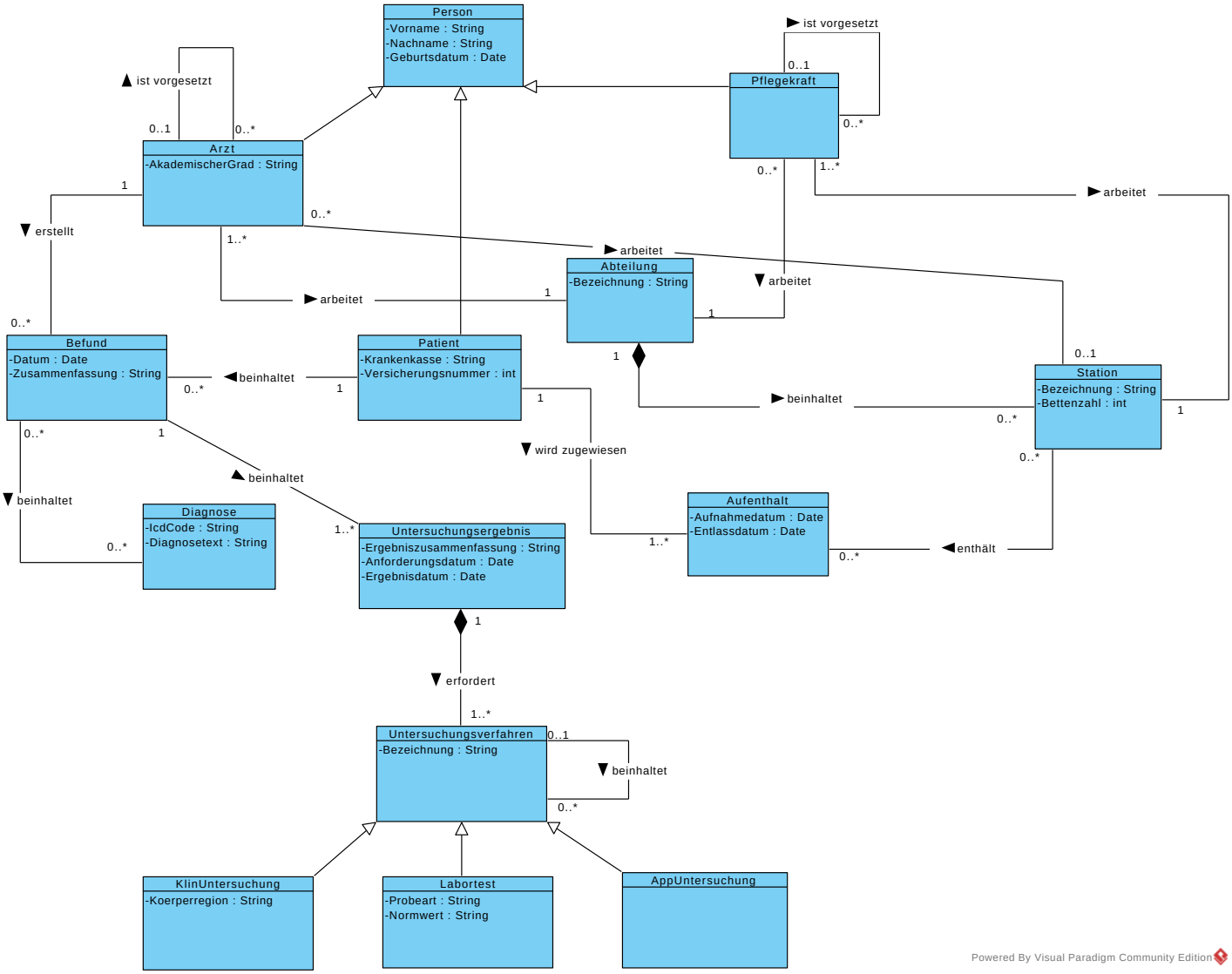
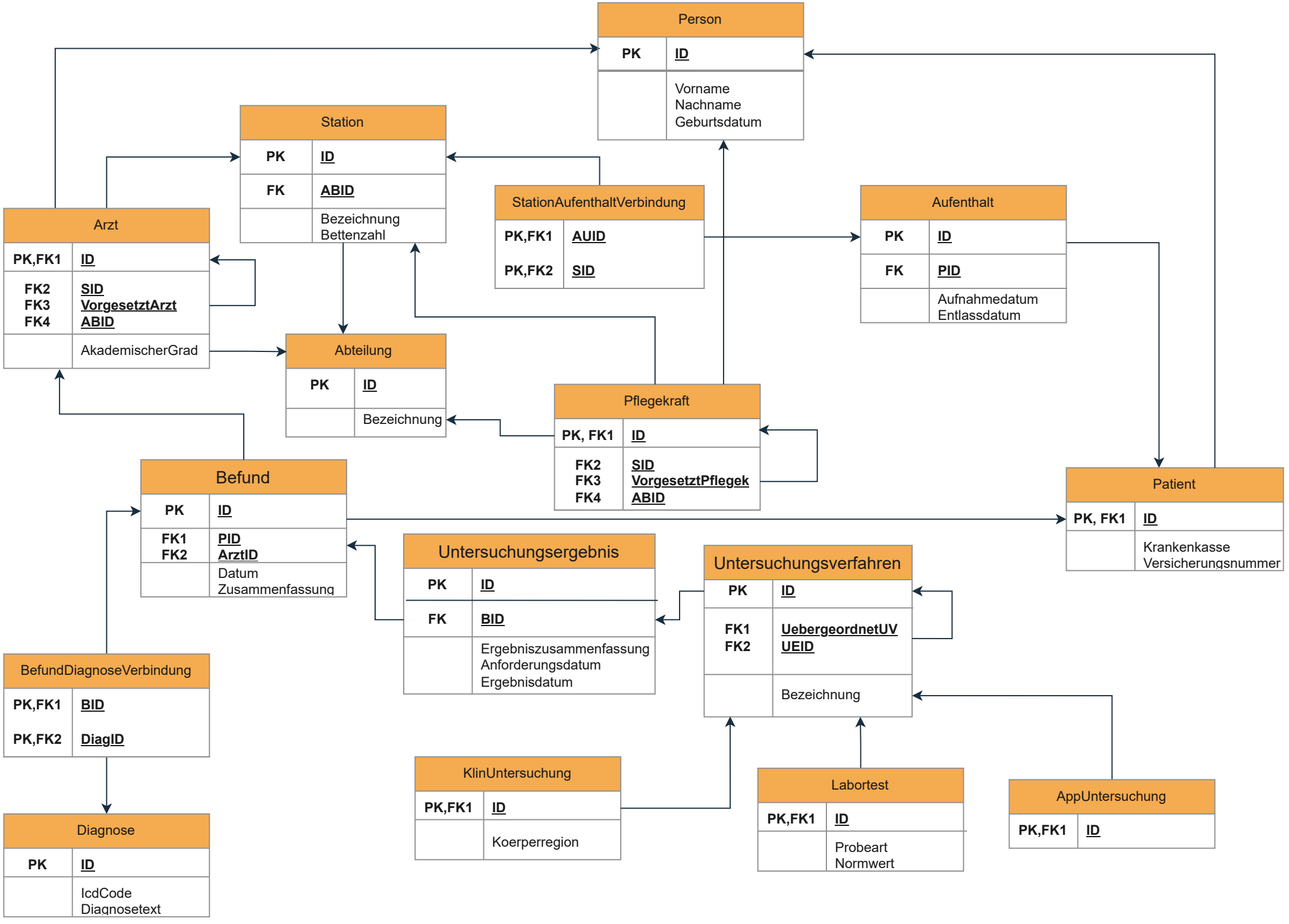

