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;

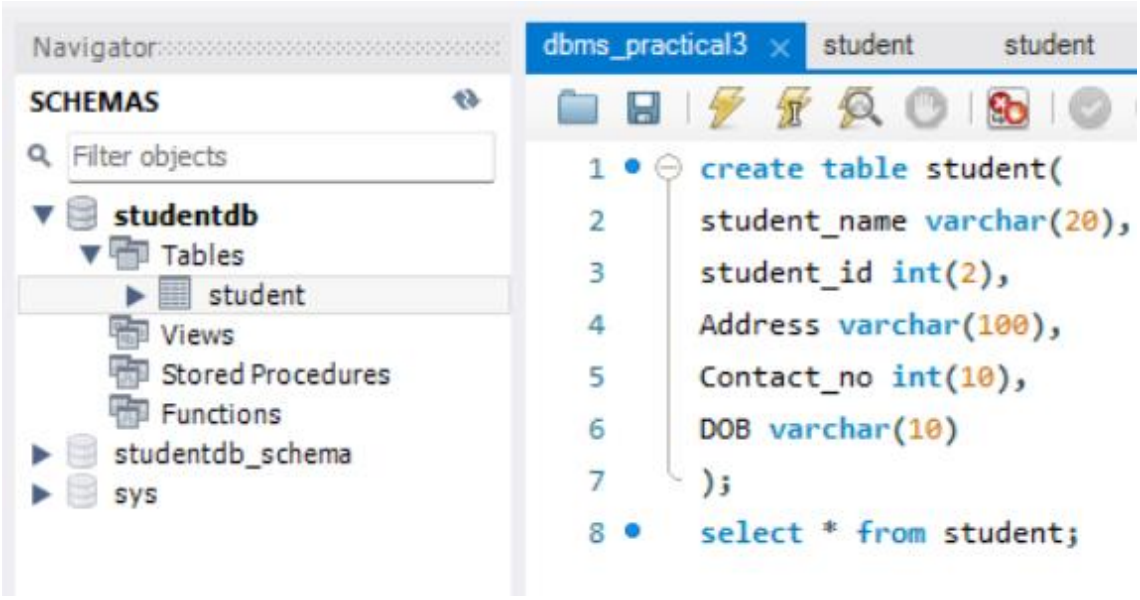
RENAME :
It is possible to change name of table with or without data in it using simple RENAME command. We can rename any table object at any point of time.

