

(Autonomous College Affiliated to University of Mumbai)

# Batch:A3 Roll No.:16010421075 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** );

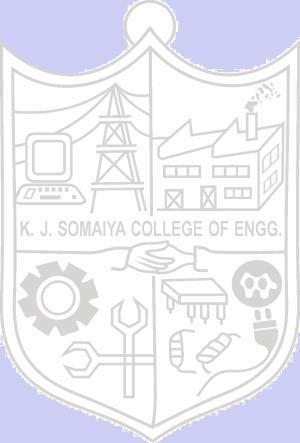
# Constraints:

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,

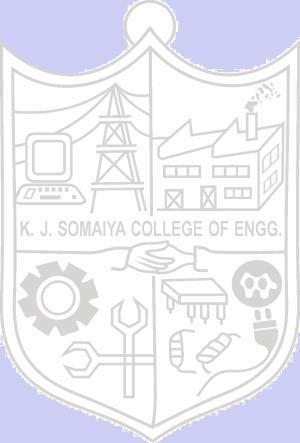
price numeric );

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(Airline Management System)

**CREATE TABLE AIRPORT(**

**Airport\_Name text PRIMARY KEY NOT NULL,**

**Country text NOT NULL,**

**City text NOT NULL,**

**Statee text NOT NULL**

**);**

**drop table AIRPORT**

**select\*from AIRPORT**

**CREATE TABLE COMPANY(**

**Company\_ID text PRIMARY KEY NOT NULL,**

**Company\_Name text NOT NULL**

**);**

**drop table COMPANY**

**select\*from COMPANY**

**CREATE TABLE PASSENGER(**

**Passenger\_ID INT PRIMARY KEY NOT NULL,**

**Passenger\_Name text NOT NULL,**

**Passenger\_Age int NOT NULL,**

**Passenger\_Address text NOT NULL,**

**Passenger\_Gender char(1)**

**CHECK (Passenger\_Gender='F' or Passenger\_Gender='M'),**

**Passenger\_DOB text NOT NULL,**

**Passenger\_Phone\_Number char(10) NOT NULL**

**CHECK (Passenger\_Phone\_Number like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')**

**);**

**drop table PASSENGER**

**select \* from PASSENGER**

**CREATE TABLE EMPLOYEE(**

**Employee\_ID INT PRIMARY KEY NOT NULL,**

**Employee\_Name text NOT NULL,**

**Employee\_Age int NOT NULL,**

**Employee\_\_Address text NOT NULL,**

**Employee\_Gender char(1)**

**CHECK (Employee\_Gender='F' or Employee\_Gender='M'),**

**Employee\_DOB text NOT NULL,**

**Employee\_Salary int NOT NULL,**

**CHECK (Employee\_Salary>0),**

**Employee\_Designation text NOT NULL,**

**Employee\_Phone\_Number char(10) NOT NULL,**

**CHECK (Employee\_Phone\_Number like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')**

**);**

**drop table EMPLOYEE**

**select \* from EMPLOYEE**

**CREATE TABLE FLIGHT(**

**Flight\_Number text PRIMARY KEY NOT NULL,**

**Sourcee text NOT NULL,**

**Destination text NOT NULL,**

**Status text NOT NULL,**

**CHECK(Status='Departed' or Status='Arrived' or Status='Delayed' or Status='Cancelled'),**

**Duration TIME NOT NULL,**

**Departure\_Time TIME NOT NULL,**

**Arrival\_Time TIME NOT NULL,**

**Connected char(1) NOT NULL,**

**CHECK(Connected='Y' or Connected='N')**

**);**

**drop table FLIGHT**

**select \* from FLIGHT**

**CREATE TABLE TICKET(**

**Seat\_Number VARCHAR(20) PRIMARY KEY NOT NULL,**

**Ticket\_Number VARCHAR(20) NOT NULL,**

**PASSENGER\_NAME VARCHAR(20) NOT NULL,**

**Company\_NAME VARCHAR(20) NOT NULL,**

**Price INT NOT NULL,**

**Classs VARCHAR(5) NOT NULL,**

**Departure\_Time TIME NOT NULL,**

**Arrival\_Time TIME NOT NULL,**

**Duration INT NOT NULL,**

**Destination VARCHAR(20) NOT NULL,**

**Sourcee VARCHAR(20) NOT NULL**

**);**

**drop table TICKET**

**select \* from TICKET**

**CREATE TABLE CONTAIN(**

**Airport\_Name VARCHAR(20),**

**Company\_ID text PRIMARY KEY NOT NULL,**

**FOREIGN KEY (Airport\_Name) REFERENCES AIRPORT(Airport\_Name),**

**FOREIGN KEY (Company\_ID) REFERENCES COMPANY(Company\_ID)**

**);**

**drop table CONTAIN**

**select \* from CONTAIN**

**CREATE TABLE HAS(**

**Employee\_ID INT PRIMARY KEY NOT NULL,**

**Airport\_Name VARCHAR(20),**

**FOREIGN KEY (Airport\_Name) REFERENCES AIRPORT(Airport\_Name),**

**FOREIGN KEY (Employee\_ID) REFERENCES EMPLOYEE(Employee\_ID)**

**);**

**drop table HAS**

**select \* from HAS**

**CREATE TABLE SERVES(**

**Employee\_ID INT,**

**Passenger\_ID INT,**

**FOREIGN KEY (Employee\_ID) REFERENCES EMPLOYEE(Employee\_ID),**

**FOREIGN KEY (Passenger\_ID) REFERENCES PASSENGER(Passenger\_ID)**

**);**

**drop table SERVES**

**select \* from SERVES**

**CREATE TABLE BOOKS(**

**Passenger\_ID INT NOT NULL,**

**Seat\_Number VARCHAR(20),**

**FOREIGN KEY (Passenger\_ID) REFERENCES PASSENGER(Passenger\_ID),**

**FOREIGN KEY (Seat\_Number) REFERENCES TICKET(Seat\_Number)**

**);**

**drop table BOOKS**

**select \* from BOOKS**

**CREATE TABLE CANCELS(**

**Passenger\_ID INT NOT NULL,**

**Seat\_Number VARCHAR(20),**

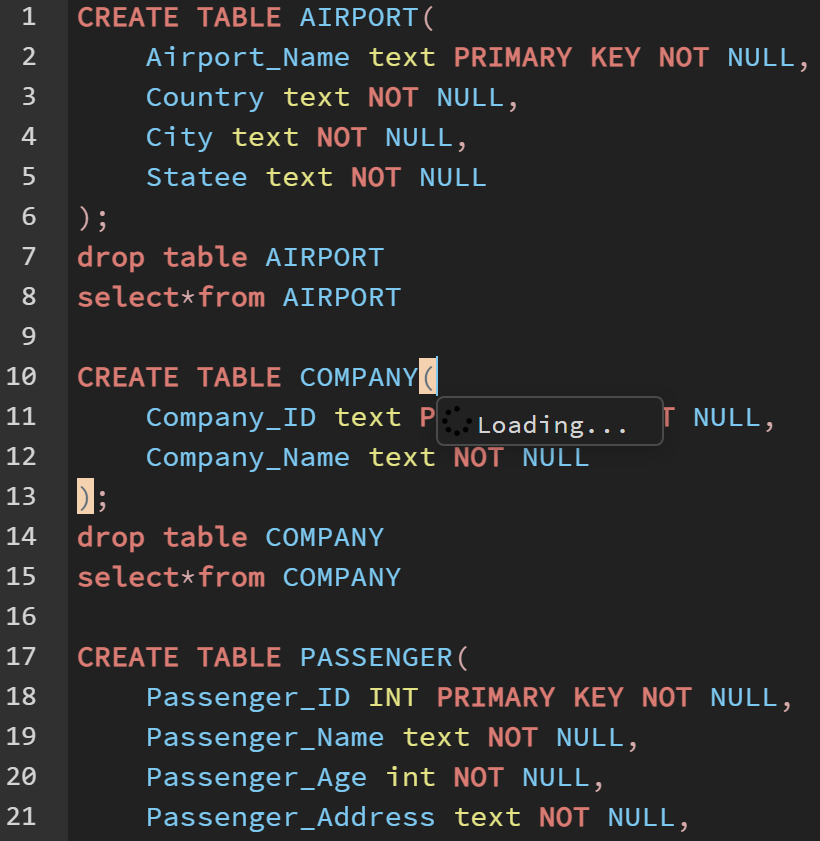
**FOREIGN KEY (Passenger\_ID) REFERENCES PASSENGER(Passenger\_ID),**

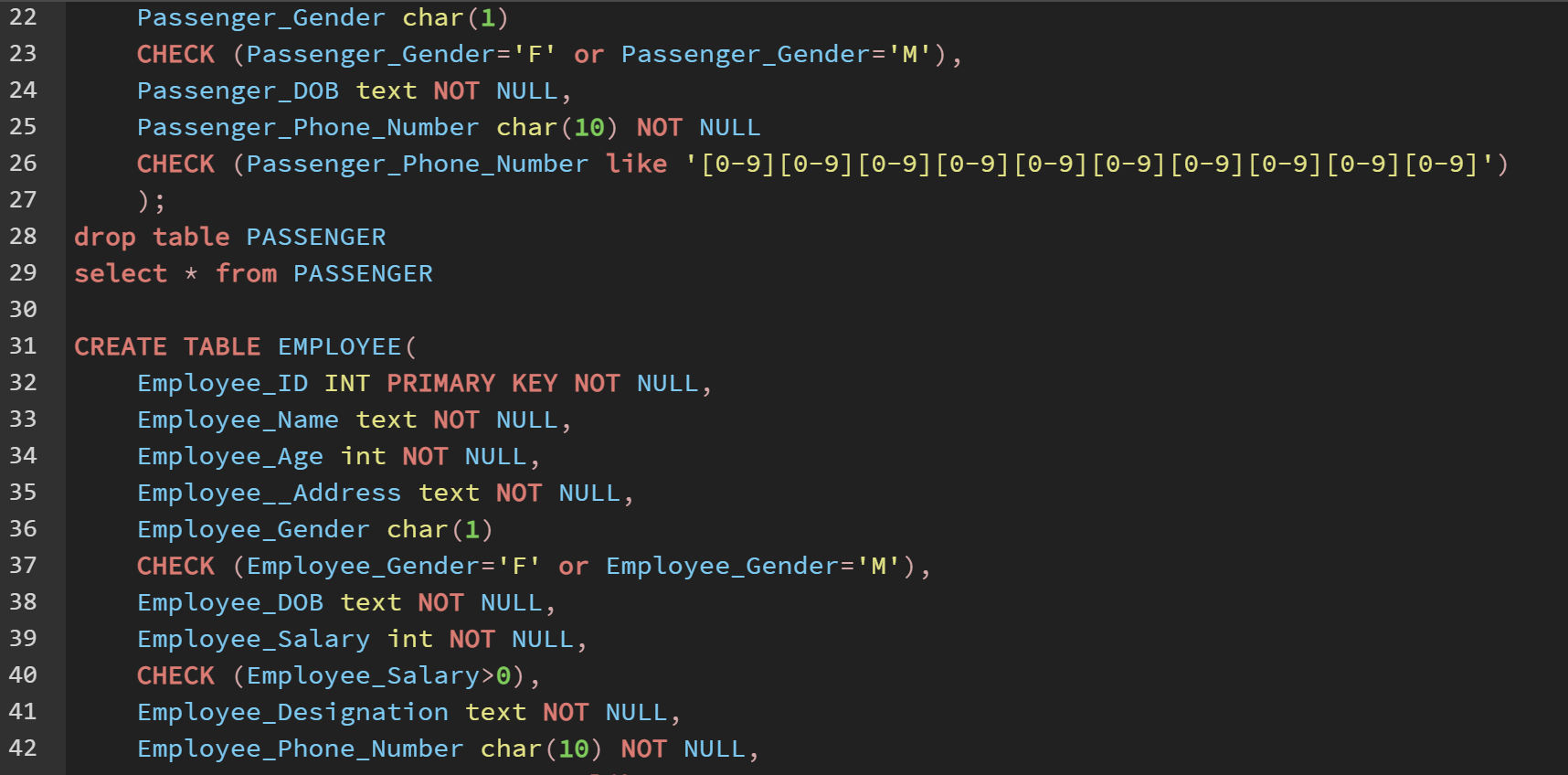
**FOREIGN KEY (Seat\_Number) REFERENCES TICKET(Seat\_Number)**

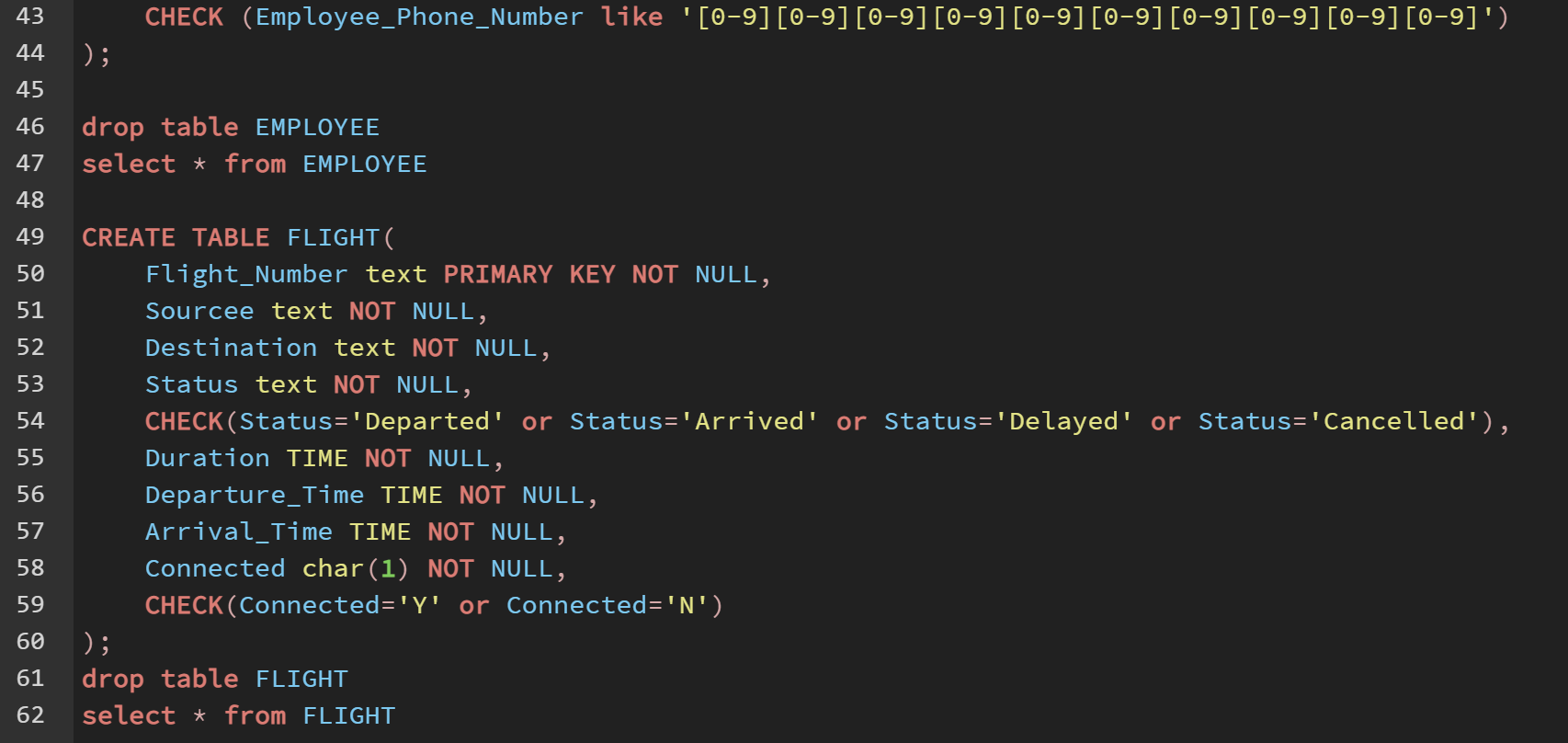
**);**

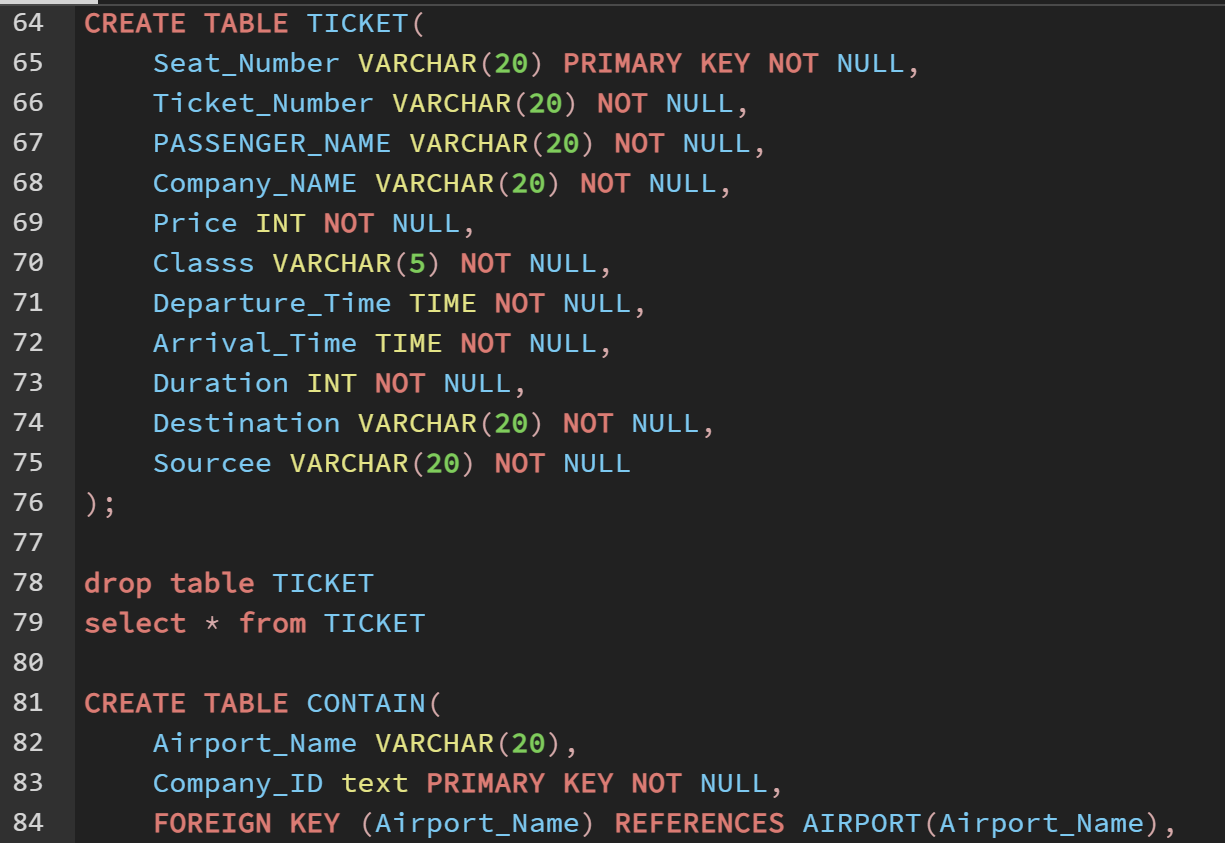
**drop table CANCELS**

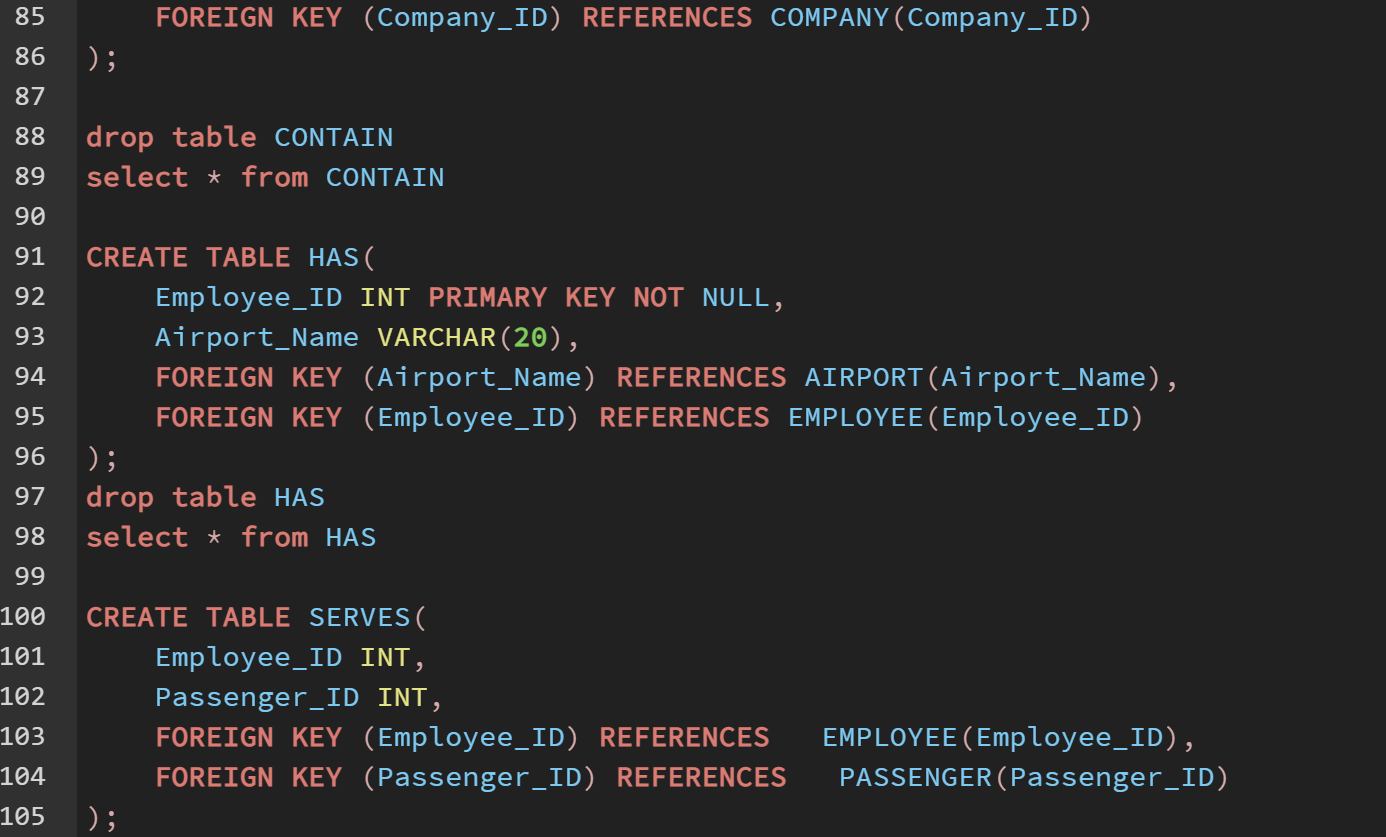
**select \* from CANCELS**

