



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:

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,  
....  
);
```

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.

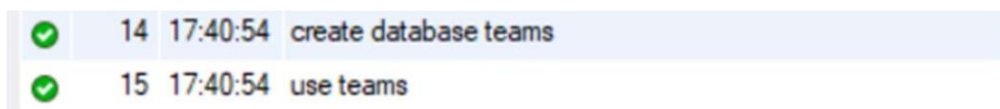
Syntax-

```
RENAME TABLE <Table_Name> To <New_Table_Name>;
```

Implementation:

Database:

```
create database teams;  
use teams;
```



✓	14	17:40:54	create database teams
✓	15	17:40:54	use teams

Table:

```
create table player (  
player_name varchar(20),  
bdate date ,  
age int,  
runs int,  
wickets int,  
player_role varchar (29),  
player_salary varchar(10)  
);
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Result Grid

Alter:

```
alter table player
add player_id varchar(10);
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	player_name	bdate	age	runs	wickets	player_role	player_salary	player_id
--	-------------	-------	-----	------	---------	-------------	---------------	-----------

Result Grid

Truncate:

```
truncate table player;
```

20	17:49:26	truncate table player
----	----------	-----------------------

Rename:

```
rename table player to players;
```

21	17:51:29	rename table player to players
----	----------	--------------------------------

teams
Tables
players
Views
Stored Procedures
Functions

Drop:

```
drop table players;
```

22	17:54:02	drop table players
----	----------	--------------------

teams
Tables
Views
Stored Procedures
Functions

Conclusion:

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

Constraints in DDL (Data Definition Language) are rules or conditions applied to the columns in a database table to enforce data integrity. They ensure that the data stored in the database meets certain criteria, preventing invalid or inconsistent data from being entered. Constraints can enforce various rules such as uniqueness, referential integrity, and data validity. For example, a primary key constraint ensures that each row in a table has a unique identifier, while a foreign key constraint ensures that values in a column match values in another table's column. By enforcing these constraints, data integrity is maintained, ensuring the accuracy, consistency, and reliability of the database.

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

Data types in DDL specify the type of data that can be stored in a column of a database table. They define the format and range of values that can be assigned to a column, ensuring proper storage and manipulation of data. Commonly used data types in DDL include:

INTEGER: Used for storing whole numbers.

VARCHAR(n): Variable-length character string with a maximum length of 'n' characters.

DATE: Used for storing dates in the format 'YYYY-MM-DD'.

DECIMAL(p, s): Exact numeric data type with 'p' total digits and 's' digits to the right of the decimal point.

BOOLEAN: Used for storing true/false or 1/0 values.

TIMESTAMP: Used for storing date and time values in the format 'YYYY-MM-DD HH:MM:SS'.

CHAR(n): Fixed-length character string with a length of 'n' characters.

FLOAT: Approximate numeric data type used for storing floating-point numbers.

TEXT: Variable-length character string for storing large blocks of text. These data types allow database designers to define the structure of the database tables and ensure the appropriate handling of data.