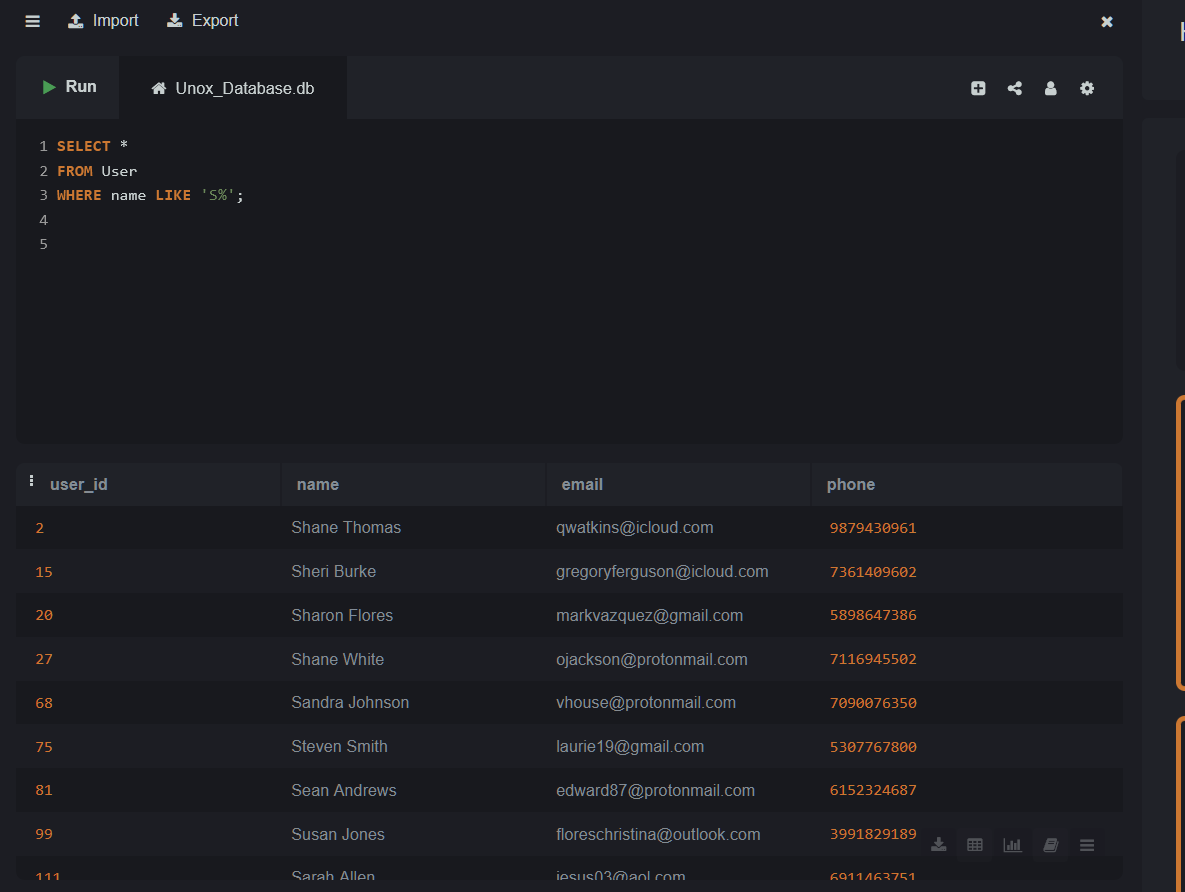
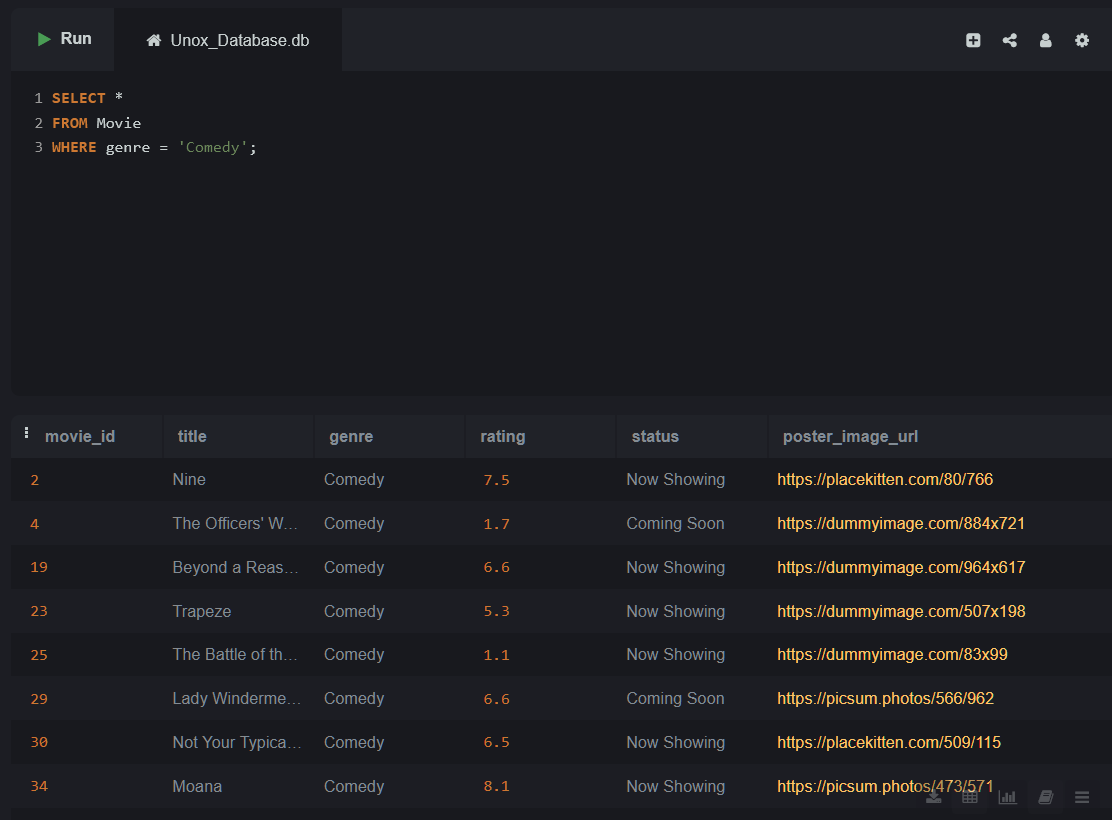
Topic 1

Lab 1.1: – Retrieve the details of all users whose name starts with

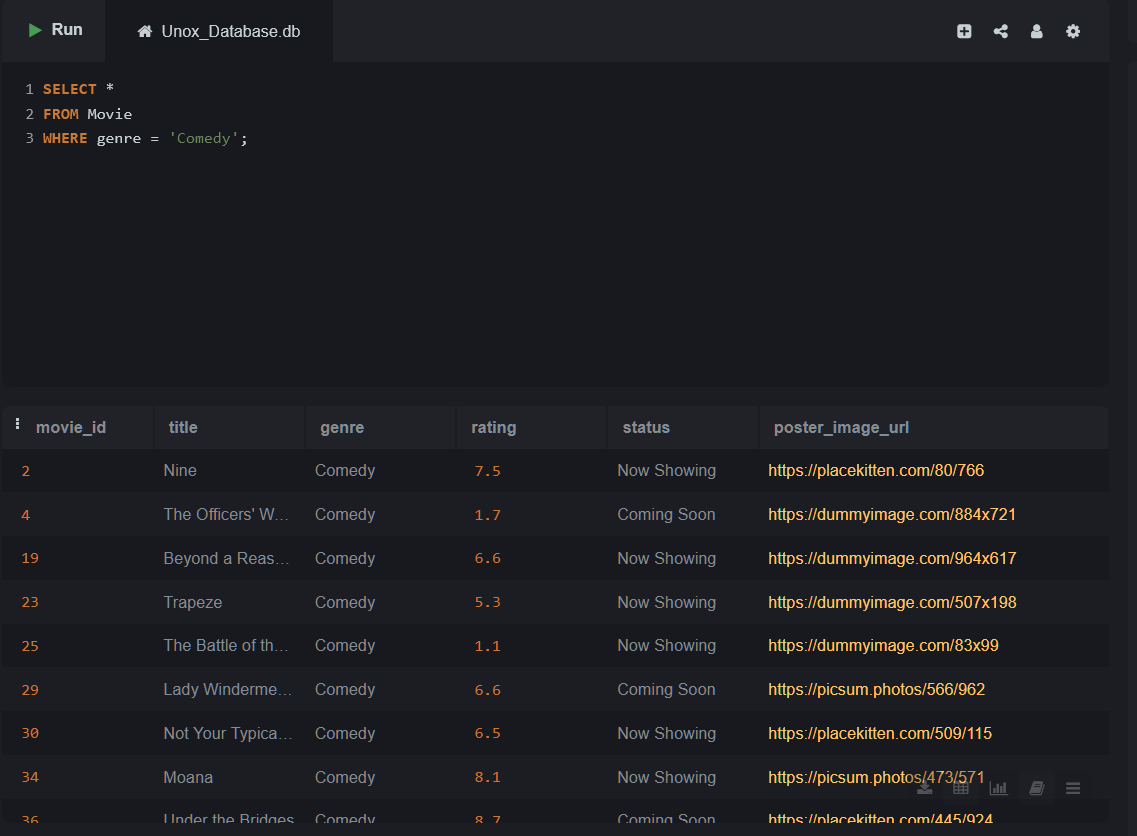
the letter 'S'.



Lab 1.2 – Retrieve all movies where the genre is 'Comedy'.

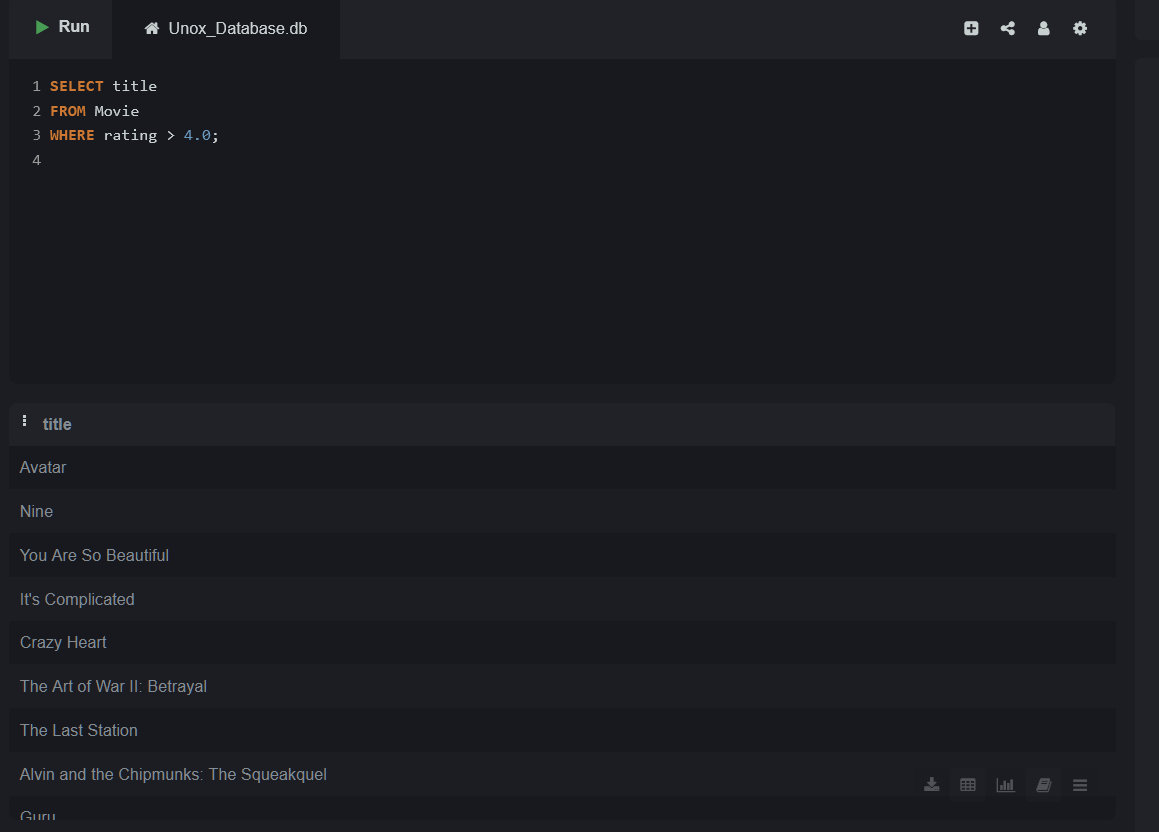


Lab 1.3:give me the sql command for this above ques using all the file



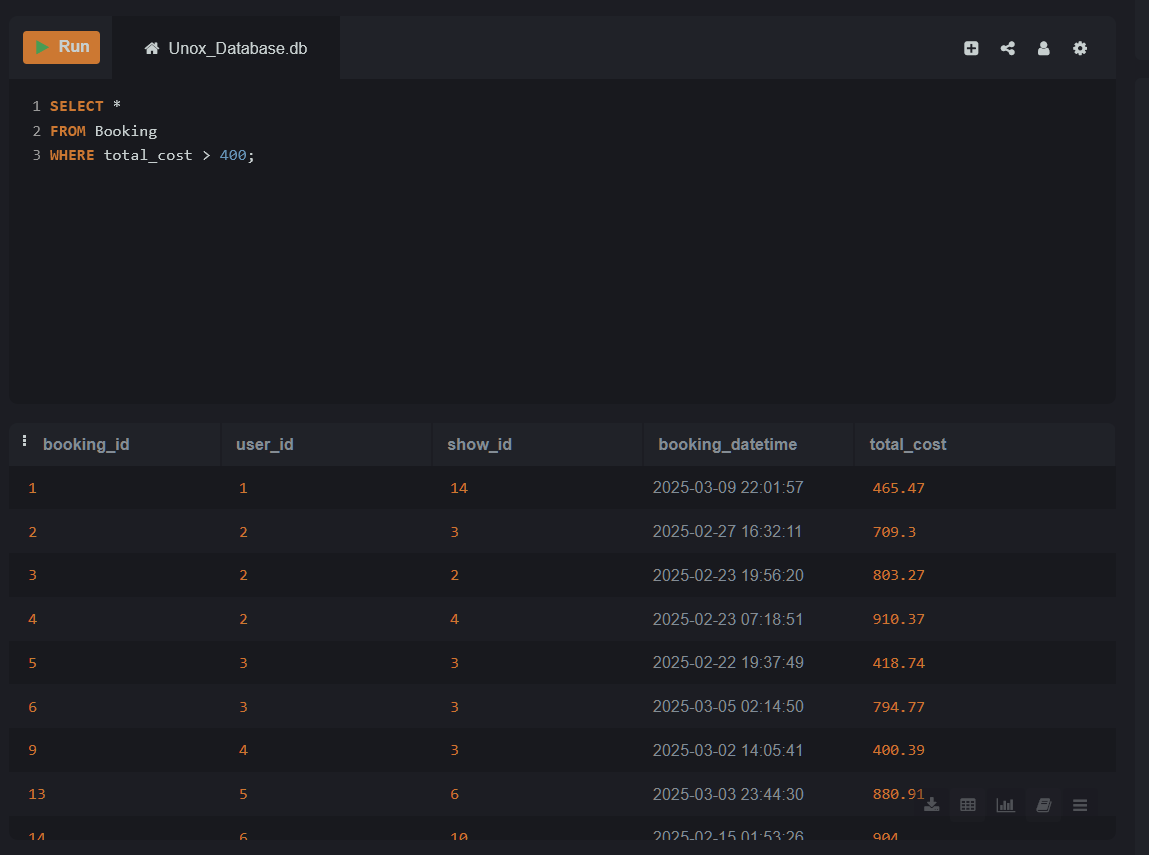
**Topic 2: SELECT with WHERE using Operators (=, >, <, !=)**

Lab 2.1 – List all movie names where the rating is greater than 4.0.



Lab 2.2 – List all bookings where the total amount is greater than

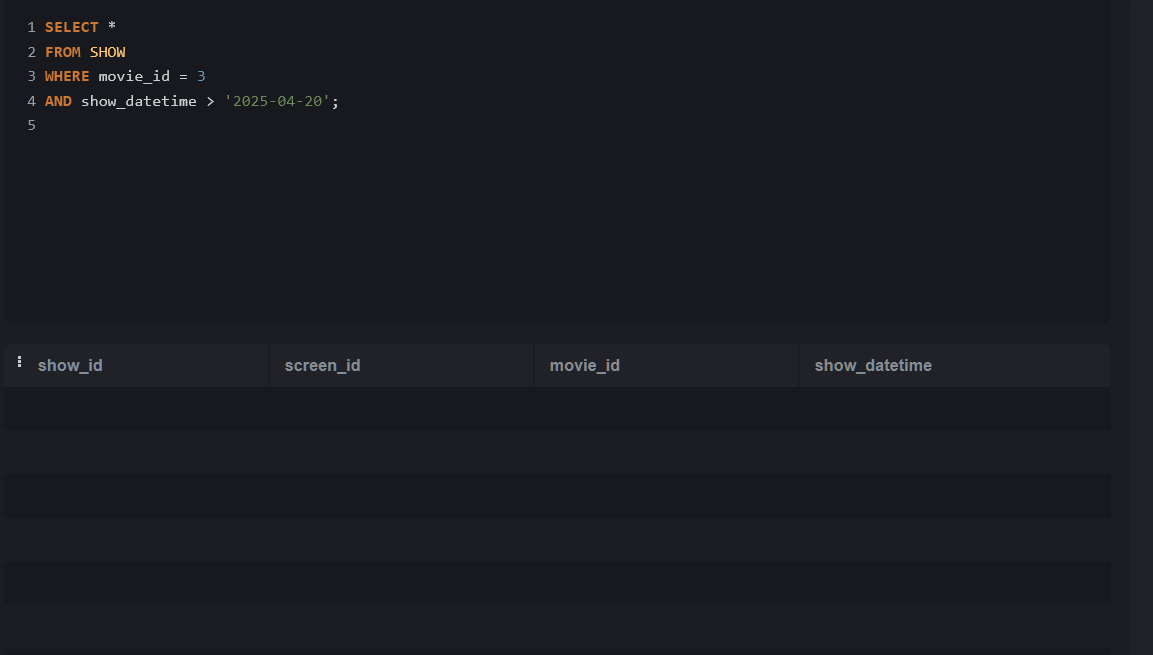
₹400.



