Table of Contents

Assignment 1: Application Description

DBMS for Hospitals - An efficient system where data between patients, nurses, and doctors need to be simple, efficient, and fast is vital for the hospital systems to run effectively. This software will deal with multiple relations within a complex hospital system. It will deal with a collection of the patient's information, diagnosis details, transactions, etc. Traditionally this has been done and organized manually by the doctors and nurses. But this system will allow for a more efficient approach where everything will be organized by a DBMS.

The management system includes the registration of patients, storing their personal information into the system, the hospital's working doctors, room numbers, computerized billings and diagnosis and many more mentioned in the chart below.

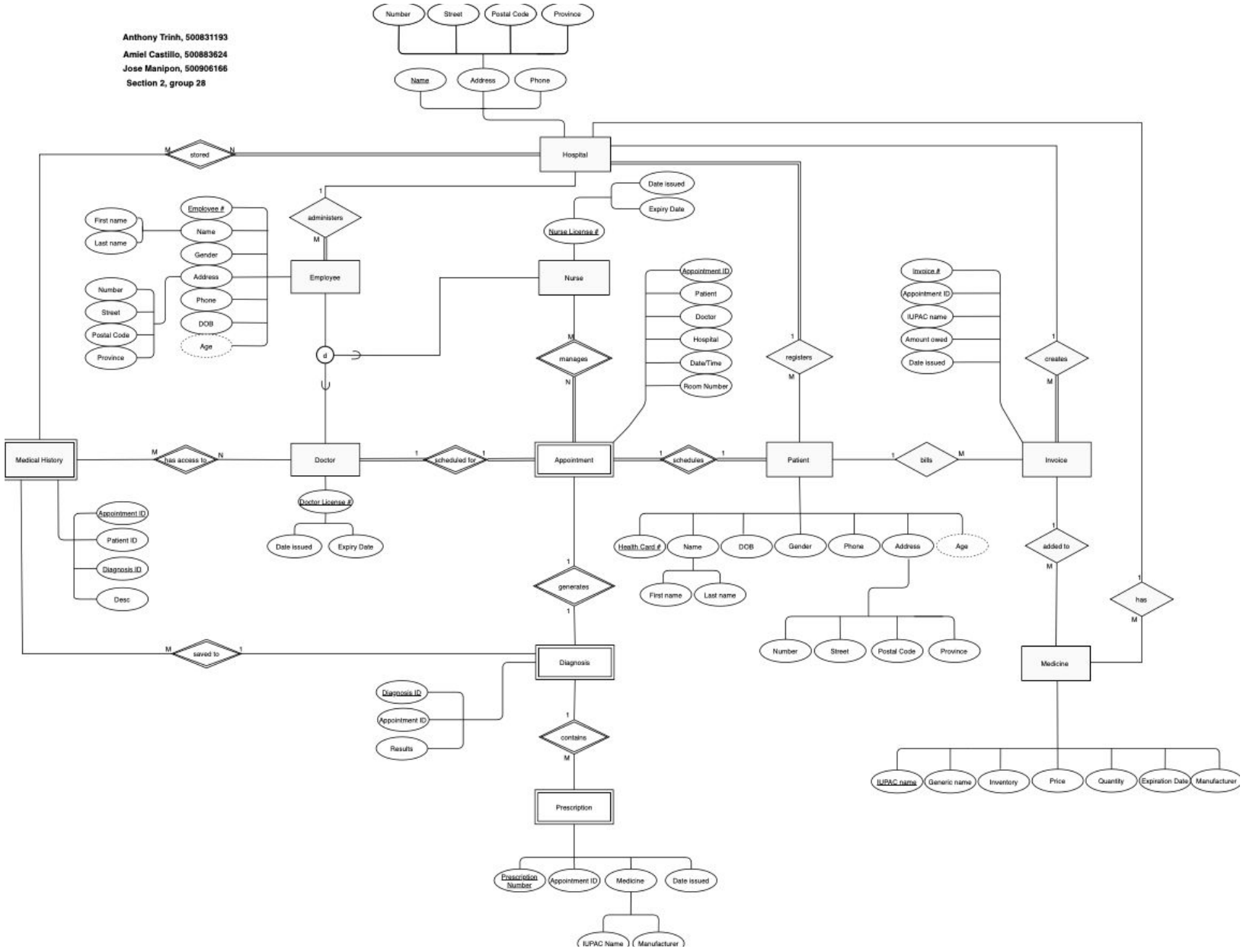
We will include 9 rows in our entity table:

Entity	Attributes
Patient	Health card #, Name, DOB, Gender, Phone, Address,Status
Doctor	Name, DOB, Gender, Phone, Address, Postal Code, Specialization
Medicine	Inventory, Name, Price, Expiration Date, Manufacturer
Medical History	**Patient**, **Hospital**, **Doctor,** Date of last visit, Past Diagnosis
Appointment	**Patient**, **Doctor**, **Type of Appointment** Time of Appointment, Date of Appointment, Location
Diagnosis	**Patient**, **Doctor**, Medicine, Diagnosis ID
Hospital	**Appointment**, Room Number
Prescription	**Medicine**, **Diagnosis,** Quantity
Transactions	**Appointment**, Payment method, Cost Amount, Paid Date, Due Date

More specifically the Hospital will conduct Appointments where the Patient will receive a Doctor. The Doctor will receive the Patient’s Medical History, and conduct the Type of Appointment. The Doctor will then give the Patient a Diagnosis, and, in some cases, a Prescription. The Patient will receive Medicine based on the Prescription, and the

Patient will conduct a Transaction with the Hospital for the Appointment and/or Medicine.

Assignment 2: ER Model



Assignment 3: Schema Design

schema.sql *Creation of tables*


```
-- create entity tables
CREATE TABLE hospital (
    hospital_name VARCHAR2(32) NOT NULL,
    address_street VARCHAR2(32) NOT NULL,
    address_street2 VARCHAR2(32),
    address_city VARCHAR2(32) NOT NULL,
    address_province VARCHAR2(2) NOT NULL,
    address_postalcode VARCHAR2(6) NOT NULL,
    address_country VARCHAR2(3) DEFAULT 'CAD',
    phone VARCHAR2(16) NOT NULL,
```

```
CREATE TABLE doctor (
    doctor_id INT NOT NULL,
    doctorlicense_expiry DATE NOT NULL,
    employee_id INT NOT NULL,
    PRIMARY KEY(doctor_id),
    FOREIGN KEY(employee_id) REFERENCES employee(employee_id)
);
```

Assignment 4: Demo of Designing Views/Simple Queries

```
-- insertion of data
-- hospitals
INSERT INTO hospital (hospital_id, hospital_name,address_street,address_city,address_province,address_postalcode,phone)
    VALUES (123, 'Toronto General Hospital', '200 Elizabeth St', 'Toronto', 'ON', 'M5G2C4','4167895297');
INSERT INTO hospital (hospital_id, hospital_name,address_street,address_city,address_province,address_postalcode,phone)
    VALUES (124, 'Michael Garron Hospital', '825 Coxwell Ave', 'East York', 'ON', 'M4C3E7','4167893127');
INSERT INTO hospital (hospital_id, hospital_name,address_street,address_city,address_province,address_postalcode,phone)
    VALUES (125, 'Toronto Western Hospital', '399 Bathurst St', 'Toronto', 'ON', 'M5T2S8','4167898713');
-- employees
INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
    VALUES (323952, 'Andrew', 'Musak', to_date('07/12/1987','mm/dd/yyyy'), 'M', 33, '121 Collins Rd.', 'Toronto', 'ON', 'M1V1N3', '4168794561','dr.andrew.musak33@gmail.com', 123);
INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
    VALUES (323678, 'Mindy', 'Ramirez', to_date('10/10/1990','mm/dd/yyyy'), 'F', 30, '126 Sisao St.', 'Toronto', 'ON', 'M1P4R2', '6475213125','mindy.ramirez@gmail.com', 124);
INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
    VALUES (323679, 'John', 'Cooper', to_date('08/10/1989','mm/dd/yyyy'), 'M', 27, 'B9 Prospect Street', 'Toronto', 'ON', 'M1PX13', '6475213256','cooper.john@gmail.com', 124);
INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
    VALUES (323680, 'Michelle', 'Stanley', to_date('10/06/1991','mm/dd/yyyy'), 'F', 30, '202 Enble Street', 'Toronto', 'ON', 'M1D13C', '6473125676','stanley.michelle@gmail.com', 123);
INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
    VALUES (323681, 'Samuel', 'Honey', to_date('10/12/1990','mm/dd/yyyy'), 'M', 31, 'A4 Laven Street', 'Toronto', 'AL', 'M2X3B1', '64753561234','samuel.honey@gmail.com', 125);
INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
    VALUES (323682, 'Ruby', 'Yul', to_date('10/11/1992','mm/dd/yyyy'), 'F', 26, 'C6 Thorne Street', 'Toronto', 'ON', 'M2R32X', '6475352132','ruby.yul@gmail.com', 125);
-- doctors
INSERT INTO doctor(doctor_id, doctorlicense_expiry, employee_id) VALUES (141524, to_date('03/06/2030','mm/dd/yyyy'), 323952);
INSERT INTO doctor(doctor_id, doctorlicense_expiry, employee_id) VALUES (141525, to_date('03/12/2020','mm/dd/yyyy'), 323678);
INSERT INTO doctor(doctor_id, doctorlicense_expiry, employee_id) VALUES (141526, to_date('04/06/2012','mm/dd/yyyy'), 323679);
-- nurses
INSERT INTO nurse(nurse_id, nurselicense_expiry, employee_id) VALUES(555879, to_date('09/10/2025','mm/dd/yyyy'), 323680);
INSERT INTO nurse(nurse_id, nurselicense_expiry, employee_id) VALUES(555880, to_date('09/11/2022','mm/dd/yyyy'), 323681);
INSERT INTO nurse(nurse_id, nurselicense_expiry, employee_id) VALUES(555881, to_date('10/10/2023','mm/dd/yyyy'), 323682);
-- patients
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567890', 'Ann', 'Smith', to_date('01/28/1997','mm/dd/yyyy'), 'F', '22','1290 Bayview Rd.','Toronto', 'ON', 'M1B2X4', '6471234567','ann.smith97@gmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567891', 'Bob', 'Brown', to_date('02/21/2000','mm/dd/yyyy'), 'M', '20','11 Clifton St.','Toronto', 'ON', 'M2C1L5','6471234568','bob.brown00@hotmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567892', 'Carl', 'Jones', to_date('07/11/1999','mm/dd/yyyy'), 'M', '21','103 Roadhouse Rd.','Oakville', 'ON', 'L6L2X6','4162578564','carl.jones99@gmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567893', 'Dan', 'Miller', to_date('04/07/1998','mm/dd/yyyy'), 'M', '22','56 Steeling Ave.','Pickering', 'ON', 'L1V0A1','4162874587','dan.miller98@gmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567894', 'Eve', 'Williams', to_date('01/22/1998','mm/dd/yyyy'), 'F', '22','23 Greentint Cres.','Markham', 'ON', 'L1C3P2','6478985674','eve.williams22@hotmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567895', 'Stan', 'Murphy', to_date('01/29/1999','mm/dd/yyyy'), 'M', '30','1111 Bayview Rd.','Toronto', 'ON', 'M1B2X4', '6471212311','stan.murphy@gmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567896', 'Alice', 'Yum', to_date('03/21/2020','mm/dd/yyyy'), 'F', '21','11 Steeling St.','Toronto', 'ON', 'M2C3CD','6471234231','yum.alice@hotmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567897', 'Steve', 'Jan', to_date('07/12/1920','mm/dd/yyyy'), 'M', '23','2 Ins Street','Oakville', 'ON', 'L6L2X','4162532313','steve.jan@gmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567898', 'Mike', 'Lanny', to_date('08/07/1990','mm/dd/yyyy'), 'M', '40','55 Steeling Ave.','Pickering', 'ON', 'L1DED','4162875561','mike.lanny@gmail.com');
INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
    VALUES ('1234567899', 'Camy', 'Hun', to_date('01/30/1932','mm/dd/yyyy'), 'F', '30','2 Hopkins Street','Markham', 'ON', 'L1QD2P','6478983164','camy.hun@hotmail.com');
```

