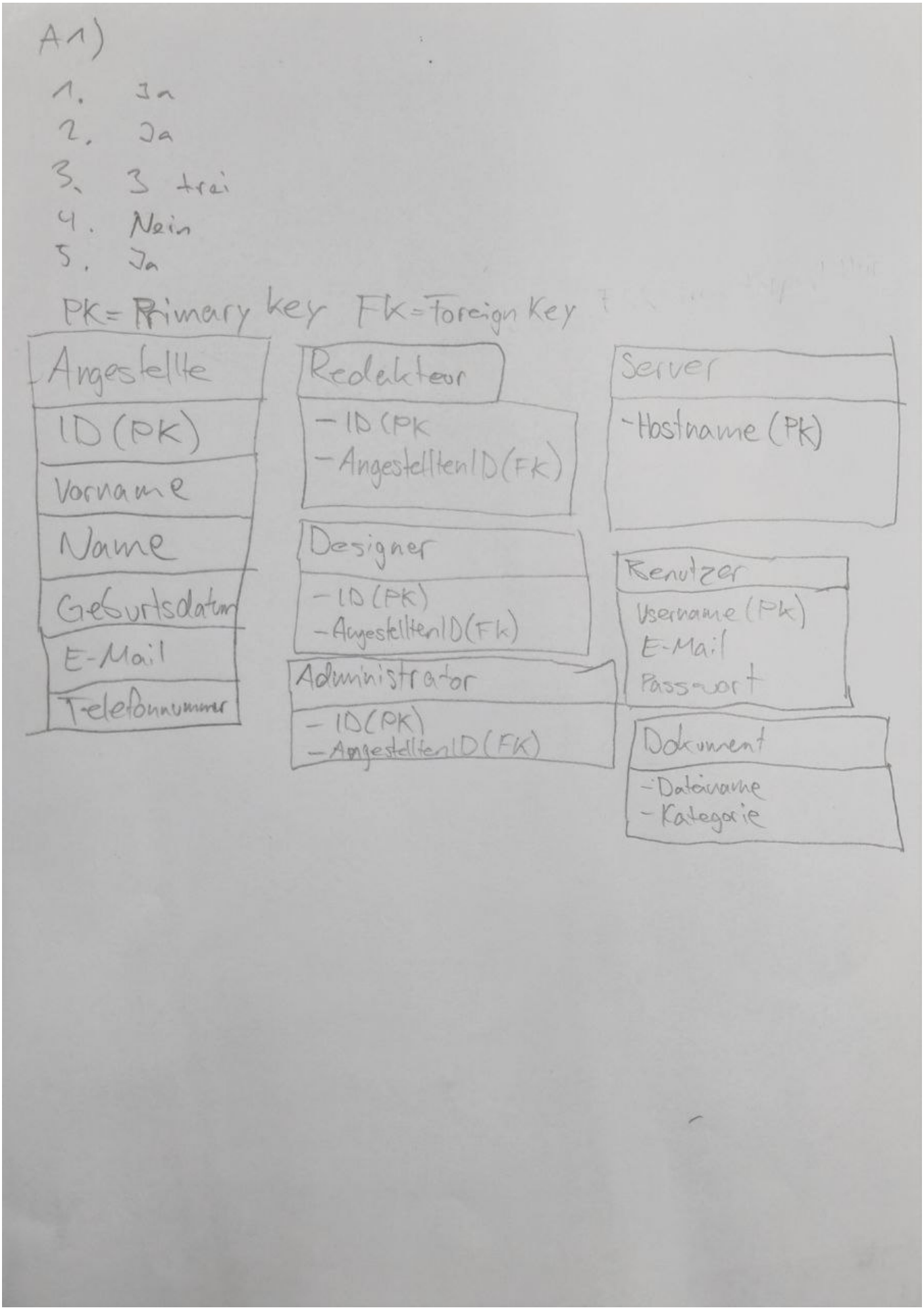
