# HOSPITAL MANAGEMENT SYSTEM

**A PROJECT REPORT**

***Submitted by***

## ABHINAY KUMAR YADAV[RA2011047010116]

## HIMANSHU KUMAR SINGH [ RA2011047010137 ]

## ABHINAV KM [RA2011047010150]

*Under the guidance of*

## Dr. R. BEAULAH JEYAVATHANA

(Assistant Professor, Department of Computational Intelligence)

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

in

## COMPUTATIONAL INTELLIGENCE

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**



### S.R.M. Nagar, Kattankulathur, Kancheepuram District

**APRIL 2022**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that this project report titled “**HOSPITAL MANAGEMENT SYSTEM**” is the bonafide work of “**HIMANSHU KUMARSINGH [RA2011047010137** ]”, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

#### SIGNATURE

Dr. R. BEAULAH JEYAVATHANA

#### GUIDE

Assistant Professor

Dept. of Computational Intelligence

Signature of the Internal Examiner

#### SIGNATURE

Dr. ANNIE UTHRA

#### HEAD OF THE DEPARTMENT

Dept. of Computational Intelligence

Signature of the External Examiner

# ACKNOWLEDGEMENTS

### I would like to express my deepest gratitude to my guide, Dr. R. BEAULAH JEYAVATHANA, her valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing the project. All through the work, in spite of her busy schedule, she has ex- tended cheerful and cordial support to me for completing this project work.

#### HIMANSHU KUMAR SINGH

iii

# CHAPTER 1

**INTRODUCTION ABOUT THE PLATFORMS WORKED**

# MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

* + - MySQL is released under an open-source license. So you have nothing to pay to use it.
    - MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
    - MySQL uses a standard form of the well-known SQL data language.
    - MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
    - MySQL works very quickly and works well even with large data sets.
    - MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
    - MySQL is very friendly to PHP, the most appreciated language for web development.
    - MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

# 

# ABSTRACT

This project Hospital 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 clinical details of every patient and hospital tests done automatically. It includes a search facility to know the current status of each patient. User can search details of a patient using the id. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. 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.

# INTRODUCTION

The project Hospital 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.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. 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.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital 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.

Hospital 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. Hospital 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

**1.2 Problem Introduction:**

**Lack of immediate retrievals: -**

The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient’s history, the user has to go through various registers. This results in in convenienceand wastage of time.

**Lack of immediate information storage: -**

The information generated by various transactions takes time and efforts to be stored at right place.

**Lack of prompt updating: -**

Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

**Error prone manual calculation: -**

Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient’s bill based on various treatments.

**Preparation of accurate and prompt reports: -**

This becomes a difficult task as information is difficult to collect from various register.

**Objective:-**

1. Define hospital
2. Recording information about the Patients that come.
3. Generating bills.
4. Recording information related to diagnosis given to Patients.
5. Keeping record of the Immunization provided to children/patients.
6. Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in a Hospital by the operational staff andDoctors. All these works are done on papers.

**Scope of the Project:-**

1. Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly.
2. Bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they all are summed up.
3. Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.
4. Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.
5. Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can’t remember them at that time.

# DESIGN

**UML Design**

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language , which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

* Visualizing
* Specifying
* Constructing
* Documenting

**Visualizing**

Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

**Specifying**

Specifying means building, models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

**Constructing**

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

**Documenting**

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes releasers, etc...

**4.2 UML Approach**

**UML Diagram**

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs . you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams , or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

* 1. Class diagram
  2. Object diagram
  3. Use case diagram
  4. Sequence diagram
  5. Collaboration diagram
  6. State chart diagram
  7. Activity diagram
  8. Component diagram
  9. Deployment diagram

**USE CASE DIAGRAM:**

A usecase diagram in the Unified Modeling Language(UML) is atype of behavioral diagram defined by and created from a use-case analysis.its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use cases),and any dependencies between those use cases.

Use case diagrams are formally included in two modeling languages defined by the OMG:theunfied modeling language(UML) and the systems modeling language(sysML)

**Use case diagram of our project:**

****

**Class Diagram**:

A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams provides the representation that developers work from. Class diagrams help on the analysis side, too.



**Sequence diagram:**

A **Sequence Diagram** is an interaction diagram that emphasis the time ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.

****

**Collaboration diagram:**

A **Collaboration Diagram** also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects. The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

****

**Deployement diagram:**

A **Deployment Diagram** shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams address the static deployment view of architecture. They are related to component diagrams in that a node typically encloses one or more components.

****

**Statechart Diagrams:**

The state diagram shows the states of an object and represents activities as arrows connecting the states. The Activity Diagram highlights the activities. Each activity is represented by a rounded rectangle-narrower and more oval-shaped than the state icon. An arrow represents the transition from the one activity to the next. The activity diagram has a starting point represented by filled-in circle, and an end point represented by bulls eye.

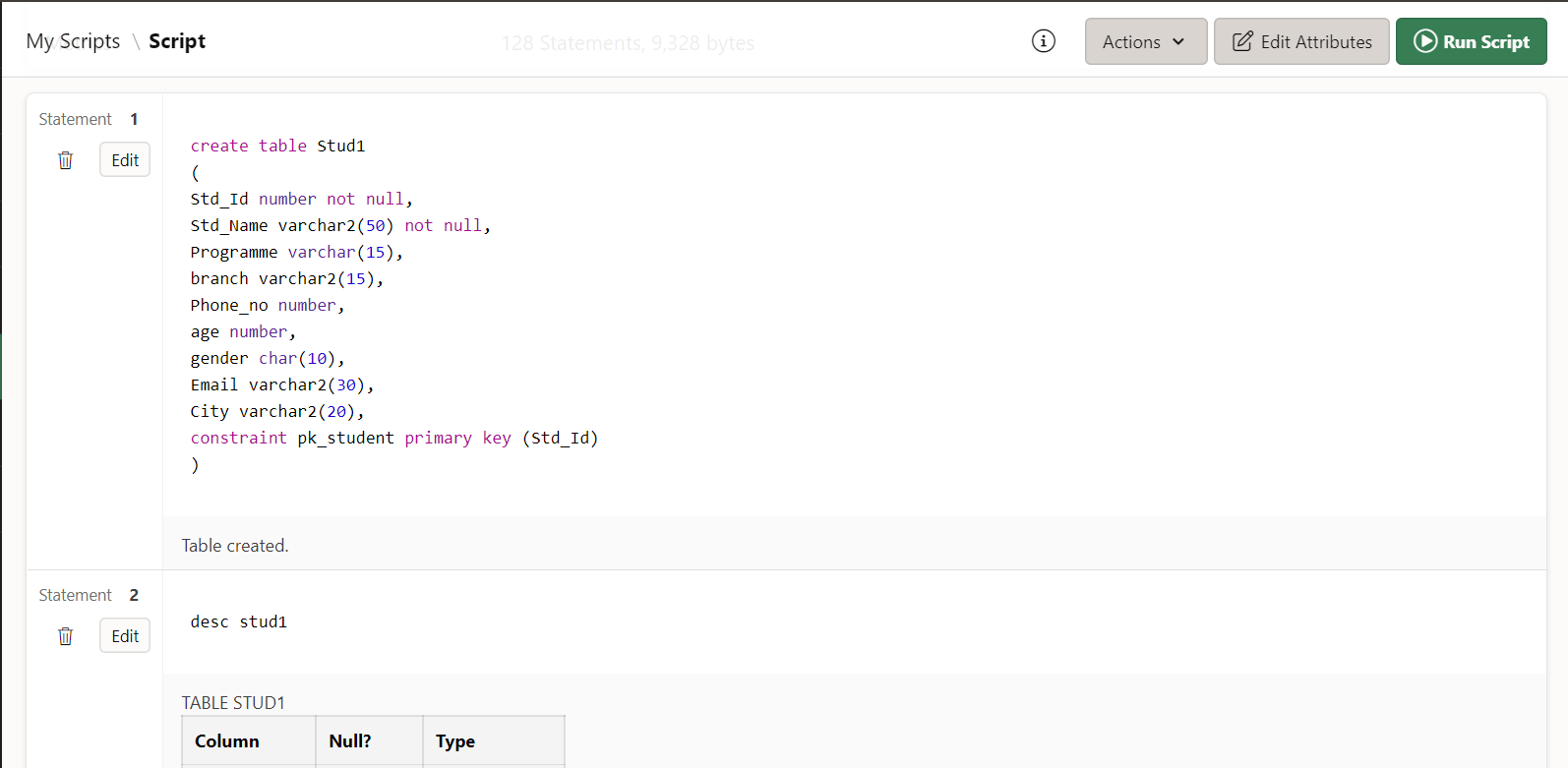


# IMPLEMENTATION

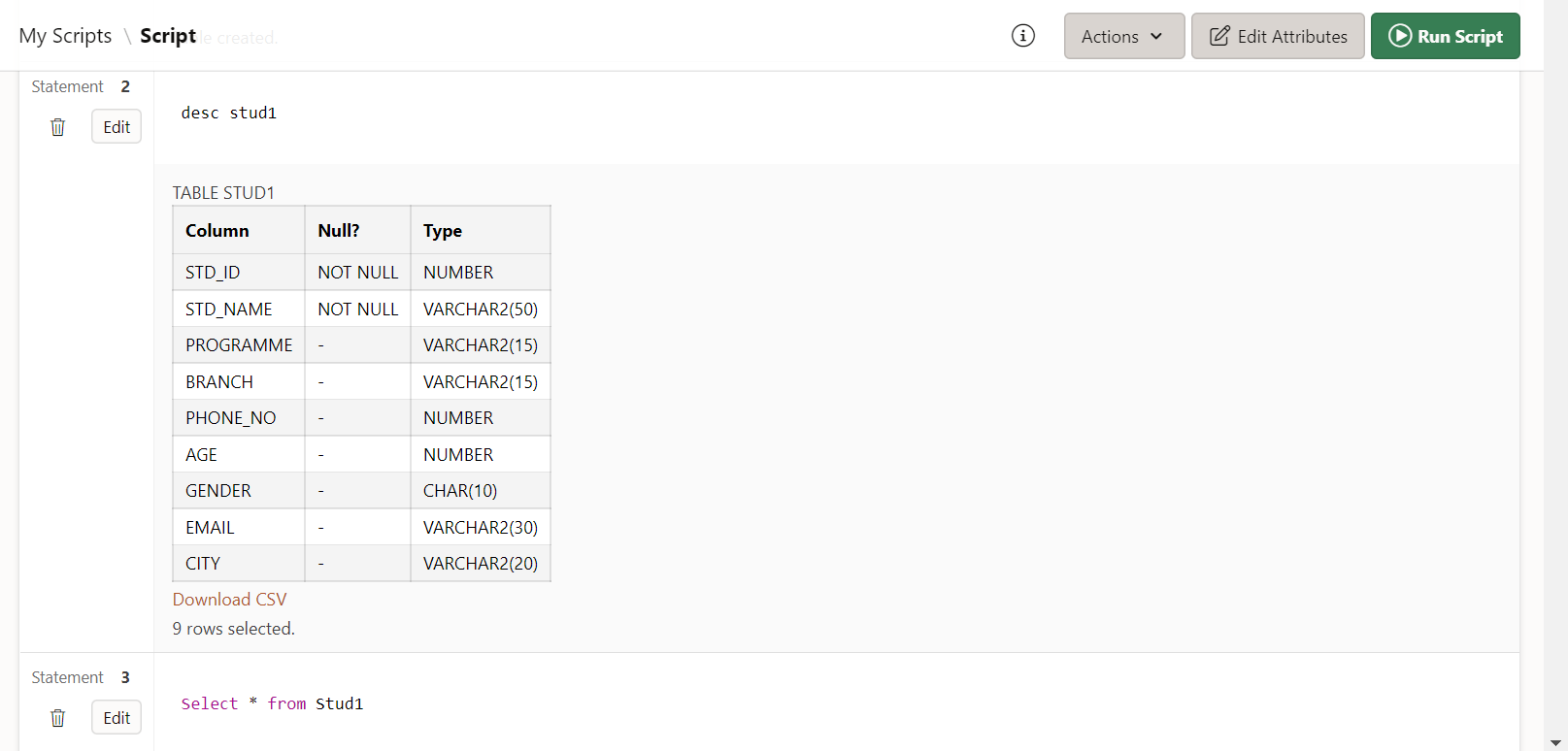
**Executing basic commands of sql**

**Aim -** To create a Table and execute the basic command in SQL.

1. creating table:

****

**Desc command**

****

**Inserting values in table**

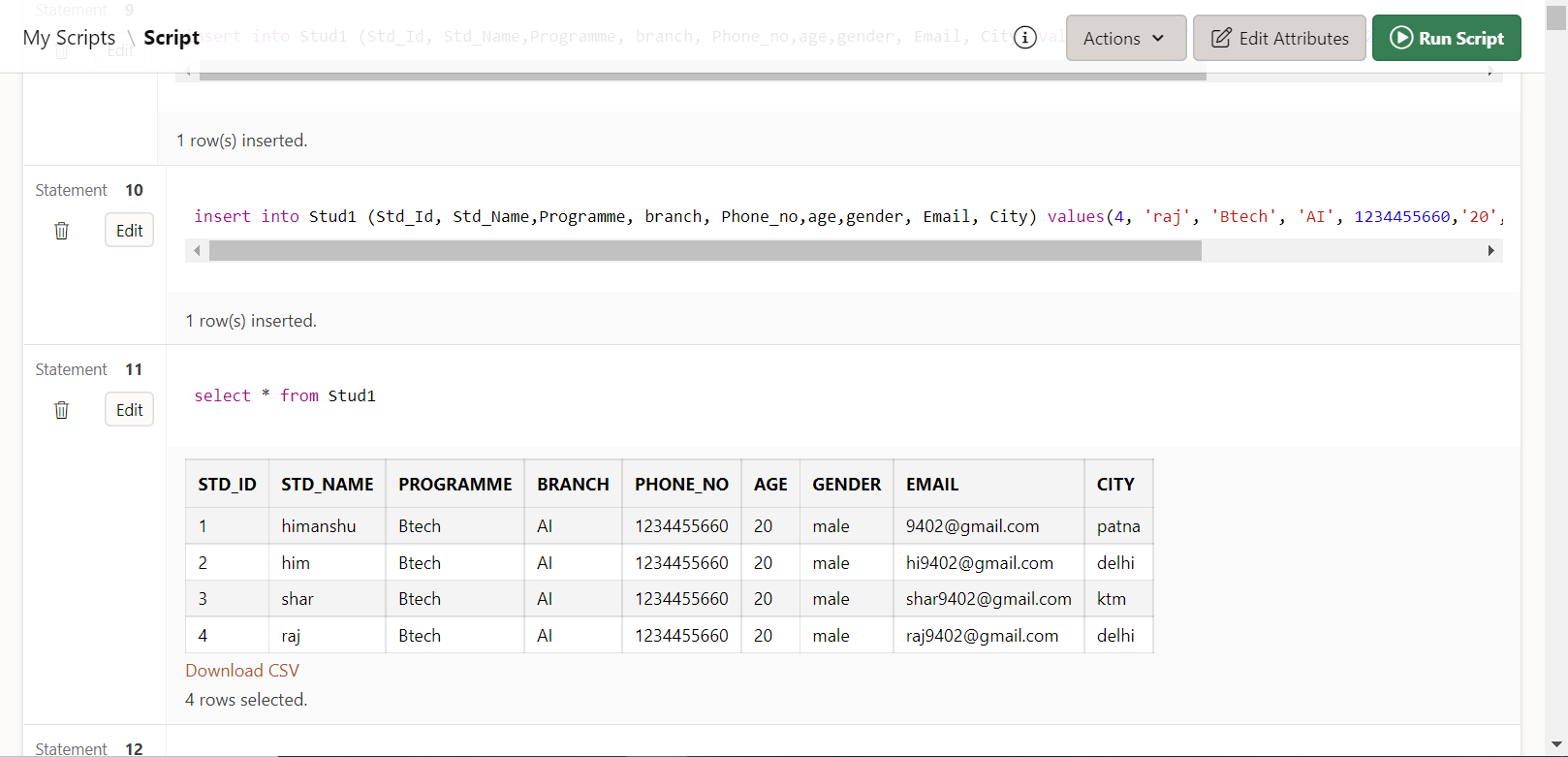
2.)INSERT - It is used to insert data into the row of a table.

SYNTAX:- INSERT INTO TABLE\_NAME(col1, col2, col3,.... col N) VALUES (value1, value2, value3, .... valueN);

EXAMPLE:-

insert into DiabetesData (Patient\_Id,Patient\_name,Pregnancies,Insulin,Blood\_Pressure,BMI,Glucose,Skin\_Thickne ss,Age,Outcome) values (1,'A',1,123,45,48,234,5,67,1);

insert into DiabetesData (Patient\_Id,Patient\_name,Pregnancies,Insulin,Blood\_Pressure,BMI,Glucose,Skin\_Thickne ss,Age,Outcome) values (2,'B',6,137,47,45,287,7,56,0)

****

**Updating a column**

**Deleting column**

**2. Alter**

It is used to alter the structure of the table.

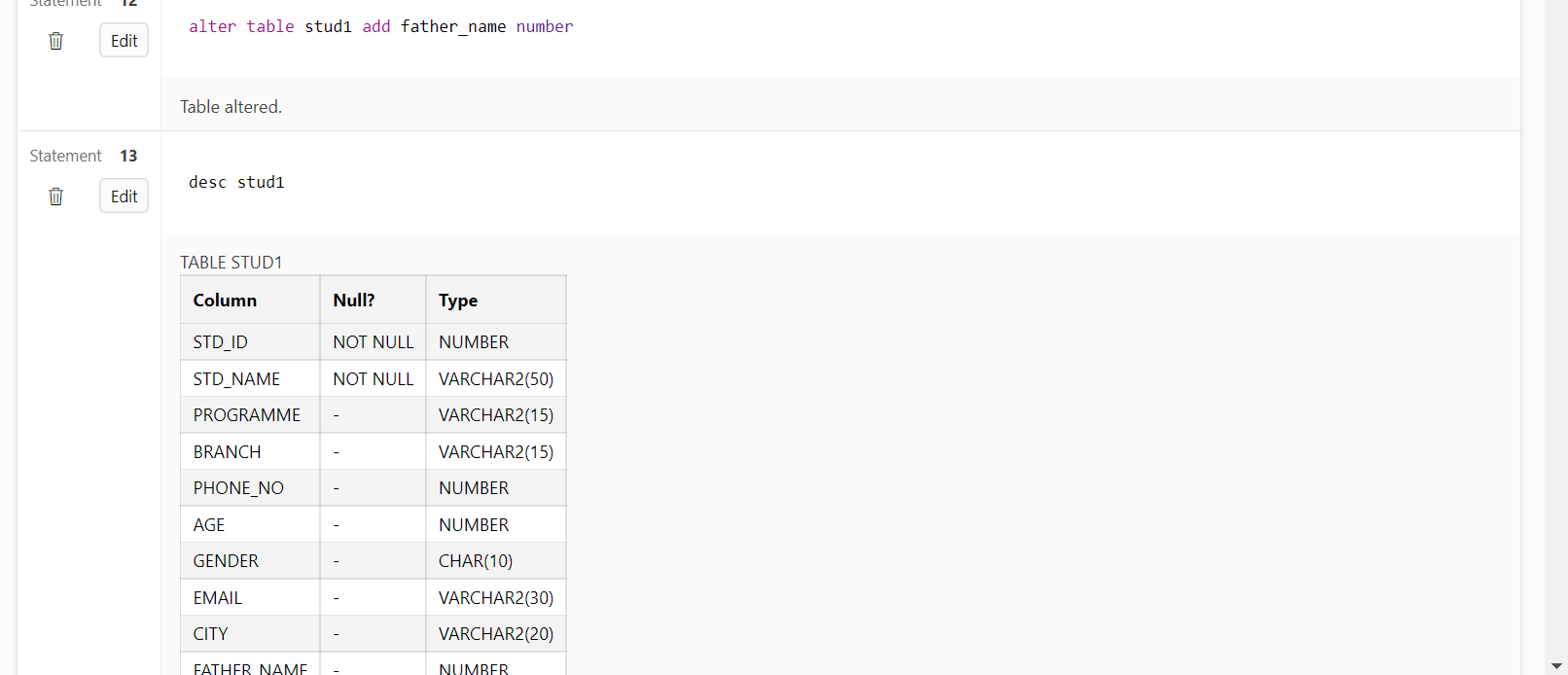
a.)ADD:-

To add a new column in the table

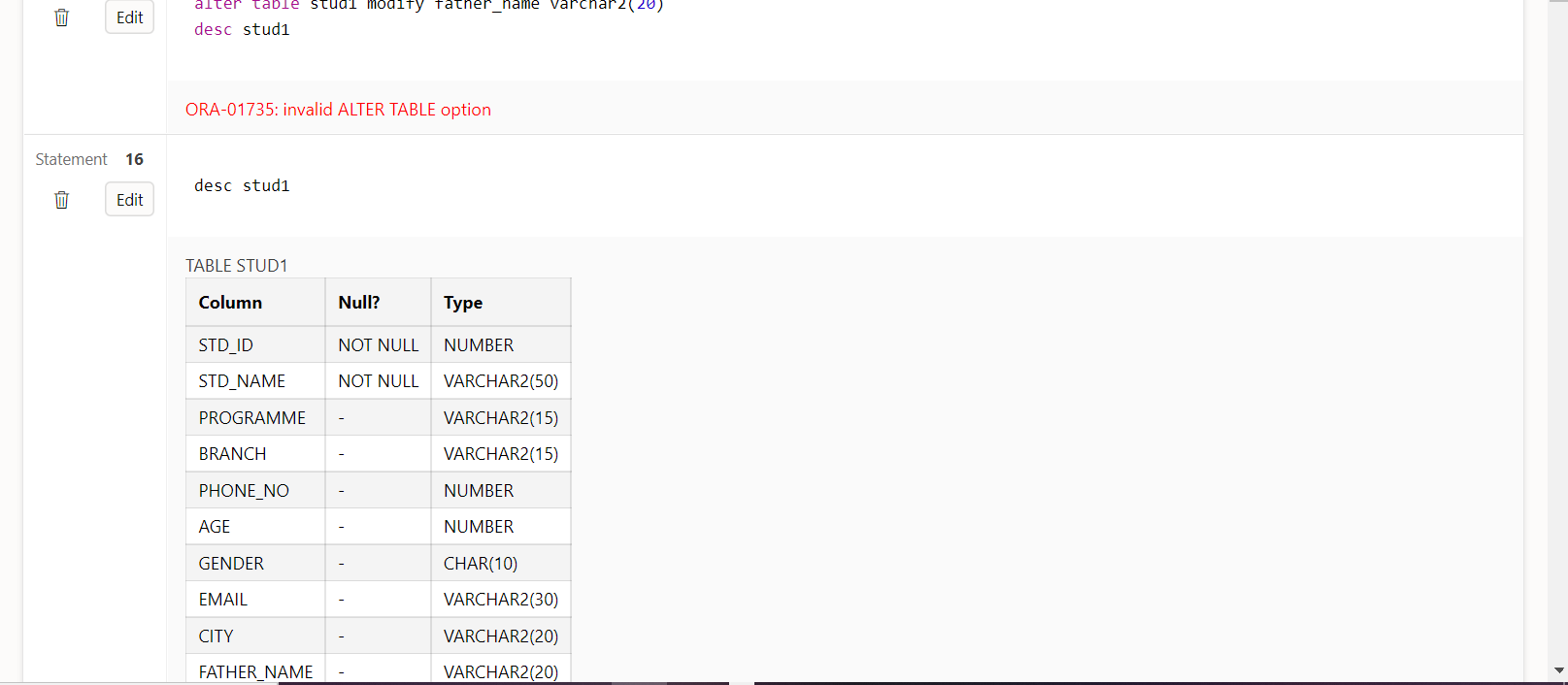
SYNTAX:- ALTER TABLE table\_name ADD column\_name COLUMN-definition;

EXAMPLE:- alter table DiabetesData add (CVD number(1))

