CS 510, FA 2021
DB Account name: joyroyweb
Password: 123456
Date: 12/03/2021

Final Project

Title: Hospital Management System Database

Submitted By

Adrianna Bowen
Bhadhan Roy Joy
Daniel Fletcher Foote
Pravasini Pati

We Adrianna Bowen, Bhadhan Roy Joy, Daniel Fletcher Foote and Pravasini Pati declare that I have completed this assignment completely and entirely on my own, without any consultation with others. I understand that any breach of the UAB Academic Honor Code may result in severe penalties.

We declare that we have completed this assignment completely and entirely on our own, without any consultation with others. We have read the UAB Academic Honor Code and understand that any breach of the Honor Code may result in severe penalties.

We also declare that the following percentage distribution *faithfully* represents individual group members' contributions to the completion of the assignment

Name	Overall Contribution (%)	Major work items completed by me	Signature or initials	Date
Adriana Bowen	25%	E-R Diagram, Relational Schema, Store Procedure, Application Background, Assumptions	AB	12-03-2021
Bhadhan Roy Joy	25%	Relation Schema, Sample data, E-R diagram, Constraints, Application Background, Assumptions	BRJ	12-03-2021
Daniel Fletcher Foote	25%	Views, Indexes, E-R diagram, Triggers, Application Background, Assumptions	DFF	12-03-2021
Pravasini Pati	25%	Triggers, E-R Diagram, Relational Schema, Sample data, Application Background, Assumptions	PP	12-03-2021

DELIVERABLE A

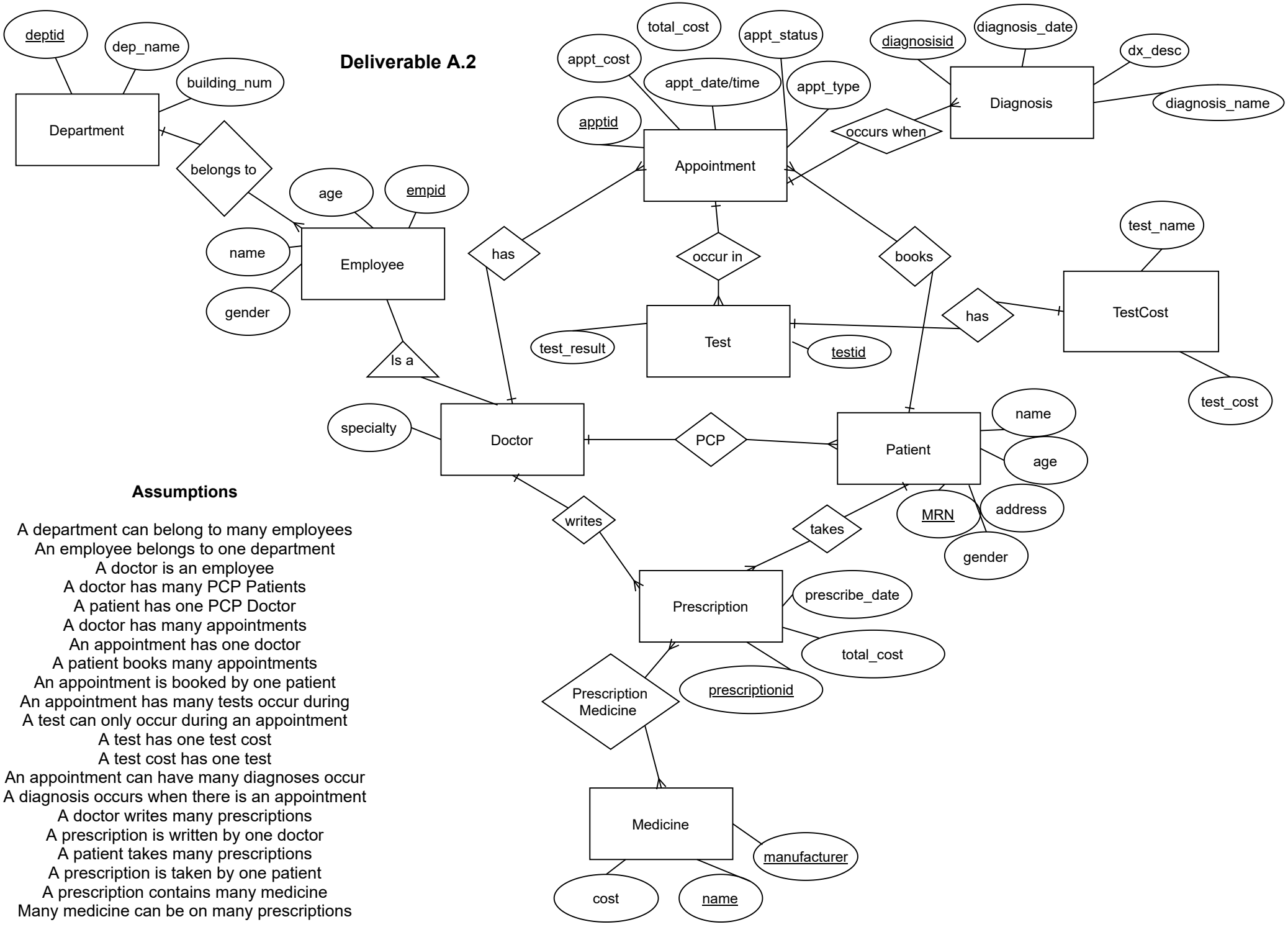
Username: joyroyweb
Password: 123456

Databases have become a critical component in the healthcare system. It allows healthcare facilities to properly assist patients and handle their medical needs, as well as manage medical operations. We chose to build a Hospital Database System. Databases in Hospitals are important for tracking employees, patients, appointments, patient information including their diagnoses, tests, prescriptions, medicines and more. Our database focuses on collecting information for healthcare operations.

