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

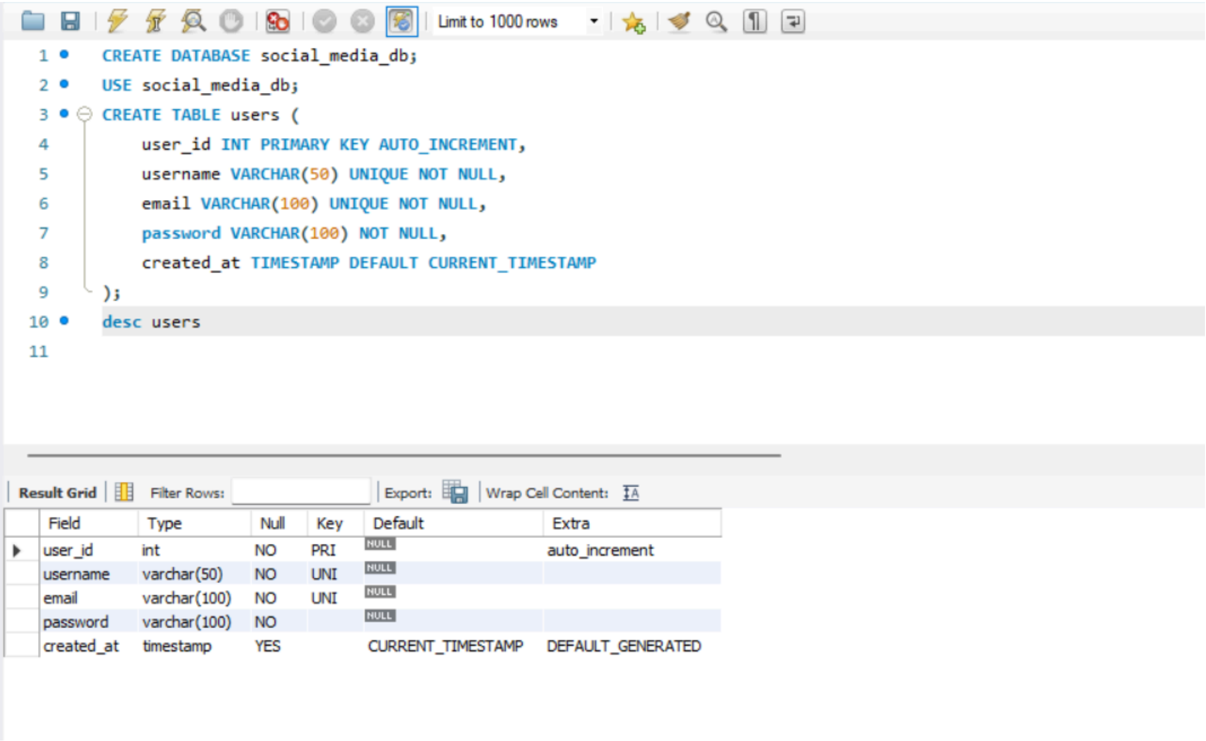
command. We can rename any table object at any point of time.

Syntax –

RENAME TABLE <Table Name> To <New_Table_Name>;

Implementation:

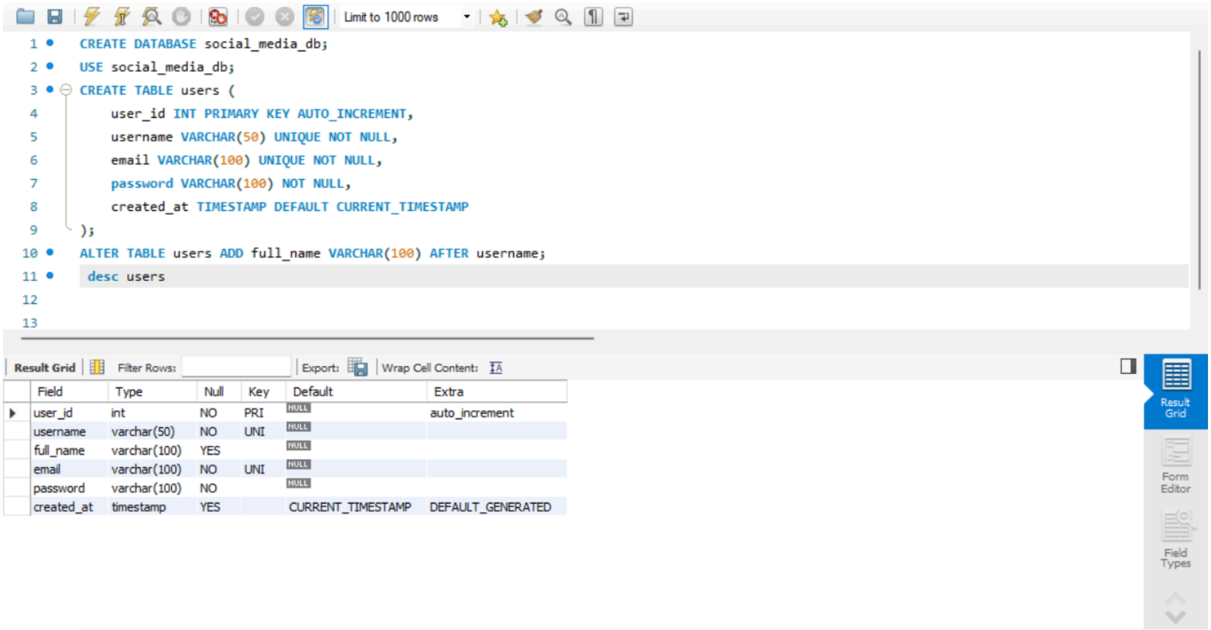
CREATE:



```sql
CREATE DATABASE social_media_db;
USE social_media_db;
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
desc users
```

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	auto_increment
username	varchar(50)	NO	UNI	NULL	
email	varchar(100)	NO	UNI	NULL	
password	varchar(100)	NO		NULL	
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

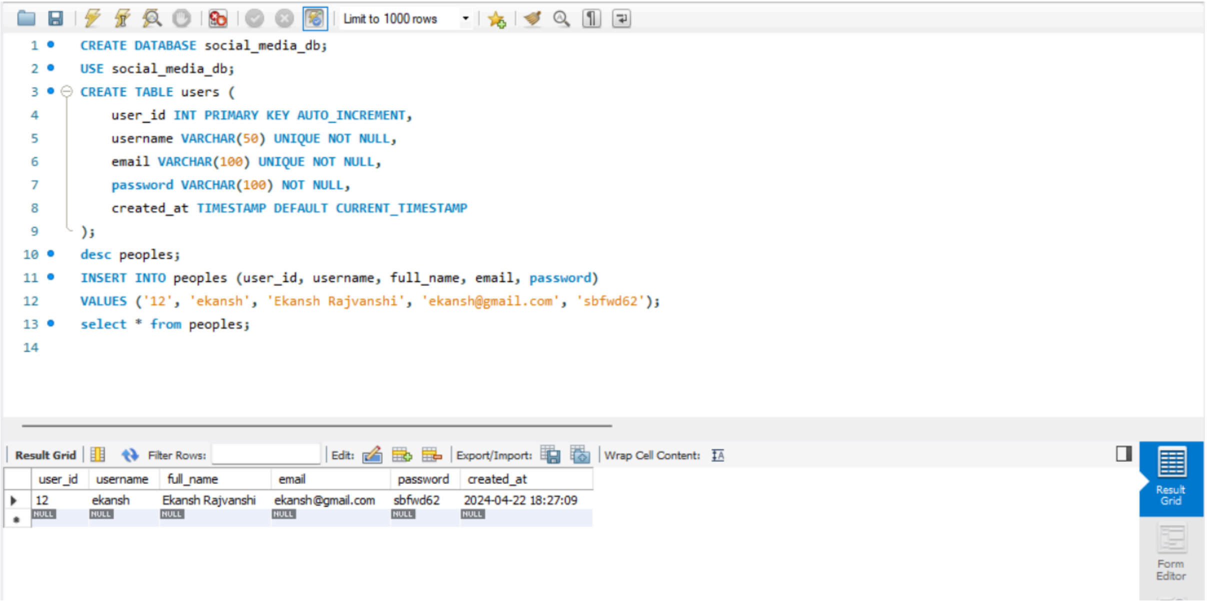
ALTER :



```sql
CREATE DATABASE social_media_db;
USE social_media_db;
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
ALTER TABLE users ADD full_name VARCHAR(100) AFTER username;
 desc users
```

