DBMS MINI PROJECT

HOSPITAL MANAGEMENT SYSTEM

Submitted By:

Name: Ananya Nigam

SRN: PES1UG20CS044

V Semester Section A

Short Description and Scope of the Project

Hospital Management System is **a system enabling hospitals to manage information and data related to all aspects of healthcare – processes, providers, patients, and more**, which in turn ensures that processes are completed swiftly and effectively.

In this Hospital Management system I have created multiple tables to store information about patients, doctors, nurses, medicines, bills, treatments, etc. Using this sytem makes tracking and keeping records very time efficient and quick to search too.

This project has vast scope and can be extended to many more tables and functions to automate many tasks in hospital management and make it easy for the hospital staff to Keep records.
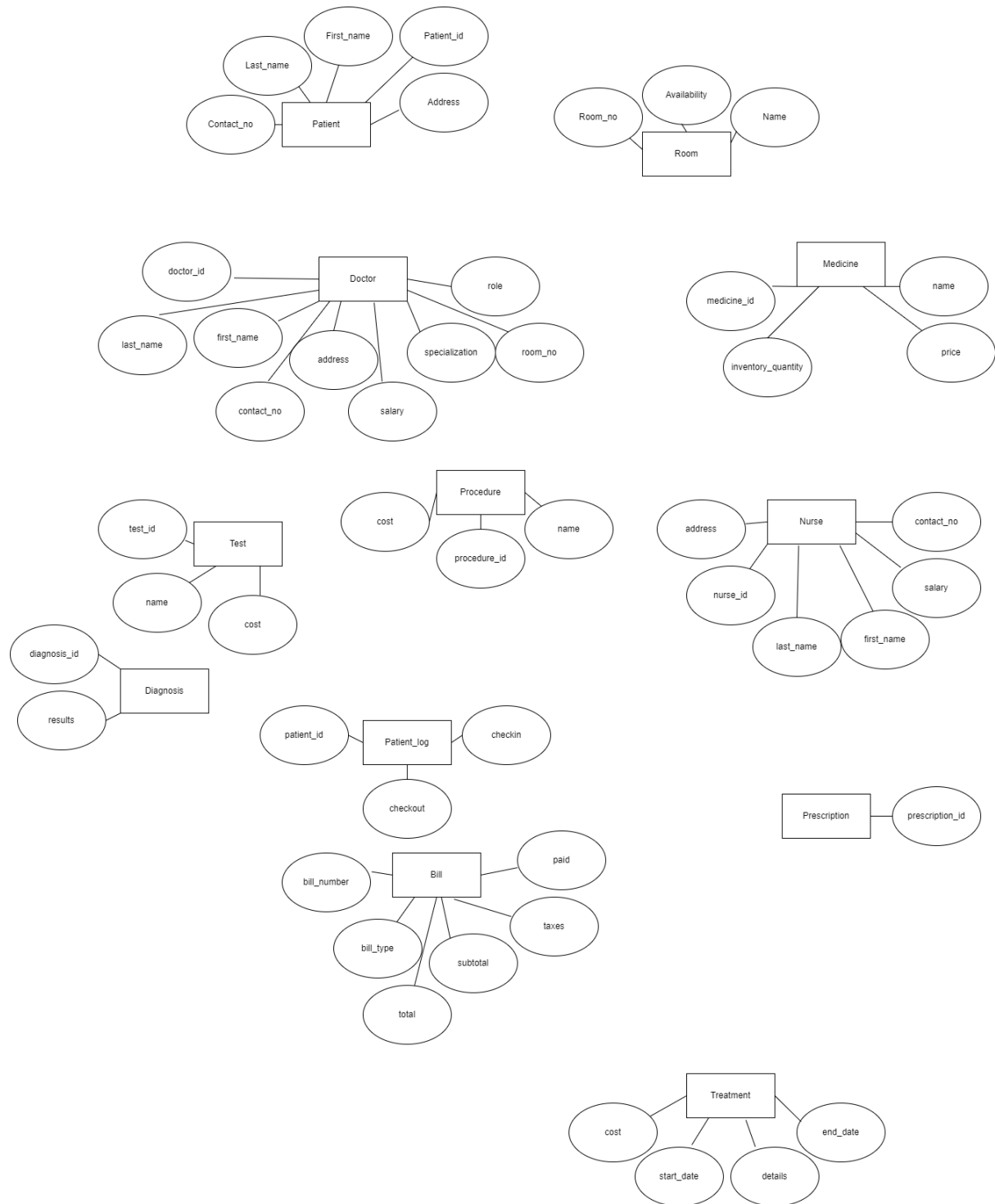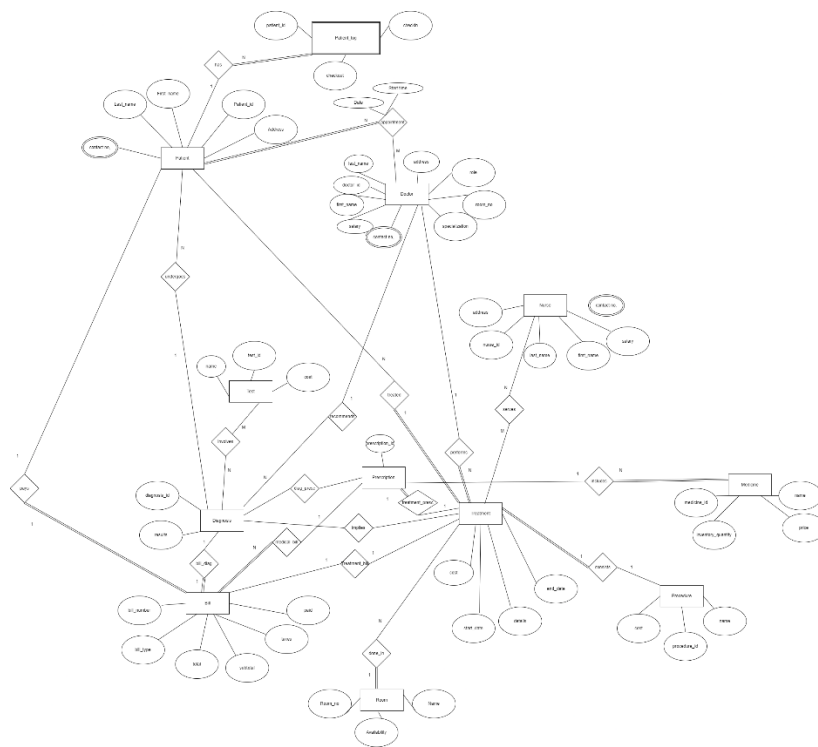
Languages used:

mySQL

Python

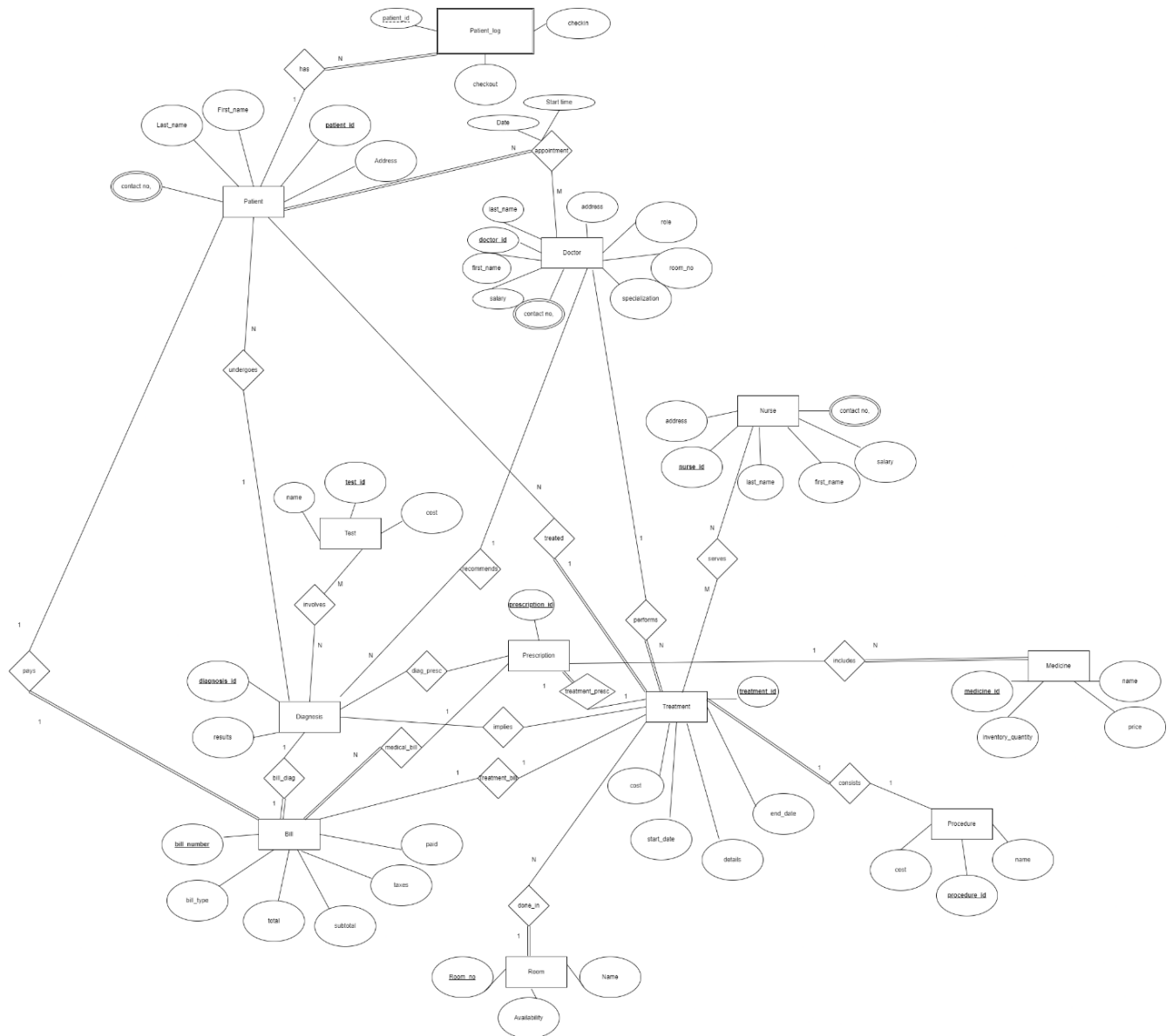Streamlit used for python and mysql connectivity to make front end.
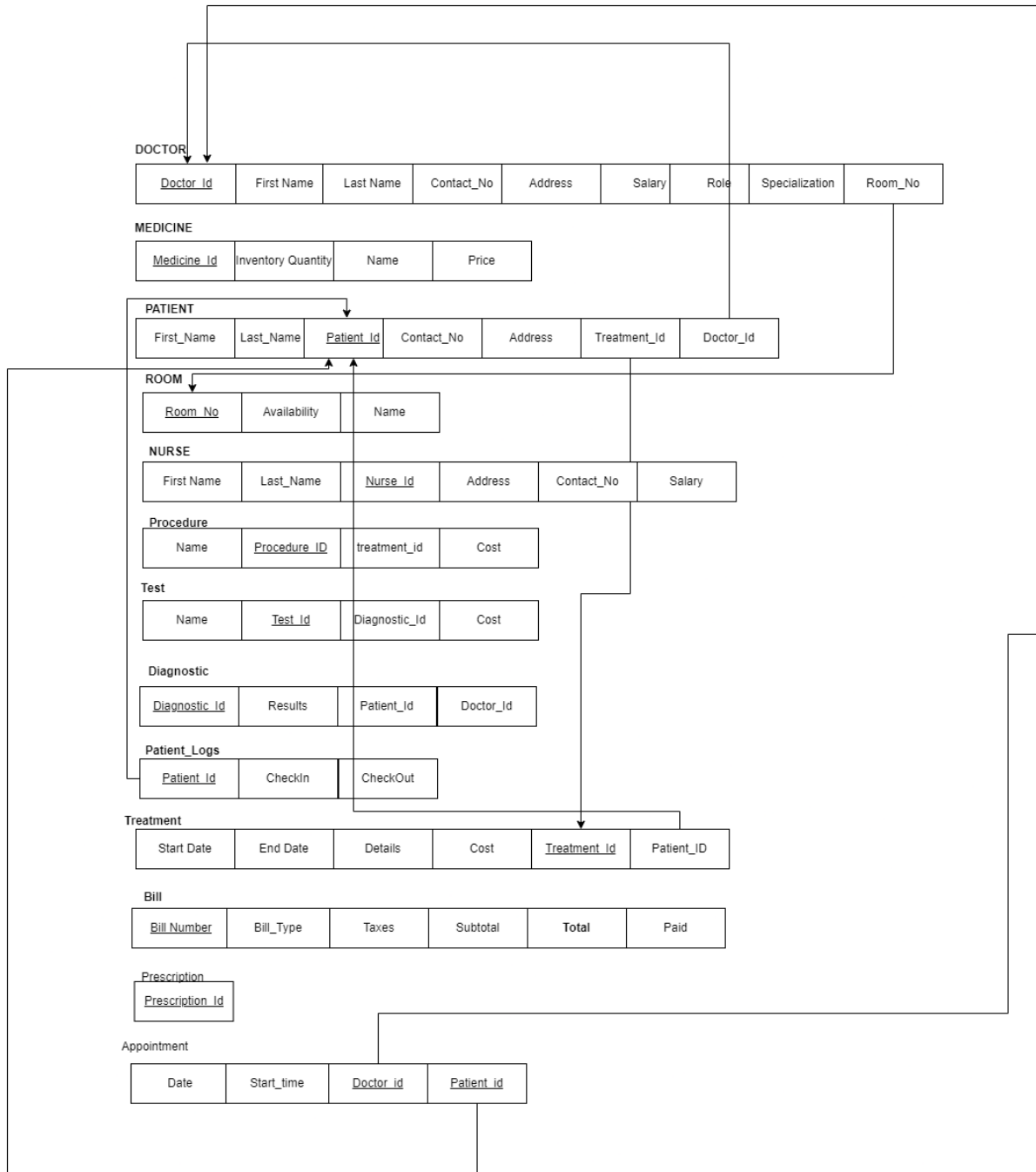
ER Diagram

