**Batch:C1 Roll No.:16010122221**

**Experiment / assignment / tutorial No. 3**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Title: :** Implementation of Database in SQL -DDL |

**Objective:** Define/modify database definitions with proper constraints

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Expected Outcome of Experiment:**

CO 2: Convert entity-relationship diagrams into relational tables, populate a relational

database and formulate SQL queries on the data Use SQL for creation and query the database.

CO 3: Define and apply integrity constraints and improve database design using normalization techniques.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pre Lab/ Prior Concepts:**

**Pre Lab/ Prior Concepts**:

Resources used: Postgresql

**Theory:** The set of relations in a database must be specifies to the system by means of a data definition language (DDL). The SQL DDL allows specification of not only a set of relations but also specific information about the relation including,

1. The schema for each relation
2. The domain of values associated with each attribute
3. The integrity constraints
4. The set of indices to be maintained for each relation
5. The security and authorization information for each relation
6. The physical storage structure of each relation on disk

**Syntax Create Table:**

create table employee(ssn,fname varchar(10), mname varchar(10), lname varchar(10), desg varchar(20), gender varchar(5), addr varchar(20), bdate datetime, sal float,primary key(ssn));

create table manages(ssn int, dept\_code int, start\_dt datetime, foreign key(ssn)

create table manages(ssn int, dept\_code int, start\_dt datetime, foreign key(ssn)

references employee, foreign key(dept\_code) refrences department, key(ssn,dept\_code) ) on delete set null;primary

**Data Constraints**

Business managers of the organization determine a set of rules that must be applied before the data is stored in the database. The application of such rules on raw data ensures **data integrity**.

**Eg:-** An employee belonging to the Sales department cannot have a salary higher than Rs. 1000.

An employee has an unique identification number.

**Applying Data Constraints**

Oracle permits data constraints to be attached to table columns using SQL syntax. Constraints can be attached to table columns using

Alter table

**Unique Constraint**

**Unique Constraint- At column level Syntax**

**<ColumnName><Datatype>(<size>)**

**UNIQUE Unique Constraint- At table level**

**CREATE TABLE<TableName>(**

**<ColumnName><Datatype>(<size>)**

**<ColumnName><Datatype>(<size>)**

**<Columnname><Datatype>(<size>) UNIQUE(<ColumnName1>,<ColumnName2>);**

**Implementation Details (Problem Statement, Query and Screenshots of Results):**

create table Users(

userId int primary key,

email varchar NOT NULL,

username varchar NOT NULL,

hashPass varchar NOT NULL

);

create table Blogs(

blogID int,

userId int,

title varchar NOT NULL,

contents varchar NOT NULL,

blogType varchar NOT NULL,

primary key(blogID, userID)

);

create table UserComments(

commentID int,

contents varchar NOT NULL,

userId int,

blogID int,

primary key(commentID,userID,blogID)

);

create table Roles(

roleId int,

userId int,

roleName varchar NOT NULL,

roleDescription varchar,

primary key(roleId, userID)

);

-- drop table Users;

-- drop table Blogs;

-- drop table UserComments;

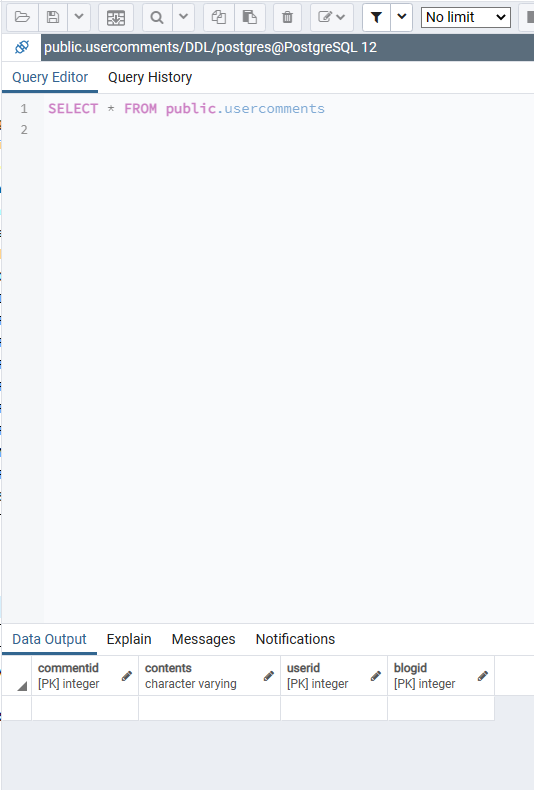
-- drop table Roles;

