

Aufgabe 1

- a. `WITH RECURSIVE OberGeo(gid, gname, gparent)`
`AS (`
`SELECT * FROM hierarchie`
`WHERE id IN (SELECT parent FROM hierarchie WHERE name = 'Geo-Datenbanken')`
`UNION`
`SELECT hierarchie.* FROM OberGeo, hierarchie`
`WHERE OberGeo.gparent = hierarchie.id`
`)`
`SELECT * FROM OberGeo;`
- b. `WITH RECURSIVE Tree(rid, rname, rparent)`
`AS (`
`SELECT * FROM hierarchie`
`WHERE name = 'Datenbanksysteme I'`
`UNION`
`SELECT hierarchie.* FROM Tree, hierarchie`
`WHERE Tree.rparent = hierarchie.id`
`)`
`SELECT count(*) from Tree;`
- c. `SELECT count(*) - 1`
`FROM hierarchie`
`WHERE parent IN (SELECT parent FROM hierarchie`
`WHERE name = 'Grafikprogrammierung II');`
- d. `WITH RECURSIVE UnterHaupt(rid, rname, rparent)`
`AS (`
`SELECT * FROM hierarchie`
`WHERE name = 'Hauptstudium'`
`UNION`
`SELECT hierarchie.* FROM UnterHaupt, hierarchie`
`WHERE hierarchie.parent = UnterHaupt.rid`
`)`
`SELECT * FROM UnterHaupt`
`WHERE rname LIKE 'S%';`

Aufgabe 2

- a.

```
UPDATE GEHALT
SET jahresgehalt = jahresgehalt - 2000
WHERE jahresgehalt > 70000;
```

```
UPDATE GEHALT
SET jahresgehalt = jahresgehalt * 1.05
WHERE jahresgehalt <= 70000;
```

- b.

```
UPDATE GEHALT
SET bonus = bonus + 10000
WHERE name LIKE 'M%';
```

Aufgabe 3

- a.

```
SELECT *
FROM GEHALT G
WHERE jahresgehalt >= ALL(SELECT jahresgehalt FROM GEHALT
                           WHERE G.mitarbeiter_id = mitarbeiter_id);
```
- b.

```
SELECT GEHALT.*
FROM GEHALT, (SELECT abteilung, avg(jahresgehalt) as AVGGehalt FROM GEHALT GROUP BY abteilung) T
WHERE GEHALT.abteilung = T.abteilung AND jahresgehalt > T.AVGGehalt;
```
- c.

```
SELECT * FROM GEHALT
WHERE abteilung IN (
    SELECT abteilung, sum(bonus)
    FROM GEHALT
    GROUP BY abteilung
    HAVING sum(bonus) > (SELECT sum(bonus)/count(distinct abteilung) FROM GEHALT));
```