Making ER Diagram:

Entities:

Patient entity with attributes: First_name, Patient_id, Last_name, Address, Contact_no

Room entity with attributes: Room_no, Availability, Name

Doctor entity with attributes: doctor_id, role, last_name, first_name, address, specialization, room_no, contact_no, salary

Medicine entity with attributes: medicine_id, name, inventory_quantity, price

Test entity with attributes: test_id, name, cost

Procedure entity with attributes: cost, name, procedure_id

Nurse entity with attributes: address, contact_no, nurse_id, salary, last_name, first_name

Diagnosis entity with attributes: diagnosis_id, results

Patient_log entity with attributes: patient_id, checkin, checkout

Prescription entity with attributes: prescription_id

Bill entity with attributes: bill_number, paid, bill_type, taxes, subtotal, total

Treatment entity with attributes: cost, start_date, details, end_date

ER DIAGRAM:

Relational Schema:

DOCTOR

Doctor_Id	First Name	Last Name	Contact_No	Address	Salary	Role	Specialization	Room_No

MEDICINE

Medicine_Id	Inventory Quantity	Name	Price

PATIENT

First_Name	Last_Name	Patient_Id	Contact_No	Address	Treatment_Id	Doctor_Id

ROOM

Room_No	Availability	Name

NURSE

First Name	Last_Name	Nurse_Id	Address	Contact_No	Salary

Procedure

Name	Procedure_ID	treatment_id	Cost

Test

Name	Test_Id	Diagnostic_Id	Cost

Diagnostic

Diagnostic_Id	Results	Patient_Id	Doctor_Id

Patient_Logs

Patient_Id	CheckIn	CheckOut

Treatment

Start Date	End Date	Details	Cost	Treatment_Id	Patient_ID

Bill

Bill Number	Bill_Type	Taxes	Subtotal	Total	Paid

Prescription

Prescription_Id

Appointment

Date	Start_time	Doctor_id	Patient_id

DDL COMMANDS

-- 1. Patient Entity

```sql
DROP TABLE IF EXISTS `Patient`;
CREATE TABLE `Patient` (
    `patient_id` int NOT NULL,
    `first_name` varchar(30),
    `last_name` varchar(30),
    `contact_no` varchar(15) UNIQUE,
    `address` varchar(100),
    PRIMARY KEY (`patient_id`)
);
```

-- 2. Room Entity

```sql
DROP TABLE IF EXISTS `Room`;
CREATE TABLE `Room` (
    `room_no` int NOT NULL,
    `name` varchar(30),
    `availability` bit,
    PRIMARY KEY (`room_no`)
);
```