The assumptions include the following:

A department can belong to many employees
An employee belongs to one department
A doctor is an employee
A doctor has many PCP Patients
A patient has one PCP Doctor
A doctor has many appointments
An appointment has one doctor
A patient books many appointments
An appointment is booked by one patient
An appointment has many tests occur during
A test can only occur during an appointment
A test has one test cost
A test cost has one test
An appointment can have many diagnoses occur
A diagnosis occurs when there is an appointment
A doctor writes many prescriptions
A prescription is written by one doctor
A patient takes many prescriptions
A prescription is taken by one patient
A prescription contains many medicine
Many medicine can be on many prescriptions

Deliverable A.2



Assumptions

A department can belong to many employees
An employee belongs to one department
A doctor is an employee
A doctor has many PCP Patients
A patient has one PCP Doctor
A doctor has many appointments
An appointment has one doctor
A patient books many appointments
An appointment is booked by one patient
An appointment has many tests occur during
A test can only occur during an appointment
A test has one test cost
A test cost has one test
An appointment can have many diagnoses occur
A diagnosis occurs when there is an appointment
A doctor writes many prescriptions
A prescription is written by one doctor
A patient takes many prescriptions
A prescription is taken by one patient
A prescription contains many medicine
Many medicine can be on many prescriptions

DELIVERABLE B – Relation Schema

Relation Department (deptid, dept_name, building_num)

The Functional dependency is: deptid -> {dept_name, building_num}

deptid is a superkey, as it determines all other attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation Employee (empid, deptid, name, gender, age)
Empid is the primary key, and deptid is the foreign key referencing department(deptid)

FD of Employee relation: empid -> {deptid, name, gender, age}.

employeeid is a superkey, as it determines all other attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation Doctor (empid, specialty)
Primary key emplid, which references Employee(empid)
FD: empid -> specialty.

employeeid is a superkey, as it determines specialty attribute. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation Patient (mrn, name, gender, age, address)

Primary key: mrn
FD: mrn -> {name, gender, age, address}

mrn is a superkey, as it determines all other attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation PCP_Patient (empid, mrn)

Primary keys: {empid, mrn}.

There are no other attributes. So, there is no transitive dependencies. The relation is in 3NF as a result.

Relation Appointment(apptid, empid, mrn, appt_date_time, appt_type, appt_cost, total_cost, appt_status)

Primary key: apptid, foreign keys empid referencing Doctor(empid), and mrn referencing Patient(mrn)

FD: apptid -> {empid, mrn, appt_date_time, appt_type, appt_cost, total_cost, appt_status }

apptid is a superkey, as it determines all other attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation Diagnosis (<u>diagnosisid</u>, apptid, diagnosis_date, diagnosis_name, dx_desc)
Primary key: diagnosisid, foreign key apptid referencing Appointment(apptid)

FD: diagnosisid -> { apptid, diagnosis_date, diagnosis_name, dx_desc }

diagnosisid is a superkey, as it determines all other attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation TestCost (<u>test_name</u>, test_cost);

Primary key: test_name

FD: test_name -> test_cost

test_name is a superkey, as it determines all other attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation Test (<u>testid</u>, apptid, test_name, test_result)

Primary key: testid, foreign key apptid referencing Appointment(apptid)

FD: testid -> {apptid, test_name, test_result}

test_id is a superkey, as it determines all other attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation Medicine (name, manufacturer, cost)

Primary keys (name, manufacturer)

FD: (name, manufacturer) -> cost

(name, manufacturer) is a superkey, as it determines the cost attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation Prescription (prescriptionid, empid, mrn, prescribe_date, total_cost)

Primary key: prescriptionid, foreign keys empid referencing Doctor(empid) and mrn referencing Patient(mrn)

FD: prescriptionid -> {empid, mrn, prescribe_date, total_cost}

prescriptionid is a superkey, as it determines all other attributes. Also, there is no transitive dependency because this is the only FD of the relation, and no non-prime attribute determines another non-prime attribute. So, the relation is in 3NF.

Relation PrescriptionMedicine (prescriptionid, name, manufacturer)

Primary keys (prescriptionid, name, manufacturer). prescriptionid references

Prescription(prescriptionid); and (name, manufacturer) references Medicine(name, manufacturer)

As all of the attributes combine to make the primary key and all are prime attributes, there is no transitive dependency. As a result, the relation is in 3NF.

DELIVERABLE C – Sample Data

```
ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu — 80×...
~~{0.00}~{abowen13@vulcan21:~}~~
$ psql -h cisdb -U joyroyweb -d joyroy
Password for user joyroyweb:
psql (9.2.24, server 9.6.3)
WARNING: psql version 9.2, server version 9.6.
         Some psql features might not work.
Type "help" for help.

joyroy=> select * from department;
 deptid |         dept_name         | buidling_num
--------+---------------------------+--------------
      1 | Medicine                  | BDB
      2 | Neurology                 | NB
      3 | Gastroenterology          | GB
      4 | Rheumatology              | kirklinB
      5 | Physical Therapy          | SpainB
      6 | Genetics                  | GenB
      7 | Urology                   | UroB
      8 | Cardio                    | CardB
      9 | General Internal Medicine | GimB
     10 | Preventive Medicine       | PrevB
(10 rows)

joyroy=>
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu — 80×...

```
     10 | Preventive Medicine       | PrevB
(10 rows)

joyroy=> select * from employee;
 empid | deptid |      name      | gender | age
-------+--------+----------------+--------+-----
     1 |      1 | John Doe       | M      |  35
     2 |      1 | Jane Doe       | F      |  53
     3 |      1 | Robert Brown   | M      |  30
     4 |      2 | Julia Brown    | M      |  64
     5 |      2 | Jan Jansen     | F      |  55
     6 |      2 | Li Li          | M      |  39
     7 |      3 | Hattori Hanzo  | M      |  55
     8 |      3 | Gaurav Patel   | F      |  41
     9 |      3 | Robin Williams | M      |  51
    10 |      4 | Robert Hook    | M      |  33
    11 |      4 | Tom Button     | F      |  56
    12 |      4 | Rob Robinson   | M      |  29
    13 |      5 | Bill Bull      | M      |  49
    14 |      5 | John Smith     | F      |  50
    15 |      5 | Mark Butcher   | M      |  39
(15 rows)

joyroy=>
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu — 80×...

```
    13 |       5 | Bill Bull       | M       |  49
    14 |       5 | John Smith      | F       |  50
    15 |       5 | Mark Butcher    | M       |  39
(15 rows)

joyroy=> select * from doctor;
 empid |      speciality
-------+----------------------
     1 | head
     2 | liver
     3 | teeth
     4 | AS
     5 | child health
     7 | childcare
     8 | cancer
     9 | COVID
    10 | family health
    11 | Infectious diseases
    12 | AIDS
    14 | Parkinson's disease
    15 | Stroke
(13 rows)

joyroy=>
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu — 119×24

