

Phase 2:

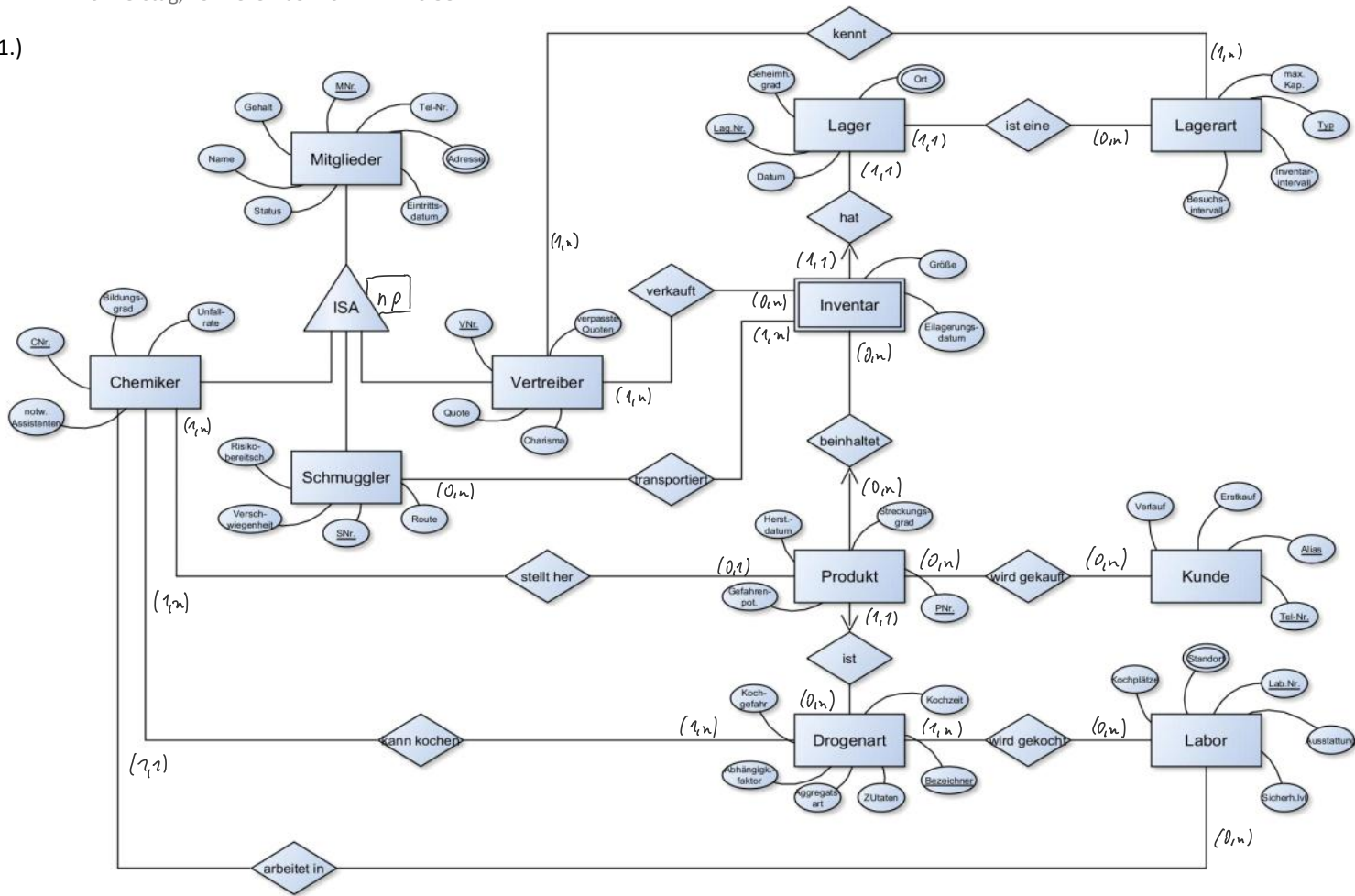
Aufgabe 1 & 2 - korrigiert

Entity-Relationship-Diagramm
&
Relationales Schema

Wandrey, Kaden - Phase 2

Donnerstag, 19. Dezember 2024 20:58

1.)



2.)

RDB:	Mitglieder	(<u>Mitarbeiternummer</u> , Name, Gehalt, Status, Telefonnummer, Adresse, Eintrittsdatum)
	Schmuggler	(<u>Schmugglernummer</u> , Route, Risiko-Bereitschaft, Verschwiegenheit)
	Vertriebsmitarbeiter	(<u>Vertrieblernummer</u> , Charisma, Quote, verpasste Quoten)
	Kunde	(<u>Alias</u> , <u>Telefonnummer</u> , Erstkaufdatum, Verlauf)
	Lagerart	(<u>Lagertyp</u> , Kapazität, Inventar-Wechsel-Intervall, Besuchs-Intervall)
	Drogenart	(<u>Bezeichner</u> , Kochzeit, Zutat, Kapazität, Abhängigkeitsfaktor, Kochgefahr)
	Labor	(<u>Labornummer</u> , Standort, Ausstattung, Kochplätze, Sicherheitsgrad)
	Chemiker	(<u>Chemikernummer</u> , Bildungsgrad, Unfallrate, Assistenten, <u>Labornummer*</u>)

RS:	wird-hergestellt	(<u>Bezeichner</u> , <u>Lab.-Nr.</u>)
	kann-herstellen	(<u>Bezeichner</u> , <u>C.-Nr.</u>)
	Produkt	(<u>P.-Nr.</u> , Preis, Herstellungsdatum, Streckungsgrad, Gefahrenpotenzial, <u>C.-Nr.*</u> , <u>Bezeichner*</u>)
	besitzt	(<u>P.-Nr.</u> , <u>Lag.-Nr.</u>)
	transportiert	(<u>S.-Nr.</u> , <u>Lag.-Nr.</u>)
	Lager-Inventar	(<u>Lag.-Nr.</u> , Startdatum, Geheim. Status, Ort, Größe, Einlagerungs-Datum, <u>Lag.-Typ</u>)
	verkauft	(<u>V.-Nr.</u> , <u>Lag.-Nr.</u>)
	wird-verkauft	(<u>Alias</u> , <u>Tel.-Nr.</u> , <u>P.-Nr.</u>)
	kennt	(<u>V.-Nr.</u> , <u>Lag.-Typ</u>)