-- 3. Doctor Entity

```sql
DROP TABLE IF EXISTS `Doctor`;
CREATE TABLE `Doctor` (
    `doctor_id` int NOT NULL AUTO_INCREMENT,
    `first_name` varchar(30),
    `last_name` varchar(30),
    `contact_no` varchar(15) UNIQUE,
    `address` varchar(50),
```

```sql
    `salary` decimal(10, 2),

    `specialization` varchar(30),

    `role` varchar(30),

    `room_no`int NOT NULL,

    PRIMARY KEY (`doctor_id`),

    FOREIGN KEY (`room_no`) REFERENCES `Room` ( `room_no` ) ON DELETE CASCADE
);


-- 4. Medicine Entity


DROP TABLE IF EXISTS `Medicine`;
CREATE TABLE `Medicine` (
    `medicine_id` int NOT NULL AUTO_INCREMENT,

    `inventory_quantity` int,

    `name` varchar(30),

    `price` decimal(10, 2),

    PRIMARY KEY (`medicine_id`)
);


-- 5. Nurse entity


DROP TABLE IF EXISTS `Nurse`;
CREATE TABLE `Nurse` (
    `nurse_id` int NOT NULL AUTO_INCREMENT,

    `first_name` varchar(30),

    `last_name` varchar(30),

    `contact_no` varchar(15) UNIQUE,

    `address` varchar(100),

    `salary` decimal(10, 2),

    PRIMARY KEY (`nurse_id`)
);


-- 6. Bill Entity


DROP TABLE IF EXISTS `Bill`;
CREATE TABLE `Bill` (
    `bill_number` int NOT NULL,

    `bill_type` int,

    `total` decimal(10, 2),

    `subtotal` decimal(10, 2),

    `taxes` decimal(10, 2),

    `paid` int,
```

```sql
    PRIMARY KEY (`bill_number`)
);
-- 8. Prescription Entity


DROP TABLE IF EXISTS `Prescription`;
CREATE TABLE `Prescription` (
    `prescription_id` int NOT NULL,
    PRIMARY KEY (`prescription_id`)
);


-- 9. Diagnosis Entity


DROP TABLE IF EXISTS `Diagnosis`;
CREATE TABLE `Diagnosis` (
    `diagnosis_id` int NOT NULL,
    `results` varchar(150),
    PRIMARY KEY (`diagnosis_id`)
);


-- 10. Treatment Entity


DROP TABLE IF EXISTS `Treatment`;
CREATE TABLE `Treatment` (
    `treatment_id` int NOT NULL,
    `start_date` Date,
    `end_date` Date,
    `details` varchar(255),
    PRIMARY KEY (`treatment_id`)
);


-- 11. Test Entity


DROP TABLE IF EXISTS `Test`;
CREATE TABLE `Test` (
    `test_id` int NOT NULL AUTO_INCREMENT,
    `name` varchar(50),
    `cost` decimal(10, 2),
    PRIMARY KEY (`test_id`)
);


-- 12. Procedure Entity
```

```sql
DROP TABLE IF EXISTS `Procedure`;
CREATE TABLE `Procedure` (
    `procedure_id` int NOT NULL AUTO_INCREMENT,
    `name` varchar(50),
    `cost` decimal(10, 2),
    PRIMARY KEY (`procedure_id`)
);


-- 13. Patient Log Entity


DROP TABLE IF EXISTS `Patient_Log`;
CREATE TABLE `Patient_Log` (
    `patient_id` int NOT NULL,
    `checkin` Datetime,
    `checkout` Datetime,
    PRIMARY KEY (`patient_id`, `checkin`),
    FOREIGN KEY (`patient_id`) REFERENCES `Patient` ( `patient_id` ) ON DELETE CASCADE
);


-- Relationship Sets


-- 1. Treated


DROP TABLE IF EXISTS `Treated`;
CREATE TABLE `Treated` (
    `patient_id` int,
    `treatment_id` int NOT NULL,
    PRIMARY KEY (`treatment_id`),
    FOREIGN KEY (`patient_id`) REFERENCES `Patient` (`patient_id`) ON DELETE CASCADE,
    FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` (`treatment_id`) ON DELETE CASCADE
);


-- 2. Performs


DROP TABLE IF EXISTS `Performs`;
CREATE TABLE `Performs` (
    `doctor_id` int NOT NULL,
    `treatment_id` int NOT NULL,
    PRIMARY KEY (`treatment_id`,`doctor_id`),
    FOREIGN KEY (`doctor_id`) REFERENCES `Doctor` ( `doctor_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` ( `treatment_id` ) ON DELETE CASCADE
```

```sql
);


-- 3. Done_In

DROP TABLE IF EXISTS `Done_In`;
CREATE TABLE `Done_In` (
    `treatment_id` int NOT NULL,
    `room_no` int NOT NULL,
    PRIMARY KEY (`treatment_id`),
    FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` ( `treatment_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`room_no`) REFERENCES `Room` ( `room_no` ) ON DELETE CASCADE
);


-- 4. Procedure_Treatment

DROP TABLE IF EXISTS `Consists`;
CREATE TABLE `Consists` (
    `procedure_id` int NOT NULL,
    `treatment_id` int NOT NULL,
    PRIMARY KEY (`procedure_id`, `treatment_id`),
    FOREIGN KEY (`procedure_id`) REFERENCES `Procedure` ( `procedure_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` ( `treatment_id` ) ON DELETE CASCADE
);


-- 5. Undergoes

DROP TABLE IF EXISTS `Undergoes`;
CREATE TABLE `Undergoes` (
    `patient_id` int,
    `diagnosis_id` int NOT NULL,
    PRIMARY KEY (`diagnosis_id`),
    FOREIGN KEY (`patient_id`) REFERENCES `Patient` (`patient_id`) ON DELETE CASCADE,
    FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` (`diagnosis_id`) ON DELETE CASCADE
);


-- 6. Recommends

DROP TABLE IF EXISTS `Recommends`;
CREATE TABLE `Recommends` (
    `doctor_id` int NOT NULL,
    `diagnosis_id` int NOT NULL,
    PRIMARY KEY (`diagnosis_id`),
```

```sql
    FOREIGN KEY (`doctor_id`) REFERENCES `Doctor` ( `doctor_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` ( `diagnosis_id` ) ON DELETE CASCADE

);


-- 7. Involves


DROP TABLE IF EXISTS `Involves`;
CREATE TABLE `Involves` (
    `diagnosis_id` int NOT NULL,
    `test_id` int NOT NULL,
    `results` varchar(100),
    PRIMARY KEY (`diagnosis_id`, `test_id`),
    FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` ( `diagnosis_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`test_id`) REFERENCES `Test` ( `test_id` ) ON DELETE CASCADE
);


-- 8. Treatment_Prescription


DROP TABLE IF EXISTS `Treatment_Presc`;
CREATE TABLE `Treatment_Presc` (
    `treatment_id` int NOT NULL,
    `prescription_id` int NOT NULL,
    PRIMARY KEY (`treatment_id`),
    FOREIGN KEY (`prescription_id`) REFERENCES `Prescription` ( `prescription_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` ( `treatment_id` ) ON DELETE CASCADE
);


-- 9. Diag_Presc


DROP TABLE IF EXISTS `Diag_Presc`;
CREATE TABLE `Diag_Presc` (
    `prescription_id` int,
    `diagnosis_id` int NOT NULL,
    PRIMARY KEY (`diagnosis_id`),
    FOREIGN KEY (`prescription_id`) REFERENCES `Prescription` (`prescription_id`) ON DELETE CASCADE,
    FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` (`diagnosis_id`) ON DELETE CASCADE
);


-- 10. Includes


DROP TABLE IF EXISTS `Includes`;
```

```sql
CREATE TABLE `Includes` (
    `medicine_id` int NOT NULL,
    `prescription_id` int NOT NULL,
    `unit` int DEFAULT 1,
    PRIMARY KEY (`medicine_id`,`prescription_id`),
    FOREIGN KEY (`medicine_id`) REFERENCES `Medicine` ( `medicine_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`prescription_id`) REFERENCES `Prescription` ( `prescription_id` ) ON DELETE CASCADE

);


-- 11. Medical_Bill


DROP TABLE IF EXISTS `Medical_Bill`;
CREATE TABLE `Medical_Bill` (
    `prescription_id` int NOT NULL,
    `bill_number` int NOT NULL,
    PRIMARY KEY (`prescription_id`),
    FOREIGN KEY (`prescription_id`) REFERENCES `Prescription` ( `prescription_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`bill_number`) REFERENCES `Bill` ( `bill_number` ) ON DELETE CASCADE
);


-- 12. Doctor_Patient


DROP TABLE IF EXISTS `Appointment`;
CREATE TABLE `Appointment` (
    `doctor_id` int NOT NULL,
    `patient_id` int NOT NULL,
    `date` Date NOT NULL,
    `start_time` Time NOT NULL,
    PRIMARY KEY (`doctor_id`,`patient_id`, `date`, `start_time`),
    FOREIGN KEY (`doctor_id`) REFERENCES `Doctor` ( `doctor_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`patient_id`) REFERENCES `Patient` ( `patient_id` ) ON DELETE CASCADE
);


-- 13. Treatment Bill


DROP TABLE IF EXISTS `Treatment_Bill`;
CREATE TABLE `Treatment_Bill` (
    `bill_number` int,
    `treatment_id` int NOT NULL,
    PRIMARY KEY (`treatment_id`),
    FOREIGN KEY (`bill_number`) REFERENCES `Bill` (`bill_number`) ON DELETE CASCADE,
```

```sql
    FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` (`treatment_id`) ON DELETE CASCADE
);


-- 14. Pays


DROP TABLE IF EXISTS `Pays`;
CREATE TABLE `Pays` (
    `patient_id` int NOT NULL,
    `bill_number` int NOT NULL,
    PRIMARY KEY (`bill_number`),
    FOREIGN KEY (`bill_number`) REFERENCES `Bill` ( `bill_number` ) ON DELETE CASCADE

);


-- 15. Nurse_Treatment


DROP TABLE IF EXISTS `Serves`;
CREATE TABLE `Serves` (
    `treatment_id` int NOT NULL,
    `nurse_id` int NOT NULL,
    PRIMARY KEY (`treatment_id`,`nurse_id`),
    FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` ( `treatment_id` ) ON DELETE CASCADE,
    FOREIGN KEY (`nurse_id`) REFERENCES `Nurse` ( `nurse_id` ) ON DELETE CASCADE
);
-- 17. Bill_Diag


DROP TABLE IF EXISTS `Bill_Diag`;
CREATE TABLE `Bill_Diag` (
    `bill_number` int,
    `diagnosis_id` int NOT NULL,
    PRIMARY KEY (`diagnosis_id`),
    FOREIGN KEY (`bill_number`) REFERENCES `Bill` (`bill_number`) ON DELETE CASCADE,
    FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` (`diagnosis_id`) ON DELETE CASCADE
);


-- 18. Implies


DROP TABLE IF EXISTS `Implies`;
CREATE TABLE `Implies` (
    `diagnosis_id` int NOT NULL,
    `treatment_id` int NOT NULL,
    PRIMARY KEY (`diagnosis_id`),
```

```
FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` ( `diagnosis_id` ) ON DELETE CASCADE,
FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` ( `treatment_id` ) ON DELETE CASCADE
);
```

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0017 seconds.)

```
CREATE TABLE `Patient` ( `patient_id` int NOT NULL, `first_name` varchar(30), `last_name` varchar(30), `contact_no` varchar(15) UNIQUE, `address` varchar(100), PRIMARY KEY
(`patient_id`) );
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
CREATE TABLE `Room` ( `room_no` int NOT NULL, `name` varchar(30), `availability` bit, PRIMARY KEY (`room_no`) );
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
CREATE TABLE `Doctor` ( `doctor_id` int NOT NULL AUTO_INCREMENT, `first_name` varchar(30), `last_name` varchar(30), `contact_no` varchar(15) UNIQUE, `address` varchar(50),
`salary` decimal(10, 2), `specialization` varchar(30), `role` varchar(30), `room_no`int NOT NULL, PRIMARY KEY (`doctor_id`), FOREIGN KEY (`room_no`) REFERENCES `Room` (
`room_no` ) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
CREATE TABLE `Medicine` ( `medicine_id` int NOT NULL AUTO_INCREMENT, `inventory_quantity` int, `name` varchar(30), `price` decimal(10, 2), PRIMARY KEY (`medicine_id`) );
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
CREATE TABLE `Nurse` ( `nurse_id` int NOT NULL AUTO_INCREMENT, `first_name` varchar(30), `last_name` varchar(30), `contact_no` varchar(15) UNIQUE, `address` varchar(100),
`salary` decimal(10, 2), PRIMARY KEY (`nurse_id`) );
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
CREATE TABLE `Bill` ( `bill_number` int NOT NULL, `bill_type` int, `total` decimal(10, 2), `subtotal` decimal(10, 2), `taxes` decimal(10, 2), `paid` int, PRIMARY KEY
(`bill_number`) );
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
CREATE TABLE `Employee` ( `employee_id` int NOT NULL AUTO_INCREMENT, `first_name` varchar(30), `last_name` varchar(30), `contact_no` varchar(15) UNIQUE, `occupation`
varchar(30), `address` varchar(50), `salary` decimal(10, 2), PRIMARY KEY (`employee_id`) );
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
CREATE TABLE `Prescription` ( `prescription_id` int NOT NULL, PRIMARY KEY (`prescription_id`) );
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```
CREATE TABLE `Diagnosis` ( `diagnosis_id` int NOT NULL, `results` varchar(150), PRIMARY KEY (`diagnosis_id`) );
```

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

CREATE TABLE `Treatment` (`treatment_id` int NOT NULL, `start_date` Date, `end_date` Date, `details` varchar(255), PRIMARY KEY (`treatment_id`));

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

CREATE TABLE `Test` (`test_id` int NOT NULL AUTO_INCREMENT, `name` varchar(50), `cost` decimal(10, 2), PRIMARY KEY (`test_id`));

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

CREATE TABLE `Procedure` (`procedure_id` int NOT NULL AUTO_INCREMENT, `name` varchar(50), `cost` decimal(10, 2), PRIMARY KEY (`procedure_id`));

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

CREATE TABLE `Patient_Log` (`patient_id` int NOT NULL, `checkin` Datetime, `checkout` Datetime, PRIMARY KEY (`patient_id`, `checkin`), FOREIGN KEY (`patient_id`) REFERENCES `Patient` (`patient_id`) ON DELETE CASCADE);

[Edit inline] [Edit] [Create PHP code]

	Table ▲	Action							Rows	Type	Collation	Size	Overhead
☐	**bill**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	**diagnosis**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	**doctor**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	**employee**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	**medicine**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	**nurse**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	**patient**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	**patient_log**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	**prescription**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	**procedure**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	**room**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	**test**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	**treatment**	⭐	🔲 Browse	🔧 Structure	🔍 Search	➕ Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	13 tables	**Sum**							0	InnoDB	utf8mb4_general_ci	288.0 KiB	0 B

Creating Relationship sets:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

CREATE TABLE `Treated` (`patient_id` int, `treatment_id` int NOT NULL, PRIMARY KEY (`treatment_id`), FOREIGN KEY (`patient_id`) REFERENCES `Patient` (`patient_id`) ON DELETE CASCADE, FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` (`treatment_id`) ON DELETE CASCADE);

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

-- 2. Performs CREATE TABLE `Performs` (`doctor_id` int NOT NULL, `treatment_id` int NOT NULL, PRIMARY KEY (`treatment_id`,`doctor_id`), FOREIGN KEY (`doctor_id`) REFERENCES `Doctor` (`doctor_id`) ON DELETE CASCADE, FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` (`treatment_id`) ON DELETE CASCADE);