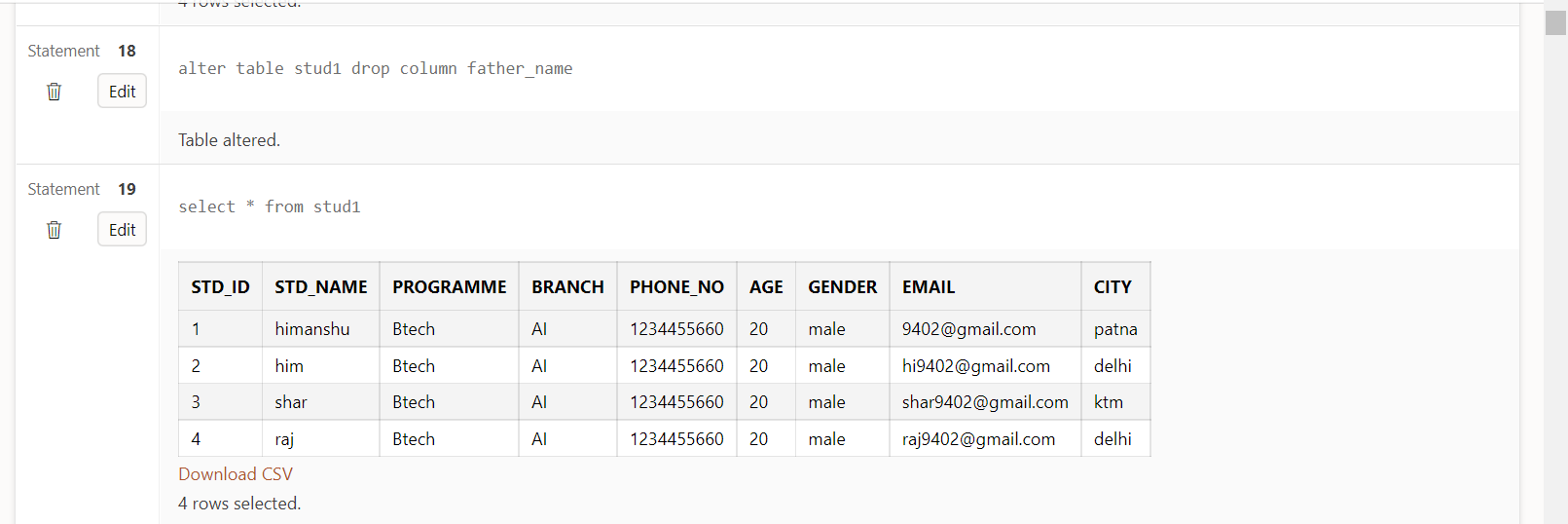
**Add column**

****

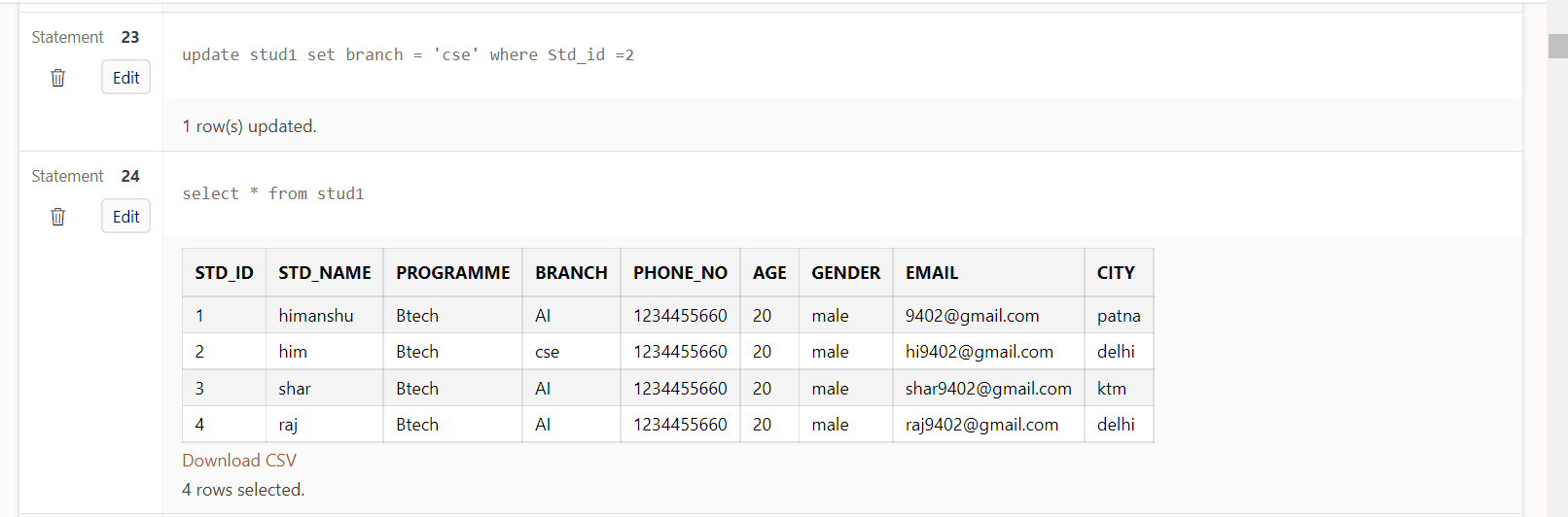
**Updating a column**

****

**Deleting column**

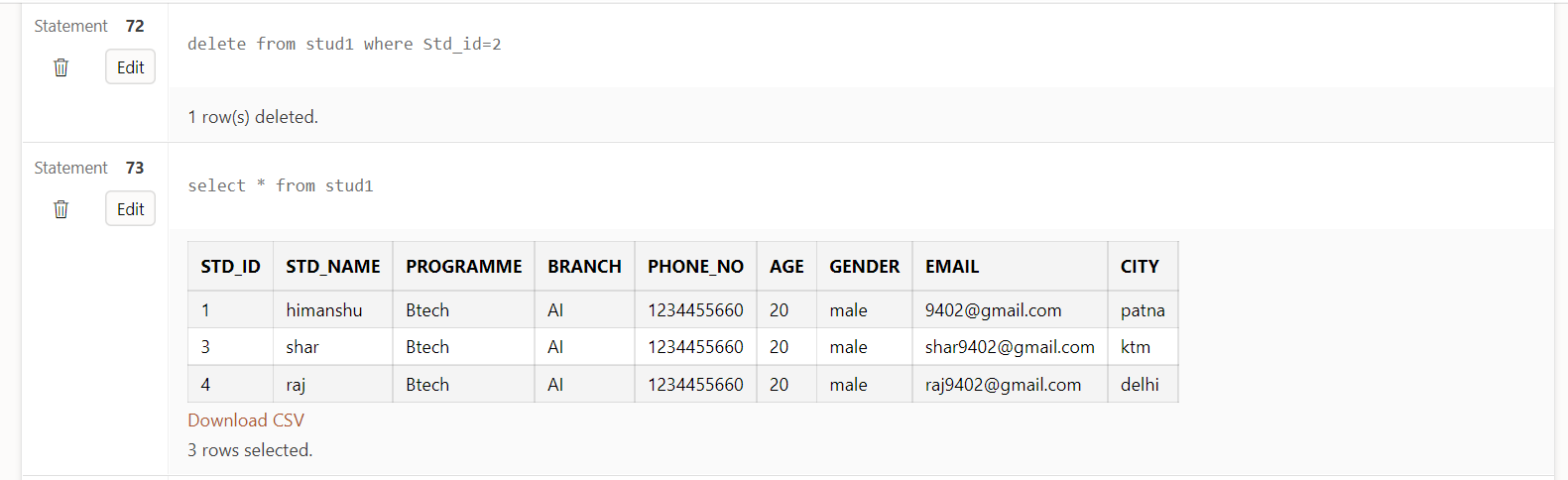
****

**Update command**

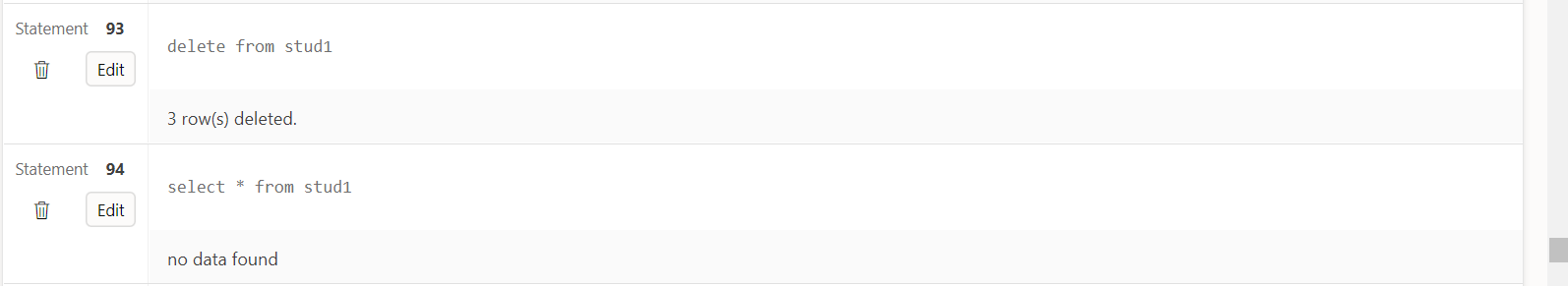
****

**Deleting a column**

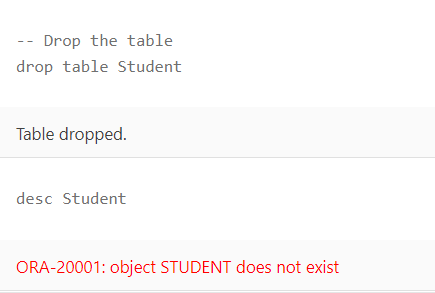
**Deleting 1 column**

****

**Deleting table**

****

**Drop Table**



**Result :**  All the basic commands of SQL have been executed successfully.

**Create a table and insert DOB in the table**

**Aim :-** To insert DOB in the table

****

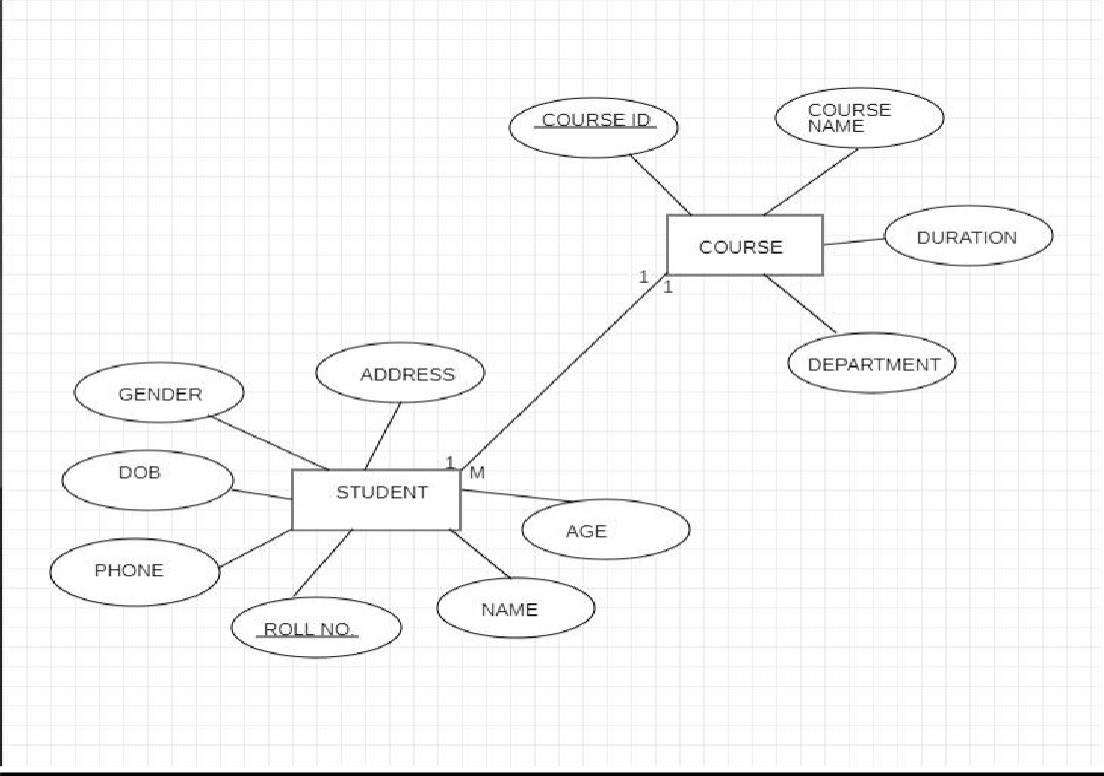
**AIM-** To make the entity relationship diagram for the student database.