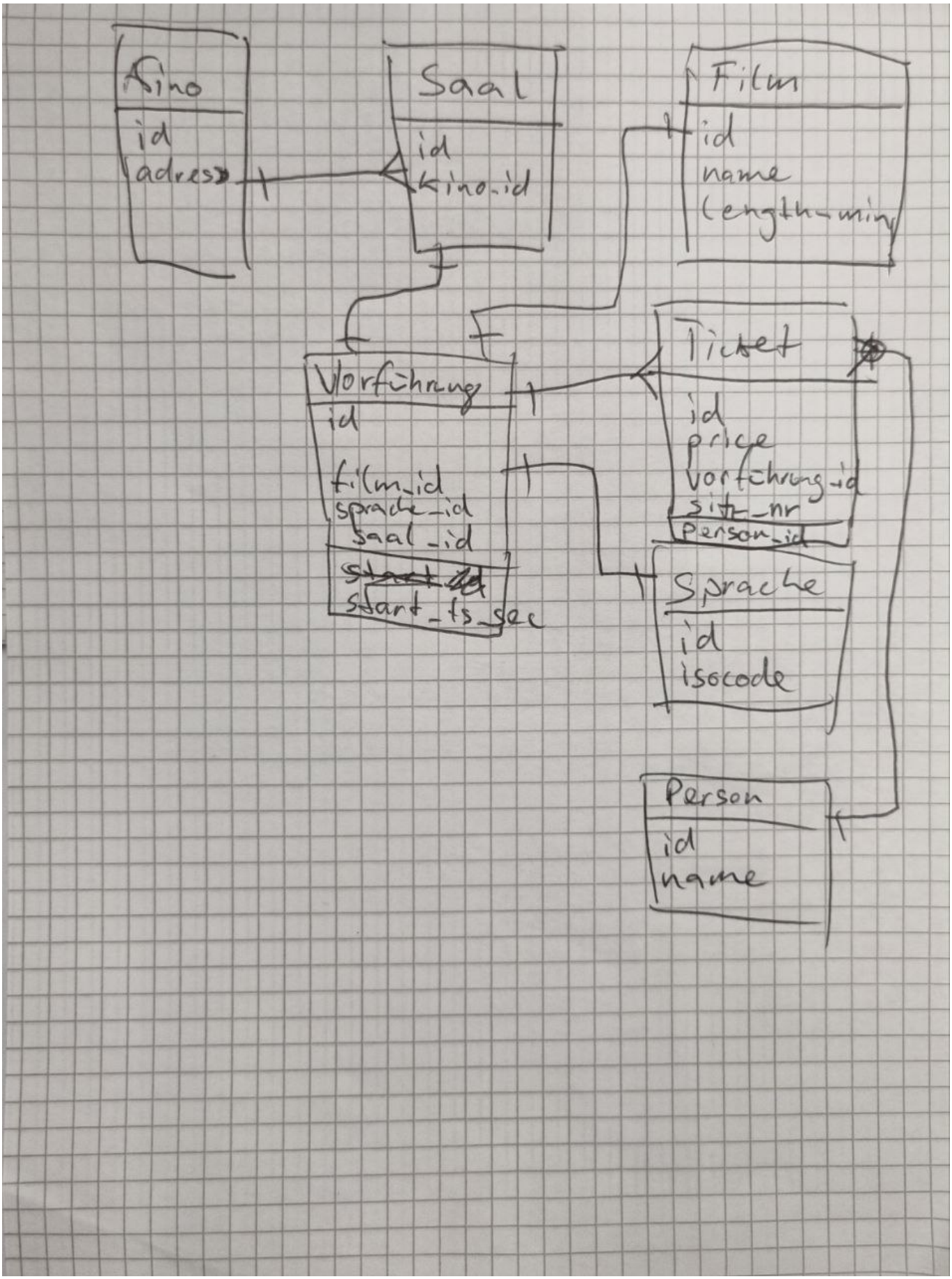
Aufgabe 1