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

-- 3. Done_In CREATE TABLE `Done_In` (`treatment_id` int NOT NULL, `room_no` int NOT NULL, PRIMARY KEY (`treatment_id`), FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` (`treatment_id`) ON DELETE CASCADE, FOREIGN KEY (`room_no`) REFERENCES `Room` (`room_no`) ON DELETE CASCADE);

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

-- 4. Procedure_Treatment CREATE TABLE `Consists` (`procedure_id` int NOT NULL, `treatment_id` int NOT NULL, PRIMARY KEY (`procedure_id`, `treatment_id`), FOREIGN KEY (`procedure_id`) REFERENCES `Procedure` (`procedure_id`) ON DELETE CASCADE, FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` (`treatment_id`) ON DELETE CASCADE);

[Edit inline] [Edit] [Create PHP code]

```
-- 5. Undergoes CREATE TABLE `Undergoes` ( `patient_id` int, `diagnosis_id` int NOT NULL, PRIMARY KEY (`diagnosis_id`), FOREIGN KEY (`patient_id`) REFERENCES `Patient`
(`patient_id`) ON DELETE CASCADE, FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` (`diagnosis_id`) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 6. Recommends CREATE TABLE `Recommends` ( `doctor_id` int NOT NULL, `diagnosis_id` int NOT NULL, PRIMARY KEY (`diagnosis_id`), FOREIGN KEY (`doctor_id`) REFERENCES
`Doctor` ( `doctor_id` ) ON DELETE CASCADE, FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` ( `diagnosis_id` ) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 7. Involves CREATE TABLE `Involves` ( `diagnosis_id` int NOT NULL, `test_id` int NOT NULL, `results` varchar(100), PRIMARY KEY (`diagnosis_id`, `test_id`), FOREIGN KEY
(`diagnosis_id`) REFERENCES `Diagnosis` ( `diagnosis_id` ) ON DELETE CASCADE, FOREIGN KEY (`test_id`) REFERENCES `Test` ( `test_id` ) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 8. Treatment_Prescription CREATE TABLE `Treatment_Presc` ( `treatment_id` int NOT NULL, `prescription_id` int NOT NULL, PRIMARY KEY (`treatment_id`), FOREIGN KEY
(`prescription_id`) REFERENCES `Prescription` ( `prescription_id` ) ON DELETE CASCADE, FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` ( `treatment_id` ) ON DELETE
CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 9. Diag_Presc CREATE TABLE `Diag_Presc` ( `prescription_id` int, `diagnosis_id` int NOT NULL, PRIMARY KEY (`diagnosis_id`), FOREIGN KEY (`prescription_id`) REFERENCES
`Prescription` (`prescription_id`) ON DELETE CASCADE, FOREIGN KEY (`diagnosis_id`) REFERENCES `Diagnosis` (`diagnosis_id`) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 10. Includes CREATE TABLE `Includes` ( `medicine_id` int NOT NULL, `prescription_id` int NOT NULL, `unit` int DEFAULT 1, PRIMARY KEY (`medicine_id`,`prescription_id`),
FOREIGN KEY (`medicine_id`) REFERENCES `Medicine` ( `medicine_id` ) ON DELETE CASCADE, FOREIGN KEY (`prescription_id`) REFERENCES `Prescription` ( `prescription_id` ) ON
DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 11. Medical_Bill CREATE TABLE `Medical_Bill` ( `prescription_id` int NOT NULL, `bill_number` int NOT NULL, PRIMARY KEY (`prescription_id`), FOREIGN KEY
(`prescription_id`) REFERENCES `Prescription` ( `prescription_id` ) ON DELETE CASCADE, FOREIGN KEY (`bill_number`) REFERENCES `Bill` ( `bill_number` ) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 12. Doctor_Patient CREATE TABLE `Appointment` ( `doctor_id` int NOT NULL, `patient_id` int NOT NULL, `date` Date NOT NULL, `start_time` Time NOT NULL, PRIMARY KEY
(`doctor_id`,`patient_id`, `date`, `start_time`), FOREIGN KEY (`doctor_id`) REFERENCES `Doctor` ( `doctor_id` ) ON DELETE CASCADE, FOREIGN KEY (`patient_id`) REFERENCES
`Patient` ( `patient_id` ) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 13. Treatment Bill CREATE TABLE `Treatment_Bill` ( `bill_number` int, `treatment_id` int NOT NULL, PRIMARY KEY (`treatment_id`), FOREIGN KEY (`bill_number`) REFERENCES
`Bill` (`bill_number`) ON DELETE CASCADE, FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` (`treatment_id`) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 14. Pays CREATE TABLE `Pays` ( `patient_id` int NOT NULL, `bill_number` int NOT NULL, PRIMARY KEY (`bill_number`), FOREIGN KEY (`bill_number`) REFERENCES `Bill` (
`bill_number` ) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 15. Nurse_Treatment CREATE TABLE `Serves` ( `treatment_id` int NOT NULL, `nurse_id` int NOT NULL, PRIMARY KEY (`treatment_id`,`nurse_id`), FOREIGN KEY (`treatment_id`)
REFERENCES `Treatment` ( `treatment_id` ) ON DELETE CASCADE, FOREIGN KEY (`nurse_id`) REFERENCES `Nurse` ( `nurse_id` ) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

```
-- 16. Treatment_Room CREATE TABLE `Aftercare` ( `treatment_id` int NOT NULL, `room_no` int NOT NULL, `start_date` Date NOT NULL, `end_date` Date, PRIMARY KEY
(`treatment_id`,`room_no`, `start_date`), FOREIGN KEY (`treatment_id`) REFERENCES `Treatment` ( `treatment_id` ) ON DELETE CASCADE, FOREIGN KEY (`room_no`) REFERENCES
`Room` ( `room_no` ) ON DELETE CASCADE );
```

[Edit inline] [Edit] [Create PHP code]

	Table ▲	Action						Rows	Type	Collation	Size	Overhead
☐	aftercare	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	appointment	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	bill	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	bill_diag	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	consists	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	diagnosis	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	diag_presc	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	doctor	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	done_in	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	employee	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	implies	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	includes	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	involves	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	medical_bill	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	medicine	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	nurse	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	patient	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	patient_log	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
		⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	

■ Console

		Action						Rows	Type	Collation	Size	Overhead
☐	pays	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	performs	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	prescription	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	procedure	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	recommends	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	room	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	serves	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	test	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	treated	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	treatment	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	treatment_bill	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	treatment_presc	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	undergoes	⭐	🔲 Browse	🔧 Structure	🔍 Search	📑 Insert	🗑 Empty	⊘ Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
	31 tables	Sum							0	InnoDB	utf8mb4_general_ci	848.0 KiB	0 B

↑ ☐ Check all   With selected: ⌄

Populating The Database:

```sql
INSERT INTO `Test` (`name`, `cost`) VALUES
("Ultrasound Imaging", 500),
("MRI", 3000),
("CT Scan", 4000),
("X-Ray", 1000),
("Blood Sugar", 5000);

INSERT INTO `Room` (`room_no`, `name`, `availability`) VALUES
(1, "ER Room", 1),
(2, "Doctor Room", 0),
(3, "Aftercare Room", 1),
(4, "Aftercare Room", 0),
(5, "Doctor Room", 0);

INSERT INTO `Employee` (`first_name`, `last_name`, `contact_no`, `occupation`, `address`, `salary`)
VALUES
("arjun", "sheikh", "9010202211", "janitor", "Raj Bhavan Road", 10000),
("Rohit", "Sharma", "9876787687", "Driver", "Andheri East, Mumbai", 20000),
("naman", "ojha", "9010233211", "ward boy", "Gachibowli, Hyderabad", 7000),
("azad", "kade", "9911234544", "cleaning staff", "JUbilee Hills, Hyderabad",10000.00),
("Pitamaha", "Khandke", "02227221431", "mechanic", "Himayathnagar, Hyderabad", 99948.03);

INSERT INTO `Doctor` (`first_name`, `last_name`, `contact_no`, `address`, `salary`, `specialization`,
`role`, `room_no`) VALUES
("Devika", "Patel", "9733277342", "E2, Near Lsr Collage, East Of Kailash, Delhi", 80000.00, "Neuro
surgeon", "surgeon", 2),
("Ameretat",  "Sachdev", "02226609802", "Rotary Service Centre, Mumbai", 150000.00, "Orthopedics",
"General Physicain", 5),
("Revati", "Mehta", "9733277444", "274 /b, Girdhari Sadan, N.c.kelkar Road", 100000, "Orthopaedics",
"Orthopedic surgeons", 2),
("Saka", "Ghate", "07925501915", "29 /b, Asopalav, Khanpur", 20000, "Cardiology", "Pathologist", 5),
("Matsya",  "Sridhar", "2660030345", "Bundi Mottu Avenue Road Cross, Avenue Road", 100000.00,
"Pediatrics", "Pediatrician", 5);

INSERT INTO `Procedure` (`name`, `cost`) VALUES
("Coronary Angiogram", 63500.00),
("Angioplasty", 412750.00),
("Abdominal aortic aneurysm", 508000.00),
```

```sql
("Knee Replacement", 539750.00),
("ACL Reconstruction", 285750.00);


INSERT INTO `Nurse` (`first_name`, `last_name`, `contact_no`, `address`, `salary`) VALUES
("Maheshvari", "Gothe", "9292451252", "241 -, Natraj Market, Sv Road, Malad, Mumbai", 40000.00),
("Atman",   "Jain", "02225123138", "Chittranjan Nagar, Rajawadi, Ghatkoper, Mumbai", 25000.00),
("Archana", "Hans", "02227895689", "22  Rachna, Sector , Vashi, Mumbai", 30000.00),
("Bharat", "Badal", "02228882593", "109 , Parasrampuria Chambers, Opp Rly Stn, Malad (west)",
35000.00),
("Manda", "Saran", "02223862592", "448  A, Girgaum, Mumbai", 25000.00);


INSERT INTO `Medicine` (`inventory_quantity`, `name`, `price`) VALUES
(145, "Ifosfamide", 551.65),
(78, "Cisplatin", 360.00),
(233, "Oxaliplatin", 4798.00),
(124, "Methotrexate", 57.85),
(112, "Busulfan", 415.00);


INSERT INTO `Patient` VALUES
(1, "Jyotish","Divan", "02223443172", "417 ,sai Chambers,  Narshi Natha St, Chinch Bunder"),
(2, "Yasmine","Kumar","02224464461", "349 ,Allied Indl Estae Off Mmc, Off M.m.c Road, Mahim"),
(3, "Supriya", "Shanker", "02224092768", "41 /, Sadar Nagar No , Sion");


INSERT INTO `Patient_Log` (`patient_id`, `checkin`, `checkout`) VALUES
(1, "2012-12-02 20:20:10", NULL),
(2, "2020-08-05 14:56:24", "2020-08-05 18:34:32"),
(3, "2015-12-02 08:20:10", "2015-12-02 11:20:10"),
(3, "2015-12-05 10:20:10", "2015-12-06 11:20:10");