Insertion of data ***HospitalDBMS_A4.sq***

```
-- appointments
INSERT INTO appointment (appointment_id,appointment_date,appointment_time,room_no,nurse_id,healthcard_no,doctor_id,hospital_id)
    VALUES (10031, to_date('12/12/2020', 'mm/dd/yyyy'), 1300, 1, 555879, 1234567890, 141524, 123) ;
INSERT INTO appointment (appointment_id,appointment_date,appointment_time,room_no,nurse_id,healthcard_no,doctor_id,hospital_id)
    VALUES (10012, to_date('10/20/2020', 'mm/dd/yyyy'), 1400, 2, 555880, 1234567891, 141525, 124) ;
INSERT INTO appointment (appointment_id,appointment_date,appointment_time,room_no,nurse_id,healthcard_no,doctor_id,hospital_id)
    VALUES (10013, to_date('10/22/2020', 'mm/dd/yyyy'), 1500, 3, 555881, 1234567892, 141526, 125);
-- medicine
INSERT INTO medicine(medicine_id, iupac_name, generic_name, inventory, price, expiration_date, manufacturer)
    VALUES(191919,'N-(4-hydroxyphenyl)acetamide','acetaminophen',500,5.00,to_date('01/01/2030','mm/dd/yyyy'), 'Tylenol') ;
INSERT INTO medicine(medicine_id, iupac_name, generic_name, inventory, price, expiration_date, manufacturer)
    VALUES(191920,'8-Chloro-1-methyl-6-phenyl-4H-[1,2,4]triazolo[4,3-a][1,4]benzodiazepine','alprazolam',200,50.00,to_date('02/02/2030','mm/dd/yyyy'), 'Tylenol') ;
INSERT INTO medicine(medicine_id, iupac_name, generic_name, inventory, price, expiration_date, manufacturer)
    VALUES(191921,'(S,S)-2-methylamino-1-phenylpropan-1-ol', 'pseudoephedrine', 100, 50.00, to_date('10/22/2030','mm/dd/yyyy'), 'Johnson and Johnson') ;
-- invoices
INSERT INTO invoice (invoice_no, medicine_id, date_issued, amount_owed, appointment_id)
    VALUES (111112,3, to_date('10/20/2020', 'mm/dd/yyyy'), 100.00, 10012) ;
INSERT INTO invoice (invoice_no,medicine_id,date_issued,amount_owed,appointment_id)
    VALUES (111113,3, to_date('10/22/2020', 'mm/dd/yyyy'), 10.00, 10013) ;
-- diagnosis
INSERT INTO diagnosis (diagnosis_id, results, appointment_id)
    VALUES(0123456, 'Stomach cancer',  10012) ;
INSERT INTO diagnosis (diagnosis_id, results, appointment_id)
    VALUES(0123457, 'Covid-19',  10013) ;
-- prescription
INSERT INTO prescription(prescription_no, dosage,date_issued,appointment_id, medicine_id, diagnosis_id)
    VALUES (1231231, '20', to_date('10/20/2020','mm/dd/yyyy'), 10012, 191919, 0123456) ;
INSERT INTO prescription(prescription_no, dosage,date_issued,appointment_id, medicine_id, diagnosis_id)
    VALUES (1231232, '50', to_date('10/22/2020', 'mm/dd/yyyy'), 10013, 191919, 0123457) ;
-- medical history
INSERT INTO medical_history(healthcard_no, appointment_id, diagnosis_id, medical_desc)
    VALUES(1234567891, 10012, 0123456, 'This guy has stomach cancer, he is allergic to tylonel so prescribe him something else') ;
INSERT INTO medical_history(healthcard_no, appointment_id, diagnosis_id, medical_desc)
    VALUES(1234567892, 10013, 0123457, 'Make sure they stay at home for two weeks') ;
```

```
---------------
--- QUERIES ---
---------------
-- Hospital Table
-- get all the existing hospitals
SELECT * FROM hospital;
-- get  hospitals located in toronto
SELECT * FROM hospital WHERE address_city = 'Toronto';

-- Employee Table
SELECT * FROM hospital;
-- get employee with hospital ids, 123
SELECT * FROM employee ORDER BY hospital_id;
SELECT * FROM employee WHERE hospital_id = 123;
-- sort employees by age
SELECT * FROM employee ORDER BY date_of_birth ASC;
-- counts
SELECT COUNT(employee_id) AS "# of Employees", hospital_id FROM employee GROUP BY hospital_id;

-- Doctor Table
SELECT * FROM doctor;
-- find doctor_id from doctors with ID's that expired in 2020
SELECT doctor_id FROM doctor WHERE doctorlicense_expiry < to_date('01/01/2020','mm/dd/yyyy');

-- Nurse Table
SELECT * FROM nurse;
-- find nurse_id from nurses with ID's that expire in the next 5 years
SELECT nurse_id FROM nurse WHERE nurselicense_expiry < to_date('01/01/2025','mm/dd/yyyy');

-- Patient Table
SELECT * FROM patient;
-- find all the women age 22 from toronto
SELECT * FROM patient WHERE gender = 'F' AND address_city = 'Toronto' AND age < 22 ;

-- Appointment Table
SELECT * FROM appointment ;
-- find appointments taken during the November 18 to 25, in the year 2020
SELECT * FROM appointment WHERE appointment_date >= to_date('10/18/2020', 'mm/dd/yyyy') AND appointment_date < to_date('10/25/2020','mm/dd/yyyy') ;

-- Invoice Table
SELECT * FROM invoice ;
--find all invoices owing 100 or more
SELECT * FROM invoice WHERE amount_owed >= 100 ;

-- Diagnosis Table
SELECT * FROM DIAGNOSIS ;
--find all diagnosises that include cancer
SELECT * FROM diagnosis WHERE results LIKE '%cancer%' ;

-- Medicine Table
SELECT * FROM medicine ;
--find all drugs made by johnson and johnson
SELECT * FROM medicine WHERE manufacturer='Johnson and Johnson' ;
```

```
-- Prescription Table--
SELECT * FROM prescription ;
--find all prescriptions for medicine_id='191919', and sort them descending by dosage
SELECT * FROM prescription WHERE medicine_id = '191919' ORDER BY dosage DESC ;

-- Medical History Table --
SELECT * FROM medical_history ;
-- sort medical histories by healthcard_no
SELECT * FROM medical_history ORDER BY healthcard_no ASC ;
```

Queries

```sql
-- ADVANCED QUERIES
-- JOIN queries
-- gets email from given name in appointment
SELECT email
FROM appointment a, patient p WHERE
    p.f_name = 'Ann'
    AND p.l_name = 'Smith'
    AND a.healthcard_no = p.healthcard_no;
-- gets appointment id, of Patient, Carl Jones and Doctor, Mindy Ramirez
SELECT appointment_id
FROM appointment a, patient p, doctor d, nurse n, employee e WHERE
    p.f_name = 'Carl' AND p.l_name = 'Jones'
    AND e.f_name = 'Mindy' AND e.l_name = 'Ramirez'
    and d.employee_id = e.employee_id
    AND a.doctor_id = d.doctor_id
    AND a.nurse_id = n.nurse_id
    AND a.healthcard_no = p.healthcard_no;
-- gets doctor and pateint for appointment_id
SELECT e.f_name AS D_firstname, e.l_name AS D_lastname, p.f_name as P_firstname, p.l_name as P_lastname
FROM appointment a, patient p, doctor d, employee e
WHERE a.appointment_id =  10031
    AND a.doctor_id = d.doctor_id
    AND d.employee_id = e.employee_id
    AND a.healthcard_no = p.healthcard_no;

-- EXISTS queries, implements intersection
SELECT generic_name, inventory, price, expiration_date, manufacturer
FROM medicine
    WHERE EXISTS
    (SELECT prescription_no
     FROM prescription
     WHERE prescription.medicine_id = medicine.medicine_id);

SELECT e.f_name, e.f_name
FROM employee e
WHERE EXISTS
    (SELECT e.f_name, e.f_name
    FROM employee e, nurse n
    WHERE
    n.employee_id = 323680
    AND e.employee_id = n.employee_id);

SELECT p1.f_name, p1.l_name
FROM patient p1
WHERE EXISTS
    (SELECT p2.f_name, p2.l_name
    FROM patient p2, appointment a, doctor d, employee e
    WHERE e.f_name = 'John' AND e.l_name = 'Cooper'
          AND e.employee_id = d.employee_id
          AND a.doctor_id = d.doctor_id
          AND a.healthcard_no = p2.healthcard_no);
```

```sql
-- UNION queries
SELECT employee_id FROM nurse
    UNION
SELECT employee_id FROM doctor;

-- MINUS queries
-- select all employees not working in Toronto General Hospital
SELECT * FROM employee
MINUS
(SELECT e.*
FROM employee e, hospital h
WHERE h.hospital_name = 'Toronto General Hospital'
AND h.hospital_id = e.hospital_id);

-- COUNT
SELECT 'Total number of patients: ', COUNT(healthcard_no)
FROM patient ;

-- GROUP BY
SELECT COUNT(medicine_id), manufacturer
FROM medicine
GROUP BY manufacturer;

SELECT COUNT(employee_id) AS "# of Employees", hospital_id
FROM employee
GROUP BY hospital_id;

-- OTHER queries
SELECT 'Average cost of medicine is from sinopharm is ', AVG(price)
FROM medicine
WHERE manufacturer='Sinopharm' ;
```