**Topic 3: SELECT with WHERE using AND/OR**

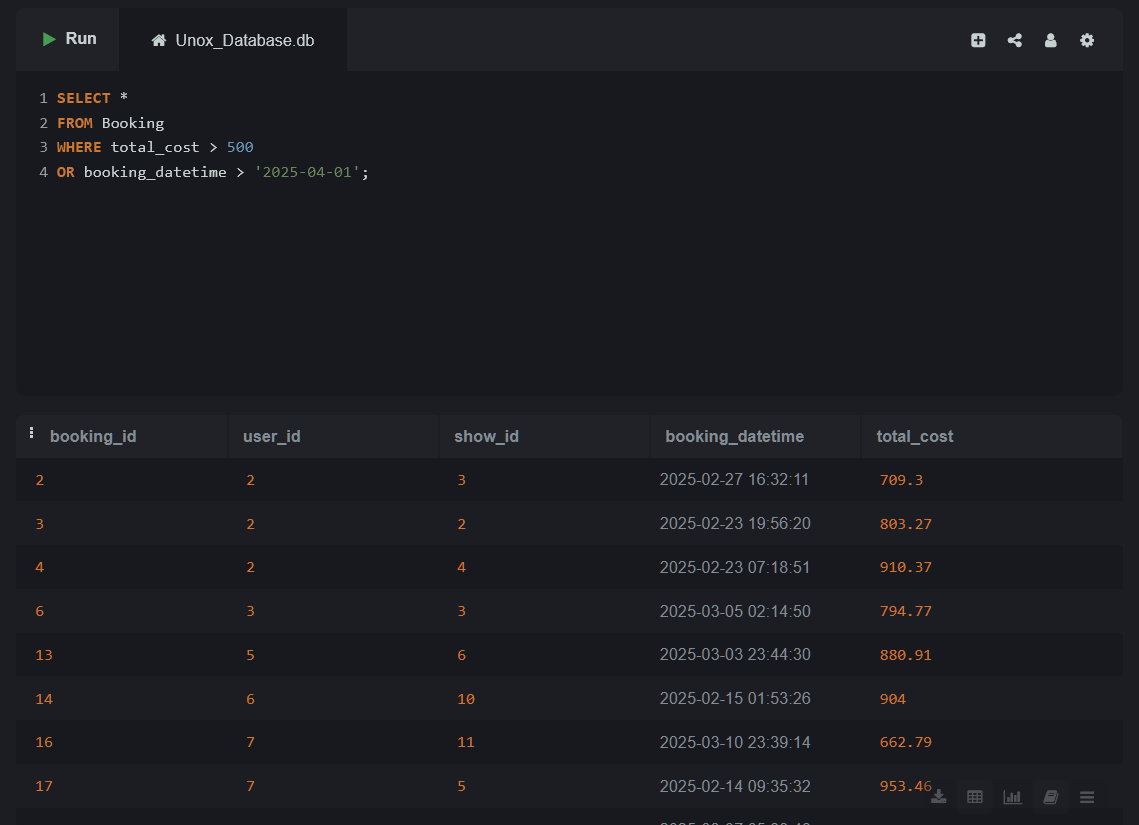
Lab 3.1 – show details where movie ID is 3 and show datetime is

after '2025-04-20'.



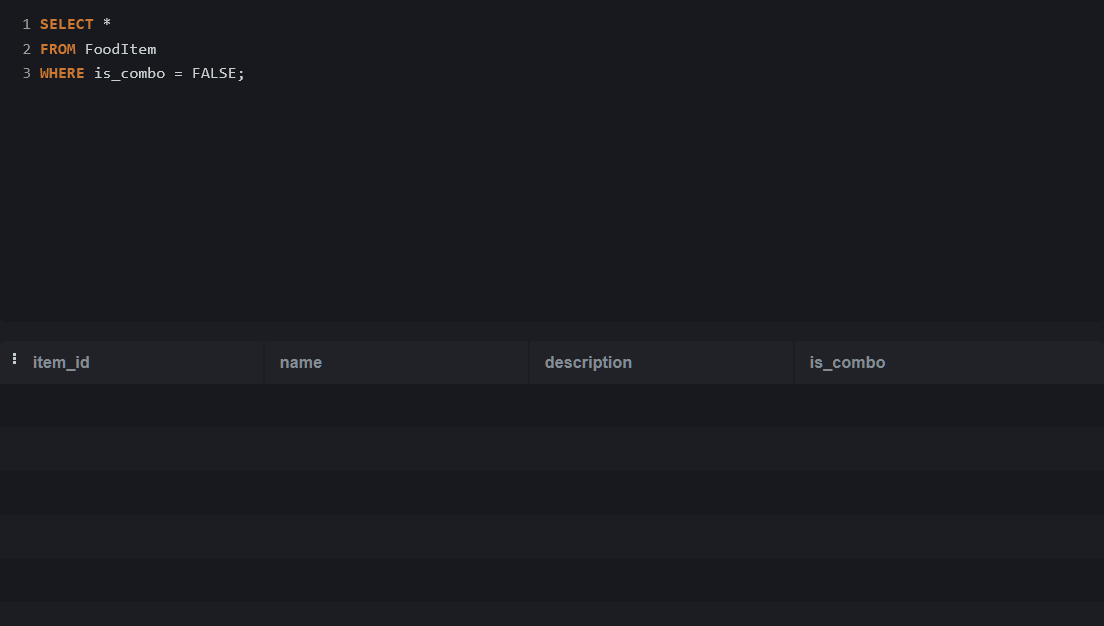
Lab 3.2 – List all bookings where the total cost is more than 500

OR the booking was made after '2025-04-01'.

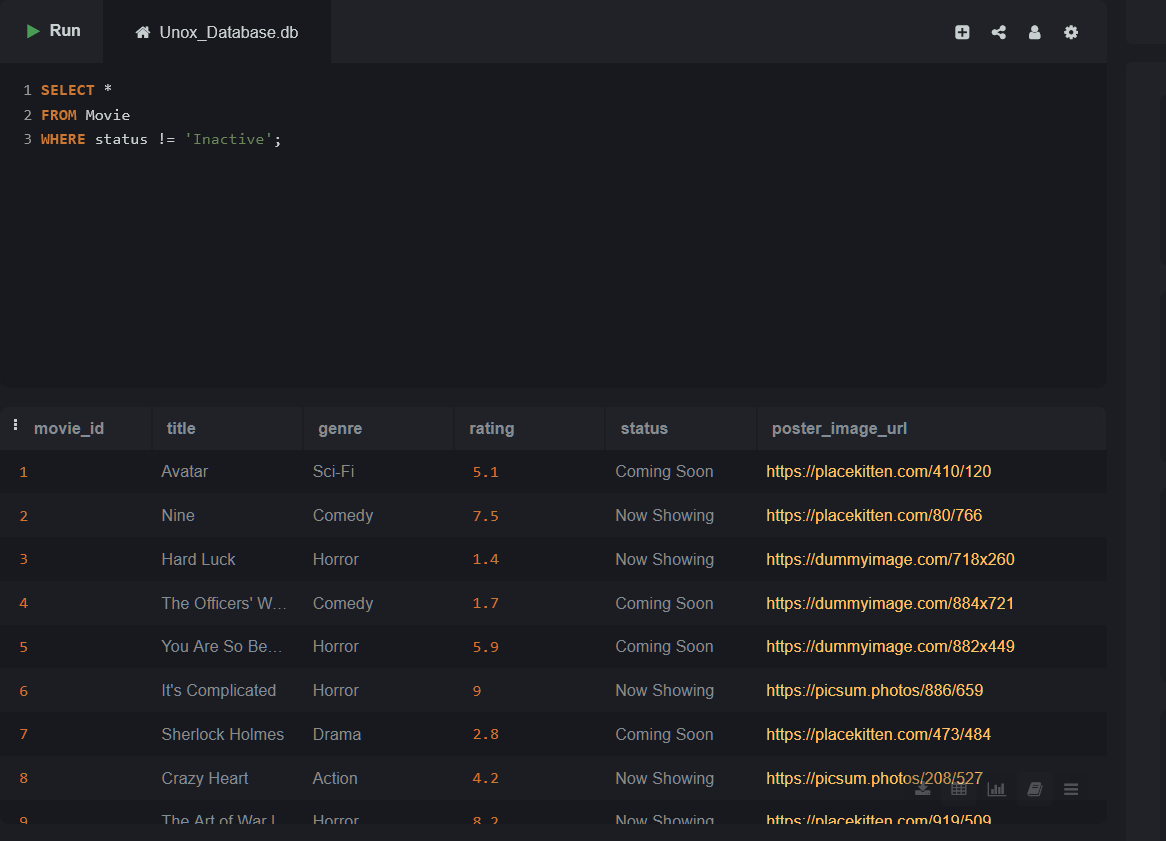


**Topic 4: SELECT with WHERE and NOT**

Lab 4.1 –Show all food items which are not combos.

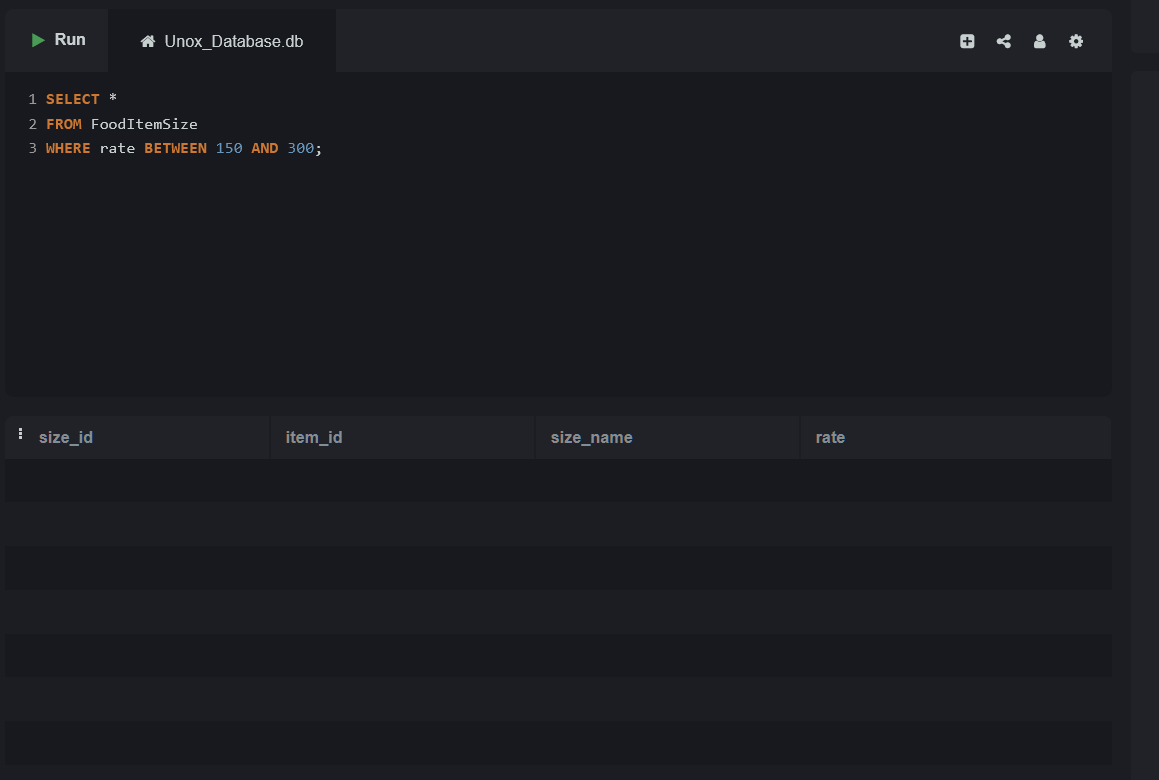


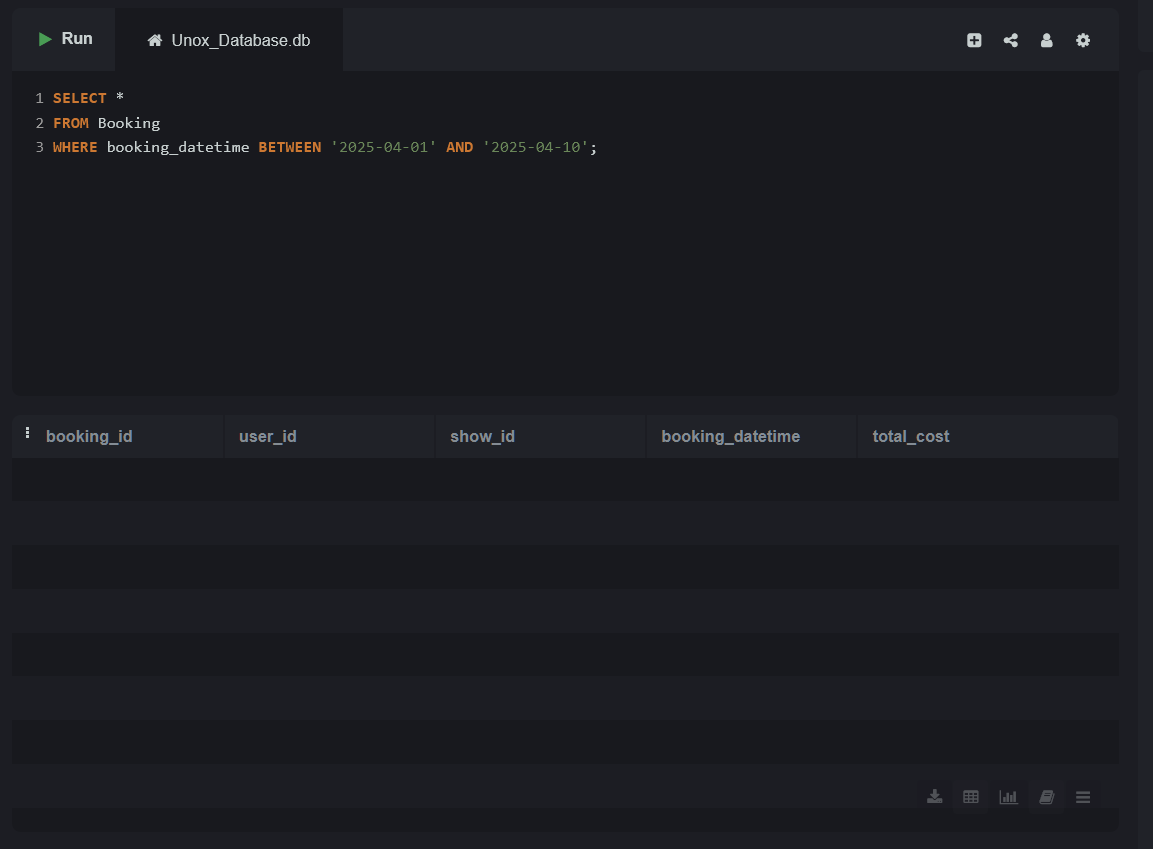
Lab 4.2 –Get all movie details that are not in status 'Inactive'.



**Topic 5: SELECT with WHERE and BETWEEN**

Lab 5.1 - List all food sizes where the rate is between 150 and 300.



Lab 5.2 – Get all ticket bookings between '2025-04-01' and '2025-04-10'.

**Topic 6: SELECT with WHERE and IN**

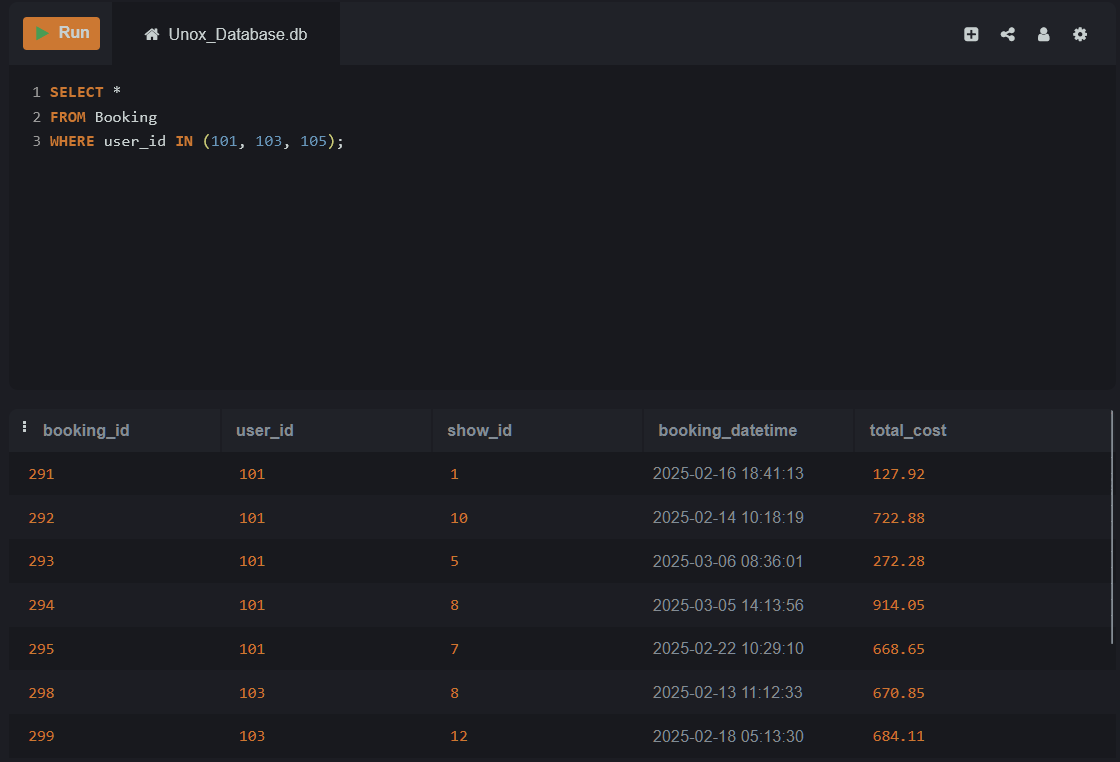
Lab 6.1 - Display food items whose names are either 'Vintage

Cola', 'Sweet & Salty Popcorn', 'Fiesta Nachos'.