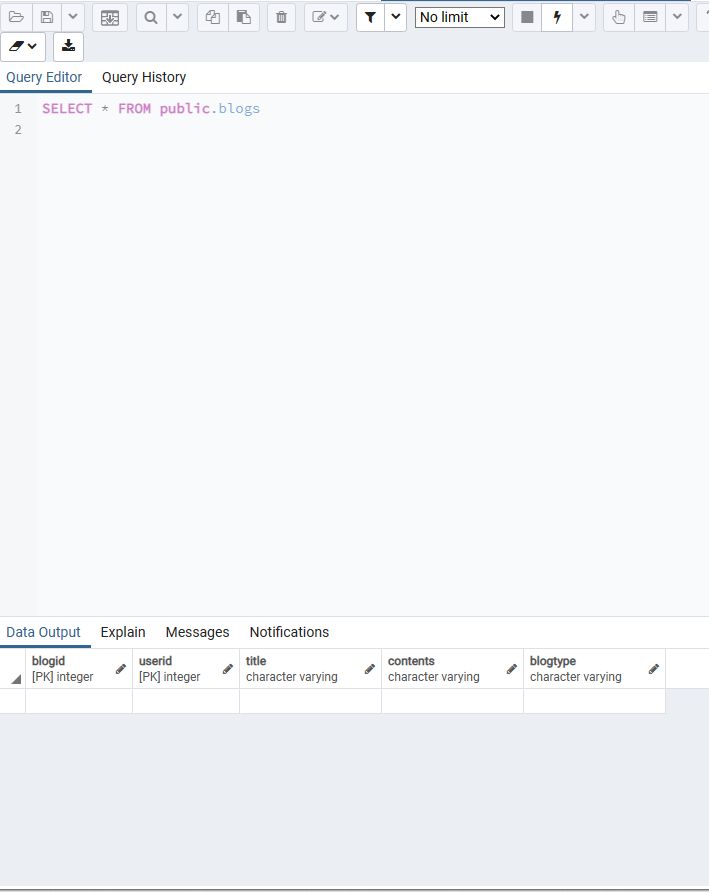
alter table UserComments add constraint fkey1 foreign key(userID) references Users(userID);

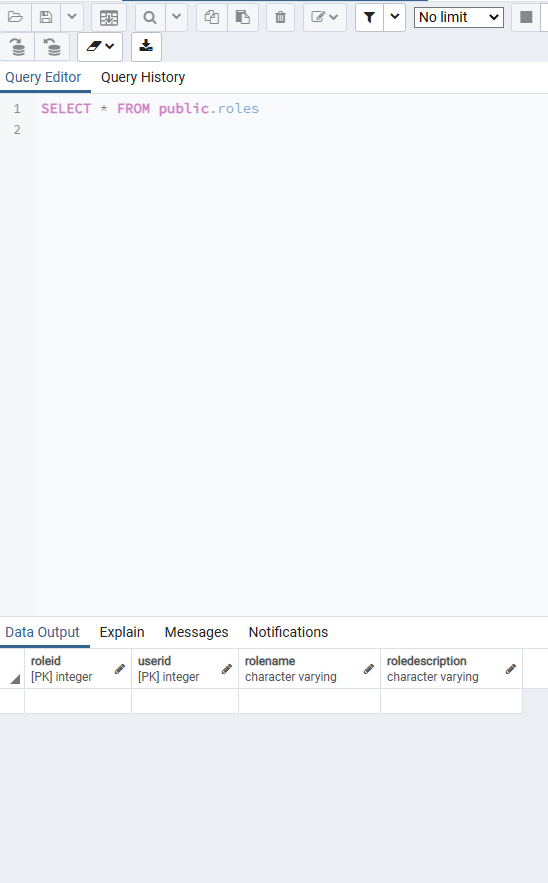
alter table UserComments add constraint fkey2 foreign key(blogID) references Blogs(blogID);