```
 apptid | empid | mrn  |    appt_date_time    | appt_type | appt_cost | total_cost | appt_status
--------+-------+------+----------------------+-----------+-----------+------------+-------------
      3 |     1 | 1002 | 2021-01-08 10:05:00 | virtual   |     30.00 |            | Cancelled
      4 |     1 | 1003 | 2021-01-08 11:05:00 | virtual   |     30.00 |            | No-show
      7 |     1 | 1006 | 2021-01-08 14:05:00 | in-person |     30.00 |            | Cancelled
      8 |     3 | 1007 | 2021-01-08 15:05:00 | virtual   |     30.00 |            | No-show
     11 |     1 | 1010 | 2021-01-08 10:05:00 | in-person |     30.00 |            | Cancelled
     12 |    12 | 1011 | 2021-01-08 11:05:00 | virtual   |     30.00 |            | No-show
     13 |    12 | 1009 | 2021-12-03 13:30:00 | virtual   |     30.00 |            | Scheduled
     14 |     2 | 1010 | 2021-12-03 14:30:00 | in-person |     30.00 |            | Scheduled
     15 |     3 | 1011 | 2021-01-08 09:05:00 | virtual   |     30.00 |            | Scheduled
     16 |     1 | 1012 | 2021-12-04 08:00:00 | virtual   |     30.00 |            | Scheduled
     17 |     1 | 1013 | 2021-12-04 09:00:00 | in-person |     30.00 |            | Scheduled
     18 |     1 | 1014 | 2021-12-04 10:00:00 | in-person |     30.00 |            | Scheduled
     19 |     1 | 1005 | 2021-12-04 11:00:00 | virtual   |     30.00 |            | Scheduled
     20 |     2 | 1002 | 2021-12-04 12:00:00 | virtual   |     30.00 |            | Scheduled
     21 |     3 | 1001 | 2021-12-04 13:00:00 | in-person |     30.00 |            | Scheduled
     22 |     1 | 1001 | 2021-12-04 14:00:00 | in-person |     30.00 |            | Scheduled
     23 |     3 | 1001 | 2021-12-04 15:00:00 | virtual   |     30.00 |            | Scheduled
     24 |     2 | 1006 | 2021-12-04 16:00:00 | in-person |     30.00 |            | Scheduled
     25 |     4 | 1007 | 2021-12-04 17:00:00 | virtual   |     30.00 |            | Scheduled
     26 |     1 | 1008 | 2021-12-05 10:00:00 | in-person |     30.00 |            | Scheduled
     27 |    12 | 1009 | 2021-12-06 11:00:00 | virtual   |     30.00 |            | Scheduled
--More--
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moa...

```
joyroy=>
joyroy=>
joyroy=> select * from test;
 testid | apptid |  test_name  |     test_result
--------+--------+-------------+----------------------
      1 |      1 | X-ray       | Normal
      2 |      2 | X-ray       | Discloation
      3 |      5 | X-ray       | Wear and Tear
      4 |      6 | X-ray       | Normal
      5 |      9 | X-ray       | Normal
      6 |     10 | X-ray       | Normal
      7 |      1 | Bloodwork   | Normal
      8 |      2 | Bloodwork   | Elevated cholesterol
      9 |      5 | Bloodwork   | Low gloucose
     10 |      6 | Bloodwork   | Low gloucose
     11 |      9 | Bloodwork   | Elevated cholesterol
     12 |     10 | Bloodwork   | Low gloucose
     13 |      1 | MRI         | Normal
     14 |      2 | MRI         | Normal
     15 |      5 | MRI         | Normal
     16 |      6 | MRI         | Normal
     17 |      9 | MRI         | Normal
     19 |      1 | Colonoscopy | Polyps
     20 |      2 | Colonoscopy | Normal
     21 |      5 | Colonoscopy | Normal
     22 |      6 | Colonoscopy | Normal
--More--
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu — 99×22

```
joyroy=>
joyroy=>
joyroy=>
joyroy=> select * from diagnosis;
 diagnosisid | apptid | diagnosis_date |  diagnosis_name  |         dx_desc
-------------+--------+----------------+------------------+------------------------
           1 |      1 | 2021-01-09     | COVID            | asympomatic
           2 |      2 | 2021-01-09     | Flu              | Seasonal Flu
           3 |      5 | 2021-01-09     | AIDS             | Under control
           4 |      6 | 2021-01-09     | Fever            | 104 degrees
           5 |      9 | 2021-01-09     | Pain             | pain in joints
           6 |     10 | 2021-01-09     | Headache         | Severe headache
           7 |      1 | 2021-01-09     | Pain             | Pain in whole body
           8 |      2 | 2021-01-09     | Amnesia          | extreme low blood count
           9 |      5 | 2021-01-09     | Loss of appetite |
          10 |      6 | 2021-01-09     | Weekness         |
          11 |      9 | 2021-01-09     | Fatigue          |
          12 |     10 | 2021-01-09     | Sleepiness       | Sleeps 12 hours a day
          13 |      1 | 2021-01-09     | Weight loss      | BMI < 20
(13 rows)

joyroy=>
```

ssh abowen13@moat.cs.uab.edu —...

```
                                ^
joyroy=> select * from testcost;
       test_name        | test_cost
------------------------+-----------
 X-ray                  |    100.00
 Bloodwork              |     50.00
 MRI                    |   1000.00
 Colonoscopy            |    200.00
 Urine Test             |    150.00
 Drug Test              |    200.00
 Ultrasound             |     30.00
 CT Scan                |    200.00
 Nerve Conduction Test  |    500.00
 COVID-19 Test          |     70.00
(10 rows)

joyroy=>
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat....

```
joyroy=>
joyroy=>
joyroy=> select * from patient;
 mrn  |      name      | gender | age |    address
------+----------------+--------+-----+----------------
 1000 | Rob Robinson   | M      |  12 | Birmingham, AL
 1001 | Bill Balmer    | F      |  90 | Pelham, AL
 1002 | John Smith     | M      |  32 | Homewood, AL
 1003 | Mark Butcher   | M      |  11 | Atlanta, GA
 1004 | Bill Turner    | F      |   2 | Auburn, AL
 1005 | Joss Butler    | M      |  59 | Birmingham, AL
 1006 | Roman Roland   | M      |  33 | Birmingham, AL
 1007 | Michael Jordan | F      |  20 | Selma, AL
 1008 | Roger Pensore  | M      |  11 | Homewood, AL
 1009 | Robert Hook    | M      |   3 | Atlanta, GA
 1010 | Tom Button     | F      |  10 | Auburn, AL
 1011 | John Paul      | M      |  85 | Birmingham, AL
 1012 | Paul Newman    | M      |  45 | Montgomery, AL
 1013 | Robert Brown   | F      |  47 | Birmingham, AL
 1014 | Agatha Brown   | U      |  37 | Birmingham, AL
(15 rows)

joyroy=>
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat....

```
joyroy=>
joyroy=>
joyroy=> select * from patient;
 mrn  |       name       | gender | age |     address
------+------------------+--------+-----+-----------------
 1000 | Rob Robinson     | M      |  12 | Birmingham, AL
 1001 | Bill Balmer      | F      |  90 | Pelham, AL
 1002 | John Smith       | M      |  32 | Homewood, AL
 1003 | Mark Butcher     | M      |  11 | Atlanta, GA
 1004 | Bill Turner      | F      |   2 | Auburn, AL
 1005 | Joss Butler      | M      |  59 | Birmingham, AL
 1006 | Roman Roland     | M      |  33 | Birmingham, AL
 1007 | Michael Jordan   | F      |  20 | Selma, AL
 1008 | Roger Pensore    | M      |  11 | Homewood, AL
 1009 | Robert Hook      | M      |   3 | Atlanta, GA
 1010 | Tom Button       | F      |  10 | Auburn, AL
 1011 | John Paul        | M      |  85 | Birmingham, AL
 1012 | Paul Newman      | M      |  45 | Montgomery, AL
 1013 | Robert Brown     | F      |  47 | Birmingham, AL
 1014 | Agatha Brown     | U      |  37 | Birmingham, AL
(15 rows)

joyroy=>
```

v

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat....

