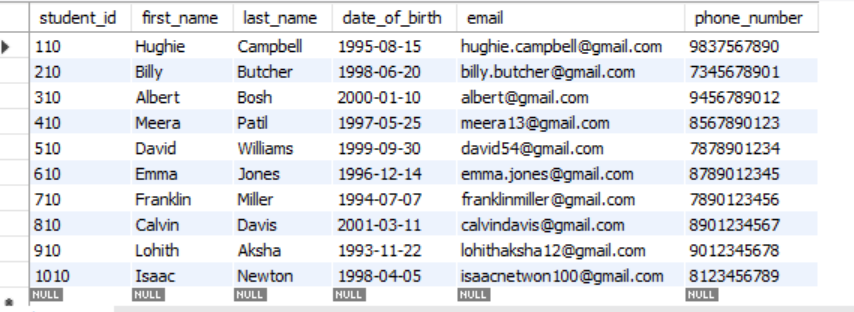
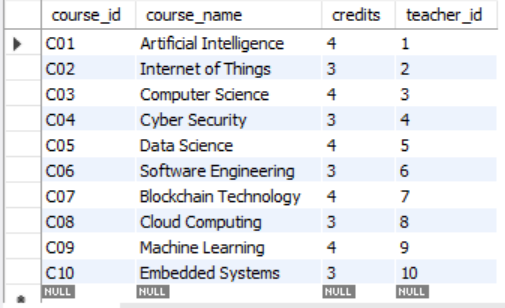
**Assignment – Student Information System (SISDB) Output Screenshots**