INSERT INTO `Appointment` (`doctor_id`, `patient_id`, `date`, `start_time`) VALUES
(3, 3, "2015-12-02", "08:35:00"),
(2, 2, "2020-08-05", "13:05:13");


INSERT INTO `Diagnosis` (`diagnosis_id`, `results`) VALUES
(3, "Type 2 diabetes"),
(2, "Scans Normal. No treatment required. Patient advised to take prescribed medicines");


INSERT INTO `Treatment` (`treatment_id`, `start_date`, `end_date`, `details`) VALUES
(1, "2012-12-02", "2012-12-03", "Success with slight knee complication"),
(2, "2015-12-05", "2015-12-06", "Sugar levels reduced");


INSERT INTO `Bill` VALUES
```

```sql
(1, 2,762000.00,700000,62000, 0),
(2, 1, 1000.5, 969.5,31, 1),
(3, 0, 6300, 6000, 300, 1),
(5, 0, 8000.00, 7500.00, 500.00, 1),
(6, 1, 952.95, 832.85, 120.10, 1),
(4, 2, 855000, 850500, 4500, 1);

INSERT INTO `Prescription` VALUES
(1),
(5);

INSERT INTO `Implies` (`diagnosis_id`, `treatment_id`) VALUES
(3, 2);

INSERT INTO `Recommends` (`doctor_id`, `diagnosis_id`) VALUES
(3, 3),
(2, 2);

INSERT INTO `Done_In` VALUES
(1, 1),
(2, 1);

INSERT INTO `Treated` VALUES
(1, 1),
(3, 2);

INSERT INTO `Performs` VALUES
(1, 1),
(4, 2),
(5, 2);

INSERT INTO `Consists` VALUES
(1, 1),
(2, 1),
(4, 1),
(4, 2),
(1, 2);

INSERT INTO `Treatment_Bill` VALUES
(1, 1),
(4, 2);
```

```sql
INSERT INTO `Pays` VALUES
(1, 1),
(3, 3),
(2, 5),
(2, 6),
(1, 2),
(3, 4);


INSERT INTO `Serves` VALUES
(1, 1);

INSERT INTO `Treatment_Presc` VALUES
(1, 1);

INSERT INTO `Includes` (`medicine_id`, `prescription_id`) VALUES
(1, 1),
(2, 1),
(4, 1),
(2, 5),
(4, 5),
(5, 5);

INSERT INTO `Medical_Bill` VALUES
(1, 2),
(5, 6);

INSERT INTO `Involves` VALUES
(2, 2, ""),
(2, 3, ""),
(3, 5, "Positive");

INSERT INTO `Undergoes` VALUES
(3, 3),
(2, 2);

INSERT INTO `Bill_Diag` VALUES
(3, 3),
(5, 2);

INSERT INTO `Diag_Presc` VALUES
(5, 2);
```

✅ 5 rows inserted.
Inserted row id: 5 (Query took 0.0036 seconds.)

```sql
INSERT INTO `Test` (`name`, `cost`) VALUES ("Ultrasound Imaging", 500), ("MRI", 3000), ("CT Scan", 4000), ("X-Ray", 1000), ("Blood Sugar", 5000);
```

[Edit inline] [Edit] [Create PHP code]

✅ 5 rows inserted.
Inserted row id: 5 (Query took 0.0017 seconds.)

```sql
INSERT INTO `Room` (`room_no`, `name`, `availability`) VALUES (1, "ER Room", 1), (2, "Doctor Room", 0), (3, "Aftercare Room", 1), (4, "Aftercare Room", 0), (5, "Doctor Room", 0);
```

[Edit inline] [Edit] [Create PHP code]

✅ 5 rows inserted.
Inserted row id: 5 (Query took 0.0004 seconds.)

```sql
INSERT INTO `Employee` (`first_name`, `last_name`, `contact_no`, `occupation`, `address`, `salary`) VALUES ("arjun", "sheikh", "9010202211", "janitor", "Raj Bhavan Road", 10000), ("Rohit", "Sharma", "9876787687", "Driver", "Andheri East, Mumbai", 20000), ("naman", "ojha", "9010233211", "ward boy", "Gachibowli, Hyderabad", 7000), ("azad", "kade", "9911234544", "cleaning staff", "JUbilee Hills, Hyderabad",10000.00), ("Pitamaha", "Khandke", "02227221431", "mechanic", "Himayathnagar, Hyderabad", 99948.03);
```

[Edit inline] [Edit] [Create PHP code]

✅ 5 rows inserted.
Inserted row id: 5 (Query took 0.0011 seconds.)

```sql
INSERT INTO `Doctor` (`first_name`, `last_name`, `contact_no`, `address`, `salary`, `specialization`, `role`, `room_no`) VALUES ("Devika", "Patel", "9733277342", "E2, Near Lsr Collage, East Of Kailash, Delhi", 80000.00, "Neuro surgeon", "surgeon", 2), ("Ameretat", "Sachdev", "02226609802", "Rotary Service Centre, Mumbai", 150000.00, "Orthopedics", "General Physicain", 5), ("Revati", "Mehta", "9733277444", "274 /b, Girdhari Sadan, N.c.kelkar Road", 100000, "Orthopaedics", "Orthopedic surgeons", 2), ("Saka", "Ghate", "07925501915", "29 /b, Asopalav, Khanpur", 20000, "Cardiology", "Pathologist", 5), ("Matsya", "Sridhar", "2660030345", "Bundi Mottu Avenue Road Cross, Avenue Road", 100000.00, "Pediatrics", "Pediatrician", 5);
```

[Edit inline] [Edit] [Create PHP code]

✅ 5 rows inserted.
Inserted row id: 5 (Query took 0.0005 seconds.)

```sql
INSERT INTO `Procedure` (`name`, `cost`) VALUES ("Coronary Angiogram", 63500.00), ("Angioplasty", 412750.00), ("Abdominal aortic aneurysm", 508000.00), ("Knee Replacement", 539750.00), ("ACL Reconstruction", 285750.00);
```

[Edit inline] [Edit] [Create PHP code]

✅ 5 rows inserted.
Inserted row id: 5 (Query took 0.0016 seconds.)

```sql
INSERT INTO `Nurse` (`first_name`, `last_name`, `contact_no`, `address`, `salary`) VALUES ("Maheshvari", "Gothe", "9292451252", "241 -, Natraj Market, Sv Road, Malad, Mumbai", 40000.00), ("Atman", "Jain", "02225123138", "Chittranjan Nagar, Rajawadi, Ghatkoper, Mumbai", 25000.00), ("Archana", "Hans", "02227895689", "22 Rachna, Sector , Vashi, Mumbai", 30000.00), ("Bharat", "Badal", "02228882593", "109 , Parasrampuria Chambers, Opp Rly Stn, Malad (west)", 35000.00), ("Manda", "Saran", "02223862592", "448 A, Girgaum, Mumbai", 25000.00);
```

[Edit inline] [Edit] [Create PHP code]

✅ 5 rows inserted.
Inserted row id: 5 (Query took 0.0019 seconds.)

```sql
INSERT INTO `Medicine` (`inventory_quantity`, `name`, `price`) VALUES (145, "Ifosfamide", 551.65), (78, "Cisplatin", 360.00), (233, "Oxaliplatin", 4798.00), (124, "Methotrexate", 57.85), (112, "Busulfan", 415.00);
```

[Edit inline] [Edit] [Create PHP code]

✅ 3 rows inserted.
Inserted row id: 3 (Query took 0.0004 seconds.)

```sql
INSERT INTO `Patient` VALUES (1, "Jyotish","Divan", "02223443172", "417 ,sai Chambers, Narshi Natha St, Chinch Bunder"), (2, "Yasmine","Kumar","02224464461", "349 ,Allied Indl Estae Off Mmc, Off M.m.c Road, Mahim"), (3, "Supriya", "Shanker", "02224092768", "41 /, Sadar Nagar No , Sion");
```

✔ 4 rows inserted.
Inserted row id: 4 (Query took 0.0012 seconds.)

```
INSERT INTO `Patient_Log` (`patient_id`, `checkin`, `checkout`) VALUES (1, "2012-12-02 20:20:10", NULL), (2, "2020-08-05 14:56:24", "2020-08-05 18:34:32"), (3, "2015-12-02 08:20:10", "2015-12-02 11:20:10"), (3, "2015-12-05 10:20:10", "2015-12-06 11:20:10");
```

[Edit inline] [Edit] [Create PHP code]

✔ 2 rows inserted.
Inserted row id: 2 (Query took 0.0015 seconds.)

```
INSERT INTO `Appointment` (`doctor_id`, `patient_id`, `date`, `start_time`) VALUES (3, 3, "2015-12-02", "08:35:00"), (2, 2, "2020-08-05", "13:05:13");
```

[Edit inline] [Edit] [Create PHP code]

✔ 2 rows inserted.
Inserted row id: 2 (Query took 0.0013 seconds.)

```
INSERT INTO `Diagnosis` (`diagnosis_id`, `results`) VALUES (3, "Type 2 diabetes"), (2, "Scans Normal. No treatment required. Patient advised to take prescribed medicines");
```

[Edit inline] [Edit] [Create PHP code]

✔ 2 rows inserted.
Inserted row id: 2 (Query took 0.0003 seconds.)

```
INSERT INTO `Treatment` (`treatment_id`, `start_date`, `end_date`, `details`) VALUES (1, "2012-12-02", "2012-12-03", "Success with slight knee complication"), (2, "2015-12-05", "2015-12-06", "Sugar levels reduced");
```

[Edit inline] [Edit] [Create PHP code]

✔ 6 rows inserted.
Inserted row id: 6 (Query took 0.0015 seconds.)

```
INSERT INTO `Bill` VALUES (1, 2,762000.00,700000,62000, 0), (2, 1, 1000.5, 969.5,31, 1), (3, 0, 6300, 6000, 300, 1), (5, 0, 8000.00, 7500.00, 500.00, 1), (6, 1, 952.95, 832.85, 120.10, 1), (4, 2, 855000, 850500, 4500, 1);
```

[Edit inline] [Edit] [Create PHP code]

✔ 1 row inserted. (Query took 0.0005 seconds.)

```
INSERT INTO `Serves` VALUES (1, 1);
```

[Edit inline] [Edit] [Create PHP code]

✔ 1 row inserted. (Query took 0.0005 seconds.)

```
INSERT INTO `Treatment_Presc` VALUES (1, 1);
```

[Edit inline] [Edit] [Create PHP code]

✔ 6 rows inserted. (Query took 0.0030 seconds.)

```
INSERT INTO `Includes` (`medicine_id`, `prescription_id`) VALUES (1, 1), (2, 1), (4, 1), (2, 5), (4, 5), (5, 5);
```

[Edit inline] [Edit] [Create PHP code]

✔ 2 rows inserted. (Query took 0.0031 seconds.)

