**Acropolis Institute Of Technology And Research, Indore(M.P.)**



**Subject – Database Management System (DBMS)**

**(CY-405)**

**Name – Samyak Jain**

**Enrollment No. - 0827CY221056**

**Branch - CS(Cyber Security)**

**Semester- 4th sem**

**Submitted To –Mrs. Nidhi Nigam Ma’am**

**Index**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.No.** | **Experiment** | **Date of Exp.** | **Date of sub.** | **Grade** |
| 1. | To study DBMS and RDBMS, its characteristic comparisons and study of popular DB software. | 11/03/24 | 18/03/24 |  |
| 2. | To study of MYSQL, features and installation of MYSQL. | 01/04/24 |  |  |
| 3 | **Implementation of Database commands in MySQL (create database,show database,use database,drop**  **database)** |  |  |  |
| 4 | **Implementation of DDL commands in MySQL (create table ,alter table ,drop table)** |  |  |  |
| 5 | **Implementation of DML commands in MySQL**  **(Insert , Update ,Delete)** |  |  |  |
| 6 | **Implementation of different clauses in SQL with suitable examples**  **· Where**  **· Group By**  **· Having**  **· Order By** |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Assignment-1**

**AIM:-**

To study DBMS and RDBMS, its characteristic comparisons and study of popular DB software.

**Database Management System(DBMS):-**

Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

## Applications of DBMS

* **Enterprise Information:** Sales, accounting, human resources, Manufacturing, online retailers.
* **Banking and Finance Sector:** Banks maintaining the customer details, accounts, loans, banking transactions, credit card transactions. Finance: Storing the information about sales and holdings, purchasing of financial stocks and bonds.
* **University:**Maintaining the information about student course enrolled information, student grades, staff roles.
* **Airlines:**Reservations and schedules.
* **Telecommunications:**Prepaid, postpaid bills maintance.

**Features:-**

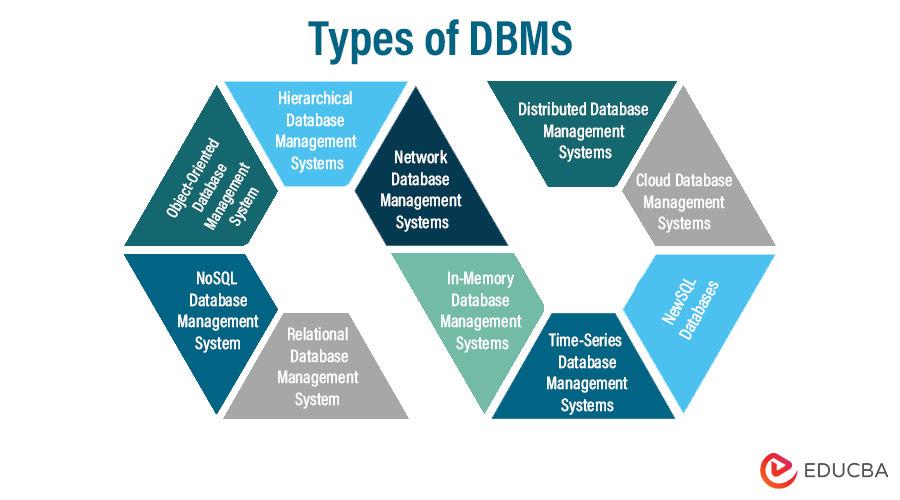
* **Data modeling:** A DBMS provides tools for creating and modifying data models, which define the structure and relationships of the data in a database.
* **Data storage and retrieval:** A DBMS is responsible for storing and retrieving data from the database, and can provide various methods for searching and querying the data.
* **Concurrency control:** A DBMS provides mechanisms for controlling concurrent access to the database, to ensure that multiple users can access the data without conflicting with each other.
* **Data integrity and security:** A DBMS provides tools for enforcing data integrity and security constraints, such as constraints on the values of data and access controls that restrict who can access the data.
* **Backup and recovery:** A DBMS provides mechanisms for backing up and recovering the data in the event of a system failure.

**Functions Of DBMS:-**

1. Data Organization
2. Data Independency
3. Data Security
4. Data Integrity

TYPES OF DBMS

There are various types of databases used for storing different varieties of data:



## 1) Centralized Database

It is the type of database that stores data at a centralized database system. It comforts the users to access the stored data from different locations through several applications. These applications contain the authentication process to let users access data securely. An example of a Centralized database can be Central Library that carries a central database of each library in a college/university.

## 2) Distributed Database

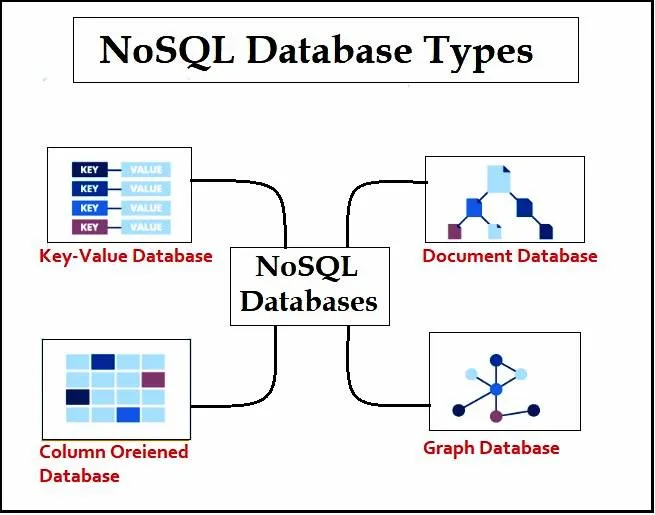
Unlike a centralized database system, in distributed systems, data is distributed among different database systems of an organization. These database systems are connected via communication links. Such links help the end-users to access the data easily. **Examples** of the Distributed database are Apache Cassandra, HBase, Ignite, etc.

## 3) Relational Database

This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation). A relational database uses SQL for storing, manipulating, as well as maintaining the data. E.F. Codd invented the database in 1970. Each table in the database carries a key that makes the data unique from others. **Examples** of Relational databases are MySQL, Microsoft SQL Server, Oracle, etc.

## 4) NoSQL Database

Non-SQL/Not Only SQL is a type of database that is used for storing a wide range of data sets. It is not a relational database as it stores data not only in tabular form but in several different ways. It came into existence when the demand for building modern applications increased. Thus, NoSQL presented a wide variety of database technologies in response to the demands. We can further divide a NoSQL database into the following four types:



## 5) Cloud Database

A type of database where data is stored in a virtual environment and executes over the cloud computing platform. It provides users with various cloud computing services (SaaS, PaaS, IaaS, etc.) for accessing the database. There are numerous cloud platforms, but the best options are:

* Amazon Web Services(AWS)
* Microsoft Azure
* Kamatera
* PhonixNAP
* ScienceSoft
* Google Cloud SQL, etc.

