WolfHospital Management System

For a hospital in North Carolina

CSC 540 - Database Systems

Project Report #2

Project Team U
Niharika Acharya, Ankit Manendra, Hasham Mukhtar, Vaibhav Nagvekar

Project Assumptions

1. SSN doesn't imply name in Patients in our design because SSN is optional so it's not guaranteed that there will always be an SSN in that table
2. Beds will always have a ward associated with them
3. There can be emergency check ins where patients don't get assigned a ward or bed and get treatment right away
    a. This means ward id, and bed id for check in info can be null
4. Sometimes a patient comes in to the hospital and doesn't need a specific treatment so it can be null in the database for medical records
5. Sometimes a patient comes in and doesn't need any tests, so the test for patients results can be null
6. Only doctor roles can have a department and professional title associated with them
7. Patients can't use this system, only hospital staff (doctors, nurse, reception, billing, admins)
8. A reserved bed is one where the reserveBed boolean value is true for a bed in the table, but there is not a patient's name
9. An assigned bed is one where the reserveBed boolean value is true and there is a patient's id for that bed
10. Ward usage is the number of occupied beds in that ward out of the total beds available for that ward
11. Bed usage is the total number of beds occupied in the hospital out of the total beds available in the hospital
12. Beds are required to have a ward number associated with them
13. The total number of beds in the beds table will equal the total capacity of all the wards in the wards table
    a. If new wards are added, new beds are added to the table
14. We have 5 users
    a. Doctors, Nurses, Reception Staff, Billing Staff and Admins
15. Admins have access to everything in the systems and can create/edit all information.
16. We have added the auto increment feature in all the tables separately as we missed on adding this while creating the table itself.
17. We are breaking down billing records into fee charge, test charge, treatment charge and ward charge
18. The fees charge attribute in the billing account table is the combined fees of consultation, medication, and registration and is saved as one number in our system.

Global Database Schema

Staff(Sid, name, age, gender, job title, phone, address, department, professional title)

- Functional Dependencies
  - Sid → name, age, gender, job title, phone, address, department, professional title
- This is in BCNF (thus in 3NF) form

Patients(Pid, SSN, name, dob, gender, age, phone, address, status, ward_id)

- Functional Dependencies
  - Pid → SSN, name, dob, gender, age, phone, address, status, ward_id
  - DOB → age
- This relation is not in 3NF so we created these two relations that are both in BCNF (thus 3NF)
  - DateOfBirth (dob, age)
  - Patients (Pid, SSN, name, dob, gender, phone, address, status, ward_id)

MedicalRecords(medical_record_id, patient_id, startDate, endDate, prescription, doctor_id, diagnosis, treatment, active)

- Functional Dependencies
  - medical_record_id→ patient_id, startDate, endDate, prescription, doctor_id, diagnosis, treatment, active
  - patient_id,StartDate →medical_record_id, endDate, prescription, doctor_id, diagnosis, treatment, active
- This relation is in 3NF form

Wards(ward_id, capacity, charges_per_day, nurse_id)

- Functional Dependencies
  - Ward_id → capacity, charges_per_day, nurse_id
- This relation is in BCNF (thus 3NF) form

Beds(bed_id, ward_id, Pid, reserved)

- Functional Dependencies
  - Bed_id → ward_id, Pid, reserved
- This relation is in BCNF (thus 3NF) form

Tests(test_id, name, price)

- Functional Dependencies
  - Test_id → name, price

    - Name → price
- This relation is not in 3NF, so we created two relations that are both in BCN (thus in 3NF) form
    - TestName(name, price)
    - Tests(test_id, name)

Treatments(treatment_id, name, price)

- Functional Dependencies
    - Treatment_id → name, price
    - Name → price
- This relation is not in 3NF, so we created two relations that are both in BCNF (thus 3NF) form
    - TreatmentName(name, price)
    - Treatments(treatment_id, name)

BillingAccounts(billing_account_id, patient_id, SSN, billing address, payment method, card number, check number, insurance company name, ward charge, test charge, treatment charge, fee charge, total charge, start date, end date, settled)

- Functional Dependencies
    - Billing_account_id→ patient_id, SSN, billing address, payment method, card number, check number, insurance company name, ward charge, test charge, treatment charge, fee charge, total charge, start date, end date, settled
    - patient_id,StartDate → billing_account_id, SSN, billing address, payment method, card number, check number, insurance company name, ward charge, test charge, treatment charge, fee charge, total charge, end date, settled
- This relation is in 3NF

Check-in-Information(check_in_id, patient_id, ward_id, bed_id, startDate, endDate)

- Functional Dependencies
    - check_in_id→ patient_id, ward_id, bed_id, startDate, endDate
    - Patient_id, startDate → check_in_id, ward_id, bed_id, endDate
    - Bed_id → ward_id
- This relation is not in 3NF, so we created two relations that are both in BCNF (thus 3NF) form
    - Bed(bed_id, ward_id)
    - Check-in-Information(check_in_id, patient_id, bed_id, startDate, endDate)

Tests_For_Patients(medical_record_id, test_id, result)

- Functional Dependencies
    - medical_record_id,test_id → result
- This relation is in BCNF (thus 3NF) form

Design Decisions & Constraints

- For each entity, we created a relations with the associated attributes
- Any exactly one or up to 1 relationships from the ER diagrams became attributes for the entities those arrows were coming from
    - Patients has a ward_id attribute for this reason
- We decided to not create separate tables for each type of user (doctor, nurse, reception, billing, admin)
    - We used the null translation for the subclasses of staff from project report 1
    - We can use the jobTitle attribute to specify a certain user
- All the constraints are listed below for each relation

Staff(<u>Sid</u>, name, age, gender, job title, phone, address, department, professional title)

- **Keys**
    - Sid
- **NULL**
    - Department
    - Professional titles
    - Amongst all the staff, only doctors have these two attributes and would be null for others.
- **NOT NULL**
    - Name, age, gender, job title, phone, address
- **Referential Integrity**
    - None

Patients(<u>Pid</u>, SSN, name, dob, gender, age, phone, address, status, ward_id)

- **Keys**
    - Pid : Unique identifier
- **NULL**
    - SSN
    - SSN is optional and can be left blank while creating a patient data.
- **NOT NULL**
    - name, dob, gender, age, phone, address, status
    - These can not be left blank for every patient.
- **Referential Integrity**

- None

MedicalRecords(<u>medical_record_id</u>, patient_id, startDate, endDate, prescription, doctor_id, diagnosis, treatment, active)

- **Keys**
  - Medical_record_id : Unique identifier
- **NULL**
  - treatment
  - diagnosis
  - endDate
  - prescription
  - doctor_id
  - When a patient checks in, a new medical record data would be created. However these columns can be left blank at that time which would be filled as and when needed in the future.
- **NOT NULL**
  - patient_id, startDate, active
  - These are the bare minimum details that would be needed to create a new medical record entry.
- **Referential Integrity**
  - Patient_id refers to pid of patients table
  - Doctor_id refers to sid of staff table
  - Treatment refers to treatment_id of treatment table.

Wards(<u>ward_id</u>, capacity, charges_per_day, nurse_id)

- **Keys**
  - Ward_id: Unique identifier
- **NULL**
  - Nurse_id
  - When creating a ward, a nurse might not be appointed at that time. This can be updated in future.
- **NOT NULL**
  - capacity, charges_per_day
  - These needs to be specified to set up a new ward.
- **Referential Integrity**

- Nurse_id refers to sid of staff table.

Beds(<u>bed_id</u>, ward_id, Pid, reserved)

- **Keys**
  - Bed_id: Unique identifier
- **NULL**
  - Pid
  - Some of the beds might be empty and consequently the patient id would be blank for them.
- **NOT NULL**
  - Ward_id, reserved
  - Every bed needs to be linked to ward and it's status shouldn't be blank.
- **Referential Integrity**
  - Ward_id refers to ward_id of wards table.
  - Pid refers to pid of patients table.

Tests(<u>test_id</u>, name, price)

- **Keys**
  - Test_id: Unique identifier
- **NULL**
  - None
- **NOT NULL**
  - name, price
  - All types of test needs to have a name and their price to either identify them or use their price for billing purposes.
- **Referential Integrity**
  - None

Treatments(<u>treatment_id</u>, name, price)

- **Keys**
  - Treatment_id: Unique identifier
- **NULL**
  - None
- **NOT NULL**
  - Name, price

    - All types of treatments needs to have a name and their price to either identify them or use their price for billing purposes.
- **Referential Integrity**
    - None

BillingAccounts(<u>billing_account_id</u>, patient_id, SSN, billing address, payment method, card number, check number, insurance company name, ward charge, test charge, treatment charge, fee charge, total charge, start date, end date, settled)

- **Keys**
    - Billing_account_id: Unique identifier
- **NULL**
    - Card number
    - Check number
    - billing address
    - payment method
    - insurance company name
    - ward charge, test charge
    - treatment charge
    - fee charge
    - total charge
    - end date
    - SSN (of person paying)
    - When a new patient is checked in, a corresponding billing account would be created for them. However, all of these details wouldn't be available at that time. These can be left blank and would be updated at the time of check out.
- **NOT NULL**
    - patient_id, start date, settled
    - We need at-least these 3 attributes to identify a patient's billing account while creating them.
- **Referential Integrity**
    - Patient_id refers to pid of patients table.

Check-in-Information(<u>check_in_id,</u> patient_id, ward_id, bed_id, startDate, endDate)

- **Keys**
    - Check_in_id: Unique identifier

- **NULL**
    - EndDate
    - Ward_id
    - Bed_id
    - There could be emergency check ins where a patient gets treatment right away
    - Just as when a patient is checked in the hospital, all these details wouldn't be available immediately and hence could store null values.
- **NOT NULL**
    - patient_id, startDate
    - To identify a check in information for a patient, these needs to be linked immediately while creating an entry for the same.
- **Referential Integrity**
    - Patient_id refers to the pid of patients table.
    - Ward_id refers to the ward_id of wards table.
    - Bed_id refers to the bed_id of beds table.

