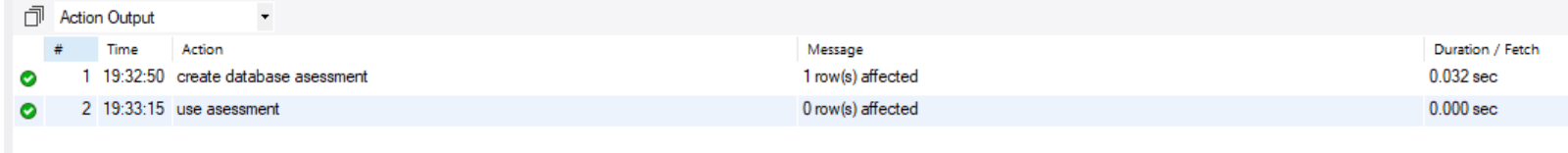
Q. Created Database and selected database

create database asessment;

use asessment;



1. Create the tables below in the database. Use foreign keys and primary keys as required.
   1. Create a table called as student with the following columns student\_id, first\_name, last\_name ,birthdate , department\_id ,address\_id .

create table student(

student\_id int,

first\_name varchar(30),

last\_name varchar(30),

birthdate date,

department\_id int,

address\_id int

);



* 1. Create Address table with following columns address\_id , street\_address, city, State,

postal\_code

create table address(

address\_id int,

street\_address varchar(100),

city varchar(30),

state varchar(30),

postal\_code int

);



* 1. Create department table department\_id, department name. Make sure you are using the right data type against all the columns.

create table department(

department\_id int,

department\_name varchar(50)

);



1. Use Sample data from sampledata.txt to insert data into the database

insert into student values(1,'john','doe','1995-04-15',1,1),

(2,'jane','smith','1967-07-22',2,2),

(3,'alice','johnson','1994-11-30',3,3),

(4,'michael','brown','1997-02-19',4,4),

(5,'sophia','davis','1998-01-05',5,5),

(6,'daniel','wilson','1995-06-10',6,6),

(7,'olivia','martinez','1995-11-25',1,7),

(8,'ethan','miler','1995-03-30',2,8);

insert into address values

(1,'123 elm st','springfield','IL',62701),

(2,'456 oak st','decatur','IL',62521),

(3,'789 pine st','champaign','IL',61820),

(4,'102 birch rd','peoria','IL',61602),

(5,'205 cedar ave','chicago','IL',60601),

(6,'310 maple dr','urbana','IL',61801),

(7,'415 oak blvd','champaign','IL',61821),

(8,'520 pine rd','carbondale','IL',62901);

insert into department values

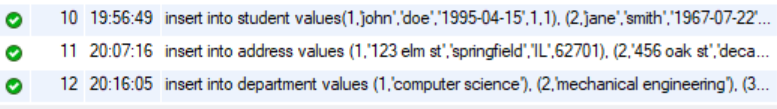
(1,'computer science'),

(2,'mechanical engineering'),

(3,'electrical engineering'),

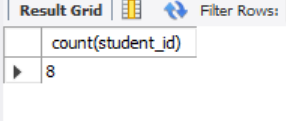
(4,'civil engineering'),

(5,'matchematics'),(6,'biology');



1. Write a query to find the total number of students.

select count(student\_id) from student;



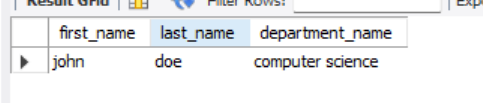
1. Write a query to find which department john belongs to.

select s.first\_name, s.last\_name, d.department\_name

from student s

join department d on s.department\_id = d.department\_id

where s.first\_name = 'john';



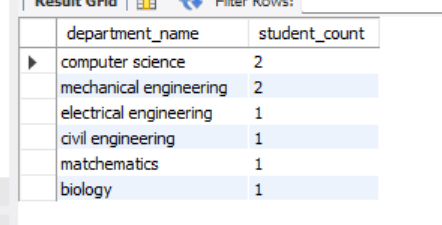
1. List All Departments with Their Number of Students (Including Departments with No Students)

select d.department\_name, count(s.student\_id) as student\_count

from department d

left join student s on d.department\_id = s.department\_id

group by d.department\_name;



1. Select all students with their department and address.

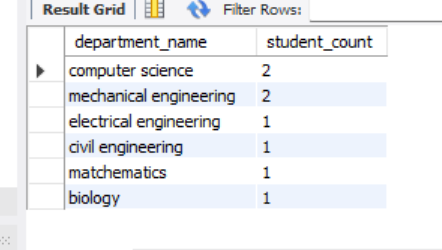
select

s.student\_id, s.first\_name, s.last\_name, d.department\_name, a.street\_address, a.city, a.state, a.postal\_code

from student s

join department d on s.department\_id = d.department\_id

join address a on s.address\_id = a.address\_id;



1. Find all students who are in the 'Computer Science' department

select

s.student\_id,

s.first\_name,

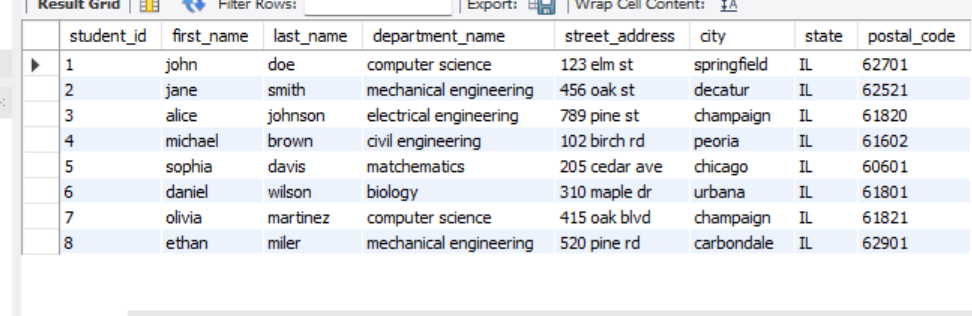
s.last\_name,

d.department\_name

from student s

join department d on s.department\_id = d.department\_id

where d.department\_name = 'computer science';



1. Update Jane’s city name to New York.

update address

set city = 'New York'

where address\_id = (

select s.address\_id

from student s

where s.first\_name = 'jane'

);

1. Delete a student from the student table.

delete from student

where student\_id = 8;

1. Select all students with their department and address in New York.

select

s.student\_id,

s.first\_name,

s.last\_name,

d.department\_name,

a.street\_address,

a.city,

a.state,

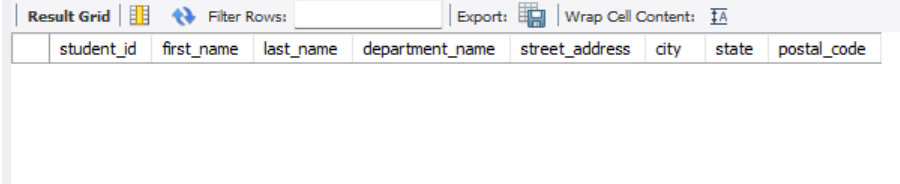
a.postal\_code

from student s

join department d on s.department\_id = d.department\_id

join address a on s.address\_id = a.address\_id

where a.city = 'New York';



1. Count how many students are in each department

select

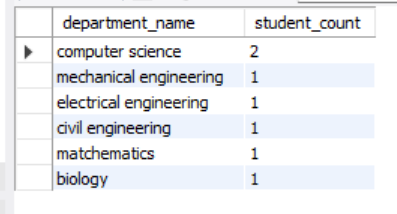
d.department\_name,

count(s.student\_id) as student\_count

from department d

left join student s on d.department\_id = s.department\_id

group by d.department\_name;



1. Find students who live in 'Springfield'

select

s.student\_id,

s.first\_name,

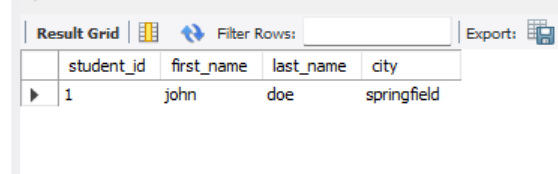
s.last\_name,

a.city

from student s

join address a on s.address\_id = a.address\_id

where a.city = 'springfield';



1. Select students whose birthday falls in February

select

student\_id,

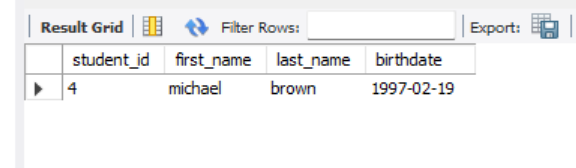
first\_name,

last\_name,

birthdate

from student

where month(birthdate) = 2;



1. Get the department and address details for a specific student, example john

select

s.first\_name,

s.last\_name,

d.department\_name,

a.street\_address,

a.city,

a.state,

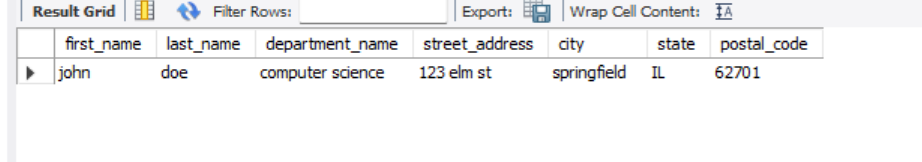
a.postal\_code

from student s

join department d on s.department\_id = d.department\_id

join address a on s.address\_id = a.address\_id

where s.first\_name = 'john';



1. Find all students who are born within 1995 to 1998

select

student\_id,

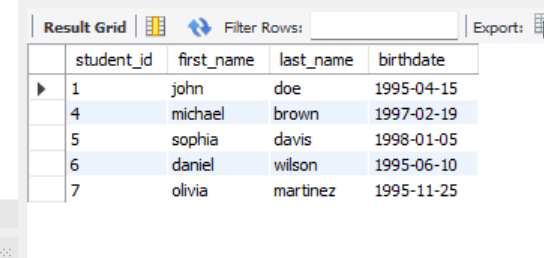
first\_name,

last\_name,

birthdate

from student

where year(birthdate) BETWEEN 1995 AND 1998;



1. List all students and their corresponding department names, sorted by department

select

s.student\_id,

s.first\_name,

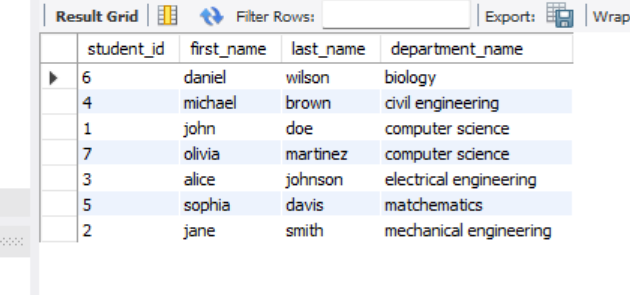
s.last\_name,

d.department\_name

from student s

join department d on s.department\_id = d.department\_id

order by d.department\_name;



1. Find the number of students in each department who are living in 'Champaign'

select

d.department\_name,

count(s.student\_id) as student\_count

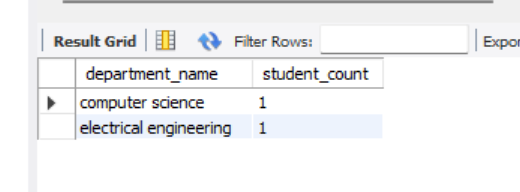
from student s

join department d on s.department\_id = d.department\_id

join address a on s.address\_id = a.address\_id

where a.city = 'champaign'

group by d.department\_name;



1. Retrieve the names of students who live on 'Pine' street

select

s.first\_name,

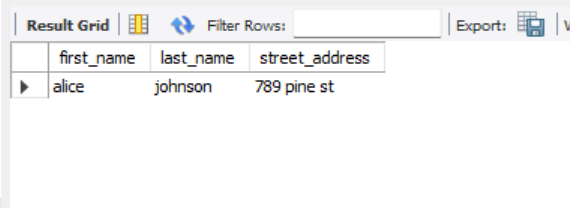
s.last\_name,

a.street\_address

from student s

join address a on s.address\_id = a.address\_id

where a.street\_address like '%Pine%';



1. Update the department of a student with student\_id = 6 to 'Mechanical Engineering'

update student

set department\_id = (

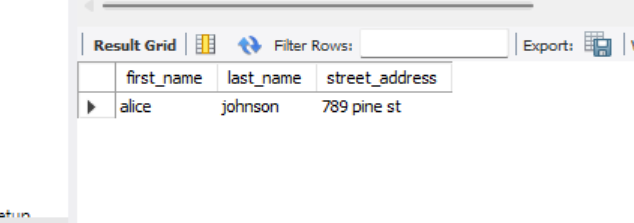
select department\_id

from department

where department\_name = 'mechanical engineering'

)

where student\_id = 6;



1. Find the student(s) who live in the city 'Chicago' and are in the 'Mathematics' department

select

s.student\_id,

s.first\_name,

s.last\_name,

d.department\_name,

a.city

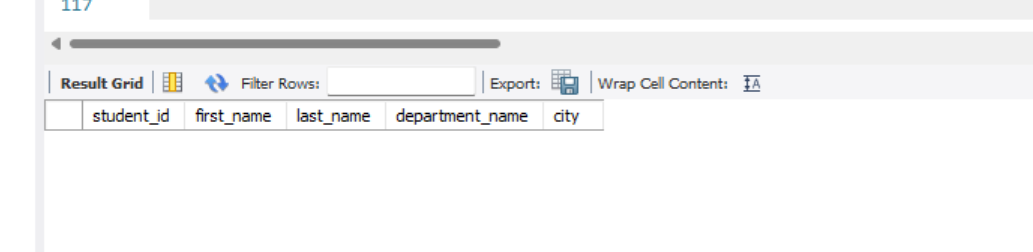
from student s

join department d on s.department\_id = d.department\_id

join address a on s.address\_id = a.address\_id

where a.city = 'chicago'

and d.department\_name = 'mathematics';



1. List all students who have an address in 'Urbana' or 'Peoria'

select

s.student\_id,

s.first\_name,

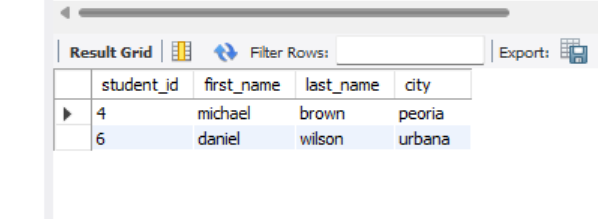
s.last\_name,

a.city

from student s

join address a on s.address\_id = a.address\_id

where a.city in ('urbana', 'peoria');



1. Find the student with the highest student\_id

select

student\_id,

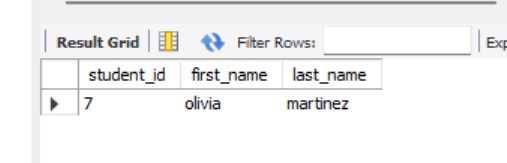
first\_name,

last\_name

from student

order by student\_id desc

limit 1;



1. Find all students who are not in the 'Computer Science' department

select

s.student\_id,

s.first\_name,

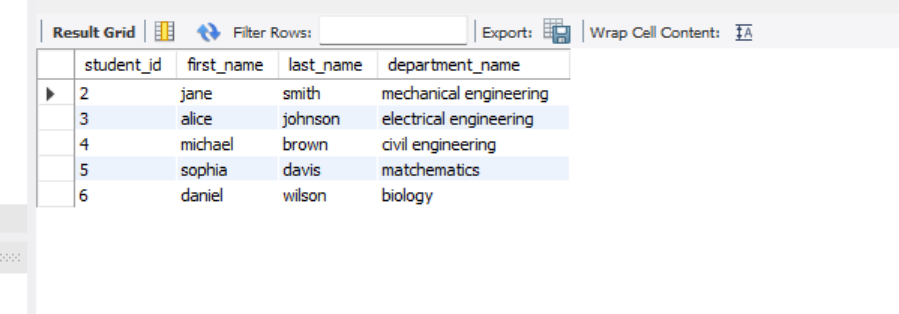
s.last\_name,

d.department\_name

from student s

join department d on s.department\_id = d.department\_id

where d.department\_name <> 'computer science';



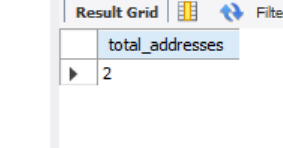
1. Count the total number of addresses in the 'Champaign' city

select

count(\*) as total\_addresses

from address

where city = 'champaign';



1. Find the name of the student who lives at '520 Pine Rd'

select

s.first\_name,

s.last\_name,

a.street\_address

from student s

join address a on s.address\_id = a.address\_id

where a.street\_address = '520 Pine Rd';

1. Get the average age of students in the 'Electrical Engineering' department

select

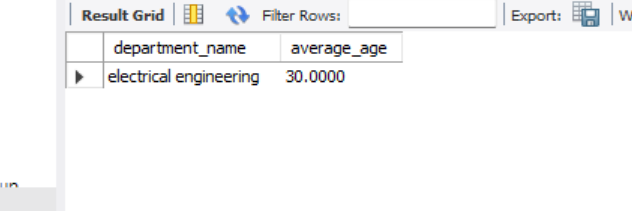
d.department\_name,

avg(timestampdiff(year, s.birthdate, curdate())) as average\_age

from student s

join department d on s.department\_id = d.department\_id

where d.department\_name = 'electrical engineering';



1. List the students, their department, and the city where they live, but only for those in departments starting with 'M'

select

s.student\_id,

s.first\_name,

s.last\_name,

d.department\_name,

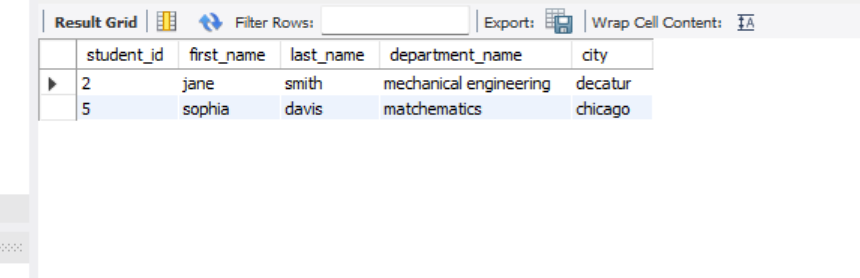
a.city

from student s

join department d on s.department\_id = d.department\_id

join address a on s.address\_id = a.address\_id

where d.department\_name like 'm%';



1. Delete a student from the 'Mechanical Engineering' department

delete from student

where department\_id = (

select department\_id

from department

where department\_name = 'mechanical engineering'

)

limit 1;



1. Retrieve All Orders with Their Customer Details and Current Status

SELECT

o.order\_id,

o.order\_date,

o.total\_amount,

CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,

c.email,

c.phone\_number,

s.status\_name AS current\_status

FROM

order\_schema.orders o

JOIN

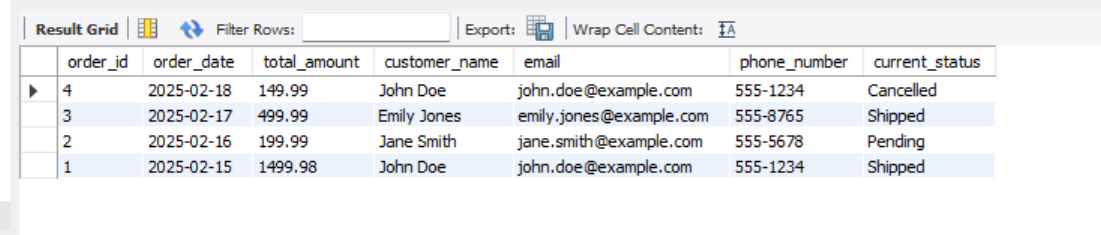
order\_schema.customer c ON o.customer\_id = c.customer\_id

JOIN

order\_schema.status s ON o.status\_id = s.status\_id

ORDER BY

o.order\_date DESC;



1. Get the Total Value of Orders for a Given Customer in a Specific Time Period

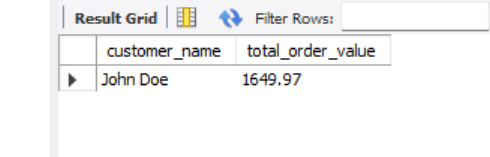
SELECT CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,SUM(o.total\_amount) AS total\_order\_value

FROM order\_schema.orders o

JOIN order\_schema.customer c ON o.customer\_id = c.customer\_id

WHERE o.customer\_id = 1 AND o.order\_date BETWEEN '2025-02-15' AND '2025-02-18'

GROUP BY customer\_name;



1. Find the Most Expensive Order by Customer

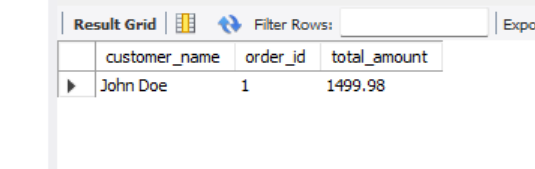
SELECT CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,o.order\_id,o.total\_amount

FROM order\_schema.orders o

JOIN order\_schema.customer c ON o.customer\_id = c.customer\_id

ORDER BY o.total\_amount DESC

LIMIT 1;



1. Find the Total Revenue for Each Product Based on Orders

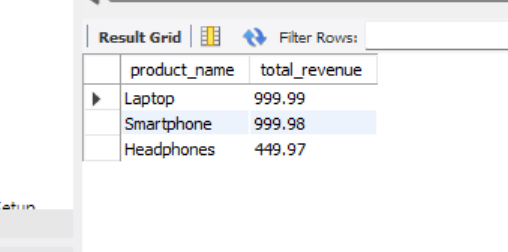
SELECT p.product\_name,SUM(oi.quantity \* oi.price) AS total\_revenue

FROM order\_schema.order\_items oi

JOIN order\_schema.product p ON oi.product\_id = p.product\_id

GROUP BY p.product\_name

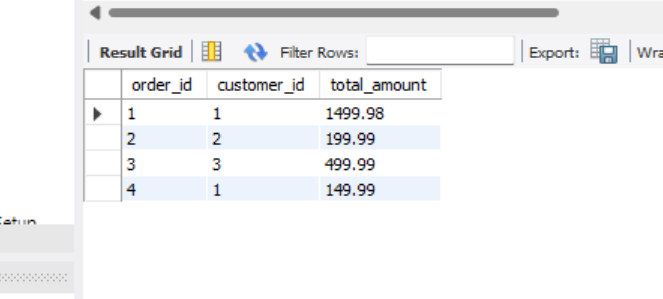
ORDER BY total\_revenue DESC;



1. Write a query to retrieve the order ID, customer ID, and the total amount of each order. If the total amount is null, display '0.00' instead.

SELECT o.order\_id,o.customer\_id,COALESCE(o.total\_amount, 0.00) AS total\_amount

FROM order\_schema.orders o;



1. Retrieve the Order History of a Specific Customer Along with Product Details

SELECT o.order\_id,o.order\_date,p.product\_name,oi.quantity,oi.price,oh.status\_change\_date,oh.status\_description

FROM order\_schema.orders o

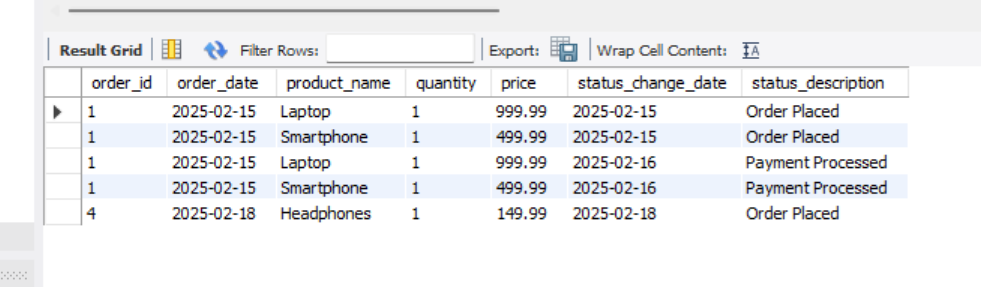
JOIN order\_schema.order\_items oi ON o.order\_id = oi.order\_id

JOIN order\_schema.product p ON oi.product\_id = p.product\_id

JOIN order\_schema.order\_history oh ON o.order\_id = oh.order\_id

WHERE o.customer\_id = 1

ORDER BY o.order\_id, oh.status\_change\_date;



1. Get the Average Order Value Per Customer in the Last 30 Days.

select o.customer\_id, avg(o.total\_amount) as average\_order\_value

from order\_schema.orders o

where o.order\_date >= date\_sub(current\_date, interval 30 day)

group by o.customer\_id;

1. Get the Top 5 Products with the Highest Number of Orders.

select p.product\_name,count(distinct oi.order\_id) as number\_of\_orders

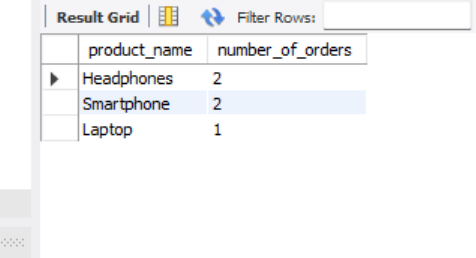
from order\_schema.order\_items oi

join order\_schema.product p on oi.product\_id = p.product\_id

group by p.product\_name

order by number\_of\_orders desc

limit 5;



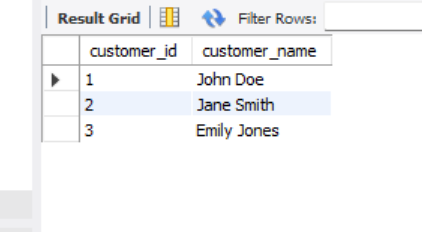
1. Get the Customers Who Have Not Placed Any Orders in the Last 60 Days

select c.customer\_id, concat(c.first\_name, ' ', c.last\_name) as customer\_name

from order\_schema.customer c

left join order\_schema.orders o on c.customer\_id = o.customer\_id and o.order\_date >= current\_date - 60

where o.order\_id is null;



1. List the Orders with Products Ordered More Than Once, Sorted by Order Date

select o.order\_id, o.order\_date, p.product\_name, oi.quantity

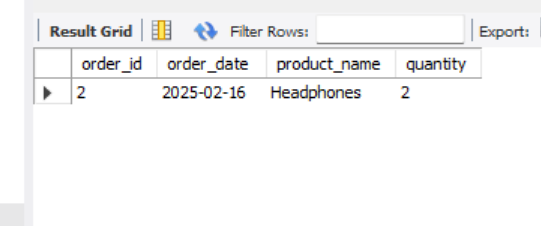
from order\_schema.orders o

join order\_schema.order\_items oi on o.order\_id = oi.order\_id

join order\_schema.product p on oi.product\_id = p.product\_id

where oi.quantity > 1

order by o.order\_date;



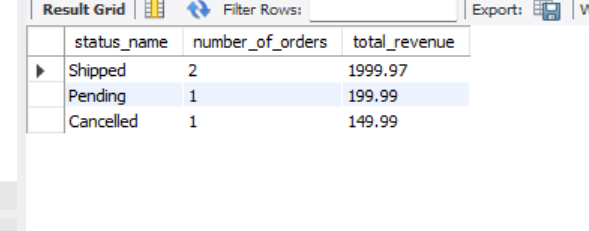
1. Retrieve the Number of Orders and Total Revenue for Each Status

select s.status\_name, count(o.order\_id) as number\_of\_orders, coalesce(sum(o.total\_amount), 0) as total\_revenue

from order\_schema.status s

left join order\_schema.orders o on s.status\_id = o.status\_id

group by s.status\_name;



1. Find Customers Who Have Ordered More Than a Specific Product (e.g., "Laptop")

select distinct c.customer\_id, concat(c.first\_name, ' ', c.last\_name) as customer\_name

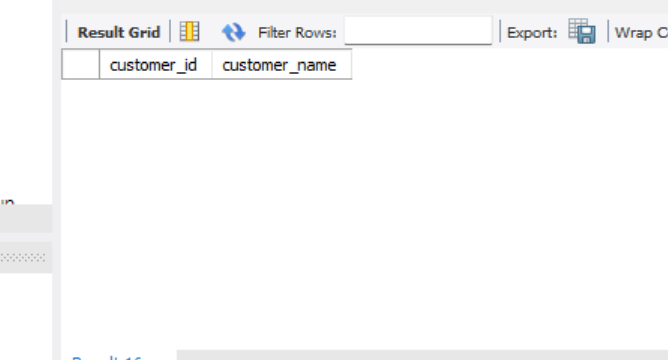
from order\_schema.customer c

join order\_schema.orders o on c.customer\_id = o.customer\_id

join order\_schema.order\_items oi on o.order\_id = oi.order\_id

join order\_schema.product p on oi.product\_id = p.product\_id

where p.product\_name = 'Laptop' and oi.quantity > 1;



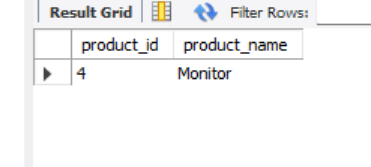
1. Find the Products That Have Never Been Ordered

select p.product\_id, p.product\_name

from order\_schema.product p

left join order\_schema.order\_items oi on p.product\_id = oi.product\_id

where oi.order\_item\_id is null;



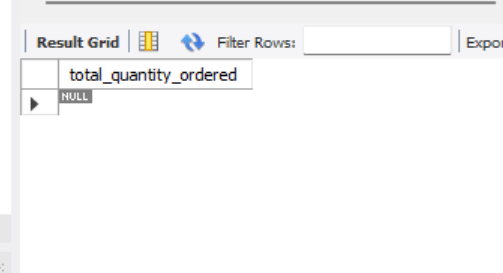
1. Get the Total Quantity of Products Ordered in the Last 7 Days

select sum(oi.quantity) as total\_quantity\_ordered

from order\_schema.order\_items oi

join order\_schema.orders o on oi.order\_id = o.order\_id

where o.order\_date >= current\_date - 7;



1. Create a view named product\_details that includes all columns from the product table.

create view order\_schema.product\_details as

select \* from order\_schema.product;

1. Create a view named order\_summary that includes the order\_id, customer\_id, order\_date, total\_amount, and status\_name (from the status table) for each order.

create view order\_schema.order\_summary as

select o.order\_id, o.customer\_id, o.order\_date, o.total\_amount, s.status\_name

from order\_schema.orders o

join order\_schema.status s on o.status\_id = s.status\_id;