**Scope: Select,Where,Between,AND,LIKE**

– 1. List all customers:  
Select \* from userr;

– 2. List all orders for a specific customer:  
Select \* from Courier where receiver\_name ='Alice Smith' ;

– 3. List all couriers:  
Select \* from Courier;

– 4. List all packages for a specific order:  
Select \* from Courier where courierID='44';

– 5. List all deliveries for a specific courier:  
 Select \* from Courier where receiver\_name ='Alice Smith' ;

– 6. List all undelivered packages:  
Select \* from Courier where statuss ='Undelivered';

– 7. List all packages that are scheduled for delivery today:  
Select \* from Courier where deliveryDate ='2023-02-12';

– 8. List all packages with a specific status:  
Select \* from Courier where statuss='In Transit';

– 9. Calculate the total number of packages for each courier.  
Select courierID, count(\*) AS TotalPackages from courier group by courierID;

– 10. Find the average delivery time for each courier  
select courierID, avg(deliveryDate) as AvgDeliveryTime from courier group by courierID;

– 11. List all packages with a specific weight range:  
select \* from Courier where Weight in (1.8,2.5) ;

– 12. Retrieve employees whose names contain 'John'  
 select \* from employee where empName ='John';

– 13. Retrieve all courier records with payments greater than 45 rupees.  
select c.\* from Courier c join Payment p on c.courierID=p.courierID where amt > 45 ;

**Scope: GroupBy, Aggregate Functions, Having, Order By, where**

– 14. Find the total number of couriers handled by each employee. select e.employeeID,e.empName as EmployeeName,count(c.courierID)as TotalCouriersHandled from employee e left join courier c on e.empName=c.sender\_name OR e.empName=c.receiver\_name group by e.employeeID , e.empName order by e.employeeID;

– 15. Calculate the total revenue generated by each location select l.locID , l.locName, sum(p.amt ) as TotalRevenue from location l left join payment p on l.locID= p.locID group by l.locID , l.locName order by l.locID;

– 16. Find the total number of couriers delivered to each location. select count(c.courierID)as TotalCouriersDelivered, l.locName , l.locID from location l left join Courier c on c.receiverAddress=l.address where c.statuss='Delivered' group by l.locID , l.locName Order by l.locID;

– 17. Find the courier with the highest average delivery time: select c.courierID, avg(DATEDIFF(now(), deliveryDate)) AS AvgDeliveryTime from courier c group by c.courierID order by AvgDeliveryTime desc limit 1;

– 18. Find Locations with Total Payments Less Than a Certain Amount select l.locID, l.locName, sum(p.amt) as TotalAmountReceived from location l left join payment p on l.locID = p.locId group by l.locID, l.locName having TotalAmountReceived < 70;

– 19. Calculate Total Payments per Location

select l.locID, l.locName, sum(p.amt) as TotalAmountReceived from location l

left join payment p on l.locID = p.locId group by l.locID, l.locName;

– 20. Retrieve couriers who have received payments totaling more than $1000 in a  
– specific location (LocationID = X):  
select c.courierID, c.sender\_name, c.receiver\_name, l.locName, sum(p.amt) as TotalPayments from courier c inner join payment p on c.courierID = p.courierID inner join location l on p.locId = l.locID where l.locName = 'Branch 1' group by c.courierID, c.sender\_name, c.receiver\_name, l.locName HAVING sum(p.amt) > 50;

– 21. Retrieve couriers who have received payments totaling more than $1000 after a – certain date (PaymentDate > 'YYYY-MM-DD'):  
SELECT c.courierID, c.sender\_name, c.receiver\_name, SUM(p.amt) AS TotalPayments FROM courier c INNER JOIN payment p ON c.courierID = p.courierID WHERE p.payDate > '2023-11-06' GROUP BY c.courierID, c.sender\_name, c.receiver\_name HAVING SUM(p.amt) > 50;

– 22. Retrieve locations where the total amount received is more than $5000 before a – certain date (PaymentDate > 'YYYY-MM-DD')  
SELECT l.locID, l.locName, SUM(p.amt) AS TotalAmountReceived FROM location l INNER JOIN payment p ON l.locID = p.locId WHERE p.payDate <= '2023-11-05' GROUP BY l.locID, l.locName HAVING SUM(p.amt) > 60;

**Scope: Inner Join,Full Outer Join, Cross Join, Left Outer Join,Right Outer Join**

– 23. Retrieve Payments with Courier Information SELECT p.\* FROM payment p JOIN courier c ON p.courierID=c.courierID ;

– 24. Retrieve Payments with Location Information SELECT p.\* FROM payment p JOIN location l ON p.locId=l.locID ;

– 25. Retrieve Payments with Courier and Location Information SELECT p.\* FROM payment p JOIN location l ON p.locId=l.locID JOIN courier c ON p.courierID=c.courierID ;

– 26. List all payments with courier details SELECT p.\* FROM payment p JOIN courier c ON p.courierID=c.courierID WHERE c.courierID='44' ;

– 27. Total payments received for each courier SELECT c.courierID, SUM(p.amt) AS TotalPaymentsReceived FROM courier c LEFT JOIN payment p ON c.courierID = p.courierID GROUP BY c.courierID;

– 28. List payments made on a specific date SELECT p.\* FROM payment p WHERE payDate='2023-11-10' ;

– 29. Get Courier Information for Each Payment SELECT p.paymentID, p.amt, p.payDate, c.courierID, c.sender\_name, c.receiver\_name FROM payment p INNER JOIN courier c ON p.courierID = c.courierID;

