

# **Datenbankenprojekt**

## Datenbank für einen Elektronikhändler Phase 2

Modul: Datenbanken  
Semester: 3

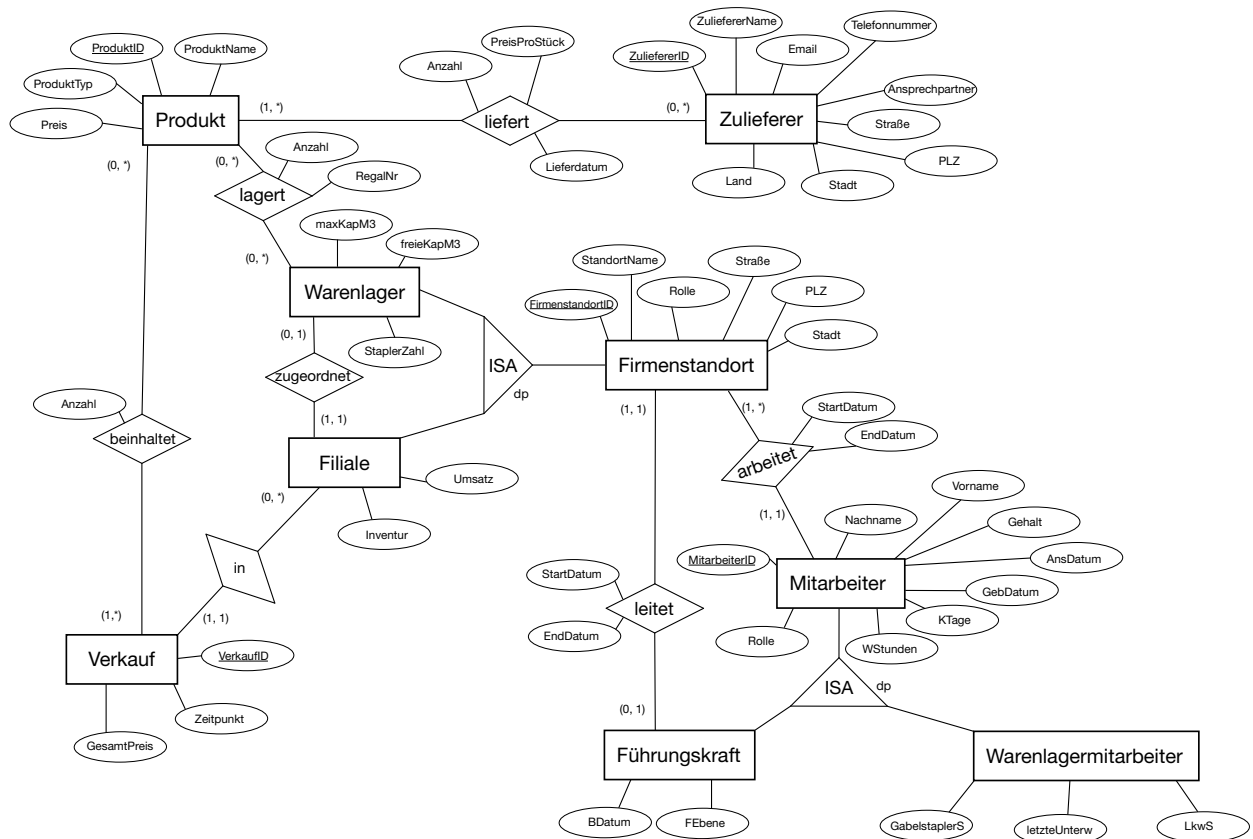
Toni Nguyen  
Matrikelnr.: 84759

Lucas Stramke  
Matrikelnr.: 84381

## Phase 2: Datenbank-Entwurf

1. Entity-Relationship-Diagramm: Das ER-Diagramm kann von Hand oder mit Hilfe eines Werkzeugs wie z.B. Visio oder EDraw erstellt werden. Beim Gebrauch von Tools müssen womöglich Einschränkungen beachtet werden (keine dreistelligen Beziehungen, m:n-Beziehungen ohne Attribute).
2. Relationales Schema (mit Schlüssel- und Spaltendefinitionen)
3. SQL-Skript zur Datenbankdefinition mit allen Integritätsbedingungen (Schlüssel, referentielle Integrität, NOT NULL, evtl. Werteinschränkungen durch CHECK-Constraints)

# Entity-Relationship-Diagramm



# Relationales Schema

unterschrieben = Primärschlüssel, Kursiv = Fremdschlüssel

---

## Schritt 1: Grundentitäten

Produkt (ProduktID, ProduktName, Preis, ProduktTyp)

Mitarbeiter (MitarbeiterID, Nachname, Vorname, Gehalt, GebDatum, AnsDatum, Rolle, WStunden, KTage)

