

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

Department of Information Technology

HEALTHCARE DATABASE MANAGEMENT SYSTEM

SUBMITTED BY:

JAHANVI CHAUDHARY (191CH063)

RASHMI VASANTH (191CH040)

SARODE UMASRI BAI (191CH044)

Contents

1.	Problem Description	03
2.	Actors	04
3.	Sample queries	05
4.	Entities that are maintained in database system	06
5.	Conceptual Model (EER)	06
6.	Global Conceptual Schema	08
7.	ER diagram	09

1. Problem description:

The project Healthcare Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

It is accessible either by a nurse or a doctor. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast. Healthcare Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Healthcare Management System is designed for multi speciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Healthcare Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Healthcare Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Healthcare Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

In our healthcare management system, A patient is identified by his Id, age, name, phone number, blood group, address, status of his condition, date of admission and date of discharge. A patient will undergo one test in lab and one test

can be taken by many patients. It is not necessary that all patients undergo a test and that all tests be conducted. A lab entity is identified by its Id, test name, lab department and test report. A care taker who takes care of the patient is identified by his blood group, phone number and name. If there are no patients then there won't be any care taker/relative. A patient will have only one care taker and a care taker will take care of only one patient. A patient makes one appointment and a particular appointment is made by one patient. An appointment is identified by the date, day and time and appointment number. It is not necessary that all patients make an appointment. A doctor treats many patients and a patient is treated by only one doctor but every patient should be treated by a doctor. A doctor is identified by his Id, name, age, salary, department and his schedule that is when he is available. A nurse can assist many doctors and a doctor can have many nurses who help him. Every nurse should help a doctor and every doctor should have a nurse to help him. A nurse is identified by her Id, age, name, salary, department and schedule. A hospital contains many patients and a patient should be in only one hospital. A hospital is identified by room number, number of births and deaths in hospital and branch of hospital. A pharmacy is there in only one hospital and a hospital has only one pharmacy. A pharmacy is identified by medicine Id, name, price and quantity of medicine. A patient pays a bill which has bill number and total cost.

2. ACTORS:

The people with interact with database are: DOCTORS AND NURSES.

3. SOME SAMPLE QUERIES:

❖ **A nurse can:**

- Manages patient.
- Assists doctors.
- Allots rooms, tests for patients.
- Provide medication according to patient prescription
- Attends appointments with doctor.
- Keep record of patient health, baby born and death of patient.
- Manages the pharmacy.
- Manages the bills of patient.
- Manage own profile.

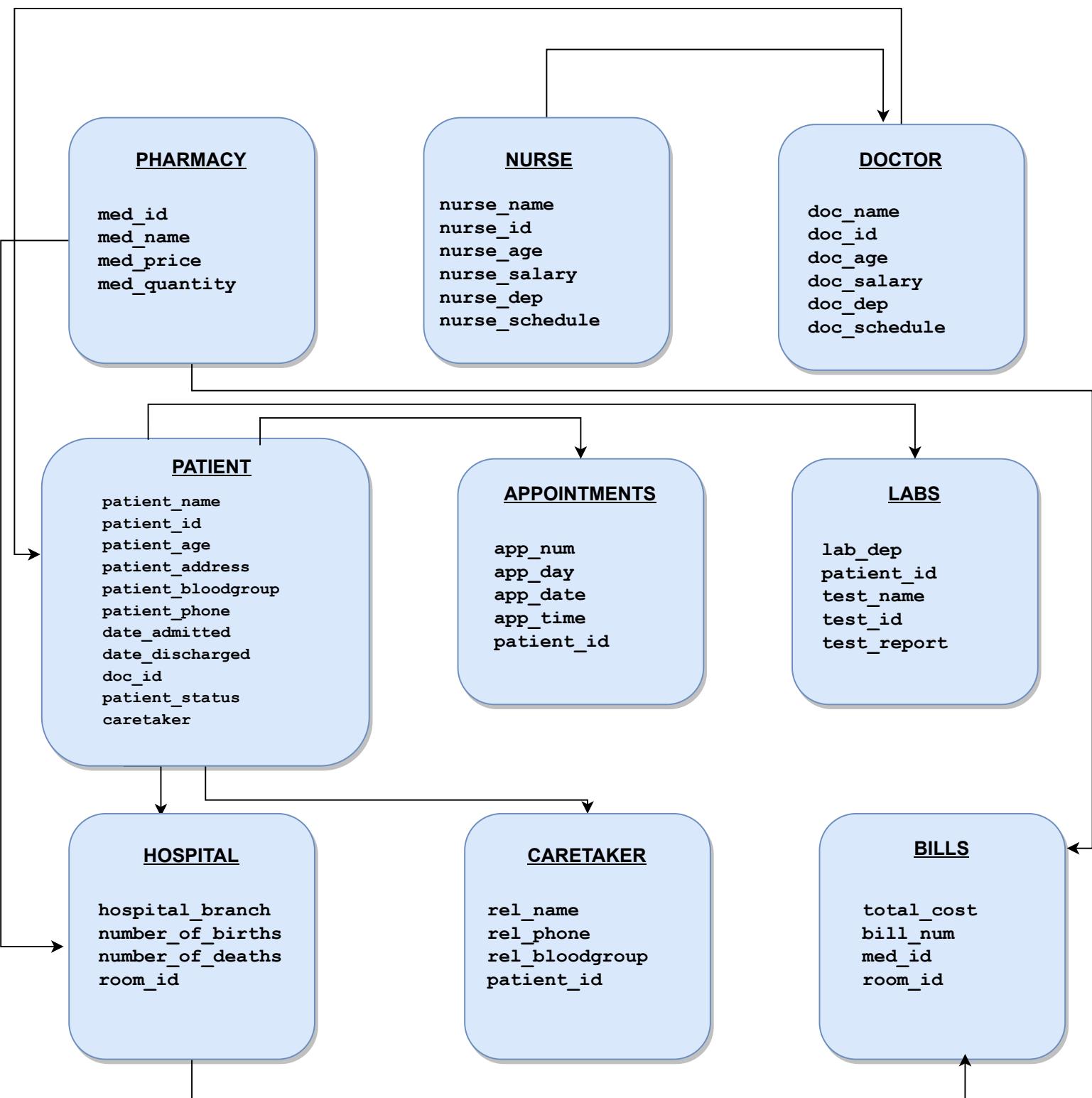
❖ **A doctor can:**

- Manage patient.
- account opening and updating.
- Create, manage appointment with patient.
- Create prescription for patient.
- Provide medication for patients.
- Issue for operation of patients and creates operation report.
- Manages the pharmacy.
- Manage own profile.