**THEORY-**

* Entity- Simple rectangular box represents an Entity
  1. Weak Entity- A weak Entity is represented using double rectangular boxes.
* Attributes for any Entity- Ellipse is used to represent attributes of any entity.
  1. Key Attribute- To represent a Key attribute, the attribute name inside the Ellipse is underlined.
  2. Composite Attribute- A composite attribute is the attribute, which also has attributes.
  3. Derived Attribute - Derived attributes are those which are derived based on other attributes,
  4. Multivalued Attribute -Double Ellipse, one inside another, represents the attribute which can have multiple values.

 Relationships- Binary Relationship means relation between two Entities. This is further divided into four types-

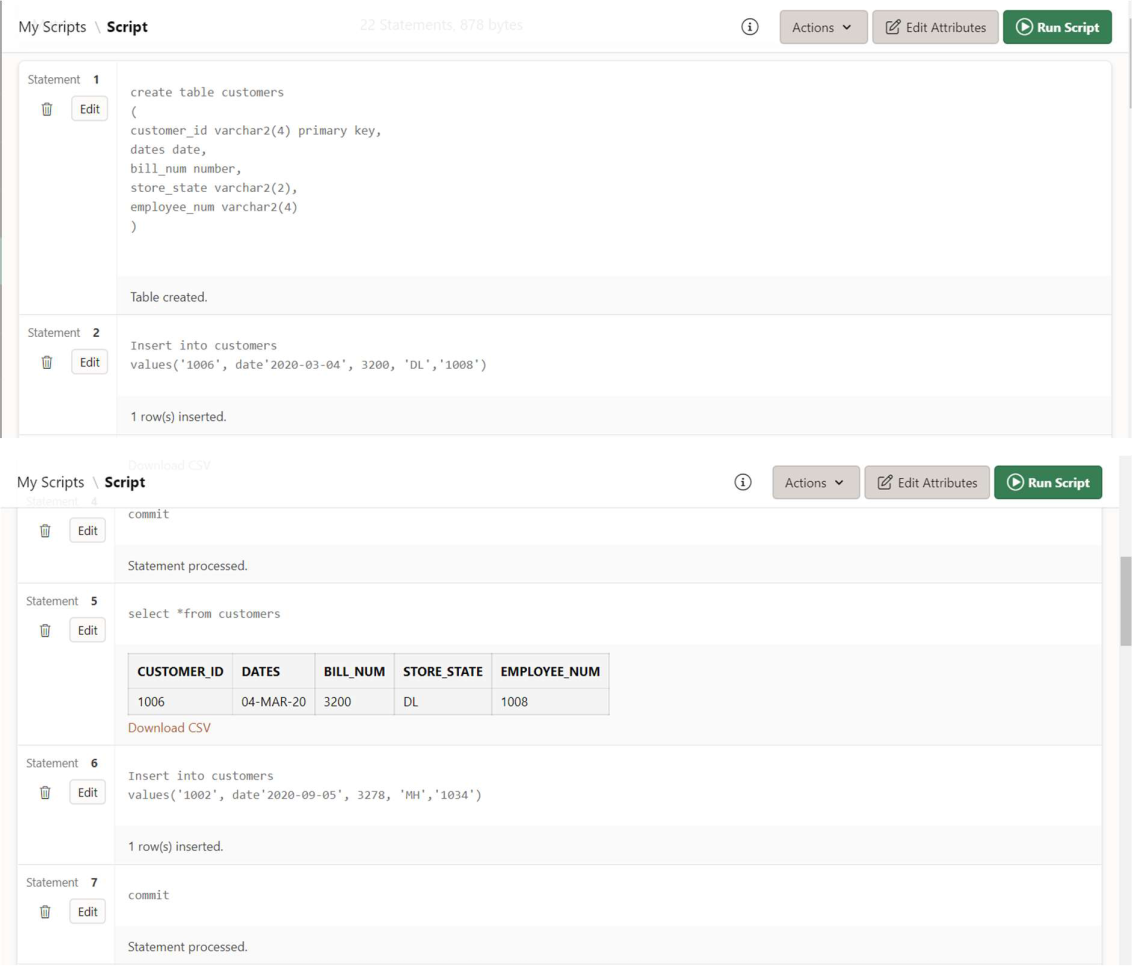
1. One to One Relationship
2. One to Many Relationship
3. Many to One Relationship
4. Many to Many Relationship **OUTPUT-**



**AIM: Execution of Transaction Control Language (TCL) Commands**

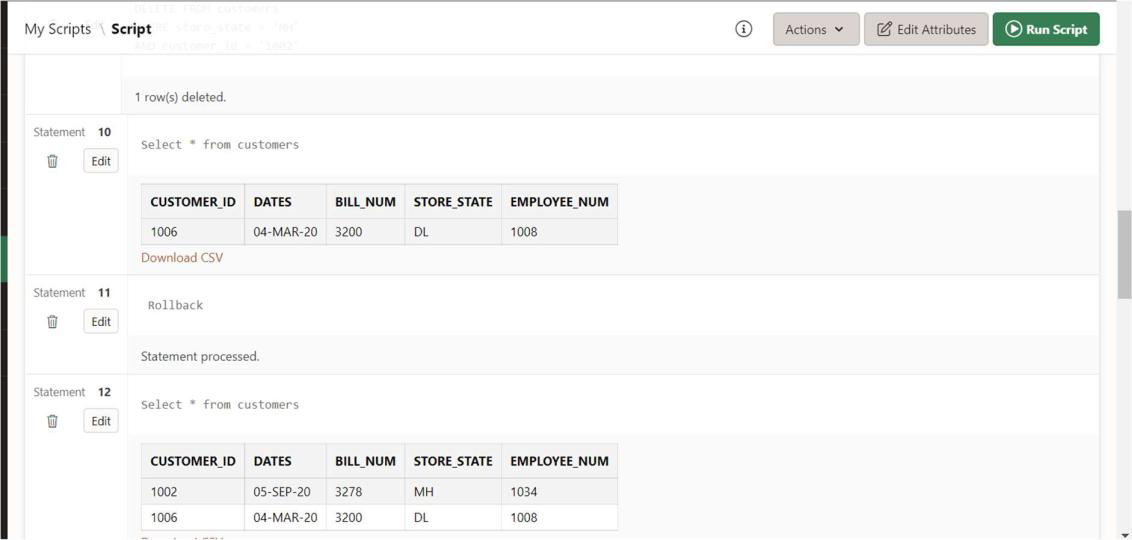
**COMMIT: Commits a Transaction.**

**Syntax:COMMIT;**



**ROLLBACK: Rollbacks a transaction in case of any error occurs.**

**Syntax:Rollback**



**SAVEPOINT:Sets a savepoint within a transaction.**

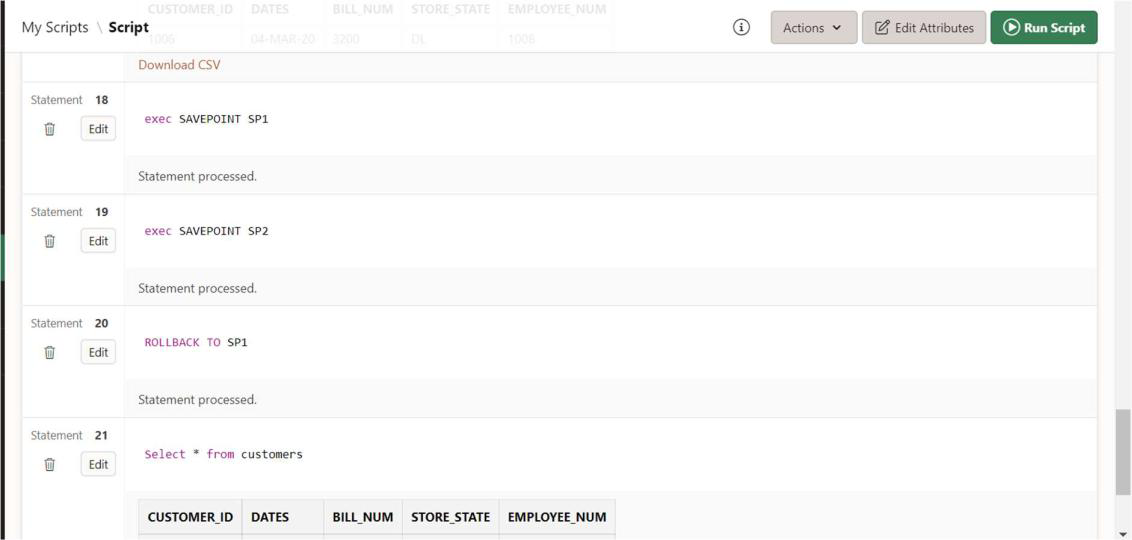
**Syntax:SAVEPOINT SAVEPOINT\_NAME;**

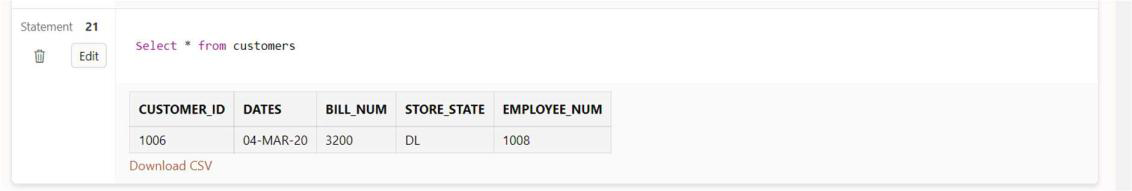
**This command is used only in the creation of SAVEPOINT among all the transactions.**

**In general ROLLBACK is used to undo a group of transactions.**

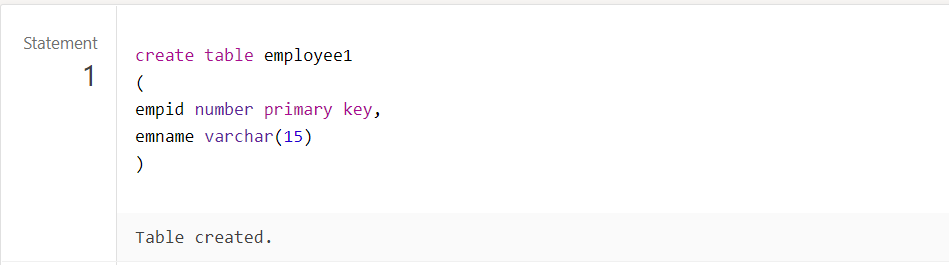
**Syntax for rolling back to Savepoint command:**