Firmenstandort (FirmenstandortID, StandortName, Straße, Postleitzahl, Stadt, Rolle)

Zulieferer (ZuliefererID, ZuliefererName, Email, Telefonnummer, Ansprechpartner, Straße, Stadt, Land, PLZ)

Verkauf (VerkaufID, Zeitpunkt, GesamtPreis)

---

## Schritt 2: ISA-Beziehungen

### Warenlagermitarbeiter ISA Mitarbeiter und Führungskraft ISA Mitarbeiter

Warenlagermitarbeiter (MitarbeiterID, GabelstaplerS, letzteUnterw, LkWS)

Fuehrungskraft (MitarbeiterID, BDatum, FEbene)

### Warenlager ISA Firmenstandort und Filiale ISA Firmenstandort

Warenlager (FirmenstandortID, maxKapM3, freieKapM3, StaplerZahl)

Filiale (FirmenstandortID, Umsatz, Inventur)

---

## Schritt 3: Beziehungen zwischen Entitäten

Arbeitet → Mitarbeiter (MitarbeiterID, Nachname, Vorname, Gehalt, GebDatum, AnsDatum, Rolle, WStunden, KTage, Arbeitsort<sup>1</sup>)

Leitet → Firmenstandort (FirmenstandortID, StandortName, Straße, Postleitzahl, Stadt, Rolle, Leiter<sup>2</sup>)

Zugeordnet → Filiale(FirmenstandortID, Umsatz, Inventur, angWarenlager<sup>3</sup>)

In → Verkauf(VerkaufID, Zeitpunkt, GesamtPreis, Verkaufsart<sup>4</sup>)

Liefert (ProduktID, ZuliefererID, Lieferdatum, Anzahl, PreisProStück)

Beinhaltet (VerkaufID, ProduktID, Anzahl)

Lagert (ProduktID, Lagerort (*siehe Fußnote 3*), Anzahl, RegalNr)

---

<sup>1</sup> entspricht einer Firmenstandort.FirmenstandortID

<sup>2</sup> entspricht einer Fuehrungskraft.MitarbeiterID

<sup>3</sup> entspricht einer Warenlager.FirmenstandortID

<sup>4</sup> entspricht einer Filiale.FirmenstandortID



---

## Ergebnis Relationen

Produkt (ProduktID, **ProduktName**, **Preis**, ProduktTyp)

Zulieferer ( ZuliefererID, **ZuliefererName**, **Email**, **Telefonnummer**, Ansprechpartner, **Straße**, **Stadt**, **Land**, **PLZ**)

Mitarbeiter (MitarbeiterID, **Nachname**, **Vorname**, Gehalt, **GebDatum**, **AnsDatum**, Rolle, WStunden, K Tage, **Arbeitsort**)

Warenlagermitarbeiter (MitarbeiterID, GabelstaplerS, letzteUnterw, LkwS)

Fuehrungskraft (MitarbeiterID, BDatum, FEbene)

Firmenstandort (FirmenstandortID, **StandortName**, **Straße**, **Postleitzahl**, **Stadt**, Rolle, **Leiter**)

Warenlager (FirmenstandortID, maxKapM3, freieKapM3, StaplerZahl)

Filiale(FirmenstandortID, Umsatz, Inventur, **angWarenlager**)

Verkauf(VerkaufID, **Zeitpunkt**, **Gesamtpreis**, **Verkaufsart**)

Liefert (ProduktID, ZuliefererID, Lieferdatum, **Anzahl**, **PreisProStück**)

Beinhaltet (VerkaufID, ProduktID, **Anzahl**)

Lagert (ProduktID, Lagerort, **Anzahl**, **RegalNr**)