Test_For_Patients(medical_record_id, test_id, result)

- **Keys**
    - Medical_record_id, test_id: These two together serves as unique identifier
- **NULL**
    - Result
    - Result for a test wouldn't be available initially and can be left blank.
- **NOT NULL**
    - None
- **Referential Integrity**
    - Medical_record_id refers to the medical_record_id of medical records table.
    - Test_id refers to the test_id of tests table.

Initial SQL Statements

CREATE TABLE Statements

Staff(<u>Sid</u>, name, age, gender, job title, phone, address, department, professional title)

```
CREATE TABLE staff
(
      id INT PRIMARY KEY AUTO_INCREMENT =1 ,
      name VARCHAR(255) NOT NULL,
      age INT NOT NULL,
      gender VARCHAR(5) NOT NULL,
      jobTitle varchar(255) NOT NULL,
      phone INT NOT NULL,
      address VARCHAR(255) NOT NULL,
      department VARCHAR(255),
      professionalTitle VARCHAR(255)
);
```

```
MariaDB [amanend]> CREATE TABLE staff
    -> (
    -> id INT PRIMARY KEY,
    -> name VARCHAR(255) NOT NULL,
    -> age INT NOT NULL,
    -> gender VARCHAR(5) NOT NULL,
    -> jobTitle varchar(255) NOT NULL,
    -> phone INT NOT NULL,
    -> address VARCHAR(255) NOT NULL,
    -> department VARCHAR(255),
    -> professionalTitle VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.32 sec)
```

```
MariaDB [amanend]> ALTER TABLE staff add id INT PRIMARY KEY AUTO_INCREMENT =
1;
Query OK, 1 row affected (0.00 sec)
```

Patients(<u>Pid</u>, SSN, name, dob, gender, age, phone, address, status, ward_id)

```
CREATE TABLE patients
(
      id INT PRIMARY KEY AUTO_INCREMENT =1,
      ssn INT,
      name VARCHAR(255) NOT NULL,
      dateOfBirth DATE NOT NULL,
      gender VARCHAR(5) NOT NULL,
      phone INT NOT NULL,
      address VARCHAR(255) NOT NULL,
      status VARCHAR(255) NOT NULL,
```

```
      ward_id INT,
      CONSTRAINT ward_id_fk FOREIGN KEY(ward_id) REFERENCES wards(id),
      age INT NOT NULL
);
```

```
MariaDB [amanend]> CREATE TABLE patients
    -> (
    -> id INT PRIMARY KEY AUTO_INCREMENT =1,
    -> ssn INT,
    -> name VARCHAR(255) NOT NULL,
    -> dateOfBirth DATE NOT NULL,
    -> gender VARCHAR(5) NOT NULL,
    -> phone INT NOT NULL,
    -> address VARCHAR(255) NOT NULL,
    -> status VARCHAR(255) NOT NULL,
    -> ward_id INT,
    -> CONSTRAINT ward_id_fk FOREIGN KEY(ward_id) REFERENCES wards(id),
    -> age INT NOT NULL
    -> );
Query OK, 0 row affected (0.07 sec)
```

MedicalRecords(<u>medical_record_id</u>, patient_id, startDate, endDate, prescription, doctor_id, diagnosis, treatment, active)

```
CREATE TABLE medical_records
(
      id INT PRIMARY KEY AUTO_INCREMENT =1,
      patient_id INT NOT NULL,
      CONSTRAINT ba_pat_id_fk FOREIGN KEY(patient_id) REFERENCES
patients(id),
      start_date DATE NOT NULL,
      end_date DATE,
      prescription VARCHAR(255),
      doctor_id INT,
      CONSTRAINT doctor_id_fk FOREIGN KEY(doctor_id) REFERENCES staff(id),
      diagnosis VARCHAR(255,
      treatment INT,
      CONSTRAINT treatment_id_fk FOREIGN KEY(treatment) REFERENCES
treatments(id),
      active BOOLEAN NOT NULL
);
```

```
MariaDB [amanend]> CREATE TABLE medical_records
    -> (
    -> id INT PRIMARY KEY,
    -> patient_id INT,
    -> CONSTRAINT pat_id_fk FOREIGN KEY(patient_id) REFERENCES patients(id) ON DELETE CASCADE,
    -> start_date DATE NOT NULL,
    -> end_date DATE NOT NULL,
    -> prescription VARCHAR(255) NOT NULL,
    -> doctor_id INT,
    -> CONSTRAINT doctor_id_fk FOREIGN KEY(doctor_id) REFERENCES staff(id) ON DELETE CASCADE,
    -> diagnosis VARCHAR(255) NOT NULL,
    -> treatment INT,
    -> CONSTRAINT treatment_id_fk FOREIGN KEY(treatment) REFERENCES treatments(id) ON DELETE CASCADE,
    -> active BOOLEAN NOT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
MariaDB [amanend]> alter table medical_records modify patient_id int not null;
Query OK, 7 rows affected (0.02 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

```
MariaDB [amanend]> ALTER TABLE medical_records add id INT PRIMARY KEY
AUTO_INCREMENT = 1;
Query OK, 1 row affected (0.00 sec)
```

Tests(<u>test_id</u>, name, price)

```
CREATE TABLE tests
(
      id INT PRIMARY KEY AUTO_INCREMENT =1,
      name VARCHAR(255) NOT NULL,
      price FLOAT NOT NULL
);
```

```
MariaDB [amanend]> CREATE TABLE tests
    -> (
    -> id INT PRIMARY KEY,
    -> name VARCHAR(255) NOT NULL,
    -> price INT NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
MariaDB [amanend]> ALTER TABLE tests add id INT PRIMARY KEY AUTO_INCREMENT =
1;
Query OK, 1 row affected (0.00 sec)
```

Treatments(<u>treatment_id</u>, name, price)

```
CREATE TABLE treatments
(
      id INT PRIMARY KEY AUTO_INCREMENT =1,
      name VARCHAR(255) NOT NULL,
      price FLOAT NOT NULL
);
```

```
MariaDB [amanend]> CREATE TABLE treatments ( id INT PRIMARY KEY, name VARCHAR(255) NOT NULL, price INT NOT NULL );
Query OK, 0 rows affected (0.08 sec)
```

```
MariaDB [amanend]> ALTER TABLE treatments add id INT PRIMARY KEY
AUTO_INCREMENT = 1;
Query OK, 1 row affected (0.00 sec)
```

BillingAccounts(<u>billing_account_id</u>, patient_id, SSN, billing address, payment method, card number, check number, insurance company name, ward charge, test charge, treatment charge, fee charge, total charge, start date, end date, settled)

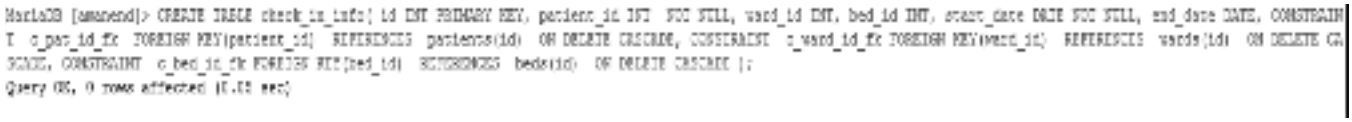
```
CREATE TABLE billing_accounts
(
      id INT PRIMARY KEY AUTO_INCREMENT =1,
      patient_id INT NOT NULL,
      CONSTRAINT pat_id_fk FOREIGN KEY(patient_id) REFERENCES patients(id),
      ssn INT,
      billing_address VARCHAR(255),
      payment_method VARCHAR(255),
      card_number INT,
      check_number INT,
      insurance_company_name VARCHAR(255),
      ward_charge FLOAT,
      test_charge FLOAT,
      treatment_charge FLOAT,
      fee_charge FLOAT,
      total_charge FLOAT,
      start_date DATE NOT NULL,
      end_date DATE,
      settled BOOLEAN NOT NULL
);
```

```
MariaDB [amanend]> CREATE TABLE billing_accounts
    -> (
    -> id INT PRIMARY KEY,
    -> patient_id INT NOT NULL,
    -> CONSTRAINT pat_id_fk FOREIGN KEY(patient_id) REFERENCES patients(id) ON DELETE CASCADE,
    -> ssn INT,
    -> billing_address VARCHAR(255),
    -> payment_method VARCHAR(255),
    -> card_number INT,
    -> check_number INT,
    -> insurance_compmany_name VARCHAR(255),
    -> ward_charge INT,
    -> test_charge INT,
    -> treatment_charge INT,
    -> fee_charge INT,
    -> total_charge INT,
    -> start_date DATE NOT NULL,
    -> end_date DATE,
    -> settled BOOLEAN NOT NULL
    -> );
```

```
MariaDB [amanend]> ALTER TABLE billing_accounts add id INT PRIMARY KEY
AUTO_INCREMENT = 1;
Query OK, 1 row affected (0.00 sec)
```

Check-in-Information(check_in_id, patient_id, ward_id, bed_id, startDate, endDate)

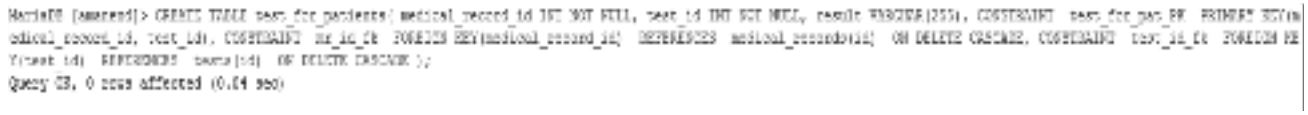
```
CREATE TABLE check_in_info
(
      id INT PRIMARY KEY AUTO_INCREMENT =1,
      patient_id INT  NOT NULL,
      ward_id INT,
      bed_id INT,
      start_date DATE NOT NULL,
      end_date DATE,
      CONSTRAINT  c_pat_id_fk  FOREIGN KEY(patient_id)  REFERENCES
patients(id),
      CONSTRAINT  c_ward_id_fk FOREIGN KEY(ward_id)  REFERENCES  wards(id) ,
      CONSTRAINT  c_bed_id_fk FOREIGN KEY(bed_id)  REFERENCES  beds(id)
);
```