**ROLLBACK TO SAVEPOINT\_NAME;**

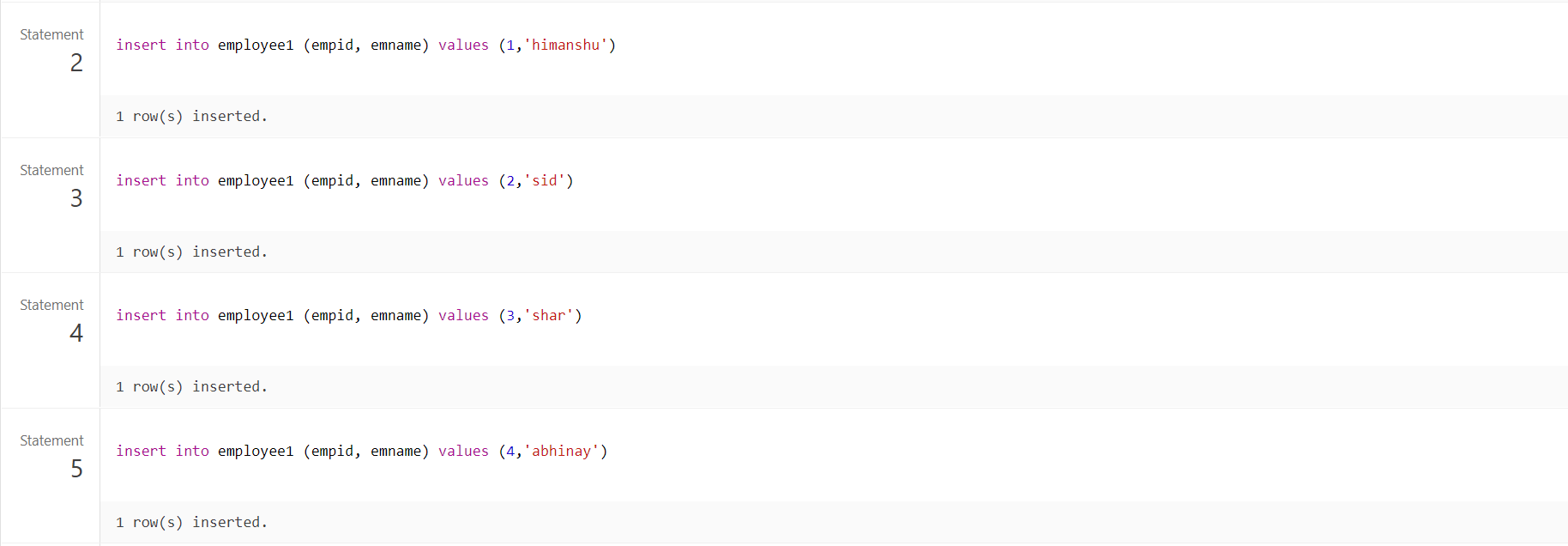




**Aim : execution of inbuilt functions**



Inserting values



Count (emname)

Count (\*)

Max

Min

Avg

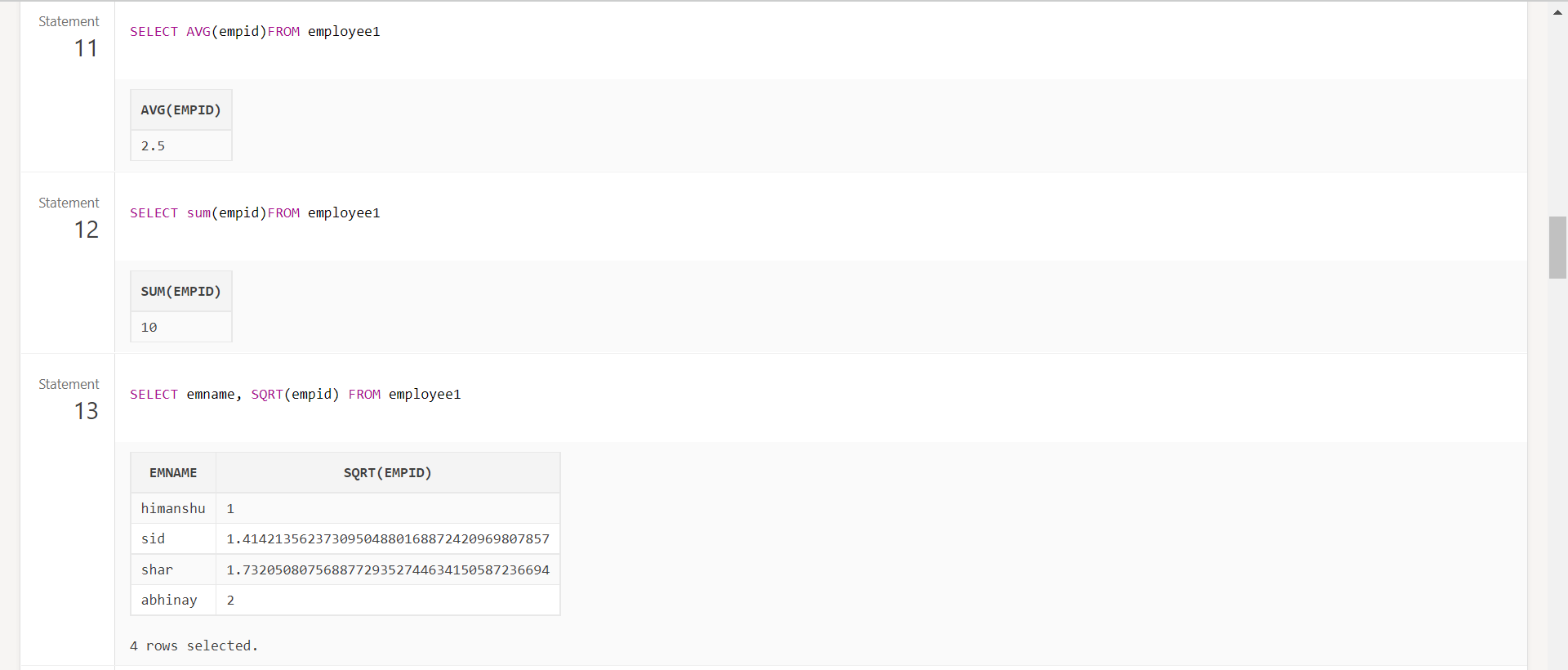
Sum

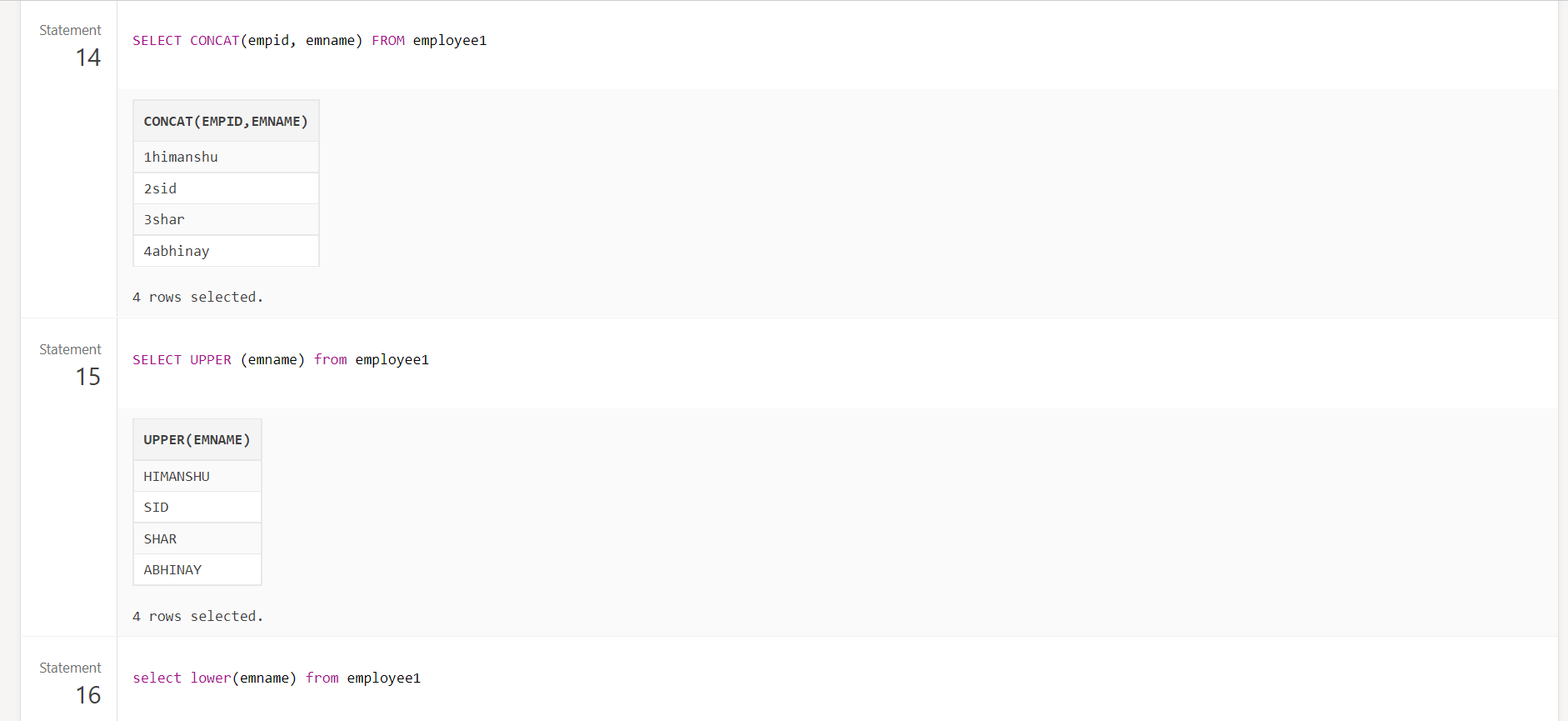
Sqrt

Concat

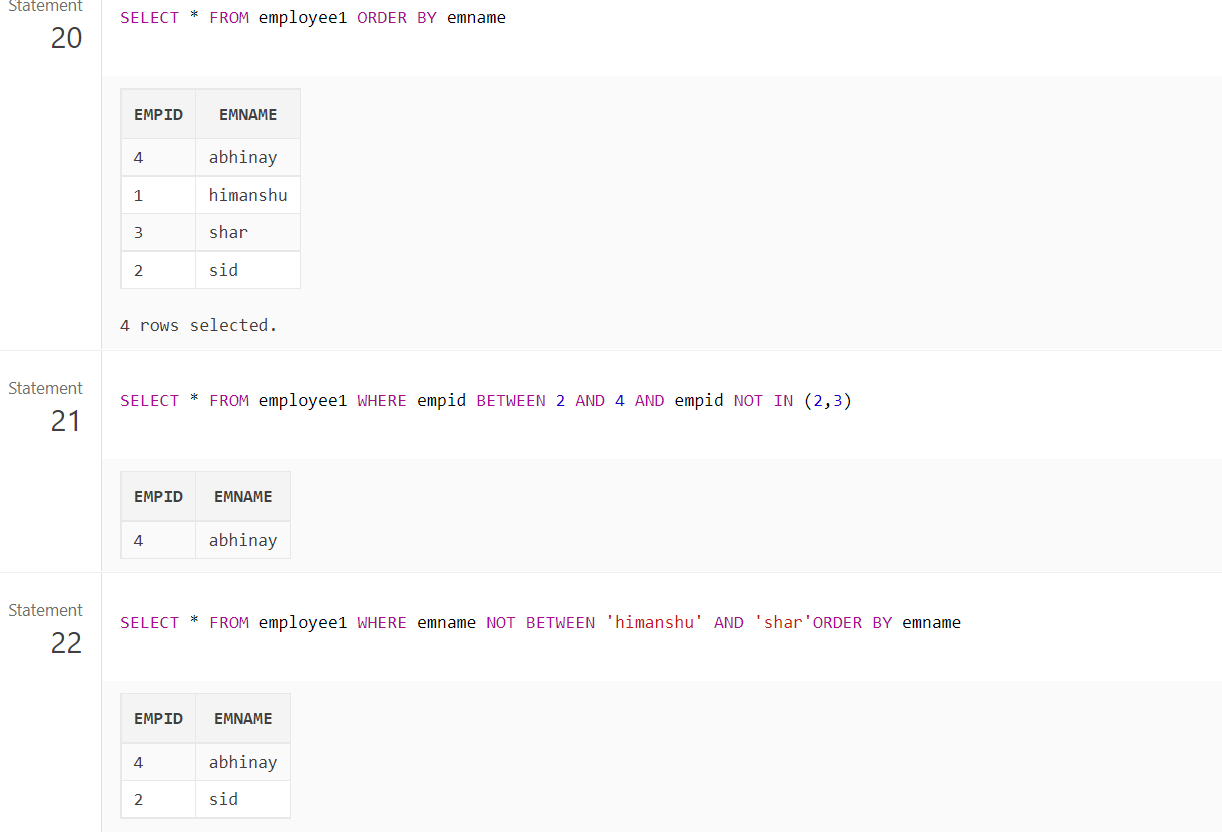
Upper lower













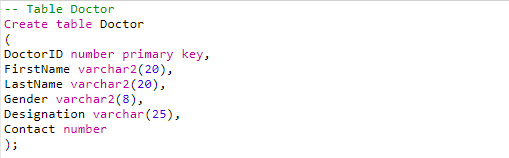


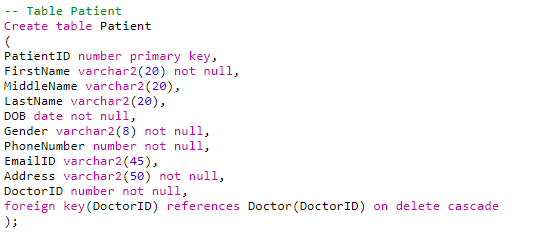
**Execute Basic Command of SQL**

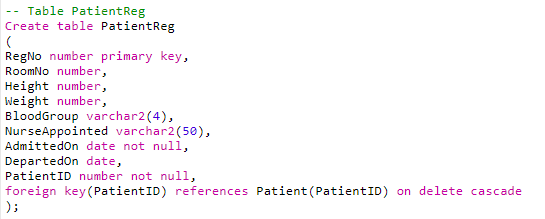
**Aim :** To execute the basic command of SQL