```
joyroy=>
joyroy=>
joyroy=> select * from patient;
 mrn  |      name      | gender | age |    address
------+----------------+--------+-----+----------------
 1000 | Rob Robinson   | M      |  12 | Birmingham, AL
 1001 | Bill Balmer    | F      |  90 | Pelham, AL
 1002 | John Smith     | M      |  32 | Homewood, AL
 1003 | Mark Butcher   | M      |  11 | Atlanta, GA
 1004 | Bill Turner    | F      |   2 | Auburn, AL
 1005 | Joss Butler    | M      |  59 | Birmingham, AL
 1006 | Roman Roland   | M      |  33 | Birmingham, AL
 1007 | Michael Jordan | F      |  20 | Selma, AL
 1008 | Roger Pensore  | M      |  11 | Homewood, AL
 1009 | Robert Hook    | M      |   3 | Atlanta, GA
 1010 | Tom Button     | F      |  10 | Auburn, AL
 1011 | John Paul      | M      |  85 | Birmingham, AL
 1012 | Paul Newman    | M      |  45 | Montgomery, AL
 1013 | Robert Brown   | F      |  47 | Birmingham, AL
 1014 | Agatha Brown   | U      |  37 | Birmingham, AL
(15 rows)

joyroy=>
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu —...

```
 1014 | Agatha Brown   | U      |  37 | Birmingham, AL
(15 rows)

joyroy=> select * from prescription;
 prescriptionid | empid | mrn  | prescribe_date | total_cost
----------------+-------+------+----------------+------------
              5 |     4 | 1009 | 2021-10-09     |       5.00
              6 |     1 | 1000 | 2021-01-10     |     100.00
              7 |     1 | 1001 | 2021-03-09     |     500.00
              8 |     2 | 1004 | 2021-01-09     |       0.00
              9 |     3 | 1005 | 2021-01-10     |       0.00
              3 |     2 | 1004 | 2021-01-10     |      15.00
              4 |     3 | 1005 | 2021-11-09     |     110.00
              1 |     1 | 1000 | 2021-11-01     |
              2 |     1 | 1001 | 2021-01-09     |    6010.00
             10 |     4 | 1009 | 2021-02-11     |       5.00
(10 rows)

joyroy=>
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu —...

```
             10 |     4 | 1009 | 2021-02-11     |       5.00
(10 rows)

joyroy=> select * from medicine;
     name      |  manufacturer   |  cost
---------------+-----------------+---------
 Humira        | Abvie           | 5000.00
 Humira        | Biosimilar      | 1000.00
 Mobic         | Square          |   10.00
 Platonix      | Beximco         |   10.00
 Tylenol       | CVS             |    5.00
 Ziagen        | Beximco         |  100.00
 Remdesivir    | Gilead Sciences |  500.00
 Covid Vaccine | Pfizer          |    0.00
 Covid Vaccine | Moderna         |    0.00
 Covid Vaccine | J&J             |    0.00
(10 rows)

joyroy=>
```

DELIVERABLE D - Views

1. # of patients that have been prescribed at least one Drug by manufacturer within the past 90 days:

Create view qtr_report_manufacturer as

Select presc_med.manufacturer, count(presc.mrn)

From prescriptionmedicine presc_med

Join prescription presc on presc.prescriptionid = presc_med.prescriptionid

Where presc.prescribe_date between current_date-90 and current_date

Group by presc_med.manufacturer;

```
joyroy=> Create view qtr_report_manufacturer as
joyroy-> Select presc_med.manufacturer, count(presc.mrn)
joyroy-> From prescriptionmedicine presc_med
joyroy-> Join prescription presc on presc.prescriptionid = presc_med.prescriptionid
joyroy-> Where presc.prescribe_date between   current_date-90 and current_date
joyroy-> Group by presc_med.manufacturer;
CREATE VIEW
joyroy=> select * from qtr_report_manufacturer;
 manufacturer | count
--------------+-------
 Beximco      |     2
 CVS          |     1
(2 rows)
```

2. patients with nonnormal xrays who were given a prescription:

Create view xray_patients as Select p.mrn, p.name, t.test_name, t.test_result from patient p

Join appointment appt on appt.mrn = p.mrn

Join test t on t.apptid = appt.apptid

Where t.test_name = 'X-ray' and t.test_result <> 'Normal'

and p.mrn in (select mrn from prescription);

```
joyroy=> Create view xray_patients as Select p.mrn, p.name, t.test_name, t.test_result  from patient p
joyroy-> Join appointment appt on appt.mrn = p.mrn
joyroy-> Join test t on t.apptid = appt.apptid
joyroy-> Where t.test_name = 'X-ray' and t.test_result <> 'Normal'
joyroy-> and p.mrn in (select mrn from prescription);
CREATE VIEW
joyroy=> select * from xray_patients;
 mrn  |    name     | test_name |  test_result
------+-------------+-----------+---------------
 1004 | Bill Turner | X-ray     | Wear and Tear
 1001 | Bill Balmer | X-ray     | Discloation
(2 rows)
```

3. pulls the full diagnosis view. Shows the patient age, primary doc, diagnosis doc, and diagnosis information.
   Create or replace view mrn_diagnosis as

Select p.mrn, p.name, emp.name as diagnosis_doc, doc.speciality, d.diagnosis_date, d.diagnosis_name,d.dx_desc
From patient p
Join appointment appt on appt.mrn = p.mrn
Join diagnosis d on appt.apptid = d.apptid
Join doctor doc on doc.empid = appt.empid
Join employee emp on emp.empid = appt.empid ;