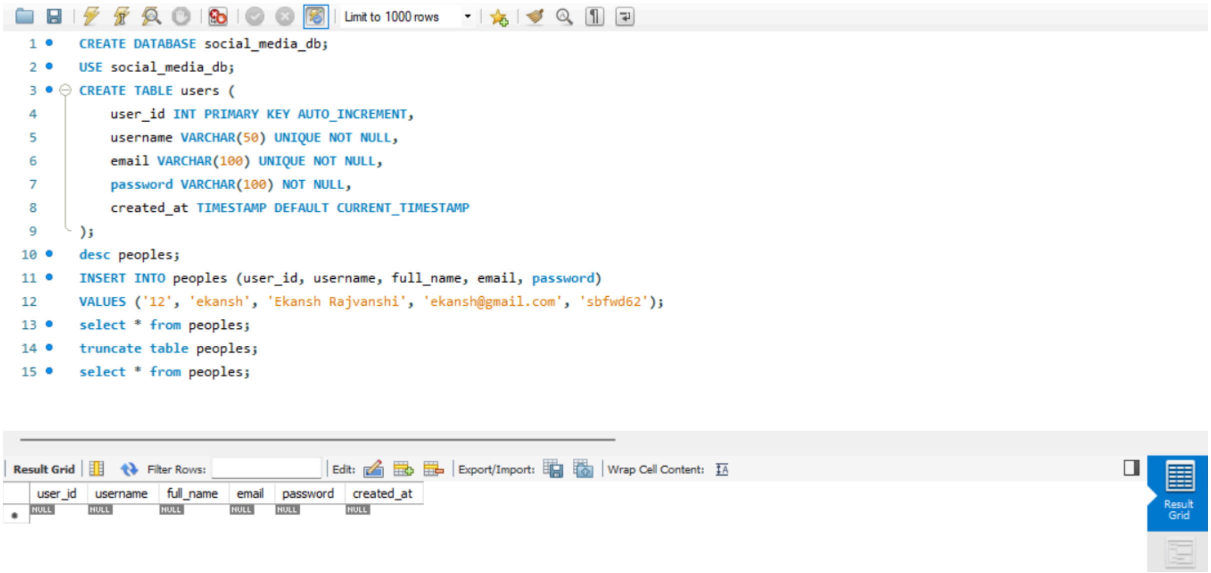
Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	auto_increment
username	varchar(50)	NO	UNI	NULL	
full_name	varchar(100)	YES		NULL	
email	varchar(100)	NO	UNI	NULL	
password	varchar(100)	NO		NULL	
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

TRUNCATE :



```
CREATE DATABASE social_media_db;
USE social_media_db;
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
desc peoples;
INSERT INTO peoples (user_id, username, full_name, email, password)
VALUES ('12', 'ekansh', 'Ekansh Rajvanshi', 'ekansh@gmail.com', 'sbfwd62');
select * from peoples;
```