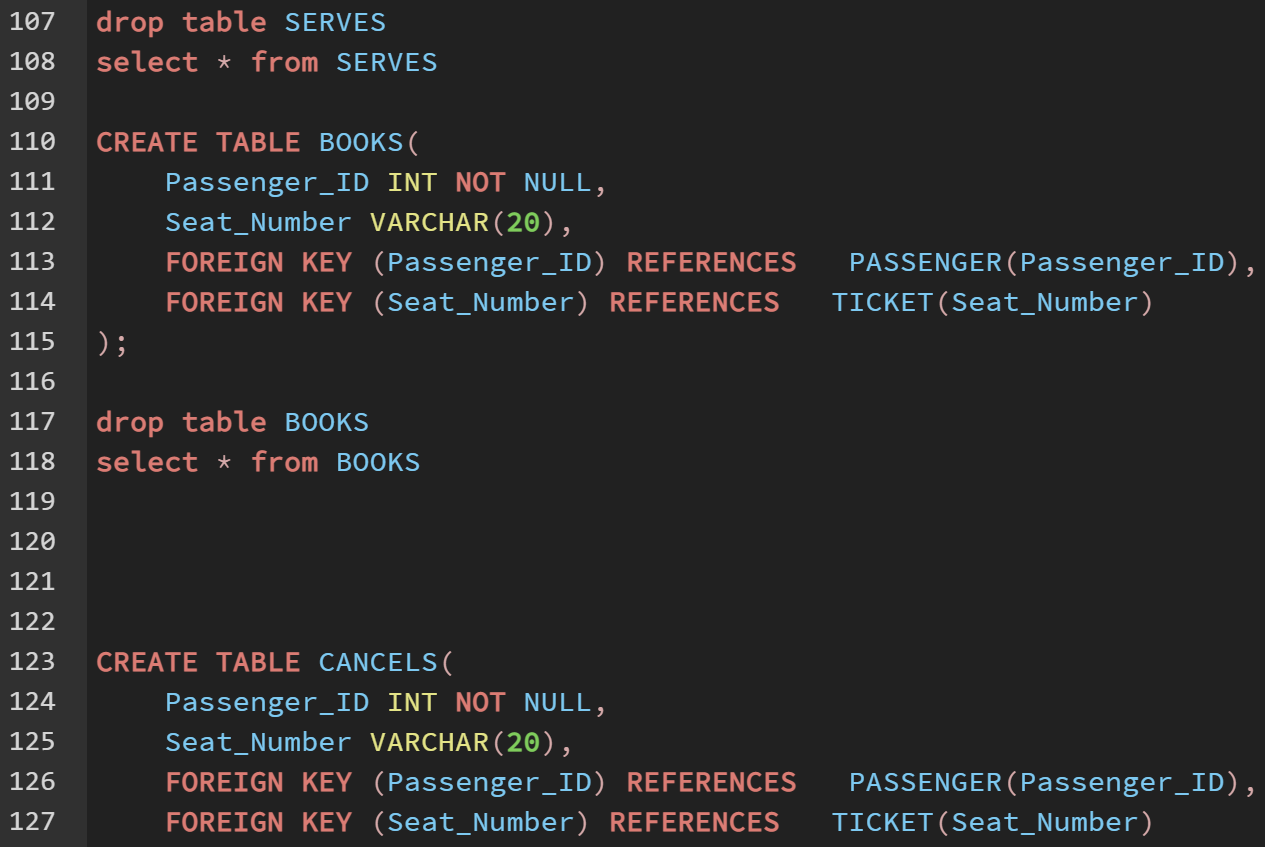


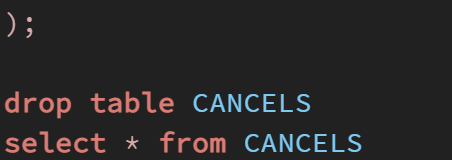












# OUTPUT

# 

# 

# 

# 

# 

# 



**Outcomes:**

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



**Questions:**

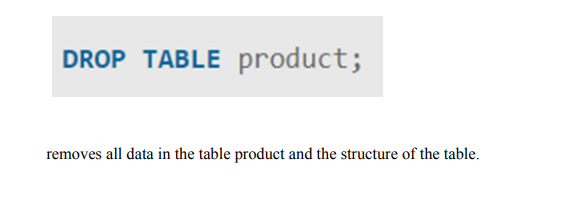
**Q1 what is difference between Truncate, Drop and delete? Explain with example**

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.

****

****

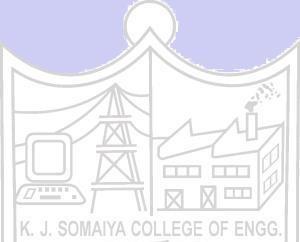


**Conclusion:**

**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)

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)