Task 1

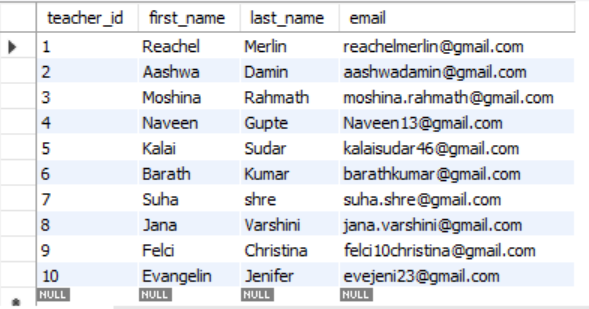
Students table



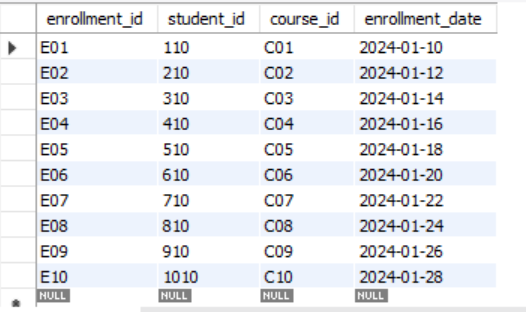
Courses table

****

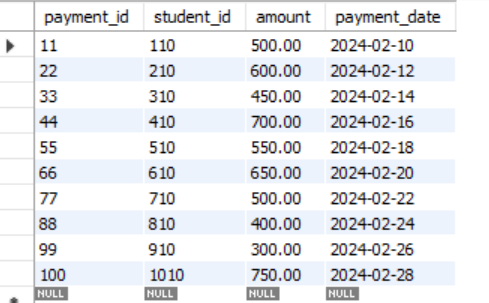
Teacher table



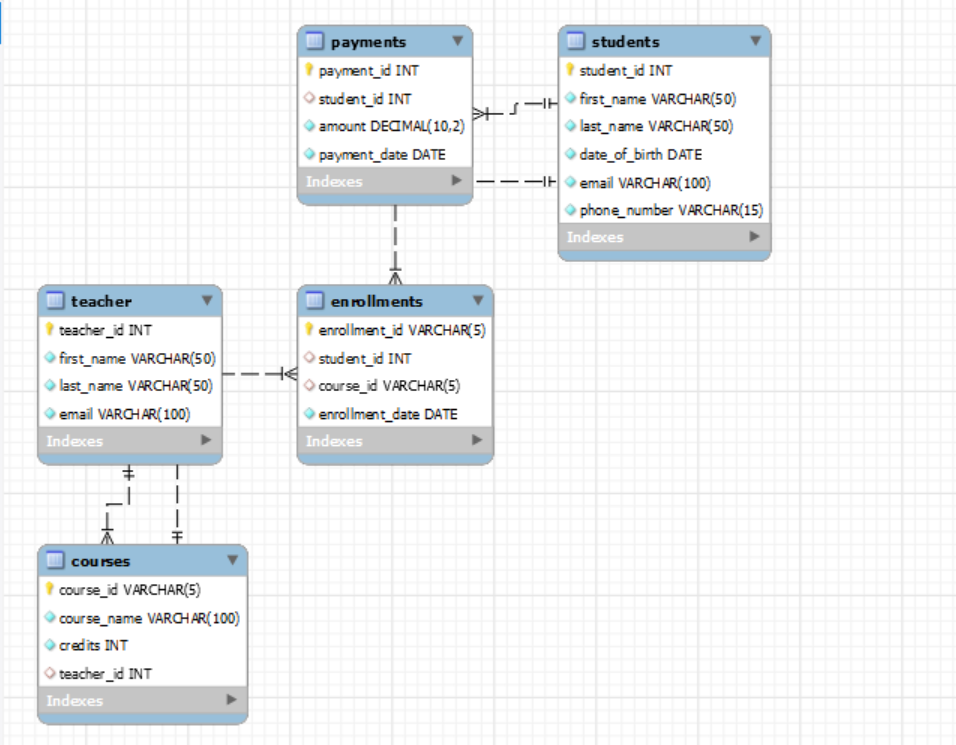
Enrollments table



Payments table

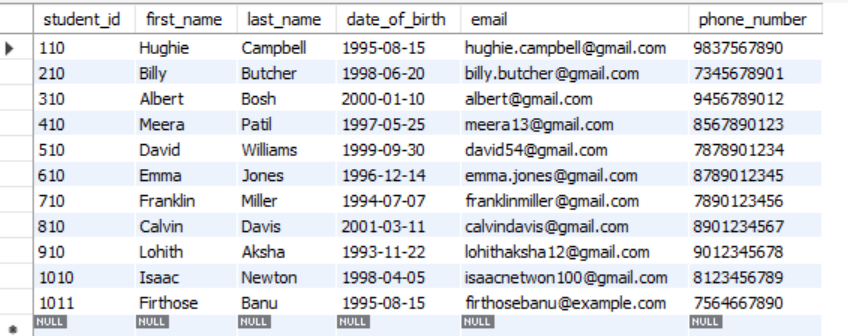


Er diagram

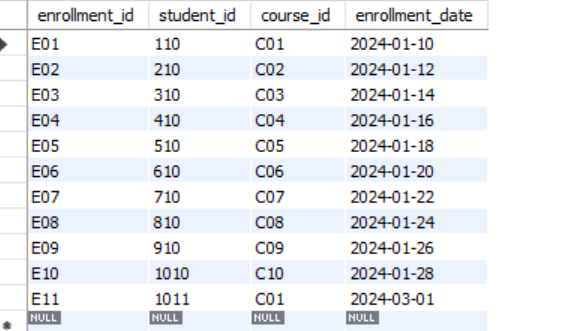


**Task 2 - Select, Where, Between, AND, LIKE:**

1) insert into Students (student\_id,first\_name, last\_name, date\_of\_birth, email, phone\_number) values(1011, 'Firthose', 'Banu', '1995-08-15', 'firthosebanu@example.com', '7564667890');

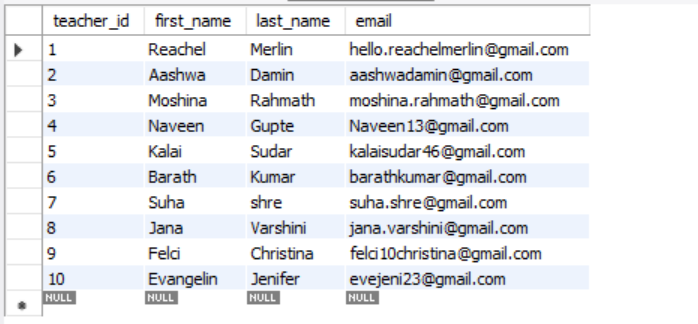


2) insert into Enrollments (enrollment\_id, student\_id, course\_id, enrollment\_date) values ('E11', 1011, 'C01', '2024-03-01');