## 6) Object-oriented Databases

The type of database that uses the object-based data model approach for storing data in the database system. The data is represented and stored as objects which are similar to the objects used in the object-oriented programming language.

## 7) Hierarchical Databases

It is the type of database that stores data in the form of parent-children relationship nodes. Here, it organizes data in a tree-like structure.

Data get stored in the form of records that are connected via links. Each child record in the tree will contain only one parent. On the other hand, each parent record can have multiple child records.

## 8) Network Databases

It is the database that typically follows the network data model. Here, the representation of data is in the form of nodes connected via links between them. Unlike the hierarchical database, it allows each record to have multiple children and parent nodes to form a generalized graph structure.

**Relational Database Management System(RDBMS):-**

A RDBMS is a form of DBMS that stores and manages facts in a tabular format, prepared as tables with rows and columns. It is based on the relational version proposed via Edgar F. Codd within the 1970s. RDBMS is widely used because of its simplicity, scalability, and efficiency in dealing with complicated relationships between entities.

**Functions Of RDBMS:-**

1. Data Structure
2. ACID Compliance
3. Query Language
4. Scalability

**Applications Of RDBMS:-**

1. Enterprise Applications
2. Web Applications
3. Scientific Research
4. Government Systems

**Comparison between DBMS and RDBMS:-**

|  |  |  |
| --- | --- | --- |
| **No.** | **DBMS** | **RDBMS** |
| 1. | DBMS applications store **data as file**. | RDBMS applications store **data in a tabular form**. |
| 2. | There is **no relation between the tables.** | Data values are stored in the form of tables, so a **relationship** between these data values will be stored in the form of a table as well. |
| 3. | Does not support distributed database. | Supports distributed database. |
| 4. | Deals with **small amount of data.** | Deals **with large amount of data.** |
| 5. | For **single user.** | For **multiple users.** |
| 6. | Less secure. | More Secure. |
| 7. | Data Redundancy is common. | No. Data Redundancy due to Keys and indexes. |
| 8. | Less than 7 Codd rules are satisfied. | More than 7 or all 12 Codd rules are satisfied. |
| 9. | Data fetching is slower. | Data fetching is faster. |
| 10. | Low software and hardware necessities. | Higher software and hardware necessities. |
| 11. | Ex:- Xml | Ex:- MySQL, Microsoft Access |

## MySQL is a relational database management system

[Databases](https://www.oracle.com/in/database/what-is-database/) are the essential data repository for all software applications. For example, whenever someone conducts a web search, logs in to an account, or completes a transaction, a database system is storing the information so it can be accessed in the future.

A [relational database](https://www.oracle.com/in/database/what-is-a-relational-database/) stores data in separate tables rather than putting all the data in one big storeroom. The database structure is organized into physical files optimized for speed. The logical data model, with objects such as data tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one to one, one to many, unique, required, or optional, and “pointers” between different tables. The database enforces these rules so that with a well-designed database your application never sees data that’s inconsistent, duplicated, orphaned, out of date, or missing.

The “SQL” part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

**Characteristic Comparison between SQL and NoSQL:-**

|  |  |  |
| --- | --- | --- |
| **No.** | **SQL** | **NoSQl** |
| 1. | RDBMS | Non-relational Or Distributed database system. |
| 2. | Fixed or static or pre-defined schema. | Dynamic Schema. |
| 3. | Not for hierarchical data storage. | For hierarchical data storage. |
| 4. | Follows ACID property. | Does not follows ACID property. |
| 5. | Ex:- MySQL, Oracle | Ex:-MongoDB,HBase |

**Assignment -2**

**Aim:-**

To study of MYSQL, features and installation of MYSQL.

**What is MySQL ?**

MySQL is an open-source [**Relational Database Management System**](https://www.hostinger.com/tutorials/dbms) (RDBMS) that enables users to store, manage, and retrieve structured data efficiently. It is widely used for various applications, from small-scale projects to large-scale websites and enterprise-level solutions.

There are a few elements of MySQL. Let’s take a closer look at them:

### **Database**

In relation to MySQL, a database is a structured collection of data organized and stored in tables. It serves as a central repository where information is efficiently managed, allowing users to store, retrieve, update, and delete data. MySQL provides the software framework to create, maintain, and interact with these databases, making data storage and retrieval seamless and reliable.

### **Client-Server Model**

Computers that install and run RDBMS software are called clients. Whenever they need to access data, they connect to the RDBMS server.

MySQL is one of many RDBMS software options. RDBMS and MySQL are often thought to be the same because of MySQL’s popularity. A few [**big web applications**](https://stackshare.io/mysql) like Facebook, Twitter, YouTube, Google, and Yahoo! all use MySQL for data storage purposes. Even though it was initially created for limited usage, it is now compatible with many important computing platforms like Linux, macOS, Microsoft Windows, and Ubuntu.

### **SQL**

[**MySQL and SQL are not the same**](https://www.hostinger.com/tutorials/difference-between-mysql-and-sql-server). Be aware that MySQL is one of the most popular RDBMS software’s brand names, which implements a client-server model.

The client and server use a domain-specific language – Structured Query Language (SQL) to communicate in an RDBMS environment. If you ever encounter other names that have SQL in them, like [**PostgreSQL**](https://www.hostinger.com/tutorials/how-to-install-postgresql-on-ubuntu/) and Microsoft SQL server, they are most likely brands which also use Structured Query Language syntax. RDBMS software is often written in other programming languages but always uses SQL as its primary language to interact with the database. MySQL itself is written in **C** and **C++**.

SQL tells the server what to do with the data. In this case, SQL statements can instruct the server to perform certain operations:

* **Data query** – requesting specific information from the existing database.
* **Data manipulation** – adding, deleting, changing, sorting, and other operations to modify the data, the values or the visuals.
* **Data identity** – defining data types, e.g. changing numerical data to integers. This also includes defining a [**schema**](https://www.hostinger.com/tutorials/database-schema) or the relationship of each table in the database
* **Data access control** – providing security techniques to protect data. This includes deciding who can view or use any information stored in the database

### **Open-Source**

Open-source means that you’re free to use and modify it. You can also learn and customize the source code to better accommodate your needs. However, The GPL ([**GNU Public License**](https://www.gnu.org/licenses/gpl-3.0.en.html)) determines what you can do depending on the conditions. The commercially licensed version is available if you need more flexible ownership and advanced support.

## Why is MySQL So Popular?

MySQL is indeed not the only RDBMS on the market, but it is one of the most popular ones. The fact that many major tech giants rely on it further solidifies the well-deserved position. Here are some of the reasons.

