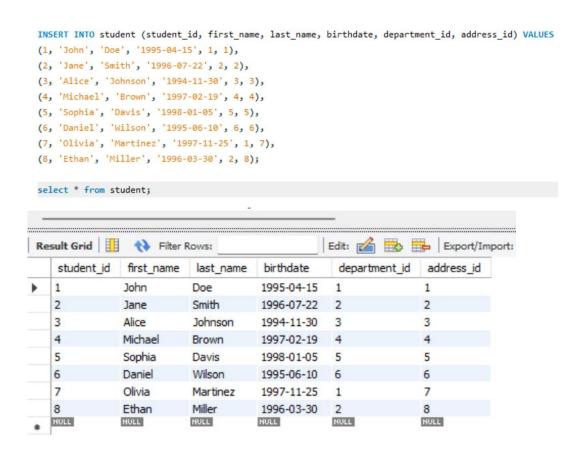
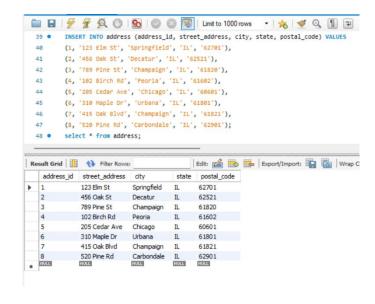
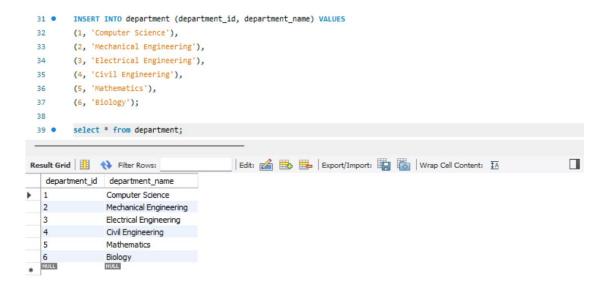
- 1. Create the tables below in the database. Use foreign keys and primary keys as required.
 - a. Create a table called as student with the following columns student_id, first_name, last_name ,birthdate , department_id ,address_id .



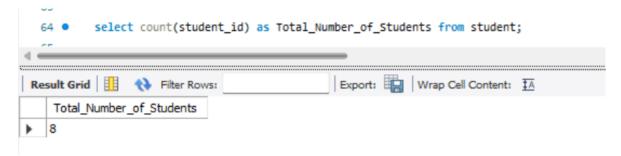
b. Create Address table with following columns address_id , street_address, city, State, postal_code



c. Create department table department_id, department name. Make sure you are using the right data type against all the columns.



- 2. Use Sample data from <u>sampledata.txt</u> to insert data into the database
- 3. Write a query to find the total number of students.



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

```
66 select d.department_name
67 from department d Left join student s
68 on s.department_id=d.department_id
69 where first_name="John";
70
71

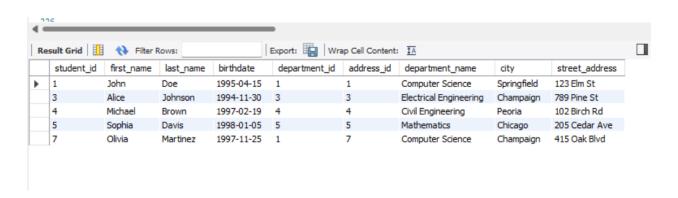
Result Grid Filter Rows: Expo

department_name
Computer Science
```

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

```
SELECT d.department_name, COUNT(s.student_id) AS student_count
    71 •
    72
                                                          FROM department d
                                                          LEFT JOIN student s ON d.department_id = s.department_id
    73
                                                          GROUP BY d.department_name;
    74
    75
    76
    77
                                                                                                                                                                                                                                                                                                          Export: Wrap Cell Content: IA
tesult Grid | | Name | 
                 department_name
                                                                                                                                                           student_count
             Computer Science
                                                                                                                                                         2
             Mechanical Engineering 2
             Electrical Engineering
             Civil Engineering
                                                                                                                                                       1
             Mathematics
                                                                                                                                                        1
            Biology
                                                                                                                                                       1
```

