

Meilenstein 3: Logischer und Physischer Entwurf

Gruppe 30
Boiko Bohdana, Alexander Grath

Logischer und Physischer Entwurf

1 Normalisierung (3. Normalform)

2 Datenbank Triggers

3 Datengenerierung (Data-Generator mit Jupyter Notebook)

4 ChatGPT

Verwendeted Prompt, aus der Angabe:

Generiere mir ein ER-Modell, ein relationales Schema und die zugehörigen SQL Statements auf Basis der folgenden Informationen: + verbatim die Abgabe aus Aufgabe 1-1.

Modell: GPT-5.2 (free)

Link: <https://chatgpt.com/share/696ba0f8-d094-8013-a1ba-5c41bd8ecab5>

Antwort:

Unten findest du ein konsistentes, prüfungs- und übungstaugliches Ergebnis bestehend aus

ER-Modell (textuell beschrieben)

Relationalem Schema

SQL-DDL-Statements (CREATE TABLE ...)

Ich halte mich eng an deine Vorgaben (Wien Praterstern, kein Zahlungssystem, Fokus auf Betrieb/Vertrieb)

1. ER-Modell (konzeptionell)

Entitäten und Spezialisierungen

Fahrzeug

FahrzeugID (PK)

Baujahr

Kapazität

Generalisierung

Subtypen (disjunkt, total):

Bus

Kennzeichen

Niederflur (Boolean)

Zug

Baureihe

Wagonanzahl

Straßenbahn

Spurweite

Stromsystem

Wartung

WartungsNr (PK)

Datum

Kosten

Wartungsart

Beziehung: Fahrzeug 1 | n Wartung

Personal

PersonalNr (PK)

Name

Gehalt

Einstellungsdatum

Generalisierung

Subtypen:

Fahrer:in

Führerscheinklasse

Kontrolleur:in

Zuständigkeitsbereich

Rekursive Beziehung:

Fahrer:in betreut Fahrer:in
(1:n, Mentor → Trainee)

Linie

LinienNr (PK)

Taktung (Minuten)

Betriebsbeginn

Betriebsende

Haltestelle

HaltestellenID (PK)

Name

Barrierefrei (Boolean)

m:n Beziehung

Linie | bedient | Haltestelle

(Reihenfolge relevant → eigene Beziehungstabelle)

Einsatz

EinsatzID (PK)

Datum

verbindet:

Fahrer:in

Fahrzeug

Linie

(„Fahrer fährt mit Fahrzeug auf Linie“)

Kunde

KundenID (PK)

Name

E-Mail

Adresse

Ticket

TicketID (PK)

Kaufdatum

Preis

GültigVon

GültigBis

Beziehungen:

Kunde 1 | n Ticket

Ticket m | n Linie

Ticket n | 1 Kontrolleur (wird geprüft von)

2. Relationales Schema

FAHRZEUG(FahrzeugID, Baujahr, Kapazität)

BUS(FahrzeugID, Kennzeichen, Niederflur)

ZUG(FahrzeugID, Baureihe, Wagonanzahl)

STRASSENBAHN(FahrzeugID, Spurweite, Stromsystem)

WARTUNG(WartungsNr, FahrzeugID, Datum, Kosten, Wartungsart)

PERSONAL(PersonalNr, Name, Gehalt, Einstellungsdatum)

FAHRER(PersonalNr, Fuehrerscheinklasse)

KONTROLLEUR(PersonalNr, Zuständigkeitsbereich)

MENTORING(MentorNr, TraineeNr)

LINIE(LinienNr, Taktung, Betriebsbeginn, Betriebsende)

HALTESTELLE(HaltestellenID, Name, Barrierefrei)

LINIE_HALTESTELLE(LinienNr, HaltestellenID, Reihenfolge)

EINSATZ(EinsatzID, Datum, PersonalNr, FahrzeugID, LinienNr)

KUNDE(KundenID, Name, Email, Adresse)

TICKET(TicketID, Kaufdatum, Preis, GueltigVon, GueltigBis, KundenID, KontrolleurNr)

TICKET_LINIE(TicketID, LinienNr)

