## Placeholder image

Winter semester 2022

CSE 250 Database Management System

**Medical Store Management System**

Group Members :-

AU2040120 Ashish Chauhan

AU2040141 Kuntal Panchal

AU2040186 Jainam R Shah

AU2040205 Nihar Jani

# Details Included In The Project

* General Description

The objective of our project is to make the medical shop owner’s life easy by handling a massive amount of data related to the medical shop.

The data relating to medical shop owners include working employees, hospitals, suppliers of medicine, patient details, doctor details ( Doctor who prescribed the particular medicine), and sales details.

The medical shop’s database manager has to login into our database. After that, he/she can enter the data of the employee, which include age, city, mobile number, joining date, or data of hospital, which include email id, city, name, mobile number, and hospital id or supplier details which include contact number of supplier, name and city or the he/she can enter all the details together also , and all these details can be fetched from the front-end interface or user interface.

* ER Digram Explanation

The above ER Diagram describes the correlated entities and relationships among the different entities. The main entity is medical\_shop which has attributes namely city, mobile\_number, email\_id, name, med\_shop\_id. Here the candidate keys are email\_id, mobile\_number and med\_shop\_id. Primary key is med\_shop\_id.

The entity hospital has a **many to one relation** with medical\_shop and it has its own attributes namely email\_id, mobile\_number, hospital\_id, city, name. Here the **primary key is hospital\_id** and the rest of the attributes are alternate keys except city and name as both of them are not included in the candidate keys.

The employee entity has **many to one relation** with medical\_shop, because in one medical shop, many employee work.The attributes of employee entity are age, city, mobile number, gender, salary, employee\_id, joined\_date, . and name. The **primary key is employee\_id**. The **candidate key are employee\_id, mobile number , and name.**

The supplier entity has **many to one relation** with medical shop, becasue one medical shop buys medicine from many suppliers and sometimes it vary from medicine to medicine.The attributes of supplier entity are name, mobile number, email id , and city. The **candidate key are email id and mobile number**.

The equipment entity has **many to one relationship** with medical shop, because one medical shop uses many equipment.The attributes of equipment entity are price, exp\_date, mfg\_date, code, product\_name, and product\_type.The **candidate key is product\_name, product\_type and mfg\_date**.

The doctor entity has **many to many relationship** with equipment, because many doctor can use many equipment.The attributes of doctor entity are doctor name, doctor id, age, gender, mobile number, specilaity. The **primary key is doctor id**. The **candidate key is doctor id and speciality.**

The patient is an entity which has **many to one relationship** with doctor, because one specilaist doctor can treat many patients of similar problem.The atributes of patient are gender, mobile number, city, age, name, patient id. The **primary key is patient id**. The **candidate key is patient id, and mobile number**.

The billing entity as **one to one relationship** with the patient, because each patient has to pay their own bill the reason behind it is the bill also depends upon the doctor selected, the medicines given, price of the medicines, due to which becomes specific to a person. The attributes of the billing entity are amount, bill id, patient name, city, mobile number, date of sale, and patient age. The **primary key is patient id**. The **candidate key is patient id and patient name.**

# Table Description

We have created 11 tables for the Medical database management system.

A brief description of the tables is described below:

* Supplier
* medical shop
* Doctor
* Patient
* Equipment
* Employee
* Hospital
* Bill
* Sale
* Works
* Prescribe

**Connection of Tables with each other and its explanation:**

* Medical shop is connected to hospitals because of records of hospital details
* Medical shop is connected to employee because the employee works in the medical shop. And the employee table stores data about employees.
* Employee shift details are stored in the work table, whose primary work is to store details about employees' shifts.
* The details from the medical shop get supplies are stored on the supplies table, such as the supplier's name, city of the supplier, and email id of the supplier.
* The data about quantity sales of equipment are stored in the equipment table,
* The equipment used by the doctor or doctor uses different equipment, and the doctor's details are stored in the doctor’s table.
* The doctor prescribes the patient, and the patient's data is stored on the patient's table.
* The patient pays the bill, therefore the details about bill is stored in bill table.

**Description about Procedures, Functions and Triggers**

For our project we have created 8 procedures, 5 functions and 5 triggers.

All of them have their different functionality .

1. The 1 st procedure that we have created is used for displaying the quantity of the required/selected item from the medical shop database and displaying that which medicine is that.
2. The 2 nd procedure that we have created is used such that the different items which arenpurchased by the customer are been added all up to create a bill.
3. The 3 rd procedure that we have created is used for appointing a patient to a particular doctor by accessing the data from the database.
4. The 4 th procedure that we have created is used for displaying the employee details.
5. The 5 th procedure that we have created is used for displaying that from shop the patient has taken the medicine.
6. The 6 th procedure that we have created is used for displaying the doctor names and also shows that which patient is treated by that doctor.
7. The 7 th procedure that we have created is used to display the details of hospitals which are being supplied medicines by a particular shop.
8. The 8 th procedure that we have created is used for displaying the doctor from its specification

Now coming to the function part we get to see that

1. The 1 st function that we have created is used for showing the total number of medicines from the prescibe table.
2. The 2nd function that we have created is used for showing the total number of patient from the patient table.
3. The 3 rd function that we have created is used for showing the total number of employee from the employee table.
4. The 4 th function that we have created is used for showing the total number of doctor which are there from the doctors table.
5. The 5 th function that we have created is used for showing the total number of supplier which are there from the supplier table.

Now coming to the Triggers Part we get to see that

1) The 1 st trigger that we have created is used for giving 10% discount on final amount which will store that in another table called discount table.

2) The 2 nd trigger that we have created is used for checking the age of a doctor before signing up into the app.

3) The 3 rd trigger that we have created is used for checking whether the product is valid or not for the medical shop.

4) The 4 th trigger that we have created is used for deleting employee records if he/she leaves the shop.

5) The 5 th trigger that we have created is displaying the salary changes of the employee.