6. Select all students with their department and address.



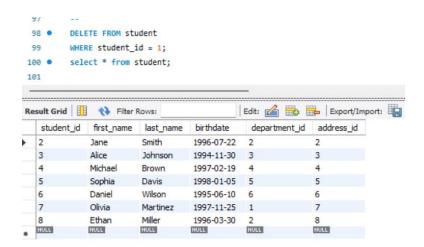
7. Find all students who are in the 'Computer Science' department

```
select s.*
 83 •
          from student s Right join department d on s.department_id=d.department_id
 84
          where d.department_name="Computer Science";
 85
 86
 87
 88
                                                  Export: Wrap Cell Content: $\frac{1}{4}
Result Grid
                 Filter Rows:
   student id
               first name
                            last name
                                        birthdate
                                                     department_id
                                                                     address_id
               John
                                                    1
                                                                     1
  1
                           Doe
                                       1995-04-15
  7
                                                                    7
               Olivia
                                                    1
                           Martinez
                                       1997-11-25
```

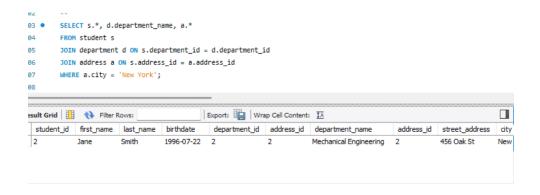
8. Update Jane's city name to New York.

```
UPDATE address
37 •
        SET city = 'New York'
88
39
     HERE address_id = (
            SELECT address_id
90
91
            FROM student
            WHERE first_name = 'Jane' AND last_name = 'Smith'
92
93
        );
94
        select city from address;
95 •
96
97
                                              Export: Wrap Cell C
sult Grid
              Filter Rows:
 city
 Springfield
 New York
 Champaign
 Peoria
 Chicago
 Urbana
 Champaign
 Carbondale
```

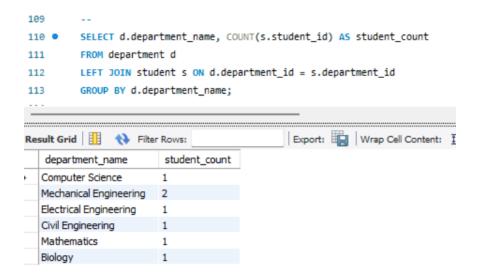
9. Delete a student from the student table.



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



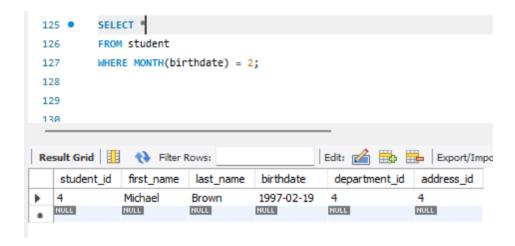
11. Count how many students are in each department



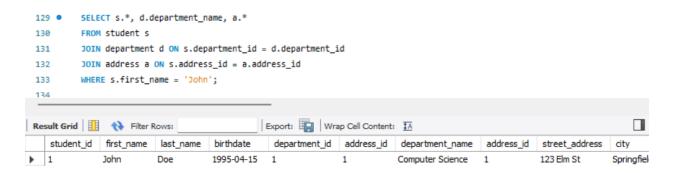
12. Find students who live in 'Springfield'

```
118
         SELECT s.*
119 •
         FROM student s
120
         JOIN address a ON s.address_id = a.address_id
121
         WHERE a.city = 'Springfield';
122
123
124
                                             Export: Wrap Cell Content: $\frac{1}{2}$
student_id
              first_name
                                    birthdate
                                                department_id
                                                              address_id
                         last_name
             John
                        Doe
                                   1995-04-15
 1
```

