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\_amount 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\_amount > 400;

Topic 3: SELECT with WHERE using AND/OR

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

Lab 3.2: SELECT \* FROM Bookings WHERE total\_cost > 500 OR booking\_date > '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 Movies WHERE status != 'Inactive';

Topic 5: SELECT with WHERE and BETWEEN

Lab 5.1: SELECT \* FROM FoodSizes 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 FoodItems 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 s.show\_datetime, m.title, sc.screen\_name FROM Shows s INNER JOIN Movies m ON s.movie\_id = m.id INNER JOIN Screens sc ON s.screen\_id = sc.id;

Topic 17: LEFT JOIN

Lab 17.1: SELECT m.title, r.content FROM Movies m LEFT JOIN Reviews r ON m.id = r.movie\_id;

Lab 17.2: SELECT m.title, c.name FROM Movies m LEFT JOIN MovieCast mc ON m.id = mc.movie\_id LEFT JOIN Cast c ON mc.cast\_id = c.id;

Topic 18: RIGHT JOIN

Lab 18.1: SELECT s.seat\_number, sh.show\_datetime FROM Seats s RIGHT JOIN Shows sh ON s.show\_id = sh.id;

Lab 18.2: SELECT b.booking\_id, u.name, b.booking\_date, m.title FROM Bookings b RIGHT JOIN Users u ON b.user\_id = u.id LEFT JOIN Movies m ON b.movie\_id = m.id;

Topic 19: FULL OUTER JOIN

Lab 19.1: SELECT m.title, r.content FROM Movies m FULL OUTER JOIN Reviews r ON m.id = r.movie\_id;

Lab 19.2: SELECT u.name, t.ticket\_id FROM Users u FULL OUTER JOIN Tickets t ON u.id = t.user\_id;

Topic 20: JOIN with Multiple Tables

Lab 20.1: SELECT o.booking\_id, f.name, s.size, oi.quantity, oi.total\_cost FROM FoodOrders o JOIN FoodOrderItems oi ON o.id = oi.order\_id JOIN FoodItems f ON oi.food\_id = f.id JOIN FoodSizes s ON f.size\_id = s.id;

Lab 20.2: SELECT f.name, oi.quantity, oi.price FROM FoodOrderItems oi JOIN FoodItems f ON oi.food\_id = f.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 s.screen\_name, COUNT(sb.seat\_id) AS seat\_count FROM Screens s JOIN SeatBookings sb ON s.id = sb.screen\_id GROUP BY s.screen\_name;

Topic 22: JOIN with WHERE and HAVING Clause

Lab 22.1: SELECT m.title, COUNT(r.id) AS review\_count FROM Movies m JOIN Reviews r ON m.id = r.movie\_id WHERE m.rating > 7.5 GROUP BY m.title HAVING COUNT(r.id) > 3;

Lab 22.2: SELECT s.show\_datetime, m.title, sc.screen\_name FROM Shows s JOIN Movies m ON s.movie\_id = m.id JOIN Screens sc ON s.screen\_id = sc.id WHERE s.show\_datetime > '2025-05-01';

Topic 23: JOIN with ORDER BY and LIMIT

Lab 23.1: SELECT name, price FROM FoodItems ORDER BY price DESC LIMIT 10;

Lab 23.2: SELECT u.name, b.total\_cost FROM Bookings b JOIN Users u ON b.user\_id = u.id ORDER BY b.booking\_datetime DESC LIMIT 3;

Topic 24: JOIN with Aggregate + WHERE + Multiple Conditions

Lab 24.1: SELECT s.id AS screen\_id, SUM(fo.total\_cost) AS total\_cost FROM Screens s JOIN Shows sh ON s.id = sh.screen\_id JOIN FoodOrders fo ON sh.id = fo.show\_id GROUP BY s.id HAVING SUM(fo.total\_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 u.name, o.id AS order\_id, SUM(oi.total\_cost) AS total\_cost FROM FoodOrders o JOIN FoodOrderItems oi ON o.id = oi.order\_id JOIN Users u ON o.user\_id = u.id GROUP BY u.name, o.id HAVING SUM(oi.total\_cost) > 100;