

**Experiment No.: 03**

**Title:** To implement database for relational model in Experiment no. 2 using DDL statements.

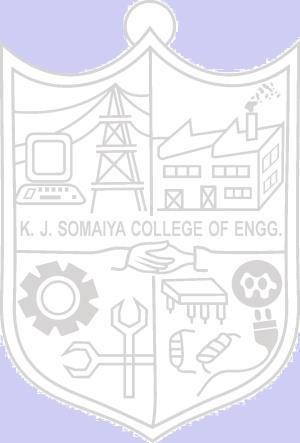
(Autonomous College Affiliated to University of Mumbai)

# Batch:A3 Roll No.:16010421073 Experiment No.: 03

**Aim:** To implement database for relational model in experiment no. 2 using DDL statements (Virtual Lab).

**Resources needed:** PostgreSQL PgAdmin3

# Theory:



The Data Definition Language (DDL) is used to create and modify the relational schema. Also it is used to add various constraints to the table like the primary key, foreign key, check constraint, not null constraint and unique constraint.

The DDL statements are: CREATE

DROP ALTER

PostgreSQL supports the standard SQL types int, smallint, real, double precision, char(N), varchar(N), date, time, timestamp, and interval for creating tables.

# Procedure:

**Create Database and use it:**

$ createdb mydb

$ psql mydb

**Delete a database:** $ dropdb mydb

# Create table:

CREATE TABLE my\_first\_table ( first\_column text,

second\_column integer

);

CREATE TABLE products ( product\_no integer,

name text, price numeric);

# Drop Table:

DROP TABLE my\_first\_table; DROP TABLE products;

# Default Value:

CREATE TABLE products ( product\_no integer,

name text,

price numeric **DEFAULT 9.99** );

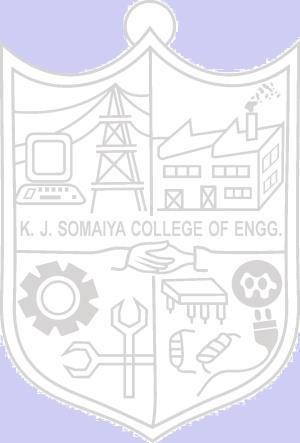
# Constraints:

1. **Primary Key**

CREATE TABLE products ( product\_no integer **PRIMARY KEY**, name text,

price numeric );

Primary keys can also constrain more than one column. CREATE TABLE example (

a integer, b integer, c integer,

# PRIMARY KEY (a, c)

);

# Check Constraint

CREATE TABLE products ( product\_no integer,

name text,

price numeric **CHECK (price** > **0)** );

# Not Null Constraint

CREATE TABLE products ( product\_no integer **NOT NULL**, name text **NOT NULL**,

price numeric );

1. **Unique Constraint** CREATE TABLE products ( product\_no integer **UNIQUE**,

name text,

price numeric );

# Foreign Key Constarint

CREATE TABLE products ( product\_no integer PRIMARY KEY, name text,

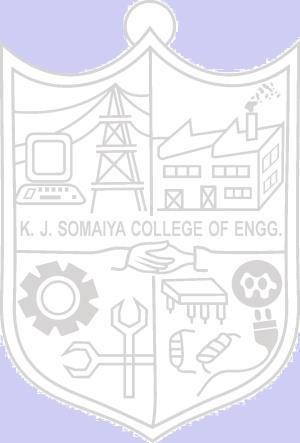
price numeric );

CREATE TABLE orders ( order\_id integer PRIMARY KEY,

product\_no integer **REFERENCES products (product\_no)**, quantity integer );

Here a foreign key constraint in the order table references the products table.

# Modifying table:

**Adding column**

ALTER TABLE products ADD COLUMN description text;

# Removing column

ALTER TABLE products DROP COLUMN description;

# Adding Constraint

ALTER TABLE products ADD CONSTRAINT some\_name UNIQUE (product\_no); ALTER TABLE products ADD FOREIGN KEY (product\_group\_id) REFERENCES product\_groups;

# Removing Constraint

ALTER TABLE products DROP CONSTRAINT some\_name;

# Adding Not Null Constraint

ALTER TABLE products ALTER COLUMN product\_no SET NOT NULL;

# Removing Not Null Constraint

ALTER TABLE products ALTER COLUMN product\_no DROP NOT NULL;

# Results: (Queries printout with output)

# Query output(Hotel Mangement system)

**CREATE TABLE** ROOMS(

Room\_ID int **PRIMARY KEY**,

Room\_Type text,

Room\_No int **NOT NULL**,

Room\_Cat text,

**check** (Room\_No <= 1000 )

);

**SELECT \* FROM** ROOMS;

**DROP TABLE** ROOMS;

**CREATE** **TABLE** BOOKING(

Booking\_ID int **PRIMARY KEY**,

Book\_Type text **NOT** NULL,

Start\_Date DATE **NOT** NULL,

End\_Date DATE **NOT** NULL

);

