(*) > 5);
```

- b) Übersetzen Sie die folgenden umgangssprachlich formulierten Anweisungen zur Änderung und Eingabe von Datensätzen jeweils in einen zugehörigen SQL-Ausdruck (Hinweis: Verwenden Sie hierfür das UPDATE-Statement bzw. das INSERT-Statement). Testen Sie die SQL-Ausdrücke auf der Übungsdatenbank. Begründen Sie, falls eine Anweisung vom Datenbanksystem zurückgewiesen werden sollte. (8 Punkte)

- i) Ersetze den aktuell gespeicherten Wohnort von Marc Webber (RID=8) durch 'Aston Clinton (United Kingdom)'. (1 Punkt)

Lösungsvorschlag:

```
UPDATE RennfahrerIn
Set Wohnort = 'Aston Clinton (United Kingdom)'
WHERE RID = 8;
```

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	4 (Lösungsvorschläge)			
	Gesamtpunktzahl	40			
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015	

- ii) Erhöhe das Budget der Teams, bei denen die durchschnittliche Platzierung ihrer RennfahrerInnen besser als Platz 5 ist, um 50. (2 Punkte)

Lösungsvorschlag:

```
UPDATE Rennstall
SET Budget = Budget + 50
WHERE RSID IN(SELECT R.Rennstall
FROM RennfahrerIn R, Platzierung P
WHERE R.RID = P.RID
GROUP BY R.Rennstall
HAVING AVG(P.Platz) < 5.0);
```

Anmerkung: Geht bei MySQL nur, wenn man bei Edit->Preferences->SQL Editor (bzw. Edit->Preferences->SQL Queries im Falle älterer Versionen) den Haken bei safe Updates entfernt (Safe Update soll wohl gewährleisten, dass immer nur ein Tupel geändert wird)


- iii) Füge den 'Deutschland GP' mit dem Streckennamen 'Hockenheimring' zu den Rennorten hinzu. (1 Punkt)

Lösungsvorschlag:

```
INSERT INTO Rennort (OID, Name, Strecke)
VALUES (1,'Deutschland GP', 'Hockenheimring');
```

ODER

```
INSERT INTO Rennort (OID, Name, Strecke)
SELECT max(OID) + 1, 'Deutschland GP', 'Hockenheimring' FROM Rennort;
```


	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	4 (Lösungsvorschläge)			
	Gesamtpunktzahl	40			
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015	

- iv) Platziere die RennfahrerIn, die beim Brasilien GP auf Platz 1 war, auch beim Deutschland GP auf Platz 1.

(2 Punkte)

Lösungsvorschlag:

```
INSERT INTO Platzierung (RID, OID, Platz)
SELECT R.RID, O.OID, 1
FROM (SELECT P.RID
      FROM Platzierung P, Rennort R0
      WHERE P.OID = R0.OID AND Platz = 1 AND Name = 'Brasilien GP') AS R,
      (SELECT OID
      FROM Rennort
      WHERE name = 'Deutschland GP') AS O;
```

ODER

```
INSERT INTO Platzierung (RID, OID, Platz)
SELECT P.RID, O1.OID, 1
FROM Rennort O1, Rennort O2, Platzierung P
WHERE O1.Name = 'Deutschland GP'
AND O2.Name = 'Brasilien GP'
AND P.OID = O2.OID
AND P.Platz = 1;
```

Hinweis: Der Name eines Rennens muss de Schema nach nicht eindeutig sein. Es kann also passieren, dass diese Statements mehr als ein Tupel in die Tabelle Platzierung eintragen.


- v) Füge einen 5ten Platz von Felipe Massa beim Großbritannien GP ein.

(2 Punkte)

Lösungsvorschlag:

erste Möglichkeit:

```
INSERT INTO Platzierung (RID, OID, Platz)
SELECT R.RID, O.OID, 5
FROM RennfahrerIn R, Rennort O
WHERE R.Vorname = 'Felipe'
AND R.Nachname = 'Massa'
AND O.Name = 'Großbritannien GP';
```

	Lehrveranstaltung	Grundlagen von Datenbanken WS 2015/16		
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015

zweite Möglichkeit:


```
INSERT INTO Platzierung (RID, OID, Platz)
SELECT R.RID, O.OID, 5
FROM (SELECT RID
      FROM RennfahrerIn
      WHERE Vorname = 'Felipe'
      AND Nachname = 'Massa') AS R,
      (SELECT OID
      FROM Rennort
      WHERE Name = 'Großbritannien GP') AS O;
```

dritte Möglichkeit:

```
INSERT INTO Platzierung (RID, OID, Platz)
Values ((SELECT RID
        FROM RennfahrerIn
        WHERE Vorname = 'Felipe'
        AND Nachname = 'Massa'),
        (SELECT OID
        FROM Rennort
        WHERE Name = 'Großbritannien GP'), 5);
```

Generell ist diese Vorgehensweise (Verwendung von Unteranfragen zur Bestimmung von Werten in der Value-Klausel) nicht zu empfehlen, da dies nur dann funktioniert, wenn jede Unteranfrage exakt einen Wert liefert.

Zudem ist bei diesen Aufgaben wichtig, dass dem Nutzer nur die Informationen zur Verfügung stehen, die in der Aufgabenstellung spezifiziert wurden (d.h. die RID von Felipe Massa ist dem Nutzer in Aufgabenteil v) nicht bekannt)

	Lehrveranstaltung	Grundlagen von Datenbanken WS 2015/16		
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015

Aufgabe 4: Anfrageoptimierung


[5 P.]

Übersetzen Sie folgende SQL-Anfrage entsprechend dem in der Vorlesung vorgestellten Erklärungsmodell in einen Operatorbaum (wählen Sie einen beliebigen der verschiedenen möglichen Operatorbäume). Führen Sie anschließend eine algebraische Optimierung entsprechend den in der Vorlesung eingeführten Regeln durch. Bewerten Sie beide Operatorbäume mit den Kardinalitäten der Zwischenergebnisse.

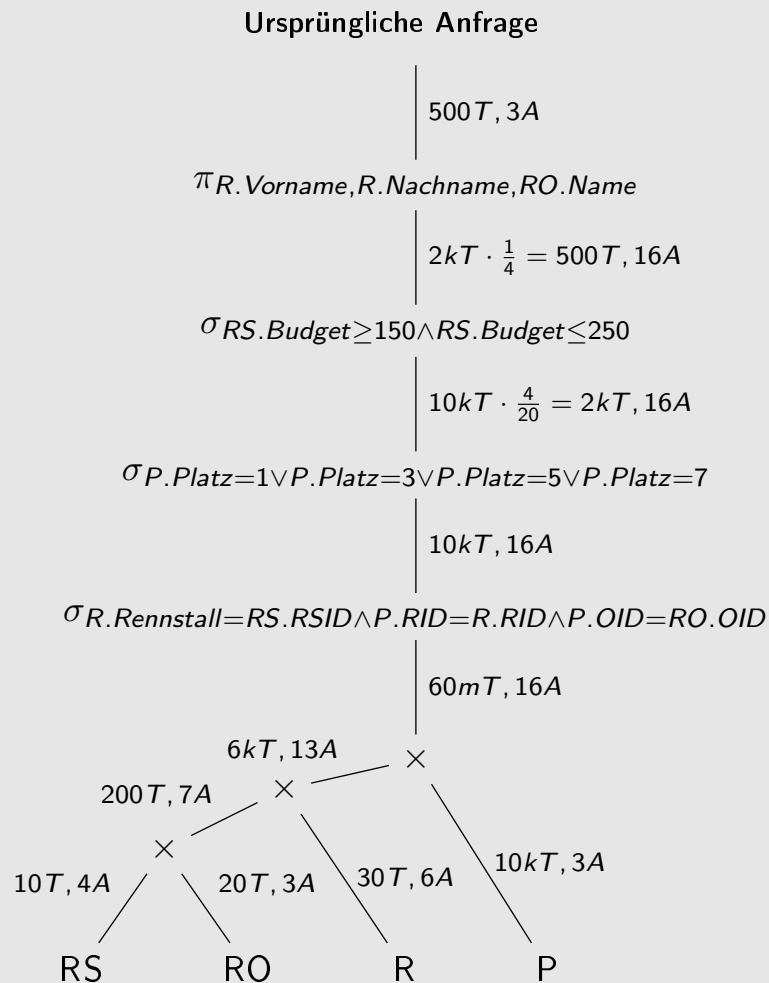
```
SELECT DISTINCT R.Vorname, R.Nachname, RO.Name
FROM  Rennstall RS,
      Rennort RO,
      RennfahrerIn R,
      Platzierung P
WHERE R.Rennstall = RS.RSID
AND   P.RID = R.RID
AND   P.OID = RO.OID
AND   P.Platz IN (1,3,5,7)
AND   RS.Budget BETWEEN 150 AND 250;
```

Für die zugehörige Datenbank werden folgende Kardinalitäten angenommen:


$\text{Card}(\text{RennfahrerIn}) = 30$, $\text{Card}(\text{Rennstall}) = 10$, $\text{Card}(\text{Rennort}) = 20$, $\text{Card}(\text{Platzierung}) = 10k$. Zudem starten bei einem Rennen immer 20 FahrerInnen und es wird angenommen, dass jede dieser FahrerInnen stets platziert wird. Hinweis: Beachten Sie, über das minimale und maximale Budget eines Rennstalles ist nichts bekannt. Daher muss in diesem Fall, die in der Vorlesung behandelte Abschätzung des Selektivitätsfaktors verwendet werden.

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	4 (Lösungsvorschläge)			
	Gesamtpunktzahl	40			
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015	

Lösungsvorschlag:



Erläuterung: T=Tupel, A=Attribute

	Lehrveranstaltung	Grundlagen von Datenbanken			WS 2015/16
	Aufgabenzettel	4 (Lösungsvorschläge)			
	Gesamtpunktzahl	40			
	Ausgabe	Mi. 25.11.2015	Abgabe	Fr. 11.12.2015	

Optimierte Anfrage