13. Select students whose birthday falls in February



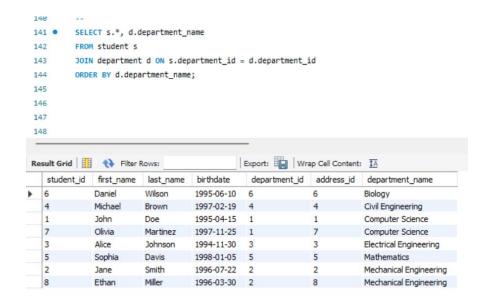
14. Get the department and address details for a specific student, example john



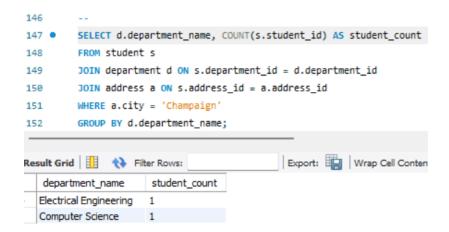
15. Find all students who are born within 1995 to 1998



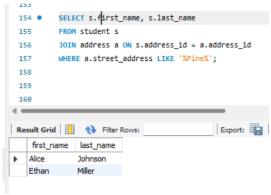
16. List all students and their corresponding department names, sorted by department



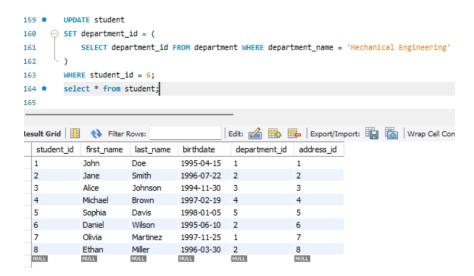
17. Find the number of students in each department who are living in 'Champaign'



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



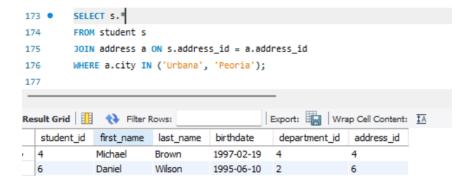
19. Update the department of a student with student_id = 6 to 'Mechanical Engineering'



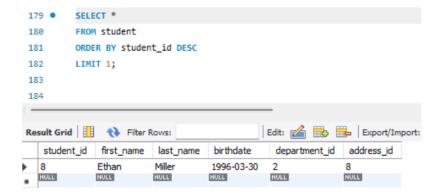
20. Find the student(s) who live in the city 'Chicago' and are in the 'Mathematics' department



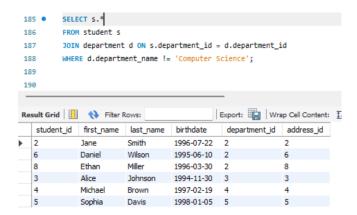
21. List all students who have an address in 'Urbana' or 'Peoria'



22. Find the student with the highest student_id



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



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

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

```
SELECT s.first_name, s.last_name
195 •
          FROM student s
196
          JOIN address a ON s.address_id = a.address_id
197
          WHERE a.street_address = '520 Pine Rd';
198
199
200
                                               Export:
Result Grid
               Filter Rows:
   first_name
               last_name
  Ethan
              Miller
```

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

```
SELECT AVG(TIMESTAMPDIFF(YEAR, birthdate, CURDATE())) AS average_age
FROM student s
JOIN department d ON s.department_id = d.department_id
WHERE d.department_name = 'Electrical Engineering';

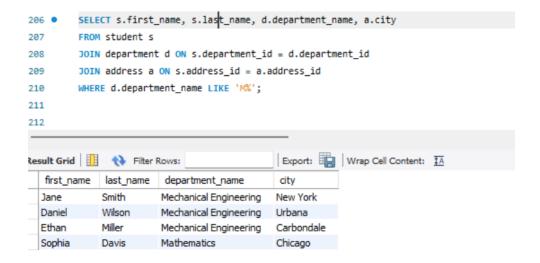
ult Grid

Filter Rows:

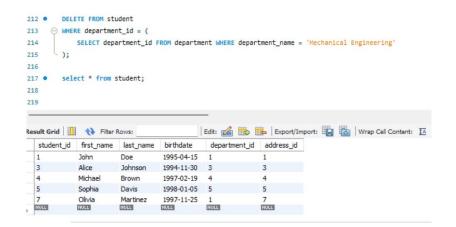
Export: Wrap Cell Content: A

average_age
30.0000
```

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



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



Download order.sql

Open PG Admin and open query tool and select any database of your choice.