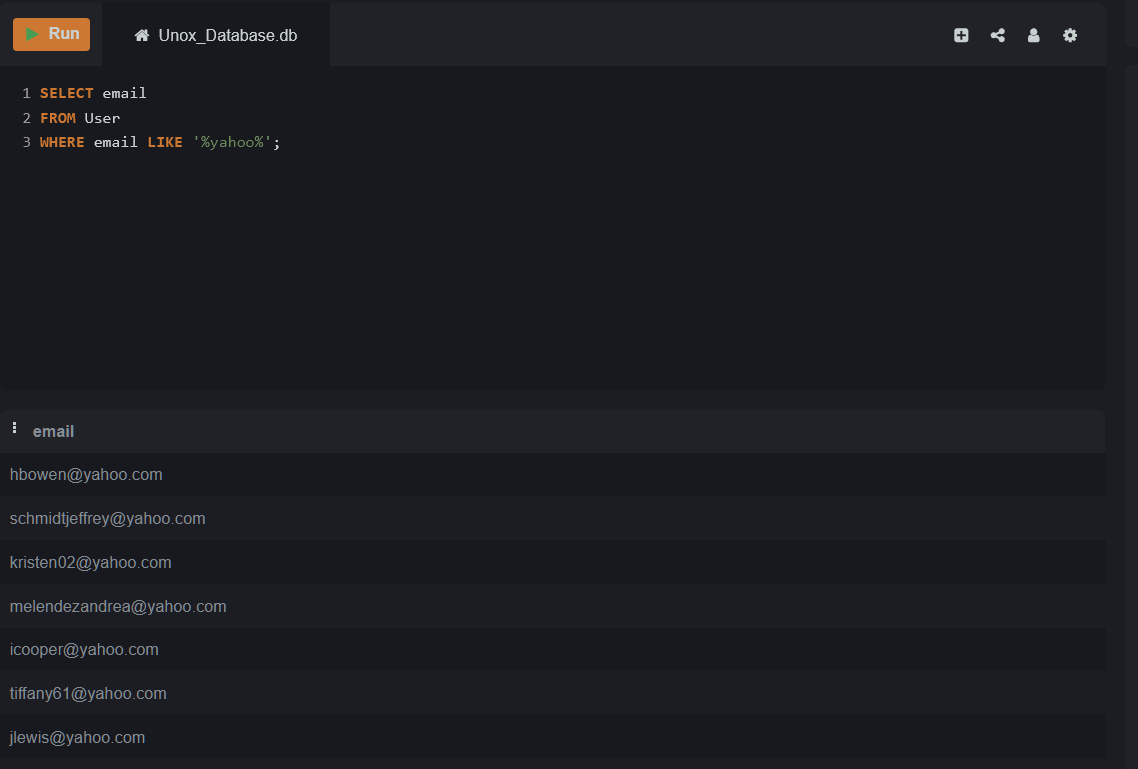
Lab 6.2 - List all bookings made by users with user IDs 101, 103, or

105

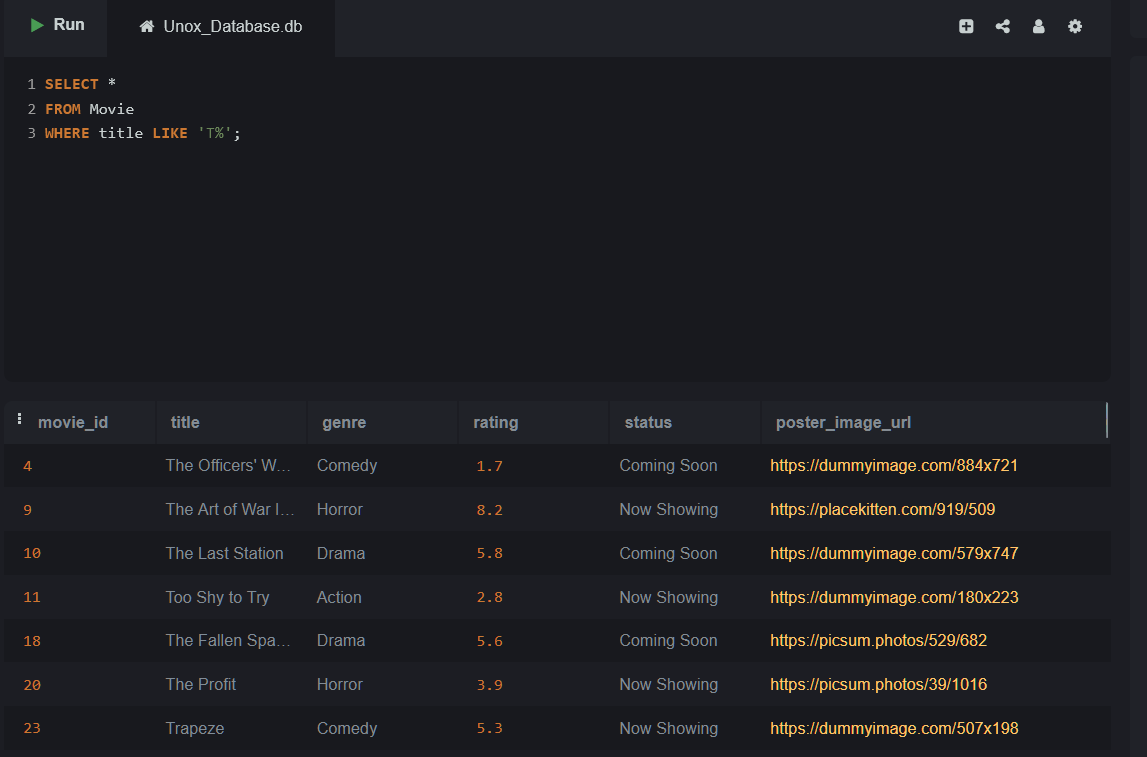


**Topic 7: SELECT with WHERE and LIKE**

Lab 7.1 – Retrieve all user emails that contain the word 'yahoo'.



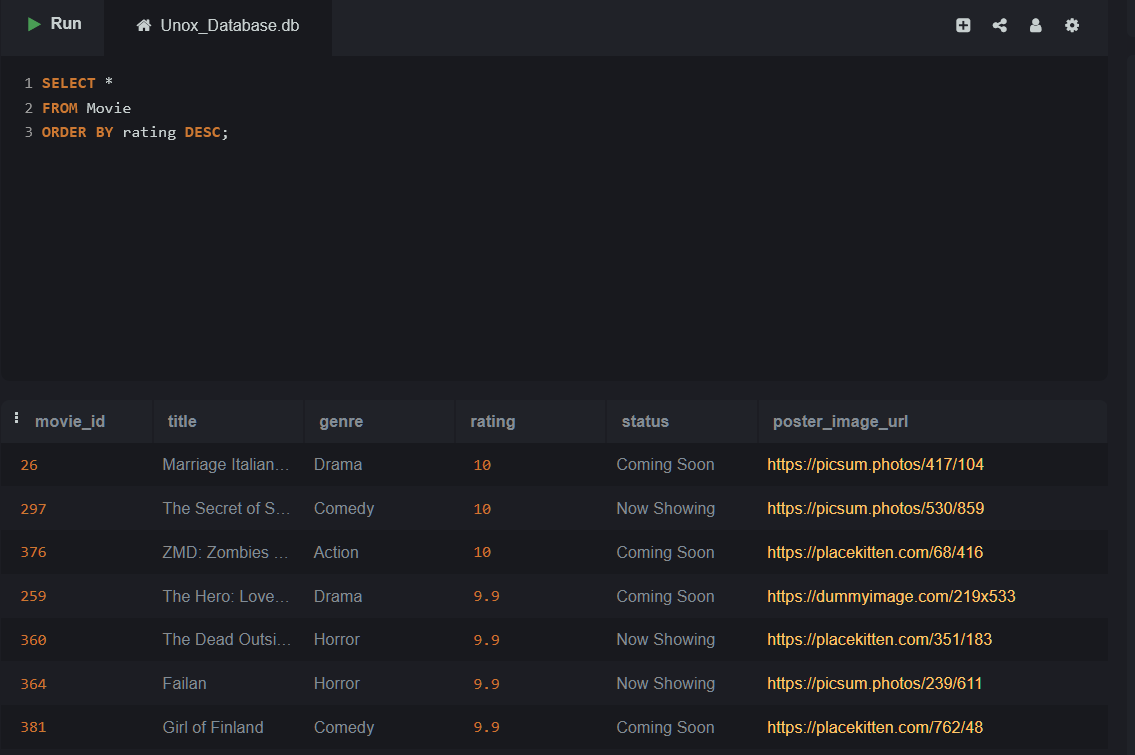
Lab 7.2 – Find all movies whose titles start with the letter 'T'.



**Topic 8: SELECT with ORDER BY (ASC/DESC)**

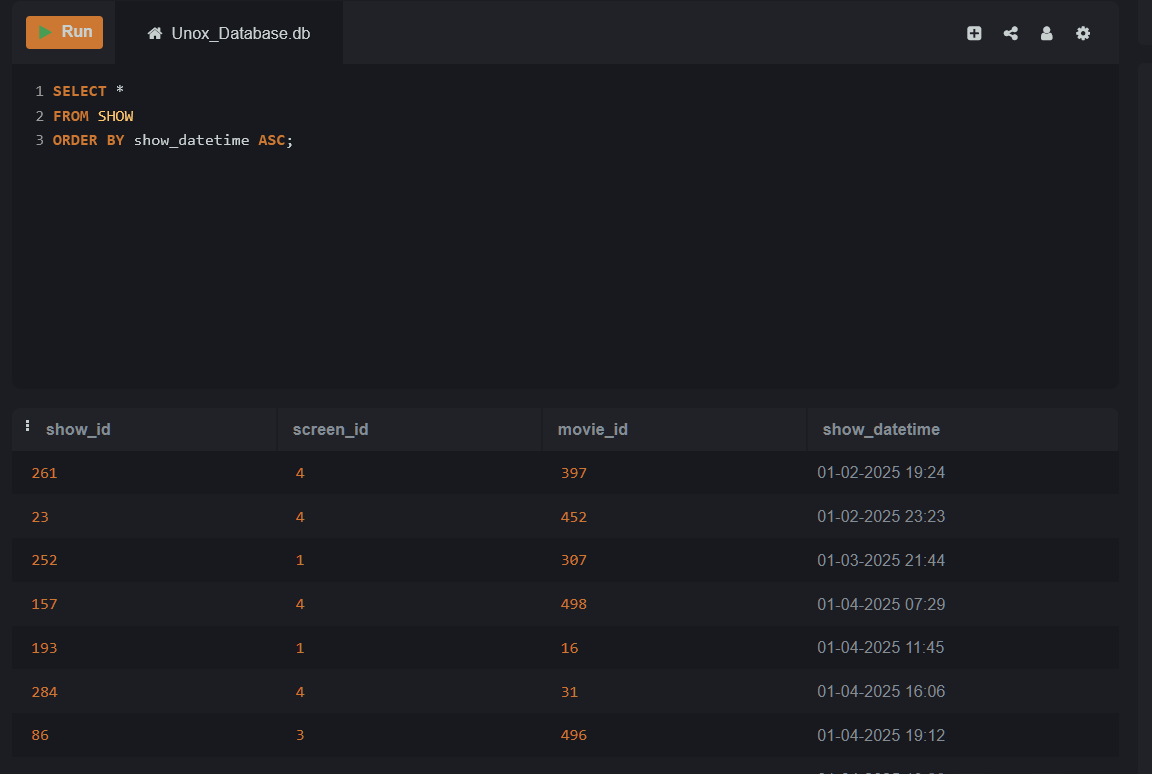
Lab 8.1 – List all movies ordered by their rating in descending

order.



Lab 8.2 – List all shows ordered by show\_datetime in ascending

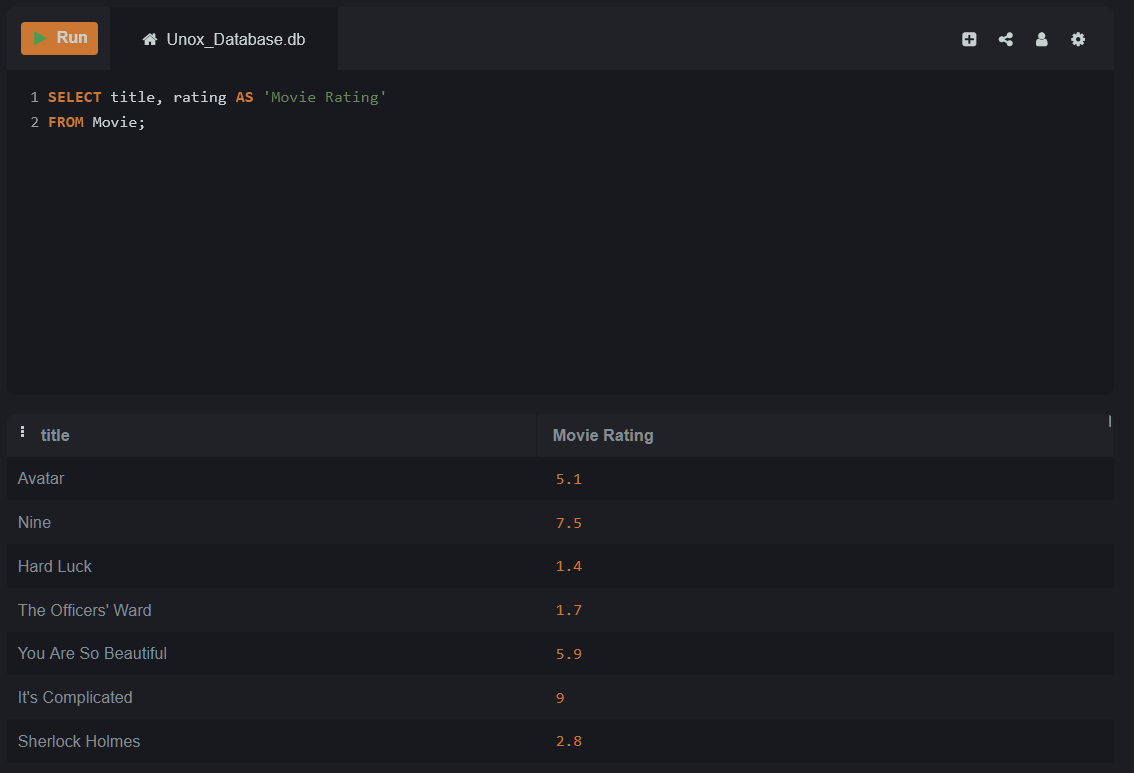
order.



**Topic 9: SELECT with ALIAS (column and table aliases)**

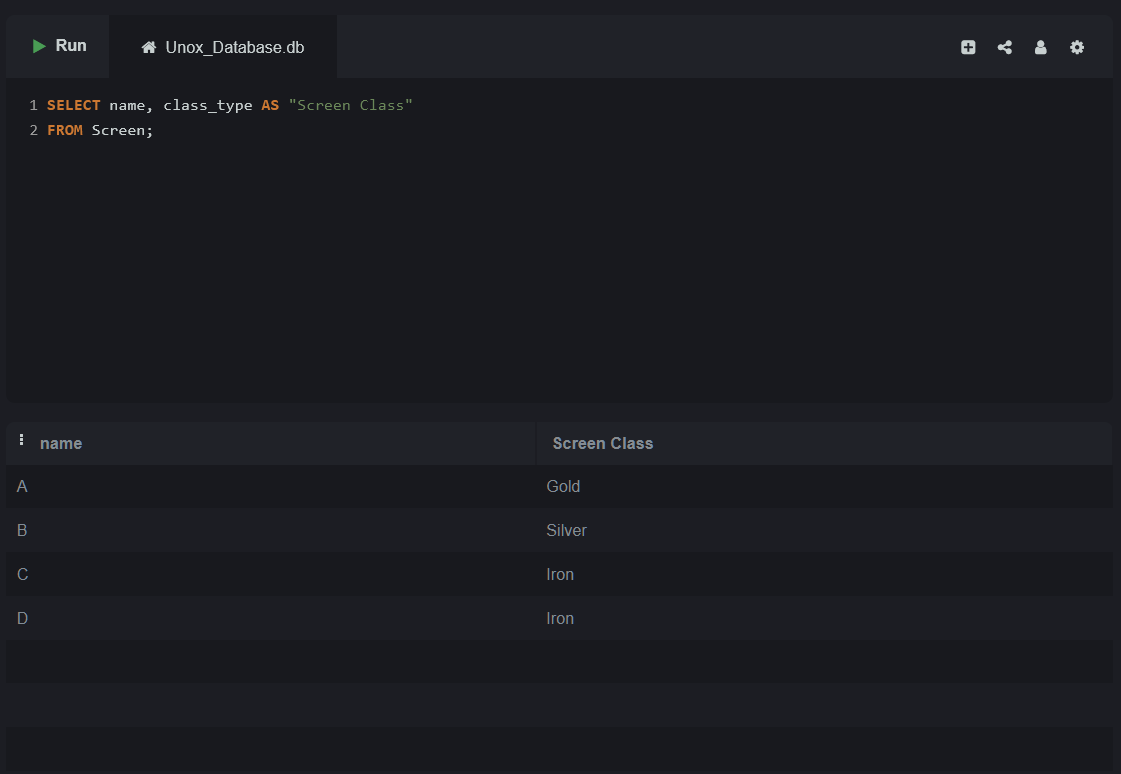
Lab 9.1 – Get the names and ratings of movies, but label the

rating as 'Movie Rating'.



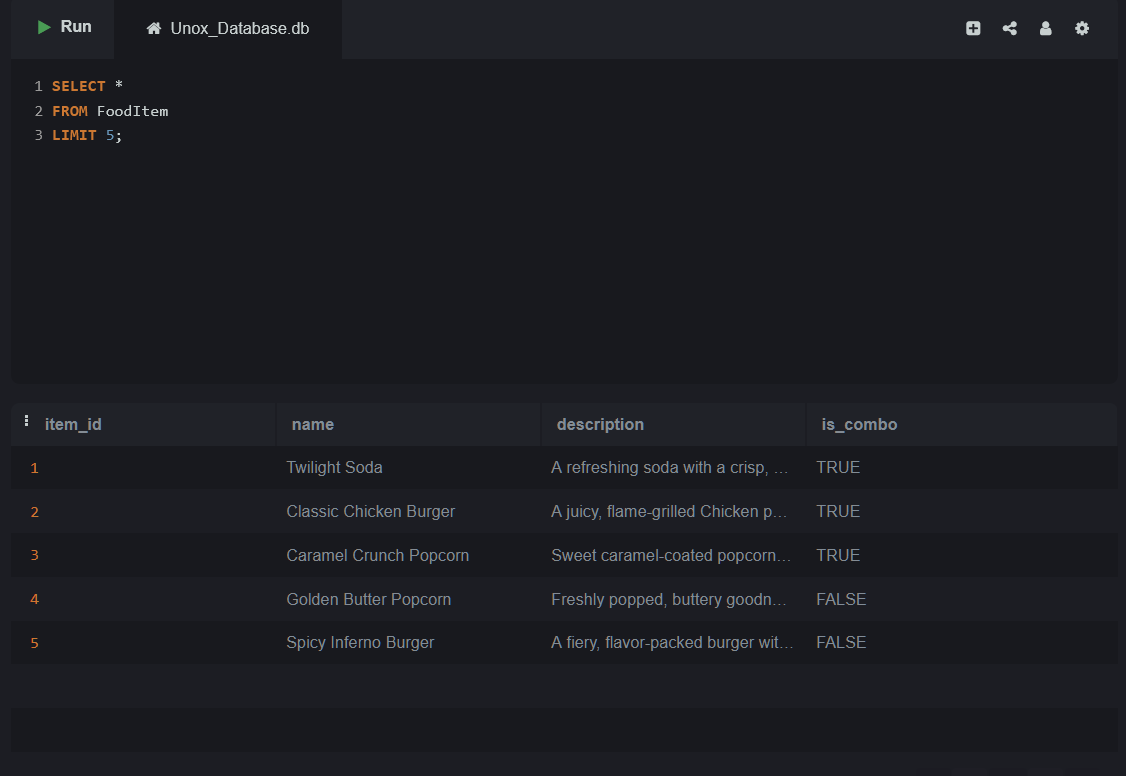
Lab 9.2 – Show screen names and their types, but alias class\_type

as "Screen Class".