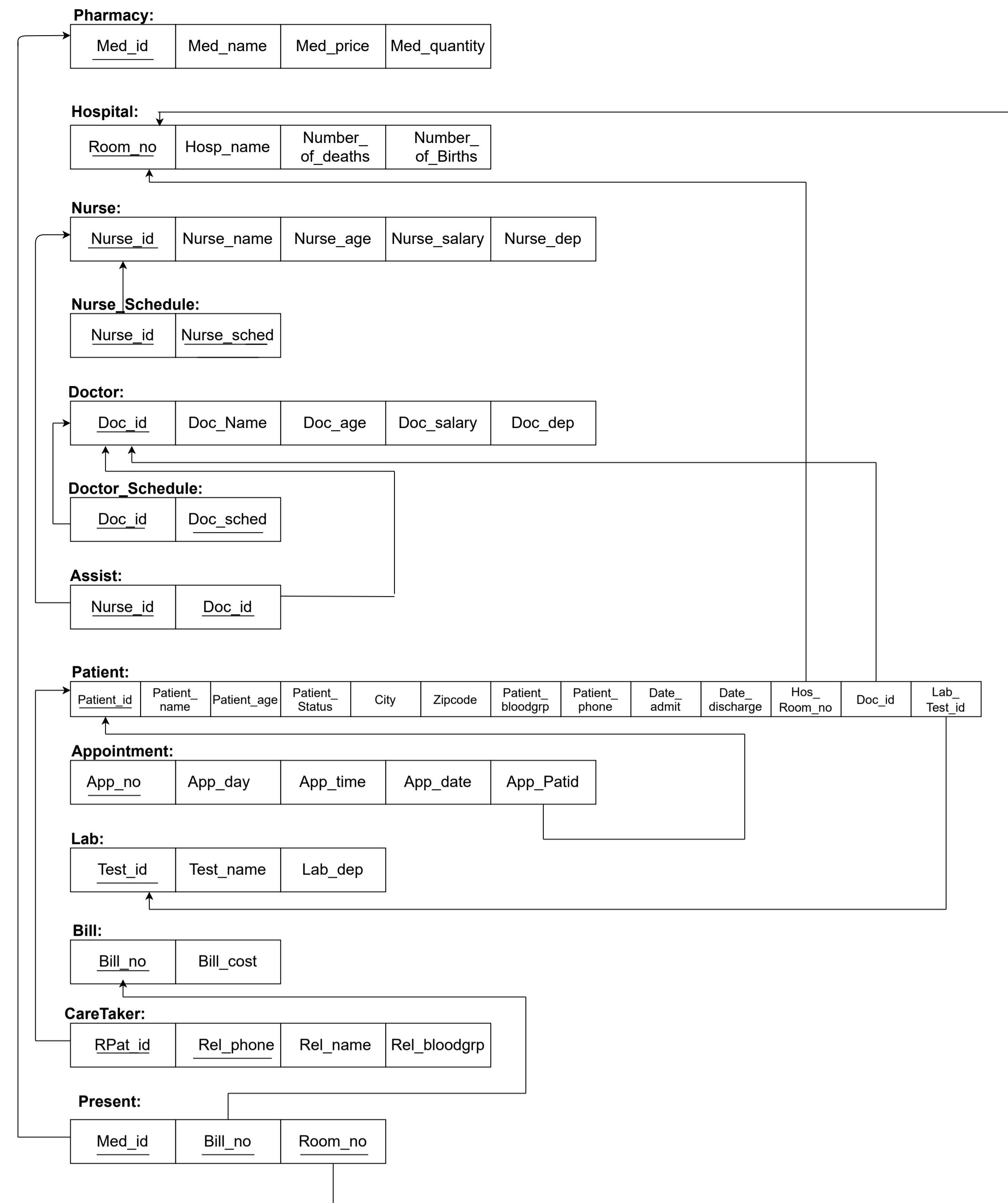
4. Entities that are maintained in database system:

- ④ Nurse
- ④ Appointments
- ④ Doctor
- ④ Labs
- ④ Patients
- ④ Caretaker
- ④ Bills
- ④ Hospital
- ④ Pharmacy

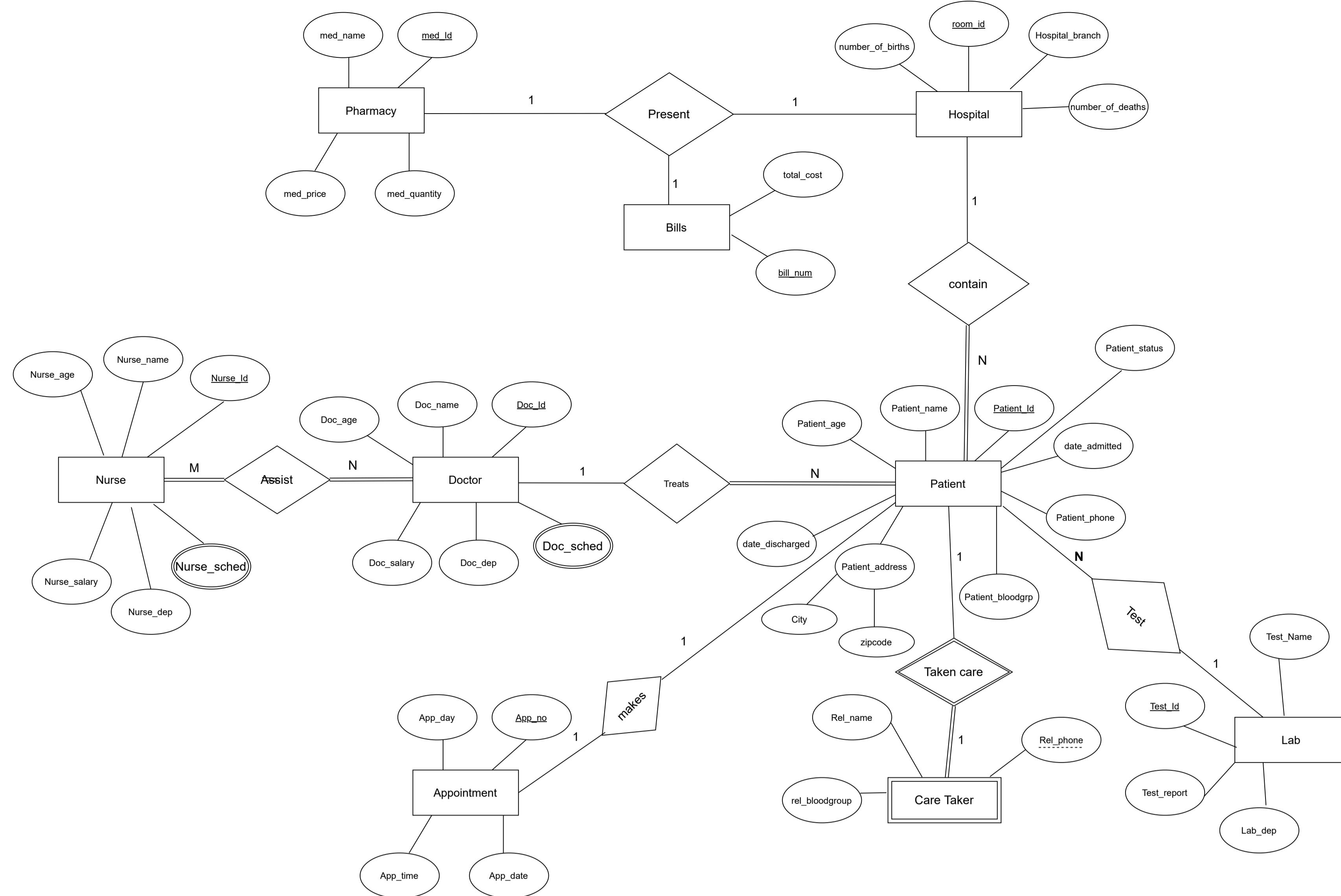
5. EER MODEL (CONCEPTUAL MODEL)



6. Global Conceptual Schema:



7. ER Diagram:



NORMALIZATION

1st Normal Form: A relation is said to be in first normal if it satisfy the following:

- No multi-valued attribute
- No composite attribute
- Identify primary key

Here, the relationship is converted to either relation or foreign key or merging relations.

Foreign Key : Giving primary key of one table as a reference to another table.

Pharmacy:

<u>Med-id</u>	Med-name	Med-price	Med-quantity

Hospital:

<u>Room-id</u>	hospital-name	No-of-deaths	No-of-Births

Nurse:

<u>Nurse-id</u>	Nurse-name	Nurse-age	Nurse-salary	Nurse-dep

Nurse-Sched is a multivalued attribute, so a table is created for Nurse-sched with Nurse-id as foreign key.

Nurse-Schedule:

<u>Nurse-id</u>	<u>Nurse-sched</u>

Doctor:

<u>Doc-id</u>	Doc-name	Doc-age	Doc-salary	Doc-dep

Doc-sched is a multivalued attribute. So a table is created for Doctor Schedule with Doctor Id as foreign key.

