

**Experiment No.: 04**

**Title:** To use DML operations and SQL queries to

Populate the database

**Batch:** A3 **Roll No.:** 16010421075 **Experiment No: 04**

**Aim:** To use DML operations and SQL queries to populate the database .

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Resources needed:** PostgreSQL PgAdmin4

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

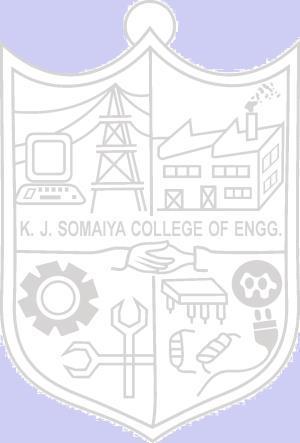
**Theory:**

The Data Manipulation Language (DML) is used to populate the table with values, modify the table values and remove the rows of the table.

The DML statements are: SELECT

INSERT

UPDATE

DELETE

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Procedure:**

CREATE TABLE products ( product\_no integer,

name text, price numeric );

Let us consider the above products table

**Inserting rows:**

The INSERT command requires the table name and column values

INSERT INTO products VALUES (1, ’Cheese’, 9.99);

If we don’t have values for all the columns, you can omit some of them. In that case, the columns will be filled with their default values. For example:

INSERT INTO products (product\_no, name) VALUES (1, ’Cheese’)

**Updating the values:**

The UPDATE command requires three pieces of information:

1. The name of the table and column to update
2. The new value of the column
3. Which row(s) to update

UPDATE products SET price = 10 WHERE price = 5;

UPDATE products SET price = price \* 1.10;

**Deleting rows:**

The syntax of the DELETE command is similar to the UPDATE command. DELETE FROM products WHERE price = 10;

**Retrieving values:**

The general syntax of the SELECT command is SELECT select\_list FROM table\_expression SELECT \* FROM table1;

SELECT \* FROM products WHERE price=10;

SELECT product\_no, name FROM products WHERE price=10;

**Example:**

insert into department values('IT', 101, 'mumbai');

insert into department values('COMP', 102, 'mumbai');

insert into department values('ETRX', 103, 'delhi');

insert into department values('EXTC', 104, 'chennai');

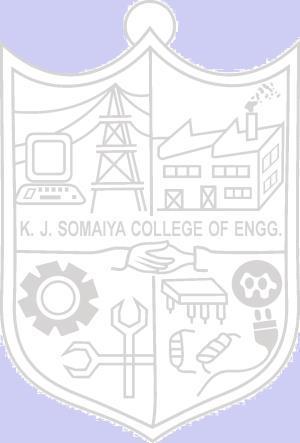
insert into department values('account', 105, 'mumbai');

insert into employee values('anita','m','sharma','emp0001',20000,'mumbai',101);

insert into employee values('nita','g','patil','emp0004',10000,'mumbai',101);

insert into employee values('krupita','v','jetali','emp0003',20000,'delhi',103);

insert into employee values('juhi','r','verma','emp0002',15000,'delhi',104);

insert into employee values('anita','m','sharma', 'emp0005',20000,'mumbai',104);

insert into project values( 1, 'mumbai','website',101);

insert into project values( 2, 'chennai','coding',101);

insert into project values( 3, 'mumbai','testing',102);

insert into project values( 4, 'delhi','documentaion',103);

insert into works\_on values(1,'emp0001', 12);

insert into works\_on values(1,'emp0002', 10);

insert into works\_on values(2,'emp0001', 6);

insert into works\_on values(3,'emp0004', 2);

insert into dependent values(‘emp0001’, ‘sunita’,’sister’);

insert into dependent values(‘emp0001’, ‘nita’,’mother’);

insert into dependent values(‘emp0002’, ‘kamal’,’brother’);

insert into dependent values(‘emp0004’, ‘krishna’,’father’);

select \* from employee;

select \* from department;

select \* from project;

select \* from dependent;

select \* from works\_on;