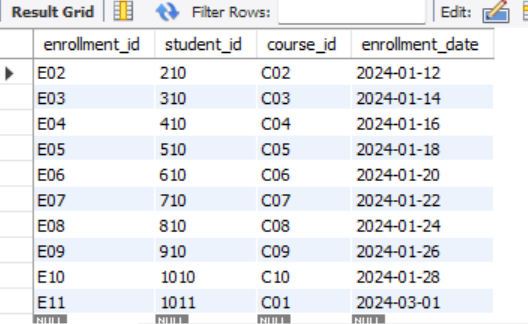


3) update Teacher set email = 'hello.reachelmerlin@gmail.com'

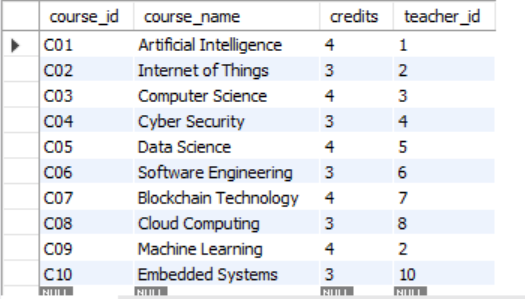
WHERE teacher\_id = 1;



4) delete from Enrollments where student\_id = 110 and course\_id = 'C01';



5) update Courses set teacher\_id = 2where course\_id = 'C09';

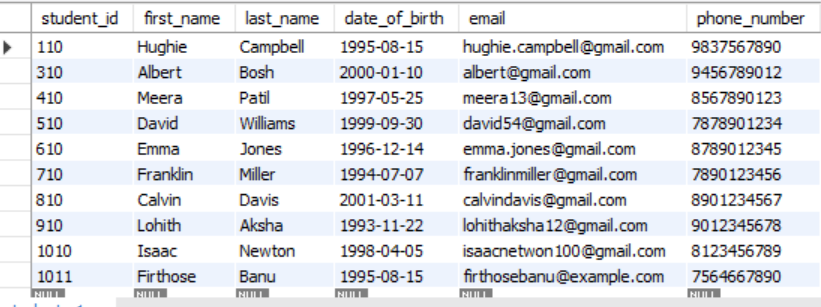


6) Delete a specific student and remove their enrollments (Deleting Student ID 210)

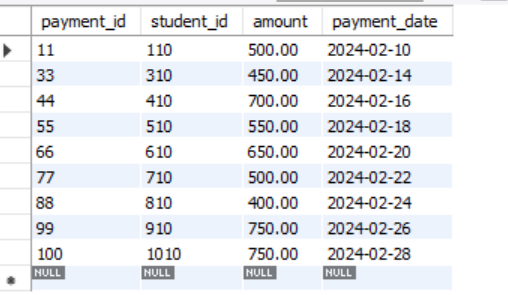
delete from Payments where student\_id = 210;

delete from Enrollments where student\_id = 210;

delete from Students where student\_id = 210;

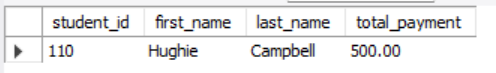


7) update Payments set amount = 750.00 where payment\_id = 22;

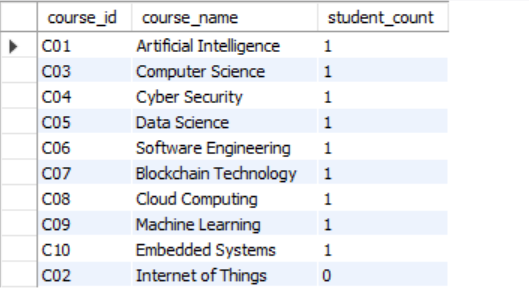


**Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:**

1. select s.student\_id, 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 = 110 group by s.student\_id, s.first\_name, s.last\_name;



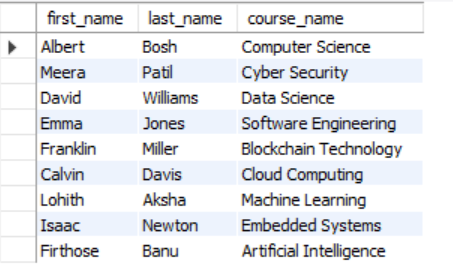
1. select c.course\_id, c.course\_name, count(e.student\_id) as student\_count from courses c left join enrollments e on c.course\_id = e.course\_id group by c.course\_id, c.course\_name order by student\_count desc;



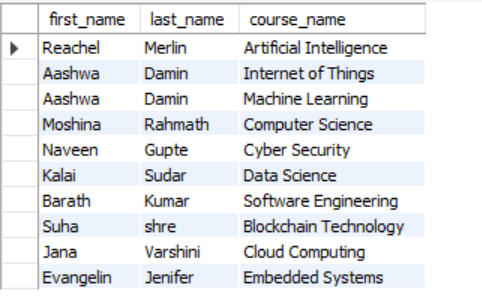
3) select s.student\_id, s.first\_name, s.last\_name from students s left join enrollments e on s.student\_id = e.student\_id where e.student\_id is null;



4) 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 e.course\_id = c.course\_id;



5) SELECT t.first\_name, t.last\_name, c.course\_name FROM Teacher tJOIN Courses c ON t.teacher\_id = c.teacher\_id;



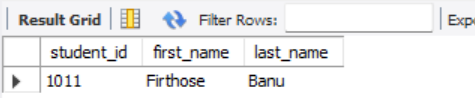
6) select s.first\_name, s.last\_name, e.enrollment\_date from students s

join enrollments e on s.student\_id = e.student\_id join courses c on e.course\_id = c.course\_idwhere c.course\_name = 'artificial intelligence'

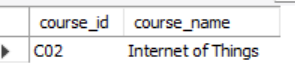
order by e.enrollment\_date;



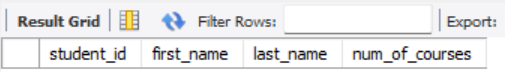
7) select s.student\_id, s.first\_name, s.last\_name from students s left join payments p on s.student\_id = p.student\_id where p.student\_id is null;



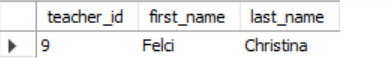
8) select c.course\_id, c.course\_name from courses c left join enrollments e on c.course\_id = e.course\_idwhere e.course\_id is null;



9) select e.student\_id, s.first\_name, s.last\_name, count(e.course\_id) as num\_of\_courses from enrollments e join students s on e.student\_id = s.student\_id group by e.student\_id, s.first\_name, s.last\_name having num\_of\_courses > 1;

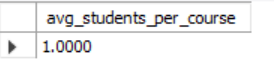


10) select t.teacher\_id, t.first\_name, t.last\_name from teacher t left join courses c on t.teacher\_id = c.teacher\_id where c.teacher\_id is null;

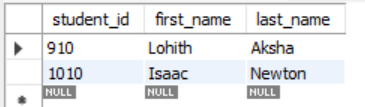


**Task 4. Subquery and its type:**

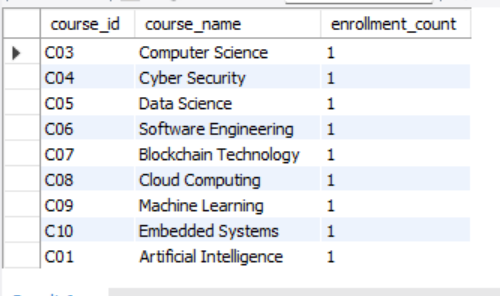
1) select avg(student\_count) as avg\_students\_per\_course from ( select course\_id, count(student\_id) as student\_count from enrollments group by course\_id) as course\_enrollments;



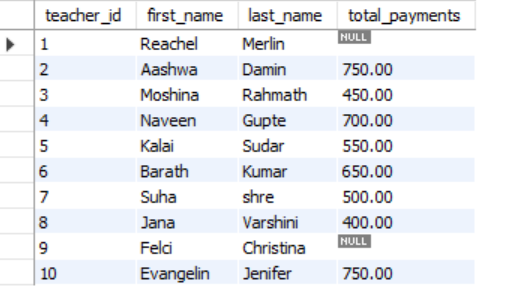
2) select student\_id, first\_name, last\_name from students where student\_id in (select student\_id from payments where amount = (select max(amount) from payments);



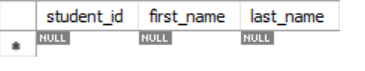
3) select avg(student\_count) as average\_enrollment from (select course\_id, count(student\_id) as student\_count from enrollments group by course\_id) as course\_enrollments;



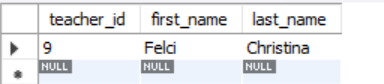
4) select t.teacher\_id, t.first\_name, t.last\_name, (select sum(p.amount) from payments p join enrollments e on p.student\_id = e.student\_id join courses c on e.course\_id = c.course\_idwhere c.teacher\_id = t.teacher\_id) as total\_payments from teacher t;



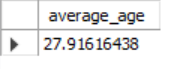
5) select s.student\_id, s.first\_name, s.last\_name from students s where (select count(distinct course\_id) from enrollments where student\_id = s.student\_id) = (select count(\*) from courses);



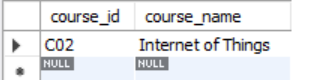
6) select teacher\_id, first\_name, last\_name from teacher where teacher\_id not in (select distinct teacher\_id from courses);



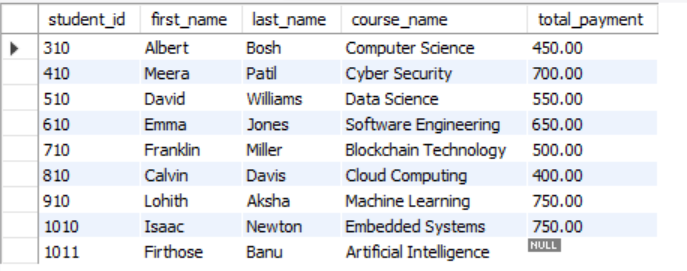
7) select avg(age) as average\_age from (select datediff(curdate(), date\_of\_birth) / 365 as age from students) as age\_subquery;



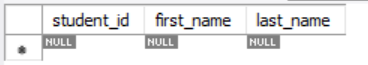
8) select course\_id, course\_name from courses where course\_id not in (select distinct course\_id from enrollments);



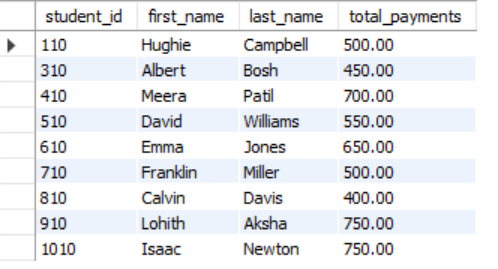
9) select s.student\_id, s.first\_name, s.last\_name, c.course\_name, (select sum(p.amount) from payments p where p.student\_id = s.student\_id) as total\_payment from students s join enrollments e on s.student\_id = e.student\_id join courses c on e.course\_id = c.course\_id;



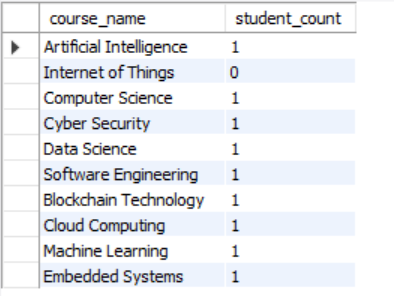
10) select student\_id, first\_name, last\_name from students where student\_id in select student\_idfrom payments group by student\_idhaving count(payment\_id) > 1);



11) select s.student\_id, s.first\_name, s.last\_name, sum(p.amount) as total\_payments from students s join payments p on s.student\_id = p.student\_id group by s.student\_id;



12) select c.course\_name, count(e.student\_id) as student\_count from courses c left join enrollments e on c.course\_id = e.course\_id group by c.course\_name;



13) select s.student\_id, s.first\_name, s.last\_name, avg(p.amount) as avg\_payment from students join payments p on s.student\_id = p.student\_id group by s.student\_id;

