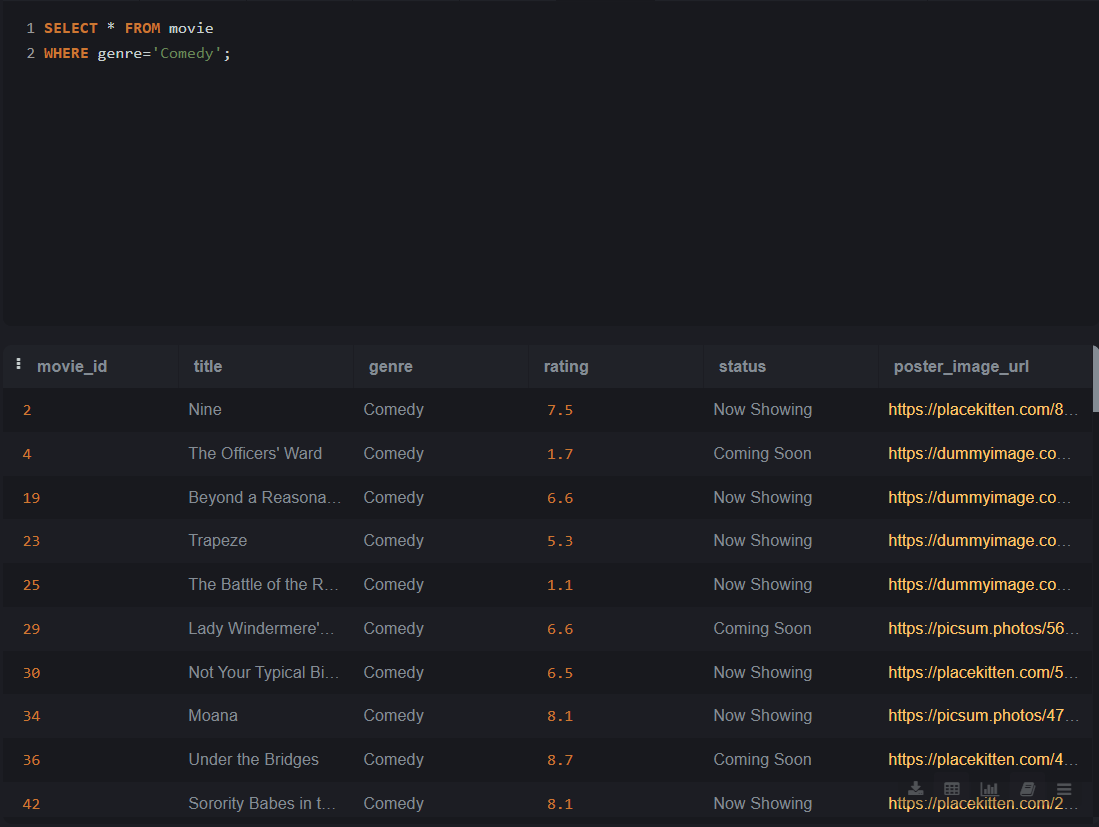
Topic 1: Basic SELECT with WHERE

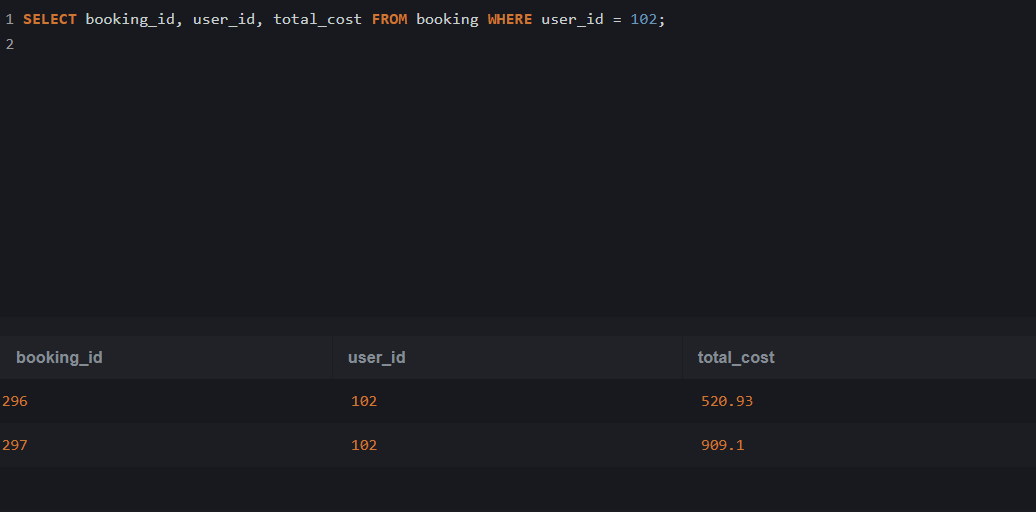
Lab 1.1: SELECT \* FROM Users WHERE name LIKE 'S%';



Lab 1.2: SELECT \* FROM Movies WHERE genre = 'Comedy';

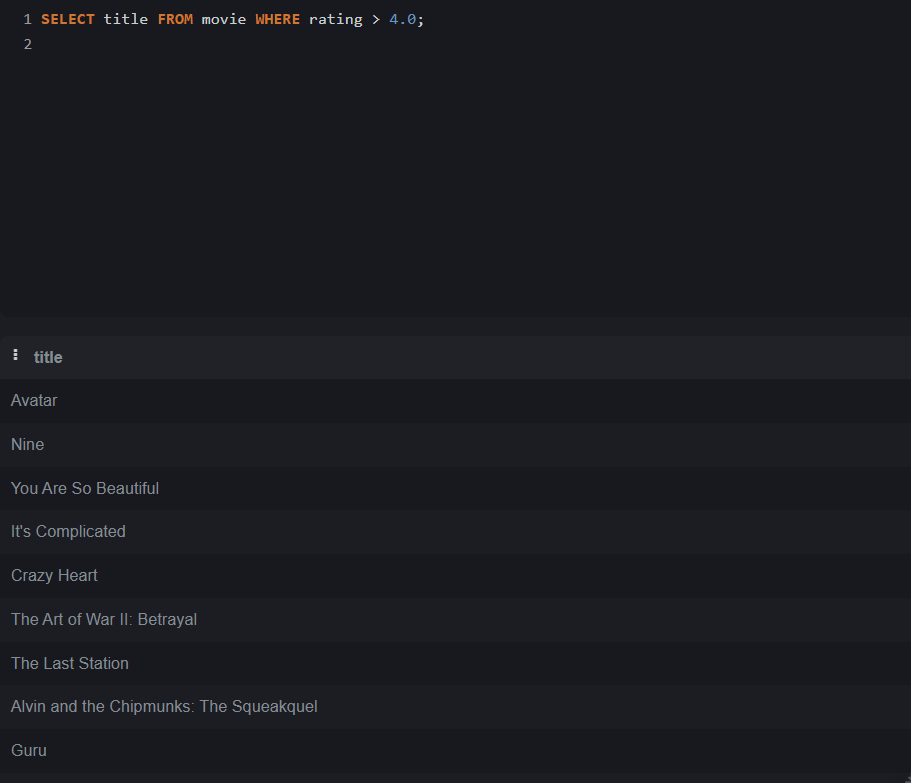


Lab 1.3: SELECT booking\_id, user\_id, total\_cost FROM Bookings WHERE user\_id = 102;

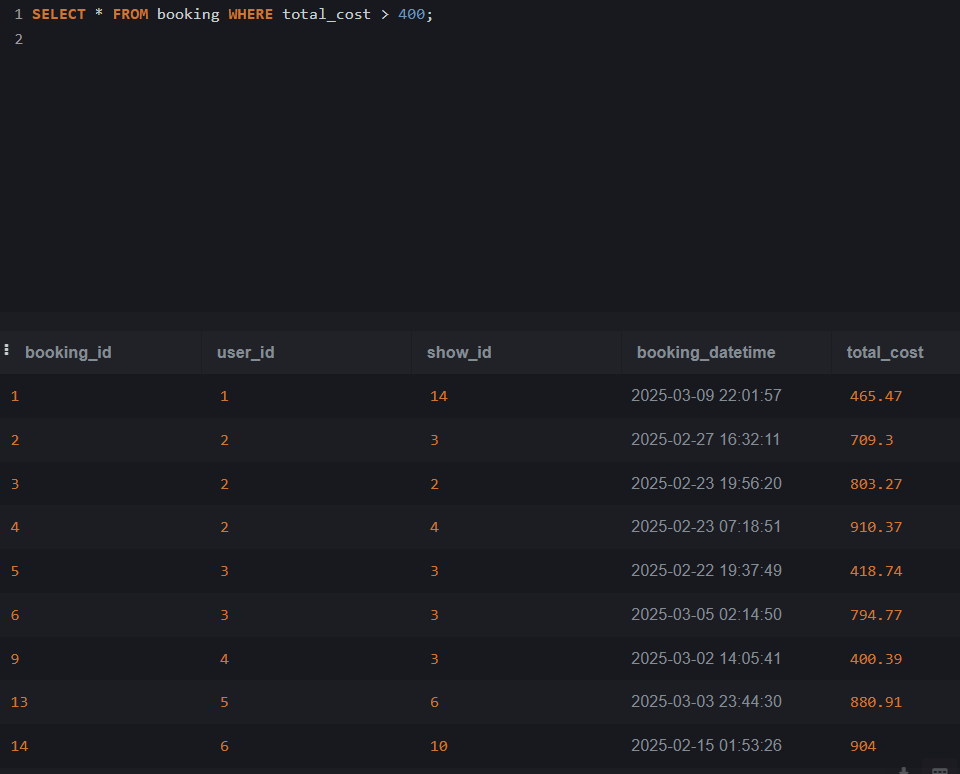


Topic 2: SELECT with WHERE using Operators

Lab 2.1: SELECT name FROM Movies WHERE rating > 4.0;

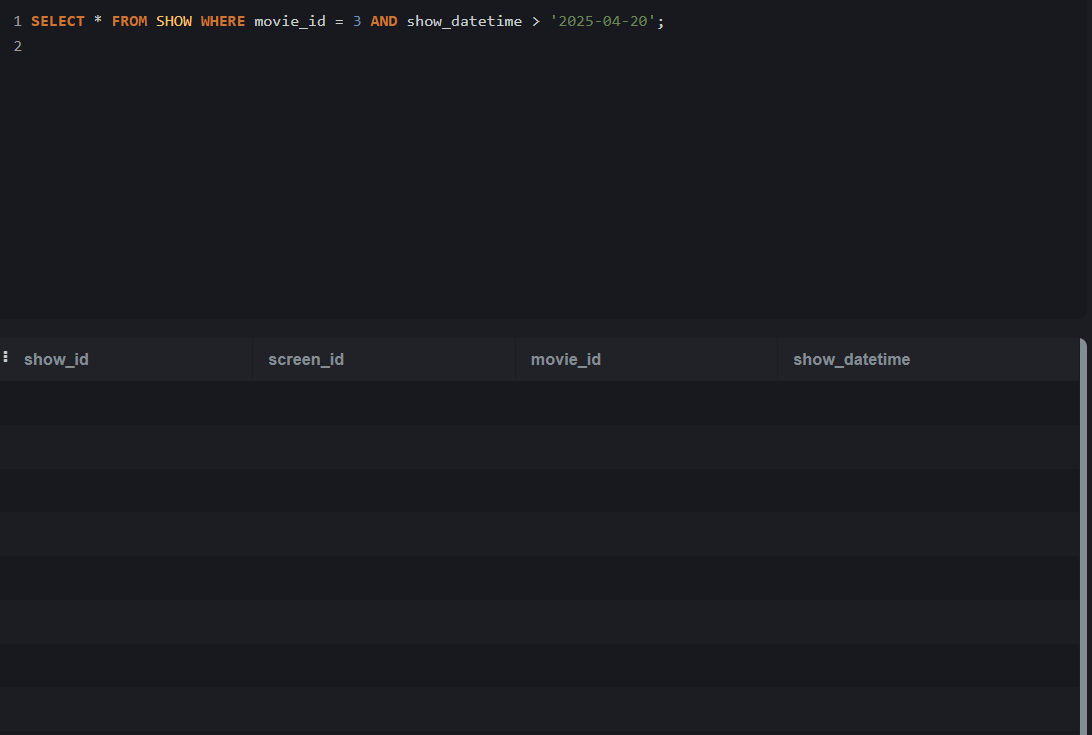


Lab 2.2: SELECT \* FROM Bookings WHERE total\_cost> 400;

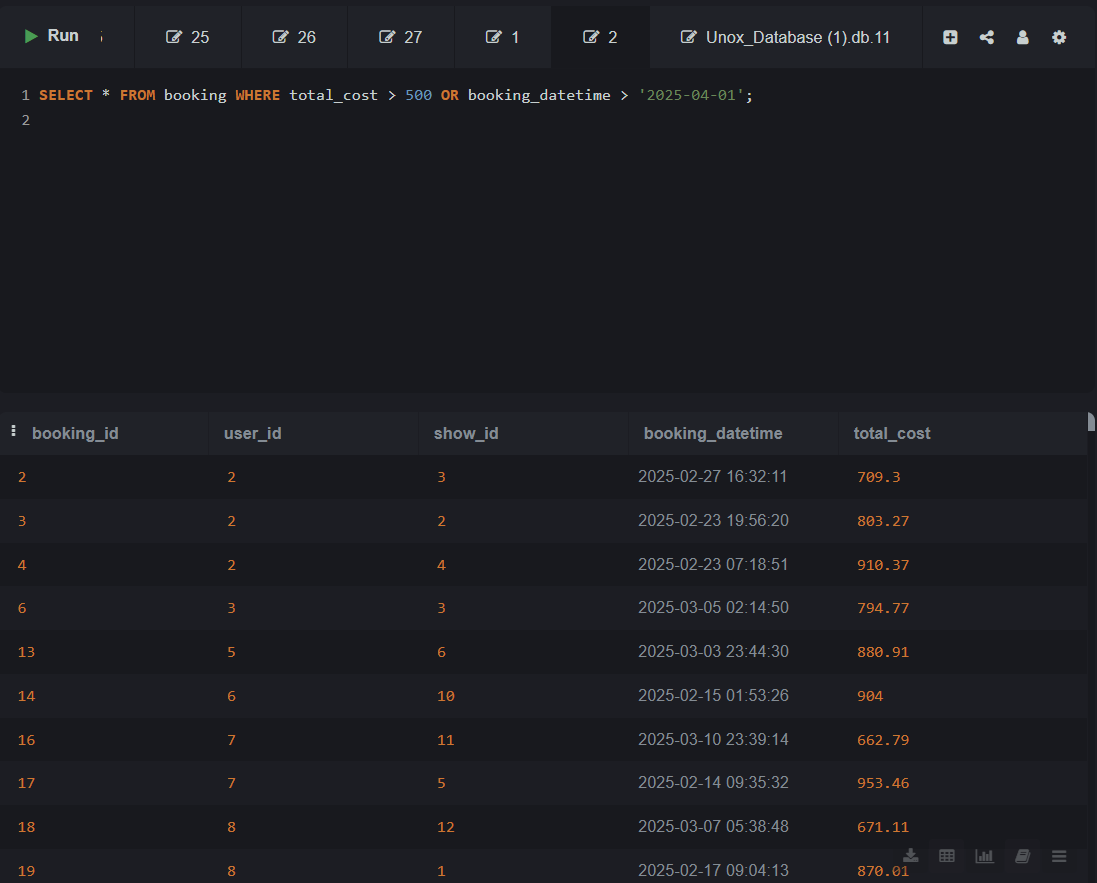


Topic 3: SELECT with WHERE using AND/OR

Lab 3.1: SELECT \* FROM Show WHERE movie\_id = 3 AND show\_datetime > '2025-04-20';

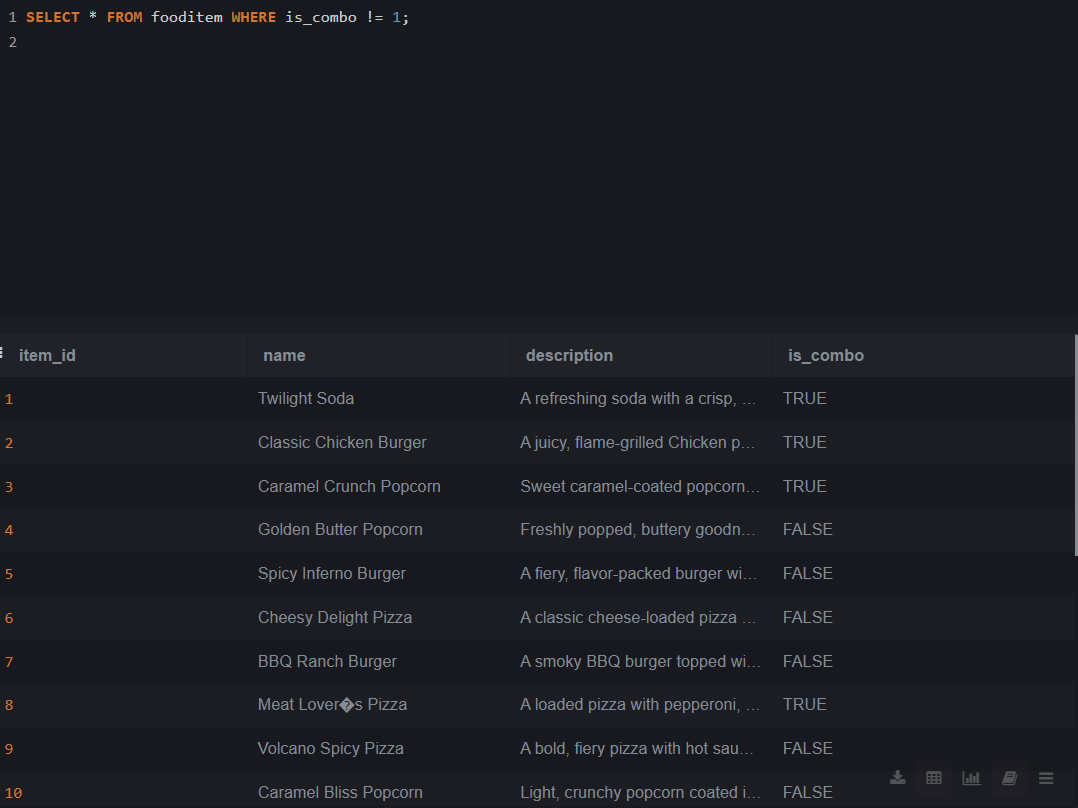


Lab 3.2: SELECT \* FROM booking WHERE total\_cost > 500 OR booking\_datetime > '2025-04-01';



Topic 4: SELECT with WHERE and NOT

Lab 4.1: SELECT \* FROM FoodItems WHERE is\_combo != 1;

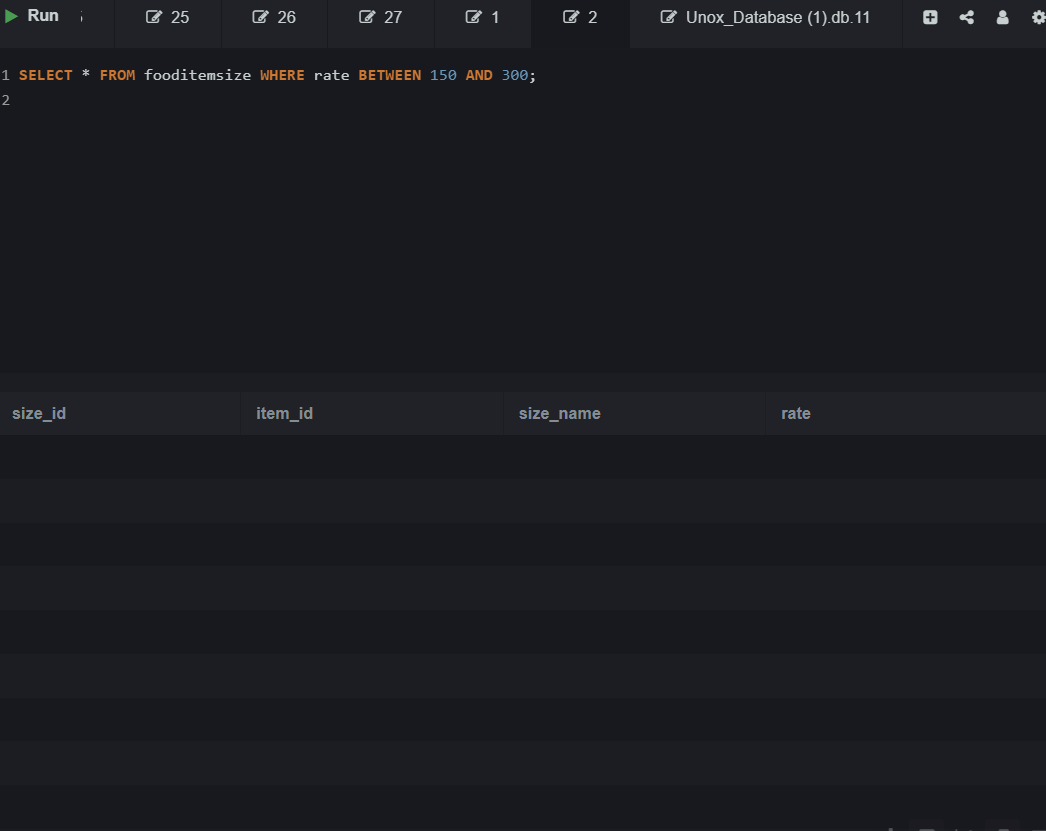


Lab 4.2: SELECT \* FROM movie WHERE status != 'Inactive';

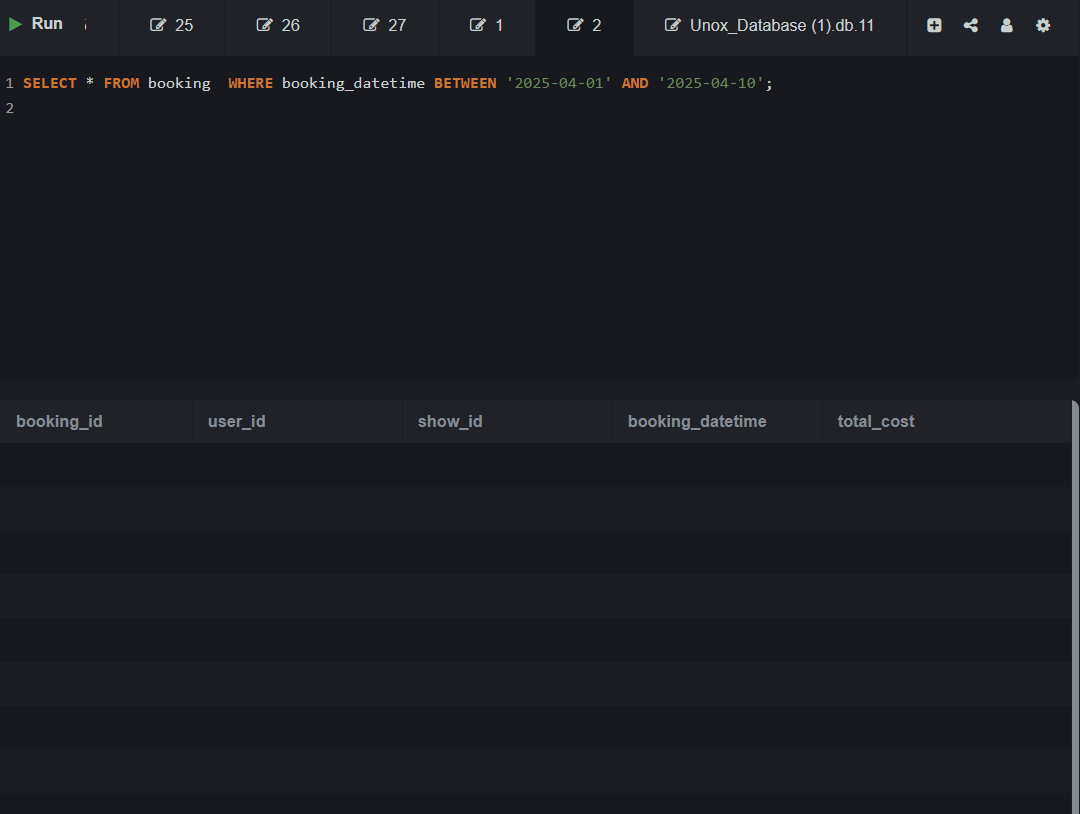


Topic 5: SELECT with WHERE and BETWEEN

Lab 5.1: SELECT \* FROM fooditemsize WHERE rate BETWEEN 150 AND 300;

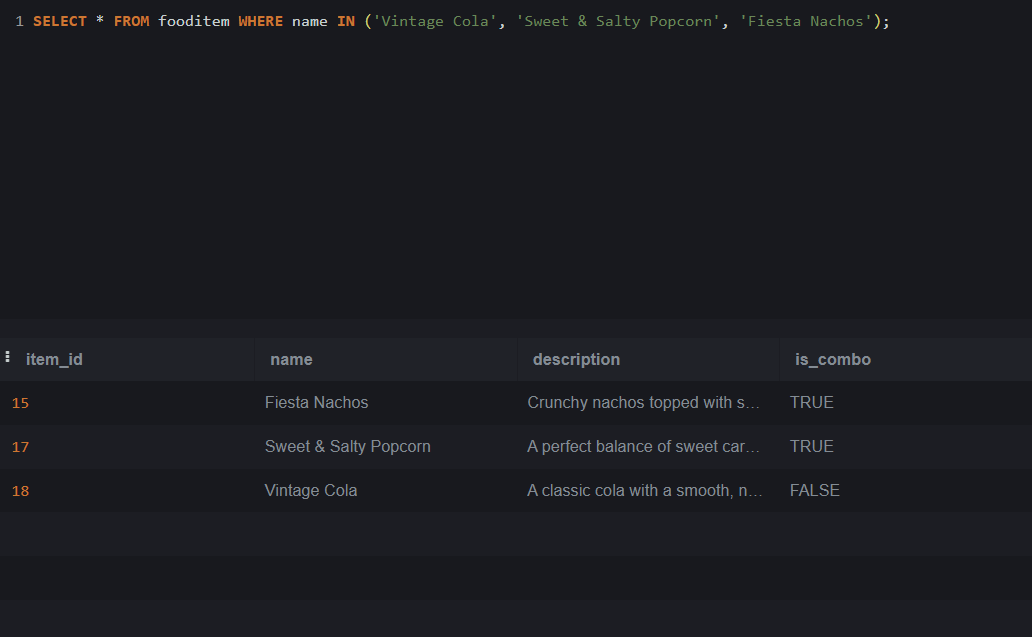


Lab 5.2: SELECT \* FROM TicketBookings WHERE booking\_date BETWEEN '2025-04-01' AND '2025-04-10';

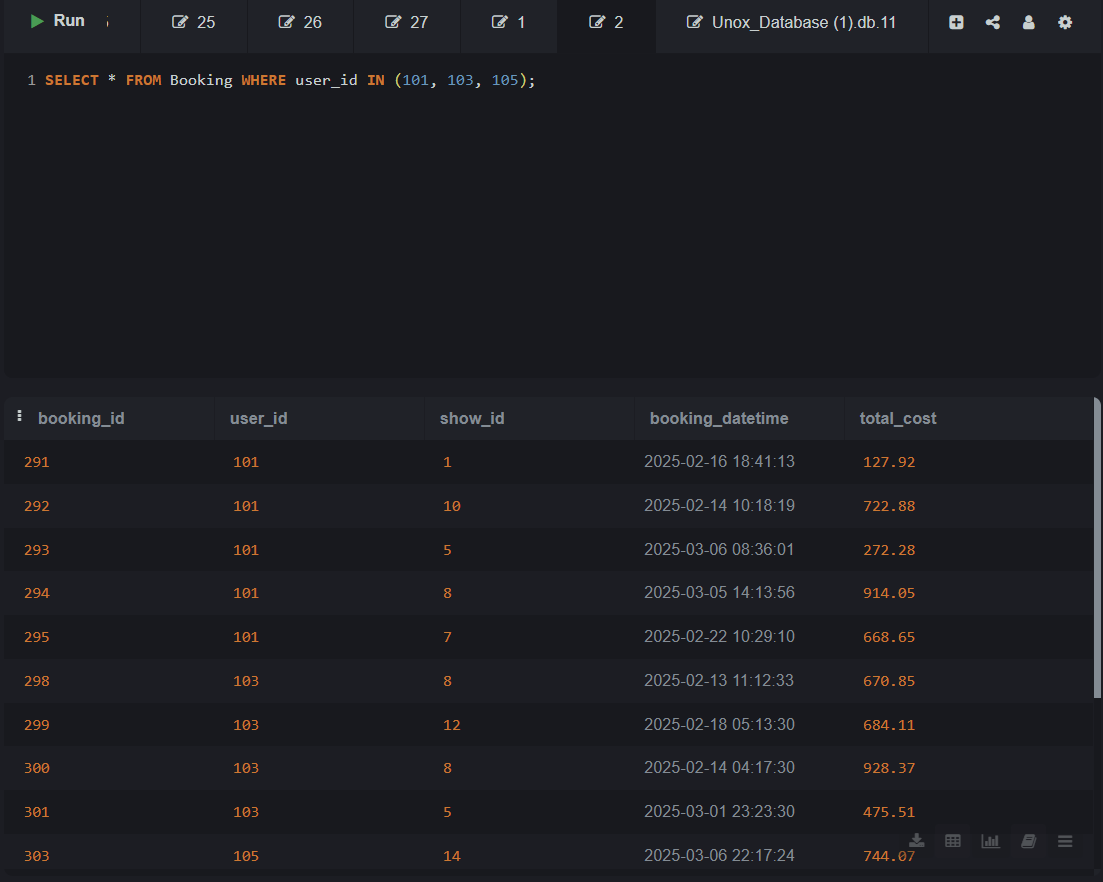


Topic 6: SELECT with WHERE and IN

Lab 6.1: SELECT \* FROM fooditem WHERE name IN ('Vintage Cola', 'Sweet & Salty Popcorn', 'Fiesta Nachos');

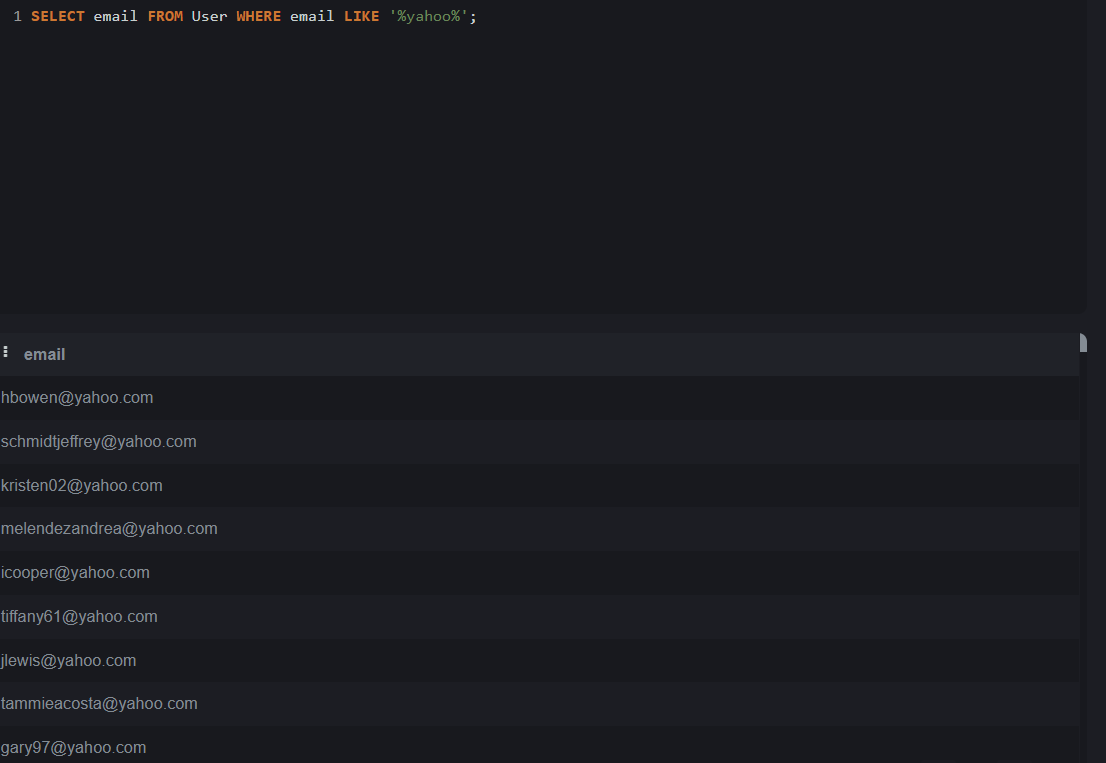


Lab 6.2: SELECT \* FROM Bookings WHERE user\_id IN (101, 103, 105);



Topic 7: SELECT with WHERE and LIKE

Lab 7.1: SELECT email FROM Users WHERE email LIKE '%yahoo%';

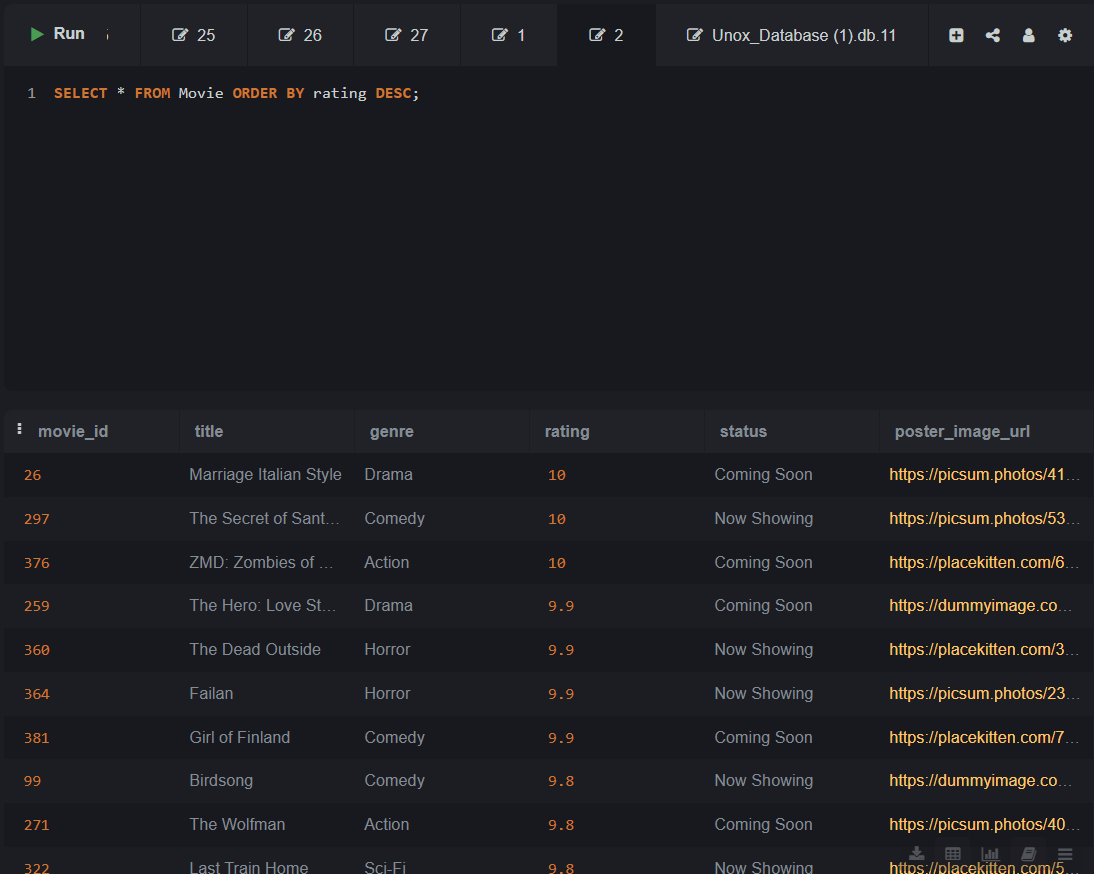


Lab 7.2: SELECT \* FROM Movies WHERE title LIKE 'T%';

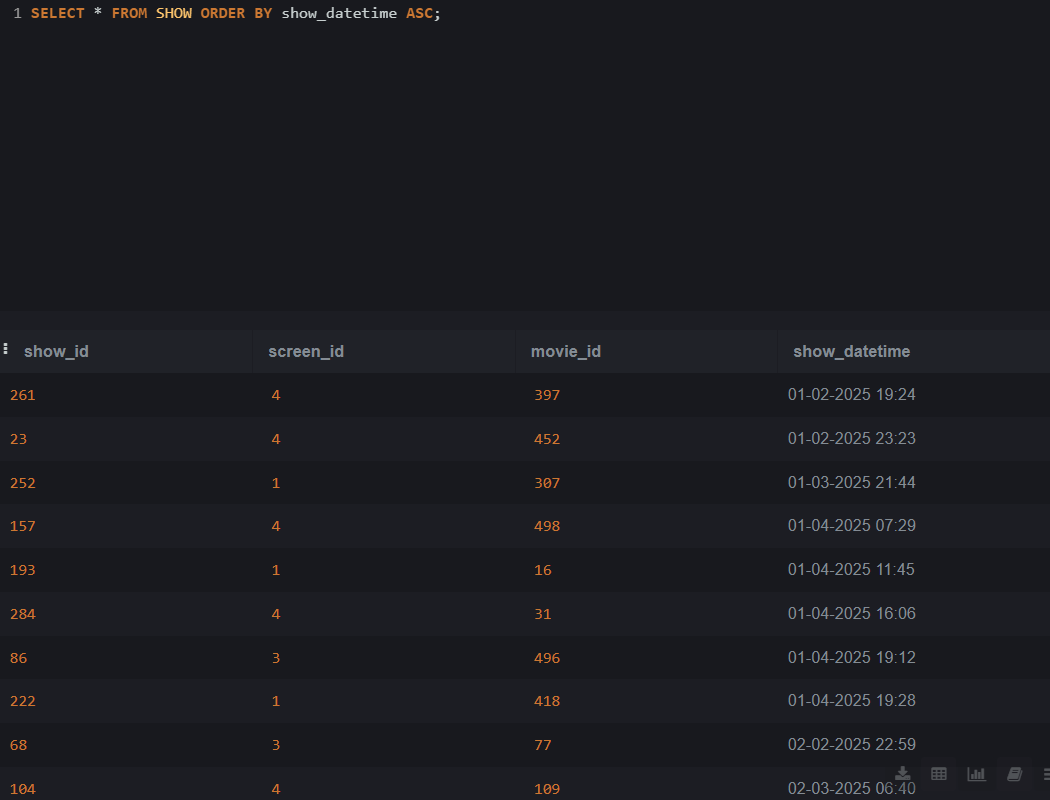


Topic 8: SELECT with ORDER BY

Lab 8.1: SELECT \* FROM Movies ORDER BY rating DESC;



Lab 8.2: SELECT \* FROM Shows ORDER BY show\_datetime ASC;

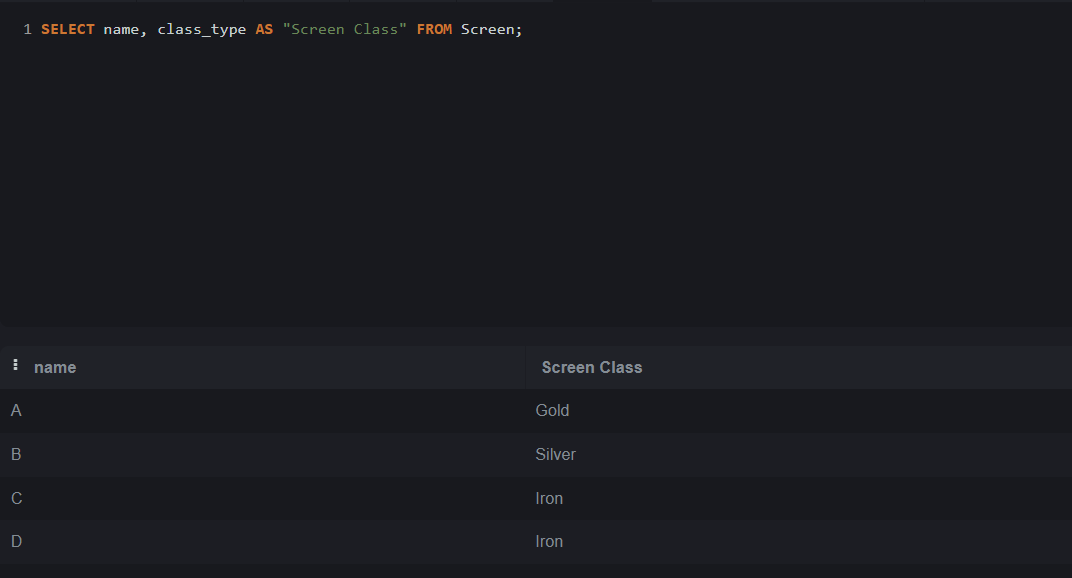


Topic 9: SELECT with ALIAS

Lab 9.1: SELECT name, rating AS "Movie Rating" FROM Movies;



Lab 9.2: SELECT screen\_name, class\_type AS "Screen Class" FROM Screens;



Topic 10: SELECT with LIMIT and OFFSET

Lab 10.1: SELECT \* FROM FoodItem LIMIT 5;

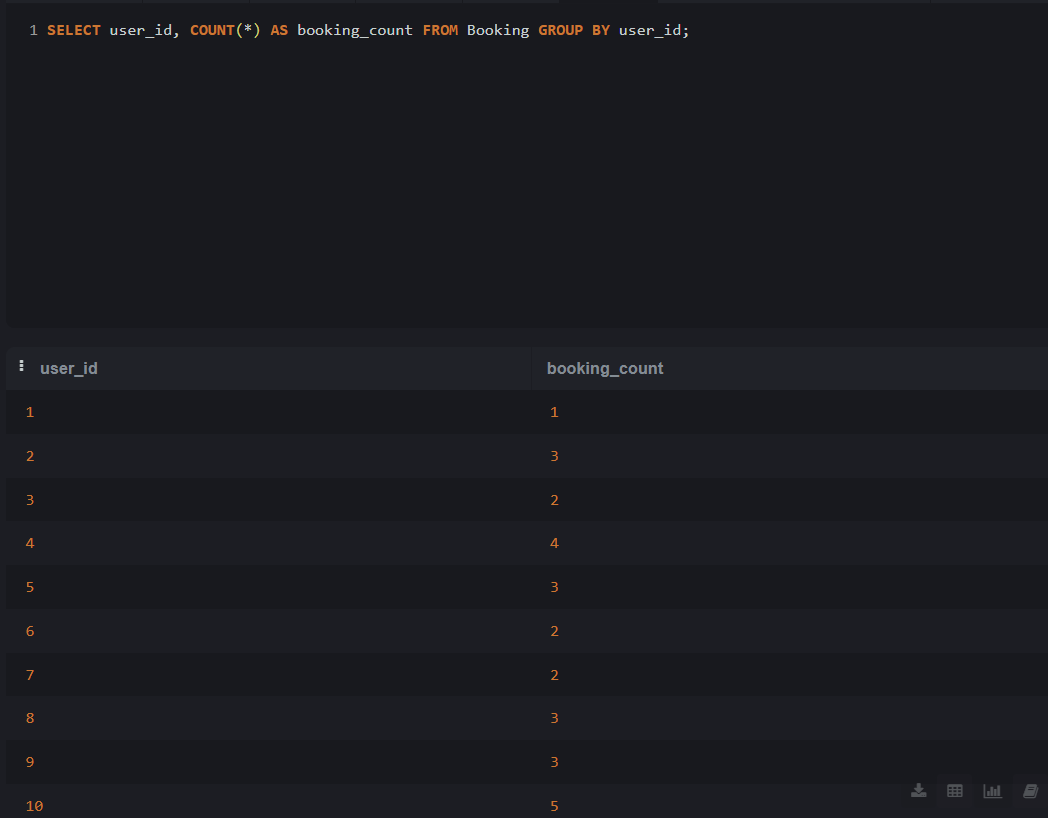


Lab 10.2: SELECT \* FROM Bookings ORDER BY booking\_datetime DESC LIMIT 5 OFFSET 3;

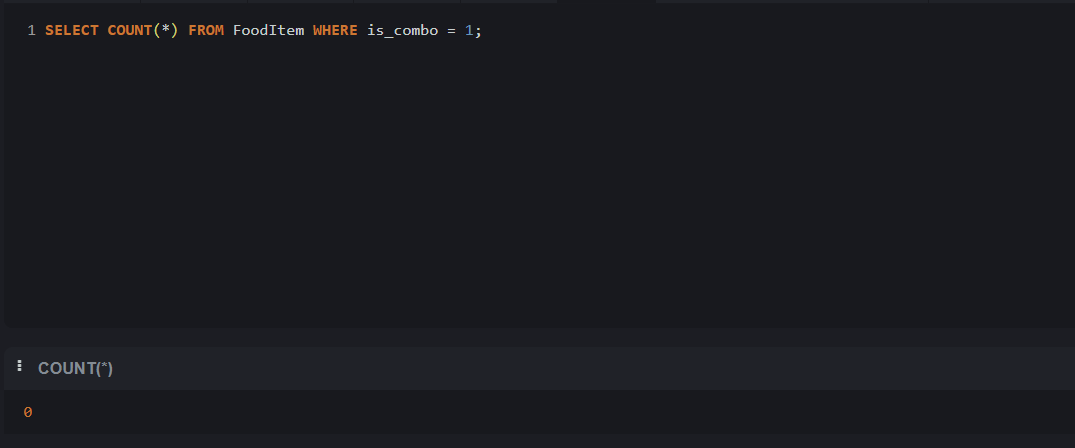


Topic 11: SELECT with Aggregate Functions

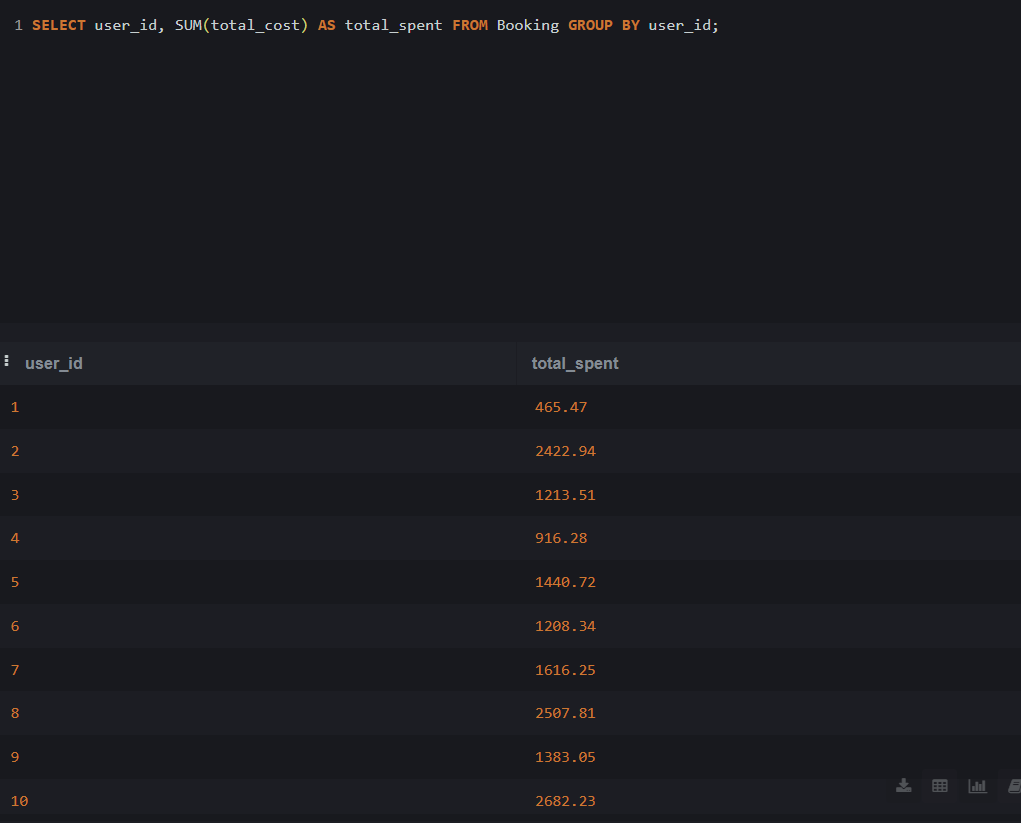
Lab 11.1.1: SELECT user\_id, COUNT(\*) AS booking\_count FROM Bookings GROUP BY user\_id;



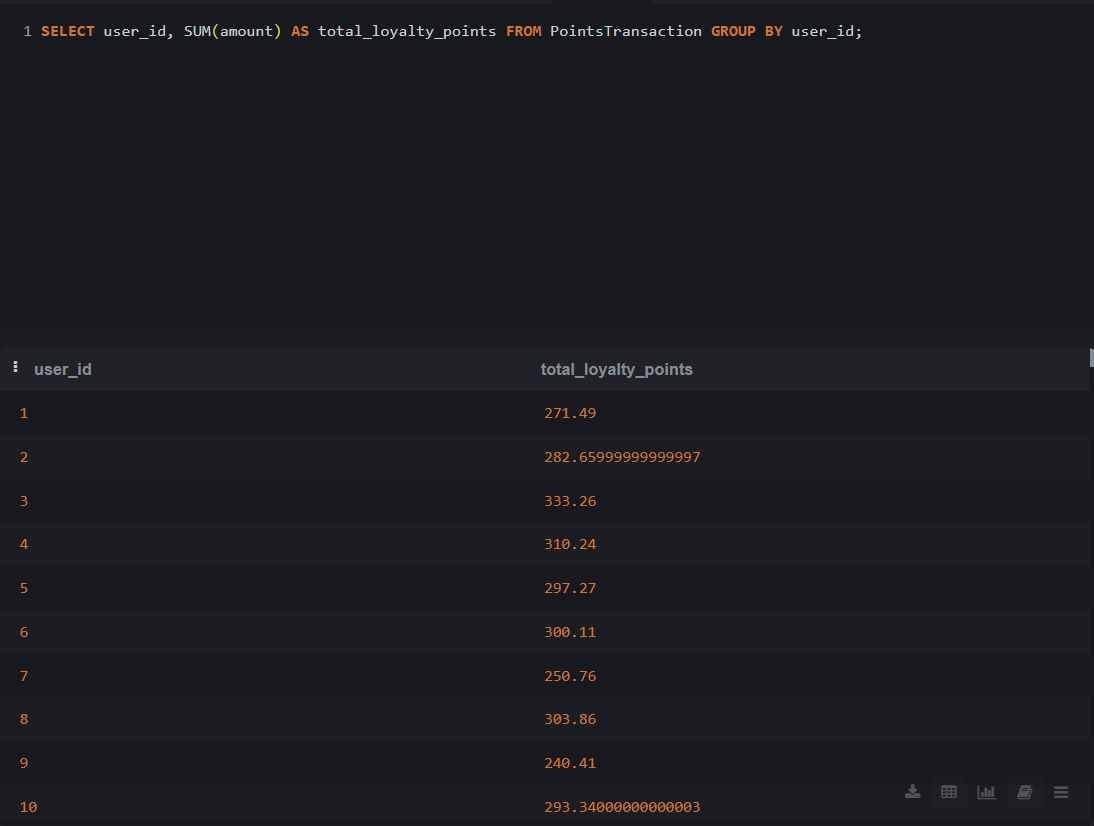
Lab 11.1.2: SELECT COUNT(\*) FROM FoodItems WHERE is\_combo = 1;



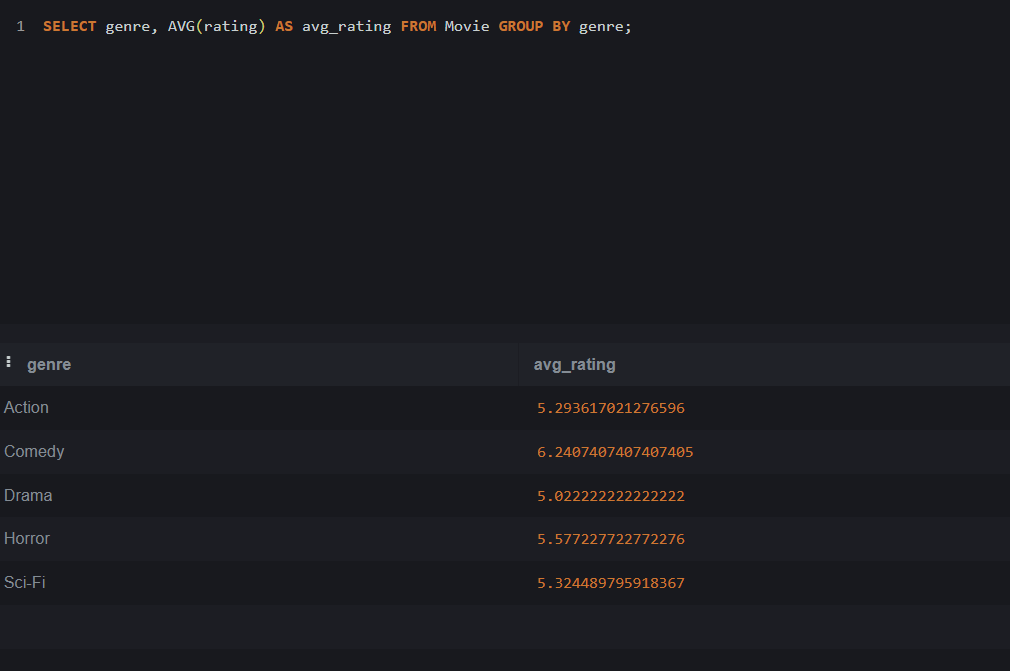
Lab 11.2.1: SELECT user\_id, SUM(total\_amount) AS total\_spent FROM Bookings GROUP BY user\_id;



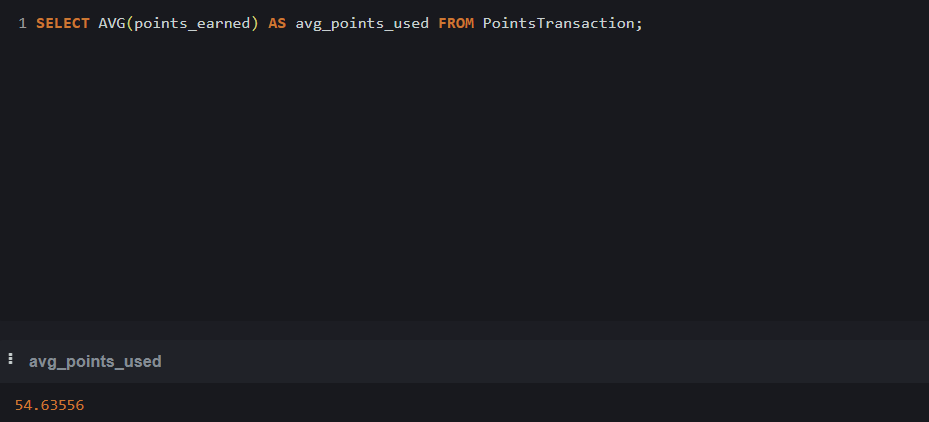
Lab 11.2.2: SELECT user\_id, SUM(amount) AS total\_loyalty\_points FROM LoyaltyPointsTransactions GROUP BY user\_id;



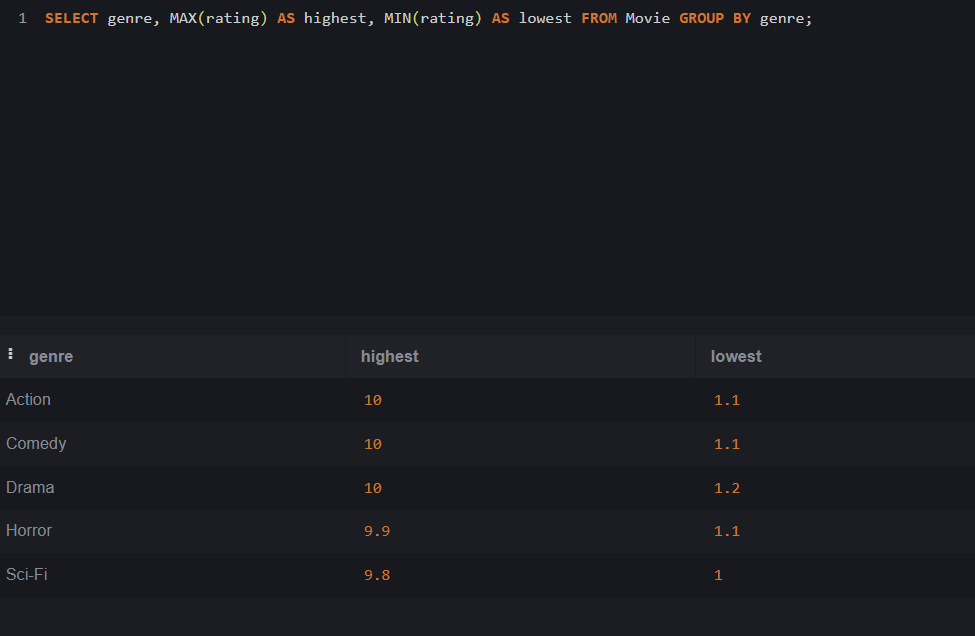
Lab 11.3.1: SELECT genre, AVG(rating) AS avg\_rating FROM Movies GROUP BY genre;



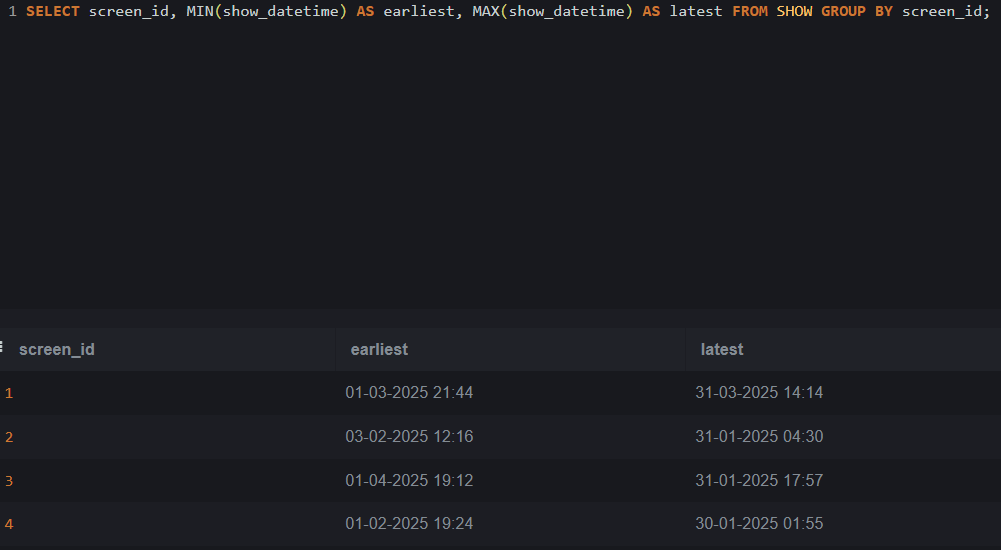
Lab 11.3.2: SELECT AVG(points\_used) AS avg\_points\_used FROM LoyaltyPointsTransactions;



Lab 11.4.1: SELECT genre, MAX(rating) AS highest, MIN(rating) AS lowest FROM Movies GROUP BY genre;

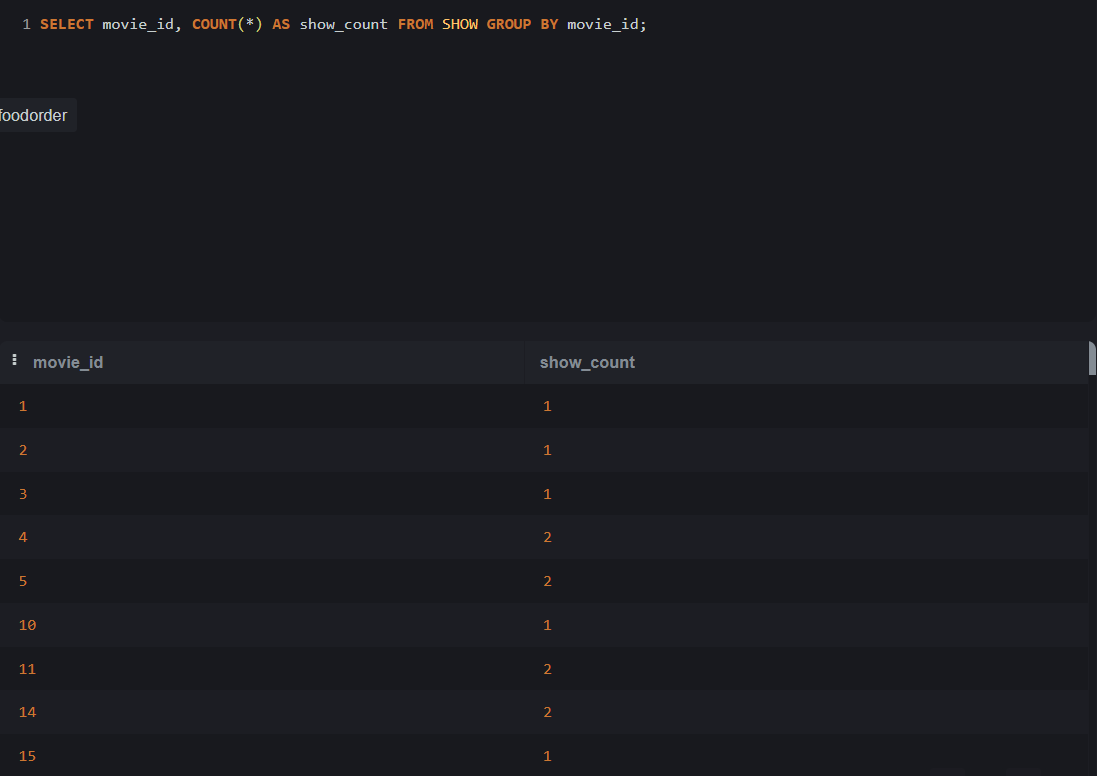


Lab 11.4.2: SELECT screen\_id, MIN(show\_datetime) AS earliest, MAX(show\_datetime) AS latest FROM Shows GROUP BY screen\_id;

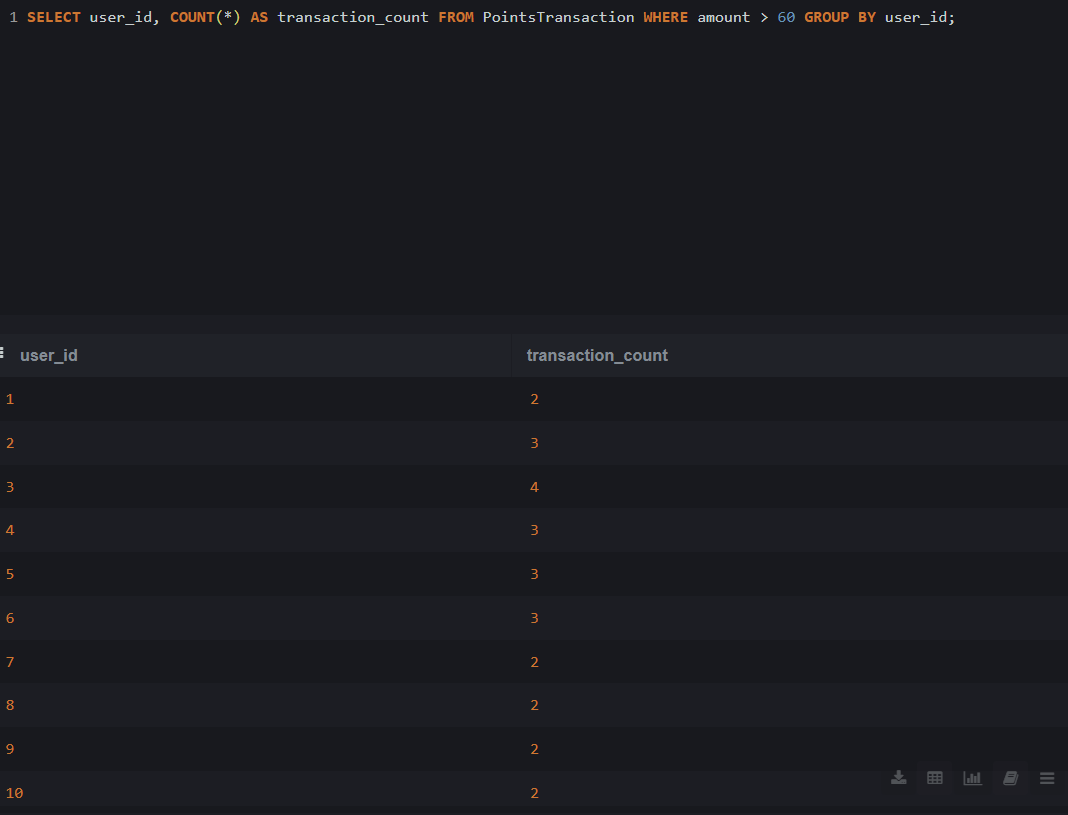


Topic 12: GROUP BY with Aggregate Functions

Lab 12.1: SELECT movie\_id, COUNT(\*) AS show\_count FROM Shows GROUP BY movie\_id;

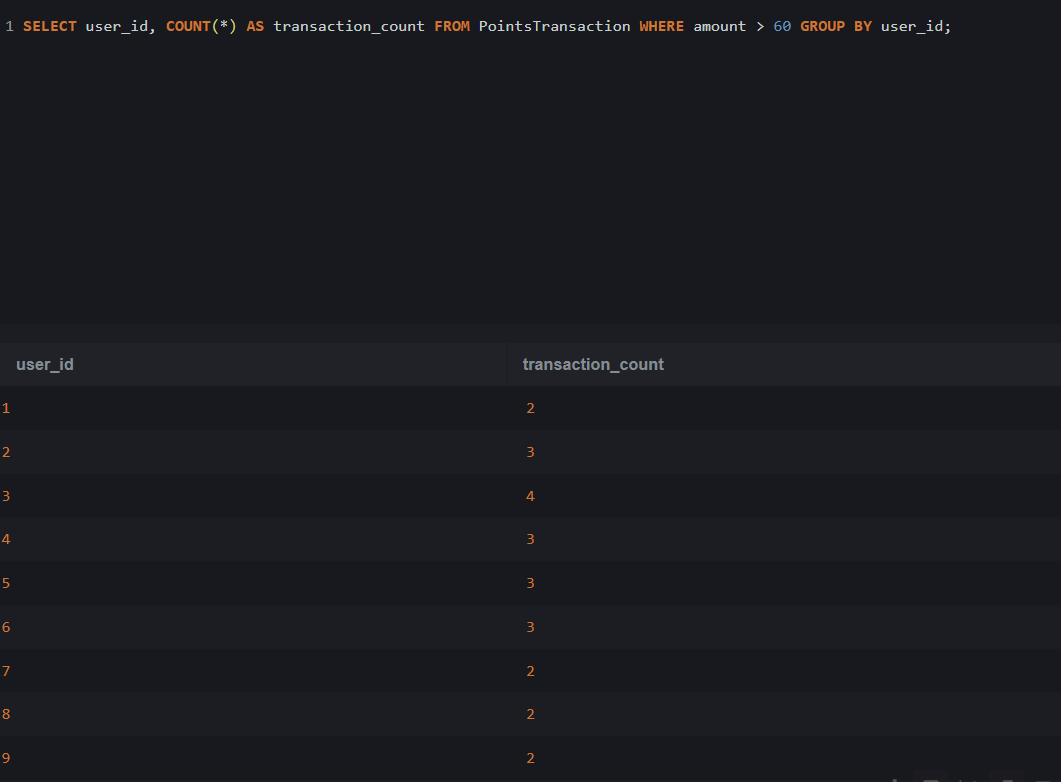


Lab 12.2: SELECT user\_id, COUNT(\*) AS transaction\_count FROM LoyaltyPointsTransactions WHERE amount > 60 GROUP BY user\_id;

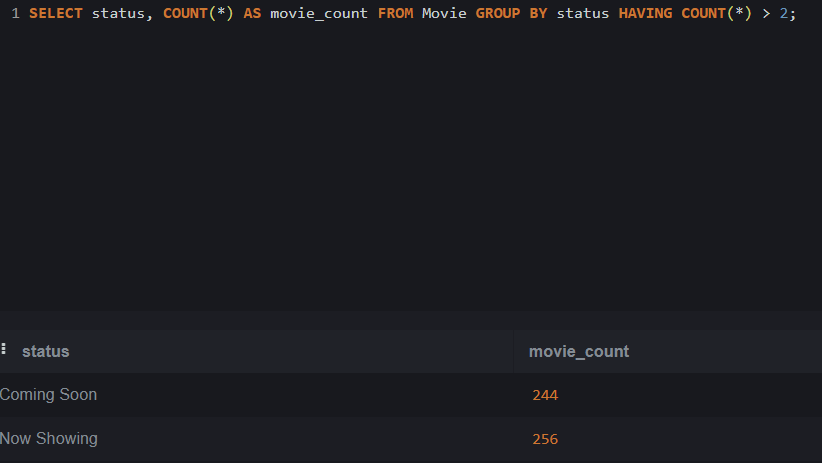


Topic 13: GROUP BY with HAVING Clause

Lab 13.1: SELECT screen\_id, COUNT(\*) AS show\_count FROM Shows GROUP BY screen\_id HAVING COUNT(\*) > 1;

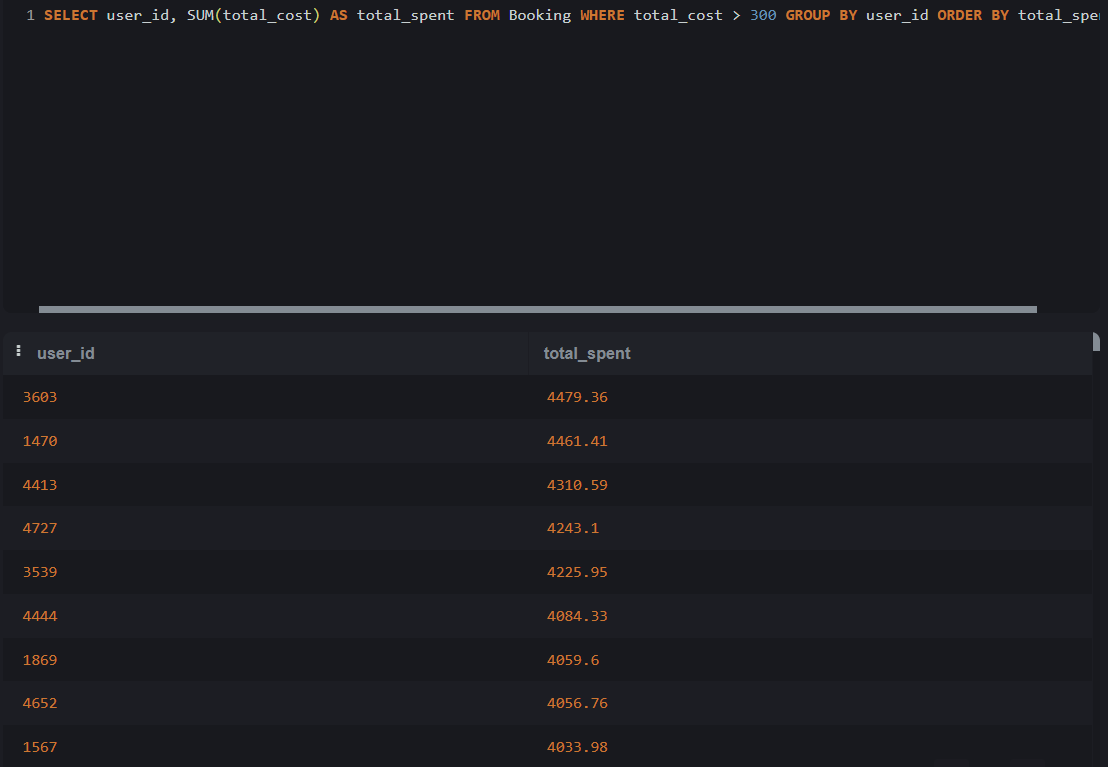


Lab 13.2: SELECT status, COUNT(\*) AS movie\_count FROM Movies GROUP BY status HAVING COUNT(\*) > 2;

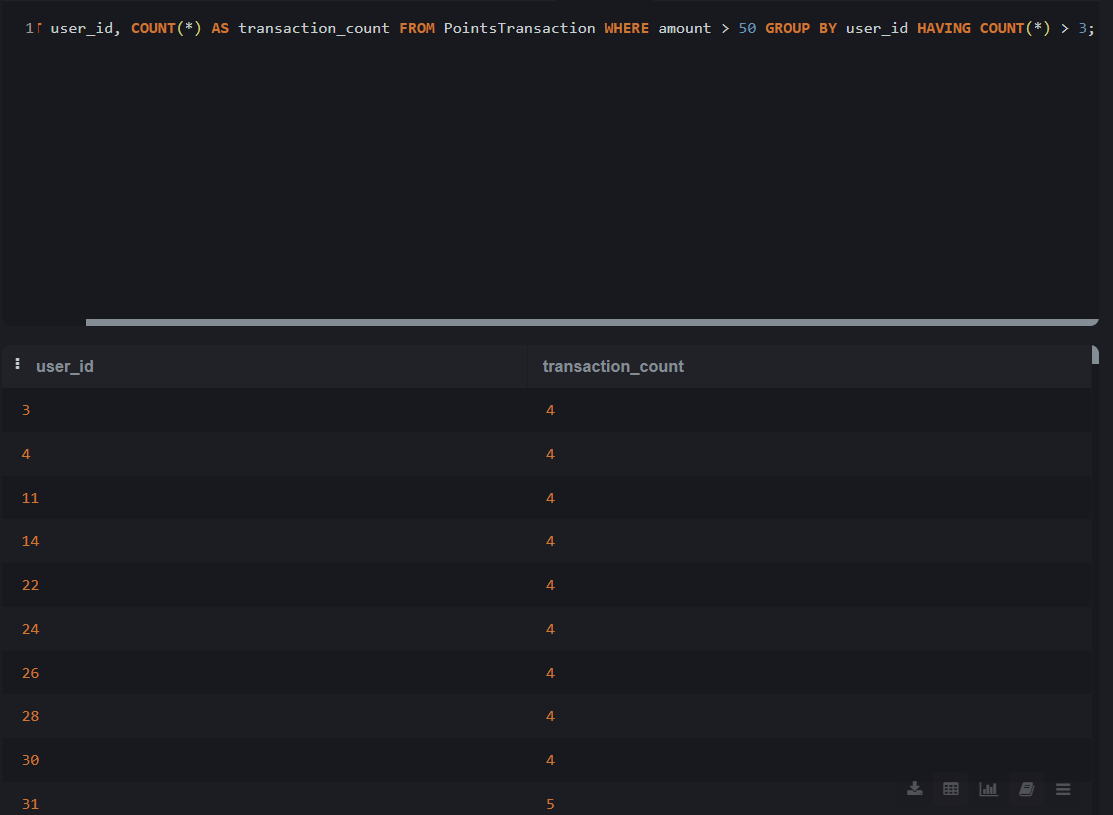


Topic 14: GROUP BY with WHERE and ORDER BY

Lab 14.1: SELECT user\_id, SUM(total\_amount) AS total\_spent FROM Bookings WHERE total\_amount > 300 GROUP BY user\_id ORDER BY total\_spent DESC;

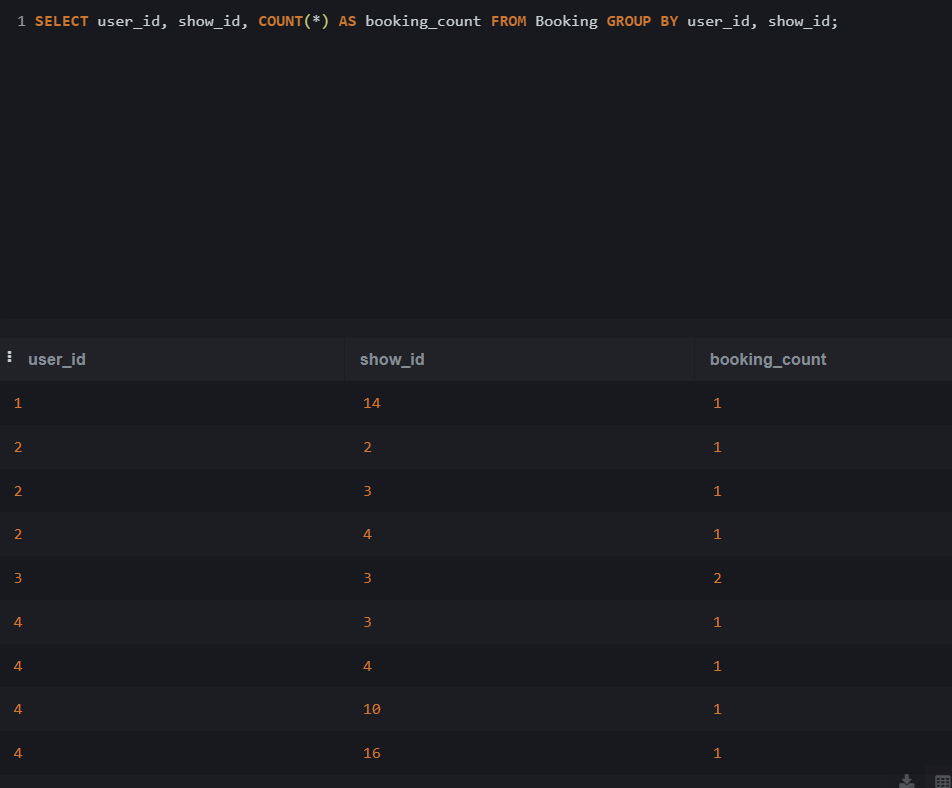


Lab 14.2: SELECT user\_id, COUNT(\*) AS transaction\_count FROM LoyaltyPointsTransactions WHERE amount > 50 GROUP BY user\_id HAVING COUNT(\*) > 3;

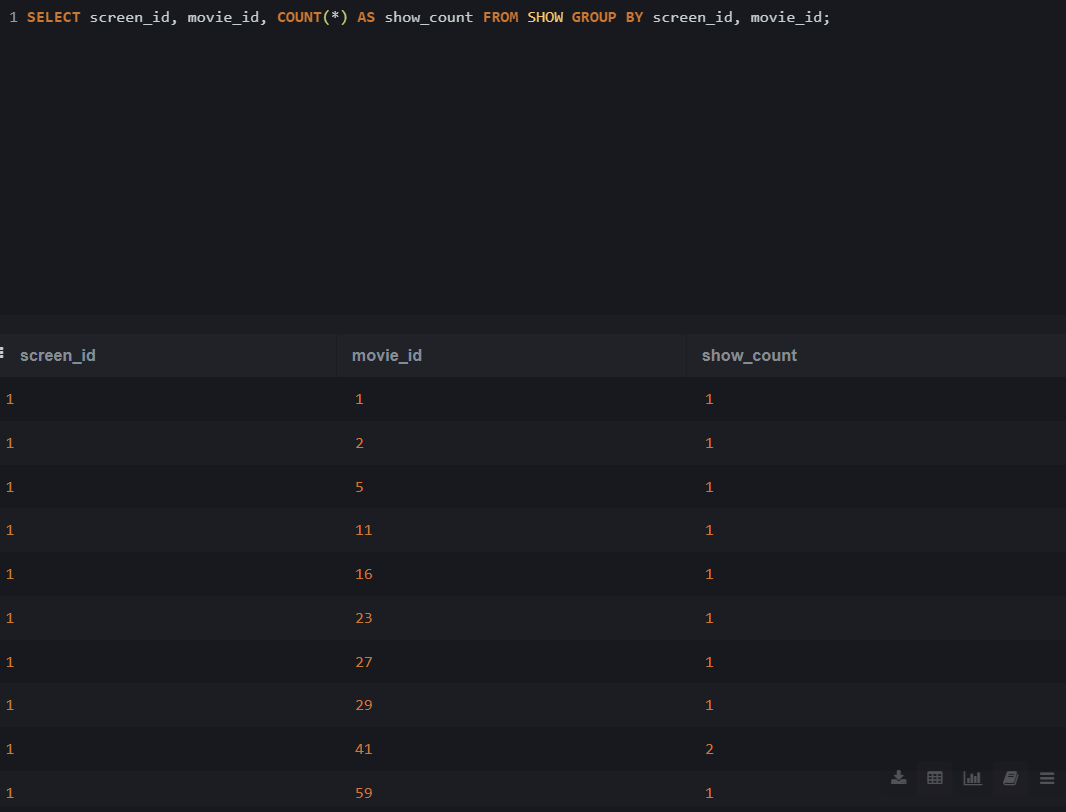


Topic 15: GROUP BY Multiple Columns

Lab 15.1: SELECT user\_id, show\_id, COUNT(\*) AS booking\_count FROM Bookings GROUP BY user\_id, show\_id;

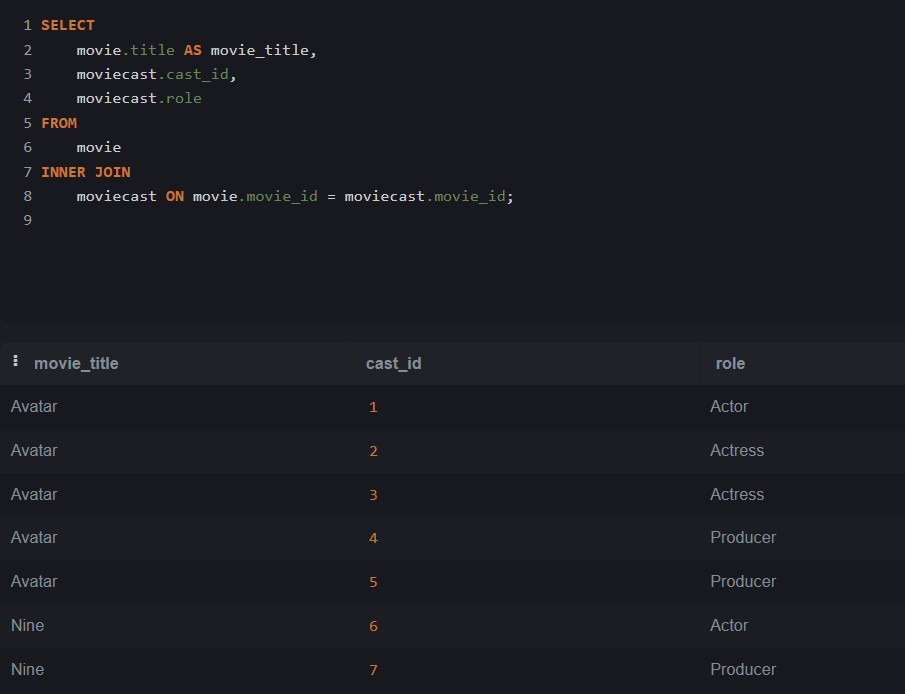


Lab 15.2: SELECT screen\_id, movie\_id, COUNT(\*) AS show\_count FROM Shows GROUP BY screen\_id, movie\_id;



Topic 16: INNER JOIN

Lab 16.1: SELECT m.title, c.name, mc.role FROM Movies m INNER JOIN MovieCast mc ON m.id = mc.movie\_id INNER JOIN Cast c ON mc.cast\_id = c.id;



Lab 16.2: SELECT

movie.title AS movie\_title,

screen.screen\_name,

show.show\_datetime

FROM

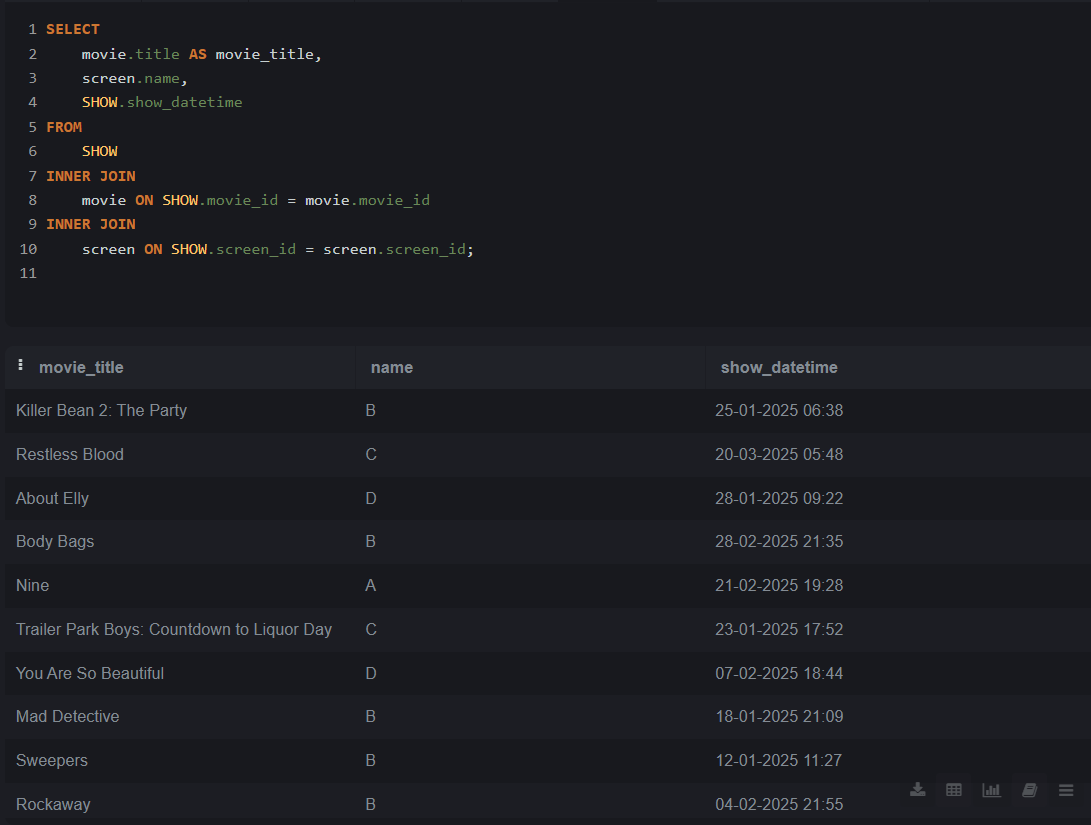
show

INNER JOIN

movie ON show.movie\_id = movie.movie\_id

INNER JOIN

screen ON show.screen\_id = screen.screen\_id;



Topic 17: LEFT JOIN

Lab 17.1 SELECT

movie.title AS movie\_title,

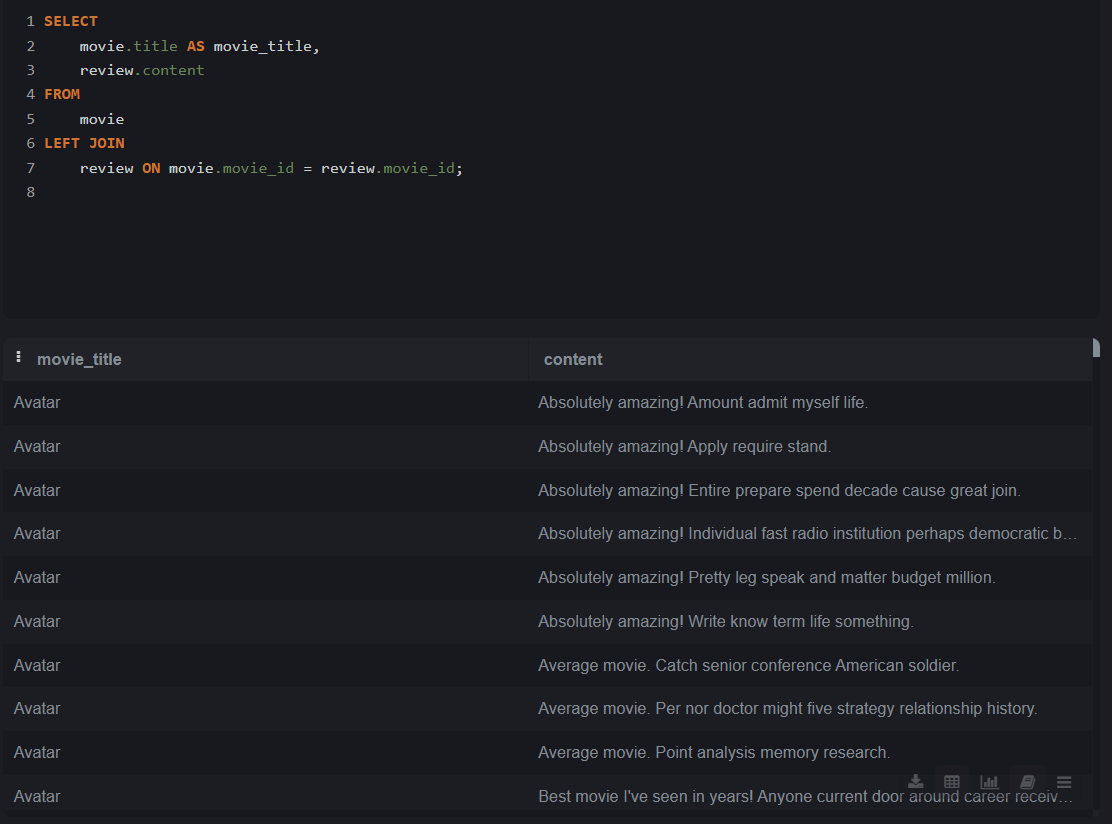
review.content

FROM

movie

LEFT JOIN

review ON movie.movie\_id = review.movie\_id;



Lab 17.2: SELECT

movie.title AS movie\_title,

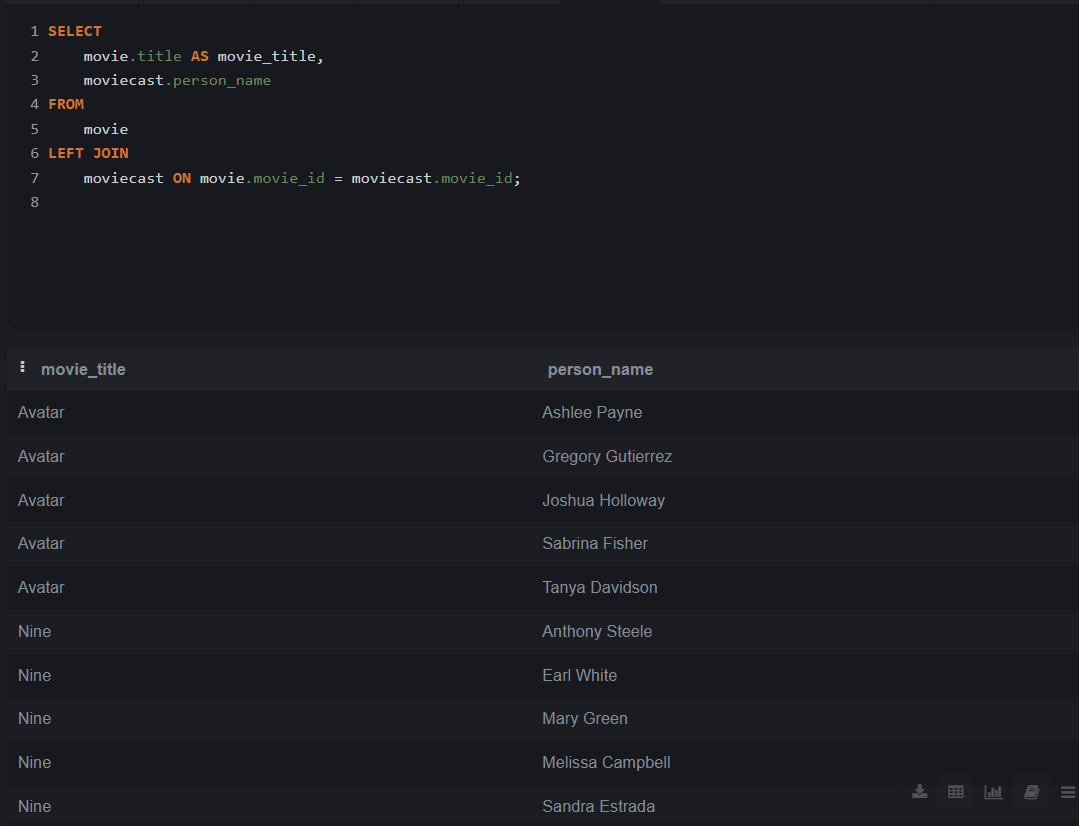
moviecast.cast\_name

FROM

movie

LEFT JOIN

moviecast ON movie.movie\_id = moviecast.movie\_id;



Topic 18: RIGHT JOIN

Lab 18.1: SELECT

seat.seat\_id,

seat.seat\_number,

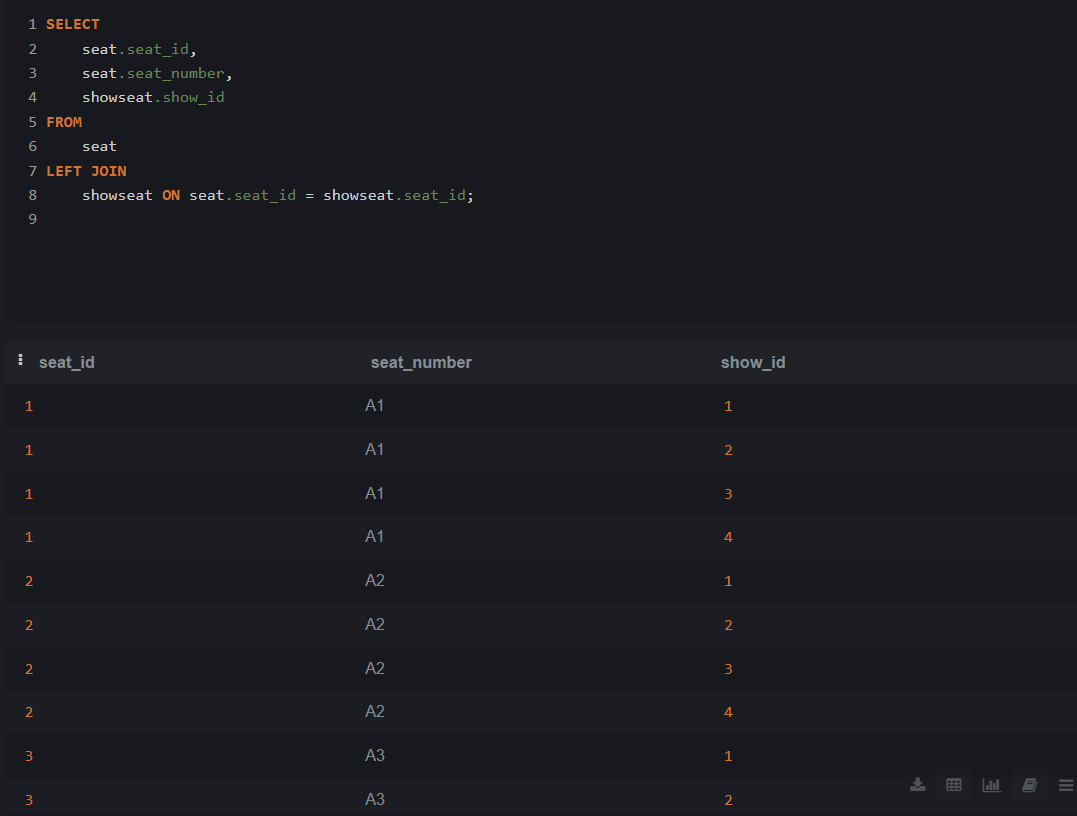
showseat.show\_id

FROM

seat

LEFT JOIN

showseat ON seat.seat\_id = showseat.seat\_id;

Lab 18.2: SELECT

booking.booking\_id,

user.name AS user\_name,

booking.booking\_datetime,

movie.title AS movie\_title

FROM

booking

INNER JOIN

user ON booking.user\_id = user.user\_id

INNER JOIN

ticket ON booking.booking\_id = ticket.booking\_id

INNER JOIN

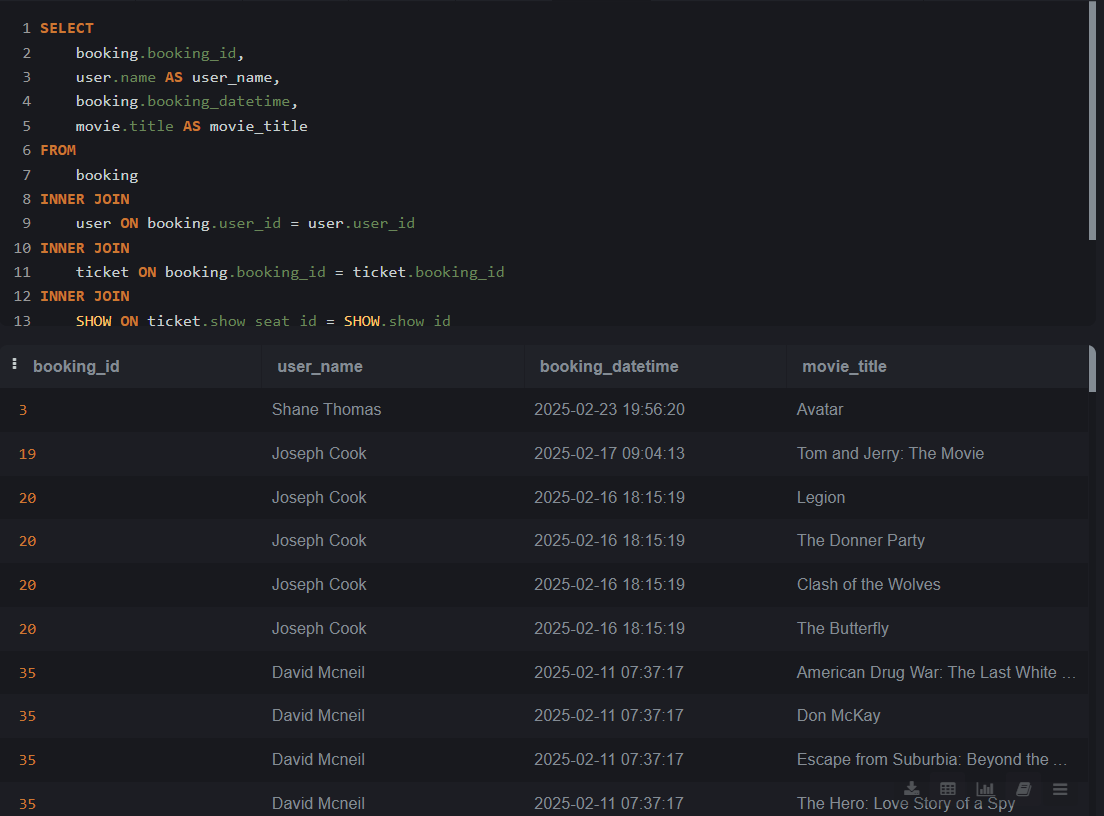
show ON ticket.show\_id = show.show\_id

INNER JOIN

movie ON show.movie\_id = movie.movie\_id

LEFT JOIN

payment ON booking.booking\_id = payment.booking\_id;



Topic 19: FULL OUTER JOIN

Lab 19.1: SELECT

movie.title AS movie\_title,

review.review\_content

FROM

movie

LEFT JOIN

review ON movie.movie\_id = review.movie\_id

UNION

SELECT

movie.title AS movie\_title,

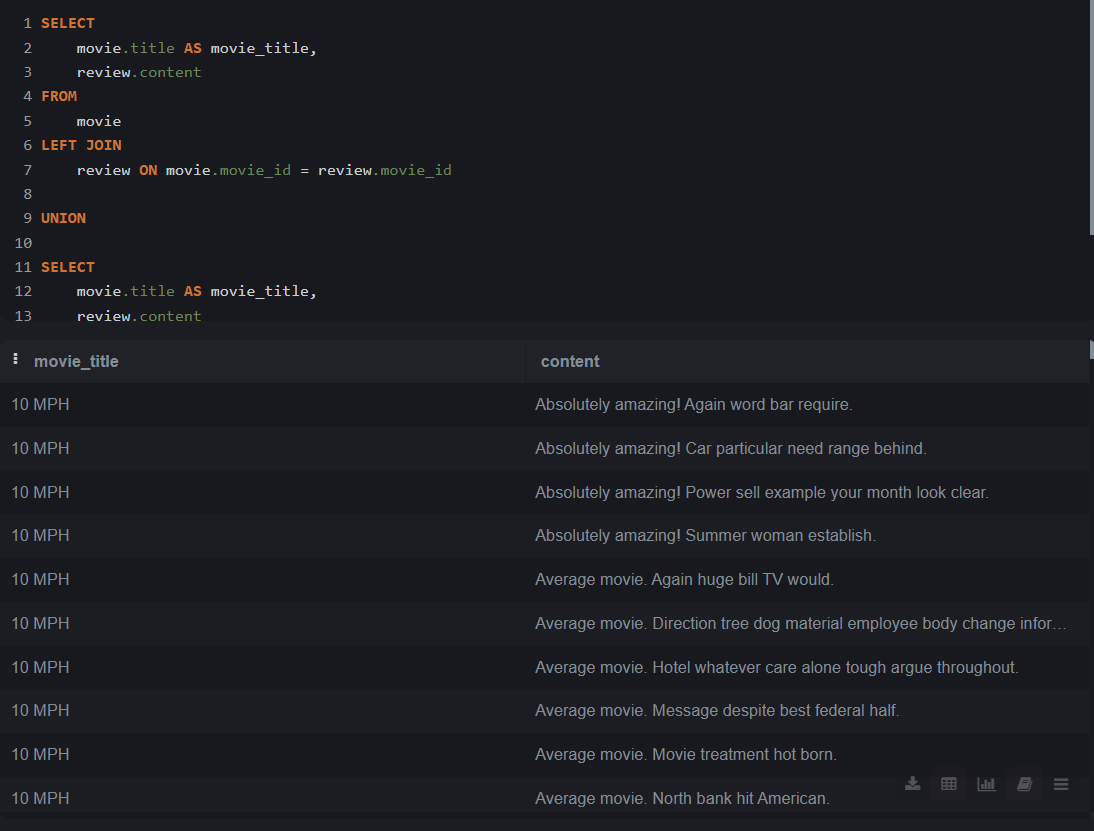
review.review\_content

FROM

review

LEFT JOIN

movie ON review.movie\_id = movie.movie\_id;

Lab 19.2: SELECT

user.name AS user\_name,

ticket.ticket\_id

FROM

user

LEFT JOIN

booking ON user.user\_id = booking.user\_id

LEFT JOIN

ticket ON booking.booking\_id = ticket.booking\_id

UNION

SELECT

user.name AS user\_name,

ticket.ticket\_id

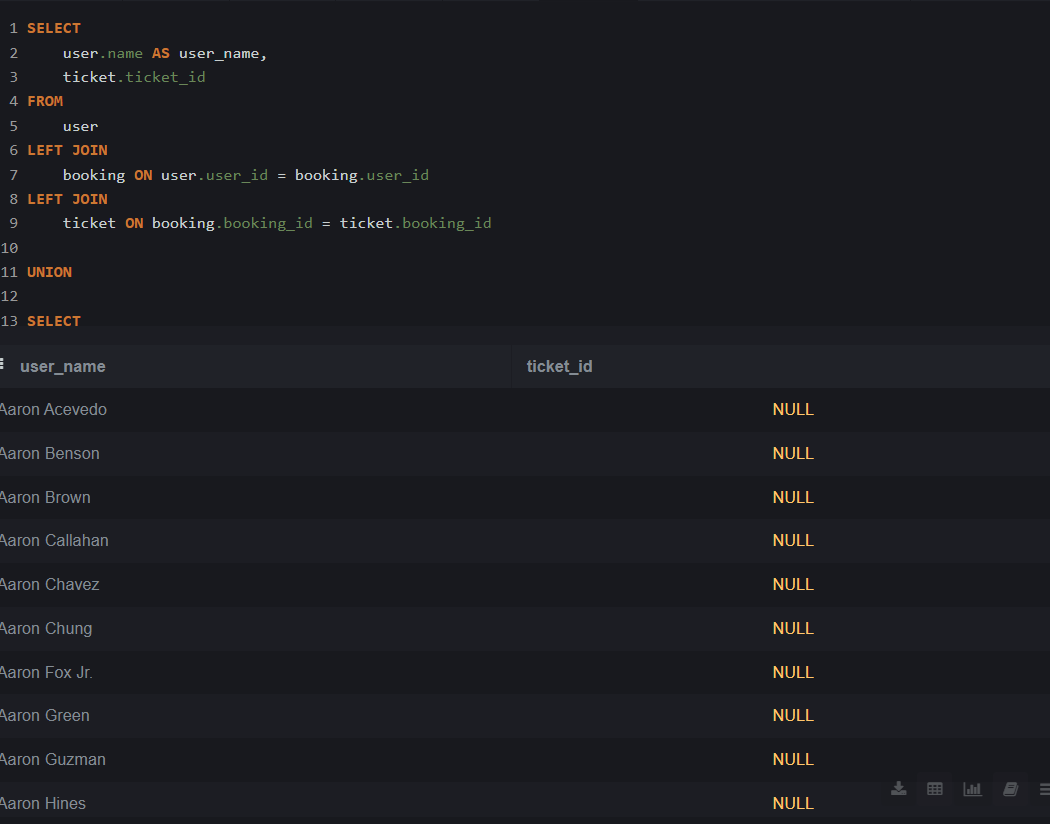
FROM

ticket

LEFT JOIN

booking ON ticket.booking\_id = booking.booking\_id

LEFT JOIN

user ON booking.user\_id = user.user\_id; 

Topic 20: JOIN with Multiple Tables

Lab 20.1: SELECT

fooditem.name AS food\_item\_name,

fooditemsize.size\_name,

foodorderitem.quantity,

(foodorderitem.quantity \* fooditemsize.price) AS total\_cost

FROM

foodorderitem

INNER JOIN

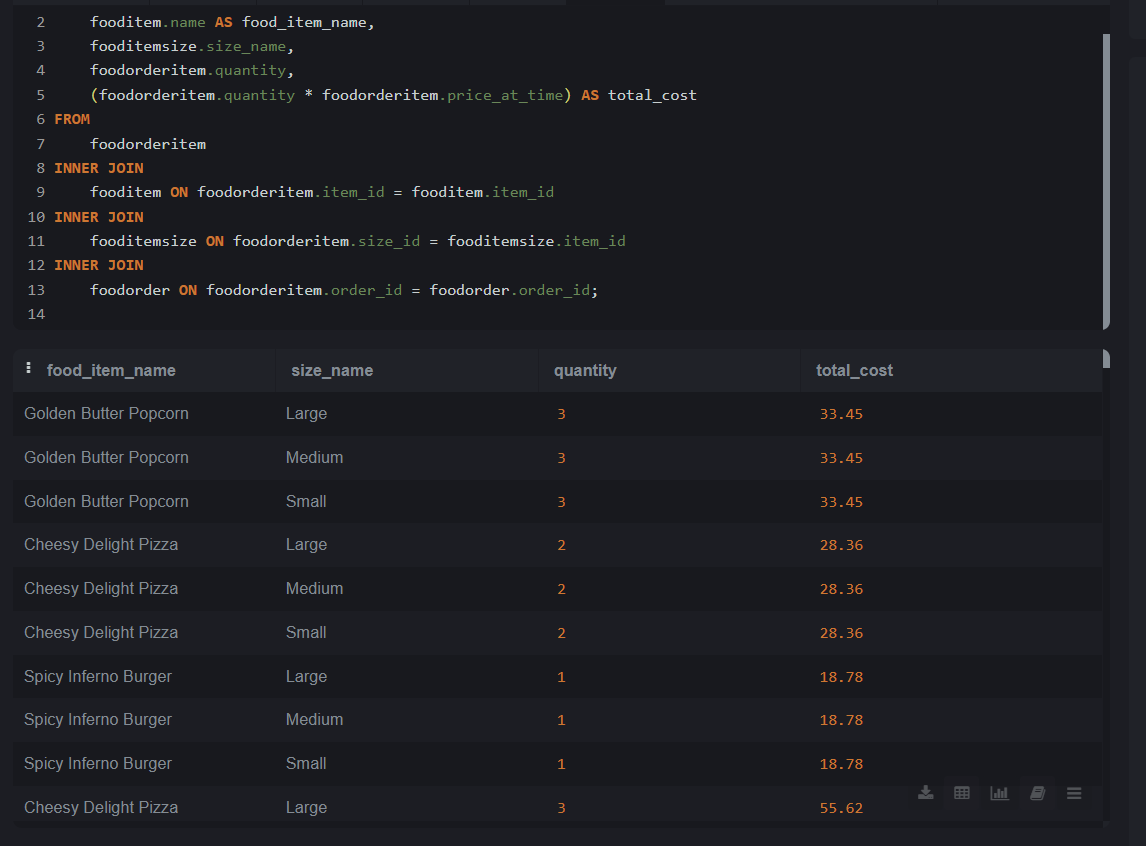
fooditemsize ON foodorderitem.fooditemsize\_id = fooditemsize.fooditemsize\_id

INNER JOIN

fooditem ON fooditemsize.fooditem\_id = fooditem.fooditem\_id

INNER JOIN

foodorder ON foodorderitem.foodorder\_id = foodorder.foodorder\_id;

Lab 20.2 SELECT

fooditem.name AS food\_item\_name,

foodorderitem.quantity,

fooditemsize.rate

FROM

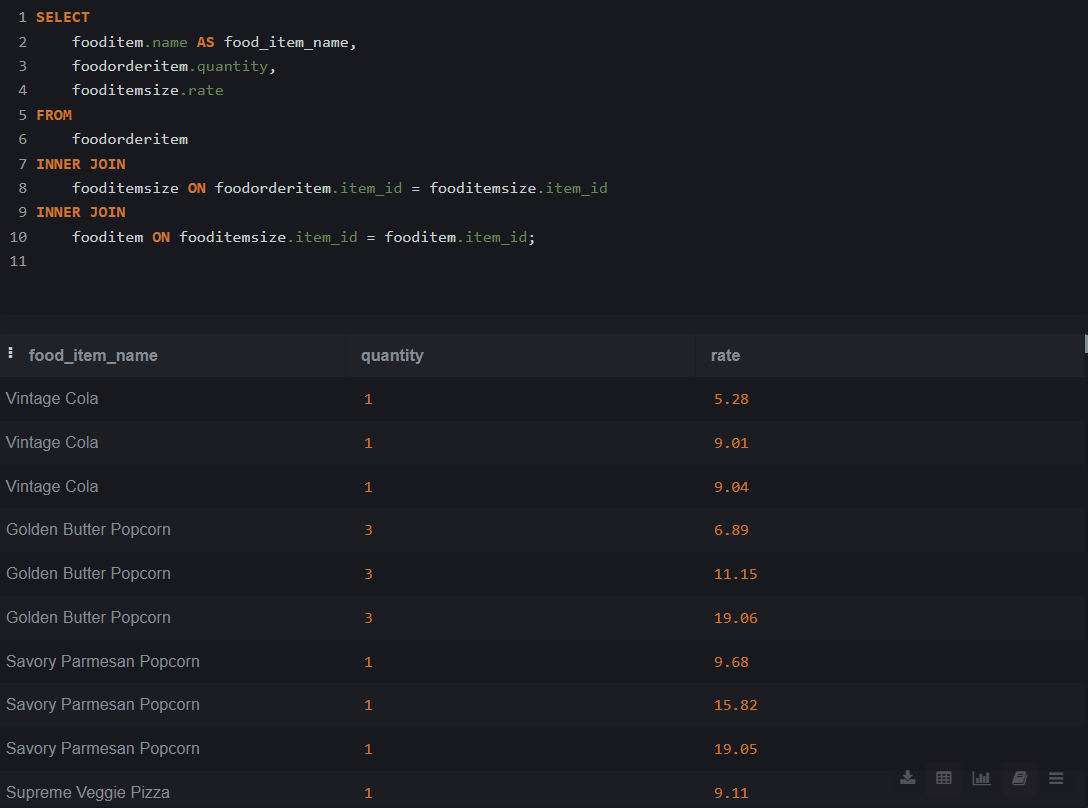
foodorderitem

INNER JOIN

fooditemsize ON foodorderitem.item\_id = fooditemsize.item\_id

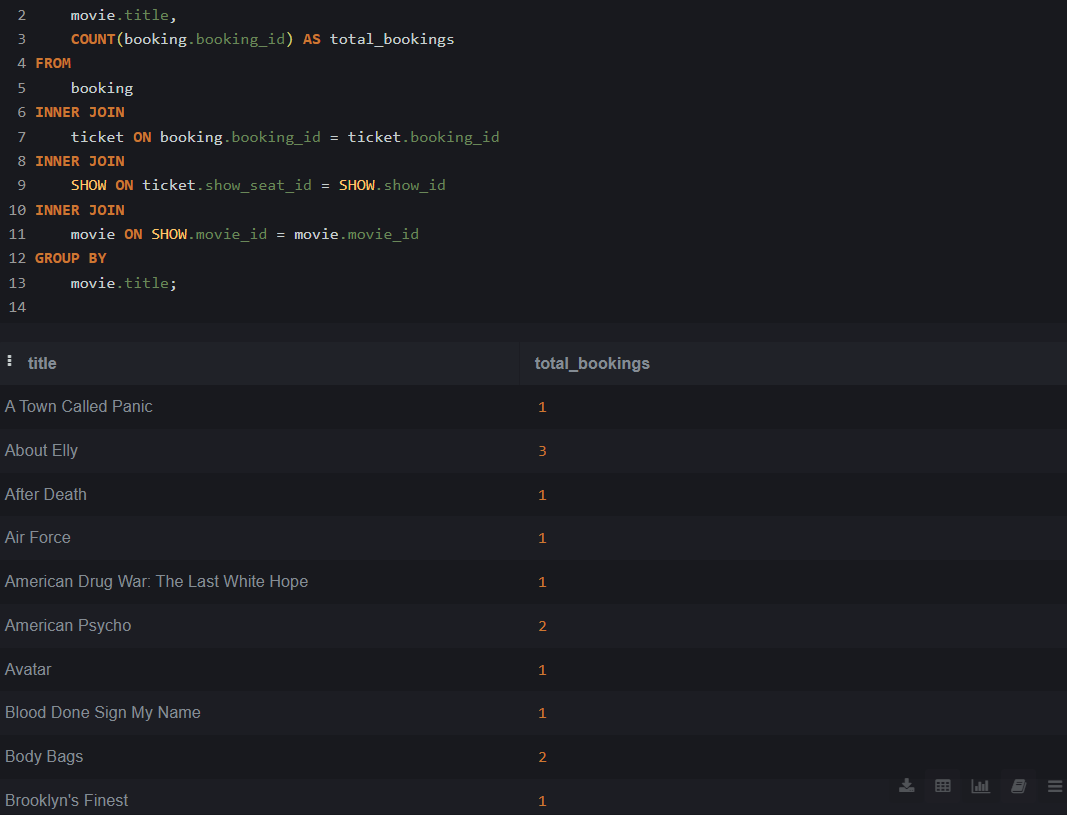
INNER JOIN

fooditem ON fooditemsize.item\_id = fooditem.item\_id;



Topic 21: JOIN with GROUP BY and Aggregate Functions

Lab 21.1: SELECT m.title, COUNT(b.id) AS total\_bookings FROM Movies m JOIN Bookings b ON m.id = b.movie\_id GROUP BY m.title;



Lab 21.2: SELECT

screen.name,

COUNT(showseat.seat\_id) AS seat\_count

FROM

showseat

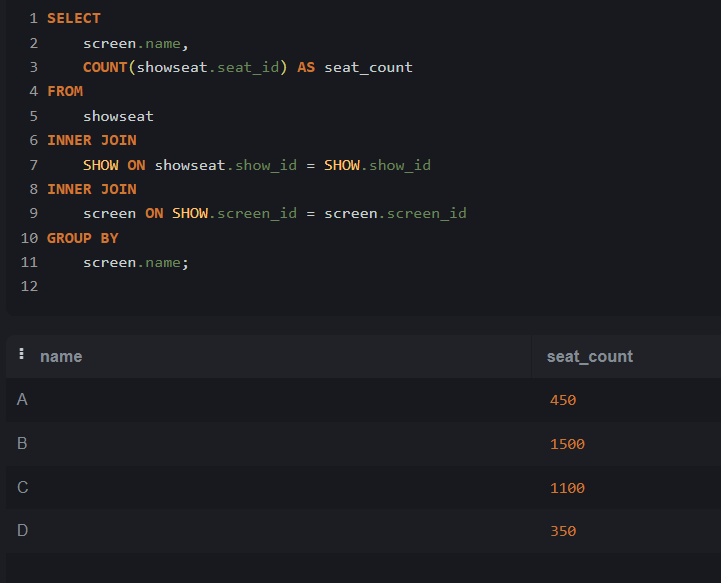
INNER JOIN

show ON showseat.show\_id = show.show\_id

INNER JOIN

screen ON show.screen\_id = screen.screen\_id

GROUP BY

screen.name;

Topic 22: JOIN with WHERE and HAVING Clause

Lab 22.1: SELECT

movie.title,

COUNT(review.review\_id) AS review\_count

FROM

movie

INNER JOIN

review ON movie.movie\_id = review.movie\_id

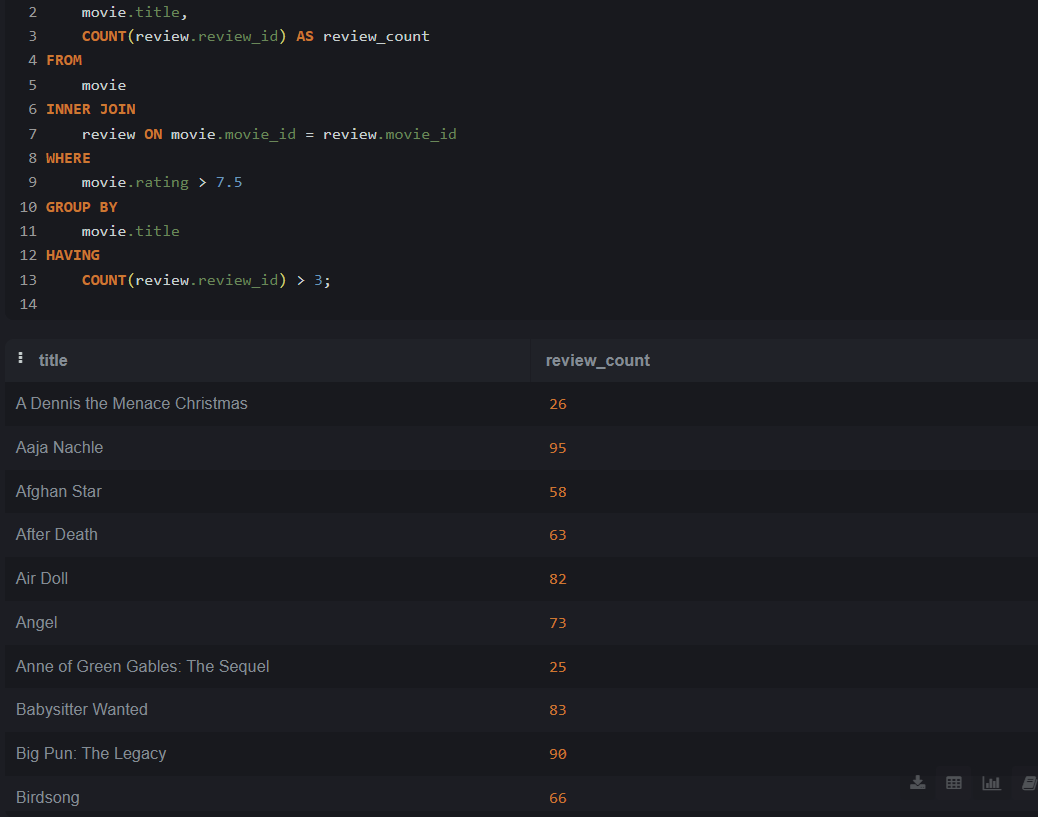
WHERE

movie.rating > 7.5

GROUP BY

movie.title

HAVING

COUNT(review.review\_id) > 3;

Lab 22.2: SELECT

movie.title,

screen.name,

show.show\_datetime

FROM

show

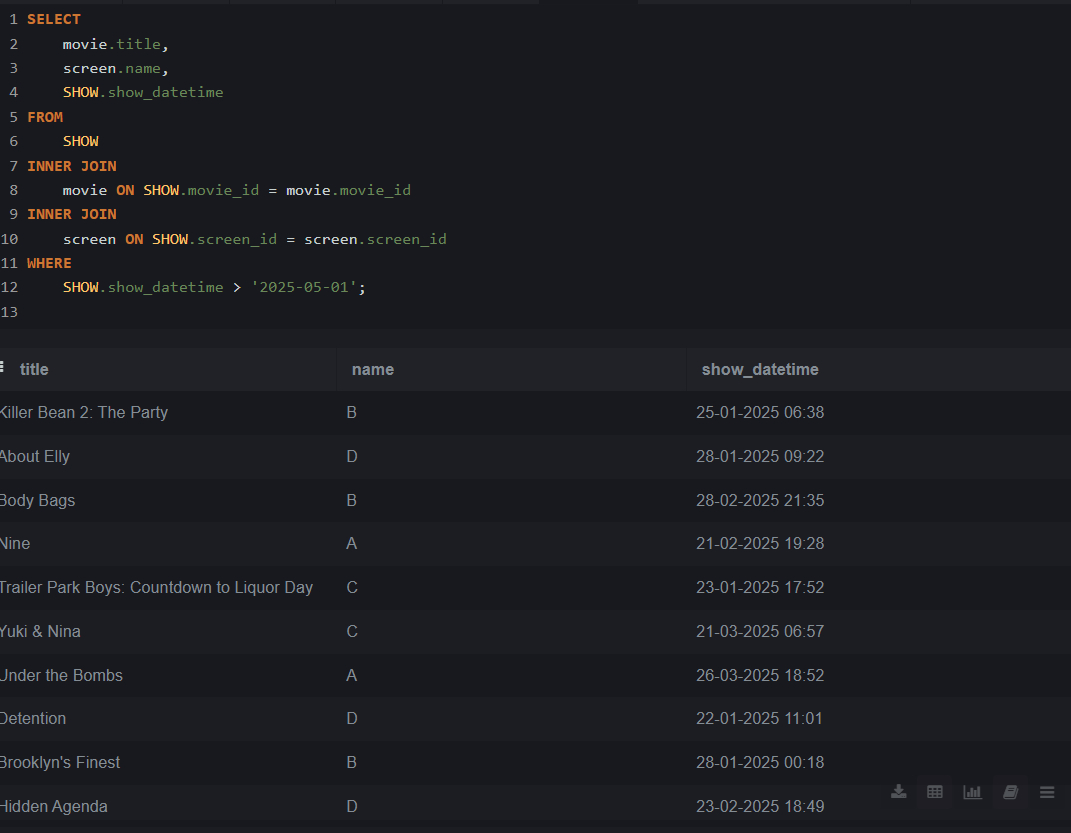
INNER JOIN

movie ON show.movie\_id = movie.movie\_id

INNER JOIN

screen ON show.screen\_id = screen.screen\_id

WHERE

show.show\_datetime > '2025-05-01';

Topic 23: JOIN with ORDER BY and LIMIT

Lab 23.1: SELECT

fooditem.name,

fooditemsize.rate

FROM

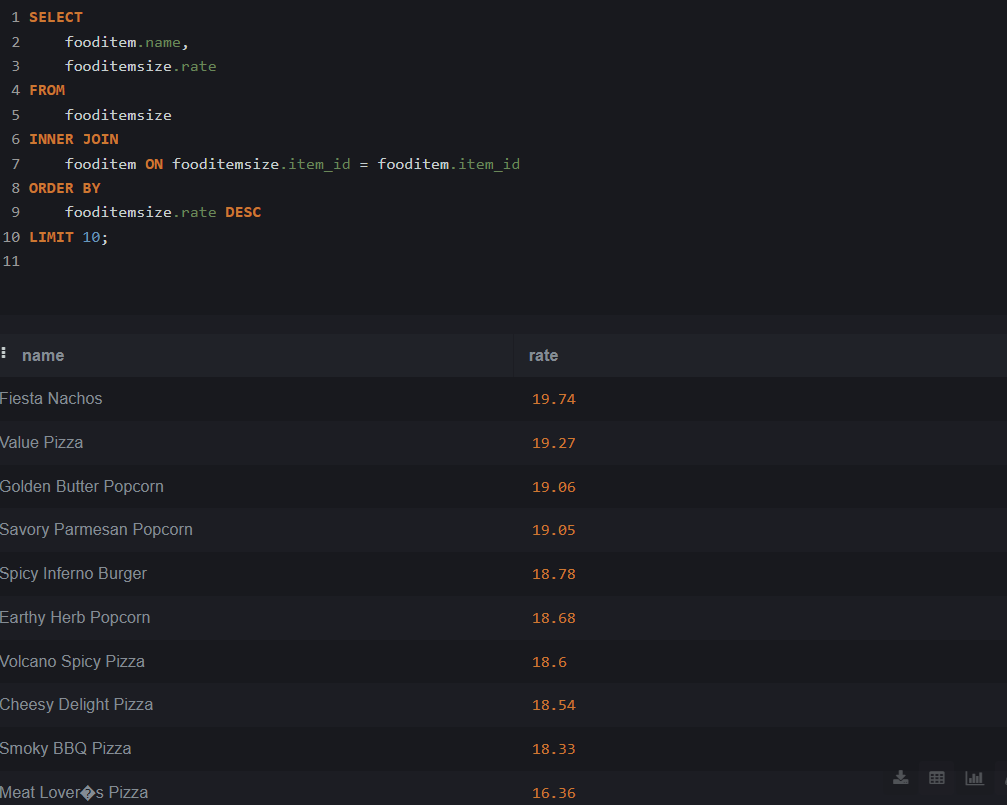
fooditemsize

INNER JOIN

fooditem ON fooditemsize.item\_id = fooditem.item\_id

ORDER BY

fooditemsize.rate DESC

LIMIT 10;

Lab 23.2 SELECT

user.name AS user\_name,

booking.total\_cost

FROM

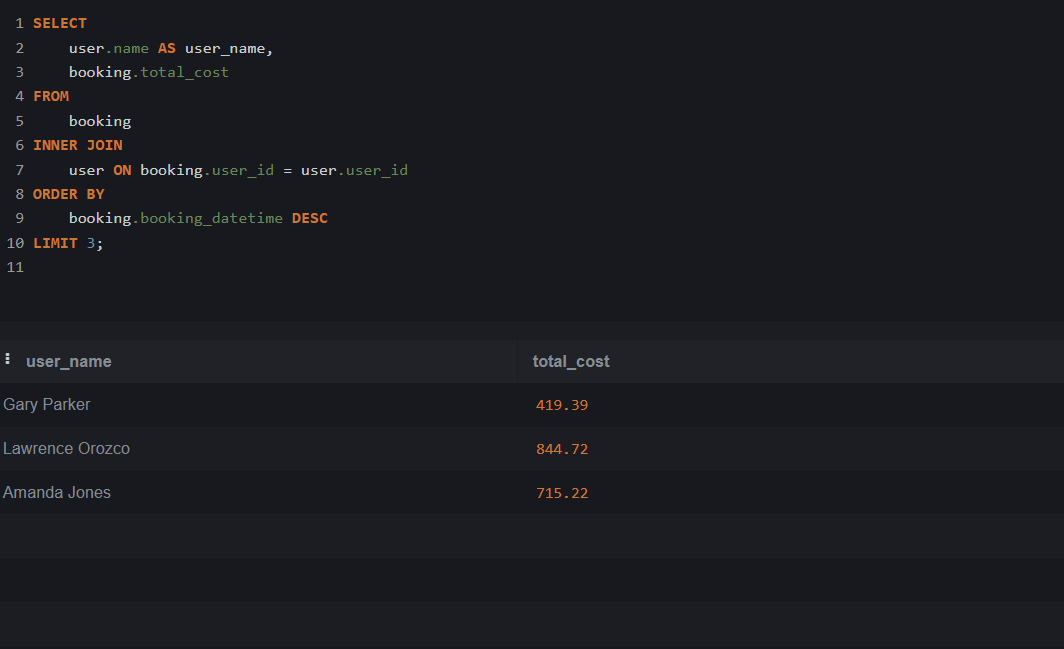
booking

INNER JOIN

user ON booking.user\_id = user.user\_id

ORDER BY

booking.booking\_datetime DESC

LIMIT 3;

Topic 24: JOIN with Aggregate + WHERE + Multiple Conditions

Lab 24.1 SELECT

screen.screen\_id,

SUM(fooditemsize.rate \* foodorderitem.quantity) AS total\_food\_cost

FROM

foodorderitem

INNER JOIN

foodorder ON foodorderitem.order\_id = foodorder.order\_id

INNER JOIN

booking ON foodorder.booking\_id = booking.booking\_id

INNER JOIN

ticket ON booking.booking\_id = ticket.booking\_id

INNER JOIN

show ON ticket.ticket\_id = show.show\_id

INNER JOIN

screen ON show.screen\_id = screen.screen\_id

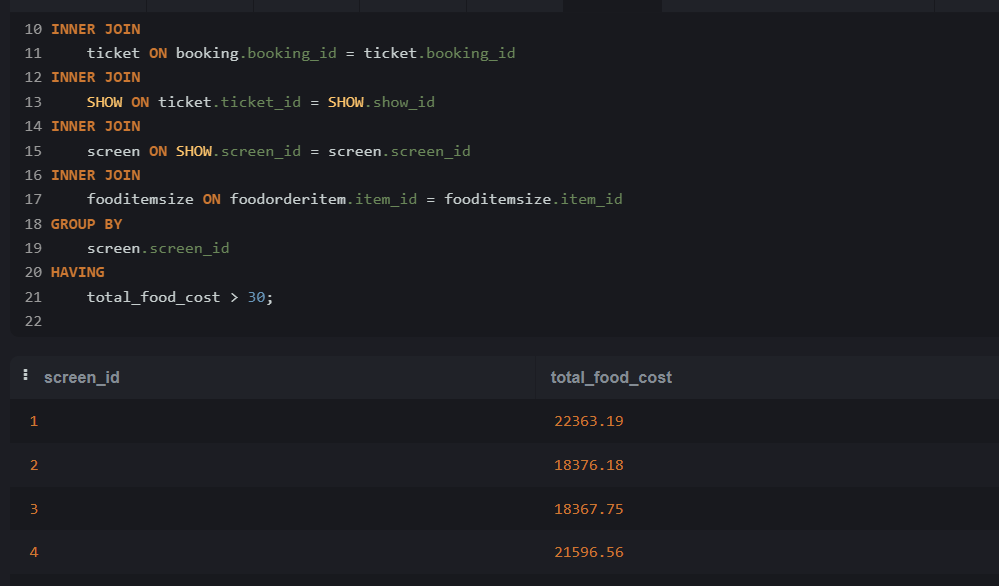
INNER JOIN

fooditemsize ON foodorderitem.item\_id = fooditemsize.item\_id

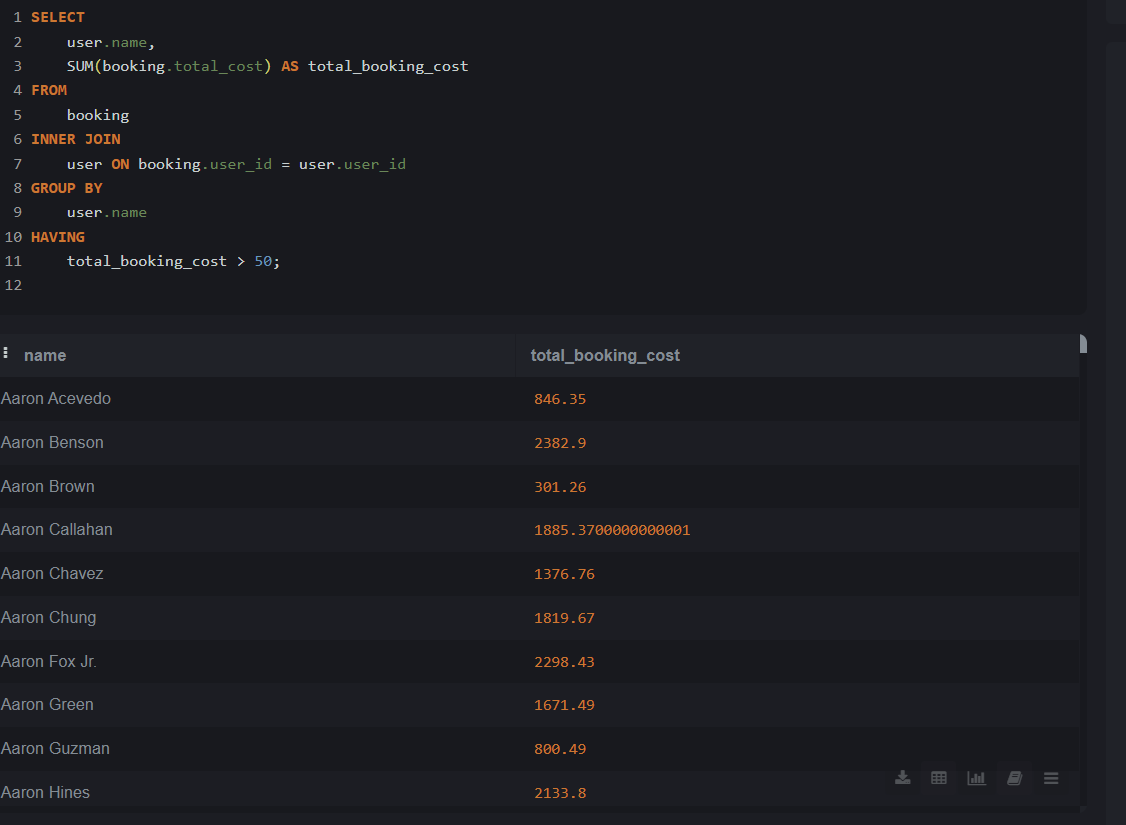
GROUP BY

screen.screen\_id

HAVING

total\_food\_cost > 30;

Lab 24.2: SELECT u.name, SUM(b.total\_amount) AS total\_booking\_cost FROM Users u JOIN Bookings b ON u.id = b.user\_id GROUP BY u.name HAVING SUM(b.total\_amount) > 50;



Topic 25: Complex JOINs involving 3+ Tables

Lab 25.1 SELECT

user.name AS user\_name,

SUM(fooditemsize.rate \* foodorderitem.quantity) AS total\_spent\_on\_food

FROM

foodorder

INNER JOIN

booking ON foodorder.booking\_id = booking.booking\_id

INNER JOIN

user ON booking.user\_id = user.user\_id

INNER JOIN

foodorderitem ON foodorder.order\_id = foodorderitem.order\_id

INNER JOIN

fooditemsize ON foodorderitem.item\_id = fooditemsize.item\_id

GROUP BY

user.name

HAVING

total\_spent\_on\_food > 100;