1) employee

fnamemnamelnamessn salary ecitydno

---------------------------------------- ------------------------------ ----------------------------------

anita m sharma emp0001 20000 mumbai101

juhi r verma emp0002 15000 delhi 104

krupita v jetali emp0003 20000 delhi 103

nita g patil emp0004 10000 mumbai 101

anita m sharma emp0005 20000 mumbai104

2) department

dnamednodlocation

------------------------------ ----------- -----------------------------------

IT 101 mumbai

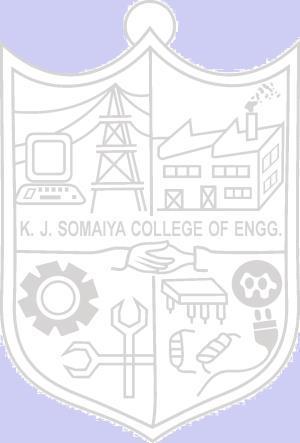
COMP 102 mumbai

ETRX 103 delhi

EXTC 104 chennai

account 105 mumbai

4) project



pnoplocationpnamedno

----------- ---------------------------------------- -------------------- -----------

1 mumbai website 101

2 chennai coding 101

3 mumbai testing 102

4 delhidocumentaion 103

5) dependents

ssndepname relation

-------------------- ------------------------------ ------------------------------

emp0001nita mother

emp0001sunita sister

emp0002kamal brother

emp0004krishna father

6) woks\_on

pnossnno\_of\_hrs

----------- -------------------- -----------

1 emp0001 12

1 emp0002 10

2 emp0001 6

3 emp0004 2

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Results: (Queries printout with output as per the format)**

1. Write 10 queries using ‘from’ and ‘where’ clause.

CREATE TABLE AIRPORT(

    Airport\_Name text PRIMARY KEY NOT NULL,

    Country text NOT NULL,

    City text NOT NULL,

    Statee text NOT NULL

);

insert into AIRPORT values('CSMIA', 'India','Mumbai', 'Maharashtra');

insert into AIRPORT values('IGIA', 'India','Delhi', 'Delhi');

insert into AIRPORT values('SVPI', 'India','Ahmedabad', 'Gujarat');

insert into AIRPORT values('RGIA', 'India','Hyderabad', 'Telangana');

insert into AIRPORT values('NSCBIA', 'India','Kolkata', 'West Bengal');

drop table AIRPORT

select\*from AIRPORT

CREATE TABLE COMPANY(

    Company\_ID text PRIMARY KEY NOT NULL,

    Company\_Name text NOT NULL

);

insert into COMPANY values('6E', 'Indigo');

insert into COMPANY values('UK', 'Vistara');

insert into COMPANY values('AI', 'Air India');

insert into COMPANY values('G8', 'Go First');

insert into COMPANY values('SG', 'Spice Jet');

drop table COMPANY

select\*from COMPANY

CREATE TABLE PASSENGER(

    Passenger\_ID INT PRIMARY KEY NOT NULL,

    CHECK(Passenger\_ID>9999 or Passenger\_ID<100000),

    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 DATE NOT NULL,

    Passenger\_Phone\_Number varchar(10) NOT NULL

);

insert into PASSENGER values(10000, 'Tirth',18, 'Maharashtra', 'M','2003-10-25',9082198360);

insert into PASSENGER values(10001, 'Keyur',18, 'Delhi', 'M','2003-08-20',9884796324);

insert into PASSENGER values(10002, 'Arsalan',18, 'Gujarat', 'M','2003-11-25',9745148752);

insert into PASSENGER values(10003, 'Harsh',18, 'Telangana', 'M','2003-09-25',9658312047);

insert into PASSENGER values(10004, 'Krishiv',18, 'West Bengal', 'M','2003-05-25',9754321984);

drop table PASSENGER

select \* from PASSENGER

CREATE TABLE EMPLOYEE(

    Employee\_ID INT PRIMARY KEY NOT NULL,

    CHECK(Employee\_ID>999 or Employee\_ID<10000),

    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 varchar(10) NOT NULL

);

insert into EMPLOYEE values(1000, 'Avery Peters',26, 'Maharashtra', 'M','2003-10-25', 15000 , 'Senior',9754321984);

insert into EMPLOYEE values(1001, 'Kayleigh Soto',26, 'Maharashtra', 'M','2003-08-20', 17000 , 'Senior',9658312047);

insert into EMPLOYEE values(1002, 'Savanna Horne',26, 'Maharashtra', 'F','2003-11-25', 14000 , 'Junior',9745148752);

insert into EMPLOYEE values(1003, 'Matteo Crosby',26, 'Maharashtra', 'M','2003-09-25', 13000 , 'Senior',9658312047);

insert into EMPLOYEE values(1004, 'Fatma Sheldon',26, 'Maharashtra', 'F','2003-05-25', 11000 , 'Junior',8745147534);

