

```

SELECT firstname || ' ' || lastname FROM customer;
SELECT firstname, lastname, country FROM customer WHERE LOWER(country) IN ('sweden', 'denmark');
SELECT firstname, lastname, country FROM customer WHERE country LIKE '%land%';
SELECT * FROM customer WHERE company IS NULL;
SELECT * FROM invoice WHERE invoicedate < TO_DATE('01.02.2009', 'DD-MM-YYYY');
SELECT * FROM invoice WHERE invoicedate BETWEEN '01.02.2012' AND '28.02.2012';

```

```

SELECT e.employeeid FROM employee e
MINUS
SELECT c.supportrepid FROM customer c;
SELECT
    e.employeeid,
    e.firstname || ' ' || e.lastname
FROM
    employee e
WHERE
    e.employeeid NOT IN (
        SELECT
            c.supportrepid
        FROM
            customer c
    );
SELECT
    vg1.bez,
    vg2.bez,
    vg3.bez,
    vg4.bez
FROM verkaufsgebiet vg1
INNER JOIN verkaufsgebiet vg2
    ON vg1.vgid = vg2.obergebiet
INNER JOIN verkaufsgebiet vg3
    ON vg2.vgid = vg3.obergebiet
INNER JOIN verkaufsgebiet vg4
    ON vg3.vgid = vg4.obergebiet
ORDER BY vg3.vgid;

SELECT
    e.employeeid,
    e.firstname || ' ' || e.lastname,
    c.supportrepid,
    c.firstname || ' ' || c.lastname
FROM
    employee e
LEFT JOIN customer c ON e.employeeid = c.supportrepid
WHERE
    c.supportrepid IS NULL;

SELECT *
FROM preisliste pl
INNER JOIN preis ON preis.plid = pl.plid
INNER JOIN produkt ON produkt.pid = preis.pid
INNER JOIN bestellposition bp ON bp.pid = produkt.pid
INNER JOIN bestellung b ON b.bid = bp.bid
WHERE
    bp.einzelpreis <> preis.betrag
    AND b.bestelldatum BETWEEN pl.datum_von AND pl.datum_bis
ORDER BY bp.bid, bp.pid;

SELECT
    vgid,
    bez,
    CASE
        WHEN obergebiet IS NULL THEN 'Wurzel'
        WHEN ebene IN (SELECT MAX(ebene) FROM verkaufsgebiet) THEN 'Blatt'
        WHEN obergebiet IS NOT NULL
            AND
            ebene NOT IN (SELECT MAX(ebene) FROM verkaufsgebiet) THEN 'Innerer Knoten'
        ELSE 'FEHLER'
    END AS knotenart
FROM verkaufsgebiet;

```

```

SELECT
    *
FROM (
    SELECT
        a.artistid,
        a.name,
        COUNT(*) AS anzahl_alben
    FROM
        artist a
    JOIN album al ON a.artistid = al.artistid
    GROUP BY
        a.artistid,
        a.name
    ) inline
WHERE anzahl_alben = 3;

SELECT
    invoiceid,
    total,
    (SELECT SUM(total) FROM invoice) AS gesamtumsatz,
    ROUND(total / (SELECT SUM(total) FROM invoice) * 100, 2) AS prozent
FROM
    invoice;

SELECT
    c.customerid,
    c.firstname || ' ' || c.lastname AS name,
    SUM(i.total) AS umsatz_pro_kunde,
    EXTRACT(YEAR FROM i.invoicedate) AS JAHR,
    COUNT(*) AS anzahl_rechnungen
FROM
    customer c
JOIN invoice i ON c.customerid = i.customerid
GROUP BY
    c.customerid,
    c.firstname || ' ' || c.lastname,
    EXTRACT(YEAR FROM i.invoicedate)
ORDER BY
    c.customerid DESC;

SELECT COUNT(*) FROM employee;
SELECT MAX(unitprice), MIN(unitprice) FROM track;
SELECT SUM(total) FROM invoice;
SELECT ROUND(AVG(milliseconds)/1000,2) FROM track;