# 

# SYSTEM REQUIREMENT SPECIFICATION

# 

# Hardware:

# RAM: Minimum 4 GB physical memory, 6 GB recommended.

# Hard Disk: Minimum 1GB free disk space for installation, 12 GB or more Recommended for working smoothly.

# Processor: Intel Core i3

# Software:

# Operating System - Windows 10

# Database Management System - Oracle 21c

# Visual Studio Code.

# 

# User Interface:

# 

# Front End :- Node JS

# Back End :- Oracle 21c

# 

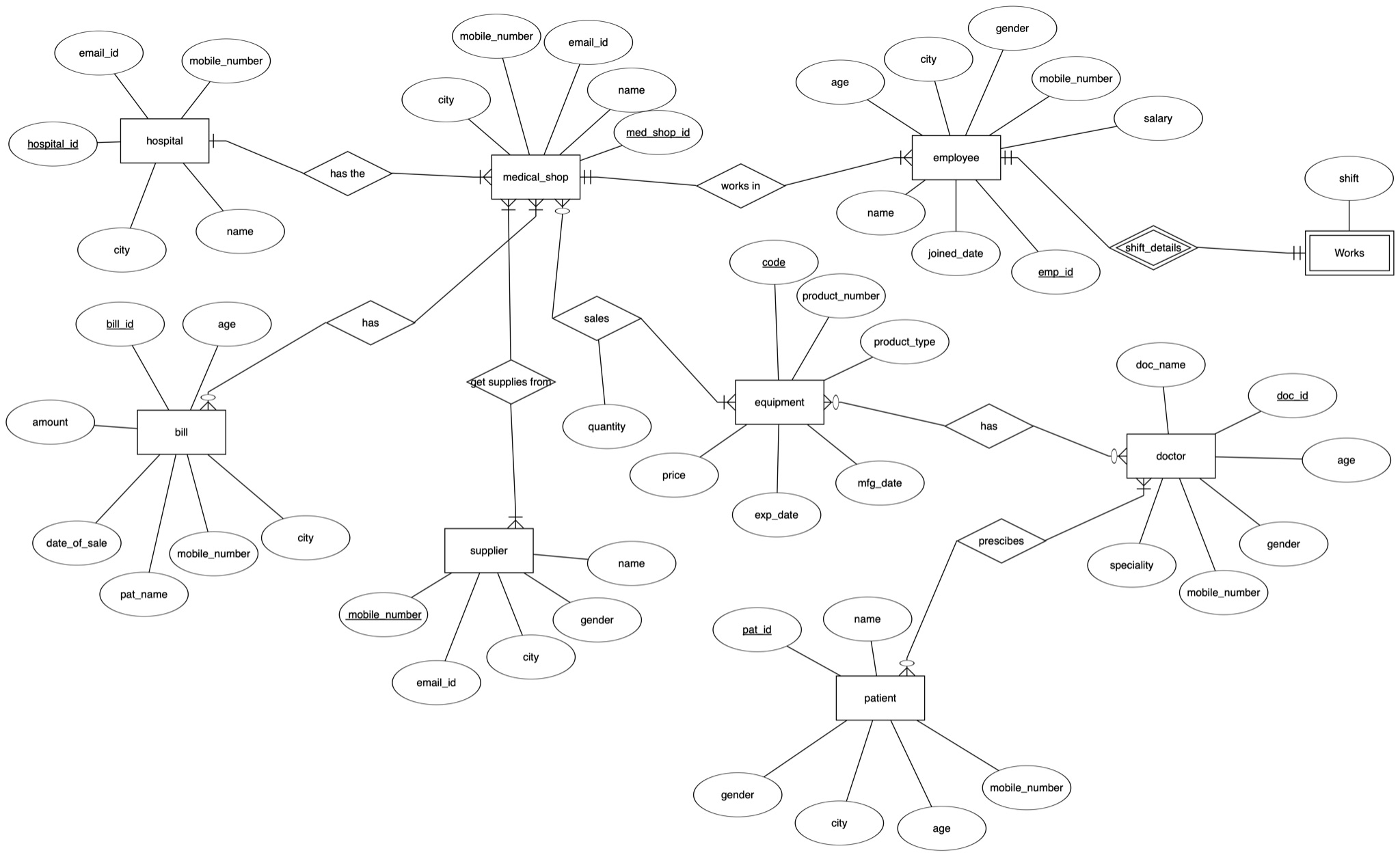
# ERD tool:

# ERDPlus

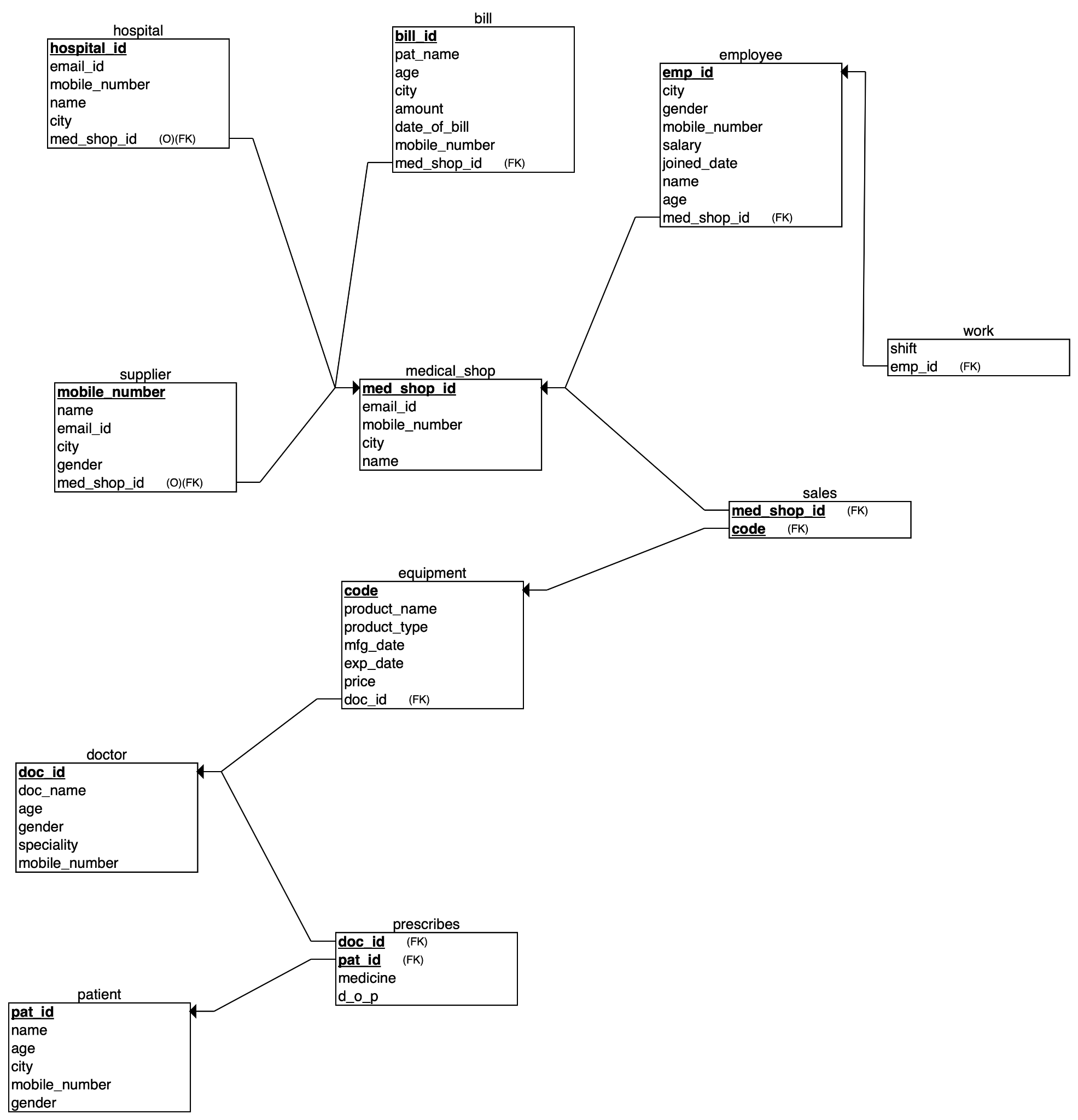
# 

# 

# ER Diagram

****

**TABLE DESIGN**



# Table 1 :- supplier

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | mobile\_number | integer | Primary Key | A unique number which every supplier has |
| 2) | name | varchar(n) | not null | Name of supplier |
| 3) | city | varchar(n) | not null | City of supplier |
| 4) | gender | varchar(n) | not null | Gender of supplier |
| 5) | email\_id | varchar(n) | not null | Email id of supplier |
| 6) | med\_shop\_id | number(n) | Foreign Key referencing medical\_shop (med\_shop\_id) | A foreign key referring to med\_shop\_id which helps in integrating this table with medical\_shop table |

# Table 2 :- medical\_shop

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | med\_shop\_id | integer | Primary Key | A unique id which every medical shop has. |
| 2) | city | varchar(n) | not null | Name of city where the medical shop is there. |
| 3) | name | varchar(n) | not null | Owner’s name . |
| 4) | email\_id | varchar(n) | unique, not null | Email id of the medical shop |
| 5) | mobile\_number | integer | unique, not nll | Contact number of the medical shop |

# 

Table 3 :-  **doctor**

# 

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | doc\_id | number(n) | Primary Key | A unique Id which every doctor have. |
| 2) | doc\_name | varchar(n) | not null | Name of doctor |
| 3) | speciality | varchar(n) | not null | Speciality of doctor |
| 4) | mobile\_number | integer | unique,not null | Mobile number of doctor |
| 5) | age | number(n) | not null | Age of doctor |
| 6) | gender | varchar(n) | not null | Gender of doctor |

# 

# 

# Table 4 :- patient

# 

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | pat\_id | number(n) | Primary Key | A unique id which every patient have |
| 2) | city | varchar(n) | not null | City of patient |
| 3) | name | varchar(n) | not null | Name of patient |
| 4) | gender | varchar(n) | not null | Gender of patient |
| 5) | mobile\_number | integer | unique, not null | Mobile number of patient |
| 6) | age | number(n) | not null | Age of patient |
| 7) | doc\_id | number(n) | Foreign Key referencing doctor(doc\_id) | A foreign key referring to doc\_id which helps in integrating this table with doctor table |

# 

# 

# Table 5 :- equipment

# 

# 

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | code | number(n) | Primary Key | A unique code which every equipment has |
| 2) | product\_name | varchar(n) | not null | Product name of equipment |
| 3) | product\_type | varchar(n) | not null | Product type of equipment |
| 4) | mfg\_date | date | not null | Manufacturing date of equipment |
| 5) | exp\_date | date | not null | Expiry date of equipment |
| 6) | price | numeric(p,s) | not null | Price of equipment |
| 7) | doc\_id | number(n) | Foreign Key referencing doctor(doc\_id) | A foreign key referring to doc\_id which helps in integrating this table with doctor table |

# 

Table 6 :-  **employee**

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | name | varchar(n) | not null | Name of employee |
| 2) | emp\_id | varchar(n) | Primary Key | A unique id which every employee has |
| 3) | city | varchar(n) | not null | City of employee |
| 4) | join\_date | date | not null | Joining date of employee |
| 5) | mobile\_number | integer | Unique, not null | Mobile numberof employee |
| 6) | age | number(n) | not null | Age of employee |
| 7) | salary | number(n) | not null | Salary of employee |
| 8) | gender | varchar(n) | not null | Gender of employee |
| 9) | med\_shop\_id | number(n) | Foreign Key referencing medical\_shop (med\_shop\_id) | A foreign key referring to med\_shop\_id which helps in integrating this table with medical\_shop table |

Table 7 :-  **hospital**

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | hospital\_id | number(n) | Primary Key | A unique id which every hospital has |
| 2) | name | varchar(n) | not null | Name of hospital |
| 3) | email\_id | varchar(n) | not null | Email id of hospital |
| 4) | mobile\_number | integer | Unique,not null | Mobile number of hospital |
| 5) | city | varchar(n) | not null | City of hospital |
| 6) | med\_shop\_id | number(n) | Foreign Key referencing medical\_shop (med\_shop\_id) | A foreign key referring to med\_shop\_id which helps in integrating this table with medical\_shop table |

Table 8 :-  **bill**

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | bill\_id | number(n) | Primary Key | A unique id which every bill has |
| 2) | date\_of\_sale | date | not null | Date when the bill was given |
| 3) | age | number(n) | not null | Age of the patient who’s bill is there |
| 4) | pat\_name | varchar(n) | not null | Patient name |
| 5) | mobile\_number | integer | Unique,not null | Mobile number of patient |
| 6) | city | varchar(n) | not null | City where the patient lives |
| 7) | product | varchar(n) | not null | The type of medicine which is given |
| 8) | amount | numeric(p,s) | not null | The total amount in bill |
| 9) | med\_shop\_id | number(n) | Foreign Key referencing medical\_shop (med\_shop\_id) | A foreign key referring to med\_shop\_id which helps in integrating this table with medical\_shop table |