Doctor-Schedule:

<u>Doc_id</u>	<u>Doc-sched</u>

Assist:

<u>Nurse_id</u>	<u>Doc_id</u>

Patient:

<u>Patient_id</u>	<u>Patient-name</u>	<u>Patient-age</u>	<u>City</u>	<u>Zipcode</u>	<u>Patient-bloodgrp</u>	<u>Patient-phone</u>

<u>Date-admit</u>	<u>Date-discharge</u>	<u>Hos-Room-no</u>	<u>Doc_id</u>	<u>Lab-Test-id</u>

Appointment:

<u>App-no</u>	<u>App-day</u>	<u>App-time</u>	<u>App-date</u>	<u>App-Patid</u>

Lab:

<u>Test_id</u>	<u>Test-name</u>	<u>Lab-dep</u>

Bill:

<u>Bill_no</u>	<u>Bill-cost</u>

Caretaker:

<u>RPat_id</u>	<u>Rel-phone</u>	<u>Rel-name</u>	<u>Rel-bloodgrp</u>

Present:

<u>Med_id</u>	<u>Bill_no</u>	<u>Room_no</u>

Outcomes of 2nd normalization:

- Primary key has been identified in each table using closure property.
- Composite attributes has been resolved
- Multi-valued attributes has been resolved.

2nd Normal Form:

- i) Repeating column values are taken out and maintained in a separate table. So that change can be done only once in the new table rather than all entries in the first table. Rule is foreign key must be on N side else again multi-value in a column will occur.
- ii) Identify prime attribute (part of candidate key that determines anything else) it is also called partial dependency and eliminate it. Because, 2nd NF is based on Full Functional dependency (key should determine all other attributes in a table)
- iii) Use foreign key on many side.

There are no partial dependencies in our hospital management schema.

3rd Normal Form:

- Only column with direct dependency of the primary key shall be in the entity.
- No transitive dependencies: Non-prime attributes transitively depending on the key.

$$A \rightarrow B \quad B \rightarrow C \Rightarrow A \rightarrow C$$
- 3rd NF should hold the condition that:
 if $X \rightarrow Y$ then Either X is super key or Y is a prime attribute. Following this condition will never allow transitive dependency.

Appointment:

App-no	App-date	App-day	App-Time	App-patid
19	2021-04-28	Wednesday	02:15:00	122
39	2021-04-13	Tuesday	01:30:00	119
45	2021-04-05	Monday	01:15:00	117
46	2021-04-17	Saturday	03:05:00	118
49	2021-04-05	Monday	03:45:00	125
51	2021-04-13	Tuesday	02:25:00	123
54	2021-04-23	Friday	01:45:00	120
59	2021-04-10	Saturday	03:15:00	124
61	2021-04-15	Thursday	04:30:00	126
97	2021-04-12	Monday	02:00:00	121

Here $\text{App-no} \rightarrow \text{App-date}, \text{App-Time}, \text{App-patid}$
 $\text{Date} \rightarrow \text{Day}$

Hence transitive dependency exists.

It is resolved by creating a table for Date and Day

Appointment

App-no	App-date	App-Time	App-patid.
19	2021-04-28	02:15:00	122
39	2021-04-13	01:30:00	119
45	2021-04-05	01:15:00	117
46	2021-04-17	03:05:00	118
49	2021-04-05	03:45:00	125
51	2021-04-13	02:25:00	123
54	2021-04-23	01:45:00	120
59	2021-04-10	03:15:00	124
61	2021-04-15	04:30:00	126
97	2021-04-12	02:00:00	121

Dateday

a-date	day
2021-04-05	Monday
2021-04-10	Saturday
2021-04-12	Monday
2021-04-13	Tuesday
2021-04-15	Thursday
2021-04-17	Saturday
2021-04-23	Friday
2021-04-28	Wednesday

Patient:

Patient-id	Patient-name	Patient-age	Patient-status	zipcode	city	Patient-Bloodgrp	Patient-phone	Date-admit	Date-discharge	hos-room-no	doc-id	lab-test-id
117	Khushi	21	fever	121445	Delhi	A+	123456	2021-3-12	NULL	101	201	33
118	Shreya	29	pain	159632	Bangalore	A+	996587	2021-3-21	2021-4-25	258	210	21
119	Bharat	25	fever	789512	Pune	B+	786324	2021-3-26	NULL	712	202	31
120	Sagarika	15	fever	121445	Delhi	AB+	225588	2021-3-12	NULL	198	205	19
121	Abhi	31	pain	159632	Bangalore	B+	769851	2021-4-12	2021-4-22	312	208	25
122	Rohit	42	infection	159632	Bangalore	O+	325874	2021-3-2	2021-4-4	265	209	40
123	Aashi	65	sideeffects	125987	Hyderabad	A+	159636	2021-4-1	NULL	412	207	37
124	Ishan	34	fine	985472	Mumbai	O+	892479	2021-4-11	NULL	300	203	44
125	Deepa	28	pain	121445	Delhi	B-	854796	2021-3-31	2021-4-30	255	204	43
126	Ayush	23	unconscious	159632	Bangalore	A+	852030	2021-3-22	NULL	384	206	22

Patient-id → Patient-name, Patient-age, zipcode, Patient-Bloodgrp, Patient-phone, Date-admit, Date-discharge, hos-roomno, doc-id, lab-test-id.

zipcode → city

Hence transitive dependency exists.

It can be resolved by creating a table for zipcode and city

Patient:

patient_id	patient_name	patient_age	patient_status	zipcode	patient_Bloodgrp	patient_phone	date-admit	date-discharge	hos-room-no	doc-id	lab-testid
117	Kushi	29	fever	121445	A+	123456	2021-3-12	NULL	101	201	33
118	Shreya	21	pain	1519632	A+	996587	2021-3-21	2021-4-25	258	210	21
119	Bharat	25	fever	789512	B+	786324	2021-3-26	NULL	712	202	31
120	Sagarika	15	fever	121445	AB+	225588	2021-3-12	NULL	198	205	19
121	Abhi	31	pain	159632	B+	769851	2021-4-12	2021-4-22	312	208	95
122	Rohit	42	infection	159632	O+	325874	2021-3-2	2021-4-4	265	209	40
123	Aashi	65	sideeffects	125987	A+	159636	2021-3-31	2021-4-30	412	207	37
124	Ishan	34	fine	985472	O+	892479	2021-4-1	NULL	300	203	44
125	Deepa	28	pain	121445	B-	854796	2021-4-11	2021-4-30	255	204	43
126	Ayush	33	unconscious	159632	A+	852030	2021-3-22	NULL	384	206	22

City:

zipcode	city
121445	Delhi
125987	Hyderabad
159632	Bangalore
789512	Pune
985472	Mumbai

BCNF:

Every 3NF is not BCNF but if a table is in BCNF then it is already in 3NF.

BCNF says every LHS of FD's must be the key of one of the tables. That is, every prime attributes in a table. Therefore, the above tables satisfy BCNF.

4th NF is not required since we eliminate repeated values in the 3NF itself.

