Employees Table

EmployeeID	FirstName	LastName	DepartmentID	Salary	HireDate	ManagerID
1	Alice	Johnson	101	60000	2018-01-15	3
2	Bob	Smith	102	75000	2017-05-20	3
3	Charlie	Brown	101	90000	2015-09-30	NULL
4	David	Williams	103	55000	2019-07-11	3
5	Eva	Davis	102	65000	2020-03-25	2

Orders Table

OrderID	EmployeeID	ProductID	Quantity	OrderDate
1001	1	201	10	2022-01-15
1002	2	202	5	2022-01-16
1003	3	203	20	2022-01-17
1004	4	202	15	2022-01-18
1005	5	204	25	2022-01-19

Products Table

ProductID	ProductName	Price	Category
201	Laptop	1200	Electronics

ProductID	ProductName	Price	Category
202	Smartphone	800	Electronics
203	Office Chair	150	Furniture
204	Desk	300	Furniture
205	Monitor	200	Electronics

SQL Queries Interview Questions

Q1. Write a query to display all records from the Employees table.

Answer:

SELECT * FROM Employees;Copy Code

```
EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate   | ManagerID
--------------------------------------------------------------------------------
1          | Alice     | Johnson  | 10           | 160000 | 2018-01-15 | 32
2          | Bob       | Smith    | 10           | 275000 | 2017-05-20 | 33
3          | Charlie   | Brown    | 10           | 190000 | 2015-09-30 | NULL
4          | David     | Williams | 10           | 355000 | 2019-07-11 | 35
5          | Eva       | Davis    | 10           | 265000 | 2020-03-25 | 32Copy Code
```

Q2. Fetch only the FirstName and LastName of employees.

Answer:

SELECT FirstName, LastName FROM Employees;

```
FirstName | LastName
```

```
--------------------
Alice     | Johnson
Bob       | Smith
Charlie   | Brown
David     | Williams
Eva       | Davis
```

Copy Code

Q3. Retrieve the unique department IDs from the Employees table.

Answer:

```
SELECT DISTINCT DepartmentID FROM Employees;
```

```
DepartmentID
-------------
10
```

Copy Code

Q4. Fetch employees with a salary greater than 60,000.

Answer:

```
SELECT * FROM Employees WHERE Salary > 60000;
```

```
EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate    | ManagerID
--------------------------------------------------------------------------------
```

```
1         | Alice    | Johnson  | 10         | 160000 | 2018-01-15  | 32
2         | Bob      | Smith    | 10         | 275000 | 2017-05-20  | 33
3         | Charlie  | Brown    | 10         | 190000 | 2015-09-30  | NULL
4         | David    | Williams | 10         | 355000 | 2019-07-11  | 35
5         | Eva      | Davis    | 10         | 265000 | 2020-03-25  | 32
```

Copy Code

Q5. Write a query to display all orders placed on or after January 17, 2022.

Answer:

```
SELECT * FROM Orders WHERE OrderDate >= '2022-01-17';
```

```
OrderID | EmployeeID | ProductID | Quantity | OrderDate
--------------------------------------------------------
1022    | 2          | 1         | 2        | 2022-01-16
1023    | 3          | 3         | 3        | 2022-01-17
1024    | 4          | 2         | 5        | 2022-01-18
1025    | 5          | 4         | 5        | 2022-01-19
```

Copy Code

Q6. Retrieve all products with a price less than 300.

Answer:

```
SELECT * FROM Products WHERE Price < 300;
```

```
ProductID | ProductName | Price | Category

--------------------------------------------

203       | Office Chair | 150   | Furniture

204       | Desk        | 300   | Furniture

205       | Monitor     | 200   | Electronics
```

Copy Code

Q7. Find the total number of orders in the Orders table.

Answer:

```
SELECT COUNT(*) AS TotalOrders FROM Orders;
```

```
TotalOrders

------------

5
```

Copy Code

Q8. Fetch the details of the product named ‘Laptop’.