# Table 9 :- sale

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | quantity | number(n) | not null | Quantity of medicine |
| 2) | code | number(n) | PK(combined) | A foreign key referring to code which helps in integrating this table with equipment table |
| 3) | med\_shop\_id | number(n) | PK(combined) | A foreign key referring to med\_shop\_id which helps in integrating this table with medical\_shop table |

# 

# Table 10 :- works

| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | shift | varchar(n) | not null | shift in which the employee works |
| 2) | emp\_id | number | Foreign Key referencing employee (emp\_id) | A foreign key referring to emp\_id which helps in integrating this table with employee table |

Table 11 :-  **prescribe**

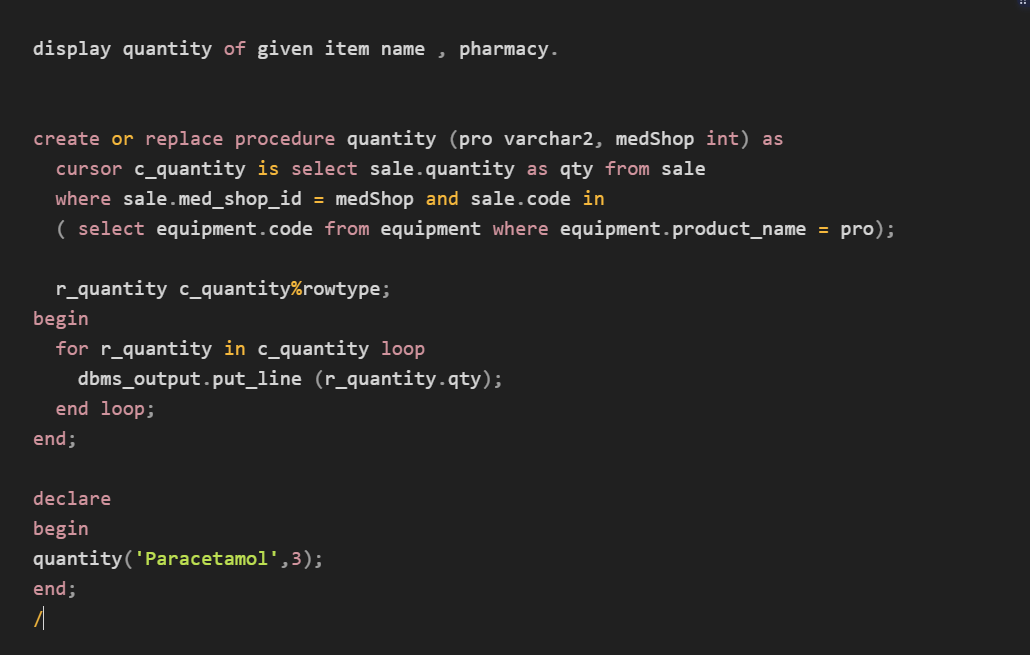
| **SR NO** | **Column Name** | **Data Type** | **Constraint** | **Description** |
| --- | --- | --- | --- | --- |
| 1) | date\_of\_pres | date | not null | Date when the medicine was prescribed |
| 2) | medicine | varchar(n) | not null | The type of medicine prescibed |
| 3) | doc\_id | number(n) | Foreign Key referencing doctor(doc\_id) | A foreign key referring to doc\_id which helps in integrating this table with doctor table |
| 4) | pat\_id | number(n) | Foreign Key referencing patient(pat\_id) | A foreign key referring to pat\_id which helps in integrating this table with patient table |

# Procedures

# 

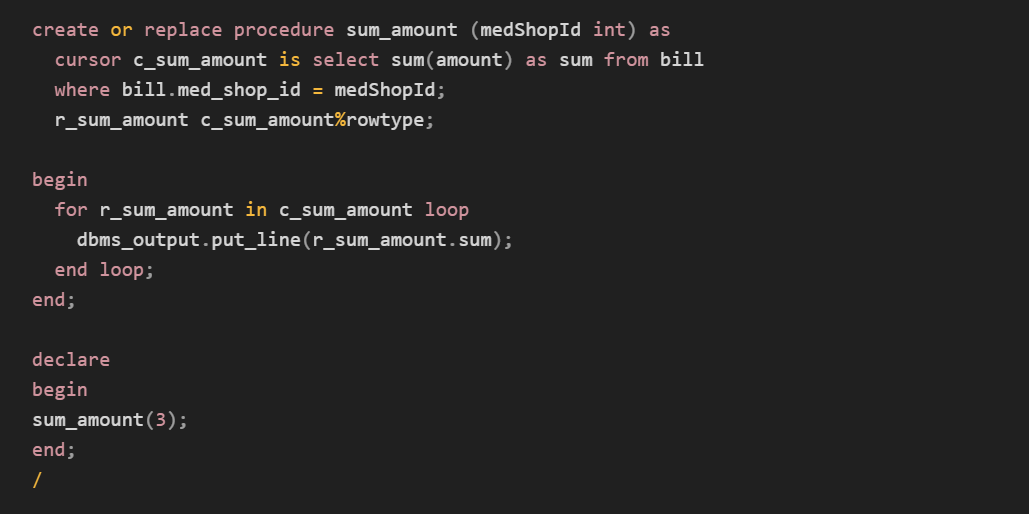
1. **Item\_counter**

The following procedure is used to display the quantity of the required/selected item from the medical shop database. The input attribute required to display the quantity of that item is ‘item name’ and ‘pharmacy’. The output example is shown below for the item ‘Paracetamol’ with its corresponding quantity i.e. 3.



