

Aufgabe 1: Oracle-SQL, 9 Punkte (3+2+4)

a)

```
SELECT datum GREATEST(wert1, wert2, wert3, wert4, wert5, wert6) gr,  
       LEAST(wert1, wert2, wert3, wert4, wert5, wert6) l  
FROM messwerte;
```

b)

```
SELECT datum, wert1, wert1-LAG(wert1, 1)  
       (ORDER BY datum);
```

c)

```
CREATE ROLE R1  
CREATE USER leser NOT IDENTIFIED  
GRANT select(datum, wert) ON R1 TO leser
```

Aufgabe 2: Trigger, 8 Punkte(2+6)

a)

```
CREATE TRIGGER OR REPLACE trigger1  
  BEFORE UPDATE ON angestellte  
  WHEN new.gehalt != old.gehalt  
Begin  
  INSERT INTO archive(:old.pers_id, :old.gehalt, :new.gehalt, to_date(now()), :new.kommentar)  
END
```

b)

```
CREATE TRIGGER OR REPLACE trigger2  
  AFTER INSERT ON archive  
  WHEN old.gehalt=new.gehalt  
Begin  
  UPDATE archive SET kommentar = 'Gehalt identisch zum' || :old.datum  
  WHERE :old.pers_id = :new.pers_id AND  
        datum = :new.datum
```

Aufgabe 3 Objekte in Oracle, 9 Punkte (3+2+2+2)

a)

```
CREATE OR REPLACE type_zimmer
CREATE TYPE zimmer_t AS OBJECT(nummer INTEGER, bettenanzahl INTEGER);
CREATE TYPE zimmer_tab IS TABLE OF zimmer_type;
CREATE TABLE hotel(tel INTEGER, name VARCHAR(20), zimmer zimmer_tab);
NASTED TABLE zimmer STORE AS hotel_zimmer;
```

b)

```
INSERT INTO hotel VALUES(
    5550101, 'Zum Orakel',
    zimmer_tab(1,2),
    zimmer_tab(1,2),
    zimmer_tab(1,2));
```

c)

```
SELECT zimmer_tab.nummer
    WHERE zimmer_tab.bettenanzahl=2
    AND name='Zum Orakel'
    FROM hotel;
```

d)

```
SELECT name, SUM(zimmer_tab.bettenanzahl) s FROM hotel h,
    TABLE(h.zimmer) z ORDER BY s desc;
```

Aufgabe 4: Rekursion in Orakle, 6 Punkte (3+1+2)

a)

```
SELEKT parent, child FROM tree
    START WITH parent = 1
    CONNECT BY parent = PRIOR child;
```

b)

```
SELECT child, LEVEL FROM tree
  CONNECT BY parent = PRIOR child;
```

c)

```
SELECT child, CONNECT_BY_ISLEAF
  FROM tree;
```