3. SQL-Statements (DDL)

```
CREATE TABLE Fahrzeug (
    FahrzeugID INT PRIMARY KEY,
    Baujahr INT,
    Kapazitaet INT
);
```

```
CREATE TABLE Bus (
    FahrzeugID INT PRIMARY KEY,
    Kennzeichen VARCHAR(20),
    Niederflur BOOLEAN,
    FOREIGN KEY (FahrzeugID) REFERENCES Fahrzeug(FahrzeugID)
);
```

```
CREATE TABLE Zug (
    FahrzeugID INT PRIMARY KEY,
    Baureihe VARCHAR(50),
    Wagonanzahl INT,
    FOREIGN KEY (FahrzeugID) REFERENCES Fahrzeug(FahrzeugID)
);
```

```
CREATE TABLE Strassenbahn (
    FahrzeugID INT PRIMARY KEY,
    Spurweite VARCHAR(30),
    Stromsystem VARCHAR(50),
    FOREIGN KEY (FahrzeugID) REFERENCES Fahrzeug(FahrzeugID)
);
```

```
CREATE TABLE Wartung (
    WartungsNr INT PRIMARY KEY,
    FahrzeugID INT,
    Datum DATE,
    Kosten DECIMAL(10,2),
    Wartungsart VARCHAR(100),
    FOREIGN KEY (FahrzeugID) REFERENCES Fahrzeug(FahrzeugID)
);
```

```
CREATE TABLE Personal (
    PersonalNr INT PRIMARY KEY,
    Name VARCHAR(100),
    Gehalt DECIMAL(10,2),
    Einstellungsdatum DATE
);
```

```
CREATE TABLE Fahrer (
    PersonalNr INT PRIMARY KEY,
    Fuehrerscheinklasse VARCHAR(10),
    FOREIGN KEY (PersonalNr) REFERENCES Personal(PersonalNr)
);
```

```
CREATE TABLE Kontrolleur (
    PersonalNr INT PRIMARY KEY,
    Zustaendigkeitsbereich VARCHAR(100),
```

```
FOREIGN KEY (PersonalNr) REFERENCES Personal(PersonalNr)
);

CREATE TABLE Mentoring (
    MentorNr INT,
    TraineeNr INT,
    PRIMARY KEY (MentorNr, TraineeNr),
    FOREIGN KEY (MentorNr) REFERENCES Fahrer(PersonalNr),
    FOREIGN KEY (TraineeNr) REFERENCES Fahrer(PersonalNr)
);

CREATE TABLE Linie (
    LinienNr INT PRIMARY KEY,
    Taktung INT,
    Betriebsbeginn TIME,
    Betriebsende TIME
);

CREATE TABLE Haltestelle (
    HaltestellenID INT PRIMARY KEY,
    Name VARCHAR(100),
    Barrierefrei BOOLEAN
);

CREATE TABLE Linie_Haltestelle (
    LinienNr INT,
    HaltestellenID INT,
    Reihenfolge INT,
    PRIMARY KEY (LinienNr, HaltestellenID),
    FOREIGN KEY (LinienNr) REFERENCES Linie(LinienNr),
    FOREIGN KEY (HaltestellenID) REFERENCES Haltestelle(HaltestellenID)
);

CREATE TABLE Einsatz (
    EinsatzID INT PRIMARY KEY,
    Datum DATE,
    PersonalNr INT,
    FahrzeugID INT,
    LinienNr INT,
    FOREIGN KEY (PersonalNr) REFERENCES Fahrer(PersonalNr),
    FOREIGN KEY (FahrzeugID) REFERENCES Fahrzeug(FahrzeugID),
    FOREIGN KEY (LinienNr) REFERENCES Linie(LinienNr)
);

CREATE TABLE Kunde (
    KundenID INT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Adresse VARCHAR(200)
);

CREATE TABLE Ticket (
    TicketID INT PRIMARY KEY,
    Kaufdatum DATE,
    Preis DECIMAL(10,2),
    GueltigVon DATE,
    GueltigBis DATE,
    KundenID INT,
    KontrolleurNr INT,
    FOREIGN KEY (KundenID) REFERENCES Kunde(KundenID),
    FOREIGN KEY (KontrolleurNr) REFERENCES Kontrolleur(PersonalNr)
```

);

```
CREATE TABLE Ticket_Linie (
    TicketID INT,
    LinienNr INT,
    PRIMARY KEY (TicketID, LinienNr),
    FOREIGN KEY (TicketID) REFERENCES Ticket(TicketID),
    FOREIGN KEY (LinienNr) REFERENCES Linie(LinienNr)
);
```

Wenn du möchtest, kann ich dir das ER-Modell auch als Diagramm (Chen oder Crow's Foot) beschreiben oder vereinfachen für eine Klausur.

Resultat Bewertung

Ein Problem ergibt sich da wir PDFs verwenden, wenn Umlaute kopiert werden die Umlautpunkte separat kopiert. Aus ö wir "o".

Wird die Antwort kopiert ergibt sich ein ähnliches Problem. ChatGPT verwendet viele Symbole (vor allem Emojis und em-dash) die in anderen Systemen nicht ordentlich dargestellt werden können.

Da unsere Angabe sehr exakt ist und auch vieles mit Textmarkierungen herausgehoben ist, hatte die KI keine Probleme den Text richtig zu interpretieren. Es würde definitiv einen guten Startpunkt bieten.

Probleme:

- die tertiäre Beziehung aus Fahrzeug, Linie und Fahrer wurde nicht erkannt bzw. es ist fraglich, wie es interpretiert worden ist da es laut KI auch einen Primärschlüssel haben sollte
- mit nur einem Prompt bekommt man kein ordentliches ER-Diagramm, ich weiß nicht wie es mit der bezahlten Version funktioniert, aber die Prompts sind auf eine gewisse Zeichenanzahl limitiert. Desto länger die "Überhaltung" desto höher ist die Chance dass die KI anfängt zu halluzinieren.
- *Wartung* wurde nicht als schwache Identität erkannt
- die Fehler aus dem ersten Schritt setzen sich in das relationale Schema fort
- beim relationalen Schema wird nicht angezeigt was ein primary / foreign key ist
- Vererbungen werden nicht ordentlich dargestellt
- die Fehler aus Schritt 1 und 2 setzen sich in die SQL-Statements fort
- es werden keine constraints verwendet um das DBS sicherer zu machen
- es wird *BOOLEAN* verwendet, es ist als Alias zu *TINYINT* in Ordnung; man wird aber nicht gewarnt das es zu Kompatibilitäts-Problemen führen kann.
- es gibt keine default Werte
- es gibt keine DELETE Regeln
- es gibt kein NOT NULL
- *CONSTRAINT* wurde nicht verwendet um error-Meldungen lesbarer zu machen