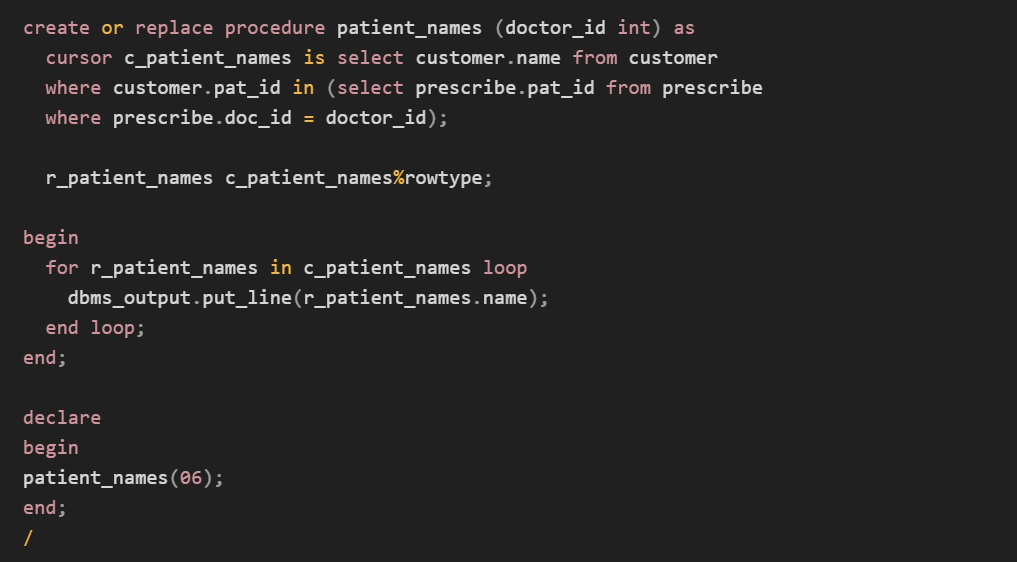
**2. Sum\_amount**

The following procedure takes the different items purchased by the customer and adds them all up to create a bill. It selects the tuples of items which are selected/purchased by the customer and adds the price attribute of the selected tuples and displays the table of selected items with their name and price.

****

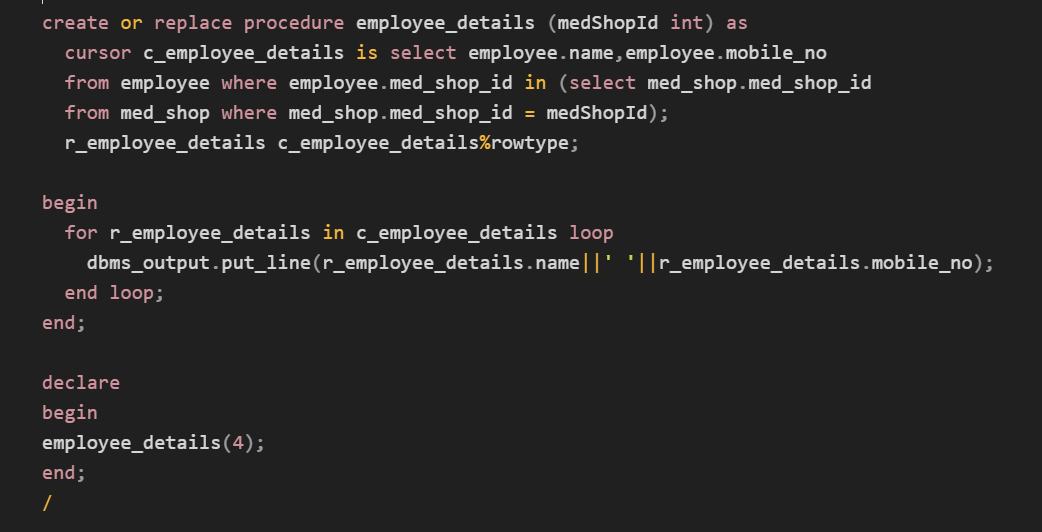
**3. Patient\_names**

The following procedure selects or appoints a patient to a particular doctor by accessing the data from the database using the cursor function.

****

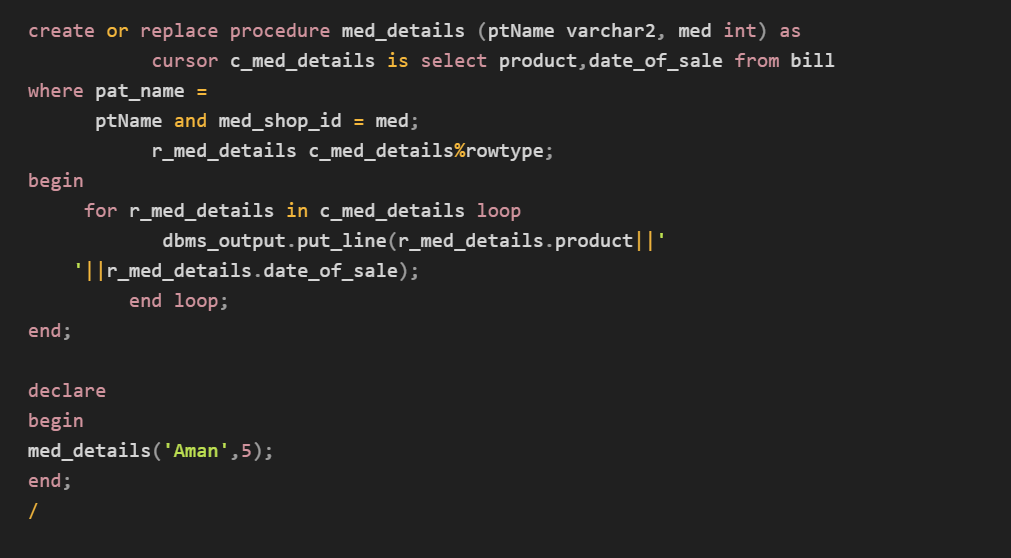
**4. Employee\_details**

The following procedure displays employee details and all the tuples which are included in the input.



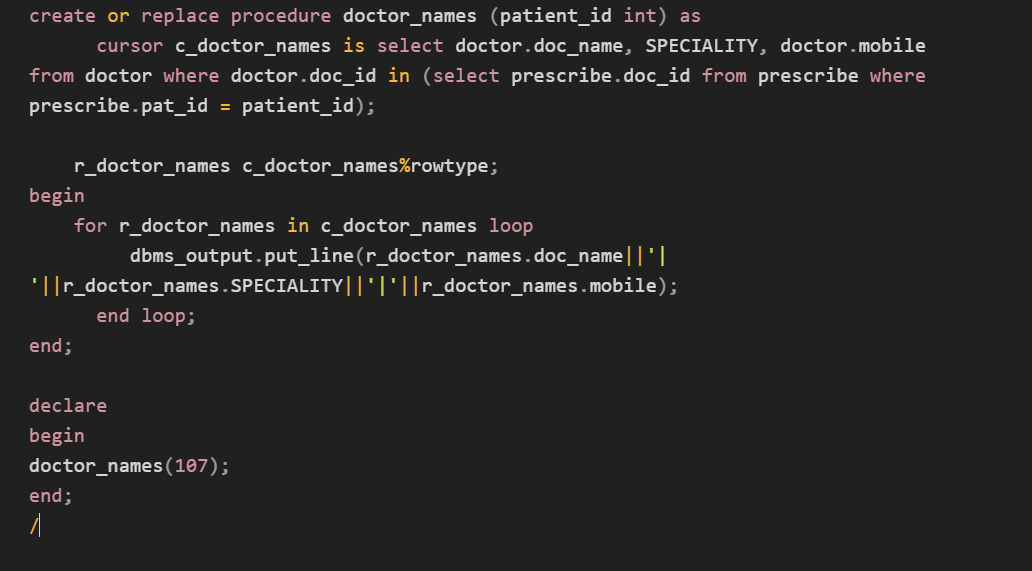
**5. Med\_details**

The following procedure extracts the attributes of the required tuples which is given by the input of a specific item name.

