

Aufgabe 1: Definitionen und Begriffe (5 Minuten)

Tabellendstruktur bei IN selekt mit einer Spalte Select mit einem Wert
Kreuzen Sie bitte die richtigen Lösungen an:

a) Welche Bestandteile einer Tabelle gehören zum zeitinvarianten Teil der Tabelle?

- | | |
|--|--|
| <input type="checkbox"/> Zeilen (Tupel) | <input checked="" type="checkbox"/> Tabellenname |
| <input checked="" type="checkbox"/> Spaltenüberschriften | <input type="checkbox"/> Anzahl der Zeilen |

b) Die Abfrage `SELECT pnr FROM personal;` liefert immer eine Tabelle

- | | |
|--|--|
| <input type="checkbox"/> mit vielen Spalten | <input type="checkbox"/> mit vielen Zeilen |
| <input checked="" type="checkbox"/> mit einer Spalte | <input type="checkbox"/> mit einer Zeile |

c) Die Abfrage `SELECT count(pnr) FROM personal;` liefert immer eine Tabelle

- | | |
|--|---|
| <input type="checkbox"/> mit vielen Spalten | <input type="checkbox"/> mit vielen Zeilen |
| <input checked="" type="checkbox"/> mit einer Spalte | <input checked="" type="checkbox"/> mit einer Zeile |

d) Bei einem `LEFT OUTER JOIN`

- | | |
|---|---|
| <input type="checkbox"/> werden mglw. links Zeilen mit NULL-Werten ergänzt. | <input checked="" type="checkbox"/> werden mglw. rechts Zeilen mit NULL-Werten ergänzt. |
| <input type="checkbox"/> werden Spalten ausgewählt. | <input type="checkbox"/> werden Tabellen hinzugefügt. |

e) Ein Primärschlüssel

- | | |
|--|---|
| <input checked="" type="checkbox"/> ist eine minimal identifizierende Attributkombination. | <input type="checkbox"/> wird stets vom Datenbanksystem vergeben. |
| <input type="checkbox"/> besteht immer aus einer Spalte. | <input checked="" type="checkbox"/> darf niemals NULL sein. |

f) Grundelemente von ER-Diagrammen sind

- | | |
|---|--|
| <input checked="" type="checkbox"/> Ovale für Attribute | <input checked="" type="checkbox"/> Rauten für Beziehungen |
| <input type="checkbox"/> Herzen für Beziehungen | <input type="checkbox"/> Rechtecke für Attribute |

Aufgabe 2: SQL (30 Minuten)

Wir betrachten die in der Vorlesung behandelte Datenbank *Firma* mit den Tabellen *Maschine*, *Personal*, *Gehalt*, *Kind*, *Abteilung* und *Prämie*. Beispieltabellen, aus denen sich auch das Datenbankschema ablesen lässt, finden sich auf der letzten Seite dieser Aufgabenstellungen. Sie dürfen dieses Blatt gerne abtrennen.

Schreiben Sie bitte SQL-Anweisungen, um die folgenden „Fragen“ zu beantworten. Wo verlangt, geben Sie bitte auch an, welche Antworten das Datenbanksystem auf Ihre Anfrage hin basierend auf den Beispieltabellen geben würde.

- a) Welche Maschinen besitzt die Firma, die vor 2000 angeschafft wurden? Geben Sie bitte jeweils die Maschinen-Nummer, den Namen der Maschine und das Anschaffungsjahr aus. Sortieren Sie die Ausgabe absteigend nach Anschaffungsjahr.

```
SELECT MNR, NAME, ANSCH_DATUM
FROM MASCHINE
WHERE year(ANSCH_DATUM) < 2000
ORDER BY ANSCH_DATUM DESC;
```

Welche konkrete Antwort liefert diese Anfrage?

MNR	NAME	ANSCH-DATUM
12	drehbank	1999
3	fräsmaschine	1998
14	hobelmaschine	1998
1	bohrmaschine	1995

- b) Welche MitarbeiterInnen arbeiten in den Abteilungen *Verwaltung* und *Projektierung*? Geben Sie bitte den Abteilungsnamen (Spaltenüberschrift *Abteilung*) und den Namen und Vornamen der MitarbeiterInnen aus.

Sortieren Sie die Ergebniszeilen bitte aufsteigend nach Abteilungsname und bei gleicher Abteilung aufsteigend nach Name und Vorname des Mitarbeiters/der Mitarbeiterin.

```
SELECT a.NAME as 'Abteilung', p.NAME, p.VORNAME
FROM ABTEILUNG as a JOIN PERSONAL as p using(pnr)
WHERE a.NAME in ('Verwaltung', 'Projektierung')
ORDER BY a.NAME, p.NAME, p.VORNAME;
```

Welcher Konflikt in den Spaltennamen tritt hier auf? Wie löst man ihn?

Es gibt zwei Spalten mit dem selben Namen, daher müssen vorher die Tabellennamen indiziert genannt werden.