Click on "Open file" and select order.sql from your device and execute it.

Questions:

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

```
112
113 •
          SELECT
114
          o.order_id,
115
           c.customer_id,
           c.first_name,
117
           c.last_name,
118
           c.email,
119
          s.status_name,
120
          o.order_date,
121
           o.total_amount
122
     FROM order_schema.orders o
123
     JOIN order_schema.customer c ON o.customer_id = c.customer_id
       JOIN order_schema.status s ON o.status_id = s.status_id;
125
Export: Wrap Cell Content: TA
   order_id customer_id first_name last_name email
                                                         status_name order_date total_amount
                    John
                             Doe
                                      john.doe@example.com
                                                          Shipped
                                                                     2025-02-15
                                                                               1499.98
  1
                                     john.doe@example.com Cancelled 2025-02-18 149.99
                   John
                             Doe
                                      jane.smith@example.com Pending
          2
                    Jane
                             Smith
                                                                     2025-02-16 199.99
                           Jones emily.jones@example.com Shipped 2025-02-17 499.99
  3
                    Emily
```

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

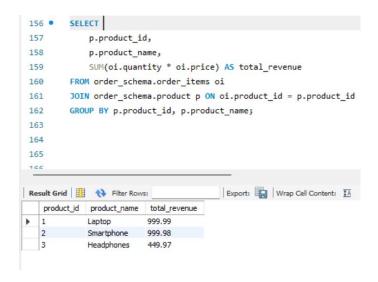
```
126
127 • SELECT
128
         c.customer_id,
129
          c.first_name,
130
          c.last_name,
          SUM(o.total_amount) AS total_order_value
132 FROM order schema.orders o
133      JOIN order_schema.customer c ON o.customer_id = c.customer_id
        WHERE o.order_date BETWEEN '2025-02-15' AND '2025-02-18'
        AND c.customer_id = 1 -- Replace with desired customer ID
135
136
        GROUP BY c.customer_id, c.first_name, c.last_name;
Result Grid ## Tilter Rows:
                                       Export: Wrap Cell Content: TA
 customer_id first_name last_name total_order_value
             John
                      Doe
```

3. Find the Most Expensive Order by Customer

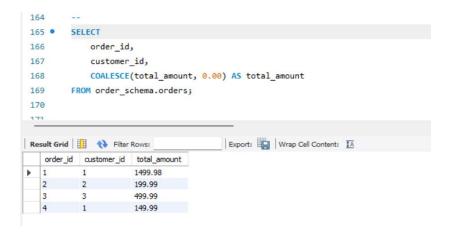
```
227
228
        SELECT customer_id, order_id, total_amount
229
        FROM order_schema.orders
230

→ WHERE (customer_id, total_amount) IN (
231
             SELECT customer_id, MAX(total_amount)
             FROM order schema.orders
232
233
             GROUP BY customer_id
234
       ٠);
235
                                          Edit: 🚄 🖶 🗒
Result Grid
             Filter Rows:
              order_id total_amount
   customer_id
                      1499.98
  1
             1
  2
             2
                      199.99
                      499.99
  3
             3
             NULL
 NULL
                      NULL
```

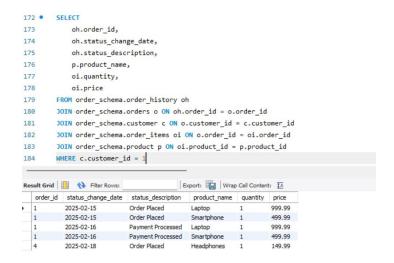
4. Find the Total Revenue for Each Product Based on Orders



5. 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.