```

```

WITH album_spielzeit (albumid, gesamtzeit) AS (
    SELECT
        albumid,
        SUM(milliseconds) / 1000
    FROM
        track
    GROUP BY
        albumid
),
grunddaten (trackname, tracktime, albumid, albumtitle) AS (
    SELECT
        t.name,
        t.milliseconds / 1000,
        a.albumid,
        a.title
    FROM
        track t
    JOIN album a ON t.albumid = a.albumid
)
SELECT
    gd.trackname,
    gd.tracktime,
    gd.albumtitle,
    az.gesamtzeit,
    ROUND(gd.tracktime / az.gesamtzeit * 100, 2) AS prozent_pro_track
FROM
    grunddaten gd
    JOIN album_spielzeit az ON gd.albumid = az.albumid
ORDER BY
    gd.albumid;

WITH bestellungen_januar_2019 (bestell_id, bestelldatum, preis) AS (
    SELECT
        bid,
        bestelldatum,
        gesamtpreis
    FROM
        bestellung
    WHERE
        EXTRACT(YEAR FROM bestelldatum) = 2019
        AND
        EXTRACT(MONTH FROM bestelldatum) = 1
)
SELECT
    *
FROM
    bestellungen_januar_2019
WHERE
    bestelldatum IN (SELECT MIN(bestelldatum) FROM bestellungen_januar_2019)
    AND
    bestell_id IN (SELECT MIN(bestell_id) FROM bestellungen_januar_2019);

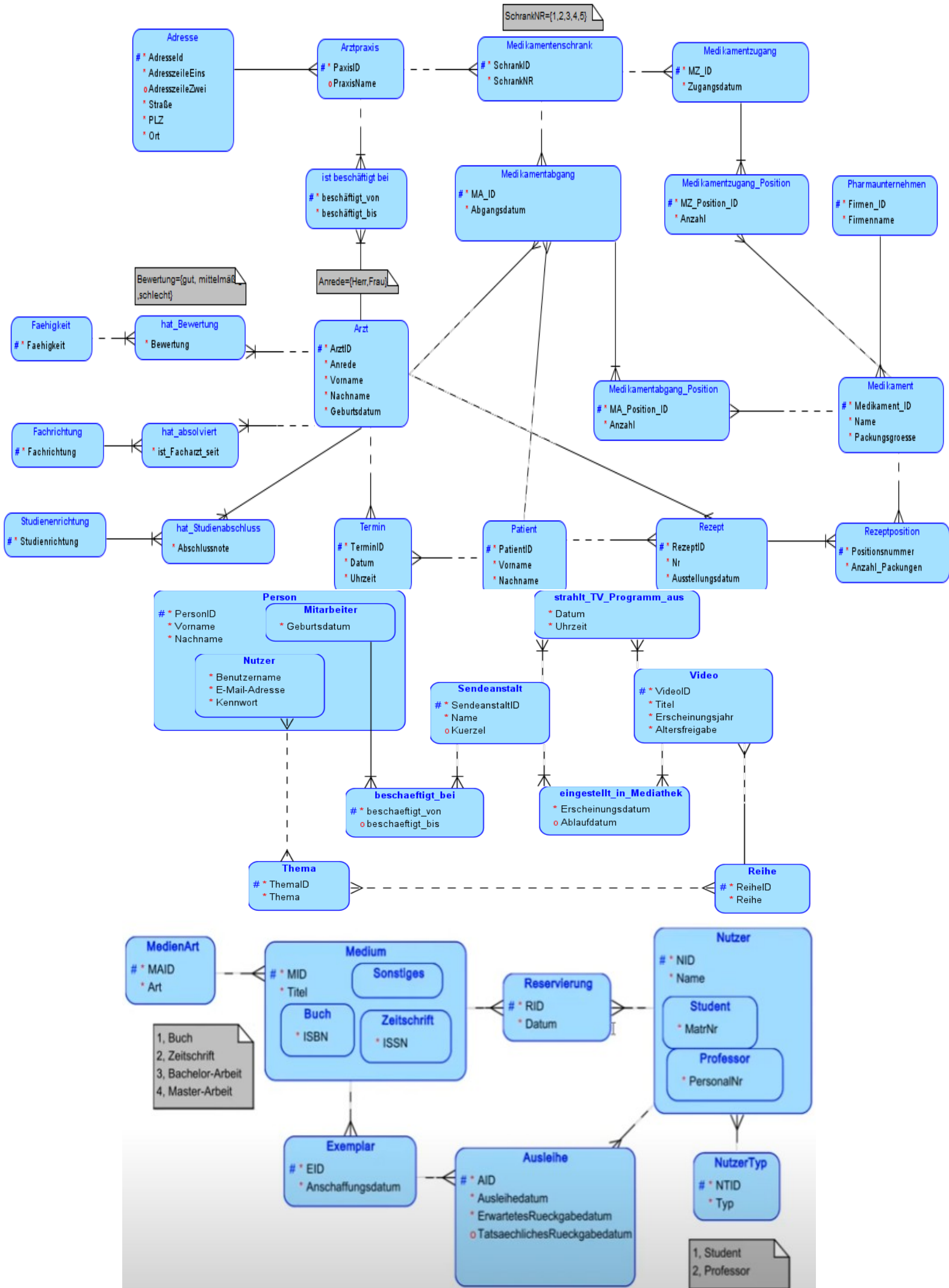
WITH zeit_pro_album (albumid, zeit_album) AS (
    SELECT
        albumid,
        SUM(milliseconds) / 1000
    FROM
        track
    GROUP BY
        albumid
)
SELECT
    t.name,
    ROUND(t.milliseconds / 1000 / zpa.zeit_album * 100, 2) AS prozent,
    a.title,
    zpa.zeit_album AS gesamt_zeit_album
FROM
    zeit_pro_album zpa
    JOIN album a ON zpa.albumid = a.albumid