```
MariaDB [amanend]> ALTER TABLE check_in_info add id INT PRIMARY KEY
AUTO_INCREMENT = 1;
Query OK, 1 row affected (0.00 sec)
```

Tests_For_Patients(medical_record_id, test_id, result)

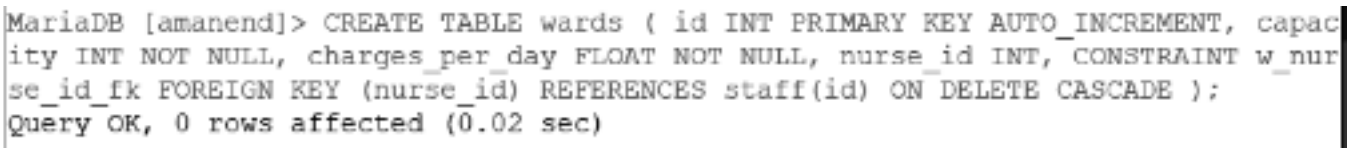
```
CREATE TABLE test_for_patients
(
      medical_record_id INT NOT NULL ,
      test_id INT NOT NULL,
      result VARCHAR(255),
      CONSTRAINT  test_for_pat_PK  PRIMARY KEY(medical_record_id, test_id),
      CONSTRAINT  mr_id_fk  FOREIGN KEY(medical_record_id)  REFERENCES
medical_records(id),
      CONSTRAINT  test_id_fk  FOREIGN KEY(test_id)  REFERENCES  tests(id)
);
```



```
MariaDB [amanend]> ALTER TABLE test_for_patients add id INT PRIMARY KEY
AUTO_INCREMENT = 1;
Query OK, 1 row affected (0.00 sec)
```

Wards(<u>ward_id</u>, capacity, charges_per_day, nurse_id)

```
CREATE TABLE wards
(
      id INT PRIMARY KEY AUTO_INCREMENT,
      capacity INT NOT NULL,
      charges_per_day FLOAT NOT NULL,
      nurse_id INT,
      CONSTRAINT w_nurse_id_fk FOREIGN KEY(nurse_id) REFERENCES staff(id)
);
```



```
MariaDB [amanend]> CREATE TABLE wards ( id INT PRIMARY KEY AUTO_INCREMENT, capac
ity INT NOT NULL, charges_per_day FLOAT NOT NULL, nurse_id INT, CONSTRAINT w_nur
se_id_fk FOREIGN KEY (nurse_id) REFERENCES staff(id) ON DELETE CASCADE );
Query OK, 0 rows affected (0.02 sec)
```

```
MariaDB [amanend]> ALTER TABLE wards add id INT PRIMARY KEY AUTO_INCREMENT =
1;
Query OK, 1 row affected (0.00 sec)
```

Beds(<u>bed_id</u>, ward_id, Pid, reserved)

```
CREATE TABLE beds
(
      id INT PRIMARY KEY AUTO_INCREMENT =1,
      ward_id INT,
      patient_id INT,
      reserved BOOLEAN NOT NULL,
      CONSTRAINT  b_ward_id_fk FOREIGN KEY(ward_id)  REFERENCES  wards(id),
      CONSTRAINT  b_pat_id_fk FOREIGN KEY(patient_id)  REFERENCES
      patients(id)
);
```

```
MariaDB [amanend]> CREATE TABLE beds(
    -> id INT PRIMARY KEY,
    -> ward_id INT,
    -> patient_id INT,
    -> reserved BOOLEAN NOT NULL,
    -> CONSTRAINT  b_ward_id_fk FOREIGN KEY(ward_id)  REFERENCES  wards(id)  ON DELETE CASCADE,
    -> CONSTRAINT  b_pat_id_fk FOREIGN KEY(patient_id)  REFERENCES  patients(id)  ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [amanend]> ALTER TABLE beds add id INT PRIMARY KEY AUTO_INCREMENT =
1;
Query OK, 1 row affected (0.00 sec)
```

INSERT Statements

Staff(name, age, gender, job title, phone, address, department, professional title)

```
INSERT INTO staff (name, age, gender, jobTitle, phone, address, department,
professionalTitle)
VALUES('Doctor 1', 45, 'M', 'doctor', 1234567890, '123 East Street, Raleigh,
NC 12345', 'Oncology Department', 'Senior Surgeon');

INSERT INTO staff (name, age, gender, jobTitle, phone, address)
VALUES('Nurse 1', 25, 'F', 'nurse', 4567891230, '123 West Street, Raleigh,
NC 12345');
INSERT INTO staff (name, age, gender, jobTitle, phone, address)
VALUES('Nurse 2', 30, 'M', 'nurse', 9085461256, '123 North Street, Raleigh,
NC 12345');

INSERT INTO staff (name, age, gender, jobTitle, phone, address)
```

```
VALUES(‘Reception Staff 1’, 21, ‘F’, ‘reception staff’, 9088961478, ‘123
South Street, Raleigh, NC 12345’);

INSERT INTO staff (name, age, gender, jobTitle, phone, address)
VALUES(‘Billing Staff 1’, 28, ‘M’, ‘billing staff’, 8564261258, ‘123
SouthWest Street, Raleigh, NC 12345’);

INSERT INTO staff
(name,age,gender,jobTitle,phone,address,department,professionalTitle) VALUES
('Admin 1',50,'F','admin',5546578765,'Gorman
Street,Raleigh','Administration','Senior admin');
```

Wards(capacity, charges_per_day, nurse_id)

```
INSERT INTO wards  (capacity, charges_per_day, nurse_id)
VALUES (1, 100.00, 2);

INSERT INTO wards  (capacity, charges_per_day, nurse_id)
VALUES (2, 125.00, 2);

INSERT INTO wards  (capacity, charges_per_day, nurse_id)
VALUES (3, 150.00, 3);

INSERT INTO wards  (capacity, charges_per_day, nurse_id)
VALUES (4, 200.00, 3);
```

Patients

```
INSERT INTO patients (ssn, name, dateOfBirth, gender, phone, address,
status, ward_id, age) VALUES (078051120,'Jimi
Hendrix','2019-02-17','M',9199549231,'2345-Avery Close,Raleigh','Processing
Treatment Plan',1,43);

INSERT INTO patients (ssn, name, dateOfBirth, gender, phone, address,
status, ward_id, age) VALUES (076052190,'Shannon
Henry','2019-02-02','F',9199249632,'Gorman St.,Raleigh','In Ward',3,32);
```

```
INSERT INTO patients (ssn, name, dateOfBirth, gender, phone, address,
status, ward_id, age) VALUES (041062193,'David
Gilmour','2019-01-03','M',9299549231,'Park Av., Charlotte','In Ward',2,48);

INSERT INTO patients (ssn, name, dateOfBirth, gender, phone, address,
status, ward_id, age) VALUES (721012190,'Zoey
King','2019-01-27','F',9299249739,'Hillsborough St., Raleigh','Completing
Treatment',2,23);

INSERT INTO patients (ssn, name, dateOfBirth, gender, phone, address,
status, ward_id, age) VALUES (321042191,'Raj
Verma','2018-12-22','M',9233249737,'Hillsborough St., Raleigh','Completing
Treatment',3,29);
```

Beds

```
INSERT INTO beds (ward_id,patient_id,reserved) Values (1,1,1);
INSERT INTO beds (ward_id,patient_id,reserved) Values (2,3,1);
INSERT INTO beds (ward_id,patient_id,reserved) Values (2,4,1);
INSERT INTO beds (ward_id,patient_id,reserved) Values (3,2,1);
INSERT INTO beds (ward_id,patient_id,reserved) Values (3,5,1);
INSERT INTO beds (ward_id,reserved) Values (3,0);
INSERT INTO beds (ward_id,reserved) Values (4,0);
INSERT INTO beds (ward_id,reserved) Values (4,0);
INSERT INTO beds (ward_id,reserved) Values (4,0);
INSERT INTO beds (ward_id,reserved) Values (4,0);
```

Medical Records

```
INSERT INTO medical_records
(patient_id,start_date,end_date,prescription,doctor_id,diagnosis,treatment,a
ctive) VALUES (1,'2003-01-01','2008-09-08','Malarone',1,'Positive CBC and
Malaria blood test',6,0);

INSERT INTO medical_records
(patient_id,start_date,end_date,prescription,doctor_id,diagnosis,treatment,a
```

```
ctive) VALUES (3,'2017-08-13','2017-09-13','Metformin',1,'High blood sugar
level and low blood cell count',2,0);

INSERT INTO medical_records
(patient_id,start_date,prescription,doctor_id,diagnosis,treatment,active)
VALUES (1,'2018-04-06','Metformin',1,'High blood sugar level',3,1);

INSERT INTO medical_records
(patient_id,start_date,prescription,doctor_id,diagnosis,treatment,active)
VALUES (3,'2018-08-05','Dostinex',1,'Tumor Diagnosed',4,1);

INSERT INTO medical_records (patient_id,start_date,doctor_id,active) VALUES
(2,'2018-01-08',1,1);

INSERT INTO medical_records (patient_id,start_date,doctor_id,active) VALUES
(4,'2018-08-25',1,1);

INSERT INTO medical_records (patient_id,start_date,doctor_id,active) VALUES
(5,'2018-08-14',1,1);
```

Billing Accounts

```
INSERT INTO
billing_accounts(patient_id,ssn,billing_address,payment_method,card_number,w
ard_charge,test_charge,treatment_charge,fee_charge,total_charge,
start_date,settled,insurance_company_name) VALUES(1,780511208,'2345-Avery
Close,Raleigh','card',
9876511127275454,100,150.25,200.50,100,550.75,'2019-1-1',1,'Livelong');