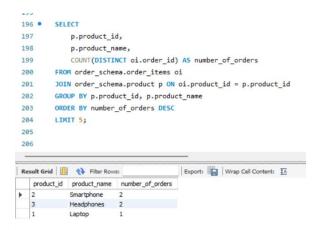
6. Retrieve the Order History of a Specific Customer Along with Product Details



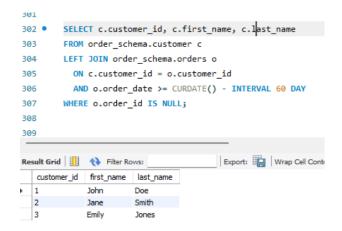
7. Get the Average Order Value Per Customer in the Last 30 Days.

```
SELECT c.customer_id, c.first_name, c.last_name,
295 •
               AVG(o.total_amount) AS avg_order_value
296
297
        FROM order schema.customer c
        JOIN order_schema.orders o ON c.customer_id = o.customer_id
        WHERE o.order_date >= CURDATE() - INTERVAL 30 DAY
299
        GROUP BY c.customer id, c.first name, c.last name;
301
302
Export: Wrap Cell Content: IA
             first_name last_name avg_order_value
  customer_id
```

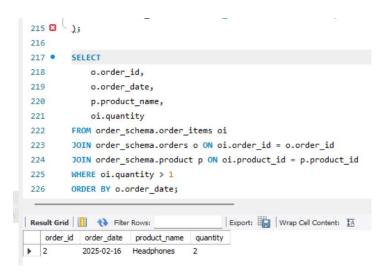
8. Get the Top 5 Products with the Highest Number of Orders.



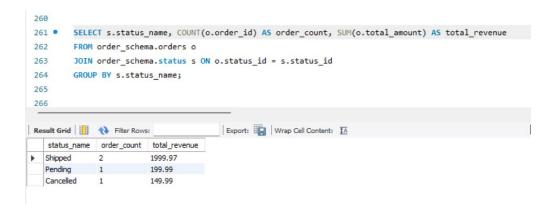
9. Get the Customers Who Have Not Placed Any Orders in the Last 60 Days



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



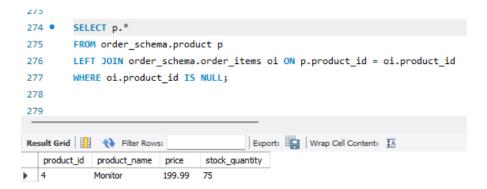
11. Retrieve the Number of Orders and Total Revenue for Each Status



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

```
SELECT DISTINCT c.customer_id, c.first_name, c.last_name
266 •
267
         FROM order_schema.customer c
         JOIN order_schema.orders o ON c.customer_id = o.customer_id
268
         JOIN order_schema.order_items oi ON o.order_id = oi.order_id
269
270
         JOIN order_schema.product p ON oi.product_id = p.product_id
271
         WHERE p.product_name = 'Laptop';
272
273
                                          Export: Wrap Cell Content: $\frac{1}{4}
Result Grid Filter Rows:
   customer_id first_name
                         last_name
              John
                        Doe
```

13. Find the Products That Have Never Been Ordered



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

```
308
        SELECT p.product_name, SUM(oi.quantity) AS total_quantity
309 •
        FROM order schema.order items oi
310
        JOIN order_schema.orders o ON oi.order_id = o.order_id
311
        JOIN order_schema.product p ON oi.product_id = p.product_id
312
        WHERE o.order_date >= CURDATE() - INTERVAL 7 DAY
313
314
        GROUP BY p.product name;
315
316
                                         Export: Wrap Cell Content: TA
Result Grid
             Filter Rows:
   product_name
                total_quantity
```

15. Create a view named product_details that includes all columns from the product table.

```
285

286 • CREATE VIEW order_schema.product_details AS

287 SELECT * FROM order_schema.product;
```

16. 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.

```
288
289 • CREATE VIEW order_schema.order_summary AS
290 SELECT o.order_id, o.customer_id, o.order_date, o.total_amount, s.status_name
291 FROM order_schema.orders o
292 JOIN order_schema.status s ON o.status_id = s.status_id;
293
```