Advance quieres

```
---------------
---- VIEWS ----
---------------
DROP VIEW JOHNSON_AND_JOHNSON_MEDICINES ;
CREATE VIEW JOHNSON_AND_JOHNSON_MEDICINES AS
SELECT medicine_id, iupac_name,generic_name,expiration_date,manufacturer,inventory
FROM medicine
WHERE manufacturer='Johnson and Johnson';
    --Selecting from above view
    SELECT * FROM JOHNSON_AND_JOHNSON_MEDICINES ;
    --Select medicines from view which expire in 2020
    SELECT * FROM JOHNSON_AND_JOHNSON_MEDICINES WHERE expiration_date < to_date('01/01/2021', 'mm/dd/yyyy') ;
    --Select from johnson_and_johnson_medcines, medcines that are low in stock
    SELECT * FROM JOHNSON_AND_JOHNSON_MEDICINES WHERE inventory <= 100 ;
    SELECT * FROM medicine MINUS SELECT * FROM JOHNSON_AND_JOHNSON_MEDICINES ;

DROP VIEW TORONTO_HOSPITALS ;
CREATE VIEW TORONTO_HOSPITALS AS
SELECT hospital_id, hospital_name, address_street,address_city,address_province, address_postalcode, address_country, phone
FROM hospital
WHERE address_city='Toronto' ;
    ---Selecting from the view
    SELECT * FROM TORONTO_HOSPITALS;

DROP VIEW ELDERLY_PATIENTS ;
CREATE VIEW ELDERLY_PATIENTS AS
SELECT *
FROM patient
WHERE age >= 65 ;
    --Selecting the view
    SELECT * FROM ELDERLY_PATIENTS ;
    --Order ELDERLY_PATIENTS by age
    SELECT * FROM ELDERLY_PATIENTS ORDER BY age DESC ;
```

View queries

Assignment 5: Demo of Adv. Queries by Unix Shell Implementation

```
#!/bin/sh
MainMenu()
{
   while [ "$CHOICE" != "START" ]
   do
      clear
   echo "============================================================="
   echo "|                 Oracle All Inclusive Tool                 |"
   echo "|        Main Menu - Select Desired Operation(s):           |"
   echo "|        <CTRL-Z Anytime to Enter Interactive CMD Prompt>   |"
   echo "-------------------------------------------------------------"
   echo " $IS_SELECTEDM M) View Manual"
   echo " "
   echo " $IS_SELECTED1 1) Drop Tables"
   echo " $IS_SELECTED2 2) Create Tables"
   echo " $IS_SELECTED3 3) Populate Tables"
   echo " $IS_SELECTED4 4) Query Tables"
   echo " "
   echo " $IS_SELECTEDX X) Force/Stop/Kill Oracle DB"
   echo " "
   echo " $IS_SELECTEDE E) End/Exit"
   echo "Choose: "

   read CHOICE
   if [ "$CHOICE" == "0" ]
   then
      echo "unknown input"

   elif [ "$CHOICE" == "1" ]
   then
      bash drop_tables.sh
      Pause

   elif [ "$CHOICE" == "2" ]
   then
      bash create_tables.sh
      Pause

   elif [ "$CHOICE" == "3" ]
   then
      bash populate_tables.sh
      Pause

   elif [ "$CHOICE" == "4" ]
   then
      bash queries.sh
      Pause

   elif [ "$CHOICE" == "E" ]
   then
      exit
   fi
done }
```

```
function Pause(){
  read -p "";
}

#--COMMENTS BLOCK--
# Main Program
#--COMMENTS BLOCK--
ProgramStart()
{
   StartMessage
   while [ 1 ]
   do
      MainMenu
   done
}

ProgramStart
```

shell_menu.sh

create_tables.sh

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
sqlplus64 "jamanipo/07226166@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
    CREATE TABLE hospital (
        hospital_id INT NOT NULL,
        hospital_name VARCHAR2(32) NOT NULL,
        address_street VARCHAR2(32) NOT NULL,
        address_street2 VARCHAR2(32),
        address_city VARCHAR2(32) NOT NULL,
        address_province VARCHAR2(2) NOT NULL,
        address_postalcode VARCHAR2(6) NOT NULL,
        address_country VARCHAR2(3) DEFAULT 'CAD',
        phone VARCHAR2(16) NOT NULL,
        PRIMARY KEY(hospital_id)
    );

    CREATE TABLE employee (
        employee_id INT NOT NULL,
        f_name VARCHAR2(64) NOT NULL,
        l_name VARCHAR2(64) NOT NULL,
        date_of_birth DATE NOT NULL,
        gender VARCHAR2(6) NOT NULL,
        age INT NOT NULL,
        address_street VARCHAR2(32) NOT NULL,
        address_street2 VARCHAR2(32),
        address_city VARCHAR2(32) NOT NULL,
        address_province VARCHAR2(2) NOT NULL,
        address_postalcode VARCHAR2(6) NOT NULL UNIQUE,
        address_country VARCHAR2(3) DEFAULT 'CAD',
        phone VARCHAR2(16) NOT NULL,
        email VARCHAR2(256),
        hospital_id INT NOT NULL,
        PRIMARY KEY(employee_id),
        FOREIGN KEY(hospital_id) REFERENCES hospital(hospital_id)
    );

    CREATE TABLE doctor (
        doctor_id INT NOT NULL,
        doctorlicense_expiry DATE NOT NULL,
        employee_id INT NOT NULL,
        PRIMARY KEY(doctor_id),
        FOREIGN KEY(employee_id) REFERENCES employee(employee_id)
    );

    CREATE TABLE nurse (
        nurse_id INT NOT NULL,
        nurselicense_expiry DATE NOT NULL,
        employee_id INT NOT NULL,
        PRIMARY KEY(nurse_id),
        FOREIGN KEY(employee_id) REFERENCES employee(employee_id)
    );
```

drop_tables.sh

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
sqlplus64 "jamanipo/07226166@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
    DROP TABLE administers;
    DROP TABLE medical_history ;
    DROP TABLE prescription;
    DROP TABLE diagnosis;
    DROP TABLE medicine;
    DROP TABLE invoice;
    DROP TABLE appointment ;
    DROP TABLE patient ;
    DROP TABLE nurse ;
    DROP TABLE doctor;
    DROP TABLE employee CASCADE CONSTRAINTS;
    DROP TABLE hospital CASCADE CONSTRAINTS;
    exit;
EOF
```

populate_tables.sh

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
sqlplus64 "jamanipo/07226166@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
  INSERT INTO hospital (hospital_id, hospital_name,address_street,address_city,address_province,address_postalcode,phone)
    VALUES (123, 'Toronto General Hospital', '200 Elizabeth St', 'Toronto', 'ON', 'M5G2C4','4167895297');
    INSERT INTO hospital (hospital_id, hospital_name,address_street,address_city,address_province,address_postalcode,phone)
        VALUES (124, 'Michael Garron Hospital', '825 Coxwell Ave', 'East York', 'ON', 'M4C3E7','4167893127');
    INSERT INTO hospital (hospital_id, hospital_name,address_street,address_city,address_province,address_postalcode,phone)
        VALUES (125, 'Toronto Western Hospital', '399 Bathurst St', 'Toronto', 'ON', 'M5T2S8','4167898713');
    -- employees
    INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
        VALUES (323952, 'Andrew', 'Musak', to_date('07/12/1987','mm/dd/yyyy'), 'M', 33, '121 Collins Rd.', 'Toronto', 'ON', 'M1V1N3', '4168794561','dr.andrew.musak33@gmail.com', 123);
    INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
        VALUES (323678, 'Mindy', 'Ramirez', to_date('10/10/1990','mm/dd/yyyy'), 'F', 30, '126 Sisao St.', 'Toronto', 'ON', 'M1P4R2', '6475213125','mindy.ramirez@gmail.com', 124);
    INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
        VALUES (323679, 'John', 'Cooper', to_date('08/10/1989','mm/dd/yyyy'), 'M', 27, 'B9 Prospect Street', 'Toronto', 'ON', 'M1PX13', '6475213256','cooper.john@gmail.com', 124);
    INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
        VALUES (323680, 'Michelle', 'Stanley', to_date('10/06/1991','mm/dd/yyyy'), 'F', 30, '202 Enble Street', 'Toronto', 'ON', 'M1D13C', '6473125676','stanley.michelle@gmail.com', 123);
    INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
        VALUES (323681, 'Samuel', 'Honey', to_date('10/12/1990','mm/dd/yyyy'), 'M', 31, 'A4 Laven Street', 'Toronto', 'AL', 'M2X3B1', '64753561234','samuel.honey@gmail.com', 125);
    INSERT INTO employee (employee_id, f_name, l_name, date_of_birth, gender, age, address_street, address_city, address_province, address_postalcode, phone, email, hospital_id)
        VALUES (323682, 'Ruby', 'Yul', to_date('10/11/1992','mm/dd/yyyy'), 'F', 26, 'C6 Thorne Street', 'Toronto', 'ON', 'M2R32X', '6475352132','ruby.yul@gmail.com', 125);
    -- doctors
    INSERT INTO doctor(doctor_id, doctorlicense_expiry, employee_id) VALUES (141524, to_date('03/06/2030','mm/dd/yyyy'), 323952);
    INSERT INTO doctor(doctor_id, doctorlicense_expiry, employee_id) VALUES (141525, to_date('03/12/2020','mm/dd/yyyy'), 323678);
    INSERT INTO doctor(doctor_id, doctorlicense_expiry, employee_id) VALUES (141526, to_date('04/06/2012','mm/dd/yyyy'), 323679);
    -- nurses
    INSERT INTO nurse(nurse_id, nurselicense_expiry, employee_id) VALUES(555879, to_date('09/10/2025','mm/dd/yyyy'), 323680);
    INSERT INTO nurse(nurse_id, nurselicense_expiry, employee_id) VALUES(555880, to_date('09/11/2022','mm/dd/yyyy'), 323681);
    INSERT INTO nurse(nurse_id, nurselicense_expiry, employee_id) VALUES(555881, to_date('10/10/2023','mm/dd/yyyy'), 323682);
    -- patients
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567890', 'Ann', 'Smith', to_date('01/28/1997','mm/dd/yyyy'), 'F', '22','1290 Bayview Rd.','Toronto', 'ON', 'M1B2X4', '6471234567','ann.smith97@gmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567891', 'Bob', 'Brown', to_date('02/21/2000','mm/dd/yyyy'), 'M', '20','11 Clifton St.','Toronto', 'ON', 'M2C1L5','6471234568','bob.brown00@hotmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567892', 'Carl', 'Jones', to_date('07/11/1999','mm/dd/yyyy'), 'M', '21','103 Roadhouse Rd.','Oakville', 'ON', 'L6L2X6','4162578564','carl.jones99@gmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567893', 'Dan', 'Miller', to_date('04/07/1998','mm/dd/yyyy'), 'M', '22','56 Steeling Ave.','Pickering', 'ON', 'L1V0A1','4162874587','dan.miller98@gmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567894', 'Eve', 'Williams', to_date('01/22/1998','mm/dd/yyyy'), 'F', '22','23 Greentint Cres.','Markham', 'ON', 'L1C3P2','6478985674','eve.williams22@hotmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567895', 'Stan', 'Murphy', to_date('01/29/1999','mm/dd/yyyy'), 'M', '30','1111 Bayview Rd.','Toronto', 'ON', 'M1B2X4', '6471212311','stan.murphy@gmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567896', 'Alice', 'Yum', to_date('03/21/2020','mm/dd/yyyy'), 'F', '21','11 Steeling St.','Toronto', 'ON', 'M2C3CD','6471234231','yum.alice@hotmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567897', 'Steve', 'Jan', to_date('07/12/1920','mm/dd/yyyy'), 'M', '23','2 Ins Street','Oakville', 'ON', 'L6L2X','4162532313','steve.jan@gmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567898', 'Mike', 'Lanny', to_date('08/07/1990','mm/dd/yyyy'), 'M', '40','55 Steeling Ave.','Pickering', 'ON', 'L1DED','4162875561','mike.lanny@gmail.com');
    INSERT INTO patient (healthcard_no,f_name,l_name, date_of_birth, gender,age,address_street,address_city,address_province,address_postalcode,phone, email)
        VALUES ('1234567899', 'Camy', 'Hun', to_date('01/30/1932','mm/dd/yyyy'), 'F', '30','2 Hopkins Street','Markham', 'ON', 'L1QD2P','6478983164','camy.hun@hotmail.com');
```

queries.sh

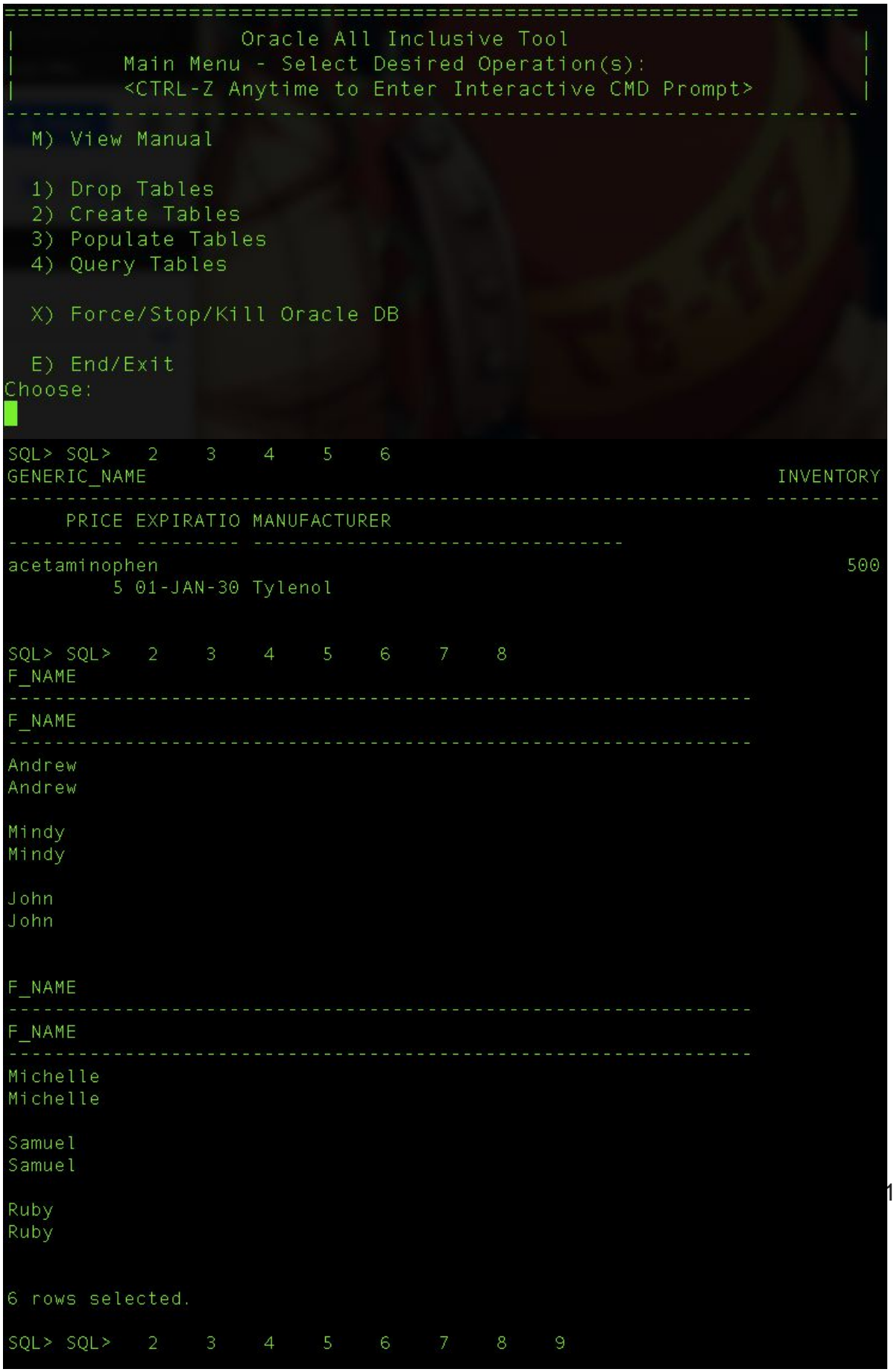
```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
sqlplus64 "jamanipo/07226166@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
   SELECT email
    FROM appointment a, patient p WHERE
        p.f_name = 'Ann'
        AND p.l_name = 'Smith'
        AND a.healthcard_no = p.healthcard_no;

