Experiment No.3
Create a database using Data Definition Language(DDL) and apply integrity constraints for the specified system
Date of Performance:
Date of Submission:

Experiment No.3

Aim:- Write a query to create tables for each relation in the relational schema of experiment no.2. Apply drop and alter commands on those tables.

Objective:- To learn commands of Data Definition Language(DDL) to create and define databases, and also learn to apply integrity constraints for the specified system.

Theory:

DDL Commands & Syntax:-

Data Definition Language (DDL) is a subset of SQL and a part of DBMS(Database Management System). DDL consist of Commands to commands like CREATE, ALTER, TRUNCATE and DROP. These commands are used to create or modify the tables in SQL. DDL Commands:

- Create
- Alter
- truncate
- drop
- Rename

CREATE:

This command is used to create a new table in SQL. The user must give information like table name, column names, and their data types.

Syntax –CREATE TABLE

table_name (

column_1

datatype,

column_2

datatype,

column_3

datatype,

....

);

ALTER :

This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can add, delete, or modify tasks easily.

Syntax –

ALTER TABLE

table_name ADD

column_name

datatype;

TRUNCATE :

This command is used to remove all rows from the table, but the structure of the table still exists.

Syntax –

TRUNCATE TABLE table_name;

DROP :

This command is used to remove an existing table along with its structure from the Database.

Syntax –

DROP TABLE table_name;

RENAME :

It is possible to change name of table with or without data in it using simple

RENAME command. We can rename any table object at any point of time.

Syntax –

RENAME TABLE <Table Name> To <New_Table_Name>;

Implementation:

```
CREATE TABLE Patients (
   patient_id INT PRIMARY KEY,
   name VARCHAR(100),
   age INT,
   gender VARCHAR(10),
   address VARCHAR(255),
   phone_number VARCHAR(20)
);

-- Create Doctors table
CREATE TABLE Doctors (
   doctor_id INT PRIMARY KEY,
   name VARCHAR(100),
   specialization VARCHAR(100),
   experience_years INT,
   phone_number VARCHAR(20)
);

-- Create Departments table
CREATE TABLE Departments (
   department_id INT PRIMARY KEY,
   name VARCHAR(100),
   head_doctor_id INT,
   FOREIGN KEY (head_doctor_id) REFERENCES Doctors(doctor_id)
);

-- Create Appointments table
CREATE TABLE Appointments (
   appointment_id INT PRIMARY KEY,
   patient_id INT,
   doctor_id INT,
   appointment_date DATE,
   appointment_time TIME,
```

```
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
-- Drop the Patients table
DROP TABLE Patients;

-- Alter the Doctors table, add a new column for email
ALTER TABLE Doctors
ADD COLUMN email VARCHAR(255);

-- Drop the Departments table
DROP TABLE Departments;

-- Alter the Appointments table, add a new column for appointment_status
ALTER TABLE Appointments
ADD COLUMN appointment_status VARCHAR(20);
```

Conclusion:

Q1)Explain the concept of constraints in DDL. How are constraints used to enforce data integrity?

Ans: In Database Management Systems (DBMS), constraints are rules defined on a database schema that enforce the integrity, accuracy, and consistency of data. These constraints are implemented using Data Definition Language (DDL), a subset of SQL (Structured Query Language), specifically used for defining and managing the structure of the database.

Here are some common types of constraints in DDL:

Primary Key Constraint: This constraint ensures that each row in a table is uniquely identified by a specified column or set of columns. It enforces the uniqueness and integrity of the primary key, preventing duplicate or null values.

Foreign Key Constraint: A foreign key constraint establishes a relationship between two tables by ensuring that the values in a column (or a set of columns) in one table match the values in a referenced column (or set of columns) in another table. It helps maintain referential integrity, ensuring that relationships between tables remain valid.

Unique Constraint: Similar to a primary key constraint, a unique constraint ensures that the values in a specified column (or set of columns) are unique across all rows in the table. However, unlike a primary key constraint, it allows null values.

Check Constraint: This constraint specifies a condition that must be satisfied for the values in a column. It allows you to define custom rules to enforce domain integrity, ensuring that only valid data is inserted or updated in the database.

Not Null Constraint: This constraint ensures that a column cannot contain null values. It enforces the requirement that every row must have a value for the specified column, preventing missing or undefined data.

Constraints are essential for enforcing data integrity in a database. They help maintain the consistency, accuracy, and reliability of the data by preventing the insertion of invalid or inconsistent data. When a constraint is defined on a table, the DBMS automatically checks the integrity of the data whenever modifications are made (such as insertions, updates, or deletions). If a modification violates any constraint, the operation is rejected, and an error is raised, ensuring that the database remains in a consistent state.Overall, constraints play a crucial role in database design and management, providing a mechanism to enforce business rules and maintain data quality.

Q2)What is the significance of data types in DDL? Provide examples of commonly used data types in DDL.

Ans: In DDL (Data Definition Language), data types are used to define the type of data that can be stored in each column of a table. Choosing appropriate data types is crucial for database design as it affects the storage requirements, performance, and integrity of the data. Data types define the range of values that can be stored in a column and the operations that can be performed on those values.

Here are some commonly used data types in DDL along with their significance and examples:

Integer: This data type is used to store whole numbers without any decimal points. It is often used for representing counts, quantities, and identifiers.
Decimal/ Numeric: These data types are used for storing fixed-point numbers with decimal precision. They are suitable for financial data, where precision is critical.
Character Strings: Character string data types are used for storing textual data. They can store fixed-length or variable-length strings.
Date and Time: Date and time data types are used for storing date, time, or both. They allow operations such as date arithmetic and date comparisons.

Boolean: This data type is used to store logical values representing true or false. It is useful for representing binary states or conditions.