**Gelb** = UNIQUE

**Blau** = NOT NULL

**Grün** = UNIQUE und NOT NULL



## SQL Script

```
create table Produkt(  
    ProduktID varchar2(8) primary key,  
    ProduktName varchar2(100) not null,  
    Preis decimal(7, 2) not null check (Preis >= 0),  
    ProduktTyp varchar2(25) check (ProduktTyp in  
        ('PC', 'Laptop', 'Handy', 'Tablet', 'Konsole', 'Drucker',  
'Bildschirm', 'Musiksystem', 'PC/Laptop-Zubehör', 'Handy/Tablet  
Zubehör', 'Druckerzubehör'))  
)  
  
create table Zuliefer(  
    ZuliefererID varchar2(8) primary key,  
    ZulieferName varchar2(50) not null unique,  
    Email varchar2(50) not null unique,  
    Telefonnummer varchar2(50) not null unique,  
    Ansprechpartner varchar2(50),  
    Straße varchar2(150) not null,  
    PLZ varchar2(8) not null,  
    Stadt varchar2(30) not null,  
    Land varchar2(10) not null  
)  
  
create table Mitarbeiter(  
    MitarbeiterID varchar2(8) primary key,  
    Nachname varchar(50) not null,  
    Vorname varchar2(50) not null,  
    Gehalt decimal(8,2),  
    GebDatum date not null,  
    AnsDatum date not null check(AnsDatum >= '01.01.1990'),  
    Rolle varchar2(100) check(Rolle in  
        ('Führungskraft', 'Warenlagermitarbeiter', 'Filialmitarbeiter',  
'Verwaltungsmitarbeiter', 'Praktikant', 'Auszubildender'))  
),  
    WStunden decimal(4,2),  
    Ktage integer check(Ktage >= 0),  
    Arbeitsort varchar2(8) not null -- foreign key Definition am Ende  
)  
  
create table Warenlagermitarbeiter(  
    MitarbeiterID varchar2(8) primary key,  
    GabelstaplerS varchar2(1) check( GabelstaplerS in ('T', 'F')),  
    letzteUnterw date,  
    LkwS varchar2(3) check( LkwS in ('C1', 'C1E', 'C', 'CE')),  
    foreign key(MitarbeiterID) references Mitarbeiter(MitarbeiterID)  
)  
  
create table Fuehrungskraft(  
    MitarbeiterID varchar2(8) primary key,  
    BDatum date not null check(BDatum >= '01.01.1990'),  
    FEbene varchar2(40) check(FEbene in  
        ('Leiter Filiale', 'Leiter Warenlager', 'Verwalter Warenlager',  
'Schichtleiter', 'Leiter Marketing', 'Leiter Einkauf', 'Leiter
```

```

Personal', 'Geschäftsführer', 'Technischer Geschäftsführer',
'Kaufmännischer Geschäftsführer')
),
foreign key(MitarbeiterID) references Mitarbeiter(MitarbeiterID)
)

create table Firmenstandort(
    FirmenstandortID varchar2(8) primary key,
    StandortName varchar2(15) not null unique,
    Straße varchar2(150) not null,
    PLZ varchar2(8) not null,
    Stadt varchar2(30) not null,
    Rolle varchar(20) check(Rolle in
        ('Filiale', 'Warenlager', 'Hauptsitz', 'Filiallager')
    ),
    Leiter varchar2(8) not null,
    foreign key (Leiter) references Fuehrungskraft(MitarbeiterID)
)

create table Warenlager (
    FirmenstandortID varchar2(8) primary key,
    maxKapM3 decimal(10,2) check (maxKapM3 >= 0) not null,
    freieKapM3 decimal(10,2) check (freieKapM3 >= 0) not null,
    StaplerZahl integer check (StaplerZahl >= 0),
    foreign key(FirmenstandortID) references
Firmenstandort(FirmenstandortID)
)

create Table Filiale(
    FirmenstandortID varchar2(8) primary key,
    Umsatz decimal(10,2) check(Umsatz >= 0),
    Inventur decimal(5,2) check (Inventur < 100),
    angWarenlager varchar2(8) not null,
    foreign key(FirmenstandortID) references
Firmenstandort(FirmenstandortID),
    foreign key(angWarenlager) references Warenlager(FirmenstandortID)
)

create table Verkauf(
    VerkaufID varchar2(8) primary key,
    Zeitpunkt date not null,
    Gesamtpreis decimal(12,2) not null check(Gesamtpreis >= 0),
    Verkaufsort varchar2(8) not null,
    foreign key Verkaufsort references Filiale(FirmenstandortID)
)

create table Liefert(
    ProduktID varchar2(8),
    ZuliefererID varchar2(8),
    Anzahl integer not null check(Anzahl > 0),
    PreisProStueck decimal(8, 2) not null check(PreisProStueck >= 0),
    primary key (ProduktID, ZuliefererID),
    foreign key ProduktID references Produkt,
    foreign key ZuliefererID references Zulieferer
)

```

```

create table Beinhaltet(
    VerkaufsID varchar2(8),
    ProduktID varchar2(8),
    Anzahl integer not null check(Anzahl > 0),
    primary key (VerkaufsID, ProduktID),
    foreign key VerkaufsID references Verkauf,
    foreign key ProduktID references Produkt
)

create table Lagert(
    ProduktID varchar2(8),
    Lagerort varchar2(8),
    Anzahl integer not null check(Anzahl >= 0),
    RegalNr varchar2(8) not null,
    primary key (ProduktID, Lagerort),
    foreign key (ProduktID) references Produkt,
    foreign key (Lagerort) references Warenlager(FirmenstandortID)
)

--nachträgliche Fremdschlüsseldeklaration die oben angekündigt wurde

alter table Mitarbeiter
add constraint fk_nachtraeglich_Mitarbeiter-Arbeitsort foreign key
(Arbeitsort) references Firmenstandort(FirmenstandortID)

```

Nochmal als .sql beigelegt