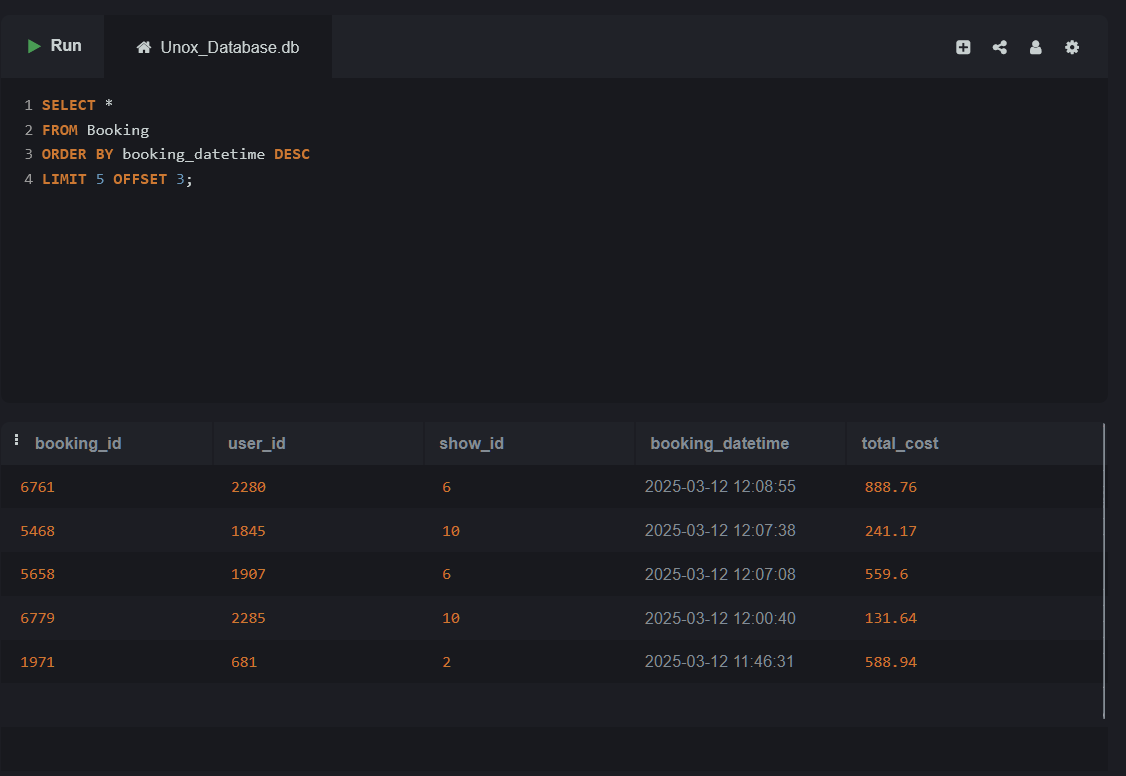
**Topic 10: SELECT with LIMIT and OFFSET**

Lab 10.1 – Fetch the first 5 food items from the fooditem table.



Lab 10.2 – Retrieve the 5 most recent bookings, skipping the

latest 3.

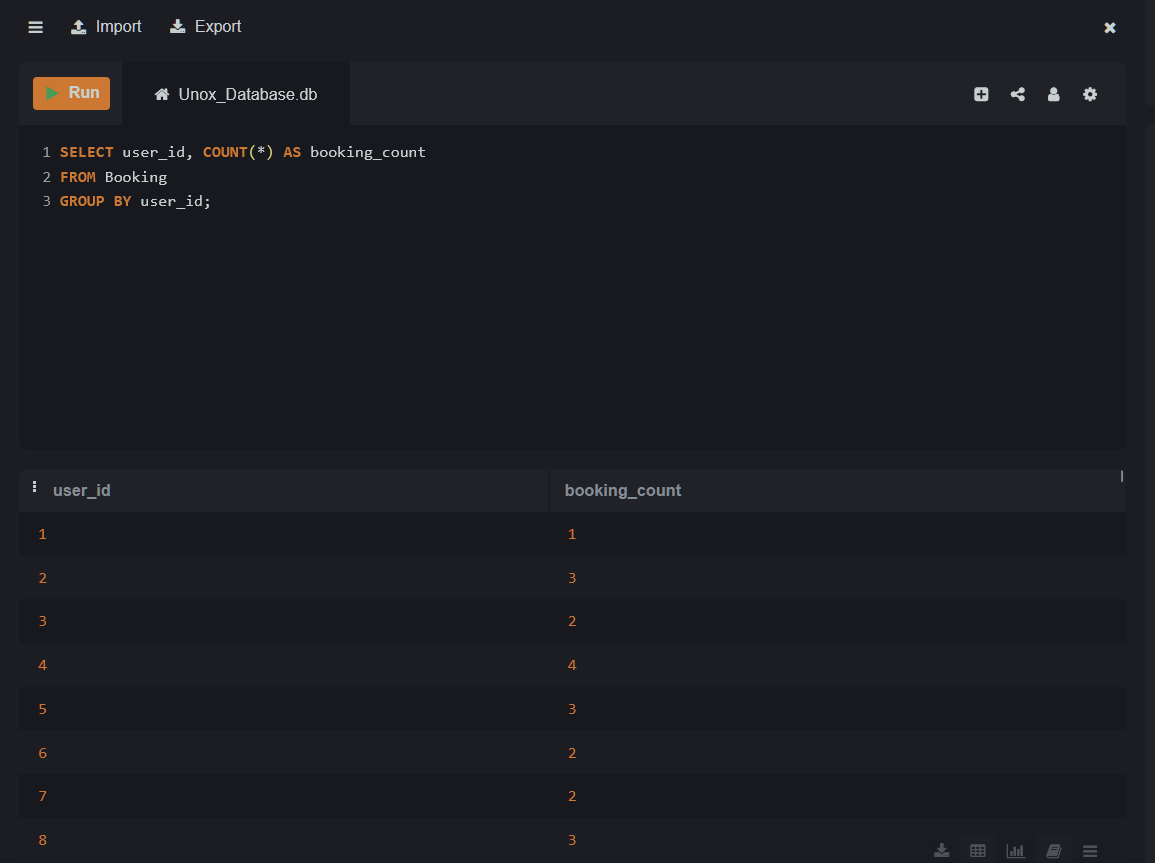


**Topic 11: SELECT with Aggregate Functions (SUM, AVG,**

**COUNT, MIN, MAX)**

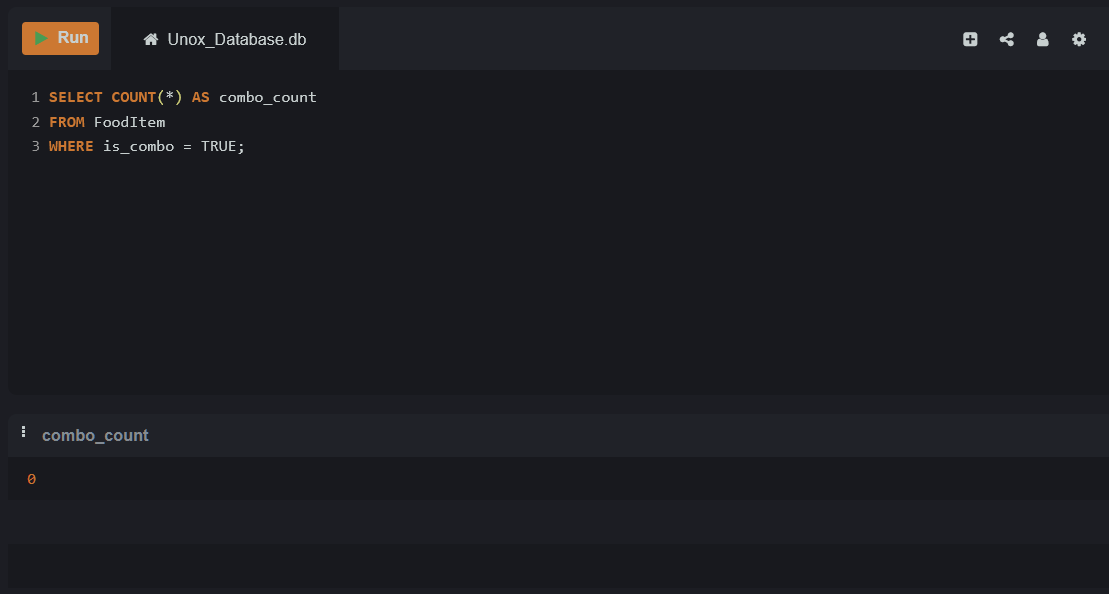
**11.1: SELECT with Aggregate Function – COUNT**

Lab 11.1.1 – Count how many bookings each user has made.



Lab 11.1.2 – Find the total number of food items marked as

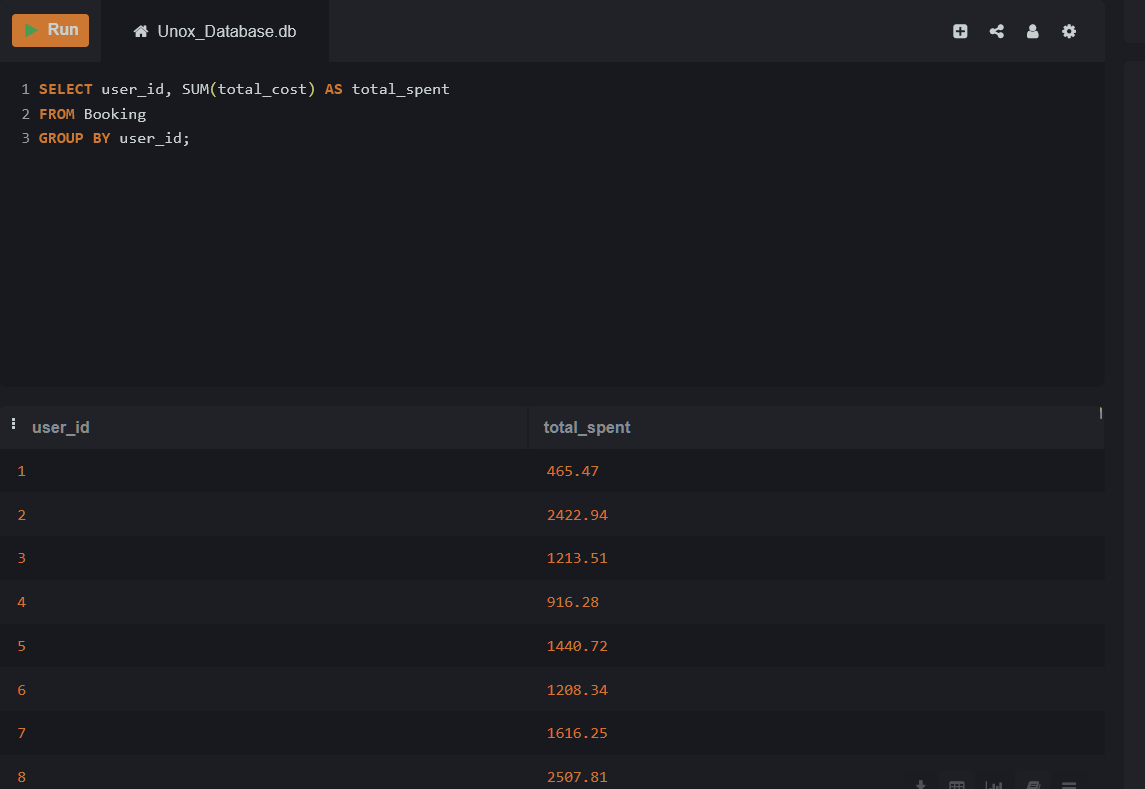
combo.



**11.2: SELECT with Aggregate Function – SUM**

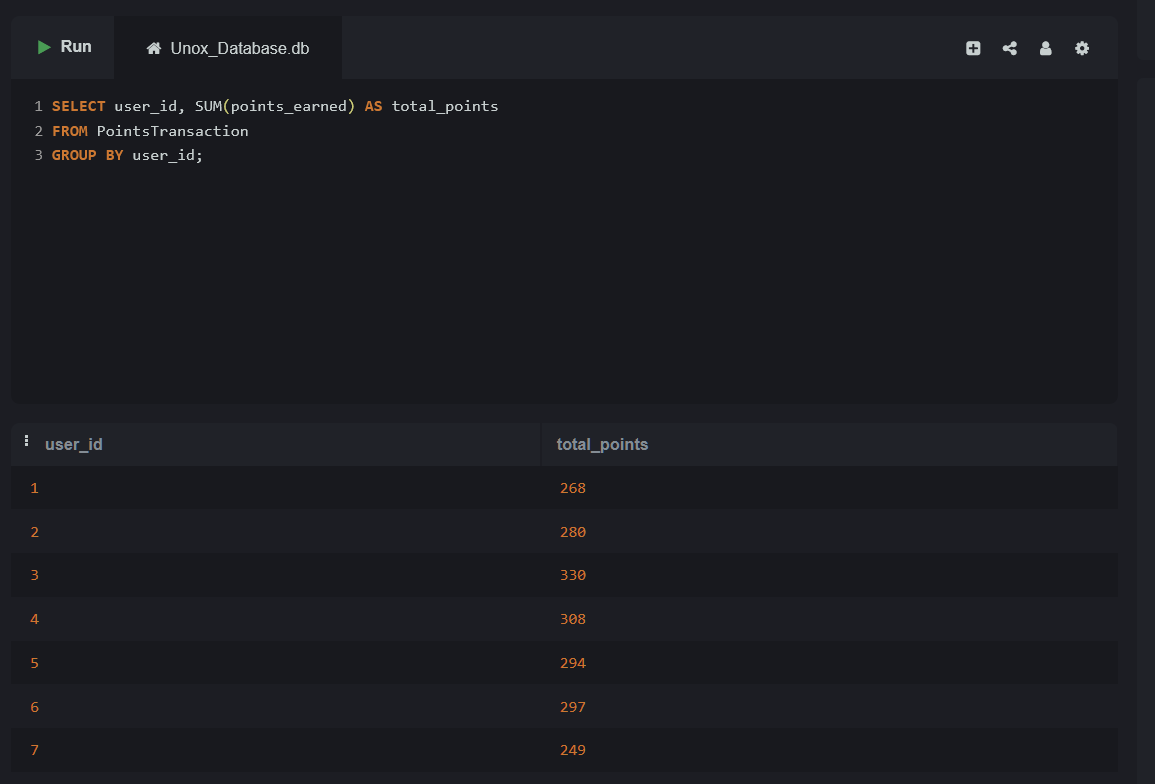
Lab 11.2.1 – Find the total amount spent by each user on

bookings.



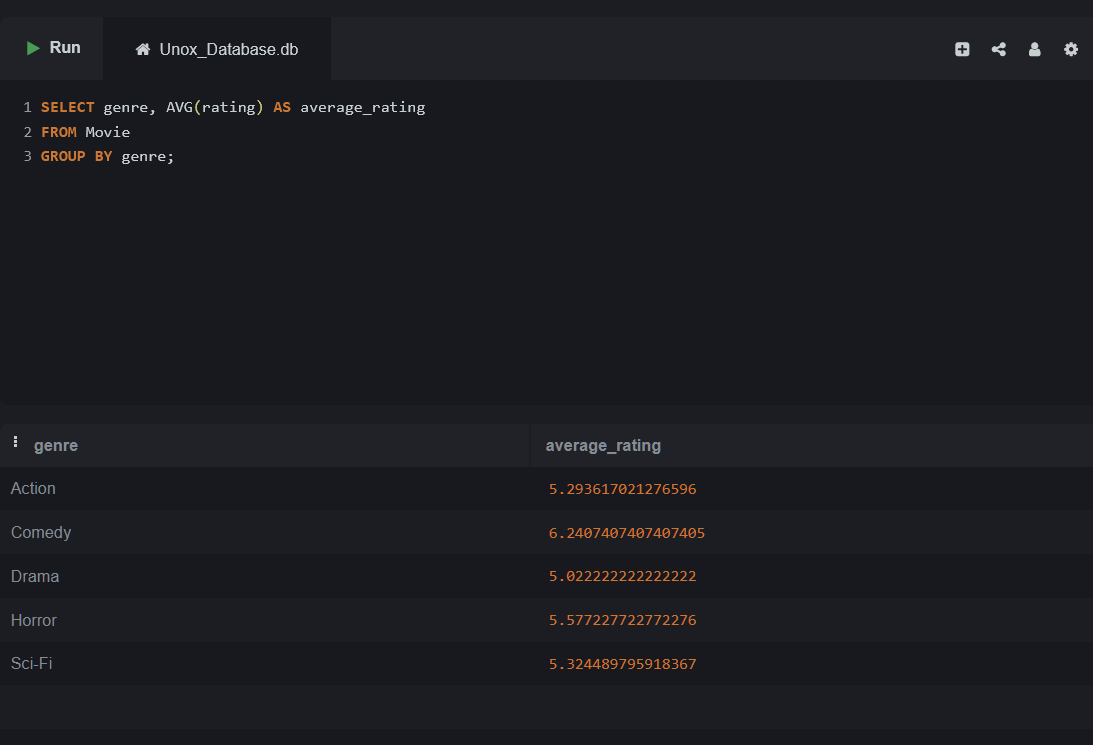
Lab 11.2.2 – Calculate the total loyalty points transactions done by

each user.



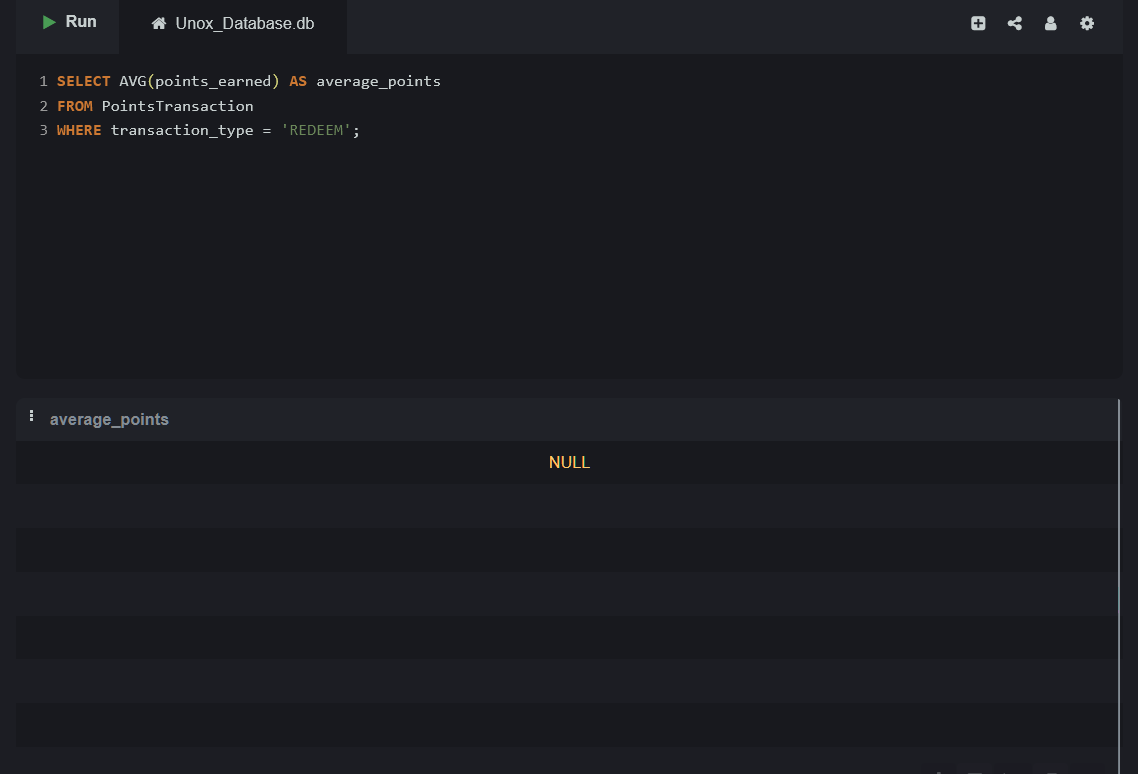
**11.3: SELECT with Aggregate Function – AVG**

Lab 11.3.1 – Find the average rating for each movie genre.