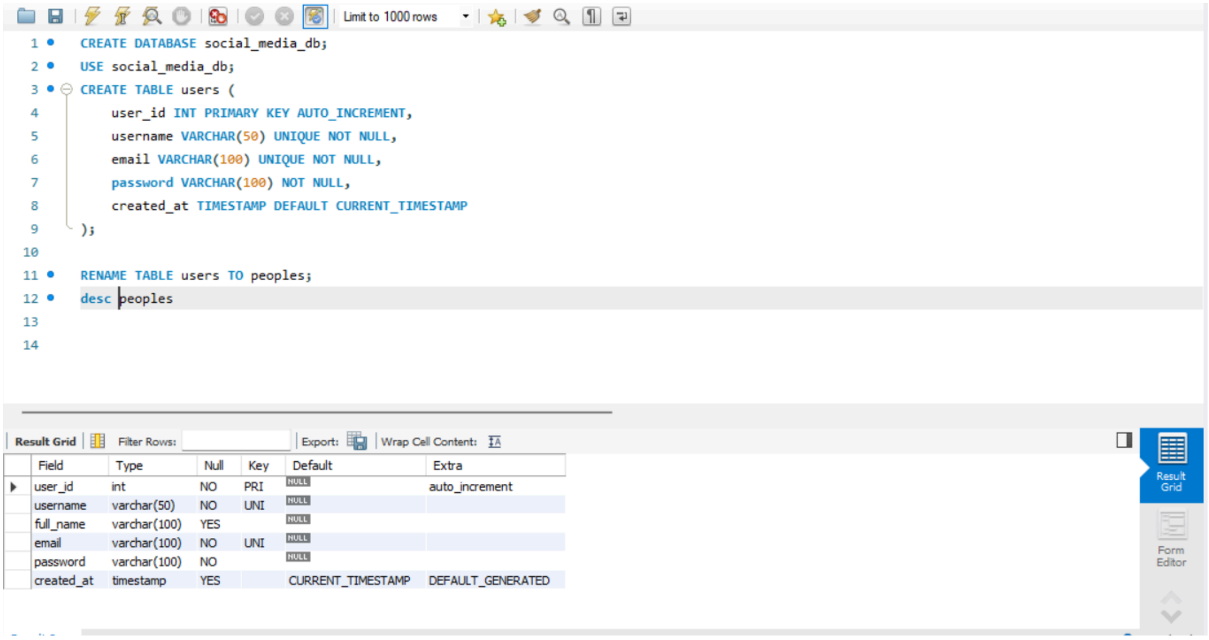
user_id	username	full_name	email	password	created_at
12	ekansh	Ekansh Rajvanshi	ekansh@gmail.com	sbfwd62	2024-04-22 18:27:09
NULL	NULL	NULL	NULL	NULL	NULL



```
CREATE DATABASE social_media_db;
USE social_media_db;
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
desc peoples;
INSERT INTO peoples (user_id, username, full_name, email, password)
VALUES ('12', 'ekansh', 'Ekansh Rajvanshi', 'ekansh@gmail.com', 'sbfwd62');
select * from peoples;
truncate table peoples;
select * from peoples;
```

user_id	username	full_name	email	password	created_at
NULL	NULL	NULL	NULL	NULL	NULL

RENAME :

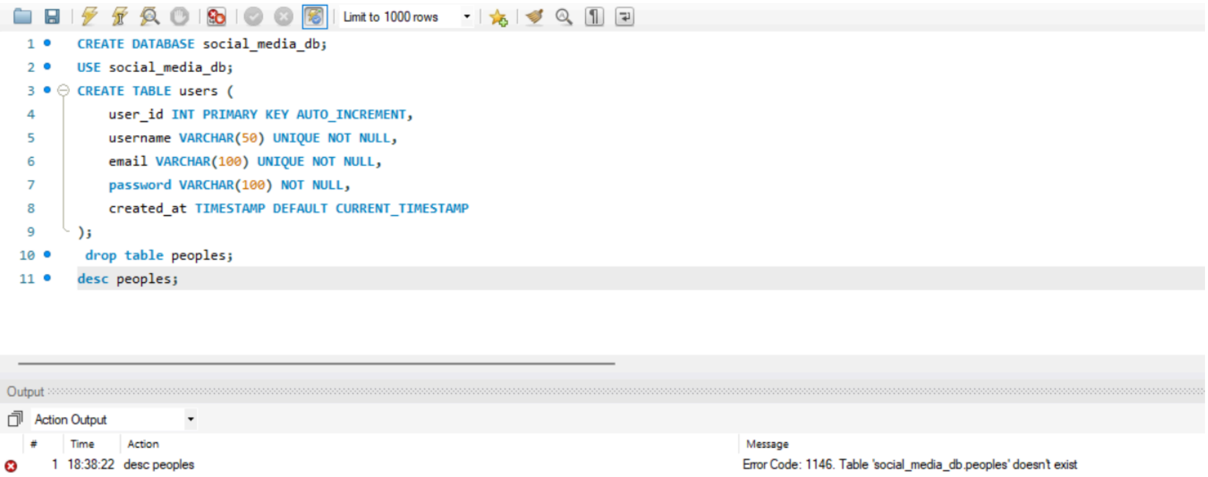


```sql
CREATE DATABASE social_media_db;
USE social_media_db;
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

RENAME TABLE users TO peoples;
desc peoples
```