INSERT INTO
billing_accounts(patient_id,ssn,billing_address,payment_method,card_number,w
ard_charge,test_charge,treatment_charge,fee_charge,total_charge,start_date,s
ettled,insurance_company_name) VALUES(2,76052190,'Gorman
St.,Raleigh','card','3232323211111111',100,250.25,300.50,100,750.75,'2019-2-
2',0,,'Livestrong');

INSERT INTO
billing_accounts(patient_id,ssn,billing_address,payment_method,check_number,
ward_charge,test_charge,treatment_charge,fee_charge,total_charge,start_date,
settled,insurance_company_name) VALUES(3,41062193,'Park Av.,
```

```
Charlotte’,’check’,’1234432189’,125,200.25,250.25,410.25,
985.5,'2018-1-8',0,’KingsInsuranceGroup’);

INSERT INTO
billing_accounts(patient_id,ssn,billing_address,payment_method,check_number,
ward_charge,test_charge,treatment_charge,fee_charge,total_charge,start_date,
end_date,settled,insurance_company_name) VALUES(4,721012190,'Hillsborough
St.,
Raleigh','check','9876512345',100,200,350.50,100.25,750.75,'2017-08-13','201
7-09-13',1,'FidelityInsurance');

INSERT INTO
billing_accounts(patient_id,ssn,billing_address,payment_method,check_number,
ward_charge,test_charge,treatment_charge,fee_charge,total_charge,start_date,
settled,insurance_company_name) VALUES (5,321042191,'Hillsborough St.,
Raleigh','check','1000056789',200,300,350.50,200.50,1050,'2018-08-05',0,'Fid
elityInsurance');
```

Tests

```
INSERT INTO tests(name,price) VALUES(“Echocardiography”, 50);

INSERT INTO tests(name, price) VALUES(“PSA Test”,150);

INSERT INTO tests(name, price) VALUES(“CBC”, 200.50);

INSERT INTO tests(name, price) VALUES(“MRI”, 175.25);
```

Treatments

```
INSERT INTO treatments (name,price) VALUES("Surgery",100.00);

INSERT INTO treatments(name,price) VALUES("Immunotherapy",200.50);

INSERT INTO treatments(name,price) VALUES("Chemotherapy",300.25);

INSERT INTO treatments(name,price) VALUES("Drug Rehabilitation",1000.50);

INSERT INTO treatments(name,price) VALUES("Intravenous therapy",575);
```

Test_for_patients

```
INSERT INTO test_for_patients(id,medical_record_id,result) values (1,2,"PSA
test positive");

INSERT INTO test_for_patients(medical_record_id,test_id,result) values
(1,3,"CBC result positive");

INSERT INTO test_for_patients(medical_record_id,test_id,result) values
(2,3,"CBC result positive");

INSERT INTO test_for_patients(medical_record_id,test_id,result) values
(2,1,"Echocardiography positive");

INSERT INTO test_for_patients(medical_record_id,test_id,result) values
(3,1,"Echocardiography positive");

INSERT INTO test_for_patients(medical_record_id,test_id,result) values
(4,4,"MRI result");

INSERT INTO test_for_patients(medical_record_id,test_id,result) values
(4,1,"Echocardiography positive");
```

Check_in_Info

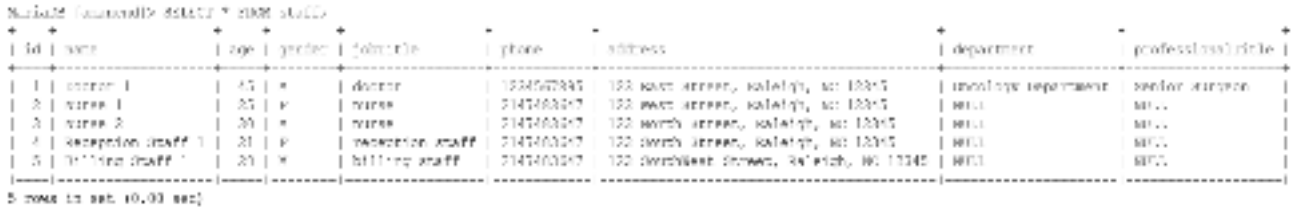
```
INSERT INTO check_in_info(id,patient_id,ward_id,bed_id,start_date,end_date)
values(1,1,1,2,"2003-01-01","2008-09-08");

INSERT INTO check_in_info(id,patient_id,ward_id,bed_id,start_date,end_date)
values(2,1,2,1,"2018-04-06",Null);

INSERT INTO check_in_info(id,patient_id,ward_id,bed_id,start_date,end_date)
values(3,2,3,4,"2018-01-08",Null);

INSERT INTO check_in_info(id,patient_id,ward_id,bed_id,start_date,end_date)
values(4,3,3,1,"2017-08-13","2017-09-13");
```

```
INSERT INTO check_in_info(id,patient_id,ward_id,bed_id,start_date,end_date)
values(5,3,2,2,"2018-08-05",Null);

INSERT INTO check_in_info(id,patient_id,ward_id,bed_id,start_date,end_date)
values(6,4,2,3,"2018-08-25",Null);

INSERT INTO check_in_info(id,patient_id,ward_id,bed_id,start_date,end_date)
values(7,5,3,5,"2018-08-14",Null);
```

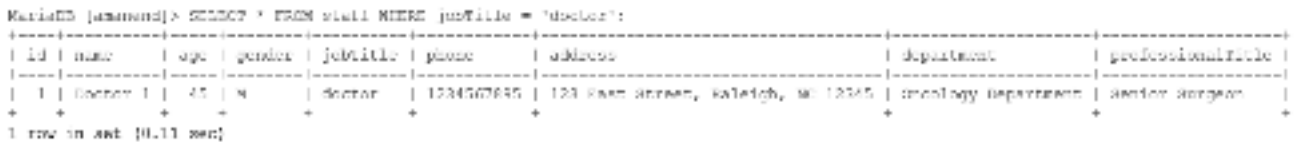
SELECT * FROM Statements

Staff

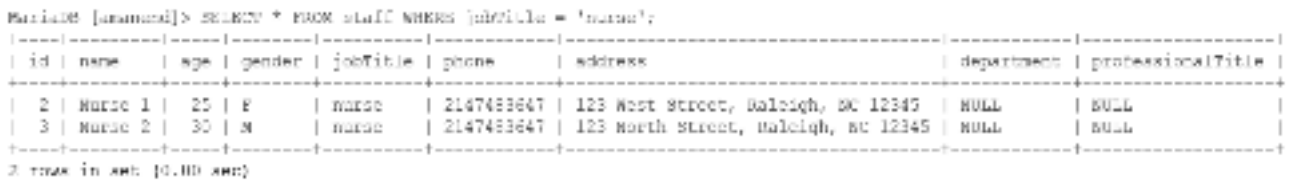
```
SELECT * FROM staff;
```



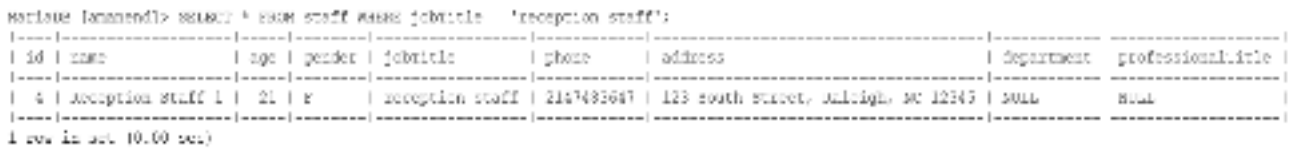
```
SELECT * FROM staff WHERE jobTitle = ‘doctor’;
```



```
SELECT * FROM staff WHERE jobTitle = ‘nurse’;
```



```
SELECT * FROM staff WHERE jobTitle = ‘reception staff’;
```



```
SELECT * FROM staff WHERE jobTitle = ‘billing staff’;
```

```
SELECT * FROM staff WHERE jobTitle = 'admin';
```

Wards

```
SELECT * FROM wards;
```

```
MariaDB [amanend]> select * from wards
    -> ;
+----+----------+------------------+----------+
| id | capacity | charges_per_day | nurse_id |
+----+----------+------------------+----------+
|  1 |        1 |             100 |        2 |
|  2 |        2 |             125 |        2 |
|  3 |        3 |             150 |        3 |
|  4 |        4 |             200 |        3 |
+----+----------+------------------+----------+
4 rows in set (0.00 sec)
```

Patients

```
SELECT * FROM patients;
```

```
MariaDB [amanend]> select * from patients;
+----+----------+--------------+-------------+--------+------------+--------------------------+--------------------------+---------+-----+
| id | ssn      | name         | dateOfBirth | gender | phone      | address                  | status                   | ward_id | age |
+----+----------+--------------+-------------+--------+------------+--------------------------+--------------------------+---------+-----+
|  1 | 78051120 | Jimi Hendrix | 1976-02-17  | M      | 2147483647 | 2345-Avery Close,Raleigh | Processing Treatment Plan |       1 |  43 |
|  2 | 76052190 | Shannon Henry | 1987-02-02 | F      | 2147483647 | Gorman St.,Raleigh       | In Ward                  |       3 |  32 |
|  3 | 41062193 | David Gilmour | 1971-01-03 | M      | 2147483647 | Park Av., Charlotte      | In Ward                  |       2 |  48 |
|  4 | 721012190 | Zoey King    | 1996-01-27  | F      | 2147483647 | Hillsborough St., Raleigh | Completing Treatment    |       2 |  23 |
|  5 | 321042191 | Raj Verma    | 1989-12-22  | M      | 2147483647 | Hillsborough St., Raleigh | Completing Treatment    |       3 |  29 |
+----+----------+--------------+-------------+--------+------------+--------------------------+--------------------------+---------+-----+
5 rows in set (0.01 sec)
```

Beds

