10/31/2023

A P, ARJUN

RBS, Amazon EU

**Advanced SQL**

**Final Project**

**TASK 1**

**Academic Management System.  
  
1. Database Creation:**  
**2. Table Creation:**

create table StudentInfo

(STU\_ID int, STU\_NAME varchar(100),

DOB DATE, PHONE\_NO VARCHAR(10),EMAIL\_ID varchar(50),

ADDRESS varchar(250),primary key (STU\_ID));

create table CourseInfo

(COURSE\_ID INT ,COURSE\_NAME VARCHAR(100), COURSE\_INSTRUCTOR\_NAME VARCHAR(100), primary key (COURSE\_ID));

create table EnrollmentInfo

(ENROLLMENT\_ID INT, STU\_ID int, COURSE\_ID INT, ENROLL\_STATUS varchar(20),

primary key (ENROLLMENT\_ID),

FOREIGN KEY (STU\_ID) REFERENCES StudentInfo(STU\_ID),

FOREIGN KEY (COURSE\_ID) REFERENCES CourseInfo(COURSE\_ID));

create database student\_database;

**Insert Values to the table:**

Insert Into StudentInfo

(STU\_ID, STU\_NAME, DOB, PHONE\_NO, EMAIL\_ID, ADDRESS) Values

('101', 'Arjun', '1993-08-23',9999999991, 'arjun@gmail.com', 'Bangalore'),

('102', 'Avinash', '1991-08-23',9999999992, 'avinash@gmail.com', 'Bangalore'),

('103', 'Rohan', '1998-08-25',9999999993, 'rohan@gmail.com', 'Chennai'),

('104', 'Samyugtha', '2000-10-23',9999999994, 'samyugtha@gmail.com', 'Mumbai'),

('105', 'Sneha', '2001-06-14',9999999995, 'sneha@gmail.com', 'Delhi'),

('106', 'Poornima', '1996-07-17',9999999996, 'poornima@gmail.com', 'Kochi');

Insert Into CourseInfo(COURSE\_ID,COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME) values

(001, 'SQL', 'Ram'),

(002, 'Python', 'Sita'),

(003, 'AWS', 'Radha'),

(004, 'JAVA', 'Krishna'),

(005, 'CSS', 'Shiva');

insert into EnrollmentInfo(ENROLLMENT\_ID, STU\_ID, COURSE\_ID, ENROLL\_STATUS) values

(10001, 101, 001,'ENROLLED'),

(10002, 103, 002,'ENROLLED'),

(10003, 104, 004,'ENROLLED'),

(10004, 102, 003,'ENROLLED'),

(10005, 105, 003,'NOT ENROLLED'),

(10006, 106, 005,'ENROLLED'),

(10007, 101, 002,'ENROLLED');  
  
Select \* From StudentInfo;

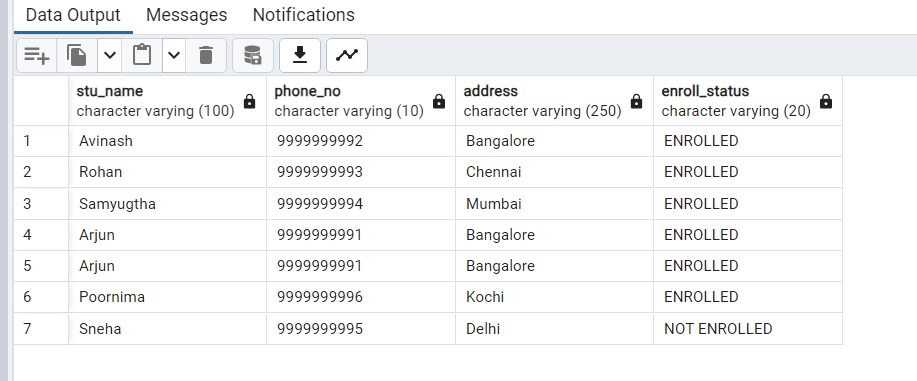
Select \* From CourseInfo;

Select \* From EnrollmentInfo;

**3. Retrieve the Student Information:**a) Write a query to retrieve Student details, Such as Student Name, Contact Information and Enrollment Status.  
  
Select StudentInfo.STU\_NAME, StudentInfo.PHONE\_NO, StudentInfo.ADDRESS,

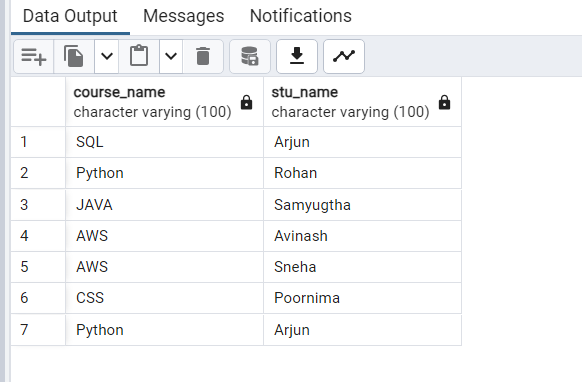
ENROLL\_STATUS

from StudentInfo , EnrollmentInfo

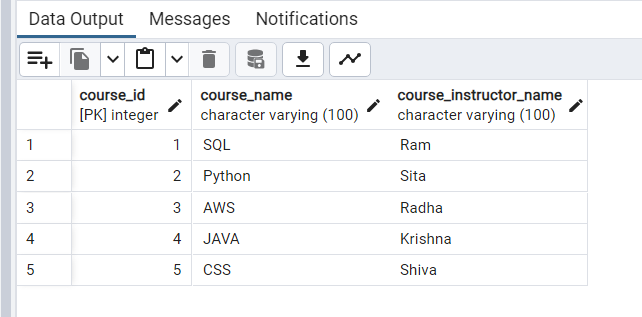
where StudentInfo.STU\_ID=EnrollmentInfo.STU\_ID order by ENROLL\_STATUS ASC;  
  
