Aufgabe 1

a)

```
SELECT COUNT(tconst) FROM tmovies
```

b)

```
SELECT DISTINCT nb.primaryname
FROM nbasics nb
         JOIN tprincipals tp ON tp.nconst = nb.nconst
WHERE tp.job = 'producer'
ORDER BY nb.primaryname
```

c)

```
SELECT DISTINCT tp1.tconst, tp2.tconst  21 s
FROM tprincipals tp1
         JOIN tprincipals tp2 ON tp1.nconst = tp2.nconst
WHERE tp1.tconst < tp2.tconst
ORDER BY tp2.tconst;
```

d)

```
SELECT nb.primaryname, COUNT(tpall.tconst) AS filmCount
FROM tprincipals tp
         JOIN nbasics nb ON nb.nconst = tp.nconst
         JOIN tprincipals tpall ON tpall.nconst = tp.nconst
WHERE tp.tconst = 'tt1375666'
  AND tp.category IN ('actor', 'actress')
GROUP BY nb.primaryname
ORDER BY filmCount DESC;
```

e)

```
SELECT DISTINCT nb.primaryname, nb.nconst
FROM nbasics nb
         JOIN tprincipals tp ON nb.nconst = tp.nconst
         JOIN tmovies tm ON tm.tconst = tp.tconst
WHERE tm.genres @> ARRAY ['Action']
  AND nb.primaryname LIKE 'T%'
```

f)

```
SELECT DISTINCT nb.primaryname, nb.nconst
FROM nbasics nb
         JOIN tprincipals tp1 ON nb.nconst = tp1.nconst
         JOIN tmovies tm1 ON tm1.tconst = tp1.tconst
WHERE tm1.genres @> ARRAY ['Action']
  AND nb.primaryname LIKE 'T%'
  AND NOT EXISTS (SELECT 1
                  FROM tprincipals tp2
                           JOIN tmovies tm2 ON tm2.tconst = tp2.tconst
                  WHERE tp2.nconst = nb.nconst
                    AND NOT (tm2.genres @> ARRAY ['Action']));
```

g)

```
SELECT tm.tconst, tm."primaryTitle", tm."startYear", tr.averagerating
FROM tmovies tm
         JOIN tratings tr ON tr.tconst = tm.tconst
WHERE tm.tconst IN (
    SELECT tp1.tconst
    FROM tprincipals tp1
    WHERE tp1.characters ILIKE '%Peter Pan%'
)
  AND tm.tconst IN (
    SELECT tp2.tconst
    FROM tprincipals tp2
    WHERE tp2.job = 'producer'
    GROUP BY tp2.tconst
    HAVING COUNT(*) >= 3
)
ORDER BY tr.averagerating DESC;
```

h)

```sql
SELECT tm."startYear", COUNT(*) AS filmAnzahl
FROM tmovies tm
WHERE tm."startYear" = (SELECT MAX("startYear") FROM tmovies)
GROUP BY tm."startYear";
```

```sql
SELECT tm."startYear", COUNT(*) AS filmAnzahl
FROM tmovies tm
GROUP BY tm."startYear"
HAVING COUNT(*) = (
    SELECT MAX(filmCount)
    FROM (
            SELECT COUNT(*) AS filmCount
            FROM tmovies
            GROUP BY "startYear"
        ) AS jahr_counts
);
```

Aufgabe 2

a)

```
SELECT Name, Kontinent
FROM Land
WHERE Bevölkerung > 200000000
```

b)

```
SELECT l.Name
FROM Land l
         JOIN Stadt s ON s.LandID = l.LandID
WHERE s.StadtName = l.Hauptstadt
  AND (Bevölkerung < 2 * s.p1959 OR Bevölkerung < 4 * s.p2000)
```

c)

```
SELECT Name
FROM Land l
         JOIN Geographie g1 ON l.LandID = g1.LandID
WHERE NOT EXISTS (SELECT 1
                  FROM Geographie g2
                  WHERE g2.urbar > g1.urbar)
```

Aufgabe 3

a)

Gebe für jedes Land den Namen des Landes und den Gesamtumsatz an, der durch Kunden in diesem Land erzielt wurde, indem sie bei Lieferanten aus demselben Land Bestellungen aufgegeben haben.