**SELECT \* FROM** BOOKING;

**DROP TABLE** BOOKING;

**CREATE TABLE** PAYMENT(

Pay\_ID int **PRIMARY KEY**,

Pay\_Cust\_ID int **UNIQUE NOT** NULL,

Pay\_Amt int **NOT** NULL

);

**SELECT \* FROM** PAYMENT;

**DROP TABLE** PAYMENT;

**CREATE TABLE** SERVICES(

Service\_ID int **PRIMARY KEY**,

SVCS\_Type text,

SVCS\_Desc text,

SVCS\_Name text,

Room\_ID int **REFERENCES** ROOMS(Room\_ID)

);

**SELECT \* FROM** SERVICES;

**DROP TABLE** SERVICES;

**CREATE TABLE** LOGIN(

Log\_ID int **PRIMARY KEY**,

Log\_Name text **NOT** NULL,

Log\_Pass text **NOT** NULL,

Mob\_no int,

Add\_ress text,

Email text

);

**SELECT \* FROM** LOGIN;

**DROP** TABLE LOGIN;

**CREATE TABLE** Customer(

Cust\_ID int **PRIMARY KEY NOT** NULL,

Cust\_Usr text **NOT** NULL,

Cust\_Pass text **NOT** NULL,

Cust\_Name int,

Cust\_Mob\_No text **NOT** NULL,

Cust\_Email text,

Cust\_Add text **NOT** NULL,

Room\_ID int **REFERENCES** ROOMS(Room\_ID),

Booking\_ID int **REFERENCES** BOOKING(Booking\_ID),

Pay\_ID int **REFERENCES** PAYMENT(Pay\_ID),

Service\_ID int **REFERENCES** SERVICES(Service\_ID),

Log\_ID int **REFERENCES** LOGIN(Log\_ID)

);

**SELECT\* FROM** Customer;

**DROP TABLE** Customer;

**CREATE TABLE** HOTEL(

Hot\_ID int **PRIMARY KEY NOT** NULL,

Hot\_name text **NOT** NULL,

Hot\_Add text **NOT** NULL,

Hot\_Rent int,

Hot\_Desc text,

**check** (Hot\_ID <= 1000),

Room\_ID int **REFERENCES** ROOMS(Room\_ID),

Booking\_ID int **REFERENCES** BOOKING(Booking\_ID),

Pay\_ID int **REFERENCES** PAYMENT(Pay\_ID),

Service\_ID int **REFERENCES** SERVICES(Service\_ID),

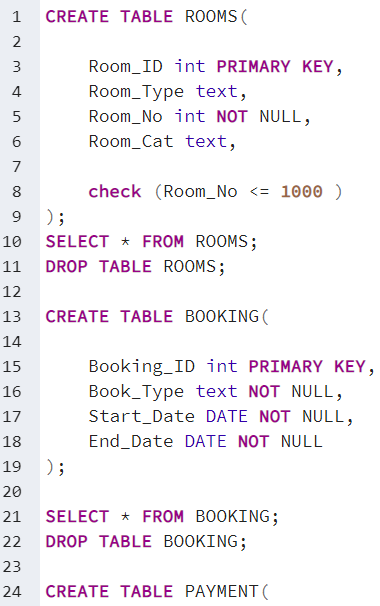
Log\_ID int **REFERENCES** LOGIN(Log\_ID),

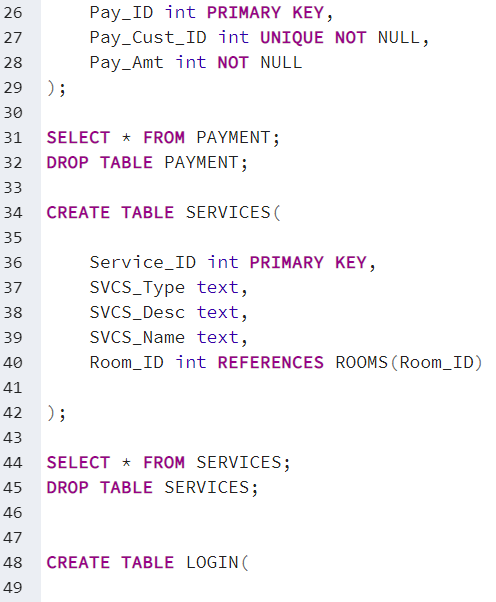
Cust\_ID int **REFERENCES** Customer(Cust\_ID)