    FROM appointment a, patient p, doctor d, nurse n, employee e WHERE
        p.f_name = 'Carl' AND p.l_name = 'Jones'
        AND e.f_name = 'Mindy' AND e.l_name = 'Ramirez'
        and d.employee_id = e.employee_id
        AND a.doctor_id = d.doctor_id
        AND a.nurse_id = n.nurse_id
        AND a.healthcard_no = p.healthcard_no;

    SELECT e.f_name AS D_firstname, e.l_name AS D_lastname, p.f_name as P_firstname, p.l_name as P_lastname
    FROM appointment a, patient p, doctor d, employee e
    WHERE a.appointment_id =  10031
        AND a.doctor_id = d.doctor_id
        AND d.employee_id = e.employee_id
        AND a.healthcard_no = p.healthcard_no;

    SELECT generic_name, inventory, price, expiration_date, manufacturer
    FROM medicine
        WHERE EXISTS
        (SELECT prescription_no
        FROM prescription
        WHERE prescription.medicine_id = medicine.medicine_id);
```

Shell results



```
================================================================
|                  Oracle All Inclusive Tool                   |
|         Main Menu - Select Desired Operation(s):             |
|         <CTRL-Z Anytime to Enter Interactive CMD Prompt>     |
----------------------------------------------------------------
  M) View Manual

  1) Drop Tables
  2) Create Tables
  3) Populate Tables
  4) Query Tables

  X) Force/Stop/Kill Oracle DB

  E) End/Exit
Choose:
SQL> SQL>   2    3    4    5    6
GENERIC_NAME                                                    INVENTORY
------------------------------------------------------------------ ----------
     PRICE EXPIRATIO MANUFACTURER
---------- --------- --------------------------------
acetaminophen                                                         500
         5 01-JAN-30 Tylenol


SQL> SQL>   2    3    4    5    6    7    8
F_NAME
----------------------------------------------------------------
F_NAME
----------------------------------------------------------------
Andrew
Andrew

Mindy
Mindy

John
John


F_NAME
----------------------------------------------------------------
F_NAME
----------------------------------------------------------------
Michelle
Michelle

Samuel
Samuel

Ruby
Ruby


6 rows selected.