Lab 11.3.2 – Find the average amount of loyalty points used per

transaction.© Talenciaglobal, 2025



**11.4: SELECT with Aggregate Function – MIN and MAX**

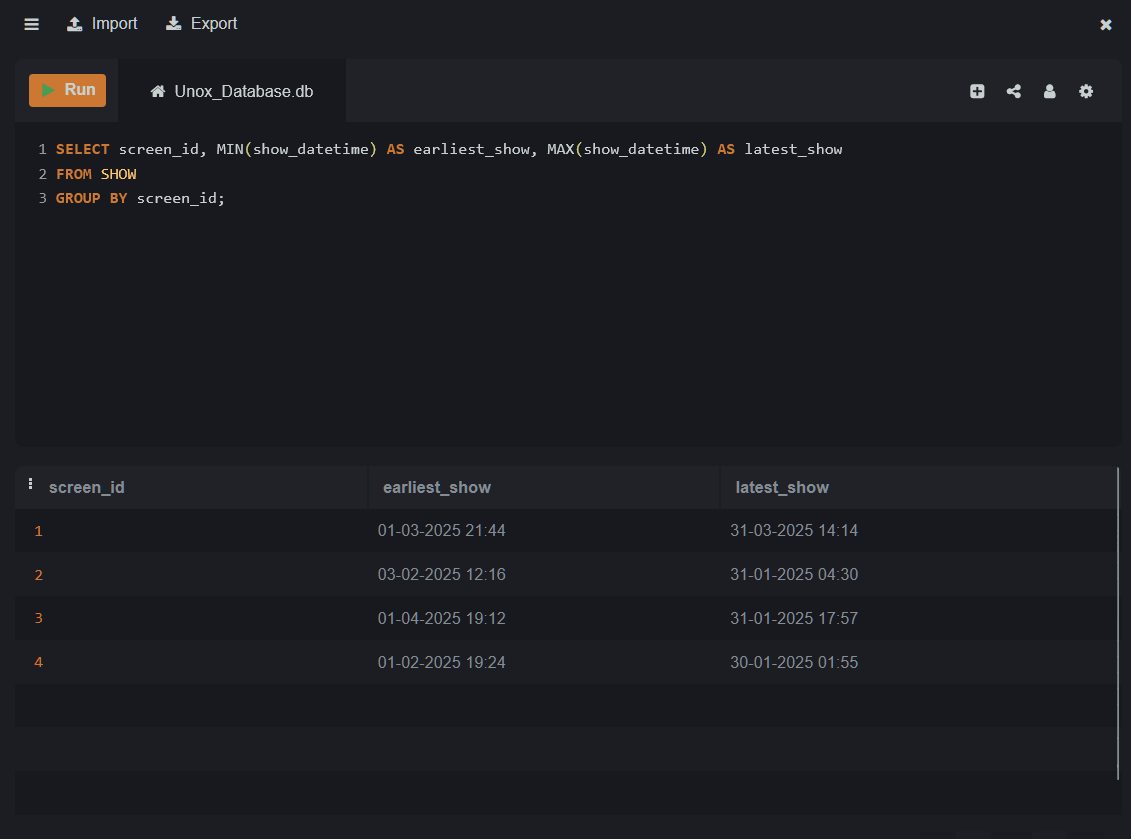
Lab 11.4.1 – Find the highest and lowest rated movies for each

genre.



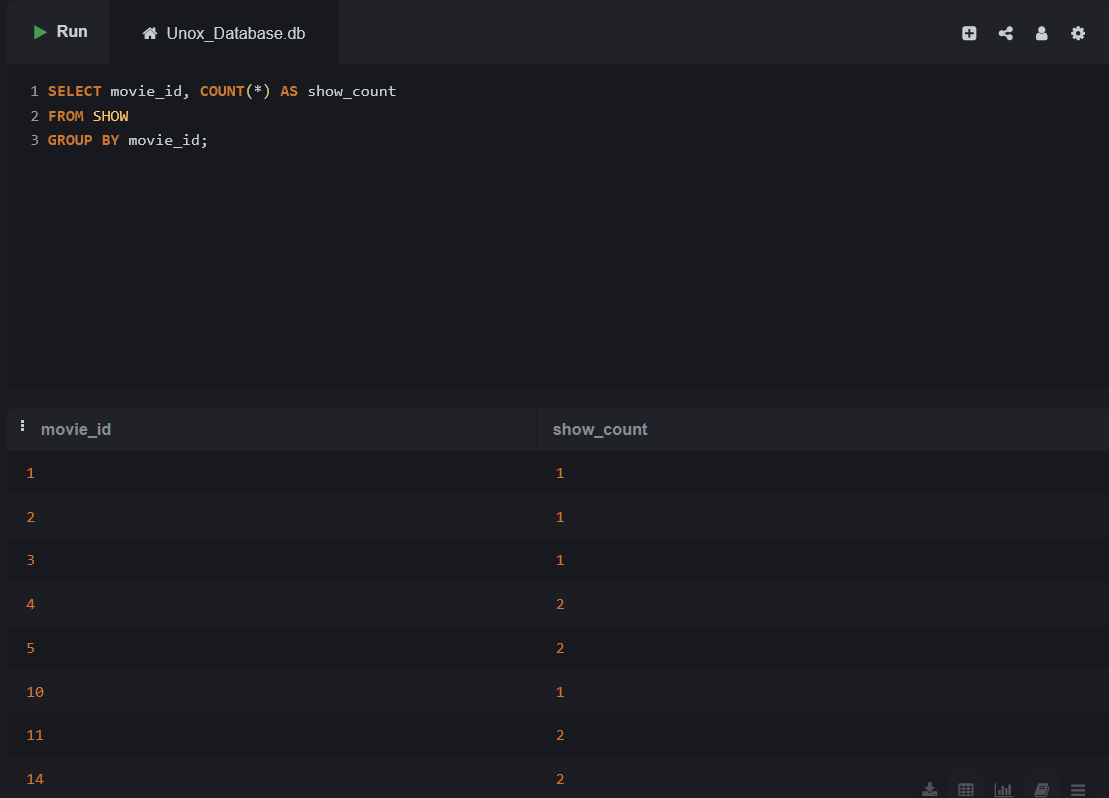
Lab 11.4.2 – Find the earliest and latest show timings for each

screen.



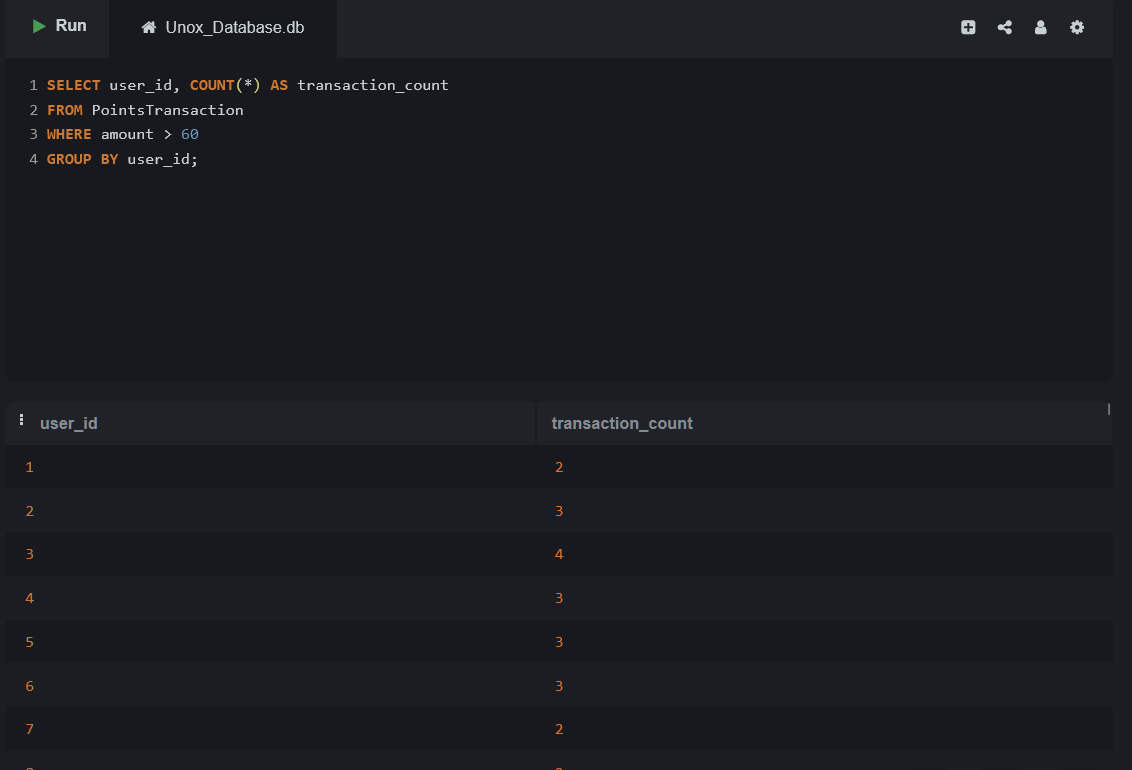
**Topic 12: GROUP BY with Aggregate Functions**

Lab 12.1 – List the number of shows scheduled for each movie.



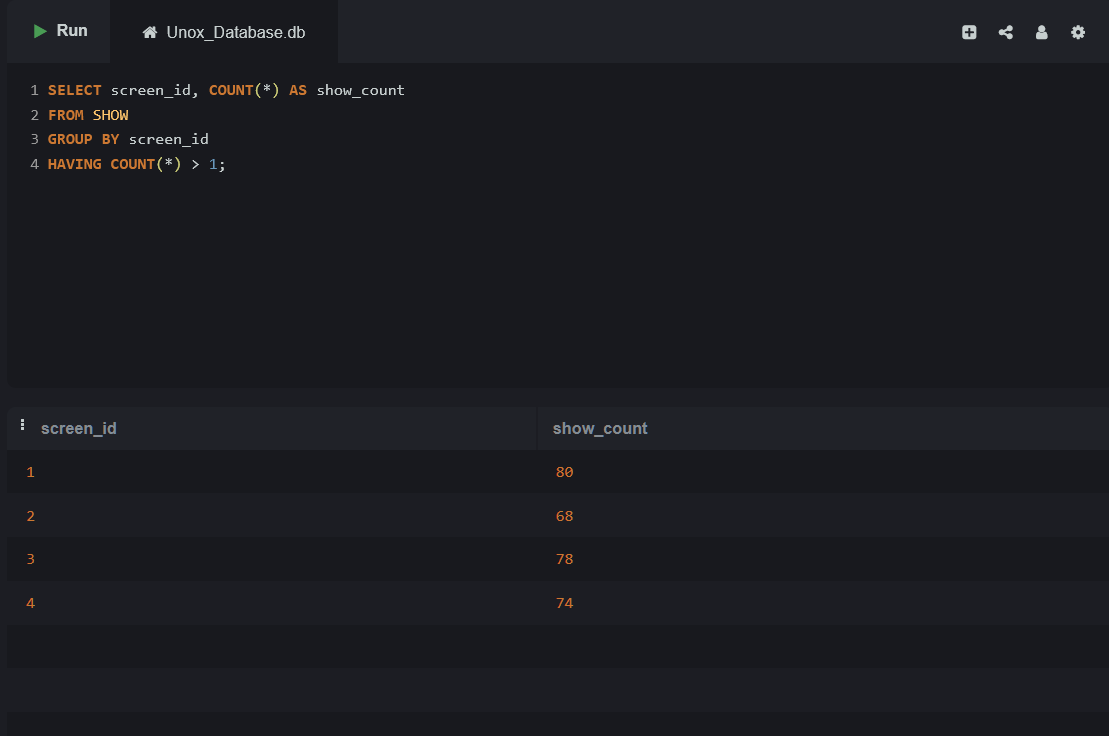
Lab 12.2 – Count the number of point transactions per user where

the amount is greater than 60.

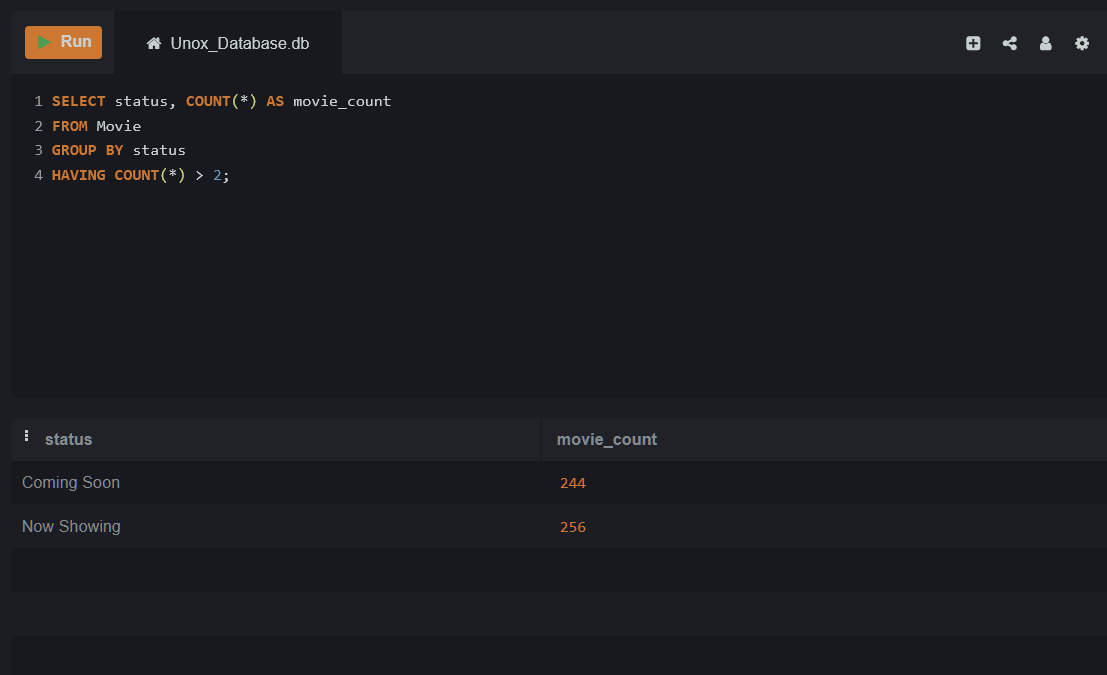


**Topic 13: GROUP BY with HAVING Clause**

Lab 13.1 – Find screens that have more than 1 show scheduled.



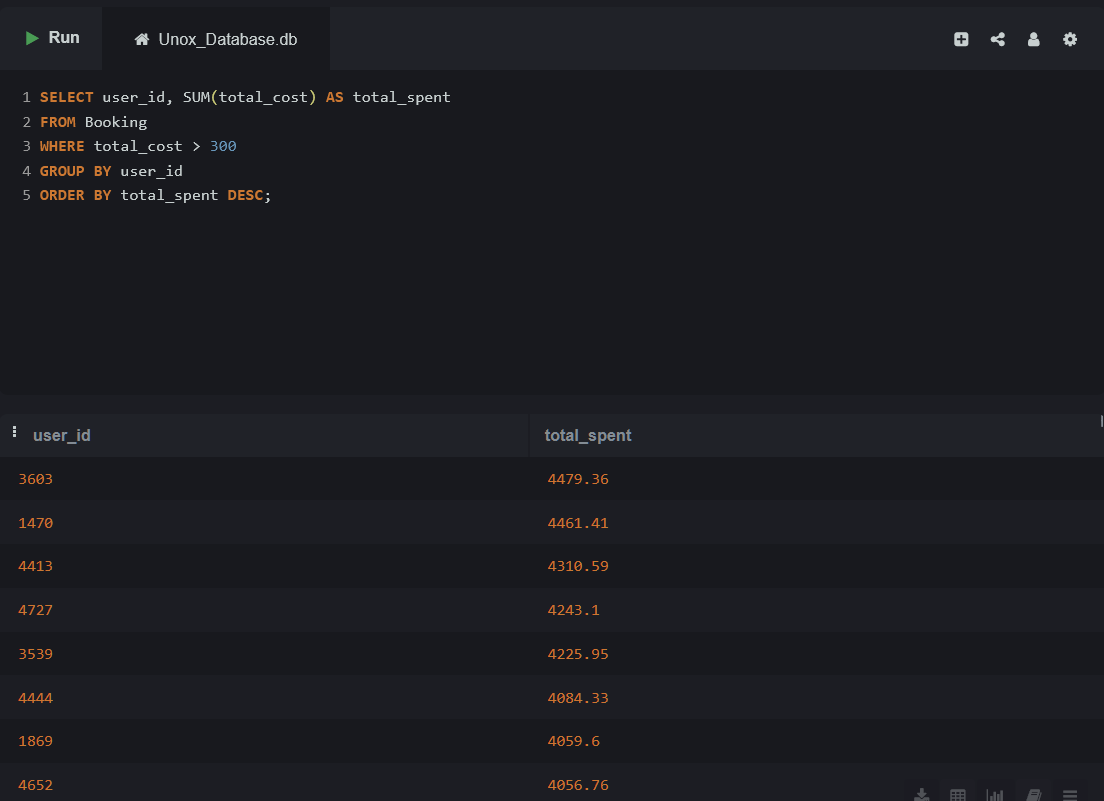
Lab 13.2 – Find movie statuses with more than 2 movies.© Talenciaglobal, 2025



**Topic 14: GROUP BY with WHERE and ORDER BY**

Lab 14.1 – Find the total amount spent per user for bookings over

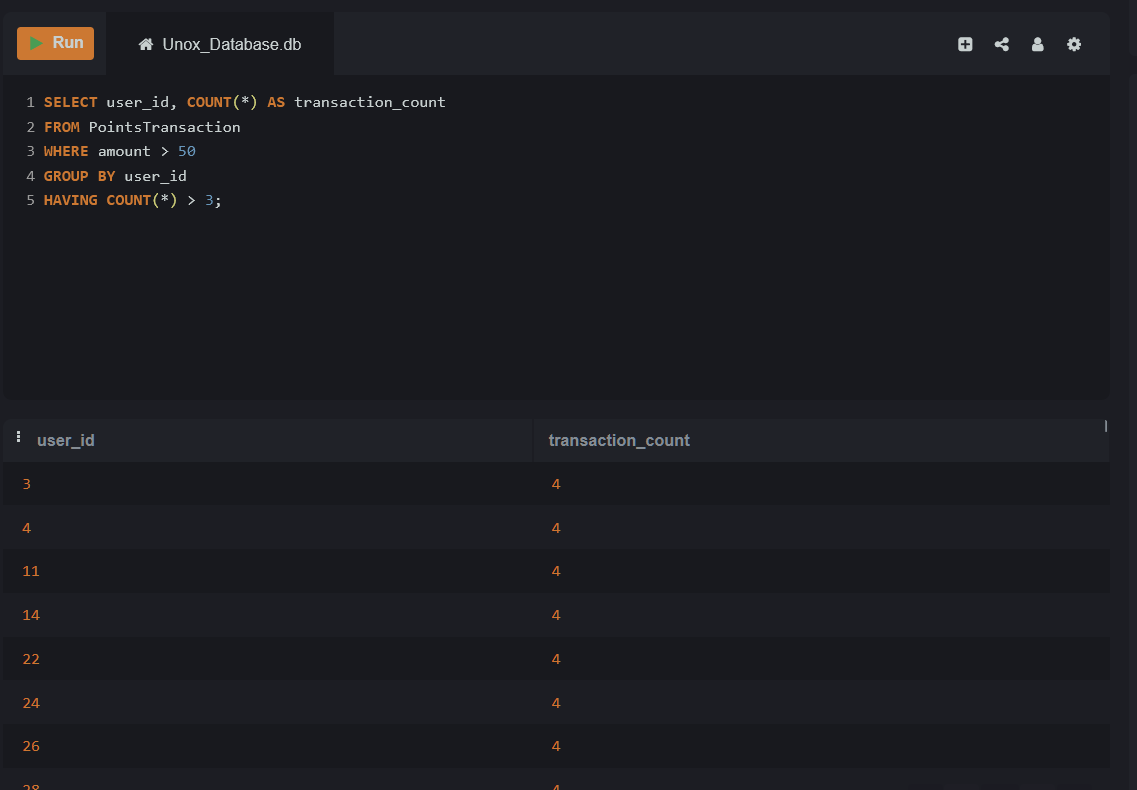
₹300, sorted in descending order of total.



Lab 14.2 – Group point transactions by user where amount is

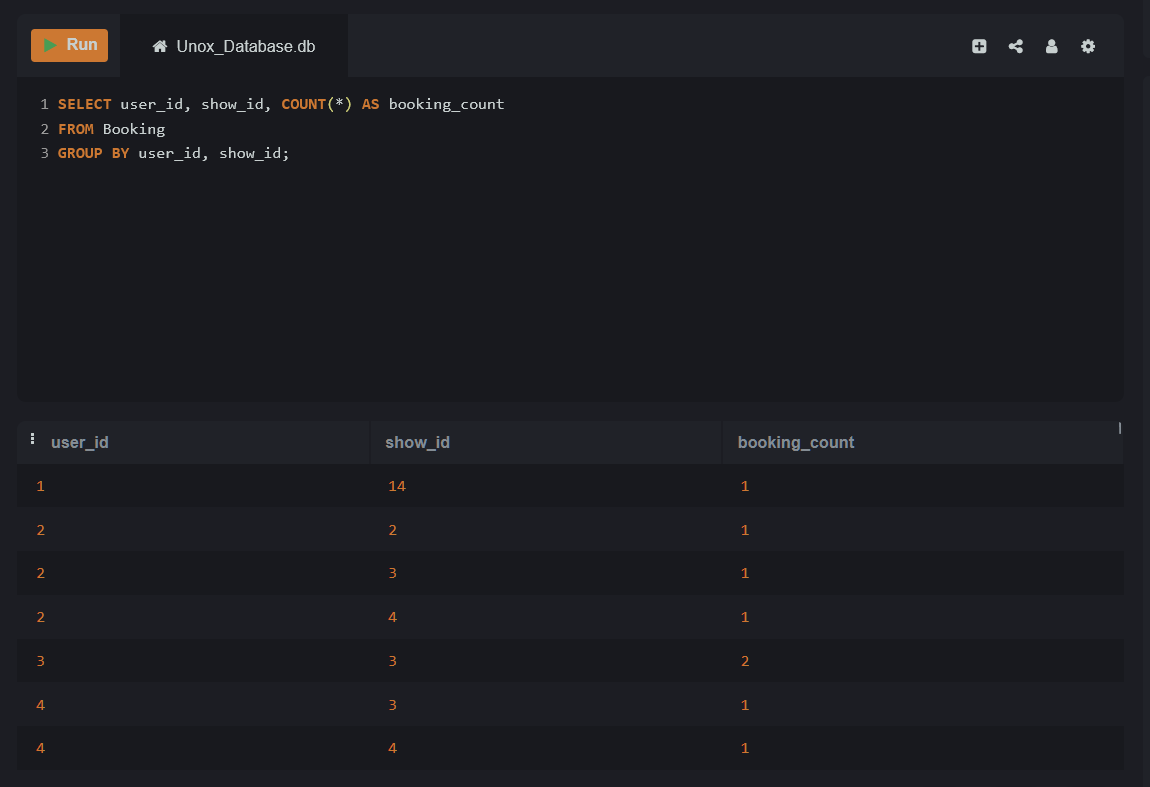
more than 50 and show only users with more than 3 such

transactions.



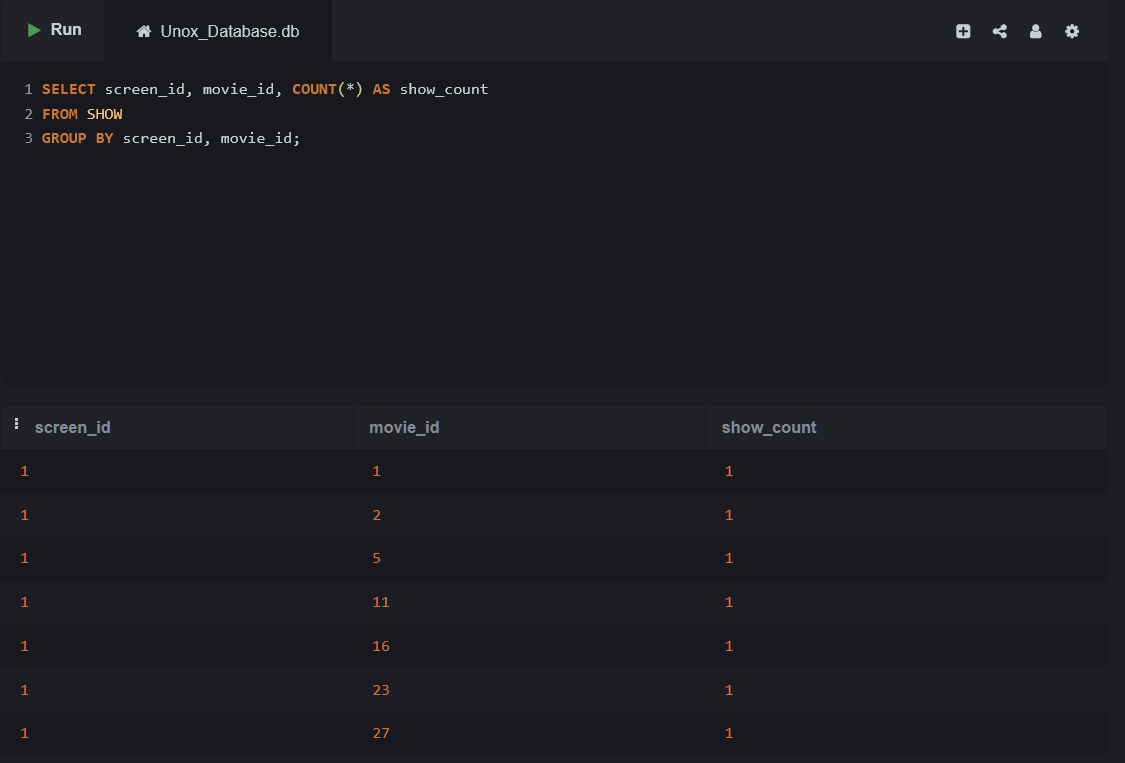
**Topic 15: GROUP BY Multiple Columns**

Lab 15.1 – Show the number of bookings per user per show.



Lab 15.2 – Get the number of shows scheduled per screen per

movie.

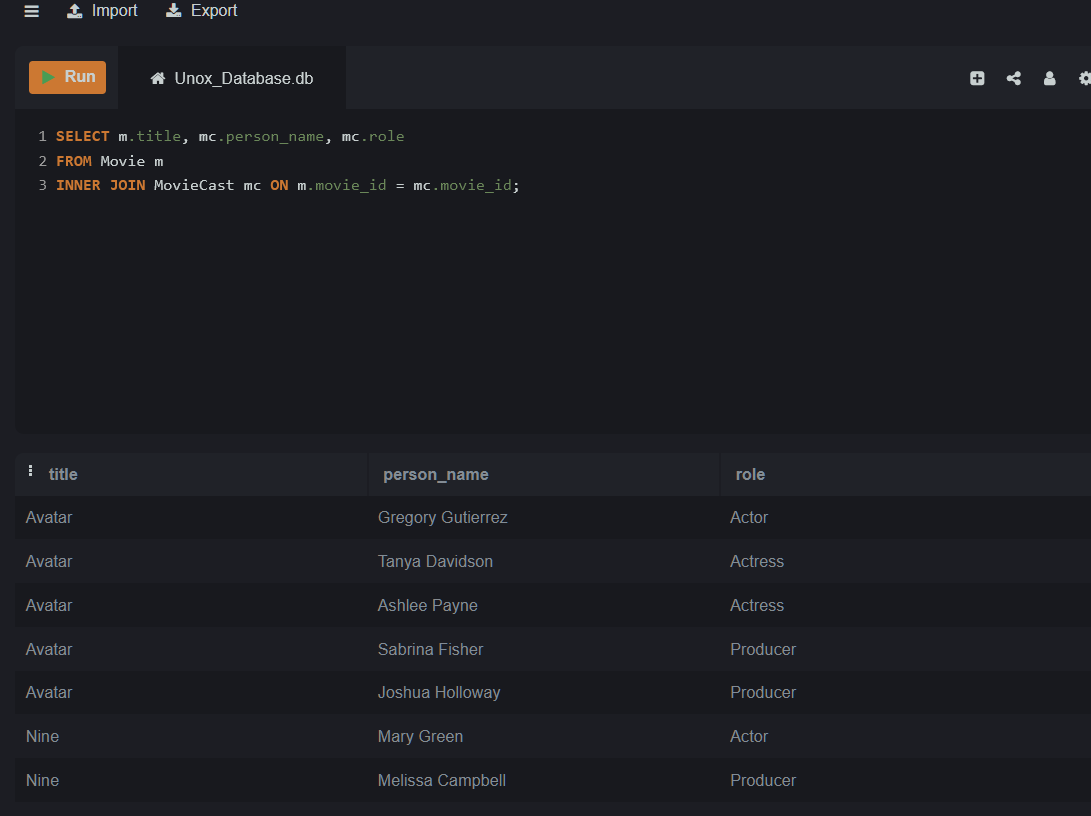


**Topic16. INNER JOIN**

Lab 16.1 – Write an SQL query to retrieve a list of all movies along

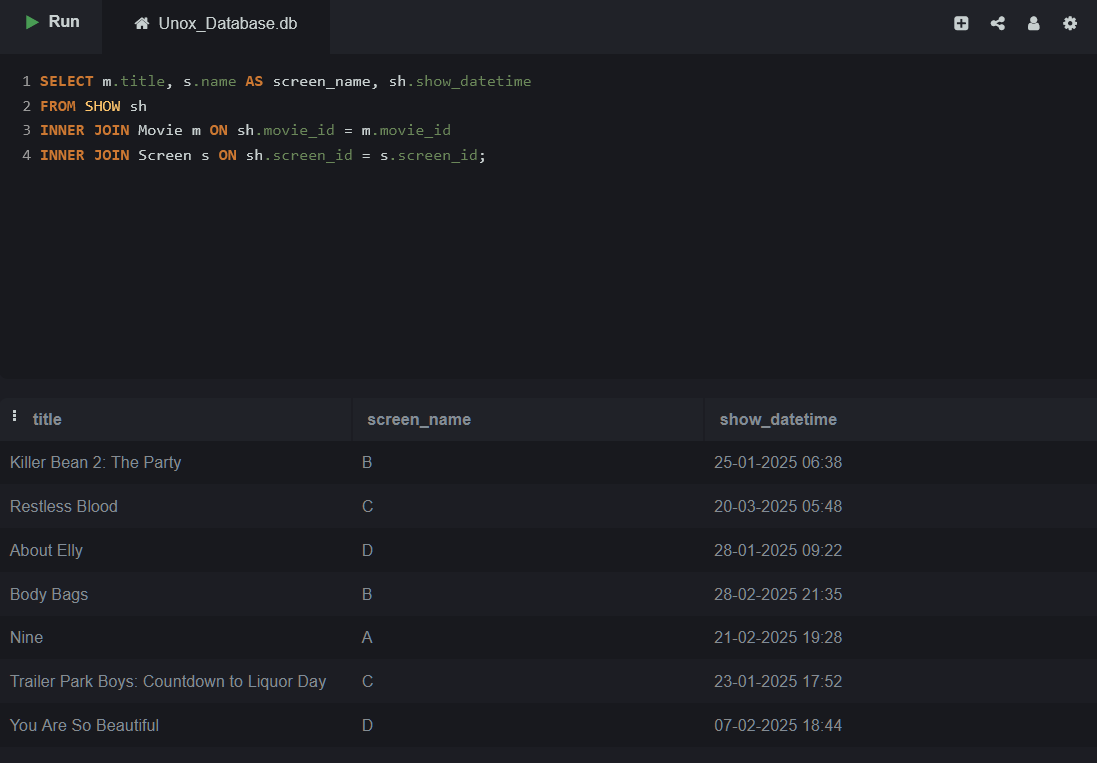
with their cast members. Include the movie title, the name of the

cast, and the role played by the cast.



Lab 16.2 – Write an SQL query to get a list of all shows, including

the movie title, the screen name, and the show datetime.© Talenciaglobal, 2025

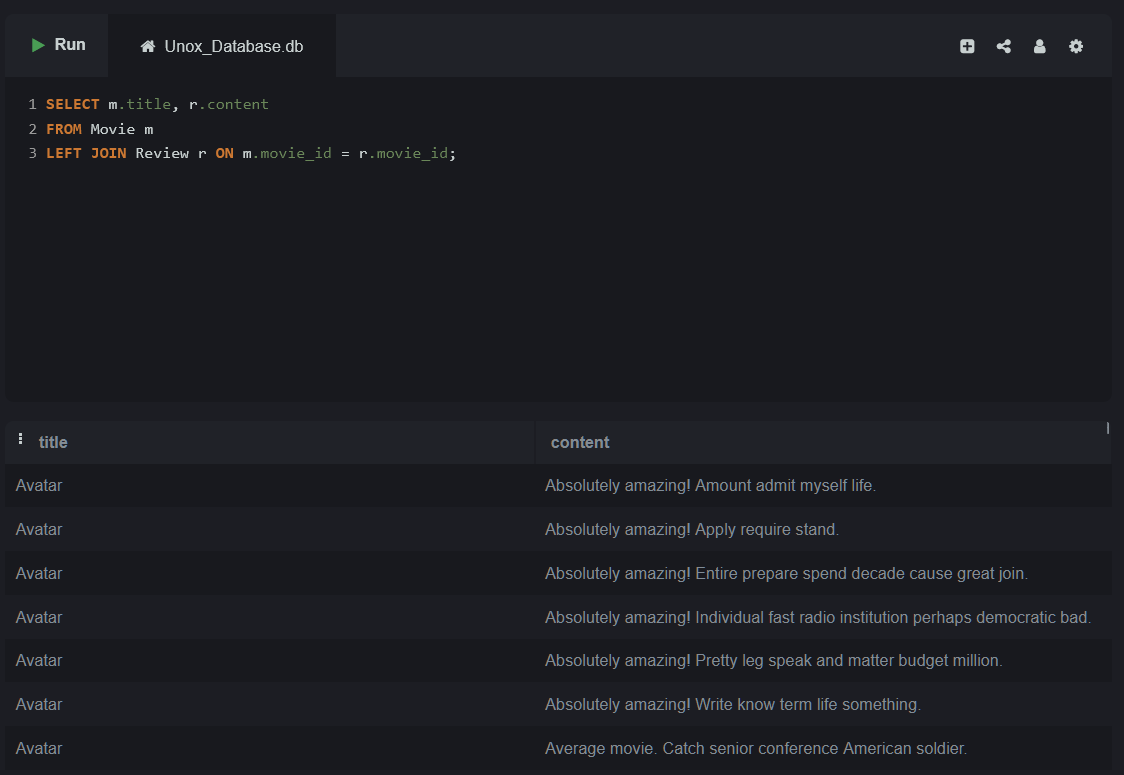


**Topic 17: LEFT JOIN**

Lab 17.1 – Write an SQL query to list all movies and their reviews,

including movies that have no reviews. The result should include

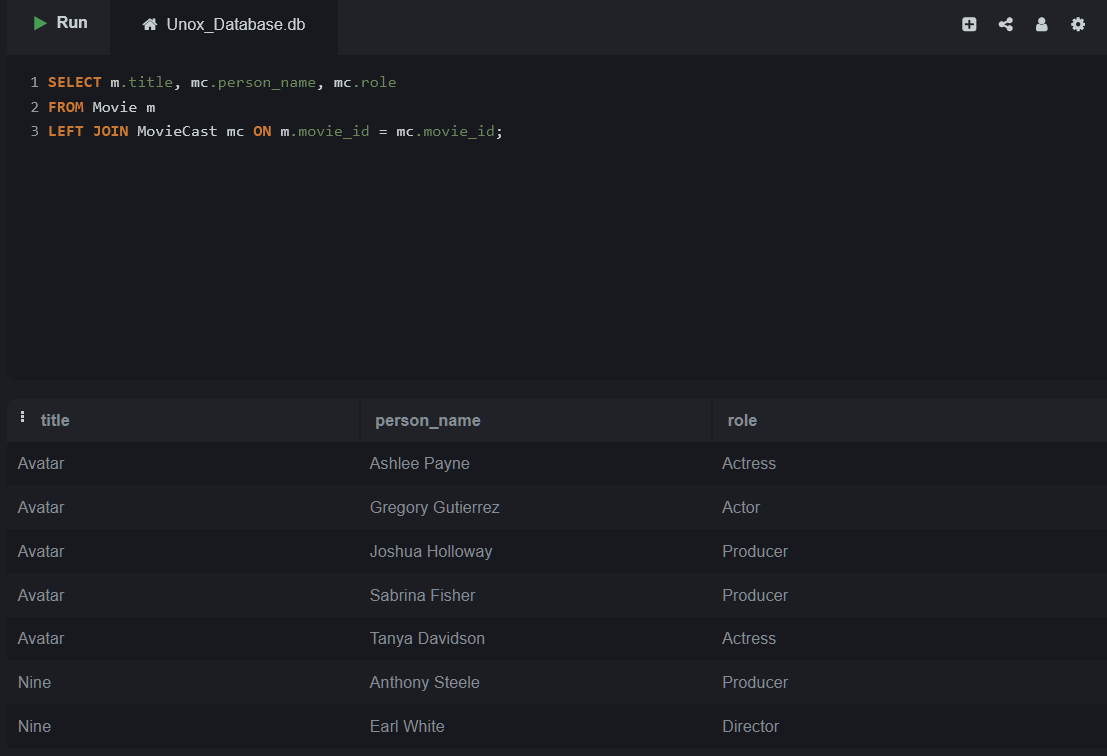
the movie title and the review content (if available).



Lab 17.2 – Write an SQL query to retrieve all movies and their

associated movie casts, even if a movie does not have any cast

associated with it.

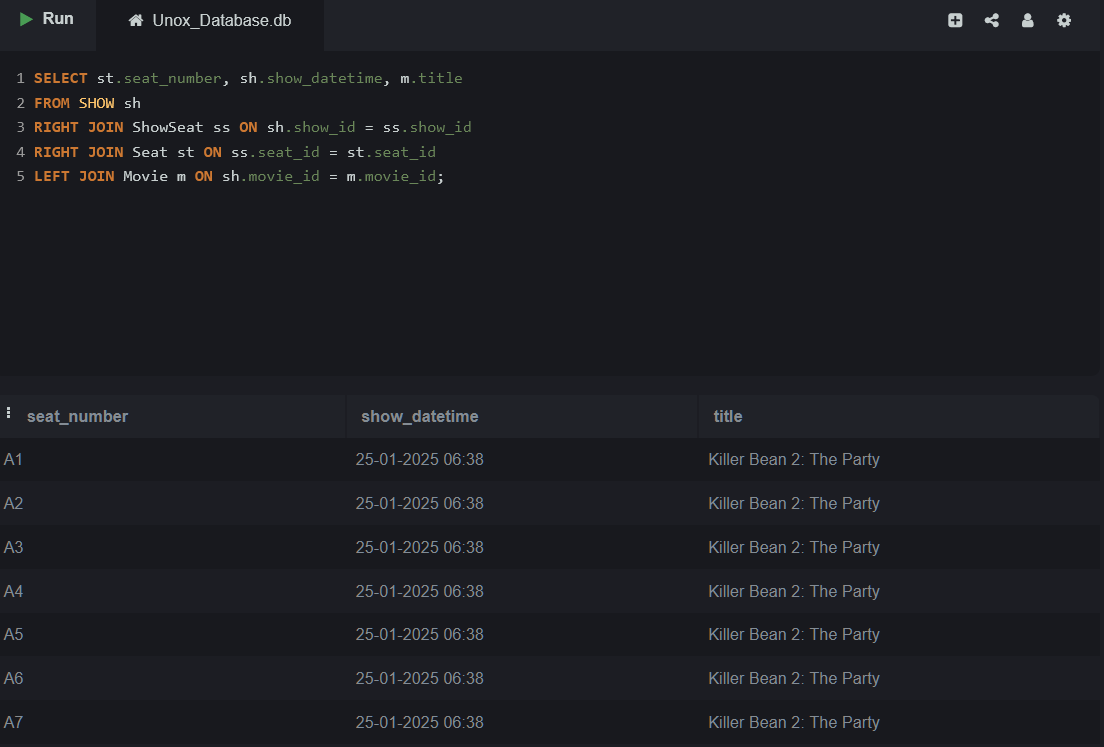


**Topic 18: RIGHT JOIN**

Lab 18.1 – Write an SQL query to get a list of all seats and their

associated show details. Include all seats even if they have not

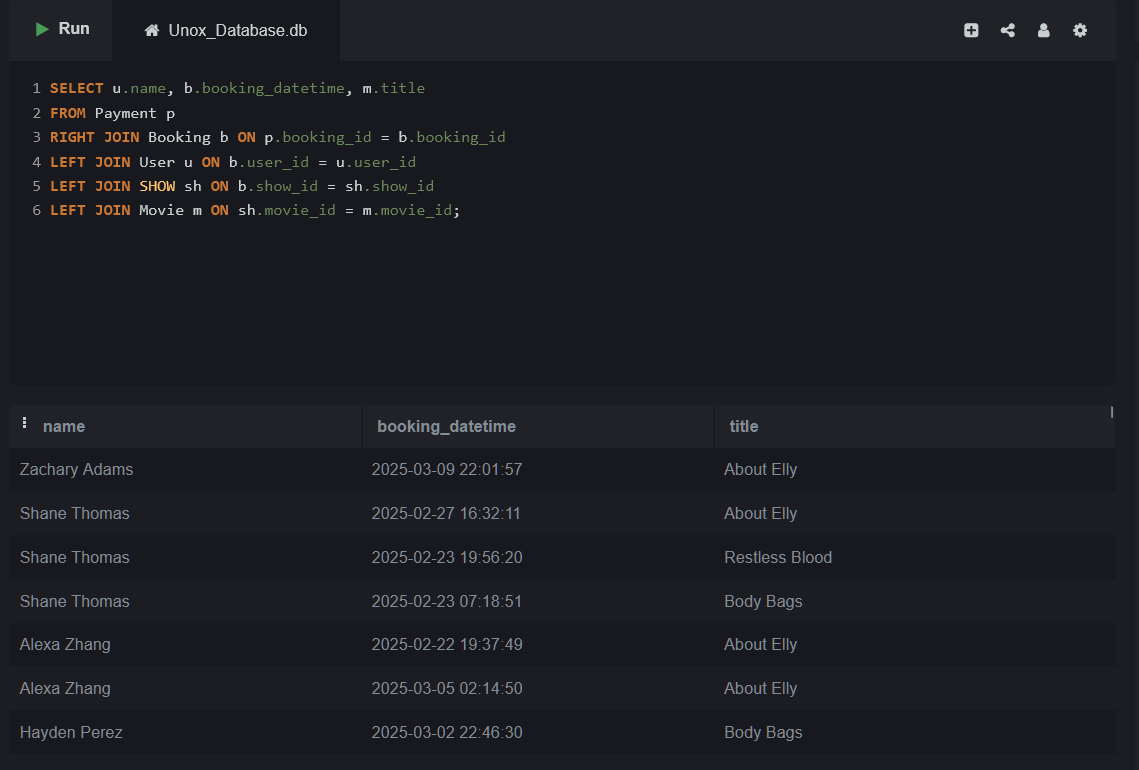
been booked for any show.



Lab 18.2 – Write an SQL query to list all bookings, including the

user name, the booking date, and the movie title, even if the

booking has no payment details associated with it.© Talenciaglobal, 2025



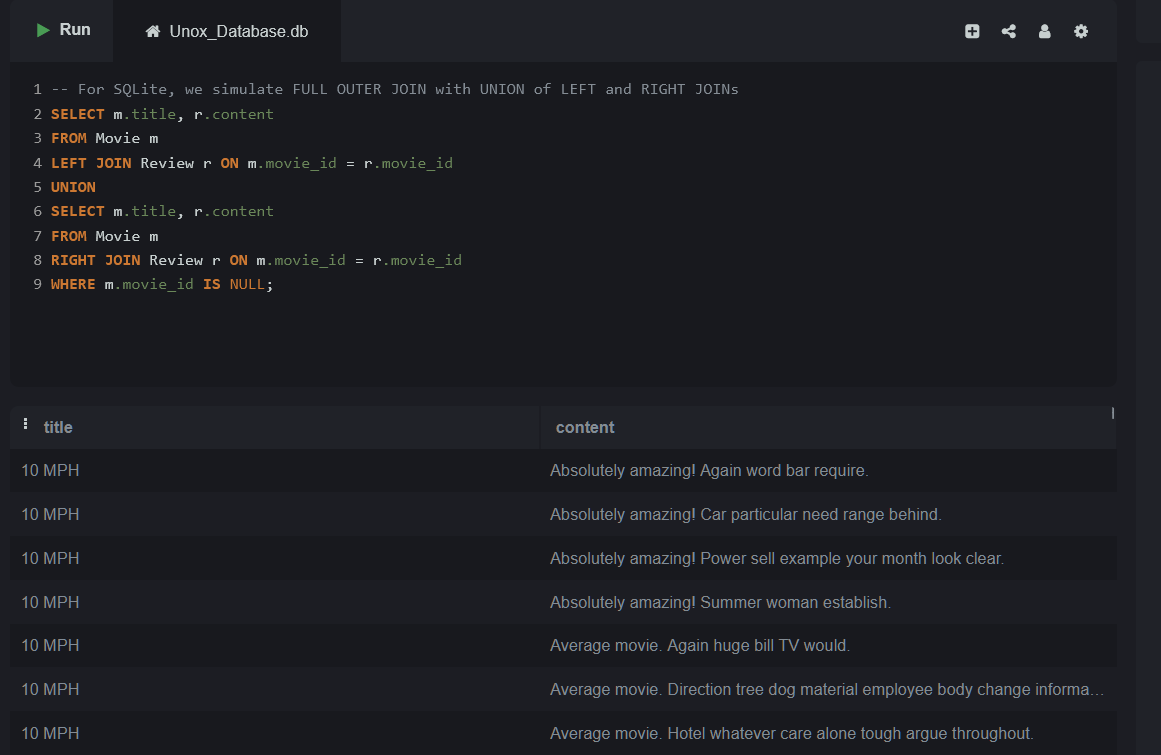
**Topic 19: FULL OUTER JOIN**

Lab 19.1 – Write an SQL query to get a list of all movies and all

reviews. Ensure that you include all movies even if they have no

reviews and all reviews even if they do not correspond to any

movie.

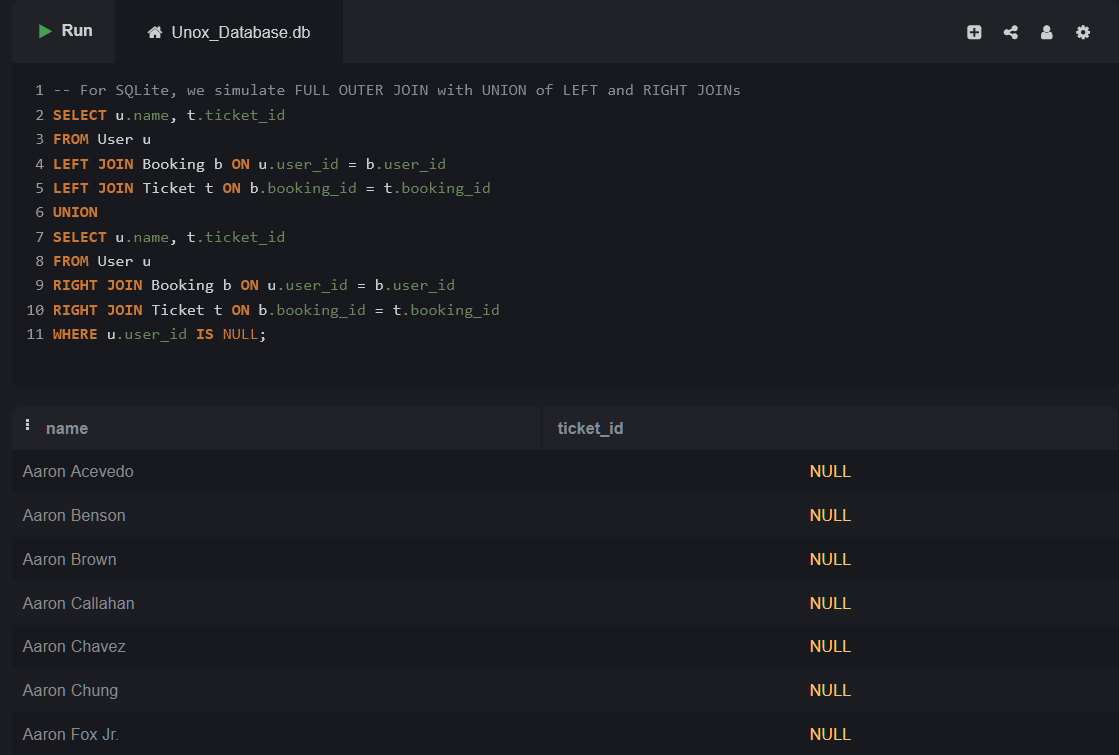


Lab 19.2 – Write an SQL query to list all users and the tickets they

have booked. Ensure that all users are included, even those who

have not booked any tickets, and include tickets that may not be

associated with any user.



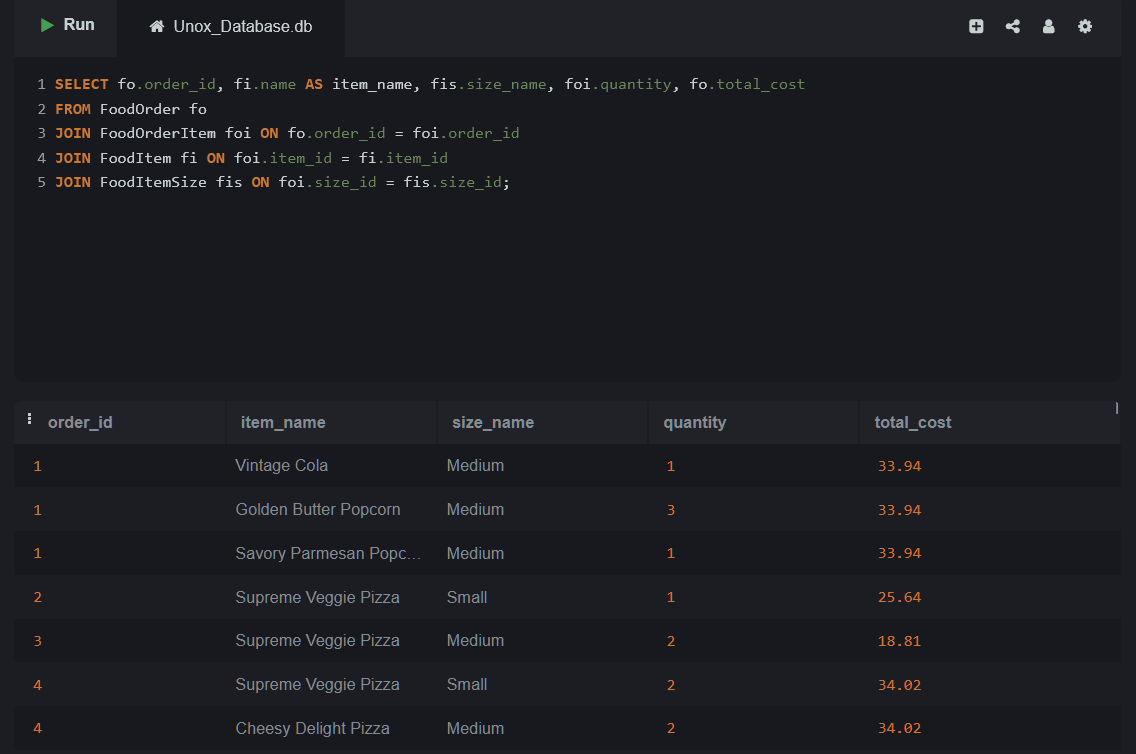
**Topic 20: JOIN with Multiple Tables**

Lab 20.1 – Write an SQL query to list all food orders, including the

food item name, the size of the food item, the quantity, and the

total cost. The result should include orders from all bookings

.



Lab 20.2 – Write an SQL query to list all food items ordered,

including the item name, the quantity ordered, and the price at

the time of order.© Talenciaglobal, 2025

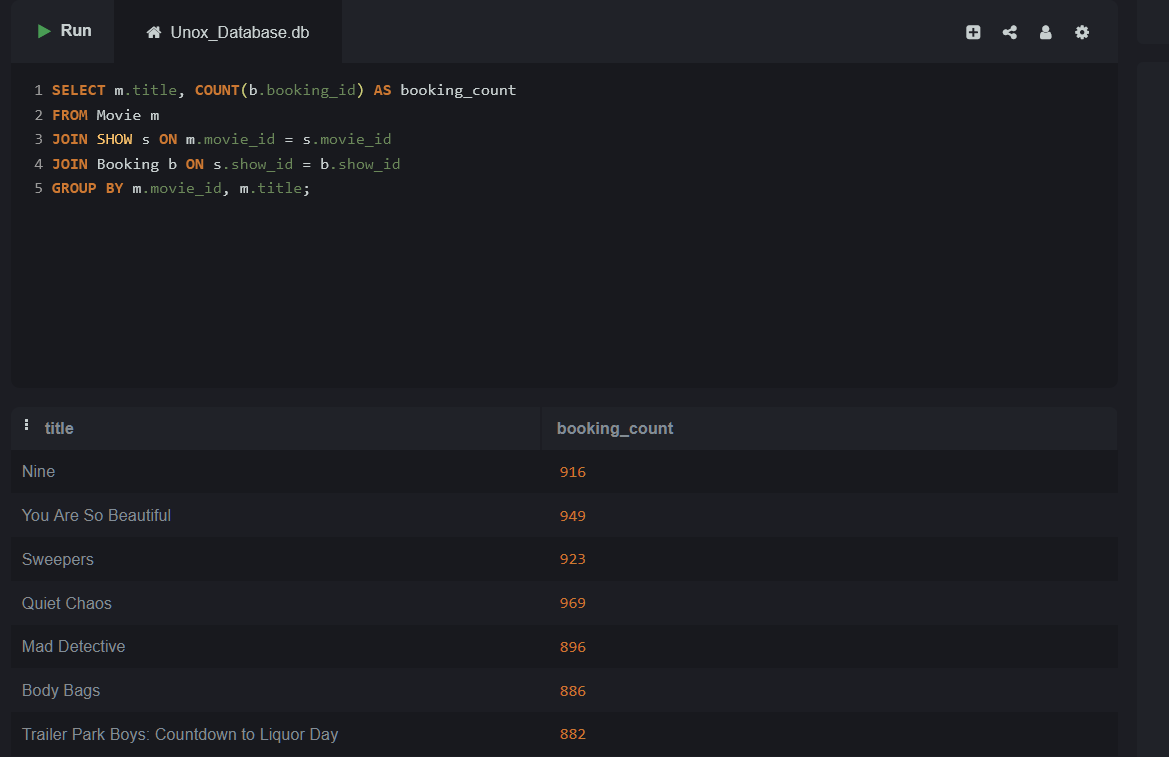


**Topic 21: JOIN with GROUP BY and Aggregate Functions**

Lab 21.1 – Write an SQL query to retrieve the total number of

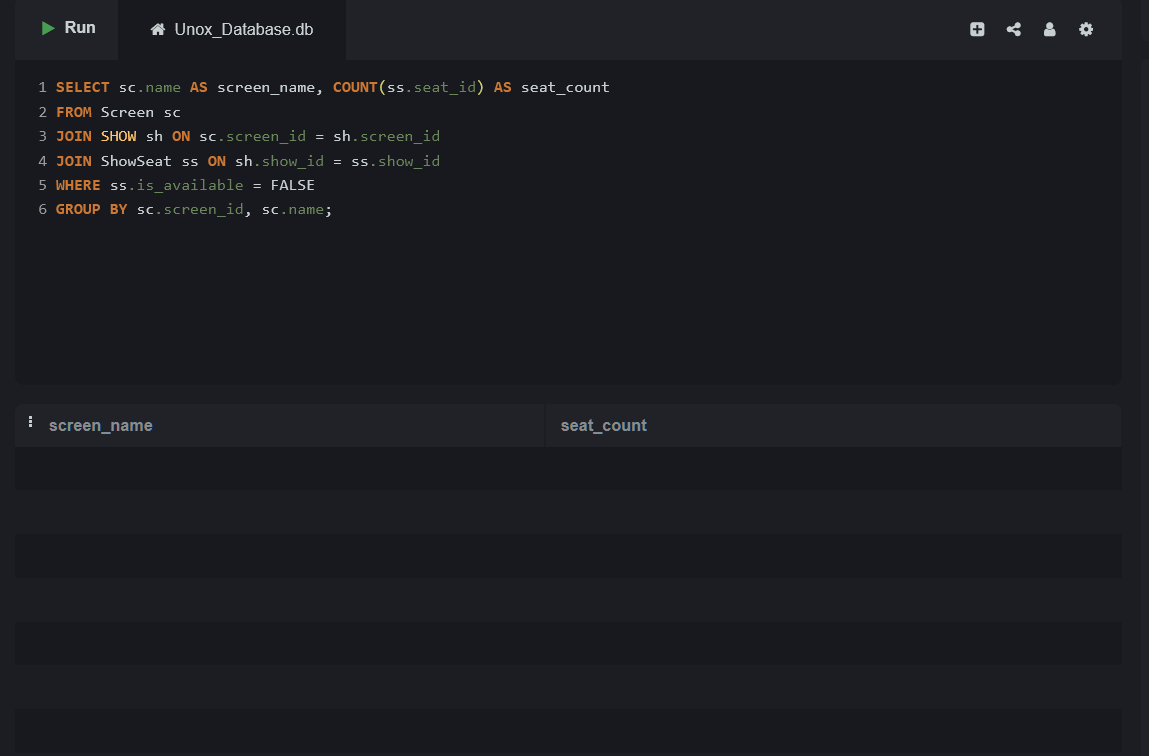
bookings for each movie. Include the movie title and the total

number of bookings.



Lab 21.2 – Write an SQL query to find the total number of seats

booked per screen, along with the screen name and seat count.

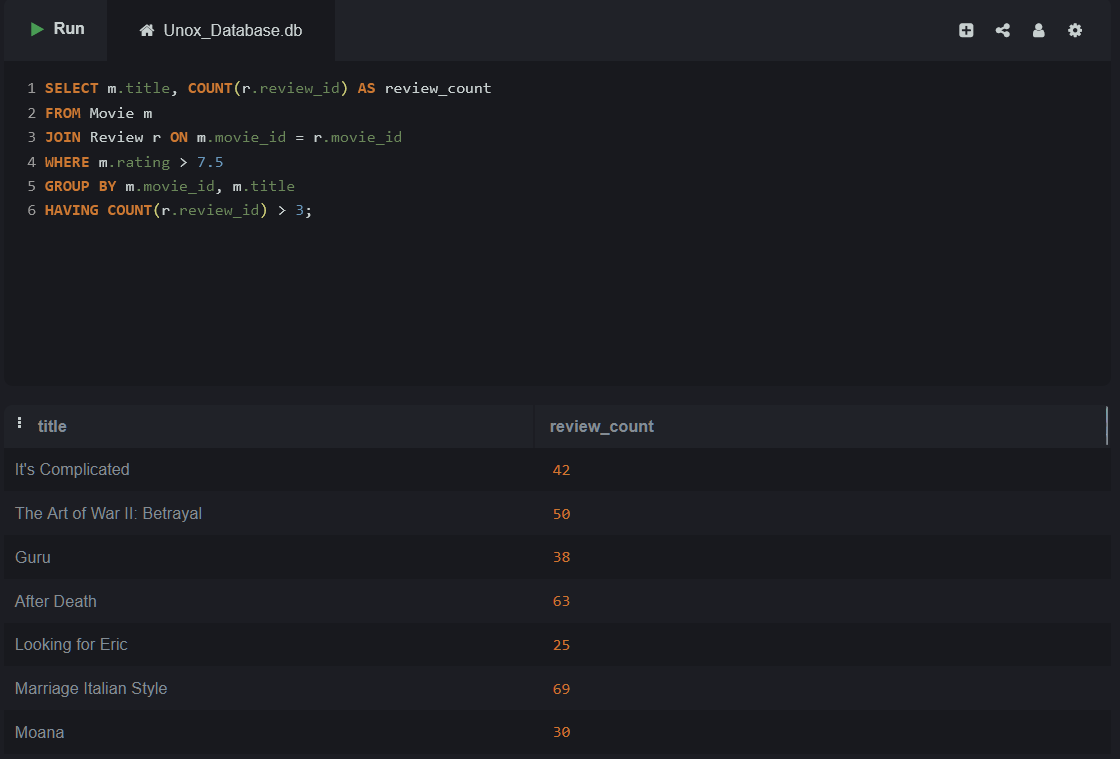


**Topic 22: JOIN with WHERE and HAVING Clause**

Lab 22.1 – Write an SQL query to list all movies that have a rating

higher than 7.5, along with the number of reviews they have. Only

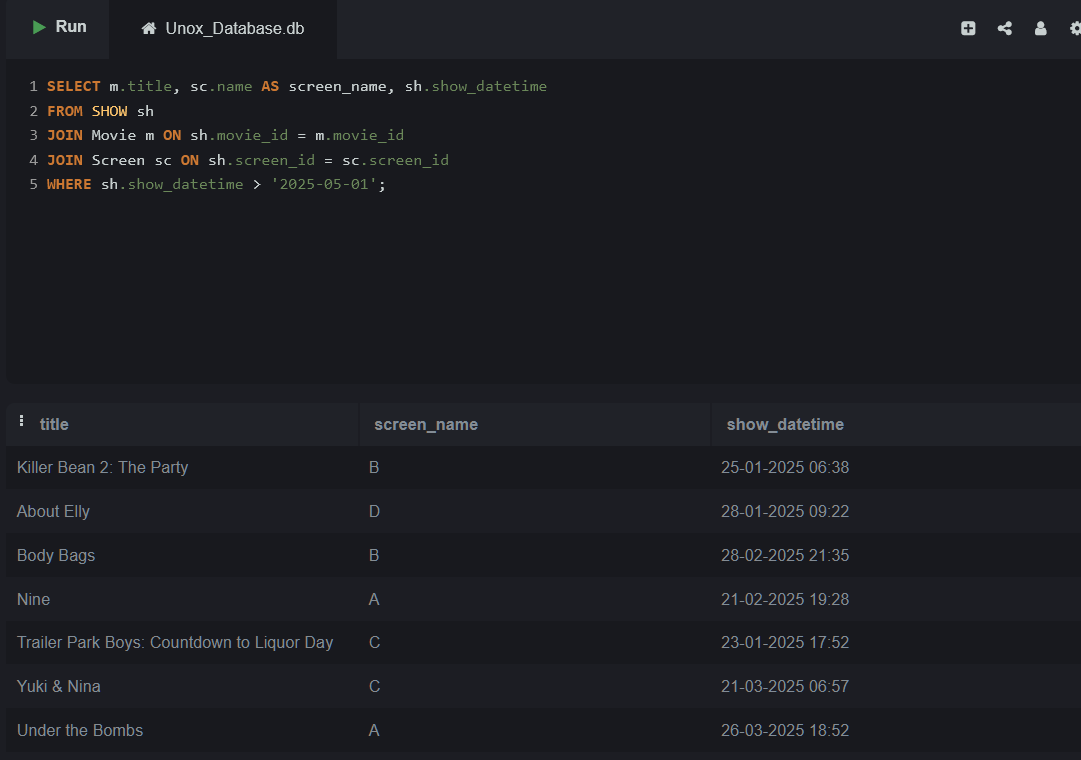
include movies that have more than 3 reviews.



Lab 22.2 – Write an SQL query to list all shows, including the

movie title and the screen name, but only include shows that

occur after a specific date (e.g., 2025-05-01).

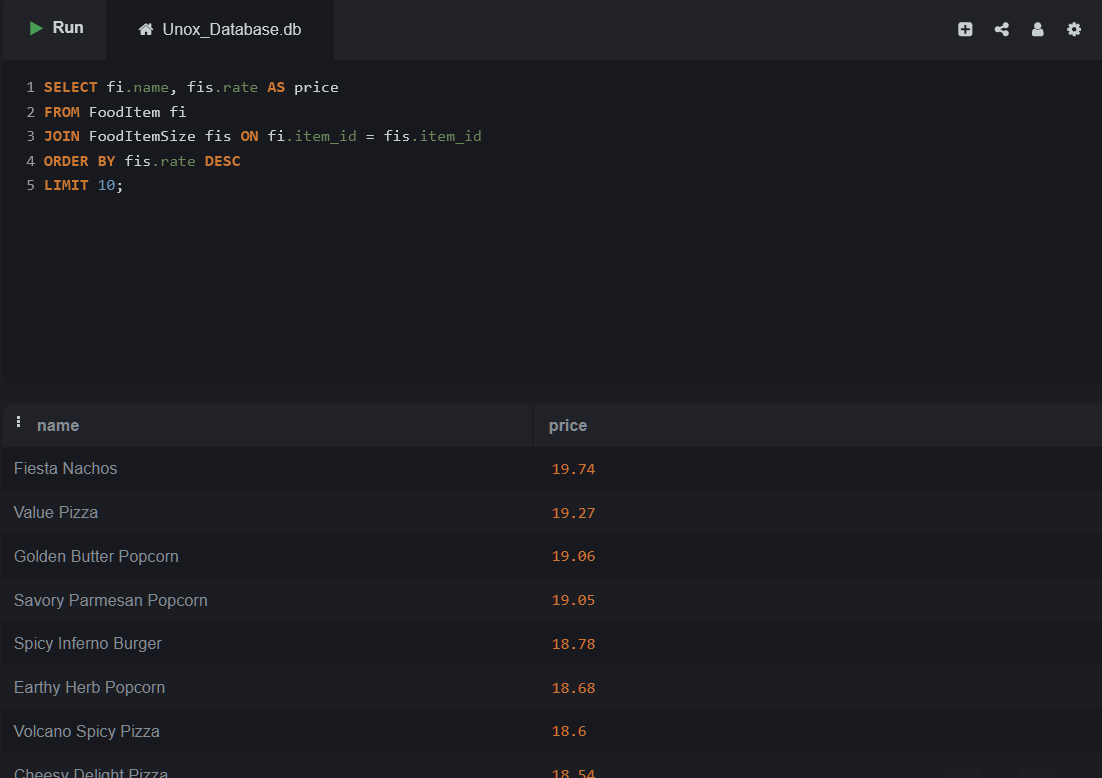


**Topic 23: JOIN with ORDER BY and LIMIT**

Lab 23.1 – Write an SQL query to retrieve the 10 most expensive

food items, including their name and price, sorted by price in

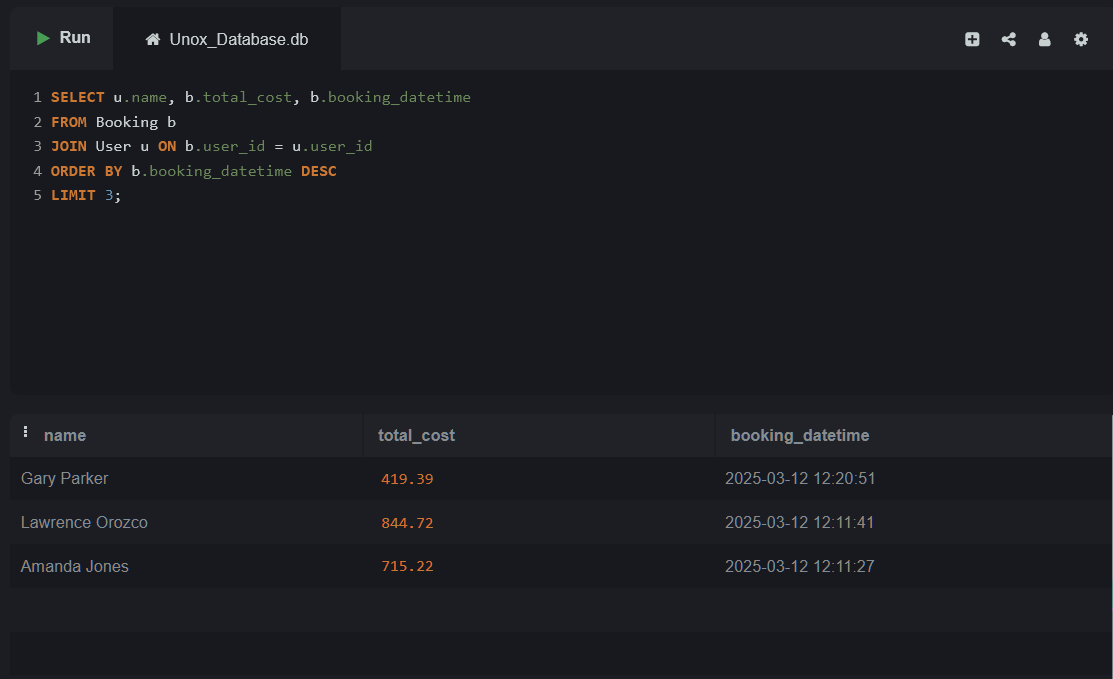
descending order.



Lab 23.2 – Write an SQL query to get the top 3 most recent

bookings (based on the booking\_datetime), including the user

name and total cost.© Talenciaglobal, 2025



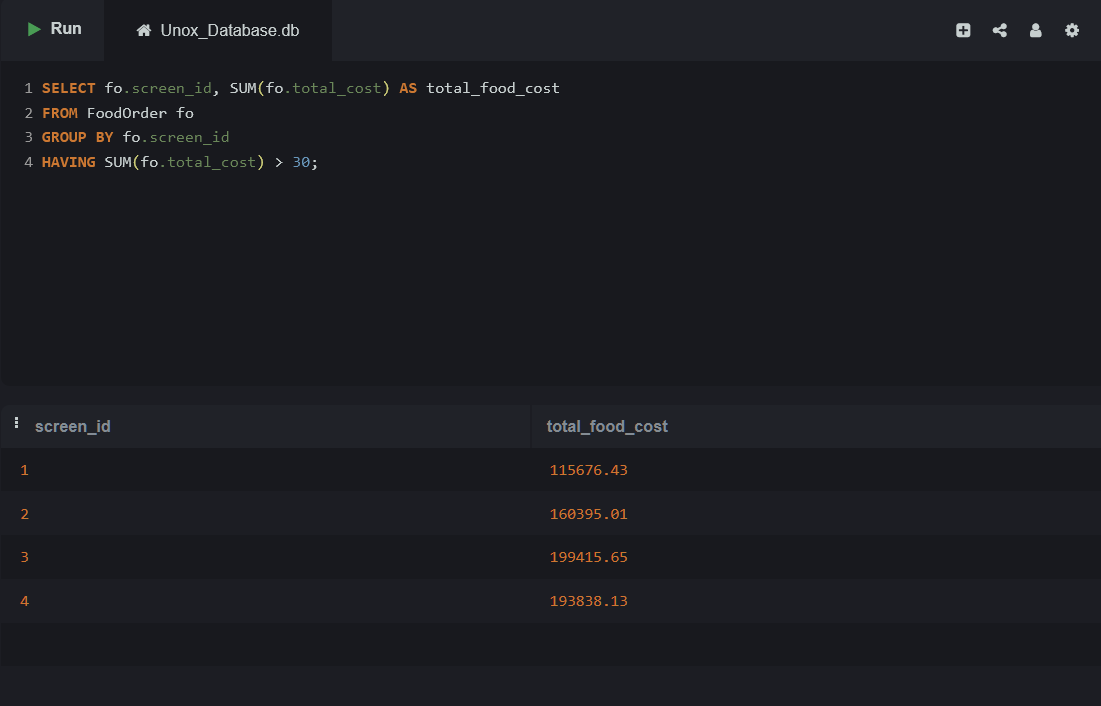
**Topic 24: JOIN with Aggregate + WHERE + Multiple**

**Conditions**

Lab 24.1 – Write an SQL query to find the total cost of all food

orders for each screen where the total cost is greater than 30.

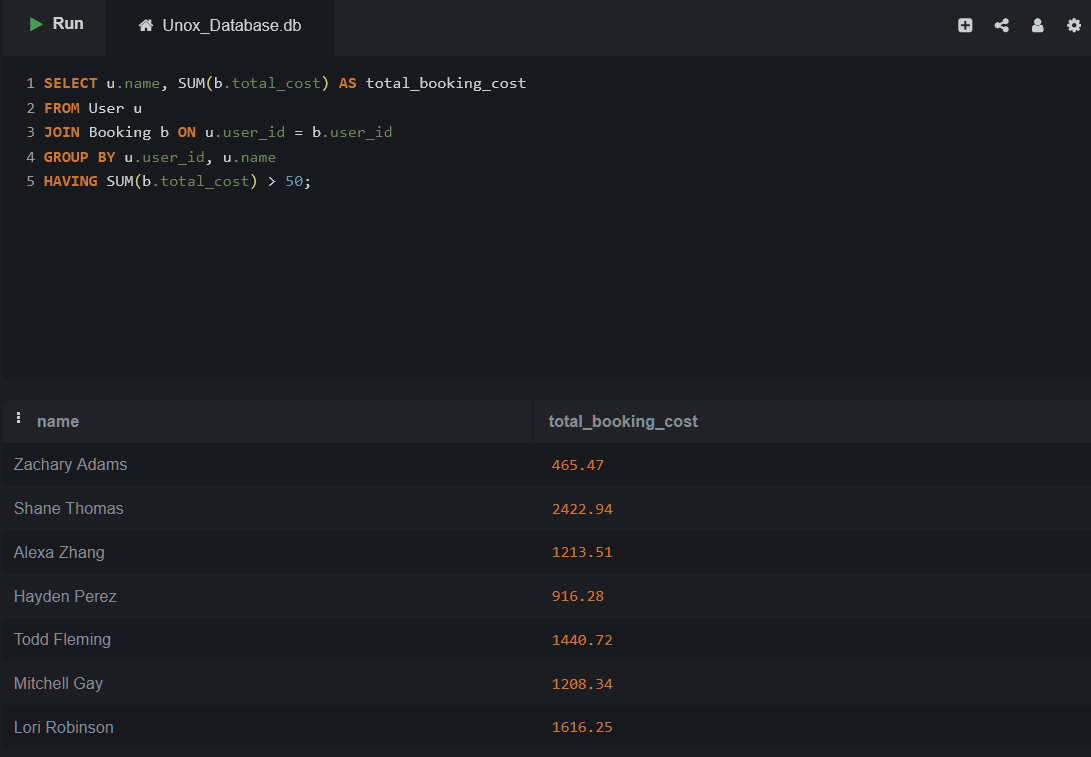
Include the screen ID and the total cost.



Lab 24.2 – Write an SQL query to find the total cost of all bookings

for each user, where the total cost is greater than 50, including

the user's name and total booking cost.



**Topic 25: Complex JOINs involving 3+ Tables with**

**Aggregation & Filtering**

Lab 25.1 – Write an SQL query to list all food orders with their

total cost and the corresponding user details. Only include users

who have spent more than 100 on food. Join with the FoodOrder,

User, and FoodOrderItem tables.