```
SELECT * FROM beds;
```

```
MariaDB [amanend]> select * from beds;
+----+----------+-------------+-----------+
| id | ward_id  | patient_id  | reserved  |
+----+----------+-------------+-----------+
|  1 |        1 |           1 |         1 |
|  2 |        2 |           3 |         1 |
|  3 |        2 |           4 |         1 |
|  4 |        3 |           2 |         1 |
|  5 |        3 |           5 |         1 |
|  6 |        3 |        NULL |         0 |
|  7 |        4 |        NULL |         0 |
|  8 |        4 |        NULL |         0 |
|  9 |        4 |        NULL |         0 |
| 10 |        4 |        NULL |         0 |
+----+----------+-------------+-----------+
10 rows in set (0.00 sec)
```

Medical Records

```
SELECT * FROM medical_records;
```

```
MariaDB [amanend]> select * from medical_records;
+----+------------+------------+------------+--------------+-----------+-----------------------------------------------+-----------+--------+
| id | patient_id | start_date | end_date   | prescription | doctor_id | diagnosis                                     | treatment | active |
+----+------------+------------+------------+--------------+-----------+-----------------------------------------------+-----------+--------+
|  1 |          1 | 2003-01-01 | 2008-09-08 | Malarone     |         1 | Positive CBC and Malaria blood test           |         6 |      0 |
|  2 |          3 | 2017-08-13 | 2017-09-13 | Metformin    |         1 | High blood sugar level and low blood cell count |         2 |      0 |
|  3 |          1 | 2018-04-06 | NULL       | Metformin    |         1 | High blood sugar level                        |         2 |      1 |
|  4 |          3 | 2018-08-05 | NULL       | Dostinex     |         1 | Tumor diagnosed                               |         4 |      1 |
|  5 |          2 | 2018-01-08 | NULL       | NULL         |         1 | NULL                                          |      NULL |      1 |
|  6 |          4 | 2018-08-25 | NULL       | NULL         |         1 | NULL                                          |      NULL |      1 |
|  7 |          5 | 2018-08-14 | NULL       | NULL         |         1 | NULL                                          |      NULL |      1 |
+----+------------+------------+------------+--------------+-----------+-----------------------------------------------+-----------+--------+
7 rows in set (0.01 sec)
```

Billing Accounts

```
SELECT * FROM billing_accounts;
```

Tests

```
SELECT * FROM tests;
```

```
MariaDB [amanend]> select * from tests;
+----+------------------+--------+
| id | name             | price  |
+----+------------------+--------+
|  1 | Echocardiography |   50.5 |
|  2 | PSA Test         | 150.25 |
|  3 | CBC              |  200.5 |
|  4 | MRI              | 175.25 |
+----+------------------+--------+
4 rows in set (0.00 sec)
```

Treatments

```
SELECT * FROM treatments;
```

```
MariaDB [amanend]> select * from treatments;
+----+---------------------+--------+
| id | name                | price  |
+----+---------------------+--------+
|  1 | Surgery             |    100 |
|  2 | Immunotherapy       |  200.5 |
|  3 | Chemotherapy        | 300.25 |
|  4 | Drug Rehabilitation | 1000.5 |
|  6 | Intravenous therapy |    575 |
+----+---------------------+--------+
5 rows in set (0.01 sec)
```

Test_for_patients

```
SELECT * FROM test_for_patients;
```

```
MariaDB [amanend]> select * from test_for_patients;
+------------------+---------+--------------------------+
| medical_record_id | test_id | result                   |
+------------------+---------+--------------------------+
|                1 |       2 | PSA test positive        |
|                1 |       3 | CBC result positive      |
|                2 |       1 | Echocardiography positive |
|                2 |       3 | CBC result positive      |
|                3 |       1 | Echocardiography positive |
|                4 |       1 | Echocardiography negative |
|                4 |       4 | MRI result               |
+------------------+---------+--------------------------+
7 rows in set (0.00 sec)
```

Check_in_Info

```
SELECT * FROM check_in_info;
```

```
MariaDB [amanend]> select * from check_in_info;
+----+------------+---------+--------+------------+------------+
| id | patient_id | ward_id | bed_id | start_date | end_date   |
+----+------------+---------+--------+------------+------------+
|  1 |          1 |       1 |      2 | 2003-01-01 | 2008-09-08 |
|  2 |          1 |       2 |      1 | 2018-04-06 | NULL       |
|  3 |          2 |       3 |      4 | 2018-01-08 | NULL       |
|  4 |          3 |       3 |      1 | 2017-08-13 | 2017-09-13 |
|  5 |          3 |       2 |      2 | 2018-08-05 | NULL       |
|  6 |          4 |       2 |      3 | 2018-08-25 | NULL       |
|  7 |          5 |       3 |      5 | 2018-08-14 | NULL       |
+----+------------+---------+--------+------------+------------+
7 rows in set (0.00 sec)
```

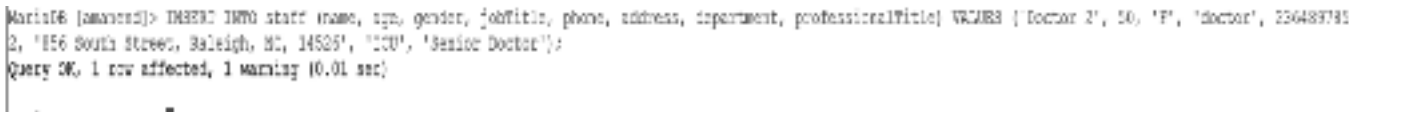
Task & Operations SQL Statements

The input parameters for each method call are user entered values that will fill in the different VALUES for the SQL statements below.

Information processing:

1. **createStaff(name, gender, age, jobTitle, phone, address) (non doctors)**
    a. General Form
        i. INSERT INTO staff (name, age, gender, jobTitle, phone, address) VALUES(name, age, gender, jobTitle, phone, address);
    b. Example
        i. INSERT INTO staff (name, age, gender, jobTitle, phone, address) VALUES(‘Nurse 3, 28, ‘M’, ‘nurse’, 5639871426, ‘234 North Street, Raleigh, NC 12655’);
        ii. Query OK, 1 row affected, 1 warning (0.00 sec)

2. **createDoctor(name, gender, age, jobTitle, phone, address, department, professionalTitle) (doctors)**
    a. General Form
        i. INSERT INTO staff (name, age, gender, jobTitle, phone, address, department, professionalTitle) VALUES (name, age, gender, jobTitle, phone, address, department, professionalTitle);
    b. Example
        i. INSERT INTO staff (name, age, gender, jobTitle, phone, address, department, professionalTitle) VALUES (‘Doctor 2’, 50, ‘F’, ‘doctor’, 2364897852, ‘856 South Street, Raleigh, NC, 14526’, ‘ICU’, ‘Senior Doctor’);
        ii. Query OK, 1 row affected, 1 warning (0.01 sec)



3. **createAdmin(name, gender, age, jobTitle, phone, address, department, professionalTitle) (admins)**
    a. General Form
        i. INSERT INTO staff (name, age, gender, jobTitle, phone, address, department, professionalTitle) VALUES (name, age, gender, jobTitle, phone, address, department, professionalTitle);

b. Example

i. INSERT INTO staff (name, age, gender, jobTitle, phone, address, department, professionalTitle) VALUES ('Admin 2', 28, 'M', 'admin', 1497458120, '704 North Street, Raleigh, NC, 14526', 'Administration', 'Junior Admin');

ii. Query OK, 1 row affected, 1 warning (0.01 sec)

```
MariaDB [amanend]> INSERT INTO staff (name, age, gender, jobTitle, phone, address, department, professionalTitle) VALUES ('Admin 2', 28, 'M', '
admin', 1497458120, '704 North Street, Raleigh, NC, 14526', 'Administration', 'Junior Admin');
Query OK, 1 row affected (0.01 sec)
```

4. **createPatient(name, DOB, gender, age, phone, address, SSN, wardID)**

a. General Form

i. INSERT INTO patients (ssn, name, dateOfBirth, gender, phone, address, status, ward_id, age) VALUES (SSN, name, DOB, gender,phone, address, status,wardID, age);

b. Example

i. INSERT INTO patients (ssn, name, dateOfBirth, gender, phone, address, status, ward_id, age) VALUES (789561258, 'Bob Marley', '1945-02-06', 'M', 1689351456, '267 Gorman Street, Raleigh, NC 27606', 'In Ward', 4, 36);

ii. Query OK, 1 row affected (0.00 sec)

```
MariaDB [amanend]> INSERT INTO patients (ssn, name, dateOfBirth, gender, phone, address, status, ward_id, age) VALUES (789561258, 'Bob Marley', '1945-02-06', 'M',
 1689351456, '267 Gorman Street, Raleigh, NC 27606', 'In Ward', 4, 36);
Query OK, 1 row affected (0.00 sec)
```

5. **createWard(capacity, nurseID, chargesPerDay)**

a. General Form

i. INSERT INTO wards (capacity, charges_per_day, nurse_id) VALUES (capacity, chargesPerDay, nurseID);

b. Example

i. INSERT INTO wards (capacity, charges_per_day, nurse_id) VALUES (1, 95.50, 6);

ii. Query OK, 1 row affected (0.00 sec)

```
MariaDB [amanend]> INSERT INTO wards  (capacity, charges_per_day, nurse_id) VALUES (1, 95.50, 6);
Query OK, 1 row affected (0.00 sec)
```

6. **createBed(wardID)**

a. General Form

i. INSERT INTO beds (ward_id, reserved) VALUES (wardID, 0);

b. Example

i. INSERT INTO beds (ward_id, reserved) VALUES (5, 0);

ii. Query OK, 1 row affected (0.00 sec)

```
MariaDB [amanend]> INSERT INTO beds (ward_id, reserved) VALUES (5, 0);
Query OK, 1 row affected (0.00 sec)
```

7. **updateStaff(name, gender, age, jobTitle, phone, address, staffID) (non doctors)**
    a. General Form
        i. UPDATE staff SET name = name, gender = gender, age = age, jobTitle = jobTitle, phone = phone, address = address WHERE id= staffID;
    b. Example
        i. UPDATE staff SET age = 26, address = ‘1234 West Street, Raleigh, NC 12345’ WHERE id= 2;
        ii. Query OK, 1 row affected (0.01 sec)
        iii. Rows matched: 1 Changed: 1 Warnings: 0