SQL> SQL>   2    3    4    5    6    7    8    9
```

Assignment 6: Normalization of The Database/Functional Dependencies

Hospital FDs:

	HOSPITAL_ID	HOSPITAL_NAME	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PROVINCE	ADDRESS_POSTALCODE	ADDRESS_COUNTRY	PHONE
1	123	Toronto General Hospital	200 Elizabeth St	(null)	Toronto	ON	M5G2C4	CAD	4167895297
2	124	Michael Garron Hospital	825 Coxwell Ave	(null)	East York	ON	M4C3E7	CAD	4167893127
3	125	Toronto Western Hospital	399 Bathurst St	(null)	Toronto	ON	M5T2S8	CAD	4167898713

Hospital ID → {Hospital Name, Address, Phone}

Employee FDs:

	EMPLOYEE_ID	F_NAME	L_NAME	DATE_OF_BIRTH	GENDER	AGE	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PROVINCE	ADDRESS_POSTALCODE	ADDRESS_COUNTRY	PHONE	EMAIL	HOSPITAL
1	323952	Andrew	Musak	87-07-12	M	33	121 Collins Rd.	(null)	Toronto	ON	M1V1N3	CAD	4168794561	dr.andrew.musak33@gmail.com	
2	323678	Mindy	Ramirez	90-10-10	F	30	126 Sisao St.	(null)	Toronto	ON	M1P4R2	CAD	6475213125	mindy.ramirez@gmail.com	
3	323679	John	Cooper	89-08-10	M	27	B9 Prospect Street	(null)	Toronto	ON	M1PX13	CAD	6475213256	cooper.john@gmail.com	
4	323680	Michelle	Stanley	91-10-06	F	30	202 Enble Street	(null)	Toronto	ON	M1D13C	CAD	6473125676	stanley.michelle@gmail.com	
5	323681	Samuel	Honey	90-10-12	M	31	A4 Laven Street	(null)	Toronto	AL	M2X3B1	CAD	64753561234	samuel.honey@gmail.com	
6	323682	Ruby	Yul	92-10-11	F	26	C6 Thorne Street	(null)	Toronto	ON	M2R32X	CAD	6475352132	ruby.yul@gmail.com	
7	323683	Robert	Smith	82-12-15	M	38	123 Bay Rd.	(null)	Toronto	ON	M1V1N2	CAD	416279454	rsmith@gmail.com	

Employee ID → { First Name, Last Name, DOB, Gender, Address, Phone, Email }

DOB → Age

Employee ID → Hospital ID

Doctor FDs:

	DOCTOR_ID	DOCTORLICENSE_EXPIRY	EMPLOYEE_ID
1	141524	30-03-06	323952
2	141525	20-03-12	323678
3	141526	12-04-06	323679

Employee ID → Doctor ID

Doctor ID → Doctor License Expiry

Nurse FDs:

	NURSE_ID	NURSELICENSE_EXPIRY	EMPLOYEE_ID
1	555879	25-09-10	323680
2	555880	22-09-11	323681
3	555881	23-10-10	323682

Employee ID → Nurse ID

Nurse ID → Nurse License Expiry

Patient FDs:

	HEALTHCA...	F_...	L_NAME	DATE_OF_BIRTH	GENDER	AGE	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PROVINCE	ADDRESS_POSTALCODE	ADDRESS_COUNTRY	PHONE	EMAIL	HOSPITAL_ID
1	1234567890	Ann	Smith	97-01-28	F	22	1290 Bayview Rd.	(null)	Toronto	ON	M1B2X4	CAD	6471234567	ann.smith97@gmail.com	123
2	1234567891	Bob	Brown	00-02-21	M	20	11 Clifton St.	(null)	Toronto	ON	M2C1L5	CAD	6471234568	bob.brown00@hotmail.com	124
3	1234567892	Carl	Jones	99-07-11	M	21	103 Roadhouse Rd.	(null)	Oakville	ON	L6L2X6	CAD	4162578564	carl.jones99@gmail.com	125
4	1234567893	Dan	Miller	98-04-07	M	22	56 Steeling Ave.	(null)	Pickering	ON	L1V0A1	CAD	4162874587	dan.miller98@gmail.com	125
5	1234567894	Eve	Williams	98-01-22	F	22	23 Greentint Cres.	(null)	Markham	ON	L1C3P2	CAD	6478985674	eve.williams22@hotmail.com	124
6	1234567895	Stan	Murphy	99-01-29	M	30	1111 Bayview Rd.	(null)	Toronto	ON	M1B2X4	CAD	6471212311	stan.murphy@gmail.com	123
7	1234567896	Alice	Yum	20-03-21	F	21	11 Steeling St.	(null)	Toronto	ON	M2C3CD	CAD	6471234231	yum.alice@hotmail.com	123
8	1234567897	Steve	Jan	20-07-12	M	23	2 Ins Street	(null)	Oakville	ON	L6L2X	CAD	4162532313	steve.jan@gmail.com	124
9	1234567898	Mike	Lanny	90-08-07	M	40	55 Steeling Ave.	(null)	Pickering	ON	L1DED	CAD	4162875561	mike.lanny@gmail.com	125
10	1234567899	Camy	Hun	32-01-30	F	30	2 Hopkins Street	(null)	Markham	ON	L1QD2P	CAD	6478983164	camy.hun@hotmail.com	123

Healthcard # → { First name, Last name, DOB, Gender, Address, Phone, Email, Hospital ID }

DOB → Age

Healthcard # → Hospital ID

Appointment FDs:

	APPOINTMENT_ID	APPOINTMENT_DATE	APPOINTMENT_TIME	ROOM_NO	NURSE_ID	HEALTHCARD_NO	DOCTOR_ID	HOSPITAL_ID
1	10031	20-12-12	1300	1	555879	1234567890	141524	123
2	10012	20-10-20	1400	2	555880	1234567897	141526	124
3	10013	20-10-22	1500	3	555881	1234567892	141525	125

Appointment ID → { Appointment Date, Appointment Time, Room Number, Nurse ID,
Healthcard #, Doctor ID, Hospital ID }

{Doctor ID, Nurse ID} → Hospital ID

Appointment ID → Doctor ID

Appointment ID → Healthcard # Appointment ID → Diagnosis ID

Invoice FDs:

	INVOICE_NO	MEDICINE_ID	DATE_ISSUED	AMOUNT_OWED	APPOINTMENT_ID	HOSPITAL_ID
1	111112	3	20-10-20	100	10012	123
2	111113	3	20-10-22	10	10013	124

Invoice # → { Medicine ID, Date Issued, Appointment ID } Amount Owed → Medicine ID

Invoice # → Hospital ID

Invoice # → Healthcard #

Diagnosis FDs:

	DIAGNOSIS_ID	RESULTS	APPOINTMENT_ID
1	123456	Stomach cancer	10012
2	123457	Covid-19	10013

Diagnosis ID → { Results, Appointment ID } Diagnosis ID → Appointment ID

Prescription FDs:

PRESCRIPTION_NO	DATE_ISSUED	APPOINTMENT_ID	MEDICINE_ID	DIAGNOSIS_ID
1231231	20-10-20	10012	191919	123456
1231232	20-10-22	10013	191919	123457
1231233	20-10-22	10013	191920	123457

Prescription # → { Date Issued, Medicine ID, Diagnosis ID, Appointment ID }

Medicine ID → Dosage

Prescription # → Diagnosis ID

Medical History FDs:

	HEALTHCARD_NO	APPOINTMENT_ID	DIAGNOSIS_ID	MEDICAL_DESC
1	1234567897	10012	123456	This guy has stomach cancer, he is allergic to tylonel so prescribe him something else
2	1234567892	10013	123457	Make sure they stay at home for two weeks

Healthcard # → { Diagnosis ID, Medical Description }

Appointment ID→ { Healthcard #, Diagnosis ID, Medical Description }

Diagnosis ID → { Healthcard #, Appointment ID, Medical Description }

Appointment ID → Diagnosis ID

Medicine FDs:

MEDICINE_ID	DOSAGE	IUPAC_NAME	GENERIC_NAME	INVENTORY	PRICE	EXPIRATION_DATE	MANUFACTURER	HOSPITAL_ID
191919	20	N-(4-hydroxyphenyl)acetamide	acetaminophen	500	5	30-01-01	Tylenol	123
191920	100	8-Chloro-1-methyl-6-phenyl-4H-[1,2,4]triazolo[4,3-a][1,4]benzodiazepine	alprazolam	200	50	20-12-25	Tylenol	123
191921	10	(S,S)-2-methylamino-1-phenylpropan-1-ol	pseudoephedrine	100	50	30-10-22	Johnson and Johnson	123
191922	50	(S)-2-Amino-3-[4-(4-hydroxy-3,5-diiodophenoxy)-3,5-diiodophenyl]propanoic acid	synthroid	50	25.5	29-06-01	Johnson and Johnson	123
191923	50	(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid	lipitor	200	60	20-12-01	Johnson and Johnson	123
191924	75	(S,S)-2-methylamino-1-phenylpropan-1-ol	pseudoephedrine	100	48.75	29-05-02	Sinopharm	123
191925	125	(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid	lipitor	1200	62.5	27-12-01	Sinopharm	123

Medicine ID → { IUPAC Name, Generic Name,Expiration Date } Medicine ID → Hospital ID

Hospital ID → { Inventory, Price }

Assignment 7: Normalization/3rd NF

Hospital:

	HOSPITAL_ID	HOSPITAL_NAME	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PROVINCE	ADDRESS_POSTALCODE	ADDRESS_COUNTRY	PHONE
1	123	Toronto General Hospital	200 Elizabeth St	(null)	Toronto	ON	M5G2C4	CAD	4167895297
2	124	Michael Garron Hospital	825 Coxwell Ave	(null)	East York	ON	M4C3E7	CAD	4167893127
3	125	Toronto Western Hospital	399 Bathurst St	(null)	Toronto	ON	M5T2S8	CAD	4167898713

Hospital ID → {Hospital Name, Address, Phone}

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: All non-primary attributes are dependent on the primary key. The table does not have a primary key which contains values >1 therefore partial dependencies do not exist.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Employee:

	EMPLOYEE_ID	F_NAME	L_NAME	DATE_OF_BIRTH	GENDER	AGE	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PROVINCE	ADDRESS_POSTALCODE	ADDRESS_COUNTRY	PHONE	EMAIL	HOSPITAL
1	323952	Andrew	Musak	87-07-12	M	33	121 Collins Rd.	(null)	Toronto	ON	M1V1N3	CAD	4168794561	dr.andrew.musak33@gmail.com	
2	323678	Mindy	Ramirez	90-10-10	F	30	126 Sisao St.	(null)	Toronto	ON	M1P4R2	CAD	6475213125	mindy.ramirez@gmail.com	
3	323679	John	Cooper	89-08-10	M	27	B9 Prospect Street	(null)	Toronto	ON	M1PX13	CAD	6475213256	cooper.john@gmail.com	
4	323680	Michelle	Stanley	91-10-06	F	30	202 Enble Street	(null)	Toronto	ON	M1D13C	CAD	6473125676	stanley.michelle@gmail.com	
5	323681	Samuel	Honey	90-10-12	M	31	A4 Laven Street	(null)	Toronto	AL	M2X3B1	CAD	64753561234	samuel.honey@gmail.com	
6	323682	Ruby	Yul	92-10-11	F	26	C6 Thorne Street	(null)	Toronto	ON	M2R32X	CAD	6475352132	ruby.yul@gmail.com	
7	323683	Robert	Smith	82-12-15	M	38	123 Bay Rd.	(null)	Toronto	ON	M1V1N2	CAD	416279454	rsmith@gmail.com	

Employee ID → { First Name, Last Name, DOB, Gender, Address, Phone, Email, Age }

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: All non-primary attributes are dependent on the primary key. The table does not have a primary key which contains values >1 therefore partial dependencies do not exist.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Doctor:

	DOCTOR_ID	DOCTORLICENSE_EXPIRY	EMPLOYEE_ID
1	141524	30-03-06	323952
2	141525	20-03-12	323678
3	141526	12-04-06	323679

Doctor ID → { Employee ID, Doctor License Expiry }

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: All non-primary attributes are dependent on the primary key. The table does not have a primary key which contains values >1 therefore partial dependencies do not exist.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF**

Nurse:

	NURSE_ID	NURSELICENSE_EXPIRY	EMPLOYEE_ID
1	555879	25-09-10	323680
2	555880	22-09-11	323681
3	555881	23-10-10	323682

Nurse ID → { Nurse License Expiry, Nurse ID }