Person
-Vorname : String
-Nachname : String
-Geburtsdatum : Date
ist vorgesetzt
0..1
0..*
Arzt
-AkademischerGrad : String
Pflegekraft
ist vorgesetzt
0..1
0..*
0..*
1..*
arbeitet
1
erstellt
0..*
1..*
arbeitet
Abteilung
-Bezeichnung : String
1
arbeitet
1
1
0..*
Befund
-Datum : Date
-Zusammenfassung : String
beinhaltet
0..*
1
Patient
-Krankenkasse : String
-Versicherungsnummer : int
1
1
beinhaltet
0..1
Station
-Bezeichnung : String
-Bettenzahl : int
1
0..*
0..*
0..*
1
beinhaltet
wird zugewiesen
beinhaltet
Diagnose
-IcdCode : String
-Diagnosetext : String
0..*
1..*
Untersuchungsergebnis
-Ergebniszusammenfassung : String
-Anforderungsdatum : Date
-Ergebnisdatum : Date
Aufenthalt
-Aufnahmedatum : Date
-Entlassdatum : Date
1..*
0..*
enthält
1
erfordert
1..*
Untersuchungsverfahren
-Bezeichnung : String
0..1
beinhaltet
0..*
KlinUntersuchung
-Koerperregion : String
Labortest
-Probeart : String
-Normwert : String
AppUntersuchung
Powered By Visual Paradigm Community Edition