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'),

    Departure\_Time TIME NOT NULL,

    Arrival\_Time TIME NOT NULL,

    Connected char(1) NOT NULL,

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

);

INSERT INTO FLIGHT VALUES('12345','Mumbai','Delhi','Departed', '09:00:00', '11:00:00','N');

INSERT INTO FLIGHT VALUES('12356','Delhi', 'Mumbai','Departed', '09:10:00', '11:05:00','N');

INSERT INTO FLIGHT VALUES('12389','Delhi', 'Mumbai','Departed', '09:15:00', '11:15:00','N');

INSERT INTO FLIGHT VALUES('12384','Ahmedabad', 'Mumbai','Departed', '19:15:00', '21:15:00','N');

INSERT INTO FLIGHT VALUES('12383','Kolkata', 'Mumbai','Departed', '15:55:00', '17:45:00','N');

drop table FLIGHT

select \* from FLIGHT

CREATE TABLE TICKET(

    PASSENGER\_NAME VARCHAR(20) NOT NULL,

    Ticket\_Number VARCHAR(20) NOT NULL,

    Company\_NAME VARCHAR(20) NOT NULL,

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

    Classs text NOT NULL,

    Price INT NOT NULL,

    Departure\_Time TIME NOT NULL,

    Arrival\_Time TIME NOT NULL,

    Destination VARCHAR(20) NOT NULL,

    Sourcee VARCHAR(20) NOT NULL

);

INSERT INTO TICKET VALUES('Keyur',6000, 'Spice Jet', 'A-2' , 'First Class', '2500', '09:10:00',  '11:05:00', 'Delhi', 'Mumbai');

INSERT INTO TICKET VALUES('Kushal',6700, 'Indigo', 'B-5' , 'First Class', '3000', '09:10:00',  '11:05:00', 'Kolkata', 'Delhi');

INSERT INTO TICKET VALUES('Srajan',6800, 'Vistara', 'A-24' , 'First Class', '1500', '09:10:00',  '11:05:00', 'Ahmedabad', 'Mumbai');

INSERT INTO TICKET VALUES('Arsalan',9000, 'Spice Jet', 'D-2' , 'First Class', '1500', '09:10:00',  '11:05:00', 'Delhi', 'Mumbai');

INSERT INTO TICKET VALUES('Shubh',6300, 'Go First', 'F-2' , 'First Class', '1500', '21:10:00',  '23:05:00', 'Delhi', 'Mumbai');

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)

);

INSERT INTO CONTAIN VALUES('CSMIA','6E');

INSERT INTO CONTAIN VALUES('CSMIA','UK');

INSERT INTO CONTAIN VALUES('RGIA','AI');

INSERT INTO CONTAIN VALUES('RGIA','G8');

INSERT INTO CONTAIN VALUES('SVPI','SG');

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)

);

INSERT INTO HAS VALUES(1000,'CSMIA');

INSERT INTO HAS VALUES(1001,'CSMIA');

INSERT INTO HAS VALUES(1002,'SVPI');

INSERT INTO HAS VALUES(1003,'RGIA');

INSERT INTO HAS VALUES(1004,'CSMIA');

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)

);

INSERT INTO SERVES VALUES(1004,10001);

INSERT INTO SERVES VALUES(1001,10002);

INSERT INTO SERVES VALUES(1003,10003);

INSERT INTO SERVES VALUES(1000,10000);

INSERT INTO SERVES VALUES(1001,10004);

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)

);

INSERT INTO BOOKS VALUES(10000, 'A-2');

INSERT INTO BOOKS VALUES(10001, 'B-5');

INSERT INTO BOOKS VALUES(10002, 'A-24');

INSERT INTO BOOKS VALUES(10003, 'D-2');

INSERT INTO BOOKS VALUES(10004, 'F-2');

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)

);

INSERT INTO CANCELS VALUES(10000, 'A-2');

INSERT INTO CANCELS VALUES(10001, 'B-5');

INSERT INTO CANCELS VALUES(10002, 'A-24');

INSERT INTO CANCELS VALUES(10003, 'D-2');