Dabei soll berücksichtigt werden, dass nur Bestellungen aus dem Jahr 1992 und der Region Europa mit einfließen dürfen.

b)

Gebe das Land, die Nation, das Jahr und den Umsatz für das jeweilige Jahr an, welches durch grenzüberschreitende Lieferungen zwischen Deutschland und der USA zwischen 1995 und 1996 erzieht wurde.

Aufgabe 4

a)

Teil 1

```
CREATE VIEW lineitemalachocolate AS
SELECT *
FROM lineitem l
         JOIN part p ON l.l_partkey = p.p_partkey
WHERE p.p_name = '%chocolate%';
```

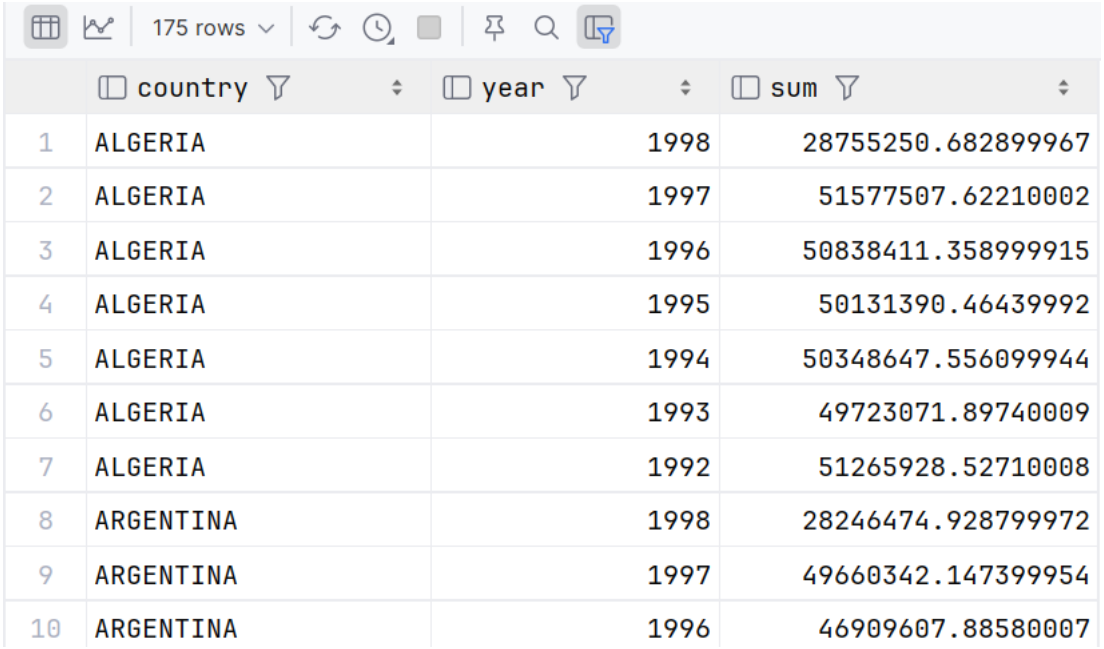
Teil 2

```
CREATE VIEW pricePerCountYearAndPart AS
SELECT DISTINCT n.n_name
                EXTRACT(year FROM o.o_orderdate)
                l.l_extendedprice * (1::double precision - l.l_discount) - ps.ps_supplycost *
                                                                           l.l_quantity
FROM lineitemalachocolate l
         JOIN orders o ON l.l_orderkey = o.o_orderkey
         JOIN supplier s ON l.l_suppkey = s.s_suppkey
         JOIN nation n ON s.s_nationkey = n.n_nationkey
         JOIN partsupp ps ON l.p_partkey = ps.ps_partkey AND s.s_suppkey = ps.ps_suppkey
ORDER BY n.n_name, (EXTRACT(year FROM o.o_orderdate)) DESC
```

Teil 3

```
SELECT cyp.country, cyp.year, sum(cyp.profit_per_lineitem)
FROM "pricePerCountryYearAndPart" cyp
GROUP BY cyp.country, cyp.year
ORDER BY country, year DESC;
```