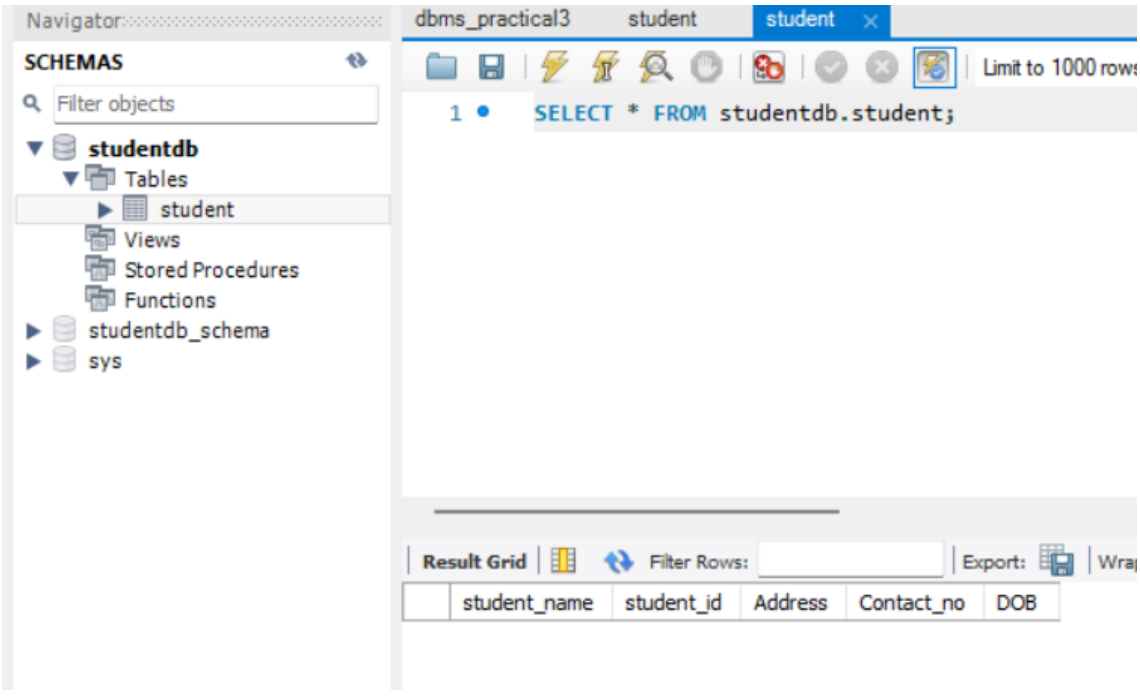
Syntax-

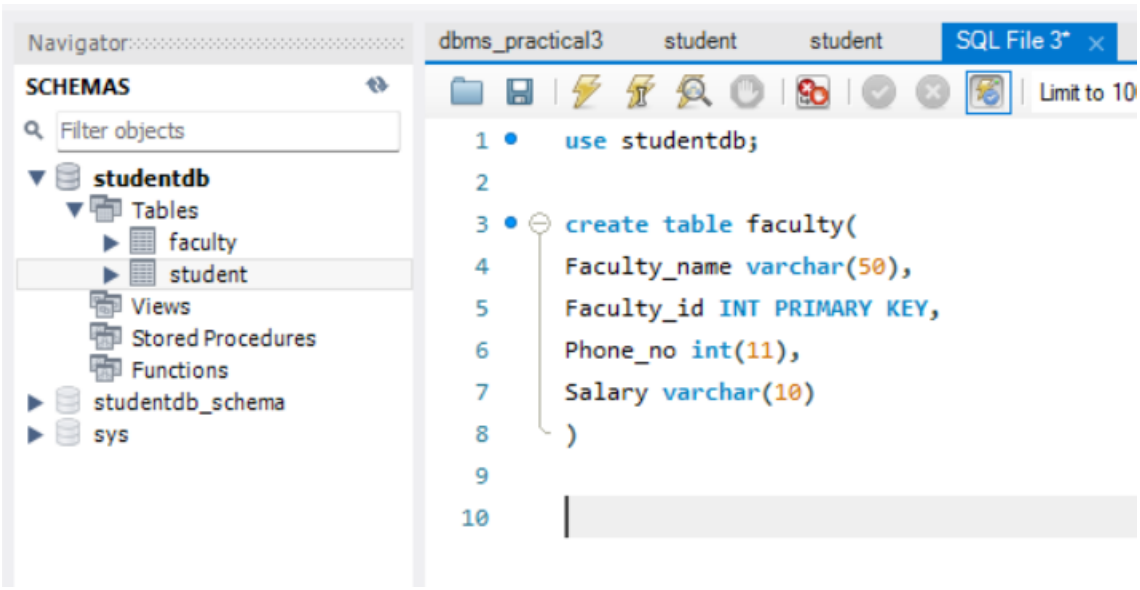
RENAME TABLE <Table_Name> To <New_Table_Name>;