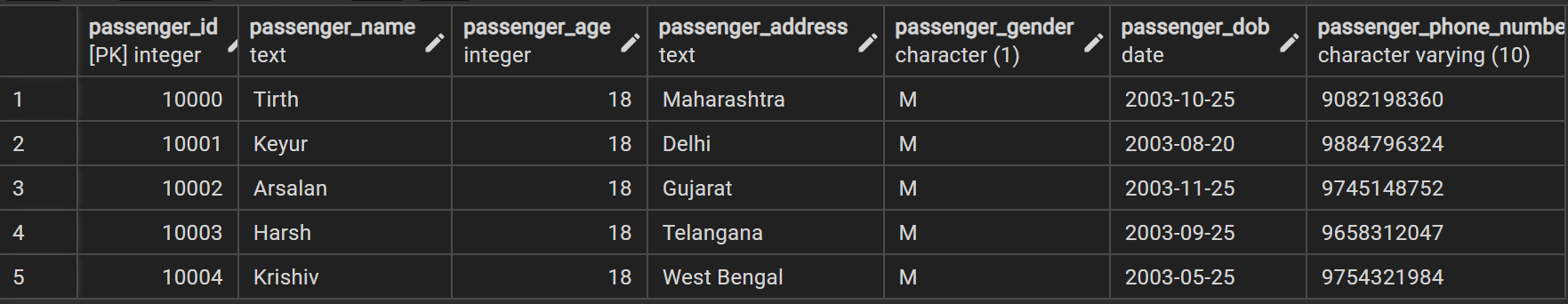
INSERT INTO CANCELS VALUES(10004, 'F-2');