Phase 2:

Aufgabe 3

SQL-Skript zur Datenbankdefinition mit allen
Integritätsbedingungen

```
-- Sicherstellen, dass alle Tabellen existieren und keine inkonsistenten
Fremdschlüssel oder CHECK-Fehler auftreten
DROP TABLE Kartell_Kunde CASCADE CONSTRAINTS;
DROP TABLE Kartell_Mitarbeiter CASCADE CONSTRAINTS;
DROP TABLE Chemiker CASCADE CONSTRAINTS;
DROP TABLE Vertriebler CASCADE CONSTRAINTS;
DROP TABLE Schmuggler CASCADE CONSTRAINTS;
DROP TABLE Drogenart CASCADE CONSTRAINTS;
DROP TABLE Labor CASCADE CONSTRAINTS;
DROP TABLE Lager_Inventar CASCADE CONSTRAINTS;
DROP TABLE Lagerart CASCADE CONSTRAINTS;
DROP TABLE Produkt CASCADE CONSTRAINTS;
DROP TABLE Wird_Hergestellt CASCADE CONSTRAINTS;
DROP TABLE Wird_Verkauft CASCADE CONSTRAINTS;
DROP TABLE verkauft CASCADE CONSTRAINTS;
DROP TABLE transportiert CASCADE CONSTRAINTS;
DROP TABLE kennt CASCADE CONSTRAINTS;
DROP TABLE Kann_Herstellen CASCADE CONSTRAINTS;
DROP TABLE besitzt CASCADE CONSTRAINTS;
```

```
-- Tabelle: Kartell-Mitarbeiter
CREATE TABLE Kartell_Mitarbeiter (
  MitarbeiterNummer VARCHAR2(10) PRIMARY KEY,
  Name VARCHAR2(50) NOT NULL,
  Status VARCHAR2(20) CHECK (Status IN ('verfügbar', 'beschäftigt', 'auf
Regierungsliste', 'in Verwahrung')) NOT NULL,
  Gehalt NUMBER(8,2),
  Eintrittsdatum DATE NOT NULL,
  Telefonnummer VARCHAR2(15),
  Adresse VARCHAR2(100) NOT NULL
);
```

```
-- Tabelle: Chemiker (Spezialisierung von Kartell-Mitarbeiter)
CREATE TABLE Chemiker (
  ChemikerNummer VARCHAR2(6) PRIMARY KEY,
  MitarbeiterNummer VARCHAR2(10) NOT NULL,
  Bildungsgrad VARCHAR2(20) CHECK (Bildungsgrad IN ('ohne', 'Diplom', 'Doktor',
'Professor', 'Andere')) NOT NULL,
  Unfallrate NUMBER(3,0) CHECK (Unfallrate BETWEEN 0 AND 100),
  notwendigeAssistenten NUMBER(2,0) CHECK (notwendigeAssistenten BETWEEN 0 AND 99),
  CONSTRAINT fk_Chemiker_Mitarbeiter FOREIGN KEY (MitarbeiterNummer) REFERENCES
Kartell_Mitarbeiter(MitarbeiterNummer)
);
```

```
-- Tabelle: Schmuggler (Spezialisierung von Kartell-Mitarbeiter)
CREATE TABLE Schmuggler (
  SchmugglerNummer VARCHAR2(6) PRIMARY KEY,
  MitarbeiterNummer VARCHAR2(10) NOT NULL,
  Route VARCHAR2(255),
  Risikobereitschaft NUMBER(3,0) CHECK (Risikobereitschaft BETWEEN 0 AND 100),
  Verschwiegenheit VARCHAR2(20) CHECK (Verschwiegenheit IN ('Plaudertasche',
```

```
'bestechlich', 'naiver Prahler', 'verschwiegen', 'Informationssafe'))),
    CONSTRAINT fk_Schmuggler_Mitarbeiter FOREIGN KEY (MitarbeiterNummer) REFERENCES
Kartell_Mitarbeiter(MitarbeiterNummer)
);
```

```
-- Tabelle: Vertriebler (Spezialisierung von Kartell-Mitarbeiter)
CREATE TABLE Vertriebler (
    VertrieblerNummer VARCHAR2(6) PRIMARY KEY,
    MitarbeiterNummer VARCHAR2(10) NOT NULL,
    Charismalevel NUMBER(1,0) CHECK (Charismalevel BETWEEN 0 AND 9),
    Quote NUMBER(5,0),
    verpassteQuoten NUMBER(2,0) CHECK (verpassteQuoten BETWEEN 0 AND 99),
    CONSTRAINT fk_Vertriebler_Mitarbeiter FOREIGN KEY (MitarbeiterNummer) REFERENCES
Kartell_Mitarbeiter(MitarbeiterNummer)
);
```

```
-- Tabelle: Kartell-Kunde
CREATE TABLE Kartell_Kunde (
    Alias VARCHAR2(20) PRIMARY KEY,
    Telefonnummer VARCHAR2(15) NOT NULL,
    Erstkaufdatum DATE NOT NULL,
    Einkaufsverlauf VARCHAR2(255)
);
```

```
-- Tabelle: Lagerart
CREATE TABLE Lagerart (
    LagerTyp VARCHAR2(20) PRIMARY KEY,
    MaximaleKapazitaet NUMBER(9,0) CHECK (MaximaleKapazitaet > 0),
    InventarWechselIntervall NUMBER(3,0),
    BesuchsIntervall NUMBER(2,0)
);
```

```
-- Tabelle: Lager
CREATE TABLE Lager_Inventar (
    LagerNummer VARCHAR2(6) PRIMARY KEY,
    Startdatum DATE NOT NULL,
    Geheimhaltungsstatus VARCHAR2(20) CHECK (Geheimhaltungsstatus IN ('unbekannt',
'vermutet', 'entdeckt')) NOT NULL,
    Ort VARCHAR2(20) NOT NULL,
    GroeÙe NUMBER(4,0),
    Einlagerungsdatum DATE NOT NULL,
    LagerTyp VARCHAR2(20),
    CONSTRAINT fk_Inventar_Lager FOREIGN KEY (LagerTyp) REFERENCES Lagerart(LagerTyp)
);
```

```
-- Tabelle: Drogenart
CREATE TABLE Drogenart (
    Bezeichnung VARCHAR2(20) PRIMARY KEY,
    Aggregatsart VARCHAR2(20) CHECK (Aggregatsart IN ('Flüssigkeit', 'Pille',
'Pulver', 'Aerosol', 'Kristall', 'Feststoff')),
    Zutaten VARCHAR2(255),
```

```

    Kochzeit NUMBER(4,0),
    Abhaengigkeitsfaktor NUMBER(3,0) CHECK (Abhaengigkeitsfaktor BETWEEN 0 AND 100),
    KochGefahrenlevel NUMBER(2,0) CHECK (KochGefahrenlevel BETWEEN 0 AND 10)
);

