Onyedikachi Okenwa

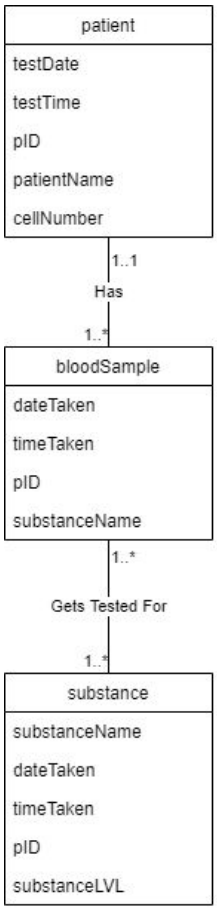
Mark Garcia

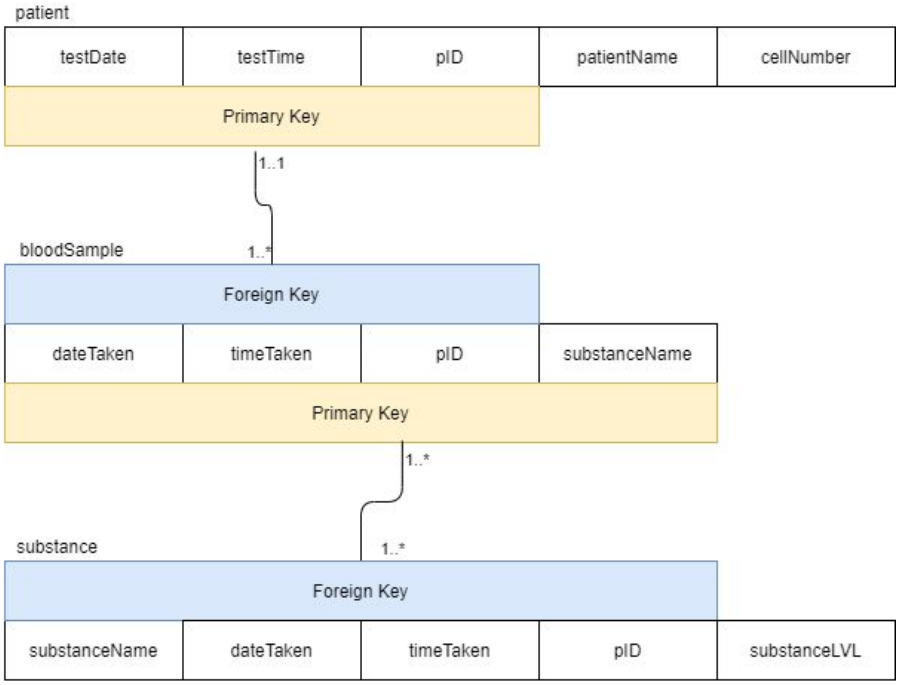
Khang Dao

HW 4 Patients and Blood Samples

In our class model, we created 3 different classes; one for patient, bloodSample, and substance. We defined the primary key of patient to be the testDate, testTime, and patientID (pID). For bloodSample, the primary key consists of dateTaken, timeTaken, pID, substanceName and the foreign key was dateTaken, timeTaken, and pID since we wanted to relate a bloodSample to a patient. Patient can have 1 to many bloodSamples but bloodSample can only belong to 1 patient. Our substance table had only a foriegn key that consisted of substanceName, dateTaken, timeTaken, and pID. In this table we are going to keep track of the level of the substance that is being tested for. It has a 1 to many relationship with bloodSample because a bloodSample can be tested for many things.

We chose our primary key for patient because the only way to differentiate between different bloodSamples was through the time and date it was taken. Since we are keeping track of multiple patients, we also included pID so that way patients with the same name can be tracked by their patient ID rather than their name. Similarly, we chose our primary key for bloodSample to be the same as the primary key for patient for the same reasons, but this time we included substanceName to make sure that one entry bloodSample cannot be tested for the same thing twice.





