

**Klausur Datenbanken 1 / Datenbanken  
im SS 2013**

# Musterlösung

**Prüfen Sie bitte zuerst, ob sie die für Sie richtige Klausur vorliegen haben.**

**Beachten Sie bitte auch, dass die Verwendung unerlaubter Hilfsmittel einen Täuschungsversuch darstellt, der entsprechend geahndet wird.**

**Studiengänge:** B\_Inf 2.0; B\_Minif 3.0, 11.0; B\_Ecom 1.0

**Bearbeitungszeit:** 90 Minuten

**Erlaubte Hilfsmittel:** ausgeteiltes Blatt mit Beispielrelation *Firma*

Als Schmierpapier stehen Ihnen die Rückseiten zur Verfügung. Die Rückseiten werden **nicht** bewertet In der Regel stehen einige Zeilen / Spalten / Tableau mehr zur Verfügung als benötigt.

Jede Teilaufgabe wird selbständig bewertet. Aufgabenlösungen werden nur korrigiert und gewertet, wenn der Rechen- bzw. Lösungsweg nachvollziehbar ist. Denken Sie an Kurzkomentare oder Kurzbegründungen innerhalb Ihrer Lösungswege! Die Zeitangaben sind nur zur Groborientierung geeignet.

Viel Erfolg!

## Aufgabe 1: SQL (35 Minuten)

Wir betrachten die in der Vorlesung behandelte Datenbank *Firma* mit den Tabellen *Maschinen*, *Mitarbeiter*, *Gehalt*, *Kind*. Beispieltabellen, aus denen sich auch das Datenbankschema ablesen lässt, finden sich auf dem zusätzlich ausgeteilten Blatt.

Schreiben Sie bitte SQL-Anweisungen, um die folgenden „Fragen“ zu beantworten.

a) Wie viele Bohrmaschinen besitzt die Firma?

### Lösung:

```
SELECT count(*)
FROM maschine
WHERE name = "Bohrmaschine"
```

b) Geben Sie bitte zu jedem **Mitarbeiter** (Spalten **Name** und **Vorname**) aus, wie hoch sein monatliches Gehalt (Spaltenüberschrift **Gehalt**) ist. Das Ergebnis soll **absteigend nach Höhe des Gehalts** der Mitarbeiter, **bei gleichem Gehalt nach Nachname, dann Vorname** sortiert werden.

### Lösung:

```
SELECT p.name, p.vorname, g.betrag
FROM personal p, gehalt g
WHERE p.geh_stufe=g.geh_stufe
ORDER BY g.betrag desc, p.name, p.vorname
```

- c) Die an Gehältern zu zahlende Gesamtsumme pro Abteilung soll absteigend sortiert nach der Gesamtsumme in einer Tabelle mit der Überschrift **ABT\_NR**, **NAME**, **Gesamtsumme** ausgegeben werden.

**Lösung:**

```
SELECT a.abt_nr, a.name, SUM(g.betrag)"Gesamtsumme"
FROM abteilung a, personal p, gehalt g
WHERE a.abt_nr=p.abt_nr and p.geh_stufe=g.geh_stufe
GROUP BY a.abt_nr, a.name, a.abt_nr
ORDER BY 3 DESC
```

- d) Benutzen Sie bitte Unterabfragen und vermeiden Sie Joins: Geben Sie bitte **Name, Vorname und Personalnummer** aller Mitarbeiter an, bei denen mindestens ein Kind im gleichen Jahr geboren ist, indem auch mindestens eine Maschine angeschafft worden ist.

**Lösung:**

```
SELECT p.name, p.vorname, p.pnr
FROM personal p
WHERE p.pnr in (
    SELECT k.pnr
    FROM kind k
    WHERE k.k_geb in (SELECT ansch_datum FROM maschine)
)
```

- e) Von allen Mitarbeitern sollen Name und Vorname und unter der Spaltenüberschrift **Anzahl-Kinder** die Zahl der Kinder angegeben werden. Sind keine Kinder vorhanden, soll für die Anzahl die Zahl 0 ausgegeben werden. Sortiert werden soll absteigend nach der Zahl der Kinder und bei gleicher Kinderzahl aufsteigend nach den Familiennamen.

**Lösung:**

```
SELECT p.name, p.vorname, COUNT(k.pnr)"Anzahl-Kinder"
FROM personal p left outer join kind k using (pnr)
GROUP BY p.name, p.vorname, p.pnr
ORDER BY 3 desc, p.name;
```

- f) Welche Abteilungen haben mehr als 2 Mitarbeiter? Es soll der Name der Abteilung und die Anzahl ihrer Mitarbeiter, sortiert nach Abteilungsname ausgegeben werden.