```
joyroy=> Create or replace view mrn_diagnosis as
joyroy-> Select p.mrn, p.name, emp.name as diagnosis_doc, doc.speciality, d.diagnosis_date, d.diagnosis_name,d.dx_desc
joyroy-> From patient p
joyroy-> Join appointment appt on appt.mrn = p.mrn
joyroy-> Join diagnosis d on appt.apptid = d.apptid
joyroy-> Join doctor doc on doc.empid = appt.empid
joyroy-> Join employee emp on  emp.empid = appt.empid ;
CREATE VIEW
joyroy=> select * from mrn_diagnosis;
 mrn  |     name      | diagnosis_doc | speciality | diagnosis_date |  diagnosis_name  |         dx_desc
------+---------------+---------------+------------+----------------+------------------+------------------------
 1000 | Rob Robinson  | John Doe      | head       | 2021-01-09     | COVID            | asympomatic
 1001 | Bill Balmer   | John Doe      | head       | 2021-01-09     | Flu              | Seasonal Flu
 1004 | Bill Turner   | Jane Doe      | liver      | 2021-01-09     | AIDS             | Under control
 1005 | Joss Butler   | Robert Brown  | teeth      | 2021-01-09     | Fever            | 104 degrees
 1008 | Roger Pensore | Jane Doe      | liver      | 2021-01-09     | Pain             | pain in joints
 1009 | Robert Hook   | Julia Brown   | AS         | 2021-01-09     | Headache         | Severe headache
 1000 | Rob Robinson  | John Doe      | head       | 2021-01-09     | Pain             | Pain in whole body
 1001 | Bill Balmer   | John Doe      | head       | 2021-01-09     | Amnesia          | extreme low blood count
 1004 | Bill Turner   | Jane Doe      | liver      | 2021-01-09     | Loss of appetite |
 1005 | Joss Butler   | Robert Brown  | teeth      | 2021-01-09     | Weekness         |
 1008 | Roger Pensore | Jane Doe      | liver      | 2021-01-09     | Fatigue          |
 1009 | Robert Hook   | Julia Brown   | AS         | 2021-01-09     | Sleepiness       | Sleeps 12 hours a day
 1000 | Rob Robinson  | John Doe      | head       | 2021-01-09     | Weight loss      | BMI < 20
(13 rows)
```

4. # of prescriptions given to patients by age

```
Create view prescriptions_by_age as
Select result.age_brackets, Count(Presc.prescriptionid)
from (Select  p.mrn,

Case when p.age between 0 and 10 then '0s'
when p.age between 10 and 20 then '10s'
when p.age between 20 and 30 then '20s'
when p.age between 30 and 40 then '30s'
when p.age between 40 and 50 then '40s'
when p.age between 50 and 60 then '50s'
when p.age between 60 and 70 then '60s'
when p.age >= 70 then '70s+'
Else null end as age_brackets
        From patient p) result

Join prescription presc on presc.mrn = result.mrn
Group by result.age_brackets;
```

```
joyroy=> Create view prescriptions_by_age as
joyroy-> Select result.age_brackets, Count(Presc.prescriptionid)
joyroy-> from (Select  p.mrn,
joyroy(>
joyroy(> Case when p.age between 0 and 10 then '0s'
joyroy(> when p.age between 10 and 20 then '10s'
joyroy(> when p.age between 20 and 30 then '20s'
joyroy(> when p.age between 30 and 40 then '30s'
joyroy(> when p.age between 40 and 50 then '40s'
joyroy(> when p.age between 50 and 60 then '50s'
joyroy(> when p.age between 60 and 70 then '60s'
joyroy(> when p.age >= 70 then '70s+'
joyroy(> Else null end as age_brackets
joyroy(> From patient p) result
joyroy->
joyroy-> Join prescription presc on presc.mrn = result.mrn
joyroy-> Group by result.age_brackets;
CREATE VIEW
joyroy=> select * from prescriptions_by_age;
 age_brackets | count
--------------+-------
 10s          |     2
 50s          |     2
 70s+         |     2
 0s           |     4
(4 rows)
```

5. # of doctors in each department

```
Create view doc_count as
Select d.dept_name , count(e.empid) as doc_count from department d

Join employee e on e.deptid = d.deptid

Inner join doctor doc on doc.empid = e.empid

Group by d.dept_name;
```

```
joyroy=> Create view doc_count as
joyroy-> Select d.dept_name , count(e.empid) as doc_count from department d
joyroy->
joyroy-> Join employee e on e.deptid = d.deptid
joyroy->
joyroy-> Inner join doctor doc on doc.empid = e.empid
joyroy->
joyroy-> Group by d.dept_name;
CREATE VIEW
joyroy=> select * from doc_count;
    dept_name     | doc_count
------------------+-----------
 Medicine         |         3
 Gastroenterology |         3
 Rheumatology     |         3
 Physical Therapy |         2
 Neurology        |         2
(5 rows)
```

6. Total cost for each patient

```
Create view Total_cost_patient as
Select result.name,
case when sum(result.total_cost) is null then 0.00
else sum(result.total_cost) end as total_cost
From (Select p.name, sum(presc.total_cost) as total_cost from prescription presc
        Join patient p on p.mrn=presc.mrn
        Group by p.name
        union
        Select p.name, sum(appt.total_cost) as total_cost from appointment appt
        Join patient p on p.mrn=appt.mrn
```

Group by p.name) result

Group by result.name;

```
joyroy=> Create view Total_cost_patient as
joyroy-> Select result.name,
joyroy-> case when sum(result.total_cost) is null then 0.00
joyroy-> else sum(result.total_cost) end as total_cost
joyroy->
joyroy-> From (Select p.name, sum(presc.total_cost) as total_cost from prescription presc
joyroy(> Join patient p on p.mrn=presc.mrn
joyroy(> Group by p.name
joyroy(> union
joyroy(> Select p.name, sum(appt.total_cost) as total_cost from appointment appt
joyroy(> Join patient p on p.mrn=appt.mrn
joyroy(> Group by p.name) result
joyroy->
joyroy-> Group by result.name;
CREATE VIEW
joyroy=> select * from total_cost_patient;
      name      | total_cost
----------------+------------
 Michael Jordan |       0.00
 Rob Robinson   |    1600.00
 Tom Button     |       0.00
 Robert Hook    |     760.00
 Roger Pensore  |    1350.00
 Paul Newman    |       0.00
 Joss Butler    |    1610.00
 John Smith     |       0.00
 Bill Balmer    |    8010.00
 Robert Brown   |       0.00
 Roman Roland   |       0.00
 Agatha Brown   |       0.00
 Bill Turner    |    1515.00
 John Paul      |       0.00
 Mark Butcher   |       0.00
(15 rows)
```

DELIVERABLE E - Indexes

Create Index manu on prescriptionmedicine(manufacturer);

The index on manufacturer helps with the group by clause on the qtr report manufacturer view

```
joyroy=> Create Index manu on prescriptionmedicine(manufacturer);
CREATE INDEX
```

create index test_results on test(test_name,test_result);

The test_results index helps with the xray patients view; t.test_name = 'X-ray' and t.test_result <> 'Normal' filter look up.

```
joyroy=> create index test_results on test(test_name,test_result);
CREATE INDEX
```

Create index age on patient(age);

This index helps with bucketing of different age brackets in the calculated field age_brackets in the prescriptions by age view.