1. **Flexible and Easy To Use**

As open-source software, you can modify the source code to suit your need and don’t need to pay anything. It includes the option for upgrading to the advanced commercial version. The installation process is relatively simple, and shouldn’t take longer than 30 minutes.

1. **High Performance**

A wide array of cluster servers backs MySQL. Whether you are storing massive amounts of big eCommerce data or doing heavy business intelligence activities, MySQL can assist you smoothly with optimum speed.

1. **An Industry Standard**

Industries have been using MySQL for years, which means that there are abundant resources for skilled developers. MySQL users can expect rapid development of the software and freelance experts willing to work for a smaller wage if they ever need them.

1. **Secure**

Your data should be your primary concern when choosing the right RDBMS software. With its Access Privilege System and User Account Management, MySQL sets the security bar high. Host-based verification and password encryption are both available.

### How query  works?

### Download MySQL

Follow these steps:

**Step 1:** Go to the [official website](https://dev.mysql.com/downloads/installer/) of MySQL and download the community server edition software. Here, you will see the option to choose the Operating System, such as Windows.

**Step 2:** Next, there are two options available to download the setup. Choose the version number for the MySQL community server, which you want. If you have good internet connectivity, then choose the mysql-installer-web-community. Otherwise, choose the other one.

**Installing MySQL on Windows**

**Step 1:** After downloading the setup, unzip it anywhere and double click the MSI **installer .exe file.** It will give the following screen:

**Step 2:** In the next wizard, choose the **Setup Type**. There are several types available, and you need to choose the appropriate option to install MySQL product and [features](https://www.javatpoint.com/mysql-features). Here, we are going to select the **Full** option and click on the Next button.

This option will install the following things: MySQL Server, MySQL Shell, MySQL Router, [MySQL Workbench](https://www.javatpoint.com/mysql-workbench), MySQL Connectors, documentation, samples and examples, and many more.

**Step 3:** Once we click on the Next button, it may give information about some features that may fail to install on your system due to a lack of requirements. We can resolve

them by clicking on the **Execute** button that will install all requirements automatically or can skip them. Now, click on the Next button.

**Step 4:** In the next wizard, we will see a dialog box that asks for our confirmation of a few products not getting installed. Here, we have to click on the **Yes** button.

After clicking on the Yes button, we will see the list of the products which are going to be installed. So, if we need all products, click on the Execute button.

**Step 5:** Once we click on the Execute button, it will download and install all the products. After completing the installation, click on the Next button.

**Step 6:** In the next wizard, we need to configure the MySQL Server and Router. Here, I am not going to configure the Router because there is no need to use it with MySQL. We are going to show you how to configure the server only. Now, click on the Next button.

**Step 7:** As soon as you will click on the Next button, you can see the screen below. Here, we have to configure the MySQL Server. Now, choose the Standalone MySQL Server/Classic MySQL Replication option and click on Next. Here, you can also choose the InnoDB Cluster based on your needs.

**Step 8:** In the next screen, the system will ask you to choose the Config Type and other connectivity options. Here, we are going to select the **Config Type** as 'Development Machine' and Connectivity as **TCP/IP,** and **Port Number** is 3306, then click on Next.

**Step 9:** Now, select the Authentication Method and click on Next. Here, I am going to select the first option.

**Step 10:** The next screen will ask you to mention the MySQL Root Password. After filling the password details, click on the Next button.

**Step 11:** The next screen will ask you to configure the Windows Service to start the server. Keep the default setup and click on the Next button.

**Step 12:** In the next wizard, the system will ask you to apply the Server Configuration. If you agree with this configuration, click on the Execute button.

**Step 13:** Once the configuration has completed, you will get the screen below. Now, click on the **Finish** button to continue.

**Step 14:** In the next screen, you can see that the Product Configuration is completed. Keep the default setting and click on the Next-> Finish button to complete the MySQL package installation.

**Step 15:** In the next wizard, we can choose to configure the Router. So click on Next->Finish and then click the Next button.

**Step 16:** In the next wizard, we will see the Connect to Server option. Here, we have to mention the root password, which we had set in the previous steps.

In this screen, it is also required to check about the connection is successful or not by clicking on the Check button. If the connection is successful, click on the Execute button. Now, the configuration is complete, click on Next.

**Step 17:** In the next wizard, select the applied configurations and click on the Execute button.

**Step 18:** After completing the above step, we will get the following screen. Here, click on the Finish button.

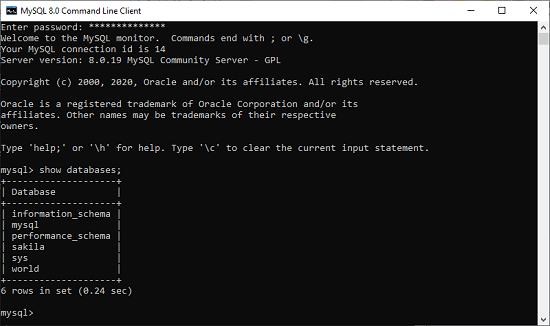
**Step 19:** Now, the MySQL installation is complete. Click on the Finish button.

### Verify MySQL installation

Once MySQL has been successfully installed, the base tables have been initialized, and the server has been started, you can verify its working via some simple tests.

Open your MySQL **Command Line Client**; it should have appeared with a **mysql> prompt**. If you have set any password, write your password here. Now, you are connected to the MySQL server, and you can execute all the SQL command at mysql> prompt as follows:

**For example**: Check the already created databases with show databases command:



**Experiment 3**

Implementation of Database commands

1. **CREATE DATABASE**

This command is used to create a database.

**Syntax** – CREATE DATABASE database\_name;

E.g. CREATE DATABASE uni;

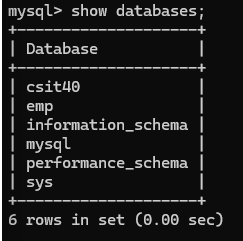


Database names ‘uni’ is created.

1. **SHOW DATABASE**

This is used to show all the databases.

**Syntax:**  SHOW DATABASES;



A list of all the databases is displayed.

1. **DROP DATABASE**

This command is used to delete a database.

**Syntax:** DROP DATABASE database\_name;

E.g. drop database uni;



Now database named uni is deleted.

1. **USE DATABASE**

To change database

**Syntax:** USE database\_name;

E.g. USE uni;



1. **Create a table**

**Syntax:**

CREATE TABLE table\_name (

    column1 datatype constraints,

    column2 datatype constraints,

    …

    PRIMARY KEY (column\_name) -- optional, defines

primary key constraint

    -- other constraints such as FOREIGN KEY, UNIQUE, etc.

);

Where, table\_name is the name of the table you need to create, column1, column2, column3, … column N are the names of the columns and datatype is the name of the datatypes of the respective columns.