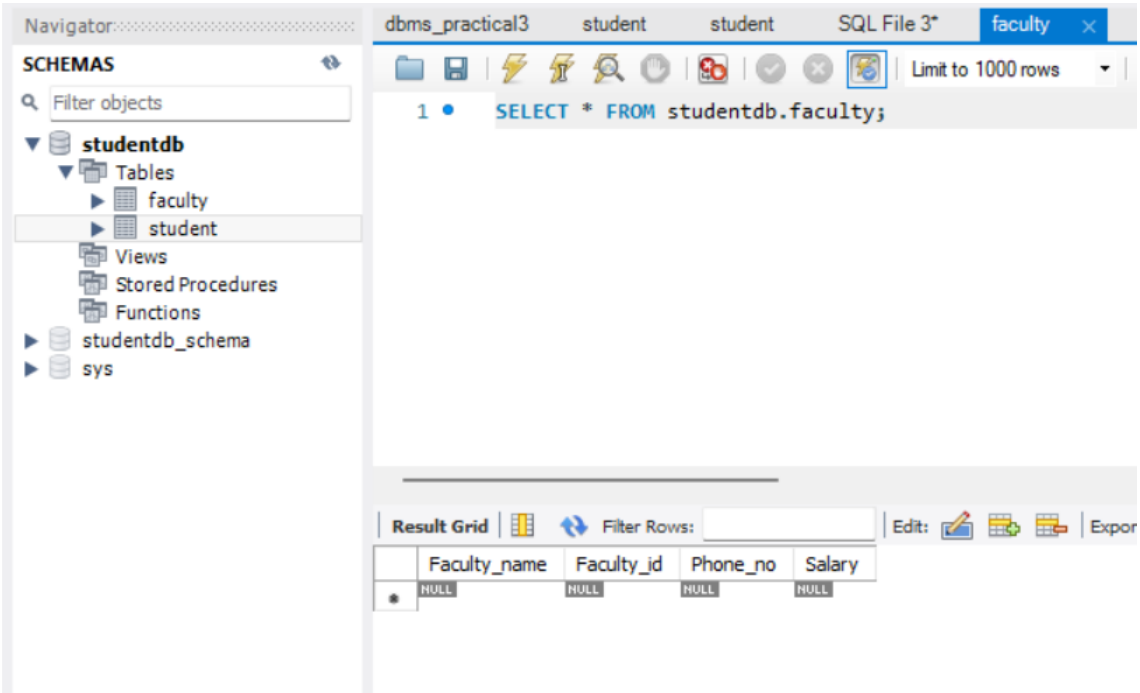
Implementation:

Table:


Navigator
SCHEMAS
Filter objects
studentdb
Tables
student
Views
Stored Procedures
Functions
studentdb_schema
sys
dbms_practical3
student
student
1 create table student(
2 student_name varchar(20),
3 student_id int(2),
4 Address varchar(100),
5 Contact_no int(10),
6 DOB varchar(10)
7);
8 select * from student;


Navigator
SCHEMAS
Filter objects
studentdb
Tables
student
Views
Stored Procedures
Functions
studentdb_schema
sys
dbms_practical3
student
student
Limit to 1000 rows
1 SELECT * FROM studentdb.student;
Result Grid
Filter Rows:
Export:
student_name
student_id
Address
Contact_no
DOB


Navigator
SCHEMAS
Filter objects
studentdb
Tables
faculty
student
Views
Stored Procedures
Functions
studentdb_schema
sys
dbms_practical3
student
student
SQL File 3*
Limit to 10
use studentdb;
create table faculty(
Faculty_name varchar(50),
Faculty_id INT PRIMARY KEY,
Phone_no int(11),
Salary varchar(10)
)


Navigator
SCHEMAS
Filter objects
studentdb
Tables
faculty
student
Views
Stored Procedures
Functions
studentdb_schema
sys
dbms_practical3
student
student
SQL File 3*
faculty
Limit to 1000 rows
SELECT * FROM studentdb.faculty;
Result Grid
Filter Rows:
Edit:
Faculty_name
Faculty_id
Phone_no
Salary
NULL