OUTPUT:  
  
****b) Write a query to retrieve a list of courses in which a specific student enrolled.  
  
select c.COURSE\_NAME, s.STU\_NAME

from EnrollmentInfo e join CourseInfo c on e.course\_id = c.course\_id

join StudentInfo s on s.STU\_ID = e.STU\_ID;

OUTPUT:  


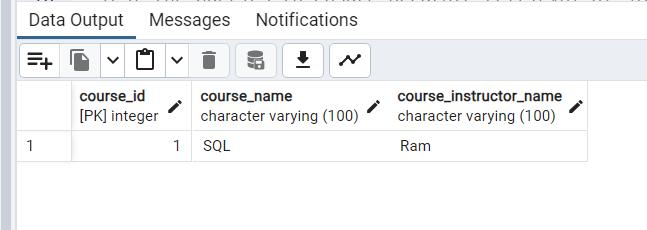
c) Write a query to retrieve course information, including course name insturctor information.

Select \* From CourseInfo;  
  
OUTPUT:   
  


d) Write a query to retrieve course information for a specific course.  
  
Select COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

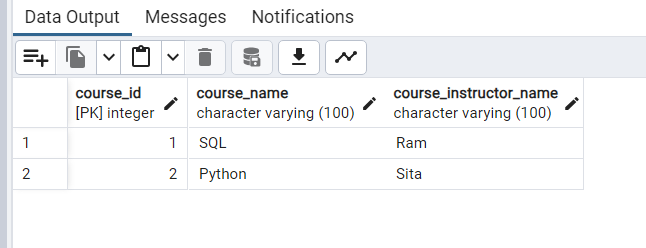
from CourseInfo where COURSE\_NAME='SQL';

OUTPUT:



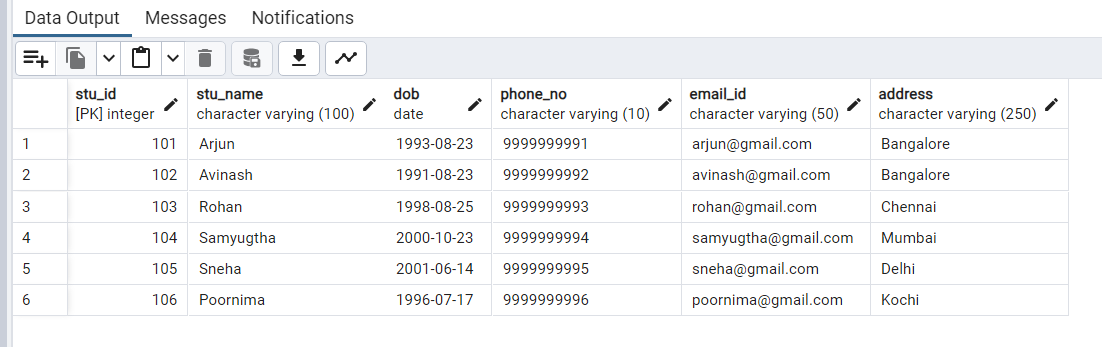
e) Write a query to retrieve course information for multiple courses.  
  
Select COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

from CourseInfo where COURSE\_NAME in ('SQL','Python');  
  
OUTPUT:



f) Test the queries to ensure accurate retrieval of Student Information.

(Execute queries and verify the results against the expected output).

Select \* From StudentInfo;  
  
OUTPUT:  
  


**4. Reporting and Analytics (using joining queries):**

a) Write a query to retrieve the number of students enrolled in each courses.

Select c.Course\_Name , count(c.course\_id) as numberofStud

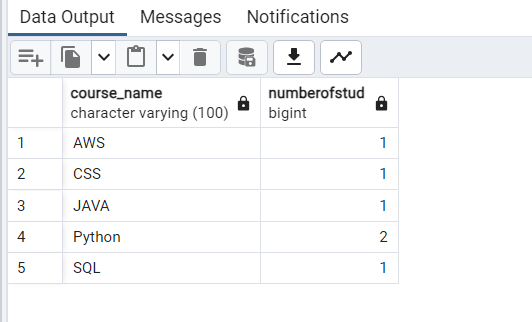
from CourseInfo c join EnrollmentInfo e

on c.course\_id=e.course\_ID

where e.enroll\_status = 'ENROLLED'

group by 1;

OUTPUT:

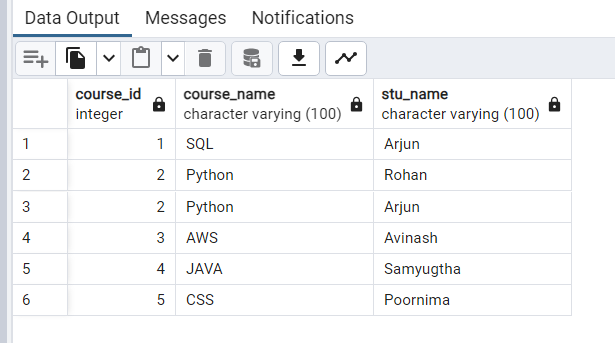


b) Write a query to retrieve the list of students enrolled in specific courses.  
  
select e.COURSE\_ID, c.COURSE\_NAME, s.STU\_NAME

from CourseInfo c join EnrollmentInfo e on c.course\_id=e.course\_ID

join StudentInfo s on s.STU\_ID = e.STU\_ID where e.enroll\_status = 'ENROLLED';

OUTPUT:



c) Write a query to retrieve the count of enrolled students for each instructor.  
  
Select c.COURSE\_INSTRUCTOR\_NAME , count(s.STU\_ID) as numberofStud

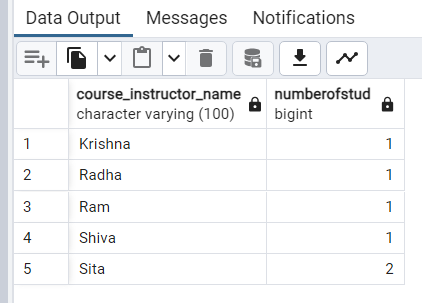
from CourseInfo c join EnrollmentInfo e

on c.course\_id=e.course\_ID

join StudentInfo s on s.STU\_ID = e.STU\_ID

where e.enroll\_status = 'ENROLLED'

group by 1;  
  
OUTPUT:



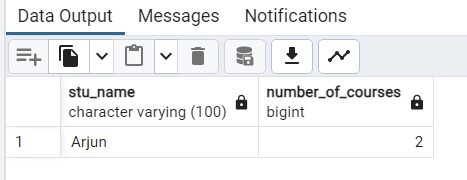
d) Write a query to retrieve the list of students who enrolled in multiple courses.  
  
Select S.STU\_Name , count(c.course\_id) as Number\_of\_courses

from CourseInfo c join EnrollmentInfo e

on c.course\_id=e.course\_ID

join StudentInfo s on s.STU\_ID = e.STU\_ID

where e.enroll\_status = 'ENROLLED' group by 1 having count(c.course\_name) >1;  
  
OUTPUT:



e) Write a query to retrieve the courses that have the highest number of enrolled students. (arrange from highest to lowest).  
  
Select S.STU\_Name , count(c.course\_id) as Number\_of\_courses

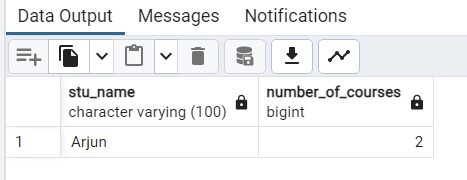
from CourseInfo c join EnrollmentInfo e

on c.course\_id=e.course\_ID

join StudentInfo s on s.STU\_ID = e.STU\_ID

where e.enroll\_status = 'ENROLLED' group by 1 having count(c.course\_name) >1

order by count(c.course\_id) desc;  
  
OUTPUT:



**TASK 2**

**Student Database Management System:  
  
1) Database setup:  
  
a) Database creation.**CREATE DATABASE "Student\_Database"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'English\_United States.1252'

LC\_CTYPE = 'English\_United States.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;  
  
**a) Student table creation:**create table Student\_table

(Student\_id int, Stu\_name varchar(100),

Department varchar(50), email\_id varchar(50),

Phone\_no numeric, Address varchar(250),

Date\_Of\_Birth DATE, Gender Varchar(30),

Major Varchar(50), GPA numeric, Grade varchar(10));