****

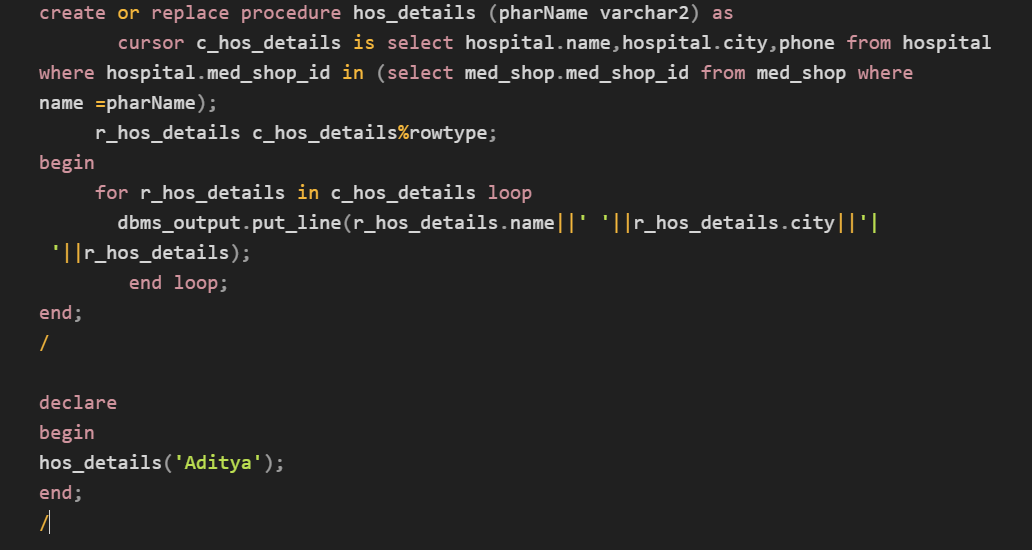
**6. Doctor\_names**

The following procedure provides us with all the names of doctors. The procedure helps us identify that a patient is being treated by which particular doctor. It takes input values as patient id in order to display the name of the corresponding doctor.



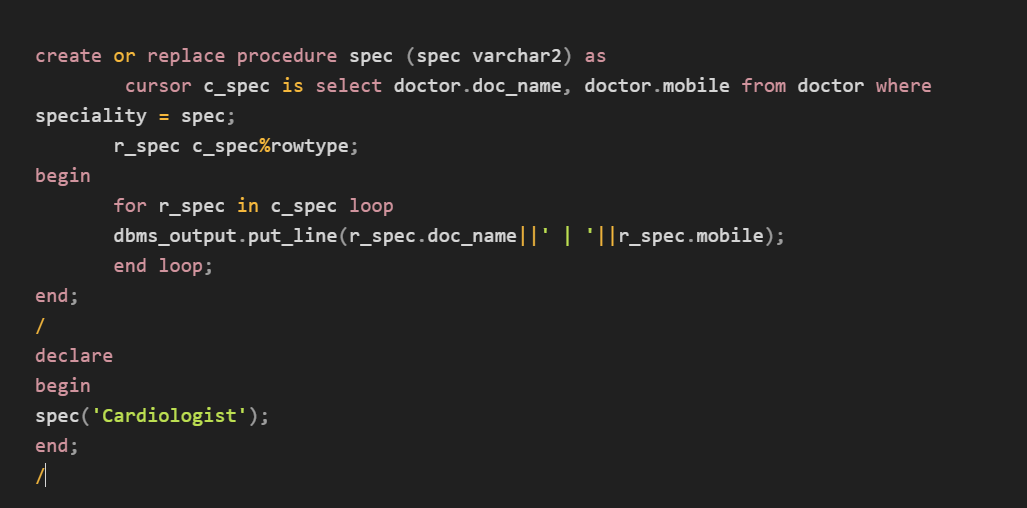
**7. Hos\_details**

The following procedure detects the address of a particular hospital from the given inputs and extracts the attributes and details of that hospital. The procedure only extracts those details which are selected by the user.



**8. Specification of Doctor**

The following procedure shows us the doctor by its specification.



