

1.) Wissensfragen

- a) Wieso nutzt man prepared Statements in JDBC in bezug auf Sicherheit
→ Sql Injection
- b) Ein Update Statement will auf eine Tabelle mit einem Trigger zugreifen.
Fehlermeldung: "Tabelle wird gerade geändert, Trigger bemerkt evtl. nicht was passiert"
"Was kann diese Fehlermeldung bedeuten?"
- c) Was ist statisches SQL?
- d) Wofür werden intentionelle Sperren genutzt ,nenne ein Beispiel!

- e) Wie kann man mit JDBC SQL Injections vermeiden?
- f) Ein Update Statement will auf eine Tabelle mit einem Trigger zugreifen.
Fehlermeldung: "Tabelle wird geändert, Trigger bemerkt Änderung evtl. nicht."
Auf was weist die Fehlermeldung hin?
- g) Kann statisches mit dynamischen SQL gemischt werden? Begründen Sie.
- h) Wie viele Relationen gibt es bei der Vererbungsstrategie Universalrelation pro Vererbungshierarchie?
Nennen Sie jeweils 2 Vor- und Nachteile dieser Vererbungsstrategie.

2.) SQL Queries (15 Punkte)

a.)

Ein Fußball Team besteht aus einem Namen, einer ID und der Liga in der es Spielt. Die Teams werden anhand der Punktzahl eingeordnet, bei Gleichstand wird die Tordifferenz genutzt. Ist hier Gleichstand besetzen zwei Teams den selben Platz und es wird weiter gezählt.

```
CREATE TABLE teams(  
    team_id number primary key,  
    team_name varchar  
    liga_name varchar,  
    Punkte number,  
    Tordifferenz number  
)
```

- a) Wählen Sie alle Teams mit einer Punktzahl über 40 aus
- b) Wählen Sie die Anzahl der Teams pro Liga und den Namen der Liga aus
- c) Wählen Sie alle Teams
- d) Wählen Sie Teamname, Liganame, Punkte pro Team und durchschnittliche Punkte pro Liga (verwenden Sie analytische Funktionen)

```
b.) CREATE TABLE Zootiere(  
    tier_id number(8) primary key name varchar(20)  
    art varchar(20)  
    gewicht number(8) //in kg
```

- a) Geben Sie alle Tiernamen aus die ein Gewicht von mehr als 100kg haben
- b) Geben Sie die Tierarten an bei denen das Gewicht durchschnittlich über 100kg ist
- c) Geben Sie alle Tiernamen aus, bei denen die Tierart durchschnittlich mehr als 100kg wiegt (Ohne Analytische Funktion!)
- d) Geben Sie alle Tiernamen aus, bei denen die Tierart durchschnittlich mehr als 100kg wiegt (mit Analytische Funktion!)

2.1 Analytische Funktionen

- a) ROW_NUMBER () OVER (ORDER BY WERT1)
- b) RANK () OVER (ORDER BY WERT1)
- c) DENSE_RANK () OVER (ORDER BY WERT1)
- d) SUM(WERT2) OVER (PARTITION BY WERT1)
- e) SUM(WERT2) OVER (PARTITION BY WERT1 ORDER BY ID)

ID	WERT1	WERT2	a.)	b.)	c.)	d.)	e.)
1	3	2					
2	5	8					
3	4	10					
4	1	1					
5	2	8					
6	5	4					
7	4	9					
8	4	9					
9	4	6					
10	2	8					

3.) JDBC (15 Punkte)

a.)

```
create table mitarbeiter (  
    id number primary key,  
    name, varchar,  
    gehalt number,  
    bonus number  
)
```

Gesucht wird eine JDBC Funktion

void bestimme_bonus(int mitarbeiter_id, double bonus)

- Der Bonus darf nicht negativ sein und auch nicht 50% des Gehaltes übersteigen. Denken Sie an Fehlerbehandlung und werfen Sie exceptions wo nötig
 - Denken Sie an das Schließen von Ressourcen
 - Es herrscht mehrfachbetrieb
 - Die Datenbank ist auf Read-commited eingestellt
 - Für die Transaktionssteuerung gibt es das statische Connection conn Element.
-

b.)

- Kunde (PK-Kunden_id)
- Fahrrad(PK-Fahrrad_id, FK-Kunden_id)

Schreibe eine Methode in JDBC die das Ausleihen handelt und Transaktion mit händelt.

void ausleihen(int fahrrad_id, int kunde_id) {...}

- Sie soll prüfen, ob das Fahrrad verfügbar ist (in Tabelle Fahrrad ist entsprechende Kunde_id=NULL).
- Wenn Fahrrad verfügbar, dann soll der Kunde hinzugefügt werden, sonst soll eine Exception geworfen werden.