```
MariaDB [amanend]> UPDATE staff SET age = 26, address = '1234 West Street, Raleigh, NC 12345' WHERE id= 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

8. **updateDoctor(name, gender, age, jobTitle, phone, address, professionalTitle, department, staffID) (doctors)**
    a. General Form
        i. UPDATE staff SET name = name, gender = gender, age = age, jobTitle = jobTitle, phone = phone, address = address, professionalTitle = professionalTitle, department = department WHERE id= staffID;
    b. Example
        i. UPDATE staff SET department = ‘Intensive Care Unit’ WHERE id= 7;
        ii. Query OK, 1 row affected (0.00 sec)
        iii. Rows matched:1 Changed: 1 Warnings: 0

```
MariaDB [amanend]> UPDATE staff SET department = 'Intensive Care Unit' WHERE id= 7;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

9. **updatePatient(name, DOB, gender, age, phone, address, SSN, status, patientID, wardID)**
    a. General Form
        i. UPDATE patients SET name = name, dateOfBirth = DOB, gender = gender, age = age, phone = phone, address = address, ssn = SSN status= status ward_id = wardID WHERE id= patientID;
    b. Example

i. UPDATE patients SET age = ‘Completing Treatment’ WHERE id= 7;
ii. Query OK, 1 row affected (0.00 sec)
iii. Rows matched: 1 Changed: 1 Warnings: 0

```
MariaDB [amanend]> UPDATE patients SET status  = 'Completing Treatment' WHERE id= 7;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

10. updateWard(capacity, nurseID, wardID, chargesPerDay)

a. General Form
   i. UPDATE wards SET capacity= capacity, nuse_id= nurseID, charges_per_day= chargesPerDay WHERE id= wardID;

b. Example
   i. UPDATE wards SET charges_per_day = 105.5 WHERE id = 5;
   ii. Query OK, 1 row affected (0.00 sec)
   iii. Rows matched: 1 Changed: 1 Warnings: 0

```
MariaDB [amanend]> UPDATE wards SET charges_per_day = 105.5 WHERE id = 5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

11. updateBed(bedID, wardID, patientID, reserveBed)

a. General Form
   i. UPDATE beds SET ward_id = wardID, patient_id= patientID, reserveBed = reserveBed WHERE id= bedID;

b. Example
   i. UPDATE beds SET patient_id = 7 where id = 7;
   ii. Query OK, 1 row affected (0.01 sec)
   iii. Rows matched:1 Changed: 1 Warnings: 0

```
MariaDB [amanend]> UPDATE  beds SET patient_id = 7 where id = 7;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

12. deleteStaff(staffID)

a. General Form
   i. DELETE staff WHERE id=staffID;

b. Example
   i. DELETE staff WHERE id = 7;
   ii. Query OK, 1 row affected (0.00 sec)

```
MariaDB [amanend]> DELETE FROM staff WHERE id = 7;
Query OK, 1 row affected (0.00 sec)
```

13. deletePatient(patientID)

a. General Form
   i. DELETE FROM patients WHERE id=patientID;
   ii. UPDATE beds SET reserved = 0 WHERE patient_id = patientID;

b. Example
   i. DELETE FROM patients WHERE id = 7;
   ii. Query OK, 1 row affected (0.01 sec)

```
MariaDB [amanend]> DELETE from patients where id = 7;
Query OK, 1 row affected (0.01 sec)
```

14. deleteWard(wardID)

a. General Form
   i. DELETE FROM wards WHERE id=wardID;

b. Example
   i. DELETE FROM wards WHERE id = 6;

```
MariaDB [amanend]> DELETE FROM wards WHERE id = 6;
Query OK, 1 row affected (0.00 sec)
```

15. deleteBed(bedID)

a. General Form
   i. DELETE FROM beds WHERE id = bedID;

b. Example
   i. DELETE FROM beds WHERE id = bedID;

```
MariaDB [amanend]> DELETE FROM beds WHERE id = 13;
Query OK, 1 row affected (0.00 sec)
```

16. reserveBed(bedID, wardID)

a. General Form
   i. UPDATE beds SET ward_id = wardID, reserved = 1 WHERE id = bedID;

b. Example
   i. UPDATE beds SET ward_id = wardID, reserved = 1 WHERE id = 14;

```
MariaDB [amanend]> UPDATE beds SET ward_id = 5, reserved = 1 WHERE id = 14;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

17. assignBed(bedID, patientID)

a. General Form

i. UPDATE beds SET patient_id = patientID, reserved = 1 WHERE id = bedID;

ii. UPDATE patients inner join beds on patients.id=beds.patient_id set patients.ward_id =beds.ward_id where beds.id=bedID;

b. Example

i. UPDATE beds SET patient_id = 8, reserved = 1 WHERE id = 14;

ii. UPDATE patients inner join beds on patients.id=beds.patient_id set patients.ward_id =beds.ward_id where beds.id=14;

```
MariaDB [amanend]> UPDATE beds SET patient_id = 8, reserved = 1 WHERE id = 14;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [amanend]> UPDATE patients inner join beds on patients.id=beds.patient_id set patients.ward_id =beds.ward_id where beds.id=14;
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

18. releaseBed(bedID, patientID)

a. General Form

i. UPDATE beds SET patient_id = NULL, reserved = 0 WHERE id = bedID;

ii. UPDATE patients SET ward_id = NULL WHERE id = patientID;

b. Example

i. UPDATE beds SET patient_id = NULL, reserved = 0 WHERE id = 14;

ii. UPDATE patients set ward_id = NULL WHERE id = 8;

```
MariaDB [amanend]> UPDATE beds SET patient_id = NULL, reserved = 0 WHERE id = 14;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [amanend]> UPDATE patients set ward_id = NULL WHERE id = 8;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

19. checkWard(wardID, numberOfBeds)

a. General Form

i. SELECT * FROM wards WHERE id = wardID AND capacity = Capacity;
ii. Or
iii. SELECT * FROM wards WHERE capacity = numberOfBeds;

b. Example
i. SELECT * FROM wards where id=1 and capacity=1;

```
MariaDB [amanend]> select * from wards where id=1 and capacity=1;
+----+----------+-----------------+----------+
| id | capacity | charges_per_day | nurse_id |
+----+----------+-----------------+----------+
|  1 |        1 |             100 |        2 |
+----+----------+-----------------+----------+
1 row in set (0.00 sec)
```

20. createTest(name, price)

a. General Form
i. INSERT INTO tests (name, price) VALUES (name, price);

b. Example:
i. INSERT INTO tests (name,price) VALUES('endoscopy',100);

```
MariaDB [amanend]> insert into tests (name,price) values('endoscopy',100);
Query OK, 1 row affected (0.01 sec)
```

21. createTreatment(name, price)

a. General Form
i. INSERT INTO treatments (name, price) VALUES(name, prce);

b. Example
i. INSERT INTO treatments s(name,price) values('biopsy',300);

```
MariaDB [amanend]> insert into treatments(name,price) values('biopsy',300);
Query OK, 1 row affected (0.00 sec)
```

22. updateTest(testID, name, price)

a. General Form
i. UPDATE test SET name = name, price = price WHERE id= testID;

b. Example
i. UPDATE tests SET name = ‘Colonoscopy’, price = 200 where id =7;

```
MariaDB [amanend]> update tests set name='Colonoscopy', price=200 where id=7;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [amanend]> select * from tests;
+----+------------------+--------+
| id | name             | price  |
+----+------------------+--------+
|  1 | Echocardiography |   50.5 |
|  2 | PSA Test         | 150.25 |
|  3 | CBC              |  200.5 |
|  4 | MRI              | 175.25 |
|  7 | Colonoscopy      |    200 |
+----+------------------+--------+
```

23. updateTreatment(treatmentID, name, price)

a. General Form

i. UPDATE treatments SET name = name, price = price WHERE id= treatmentID;

b. Example:

i. UPDATE treatments SET name='Intravenous therapy', price=575 where id=6;

```
MariaDB [amanend]> update treatments set name='Intravenous therapy', price=575 where id=6;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [amanend]> select * from treatments;
+----+---------------------+--------+
| id | name                | price  |
+----+---------------------+--------+
|  1 | Surgery             |    100 |
|  2 | Immunotherapy       |  200.5 |
|  3 | Chemotherapy        |  300.5 |
|  4 | Drug rehabilitation | 1000.5 |
|  6 | Intravenous therapy |    575 |
|  8 | Cataract surgery    |    100 |
+----+---------------------+--------+
```

24. deleteTest(testID)

a. General Form

i. DELETE FROM tests WHERE id = testID;

b. Example

i. DELETE FROM tests WHERE id=6;

ii. Query OK, 1 row affected (0.00 sec)

```
MariaDB [amanend]> delete from tests where id=6;
Query OK, 1 row affected (0.00 sec)
```

25. deleteTreatment(treatmentID)

a. General Form
    i. DELETE FROM treatments WHERE id = treatmentID
b. Example
    i. DELETE FROM treatments WHERE id=7;
    ii. Query OK, 1 row affected (0.00 sec)

```
MariaDB [amanend]> delete from treatments where id=7;
Query OK, 1 row affected (0.00 sec)
```

Maintaining Medical Records:

1. **emergencyCheckIn(patientID, startDate)**
    a. General Form
        INSERT INTO check_in_info(patient_id,start_date) VALUES (patientID, startDate);
    b. Example
        i. INSERT INTO check_in_info(patient_id,start_date) VALUES (6, '2018-09-08');

```
MariaDB [amanend]> insert into check_in_info(patient_id,start_date) values(6,'2018-09-08');
Query OK, 1 row affected (0.00 sec)
```

2. **normalCheckIn(patientID, startDate, wardID, bedID)**
    a. General Form
        i. INSERT INTO check_in_info (patient_id,start_date,ward_id,bed_id) VALUES(patientID, startDate, wardID, bedID);
    b. Example
        i. INSERT INTO check_in_info (patient_id,start_date,ward_id,bed_id) VALUES(7, '2017-02-09', 2, 2);