```

```

-- Tabelle: Produkt
CREATE TABLE Produkt (
    ProduktNummer VARCHAR2(6) PRIMARY KEY,
    Preis NUMBER(8,2),
    Herstellungsdatum DATE NOT NULL,
    Streckungsgrad NUMBER(3,0),
    Gefahrenpotenzial NUMBER(3,0) CHECK (Gefahrenpotenzial BETWEEN 0 AND 100),
    Chemiker VARCHAR2(6),
    Bezeichner VARCHAR2(20),
    CONSTRAINT fk_Produkt_Bezeichnung FOREIGN KEY (Bezeichner) REFERENCES
Drogenart(Bezeichnung),
    CONSTRAINT fk_Produkt_ChemikerNummer FOREIGN KEY (Chemiker) REFERENCES
Chemiker(ChemikerNummer)
);

```

```

-- Tabelle: Labor
CREATE TABLE Labor (
    LaborNummer VARCHAR2(6) PRIMARY KEY,
    Standort VARCHAR2(50) NOT NULL,
    Ausstattung VARCHAR2(50),
    Kochplaetze NUMBER(2,0) CHECK (Kochplaetze > 0),
    Sicherheitslevel NUMBER(1,0) CHECK (Sicherheitslevel BETWEEN 0 AND 9)
);

```

```

-- Beziehungen einfügen

```

```

-- Tabelle: wird-hergestellt (Bezeichnung, LaborNummer)
CREATE TABLE Wird_Hergestellt (
    Bezeichnung VARCHAR2(20),
    LaborNummer VARCHAR2(6),
    CONSTRAINT fk_WirdHergestellt_Drogenart FOREIGN KEY (Bezeichnung) REFERENCES
Drogenart(Bezeichnung),
    CONSTRAINT fk_WirdHergestellt_Labor FOREIGN KEY (LaborNummer) REFERENCES
Labor(LaborNummer),
    PRIMARY KEY (Bezeichnung, LaborNummer)
);

```

```

-- Tabelle: kann-herstellen (Bezeichnung, ChemikerNummer)
CREATE TABLE Kann_Herstellen (
    Bezeichnung VARCHAR2(20),
    ChemikerNummer VARCHAR2(6),
    CONSTRAINT fk_KannHerstellen_Drogenart FOREIGN KEY (Bezeichnung) REFERENCES
Drogenart(Bezeichnung),
    CONSTRAINT fk_KannHerstellen_Chemiker FOREIGN KEY (ChemikerNummer) REFERENCES
Chemiker(ChemikerNummer),
    PRIMARY KEY (Bezeichnung, ChemikerNummer)
);

```

```
);
```

```
-- Tabelle: besitzt (ProduktNummer, LagerNummer)
```

```
CREATE TABLE Besitzt (  
    ProduktNummer VARCHAR2(6),  
    LagerNummer VARCHAR2(6),  
    CONSTRAINT fk_Besitzt_Produkt FOREIGN KEY (ProduktNummer) REFERENCES  
Produkt(ProduktNummer),  
    CONSTRAINT fk_Besitzt_Lager FOREIGN KEY (LagerNummer) REFERENCES  
Lager_Inventar(LagerNummer),  
    PRIMARY KEY (ProduktNummer, LagerNummer)  
);
```

```
-- Tabelle: transportiert (SchmugglerNummer, LagerNummer)
```

```
CREATE TABLE Transportiert (  
    SchmugglerNummer VARCHAR2(6),  
    LagerNummer VARCHAR2(6),  
    CONSTRAINT fk_Transportiert_Schmuggler FOREIGN KEY (SchmugglerNummer) REFERENCES  
Schmuggler(SchmugglerNummer),  
    CONSTRAINT fk_Transportiert_Lager FOREIGN KEY (LagerNummer) REFERENCES  
Lager_Inventar(LagerNummer),  
    PRIMARY KEY (SchmugglerNummer, LagerNummer)  
);
```

```
-- Tabelle: verkauft (VertrieblerNummer, LagerNummer)
```

```
CREATE TABLE Verkauft (  
    VertrieblerNummer VARCHAR2(6),  
    LagerNummer VARCHAR2(6),  
    CONSTRAINT fk_Verkauft_Vertriebler FOREIGN KEY (VertrieblerNummer) REFERENCES  
Vertriebler(VertrieblerNummer),  
    CONSTRAINT fk_Verkauft_Lager FOREIGN KEY (LagerNummer) REFERENCES  
Lager_Inventar(LagerNummer),  
    PRIMARY KEY (VertrieblerNummer, LagerNummer)  
);
```

```
-- Tabelle: wird-verkauft (Alias, Telefonnummer, ProduktNummer)
```

```
CREATE TABLE Wird_Verkauft (  
    Alias VARCHAR2(20),  
    Telefonnummer VARCHAR2(15),  
    ProduktNummer VARCHAR2(6),  
    CONSTRAINT fk_WirdVerkauft_Kunde FOREIGN KEY (Alias) REFERENCES  
Kartell_Kunde(Alias),  
    CONSTRAINT fk_WirdVerkauft_Produkt FOREIGN KEY (ProduktNummer) REFERENCES  
Produkt(ProduktNummer),  
    PRIMARY KEY (Alias, ProduktNummer)  
);
```

```
-- Tabelle: kennt (VertrieblerNummer, LagerTyp)
```

```
CREATE TABLE Kennt (  
    VertrieblerNummer VARCHAR2(6),  
    LagerTyp VARCHAR2(20),
```

```
CONSTRAINT fk_Kennt_Vertriebler FOREIGN KEY (VertrieblerNummer) REFERENCES  
Vertriebler(VertrieblerNummer),  
CONSTRAINT fk_Kennt_Lagerart FOREIGN KEY (LagerTyp) REFERENCES  
Lagerart(LagerTyp),  
PRIMARY KEY (VertrieblerNummer, LagerTyp)  
);  
  
-- end
```