    JOIN track t ON a.albumid = t.albumid
ORDER BY
    a.albumid;

WITH landumsatz (country, umsatz) AS (
    SELECT
        c.country,
        SUM(i.total)
    FROM
        customer c
    JOIN invoice i ON c.customerid = i.customerid
    WHERE
        c.country LIKE 'A%'
    GROUP BY
        c.country
)
SELECT
    *
FROM
    landumsatz l1, landumsatz l2
WHERE
    l1.umsatz = l2.umsatz
    AND
    l1.country > l2.country;

WITH anzahl_alben_pro_kunstler (artistid, name, anzahl_alben) AS (
    SELECT
        ar.artistid,
        ar.name,
        COUNT(*)
    FROM
        album al
    INNER JOIN artist ar ON al.artistid = ar.artistid
    GROUP BY
        ar.artistid,
        ar.name
)
SELECT
    *
FROM
    anzahl_alben_pro_kunstler
WHERE
    anzahl_alben IN (SELECT MAX(anzahl_alben) FROM anzahl_alben_pro_kunstler);

WITH neben_kategorie (kid, bez, pid) AS (
    SELECT
        k.kid,
        k.bez,
        pk.pid
    FROM
        kategorie k
    JOIN produktkategorie pk ON k.kid = pk.kid
    WHERE
        pk.hk = 'N'
),
haupt_kategorie (kid, bez, pid) AS (
    SELECT
        k.kid,
        k.bez,
        pk.pid
    FROM
        kategorie k
    JOIN produktkategorie pk ON k.kid = pk.kid
    WHERE
        pk.hk = 'J'
)
SELECT
    p.pid,
    p.bez,
    hk.bez AS hauptkategorie,
    NVL(nk.bez, ' ') AS nebenkategorie
FROM
    haupt_kategorie hk
    JOIN produkt p ON hk.pid = p.pid
    LEFT JOIN neben_kategorie nk ON p.pid = nk.pid
WHERE
    p.bez LIKE '%sessel%'
    OR
    p.bez LIKE '%tisch%'
ORDER BY
    p.pid;

