CAPSTONE PROJECT

Task 1 (Workbench)

1) Database Creation

create database student_database;

use student_database;

a) 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));

a) create table CourseInfo

(COURSE_ID INT ,COURSE_NAME VARCHAR(100), COURSE_INSTRUCTOR_NAME VARCHAR(100), primary key (COURSE_ID));

b) 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));

2) Data Creation

a) Insert Into StudentInfo

(STU_ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID ,ADDRESS) Values

('1001', 'Tom Hardy', '1993-08-23',9999999991, 'tom101@gamil.com', 'Banglore'),

('1002', 'Sam Joseph', '1994-08-23',9999999992, 'sam102@gamil.com', 'Banglore'),

('1003', 'Ben Issac', '1993-08-25',9999999993, 'ben103@gamil.com', 'Chennai'),

('1004', 'Kane Lewis', '1993-10-23',9999999994, 'kane104@gamil.com', 'Mumbai'),

('1005', 'Ian Robert', '1994-06-14',9999999995, 'ian105@gamil.com', 'Delhi'),

('1006', 'John Austin', '1991-07-17',9999999996, 'john106@gamil.com', 'Kochi');

b) Insert Into CourseInfo(COURSE_ID,COURSE_NAME, COURSE_INSTRUCTOR_NAME) values

(1, 'SQL','David'),

(2, 'Python','Artur'),

(3, 'AWS','Sebastian'),

(4, 'JAVA','Harry'),

(5, 'CSS','Jack');

c) insert into EnrollmentInfo(ENROLLMENT_ID, STU_ID, COURSE_ID, ENROLL_STATUS) values

(10001, 1001, 001,'ENROLLED'),

(10002, 1003, 002,'ENROLLED'),

(10003, 1004, 004,'ENROLLED'),

(10004, 1002, 003,'ENROLLED'),

(10005, 1005, 003,'NOT ENROLLED'),

(10006, 1006, 005,'ENROLLED');

3) **Retrieve the Student Information**

a) Write a query to retrieve Student details, Such as Student Name, Contact Informations and Enrollment Status

Select s.STU_NAME, s.PHONE_NO, s.ADDRESS,

e.ENROLL_STATUS

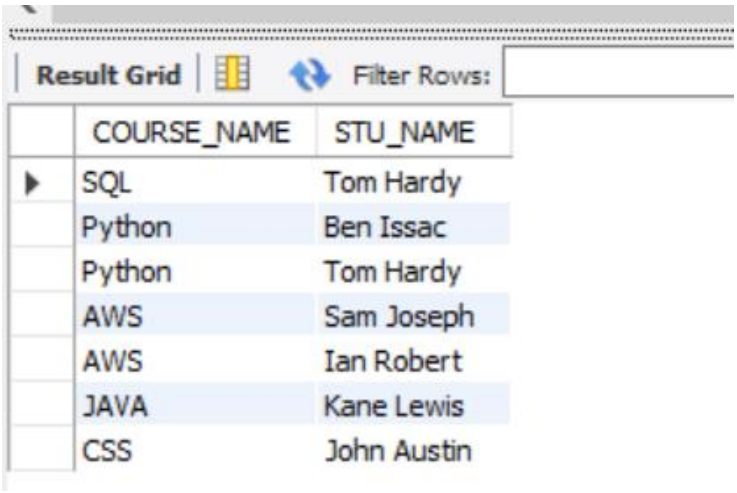
from StudentInfo s join EnrollmentInfo e

on s.STU_ID=e.STU_ID order by e.ENROLL_STATUS ASC;

STU_NAME	PHONE_NO	ADDRESS	ENROLL_STATUS
Tom Hardy	9999999991	Banglore	ENROLLED
Tom Hardy	9999999991	Banglore	ENROLLED
Sam Joseph	9999999992	Banglore	ENROLLED
Ben Issac	9999999993	Chennai	ENROLLED
Kane Lewis	9999999994	Mumbai	ENROLLED

b) Write a query to retrieve a list of courses in which a specific student enrolled

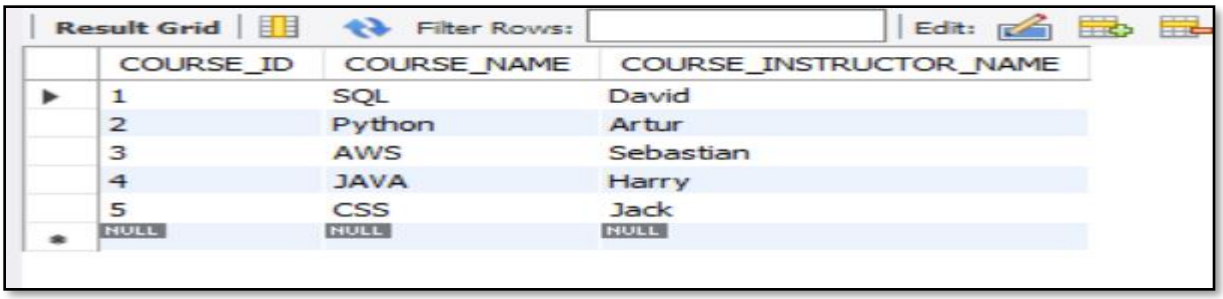
```
select c.COURSE_NAME, s.STU_NAME
from EnrollmentInfo e join CourseInfo c on e.course_id = c.course_id
join StudentInfo s on s.STU_ID = e.STU_ID
```



COURSE_NAME	STU_NAME
SQL	Tom Hardy
Python	Ben Issac
Python	Tom Hardy
AWS	Sam Joseph
AWS	Ian Robert
JAVA	Kane Lewis
CSS	John Austin

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

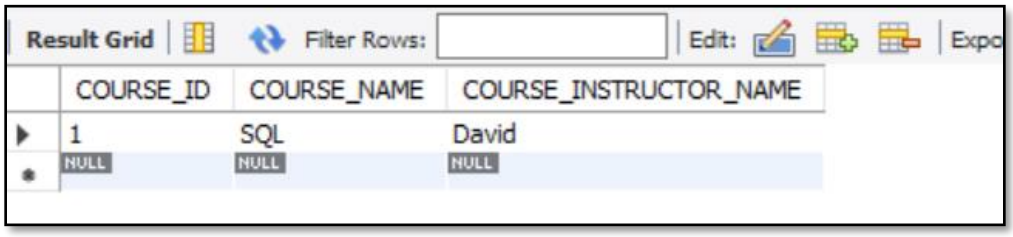
```
Select * From CourseInfo;
```



COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME
1	SQL	David
2	Python	Artur
3	AWS	Sebastian
4	JAVA	Harry
5	CSS	Jack
NULL	NULL	NULL

d) Write a query to retrieve course information for a specific course

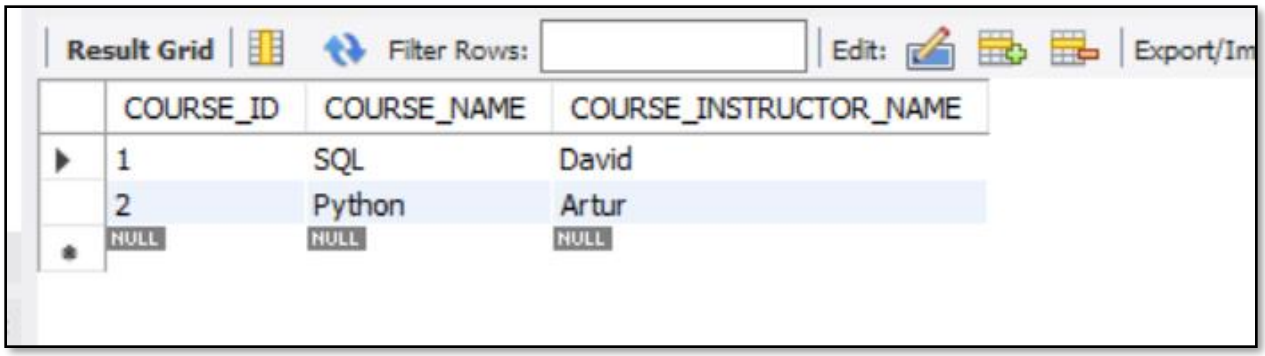
```
Select COURSE_ID,COURSE_NAME, COURSE_INSTRUCTOR_NAME
 from CourseInfo
 where COURSE_NAME='SQL';
```



COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME
1	SQL	David
NULL	NULL	NULL

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');
```

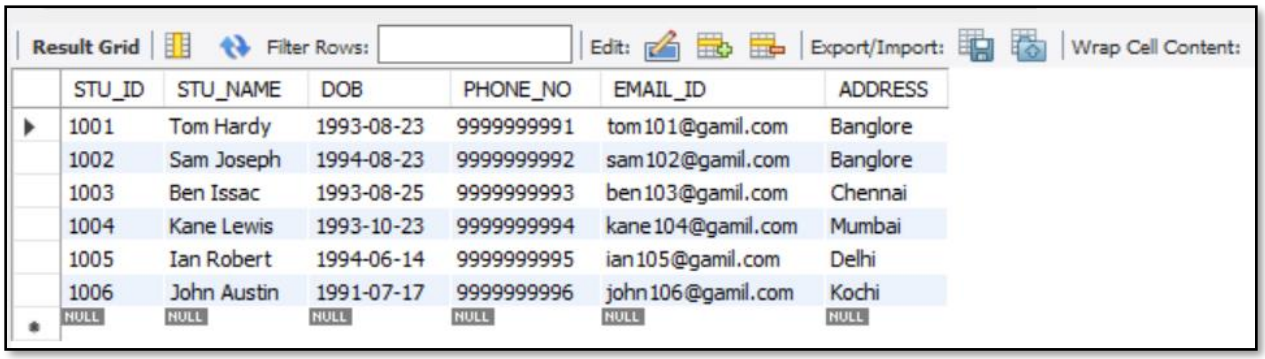


	COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME
▶	1	SQL	David
	2	Python	Artur
*	NULL	NULL	NULL

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

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

Select * From StudentInfo;

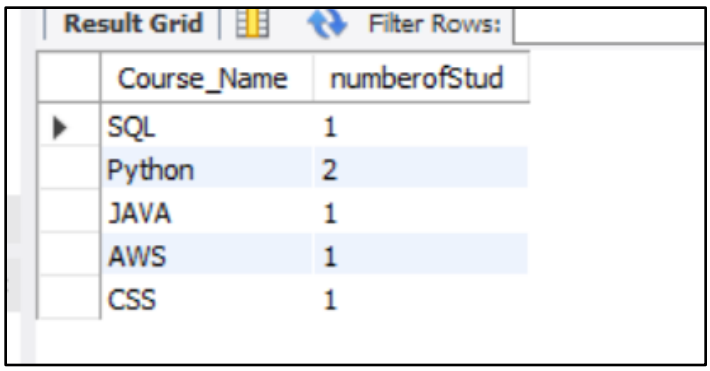


	STU_ID	STU_NAME	DOB	PHONE_NO	EMAIL_ID	ADDRESS
▶	1001	Tom Hardy	1993-08-23	9999999991	tom101@gamil.com	Banglore
	1002	Sam Joseph	1994-08-23	9999999992	sam102@gamil.com	Banglore
	1003	Ben Issac	1993-08-25	9999999993	ben103@gamil.com	Chennai
	1004	Kane Lewis	1993-10-23	9999999994	kane104@gamil.com	Mumbai
	1005	Ian Robert	1994-06-14	9999999995	ian105@gamil.com	Delhi
	1006	John Austin	1991-07-17	9999999996	john106@gamil.com	Kochi
*	NULL	NULL	NULL	NULL	NULL	NULL

4) Reporting and Analytics (Using joining queries)

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

```
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
```

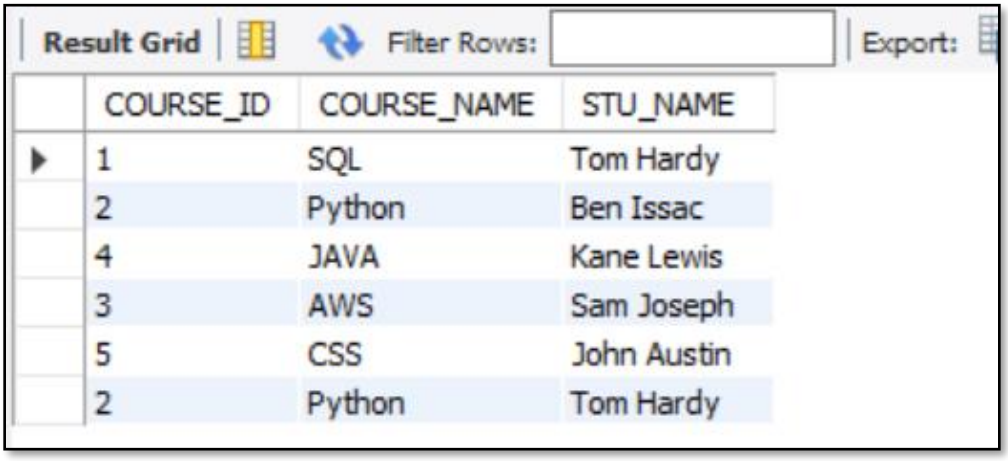


Result Grid | Filter Rows:

Course_Name	numberofStud
SQL	1
Python	2
JAVA	1
AWS	1
CSS	1

b) Write a query to retrieve the list of students enrolled in a specific course

```
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';
```



Result Grid | Filter Rows: | Export:

COURSE_ID	COURSE_NAME	STU_NAME
1	SQL	Tom Hardy
2	Python	Ben Issac
4	JAVA	Kane Lewis
3	AWS	Sam Joseph
5	CSS	John Austin
2	Python	Tom Hardy

c) Write a query to retrieve the count of enrolled students for each instructor

```
Select c.COURSE_INSTRUCTOR_NAME , count(e.Stu_id) as numberofStud
from CourseInfo c join EnrollmentInfo e
on c.course_id=e.course_ID
where e.enroll_status = 'ENROLLED'
group by 1;
```

	COURSE_INSTRUCTOR_NAME	numberofStud
▶	David	1
	Artur	2
	Harry	1
	Sebastian	1
	Jack	1

d) Write a query to retrieve the list of students who are enrolled in a multiple course

```
Select e.stu_id , 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
having count(c.course_id) >1
```

Result Grid | Filter Rows:

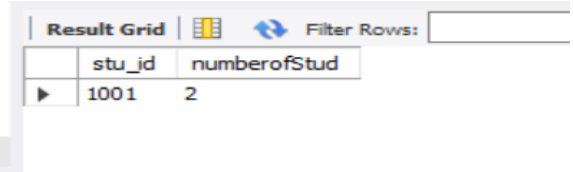
	stu_id	numberofStud
▶	1001	2

e) Write a query to retrieve the courses that have the highest number of enrolled students(arranging from highest to lowest)

```
Select e.stu_id , 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
```

having count(c.course_id) >1

order by count(c.course_id) desc



Task 2 (Postgresql)

-- Database: Student_Database

-- DROP DATABASE IF EXISTS "Student_Database";

1) Database setup

```
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;
```

```
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', 'Muskaan Arya', 'Business', 'muskaan@gmail.com', '9999999991',  'Delhi', '1999-04-26', 'Female',
'MBA', '8.8', 'A'),
 ('2', 'Kundan Kumar', 'Arts and Sciences', 'kundan@gmail.com',  '9999999992', 'Bangalore', '1992-07-15',
'Male', 'Mathematics', '8.6', 'A'),
```

('3', 'Rajat Nema', 'Business', 'Rajat@gmail.com', '9999999993', 'Delhi', '1995-06-28', 'Male', 'MBA', '8.5', 'A'),

('4', 'Devashish Negi', 'Arts and Sciences', 'Devashish@gmail.com', '9999999994', 'Dehradhun', '1997-01-12', 'Male', 'Physics', '7.6', 'B'),

('5', 'Karishma Roy', 'Arts and communication', 'Karishma@gmail.com', '9999999995', 'Mumbai', '1995-10-26', 'Female', 'Communication', '6.9', 'B'),

('6', 'Shivani Saini', 'Arts and Sciences', 'Shivani@gmail.com', '9999999996', 'Goa', '2000-01-03', 'Female', 'Computer Science', '5.5', 'C'),

('7', 'Himanshu Chawla', 'Human Development', 'Himanshu@gmail.com', '9999999997', 'Uttar Pradesh', '2000-10-05', 'Male', 'Counseling', '7.0', 'B'),

('8', 'Pranshu Yadav', 'Arts and communication', 'Pranshu@gmail.com', '9999999998', 'Gujarat', '1999-11-16', 'Male', 'Theatre', '9.2', 'A'),

('9', 'Sarthak Arya', 'Business', 'Sarthak@gmail.com', '9999999999', 'Rajasthan', '2003-08-14', 'Male', 'Accounting', '7.9', 'B'),

('10', 'Sanya Gambhir', 'Arts and Sciences', 'Sanya@gmail.com', '9999999911', 'Mumbai', '1997-03-12', 'Female', 'Physics', '4.9', 'C');

Select * from Student_table;

	student_id integer	stu_name character varying (100)	department character varying (50)	email_id character varying	phone_no numeric	address character varying	date_of_birth date	gender character var	major character varying (50)	gpa numeric	grade characte
1	1	Muskaan Arya	Business	muskaan@g...	9999999991	Delhi	1999-04-26	Female	MBA	8.8	A
2	2	Kundan Kumar	Arts and Sciences	kundan@gm...	9999999992	Bangalore	1992-07-15	Male	Mathematics	8.6	A
3	3	Rajat Nema	Business	Rajat@gmail....	9999999993	Delhi	1995-06-28	Male	MBA	8.5	A
4	4	Devashish Negi	Arts and Sciences	Devashish@...	9999999994	Dehradhun	1997-01-12	Male	Physics	7.6	B
5	5	Karishma Roy	Arts and communication	Karishma@g...	9999999995	Mumbai	1995-10-26	Female	Communication	6.9	B
6	6	Shivani Saini	Arts and Sciences	Shivani@gm...	9999999996	Goa	2000-01-03	Female	Computer Science	5.5	C
7	7	Himanshu Chawla	Human Development	Himanshu@...	9999999997	Uttar Pradesh	2000-10-05	Male	Counseling	7.0	B
8	8	Pranshu Yadav	Arts and communication	Pranshu@g...	9999999998	Gujarat	1999-11-16	Male	Theatre	9.2	A
9	9	Sarthak Arya	Business	Sarthak@gm...	9999999999	Rajasthan	2003-08-14	Male	Accounting	7.9	B
10	10	Sanya Gambhir	Arts and Sciences	Sanya@gmai...	9999999911	Mumbai	1997-03-12	Female	Physics	4.9	C

3) Student information retrieval

Select * from Student_table

Order By GPA desc, Grade;

	student_id integer	stu_name character varying	department character varying (5	email_id character varying (50)	phone_no numeric	address character varying (250)	date_of_birth date	gender character vary	major character vary	gpa numeric	grade character vary
1	8	Pranshu Yadav	Arts and comm...	Pranshu@gmail.com	9999999998	Gujarat	1999-11-16	Male	Theatre	9.2	A
2	1	Muskaan Arya	Business	muskaan@gmail.com	9999999991	Delhi	1999-04-26	Female	MBA	8.8	A
3	2	Kundan Kumar	Arts and Scienc...	kundan@gmail.com	9999999992	Bangalore	1992-07-15	Male	Mathema...	8.6	A
4	3	Rajat Nema	Business	Rajat@gmail.com	9999999993	Delhi	1995-06-28	Male	MBA	8.5	A
5	9	Sarthak Arya	Business	Sarthak@gmail.com	9999999999	Rajasthan	2003-08-14	Male	Accounting	7.9	B
6	4	Devashish Negi	Arts and Scienc...	Devashish@gmail.com	9999999994	Dehradhun	1997-01-12	Male	Physics	7.6	B
7	7	Himanshu Ch...	Human Develop...	Himanshu@gmail.com	9999999997	Uttar Pradesh	2000-10-05	Male	Counseling	7.0	B
8	5	Karishma Roy	Arts and comm...	Karishma@gmail.com	9999999995	Mumbai	1995-10-26	Female	Communi...	6.9	B
9	6	Shivani Saini	Arts and Scienc...	Shivani@gmail.com	9999999996	Goa	2000-01-03	Female	Computer...	5.5	C
10	10	Sanya Gambhir	Arts and Scienc...	Sanya@gmail.com	9999999911	Mumbai	1997-03-12	Female	Physics	4.9	C

4) Male students

```
Select * from Student_table

where Gender = 'Male'

;
```

	student_id integer	stu_name character varying (100)	department character varying (50)	email_id character varying (50)	phone_no numeric	address character vary	date_of_birth date	gender character vary	major character vary	gpa numeric	grade character vary
1	2	Kundan Kumar	Arts and Sciences	kundan@gmail.com	9999999992	Bangalore	1992-07-15	Male	Mathema...	8.6	A
2	3	Rajat Nema	Business	Rajat@gmail.com	9999999993	Delhi	1995-06-28	Male	MBA	8.5	A
3	4	Devashish Negi	Arts and Sciences	Devashish@gmail.com	9999999994	Dehradhun	1997-01-12	Male	Physics	7.6	B
4	7	Himanshu Chawla	Human Development	Himanshu@gmail.com	9999999997	Uttar Pra...	2000-10-05	Male	Counseling	7.0	B
5	8	Pranshu Yadav	Arts and communication	Pranshu@gmail.com	9999999998	Gujarat	1999-11-16	Male	Theatre	9.2	A
6	9	Sarthak Arya	Business	Sarthak@gmail.com	9999999999	Rajasthan	2003-08-14	Male	Accounting	7.9	B

5) GPA less than 5

```
Select * from Student_table

where GPA <5.0

;
```

	student_id integer	stu_name character vary	department character varying (50)	email_id character varying (50)	phone_no numeric	address character vary	date_of_birth date	gender character vary	major character varying (50)	gpa numeric	grade character vary
1	10	Sanya Ga...	Arts and Sciences	Sanya@gmail.com	9999999911	Mumbai	1997-03-12	Female	Physics	4.9	C

6) Update email_id and Grade

```
UPDATE Student_table

SET email_id = 'Sanyaa@gmail.com', Grade = 'D'

Where Student_id = '10';
```

7) Grade B

```
SELECT Stu_name, date_part('year',age(Date_Of_Birth)) as Age

FROM Student_table

Where Grade = 'B'

;
```

	stu_name character varying (100)	age double precision
1	Devashish Negi	26
2	Karishma Roy	28
3	Himanshu Chawla	23
4	Sarthak Arya	20

8) Grouping and calculation

```
Select Department, Gender, Avg(GPA)

From Student_table

Group By 1,2

;
```

	department character varying (50)	gender character varying (30)	avg numeric
1	Business	Male	8.2000000000000000
2	Arts and communication	Female	6.9000000000000000
3	Business	Female	8.8000000000000000
4	Arts and Sciences	Male	8.1000000000000000
5	Arts and communication	Male	9.2000000000000000
6	Human Development	Male	7.0000000000000000
7	Arts and Sciences	Female	5.2000000000000000

9) Renaming

Alter table Student_table

Rename to Student_info;

Select * from Student_info

	student_id integer	stu_name character varying (1	department character varying (50	email_id character vary	phone_no numeric	address character vary	date_of_birth date	gender character vary	major character vary	gpa numeric	grade character var
1	1	Muskaan Arya	Business	muskaan...	9999999991	Delhi	1999-04-26	Female	MBA	8.8	A
2	2	Kundan Kumar	Arts and Sciences	kundan@...	9999999992	Bangalore	1992-07-15	Male	Mathemati...	8.6	A
3	3	Rajat Nema	Business	Rajat@g...	9999999993	Delhi	1995-06-28	Male	MBA	8.5	A
4	4	Devashish Negi	Arts and Sciences	Devashis...	9999999994	Dehradhun	1997-01-12	Male	Physics	7.6	B
5	5	Karishma Roy	Arts and commun...	Karishma...	9999999995	Mumbai	1995-10-26	Female	Communic...	6.9	B
6	6	Shivani Saini	Arts and Sciences	Shivani@...	9999999996	Goa	2000-01-03	Female	Computer ...	5.5	C
7	7	Himanshu Chaw...	Human Developm...	Himansh...	9999999997	Uttar Pra...	2000-10-05	Male	Counseling	7.0	B
8	8	Pranshu Yadav	Arts and commun...	Pranshu...	9999999998	Gujarat	1999-11-16	Male	Theatre	9.2	A
9	9	Sarthak Arya	Business	Sarthak@...	9999999999	Rajasthan	2003-08-14	Male	Accounting	7.9	B
10	10	Sanya Gambhir	Arts and Sciences	Sanyaa@...	9999999911	Mumbai	1997-03-12	Female	Physics	4.9	D

10) Students with highest GPA

Select Stu_name, GPA

From Student_info where GPA = (Select Max(GPA) From Student_info)

;

	stu_name character varying (100)	gpa numeric
1	Pranshu Yadav	9.2

Task 3 (Postgresql)

```
-- Database: EventsManagement

-- DROP DATABASE IF EXISTS "EventsManagement";


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;
```

1) Database 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', 'Ed Sheeran', '2024-03-03', 'Mumbai', 'Music Show'),

('102', 'Comicon 2023', '2023-11-19', 'Bangalore', 'Exhibition'),

('103', 'IIMF', '2023-11-12', 'Kerala', 'Music Show'),

('104', 'Who are You', '2024-01-06', 'Delhi', 'Comedy'),

('105', 'Champak chacha', '2024-02-13', 'Kolkata', 'Comedy')

;

Select * from Events;

	event_id [PK] integer	event_name character varying (30)	event_date date	event_location character varying (100)	event_description character varying (200)
1	101	Ed Sheeran	2024-03-03	Mumbai	Music Show
2	102	Comicon 2023	2023-11-19	Bangalore	Exhibition
3	103	IIMF	2023-11-12	Kerala	Music Show
4	104	Who are You	2024-01-06	Delhi	Comedy
5	105	Champak chacha	2024-02-13	Kolkata	Comedy

Insert into Attendees (Attendee_Id, Attendee_Name, Attendee_Phone, Attendee_Email, Attendee_City) values

('1001', 'Muskaan', '9999999991', 'Muskaan@gmail.com', 'Delhi'),

('1002', 'Rajat', '9999999992', 'Rajat@gmail.com', 'Delhi'),

('1003', 'Abhinav', '9999999993', 'Abhinav@gmail.com', 'Mumbai'),

('1004', 'Kundan', '9999999994', 'Kundan@gmail.com', 'Uttar Pradesh'),

('1005', 'Karishma', '9999999995', 'Karishma@gmail.com', 'Mumbai'),

('1006', 'Shivani', '9999999996', 'Shivani@gmail.com', 'Goa'),

('1007', 'Devashish', '9999999997', 'Devashish@gmail.com', 'Uttrakhand'),

('1008', 'Sarthak', '9999999998', 'Sarthak@gmail.com', 'Delhi')

;

Select * from Attendees;

	attendee_id [PK] integer	attendee_name character varying (30)	attendee_phone numeric	attendee_email character varying (30)	attendee_city character varying (20)
1	1001	Muskaan	9999999991	Muskaan@gmail.com	Delhi
2	1002	Rajat	9999999992	Rajat@gmail.com	Delhi
3	1003	Abhinav	9999999993	Abhinav@gmail.com	Mumbai
4	1004	Kundan	9999999994	Kundan@gmail.com	Uttar Pradesh
5	1005	Karishma	9999999995	Karishma@gmail.com	Mumbai
6	1006	Shivani	9999999996	Shivani@gmail.com	Goa
7	1007	Devashish	9999999997	Devashish@gmail.com	Uttrakhand
8	1008	Sarthak	9999999998	Sarthak@gmail.com	Delhi

Insert into Registrations

(Registration_Id, Event_Id, Attendee_Id, Registration_Date, Registration_Amount) values

('10001', '101', '1001', '2023-10-12', '7500'),

('10002', '102', '1003', '2023-09-25', '1000'),

('10003', '103', '1002', '2023-10-29', '2000'),

('10004', '104', '1004', '2023-10-20', '500'),

('10005', '101', '1005', '2023-09-10', '7500'),

('10006', '103', '1008', '2023-09-15', '2000'),

('10007', '101', '1007', '2023-11-01', '7500'),

('10008', '102', '1006', '2023-11-05', '1000')

;

Select *from Registrations;

	registration_id [PK] integer	event_id integer	attendee_id integer	registration_date date	registration_amount numeric
1	10001	101	1001	2023-10-12	7500
2	10002	102	1003	2023-09-25	1000
3	10003	103	1002	2023-10-29	2000
4	10004	104	1004	2023-10-20	500
5	10005	101	1005	2023-09-10	7500
6	10006	103	1008	2023-09-15	2000
7	10007	101	1007	2023-11-01	7500
8	10008	102	1006	2023-11-05	1000

3) Manage Event Details

a) Insert New event

```
Insert INTO Events (Event_Id, Event_Name, Event_Date, Event_Location, Event_Description) VALUES
('106', 'Unheard Diaries', '2023-10-29', 'Delhi', 'Storytelling');
```

b) Update event's information

```
UPDATE Events
SET Event_Location = 'Bangalore' Where Event_Id = '104'
;
```

c) Deleting an event

```
Delete from Events where Event_Id = '105'
;
```

4) Manage Track attendees and handle events

a) Insert new attendee

```
Insert into Attendees (Attendee_Id, Attendee_Name, Attendee_Phone, Attendee_Email, Attendee_City)
values ('1009', 'Krishna', '9999999999', 'Krishna@gmail.com', 'Tamil Nadu');
```

b) Register attendee

```
Insert into Registrations (Registration_Id, Event_Id, Attendee_Id, Registration_Date, Registration_Amount)
values ('10009', '101', '1009', '2023-11-11', '7500');
```

5) Retrieve event information, Generate attendee list, Calculate event attendee statistics

```
Select * from Events;
Select * from Attendees;
select * from Registrations;
```

```
with Event1 as(
select E.Event_id, E.event_name, E.event_date, E.event_location,
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
from Event1
group by 1,2,3,4
```

	event_id [PK] integer	event_name character varying (30)	event_location character varying (100)	amountgenperevent numeric
1	103	IIMF	Kerala	4000
2	101	Ed Sheeran	Mumbai	30000
3	104	Who are You	Bangalore	500
4	102	Comicon 2023	Bangalore	2000

Task 4 (Postgresql)

-- Database: Sales Data

-- DROP DATABASE IF EXISTS "Sales Data ";

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;

1) Database 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', 'East', '2023-10-10', '20000'),

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

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

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

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

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

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

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

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

('1', 'West', '2023-09-29', '70000')

;

Select * from Sales_Sample;

	product_id integer	region character varying (50)	on_date date	sales_amount numeric
1	1	East	2023-10-10	20000
2	2	West	2023-09-19	50000
3	2	East	2023-10-21	40000
4	3	North	2023-09-20	15000
5	4	North	2023-08-06	45000
6	2	South	2023-08-25	45000
7	5	North	2023-11-23	20000
8	5	West	2023-11-11	60000
9	3	East	2023-09-19	50000
10	1	West	2023-09-29	70000

3) 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

;

	region character varying (50)	product_id integer	sales_amount numeric
1	East	1	20000
2	East	2	40000
3	East	3	50000
4	North	3	15000
5	North	4	45000
6	North	5	20000
7	South	2	45000
8	West	1	70000
9	West	2	50000
10	West	5	60000

b) Roll Up

Select Region, Product_Id, Sum(Sales_Amount) as Sales_Amount

From Sales_Sample

Group By Rollup (1,2)

Order By Region

;

	region character varying (50)	product_id integer	sales_amount numeric
1	East	1	20000
2	East	2	40000
3	East	3	50000
4	East	[null]	110000
5	North	3	15000
6	North	4	45000
7	North	5	20000
8	North	[null]	80000
9	South	2	45000
10	South	[null]	45000
11	West	1	70000
12	West	2	50000
13	West	5	60000
14	West	[null]	180000
15	[null]	[null]	415000

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

;

	region character varying (50)	product_id integer	on_date date	sales_amount numeric
1	East	1	2023-10-10	20000
2	East	1	[null]	20000
3	East	2	2023-10-21	40000
4	East	2	[null]	40000
5	East	3	2023-09-19	50000
6	East	3	[null]	50000
7	East	[null]	2023-09-19	50000
8	East	[null]	2023-10-10	20000
9	East	[null]	2023-10-21	40000
10	East	[null]	[null]	110000
11	North	3	2023-09-20	15000
12	North	3	[null]	15000
13	North	4	2023-08-06	45000
14	North	4	[null]	45000
15	North	5	2023-11-23	20000
16	North	5	[null]	20000

d) Slice

Select Region, Product_Id, On_Date, Sum(Sales_Amount) as Sales_Amount

From Sales_Sample

Where Region in('North', 'South') 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

;

	region character varying (50)	product_id integer	on_date date	sales_amount numeric
1	East	1	2023-10-10	20000
2	East	3	2023-09-19	50000
3	North	3	2023-09-20	15000
4	North	4	2023-08-06	45000
5	North	5	2023-11-23	20000
6	South	2	2023-08-25	45000
7	West	1	2023-09-29	70000
8	West	2	2023-09-19	50000

e) Dice

Select Region, Product_Id, On_Date, Sum(Sales_Amount) as Sales_Amount

From Sales_Sample

Where Region in('North', 'South') 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

;

	region character varying (50)	product_id integer	on_date date	sales_amount numeric
1	South	2	2023-08-25	45000