

ASSIGNMENT 3

STUDENT MANAGEMENT SYSTEM

INSERTIONS:

```
insert into
students(first_name,last_name,date_of_birth,email,phone_number) values
('Abinaya','Shri','2002-03-21','abinayashri12@gmail.com','23456764'),
('harry','potter','2003-05-08','harrypotter1@gmail.com','64743748'),
('Anand ','Murthy','2002-07-28','anandmurthy@gmail.com','67463784'),
('Priya','Dharshini','2003-07-02','priyadharshini4@gmail.com','8496764'),
('chotta','bheem','2003-06-04','cottabheem@gmail.com','23457485'),
('Jaya','krishnan','2002-04-30','jayakrishnan2@gmail.com','7483784');
select * from students;
```

```
insert into teachers(first_name,last_name,email) values
('Bala','ganesh','balaganesh@gmail.com'),
('Deepika','gopal','deepika@gmail.com'),
('Anitha','krishnan','anitha@gmail.com'),
('Harish','selvan','harish@gmail.com'),
('lavanya','shri','lavanya@gmail.com');
select * from teachers;
```

```
insert into payments(amount,payment_date,student_id) values
(50000,'2024-03-12',1),
(55000,'2024-01-24',2),
(60000,'2024-01-18',3),
(90000,'2024-04-05',4),
(54550,'2024-02-19',5),
```

```
(80000,'2024-03-30',6);
```

```
select * from payments;
```

```
insert into courses(course_name,credits,teacher_id) values
```

```
('Python programming','3',3),
```

```
('Database Management Systems','4',4),
```

```
('Cybersecurity Fundamentals','3',2),
```

```
('Network Administration','3',5),
```

```
('Cloud Computing','3',1),
```

```
('Machine Learning','4',4);
```

```
select * from courses;
```

```
insert into enrollments(enrollment_id,enrollment_date,student_id,course_id)  
values
```

```
(1,'2023-05-13', 1 , 5),
```

```
(2,'2023-05-13', 3 , 6),
```

```
(3,'2023-05-13', 6 , 4),
```

```
(4,'2023-05-13', 5 , 1),
```

```
(5,'2023-05-13', 2 , 3);
```

```
select * from enrollments;
```

QUERIES:

```
-- TASK 2 --
```

```
-- 1.query to insert a new student into the student table of students details
```

```
-- firstname : John, lastname : Doe ,date of birth :'1995-08-15', email:
```

```
john.doe@example.com, phone number:1234567
```

```
insert into
students(first_name,last_name,date_of_birth,email,phone_number) values
('John','Doe','1995-08-15','john.doe@example.com','1234567');

select * from students;
```

-- 2. query to enroll a student in a course. Choose an existing student and course

```
-- and insert a record into the "Enrollments"with enrollment_date

insert into enrollments(enrollment_id,enrollment_date,student_id,course_id)
values
(6,'2023-03-24', 4 , 2);

select * from enrollments;
```

-- 3. query to update the email address of a specific teacher in the teachers table.

```
update teachers set email = 'deepikagopal@gmail.com' where teacher_id = 2;

select * from teachers;
```

-- 4. query to delete a specific enrollment record from enrollment table

```
delete from enrollments where course_id = 2 and student_id = 4;
```

-- 5. update courses table to assign a

-- specific teacher to a course. choose any course and teacher from the respective table

```
update courses set teacher_id = 2 where course_name = 'Database
Management Systems';
```

-- 6.delete a specific student from the students table and remove all their enrollment record

-- from the enrollment table be sure to maintain referential integrity

delete from enrollments

where student_id = 6;

delete from students

where student_id = 6;

-- 7.update the payment amount for a specific payment record in the payments table.

-- choose any payment record and modify the payment amount

update payments set amount = 70000 where payment_id = 2;

-- TASK 3 --

-- 1.query to calculate total payments made by specific student

-- need to join the payments table with students table based on students id

select s.first_name,sum(p.amount)

from students s join payments p on s.student_id = p.student_id

where s.student_id = 2;

OUTPUT:

first_name	sum(p.amount)
------------	---------------

harry	70000
-------	-------

-- 2.query to retrieve a list of courses along with count of students enrolled in each course

-- use a join operation between course and enrollments

select c.course_name, count(student_id) as no_of_students

```

from courses c join enrollments e on c.course_id = e.course_id
group by c.course_name;

```

OUTPUT:

course_name	no_of_students
Cloud Computing	1
Cybersecurity Fundamentals	1
Machine Learning	1
Python programming	1

-- 3. query to find names of students who have not enrolled in any course

-- use left join between student and enrollments

```
select *
```

```
from students s left join enrollments e on e.student_id = s.student_id
```

```
where e.student_id is null;
```

OUTPUT:

student_id	first_name	last_name	date_of_birth	email	phone_number	student_id	course_id	enrollment_id	enrollment_date
4	Priya	Dharsini	2003-07-02	priyadharshini4@gmail.com	8496764				
6	Jaya	krishnan	2002-04-30	jayakrishnan2@gmail.com	7483784				
7	John	Doe	1995-08-15	john.doe@example.com	1234567				

-- 4. query to retrieve the first name last name of students and the names of the courses

-- they are enrolled in .use joins between enroll, course and student

```
select s.first_name,s.last_name,c.course_name
```

```
from students s join enrollments e on s.student_id = e.student_id
```

```
join courses c on c.course_id = e.course_id;
```

OUTPUT:

first_name	last_name	course_name
Abinaya	Shri	Cloud Computing
Anand	Murthy	Machine Learning
chotta	bheem	Python programming
harry	potter	Cybersecurity Fundamentals

-- 5. query to list the names of teachers and the courses they are assigned to

```
select t.first_name,t.last_name,c.course_name
```

```
from teachers t join courses c on c.teacher_id = t.teacher_id;
```

OUTPUT:

first_name	last_name	course_name
Bala	ganesh	Cloud Computing
Deepika	gopal	Database Management Systems
Deepika	gopal	Cybersecurity Fundamentals
Anitha	krishnan	Python programming
Harish	selvan	Machine Learning
lavanya	shri	Network Administration

-- 6. retrieve a list of students and their enrollment dates for a specific course

```
select s.*,e.enrollment_date,c.course_name
```

```
from students s join enrollments e on s.student_id = e.student_id
```

```
join courses c on c.course_id = e.course_id
```

```
where c.course_name = 'Cloud Computing';
```

OUTPUT:

student_id	first_name	last_name	date_of_birth	email	phone_number	enrollment_date	course_name
1	Abinaya	Shri	2002-03-21	abinayashri12@gmail.com	23456764	2023-05-13	Cloud Computing

-- 7. name of students who have not made any payments

select *

from students s left join payments p on s.student_id = p.student_id

where p.student_id is null;

OUTPUT:

student_id	first_name	last_name	date_of_birth	email	phone_number	payment_id	amount	student_id	payment_date
7	John	Doe	1995-08-15	john.doe@example.com	1234567				

-- 8. query to identify courses that have no enrollments .use left join

select *

from courses c left join enrollments e on c.course_id = e.course_id

where e.course_id is null;

OUTPUT:

course_id	course_name	credits	teacher_id	student_id	course_id	enrollment_id	enrollment_date
2	Database Management Systems	4	2				
4	Network Administration	3	5				

-- 9. identify students who are enrolled in more than one course

select s.*

from students s join enrollments e on s.student_id = e.student_id

join courses c on c.course_id = e.course_id

group by s.student_id

```
having count(e.course_id) > 1;
```

```
-- 10. teachers who are not assigned to any courses
```

```
select *
```

```
from teachers t left join courses c on c.teacher_id = t.teacher_id
```

```
where c.teacher_id is null;
```

```
-- TASK 4 --
```

```
-- 1. query to calculate the avg no of students enrolled in each course
```

```
select c.course_name, (count(s.student_id)/(select count(*) from courses)) as  
avg_students
```

```
from students s join enrollments e on s.student_id = e.student_id
```

```
join courses c on c.course_id = e.course_id
```

```
group by c.course_name;
```

OUTPUT:

course_name	avg_students
Cloud Computing	0.1667
Cybersecurity Fundamentals	0.1667
Machine Learning	0.1667
Python programming	0.1667

```
-- 2. query to identify the student who made the highest payment.use a  
subquery to find the maximum payment amount
```

```
-- and then retrieve the students associated with that amount
```

```
-- projection : students
```

```
-- criteria : payments
```

```
select *
```

```
from students s join payments p on s.student_id = p.student_id
```

```
where p.amount in (select max(amount) from payments);
```


OUTPUT:

stude nt_id	first_n ame	last_n ame	date_of _birth	email	phone_n umber	payme nt_id	amo unt	stude nt_id	payment _date
4	Priya	Dhars hini	2003- 07-02	priyadharshini4 @gmail.com	8496764	4	9000 0	4	2024-04- 05

-- 3.retrieve a list of course with the highest number of enrollments use subqueries to find the courses with max enrollment count

```
select c.course_name,count(*)
from courses c join enrollments e on c.course_id = e.course_id
group by c.course_name
order by count(*) desc;
```

OUTPUT:

course_name	count(*)
Python programming	1
Cybersecurity Fundamentals	1
Cloud Computing	1
Machine Learning	1

-- 4. calculate total payments made to courses taught by each teacher.

```
-- payment,courses, teachers
select t.first_name,t.last_name,sum(p.amount) as
total_payment,c.course_name
from teachers t
JOIN courses c ON t.teacher_id = c.teacher_id
JOIN enrollments e ON c.course_id = e.course_id
JOIN payments p ON e.student_id = p.student_id
group by t.teacher_id;
```

OUTPUT:

first_name	last_name	total_payment	course_name
Bala	ganesh	50000	Cloud Computing
Deepika	gopal	70000	Cybersecurity Fundamentals
Anitha	krishnan	54550	Python programming
Harish	selvan	60000	Machine Learning

```
-- 5.identify students who are enrolled in all available courses
-- use sub query to compare the students enrollment with total no of courses
-- projection: students
-- criteria: courses
```

```
select *
from students s join enrollments e on s.student_id = e.student_id
                join courses c on c.course_id = e.course_id
where s.student_id = ALL (select course_name from courses);
```

```
SELECT s.*
FROM students s
WHERE (SELECT COUNT(DISTINCT e.course_id) FROM enrollments e) = (SELECT
COUNT(*) FROM courses);
```

```
SELECT COUNT(DISTINCT e.course_id) FROM enrollments e;
SELECT COUNT(*) FROM courses;
```

OUTPUT:

student_id	first_name	last_name	date_of_birth	email	phone_number
-------------------	-------------------	------------------	----------------------	--------------	---------------------

```
-- 6. retrieve name of teachers who have not been assigned to any courses
select *
```

```

from teachers
where teacher_id not in (select t.teacher_id
                        from teachers t join courses c on
                        c.teacher_id = t.teacher_id);

```

OUTPUT:

```

teacher_id first_name last_name email

```

-- 7.average age of all students

```

select avg(2024-year(date_of_birth)) as average_age
from students;

```

OUTPUT:

```

average_age
22.5714

```

-- 8.identify courses with no enrollment.

```

select *
from courses
where course_id not in (select c.course_id
                        from courses c join enrollments e on c.course_id = e.course_id);

```

OUTPUT:

course_id	course_name	credits	teacher_id
2	Database Management Systems	4	2
4	Network Administration	3	5

-- 9.calculate the total payments made by each student for each course they are enrolled insert

```

select s.first_name,s.last_name,sum(p.amount) as
total_payment,c.course_name

```

```

from students s
JOIN enrollments e ON s.student_id = e.student_id
JOIN courses c ON e.course_id = c.course_id
JOIN payments p ON s.student_id = p.student_id
group by s.student_id,c.course_name;

```

OUTPUT:

first_name	last_name	total_payment	course_name
Abinaya	Shri	50000	Cloud Computing
harry	potter	70000	Cybersecurity Fundamentals
Anand	Murthy	60000	Machine Learning
chotta	bheem	54550	Python programming

-- 10. identify students who have made more than one payment

```

select *,count(p.payment_id) as num_payments
from students s join payments p on s.student_id = p.student_id
group by s.student_id
having count(p.payment_id) > 1;

```

-- 11. query to calculate the total payments made by each student

```

select *,sum(p.amount) as tot_payment
from students s join payments p on s.student_id = p.student_id
group by s.student_id;

```

OUTPUT:

stude nt_id	first_ name	last_ name	date_of _birth	email	phone_ number	paym ent_id	am oun t	stude nt_id	payme nt_date	tot_pa yment
1	Abina ya	Shri	2002- 03-21	abinayashri12 @gmail.com	2345676 4	1	500 00	1	2024- 03-12	50000

stude nt_id	first_ name	last_ name	date_of _birth	email	phone_ number	paym ent_id	am oun t	stude nt_id	payme nt_date	tot_pa yment
2	harry	potter	2003-05-08	harrypotter1@gmail.com	64743748	2	70000	2	2024-01-24	70000
3	Anand	Murthy	2002-07-28	anandmurthy@gmail.com	67463784	3	60000	3	2024-01-18	60000
4	Priya	Dharsini	2003-07-02	priyadharshini4@gmail.com	8496764	4	90000	4	2024-04-05	90000
5	chotta	bheem	2003-06-04	cottabheem@gmail.com	23457485	5	54550	5	2024-02-19	54550
6	Jaya	krishnan	2002-04-30	jayakrishnan2@gmail.com	7483784	6	80000	6	2024-03-30	80000

-- 12. retrieve a list of courses names along with the count of students enrolled in each course

```
select c.*,count(s.student_id) as tot_students
```

```
from students s join enrollments e on s.student_id = e.student_id
```

```
join courses c on c.course_id = e.course_id
```

```
group by c.course_name;
```

OUTPUT:

course_id	course_name	credits	teacher_id	tot_students
5	Cloud Computing	3	1	1
3	Cybersecurity Fundamentals	3	2	1
6	Machine Learning	4	4	1
1	Python programming	3	3	1

-- 13. average payment amount made by students

```
select *,avg(p.amount) as avg_payment
```

```
from students s join payments p on s.student_id = p.student_id
```

```
group by s.student_id;
```

OUTPUT:

stude nt_id	first_ name	last_ name	date_of _birth	email	phone_ number	paym ent_id	am oun t	stude nt_id	payme nt_date	avg_pa yment
1	Abina ya	Shri	2002- 03-21	abinayashri12 @gmail.com	2345676 4	1	500 00	1	2024- 03-12	50000
2	harry	potter	2003- 05-08	harrypotter1@ gmail.com	6474374 8	2	700 00	2	2024- 01-24	70000
3	Anan d	Murt hy	2002- 07-28	anandmurthy@ gmail.com	6746378 4	3	600 00	3	2024- 01-18	60000
4	Priya	Dhars hini	2003- 07-02	priyadharshini4 @gmail.com	8496764	4	900 00	4	2024- 04-05	90000
5	chotta	bhee m	2003- 06-04	cottabheem@g mail.com	2345748 5	5	545 50	5	2024- 02-19	54550
6	Jaya	krish nan	2002- 04-30	jayakrishnan2 @gmail.com	7483784	6	800 00	6	2024- 03-30	80000