Ausgabe und Anzahl Tupel (175 rows)



175 rows

	country	year	sum
1	ALGERIA	1998	28755250.682899967
2	ALGERIA	1997	51577507.62210002
3	ALGERIA	1996	50838411.358999915
4	ALGERIA	1995	50131390.46439992
5	ALGERIA	1994	50348647.556099944
6	ALGERIA	1993	49723071.89740009
7	ALGERIA	1992	51265928.52710008
8	ARGENTINA	1998	28246474.928799972
9	ARGENTINA	1997	49660342.147399954
10	ARGENTINA	1996	46909607.88580007

b)

Teil 1

```
CREATE VIEW germanSupps AS
SELECT s.s_suppkey
FROM supplier s
         JOIN nation n ON s.s_nationkey = n.n_nationkey
WHERE n.n_name = 'GERMANY';
```

Teil 2

```
CREATE VIEW germanpartswithcount AS
SELECT ps.ps_partkey,
       sum(DISTINCT ps.ps_availqty) AS totalcount
FROM partsupp ps
         JOIN germansupps dl ON ps.ps_suppkey = dl.s_suppkey
GROUP BY ps.ps_partkey
```

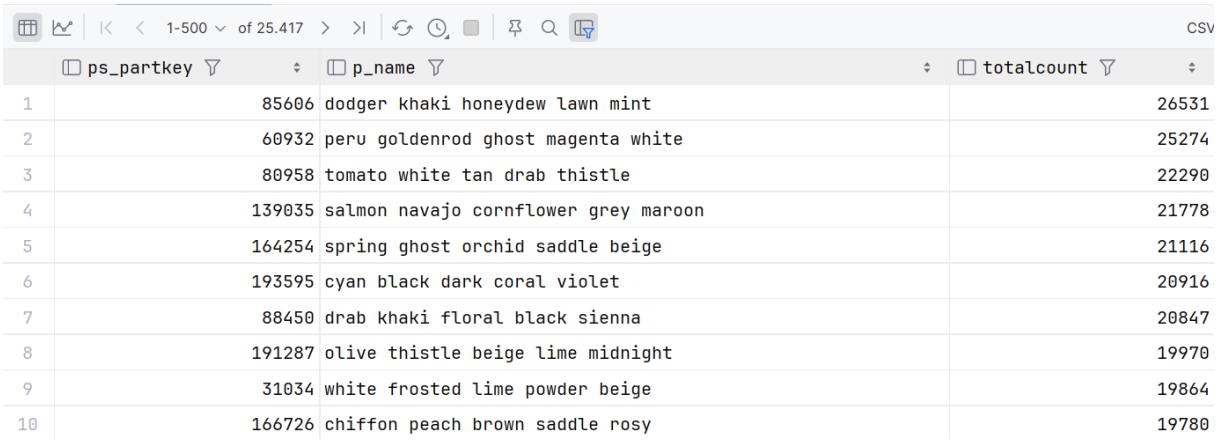
Teil 3

```
CREATE VIEW germantotalcount AS
SELECT sum(germanpartswithcount.totalcount) AS sum
FROM germanpartswithcount
```

Teil 4

```
SELECT DISTINCT
    d.ps_partkey,
    p.p_name,
    d.totalcount
FROM
    germanpartswithcount d
        JOIN part p ON d.ps_partkey = p.p_partkey
        JOIN germantotalcount g ON TRUE
WHERE d.totalcount > g.sum * 0.00001
ORDER BY d.totalcount DESC;
```

Ausgabe & Anzahl Tupel (25.417 rows)



1-500 of 25.417 CSV

	ps_partkey	p_name	totalcount
1	85606	dodger khaki honeydew lawn mint	26531
2	60932	peru goldenrod ghost magenta white	25274
3	80958	tomato white tan drab thistle	22290
4	139035	salmon navajo cornflower grey maroon	21778
5	164254	spring ghost orchid saddle beige	21116
6	193595	cyan black dark coral violet	20916
7	88450	drab khaki floral black sienna	20847
8	191287	olive thistle beige lime midnight	19970
9	31034	white frosted lime powder beige	19864
10	166726	chiffon peach brown saddle rosy	19780