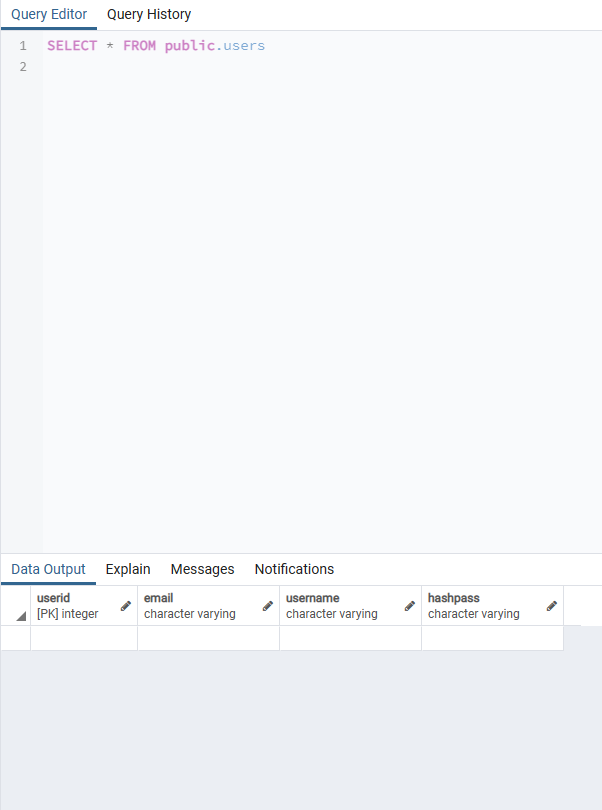
alter table Roles add constraint fkey3 foreign key(userID) references Users(userID);

alter table Blogs add constraint fkey4 foreign key(userID) references Users(userID);









**Post Lab Questions:**

1. **Explain in brief the following terms:**

a).Database:

- A database is a structured collection of data organized for efficient retrieval, storage, and manipulation. It acts as a central repository for storing and managing various types of data, allowing users to easily access, update, and analyze information. Databases are used in various applications such as websites, mobile apps, enterprise systems, and more.

b). Types of Databases:

Relational Databases: Organize data into tables with rows and columns, and establish relationships between them. Examples include MySQL, PostgreSQL, Oracle, and SQL Server.

NoSQL Databases: Designed for handling unstructured or semi-structured data. They are schema-less and offer flexible data models. Examples include MongoDB, Cassandra, Redis, and Couchbase.

Graph Databases: Specifically designed for dealing with data whose relations are best represented as graphs. Examples include Neo4j, Amazon Neptune, and OrientDB.

Document Databases: Store data in flexible, JSON-like documents. Suitable for content management systems, real-time analytics, and other applications. Examples include MongoDB, CouchDB, and Elasticsearch.

Time-Series Databases: Optimized for handling time-series data, such as IoT sensor data, financial data, or log data. Examples include InfluxDB, Prometheus, and TimescaleDB.

C).SQL Data Types:

SQL (Structured Query Language) data types define the type of data that can be stored in a column of a database table. Common SQL data types include:

Numeric Types: Integers (e.g., INT, SMALLINT, BIGINT), floating-point numbers (e.g., FLOAT, REAL, DOUBLE PRECISION), and fixed-point numbers (e.g., DECIMAL, NUMERIC).

Character Types: Used for storing text data, including fixed-length (CHAR) and variable-length (VARCHAR) character strings.

Date/Time Types: Used for storing date and time values, including DATE, TIME, DATETIME, and TIMESTAMP.

Boolean Type: Represents true or false values.

Binary Types: Store binary data, such as images or files, using types like BLOB (Binary Large Object) or BYTEA.

Specialized Types: Various databases offer specialized types for handling specific data, such as arrays, JSON, XML, spatial data types, etc.

2.What are the different commands to:

Here are the commands for each of the tasks you mentioned:

a. Delete an entire table:

To delete an entire table in SQL, you can use the `DROP TABLE` command followed by the table name. Be very careful with this command as it permanently deletes the table and all of its data.