Phase 3:

Aufgabe 1

Berichte und Diagramme für Datenbank-Auswertungen

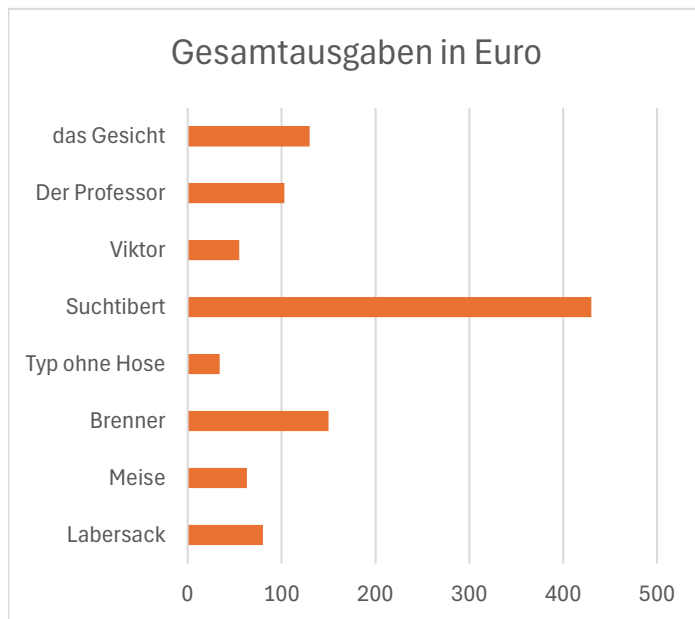
```
-- View 1: Gesamtkäufe Kunden
CREATE OR REPLACE VIEW Kunden_Produnkte_Ausgaben AS
SELECT
    k.Alias AS Kunde,
    k.Telefonnummer,
    wv.ProduktNummer,
    p.Bezeichner AS Produkt,
    p.Preis AS Einzelpreis,
    COUNT(wv.ProduktNummer) AS Gekaufte_Anzahl,
    SUM(p.Preis) AS Gesamtausgabe
FROM
    Kartell_Kunde k
JOIN
    Wird_Verkauft wv ON k.Alias = wv.Alias
JOIN
    Produkt p ON wv.ProduktNummer = p.ProduktNummer
GROUP BY
    k.Alias, k.Telefonnummer, wv.ProduktNummer, p.Bezeichner, p.Preis
ORDER BY
    k.Alias;
```

```
-- View 2: Gesamtausgaben
CREATE OR REPLACE VIEW Kunden_Gesamtausgaben AS
SELECT
    k.Alias AS Kunde,
    k.Telefonnummer,
    SUM(p.Preis) AS Gesamtausgabe
FROM
    Kartell_Kunde k
JOIN
    Wird_Verkauft wv ON k.Alias = wv.Alias
JOIN
    Produkt p ON wv.ProduktNummer = p.ProduktNummer
GROUP BY
    k.Alias, k.Telefonnummer;
```

```
-- View 3: Restkapazität der genutzten Lager
CREATE OR REPLACE VIEW Schmuggler_Lager_Kapazitaet AS
SELECT
    s.SchmugglerNummer,
    k.Name AS Schmuggler_Name,
    l.LagerNummer,
    l.Ort AS Lager_Ort,
    la.LagerTyp,
    la.MaximaleKapazitaet,
    COUNT(b.ProduktNummer) AS Belegte_Produnkte,
    la.MaximaleKapazitaet - COUNT(b.ProduktNummer) AS Verfuegbare_Kapazitaet
FROM
    Schmuggler s
JOIN
    Kartell_Mitarbeiter k ON s.MitarbeiterNummer = k.MitarbeiterNummer
```

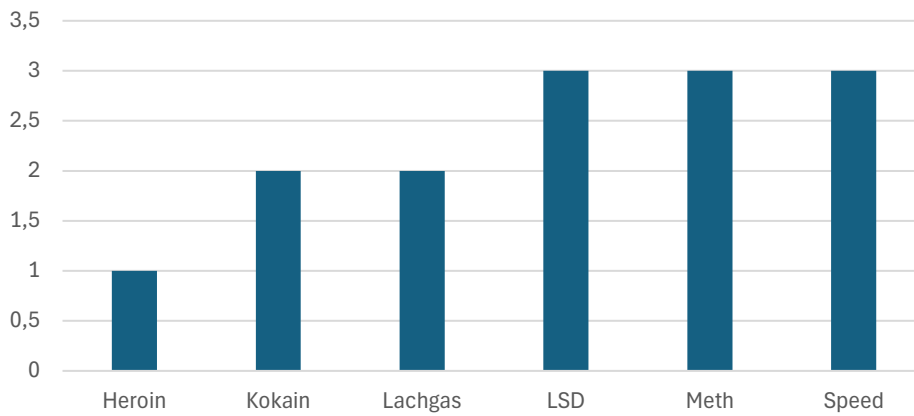
```
JOIN
  Transportiert t ON s.SchugglerNummer = t.SchugglerNummer
JOIN
  Lager_Inventar l ON t.LagerNummer = l.LagerNummer
JOIN
  Lagerart la ON l.LagerTyp = la.LagerTyp
LEFT JOIN
  Besitzt b ON l.LagerNummer = b.LagerNummer
GROUP BY
  s.SchugglerNummer, k.Name, l.LagerNummer, l.Ort, la.LagerTyp,
  la.MaximaleKapazitaet;
```

