

Fallstudie Pflegedienst

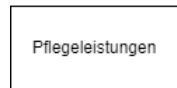
Vor der Erweiterung



MNr., m_Name,
m_Vorname,
m_Geb.Datum,
m_Anschrift, m_PLZ,
m_Stadt

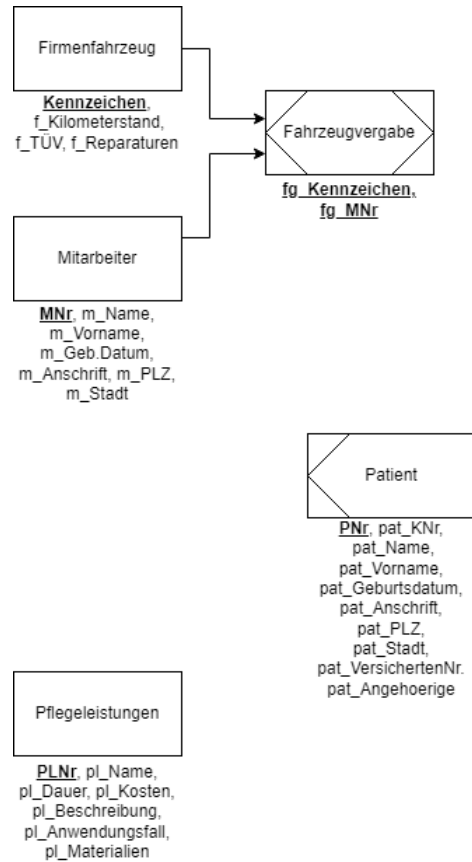


PNr., pat_KNr,
pat_Name,
pat_Vorname,
pat_Geburtsdatum,
pat_Anschrift,
pat_PLZ,
pat_Stadt,
pat_VersichertenNr.
pat_Angehoerige

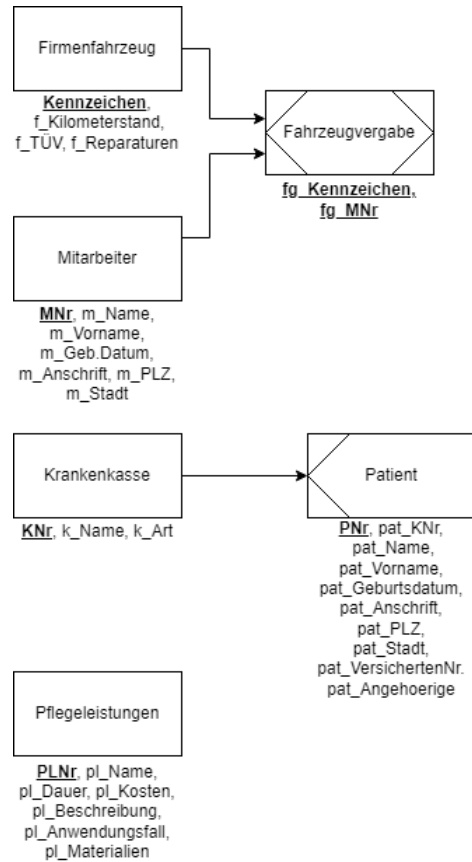


PLNr., pl_Name,
pl_Dauer, pl_Kosten,
pl_Beschreibung,
pl_Anwendungsfall,
pl_Materialien