TABLES:

```
+-----+  
| Tables_in_health |  
+-----+  
| appointment      |  
| assist           |  
| bill             |  
| caretaker        |  
| city             |  
| dateday          |  
| doctor           |  
| doctor_schedule  |  
| hospital          |  
| labs             |  
| nurse            |  
| nurse_schedule   |  
| patient          |  
| pharmacy          |  
| present          |  
+-----+  
15 rows in set (0.01 sec)  
  
mysql> select *from appointment;  
+-----+-----+-----+-----+  
| app_no | app_time | app_date | app_patid |  
+-----+-----+-----+-----+  
| 19    | 02:15:00 | 2021-04-28 | 122     |  
| 39    | 01:30:00 | 2021-04-13 | 119     |  
| 45    | 01:15:00 | 2021-04-05 | 117     |  
| 46    | 03:05:00 | 2021-04-17 | 118     |  
| 49    | 03:45:00 | 2021-04-05 | 125     |  
| 51    | 02:25:00 | 2021-04-13 | 123     |  
| 54    | 01:45:00 | 2021-04-23 | 120     |  
| 59    | 03:15:00 | 2021-04-10 | 124     |  
| 61    | 04:30:00 | 2021-04-15 | 126     |  
| 97    | 02:00:00 | 2021-04-12 | 121     |  
+-----+-----+-----+-----+  
10 rows in set (0.01 sec)  
  
mysql> select *from assist;  
+-----+-----+  
| doc_id | nurse_id |  
+-----+-----+  
| 202    | 101     |  
| 203    | 102     |  
| 207    | 103     |  
| 209    | 104     |  
| 208    | 105     |  
| 204    | 106     |  
| 210    | 107     |  
| 206    | 108     |  
| 201    | 109     |  
| 205    | 110     |
```

```
mysql> select *from bill;
```

bill_no	bill_cost
1011	1500
1013	1020
1015	2500
1016	5500
1017	1950
1020	500
1025	1550
1027	2250
1028	1930
1029	1790

```
10 rows in set (0.01 sec)
```

```
mysql> select *from caretaker;
```

rpat_id	rel_phone	rel_name	rel_bloodgrp
117	1234	ankita	A+
118	3698	manmeen	AB+
119	7854	ram	O+
120	9851	sam	B+
121	2584	vidit	A-
122	2793	kartik	A+
123	7658	achint	B+
124	1597	raysha	A+
125	8753	niharika	O+
126	2468	sonali	B-

```
10 rows in set (0.01 sec)
```

```
mysql> select *from city;
```

zipcode	city
121445	delhi
125987	hyderabad
159632	bangalore
789512	pune
985472	mumbai

```
5 rows in set (0.01 sec)
```

```
mysql> select *from dateday;
```

a_date	day
2021-04-05	monday
2021-04-10	saturday
2021-04-12	monday
2021-04-13	tuesday
2021-04-15	thursday
2021-04-17	saturday
2021-04-23	friday
2021-04-28	wednesday

```
8 rows in set (0.01 sec)
```

```
mysql> select *from doctor;
```

doc_id	doc_name	doc_age	doc_salary	doc_dep
201	Rashmi	30	4055984	Oncologist
202	Tanisha	33	1272830	Allergist
203	Aditya	46	2839379	Anesthesiologist
204	Sanjay	38	45393021	Hematologist
205	Uday	36	65839022	Ophthalmologist
206	Smitha	42	45272911	Neurologist
207	Uma	44	3648202	Dermatologist
208	Shravya	45	5678907	Gastroenterologist
209	Shreesha	50	78291012	Endocrinologist
210	Sindhu	53	49200118	Cardiologist

```
10 rows in set (0.00 sec)
```

```
mysql> select *from doctor_schedule;
```

doc_id	doc_sched
201	01:00:00
201	15:00:00
202	06:30:00
203	11:00:00
204	12:45:00
204	17:00:00
205	09:35:00
206	08:15:00
206	12:00:00
207	09:00:00
208	10:55:00
208	16:00:00
209	08:20:00
210	07:25:00
210	18:00:00

```
mysql> select *from hospital;
+-----+-----+-----+-----+
| room_id | hospital_name | no_of_deaths | no_of_birth |
+-----+-----+-----+-----+
| 101 | healthcare hospital | 305 | 250 |
| 198 | manath hospital | 44 | 39 |
| 255 | anand hospital | 69 | 75 |
| 258 | starshine hospital | 20 | 15 |
| 265 | vasanth hospital | 28 | 37 |
| 300 | suncare hospital | 73 | 81 |
| 312 | getwell hospital | 92 | 90 |
| 384 | bharathi hospital | 30 | 51 |
| 412 | swaraj hospital | 127 | 110 |
| 712 | carewell hospital | 198 | 208 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

```
mysql> select *from labs;
+-----+-----+-----+
| test_id | test_name | lab_dep |
+-----+-----+-----+
| 19 | corneal topography | Ophthalmology |
| 21 | chest xray | cardiology |
| 22 | CT scan | neurology |
| 25 | endoscopy | gastroenterology |
| 31 | blood test | allergies |
| 33 | ultrasonogram | oncology |
| 37 | skin test | dermatology |
| 40 | bone density test | endocrinology |
| 43 | WBC count | hematology |
| 44 | PAC | anesthesiology |
+-----+-----+-----+
10 rows in set (0.01 sec)
```

```
mysql> select *from nurse;
+-----+-----+-----+-----+-----+
| nurse_id | nurse_name | nurse_age | nurse_salary | nurse_dep |
+-----+-----+-----+-----+-----+
| 101 | Sunaina | 30 | 500000 | allergist |
| 102 | Vasanth | 40 | 909200 | Anesthesiologist |
| 103 | Rahul | 28 | 1290300 | Dermatologist |
| 104 | Ram | 39 | 1292813 | Endocrinologist |
| 105 | Sham | 29 | 1827793 | Gastroenterologist |
| 106 | Raju | 32 | 364849 | Hematologist |
| 107 | Tanvi | 41 | 8935282 | Cardiologist |
| 108 | Nandana | 34 | 2844003 | Neurologist |
| 109 | Karpaga | 26 | 789674 | Oncologist |
| 110 | Kriti | 33 | 457328 | Ophthalmologist |
+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

```

mysql> select *from nurse_schedule;
+-----+-----+
| nurse_id | nurse_sched |
+-----+-----+
| 101 | 10:15:00 |
| 101 | 12:00:00 |
| 102 | 09:05:00 |
| 103 | 09:10:00 |
| 103 | 14:00:00 |
| 104 | 10:45:00 |
| 105 | 04:30:00 |
| 106 | 05:15:00 |
| 106 | 17:00:00 |
| 107 | 07:50:00 |
| 108 | 06:10:00 |
| 109 | 11:00:00 |
| 109 | 15:00:00 |
| 110 | 12:00:00 |
| 110 | 18:00:00 |
+-----+-----+
15 rows in set (0.00 sec)

mysql> select *from patient;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| patient_id | patient_name | patient_age | patient_status | zipcode | patient_bloodgrp | patient_phone | date_admitt | date_discharge | hos_room_no | doc_id | lab_test_id | 
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 117 | khushi | 21 | fever | 121445 | A+ | 123456 | 2021-03-12 | NULL | 101 | 201 | 33 |
| 118 | shreya | 29 | pain | 159632 | A+ | 996587 | 2021-03-21 | 2021-04-25 | 258 | 210 | 21 |
| 119 | bharat | 25 | fever | 789512 | B+ | 786324 | 2021-03-26 | NULL | 712 | 202 | 31 |
| 120 | sagarika | 15 | fever | 121445 | AB+ | 225588 | 2021-03-12 | NULL | 198 | 205 | 19 |
| 121 | abhi | 31 | pain | 159632 | B+ | 769851 | 2021-04-12 | 2021-04-22 | 312 | 208 | 25 |
| 122 | rohit | 42 | infection | 159632 | O+ | 325874 | 2021-03-02 | 2021-04-04 | 265 | 209 | 40 |
| 123 | aashi | 65 | side effects | 125987 | A+ | 159636 | 2021-03-31 | NULL | 412 | 207 | 37 |
| 124 | ishan | 34 | fine | 985472 | O+ | 892479 | 2021-04-01 | NULL | 300 | 203 | 44 |
| 125 | deepa | 28 | pain | 121445 | B- | 854796 | 2021-04-11 | 2021-04-30 | 255 | 204 | 43 |
| 126 | ayush | 33 | unconscious | 159632 | A+ | 852030 | 2021-03-22 | NULL | 384 | 206 | 22 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)

```

```

mysql> select *from pharmacy;
+-----+-----+-----+-----+
| med_id | med_name | med_price | med_quantity |
+-----+-----+-----+-----+
| 259 | paracetamol | 30 | 250 |
| 260 | crocin | 59 | 512 |
| 261 | disprin | 45 | 498 |
| 262 | brufen | 132 | 314 |
| 263 | acilok | 78 | 369 |
| 264 | wincold | 98 | 135 |
| 265 | cefaxime | 158 | 201 |
| 266 | actos | 112 | 125 |
| 267 | azithromycin | 204 | 74 |
| 268 | dimetapp | 158 | 169 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)