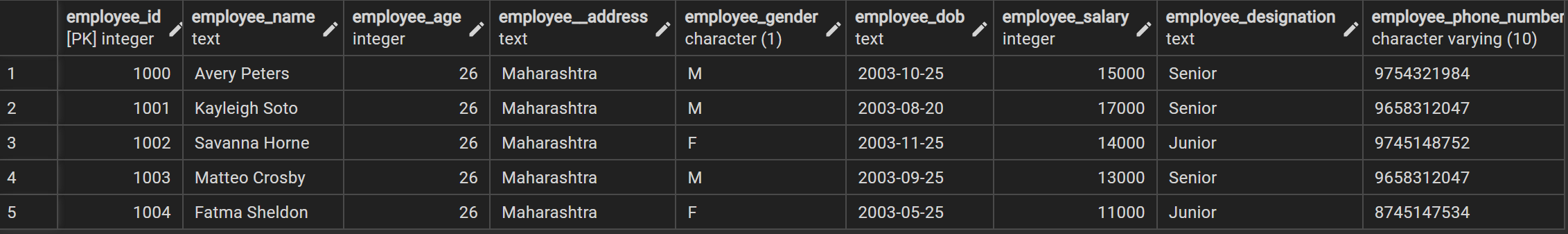
drop table CANCELS

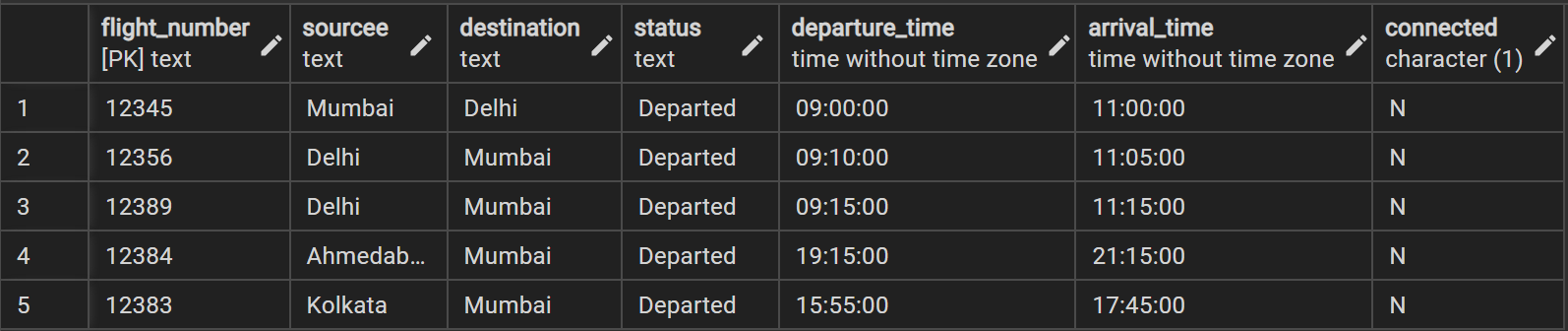
select \* from CANCELS



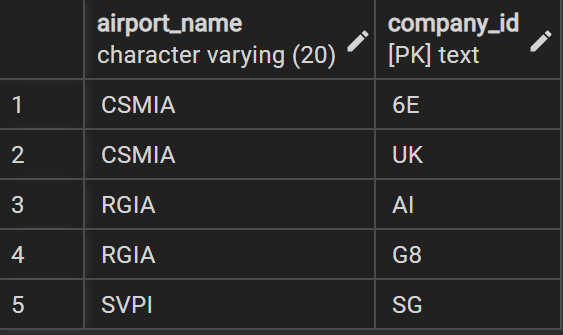


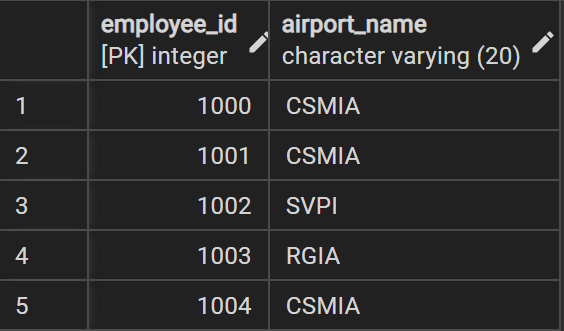


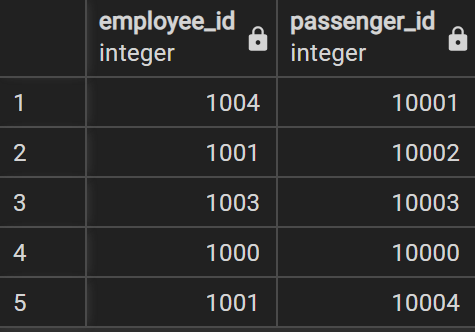


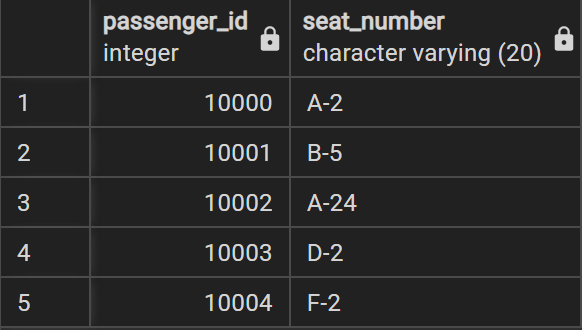


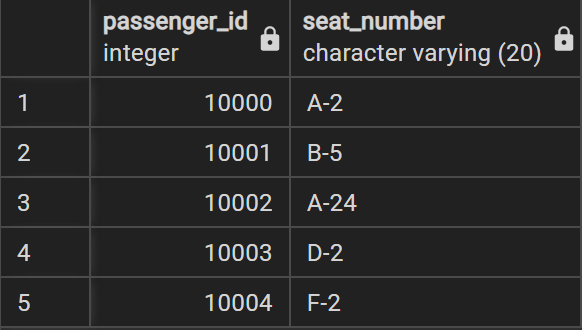




****

****

****

****

**Example:**

1. **To extract the name and ssn of all the employees:**

Select fname, mname, lname, ssn from employee;

fnamemnamelnamessn

---------------------------------------- ---------------------------------------- -----------------------------

anitasharmam emp0001

juhiverma r emp0002

krupitajetali v emp0003

nitapatil g emp0004

anitasharma m emp0005

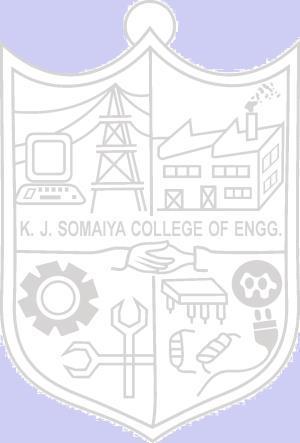
1. **To select names and city of the employees earning salary more then 10000:**