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: All non-primary attributes are dependent on the primary key. The table does not have a primary key which contains values >1 therefore partial dependencies do not exist.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Patient FDs:

	HEALTHCA...	F_...	L_NAME	DATE_OF_BIRTH	GENDER	AGE	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PROVINCE	ADDRESS_POSTALCODE	ADDRESS_COUNTRY	PHONE	EMAIL	HOSPITAL_ID
1	1234567890	Ann	Smith	97-01-28	F	22	1290 Bayview Rd.	(null)	Toronto	ON	M1B2X4	CAD	6471234567	ann.smith97@gmail.com	123
2	1234567891	Bob	Brown	00-02-21	M	20	11 Clifton St.	(null)	Toronto	ON	M2C1L5	CAD	6471234568	bob.brown00@hotmail.com	124
3	1234567892	Carl	Jones	99-07-11	M	21	103 Roadhouse Rd.	(null)	Oakville	ON	L6L2X6	CAD	4162578564	carl.jones99@gmail.com	125
4	1234567893	Dan	Miller	98-04-07	M	22	56 Steeling Ave.	(null)	Pickering	ON	L1V0A1	CAD	4162874587	dan.miller98@gmail.com	125
5	1234567894	Eve	Williams	98-01-22	F	22	23 Greentint Cres.	(null)	Markham	ON	L1C3P2	CAD	6478985674	eve.williams22@hotmail.com	124
6	1234567895	Stan	Murphy	99-01-29	M	30	1111 Bayview Rd.	(null)	Toronto	ON	M1B2X4	CAD	6471212311	stan.murphy@gmail.com	123
7	1234567896	Alice	Yum	20-03-21	F	21	11 Steeling St.	(null)	Toronto	ON	M2C3CD	CAD	6471234231	yum.alice@hotmail.com	123
8	1234567897	Steve	Jan	20-07-12	M	23	2 Ins Street	(null)	Oakville	ON	L6L2X	CAD	4162532313	steve.jan@gmail.com	124
9	1234567898	Mike	Lanny	90-08-07	M	40	55 Steeling Ave.	(null)	Pickering	ON	L1DED	CAD	4162875561	mike.lanny@gmail.com	125
10	1234567899	Camy	Hun	32-01-30	F	30	2 Hopkins Street	(null)	Markham	ON	L1QD2P	CAD	6478983164	camy.hun@hotmail.com	123

Healthcard # → { First name, Last name, DOB, Gender, Address, Phone, Email, Hospital ID }

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: All non-primary attributes are dependent on the primary key. The table does not have a primary key which contains values >1 therefore partial dependencies do not exist.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Appointment:

	APPOINTMENT_ID	APPOINTMENT_DATE	APPOINTMENT_TIME	ROOM_NO	NURSE_ID	HEALTHCARD_NO	DOCTOR_ID	HOSPITAL_ID
1	10031	20-12-12	1300	1	555879	1234567890	141524	123
2	10012	20-10-20	1400	2	555880	1234567897	141526	124
3	10013	20-10-22	1500	3	555881	1234567892	141525	125

Appointment ID → { Appointment Date, Appointment Time, Room Number, Nurse ID, Healthcard #, Doctor ID, Hospital ID }

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: All non-primary attributes are dependent on the primary key. The table does not have a primary key which contains values >1 therefore partial dependencies do not exist.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Invoice FDs:

	INVOICE_NO	MEDICINE_ID	DATE_ISSUED	AMOUNT_OWED	APPOINTMENT_ID	HOSPITAL_ID
1	111112	3	20-10-20	100	10012	123
2	111113	3	20-10-22	10	10013	124

Invoice # → { Medicine ID, Date Issued, Appointment ID, Amount Owed, Health Card # }

Invoice # → Hospital ID

Invoice # → Healthcard #

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: All non-primary attributes are dependent on the primary key. The table does not have a primary key which contains values >1 therefore partial dependencies do not exist.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Diagnosis FDs:

	DIAGNOSIS_ID	RESULTS	APPOINTMENT_ID
1	123456	Stomach cancer	10012
2	123457	Covid-19	10013

Diagnosis ID → { Results, Appointment ID }

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: All non-primary attributes are dependent on the primary key. The table does not have a primary key which contains values >1 therefore partial dependencies do not exist.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Prescription FDs:

PRESCRIPTION_NO	DATE_ISSUED	APPOINTMENT_ID	MEDICINE_ID	DIAGNOSIS_ID
1231231	20-10-20	10012	191919	123456
1231232	20-10-22	10013	191919	123457
1231233	20-10-22	10013	191920	123457

Prescription # → { Date Issued, Medicine ID, Diagnosis ID, Appointment ID }
Prescription # → Diagnosis ID

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: This table contains a composite key but a partial dependency does not exists because all the non-primary attributes are dependent on the composite key.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Medical History:

	HEALTHCARD_NO	APPOINTMENT_ID	DIAGNOSIS_ID	MEDICAL_DESC
1	1234567897	10012	123456	This guy has stomach cancer, he is allergic to tylonel so prescribe him something else
2	1234567892	10013	123457	Make sure they stay at home for two weeks

Healthcard # → { Diagnosis ID, Medical Description }
Appointment ID→ { Healthcard #, Diagnosis ID, Medical Description }
Diagnosis ID → { Healthcard #, Appointment ID, Medical Description }
Appointment ID → Diagnosis ID

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.
2NF: This table contains a composite key but a partial dependency does not exist because all the non-primary attributes are dependent on the composite key.
3NF: All non-primary attributes are determined Only by the primary key in the table therefore transitive dependencies do not exist

Therefore, **this table is in 3NF.**

Medicine:

MEDICINE_ID	DOSAGE	IUPAC_NAME	GENERIC_NAME	INVENTORY	PRICE	EXPIRATION_DATE	MANUFACTURER	HOSPITAL_ID
191919	20	N-(4-hydroxyphenyl)acetamide	acetaminophen	500	5	30-01-01	Tylenol	123
191920	100	8-Chloro-1-methyl-6-phenyl-4H-[1,2,4]triazolo[4,3-a][1,4]benzodiazepine	alprazolam	200	50	20-12-25	Tylenol	123
191921	10	(S,S)-2-methylamino-1-phenylpropan-1-ol	pseudoephedrine	100	50	30-10-22	Johnson and Johnson	123
191922	50	(S)-2-Amino-3-[4-(4-hydroxy-3,5-diiodophenoxy)-3,5-diiodophenyl]propanoic acid	synthroid	50	25.5	29-06-01	Johnson and Johnson	123
191923	50	(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid	lipitor	200	60	20-12-01	Johnson and Johnson	123
191924	75	(S,S)-2-methylamino-1-phenylpropan-1-ol	pseudoephedrine	100	48.75	29-05-02	Sinopharm	123
191925	125	(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid	lipitor	1200	62.5	27-12-01	Sinopharm	123

Medicine ID → { IUPAC Name, Generic Name,Expiration Date }

Medicine ID → Hospital ID → { Inventory, Price }

1NF: This table has no multivalued (atomic values) attributes or nested relations, all entries in column are the same type and unique column identifiers are used.

2NF: This table does not have a primary key which contains >1 value therefore partial dependencies do not exist.

3NF: We have no transitive dependencies in this table.

Therefore, **this table is in 3NF.**

Assignment 8: Normalization 3NF/BCNF by Bernstein Algorithm

Patient Table: (Healthcard #, First name, Last name, DOB, Gender, Address, Phone, Email, Hospital ID)

HEALTHC...	F_NAME	L_NAME	DATE_OF...	GENDER	AGE	ADDRESS...	ADDRESS...	ADDRESS...	ADDRESS...	ADDRESS...	ADDRESS...	PHONE	EMAIL
1234567890	Ann	Smith	97-01-28	F	23	1290 Ba...	(null)	Toronto	ON	M1B2X4	CAD	6471234567	ann.smi...
1234567891	Bob	Brown	00-02-21	M	20	11 Clif...	(null)	Toronto	ON	M2C1L5	CAD	6471234568	bob.bro...
1234567892	Carl	Jones	99-07-11	M	21	103 Roa...	(null)	Oakville	ON	L6L2X6	CAD	4162578564	carl.jo...
1234567893	Dan	Miller	98-04-07	M	22	56 Stee...	(null)	Pickering	ON	L1V0A1	CAD	4162874587	dan.mil...
1234567894	Eve	Williams	98-01-22	F	22	23 Gree...	(null)	Markham	ON	L1C3P2	CAD	6478985674	eve.wil...
1234567895	Stan	Murphy	99-01-29	M	21	1111 Ba...	(null)	Toronto	ON	M1B2X4	CAD	6471212311	stan.mu...
1234567896	Alice	Yum	20-03-21	F	1	11 Stee...	(null)	Toronto	ON	M2C3CD	CAD	6471234231	yum.ali...
1234567897	Steve	Jan	20-07-12	M	100	2 Ins S...	(null)	Oakville	ON	L6L2X	CAD	4162532313	steve.j...
1234567898	Mike	Lanny	90-08-07	M	30	55 Stee...	(null)	Pickering	ON	L1DED	CAD	4162875561	mike.la...
1234567899	Camy	Hun	32-01-30	F	88	2 Hopki...	(null)	Markham	ON	L1QD2P	CAD	6478983164	camy.hu...
1234567900	Anthony	Trinh	99-03-12	M	21	135 Hil...	(null)	Mississ...	ON	L5B3Z2	CAD	6478983164	anthony...

Step 1: Determine functional dependencies

Health card # → First name
Health card # → Last name
Health card # → Address
Health card # → Phone
Health card # → Email
Health card # → DOB
Health card # → Age
First name, Last name → Health card #
Last name, Health card # → First name
Last name, Health card # → Age
Last name, Health card # → DOB
Last name, Health card # → Address
Last name, Health card # → Phone
Last name, Health card # → Email

Step 2: Find redundancies

We have to get rid of the redundant dependencies

Health card # → First name
Health card # → Last name
Health card # → Address
Health card # → Phone
Health card # → Email
Health card # → DOB

Health card # → Age
First name, Last name → Health card #
Last name, Health card # → First name
Last name, Health card # → Age
Last name, Health card # → DOB
Last name, Health card # → Address
Last name, Health card # → Phone
Last name, Health card # → Email

The last six FD's are redundant, so we can get rid of them.
There are no partial dependencies since in the dependency First name, Last name → Health card #, first names and last names are not unique, so they cannot alone determine the health card #.

Step 3: Find Keys

We have two candidate keys.
{Health Card #}
{First name, Last name}
{DOB}

Step 4: Find relations
R1(Health card #, First name, Last name, DOB, Age1, Address, Phone, Email)
R2(First name, Last name, Health card #)