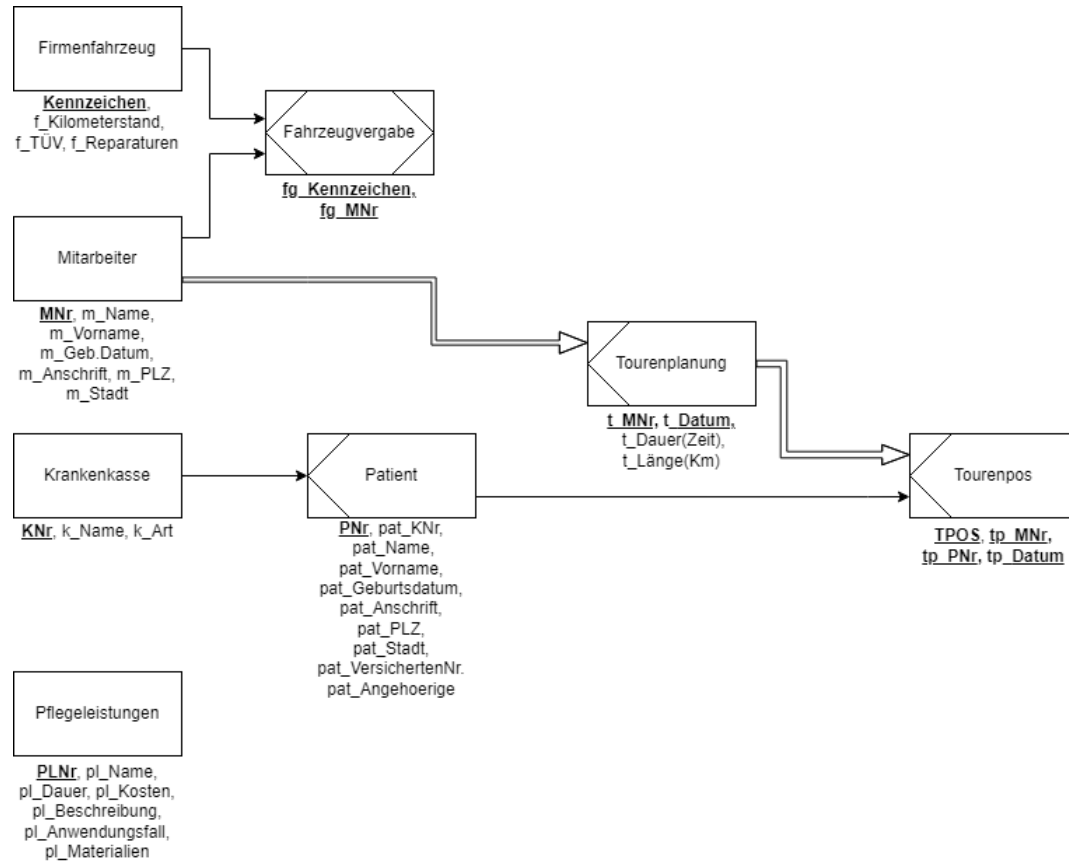
Vor der Erweiterung



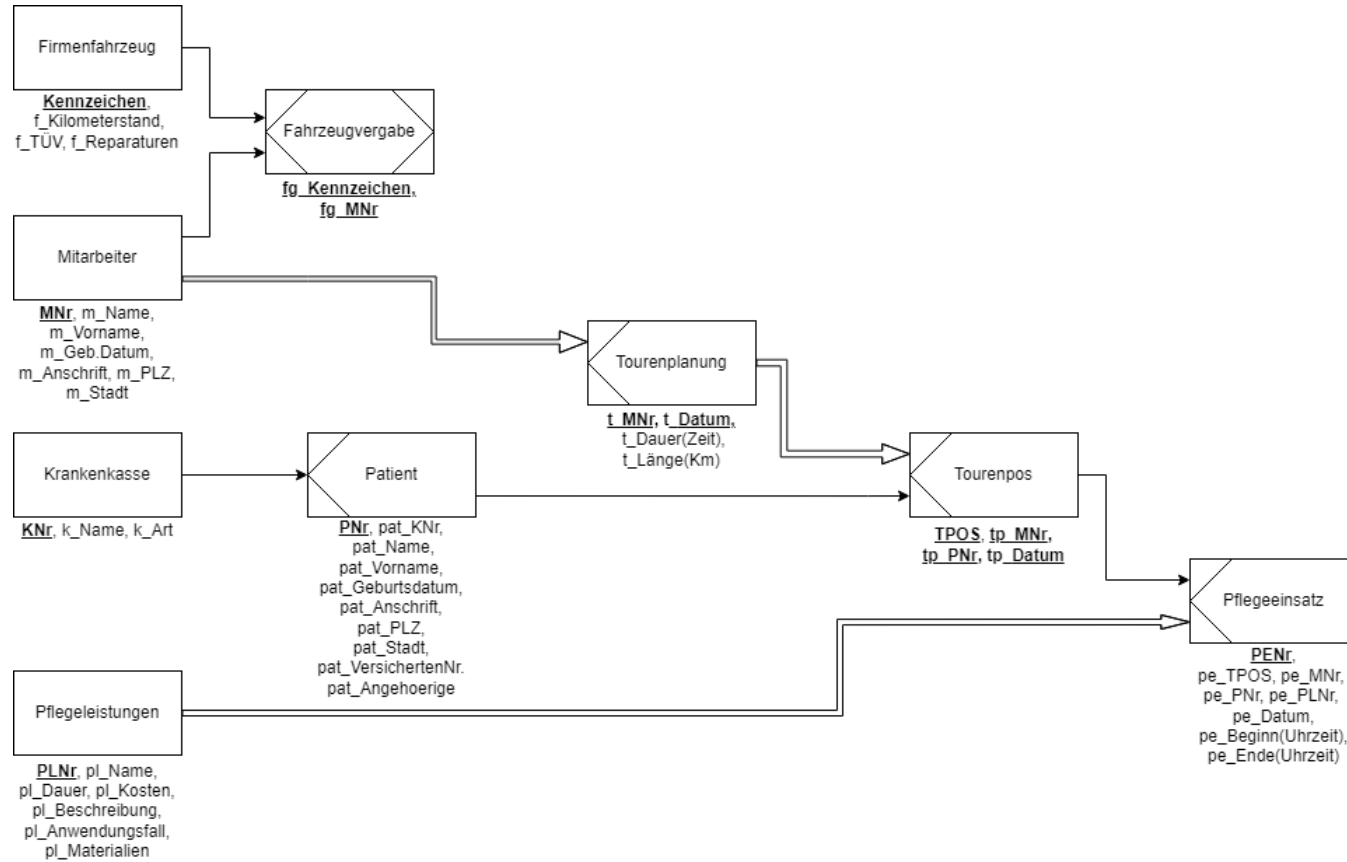
Vor der Erweiterung



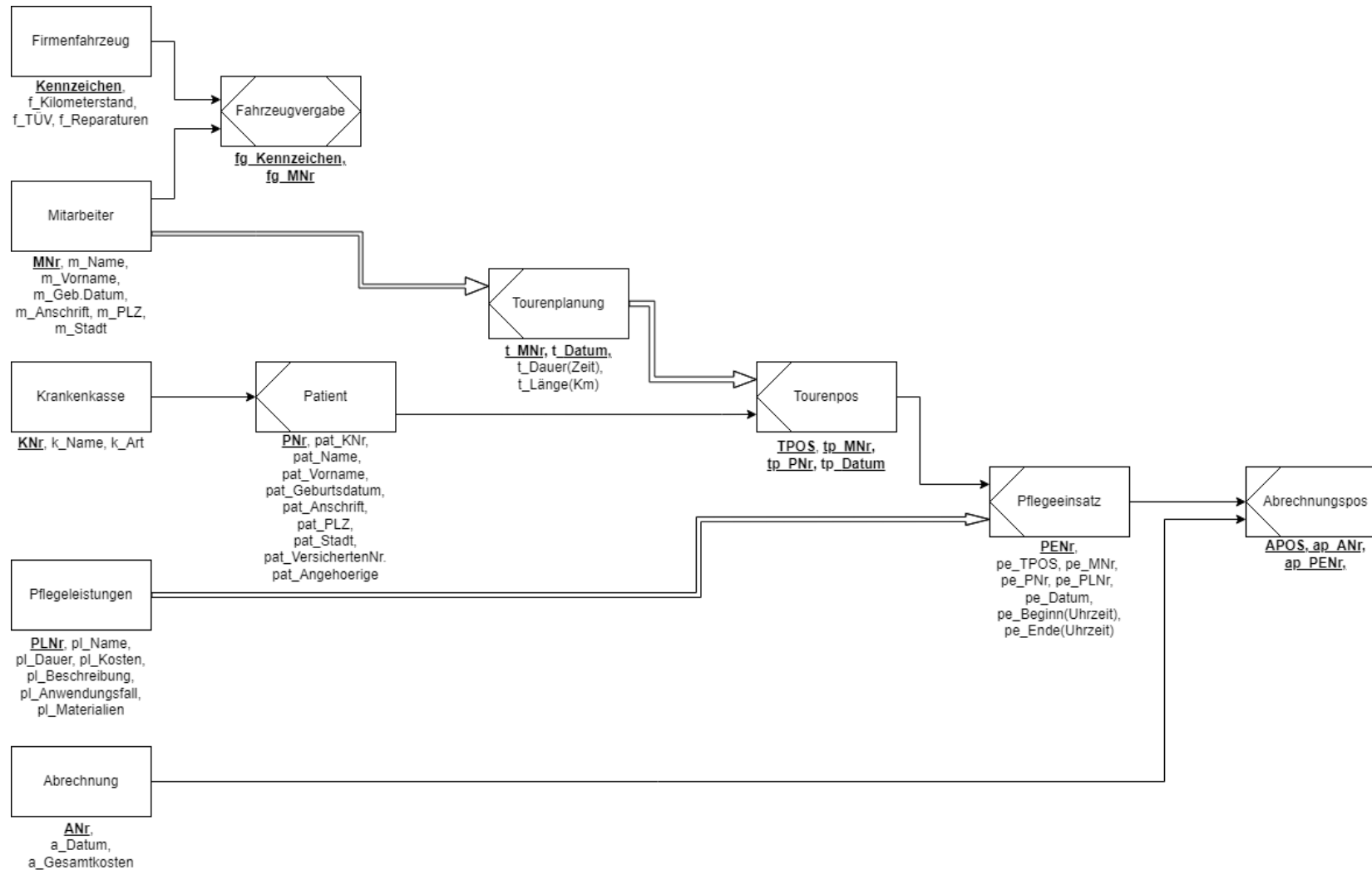
Vor der Erweiterung



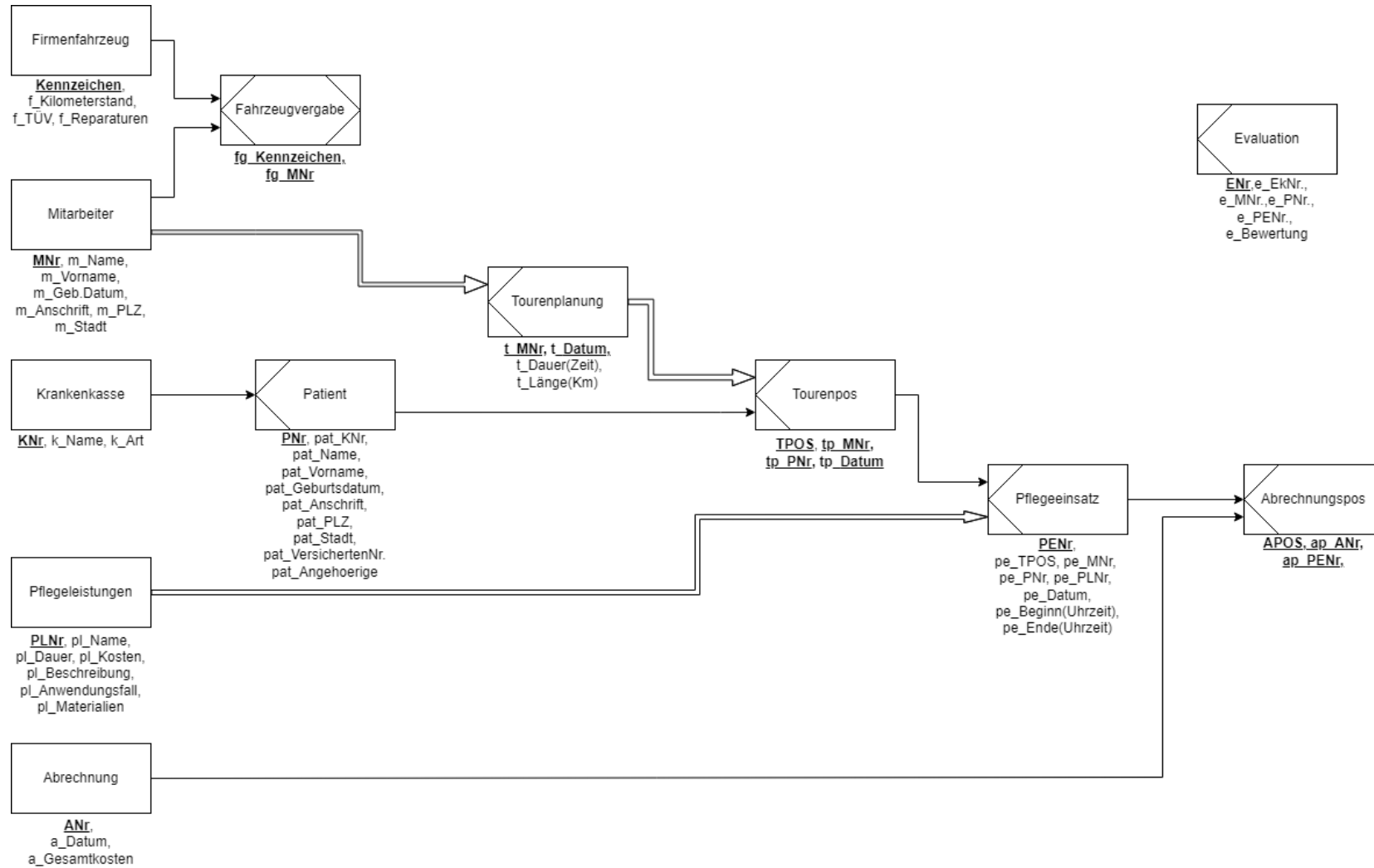
Vor der Erweiterung



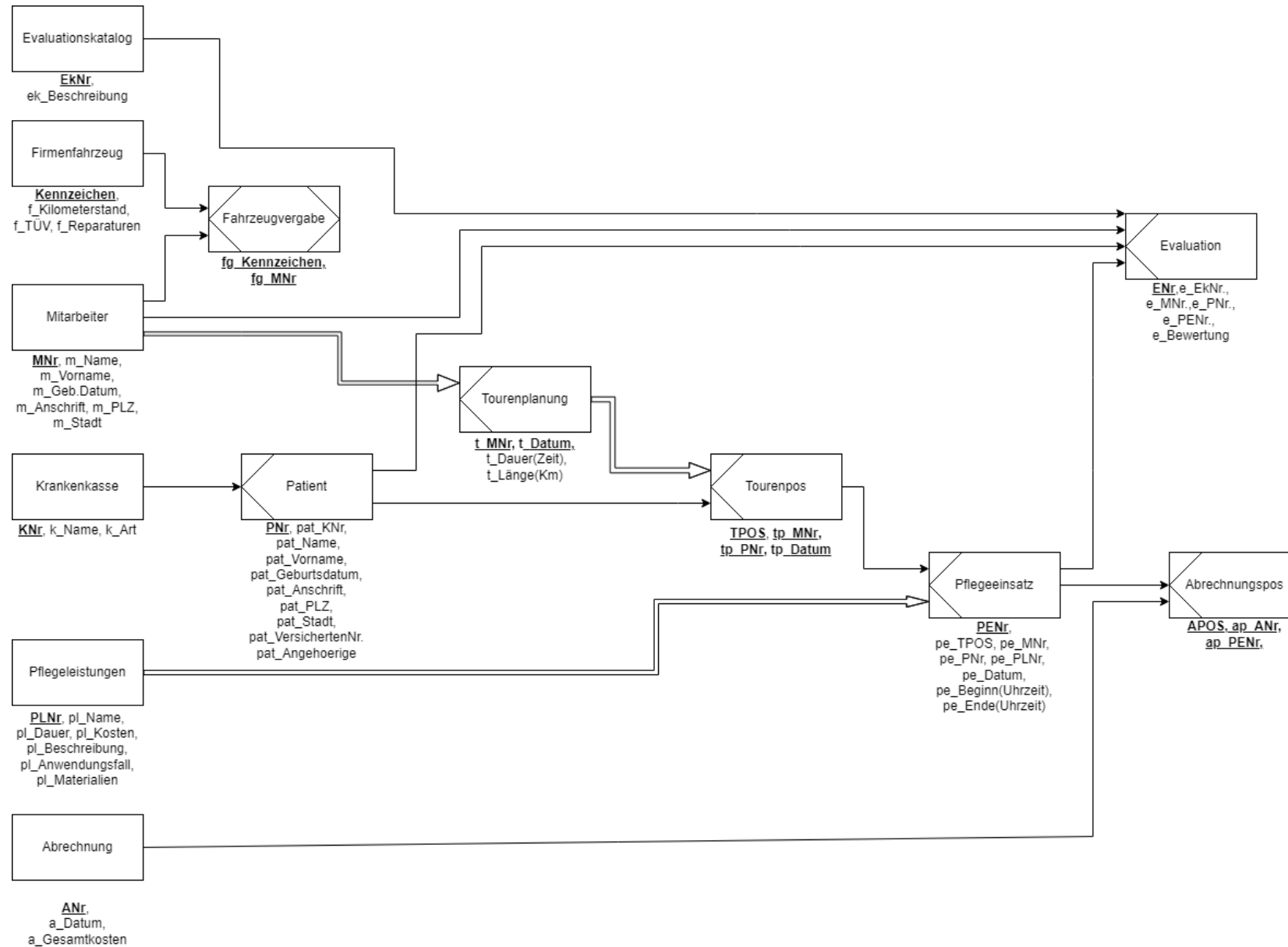
Vor der Erweiterung



Vor der Erweiterung



Vor der Erweiterung



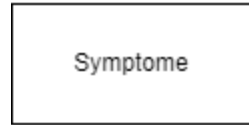
Nach der Erweiterung



Nach der Erweiterung



ErkNr, erk_Name

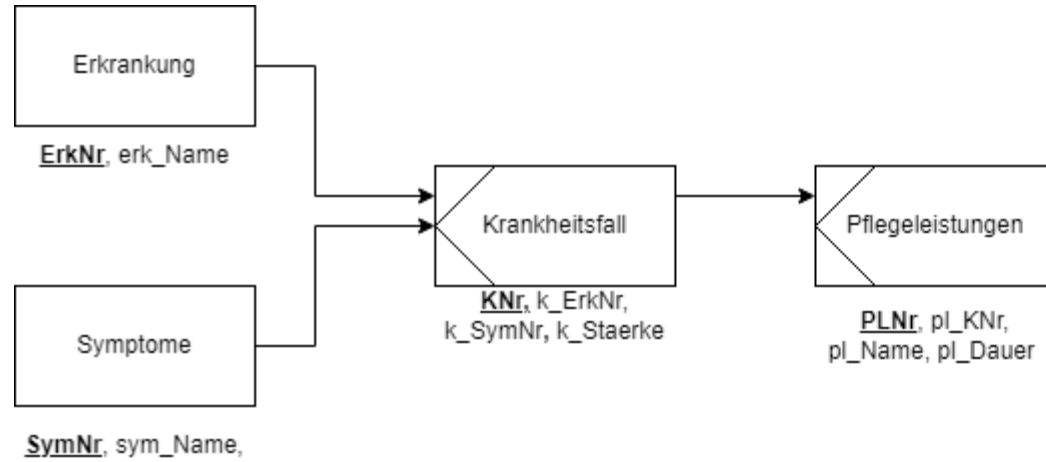


SymNr, sym_Name,