Example:

sql

DROP TABLE table\_name;

b. To view a database:

In SQL, you typically don't "view" a database directly. Instead, you connect to a database management system (DBMS) and interact with the databases within it. To list all available databases in a DBMS, the command varies depending on the specific DBMS you're using. Here are a few examples:

- MySQL: `SHOW DATABASES;`

- PostgreSQL: `\l`

- SQL Server: `SELECT name FROM sys.databases;`

c. To select & view all the columns:

To select and view all columns from a table in SQL, you use the `SELECT` statement without specifying particular column names. Instead, you use the wildcard character `\*` to select all columns.

Example:

sql

SELECT \* FROM table\_name;

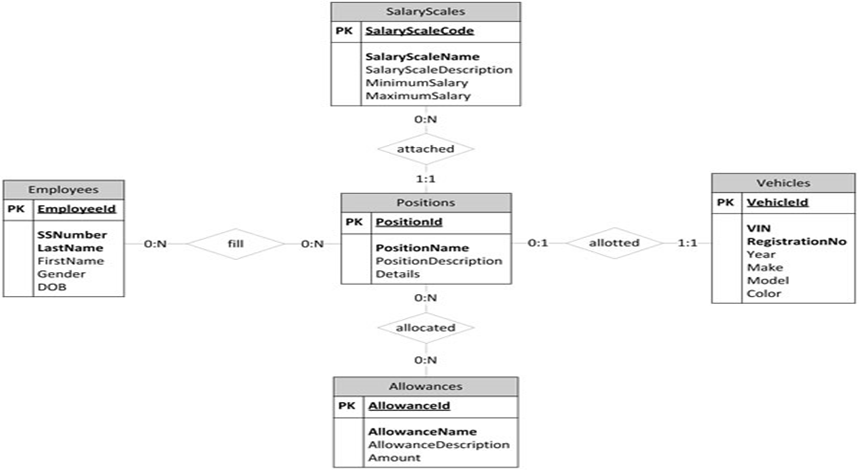
This will retrieve and display all columns from the specified table.

3.

For the given ER model, using DDL command: Write syntax to create CREATE Tables with all possible integrity constraints.

# Problem Statement:

# A small accounting firm wants a simple HR application that will help it to keep track of its employees, their positions, allowances, salary scales, and which company vehicles their employees drive. The application must keep track of all the positions at the firm, the employees filling these positions, the allowances for these positions, the salary scales for these positions, and the company vehicles assigned to these positions.



CREATE TABLE position (

position\_id INT PRIMARY KEY,

position\_title VARCHAR(50) NOT NULL,

salary\_scale DECIMAL(10,2) NOT NULL,

UNIQUE(position\_title)

);

CREATE TABLE employee (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

position\_id INT NOT NULL,

FOREIGN KEY (position\_id) REFERENCES position(position\_id) ON UPDATE CASCADE ON DELETE RESTRICT

);

CREATE TABLE allowance (

allowance\_id INT PRIMARY KEY,

allowance\_name VARCHAR(50) NOT NULL,

allowance\_amount DECIMAL(10,2) NOT NULL,

UNIQUE(allowance\_name)

);

CREATE TABLE employee\_allowance (

employee\_id INT NOT NULL,

allowance\_id INT NOT NULL,

amount DECIMAL(10,2) NOT NULL,

PRIMARY KEY (employee\_id, allowance\_id),

FOREIGN KEY (employee\_id) REFERENCES employee(employee\_id) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (allowance\_id) REFERENCES allowance(allowance\_id) ON UPDATE CASCADE ON DELETE CASCADE

);

CREATE TABLE vehicle (

vehicle\_id INT PRIMARY KEY,

make VARCHAR(50) NOT NULL,

model VARCHAR(50) NOT NULL,

year INT NOT NULL

);

CREATE TABLE position\_vehicle (

position\_id INT NOT NULL,

vehicle\_id INT NOT NULL,

PRIMARY KEY (position\_id, vehicle\_id),

FOREIGN KEY (position\_id) REFERENCES position(position\_id) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (vehicle\_id) REFERENCES vehicle(vehicle\_id) ON UPDATE CASCADE ON DELETE CASCADE

);

**Conclusion:**

This experiment enabled a systematic understanding of Data requirements of application and EER diagram design, laying the groundwork for effective and optimized database development