```
joyroy=> Create index age on patient(age);
CREATE INDEX
```

DELIVERABLE F - Constraints

Employee constraints:

CHECK(gender in('M', 'F', 'U')) -- gender would be either Male, Female or Unknown
CHECK(age >= 25 AND age <= 100) – employee age between 25 and 100, inclusive

Patient constraints:

CHECK(gender in('M', 'F', 'U')) -- gender would be either Male, Female or Unknown

Appointment constraints:

CHECK(appt_type in('in-person', 'virtual')) -- appointment type either in-person or virtual
CHECK(appt_status in('Scheduled', 'Arrived', 'Cancelled', 'No-show')) – appointment status cannot be other than these four types

DELIVERABLE G - Triggers

Trigger 1: Auto_Reschedule_Appointment

If the appt status is updated to cancelled, it automatically changes the appointment to a date 7 days out, and the appointment status back to scheduled.

```
joyroy=> CREATE OR REPLACE FUNCTION auto_reschedule_appt () RETURNS TRIGGER AS
$auto_reschedule_appt$
BEGIN
IF new.appt_status = 'Cancelled' THEN

Update appointment set appt_date_time =
(select

 (select appt.appt_date_time from appointment appt where appt.apptid = new.apptid)

+ interval '7 days'),
Appt_status = 'Scheduled'
where appointment.apptid = new.apptid;

END IF;
RETURN NULL;
END;
$auto_reschedule_appt$ LANGUAGE plpgsql;
CREATE FUNCTION
joyroy=> CREATE TRIGGER auto_reschedule
joyroy-> AFTER UPDATE ON Appointment
joyroy-> FOR EACH ROW EXECUTE PROCEDURE auto_reschedule_appt();
CREATE TRIGGER
joyroy=>
```

```
joyroy=> select * from appointment where apptid = 31;
 apptid | empid | mrn  |   appt_date_time    | appt_type | appt_cost | total_cost | appt_status
--------+-------+------+---------------------+-----------+-----------+------------+-------------
     31 |     1 | 1000 | 2021-12-07 15:00:00 | virtual   |     30.00 |            | Scheduled
(1 row)

joyroy=> update appointment set appt_status='Cancelled' where apptid = 31;
UPDATE 1
joyroy=> select * from appointment where apptid = 31;
 apptid | empid | mrn  |   appt_date_time    | appt_type | appt_cost | total_cost | appt_status
--------+-------+------+---------------------+-----------+-----------+------------+-------------
     31 |     1 | 1000 | 2021-12-14 15:00:00 | virtual   |     30.00 |            | Scheduled
(1 row)
```

Trigger 2: Total_amount_appt

This trigger sums up the costs of the appointment including the tests.

```
joyroy=>
joyroy=> CREATE OR REPLACE FUNCTION Total_amount_appt() RETURNS TRIGGER AS $Total_amount_appt$

joyroy$> Begin
joyroy$> IF(TG_OP = 'INSERT')   THEN
joyroy$>     update appointment set total_cost =
joyroy$> (select sum(result.cost) as total_cost
joyroy$> from (select appt.apptid, testcost.test_cost as cost from test
joyroy$> join appointment appt on appt.apptid = test.apptid
joyroy$> join testcost on test.test_name = testcost.test_name
joyroy$> where appt.apptid = new.apptid) result
joyroy$> group by result.apptid
joyroy$> having result.apptid = new.apptid)
joyroy$> where apptid = new.apptid;
joyroy$> ELSIF(TG_OP = 'DELETE')   THEN
joyroy$>
joyroy$> update appointment set total_cost =
joyroy$> (select sum(result.cost) as total_cost
joyroy$> from (select appt.apptid, testcost.test_cost as cost from test
joyroy$> join appointment appt on appt.apptid = test.apptid
joyroy$> join testcost on test.test_name = testcost.test_name
joyroy$> where appt.apptid = old.apptid) result
joyroy$> group by result.apptid
joyroy$> having result.apptid = old.apptid)
joyroy$> where apptid = old.apptid;
joyroy$> ELSIF(TG_OP = 'UPDATE')   THEN
joyroy$>
joyroy$> update appointment set total_cost =
joyroy$> (select sum(result.cost) as total_cost
joyroy$> from (select appt.apptid, testcost.test_cost as cost from test
joyroy$> join appointment appt on appt.apptid = test.apptid
joyroy$> join testcost on test.test_name = testcost.test_name
joyroy$> where appt.apptid = new.apptid) result
joyroy$> group by result.apptid
joyroy$> having result.apptid = new.apptid)
joyroy$> where apptid = new.apptid;
joyroy$> end if;
joyroy$> return null;
joyroy$> end;
joyroy$> $Total_amount_appt$ LANGUAGE plpgsql;
CREATE FUNCTION
```

CREATE TRIGGER appt_total_amount
AFTER INSERT OR DELETE OR UPDATE ON Test
FOR EACH ROW EXECUTE PROCEDURE Total_amount_appt();

```
joyroy=> select * from appointment where appt_status='Arrived';
 apptid | empid | mrn  |   appt_date_time    | appt_type | appt_cost | total_cost | appt_status
--------+-------+------+---------------------+-----------+-----------+------------+-------------
      1 |     1 | 1000 | 2021-01-08 08:05:00 | in-person |     30.00 |    1500.00 | Arrived
      2 |     1 | 1001 | 2021-01-08 09:00:00 | in-person |     30.00 |    1500.00 | Arrived
      5 |     2 | 1004 | 2021-01-08 12:05:00 | virtual   |     30.00 |    1500.00 | Arrived
      6 |     3 | 1005 | 2021-01-08 13:05:00 | in-person |     30.00 |    1500.00 | Arrived
      9 |     2 | 1008 | 2021-01-09 08:05:00 | virtual   |     30.00 |    1500.00 | Arrived
     10 |     4 | 1009 | 2021-01-08 09:05:00 | in-person |     30.00 |    1350.00 | Arrived
(6 rows)

joyroy=> delete from test where testid=29;
DELETE 1
joyroy=> insert into Test values(31, 10, 'CT Scan', 'Normal');
INSERT 0 1
joyroy=> update Test set test_name='Drug Test' where testid=18;
UPDATE 1
joyroy=> select * from appointment where appt_status='Arrived';
 apptid | empid | mrn  |   appt_date_time    | appt_type | appt_cost | total_cost | appt_status
--------+-------+------+---------------------+-----------+-----------+------------+-------------
      1 |     1 | 1000 | 2021-01-08 08:05:00 | in-person |     30.00 |    1500.00 | Arrived
      2 |     1 | 1001 | 2021-01-08 09:00:00 | in-person |     30.00 |    1500.00 | Arrived
      5 |     2 | 1004 | 2021-01-08 12:05:00 | virtual   |     30.00 |    1500.00 | Arrived
      6 |     3 | 1005 | 2021-01-08 13:05:00 | in-person |     30.00 |    1500.00 | Arrived
      9 |     2 | 1008 | 2021-01-09 08:05:00 | virtual   |     30.00 |    1350.00 | Arrived
     10 |     4 | 1009 | 2021-01-08 09:05:00 | in-person |     30.00 |     750.00 | Arrived
(6 rows)
```

Trigger 3: Total_amount_presciption

Adds up the cost of the medicine by each prescription and adds to total cost field in prescription.