**2) Data Entry:**Insert into Student\_table

(Student\_id, Stu\_name, Department, email\_id, Phone\_no, Address

,Date\_Of\_Birth, Gender, Major, GPA, Grade) Values

('1', 'Arjun A P', 'Engineering', 'arjun@gmail.com', '9999999991',

'Bangalore', '1999-04-26', 'Male', 'Computer Science', '8.8', 'A'),

('2', 'Avinash Murali', 'Arts and Sciences', 'avinash@gmail.com',

'9999999992', 'Chennai', '1992-07-15', 'Male', 'Mathematics', '8.6', 'A'),

('3', 'Rohan Shetty', 'Business', 'rohan@gmail.com', '9999999993',

'Mangalore', '1995-06-28', 'Male', 'MBA', '8.5', 'A'),

('4', 'Samyugtha C K', 'Arts and Sciences', 'samyugtha@gmail.com',

'9999999994', 'Chennai', '1997-01-12', 'Female', 'Physics', '7.6', 'B'),

('5', 'Sneha Sasi', 'Arts and communication', 'sneha@gmail.com',

'9999999995', 'Mumbai', '1995-10-26', 'Female', 'Communication', '6.9', 'B'),

('6', 'Poornima B', 'Engineering', 'poornima@gmail.com',

'9999999996', 'Goa', '2000-01-03', 'Female', 'Computer Science', '5.5', 'C'),

('7', 'Raghava V J', 'Human Development', 'raghava@gmail.com',

'9999999997', 'Uttar Pradesh', '2000-10-05', 'Male', 'Counseling', '7.0', 'B'),

('8', 'Parul Arora', 'Arts and communication', 'parul@gmail.com',

'9999999998', 'Gujarat', '1999-11-16', 'Female', 'Theatre', '9.2', 'A'),

('9', 'Naveen Kumar', 'Business', 'naveen@gmail.com', '9999999999',

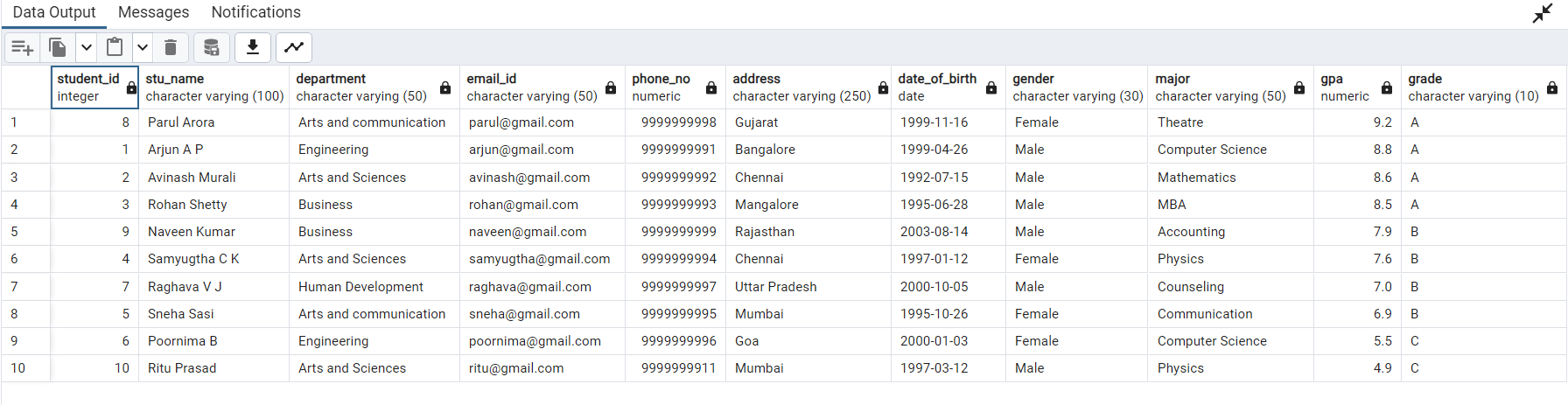
'Rajasthan', '2003-08-14', 'Male', 'Accounting', '7.9', 'B'),

('10', 'Ritu Prasad', 'Arts and Sciences', 'ritu@gmail.com',

'9999999911', 'Mumbai', '1997-03-12', 'Male', 'Physics', '4.9', 'C');

**3) Student information retrieval:**Select \* from Student\_table

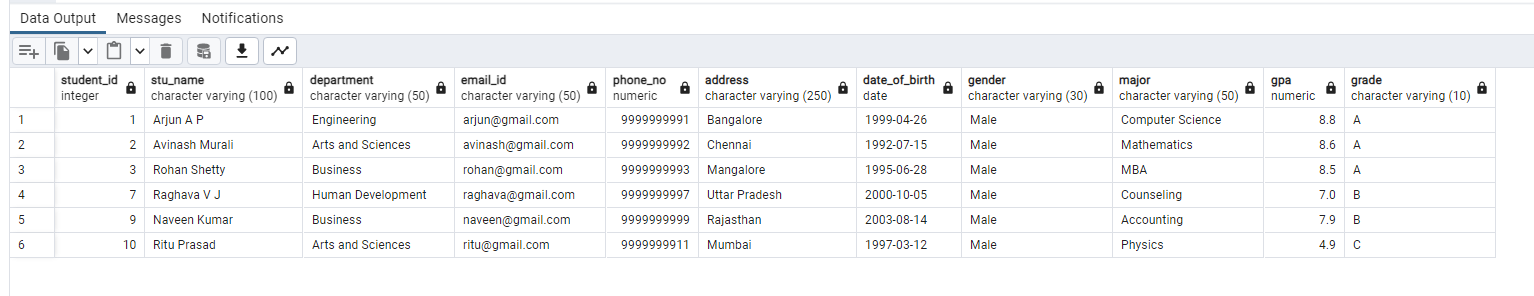
Order By GPA desc, Grade;  
  
OUTPUT:



**4) Query for Male students:**Select \* from Student\_table

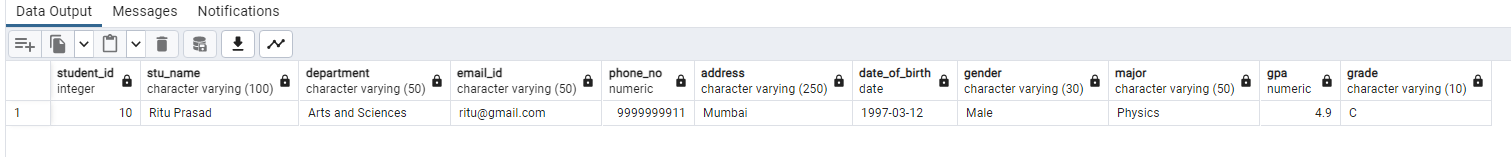
where Gender = 'Male';

