

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2013/14
	Aufgabenzettel	<b>4</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 27.11.2013</b>	Abgabe	<b>Do. 12.12.2013</b>

## Aufgabe 1: Relationenalgebra

[6 P.]

Gegeben seien die folgenden Relationenschemata:

*Person*(PNR, Vorname, Nachname, DOB, Lieblingsobst → Obst.ONR)

*Obst*(ONR, Sorte, Entdecker → Person.PNR)

*Allergie*(Person → Person.PNR, Obst → 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'
```

- 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.]
- 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 Würgreiz auslöst. [2 P.]

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2013/14
	Aufgabenzettel	<b>4</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 27.11.2013</b>	Abgabe	<b>Do. 12.12.2013</b>

## Aufgabe 2: SQL – Schemadefinition

[18 P.]

Wir verwenden das gleiche Datenbankschema wie in der dritten Aufgabe von Aufgabenblatt 3:

<u>Rennfahrer</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	Kimi	Räikkönen	1979-10-17	Espoo (Finnland)	34

Rennstall → Rennstall.RSID

<u>Rennstall</u>	<u>RSID</u>	Name	Teamchef	Budget
	2	Red Bull	Christian Horner	370
	5	Ferrari	Stefano Domenicali	350
	31	McLaren	Martin Whitmarsh	220
	34	Lotus F1	Eric Boullier	100

<u>Rennort</u>	<u>OID</u>	Name	Strecke
	4	Australien GP	Albert Park Circuit
	15	Malaysia GP	Sepang International Circuit
	21	China GP	Shanghai International Circuit

<u>Platzierung</u>	<u>RID</u>	<u>OID</u>	Platz
	8	4	6
	4	15	1
	20	15	17
	4	4	3
	6	4	2
	8	15	2
	6	21	1
	9	4	5
	21	15	5
	20	4	9
	21	4	4

RID → Rennfahrer.RID, OID → Rennorte.OID

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2013/14
	Aufgabenzettel	<b>4</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 27.11.2013</b>	Abgabe	<b>Do. 12.12.2013</b>

Um die Konsistenz der Daten sicherzustellen, sollen folgende Integritätsbedingungen gelten:

IB1: Das Rennstallbudget muss zwischen 0 und 500 liegen.

IB2: Alle Felder bis auf `Rennfahrer.Wohnort` und `Rennstall.Teamchef` sind Pflichtfelder.

IB3: Rennstallnamen sind eindeutig.

- a) Definieren Sie das angegebene Schema mithilfe von Befehlen der SQL DDL (Data Definition Language). Zur Prüfung Ihrer Lösung führen Sie die DDL-Befehle bitte in MySQL aus. Legen Sie die Prüfung des Budgets als eine Check-Klausel an, auch wenn MySQL diese (ohne Fehler) ignoriert. Legen Sie Fremdschlüssel- und Check-Constraints bitte als benannte Constraints an. [8 P.]
- b) In MySQL wird die referentielle Integrität von Fremdschlüsseln nicht verzögert am Ende der Transaktion (*deferred*) geprüft, sondern stets direkt. Welche Einschränkungen bringt dies mit sich?  
Was müsste man bei der Definition des Schemas in SQL DDL beachten, wenn die `Rennstall`-Relation ein neues Attribut `Star` erhalten würde, welches einen `Rennfahrer` referenziert? Welches Problem träte bei der Manipulation der Daten in SQL DML auf? [4 P.]
- c) Geben Sie SQL-Befehle an, die die Datenbank mit den in der Tabelle angegebenen Datensätzen füllen. [4 P.]
- d) Geben Sie SQL Befehle an, um: [2 P.]
  - alle `Rennfahrer` zu löschen, deren Vorname mit einem „F“ beginnt.
  - alle Tabellen zu löschen.

### Aufgabe 3: SQL – Anfragen

[10 P.]

Gegeben seien die aus Aufgabe 1 bekannten Relationenschemata.

Formulieren Sie für die in den nachfolgenden Teilaufgaben angeführten, natürlichsprachlich formulierten Anfragen entsprechende SQL-Anweisungen. **Verwenden Sie den in der Vorlesung verwendeten SQL-Standard.** Das SQL-Schlüsselwort `JOIN` darf dabei nicht verwendet werden.

- a) Alle Obstsorten, gegen die ein Peter Meyer allergisch ist, ohne Duplikate in absteigender Sortierung. [2 P.]
- b) Die PNR, den Nachnamen und die Anzahl der Allergien jedes Allergikers. [2 P.]
- c) Die PNR aller Personen, die mehr als 6 Obstsorten entdeckt haben. [2 P.]
- d) Vorname und Nachname aller Personen, deren Lieblingsobst von einer Person mit dem gleichen Vornamen entdeckt wurde. [2 P.]
- e) Die PNR, Vorname und Nachname aller Personen, die noch keine Obstsorte entdeckt haben. [2 P.]

	Lehrveranstaltung	<b>Grundlagen von Datenbanken</b>		WS 2013/14
	Aufgabenzettel	<b>4</b>		
	Gesamtpunktzahl	<b>40</b>		
	Ausgabe	<b>Mi. 27.11.2013</b>	Abgabe	<b>Do. 12.12.2013</b>

## Aufgabe 4: Optimierung

[6 P.]

Gegeben seien die aus Aufgabe 1 bekannten Relationenschemata:

Für die nachfolgende Anfrage soll eine algebraische Optimierung durchgeführt werden. Zeichnen Sie dafür als erstes den Operatorbaum für die vorgegebene Anfrage und optimieren Sie diese anschließend anhand der in der Vorlesung eingeführten Regeln. Bewerten Sie den Operatorbaum mit den Kardinalitäten der Zwischenergebnisse. Benutzen Sie bei SQL-Anweisungen das in der Vorlesung verwendete SQL-Erklärungsmodell, um die Anfrage in einen Operatorbaum zu überführen.

Für die zugehörige Datenbank werden folgende Kardinalitäten angenommen:

$Card(Person) = 2.000$ ,  $Card(Obst) = 25$ ,  $Card(Allergie) = 10.000$ . Es gibt 400 verschiedene Nachnamen und jedes Obst besitzt eine eindeutige Sorte, wobei es genau 5 Obstsorten gibt, die mit einem 'K' beginnen.

```

SELECT DISTINCT p.PNR, p.Vorname, p.Nachname
FROM Person p, Obst o
WHERE p.Lieblingsobst = o.ONR
      AND o.Sorte LIKE "K%"

```