```
joyroy=> CREATE OR REPLACE FUNCTION Total_amount_prescription() RETURNS TRIGGER AS $Total_amount_prescription$

joyroy$> Begin
joyroy$> IF(TG_OP = 'INSERT') THEN
joyroy$>     update prescription set total_cost =
joyroy$> (select sum(result.cost) as total_cost
joyroy$> from (select presc.prescriptionid, medicine.cost from prescriptionmedicine pres_med
joyroy$> join prescription presc on pres_med.prescriptionid = presc.prescriptionid
joyroy$> join medicine on medicine.name  = pres_med.name   and medicine.manufacturer = pres_med.manufacturer
joyroy$> where presc.prescriptionid = new.prescriptionid) result
joyroy$> group by result.prescriptionid
joyroy$> having result.prescriptionid = new.prescriptionid)
joyroy$> where prescriptionid = new.prescriptionid;
joyroy$> ELSIF(TG_OP = 'DELETE')   THEN
joyroy$> update prescription set total_cost =
joyroy$> (select sum(result.cost) as total_cost
joyroy$> from (select presc.prescriptionid, medicine.cost from prescriptionmedicine pres_med
joyroy$> join prescription presc on pres_med.prescriptionid = presc.prescriptionid
joyroy$> join medicine on medicine.name  = pres_med.name   and medicine.manufacturer = pres_med.manufacturer
joyroy$> where presc.prescriptionid = old.prescriptionid) result
joyroy$> group by result.prescriptionid
joyroy$> having result.prescriptionid = old.prescriptionid)
joyroy$> where prescriptionid = old.prescriptionid;
joyroy$> ELSIF(TG_OP = 'UPDATE')   THEN
joyroy$> update prescription set total_cost =
joyroy$> (select sum(result.cost) as total_cost
joyroy$> from (select presc.prescriptionid, medicine.cost from prescriptionmedicine pres_med
joyroy$> join prescription presc on pres_med.prescriptionid = presc.prescriptionid
joyroy$> join medicine on medicine.name  = pres_med.name   and medicine.manufacturer = pres_med.manufacturer
joyroy$> where presc.prescriptionid = new.prescriptionid) result
joyroy$> group by result.prescriptionid
joyroy$> having result.prescriptionid = new.prescriptionid)
joyroy$> where prescriptionid = new.prescriptionid;
joyroy$> end if;
joyroy$> return null;
joyroy$> end;
joyroy$> $Total_amount_prescription$ LANGUAGE plpgsql;
```

CREATE TRIGGER amount_update
AFTER INSERT OR DELETE OR UPDATE ON prescriptionmedicine
FOR EACH ROW EXECUTE PROCEDURE Total_amount_prescription();

```
joyroy=> select * from prescription;
 prescriptionid | empid | mrn  | prescribe_date | total_cost
----------------+-------+------+----------------+------------
              5 |     4 | 1009 | 2021-10-09     |       5.00
              6 |     1 | 1000 | 2021-01-10     |     100.00
              7 |     1 | 1001 | 2021-03-09     |     500.00
              8 |     2 | 1004 | 2021-01-09     |       0.00
              9 |     3 | 1005 | 2021-01-10     |       0.00
             10 |     4 | 1009 | 2021-02-11     |       0.00
              1 |     1 | 1000 | 2021-11-01     |    5010.00
              2 |     1 | 1001 | 2021-01-09     |    1010.00
              3 |     2 | 1004 | 2021-01-10     |      15.00
              4 |     3 | 1005 | 2021-11-09     |     110.00
(10 rows)

joyroy=> delete from prescriptionmedicine where prescriptionid =1;
DELETE 2
joyroy=> insert into prescriptionmedicine values(2,'Humira','Abvie');
INSERT 0 1
joyroy=> update prescriptionmedicine set name = 'Tylenol',manufacturer = 'CVS'
joyroy-> where prescriptionid = 10;
UPDATE 1
joyroy=> select * from prescription;
 prescriptionid | empid | mrn  | prescribe_date | total_cost
----------------+-------+------+----------------+------------
              5 |     4 | 1009 | 2021-10-09     |       5.00
              6 |     1 | 1000 | 2021-01-10     |     100.00
              7 |     1 | 1001 | 2021-03-09     |     500.00
              8 |     2 | 1004 | 2021-01-09     |       0.00
              9 |     3 | 1005 | 2021-01-10     |       0.00
              3 |     2 | 1004 | 2021-01-10     |      15.00
              4 |     3 | 1005 | 2021-11-09     |     110.00
              1 |     1 | 1000 | 2021-11-01     |
              2 |     1 | 1001 | 2021-01-09     |    6010.00
             10 |     4 | 1009 | 2021-02-11     |       5.00
(10 rows)
```

DELIVERABLE H

Created a function to change all appointments with a status of Scheduled and a past appointment date to 'No-Show'. The version of Postgresql in the server does not support stored procedures so we made a function. Stored Procedures are for Postgresql version 11 or greater.

```
                                                version
-----------------------------------------------------------------------------------------------------------
 PostgreSQL 9.6.3 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-11), 64-bit
(1 row)

joyroy=>
```

```
create or replace function noshows()
RETURNS table (apptid int, empid int, mrn int, appt_date_time timestamp, appt_type varchar,
cost decimal, total_cost decimal, appt_status varchar) AS $$
BEGIN
UPDATE appointment
        SET appt_status = 'No-show'
        WHERE appointment.appt_date_time < localtimestamp
        AND appointment.appt_status = 'Scheduled';
        RETURN QUERY (select appointment.* from appointment);
END;$$
LANGUAGE PLPGSQL;

Select * from noshows();
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu —...

```
CREATE          INSERT          ROLLBACK        VALUES
DEALLOCATE      LISTEN          SAVEPOINT       WITH
joyroy=> create or replace function noshows()
joyroy-> RETURNS table (apptid int, empid int, mrn int, appt_date_time times
tamp, appt_type varchar, cost decimal, appt_status varchar, total_cost decim
al) AS $$
joyroy$> BEGIN
joyroy$> UPDATE appointment
joyroy$> SET appt_status = 'No-show'
joyroy$> WHERE appointment.appt_date_time < localtimestamp
joyroy$> AND appointment.appt_status = 'Scheduled';
joyroy$> RETURN QUERY (select appointment.* from appointment);
joyroy$> END;$$
joyroy-> LANGUAGE PLPGSQL;
CREATE FUNCTION
joyroy=>
```

