## **Joins - SQL (Inner, LEFT, RIGHT, Full)**

1. Retrieve a list of all customers with their corresponding product names they ordered (use an INNER JOIN between customers and sales tables).
2. List all products and show the details of customers who have placed orders for them. Include products that have no orders (use a LEFT JOIN between products and sales tables).
3. List all orders and their shipping status. Include orders that do not have any shipping records (use a LEFT JOINbetween sales and shippings tables).
4. Retrieve all products, including those with no orders, along with their price. Use a RIGHT JOIN between the products and sales tables.
5. Get a list of all customers who have placed orders, including those with no payment records. Use a FULL OUTER JOIN between the customers and payments tables.

## 

## **Joins + Where Clause**

* Find the total number of completed orders made by customers from the state 'Delhi' (use INNER JOIN between customers and sales and apply a WHERE condition).
* Retrieve a list of products ordered by customers from the state 'Karnataka' with price greater than 10,000 (use INNER JOIN between sales, customers, and products).
* List all customers who have placed orders where the product category is 'Accessories' and the order status is 'Completed' (use INNER JOIN with sales, customers, and products).
* Show the order details of customers who have paid for their orders, excluding those who have cancelled their orders (use INNER JOIN between sales and payments and apply WHERE for order\_status).
* Retrieve products ordered by customers who are in the 'Gujarat' state and whose total order price is greater than 15,000 (use INNER JOIN between sales, customers, and products).

## 

## **Joins + Group BY + Having**

* Find the total quantity of each product ordered by customers from 'Delhi' and only include products with a quantity greater than 5 (use INNER JOIN with sales, customers, and products and group by product).
* Get the average payment amount per customer who has placed more than 3 orders (use INNER JOIN between paymentsand sales, group by customer, and apply a HAVING clause).
* Retrieve the total sales for each product category and only include categories where the total sales exceed 100,000 (use INNER JOIN between sales and products, group by category).
* Show the number of customers in each state who have made purchases with a total spend greater than 50,000 (use INNER JOIN between sales and customers).
* List the total sales by brand for products that have been ordered more than 10 times (use INNER JOIN between salesand products, group by brand).

## 

## **Joins + WHERE + Group BY + HAVING + ORDER BY**

* Retrieve the total sales per customer in 'Delhi' where the order status is 'Completed', only include those with total sales greater than 50,000, and order the results by total sales (use INNER JOIN between sales and customers).
* Show the total quantity sold per product in the 'Accessories' category where the total quantity sold is greater than 50 and order the results by product name (use INNER JOIN between sales and products).
* Find the total number of orders for customers from 'Maharashtra' who have spent more than 1,00,000, and order the results by the total amount spent (use INNER JOIN between sales and customers).
* Get the number of orders per product and filter to include only products that have been ordered more than 10 times, then order the results by the highest number of orders (use INNER JOIN between sales and products).
* Retrieve the number of payments made per customer where the payment status is 'Payment Successed' and group by customer, ordering by payment count (use INNER JOIN between payments and customers).

## 

## **DATE FUNCTIONS**

* List all orders that were placed within the year 2023 (use order\_date with the EXTRACT function).
* Retrieve customers who have made purchases in the month of January (use order\_date and TO\_CHAR to extract the month).
* Calculate the number of days between the payment\_date and order\_date for each order (use the AGE function).
* Find the total sales for each year (use EXTRACT with order\_date to group by year).
* Show all orders where the shipping date is after the payment date (use date comparison).