Alter:

dbms_practical3 | student | student | SQL File 3* | SQL File 4* | faculty

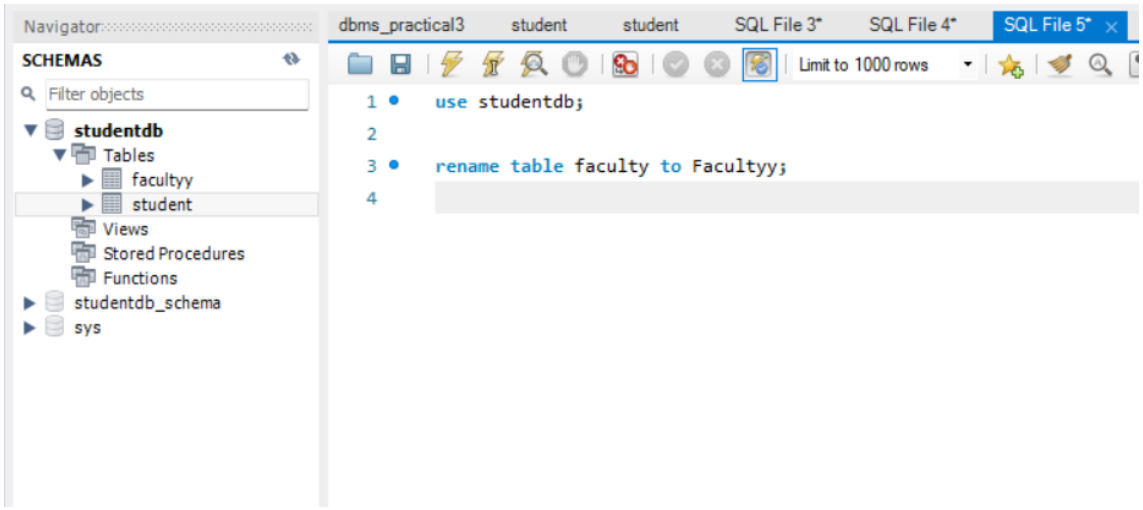
Limit to 1000 rows

```
1  use studentdb;
2  alter table faculty
3  add column email varchar(50);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

Faculty_name	Faculty_id	Phone_no	Salary	email
NULL	NULL	NULL	NULL	NULL

Rename:



Conclusion:

Constraints in Data Definition Language (DDL) are rules applied to the data in a database table to enforce data integrity. They ensure that the data stored in the database meets certain criteria or conditions. Constraints can enforce uniqueness, referential integrity, and data validity. By defining constraints, such as primary key, unique, foreign key, and check constraints, developers can maintain the accuracy, consistency, and reliability of the data stored in the database. Constraints prevent the insertion, deletion, or modification of data that would violate the specified rules, thereby ensuring data integrity.

Data types in Data Definition Language (DDL) define the type of data that can be stored in each column of a database table. They specify the format and range of values that can be assigned to a column, ensuring data integrity and efficient storage.

The significance of data types in DDL lies in:

1. Data Integrity: Data types enforce constraints on the values that can be stored in a column, preventing invalid or inappropriate data from being inserted.

2. Storage Efficiency: Different data types require different amounts of storage space. By choosing appropriate data types, database designers can optimize storage efficiency and performance.

3. Data Validation: Data types help validate the format and range of values entered into a column, ensuring consistency and accuracy of the data.

Commonly used data types in DDL include:

1. INTEGER/INT: Used for whole numbers without decimal points, such as 1, 10, -5.

2. VARCHAR(n): Variable-length character strings with a maximum length of 'n'. For example, VARCHAR(50) can store strings of up to 50 characters.

3. CHAR(n): Fixed-length character strings with a length of 'n'. Unused space is padded with spaces. For example, CHAR(10) can store strings of exactly 10 characters.

4.DECIMAL(p, s): Fixed-point numeric data type, where 'p' specifies the total number of digits and 's' specifies the number of digits after the decimal point. For example, DECIMAL(10, 2) can store numbers with up to 10 digits, 2 of which are after the decimal point.

5. DATE: Used to store date values in YYYY-MM-DD format.

6. BOOLEAN: Represents true or false values.

These are just a few examples of commonly used data types in DDL. The choice of data type depends on the nature of the data being stored and the requirements of the application.