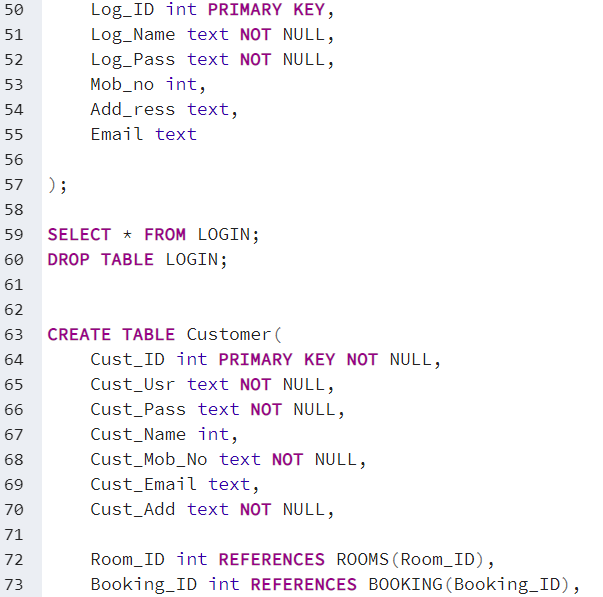
);

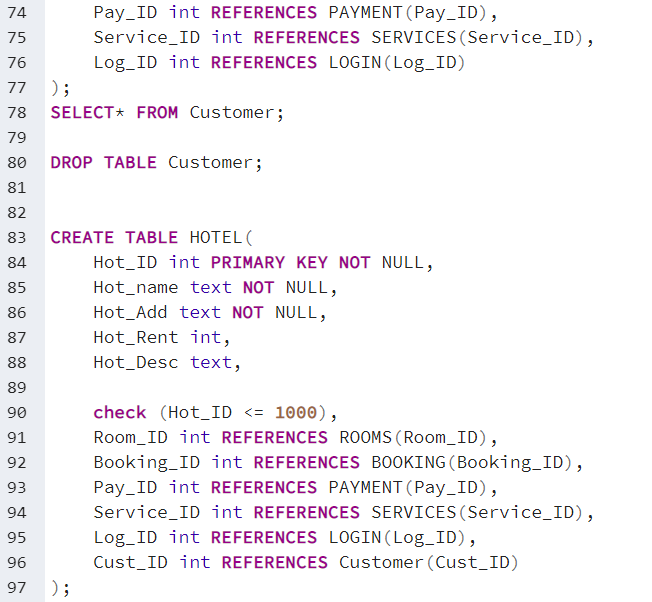
**SELECT\* FROM** HOTEL;

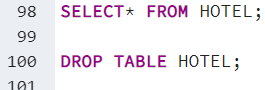
**DROP TABLE** HOTEL;











# OUTPUT

# 

# 

# 

# 

# 

# 

# 

**Outcomes:**

**CO3:** Illustrate the concept of security, Query processing, indexing and Normalization for Relational database.

**Questions:**

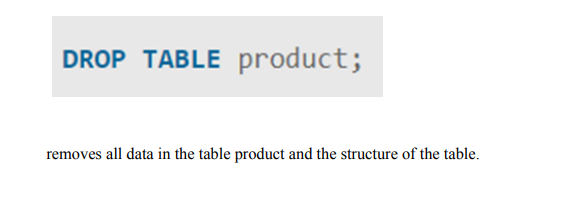
**Q1 what is difference between Truncate, Drop and delete? Explain with example**

The DELETE command is used to remove some or all rows from a table. A WHERE clause can be used to only remove some rows. If no WHERE condition is specified, all rows will be removed.

TRUNCATE removes all rows from a table. The operation cannot be rolled back and no triggers will be fired. As such, TRUNCATE is faster and doesn't use as much undo space as a DELETE.

The DROP command removes a table from the database. All the tables' rows, indexes and privileges will also be removed. The operation cannot be rolled back.

****

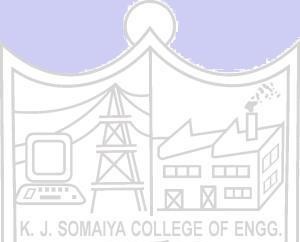
****

**Conclusion:**

**We learned about how to create and modify the relational schema and also to add various constraints to the table like the primary key, foreign key, check constraint, not null constraint and unique constraint by implementing various DDL statements like create,drop and alter.**

**Grade: AA / AB / BB / BC / CC / CD /DD Signature of faculty in-charge with date**

**Reference books:**

(Autonomou of Mumbai)

s College Affiliated to University

1. Elmasri and Navathe, “Fundamentals of Database Systems”, 6th Edition, Pearson Education
2. Korth, Slberchatz,Sudarshan, :”Database System Concepts”, 6th Edition, McGraw – Hill.

# WebSite:

1. <http://www.tutorialspoint.com/postgresql/>
2. <http://sage.virtual-labs.ac.in/home/pub/21/>

(Autonomous College Affiliated to University of Mumbai)