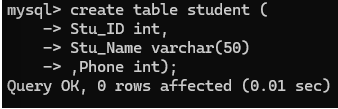
E.g.

 CREATE TABLE student (

    -> Stu\_ID int,

    -> Stu\_Name varchar (50)

    ->, Phone int);



1. **INSERT TABLE**

Syntax:

INSERT INTO table\_name (column1, column2, column3, ..., column N)VALUES (value1, value2, value3, ..., value N);

Where, table\_name is the name of the table into which you need to insert data, (column1, column2, column3, ..., column N) are the names of the columns and (value1, value2, value3, ..., value N) are the values in the record.

E.g.

 INSERT INTO student VALUES (101, 'ABC',12459), (102, 'PQR',7890), (103, 'XYZ',87541);



1. **Display data**

Use the SELECT statement

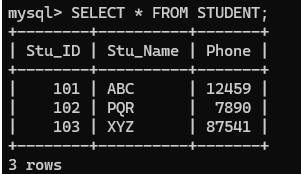
**Syntax:**

SELECT \* FROM table\_name;

This query will retrieve all rows and columns from the specified table.

E.g.

SELECT \* FROM STUDENT;

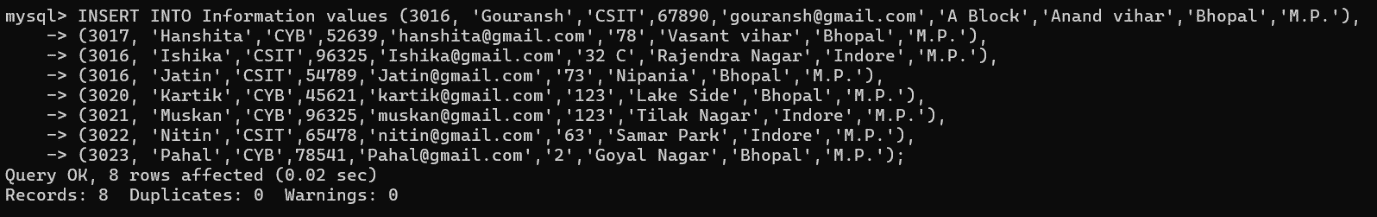


**Experiment 4**

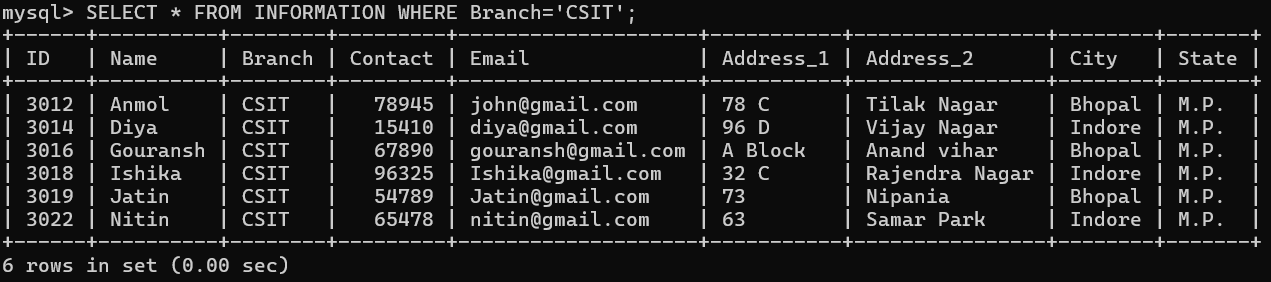
  Implementation of Database commands

Q.1 Insert at least 10 12 records in this table having the city name Indore and Bhopal and beach CSIT and CYB

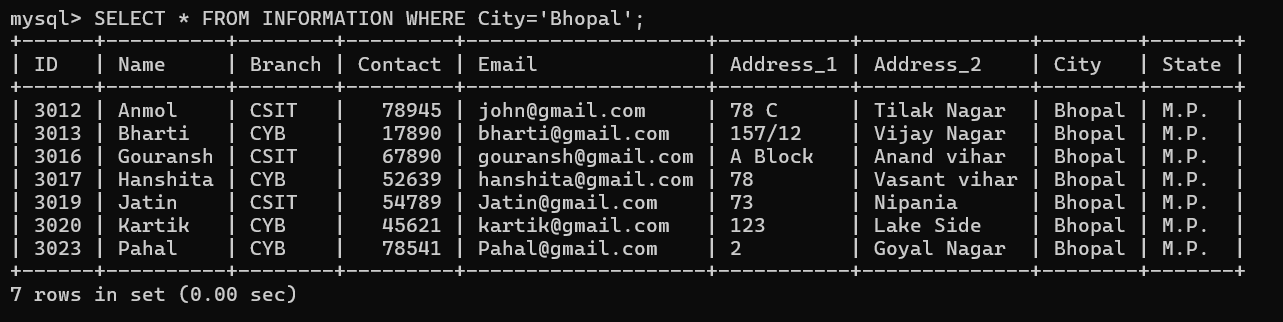




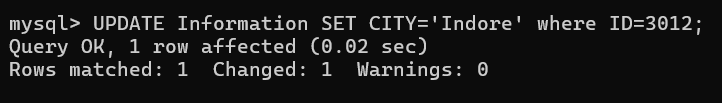
Q.2 Retrieve all the details of CSIT branch students

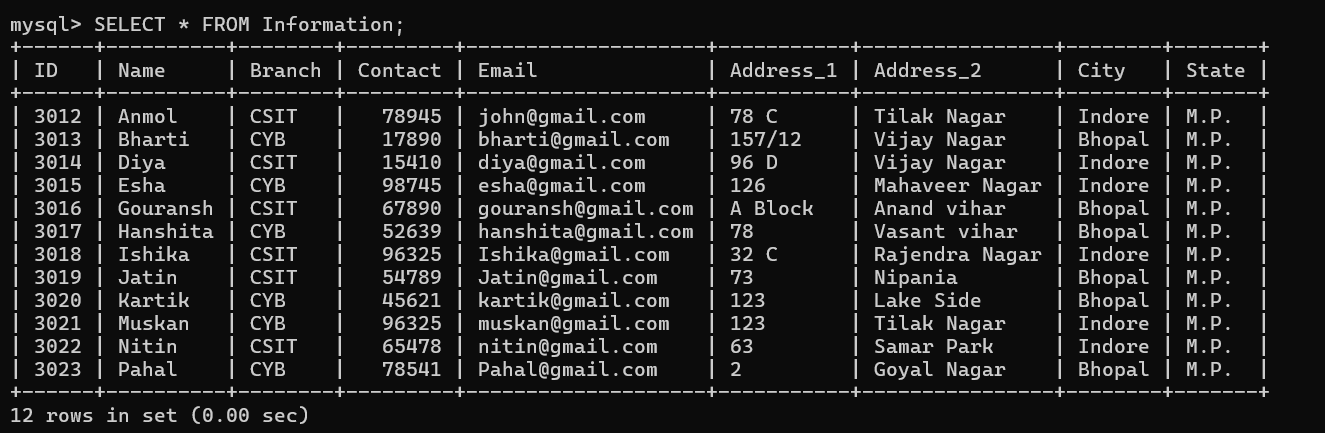


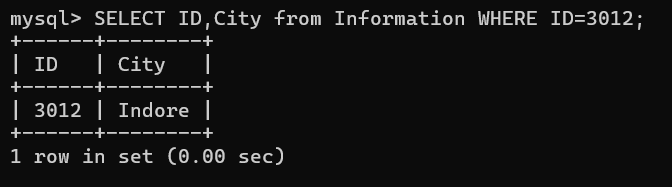
Q.3 Retrieve all the details of who is located at Bhopal