KUNDE	GESAMTAUSGABE
Labersack	80
Meise	63
Brenner	150
Typ ohne Hose	34
Suchtibert	430
Viktor	55
Der Professor	103
das Gesicht	130



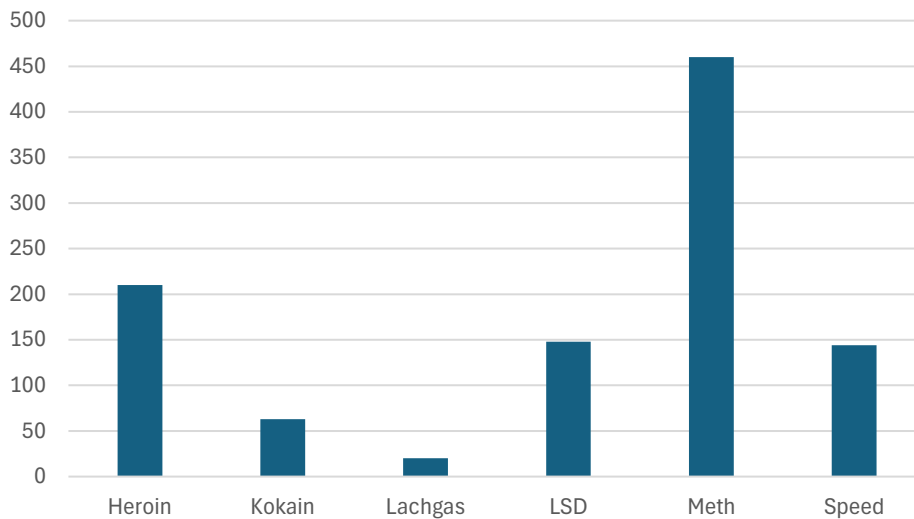
Summe von GEKAUFTE_ANZAHL

Summe von GEKAUFTE_ANZAHL bis PRODUKT

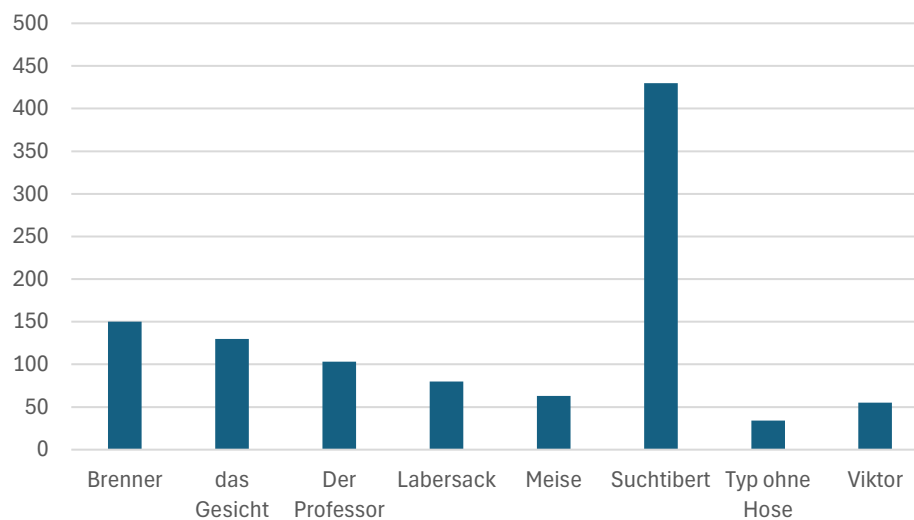


PRODUKT

Summe von EINZELPREIS bis PRODUKT



Summe von EINZELPREIS bis KUNDE



Phase 3:

Aufgabe 2

Programmierung eines Triggers

```

-- Trigger 1:
-- Prozedur für Produkt-ID
CREATE OR REPLACE PROCEDURE gen_product_id
(new_num OUT VARCHAR2)
AS
    max_num NUMBER;
BEGIN
    -- höchste Produkt-id ermitteln und erhöhen
    SELECT NVL(MAX(TO_NUMBER(SUBSTR(ProduktNummer, 2))), 0) + 1
    INTO max_num
    FROM Produkt;

    -- error wenn maximale Grenze erreicht wurde
    IF max_num > 999 THEN
        RAISE_APPLICATION_ERROR(-20202, 'Maximale Anzahl an Produkten erreicht. Bitte
das Lager aufräumen!');
    END IF;

    -- neue Produkt-id im Format 'PXXX' generieren
    new_num := 'P' || LPAD(max_num, 3, '0');
END;

-- Trigger Produkt-ID
CREATE OR REPLACE TRIGGER trg_product_id_gen
BEFORE INSERT ON Produkt
FOR EACH ROW
BEGIN
    -- Produkt-id wird sie automatisch hinzugefügt
    IF :NEW.ProduktNummer IS NULL THEN
        gen_product_id(:NEW.ProduktNummer);
    END IF;
END;

-- Trigger 2:
-- Kunden-ID
CREATE OR REPLACE TRIGGER trg_customer_no_dbl_alias
BEFORE INSERT ON Kartell_Kunde
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    -- Prüfen, ob der Alias (unabhängig von Groß-/Kleinschreibung) bereits existiert
    SELECT COUNT(*)
    INTO v_count
    FROM Kartell_Kunde
    WHERE UPPER(Alias) = UPPER(:NEW.Alias);

    -- Wenn der Alias existiert, eine benutzerdefinierte Fehlermeldung auslösen
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20201, 'Der Alias ist schon vergeben, du Erbse!');
    END IF;
END;

```

Phase 3:

Aufgabe 3

Testdaten


```
-- 1. Tabellen mit Eltern-Schlüsseln (Primärschlüssel) füllen
```

```
-- Kartell_Mitarbeiter (MitarbeiterNummer, Name, Status, Gehalt, Eintrittsdatum, Telefonnummer, Adresse)
```

```
INSERT INTO Kartell_Mitarbeiter (MitarbeiterNummer, Name, Status, Gehalt, Eintrittsdatum, Telefonnummer, Adresse)
```

```
VALUES ('K001', 'Max Wandrey', 'verfügbar', 5500.00, TO_DATE('2020-05-10', 'YYYY-MM-DD'), '1234567890', 'Brandvorwerkstr. 8');
```

```
INSERT INTO Kartell_Mitarbeiter (MitarbeiterNummer, Name, Status, Gehalt, Eintrittsdatum, Telefonnummer, Adresse)
```

```
VALUES ('K002', 'Ben Kaden', 'beschäftigt', 4800.00, TO_DATE('2020-03-15', 'YYYY-MM-DD'), '0987654321', 'Kurt-Eisner-Str. 35');
```

```
INSERT INTO Kartell_Mitarbeiter (MitarbeiterNummer, Name, Status, Gehalt, Eintrittsdatum, Telefonnummer, Adresse)
```

```
VALUES ('K003', 'Farin Förster', 'auf Regierungsliste', 4350.98, TO_DATE('2023-03-15', 'YYYY-MM-DD'), '09876545321', 'Hauptstraße 5');
```

```
INSERT INTO Kartell_Mitarbeiter (MitarbeiterNummer, Name, Status, Gehalt, Eintrittsdatum, Telefonnummer, Adresse)
```

```
VALUES ('K004', 'Chiara Geier', 'verfügbar', 3125.00, TO_DATE('2024-03-15', 'YYYY-MM-DD'), '0987612321', 'Breitenfelder Str. 9');
```

```
INSERT INTO Kartell_Mitarbeiter (MitarbeiterNummer, Name, Status, Gehalt, Eintrittsdatum, Telefonnummer, Adresse)
```

```
VALUES ('K005', 'Ben Beckschäfer', 'in Verwahrung', 0.00, TO_DATE('2022-03-15', 'YYYY-MM-DD'), '8817654321', 'Sonderhausener Weg 32');
```

```
INSERT INTO Kartell_Mitarbeiter (MitarbeiterNummer, Name, Status, Gehalt, Eintrittsdatum, Telefonnummer, Adresse)
```

```
VALUES ('K006', 'Robin Wiemer', 'verfügbar', 2408.00, TO_DATE('2024-03-15', 'YYYY-MM-DD'), '1087654323', 'Woauchimmer 44');
```

```
INSERT INTO Kartell_Mitarbeiter (MitarbeiterNummer, Name, Status, Gehalt, Eintrittsdatum, Telefonnummer, Adresse)
```

```
VALUES ('K007', 'Matthes Böttcher', 'verfügbar', 5200.45, TO_DATE('2024-03-15', 'YYYY-MM-DD'), '5930154321', 'Nobodycares 12');
```

```
-- Drogenart (Bezeichnung, Aggregatsart, Zutaten, Kochzeit, Abhaengigkeitsfaktor, KochGefahrenlevel)
```

```
INSERT INTO Drogenart (Bezeichnung, Aggregatsart, Zutaten, Kochzeit, Abhaengigkeitsfaktor, KochGefahrenlevel)
```

```
VALUES ('Kokain', 'Pulver', 'Kerosin, Kokapflanze', 180, 80, 1);
```

```
INSERT INTO Drogenart (Bezeichnung, Aggregatsart, Zutaten, Kochzeit, Abhaengigkeitsfaktor, KochGefahrenlevel)
```

```
VALUES ('Speed', 'Flüssigkeit', 'Koffein, Ephedrin, Paracetamol, Amphetamin', 30, 50, 5);
```

```
INSERT INTO Drogenart (Bezeichnung, Aggregatsart, Zutaten, Kochzeit, Abhaengigkeitsfaktor, KochGefahrenlevel)
```

```
VALUES ('LSD', 'Pille', 'Kohlenstoff, Wasserstoff, Stickstoff', 90, 20, 3);
```

```
INSERT INTO Drogenart (Bezeichnung, Aggregatsart, Zutaten, Kochzeit, Abhaengigkeitsfaktor, KochGefahrenlevel)
```

```
VALUES ('Lachgas', 'Aerosol', 'Stickstoff, Sauerstoff', 45, 5, 2);
```

```
INSERT INTO Drogenart (Bezeichnung, Aggregatsart, Zutaten, Kochzeit, Abhaengigkeitsfaktor, KochGefahrenlevel)
```

```
VALUES ('Meth', 'Kristall', 'Ephedrin, Ammoniak-Hydroxid', 240, 95, 10);
```

```

INSERT INTO Drogenart (Bezeichnung, Aggregatsart, Zutaten, Kochzeit,
Abhaengigkeitsfaktor, KochGefahrenlevel)
VALUES ('Heroin', 'Feststoff', 'Schlafmohn, Ammoniumchlorid', 120, 100, 6);
INSERT INTO Drogenart (Bezeichnung, Aggregatsart, Zutaten, Kochzeit,
Abhaengigkeitsfaktor, KochGefahrenlevel)
VALUES ('Ecstasy', 'Pille', '3,4-Methylenedioxy-N-methylamphetamin', 40, 25, 4);

