

1. SQL vs. Relationale Algebra

a)

1.

```
SELECT PersID, Vorname, Alter
FROM Person
WHERE Alter<25;
```
2.

```
SELECT Stadt
FROM Mechaniker
NATURAL JOIN (
    SELECT *
    FROM Repariert
    WHERE Grund = 'def.Bremse'
) AS A;
```
3.

```
SELECT Alter, Bezeichnung
FROM Auto
JOIN Repariert
ON AutoID = Auto
WHERE Datum > '1980-08-20';
```
4.

```
SELECT A.Auto
FROM Besitzt AS A
JOIN Besitzt AS B
ON (A.Auto = B.Auto AND A.KAUFDATUM != B.Kaufdatum);
```

2. SQL vs. Umgangssprache

a)

1.

```
SELECT count(*)
FROM Auto
WHERE Bezeichnung LIKE '%BMW%';
```
2.

```
SELECT Stadt, avg(Gehalt)
FROM Mechaniker
GROUP BY Stadt
HAVING avg(Gehalt)<3000;
```
3.

```
WITH mehralszweiautos AS (
    SELECT Besitzer
    FROM Besitzt
    GROUP BY Besitzer
    HAVING count(*)>2
)
SELECT avg(Alter)
FROM mehralszweiautos
JOIN Person
ON Besitzer=PersID;
```

```

4.    WITH mehrals10 AS (
        SELECT PersID
        FROM Mechaniker
        GROUP BY PersID
        HAVING count(*)>10
    )
    SELECT PersID, Vorname
    FROM mehrals10
    NATURAL JOIN Person;

```

b)

- Die IDs der Personen, die entweder Peter oder Lukas Schneider heißen.
- Vor- und Nachnamen aller Mechaniker
- Vor- und Nachnamen der Autobesitzer, die ihr Auto schon einmal selbst repariert haben, in aufsteigender Reihenfolge nach Nachnamen geordnet

3. SQL Fehlersuche

1. Reihenfolge falsch, FROM kommt vor WHERE; Alter ist kein Schlüsselwort
2. im WHERE fehlt Präfix, schreibe M.PersID oder P.PersID
3. WHERE durch HAVING ersetzen; PersID darf zudem nicht im SELECT stehen weil nicht darüber gruppiert wurde (und ist falsch geschrieben), besser PARTITION BY benutzen
4. Unteranfrage muss Namen bekommen (mit AS); schlägt fehl da „Durchschnitt“ bei Auswertung von HAVING noch nicht definiert (SELECT wird erst danach)

4. Tabellen erstellen mit SQL

```

a)    CREATE TABLE User (
        Mail          character varying PRIMARY KEY
        Nickname      character varying UNIQUE
        Passwort      character varying
    );

    CREATE TABLE Post(
        ID            integer
        Author        character varying REFERENCES User
        Text          character varying NOT NULL
    );

b)    INSERT INTO User
    VALUES      ("chris@infosys.de", "chris", "123456"),
                ("marv@infosys.de", null, "äsd")

    INSERT INTO Post
    VALUES      (1, "chris@infosys.de", "Hallo Welt"),
                (2, "marv@infosys.de", "Hallo Christian")

c)    UPDATE Post
    SET Text="HALLO CHRISTIAN"
    WHERE ID=2;

```