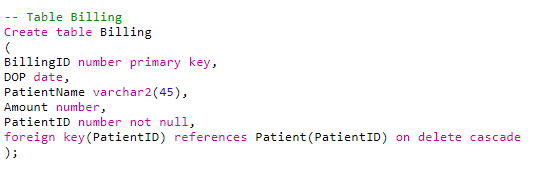
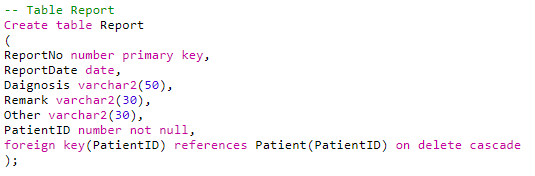
**Command:**

* **Create Table**

****

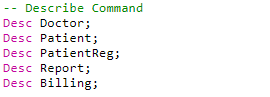
****

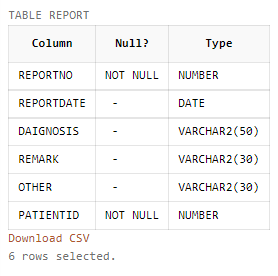
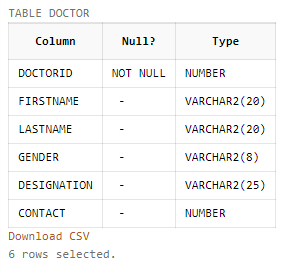
****

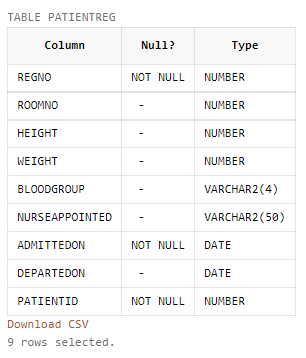
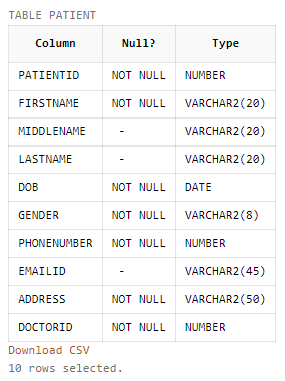
****