```

```

-- Lagerart (LagerTyp, MaximaleKapazitaet, InventarWechselIntervall,
BesuchsIntervall)
INSERT INTO Lagerart (LagerTyp, MaximaleKapazitaet, InventarWechselIntervall,
BesuchsIntervall)
VALUES ('Container', 1000, 365, 30);
INSERT INTO Lagerart (LagerTyp, MaximaleKapazitaet, InventarWechselIntervall,
BesuchsIntervall)
VALUES ('Toter Briefkasten', 20, 1, 1);
INSERT INTO Lagerart (LagerTyp, MaximaleKapazitaet, InventarWechselIntervall,
BesuchsIntervall)
VALUES ('Lagerhalle', 50000, 90, 15);
INSERT INTO Lagerart (LagerTyp, MaximaleKapazitaet, InventarWechselIntervall,
BesuchsIntervall)
VALUES ('Kofferraum', 150, 5, 5);
INSERT INTO Lagerart (LagerTyp, MaximaleKapazitaet, InventarWechselIntervall,
BesuchsIntervall)
VALUES ('Dünndarm', 5, 3, 3);
INSERT INTO Lagerart (LagerTyp, MaximaleKapazitaet, InventarWechselIntervall,
BesuchsIntervall)
VALUES ('Werkstatt', 500, 30, 23);

```

```

-- 2. Weitere Tabellen füllen, die auf die primären Schlüssel der vorherigen
Tabellen verweisen

```

```

-- Chemiker (ChemikerNummer, MitarbeiterNummer, Bildungsgrad, Unfallrate,
notwendigeAssistenten)
INSERT INTO Chemiker (ChemikerNummer, MitarbeiterNummer, Bildungsgrad, Unfallrate,
notwendigeAssistenten)
VALUES ('C001', 'K003', 'Doktor', 9, 2);
INSERT INTO Chemiker (ChemikerNummer, MitarbeiterNummer, Bildungsgrad, Unfallrate,
notwendigeAssistenten)
VALUES ('C002', 'K005', 'Diplom', 45, 4);

```

```

-- Schmuggler (SchmugglerNummer, MitarbeiterNummer, Route, Risikobereitschaft,
Verschwiegenheit)
INSERT INTO Schmuggler (SchmugglerNummer, MitarbeiterNummer, Route,
Risikobereitschaft, Verschwiegenheit)
VALUES ('S001', 'K004', 'Route 66', 75, 'verschwiegen');
INSERT INTO Schmuggler (SchmugglerNummer, MitarbeiterNummer, Route,
Risikobereitschaft, Verschwiegenheit)
VALUES ('S002', 'K006', 'Route 42', 50, 'bestechlich');

```

```

-- Vertriebler (VertrieblerNummer, MitarbeiterNummer, Charismalevel, Quote,
verpassteQuoten)

```

```

INSERT INTO Vertriebler (VertrieblerNummer, MitarbeiterNummer, Charismalevel,
Quote, verpassteQuoten)
VALUES ('V001', 'K002', 9, 210, 0);
INSERT INTO Vertriebler (VertrieblerNummer, MitarbeiterNummer, Charismalevel,
Quote, verpassteQuoten)
VALUES ('V002', 'K007', 6, 120, 7);

```

```

-- Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
INSERT INTO Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
VALUES ('Viktor', '01123456789', TO_DATE('2021-06-01', 'YYYY-MM-DD'), 'Lachgas,
LSD');
INSERT INTO Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
VALUES ('Brenner', '11234567890', TO_DATE('2022-06-25', 'YYYY-MM-DD'), 'Meth,
Heroin, Speed');
INSERT INTO Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
VALUES ('Meise', '56234567430', TO_DATE('2023-07-11', 'YYYY-MM-DD'), 'Kokain');
INSERT INTO Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
VALUES ('Typ ohne Hose', '10534567777', TO_DATE('2024-02-18', 'YYYY-MM-DD'),
'Speed, Lachgas');
INSERT INTO Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
VALUES ('Der Professor', '86234567567', TO_DATE('2023-09-11', 'YYYY-MM-DD'),
'LSD');
INSERT INTO Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
VALUES ('Labersack', '33034567840', TO_DATE('2021-03-22', 'YYYY-MM-DD'), 'Kokain,
Speed');
INSERT INTO Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
VALUES ('das Gesicht', '59034567897', TO_DATE('2024-12-09', 'YYYY-MM-DD'), 'Meth,
Kokain, Speed');
INSERT INTO Kartell_Kunde (Alias, Telefonnummer, Erstkaufdatum, Einkaufsverlauf)
VALUES ('Suchtibert', '44134567223', TO_DATE('2020-5-01', 'YYYY-MM-DD'), 'Lachgas,
Speed, Heroin, LSD, Kokain, Meth');

```

```

-- Lager_Inventar (LagerNummer, Startdatum, Geheimhaltungsstatus, Ort, GroeÙe,
Einlagerungsdatum, LagerTyp)
INSERT INTO Lager_Inventar (LagerNummer, Startdatum, Geheimhaltungsstatus, Ort,
GroeÙe, Einlagerungsdatum, LagerTyp)
VALUES ('LG001', TO_DATE('2021-01-01', 'YYYY-MM-DD'), 'unbekannt', 'Leipzig', 1000,
TO_DATE('2021-01-15', 'YYYY-MM-DD'), 'Container');
INSERT INTO Lager_Inventar (LagerNummer, Startdatum, Geheimhaltungsstatus, Ort,
GroeÙe, Einlagerungsdatum, LagerTyp)
VALUES ('LG002', TO_DATE('2022-02-20', 'YYYY-MM-DD'), 'vermutet', 'Breitenfeld',
400, TO_DATE('2022-02-25', 'YYYY-MM-DD'), 'Werkstatt');
INSERT INTO Lager_Inventar (LagerNummer, Startdatum, Geheimhaltungsstatus, Ort,
GroeÙe, Einlagerungsdatum, LagerTyp)
VALUES ('LG003', TO_DATE('2022-03-15', 'YYYY-MM-DD'), 'entdeckt', 'Hauptbahnhof
Dresden', 10, TO_DATE('2023-12-02', 'YYYY-MM-DD'), 'Toter Briefkasten');
INSERT INTO Lager_Inventar (LagerNummer, Startdatum, Geheimhaltungsstatus, Ort,
GroeÙe, Einlagerungsdatum, LagerTyp)
VALUES ('LG004', TO_DATE('2024-08-11', 'YYYY-MM-DD'), 'unbekannt', 'Gohlis', 5,
TO_DATE('2024-08-12', 'YYYY-MM-DD'), 'Dünndarm');