```

```

mysql> select *from present;
+-----+-----+-----+
| med_id | bill_no | room_no |
+-----+-----+-----+
| 259 | 1011 | 101 |
| 260 | 1013 | 384 |
| 261 | 1015 | 258 |
| 262 | 1016 | 712 |
| 263 | 1017 | 312 |
| 264 | 1020 | 198 |
| 265 | 1025 | 255 |
| 266 | 1027 | 300 |
| 267 | 1028 | 412 |
| 268 | 1029 | 265 |
+-----+-----+-----+
10 rows in set (0.00 sec)

```

Simple Queries:

1. Select the name and price of medicines which have price greater than 100.

```
mysql> select med_name,med_price from pharmacy where med_price>100;
+-----+-----+
| med_name      | med_price |
+-----+-----+
| brufen        |      132 |
| cefaxime      |      158 |
| actos         |      112 |
| azithromycin  |      204 |
| dimetapp      |      158 |
+-----+-----+
5 rows in set (0.00 sec)
```

2. Display the total number of doctors present whose age is lesser than 40.

```
mysql> select count(*) from doctor where doc_age<40;
+-----+
| count(*) |
+-----+
|        4 |
+-----+
1 row in set (0.01 sec)

mysql>
```

3. Display the number of patients of each blood group.

```
mysql> select patient_bloodgrp,count(patient_bloodgrp) from patient group by patient_bloodgrp;
+-----+-----+
| patient_bloodgrp | count(patient_bloodgrp) |
+-----+-----+
| A+                |                      4 |
| B+                |                      2 |
| AB+               |                      1 |
| O+                |                      2 |
| B-                |                      1 |
+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

4. Display the details of the patients who have not been discharged.

5.Display the average salary of all the nurses.

```
mysql> select avg(nurse_salary) from nurse;
+-----+
| avg(nurse_salary) |
+-----+
|      1921124.2000 |
+-----+
1 row in set (0.00 sec)

mysql>
```

6.Display the appointment details of the patient whose appointment is between 5th april 2021 and 20th april 2021 and sort by the appointment date.

```
mysql> select *from appointment where app_date between cast('2021-04-05' as date) and cast('2021-04-20' as date) order by app_date asc;
+-----+-----+-----+-----+
| app_no | app_time | app_date   | app_patid |
+-----+-----+-----+-----+
|    45  | 01:15:00  | 2021-04-05 |      117   |
|    49  | 03:45:00  | 2021-04-05 |      125   |
|    59  | 03:15:00  | 2021-04-10 |      124   |
|    97  | 02:00:00  | 2021-04-12 |      121   |
|    39  | 01:30:00  | 2021-04-13 |      119   |
|    51  | 02:25:00  | 2021-04-13 |      123   |
|    61  | 04:30:00  | 2021-04-15 |      126   |
|    46  | 03:05:00  | 2021-04-17 |      118   |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

7.Apply a 10% discount to the final bill.

```
mysql> select bill_no,(0.9*bill_cost) as discount_amt from bill;
+-----+-----+
| bill_no | discount_amt |
+-----+-----+
|    1011 |      1350.0 |
|    1013 |      918.0  |
|    1015 |      2250.0 |
|    1016 |      4950.0 |
|    1017 |      1755.0 |
|    1020 |      450.0  |
|    1025 |      1395.0 |
|    1027 |      2025.0 |
|    1028 |      1737.0 |
|    1029 |      1611.0 |
+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

8.Display the details of the three hospitals which have maximum number of deaths.

```
mysql> select *from hospital order by no_of_deaths desc limit 3;
+-----+-----+-----+
| room_id | hospital_name      | no_of_deaths | no_of_birth |
+-----+-----+-----+
|    101 | healthcare hospital |      305 |      250 |
|    712 | carewell hospital   |      198 |      208 |
|    412 | swaraj hospital     |      127 |      110 |
+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```

9.Display the distinct blood groups of all the caretakers.

```
mysql> select distinct rel_bloodgrp from caretaker;
+-----+
| rel_bloodgrp |
+-----+
| A+          |
| AB+         |
| O+          |
| B+          |
| A-          |
| B-          |
+-----+
6 rows in set (0.00 sec)
```

10.Display the id of all the nurses who have more than or equal to 2 schedules.

```
mysql> select nurse_id from nurse_schedule group by nurse_id having count(*)>=2;
+-----+
| nurse_id |
+-----+
|    101   |
|    103   |
|    106   |
|    109   |
|    110   |
+-----+
5 rows in set (0.00 sec)
```

Join:

- 1.Display patient id,patient name and the names and phone numbers of the relatives who take care of the patient.

```
mysql> select patient_id,patient_name,rel_name,rel_phone from patient,caretaker where patient_id=rpat_id;
+-----+-----+-----+
| patient_id | patient_name | rel_name | rel_phone |
+-----+-----+-----+
| 117 | khushi | ankita | 1234 |
| 118 | shreya | manmeen | 3698 |
| 119 | bharat | ram | 7854 |
| 120 | sagarika | sam | 9851 |
| 121 | abhi | vidit | 2584 |
| 122 | rohit | kartik | 2793 |
| 123 | aashi | achint | 7658 |
| 124 | ishan | raysha | 1597 |
| 125 | deepa | niharika | 8753 |
| 126 | ayush | sonali | 2468 |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

- 2.Display the patient name and the respective test which the patient has to undergo and the name of the doctor who is treating the patient.

```
mysql> select patient_name,doc_name,test_name from patient,doctor,labs where doctor.doc_id=patient.doc_id and patient.lab_test_id=labs.test_id;
+-----+-----+-----+
| patient_name | doc_name | test_name |
+-----+-----+-----+
| khushi | Rashmi | ultrasonogram |
| shreya | Sindhu | chest xray |
| bharat | Tanisha | blood test |
| sagarika | Uday | corneal topography |
| abhi | Shravya | endoscopy |
| rohit | Shreesha | bone density test |
| aashi | Uma | skin test |
| ishan | Aditya | PAC |
| deepa | Sanjay | WBC count |
| ayush | Smitha | CT scan |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

Correlated subqueries:

1. Display patient's id, name and zipcode if and only if one or more patients are from Bangalore.

```
mysql> select patient_id,patient_name,zipcode from patient where exists (select *from patient,city where patient.zipcode=city.zipcode and city.city='bangalore');
+-----+-----+-----+
| patient_id | patient_name | zipcode |
+-----+-----+-----+
|    117 | khushi      | 121445   |
|    118 | shreya     | 159632   |
|    119 | bharat     | 789512   |
|    120 | sagarika   | 121445   |
|    121 | abhi       | 159632   |
|    122 | rohit     | 159632   |
|    123 | aashi      | 125987   |
|    124 | ishan      | 985472   |
|    125 | deepa      | 121445   |
|    126 | ayush      | 159632   |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

2. Display the hospital details if there is no hospital with number of deaths greater than 500

```
mysql> select *from hospital where not exists(select *from hospital where no_of_deaths>500);
+-----+-----+-----+-----+
| room_id | hospital_name | no_of_deaths | no_of_birth |
+-----+-----+-----+-----+
|    101 | healthcare hospital | 305 | 250 |
|    198 | manath hospital | 44 | 39 |
|    255 | anand hospital | 69 | 75 |
|    258 | starshine hospital | 20 | 15 |
|    265 | vasanth hospital | 28 | 37 |
|    300 | suncare hospital | 73 | 81 |
|    312 | getwell hospital | 92 | 90 |
|    384 | bharathi hospital | 30 | 51 |
|    412 | swaraj hospital | 127 | 110 |
|    712 | carewell hospital | 198 | 208 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