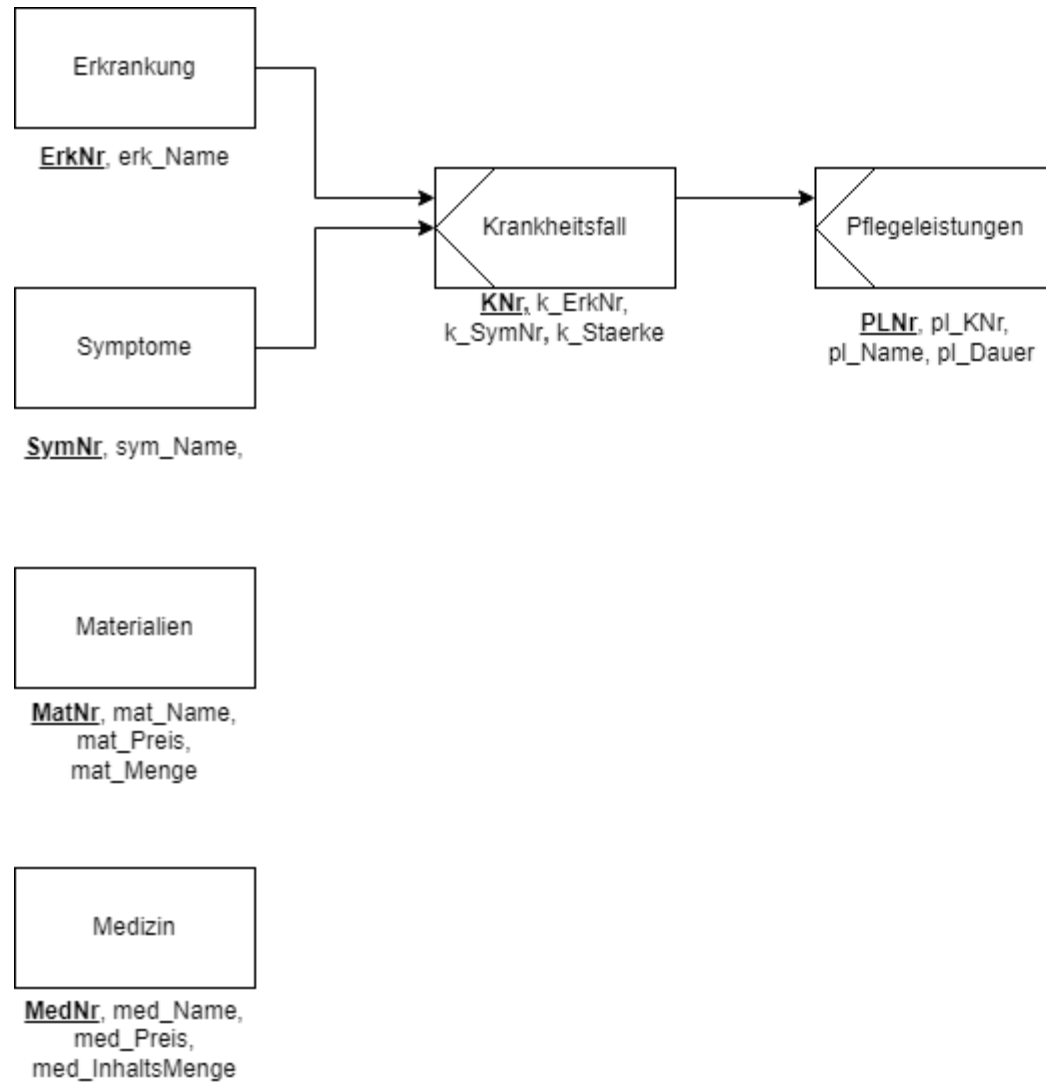


PLNr, pl_KNr,
pl_Name, pl_Dauer

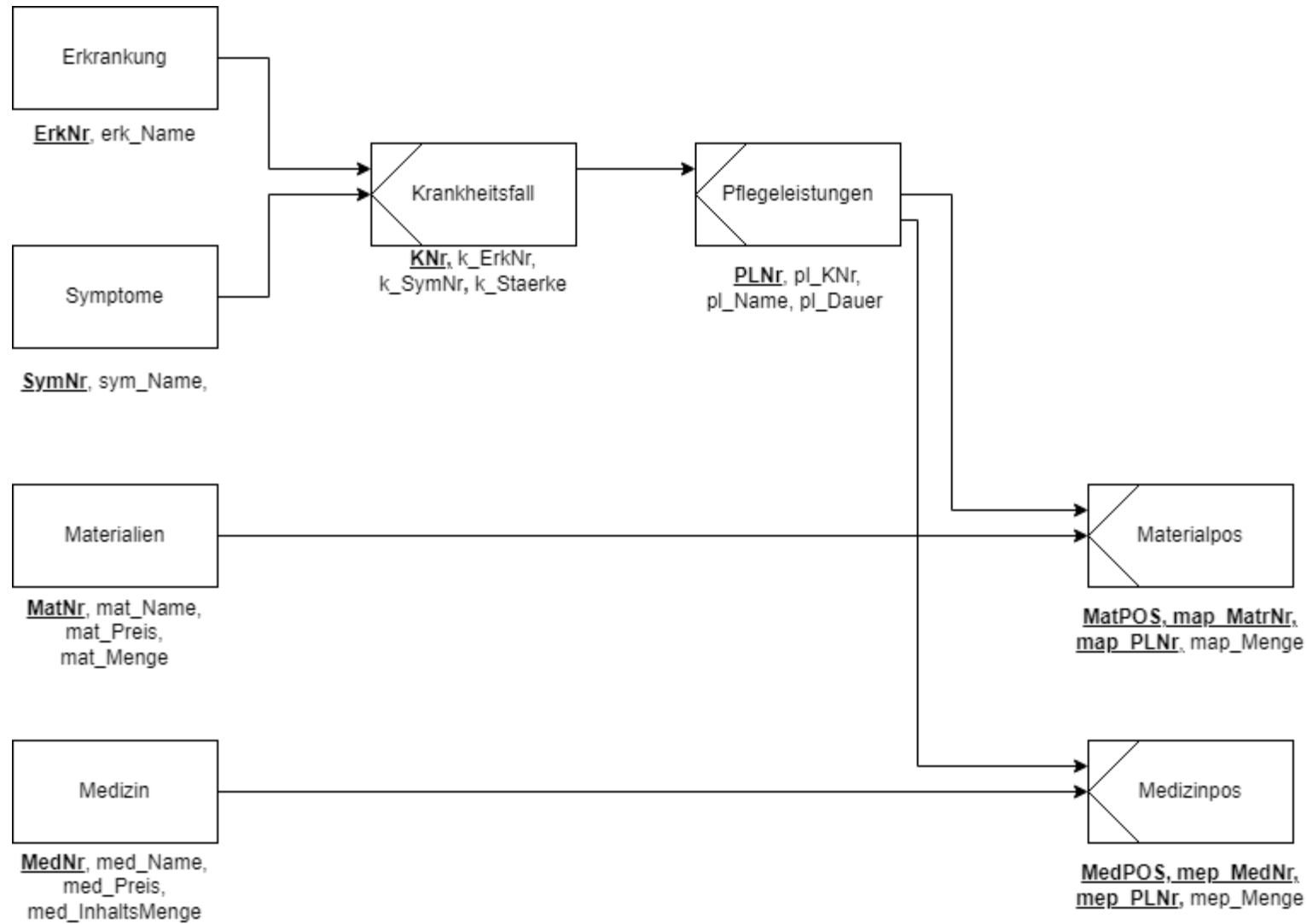
Nach der Erweiterung



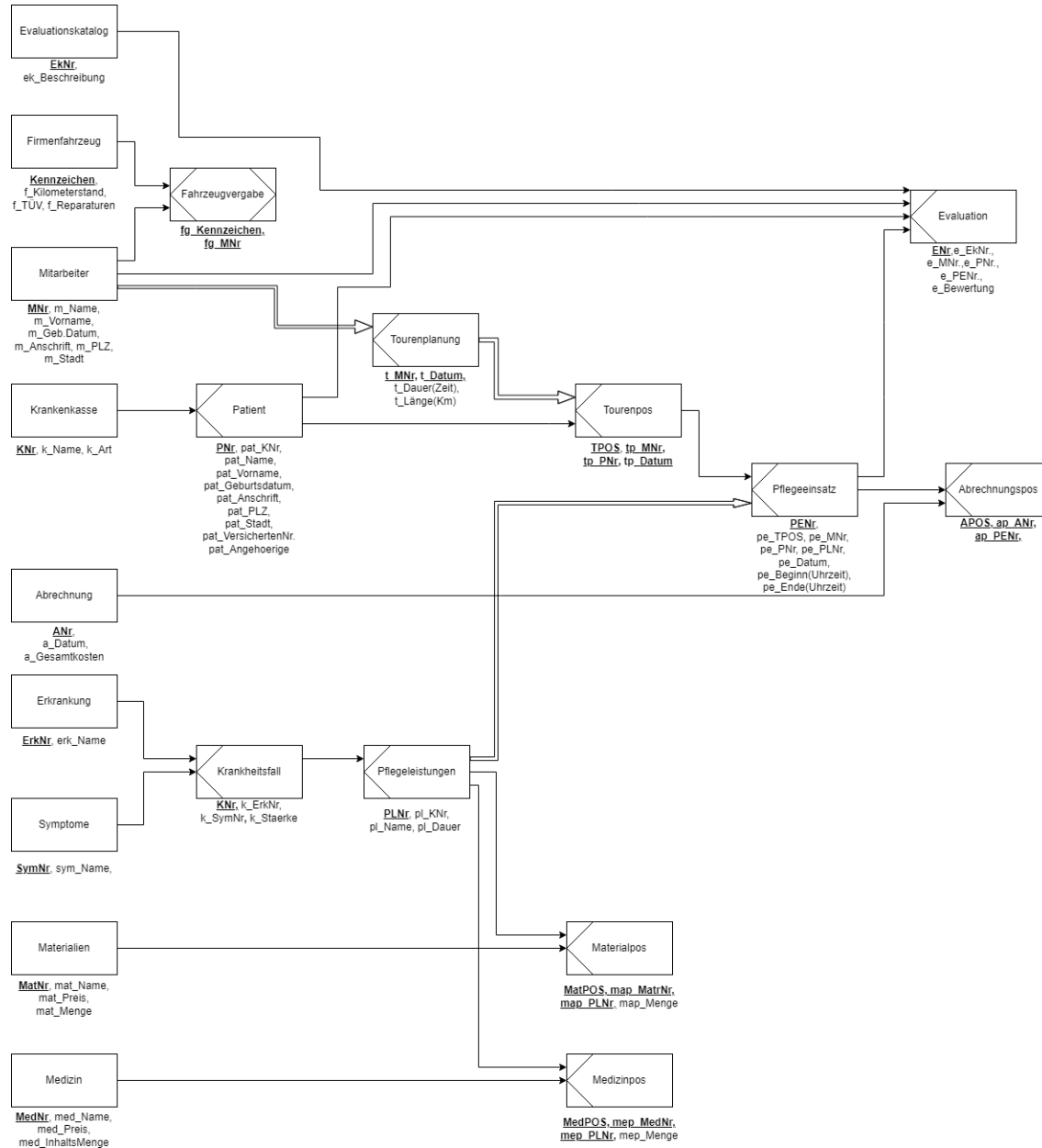
Nach der Erweiterung



Nach der Erweiterung




Nach der Erweiterung



Fallbeispiel Frage 2: Wie alt ist der durchschnittliche Patient?

```
SELECT (SUM(CURRENT_DATE - pat_Geburtsdatum) / COUNT(pat_Geburtsdatum)) / 365.0 AS Durchschnittsalter  
FROM patient  
;
```

	durchschnittsalter numeric 
1	80.8739726027397260

Eigene Frage 2: Wie lange dauerte der längste Einsatz des letzten Jahres und wer war daran beteiligt?