# 

# 

# 

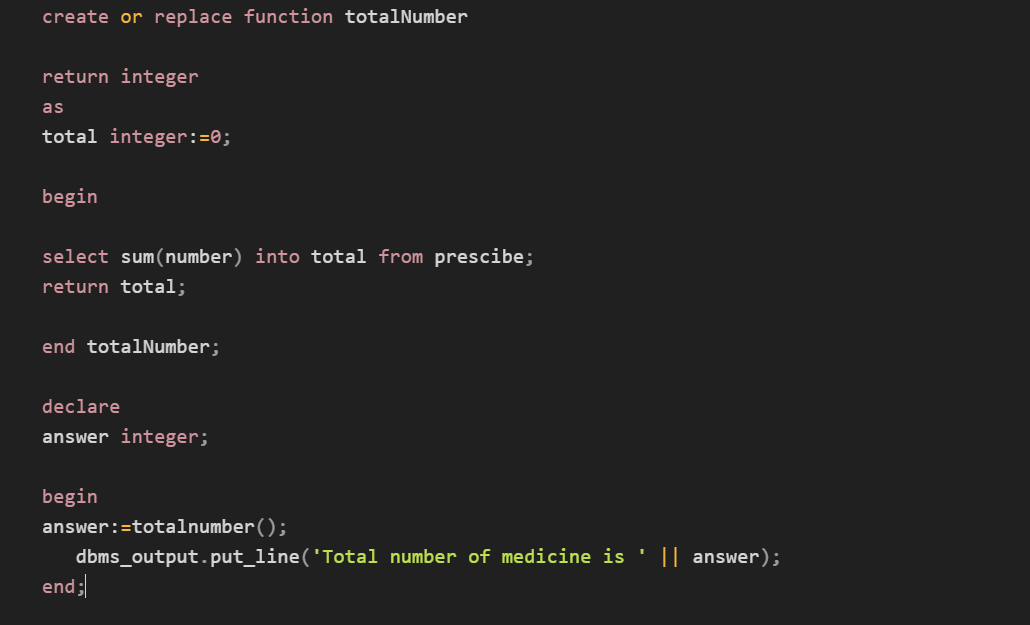
# 

# 

# Functions

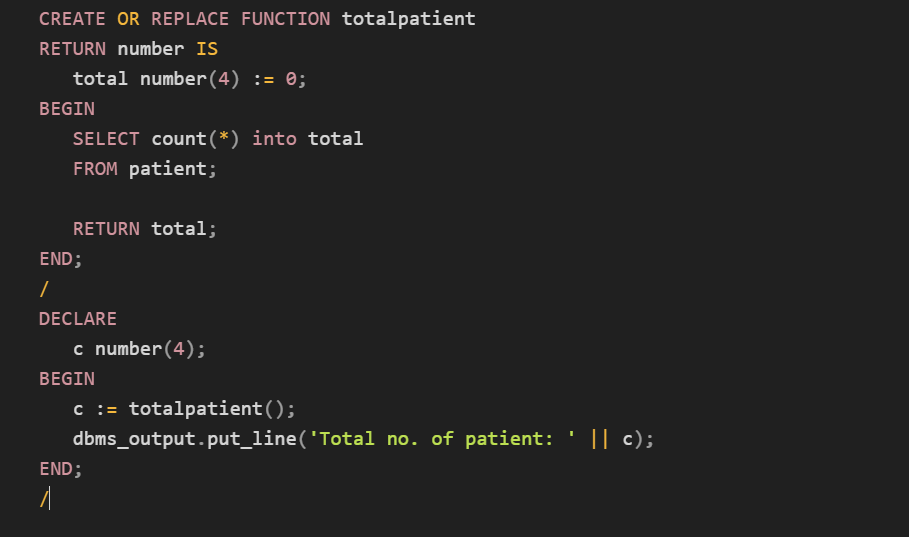
1. **Total\_number**

The mentioned function in our database is used to fetch the total number of medicines available in the medicine shop. It helps the chemist with managing his/her stock efficiently. The input value provided will be the item name of its id.



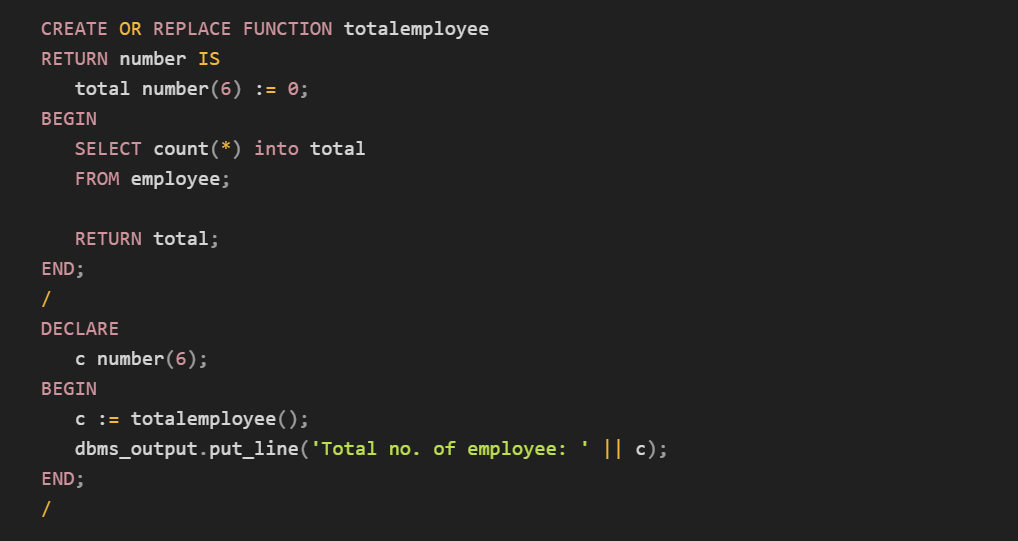
1. **Total\_patient**

The ‘Total\_patient function is used to display the number of patients being treated by a particular doctor or total number of patients in a particular hospital provided doctor name or the hospital id.



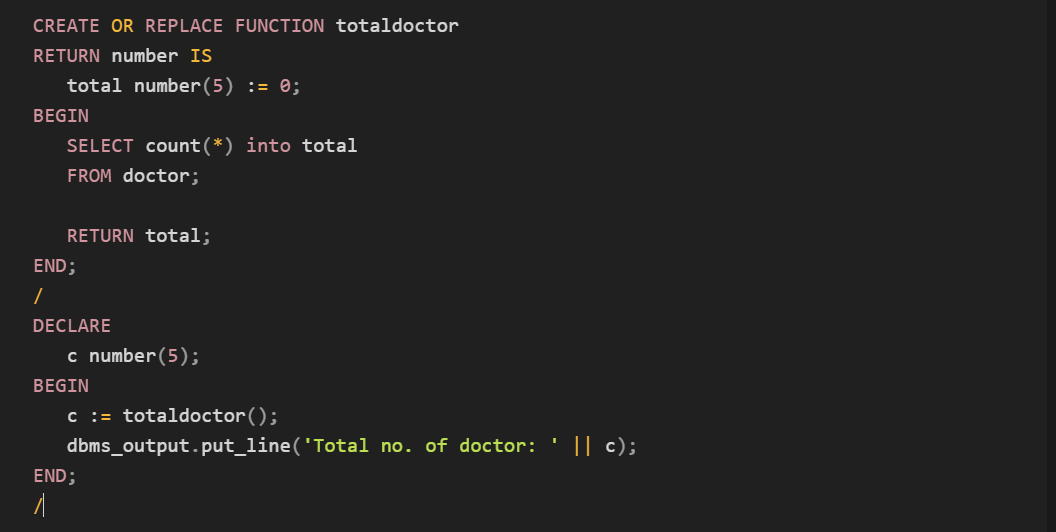
**3. Total\_employee**

The Total\_employee function helps us fetch the data of the number of employees working in a medical shop. It provides us with the data of the number of employees given the medical shop id.