Select fname, mname, lname, ecity from the employee where salary>10000;

fnamemnamelname ecity

---------------------------------------- ------------------------- ----------------------------------------

anitasharmam mumbai

juhivermar delhi

krupitajetaliv delhi

anitasharma m mumbai

1. **TO get the details of the cities of the employees in our company:**

select distinct ecity from employee;

ecity

------------

delhi

mumbai

1. **To find the name of the department located in Mumbai and with department number 101:**

select dname from department where dlocation=’Mumbai’ and dno=101;

dname

--------------

1. **To delete all dependent whose relation is mother with employee:**

delete form dependent where relation=’mother’;

ssndepname relation

-------------------- ------------------------------ ------------------------------

emp0001sunita sister

emp0002kamal brother

emp0004krishna father

1. **Update relation employee to increment salary of all employees working in Department 101 by Rs. 10000:**

update employee set salary=salary+10000 where dno=101;

fnamemnamelnamessn salary ecitydno

---------------------------------------- ------------------------------ ----------------------------------

anita m sharma emp0001 30000 mumbai101

juhi r verma emp0002 15000 delhi 104

krupita v jetali emp0003 20000 delhi 103

nita g patil emp0004 20000 mumbai 101

anita m sharma emp0005 20000 mumbai104

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Outcomes:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Questions:**

**Q1 Explain various data types used in SQL**

1. **CHARACTER [(length)] or CHAR [(length)]**: The CHARACTER data type accepts character strings, including Unicode, of a fixed length. The length of the character string should be specified in the data type declaration; for example, CHARACTER (n) where n represents the desired length of the character string. If no length is specified during the declaration, the default length is 1.
2. **VARCHAR (length)**: The VARCHAR data type accepts character strings, including Unicode, of a variable length is up to the maximum length specified in the data type declaration. A VARCHAR declaration must include a positive integer in parentheses to define the maximum allowable character string length. For example, VARCHAR (n) can accept any length of character string up to n characters in length. The length parameter may take any value from 1 to the current table page size. Attempting to assign a value containing more characters than the defined maximum length results in the truncation of the character string to the defined length. If any of the truncated characters are not blank, an error is raised.
3. **BOOLEAN:** The BOOLEAN data type supports the storage of two values: TRUE or FALSE. No parameters are required when declaring a BOOLEAN data type.
4. **SMALLINT :** The SMALLINT data type accepts numeric values with an implied scale of zero. It stores any integer value between the range 2^ -15 and 2^15 -1. Attempting to assign values outside this range causes an error.
5. **INTEGER or INT** : The INTEGER data type accepts numeric values with an implied scale of zero. It stores any integer value between the range 2^ -31 and 2^31 -1. Attempting to assign values outside this range causes an error.
6. **NUMERIC [(p[,s])]** : Point Base treats the NUMERIC data type in exactly the same way as the DECIMAL data type.
7. **FLOAT(p)** : The FLOAT data type accepts approximate numeric values, for which you may define a precision up to a maximum of 64. If no precision is specified during the declaration, the default precision is 64. Attempting to assign a value lager than the declared precision will cause an error to be raised.

**Q2 what is outer JOIN and why it is used? Explain its type with example**

: The joined table retains each row—even if no other matching row exists.

OUTER JOIN should be used when you want to include rows that have no matching rows in the related table.

Outer joins subdivide further into left outer joins, right outer joins, and full outer joins, depending on which table's rows are retained (left, right, or full).

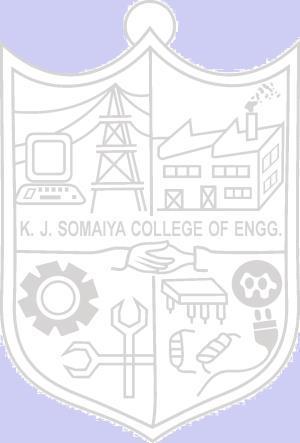
**Left Outer Join** : The left outer join returns a result table with the matched data of two tables then remaining rows of the left table and null for the right table's column.

**Right Outer Join :** The right outer join returns a result table with the matched

data of two tables then remaining rows of the right table and null for the left table's columns

### Full Outer Join :

The full outer join returns a result table with the **matched data** of two table then remaining rows of both **left** table and then the **right** table.



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion: (Conclusion to be based on the objectives and outcomes achieved)**

#### In this experiment, we have used the ‘select-from-where’ clause and have successfully implemented it to write 10 queries. We have also implemented two queries each using the ‘update’ and the ‘delete’ clause. Hence, in this experiment, we have understood the use of data manipulation statements and their implementation.

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

**Signature of faculty in-charge with date**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**References:**

**Books:**

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/