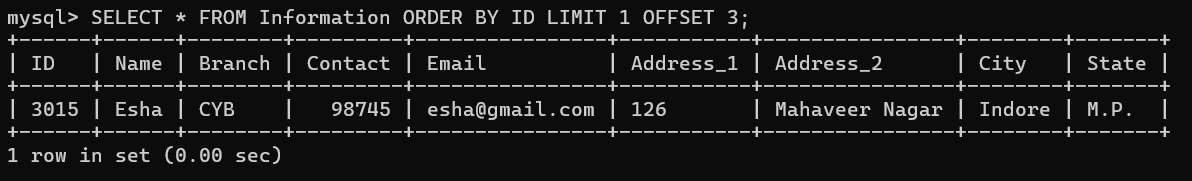
Q. 4



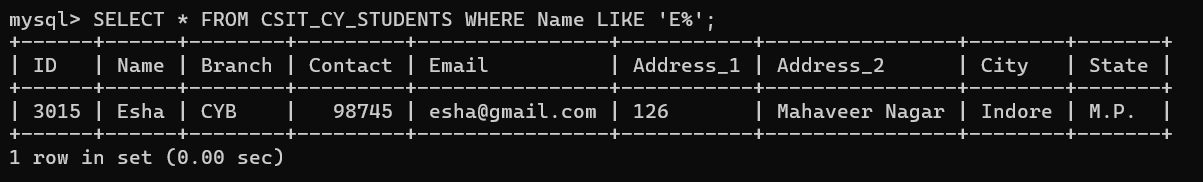




Q.5 Show the name and contact of the 4th record of your table



Q.6 Display the name that are starting with the alphabet e

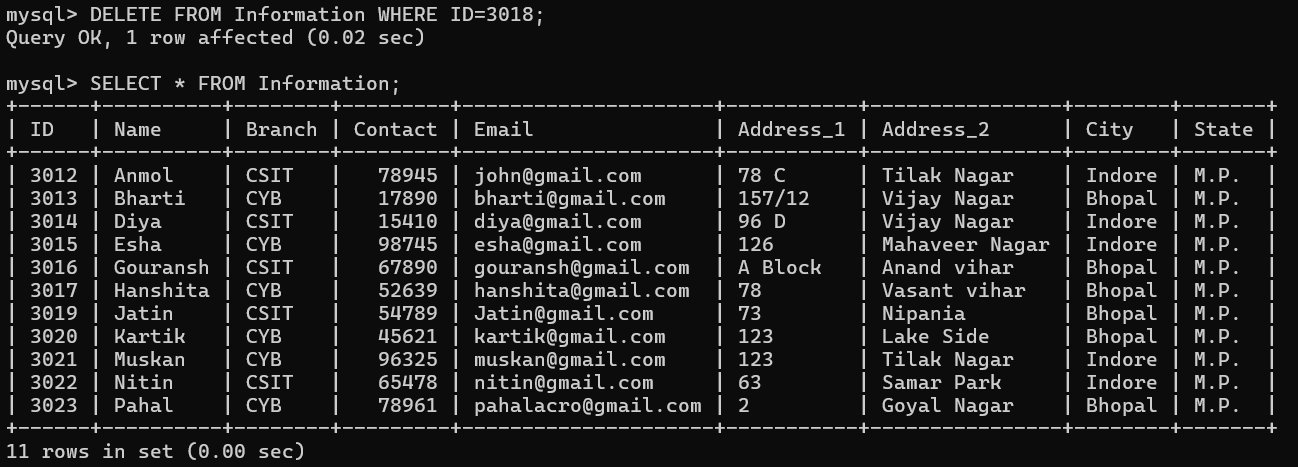


Q.7 Update the email and contact of the last record of your table

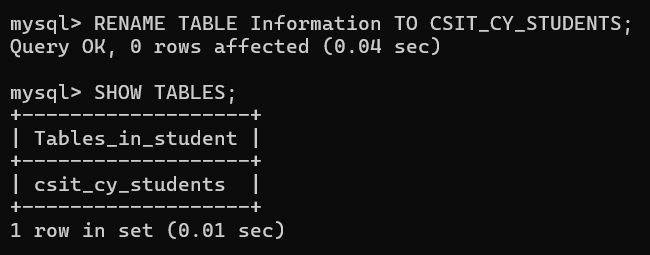




Q.8 Delete all the information of the 7th record on your table



Q.9 Change the name of the table from student to csit\_cy\_students



Experi**ment 5:**

**IMPLEMENTATIOIN OF DML COMMANDS IN SQL:**

**DML(Data Manipulation Language)**

The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

List of DML commands:

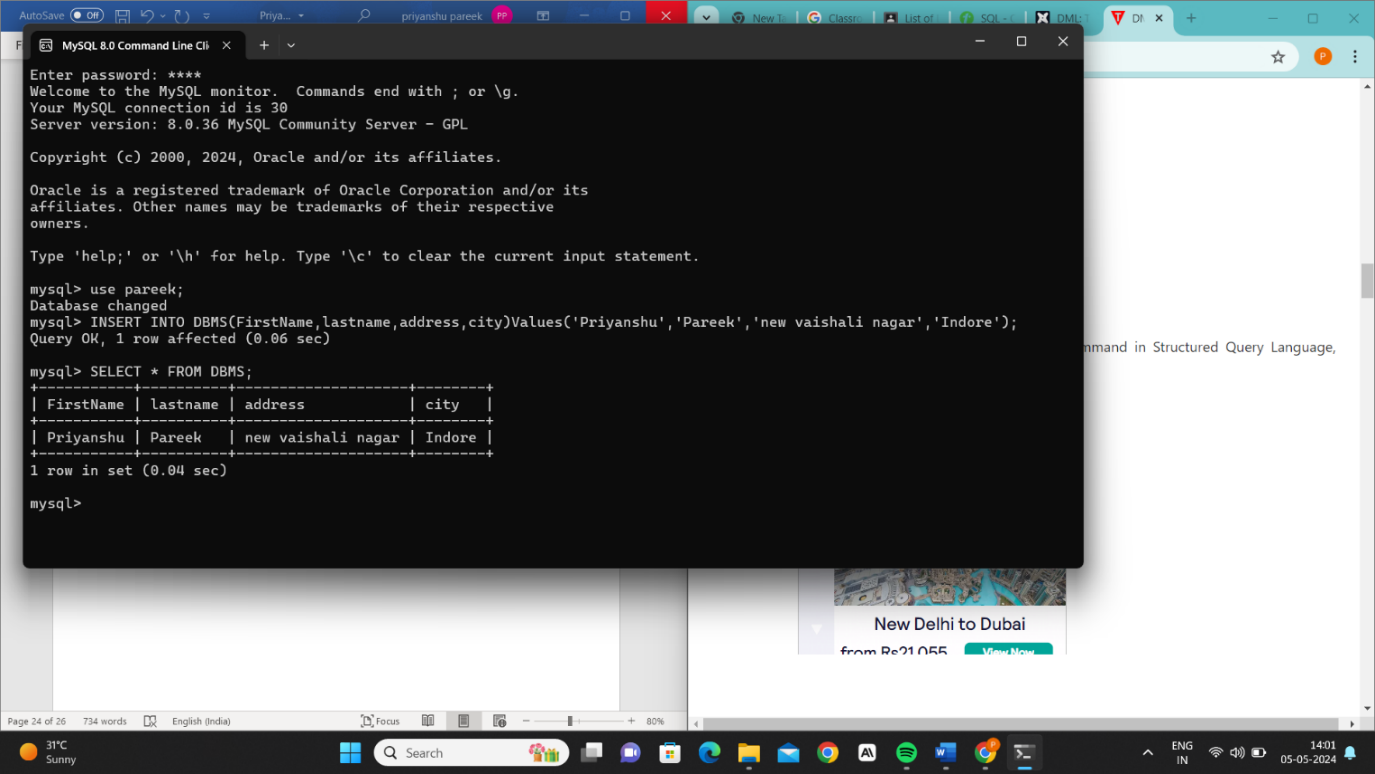
* [**INSERT**](https://www.geeksforgeeks.org/sql-insert-statement/): It is used to insert data into a table.
* [**UPDATE**](https://www.geeksforgeeks.org/sql-update-statement/)**:** It is used to update existing data within a table.
* [**DELETE**](https://www.geeksforgeeks.org/sql-delete-statement/): It is used to delete records from a database table.

### **INSERT** **:**

Using the INSERT DML command, you can add rows to a table that exists in your database. To be honest, data folks are rarely inserting data into tables manually with the INSERT command. Instead, data team members will most often use data that’s already been inserted by an [ELT](https://docs.getdbt.com/terms/elt) tool or other data ingestion process.

SYNTAX :

**INSERT** **INTO** TABLE\_NAME ( column\_Name1 , column\_Name2 , column\_Name3 , .... column\_NameN ) **VALUES** (value\_1, value\_2, value\_3, .... value\_N ) ;

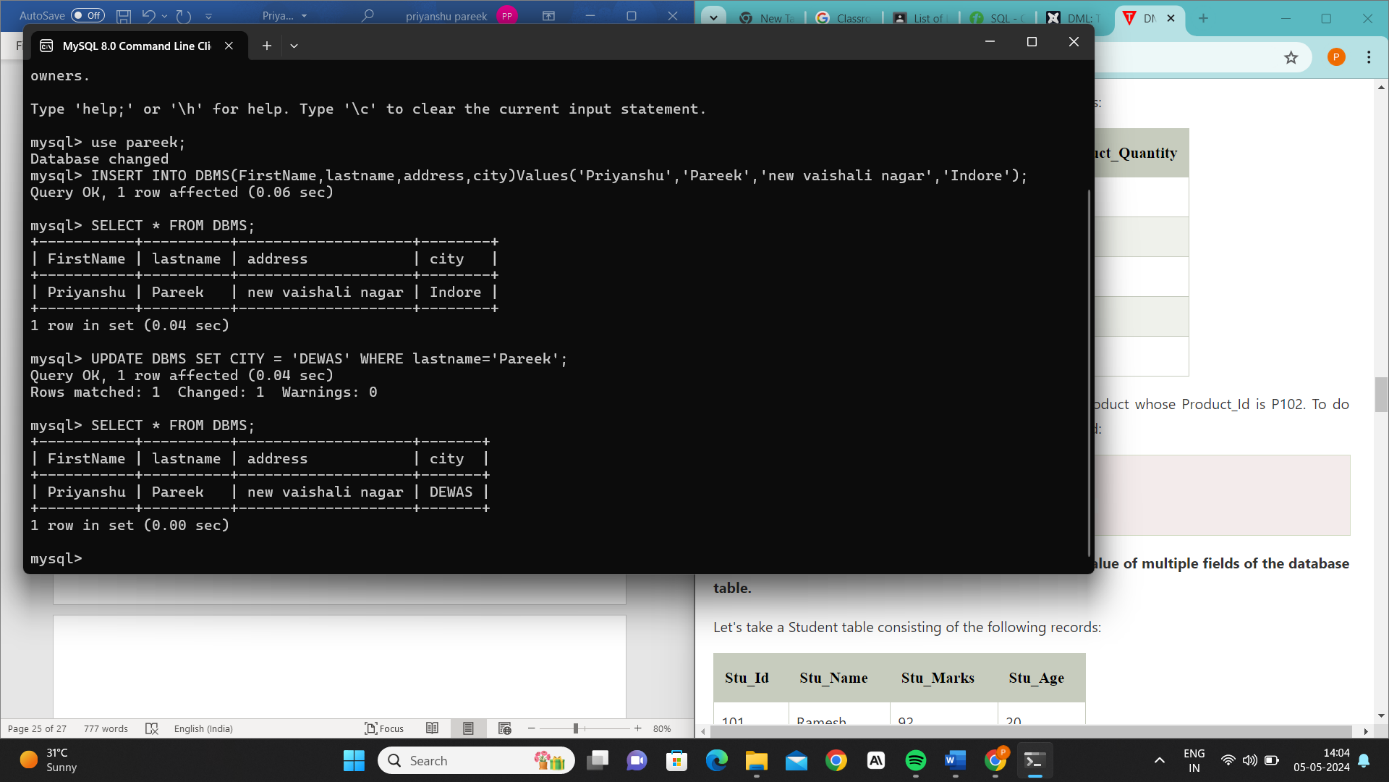


## UPDATE DML Command

UPDATE is another most important data manipulation command in Structured Query Language, which allows users to update or modify the existing data in database tables.