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	auto_increment
username	varchar(50)	NO	UNI	NULL	
full_name	varchar(100)	YES		NULL	
email	varchar(100)	NO	UNI	NULL	
password	varchar(100)	NO		NULL	
created_at	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

DROP :



```sql
CREATE DATABASE social_media_db;
USE social_media_db;
CREATE TABLE users (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
 drop table peoples;
desc peoples;
```

```
Output
Action Output
#   Time      Action          Message
1   18:38:22  desc peoples    Error Code: 1146. Table 'social_media_db.peoples' doesn't exist
```

Conclusion:

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

   ANS : Constraints in Database Definition Language (DDL) are rules and limitations applied to the data in a database, ensuring data integrity by enforcing certain conditions on the data. These constraints help maintain the accuracy, consistency, and reliability of the data stored in the database. Here's how constraints are used to enforce data integrity: a) Primary Key Constraint: Ensures that each record in a table has a unique identifier, which cannot be null. This prevents duplicate records and provides a unique reference for each row. b) Foreign Key Constraint: Enforces referential integrity by ensuring that values in a column (or set of columns) in one table match values in another table's primary key. It helps maintain consistency between related tables, preventing orphaned records. c) Unique Constraint: Ensures that the values in a column (or set of columns) are unique across the table. It prevents duplicate entries within the specified columns, maintaining data accuracy. d) Check Constraint: Defines a condition that all data in a column must satisfy. It restricts the range of values that can be inserted into a column, ensuring data validity and consistency. e) Default Constraint: Specifies a default value for a column when no explicit value is provided during insertion. It ensures that the column always has a valid value, even if one is not explicitly provided. By applying these constraints, database management systems (DBMS) can automatically enforce data integrity rules, preventing the insertion or modification of data that violates these rules. Constraints act as safeguards against erroneous or inconsistent data, promoting data reliability and accuracy within the database.

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

   ANS : Data types in Database Definition Language (DDL) define the kind of data that can be stored in a column of a table. They specify the format, size, and range of values that can be assigned to a particular attribute, ensuring consistency and accuracy of data storage. The significance of data types lies in their ability to: a) Optimize Storage: Different data types occupy varying amounts of storage space. Choosing appropriate data types can help optimize storage efficiency and reduce disk space usage. b) Enforce Data Integrity: Data types enforce constraints on the values that can be stored in a column, preventing the insertion of invalid or incompatible data. c) Facilitate Data Operations: Data types determine the operations that can be performed on the data, such as arithmetic operations, comparisons, and string manipulations. d) Ensure Data Accuracy: By specifying data types, databases can ensure that only valid data is stored, helping to maintain the integrity and accuracy of the stored information. Examples of commonly used data types in DDL include: a) INTEGER: Used for storing whole numbers (positive or negative) without fractional components. b) VARCHAR(n): Variable-length character string with a maximum length of n characters. c) CHAR(n): Fixed-length character string with a length of exactly n characters. d) DATE: Used for storing date values in the format YYYY-MM-DD. e) TIME: Used for storing time values in the format HH:MM:SS. f) FLOAT: Used for storing floating-point numbers with decimal precision. g) BOOLEAN: Used for storing boolean values (true or false). h) DECIMAL(p, s): Used for storing fixed-point numbers with precision p and scale s. i) BLOB: Used for storing large binary objects, such as images or files. j) CLOB: Used for storing large character strings, such as documents or text files