```
MariaDB [amanend]> insert into check_in_info (patient_id,start_date,ward_id,bed_id) values (7,'2017-02-09',2,2);
Query OK, 1 row affected (0.00 sec)
```

3. **createMedicalRecord(patientID, startDate)**
    a. General Form
        i. INSERT INTO medical_records (patient_id, start_date) values (patientID,startDate);
    b. Example

i. INSERT INTO medical_records (patient_id, start_date) values (4,’2018-06-06’);

```
MariaDB [amanend]> insert into medical_records(patient_id,start_date) values(4,'2018-06-06');
Query OK, 1 row affected, 1 warning (0.00 sec)
```

4. **updateCheckIn(checkInID, patientID, startDate, endDate, wardID, bedID)**
   a. General Form
      i. UPDATE check_in_info SET end_date= endDate, ward_id= wardID, bed_id= bedID WHERE patient_id= patientID and start_date=startDate;
      ii. Or
      iii. UPDATE check_in_info SET patient_id = patientID, start_date = startDate, end_date= endDate, ward_id= wardID, bed_id= bedID WHERE id = checkInID;
   b. Example
      i. UPDATE check_in_info SET end_date= ‘2017-03-10’, ward_id= 3, bed_id= 3 WHERE patient_id= 7 and start_date=’2017-02-09’;

```
MariaDB [amanend]> update check_in_info set end_date='2017-03-10', ward_id=3,bed_id=3 where patient_id=7 and start_date='2017-02-09';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

5. **updateMedicalRecord(medicalRecordID, patientID, startDate, endDate, doctorID, prescription, diagnosis, treatment, active)**
   a. General Form
      i. UPDATE medical_records SET patient_id = patientID, start_date = startDate, end_date = endDate, doctor_id = doctorID, prescription = prescription, diagnosis = diagnosis, treatment = treatment, active = active WHERE id = medicalRecordID;
      ii. Or
      iii. UPDATE check_in_info SET end_date = endDate, doctor_id = doctorID, prescription = prescription, diagnosis = diagnosis, treatment = treatment, active = active WHERE patient_id = patientID AND start_date = startDate;
   b. Example
      i. UPDATE medical_records SET patient_id = 7, start_date = '2018-08-11', end_date = '2018-09-12', doctor_id = 1, prescription = 'Dostinex', diagnosis = 'Tumor Diagnosed', treatment = 6, active = 0 WHERE id = 2;

```
MariaDB [amanend]> UPDATE medical_records SET patient_id = 7, start_date = '2018-08-11', end_date = '2018-09-12', doctor_id = 1, prescription = 'Dostinex', diagnosis = 'Tumor Diagnosed
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

6. **insertTestForPatient(medicalRecordID, patientID, result)**
    a. General Form
        i. `INSERT INTO test_for_patient(medical_record_id, test_id, result) VALUES (medicalRecordID, patientID, result)`
    b. Example
        i. INSERT INTO test_for_patients(medical_record_id, test_id, result) VALUES (5, 5, 'MRI Result');

```
MariaDB [amanend]> INSERT INTO test_for_patients(medical_record_id, test_id, result) VALUES (5, 5, 'MRI Result');
Query OK, 1 row affected (0.00 sec)
```

7. **updateTestForPatient(medicalRecordID, patientID, result)**
    a. General Form
        i. `UPDATE test_for_patients SET result = result WHERE test_id= test_ID AND medical_record_id = medicalRecordID;`
    b. Example
        i. `update test_for_patients set result='PSA test negative' wh ere test_id=2 and medical_record_id=1;`

```
MariaDB [amanend]> update test_for_patients set result='PSA test negative' where test_id=2 and medical_record_id=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Maintaining Billing Accounts:

1. **createBillingAccount(SSN, billAddr, payment type, credit card number, check number, test charge, ward charge, treatment charge, fee charge, total charge, patientID, startDate, endDate, insurance company, settled)**
    a. General Form
        i. `INSERT INTO billing_accounts(patient_id, ssn, billing_address, payment_method, card_number, check_number, ward_charge, test_charge, treatment_charge, fee_charge, total_charge, start_date, end_date, settled, insurance_company, name) VALUES(patientID, SSN, billAddr, payment type, credit card number, check number, ward charge, test charge, treatment charge, fee charge, total charge, startDate, endDate, settled, insurance company);`
    b. Example

i. INSERT INTO billing_accounts(patient_id,start_date,settled) values(7,'2019-01-09',0);

```
MariaDB [amanend]> insert into billing_accounts(patient_id,start_date,settled) values(7,'2019-01-09',0);
Query OK, 1 row affected (0.01 sec)
```

ii.

2. **updateBillingAccount(billingAccountID, SSN, billAddr, payment type, credit card number, check number, test charge, ward charge, treatment charge, fee charge, total charge, patientID, startDate, endDate, settled, insurance company)**
   a. General Form
      i. update billing_accounts SET patient_id = patientID, ssn = SSN, billing_address = billAddr, payment_type = payment type, card_number = credit card number, check_number = check number, test_charge = test charge, ward_charge = ward charge, treatment_charge = treatment charge, fee_charge = fee charge, total_charge = total charge, start_date = startDate, end_date = endDate, settled = settled, insurance_company_name = insurance company WHERE id = billingAccountID;
   b. Example
      i. UPDATE billing_accounts SET ward_charge=500, total_charge=950.75 WHERE id=1;

```
MariaDB [amanend]> update billing_accounts set ward_charge=500, total_charge=950.75 where id=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

      ii.

3. **checkHospitalSpace()**
   a. General Form
      i. SELECT * FROM beds where reserved=0;
   b. Example:
      i. SELECT * FROM beds where reserved=0;

```
MariaDB [amanend]> select * from beds where reserved=0;
+----+---------+------------+----------+
| id | ward_id | patient_id | reserved |
+----+---------+------------+----------+
|  6 |       3 |       NULL |        0 |
|  8 |       4 |       NULL |        0 |
|  9 |       4 |       NULL |        0 |
| 10 |       4 |       NULL |        0 |
| 11 |       5 |       NULL |        0 |
| 12 |       4 |       NULL |        0 |
+----+---------+------------+----------+
6 rows in set (0.01 sec)
```

      ii.

4. **releasePatient(patientID, startDate, endDate)**
    a. General Form
        i. Makes the patient settled so they are ready to leave hospital
            1. UPDATE billing_accounts SET settled = 1 WHERE patient_id = patientID AND start_date = startDate AND end_date = endDate;
    b. Example
        i. UPDATE billing_accounts SET settled = 1 WHERE patient_id = 1 AND start_date = '2003-01-01' AND end_date = '2008-09-08';

```
MariaDB [amanend]> UPDATE billing_accounts SET settled = 1 WHERE patient_id = 1 AND start_date = '2003-01-01' AND end_date = '2008-09-08';
Query OK, 0 rows affected (0.01 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```

Reports:

1. **generateMedicalReport(month,year, patientID)**
    a. General Form
        i. SELECT* FROM medical_records WHERE start_date <= 'year-month-01' and (end_date IS NULL or end_date >= 'year-month-01') AND patient_id=patientID
    b. Example
        i. SELECT* FROM medical_records WHERE start_date<= '2019-01-01' and (end_date IS NULL or end_date >= '2019-01-01') AND patient_id=3;

```
MariaDB [amanend]> SELECT* FROM medical_records WHERE start_date<= '2019-01-01' and (end_date IS NULL or end_date >= '2019-01-01') AND patient_id=3;
+----+------------+------------+----------+--------------+-----------+-----------------+-----------+--------+
| id | patient_id | start_date | end_date | prescription | doctor_id | diagnosis       | treatment | active |
+----+------------+------------+----------+--------------+-----------+-----------------+-----------+--------+
|  4 |          3 | 2018-08-05 | NULL     | Dostinex     |         1 | Tumor diagnosed |         4 |      1 |
+----+------------+------------+----------+--------------+-----------+-----------------+-----------+--------+
1 row in set (0.00 sec)
```

2. **generatePatientsPerMonth (month,year)**
    a. General Form
        i. SELECT COUNT(*) FROM medical_records WHERE MONTH(start_date) = month AND YEAR(start_date) = year;
    b. Example
        i. SELECT COUNT(*) FROM medical_records WHERE MONTH(start_date) = 8 AND YEAR(start_date) = 2018;

```
MariaDB [amanend]> select count(*) from medical_records where MONTH(start_date)=8 and YEAR(start_date)=2018;
+----------+
| count(*) |
+----------+
|        1 |
+----------+
1 row in set (0.00 sec)
```

3. **generatePatientsPerMonth ()**
    a. General Form
        i. SELECT COUNT(*),month(start_date),year(start_date) FROM medical_records GROUP BY month(start_date),year(start_date) ORDER BY start_date;
    b. Example
        i. select count(*),month(start_date),year(start_date) from medical_records group by month(start_date),year(start_date) order by start_date;

```
MariaDB [amanend]> select count(*),month(start_date),year(start_date) from medical_records group by month(start_date),year(start_date) order by start_date;
+----------+-------------------+------------------+
| count(*) | month(start_date) | year(start_date) |
+----------+-------------------+------------------+
|        1 |                 1 |             2003 |
|        1 |                 8 |             2017 |
|        1 |                 4 |             2018 |
|        1 |                 8 |             2018 |
+----------+-------------------+------------------+
4 rows in set (0.00 sec)
```