4.) JPA (20 Punkte)

a.) JAVA CODE MIT PROJEKT, KUNDE UND GESCHÄFTSKUNDE

Fügen Sie nur die nötigen Annotationen ein. Wenn nötig, streichen Sie Zeilen.

```
public class Projekt{  
    private int id;  
    private String name;  
    private Kunde kunde;  
}
```

```
public class Kunde{  
    private int id;  
    private String name;  
    private List<Projekt> hatBeauftrag = new ArrayList<Projekt>();  
}
```

```
public class Geschäftskunde extends Kunde{  
    private int id;  
    private String rechtsform;  
}
```

- Welche Abbildungsmöglichkeiten gibt es für Klassen auf Entitäten?
- Was sind "Detached" Objekte? Wozu sind die gut?

b.) Gegeben sind 2 Entities in einem ER-Diagramm.

Kunde <-1-----0...n-> Journey

- int ID	- int ID
- string name	- string name
- <nochwas>	- <nochwas>

- Schreiben Sie den Java Code mit allen nötigen Annotationen für JPA, damit das ER-Diagramm so genau wie möglich abgebildet wird.
- Methoden können vernachlässigt werden
- Alle Entitäts-Tabellen sollen in der Datenbank mit "E" beginnen.
- Alle Attribute sollen in der Datenbank mit "A" beginnen.

5.) Transaktion (20 Punkte)

a.) Gegeben sind zwei Schedules mit je zwei bzw. drei Transaktionen

- Sind die Schedules Konfliktserialisierbar? Begründen Sie mit einem Abhängigkeitsgraphen.
- Sind die Schedules Seriell? Begründen Sie kurz

S1		S2		

- Geben Sie den Ablauf der Schedules an, wenn sie ein Striktes Zwei Phasen Sperrprotokoll anwenden
 - Was ist ein Lost Update und wieso wird es durch serialisierbarkeit verhindert?
-

b.) Gegeben ist ein Schedule mit 3 Transaktionen.

- Prüfen Sie ob der Schedule konflikt-serialisierbar ist.
(Zeichnen Sie den Abhängigkeitsgraphen.)
- Füllen Sie die Tabelle für das Zeitstempelverfahren mit dem gegebenen Schedule durch.
(Klappt das? Begründen.)
- Füllen Sie die Tabelle für das strikte 2PL Protokoll mit Mehrfachmodi-Sperren durch.
(Klappt das? Begründen.)
- Gegeben ist folgender Schedule.
Zeichnen Sie ein an welchen Stellen Aborts/Commits eingefügt werden müssen, um
 - a) einen kaskadierenden Abbruch zu verursachen
 - b) den Schedule nicht rücksetzbar zu machen

6.) Trigger (10 Punkte)

a.)

- a) Beantworten sie kurz und präzise was der Unterschied zwischen einem Zeilentrigger und einem Statement-trigger ist
 - b) Wenn zwei Trigger auf das gleich Ereignis ausgelöst werden: In welcher Reihenfolge werden die Trigger abgearbeitet? Wie wirkt sich das auf das Programm aus?
-

b.) Erklären sie was hierbei genau passiert!

```
CREATE OR REPLACE TRIGGER trg_projekt
AFTER DELETE OR UPDATE OF projektnr ON Projekte_Angestellte
DECLARE anzAngest NUMBER;
BEGIN
    SELECT min(cnt) INTO anzAngest FROM
        (SELECT COUNT(*) cnt, abtnr FROM Projekte_Angestellte GROUP BY
projektnr);
    IF (anzAngest < 2) THEN
        RAISE_APPLICATION_ERROR(num => -20000,
            msg => 'Angest2Min verletzt!')
    );
    END IF;
END;
```

7.) Sicherheit (10)

a.) Geben Sie die GRANT Befehle für die Situationen.

Wenn ein GRANT nicht genügt, geben Sie eine Lösungsskizze an

- a) Die Rolle Mitarbeiter soll die Tabelle Bestellung Lesen können
 - b) Die Rolle Mitarbeiter soll die Tabelle X lesen können, aber ohne die Spalte Y
 - c) Welche Zwei Anmeldemöglichkeiten gibt es bei einer Datenbank? Nennen Sie je einen Vorteil
-

b.) Geben Sie den Befehl an, um folgende SQL Berechtigungen zu ermöglichen.

Falls es keinen geeigneten Befehl gibt, skizzieren Sie einen Lösungsweg.

- a) Die Rolle Mitarbeiter soll Leserechte auf die Tabelle Angestellte haben.
- b) Die Rolle Mitarbeiter soll alle Spalten außer der Spalte Gehalt lesen können.
- c) Die Rolle Mitarbeiter soll alle Aufträge updaten dürfen, außer Aufträge, die im Status "Abgeschlossen" sind.