



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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:

1. Create
2. Alter
3. truncate
4. drop
5. 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,  
....  
);
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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:

Database:

```
1 • create database ATM;
2 • use ATM;
```

Create:

```
4 • CREATE TABLE customer (  
5     customer_id INTEGER PRIMARY KEY,  
6     fname VARCHAR(25),  
7     lname VARCHAR(25),  
8     phoneno INTEGER,  
9     email VARCHAR(25),  
10    password VARCHAR(25)  
11 );
```

Alter:

```
13 • ALTER TABLE customer  
14 ADD address VARCHAR(255);
```

Truncate:

```
18 • TRUNCATE TABLE customer;
```

Rename:

```
22 • RENAME TABLE customer TO new_customer_table;
```

Drop:

```
24 • DROP TABLE new_customer_table;
```

Conclusion:

1. Explain the concept of constraints in DDL. How are constraints used to enforce data integrity?
 1. Constraints in DDL (Data Definition Language) are rules or conditions applied to the database schema to enforce data integrity.
 2. They ensure that data stored in tables adheres to predefined rules, preventing inconsistencies or errors.
 3. Constraints are used to specify requirements for data values, such as uniqueness, validity, or relationships between tables.
 4. By enforcing constraints, DDL ensures the accuracy, consistency, and reliability of data stored in the database.
 5. Common constraints include primary key, foreign key, unique, check, not null, default, and referential integrity constraints.
 6. Primary key constraints uniquely identify each record in a table, while foreign key constraints maintain relationships between tables.
 7. Unique constraints enforce uniqueness of values, check constraints validate data based on specified conditions, and not null constraints require values to be present.
 8. Default constraints provide default values when none are specified, and referential integrity constraints ensure the consistency of relationships between tables.

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

1. Data Type Significance:

- Data types in Data Definition Language (DDL) specify the type of data that can be stored in a database column, ensuring data integrity and optimal storage efficiency.
- They enforce constraints on the values that can be inserted into a column, preventing data inconsistencies and errors.

2. Examples of Commonly Used Data Types:

- Integer: Stores whole numbers, such as age or quantity.
- Varchar: Variable-length character string, suitable for storing text data like names or addresses.
- Decimal: Holds fixed-point numbers with decimal precision, commonly used for monetary values.
- Date: Stores date values, facilitating date-related operations and queries.
- Boolean: Represents true/false or yes/no values, useful for logical operations and conditions.
- Blob: Binary large object, used for storing large binary data like images or documents.