Since R2 is a subset of R1, we can eliminate R2 and thus our final schema is R1.

Medicine Table: (Medicine ID, Dosage, IUPAC name, Generic name, Inventory, Price, Expiration date, Manufacturer, Hospital ID)

	MEDICINE_ID	DOSAGE	IUPAC_NAME	GENERIC_NAME	INVENTORY	PRICE	EXPIRATION_DATE	MANUFACTURER	HOSPITAL_ID
1	191919	20	N-(4-hydroxyphenyl)acetamide	acetaminophen	500	5	30-01-01	Tylenol	12
2	191920	100	8-Chloro-1-methyl-6-phenyl-4H-[1,2,4]triazolo[4,3-a][1,4]benzodiazepine	alprazolam	200	50	20-12-25	Tylenol	12
3	191921	10	(S,S)-2-methylamino-1-phenylpropan-1-ol	pseudoephedrine	100	50	30-10-22	Johnson and Johnson	12
4	191922	50	(S)-2-Amino-3-[4-(4-hydroxy-3,5-diiodophenoxy)-3,5-diiodophenyl]propanoic acid	synthroid	50	25.5	29-06-01	Johnson and Johnson	12
5	191923	50	(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid	lipitor	200	60	20-12-01	Johnson and Johnson	12
6	191924	75	(S,S)-2-methylamino-1-phenylpropan-1-ol	pseudoephedrine	100	48.75	29-05-02	Sinopharm	12
7	191925	125	(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid	lipitor	1200	62.5	27-12-01	Sinopharm	12

Step 1: Determine functional dependencies

Medicine ID → Dosage, IUPAC name, Generic name, Inventory, Price, Expiration date, Manufacturer, Hospital ID
Hospital ID → Inventory, Price
IUPAC name → Generic name
Manufacturer, Hospital ID, Dosage → Inventory, Price

Step 2: Find redundancies

Manufacturer, Hospital ID, Dosage → Inventory, Price and *Hospital ID → Inventory, Price* are redundant, so we will remove the latter.

Step 3: Find keys

Candidate keys are
{Medicine ID}
{Hospital ID}
{IUPAC Name}

Step 4: Find relations

R1(Medicine ID,Dosage, IUPAC name, Generic name, Inventory, Price, Expiration date, Manufacturer, Hospital ID)
R2(Hospital ID, Inventory, Price)
R3(IUPAC Name, Generic Name)

Since R2, R3 are subsets of R1, our final schema is R1.

Appointment Table: (Appointment ID, Appointment date, Appointment time, Room #, Nurse ID, Healthcard #, Doctor ID, Hospital ID)

APPOINTMENT_ID	APPOINTMENT_DATE	APPOINTMENT_TIME	ROOM_NO	NURSE_ID	HEALTHCARD_NO	DOCTOR_ID	HOSPITAL_ID
10031	20-12-12	1300	1	555879	1234567890	141524	123
10012	20-10-20	1400	2	555880	1234567897	141526	124
10013	20-10-22	1500	3	555881	1234567892	141525	125

Step 1: Determine functional dependencies

Appointment ID → Appointment date, Appointment time, Room #, Nurse ID, Healthcard #, Doctor ID, Hospital ID
Appointment date, Appointment time, Healthcard # → Appointment ID
Appointment date, Appointment time, Doctor ID → Appointment ID
Appointment date, Appointment time, Nurse ID → Appointment ID

Step 2: Find redundancies

Appointment date, Appointment time, Healthcard # → Appointment ID*, Appointment date, Appointment time, Doctor ID* → Appointment ID*, and Appointment date, Appointment time, Nurse ID* → Appointment ID are redundant, so we will remove the latter.

Step 3: Find keys

Candidate keys are
{ Appointment ID }

{ Appointment date, Appointment time, Healthcard # }
{ Appointment date, Appointment time, Doctor ID }
{ Appointment date, Appointment time, Nurse ID }

Step 4: Find relations

R1(Appointment ID, Appointment date, Appointment time, Room #, Nurse ID, Healthcard #, Doctor ID, Hospital ID)
R2(Appointment date, Appointment time, Healthcard # → Appointment ID)
R3(Appointment date, Appointment time, Doctor ID → Appointment ID)
R4(Appointment date, Appointment time, Nurse ID → Appointment ID)

Since R2, R3, R4 are subsets of R1, we can eliminate R2, R3, R4.
Our final schema is R1.

Hospital Table: (Hospital ID Hospital Name, Address, Phone)

	HOSPITAL_ID	HOSPITAL_NAME	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PR...	ADDRESS_POSTAL...	ADDRESS_COUNTRY	PHONE
1	124	Michael Garron Hospital	825 Coxwell Ave	(null)	East York	ON	M4C3E7	CAD	4167893127
2	125	Toronto Western Hospital	399 Bathurst St	(null)	Toronto	ON	M5T2S8	CAD	4167898713
3	123	Toronto General Hospital	200 Elizabeth St	(null)	Toronto	ON	M5G2C4	CAD	4167895297

Step 1: Determine the FD's that violate BCNF

Hospital ID → Hospital Name, Address, Phone

This is already in BCNF

Employee Table: (Employee ID, First name, Last name, DOB, Gender, Age, Address, Phone, Email, Hospital ID)

EMPLOYEE_ID	F_NAME	L_NAME	DATE_OF_BIRTH	GENDER	AGE	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PROVINCE	ADDRESS_POSTALCODE	ADDRESS_COUNTRY	PHONE	EMAIL	HOSPITAL_ID
323952	Andrew	Musak	87-07-12	M	33	121 Collins Rd.	(null)	Toronto	ON	M1V1N3	CAD	4168794561	dr.andrew.musak33@gmail.com	123
323678	Mindy	Ramirez	90-10-10	F	30	126 Sisao St.	(null)	Toronto	ON	M1P4R2	CAD	6475213125	mindy.ramirez@gmail.com	124
323679	John	Cooper	89-08-10	M	27	B9 Prospect Street	(null)	Toronto	ON	M1PX13	CAD	6475213256	cooper.john@gmail.com	124
323680	Michelle	Stanley	91-10-06	F	30	202 Enble Street	(null)	Toronto	ON	M1D13C	CAD	6473125676	stanley.michelle@gmail.com	123
323681	Samuel	Honey	90-10-12	M	31	A4 Laven Street	(null)	Toronto	AL	M2X3B1	CAD	64753561234	samuel.honey@gmail.com	125
323682	Ruby	Yul	92-10-11	F	26	C6 Thorne Street	(null)	Toronto	ON	M2R32X	CAD	6475352132	ruby.yul@gmail.com	125
323683	Robert	Smith	82-12-15	M	38	123 Bay Rd.	(null)	Toronto	ON	M1V1N2	CAD	416279454	rsmith@gmail.com	123

Step 1: Determine the FD's that violate BCNF
Employee ID → First name, Last name, DOB, Gender, Age, Address, Phone, Email, Hospital ID
DOB → Age

The last FD violates BCNF

Step 2: Decompose the tables

Now we have…

R1(DOB,Age)

R2(Employee ID, First name, Last name, DOB, Gender, Address, Phone, Email, Hospital)

Now, both of these tables are in BCNF.

Doctor Table: (Doctor ID, Doctor License Expiry, Employee ID)

DOCTOR_ID	DOCTORLICENSE_EXPIRY	EMPLOYEE_ID
141524	30-03-06	323952
141525	20-03-12	323678
141526	12-04-06	323679

Step 1: Determine the FD's that violate BCNF

Doctor ID → Doctor License Expiry, Employee ID

No FD violates BCNF.

Nurse Table: (Nurse ID, Nurse License Expiry, Employee ID)

NURSE_ID	NURSELICENSE_EXPIRY	EMPLOYEE_ID
555879	25-09-10	323680
555880	22-09-11	323681
555881	23-10-10	323682

Step 1: Determine the FD's that violate BCNF

Nurse ID → NurseLicense Expiry, Employee ID

No FD violates BCNF.

Patient Table: (Healthcard #, First name, Last name, DOB, Gender, Address, Phone, Email, Hospital ID)

HEALTHCA...	F_...	L_NAME	DATE_OF_BIRTH	GENDER	AGE	ADDRESS_STREET	ADDRESS_STREET2	ADDRESS_CITY	ADDRESS_PROVINCE	ADDRESS_POSTALCODE	ADDRESS_COUNTRY	PHONE	EMAIL	HOSPITAL_ID
1234567890	Ann	Smith	97-01-28	F	22	1290 Bayview Rd.	(null)	Toronto	ON	M1B2X4	CAD	6471234567	ann.smith97@gmail.com	123
1234567891	Bob	Brown	00-02-21	M	20	11 Clifton St.	(null)	Toronto	ON	M2C1L5	CAD	6471234568	bob.brown00@hotmail.com	124
1234567892	Carl	Jones	99-07-11	M	21	103 Roadhouse Rd.	(null)	Oakville	ON	L6L2X6	CAD	4162578564	carl.jones99@gmail.com	125
1234567893	Dan	Miller	98-04-07	M	22	56 Steeling Ave.	(null)	Pickering	ON	L1V0A1	CAD	4162874587	dan.miller98@gmail.com	125
1234567894	Eve	Williams	98-01-22	F	22	23 Greentint Cres.	(null)	Markham	ON	L1C3P2	CAD	6478985674	eve.williams22@hotmail.com	124
1234567895	Stan	Murphy	99-01-29	M	30	1111 Bayview Rd.	(null)	Toronto	ON	M1B2X4	CAD	6471212311	stan.murphy@gmail.com	123
1234567896	Alice	Yum	20-03-21	F	21	11 Steeling St.	(null)	Toronto	ON	M2C3CD	CAD	6471234231	yum.alice@hotmail.com	123
1234567897	Steve	Jan	20-07-12	M	23	2 Ins Street	(null)	Oakville	ON	L6L2X	CAD	4162532313	steve.jan@gmail.com	124
1234567898	Mike	Lanny	90-08-07	M	40	55 Steeling Ave.	(null)	Pickering	ON	L1DED	CAD	4162875561	mike.lanny@gmail.com	125
1234567899	Camy	Hun	32-01-30	F	30	2 Hopkins Street	(null)	Markham	ON	L1QD2P	CAD	6478983164	camy.hun@hotmail.com	123

Step 1: Determine the FD's that violate BCNF

Healthcard # → First name, Last name, DOB, Gender, Age, Address, Phone, Email, Hospital ID)

Step 2: Decompose the tables

Now we have…
R1(DOB,Age)
R2(Heatlhcard #, First name, Last name, DOB, Gender, Address, Phone, Email, Hospital ID)

Now, both of these tables are in BCNF.

Appointment Table: (Appointment ID, Appointment date, Appointment time, Room #, Nurse ID, Healthcard #, Doctor ID, Hospital ID)

APPOINTMENT_ID	APPOINTMENT_DATE	APPOINTMENT_TIME	ROOM_NO	NURSE_ID	HEALTHCARD_NO	DOCTOR_ID	HOSPITAL_ID
10031	20-12-12	1300	1	555879	1234567890	141524	123
10012	20-10-20	1400	2	555880	1234567897	141526	124
10013	20-10-22	1500	3	555881	1234567892	141525	125

Step 1: Determine the FD's that violate BCNF

Appointment ID → Appointment date, Appointment time, Room #, Nurse ID, Healthcard #, Doctor ID, Hospital ID

No FD violates BCNF.

Diagnosis Table: (Diagnosis ID, Results, Appointment ID)

	DIAGNOSIS_ID	RESULTS	APPOINTMENT_ID
1	123456	Stomach cancer	10012
2	123457	Covid-19	10013

Step 1: Determine the FD's that violate BCNF
Diagnosis ID → Results, Appointment ID
No FD violates BCNF.

Invoice Table: (Invoice #, Medicine ID, Date Issued, Amount owed, appointment ID)

INVOICE_NO	MEDICINE_ID	DATE_ISSUED	AMOUNT_OWED	APPOINTMENT_ID
111112	3	20-10-20	100	10012
111113	3	20-10-22	10	10013

Step 1: Determine the FD's that violate BCNF

Invoice # → Medicine ID, Date issued, amount owed, appointment ID

No FD violates BCNF.

Prescription Table: (Prescription #, Appointment ID, Medicine ID, Diagnosis ID)

PRESCRIPTION_NO	DATE_ISSUED	APPOINTMENT_ID	MEDICINE_ID	DIAGNOSIS_ID
1231231	20-10-20	10012	191919	123456