4. **generatePatientsUnderDoctor (staffID)**
    a. General Form
        i. SELECT p.id,ssn,name,dateOfBirth,gender,phone,address,status,ward_id,start_date,prescription,diagnosis,treatment,active FROM patients p INNER JOIN medical_records mr on p.id=mr.patient_id WHERE mr.doctor_id=staffID AND mr.active=1;
    b. Example
        i. SELECT p.id,ssn,name,dateOfBirth,gender,phone,address,status,ward_id,start_date,prescription,diagnosis,treatment,active FROM patients p INNER JOIN medical_records mr on p.id=mr.patient_id WHERE mr.doctor_id=1 AND mr.active=1;

```
MariaDB [amanend]> select p.id,ssn,name,dateOfBirth,gender,phone,address,status,ward_id,start_date,prescription,diagnosis,treatment,active from patients p  inner join medic al_records mr on p.id=mr.patient_id
where mr.doctor_id=1 and mr.active=1;
+----+-----------+--------------+-------------+--------+-----------+-----------------------+---------------------------+---------+------------+--------------+-----------------------+-----------+--------+
| id | ssn       | name         | dateOfBirth | gender | phone     | address               | status                    | ward_id | start_date | prescription | diagnosis             | treatment | active |
+----+-----------+--------------+-------------+--------+-----------+-----------------------+---------------------------+---------+------------+--------------+-----------------------+-----------+--------+
|  1 | 78051120  | Jimi Hendrix | 1976-02-17  | M      | 2147483647 | 2345-Avery Close,Raleigh | Processing Treatment Plan |       1 | 2018-04-06 | Metformin    | High blood sugar level |         2 |      1 |
|  3 | 41062193  | David Gilmour | 1971-01-03  | M      | 2147483647 | Park Av., Charlotte   | In Ward                   |       2 | 2018-08-05 | Dostinex     | Tumor diagnosed       |         4 |      1 |
+----+-----------+--------------+-------------+--------+-----------+-----------------------+---------------------------+---------+------------+--------------+-----------------------+-----------+--------+
2 rows in set (0.00 sec)
```

5. **generateHospitalStaff (jobTitle)**
    a. General Form
        i. `SELECT * FROM staff WHERE jobTitle=jobTitle;`
    b. Example
        i. `SELECT * FROM staff WHERE jobTitle='nurse';`

```
MariaDB [amanend]> select * from staff where jobTitle='nurse';
+----+---------+-----+--------+----------+------------+-----------------------------------+------------+--------------------+
| id | name    | age | gender | jobTitle | phone      | address                           | department | professionalTitle  |
+----+---------+-----+--------+----------+------------+-----------------------------------+------------+--------------------+
|  2 | Nurse 1 |  26 | F      | nurse    | 2147483647 | 1234 West Street, Raleigh, NC 12345 | NULL       | NULL               |
|  3 | Nurse 2 |  30 | M      | nurse    | 2147483647 | 123 North Street, Raleigh, NC 12345 | NULL       | NULL               |
|  6 | Nurse 3 |  28 | M      | nurse    | 2147483647 | 234 North Street, Raleigh, NC 12655 | NULL       | NULL               |
+----+---------+-----+--------+----------+------------+-----------------------------------+------------+--------------------+
3 rows in set (0.01 sec)
```

6. **generateCurrentUsageWards(wardID)**
    a. General Form
        i. `SELECT ward_id,count(*) as total_beds,((select count(*) FROM beds WHERE ward_id=wardID AND patient_id IS NOT NUll)/count(*))*100 as 'usage %' from beds WHERE ward_id=wardID;`
    b. Example
        i. `SELECT ward_id,count(*) as total_beds,((select count(*) FRO beds WHERE ward_id=3 AND patient_id IS NOT NUll)/count(*))*100 as 'usage %' from beds WHERE ward_id=3;`

```
MariaDB [amanend]> select ward_id,count(*) as total_beds,((select count(*) from beds where ward_id=3 and
patient_id IS NOT NUll)/count(*))*100 as 'usage %' from beds where ward_id=3;
+---------+------------+---------+
| ward_id | total_beds | usage % |
+---------+------------+---------+
|       3 |          3 | 66.6667 |
+---------+------------+---------+
1 row in set (0.00 sec)
```

7. **generateCurrentUsageBeds()**
    a. General Form

i. Select * from beds;

b. Example

i. Select * from beds;

```
MariaDB [amanend]> select * from beds;
+----+---------+------------+----------+
| id | ward_id | patient_id | reserved |
+----+---------+------------+----------+
|  1 |       1 |          1 |        1 |
|  2 |       2 |          3 |        1 |
|  3 |       2 |          4 |        1 |
|  4 |       3 |          2 |        1 |
|  5 |       3 |          5 |        1 |
|  6 |       3 |       NULL |        0 |
|  8 |       4 |       NULL |        0 |
|  9 |       4 |       NULL |        0 |
| 10 |       4 |       NULL |        0 |
| 11 |       5 |       NULL |        0 |
| 12 |       4 |       NULL |        0 |
+----+---------+------------+----------+
11 rows in set (0.01 sec)
```

8. **generateAllTestForPatVisit(patientID, startDate)**

a. General Form

i. select t.test_id,t.result,a.id from test_for_patients t inner join medical_records a on t.medical_record_id=a.id where a.patient_id=patientID and a.start_date = startDate;

b. Example

i. select t.test_id,t.result,a.id from test_for_patients t inner join medical_records a on t.medical_record_id=a.id where a.patient_id=1 and a.start_date = '2003-01-01';

```
MariaDB [amanend]> select t.test_id,t.result,a.id from test_for_patients t inner join medical_records a
on t.medical_record_id=a.id where a.patient_id=1 and a.start_date =
'2003-01-01';
+---------+--------------------+----+
| test_id | result             | id |
+---------+--------------------+----+
|       2 | PSA test negative  |  1 |
|       3 | CBC result positive |  1 |
+---------+--------------------+----+
2 rows in set (0.01 sec)
```

EXPLAIN Directive

1. SELECT * FROM staff WHERE jobTitle= ‘doctor’;

a. Select all staff who are doctors;

b. Query output

c. EXPLAIN SELECT * FROM staff WHERE jobTitle= ‘doctor’;

i. Explain query output

```
MariaDB [amanend]> explain select * from staff where jobTitle = 'doctor';
+------+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id   | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+------+-------------+-------+------+---------------+------+---------+------+------+-------------+
|    1 | SIMPLE      | staff | ALL  | NULL          | NULL | NULL    | NULL |    6 | Using where |
+------+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 row in set (0.00 sec)
```

d. CREATE INDEX jobTitle_index on staff(jobTitle);

```
MariaDB [amanend]> CREATE INDEX jobTitle_index on staff(jobTitle);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

e. EXPLAIN SELECT * FROM staff WHERE jobTitle = ‘doctor’;

2. SELECT * FROM wards WHERE capacity = 4;

a. Select all wards that have 4 beds in them

b. Query result

```
MariaDB [amanend]> select * from wards where capacity = 4;
+----+----------+-----------------+----------+
| id | capacity | charges_per_day | nurse_id |
+----+----------+-----------------+----------+
|  4 |        4 |             200 |        3 |
+----+----------+-----------------+----------+
1 row in set (0.00 sec)
```

c. EXPLAIN SELECT * FROM wards WHERE capacity = 4;

```
MariaDB [amanend]> explain select * from wards where capacity = 4;
+------+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id   | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+------+-------------+-------+------+---------------+------+---------+------+------+-------------+
|    1 | SIMPLE      | wards | ALL  | NULL          | NULL | NULL    | NULL |    5 | Using where |
+------+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 row in set (0.00 sec)
```

d. CREATE INDEX capacity_index on wards(capacity)

```
MariaDB [amanend]> create index capacity_index on wards(capacity);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

e. EXPLAIN SELECT * FROM wards WHERE capacity = 4;

```
MariaDB [amanend]> explain select * from wards where capacity = 4;
+------+-------------+-------+------+----------------+----------------+---------+-------+------+-------+
| id   | select_type | table | type | possible_keys  | key            | key_len | ref   | rows | Extra |
+------+-------------+-------+------+----------------+----------------+---------+-------+------+-------+
|    1 | SIMPLE      | wards | ref  | capacity_index | capacity_index | 4       | const |    1 |       |
+------+-------------+-------+------+----------------+----------------+---------+-------+------+-------+
1 row in set (0.00 sec)
```

Query Correctness Proof

Query 1:

1. SELECT p.id,ssn,name,dateOfBirth,gender,phone,address,status,ward_id,start_date,prescription,diagnosis,treatment,active FROM patients p inner join medical_records mr on p.id=mr.patient_id WHERE mr.doctor_id=1 and mr.active=1;

2. Relational Algebra:

$$\Pi_{\text{patients.id,ssn,name,dateOfBirth,gender,phone,address,status,ward\_id,start\_date,prescription,diagnosis,treatment,active}}(\text{patients} \bowtie_{\text{patients.id = medical\_records,patient\_id}} (\sigma_{\text{doctor\_id=1 and active=1}} \text{medical\_records}))$$

3. Specification:
   The query returns all the patients under a given doctor who are currently in the hospital.

4. Proof of correctness:
   Suppose m is one tuple in the medical records relation such that it has doctor id as ‘d’ and is currently in the hospital given by active=1. The patient_id in tuple m is same as the id in tuple p of the patients table. Each combination of (p,m) gives all the information of patients under doctor with id ‘d’ and who are active(currently present in the hospital). The above query returns patients id,ssn,name,date of birth,gender,phone,address,status,ward_id,start_date,prescription,diagnosis,treatment,active for all the patients under a given doctor.

Query 2:

1. select t.test_id,t.result from test_for_patients t inner join (select id from medical_records where patient_id=1 and start_date='2003-01-01')as a on t.medical_record_id=a.id;

2. Relational Algebra:
   $\Pi_{t.test\_id,t.result}$ (ϱ_t (test_for_patients)) $\bowtie_{t.medical\_record\_id=a.id}$ $\rho_a(\Pi_{id}$ ($\sigma_{patient\_id=1 \text{ and } start\_date='2003-01-01'}$ (medical_records))

3. Specification:
   The query returns all the tests taken by a patient for each visit in the hospital.

4. Proof of correctness:
   Tuple p in the subquery corresponds to a tuple with unique medical_record_id from the medical_records relation where patient_id and start date are equal to given as input to the query.Tuple q,r in test_for_patients relation has the medical_record_id given by the subquery. The join of id from subquery with test_for_patients relation gives us the tuple q and r. Tuples q and r have all the information about the tests and the results that a patient takes.