

```
-- 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
```