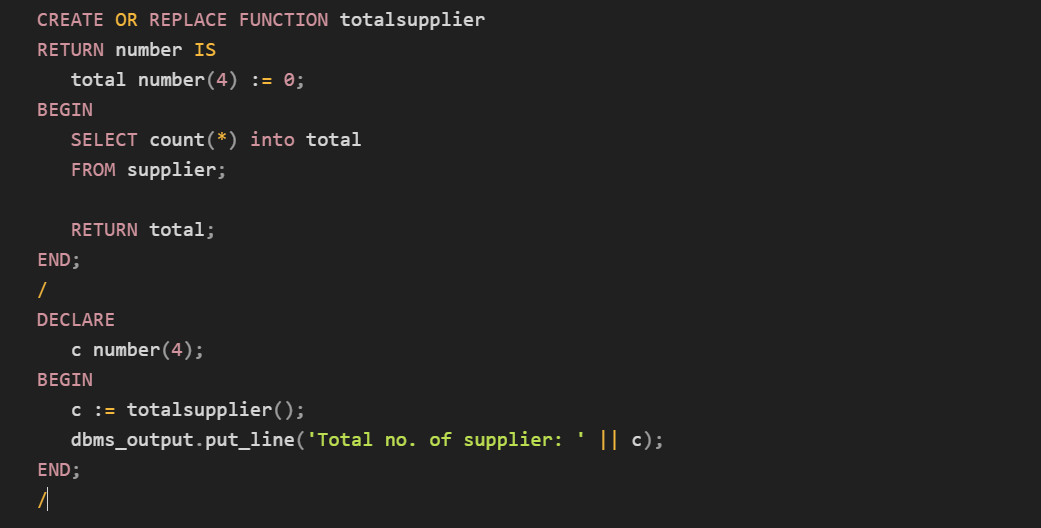
**4. Total\_doctor**

The below mentioned function helps us determine the total number of doctors treating a particular patient or total number of patients working in a hospital given the name of that hospital or the patient name.



**5. Total\_supplier**

The below mentioned function is used to determine the total number of suppliers of a given item or a total number of suppliers supplying items to a given medical shop. It takes input as an item name.



# 

# 

# 

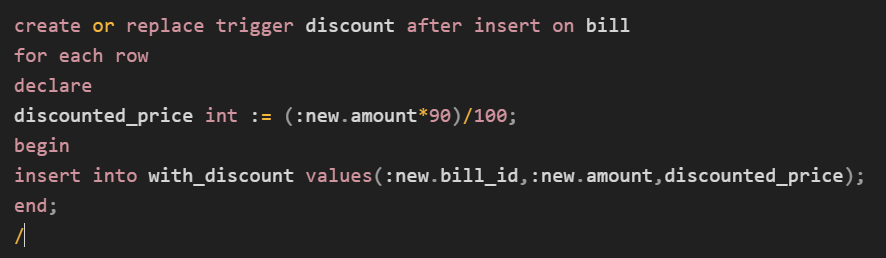
# 

# 

# Triggers

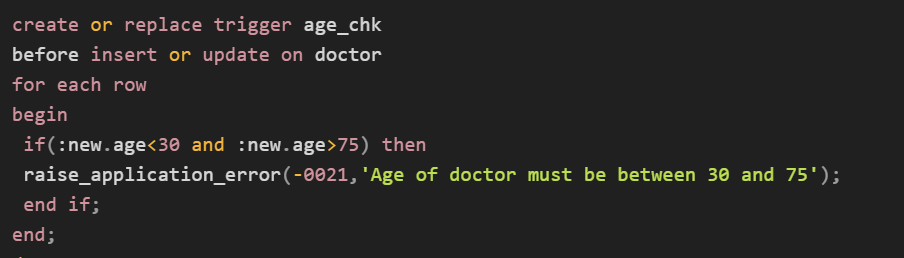
**1) Display\_discount**

The following trigger is used to display the discounted bill amount after a purchase. The cause of the trigger is generating the bill. It takes bill amount and discount value as input and provides the discount amount.



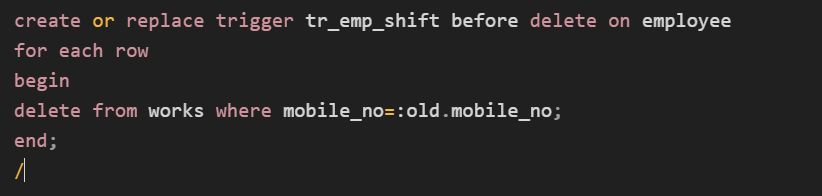
**2) Age\_check**

This trigger is involved in checking and verifying the age of a particular doctor and decides whether a doctor is eligible in the given criteria or not. It takes age as input and shows the eligibility of a doctor as output. The criteria used is a doctor should be older than 30 but his age should not exceed 75.



**3 )tr\_emp\_shift**

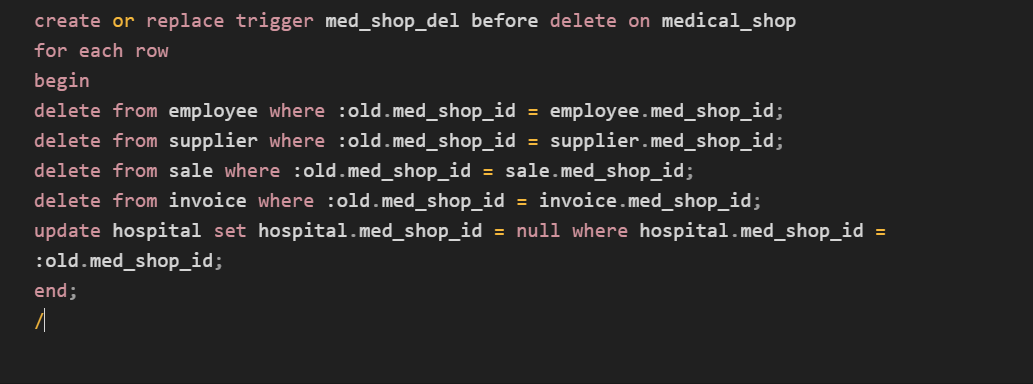
The following trigger function deletes the older number of an employee. It accesses the data of the age of every employee.



# 

**4) Med\_shop\_del**

The following trigger is useful in restricting the age limit of the medical shop. If a medical shop in a database is at any point found to be disobeying the age criteria the trigger removes that medical shop from the database.



**5) Display\_salary\_changes**

The following trigger is used to generate the changes in the salary of an employee. It uses the previous value of the salary and compares it with the new salary and displays the difference in the salaries.