```
CREATE OR REPLACE VIEW einsDauer AS
SELECT (pe.pe_Ende - pe.pe_Beginn) AS Dauer, m.m_Name, m.m_Vorname, pat.pat_Name, pat.pat_Vorname
FROM mitarbeiter m JOIN tourenplanung t ON (MNr = t_MNr)
      JOIN tourenpos tp ON (t_MNr = tp_MNr)
      JOIN patient pat ON (tp_PNr = PNr)
      JOIN pflegeeinsatz pe ON (TPOS = pe_TPOS);

SELECT DISTINCT Dauer, m_Name AS "Mitarbeiter Nachname", m_Vorname AS "Mitarbeiter Vorname",
      pat_Name AS "Patient Nachname", pat_Vorname AS "Patient Vorname"
FROM einsDauer
WHERE Dauer IN (SELECT MAX(Dauer)
                FROM einsDauer)
;
```

	dauer interval	Mitarbeiter Nachname character varying (30)	Mitarbeiter Vorname character varying (30)	Patient Nachname character varying (30)	Patient Vorname character varying (30)
1	01:30:00	Himmel	Fabian	Schneider	Emma
2	01:30:00	Maier	Tom	Becker	Gabriel

Trigger

```
DELIMITER //
CREATE TRIGGER patientenSichern
BEFORE DELETE ON patienten FOR EACH ROW
BEGIN
    CREATE TABLE IF NOT EXISTS sicherungPatienten(
        s_PNr INTEGER,
        s_pat_KNr INTEGER,