OUTPUT:

  
  
  
**5) GPA less than 5:**  
Select \* from Student\_table

where GPA <5.0;

OUTPUT:



**6) Update email\_id and Grade:**UPDATE Student\_table

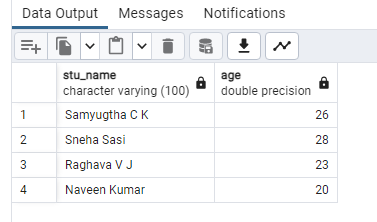
SET email\_id = 'ritu123@gmail.com', GPA='3.0', Grade = 'D'

Where Student\_id = '10';

**7) Query Grade B:**SELECT Stu\_name, date\_part('year',age(Date\_Of\_Birth)) as Age

FROM Student\_table

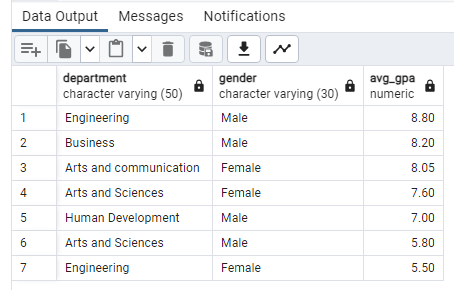
Where Grade = 'B';  
  
OUTPUT:



**8) Grouping and calculation:**Select Department, Gender, round(Avg(GPA),2) as Avg\_GPA

From Student\_table

Group By 1,2 Order By Avg(GPA) DESC;  
  
OUTPUT:

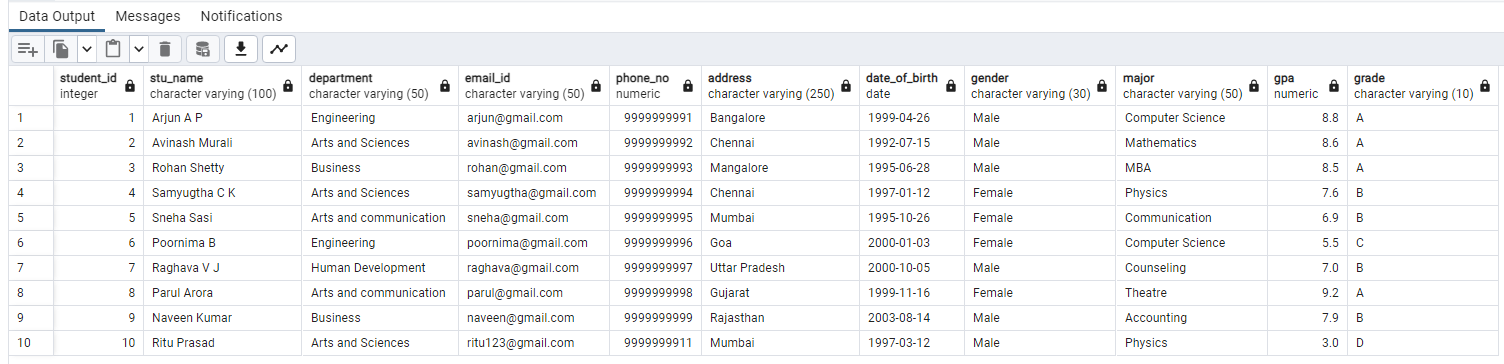


**9) Table Renaming:**

Alter table Student\_table

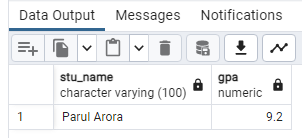
Rename to Student\_info;

Select \* from Student\_info;

OUTPUT:  
  
****

**10) Retrieve student with highest GPA:**Select Stu\_name, GPA

From Student\_info where GPA = (Select Max(GPA) From Student\_info);

OUTPUT:  
  


**TASK 3**

**Event management system using PostgreSQL.  
  
1) Database creation:**CREATE DATABASE "EventsManagement"

WITH OWNER = postgres ENCODING = 'UTF8'

LC\_COLLATE = 'English\_United States.1252'

LC\_CTYPE = 'English\_United States.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;  
  
**a) Table creation:**Create table Events (Event\_Id Int, Event\_Name Varchar(30), Event\_Date Date,

Event\_Location Varchar(100), Event\_Description Varchar(200), primary key (Event\_Id));

Create table Attendees (Attendee\_Id Int, Attendee\_Name Varchar(30),

Attendee\_Phone numeric, Attendee\_Email Varchar(30), Attendee\_City Varchar(20),

primary key (Attendee\_Id));

Create table Registrations (Registration\_Id Int, Event\_Id Int, Attendee\_Id Int,

Registration\_Date Date, Registration\_Amount numeric, primary key (Registration\_Id),

FOREIGN KEY (Event\_Id) REFERENCES Events(Event\_Id),

FOREIGN KEY (Attendee\_Id) REFERENCES Attendees(Attendee\_Id));

**2) Data creation:**insert into Events (Event\_Id, Event\_Name, Event\_Date, Event\_Location, Event\_Description) values