A1)
1. Ja
2. Ja
3. 3 drei
4. Nein
5. Ja
PK=Primary key FK=Foreign Key
Angestellte
ID (PK)
Vorname
Name
Geburtsdatum
E-Mail
Telefonnummer
Redakteur
- ID (PK
- AngestelltenID (FK)
Designer
- ID (PK)
- AngestelltenID (FK)
Administrator
- ID (PK)
- AngestelltenID (FK)
Server
- Hostname (PK)
Benutzer
Username (PK)
E-Mail
Passwort
Dokument
- Dateiname
- Kategorie

Aufgabe 2 (ER-Diagram fur Kinokette)

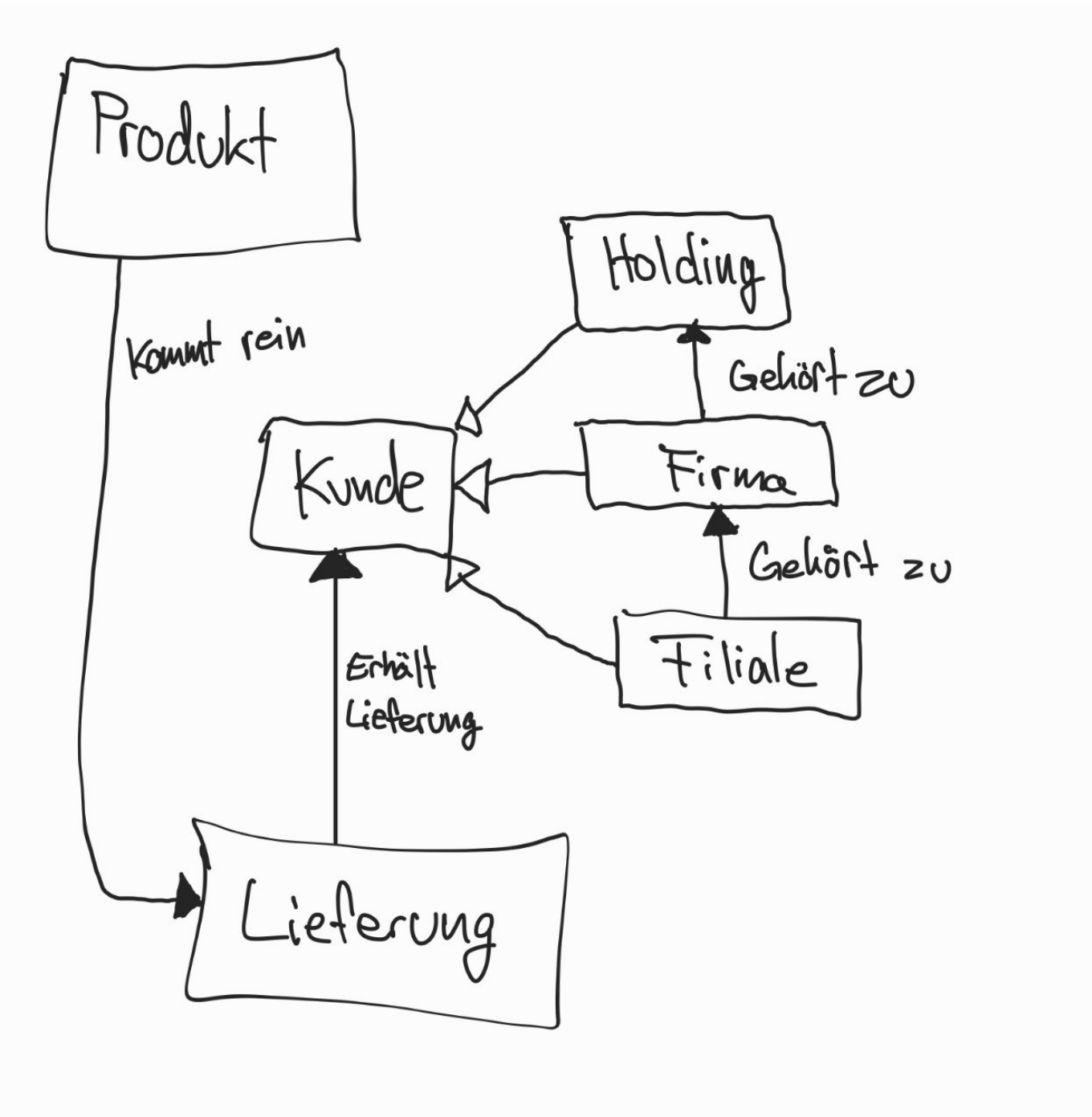


Schema aus ERD

```
CREATE TABLE kino (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    address TEXT NOT NULL,
);


CREATE TABLE saal (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    kino_id INTEGER NOT NULL
    FOREIGN KEY (kino_id)
       REFERENCES kino (kino_id)
);


CREATE TABLE film (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    length_min REAL NOT NULL,
);

CREATE TABLE vorfuehrung (
    id INTEGER PRIMARY KEY,
    film_id INTEGER NOT NULL
    FOREIGN KEY (film_id)
       REFERENCES film (film_id)
    sprache_id INTEGER NOT NULL
    FOREIGN KEY (sprache_id)
       REFERENCES sprache (sprache_id)
    saal_id INTEGER NOT NULL
    FOREIGN KEY (saal_id)
       REFERENCES saal (saal_id)
    start_ts_sec INTEGER NOT NULL
);

CREATE TABLE ticket (
    id INTEGER PRIMARY KEY,
    price REAL NOT NULL,
    vorfuehrung_id INTEGER NOT NULL
    FOREIGN KEY (vorfuehrung_id)
       REFERENCES vorfuehrung (vorfuehrung_id)
);

CREATE TABLE sprache (
    id INTEGER PRIMARY KEY,
    iso_code TEXT NOT NULL,
);

CREATE TABLE person (
    id INTEGER PRIMARY KEY,
```

```
    name TEXT NOT NULL,
);
```

Aufgabe 3



2. DB Schema

```
CREATE TABLE produkt (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
);
```

```