**Syntax of UPDATE Command:**

**UPDATE** Table\_name **SET** [column\_name1= value\_1, ….., column\_nameN = value\_N] **WHERE** CONDITION;



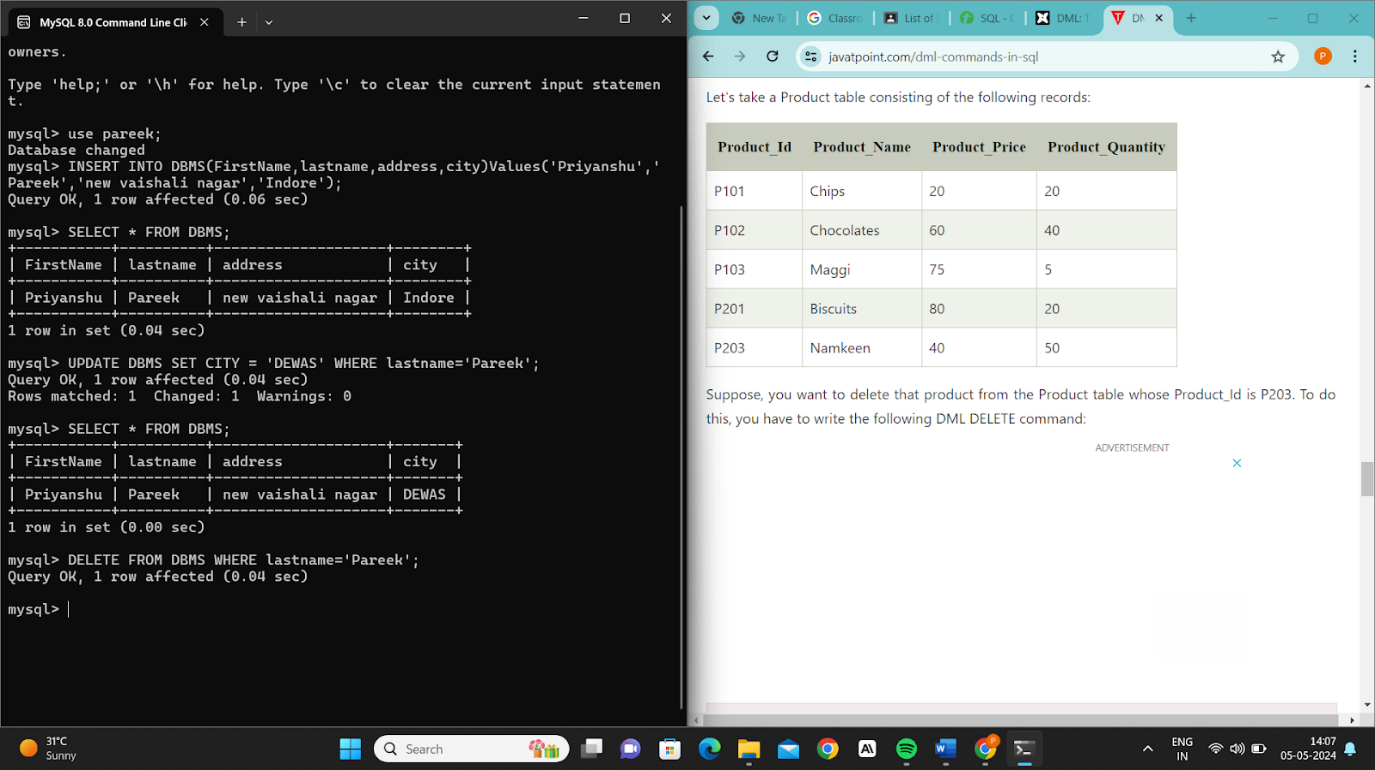
## DELETE DML Command

DELETE is a DML command which allows SQL users to remove single or multiple existing records from the database tables.

This command of Data Manipulation Language does not delete the stored data permanently from the database. We use the WHERE clause with the DELETE command to select specific rows from the table.

**Syntax of DELETE Command**

**DELETE** **FROM** Table\_Name **WHERE** condition;



### 

### **Experiment 6:**

### **IMPLEMENTATION OF CLAUSES IN SQL:**

**Different clauses available in the Structured Query Language are as follows:**

1. WHERE CLAUSE
2. GROUP BY CLAUSE
3. HAVING CLAUSE
4. ORDER BY CLAUSE

Let's see each clause one by one with an example. We will use MySQL database for writing the queries in examples.

1. WHERE CLAUSE

A WHERE clause in SQL is used with the SELECT query, which is one of the data manipulation language commands. WHERE clauses can be used to limit the number of rows to be displayed in the result set, it generally helps in filtering the records. It returns only those queries which fulfill the specific conditions of the WHERE clause. WHERE clause is used in SELECT, UPDATE, DELETE statement, etc.