Person
PK
ID
Vorname
Nachname
Geburtsdatum
Station
PK
ID
FK
ABID
Bezeichnung
Bettenzahl
StationAufenthaltVerbindung
PK,FK1
AUID
PK,FK2
SID
Aufenthalt
PK
ID
FK
PID
Aufnahmedatum
Entlassdatum
Arzt
PK,FK1
ID
FK2
FK3
FK4
SID
VorgesetztArzt
ABID
AkademischerGrad
Abteilung
PK
ID
Bezeichnung
Pflegekraft
PK, FK1
ID
FK2
FK3
FK4
SID
VorgesetztPflegek
ABID
Befund
PK
ID
FK1
FK2
PID
ArztID
Datum
Zusammenfassung
Patient
PK, FK1
ID
Krankenkasse
Versicherungsnummer
Untersuchungsergebnis
PK
ID
FK
BID
Ergebniszusammenfassung
Anforderungsdatum
Ergebnisdatum
Untersuchungsverfahren
PK
ID
FK1
FK2
UebergeordnetUV
UEID
Bezeichnung
BefundDiagnoseVerbindung
PK,FK1
BID
PK,FK2
DiagID
KlinUntersuchung
PK,FK1
ID
Koerperregion
Labortest
PK,FK1
ID
Probeart
Normwert
AppUntersuchung
PK,FK1
ID
Diagnose
PK
ID
IcdCode
Diagnosetext

```
  CREATE TABLE Person (
    ID BIGINT PRIMARY KEY,
    Vorname VARCHAR(50),
  Nachname VARCHAR(50),
  Geburtsdatum Date
);
CREATE TABLE Patient(
  ID BIGINT PRIMARY KEY references Person(ID)
  on delete cascade on update cascade,
  Krankenkasse VARCHAR(50),
  Versicherungsnummer VARCHAR(50)
);
CREATE TABLE Aufenthalt (
  ID BIGINT PRIMARY KEY,
  PID BIGINT references Patient(ID)
  on delete set null
on update cascade,
  Aufnahmedatum Date,
  Entlassdatum Date
);
CREATE TABLE Abteilung (
    ID BIGINT PRIMARY KEY,
   Bezeichnung VARCHAR(50)
);
CREATE TABLE Station (
    ID BIGINT PRIMARY KEY,
  ABID BIGINT references Abteilung(ID)
  on delete set null
on update cascade,
  Bezeichnung VARCHAR(50),
  Bettenzahl INT
);
CREATE TABLE StationAufenthaltVerbindung(
  AUID BIGINT,
    SID BIGINT,
    PRIMARY KEY (AUID, SID),
    FOREIGN KEY (AUID) REFERENCES Aufenthalt(ID)
  on delete cascade on update cascade,
    FOREIGN KEY (SID) REFERENCES Station(ID)
  on delete cascade on update cascade
);
CREATE TABLE Pflegekraft (
    ID BIGINT PRIMARY KEY references Person(ID)
  on delete cascade on update cascade,
     SID BIGINT references Station(ID)
```

```
  on delete set null
on update cascade,
  ABID BIGINT references Abteilung(ID)
  on delete set null
on update cascade,
  VorgesetztPflegek BIGINT references Pflegekraft(ID)
  on delete set null
on update cascade
);

CREATE TABLE Arzt (
    ID BIGINT PRIMARY KEY references Person(ID)
  on delete cascade on update cascade,
    SID BIGINT references Station(ID)
  on delete set null
on update cascade,
  ABID BIGINT references Abteilung(ID)
  on delete set null
on update cascade,
  VorgesetztArzt BIGINT references Arzt(ID)
  on delete set null
on update cascade,
  AkademischerGrad VARCHAR(50)
);
CREATE TABLE Befund (
    ID BIGINT PRIMARY KEY,
  ArztID BIGINT references Arzt(ID)
  on delete set null
on update cascade,
  PID BIGINT references Patient(ID)
  on delete set null
on update cascade,
  Datum DATE,
  Zusammenfassung VARCHAR(50)
);
CREATE TABLE Diagnose (
    ID BIGINT PRIMARY KEY,
  IcdCode CHAR(7),
  DiagnoseText VARCHAR(50)
);
CREATE TABLE BefundDiagnoseVerbindung (
  BID BIGINT,
  DiagID BIGINT,
  PRIMARY KEY(BID, DiagID),
  FOREIGN KEY(BID)  references Befund(ID)
```

```
  on delete cascade on update cascade,
  FOREIGN KEY(DiagID)  references Diagnose(ID)
  on delete cascade on update cascade
);
CREATE TABLE Untersuchungsergebnis (
    ID BIGINT PRIMARY KEY,
  BID BIGINT references Befund(ID)
  on delete set null
on update cascade,
  Ergebnisdatum DATE,
  Anforderungsdatum DATE,
  Ergebniszusammenfassung VARCHAR(50)
);
CREATE TABLE Untersuchungsverfahren (
    ID BIGINT PRIMARY KEY,
  Bezeichnung VARCHAR(50),
  UEID BIGINT REFERENCES Untersuchungsergebnis(ID)
  on delete set null
on update cascade,
  UebergeordnetUV BIGINT REFERENCES Untersuchungsverfahren(ID)
  on delete cascade
on update cascade
);
CREATE TABLE KlinUntersuchung (
  ID BIGINT PRIMARY KEY references Untersuchungsverfahren(ID)
  on delete cascade on update cascade,
  Koerperregion VARCHAR(50)
);
CREATE TABLE Labortest  (

  ID BIGINT PRIMARY KEY references Untersuchungsverfahren(ID)
  on delete cascade on update cascade,
  Probeart VARCHAR(50),
  Normwert VARCHAR(50)
);
CREATE TABLE AppUntersuchung  (
  ID BIGINT PRIMARY KEY references Untersuchungsverfahren(ID)
  on delete cascade on update cascade
);
```