1231232	20-10-22	10013	191919	123457
1231233	20-10-22	10013	191920	123457

Step 1: Determine the FD's that violate BCNF

Prescription # → Date Issued, Medicine ID, Diagnosis ID, Appointment ID

No FD violates BCNF.

Medical History Table: (Healthcard #, Appointment ID, Diagnosis ID, Medical Desc)

HEALTHCARD_NO	APPOINTMENT_ID	DIAGNOSIS_ID	MEDICAL_DESC
1234567897	10012	123456	This guy has stomach cancer, he is allergic to tylonel so prescribe him something else
1234567892	10013	123457	Make sure they stay at home for two weeks

Step 1: Determine the FD's that violate BCNF

Healthcard #, Appointment ID, Diagnosis ID → Medical Desc

No FD violates BCNF.

Medicine Table: (Medicine ID, Dosage, IUPAC Name, Generic Name, Inventory, Price, Expiration date, Manufacturer, Hospital ID)

	MEDICINE_ID	DOSAGE	IUPAC_NAME	GENERIC_NAME	INVENTORY	PRICE	EXPIRATION_DATE	MANUFACTURER	HOSPITAL
1	191919	20	N-(4-hydroxyphenyl)acetamide	acetaminophen	500	5	30-01-01	Tylenol	
2	191920	100	8-Chloro-1-methyl-6-phenyl-4H-[1,2,4]triazolo[4,3-a][1,4]benzodiazepine	alprazolam	200	50	20-12-25	Tylenol	
3	191921	10	(S,S)-2-methylamino-1-phenylpropan-1-ol	pseudoephedrine	100	50	30-10-22	Johnson and Johnson	
4	191922	50	(S)-2-Amino-3-[4-(4-hydroxy-3,5-diiodophenoxy)-3,5-diiodophenyl]propanoic acid	synthroid	50	25.5	29-06-01	Johnson and Johnson	
5	191923	50	(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid	lipitor	200	60	20-12-01	Johnson and Johnson	
6	191924	75	(S,S)-2-methylamino-1-phenylpropan-1-ol	pseudoephedrine	100	48.75	29-05-02	Sinopharm	
7	191925	125	(3R,5R)-7-[2-(4-Fluorophenyl)-3-phenyl-4-(phenylcarbamoyl)-5-propan-2-ylpyrrol-1-yl]-3,5-dihydroxyheptanoic acid	lipitor	1200	62.5	27-12-01	Sinopharm	

Step 1: Determine the FD's that violate BCNF

Medicine ID → Dosage, IUPAC Name, Generic name, Inventory, Price, Expiration date, Manufacturer, Hospital ID

Hospital ID → Inventory, Price
IUPAC Name → Generic Name

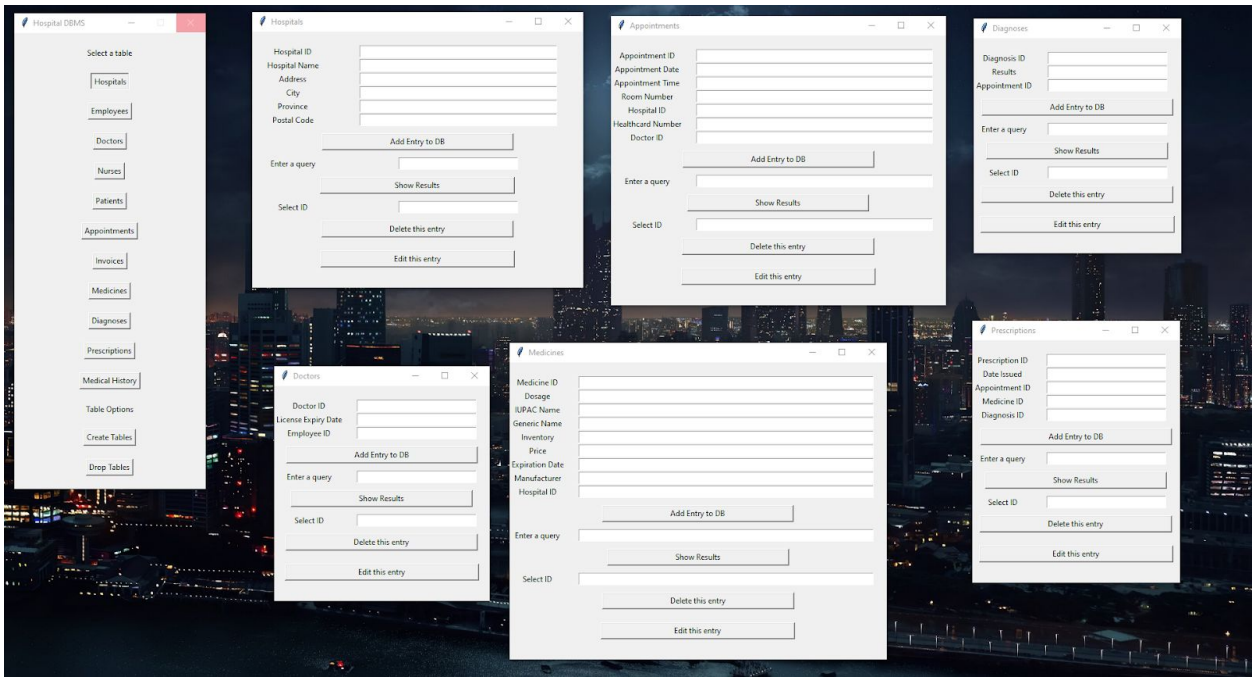
The latter two FD's violate BCNF

Step 2: Decompose the Tables
R1(Hospital ID, Inventory, Price)
R2(Medicine ID, Dosage, IUPAC Name, Generic Name, Expiration date, manufacturer, Hospital ID)
R3(IUPAC Name, generic name)
R4(Medicine ID, Dosage, IUPAC Name, Expiration date, Manufacturer, Hospital ID)

So finally, the tables decomposed into BCNF are R1, R3, R4. We omit R2 since it is a subset of R4.

In conclusion, all tables are now BCNF/3NF since all of their attributes are independent of one another, and depend ONLY on their respective tables Primary Key.

Assignment 9: Demonstration of Application by Java/Web Based UI



Code: https://github.com/anthony-trinh/tkinter-sqlite-guis

Assignment 10: Relational Algebra Notation

Hospital Table

English	SQL	RA
get all the existing hospitals	SELECT * FROM hospital;	hospital
get hospitals located in toronto	SELECT * FROM hospital WHERE address_city = 'Toronto';	$\pi_{\text{hospital\_id, address\_city}}(\sigma_{\text{address\_city = "Toronto"}}(\text{hospital}))$

Employee Table

English	SQL	RA
get all the existing employees	SELECT * FROM employee;	employee
get employees in hospital 123	SELECT * FROM employee WHERE hospital_id = 123;	$\pi_{\text{employee\_id, hospital\_id}}(\sigma_{\text{hospital\_id = 123}}(\text{employee}))$

Doctor Table

English	SQL	RA
get all the existing doctors	SELECT * FROM doctor;	doctor
find doctor_id from doctors with ID's that expired in 2020	SELECT doctor_id FROM doctor WHERE doctorlicense_expiry < to_date('01/01/2020','mm/dd/yyyy');	$\pi_{\text{doctor\_id, doctorlicense\_expiry}}(\sigma_{\text{doctorlicense\_expiry < to\_date('01/01/2020','mm/dd/yyyy')}}(\text{doctor}))$

Nurse Table

English	SQL	RA
get all the existing nurses	SELECT * FROM nurse;	nurse
find nurse_id from nurses with ID's that expire in the next 5 years	SELECT nurse_id FROM nurse WHERE nurselicense_expiry < to_date('01/01/2025','mm/dd/yyyy');	$\pi_{\text{nurse\_id, nurselicense\_expiry}}(\sigma_{\text{doctorlicense\_expiry < to\_date('01/01/2025','mm/dd/yyyy')}}(\text{nurse}))$

Patient Table

English	SQL	RA
get all the existing patients	SELECT * FROM patient;	patient
find all the women age 22 from toronto	SELECT * FROM patient WHERE gender = 'F' AND address_city = 'Toronto' AND age < 22 ;	$\pi_{\text{healthcard\_no, age, gender, adress\_city}}$ ($\sigma_{\text{gender = 'F' AND address\_city = 'Toronto' AND age < 22}}$ (patient))

Appointment Table

English	SQL	RA
get all the existing appointments	SELECT * FROM appointment ;	appointment
find appointments taken during the November 18 to 25, in the year 2020	SELECT * FROM appointment WHERE appointment_date >= to_date('10/18/2020', 'mm/dd/yyyy') AND appointment_date < to_date('10/25/2020','mm/dd/yyyy') ;	$\pi_{\text{appointment\_id, healthcard\_no}}$($\sigma_{\text{appointment\_date >= to\_date('10/18/2020', 'mm/dd/yyyy') AND appointment\_date < to\_date('10/25/2020','mm/dd/yyyy')}}$(appointment))

Invoice Table

English	SQL	RA
get all the existing invoices	SELECT * FROM invoice;	invoice
find all invoices owing 100 or more	SELECT * FROM invoice WHERE amount_owed >= 100 ;	$\pi_{\text{invoice\_no, amount\_owed}}$($\sigma_{\text{amount\_owed >= 100}}$(invoice))

Diagnosis Table

English	SQL	RA
get all the existing	SELECT * FROM diagnosis;	diagnosis

diagnoses		
find all diagnoses that include cancer	SELECT * FROM diagnosis WHERE results LIKE '%cancer%' ;	$\pi_{diagnosis\_id,\ results}(\sigma_{results\ LIKE\ '\%cancer\%'}(diagnosis))$

Medicine Table

English	SQL	RA
get all the existing medicine	SELECT * FROM medicine ;	medicine
find all drugs made by johnson and johnson	SELECT * FROM medicine WHERE manufacturer='Johnson and Johnson' ;	$\pi_{medicine\_id,\ manufacturer}(\sigma_{manufacturer\ =\ 'Johnson\ and\ Johnson'}(medicine))$

Prescription Table

English	SQL	RA
get all the existing appointments	SELECT * FROM prescription ;	prescription
find all prescriptions for medicineId='191919' and sort them descending by dosage	SELECT * FROM prescription WHERE medicine_id = '191919' ORDER BY dosage DESC ;	$\pi_{perscription\_id,\ medicine\_id}(\sigma_{medicine\_id\ =\ '191919'}(prescription))$

Medical History Table

English	SQL	RA
get all the existing medical histories	SELECT * FROM medical_history ;	medical_history

-- ADVANCED QUERIES

-- JOIN queries

English	SQL	RA
gets email from given name in appointment	SELECT email FROM appointment a, patient p WHERE p.f_name = 'Ann' AND p.l_name = 'Smith' AND a.healthcard_no = p.healthcard_no;	$\pi_{appointment\_id,\ healthcard\_no,\ email}(\sigma_{p.f\_name = 'Ann'}(\sigma_{p.l\_name = 'Smith'}(Appointment \|\| Patient))$
get first name of patient from appointment 10031	SELECT f_name FROM appointment a, patient p WHERE a.appointment_id = 10031 AND a.healthcard_no = p.healthcard_no; $\pi_{f\_name}(\sigma_{appointment\_id = 10031}(Appointment \|\| Patient))$	$\pi_{f\_name}(\sigma_{appointment\_id = 10031}(Appointment \|\| Patient))$

-- EXISTS queries, implements intersection

English	SQL	RA
get mentioned attributes from existing tables	SELECT generic_name, inventory, price, expiration_date, manufacturer FROM medicine WHERE EXISTS (SELECT prescription_no FROM prescription	$\pi_{generic\_name,\ inventory,\ price,\ expiration\_date,\ manufacturer}(\sigma$ $EXISTS(\pi_{prescription}(Prescription \|\| Patient))$

	WHERE prescription.medicine_id = medicine.medicine_id);	

-- UNION queries

English	SQL	RA
get all the nurses and doctors	SELECT employee_id FROM nurse UNION SELECT employee_id FROM doctor;	nurse ∪ doctor
get all hospital cities and patient address cities	SELECT address_city FROM hospital UNION SELECT address_city FROM patient ORDER BY address_city ;	$\pi_{hospital\_id,\ address\_city}(\sigma_{address\_city}$ (hospital)) ∪ $\pi_{healthcard\_no,\ address\_city}(\sigma_{address\_city}$ (patient))

-- MINUS queries

English	SQL	RA
get all the employees not working in Toronto General Hospital	SELECT * FROM employee MINUS (SELECT e.* FROM employee e, hospital h WHERE h.hospital_name = 'Toronto General Hospital' AND h.hospital_id = e.hospital_id);	employee - $\sigma_{hospital\_name = \text{'Toronto General Hospital'}}$ (employee)

-- COUNT

English	SQL	RA

Get total numbers of patients where their first name is ‘John’	SELECT 'Total number of patients: ', COUNT(healthcard_no) WHERE f_name = ‘John’ FROM patient ;	$\pi_{COUNT(health\_no)}(\sigma_{f\_name = \text{‘John’}}(Patient))$

Conclusion

The document summarizes all the processes taken of our Hospital Database. It utilizes SQL and the fundamentals of database management in order to manage and display the information required to operate a hospital. Many layers of normal forms were explored which were needed to normalize the tables storing information on important components of a hospital such as the doctors, patients, prescriptions, and appointments.The SQL queries made during the course allows for easy retrieval for key information. These queries were later rewritten using Relational Algebra notation to yield instances of relations as output. Overall, our database has evolved from a database with a beak ER diagram, to a normalized and standardized one. Nevertheless, this project has given our group the skills required to build any database system.