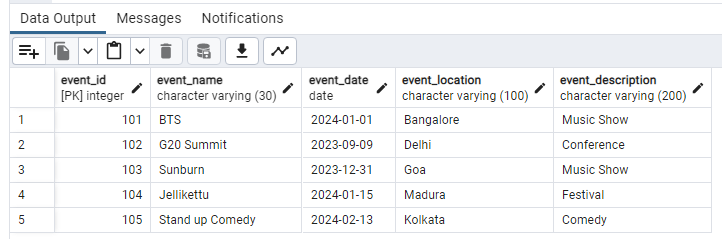
('101', 'BTS', '2024-01-01', 'Bangalore', 'Music Show'),

('102', 'G20 Summit', '2023-09-09', 'Delhi', 'Conference'),

('103', 'Sunburn', '2023-12-31', 'Goa', 'Music Show'),

('104', 'Jellikettu', '2024-01-15', 'Madura', 'Festival'),

('105', 'Stand up Comedy', '2024-02-13', 'Kolkata', 'Comedy');

Select \* from Events;OUTPUT:  
  
****Insert into Attendees (Attendee\_Id, Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City) values

('1001', 'Arjun', '9999999991', 'arjun@gmail.com', 'Bangalore'),

('1002', 'Avinash', '9999999992', 'avinash@gmail.com', 'Chennai'),

('1003', 'Samyugtha', '9999999993', 'samyugtha@gmail.com', 'Chennai'),

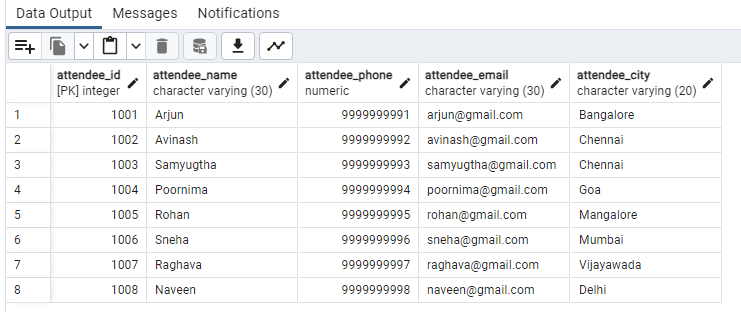
('1004', 'Poornima', '9999999994', 'poornima@gmail.com', 'Goa'),

('1005', 'Rohan', '9999999995', 'rohan@gmail.com', 'Mangalore'),

('1006', 'Sneha', '9999999996', 'sneha@gmail.com', 'Mumbai'),

('1007', 'Raghava', '9999999997', 'raghava@gmail.com', 'Vijayawada'),

('1008', 'Naveen', '9999999998', 'naveen@gmail.com', 'Delhi');

Select \* from Attendees;  
  
OUTPUT:  
  


Insert into Registrations

(Registration\_Id, Event\_Id, Attendee\_Id, Registration\_Date, Registration\_Amount) values

('10001', '101', '1001', '2023-11-30', '7500'),

('10002', '102', '1003', '2023-09-01', '1000'),

('10003', '103', '1002', '2023-12-12', '2000'),

('10004', '104', '1004', '2023-11-25', '500'),

('10005', '101', '1005', '2023-12-10', '7500'),

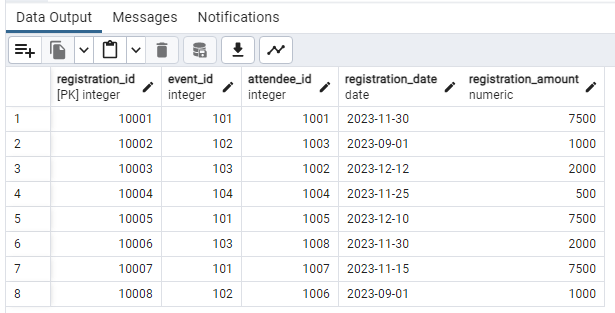
('10006', '103', '1008', '2023-11-30', '2000'),

('10007', '101', '1007', '2023-11-15', '7500'),

('10008', '102', '1006', '2023-09-01', '1000');

Select \*from Registrations;

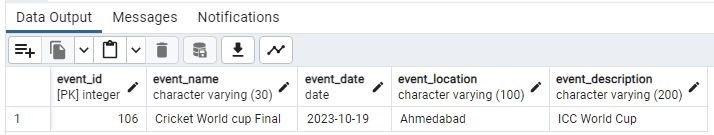
OUTPUT:



**3) Manage Event Details  
  
a) Insert New event:**

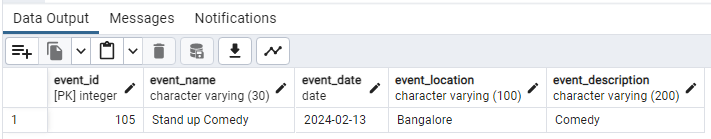
Insert INTO Events (Event\_Id, Event\_Name, Event\_Date, Event\_Location, Event\_Description) VALUES

('106', 'Cricket World cup Final', '2023-10-19', 'Ahmedabad', 'ICC World Cup');

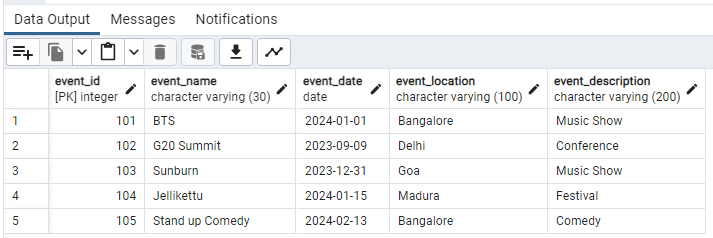
Select \* from Events where Event\_Id = '106';  
  