```

```
-- 3. Tabellen füllen, die auf bereits erstellte Schlüssel referenzieren
```

```
-- Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, C-Nr., Bezeichnung)
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P001', 180.00, TO_DATE('2024-05-01', 'YYYY-MM-DD'), 1, 2, 'C001', 'Meth');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P002', 40.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 4, 40, 'C002',
'Kokain');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P003', 10.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 0, 0, 'C002',
'Lachgas');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P004', 10.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 0, 0, 'C002',
'Lachgas');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P005', 149.99, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 3, 16, 'C002',
'Meth');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P006', 80.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 5, 50, 'C001',
'Speed');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P007', 64.49, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 2, 15, 'C001', 'LSD');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P008', 210.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 1, 1, 'C001',
'Heroin');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P009', 30.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 5, 23, 'C002',
'Speed');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P010', 34.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 2, 20, 'C001',
'Speed');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P011', 130.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 7, 22, 'C002',
'Meth');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P012', 38.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 4, 20, 'C002', 'LSD');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
```

```
VALUES ('P013', 22.95, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 8, 80, 'C001',
'Kokain');
INSERT INTO Produkt (ProduktNummer, Preis, Herstellungsdatum, Streckungsgrad,
Gefahrenpotenzial, Chemiker, Bezeichner)
VALUES ('P014', 45.00, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 5, 45, 'C001', 'LSD');
```

```
-- Labor (LaborNummer, Standort, Ausstattung, Kochplaetze, Sicherheitslevel)
INSERT INTO Labor (LaborNummer, Standort, Ausstattung, Kochplaetze,
Sicherheitslevel)
VALUES ('LB001', 'Grünau', 'Ausrüstung A', 5, 8);
INSERT INTO Labor (LaborNummer, Standort, Ausstattung, Kochplaetze,
Sicherheitslevel)
VALUES ('LB002', 'Paunsdorf', 'Ausrüstung B', 3, 4);
```

```
-- Wird_Hergestellt (Bezeichnung, LaborNummer)
INSERT INTO Wird_Hergestellt (Bezeichnung, LaborNummer)
VALUES ('Meth', 'LB001');
INSERT INTO Wird_Hergestellt (Bezeichnung, LaborNummer)
VALUES ('Heroin', 'LB001');
INSERT INTO Wird_Hergestellt (Bezeichnung, LaborNummer)
VALUES ('Speed', 'LB001');
INSERT INTO Wird_Hergestellt (Bezeichnung, LaborNummer)
VALUES ('Kokain', 'LB002');
INSERT INTO Wird_Hergestellt (Bezeichnung, LaborNummer)
VALUES ('Lachgas', 'LB002');
INSERT INTO Wird_Hergestellt (Bezeichnung, LaborNummer)
VALUES ('LSD', 'LB002');
```

```
-- Kann_Herstellen (Bezeichnung, ChemikerNummer)
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('Meth', 'C001');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('Heroin', 'C001');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('Speed', 'C001');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('Kokain', 'C001');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('LSD', 'C001');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('Kokain', 'C002');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('LSD', 'C002');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('Lachgas', 'C002');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('Meth', 'C002');
INSERT INTO Kann_Herstellen (Bezeichnung, ChemikerNummer)
VALUES ('Speed', 'C002');
```

```
-- Besitzt (ProduktNummer, LagerNummer)
```

```

INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P001', 'LG001');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P002', 'LG002');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P003', 'LG002');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P004', 'LG001');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P005', 'LG002');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P006', 'LG001');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P007', 'LG001');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P008', 'LG001');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P009', 'LG002');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P010', 'LG002');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P011', 'LG001');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P012', 'LG002');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P013', 'LG001');
INSERT INTO Besitzt (ProduktNummer, LagerNummer)
VALUES ('P014', 'LG001');

-- Transportiert (SchmugglerNummer, LagerNummer)
INSERT INTO Transportiert (SchmugglerNummer, LagerNummer)
VALUES ('S001', 'LG001');
INSERT INTO Transportiert (SchmugglerNummer, LagerNummer)
VALUES ('S002', 'LG002');

-- Verkauft (VertrieblerNummer, LagerNummer)
INSERT INTO Verkauft (VertrieblerNummer, LagerNummer)
VALUES ('V001', 'LG001');
INSERT INTO Verkauft (VertrieblerNummer, LagerNummer)
VALUES ('V002', 'LG002');

-- Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Suchtibert', '44134567223', 'P001');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Meise', '56234567430', 'P002');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Suchtibert', '44134567223', 'P003');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Viktor', '01123456789', 'P004');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)

```

```

VALUES ('Brenner', '11234567890', 'P005');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Labersack', '33034567840', 'P006');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Der Professor', '86234567567', 'P007');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Suchtibert', '44134567223', 'P008');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Suchtibert', '44134567223', 'P009');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Typ ohne Hose', '10534567777', 'P010');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('das Gesicht', '59034567897', 'P011');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Der Professor', '86234567567', 'P012');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Meise', '56234567430', 'P013');
INSERT INTO Wird_Verkauft (Alias, Telefonnummer, ProduktNummer)
VALUES ('Viktor', '01123456789', 'P014');

```

```

-- Kennt (VertrieblerNummer, LagerTyp)
INSERT INTO Kennt (VertrieblerNummer, LagerTyp)
VALUES ('V001', 'Container');
INSERT INTO Kennt (VertrieblerNummer, LagerTyp)
VALUES ('V001', 'Werkstatt');
INSERT INTO Kennt (VertrieblerNummer, LagerTyp)
VALUES ('V001', 'Kofferraum');
INSERT INTO Kennt (VertrieblerNummer, LagerTyp)
VALUES ('V001', 'Lagerhalle');
INSERT INTO Kennt (VertrieblerNummer, LagerTyp)
VALUES ('V002', 'Toter Briefkasten');
INSERT INTO Kennt (VertrieblerNummer, LagerTyp)
VALUES ('V002', 'Dünndarm');

```

```

-- Änderungen eintragen
COMMIT;

```