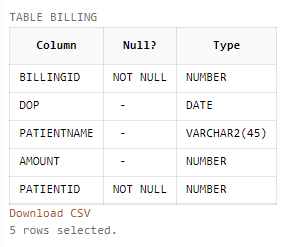
****

* **Describe Table**

****

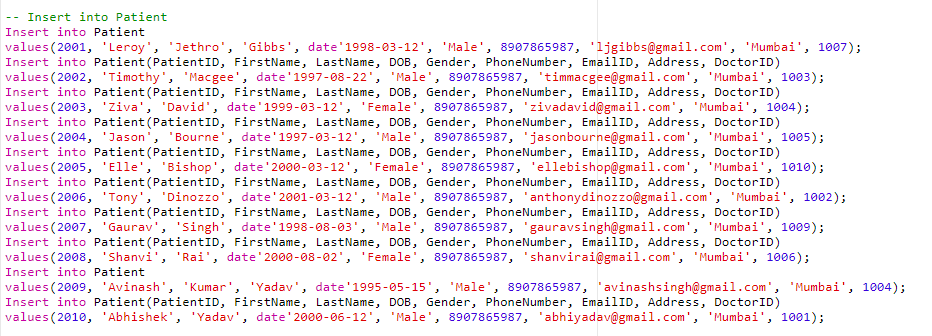
****

****

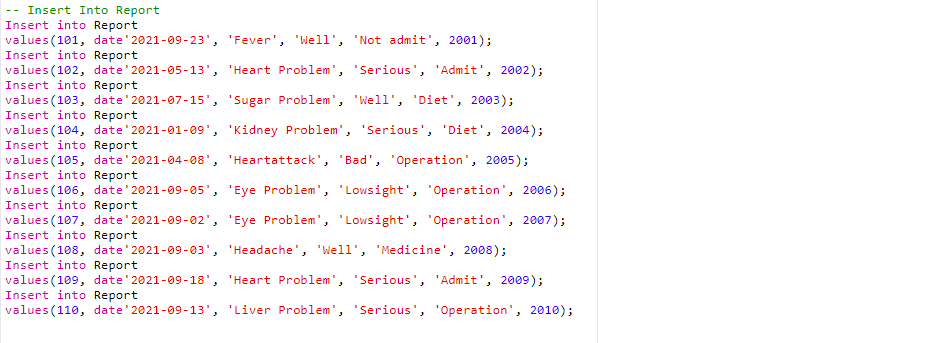
****

* **Insert into Tables**

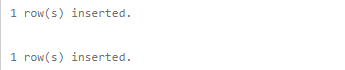
****

****

****

****

****

****

**.**

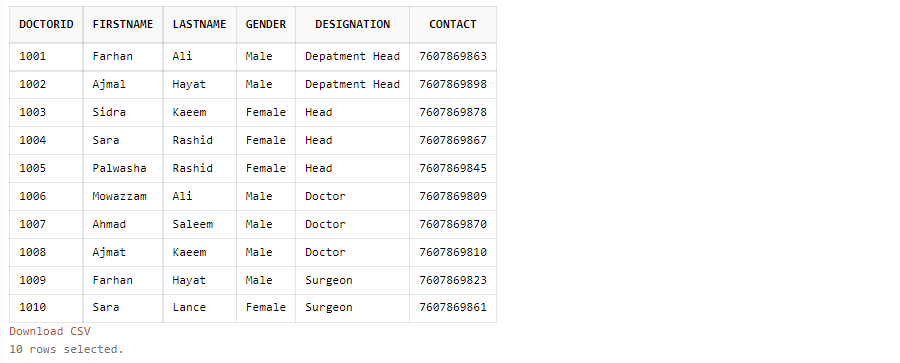
**.**

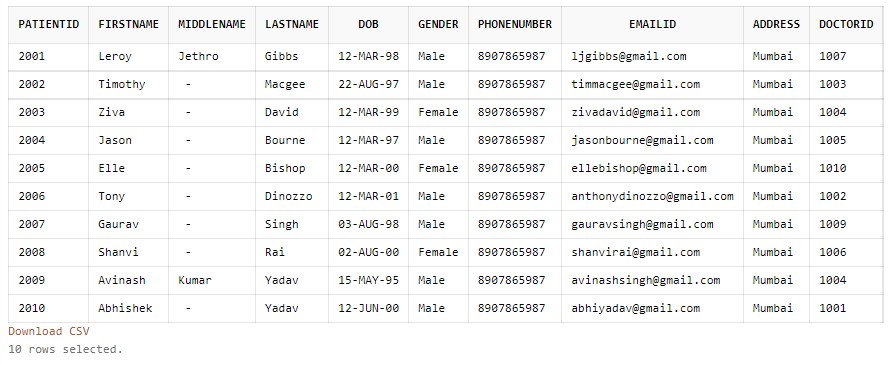
**.**

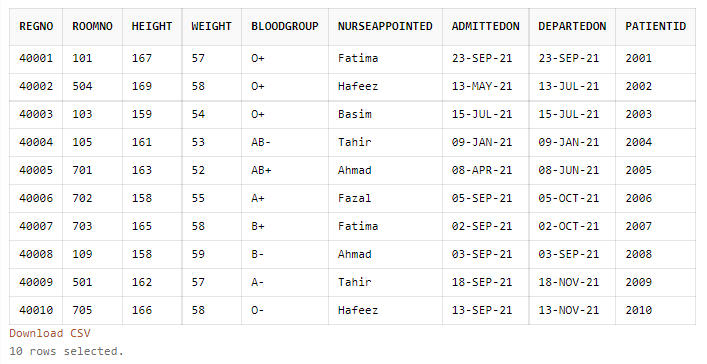
****

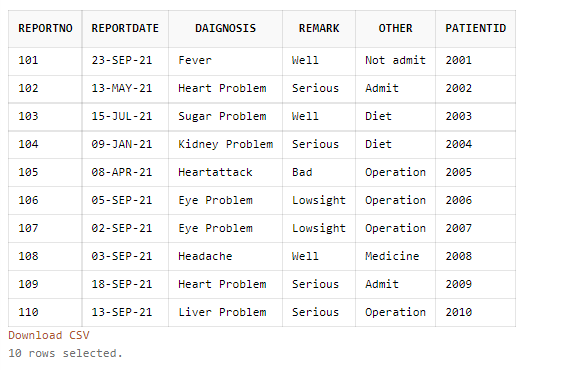
* **Select Statement :**

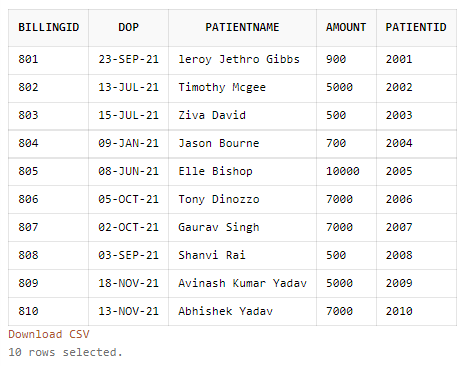
****

****

****

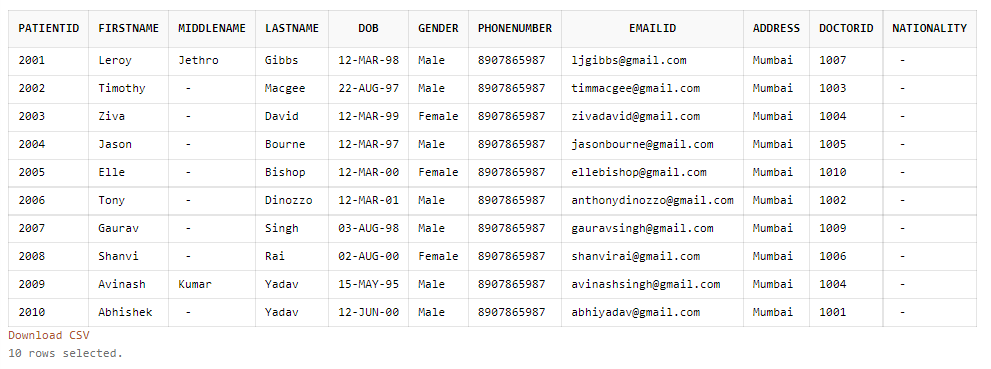
****

****

****

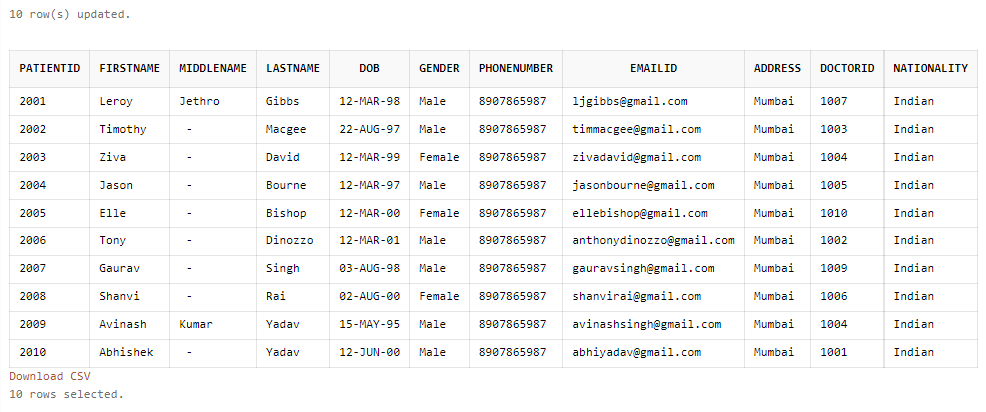
* **Alter Table - Add column**

****

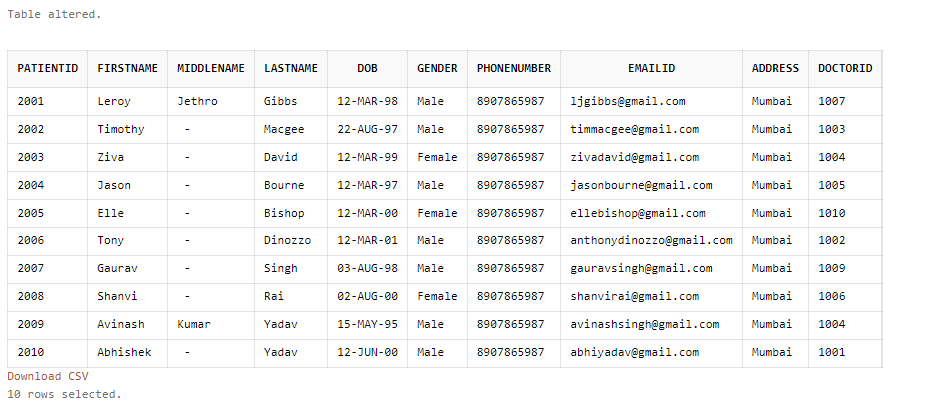
****

* **Update Table**

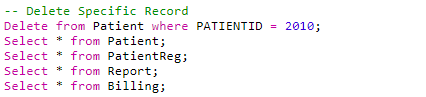
****

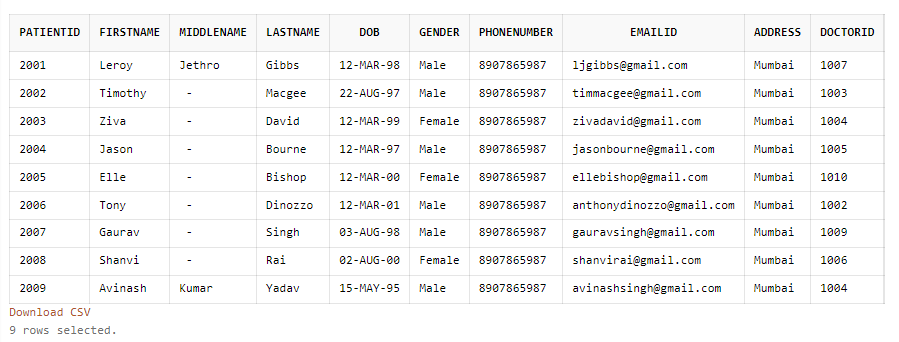
****

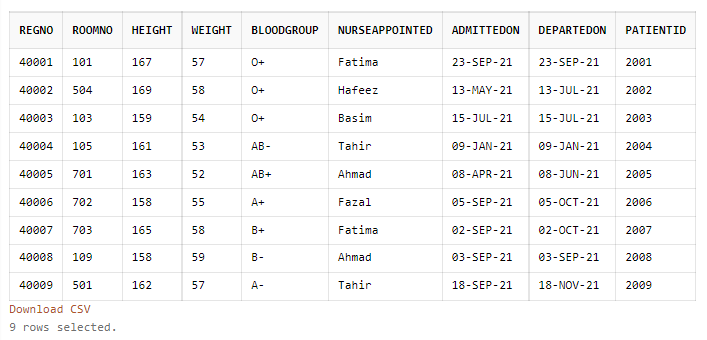
****

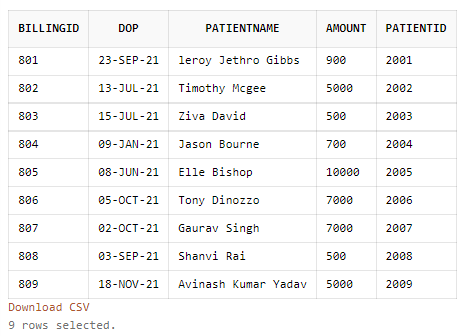
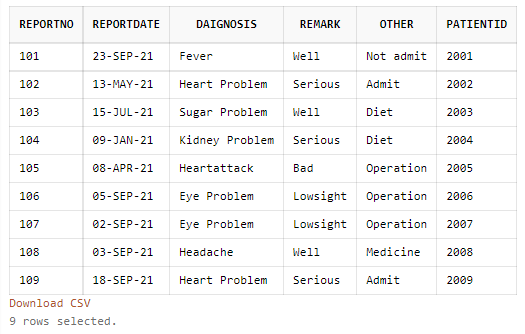
****

* **Delete Specific Record from the table :**

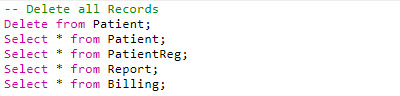
****

****

****

****

* **Delete all Records :**

****

****

**Execution of TCL Command**

**Aim :** To execute the TCL command on the Hospital Database.

**Command:**

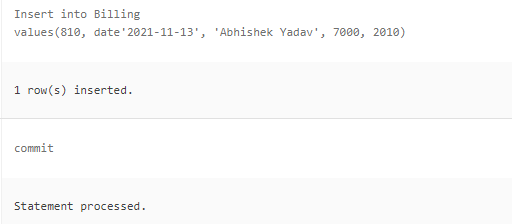
**1. Commit -**



**.**

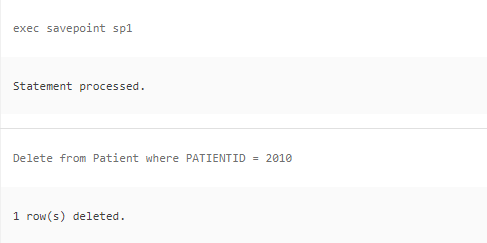
**.**

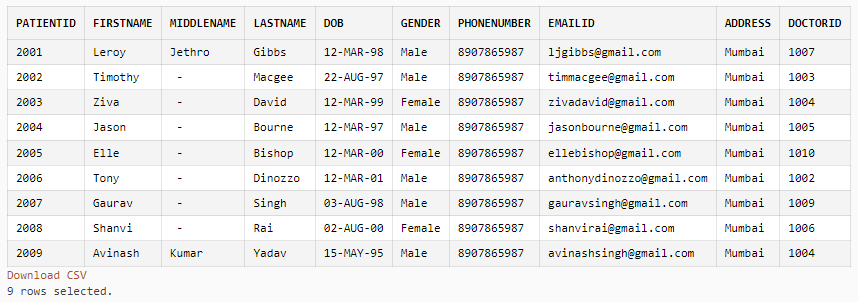
**.**

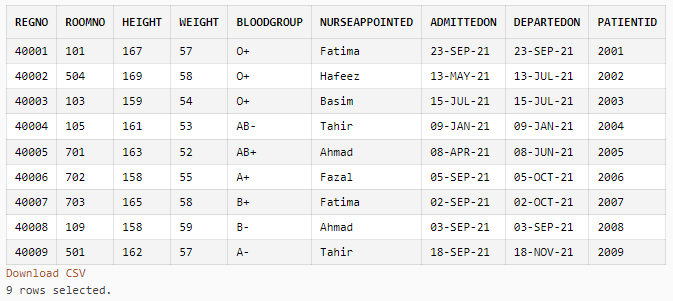


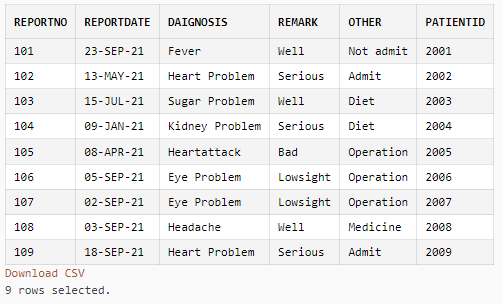
**Result :** TCL command Commit has been successfully executed.

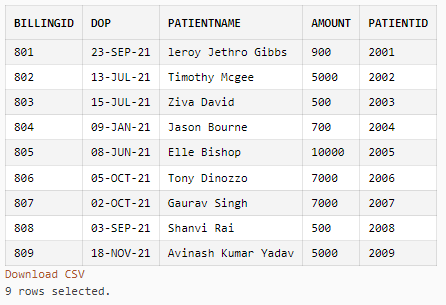
**2. Savepoint -**

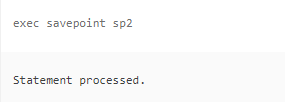






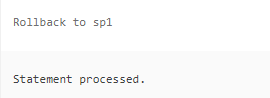


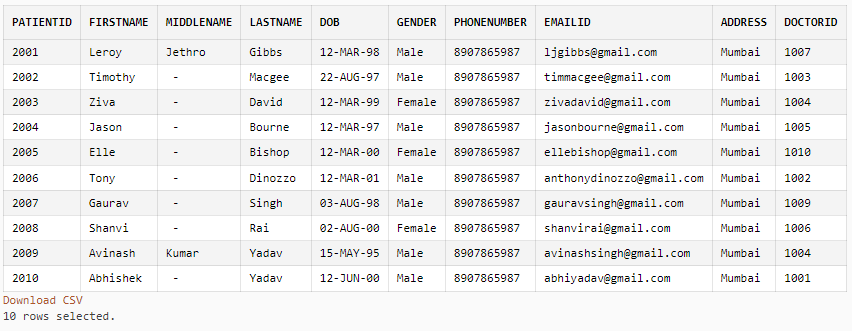


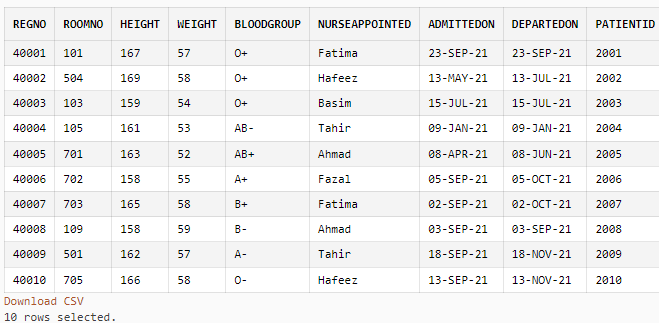


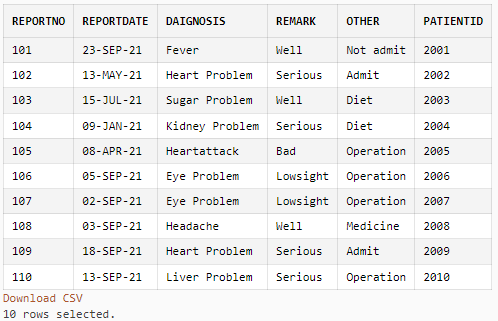
**Result :-** TCL command Savepoint has been successfully executed.

**3. Rollback -**







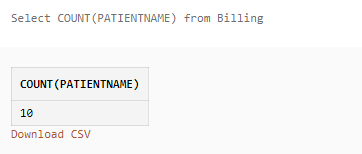


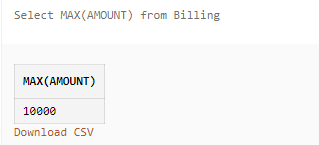
**Execution the Inbuilt function**

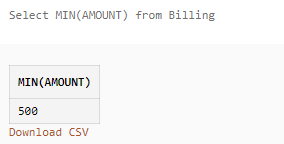
**Aim :** To execute the Inbuilt function on the Hospital Database.