– 30. Get Payment Details with Location SELECT p.paymentID, p.amt, p.payDate, l.locID, l.locName, l.address FROM payment p INNER JOIN location l ON p.locId = l.locID;

– 31. Calculating Total Payments for Each Courier SELECT courierID, SUM(amt) AS TotalPayments FROM payment GROUP BY courierID;

– 32. List Payments Within a Date Range SELECT \* FROM payment WHERE payDate BETWEEN '2023-11-02' AND '2023-11-08';

– 33. Retrieve a list of all users and their corresponding courier records, including cases – where there are no matches on either side  
SELECT \* FROM userr u LEFT JOIN courier c ON u.userID = c.courierID UNION

SELECT \* FROM userr RIGHT JOIN courier ON u.userID=c.courierID WHERE u.userID IS NULL ;

– 34. Retrieve a list of all couriers and their corresponding services, including cases – where there are no matches on either side  
SELECT \* FROM courier c LEFT JOIN C\_service s ON c.courierID = s.serviceID UNION SELECT \* FROM courier c RIGHT JOIN C\_service s ON c.courierID = s.serviceID WHERE c.courierID IS NULL ;

– 35. Retrieve a list of all employees and their corresponding payments, including cases – where there are no matches on either side  
 SELECT \* FROM employee e LEFT JOIN Payment ON e.employeeID=p.courierID UNION SELECT \* FROM employee e RIGHT JOIN Payment ON e.employeeID=p.courierID WHERE e.employeeID IS NULL;

– 36. List all users and all courier services, showing all possible combinations. SELECT \* FROM userr CROSS JOIN C\_services;

– 37. List all employees and all locations, showing all possible combinations: SELECT \* FROM employee CROSS JOIN location;

– 38. Retrieve a list of couriers and their corresponding sender information (if available) SELECT c.courierID, c.sender\_name, c.senderAddress, c.receiver\_name, c.receiverAddress FROM courier c;

– 39. Retrieve a list of couriers and their corresponding receiver information (if – available): SELECT c.courierID, c.sender\_name, c.senderAddress, c.receiver\_name, c.receiverAddress FROM courier c;

– 40. Retrieve a list of couriers along with the courier service details (if available): SELECT c.\*, s.serviceName, s.cost FROM courier c LEFT JOIN C\_services s ON c.ServiceID = s.ServiceID;

– 41. Retrieve a list of employees and the number of couriers assigned to each – employee: SELECT e.employeeID, e.empName, COUNT(c.courierID) AS NumberOfCouriers FROM employee e LEFT JOIN courier c ON e.employeeID = c.employeeID GROUP BY e.employeeID, e.empName;

– 42. Retrieve a list of locations and the total payment amount received at each location: SELECT l.locID, l.locName, SUM(p.amt) AS TotalPaymentAmount FROM location l LEFT JOIN payment p ON l.locID = p.locId GROUP BY l.locID, l.locName;

– 43. Retrieve all couriers sent by the same sender (based on SenderName). SELECT c1.\* FROM courier c1 JOIN courier c2 ON c1.sender\_name = c2.sender\_name WHERE c1.courierID <> c2.courierID;

– 44. List all employees who share the same role. SELECT e1.\* FROM employee e1 JOIN employee e2 ON e1.empRole = e2.empRole WHERE e1.employeeID <> e2.employeeID;

– 45. Retrieve all payments made for couriers sent from the same location. SELECT p.\* FROM payment p JOIN courier c1 ON p.courierID = c1.courierID JOIN courier c2 ON c1.locId = c2.locId WHERE c1.courierID <> c2.courierID ;

– 46. Retrieve all couriers sent from the same location (based on SenderAddress). SELECT c1.\* FROM courier c1 JOIN courier c2 ON c1.senderAddress = c2.senderAddress WHERE c1.courierID <> c2.courierID;

– 47. List employees and the number of couriers they have delivered:  
 SELECT e.employeeID, e.empName, COUNT(c.courierID) AS NumberOfCouriersDelivered FROM employee e LEFT JOIN courier c ON e.employeeID = c.employeeID GROUP BY e.employeeID, e.empName;

– 48. Find couriers that were paid an amount greater than the cost of their respective  
– courier services SELECT c.\*, s.cost AS ServiceCost, p.amt AS PaymentAmount FROM courier c INNER JOIN payment p ON c.courierID = p.courierID INNER JOIN C\_services s ON c.serviceID = s.serviceID WHERE p.amt > s.cost;

**Scope: Inner Queries, Non Equi Joins, Equi joins,Exist,Any,All**

– 49. Find couriers that have a weight greater than the average weight of all couriers SELECT \* FROM courier WHERE Weight > ( SELECT AVG(Weight) FROM courier);

– 50. Find the names of all employees who have a salary greater than the average – salary: SELECT empName FROM employee WHERE salary > (SELECT AVG(salary) FROM employee );

– 51. Find the total cost of all courier services where the cost is less than the maximum – cost SELECT SUM(cost) AS TotalCost FROM C\_services WHERE cost < ( SELECT MAX(cost) FROM C\_services);

– 52. Find all couriers that have been paid for orders  
SELECT c.\* FROM courier JOIN payment p ON c.courierID=p.courierID;

– 53. Find the locations where the maximum payment amount was made  
SELECT l.\* FROM location JOIN payment p ON l.locID=p.locID WHERE p.amt=(SELECT MAX(amt) FROM payment);

– 54. Find all couriers whose weight is greater than the weight of all couriers sent by a

– specific sender (e.g., 'SenderName'): SELECT \* FROM courier WHERE Weight > (SELECT MAX(Weight) FROM courier

WHERE sender\_name = 'Bob Johnson');