OUTPUT:  
  


**b) Updating an event's information:**UPDATE Events

SET Event\_Location = 'Bangalore' Where Event\_Id = '105';

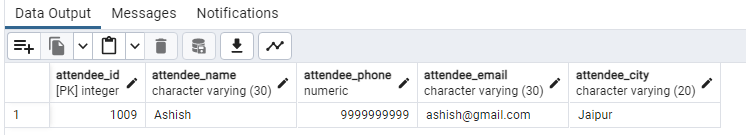
Select \* from Events where Event\_Id = '105';  
  
OUTPUT:  
  


**c) Deleting an event:**Delete from Events where Event\_Id = '106';

Select \* from Events;  
  
OUTPUT:  
  


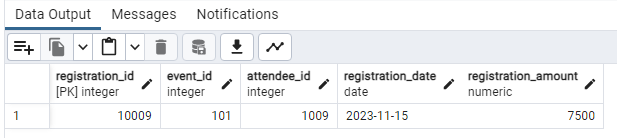
**4) Manage Event Details  
  
a) Insert new attendee:**Insert into Attendees (Attendee\_Id, Attendee\_Name, Attendee\_Phone, Attendee\_Email, Attendee\_City)

values ('1009', 'Ashish', '9999999999', 'ashish@gmail.com', 'Jaipur');

Select \* from Attendees where Attendee\_Id = '1009';  
  
OUTPUT:  
  


**b) Registering an attendee of an event:**Insert into Registrations (Registration\_Id, Event\_Id, Attendee\_Id, Registration\_Date, Registration\_Amount)

values ('10009', '101', '1009', '2023-11-15', '7500');

Select \* from Registrations where Registration\_Id = '10009';  
  
OUTPUT:  
  


**5) Develop the queries to retrieve event information, generate attendee lists and calculate event attendance statistics.**

with Event1 as(

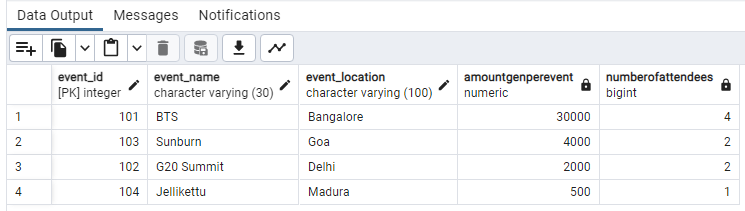
select E.Event\_id, E.event\_name, E.event\_date, E.event\_location,  
 count(R.Event\_Id) over(partition by R.Event\_Id) as NumberofAttendees,

sum(R.registration\_amount) over(partition by E.event\_id) as Amountgenperevent

from Events E join Registrations R on E.event\_id = R.event\_id

join Attendees A on A.attendee\_id = R.attendee\_id)

select Event\_id, event\_name, event\_location, Amountgenperevent , NumberofAttendees

from Event1 group by 1,2,3,4,5;  
OUTPUT:  
  
****

**TASK 4**

**OLAP operations using PostgreSQL.  
  
1) Database creation:**CREATE DATABASE "Sales Data";

WITH OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'English\_United States.1252'

LC\_CTYPE = 'English\_United States.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1

IS\_TEMPLATE = False; **a) Table creation:** Create table Sales\_sample (Product\_Id Int, Region Varchar(50), On\_date Date,

Sales\_Amount Numeric);  
 **2) Data creation:**Insert into Sales\_sample (Product\_Id, Region, On\_date, Sales\_Amount) values

('1', 'Kolkata', '2023-10-10', '20000'),

('2', 'Ahmedabad', '2023-09-19', '50000'),

('2', 'Kolkata', '2023-10-21', '40000'),

('3', 'Delhi', '2023-09-20', '15000'),

('4', 'Delhi', '2023-08-06', '45000'),

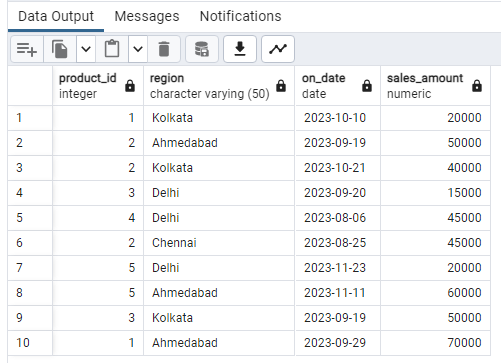
('2', 'Chennai', '2023-08-25', '45000'),

('5', 'Delhi', '2023-11-23', '20000'),

('5', 'Ahmedabad', '2023-11-11', '60000'),

('3', 'Kolkata', '2023-09-19', '50000'),

('1', 'Ahmedabad', '2023-09-29', '70000');