3. Display the details of the doctors who have 2 schedules.

```
mysql> select * from doctor where exists(select *from doctor_schedule where doctor.doc_id=doctor_schedule.doc_id group by doc_id having count(*)=2);
+-----+-----+-----+-----+
| doc_id | doc_name | doc_age | doc_salary | doc_dep |
+-----+-----+-----+-----+
|    201 | Rashmi | 30 | 4055984 | Oncologist |
|    204 | Sanjay | 38 | 45393021 | Hematologist |
|    206 | Smitha | 42 | 45272911 | Neurologist |
|    208 | Shravya | 45 | 5678907 | Gastroenterologist |
|    210 | Sindhу | 53 | 49200118 | Cardiologist |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Nested queries:

1. Display the details of the doctors who have salary greater than the salaries of all the nurses

```
mysql> select *from doctor where doc_salary>all(select nurse_salary from nurse);
+-----+-----+-----+-----+
| doc_id | doc_name | doc_age | doc_salary | doc_dep
+-----+-----+-----+-----+
| 204 | Sanjay | 38 | 45393021 | Hematologist |
| 205 | Uday | 36 | 65839022 | Ophthalmologist |
| 206 | Smitha | 42 | 45272911 | Neurologist |
| 209 | Shreesha | 50 | 78291012 | Endocrinologist |
| 210 | Sindhu | 53 | 49200118 | Cardiologist |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

2. Display the appointment details where the day of the appointment is a Monday.

```
mysql> select *from appointment where app_date in (select a_date from dateday where day='monday');
+-----+-----+-----+
| app_no | app_time | app_date | app_patid
+-----+-----+-----+
| 45 | 01:15:00 | 2021-04-05 | 117 |
| 49 | 03:45:00 | 2021-04-05 | 125 |
| 97 | 02:00:00 | 2021-04-12 | 121 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

3. Select the details of the oldest nurse.

```
mysql> select *from nurse where nurse_age=(select max(nurse_age) from nurse);
+-----+-----+-----+-----+
| nurse_id | nurse_name | nurse_age | nurse_salary | nurse_dep
+-----+-----+-----+-----+
| 107 | Tanvi | 41 | 8935282 | Cardiologist |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

4.Display the details of the doctors who are available after 12:00:00

```
mysql> select *from doctor where doc_id=any( select doc_id from doctor_schedule where doc_sched>'12:00:00');
+-----+-----+-----+-----+
| doc_id | doc_name | doc_age | doc_salary | doc_dep
+-----+-----+-----+-----+
| 201 | Rashmi | 30 | 4055984 | Oncologist
| 204 | Sanjay | 38 | 45393021 | Hematologist
| 208 | Shravya | 45 | 5678907 | Gastroenterologist
| 210 | Sindhu | 53 | 49200118 | Cardiologist
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

5.Display the medicine details whose price is greater than all the medicines whose quantity is greater than 300

```
mysql> select *from pharmacy where med_price>all(select med_price from pharmacy where med_quantity>300);
+-----+-----+-----+
| med_id | med_name | med_price | med_quantity |
+-----+-----+-----+
| 265 | cefaxime | 158 | 201 |
| 267 | azithromycin | 204 | 74 |
| 268 | dimetapp | 158 | 169 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

VIEWS:

- 1.Create a view containing the caretaker's name and id.

```
mysql> create or replace view care as select rpat_id,rel_name from caretaker;
Query OK, 0 rows affected (0.05 sec)

mysql> select *from care;
+-----+-----+
| rpat_id | rel_name |
+-----+-----+
| 117 | ankita |
| 118 | marmeen |
| 119 | ram |
| 120 | sam |
| 121 | vidit |
| 122 | kartik |
| 123 | achint |
| 124 | raysha |
| 125 | niharika |
| 126 | sonali |
+-----+-----+
10 rows in set (0.00 sec)
```

- 2.Create a view containing the patients name,phone number and the date,time of the patient's appointment.

```
mysql> create or replace view patient_appointment as select patient_name,patient_phone,app_time as appointment_time,app_date as appointment_date from patient,appointment where appointment.app_patid=patient.patient_id;
Query OK, 0 rows affected (0.08 sec)

mysql> select *from patient_appointment;
+-----+-----+-----+-----+
| patient_name | patient_phone | appointment_time | appointment_date |
+-----+-----+-----+-----+
| khushi | 123456 | 01:15:00 | 2021-04-05 |
| shreya | 996587 | 03:05:00 | 2021-04-17 |
| bharat | 786324 | 01:30:00 | 2021-04-13 |
| sagarika | 225588 | 01:45:00 | 2021-04-23 |
| abhi | 769851 | 02:00:00 | 2021-04-12 |
| rohit | 325874 | 02:15:00 | 2021-04-28 |
| aashi | 159636 | 02:25:00 | 2021-04-13 |
| ishan | 892479 | 03:15:00 | 2021-04-10 |
| deepa | 854796 | 03:45:00 | 2021-04-05 |
| ayush | 852030 | 04:30:00 | 2021-04-15 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

- 3.Create a view which contains the name of the doctor, nurse who is assisting the doctor and the department they are in.Sort by the department.

```
mysql> create or replace view doc_nurse as select doc_name as doctor_name,nurse_name,doc_dep as department from doctor,nurse,assist where doctor.doc_id=assist.doc_id and nurse.nurse_id=assist.nurse_id order by department;
Query OK, 0 rows affected (0.06 sec)

mysql> select *from doc_nurse;
+-----+-----+-----+
| doctor_name | nurse_name | department |
+-----+-----+-----+
| Tanisha | Sunaina | Allergist |
| Aditya | Vasanth | Anesthesiologist |
| Sindhu | Tanvi | Cardiologist |
| Uma | Rahul | Dermatologist |
| Shreesta | Ravi | Endocrinologist |
| Shravya | Sham | Gastroenterologist |
| Sanjay | Raju | Hematologist |
| Smitha | Nandana | Neurologist |
| Rashmi | Karpaga | Oncologist |
| Uday | Kriti | Ophthalmologist |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

4.Create a view containing the nurse's name and schedule.

```
mysql> create or replace view nurse_details as select nurse_name,nurse_sched from nurse,nurse_schedule where nurse.nurse_id=nurse_schedule.nurse_id;
Query OK, 0 rows affected (0.12 sec)

mysql> select *from nurse_details;
+-----+-----+
| nurse_name | nurse_sched |
+-----+-----+
| Sunaina   | 10:15:00
| Sunaina   | 12:00:00
| Vasanth    | 09:05:00
| Rahul      | 09:10:00
| Rahul      | 14:00:00
| Ram        | 10:45:00
| Sham       | 04:30:00
| Raju       | 05:15:00
| Raju       | 17:00:00
| Tanvi      | 07:50:00
| Nandana    | 06:10:00
| Karpaga    | 11:00:00
| Karpaga    | 15:00:00
| Kriti      | 12:00:00
| Kriti      | 18:00:00
+-----+-----+
15 rows in set (0.01 sec)

mysql>
```

5.Create a view which contains the status of patient and number of patients of the same status .

```
mysql> create or replace view patient_details as select patient_status,count(*) as number_of_patients from patient group by patient_status;
Query OK, 0 rows affected (0.09 sec)

mysql> select *from patient_details;
+-----+-----+
| patient_status | number_of_patients |
+-----+-----+
| fever          |            3 |
| pain           |            3 |
| infection      |            1 |
| side effects   |            1 |
| fine           |            1 |
| unconscious    |            1 |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

FUNCTIONS:

1.Create a function to find whether a patient is major.

```
MySQL 8.0 Command Line Client
mysql> DELIMITER $$
mysql> CREATE FUNCTION major_minor(age INT)
-> RETURNS VARCHAR(20)
-> DETERMINISTIC
-> BEGIN
->     DECLARE division VARCHAR(20);
->     IF age > 18 THEN SET division = 'MAJOR';
->     ELSE SET division = 'MINOR';
->     END IF;
->     RETURN (division);
-> END $$
Query OK, 0 rows affected (0.17 sec)

mysql> DELIMITER ;
mysql> SELECT patient_id, patient_name, patient_age, major_minor(patient_age) FROM patient;
+-----+-----+-----+-----+
| patient_id | patient_name | patient_age | major_minor(patient_age) |
+-----+-----+-----+-----+
|      117 | khushi       |      21 | MAJOR
|      118 | shreya       |      29 | MAJOR
|      119 | bharat       |      25 | MAJOR
|      120 | sagarika     |      15 | MINOR
|      121 | abhi         |      31 | MAJOR
|      122 | rohit         |      42 | MAJOR
|      123 | aashi         |      65 | MAJOR
|      124 | ishan         |      34 | MAJOR
|      125 | deepa         |      28 | MAJOR
|      126 | ayush         |      33 | MAJOR
+-----+-----+-----+-----+
10 rows in set (0.26 sec)
```

2.Create a function to calculate the GST on medicines.

```
MySQL 8.0 Command Line Client
mysql> DELIMITER $$
mysql> CREATE FUNCTION GST(price INT)
-> RETURNS INT
-> DETERMINISTIC
-> BEGIN
->     DECLARE gst_price INT;
->     IF price < 100 THEN SET gst_price = 0;
->     ELSEIF (price >=100 AND price < 500) THEN SET gst_price = 0.1*price;
->     ELSE SET gst_price = 0.25*price;
->     END IF;
->     RETURN (gst_price);
-> END$$
Query OK, 0 rows affected (0.10 sec)

mysql> DELIMITER ;
mysql> SELECT med_id, med_name, med_price, med_quantity, GST(med_price) AS gst, med_price+GST(med_price) total_amt FROM pharmacy;
+-----+-----+-----+-----+-----+-----+
| med_id | med_name | med_price | med_quantity | gst | total_amt |
+-----+-----+-----+-----+-----+-----+
|    259 | paracetamol |      30 |      250 |    0 |      30 |
|    260 | crocin     |      59 |      512 |    0 |      59 |
|    261 | disprin    |      45 |      498 |    0 |      45 |
|    262 | brufen     |     132 |      314 |   13 |     145 |
|    263 | acilok     |      78 |      369 |    0 |      78 |
|    264 | wincold    |      98 |      135 |    0 |      98 |
|    265 | cefaxime   |     158 |      201 |   16 |     174 |
|    266 | actos      |     112 |      125 |   11 |     123 |
|    267 | azithromycin |     204 |      74 |   20 |     224 |
|    268 | dimetapp   |     158 |      169 |   16 |     174 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.13 sec)

mysql>
```

3.Create a function to calculate bonus salary for each nurse.

```
MySQL 8.0 Command Line Client

mysql> DELIMITER $$  
mysql> CREATE FUNCTION Bonus(age INT)  
    -> RETURNS INT  
    -> DETERMINISTIC  
    -> BEGIN  
    ->     DECLARE bonus_amt INT;  
    ->     IF age > 30 THEN SET bonus_amt = 10000;  
    ->     ELSE SET bonus_amt = 0;  
    ->     END IF;  
    ->     RETURN (bonus_amt);  
    -> END$$  
Query OK, 0 rows affected (0.10 sec)

mysql> DELIMITER ;  
mysql> SELECT nurse_id, nurse_name, Bonus(nurse_age) FROM nurse;  
+-----+-----+-----+  
| nurse_id | nurse_name | Bonus(nurse_age) |  
+-----+-----+-----+  
| 101 | Sunaina | 0 |  
| 102 | Vasanth | 10000 |  
| 103 | Rahul | 0 |  
| 104 | Ram | 10000 |  
| 105 | Sham | 0 |  
| 106 | Raju | 10000 |  
| 107 | Tanvi | 10000 |  
| 108 | Nandana | 10000 |  
| 109 | Karpaga | 0 |  
| 110 | Kriti | 10000 |  
+-----+-----+-----+  
10 rows in set (0.42 sec)
```

4.Create a function to calculate the discount on the bill amount.

```
MySQL 8.0 Command Line Client

mysql> DELIMITER $$  
mysql> CREATE FUNCTION discount(cost INT)  
    -> RETURNS INT  
    -> DETERMINISTIC  
    -> BEGIN  
    ->     DECLARE discount INT;  
    ->     IF (cost >= 1000 AND cost < 2000) THEN SET discount = 0.05*cost;  
    ->     ELSEIF cost >=2000 THEN SET discount = 0.1*cost;  
    ->     ELSE SET discount = 0;  
    ->     END IF;  
    ->     RETURN (discount);  
    -> END$$  
Query OK, 0 rows affected (0.67 sec)

mysql> DELIMITER ;  
mysql> SELECT bill_no, bill_cost, discount(bill_cost), bill_cost-discount(bill_cost) AS Bill_amount FROM bill;  
+-----+-----+-----+-----+  
| bill_no | bill_cost | discount(bill_cost) | Bill_amount |  
+-----+-----+-----+-----+  
| 1011 | 1500 | 75 | 1425 |  
| 1013 | 1020 | 51 | 969 |  
| 1015 | 2500 | 250 | 2250 |  
| 1016 | 5500 | 550 | 4950 |  
| 1017 | 1950 | 98 | 1852 |  
| 1020 | 500 | 0 | 500 |  
| 1025 | 1550 | 78 | 1472 |  
| 1027 | 2250 | 225 | 2025 |  
| 1028 | 1930 | 97 | 1833 |  
| 1029 | 1790 | 90 | 1700 |  
+-----+-----+-----+-----+  
10 rows in set (0.12 sec)
```

5.Create a function to set the doctor level based on salary.

```
MySQL 8.0 Command Line Client

mysql> DELIMITER $$  
mysql> CREATE FUNCTION doctor_level( salary INT)  
-> RETURNS VARCHAR(20)  
-> DETERMINISTIC  
-> BEGIN  
->     DECLARE level VARCHAR(20);  
->     IF salary < 5000000 THEN SET level = 'SILVER';  
->     ELSEIF (salary >= 5000000 AND salary <50000000) THEN SET level = 'GOLD';  
->     ELSEIF salary >=50000000 THEN SET level = 'PLATINUM';  
->     END IF;  
->     RETURN (level);  
-> END$$  
Query OK, 0 rows affected (0.68 sec)

mysql> DELIMITER ;  
mysql> SELECT doc_id, doc_name, doctor_level(doc_salary) AS doc_level FROM doctor;  
+-----+-----+-----+  
| doc_id | doc_name | doc_level |  
+-----+-----+-----+  
| 201   | Rashmi  | SILVER  |  
| 202   | Tanisha | SILVER  |  
| 203   | Aditya  | SILVER  |  
| 204   | Sanjay  | GOLD    |  
| 205   | Uday    | PLATINUM|  
| 206   | Smitha  | GOLD    |  
| 207   | Uma     | SILVER  |  
| 208   | Shravya | GOLD    |  
| 209   | Shreesha| PLATINUM|  
| 210   | Sindhu  | GOLD    |  
+-----+-----+-----+  
10 rows in set (0.43 sec)
```

STORED PROCEDURES

1. DISPLAY ALL DETAILS OF THE CARETAKER OF THE PATIENT FROM THE PATIENT ID.

```
MySQL 8.0 Command Line Client
mysql> DELIMITER $$

mysql> create procedure caretaker_details( in c_id int)
    -> begin
    -> select * from caretaker
    -> where rpat_id=c_id;
    -> end $$
Query OK, 0 rows affected (0.45 sec)

mysql> DELIMITER ;
mysql> call caretaker_details(117);
+-----+-----+-----+
| rpat_id | rel_phone | rel_name | rel_bloodgrp |
+-----+-----+-----+
|     117 |      1234 |   ankita |       A+ |
+-----+-----+-----+
1 row in set (0.23 sec)

Query OK, 0 rows affected (0.25 sec)
```

2. DISPLAY THE MEDICINE NAME FROM THE MEDICINE ID.