        s_pat_Name VARCHAR(30),
        s_pat_Vorname VARCHAR(30),
        s_pat_Geburtsdatum DATE,
        s_pat_Anschrift VARCHAR(100),
        s_pat_PLZ DECIMAL(5,0),
        s_pat_Stadt VARCHAR(40),
        s_pat_VersichertenNr VARCHAR(30),
        s_pat_Angehoerige VARCHAR(200),
        CONSTRAINT pksicherungpatient PRIMARY KEY (s_PNr),
        CONSTRAINT fksicherungpatient FOREIGN KEY (s_pat_KNr) REFERENCES Krankenkasse (KNr)
    );
    INSERT INTO sicherungPatienten(s_PNr, s_pat_KNr, s_pat_Name, s_pat_Vorname, s_pat_Geburtsdatum,
        s_pat_Anschrift, s_pat_PLZ, s_pat_Stadt, s_pat_VersichertenNr,
        s_pat_Angehoerige) VALUES
        (old.s_PNr, old.s_pat_KNr, old.s_pat_Name, old.s_pat_Vorname,
        old.s_pat_Geburtsdatum, old.s_pat_Anschrift, old.s_pat_PLZ,
        old.s_pat_Stadt, old.s_pat_VersichertenNr, old.s_pat_Angehoerige);

END //
DELIMITER ;
```

Stored Procedure

```
DELIMITER //
CREATE PROCEDURE abrGesKost (IN ANr INTEGER, OUT a_Gesamtkosten DECIMAL(10,2))
BEGIN

    DECLARE a_Gesamtkosten DECIMAL(10,2) DEFAULT 0;

    SET a_Gesamtkosten = (SELECT SUM((mat.mat_Preis * map_Menge) + (med.med_Preis * mep_Menge))
        FROM Abrechnung a JOIN Abrechnungspos ap ON (ANr = ap_ANr)
            JOIN Pflegeeinsatz pe ON (ap_PENr = PENr)
            JOIN Pflegeleistungen pl ON (pe_PLNr = PLNr)
            JOIN Materialpos map ON (PLNr = map_PLNr)
            JOIN Materialien mat ON (map_MatNr = MatNr)
            JOIN Medizinpos mep ON (PLNr = mep_PLNr)
            JOIN Medizin med ON (mep_MedNr = MedNr)
        WHERE a.ANr = ANr) AS gesKostenErfassung;

END //
DELIMITER ;
```