Triggers:

1)

Trigger Semantik: Man möchte sicherstellen, dass die Anzahl der Patienten, die einer Station zugeschreiben sind, nicht die Bettenzahl überschreitet.

Trigger Typ : Der Datensatz wird vor dem Einfügen in die StationAufenthaltVerbindung Tabelle den Trigger auslösen.

Trigger Logik: Vor dem Zuschreibungsversuchs eines Aufenthaltes einer bestimmten Station wird mittels JOIN überprüft, ob die Anzahl der sich auf der erwähnten Station befindenden Patienten mit keinem festgelegten Entlassdatum kleiner als die Bettenzahl ist. Falls das der Fall ist, wird der Datensatz erfolgreich eingetragen. Ansonsten wird eine Ausnahme geworfen.

2)

Trigger Semantik: Man möchte sicherstellen, dass ein Patient mindestens über einen Aufenthalt im Krankenhaus verfügt. Ansonsten ist er kein Patient.

Trigger Typ: Beim Löschen eines Aufenthalts als Ereignis, wird der Trigger ausgelöst.

Trigger Logik: Man ermittelt den Patienten, dem der Aufenthalt zugehörig ist. Danach ermittelt man alle Aufenthalteinträge, die der Person gehört. Wenn die Anzahl der Aufenthalte kleiner gleich eins ist, wird eine Ausnahme geworfen. Ansonsten wird der Eintrag erfolgreich vorgenommen.

3)

Trigger Semantik: Man möchte sicherstellen, dass in der Datenbank keine Befunde existieren, die als Datum einen späteren Zeitpunkt als das Geburtsdatum der zugehörigen Person haben.

Trigger Typ: Die Überprüfung wird vor dem Einfügen eines Befundes ausgelöst.

Trigger Logik: Man verbindet die Person, Arzt und Befund Tabellen miteinander mittels JOIN-Verfahren und vergleicht das Geburtsdatum des Patienten mit dem Datum des Befundes. Falls das Datum des Befundes vor dem Geburtsdatum des Patienten liegt, wird eine Ausnahme geworfen. Ansonsten wird der Eintrag erfolgreich vorgenommen.