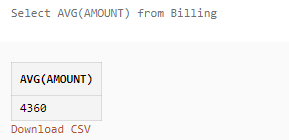
**Command:** The Inbuilt functions are as follows:

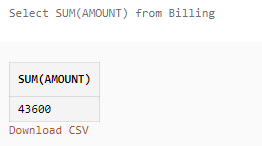
Count(), Max(), Min(), Avg(), Sum(), Sqrt(), Concat(), Upper(), lower()



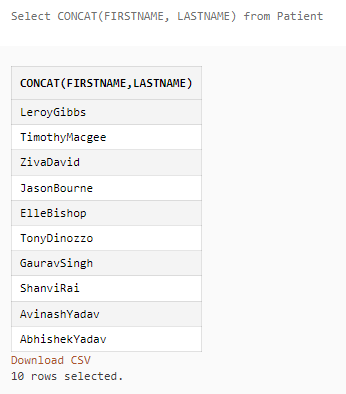


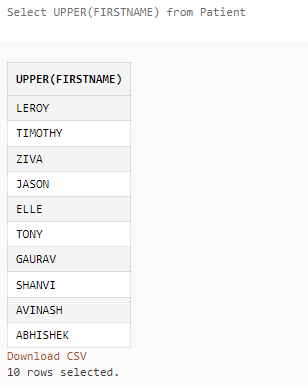




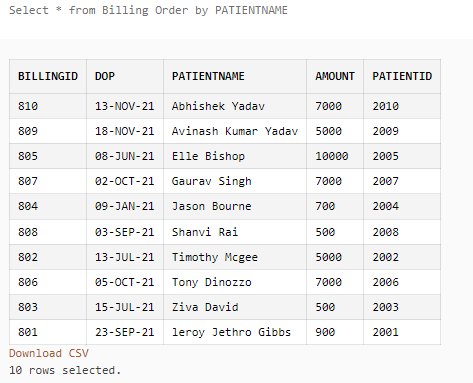






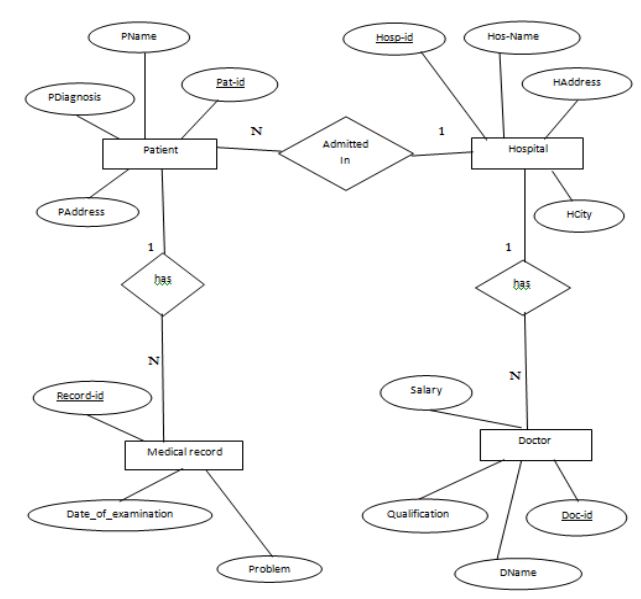






**E. R. Relationship diagram**

**Hospital management system**

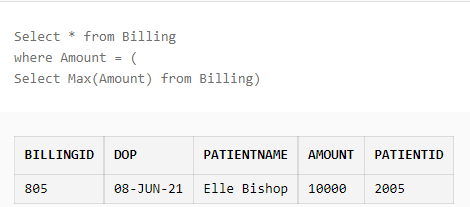
****

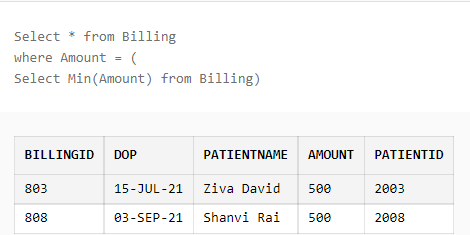
For table :

* Patient
* Medical record
* Doctor
* Hospital

**Aim :** To execute a Nested Query.

**Code and Output:**





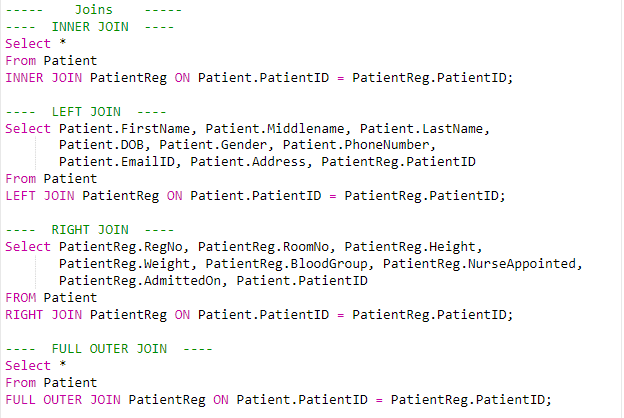
**JOINs Clause**

**Aim :** To execute the Join Clause.

There are four Joins Clause

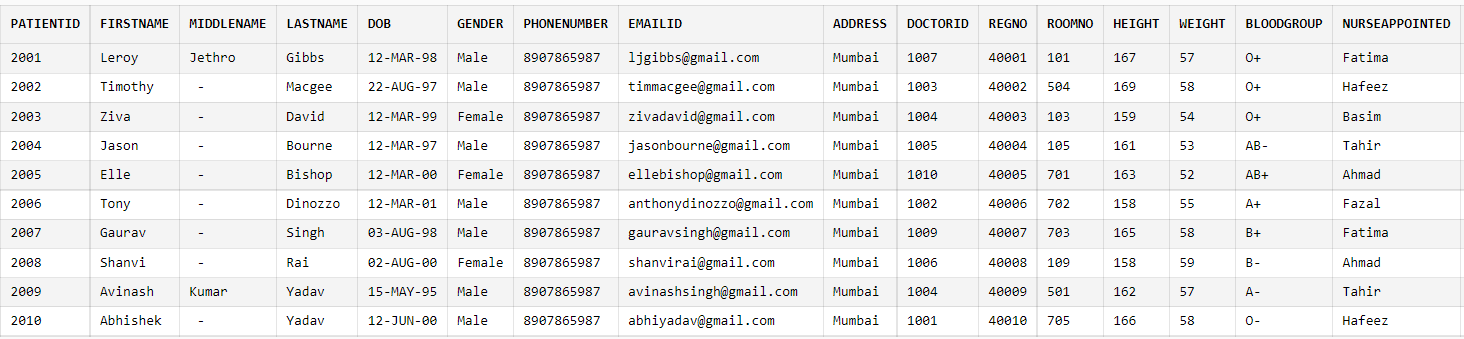
* Inner Join
* Left Join
* Right Join
* Full Outer Join

**Code :**

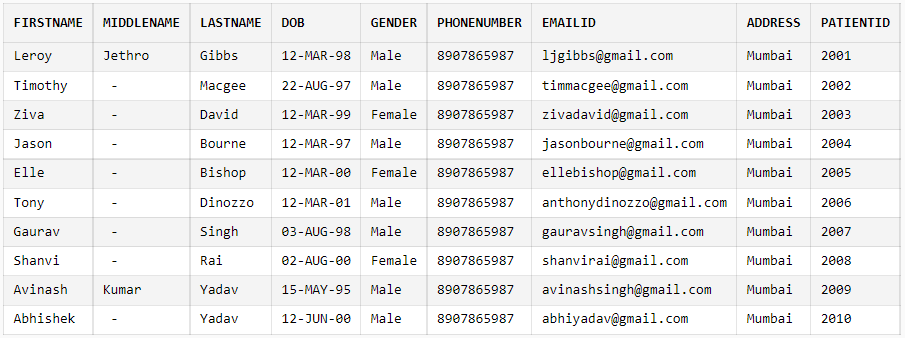


**Output :**

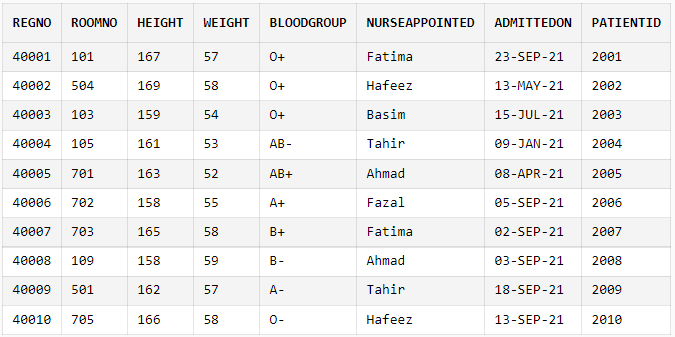
**Inner Join -**

****

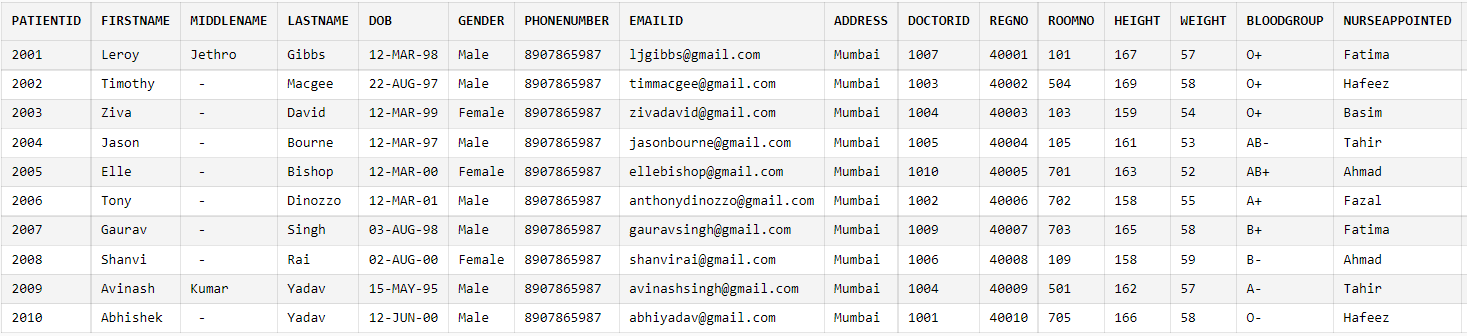
**Left Join -**



**Right Join -**



Full Outer Join -



Triggers

Aim : To create and execute the Triggers.

Command :

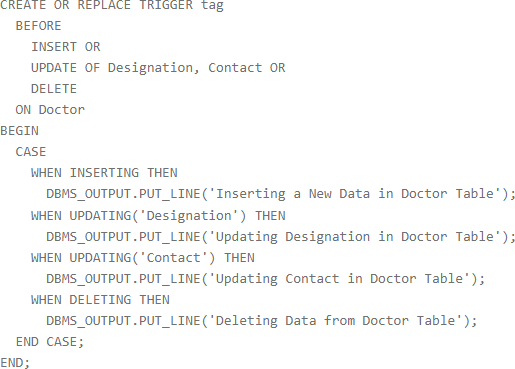
CREATE TRIGGER Trigger\_Name

[ BEFORE | AFTER ] [ Insert | Update | Delete] ON [Table\_Name]

[ FOR EACH ROW | FOR EACH COLUMN ] AS

Set of SQL Statement

Code :



Output :



**Exception Handling**

**Aim :** To execute a Nested Query.

**Syntax:** Raise with Exception name

CREATE [ PROCEDURE | FUNCTION ]

AS

BEGIN

<Execution block>

RAISE <exception\_name>

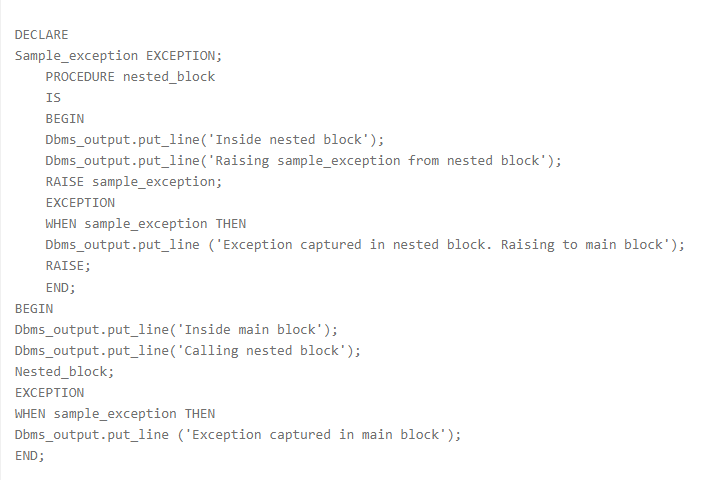
EXCEPTION

WHEN <exception\_name> THEN

<Handler>

END;

**Code :**



**Output :**