**Lösung:**

```
SELECT a.name, count(p.pnr)
FROM personal p join abteilung a USING (abt_nr)
GROUP BY a.name, a.abt_nr
HAVING COUNT(p.pnr)>2
ORDER BY a.name
```

- g) Welche Mitarbeiter haben keine Prämie erhalten?

**Lösung:**

```
SELECT pnr, name, vorname
FROM personal
WHERE pnr NOT IN (SELECT pnr FROM praemie)

oder

SELECT pnr, name, vorname
FROM personal left outer join praemie using(pnr)
GROUP BY pnr, name, vorname
HAVING count(p_betrag)=0
```

## Aufgabe 2: Datenbankentwurf (40 Minuten)

Ein Meeresforschungsinstitut beginnt für sein Aquarium eine Datenbank aufzubauen.

Informationen sollen über die **Meerestiere** des Aquariums zur Verfügung gestellt werden und über die **Becken**, in denen die Tiere leben.

Jedes Tier wird in genau einem Becken gehalten, in einem Becken können mehrere Tiere und auch mehrere Tierarten gehalten werden.

Jedes **Becken** hat eine Nummer (**BassinNo**). Des weiteren interessiert die Größe in Liter (**Groesse**), der vorgesehene Salzgehalt (**Salzgehalt**) und die Normtemperatur (**B\_Temperatur**) des Beckens.

Jedes **Meerestier** bekommt sofort nach Geburt im Aquarium bzw. Kauf eine eindeutige Registriernummer (**RegNr**). Weiterhin werden zu jedem Meerestier das Geburtsdatum (**Geb\_Dat**), der wissenschaftliche Name seiner Tierart (**Spezies**) und seine Wohlfühltemperatur (**T\_Temperatur**) gespeichert.

Im Aquarium gibt es **Fische** und **wirbellose Tiere** (wie Krebse oder Weichtiere, z.B. Seeesterne). **Meeressäugetiere** sollten erst später erfasst werden.

Bei den **Fischen** werden die allgemeinen Informationen über die Meerestiere ergänzt durch das Geschlecht (**Geschlecht**) soweit festgestellt.

Bei den **wirbellosen Tieren** werden die allgemeinen Informationen über Tiere ergänzt durch die Art ihres Panzers (**Panzer**) und die Anzahl ihrer Extremitäten (**Beinanzahl**).

Ein Meerestier hat viele **Fressfeinde** und kann sich wiederum selber von vielen anderen Tierarten ernähren.

### a) Entity-Relationship-Diagramm

Erstellen Sie bitte ein Entity-Relationship-Diagramm, das die oben skizzierten Sachverhalte wiedergibt. Charakterisieren Sie dabei bitte insbesondere die Beziehung zwischen Meerestieren, Fischen und wirbellosen Tieren genau.

**b) Entity-Relationship-Modell**

Leiten Sie aus dem ER-Diagramm bitte ein Entity-Relationship-Modell ab und geben Sie bitte die zugehörigen Entity- und Relationship-Deklarationen an.

**Entity-Deklarationen:**

**Relationship-Deklarationen:**

**c) Relationales Modell**

Transformieren Sie bitte das ER-Modell in ein relationales Modell und geben sie bitte entsprechende R-Schema-Definitionen sowie Integritätsbedingungen an.

**d) Normalisierung**

Betrachten Sie bitte die funktionalen Abhängigkeiten der Meerestiere. Zwischen welchen Attributen bestehen funktionale Abhängigkeiten?

In welcher Normalform befindet sich die zugehörige Relationen für Meerestiere Ihres relationalen Modells?

Falls es sich nicht in dritter Normalform befindet: Wie sieht ein verändertes relationales Modell aus, bei dem die Relation für Meerestiere in dritten Normalform ist?

**e) SQL-Datendefinitionen**

Wie sieht die zugehörige Tabellendefinition in SQL für die Tabelle **Becken** aus?

**f) SQL-Anfrage**

Wie sieht eine SQL-Anfrage aus, die diejenigen Becken ermittelt in denen sich Fressfeinde befinden?



### Aufgabe 3: Begriffe (15 Minuten)

Erläutern Sie bitte die folgenden Begriffe:

- a) Was ist ein **Entity**?
- b) Was ist eine **Entity-Deklaration** und welche allgemeine Form hat sie?
- c) Was ist ein **Attribut** und wodurch ist es charakterisiert?
- d) Was ist ein **Fremdschlüssel** und wozu dient er?
- e) Was ist **null** und wofür wird es verwendet?