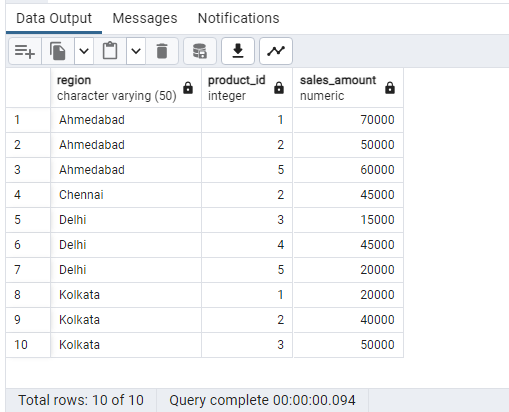
Select \* from Sales\_Sample;  
  
OUTPUT:  
  


**3) Perform OLAP operations:  
  
a) Drill Down:**Select Region, Product\_Id, Sum(Sales\_Amount) as Sales\_Amount

From Sales\_Sample

Group By 1,2 Order By Region, Product\_Id, Sales\_Amount;

OUTPUT:



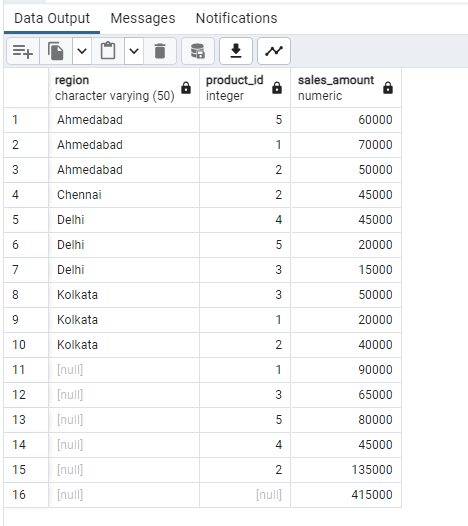
**b) Roll up:**Select Region, Product\_Id, Sum(Sales\_Amount) as Sales\_Amount

From Sales\_Sample

Group By Rollup (2,1)

Order By Region;

OUTPUT:

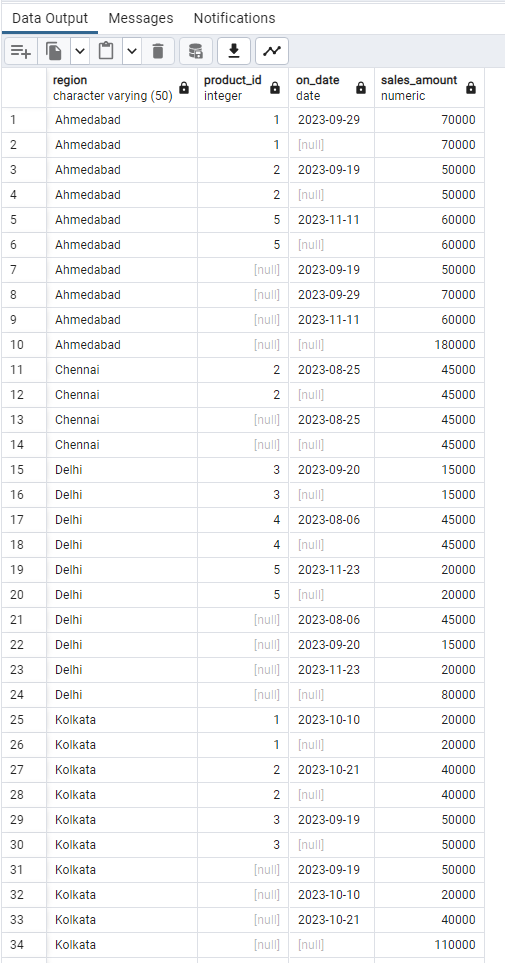
****

**c) Cube:**Select Region, Product\_Id, On\_Date, Sum(Sales\_Amount) as Sales\_Amount

From Sales\_Sample

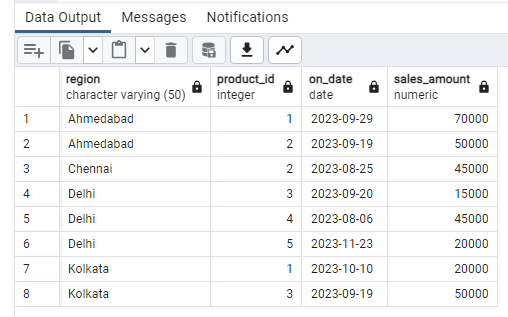
Group By Cube (1,2,3)

Order By Region, Product\_Id, On\_Date, Sales\_Amount;

OUTPUT:  
  
  
**d) Slice:**Select Region, Product\_Id, On\_Date, Sum(Sales\_Amount) as Sales\_Amount From Sales\_Sample

Where Region in('Delhi', 'Chennai') OR On\_Date between To\_date('2023-08-20','YYYY-MM-DD') And To\_Date('2023-10-20','YYYY-MM-DD')

Group By 1,2,3 Order By Region, Product\_Id, On\_Date, Sales\_Amount;

OUTPUT:  
  


**e) Dice:**Select Region, Product\_Id, On\_Date, Sum(Sales\_Amount) as Sales\_Amount

From Sales\_Sample Where Region in('Delhi', 'Chennai') AND Product\_Id IN (1,2) AND On\_Date between To\_date('2023-08-20','YYYY-MM-DD') And To\_Date('2023-10-20','YYYY-MM-DD')

Group By 1,2,3 Order By Region, Product\_Id, On\_Date, Sales\_Amount;  
OUTPUT: