```
show databases;
use sisdb;

-- insertions
insert into students (student_id,first_name,last_name,date_of_birth,email,phone_number)
values
( 1 ,'harry','potter','2002-09-12','har@gmail.com',9897654),
(2, 'Hermione', 'Granger', '2002-07-30', 'hermione@gmail.com', 9876543),
(3, 'Ron', 'Weasley', '2002-03-01', 'ron@gmail.com', 9765432),
(4, 'Luna', 'Lovegood', '2002-02-13', 'luna@gmail.com', 9654321),
(5, 'Neville', 'Longbottom', '2002-07-30', 'neville@gmail.com', 9543210),
(6, 'Ginny', 'Weasley', '2003-08-11', 'ginny@gmail.com', 9432109),
(7, 'Draco', 'Malfoy', '2001-06-05', 'draco@gmail.com', 9321098),
(8, 'Fred', 'Weasley', '2002-04-01', 'fred@gmail.com', 9210987),
(9, 'George', 'Weasley', '2002-04-01', 'george@gmail.com', 9109876),
(10, 'Cho', 'Chang', '2002-09-15', 'cho@gmail.com', 9876543);

select * from students;

insert into teacher (teacher_id,first_name,last_name,email)
values
(1, 'Severus', 'Snape', 'snape@example.com'),
(2, 'Minerva', 'McGonagall', 'mcgonagall@example.com'),
(3, 'Albus', 'Dumbledore', 'dumbledore@example.com'),
(4, 'Remus', 'Lupin', 'lupin@example.com'),
(5, 'Sybill', 'Trelawney', 'trelawney@example.com'),
(6, 'Filius', 'Flitwick', 'flitwick@example.com'),
(7, 'Pomona', 'Sprout', 'sprout@example.com'),
(8, 'Rubeus', 'Hagrid', 'hagrid@example.com'),
(9, 'Gilderoy', 'Lockhart', 'lockhart@example.com'),
(10, 'Quirinus', 'Quirrell', 'quirrell@example.com');

select * from teacher;

insert into courses(course_id,course_name,credits,teacher_id)
values
(1, 'Potions', 3, 1),
(2, 'Transfiguration', 4, 2),
(3, 'Defense Against the Dark Arts', 4, 4),
(4, 'Charms', 3, 6),
(5, 'Herbology', 3, 7),
(6, 'Care of Magical Creatures', 2, 8),
(7, 'Divination', 2, 5),
(8, 'History of Magic', 3, 3),
(9, 'Flying', 2, 10),
(10, 'Alchemy', 4, 9);

select * from courses;

insert into enrollments(enrollment_id,enrollment_date,student_id,course_id)
values
(1, '2024-04-09', 1, 1),
(2, '2024-04-09', 1, 2),
(3, '2024-04-09', 1, 3),
(4, '2024-04-09', 2, 4),
(5, ’2024-04-09’, 2, 5),
(6, ’2024-04-09’, 3, 6),
(7, ’2024-04-09’, 3, 7),
(8, ’2024-04-09’, 4, 8),
(9, ’2024-04-09’, 4, 9),
(10, ’2024-04-09’, 5, 10);
select * from enrollments;

ALTER TABLE payments
MODIFY COLUMN amount INT;

describe payments;

insert into payments(payment_id,amount,student_id)
values
(1, 100, 1),
(2, 150, 2),
(3, 200, 3),
(4, 120, 4),
(5, 180, 5),
(6, 220, 1),
(7, 130, 2),
(8, 190, 3),
(9, 140, 4),
(10, 210, 5);
select * from payments;

-- Task 2
-- 1. Write an sql query to insert a new student into the students table with the following details
insert into students(student_id,first_name,last_name,date_of_birth,email,phone_number)
values
(11,’John’,’Doe’,’1995-08-15’,’john.doe@example.com’,1234567890);
select * from students;
/*
Output
1 harry potter 2002-09-12 har@gmail.com 9897654
2 Hermione Granger 2002-07-30 hermione@gmail.com 9876543
3 Ron Weasley 2002-03-01 ron@gmail.com 9765432
4 Luna Lovegood 2002-02-13 luna@gmail.com 9654321
5 Neville Longbottom 2002-07-30 neville@gmail.com 9543210
6 Ginny Weasley 2003-08-11 ginny@gmail.com 9432109
7 Draco Malfoy 2001-06-05 draco@gmail.com 9321098
8 Fred Weasley 2002-04-01 fred@gmail.com 9210987
9 George Weasley 2002-04-01 george@gmail.com 9109876
*/



-- 2.Write an sql query to enroll a student in a course
insert into enrollments (enrollment_id,enrollment_date,student_id,course_id)
values
(11,’2001-09-11’,11,1);
select * from enrollments;
/*
Output
```

```
1 2024-04-09 1 1
2 2024-04-09 1 2
3 2024-04-09 1 3
4 2024-04-09 2 4
5 2024-04-09 2 5
6 2024-04-09 3 6
7 2024-04-09 3 7
8 2024-04-09 4 8
9 2024-04-09 4 9
10 2024-04-09 5 10
*/



-- 3. Update the email address of a specific teacher in a teacher table;
update teacher
set email='severus@example.com'
where teacher_id=1;
select * from teacher;
/*
Output
1 Severus Snape severus@example.com
2 Minerva McGonagall mcgonagall@example.com
3 Albus Dumbledore dumbledore@example.com
4 Remus Lupin lupin@example.com
5 Sybill Trelawney trelawney@example.com
6 Filius Flitwick flitwick@example.com
7 Pomona Sprout sprout@example.com
8 Rubeus Hagrid hagrid@example.com
9 Gilderoy Lockhart lockhart@example.com
10 Quirinus Quirrell quirrell@example.com
*/



-- 4. Write an sql query to delete a specific enrollment record from the Enrollments table.
delete from
enrollments where enrollment_id=11;
select * from enrollments;
/*
Output
*/



-- 5. Update the courses table to assign a specific teacher to a course.
update courses
set teacher_id=7
where teacher_id=9;
select * from courses;
/*
Output
*/
-- 6. Delete a specific student from the students table and remove all their enrollments record.
delete from
students where student_id=11;
```

```
select * from students;
/*
Output
*/
-- 7. Update the payment amount for a specific payment record in the payments table.
update payments
set amount=amount+40
where payment_id=7;
select * from payments;
/*
Output
1 100 1
2 150 2
3 200 3
4 120 4
5 180 5
6 220 1
7 170 2
8 190 3
9 140 4
10 210 5
*/

-- Task 3
-- 1.Write an sql query to calculate the total payments made by a specific student.
select s.first_name,s.last_name,sum(p.amount) as total_payment
from students s JOIN payments p ON s.student_id=p.student_id
where s.student_id =1;
/*
Output
harry potter 320
*/


-- 2. Write an sql query to retrieve a list of courses along with the count of students enrolled in each
course .p-courses,c-enrollments
select c.course_id, c.course_name, count(e.student_id) AS student_count
from courses c JOIN enrollments e ON c.course_id=e.course_id
group by c.course_name;
/*
Output
10 Alchemy 1
6 Care of Magical Creatures 1
4 Charms 1
3 Defense Against the Dark Arts 1
7 Divination 1
9 Flying 1
5 Herbology 1
8 History of Magic 1
1 Potions 1
2 Transfiguration 1
*/


-- 3. Write an sql query to find the names of students who have not enrolled in any
```

```
course.p:students,c:enrollments
select s.*
from students s
where s.student_id NOT IN (select e.student_id
from enrollments e LEFT JOIN students s  ON s.student_id=e.student_id);
/*
Output
6 Ginny Weasley 2003-08-11 ginny@gmail.com 9432109
7 Draco Malfoy 2001-06-05 draco@gmail.com 9321098
8 Fred Weasley 2002-04-01 fred@gmail.com 9210987
9 George Weasley 2002-04-01 george@gmail.com 9109876
10 Cho Chang 2002-09-15 cho@gmail.com 9876543
*/



-- 4. Write an sql query to retrieve the first_name,last_name of students, and the names of the courses
they are enrolled in.p:students,courses,c:enrollent
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
*/
-- 5. Create a query to list the names of teachers and the courses they are assigned
to.p:teacher,c:courses
select t.first_name,t.last_name,c.course_name
from teacher t JOIN courses c ON t.teacher_id=c.teacher_id;
/*
Output
harry potter Potions
harry potter Transfiguration
harry potter Defense Against the Dark Arts
Hermione Granger Charms
Hermione Granger Herbology
Ron Weasley Care of Magical Creatures
Ron Weasley Divination
Luna Lovegood History of Magic
Luna Lovegood Flying
Neville Longbottom Alchemy
*/

-- 6. Retrieve a list of students and their enrollment dates for a specific
course.p:students,enrollments,c:course
select s.first_name,s.last_name,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='Potions';
 /*
Output
harry potter 2024-04-09 Potions
*/


-- 7. Find the names of students who have not made any payments.
```

```
select *
from students s LEFT JOIN payments p ON s.student_id=p.payment_id
where p.student_id is NULL;
/*
Output

No Output
*/



-- 8. Write a query to identify courses that have no enrollments.
select c.*
from courses c
where c.course_id NOT IN (select e.course_id
from  enrollments e LEFT JOIN courses c ON c.course_id=e.course_id);

SELECT c.course_id,c.course_name
FROM courses c
LEFT JOIN enrollments e ON c.course_id = e.course_id
WHERE e.course_id IS NULL;
/*
Output

No Output
*/



-- 9. Identify students who are enrolled in more than one course.
select s.first_name,s.last_name,count(*) as count
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>1;
/*
Output
harry potter 3
Hermione Granger 2
Ron Weasley 2
Luna Lovegood 2
*/



-- 10. Find teachers who are not assigned to any courses.
select t.teacher_id,t.first_name,t.last_name
from teacher t
where t.teacher_id NOT IN (select c.teacher_id
      from courses c LEFT JOIN teacher t ON t.teacher_id=c.teacher_id);
/*
Output
9 Gilderoy Lockhart
*/




-- Task 4
-- 2.Identify the students who made the highest payment.
SELECT s.student_id,s.first_name,s.last_name,p.amount AS highest_payment
FROM students s JOIN payments p ON s.student_id = p.student_id
ORDER BY p.amount DESC
LIMIT 1;
/*
Output
1 harry potter 220
*/



-- 3. Retrieve a list of courses with the highest no. of enrollments.
SELECT c.course_id,c.course_name,COUNT(e.student_id) AS enrollment_count
FROM courses c
JOIN enrollments e ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name
ORDER BY enrollment_count DESC
limit 1;
/*
Output
1 Potions 1
*/




-- 4.Calculate the total payments made to courses taught by each teacher. Use subqueries to sum
payments for each teacher’s courses.
SELECT t.teacher_id,t.first_name,t.last_name,SUM(p.amount) AS total_payments
FROM teacher 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
1 Severus Snape 320
2 Minerva McGonagall 320
3 Albus Dumbledore 260
4 Remus Lupin 320
5 Sybill Trelawney 390
6 Filius Flitwick 320
7 Pomona Sprout 710
8 Rubeus Hagrid 390
10 Quirinus Quirrell 260
*/


-- 6. Retrieve the names of teachers who have not been assigned to any courses. Use subqueries to find
teachers with no course assignments.
SELECT t.first_name,t.last_name
FROM teacher t
WHERE t.teacher_id NOT IN (SELECT DISTINCT teacher_id FROM courses);
/*
```

```
Output
Gilderoy Lockhart
*/


-- 7. Calculate the average age of all students. Use subqueries to calculate the age of each student based
on their date of birth.
SELECT AVG(TIMESTAMPDIFF(YEAR, date_of_birth, CURDATE())) AS average_age
FROM students;
/*
Output
21.400
*/




-- 8. Identify courses with no enrollments. Use subqueries to find courses without enrollment records.
SELECT course_id,course_name
FROM courses
WHERE course_id NOT IN (SELECT DISTINCT course_id FROM enrollments);
/*
Output

No Output
*/



-- 9. Calculate the total payments made by each student for each course they are enrolled in. Use
subqueries and aggregate functions to sum payments.
SELECT e.student_id,e.course_id,SUM(p.amount) AS total_payments
FROM enrollments e
LEFT JOIN payments p ON e.student_id = p.student_id
GROUP BY e.student_id, e.course_id;
/*
Output
1 1 320
1 2 320
1 3 320
2 4 320
2 5 320
3 6 390
3 7 390
4 8 260
4 9 260
5 10 390
*/


-- 10. Identify students who have made more than one payment. Use subqueries and aggregate functions
to count payments per student and filter for those with counts greater than one.
SELECT student_id,COUNT(*) AS payment_count
FROM payments
GROUP BY student_id
```

```
HAVING COUNT(*) > 1;
/*
Output
1 2
2 2
3 2
4 2
5 2
*/


-- 11. Write an SQL query to calculate the total payments made by each student. Join the "Students" table with the "Payments" table and use GROUP BY to calculate the sum of payments for each student.
SELECT s.student_id,s.first_name,s.last_name,SUM(p.amount) AS total_payments
FROM Students s
LEFT JOIN Payments p ON s.student_id = p.student_id
GROUP BY s.student_id, s.first_name, s.last_name;
/*
Output
1 harry potter 320
2 Hermione Granger 320
3 Ron Weasley 390
4 Luna Lovegood 260
5 Neville Longbottom 390
6 Ginny Weasley
7 Draco Malfoy
8 Fred Weasley
9 George Weasley
10 Cho Chang
*/



-- 12. Retrieve a list of course names along with the count of students enrolled in each course. Use JOIN operations between the "Courses" table and the "Enrollments" table and GROUP BY to count enrollments.
SELECT c.course_name,COUNT(e.student_id) AS enrollment_count
FROM  Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
GROUP BY c.course_name;
/*
Output
Alchemy 1
Care of Magical Creatures 1
Charms 1
Defense Against the Dark Arts 1
Divination 1
Flying 1
Herbology 1
History of Magic 1
Potions 1
Transfiguration 1
*/
```

```
/*
13. Calculate the average payment amount made by students. Use JOIN operations between the
"Students" table and the "Payments" table and GROUP BY to calculate the average.
*/
SELECT AVG(p.amount) AS average_payment_amount
FROM Payments p
JOIN Students s ON p.student_id = s.student_id;
/*
Output
168.0000
*/
```