SQL TO CREATE TABLES & INSERT VALUES:

```
CREATE TABLE patient(
   testDate       date       NOT NULL,
   testTime       time       NOT NULL,
   pID            int        NOT NULL,
   patientName    varchar(100)       NOT NULL,
   cellNumber     int        NOT NULL,

   CONSTRAINT     patient_PK     PRIMARY KEY     (testDate, testTime, pID)
);
```

```
CREATE TABLE bloodSample(
   dateTaken       date        NOT NULL,
   timeTaken       time        NOT NULL,
   pID             int         NOT NULL,
   substanceName   varchar(100)       NOT NULL,


   CONSTRAINT      bloodSample_FK      FOREIGN KEY     (dateTaken, timeTaken, pID)
REFERENCES patient (testDate, testTime, pID),
   CONSTRAINT      bloodSample_PK      PRIMARY KEY     (dateTaken, timeTaken, pID,
substanceName)
);

CREATE TABLE substance(
   dateTaken       date        NOT NULL,
   timeTaken       time        NOT NULL,
   pID             int         NOT NULL,
   substanceName       varchar(100)       NOT NULL,
   substanceLVL        int                NOT NULL,

   CONSTRAINT      substance_FK        FOREIGN KEY     (dateTaken, timeTaken, pID,
substanceName) REFERENCES bloodSample(dateTaken, timeTaken, pID, substanceName)
);


INSERT INTO patient (testDate, testTime, pID, patientName, cellNumber)
```

```
VALUES ('2020-01-21', '10:24:20', 123456, 'John Doe', 56222),
    ('2019-05-05', '12:56:06', 123415, 'Bob Thompson', 98342),
    ('2020-12-21', '14:31:04', 098123, 'James Smith', 21353);

INSERT INTO bloodSample(dateTaken, timeTaken, pID, substanceName)
VALUES ('2020-01-21', '10:24:20', 123456, 'sugar'),
    ('2020-01-21', '10:24:20', 123456, 'LDL'),
    ('2019-05-05', '12:56:06', 123415, 'HDL'),
    ('2019-05-05', '12:56:06', 123415, 'alcohol'),
    ('2020-12-21', '14:31:04', 098123, 'calcium'),
    ('2020-12-21', '14:31:04', 098123, 'sodium');

INSERT INTO bloodSample(dateTaken, timeTaken, pID, substanceName)
VALUES ('2019-05-05', '12:56:06', 123415, 'LDL'),
    ('2020-12-21', '14:31:04', 098123, 'LDL');


INSERT INTO substance(dateTaken, timeTaken, pID, substanceName, substanceLVL)
VALUES ('2020-01-21', '10:24:20', 123456, 'sugar', 122),
    ('2020-01-21', '10:24:20', 123456, 'LDL', 423),
    ('2019-05-05', '12:56:06', 123415, 'HDL', 314),
    ('2019-05-05', '12:56:06', 123415, 'alcohol', 123),
    ('2020-12-21', '14:31:04', 098123, 'calcium', 30),
    ('2020-12-21', '14:31:04', 098123, 'sodium', 192),
    ('2019-05-05', '12:56:06', 123415, 'LDL', 99),
    ('2020-12-21', '14:31:04', 098123, 'LDL', 160);
```

SQL THAT WILL LIST THE PATIENT NAME, TIME, AND DATE OF ANY BLOOD SAMPLES SHOW A VALUE OF LDL CHOLESTEROL > 160 mg/dL

```
SELECT patientName, testDate, testTime FROM patient p

NATURAL JOIN bloodSample b

NATURAL JOIN substance s

WHERE (s.SUBSTANCELVL > 160 AND s.SUBSTANCENAME = 'LDL');
```

SAME QUERY USING RELATIONAL ALGEBRA

π patient.patientName, patient.testDate, patient.testTime (σ substance.substanceName = 'LDL'

⋀ substance.substanceLVL > 160) (((patient) ⋈ patient.pID = bloodSample.pID (bloodSample))

⋈ bloodSample.substanceName = substance.substanceName (substance))