```
INSERT INTO `Medical_Bill` VALUES (1, 2), (5, 6);
```

[Edit inline] [Edit] [Create PHP code]

✔ 3 rows inserted. (Query took 0.0028 seconds.)

```
INSERT INTO `Involves` VALUES (2, 2, ""), (2, 3, ""), (3, 5, "Positive");
```

[Edit inline] [Edit] [Create PHP code]

✔ 2 rows inserted. (Query took 0.0035 seconds.)

```sql
INSERT INTO `Undergoes` VALUES (3, 3), (2, 2);
```

[Edit inline] [Edit] [Create PHP code]

✔ 2 rows inserted. (Query took 0.0023 seconds.)

```sql
INSERT INTO `Bill_Diag` VALUES (3, 3), (5, 2);
```

[Edit inline] [Edit] [Create PHP code]

✔ 1 row inserted. (Query took 0.0003 seconds.)

```sql
INSERT INTO `Diag_Presc` VALUES (5, 2);
```

[Edit inline] [Edit] [Create PHP code]

Tables after Populating

Test:

←T→			test_id	name	cost
☐	🖉 Edit	🏛 Copy ⊖ Delete	1	Ultrasound Imaging	500.00
☐	🖉 Edit	🏛 Copy ⊖ Delete	2	MRI	3000.00
☐	🖉 Edit	🏛 Copy ⊖ Delete	3	CT Scan	4000.00
☐	🖉 Edit	🏛 Copy ⊖ Delete	4	X-Ray	1000.00
☐	🖉 Edit	🏛 Copy ⊖ Delete	5	Blood Sugar	5000.00

Room

←T→				room_no	name	availability
☐	🖉 Edit	📑 Copy	⊖ Delete	1	ER Room	1
☐	🖉 Edit	📑 Copy	⊖ Delete	2	Doctor Room	0
☐	🖉 Edit	📑 Copy	⊖ Delete	3	Aftercare Room	1
☐	🖉 Edit	📑 Copy	⊖ Delete	4	Aftercare Room	0
☐	🖉 Edit	📑 Copy	⊖ Delete	5	Doctor Room	0

Employee

←	→				employee_id	first_name	last_name	contact_no	occupation	address	salary
☐	🖉 Edit	📑 Copy	⊖ Delete		1	arjun	sheikh	9010202211	janitor	Raj Bhavan Road	10000.00
☐	🖉 Edit	📑 Copy	⊖ Delete		2	Rohit	Sharma	9876787687	Driver	Andheri East, Mumbai	20000.00
☐	🖉 Edit	📑 Copy	⊖ Delete		3	naman	ojha	9010233211	ward boy	Gachibowli, Hyderabad	7000.00
☐	🖉 Edit	📑 Copy	⊖ Delete		4	azad	kade	9911234544	cleaning staff	JUbilee Hills, Hyderabad	10000.00
☐	🖉 Edit	📑 Copy	⊖ Delete		5	Pitamaha	Khandke	02227221431	mechanic	Himayathnagar, Hyderabad	99948.03

Doctor

←T→				doctor_id	first_name	last_name	contact_no	address	salary	specialization	role	room_no
☐	🖉 Edit	📑 Copy	⊖ Delete	1	Devika	Patel	9733277342	E2, Near Lsr Collage, East Of Kailash, Delhi	80000.00	Neuro surgeon	surgeon	2
☐	🖉 Edit	📑 Copy	⊖ Delete	2	Ameretat	Sachdev	02226609802	Rotary Service Centre, Mumbai	150000.00	Orthopedics	General Physicain	5
☐	🖉 Edit	📑 Copy	⊖ Delete	3	Revati	Mehta	9733277444	274 /b, Girdhari Sadan, N.c.kelkar Road	100000.00	Orthopaedics	Orthopedic surgeons	2
☐	🖉 Edit	📑 Copy	⊖ Delete	4	Saka	Ghate	07925501915	29 /b, Asopalav, Khanpur	20000.00	Cardiology	Pathologist	5
☐	🖉 Edit	📑 Copy	⊖ Delete	5	Matsya	Sridhar	2660030345	Bundi Mottu Avenue Road Cross, Avenue Road	100000.00	Pediatrics	Pediatrician	5

Procedure

←T→				procedure_id	name	cost
☐	🖉 Edit	📑 Copy	⊖ Delete	1	Coronary Angiogram	63500.00
☐	🖉 Edit	📑 Copy	⊖ Delete	2	Angioplasty	412750.00
☐	🖉 Edit	📑 Copy	⊖ Delete	3	Abdominal aortic aneurysm	508000.00
☐	🖉 Edit	📑 Copy	⊖ Delete	4	Knee Replacement	539750.00
☐	🖉 Edit	📑 Copy	⊖ Delete	5	ACL Reconstruction	285750.00

Nurse

			nurse_id	first_name	last_name	contact_no	address	salary
☐	Edit	Copy	Delete	1	Maheshvari	Gothe	9292451252	241 -, Natraj Market, Sv Road, Malad, Mumbai	40000.00
☐	Edit	Copy	Delete	2	Atman	Jain	02225123138	Chittranjan Nagar, Rajawadi, Ghatkoper, Mumbai	25000.00
☐	Edit	Copy	Delete	3	Archana	Hans	02227895689	22 Rachna, Sector , Vashi, Mumbai	30000.00
☐	Edit	Copy	Delete	4	Bharat	Badal	02228882593	109 , Parasrampuria Chambers, Opp Rly Stn, Malad (...	35000.00
☐	Edit	Copy	Delete	5	Manda	Saran	02223862592	448 A, Girgaum, Mumbai	25000.00

Medicine

			medicine_id	inventory_quantity	name	price
☐	Edit	Copy	Delete	1	145	Ifosfamide	551.65
☐	Edit	Copy	Delete	2	78	Cisplatin	360.00
☐	Edit	Copy	Delete	3	233	Oxaliplatin	4798.00
☐	Edit	Copy	Delete	4	124	Methotrexate	57.85
☐	Edit	Copy	Delete	5	112	Busulfan	415.00

Patient

			patient_id	first_name	last_name	contact_no	address
☐	Edit	Copy	Delete	1	Jyotish	Divan	02223443172	417 ,sai Chambers, Narshi Natha St, Chinch Bunder
☐	Edit	Copy	Delete	2	Yasmine	Kumar	02224464461	349 ,Allied Indl Estae Off Mmc, Off M.m.c Road, Ma...
☐	Edit	Copy	Delete	3	Supriya	Shanker	02224092768	41 /, Sadar Nagar No , Sion

Patient_log

			patient_id	checkin	checkout
☐	Edit	Copy	Delete	1	2012-12-02 20:20:10	NULL
☐	Edit	Copy	Delete	2	2020-08-05 14:56:24	2020-08-05 18:34:32
☐	Edit	Copy	Delete	3	2015-12-02 08:20:10	2015-12-02 11:20:10
☐	Edit	Copy	Delete	3	2015-12-05 10:20:10	2015-12-06 11:20:10

Appointment:

			doctor_id	patient_id	date	start_time
☐	Edit	Copy	Delete	2	2	2020-08-05	13:05:13
☐	Edit	Copy	Delete	3	3	2015-12-02	08:35:00

Diagnosis

					diagnosis_id	results
☐	🖉 Edit	⋽ Copy	⊖ Delete		2	Scans Normal. No treatment required. Patient advis...
☐	🖉 Edit	⋽ Copy	⊖ Delete		3	Type 2 diabetes

Treatment

					treatment_id	start_date	end_date	details
☐	🖉 Edit	⋽ Copy	⊖ Delete		1	2012-12-02	2012-12-03	Success with slight knee complication
☐	🖉 Edit	⋽ Copy	⊖ Delete		2	2015-12-05	2015-12-06	Sugar levels reduced

Bill

					bill_number	bill_type	total	subtotal	taxes	paid
☐	🖉 Edit	⋽ Copy	⊖ Delete		1	2	762000.00	700000.00	62000.00	0
☐	🖉 Edit	⋽ Copy	⊖ Delete		2	1	1000.50	969.50	31.00	1
☐	🖉 Edit	⋽ Copy	⊖ Delete		3	0	6300.00	6000.00	300.00	1
☐	🖉 Edit	⋽ Copy	⊖ Delete		4	2	855000.00	850500.00	4500.00	1
☐	🖉 Edit	⋽ Copy	⊖ Delete		5	0	8000.00	7500.00	500.00	1
☐	🖉 Edit	⋽ Copy	⊖ Delete		6	1	952.95	832.85	120.10	1

JOIN QUERIES

1.Show details of all the doctors(who have/don't) have appointment along with their respective appointment details.

```
Server: 127.0.0.1 »  Database: hms

 Structure    SQL    Search    Query    Export    Import    Operations    Privileges

Run SQL query/queries on database hms:

1 SELECT  Doctor.doctor_id, first_name, last_name, role, specialization, date, start_time, room_no
2 FROM Appointment RIGHT OUTER JOIN
3 Doctor ON Doctor.doctor_id=Appointment.doctor_id;
```
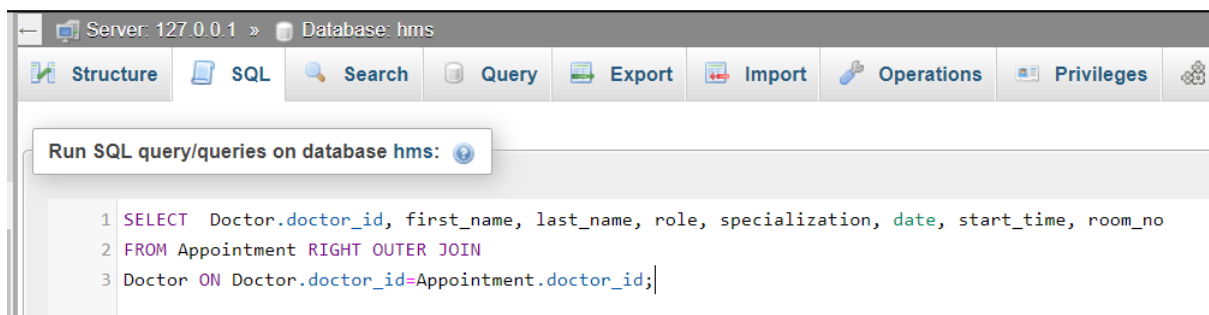
doctor_id	first_name	last_name	role	specialization	date	start_time	room_no
1	Devika	Patel	surgeon	Neuro surgeon	*NULL*	*NULL*	2
2	Ameretat	Sachdev	General Physicain	Orthopedics	2020-08-05	13:05:13	5
3	Revati	Mehta	Orthopedic surgeons	Orthopaedics	2015-12-02	08:35:00	2
4	Saka	Ghate	Pathologist	Cardiology	*NULL*	*NULL*	5
5	Matsya	Sridhar	Pediatrician	Pediatrics	*NULL*	*NULL*	5

2.Show a prescription report with medicine name, prescription_id and unit number:

```
1 SELECT Includes.prescription_id, Includes.unit, Medicine.name
2 FROM Includes LEFT JOIN
3 Medicine ON Includes.medicine_id = Medicine.medicine_id;
```

✔ Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

SELECT Includes.prescription_id, Includes.unit, Medicine.name FROM Includes LEFT JOIN Medicine ON Includes.medicine_id = Medicine.medicine_id;

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

Extra options

prescription_id	unit	name
1	1	Ifosfamide
1	1	Cisplatin
5	1	Cisplatin
1	1	Methotrexate
5	1	Methotrexate
5	1	Busulfan

3.Display an appointment Menu containing only the doctors who have appointment booked.

```
1 SELECT D.doctor_id, CONCAT(D.first_name, " ", D.last_name), D.specialization, A.date, A.start_time
2 FROM (Doctor AS D) JOIN (Appointment AS A)
3 ON D.doctor_id = A.doctor_id
4 ORDER BY D.first_name ASC, D.last_name ASC
5
6
```

☐ Show all | Number of rows: 25 ⌄ | Filter rows: [Search this table]

Extra options

doctor_id	CONCAT(D.first_name, " ", D.last_name)	specialization	date	start_time
2	Ameretat Sachdev	Orthopedics	2020-08-05	13:05:13
3	Revati Mehta	Orthopaedics	2015-12-02	08:35:00

☐ Show all | Number of rows: 25 ⌄ | Filter rows: [Search this table]

4.Display a Bill Report containing all the details of test, procedure, medicine and patient with their respective bill details.

Run SQL query/queries on database hms: ⓘ

```sql
1  SELECT Bill.*, Test.name as `Tests`, Test.cost as `Test_Cost`,`Procedure`.name as `Procedures`, `Procedure`.cost as `Procedure_Cost`,
2  Medicine.name as `drug`, Includes.unit as `Medicine_Qt`, Medicine.price as `Medicine_Unit_Cost`,
3  Patient.patient_id, CONCAT(Patient.first_name, " ", Patient.last_name) as `Patient_Name`
4  FROM Bill LEFT JOIN
5  Pays ON Pays.bill_number = Bill.bill_number LEFT JOIN
6  Patient ON Patient.patient_id = Pays.patient_id LEFT JOIN
7  Medical_Bill ON Medical_Bill.bill_number = Bill.bill_number LEFT JOIN
8  Includes ON Includes.prescription_id = Medical_Bill.prescription_id LEFT JOIN
9  Medicine ON Medicine.medicine_id = Includes.medicine_id LEFT JOIN
10 Bill_Diag ON Bill_Diag.bill_number = Bill.bill_number LEFT JOIN
11 Involves ON Involves.diagnosis_id = Bill_Diag.diagnosis_id LEFT JOIN
12 Test ON Test.test_id = Involves.test_id LEFT JOIN
13 Treatment_Bill ON Treatment_Bill.bill_number = Bill.bill_number LEFT JOIN
14 Consists ON Consists.treatment_id = Treatment_Bill.treatment_id LEFT JOIN
15 `Procedure` ON `Procedure`.procedure_id = Consists.procedure_id;
```

☐ Show all | Number of rows: 25 ⌄ | Filter rows: [Search this table] | Sort by key: None ⌄

bill_number	bill_type	total	subtotal	taxes	paid	Tests	Test_Cost	Procedures	Procedure_Cost	drug	Medicine_Qt	Medicine_Unit_Cost	patient_id	Patient_Name
1	2	762000.00	700000.00	62000.00	0	NULL	NULL	Coronary Angiogram	63500.00	NULL	NULL	NULL	1	Jyotish Divan
1	2	762000.00	700000.00	62000.00	0	NULL	NULL	Angioplasty	412750.00	NULL	NULL	NULL	1	Jyotish Divan
1	2	762000.00	700000.00	62000.00	0	NULL	NULL	Knee Replacement	539750.00	NULL	NULL	NULL	1	Jyotish Divan
2	1	1000.50	969.50	31.00	1	NULL	NULL	NULL	NULL	Ifosfamide	1	551.65	1	Jyotish Divan
2	1	1000.50	969.50	31.00	1	NULL	NULL	NULL	NULL	Cisplatin	1	360.00	1	Jyotish Divan
2	1	1000.50	969.50	31.00	1	NULL	NULL	NULL	NULL	Methotrexate	1	57.85	1	Jyotish Divan
3	0	6300.00	6000.00	300.00	1	Blood Sugar	5000.00	NULL	NULL	NULL	NULL	NULL	3	Supriya Shanker
4	2	855000.00	850500.00	4500.00	1	NULL	NULL	Coronary Angiogram	63500.00	NULL	NULL	NULL	3	Supriya Shanker
4	2	855000.00	850500.00	4500.00	1	NULL	NULL	Knee Replacement	539750.00	NULL	NULL	NULL	3	Supriya Shanker
5	0	8000.00	7500.00	500.00	1	MRI	3000.00	NULL	NULL	NULL	NULL	NULL	2	Yasmine Kumar
5	0	8000.00	7500.00	500.00	1	CT Scan	4000.00	NULL	NULL	NULL	NULL	NULL	2	Yasmine Kumar
6	1	952.95	832.85	120.10	1	NULL	NULL	NULL	NULL	Cisplatin	1	360.00	2	Yasmine Kumar
6	1	952.95	832.85	120.10	1	NULL	NULL	NULL	NULL	Methotrexate	1	57.85	2	Yasmine Kumar
6	1	952.95	832.85	120.10	1	NULL	NULL	NULL	NULL	Busulfan	1	415.00	2	Yasmine Kumar

AGGREGATE FUNCTIONS

1.Display count of doctors in the same room with respective room number.

```
Run SQL query/queries on table hms.doctor:
1 SELECT doctor.room_no,count(doctor.doctor_id)
2 from doctor
3 group by room_no
4 order by count(doctor.doctor_id)DESC;
```

				room_no	count(doctor.doctor_id) ▾ 1
☐	🖉 Edit	Copy	⊖ Delete	5	3
☐	🖉 Edit	Copy	⊖ Delete	2	2

2. Displays the doctor who has minimum salary.

SELECT * FROM doctor

WHERE salary = (

  SELECT MIN(salary)

  FROM doctor);

```
Run SQL query/queries on table hms.doctor:
1 SELECT * FROM doctor
2 WHERE salary = (
3     SELECT MIN(salary)
4     FROM doctor);
```

✔ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

SELECT * FROM doctor WHERE salary = (SELECT MIN(salary) FROM doctor);

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ | Filter rows: Search this table

Extra options

				doctor_id	first_name	last_name	contact_no	address	salary	specialization	role	room_no
☐	🖉 Edit	Copy	⊖ Delete	4	Saka	Ghate	07925501915	29 /b, Asopalav, Khanpur	20000.00	Cardiology	Pathologist	5

3. Displays the doctor who has maximum salary.

Run SQL query/queries on table hms.doctor:

```
1 SELECT * FROM doctor
2 WHERE salary = (
3     SELECT MAX(salary)
4     FROM doctor);
```

	doctor_id	first_name	last_name	contact_no	address	salary	specialization	role	room_no
☐ ✎ Edit ⅋ Copy ⊝ Delete	2	Ameretat	Sachdev	02226609802	Rotary Service Centre, Mumbai	150000.00	Orthopedics	General Physicain	5

4.Display average amount of the total amount on Bill of type 2

select AVG(total) as average_of_type_2_bill

from bill

where bill_type="2"

Run SQL query/queries on table hms.bill: ⓘ

```
1 select AVG(total) as average_of_type_2_bill
2 from bill
3 where bill_type="2"
```

average_of_type_2_bill
808500.000000

SET OPERATIONS

1. Display first_name of all the doctors and nurses

Run SQL query/queries on table hms.doctor: ⓘ

```
1 SELECT doctor.first_name FROM doctor
2 UNION
3 SELECT nurse.first_name FROM nurse;
```

first_name
Devika
Ameretat
Revati
Saka
Matsya
Maheshvari
Atman
Archana
Bharat
Manda

2.Display addresses of all nurses and patients

SELECT nurse.first_name,nurse.address FROM nurse

UNION

SELECT patient.first_name,patient.address FROM patient;

Run SQL query/queries on table hms.patient: ⓘ

```
1 SELECT nurse.first_name,nurse.address FROM nurse
2 UNION
3 SELECT patient.first_name,patient.addressX FROM patient;
4
```

first_name	address
Maheshvari	241 -, Natraj Market, Sv Road, Malad, Mumbai
Atman	Chittranjan Nagar, Rajawadi, Ghatkoper, Mumbai
Archana	22 Rachna, Sector , Vashi, Mumbai
Bharat	109 , Parasrampuria Chambers, Opp Rly Stn, Malad (...
Manda	448 A, Girgaum, Mumbai
Jyotish	417 ,sai Chambers, Narshi Natha St, Chinch Bunder
Yasmine	349 ,Allied Indl Estae Off Mmc, Off M.m.c Road, Ma...
Supriya	41 /, Sadar Nagar No , Sion

4.Display first_name of all the doctors and patients

 SELECT doctor.first_name FROM doctor

UNION

SELECT patient.first_name FROM patient;

```
1 SELECT doctor.first_name FROM doctor
2 UNION
3 SELECT patient.first_name FROM patient;
```

first_name
Devika
Ameretat
Revati
Saka
Matsya
Jyotish
Yasmine
Supriya

FUNCTIONS AND PROCEDURES:

1.Function: Create an appointment by the consultant

DELIMITER $$

CREATE FUNCTION create_appointment(doctor int, patient_id int, start_datetime datetime)

RETURNS int

BEGIN

DECLARE created int;

IF (SELECT start_time FROM Appointment

   WHERE `date` = DATE(start_datetime) AND

   doctor_id = doctor AND start_time

   BETWEEN (SELECT subtime(TIME(start_datetime),'00:29:59'))

   AND (SELECT subtime(TIME(start_datetime),'-00:29:59')))

   THEN SET created = 0;

ELSE

   INSERT INTO Appointment VALUES

   (doctor, patient_id, DATE(start_datetime), TIME(start_datetime));

   SET created = 1;

END IF;

RETURN created;

END;

$$ DELIMITER ;

```
Run SQL query/queries on database hms:  ⓘ

 1  -- Create an appointment by the consultant
 2  DELIMITER $$
 3  CREATE FUNCTION create_appointment(doctor int, patient_id int, start_datetime datetime)
 4  RETURNS int
 5  BEGIN
 6  DECLARE created int;
 7  IF (SELECT start_time FROM Appointment
 8      WHERE `date` = DATE(start_datetime) AND
 9      doctor_id = doctor AND start_time
10      BETWEEN (SELECT subtime(TIME(start_datetime),'00:29:59'))
11      AND (SELECT subtime(TIME(start_datetime),'-00:29:59')))
12      THEN SET created = 0;
13  ELSE
14      INSERT INTO Appointment VALUES
15      (doctor, patient_id, DATE(start_datetime), TIME(start_datetime));
16      SET created = 1;
17  END IF;
18  RETURN created;
19  END;
20  $$ DELIMITER ;
```

```
CREATE FUNCTION create_appointment(doctor int, patient_id int, start_datetime datetime) RETURNS int BEGIN DECLARE created int; IF (SELECT start_time FROM Appointment WHERE
`date` = DATE(start_datetime) AND doctor_id = doctor AND start_time BETWEEN (SELECT subtime(TIME(start_datetime),'00:29:59')) AND (SELECT
subtime(TIME(start_datetime),'-00:29:59'))) THEN SET created = 0; ELSE INSERT INTO Appointment VALUES (doctor, patient_id, DATE(start_datetime), TIME(start_datetime)); SET
created = 1; END IF; RETURN created; END;;
```

[Edit inline] [Edit] [Create PHP code]

```
1 select doctor_id,create_appointment(1,3,"2020-08-05 05:56:24") from doctor;
```

✔ Showing rows 0 - 4 (5 total, Query took 0.0066 seconds.)

```
select doctor_id,create_appointment(1,3,"2020-08-05 05:56:24") from doctor;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ⌄   Filter rows: Search this table   Sort by key: N

Extra options

←T→			doctor_id	create_appointment(1,3,"2020-08-05 05:56:24")
☐ 🖉 Edit ⧉ Copy ⊖ Delete			1	1
☐ 🖉 Edit ⧉ Copy ⊖ Delete			3	0
☐ 🖉 Edit ⧉ Copy ⊖ Delete			2	0
☐ 🖉 Edit ⧉ Copy ⊖ Delete			4	0
☐ 🖉 Edit ⧉ Copy ⊖ Delete			5	0

⬆ ☐ Check all   With selected: 🖉 Edit   ⧉ Copy   ⊖ Delete   📊 Export

A new appointment has been for doctor_id 1 and patient_id 3created using function

←T→			doctor_id	patient_id	date	start_time
☐ 🖉 Edit ⧉ Copy ⊖ Delete			1	3	2020-08-05	05:56:24
☐ 🖉 Edit ⧉ Copy ⊖ Delete			2	2	2020-08-05	13:05:13
☐ 🖉 Edit ⧉ Copy ⊖ Delete			3	3	2015-12-02	08:35:00

Procedure

```
1  -- mark bill paid
2
3  DELIMITER $$
4  CREATE OR REPLACE PROCEDURE mark_bill_paid(
5  IN bill_id INT,
6  OUT success_state INT
7  )
8  BEGIN
9
10 IF EXISTS(SELECT * FROM Bill WHERE bill_number=bill_id) THEN
11 IF ((SELECT paid FROM Bill WHERE bill_number=bill_id) = 0) THEN
12 UPDATE Bill SET paid = 1 WHERE bill_number = bill_id;
13 SET success_state = 0;
14 ELSE
15 SET success_state = -1;
16 END IF;
17 END IF;
18
19 END;
20 $$ DELIMITER ;
```

Clear   Format   Get auto-saved query

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0066 seconds.)

CREATE OR REPLACE PROCEDURE mark_bill_paid(IN bill_id INT, OUT success_state INT) BEGIN IF EXISTS(SELECT * FROM Bill WHERE bill_number=bill_id) THEN IF ((SELECT paid FROM Bill WHERE bill_number=bill_id) = 0) THEN UPDATE Bill SET paid = 1 WHERE bill_number = bill_id; SET success_state = 0; ELSE SET success_state = -1; END IF; END IF; END;;

[Edit inline] [Edit] [Create PHP code]

BEFORE PROCEDURE CALL:

		bill_number	bill_type	total	subtotal	taxes	paid
☐	✎ Edit ⧉ Copy ⊖ Delete	1	2	762000.00	700000.00	62000.00	0
☐	✎ Edit ⧉ Copy ⊖ Delete	2	1	1000.50	969.50	31.00	1
☐	✎ Edit ⧉ Copy ⊖ Delete	3	0	6300.00	6000.00	300.00	1
☐	✎ Edit ⧉ Copy ⊖ Delete	4	2	855000.00	850500.00	4500.00	1
☐	✎ Edit ⧉ Copy ⊖ Delete	5	0	8000.00	7500.00	500.00	1
☐	✎ Edit ⧉ Copy ⊖ Delete	6	1	952.95	832.85	120.10	1

Run SQL query/queries on database hms:

```
1 CALL mark_bill_paid( 1, @out_value)
```

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0061 seconds.)

CALL mark_bill_paid(1, @out_value);

[Edit inline] [Edit] [Create PHP code]

AFTER PROCEDURE CALL:

←T→				bill_number	bill_type	total	subtotal	taxes	paid
☐	Edit	Copy	Delete	1	2	762000.00	700000.00	62000.00	1
☐	Edit	Copy	Delete	2	1	1000.50	969.50	31.00	1
☐	Edit	Copy	Delete	3	0	6300.00	6000.00	300.00	1
☐	Edit	Copy	Delete	4	2	855000.00	850500.00	4500.00	1
☐	Edit	Copy	Delete	5	0	8000.00	7500.00	500.00	1
☐	Edit	Copy	Delete	6	1	952.95	832.85	120.10	1

TRIGGERS AND CURSORS

Trigger

1. Create a trigger which initiates a delete first on prescription and bill before Diagnosis if there is an attempt to delete data in diagnosis table.
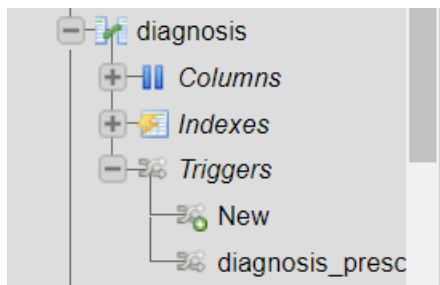
```
1 DELIMITER $$
2 CREATE OR REPLACE TRIGGER diagnosis_presc_bill_delete BEFORE DELETE ON Diagnosis
3 FOR EACH ROW
4 BEGIN
5 DELETE FROM Bill WHERE bill_number = (SELECT B.bill_number FROM Bill_Diag B WHERE B.diagnosis_id = OLD.diagnosis_id);
6 DELETE FROM Prescription WHERE prescription_id = (SELECT P.prescription_id FROM Diag_Presc P WHERE P.diagnosis_id = OLD.diagnosis_id);
7 END;
8 $$
9 DELIMITER ;
```

Show query box

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0122 seconds.)

CREATE OR REPLACE TRIGGER diagnosis_presc_bill_delete BEFORE DELETE ON Diagnosis FOR EACH ROW BEGIN DELETE FROM Bill WHERE bill_number = (SELECT B.bill_number FROM Bill_Diag B WHERE B.diagnosis_id = OLD.diagnosis_id); DELETE FROM Prescription WHERE prescription_id = (SELECT P.prescription_id FROM Diag_Presc P WHERE P.diagnosis_id = OLD.diagnosis_id); END;;

[Edit inline] [Edit] [Create PHP code]

diagnosis
  Columns
  Indexes
  Triggers
    New
    diagnosis_presc

Before:

			bill_number	bill_type	total	subtotal	taxes	paid
☐	Edit	Copy	Delete	1	2	762000.00	700000.00	62000.00	1
☐	Edit	Copy	Delete	2	1	1000.50	969.50	31.00	1
☐	Edit	Copy	Delete	3	0	6300.00	6000.00	300.00	1
☐	Edit	Copy	Delete	4	2	855000.00	850500.00	4500.00	1
☐	Edit	Copy	Delete	5	0	8000.00	7500.00	500.00	1
☐	Edit	Copy	Delete	6	1	952.95	832.85	120.10	1

| Browse | Structure | SQL | Search | Insert | Exp |

Show query box

✔ 1 row affected. (Query took 0.0003 seconds.)

DELETE FROM diagnosis WHERE diagnosis_id=2;

[Edit inline] [Edit] [Create PHP code]

After:

←T→				bill_number	diagnosis_id
☐	🖉 Edit	⚏ Copy	⊖ Delete	3	3

↑⌐ ☐ Check all   *With selected:* 🖉 Edit   ⚏ Copy   ⊖

←T→				prescription_id
☐	🖉 Edit	⚏ Copy	⊖ Delete	1

←T→				bill_number	bill_type	total	subtotal	taxes	paid
☐	🖉 Edit	⚏ Copy	⊖ Delete	1	2	762000.00	700000.00	62000.00	1
☐	🖉 Edit	⚏ Copy	⊖ Delete	2	1	1000.50	969.50	31.00	1
☐	🖉 Edit	⚏ Copy	⊖ Delete	3	0	6300.00	6000.00	300.00	1
☐	🖉 Edit	⚏ Copy	⊖ Delete	4	2	855000.00	850500.00	4500.00	1
☐	🖉 Edit	⚏ Copy	⊖ Delete	6	1	952.95	832.85	120.10	1

```
DELIMITER $$

CREATE OR REPLACE TRIGGER diagnosis_presc_bill_delete BEFORE DELETE ON Diagnosis

FOR EACH ROW

BEGIN

DELETE FROM Bill WHERE bill_number = (SELECT B.bill_number FROM Bill_Diag B WHERE
B.diagnosis_id = OLD.diagnosis_id);

DELETE FROM Prescription WHERE prescription_id = (SELECT P.prescription_id FROM Diag_Presc P
WHERE P.diagnosis_id = OLD.diagnosis_id);

END;

$$

DELIMITER ;
```

Cursor

1.Using a cursor shift the doctor names and specialization from the doctor table to a new table crs_patconc.

```
DELIMITER $$

CREATE PROCEDURE move_doc_id_specialization()

BEGIN

DECLARE Doctor_names varchar(50);

DECLARE specialization varchar(50);

DECLARE done1 int DEFAULT 0;

DECLARE done2 int DEFAULT 0;

DECLARE crs1 CURSOR FOR SELECT doctor.first_name FROM doctor;

DECLARE crs2 CURSOR FOR SELECT doctor.specialization FROM doctor;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done1 = 1;

OPEN crs1;

OPEN crs2;

getnames : LOOP

FETCH crs1 INTO Doctor_names;

FETCH crs2 INTO specialization;

IF done1=1 THEN LEAVE getnames;

END IF;

INSERT INTO crs_doc_special VALUES (doctor_names,specialization);

END LOOP getnames;

CLOSE crs1;

CLOSE crs2;
```

END $$;DELIMITER;

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0114 seconds.)

```
call move_det();
```

[Edit inline] [Edit] [Create PHP code]

✔ Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.)

```
SELECT * FROM `crs_patconc`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Re

☐ Show all | Number of rows: 25 ⌄ | Filter rows: Searc

Extra options

doctor_names	specialization
1	Neuro surgeon
3	Orthopedics
2	Orthopaedics
4	Cardiology
5	Pediatrics

DEVELOPING A FRONTEND