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu — 106×28

apptid	empid	mrn	appt_date_time	appt_type	appt_cost	total_cost	appt_status
3	1	1002	2021-01-08 10:05:00	virtual	30.00		Cancelled
4	1	1003	2021-01-08 11:05:00	virtual	30.00		No-show
7	1	1006	2021-01-08 14:05:00	in-person	30.00		Cancelled
8	3	1007	2021-01-08 15:05:00	virtual	30.00		No-show
11	1	1010	2021-01-08 10:05:00	in-person	30.00		Cancelled
12	12	1011	2021-01-08 11:05:00	virtual	30.00		No-show
13	12	1009	2021-12-03 13:30:00	virtual	30.00		Scheduled
14	2	1010	2021-12-03 14:30:00	in-person	30.00		Scheduled
15	3	1011	2021-01-08 09:05:00	virtual	30.00		Scheduled
16	1	1012	2021-12-04 08:00:00	virtual	30.00		Scheduled
17	1	1013	2021-12-04 09:00:00	in-person	30.00		Scheduled
18	1	1014	2021-12-04 10:00:00	in-person	30.00		Scheduled
19	1	1005	2021-12-04 11:00:00	virtual	30.00		Scheduled
20	2	1002	2021-12-04 12:00:00	virtual	30.00		Scheduled
21	3	1001	2021-12-04 13:00:00	in-person	30.00		Scheduled
22	1	1001	2021-12-04 14:00:00	in-person	30.00		Scheduled
23	3	1001	2021-12-04 15:00:00	virtual	30.00		Scheduled
24	2	1006	2021-12-04 16:00:00	in-person	30.00		Scheduled
25	4	1007	2021-12-04 17:00:00	virtual	30.00		Scheduled
26	1	1008	2021-12-05 10:00:00	in-person	30.00		Scheduled
27	12	1009	2021-12-06 11:00:00	virtual	30.00		Scheduled
28	7	1010	2021-12-07 12:00:00	virtual	30.00		Scheduled
29	7	1011	2021-12-07 13:00:00	in-person	30.00		Scheduled
30	15	1012	2021-12-07 14:00:00	in-person	30.00		Scheduled
32	1	1000	2021-12-07 16:00:00	in-person	30.00		Scheduled
1	1	1000	2021-01-08 08:05:00	in-person	30.00	1500.00	Arrived

ssh abowen13@moat.cs.uab.edu — -ssh abowen13@moat.cs.uab.edu — 106×43

```
CREATE FUNCTION
joyroy=> Select * from noshows();
 apptid | empid | mrn  |    appt_date_time    | appt_type | cost  | total_cost | appt_status
--------+-------+------+----------------------+-----------+-------+------------+-------------
      3 |     1 | 1002 | 2021-01-08 10:05:00 | virtual   | 30.00 |            | Cancelled
      4 |     1 | 1003 | 2021-01-08 11:05:00 | virtual   | 30.00 |            | No-show
      7 |     1 | 1006 | 2021-01-08 14:05:00 | in-person | 30.00 |            | Cancelled
      8 |     3 | 1007 | 2021-01-08 15:05:00 | virtual   | 30.00 |            | No-show
     11 |     1 | 1010 | 2021-01-08 10:05:00 | in-person | 30.00 |            | Cancelled
     12 |    12 | 1011 | 2021-01-08 11:05:00 | virtual   | 30.00 |            | No-show
     16 |     1 | 1012 | 2021-12-04 08:00:00 | virtual   | 30.00 |            | Scheduled
     17 |     1 | 1013 | 2021-12-04 09:00:00 | in-person | 30.00 |            | Scheduled
     18 |     1 | 1014 | 2021-12-04 10:00:00 | in-person | 30.00 |            | Scheduled
     19 |     1 | 1005 | 2021-12-04 11:00:00 | virtual   | 30.00 |            | Scheduled
     20 |     2 | 1002 | 2021-12-04 12:00:00 | virtual   | 30.00 |            | Scheduled
     21 |     3 | 1001 | 2021-12-04 13:00:00 | in-person | 30.00 |            | Scheduled
     22 |     1 | 1001 | 2021-12-04 14:00:00 | in-person | 30.00 |            | Scheduled
     23 |     3 | 1001 | 2021-12-04 15:00:00 | virtual   | 30.00 |            | Scheduled
     24 |     2 | 1006 | 2021-12-04 16:00:00 | in-person | 30.00 |            | Scheduled
     25 |     4 | 1007 | 2021-12-04 17:00:00 | virtual   | 30.00 |            | Scheduled
     26 |     1 | 1008 | 2021-12-05 10:00:00 | in-person | 30.00 |            | Scheduled
     27 |    12 | 1009 | 2021-12-06 11:00:00 | virtual   | 30.00 |            | Scheduled
     28 |     7 | 1010 | 2021-12-07 12:00:00 | virtual   | 30.00 |            | Scheduled
     29 |     7 | 1011 | 2021-12-07 13:00:00 | in-person | 30.00 |            | Scheduled
     30 |    15 | 1012 | 2021-12-07 14:00:00 | in-person | 30.00 |            | Scheduled
     32 |     1 | 1000 | 2021-12-07 16:00:00 | in-person | 30.00 |            | Scheduled
      1 |     1 | 1000 | 2021-01-08 08:05:00 | in-person | 30.00 |    1500.00 | Arrived
      2 |     1 | 1001 | 2021-01-08 09:00:00 | in-person | 30.00 |    1500.00 | Arrived
      5 |     2 | 1004 | 2021-01-08 12:05:00 | virtual   | 30.00 |    1500.00 | Arrived
      6 |     3 | 1005 | 2021-01-08 13:05:00 | in-person | 30.00 |    1500.00 | Arrived
      9 |     2 | 1008 | 2021-01-09 08:05:00 | virtual   | 30.00 |    1350.00 | Arrived
     10 |     4 | 1009 | 2021-01-08 09:05:00 | in-person | 30.00 |     750.00 | Arrived
     31 |     1 | 1000 | 2021-12-14 15:00:00 | virtual   | 30.00 |            | Scheduled
     13 |    12 | 1009 | 2021-12-03 13:30:00 | virtual   | 30.00 |            | No-show
     14 |     2 | 1010 | 2021-12-03 14:30:00 | in-person | 30.00 |            | No-show
     15 |     3 | 1011 | 2021-01-08 09:05:00 | virtual   | 30.00 |            | No-show
(32 rows)

joyroy=>
joyroy=>
joyroy=>
joyroy=>
joyroy=>
```