WHERE clause with SELECT Query

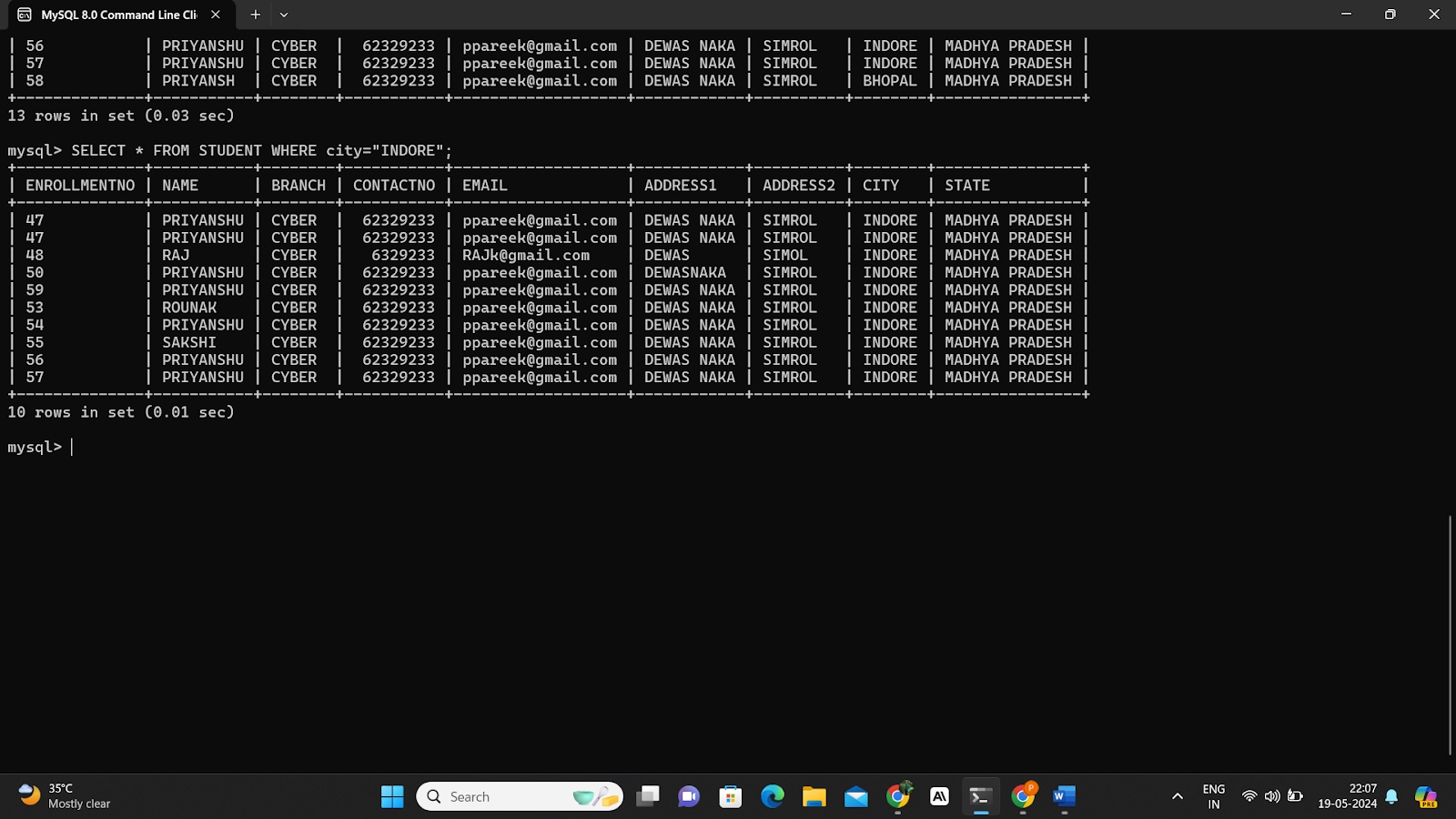
Asterisk symbol is used with a WHERE clause in a SELECT query to retrieve all the column values for every record from a table.

**Syntax of where clause with a select query to retrieve all the column values for every record from a table:**

**SELECT** \* **FROM** TABLENAME **WHERE** CONDITION;

**If according to the requirement, we only want to retrieve selective columns, then we will use below syntax:**

**SELECT** COLUMNNAME1, COLUMNNAME2 **FROM** TABLENAME **WHERE** CONDITION;



2. GROUP BY CLAUSE

The Group By clause is used to arrange similar kinds of records into the groups in the Structured Query Language. The Group by clause in the Structured Query Language is used with Select Statement. Group by clause is placed after the where clause in the SQL statement. The Group By clause is specially used with the aggregate function, i.e., max (), min (), avg (), sum (), count () to group the result based on one or more than one column.

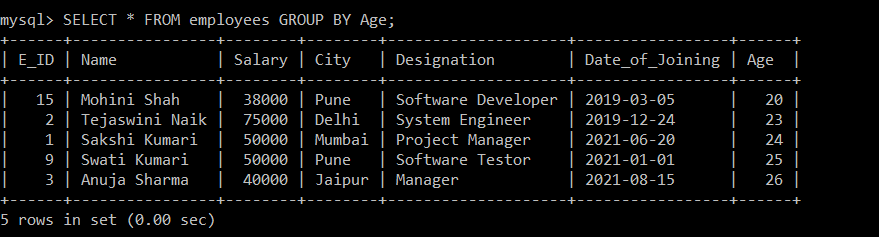
The syntax of Group By clause:

**SELECT** \* **FROM** TABLENAME **GROUP** **BY** COLUMNNAME;

The above syntax will select all the data or records from the table, but it will arrange all those data or records in the groups based on the column name given in the query.

The syntax of Group By clause with Aggregate Functions:

**SELECT** COLUMNNAME1, Aggregate\_FUNCTION (COLUMNNAME) **FROM** TABLENAME **GROUP** **BY** COLUMNNAME;

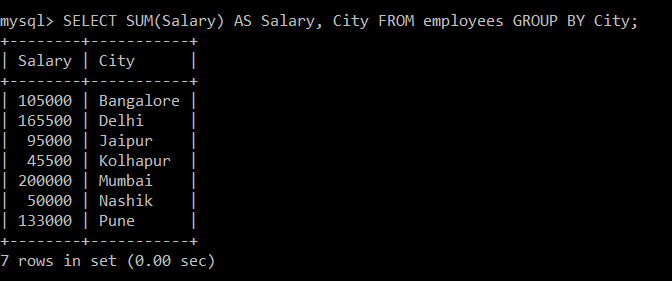


## 3. HAVING CLAUSE:

When we need to place any conditions on the table's column, we use the WHERE clause in SQL. But if we want to use any condition on a column in Group By clause at that time, we will use the HAVING clause with the Group By clause for column conditions.

### Syntax:

TABLENAME **GROUP** **BY** COLUMNNAME **HAVING** CONDITION;



## 4. ORDER BY CLAUSE

Whenever we want to sort anything in SQL, we use the ORDER BY clause. The ORDER BY clause in SQL will help us to sort the data based on the specific column of a table. This means that all the data stored in the specific column on which we are executing the ORDER BY clause will be sorted. The corresponding column values will be displayed in the sequence in which we have obtained the values in the earlier step.

As we all know, sorting means either in ASCENDING ORDER or DESCENDING ORDER. In the same way, ORDER BY CLAUSE sorts the data in ascending or descending order as per our requirement. The data will be sorted in ascending order whenever the **ASC keyword** is used with ORDER by clause, and the **DESC keyword** will sort the records in descending order.

By default, sorting in the SQL will be done using the ORDER BY clause in ASCENDING order if we didn't mention the sorting order.

Before moving towards the example of the ORDER BY clause to sort the records, first, we will look at syntax so it will be easy for us to go through the example.

### Syntax of ORDER BY clause without asc and desc keyword:

**SELECT** COLUMN\_NAME1, COLUMN\_NAME2 **FROM** TABLE\_NAME **ORDER** **BY** COLUMNAME;

### Syntax of ORDER BY clause to sort in ascending order:

**SELECT** COLUMN\_NAME1, COLUMN\_NAME2 **FROM** TABLE\_NAME **ORDER** **BY** COLUMN\_NAME **ASC**;

### Syntax of ORDER BY clause to sort in descending order:

**SELECT** COLUMN\_NAME1, COLUMN\_NAME2 **FROM** TABLE\_NAME **ORDER** **BY** COLUMN\_NAME **DESC**;