```



SQL AUSWERTEN

Adressart	
AAID	Bez
10	Privat
20	Arbeit

Adresse			
AID	PLZ	PID	AAID
1	13297	1	10
2	24188	1	20
3	80900	2	20
4	63452	2	20
5	14187	2	20

Person	
PID	Name
1	Meier
2	Müller

1) Name

```
select Name
from Person P
    inner join Adresse A on P.PID = A.PID
    inner join Adressart AA on A.AAID = AA.AAID
where AA.Bez = 'Privat'
```

Meier

2)

```
select PID, count(*) as Anz
from Adresse A
group by PID
having count(*) >= 2
```

PID ANZ

1 2

2 3

3)

```
select Name
from Person P
where (
    select count(*)
    from
        Adresse A
        inner join Adressart AA on A.AAID = AA.AAID
    where
        P.PID = A.PID and
        AA.Bez = 'Privat'
) > 1
```

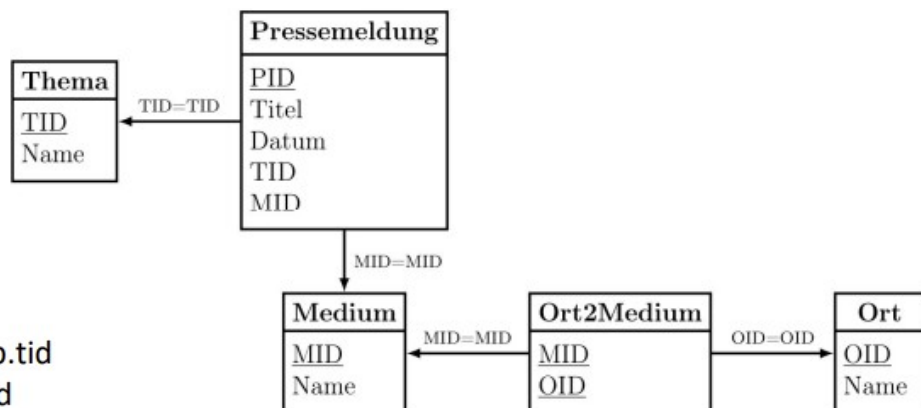
Name

Kein Datensatz

SQL ERSTELLEN

1) Titel der Pressemeldungen zum Thema "Sport", die im "Darmstädter Echo" erschienen sind.

```
select titel
from thema t
    join pressemeldung p on t.tid=p.tid
    join medium m on m.mid=p.mid
where t.name = 'Sport' and m.name='Darmstädter Echo'
```



2) Namen der Medien und Anzahl der Pressemeldungen, die darin erschienen sind, wobei nur Medien aus Darmstadt berücksichtigt werden sollen.

```
select m.name, count(pid)
from medium m
    left join pressemeldung p on m.mid=p.mid
    left join ort2medium o2m on m.mid=o2m.mid
    join ort o on o2m.oid=o.oid
where o.name='Darmstadt'
group by m.name
```


3) **Namen der Medien und Anzahl der Pressemeldungen, die darin erschienen sind, wobei nur Medien aus Darmstadt und Offenbach berücksichtigt werden sollen.**

```
select name, count(pid)
from medium m
      left join pressemeldung p on m.mid=p.mid
where m.mid in
      (select mid
       from ort2medium o2m
        join ort o on o2m.oid=o.oid
       where o.name in ('Darmstadt', 'Offenbach'))
group by m.name
```

4) **Namen der Orte, in denen es die meisten Medien gibt.**

```
with mcount as (
  select name, count(*) as anzahl
  from ort o
        join ort2medium o2m on o.oid=o2m.oid
  group by name
)
select name
From mcount
where mcount.anzahl=
      (select max(anzahl) from mcount)
```

ER-MODELLIERUNG

Patienten haben einen Namen, belegen Zimmer in bestimmten Zeiträumen, wobei jede Belegung durch den Patienten, das Zimmer und den Beginn des Belegungszeitraumes identifiziert wird. Jedes Zimmer hat eine Anzahl von Betten.

Das Krankenhaus besteht aus einer Menge von Stationen, die jeweils eine Bezeichnung (z.B. Chirurgie) haben. Zimmer sind genau einer Station zugeordnet. Des Weiteren hat das Krankenhaus mehrere Operationssäle.

Neben der ärztlichen Betreuung können für einen Patienten verschiedene Maßnahmen durchgeführt werden. Es kann sich hierbei um eine Operation handeln, für die der Durchführungstermin inklusive Start- und Endezeit gespeichert wird sowie der Operationssaal. Das ist notwendig, um die Terminplanungen für Operationen zu unterstützen.

Eine andere Art von Maßnahme ist die Laboruntersuchung, die sich auf eine Untersuchungsposition bezieht und für die das Untersuchungsergebnis gespeichert wird. Untersuchungspositionen (Blutzuckerwerte, Leberwerte, ...) sollen als dynamische Domäne modelliert werden.

Alle Maßnahmen haben einen Termin und einen Preis.

