

SQL QUERIES

1. Display all customer details.
SYNTAX:
SELECT * FROM Customers
2. List all product names and their prices
SYNTAX :
SELECT productname, price FROM Products
3. Show customers from India only.
SYNTAX:
SELECT customername FROM Customers
WHERE country = 'India' ;
4. Find all orders placed after June 2024.
SYNTAX:
SELECT orderid FROM Orders
where orderdate > '2024-06-30';
5. Show the total number of customers.
SYNTAX:
select COUNT(*) as TotalCustomer from Customers;
6. Show the highest product price.
SYNTAX:
select productname,max(price) as HighestProductprice from Products;
7. Display the average price of all products
SYNTAX:
SELECT AVG(Price) AS Average_Price FROM Products;
8. Sort customers alphabetically by name.
SYNTAX:
select * from Customers
order by customername ASC;
9. Show all products priced between ₹1000 and ₹10000.
SYNTAX:
Select * from Products
WHERE price BETWEEN 1000 and 10000;
10. List distinct cities from which customers belong.
SYNTAX:
select * from Customers
order by city
11. Find total sales (TotalAmount) made by each customer.
SYNTAX:
select c.customername, sum(o.totalamount) from Orders as o
join Customers as c
on c.CustomerID = o.CustomerID
group by customername
12. Show the number of orders placed by each customer.
SYNTAX:
select c.customername , COUNT(orderid) as totalorders from Customers as c
join Orders as o on c.CustomerID = o.CustomerID

GROUP by customername

13. List product names with quantity > 1 in orders.
SYNTAX:
Select productname from OrderDetails as od
join Products p on p.. ProductID = od.ProductID
where quantity >1
14. Display the total number of products sold (sum of quantities).
SYNTAX:
select sum(quantity) as totalquantity from OrderDetails
15. Find the most expensive product.
SYNTAX:
SELECT * FROM Products WHERE Price = (SELECT MAX(Price) FROM Products);
16. Show customer names and their corresponding order IDs.
SYNTAX:
SELECT customername,orderid from Customers c
Join Orders on c.CustomerID = o.CustomerID
17. Find all products ordered by "Aarav Shah".
SYNTAX:
SELECT productname from Products p
join OrderDetails od on od.ProductID = p.ProductID
join Orders on o.OrderID = od.OrderID
join Customers c on c.CustomerID = o.CustomerID
where customername = 'Aarav Shah';
18. List the top 3 highest order amounts.
SYNTAX:
select DISTINCT totalamount from Orders
order by totalamount DESC
limit 3
19. Find orders where the total amount > 10,000.
SYNTAX:
select * from Orders
where totalamount>10000
20. Display all customers who have not placed any order (use LEFT JOIN).
SYNTAX:
SELECT c.CustomerName
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE o.OrderID IS NULL;
21. Show each product's total quantity sold.
SYNTAX:
SELECT sum(quantity), productname, price from Products p
join OrderDetails od on od.ProductID = p.ProductID
group by productname

22. Find total revenue per product.
 SYNTAX:
 SELECT sum(quantity*price) as revenue, productname from Products p
 join OrderDetails od on od.ProductID = p.ProductID
 group by productname
23. Find total revenue per category.
 SYNTAX:
 select category, sum(quantity*price) as totalrevenue from OrderDetails od
 inner join Products p
 on p.ProductID=od.ProductID
 GROUP by category
24. Display the top-selling product by quantity.
 SYNTAX:
 select productname, sum(quantity) as totalquantity FROM OrderDetails as od
 join Products as p on p.. ProductID=od.ProductID
 group by productname
 order by totalquantity DESC
 limit 1
25. Display the total number of customers per city.
 SYNTAX:
 select city , COUNT(customerid) FROM Customers
 group by city;
26. Find month-wise total revenue (June vs July).
 SYNTAX:
 select strftime('%m',orderdate) as month,sum(totalamount) as totalrevenue from Orders
 group by month;
27. Use CASE to categorize orders:
- ₹50,000 → 'High Value'
 - ₹10,000–₹50,000 → 'Medium Value'
 - < ₹10,000 → 'Low Value'
- SYNTAX:
 Select orderid, totalamount,
 CASE
 when totalamount >50000 THEN 'high value'
 when totalamount BETWEEN 10000 and 50000 THEN 'medium value'
 ELSE 'low value'
 end as ordercategory
 from Orders;
28. Rank customers by total spending.
 SYNTAX:
 select customername ,sum(totalamount) as totalspending,
 rank() over (order by sum(totalamount) DESC)
 from Customers as c
 join orders as o on o.CustomerID=c.CustomerID
 GROUP by customername

29. Find the average quantity per order.

SYNTAX:

```
select avg(quantity) as avgquantityperorder from OrderDetails;
```

30. Display all products that were never sold (NOT IN OrderDetails).

SYNTAX:

```
select productname from Products as p
join OrderDetails od on p.. ProductID = od.ProductID
where p.ProductID <> od.ProductID
or
SELECT productname FROM Products
where productid not in
(select DISTINCT productid from OrderDetails)
```