- c) Für **jeden Mitarbeiter (Name, Vorname)** und jede Mitarbeiterin soll ermittelt werden, **wieviele Prämien** er oder sie bekommen hat (**Spaltenname Prämienanzahl**). MitarbeiterInnen ohne Prämie sollen dabei mit **Prämienanzahl 0** auftreten. Das Ergebnis soll **absteigend nach Anzahl der Prämien** sortiert sein, bei gleicher Anzahl **alphabetisch aufsteigend** nach Name und Vorname.

```
SELECT p.NAME, p.VORNAME, coalesce (count(pr.pnr), 0) as 'Prämienanzahl'
FROM PERSONAL as p LEFT JOIN PRAEMIE as pr on p.pnr = pr.pnr
GROUP BY p.pnr
ORDER BY count(pr.pnr) desc, p.NAME, p.VORNAME;
```

d) Beantworten Sie zunächst die Fragen nach den konkreten Werten:

1. Wie hoch ist konkret das höchste Gehalt? (Zahlenwert als Antwort erwartet)

3782

2. In welche konkreten Gehaltsstufe wird dieses höchste Gehalt gezahlt?
(Gehaltsstufen-Kürzel als Antwort erwartet)

it5

3. An welche konkreten Mitarbeiter wird diese Gehaltsstufe gezahlt? (Personalnummern als Antwort erwartet)

124, 134

4. In welchen konkreten Abteilungen arbeiten diese Mitarbeiter? (Abteilungs-
namen als Antworten erwartet)

Verwaltung, Produktion

Erstellen Sie nun eine Datenbankabfrage hierzu: Benutzen Sie dabei bitte Unterabfragen und **vermeiden** Sie Joins (auch keine Join-Bedingungen in WHERE):

In welchen Abteilungen wird das höchste Gehalt gezahlt?

```
SELECT a.NAME
FROM ABTEILUNG as a
WHERE EXISTS ( SELECT *
                FROM PERSONAL as p
                WHERE EXISTS ( SELECT g.GEH-STUFE
                              FROM GEHALT as g
                              WHERE g.GEH-STUFE = p.GEH-STUFE AND
                                    g.BETRAG = ( SELECT max(gehalt.Betrag)
                                                  FROM gehalt))))
```

Sind Ihre Unterabfragen korreliert oder unkorreliert? Bitte begründen Sie Ihre Antwort.

korreliert, da in der letzten Subquery immer wieder aus der äußeren Query zugegriffen wird, nämlich bei $(g. \text{geh-stufe} = p. \text{geh-stufe})$

- e) Wie würde die Abfrage „In welchen Abteilungen wird das höchste Gehalt gezahlt?“ aus d) mit Joins aussehen?

```
SELECT a.NAME
FROM ABTEILUNG as a join PERSONAL as p using(abt-nr) join GEHALT as g using(geh-stufe)
WHERE g.betrag = (SELECT max(BETRAG
                      FROM GEHALT)
GROUP BY a.NAME;
```

Aufgabe 3: Datenbankentwurf (25 Minuten)

Wintermarkt

Ein Veranstalter möchte die Platzvergabe der Stände auf einem Wintermarkt mit einer Datenbank verwalten.

Dazu sollen Informationen über Inhaber, Marktstände, die zu vergebenden Plätze erfasst werden. Von den Inhabern der Marktstände wird ihr Vorname und Name aufgenommen. Ihre Email-Adresse erlaubt ihre eindeutige Unterscheidung, so dass auch mehrere Inhaber gleichen Namens (etwa Thomas Müller) verwaltet werden können.

Die Plätze auf dem Wintermarkt werden durch eine Platznummer festgelegt und haben unterschiedliche Größen, die durch Breite und Tiefe des Platzes bestimmt sind. Für jeden Tag, auf dem ein Stand auf einem Platz steht, fällt eine Tagesmiete an.

Jeder Marktstand ist durch eine Standnummer identifiziert und besitzt wiederum eine Größe gegeben durch Breite und Tiefe des Standes. (Bei der Vergabe der Plätze können natürlich nur solche Plätze für einen Stand vergeben werden, auf denen er auch ausreichend Raum findet. Diese Einschränkung soll uns hier aber nicht weiter interessieren.) Marktstände sollen zudem eine Beschriftung haben, die ebenfalls erfasst wird.

Jeder Platz wird an höchstens einen Marktstand vergeben und jeder Marktstand bekommt höchstens einen Platz. Es kann vorkommen, dass Plätze unbesetzt bleiben (zu wenig Zuspruch) oder aber auch, dass es Marktstände gibt, für die kein Platz mehr vergeben werden kann (zu viel Zuspruch).

Ein Inhaber kann mehrere Marktstände betreiben. Jeder Marktstand wird aber nur von genau einem Inhaber betrieben.

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 Inhaber, Marktstand und Platz genau. Geben Sie bitte auch die Kardinalitäten der Beziehungstypen an.