CREATE TABLE lieferung (
    id INTEGER PRIMARY KEY,
    preis TEXT NOT NULL,
    kunde_id INTEGER NOT NULL,
   date DATETIME NOT NULL,
   FOREIGN KEY (kunde_id)
       REFERENCES kunde (kunde_id)

    product_id INTEGER NOT NULL
   FOREIGN KEY (product_id)
       REFERENCES product (product_id)
    holding_id INTEGER
   FOREIGN KEY (holding_id)
       REFERENCES holding (holding_id)
    firma_id INTEGER
   FOREIGN KEY (firma_id)
       REFERENCES firma (firma_id)
    filiale_id INTEGER
   FOREIGN KEY (filiale_id)
       REFERENCES filiale (filiale_id)
);


CREATE TABLE kunde (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    lieferung_id INTEGER
   FOREIGN KEY (lieferung_id)
       REFERENCES lieferung (lieferung_id)
    holding_id INTEGER
   FOREIGN KEY (holding_id)
       REFERENCES holding (holding_id)
    firma_id INTEGER
   FOREIGN KEY (firma_id)
       REFERENCES firma (firma_id)
    filiale_id INTEGER
   FOREIGN KEY (filiale_id)
       REFERENCES filiale (filiale_id)

);

CREATE TABLE holding (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
   adresse TEXT NOT NULL
);
CREATE TABLE firma (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL
   adresse TEXT NOT NULL,
);
CREATE TABLE filiale (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
```

```
    adresse TEXT NOT NULL
);
```

3. sql queries

a.

```
select * from lieferung where date > '2005-01-12'
```

b.

```
select * from lieferung
    join holding on lieferung.holding_id = holding.id
    where holding.name = "ABC Holding"
```

c.

```
select ALL from lieferung
    join holding on lieferung.holding_id = holding.id
    where holding.name = "ABC Holding"
    UNION ALL
    WHERE TRUE;
```