```
mysql> DELIMITER $$

mysql> create procedure medicine_name( in id int, out name varchar(30))
    -> begin
    -> select med_name into name from pharmacy
    -> where med_id=id;
    -> end $$
Query OK, 0 rows affected (0.13 sec)
```

```
mysql> DELIMITER ;
mysql> call medicine_name (264, @name);
ERROR 1305 (42000): PROCEDURE project.medicine_name does not exist
mysql> call medicine_name (264, @name);
Query OK, 1 row affected (0.09 sec)

mysql> select(@name);
+-----+
| (@name) |
+-----+
| wincold |
+-----+
1 row in set (0.00 sec)
```

3. DISPLAY STATUS OF THE DOCTOR FROM DOCTOR ID.

```
mysql> create procedure senior_doctor( in id int, out status varchar(20))
    -> begin
    -> declare age int default 0;
    -> select doc_age into age from doctor
    -> where doc_id=id;
    -> if age > 50 then set status= 'senior';
    -> else
    -> set status='not senior';
    -> end if;
    -> end $$
Query OK, 0 rows affected (0.17 sec)

mysql> DELIMITER ;
mysql> call senior_doctor(210,@status);
Query OK, 1 row affected (0.07 sec)

mysql> select(@status);
+-----+
| (@status) |
+-----+
| senior   |
+-----+
1 row in set (0.00 sec)
```

4. DISPLAY THE NAME OF THE LAB TEST OF THE GIVEN PATIENT.

```
mysql> DELIMITER $$  
mysql> create procedure lab( in id int )  
    -> begin  
    -> select l.test_name, p.patient_name from patient p, labs l  
    -> where id=p.patient_id and p.lab_test_id=l.test_id;  
    -> end $$  
Query OK, 0 rows affected (0.10 sec)  
  
mysql> DELIMITER ;  
mysql> call lab(29);  
Empty set (0.05 sec)  
  
Query OK, 0 rows affected (0.05 sec)  
  
mysql> call lab(118);  
+-----+-----+  
| test_name | patient_name |  
+-----+-----+  
| chest xray | shreya |  
+-----+-----+  
1 row in set (0.09 sec)  
  
Query OK, 0 rows affected (0.09 sec)  
  
mysql> call lab(120);  
+-----+-----+  
| test_name | patient_name |  
+-----+-----+  
| corneal topography | sagarika |  
+-----+-----+  
1 row in set (0.00 sec)
```

5. DISPLAY THE NAMES OF THE DOCTORS AND NURSES WORKING IN GIVEN DEPARTEMENT.

```
mysql> DELIMITER $$  
mysql> create procedure name( in dept varchar(20) )  
    -> begin  
    -> select d.doc_name, n.nurse_name from doctor d, nurse n  
    -> where d.doc_dep=dept and n.nurse_dep=dept;  
    -> end $$  
Query OK, 0 rows affected (0.12 sec)  
  
mysql> DELIMITER ;  
mysql> call name('cardiologist');  
+-----+-----+  
| doc_name | nurse_name |  
+-----+-----+  
| Sindhu | Tanvi |  
+-----+-----+  
1 row in set (0.00 sec)  
  
Query OK, 0 rows affected (0.03 sec)  
  
mysql> call name('dermatologist');  
+-----+-----+  
| doc_name | nurse_name |  
+-----+-----+  
| Uma | Rahul |  
+-----+-----+  
1 row in set (0.00 sec)  
  
Query OK, 0 rows affected (0.04 sec)
```

TRIGGERS

1. CHECK THE BILL AMOUNT BEFORE INSERTING IN TABLE.

```
mysql> DELIMITER $$  
mysql> create trigger bill_amt before insert  
    -> on bill for each row  
    -> begin  
    -> if new.bill_cost<0 then  
    -> set new.bill_cost=500;  
    -> end if;  
    -> end $$  
Query OK, 0 rows affected (0.39 sec)  
  
mysql> DELIMITER ;  
mysql> insert into bill values(1031,0);  
Query OK, 1 row affected (0.34 sec)  
  
mysql> select * from bill;  
+-----+-----+  
| bill_no | bill_cost |  
+-----+-----+  
| 1011   |    1500 |  
| 1013   |    1020 |  
| 1015   |    2500 |  
| 1016   |    5500 |  
| 1017   |    1950 |  
| 1020   |     500 |  
| 1025   |    1550 |  
| 1027   |    2250 |  
| 1028   |    1930 |  
| 1029   |    1790 |  
| 1031   |     500 |  
+-----+-----+  
11 rows in set (0.03 sec)
```

2. CREATE A TRIGGER ALLOWING THE UPDATE OF MEDICINE QUANTITY ONLY IF THE NEW VALUE IS GREATER THAN ZERO.

```
mysql> delimiter //  
mysql> create trigger quan_update before update  
    -> on pharmacy for each row  
    -> begin  
    -> if new.med_quantity<=0 then  
    -> set new.med_quantity=old.med_quantity;  
    -> end if;  
    -> end //  
Query OK, 0 rows affected (0.14 sec)  
  
mysql> delimiter ;  
mysql> update pharmacy set med_quantity=0 where med_id=267;  
Query OK, 0 rows affected (0.05 sec)  
Rows matched: 1  Changed: 0  Warnings: 0  
  
mysql> select *from pharmacy;  
+-----+-----+-----+-----+  
| med_id | med_name      | med_price | med_quantity |  
+-----+-----+-----+-----+  
| 259   | paracetamol    |    30      |      250     |  
| 260   | crocin        |    59      |      512     |  
| 261   | disprin       |    45      |      498     |  
| 262   | brufen         |   132      |      314     |  
| 263   | acilok        |    78      |      369     |  
| 264   | wincold        |    98      |      135     |  
| 265   | cefaxime       |   158      |      201     |  
| 266   | actos          |   112      |      125     |  
| 267   | azithromycin   |   204      |       74     |  
| 268   | dimetapp       |   158      |      169     |  
+-----+-----+-----+-----+  
10 rows in set (0.00 sec)
```

3. TRIGGER TO STORE OLD DETAILS OF THE DOCTOR BEFORE UPDATING THE VALUES.

```
MySQL 8.0 Command Line Client

mysql> CREATE TABLE Doc_log(doc_id INT PRIMARY KEY, doc_name VARCHAR(20), old_salary INT);
Query OK, 0 rows affected (1.10 sec)

mysql> DELIMITER $$ 
mysql> CREATE TRIGGER Doc_trigger
    -> BEFORE update ON doctor
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO Doc_log VALUES(old.doc_id, old.doc_name, old.doc_salary);
    -> END$$
Query OK, 0 rows affected (0.11 sec)

mysql> DELIMITER ;
mysql> UPDATE doctor SET doc_salary = 70000000 WHERE doc_id = 207;
Query OK, 1 row affected (0.43 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT *FROM doctor;
+-----+-----+-----+-----+-----+
| doc_id | doc_name | doc_age | doc_salary | doc_dep |
+-----+-----+-----+-----+-----+
| 201 | Rashmi | 30 | 12345678 | Oncologist |
| 202 | Tanisha | 33 | 1272830 | Allergist |
| 203 | Aditya | 46 | 2839379 | Anesthesiologist |
| 204 | Sanjay | 38 | 45393021 | Hematologist |
| 205 | Uday | 36 | 65839022 | Ophthalmologist |
| 206 | Smitha | 42 | 45272911 | Neurologist |
| 207 | Uma | 44 | 70000000 | Dermatologist |
| 208 | Shravya | 45 | 5678907 | Gastroenterologist |
| 209 | Shreesta | 50 | 78291012 | Endocrinologist |
| 210 | Sindhu | 53 | 49200118 | Cardiologist |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> SELECT *FROM doc_log;
+-----+-----+-----+
| doc_id | doc_name | old_salary |
+-----+-----+-----+
| 207 | Uma | 3648202 |
+-----+-----+-----+
1 row in set (0.00 sec)
```