Answer:

```
SELECT * FROM Products WHERE ProductName = 'Laptop';
```

```
ProductID | ProductName | Price | Category

--------------------------------------------

201       | Laptop      | 1200  | Electronics
```

Copy Code

Q9. Write a query to sort employees by their HireDate in ascending order.

Answer:

```
SELECT * FROM Employees ORDER BY HireDate ASC;
```

```
EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate   | ManagerID
--------------------------------------------------------------------------------
3          | Charlie   | Brown    | 10           | 190000 | 2015-09-30 | NULL
2          | Bob       | Smith    | 10           | 275000 | 2017-05-20 | 33
1          | Alice     | Johnson  | 10           | 160000 | 2018-01-15 | 32
4          | David     | Williams | 10           | 355000 | 2019-07-11 | 35
5          | Eva       | Davis    | 10           | 265000 | 2020-03-25 | 32
```

Copy Code

Q10. Retrieve the maximum price of products in the Electronics category.

Answer:

```
SELECT MAX(Price) AS MaxPrice FROM Products WHERE Category = 'Electronics';
```

```
MaxPrice
--------
1200
```

Copy Code

Q11. Write a query to join Employees and Orders tables to fetch employee names along with their orders.

Answer:

```
SELECT e.FirstName, e.LastName, o.OrderID, o.OrderDate

FROM Employees e

JOIN Orders o ON e.EmployeeID = o.EmployeeID;
```

```
FirstName | LastName | OrderID | OrderDate

--------------------------------------------

Alice     | Johnson  | 1022    | 2022-01-16

Bob       | Smith    | 1023    | 2022-01-17

Charlie   | Brown    | 1024    | 2022-01-18

David     | Williams | 1025    | 2022-01-19
```

Copy Code

Q12. Calculate the total salary by department.

Answer:

```
SELECT DepartmentID, SUM(Salary) AS TotalSalary

FROM Employees

GROUP BY DepartmentID;
```

```
DepartmentID | TotalSalary

-------------------------

10           | 1355000
```

Copy Code

Q13. Find the employees who do not have a manager.

Answer:

```
SELECT * FROM Employees WHERE ManagerID IS NULL;
```

```
EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate    | ManagerID
-------------------------------------------------------------------------------
3          | Charlie   | Brown    | 10           | 190000 | 2015-09-30  | NULL
```

Copy Code

Q14. Write a query to display the average product price for each category.

Answer:

```
SELECT Category, AVG(Price) AS AvgPrice

FROM Products

GROUP BY Category;
```

```
Category    | AvgPrice
-----------------------
Electronics | 800
Furniture   | 216.67
```

Copy Code

Q15. Fetch the details of the top 3 highest-paid employees.

Answer:

```
SELECT * FROM Employees

ORDER BY Salary DESC

LIMIT 3;
```

```
EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate    | ManagerID

---------------------------------------------------------------------------

4         | David    | Williams | 10          | 355000 | 2019-07-11  | 35

2         | Bob      | Smith    | 10          | 275000 | 2017-05-20  | 33

5         | Eva      | Davis    | 10          | 265000 | 2020-03-25  | 32
```

Copy Code

Q16. Retrieve the order details along with the product name.

Answer:

```
SELECT o.OrderID, o.Quantity, p.ProductName, p.Price

FROM Orders o

JOIN Products p ON o.ProductID = p.ProductID;
```

```
OrderID | Quantity | ProductName  | Price

-------------------------------------------

1022    | 2       | Laptop       | 1200

1023    | 3       | Office Chair | 150
```

```
1024    | 5       | Smartphone   | 800

1025    | 5       | Desk        | 300
```

Copy Code

Q17. Find the total quantity of products ordered for each product.

Answer:

```
SELECT ProductID, SUM(Quantity) AS TotalQuantity

FROM Orders

GROUP BY ProductID;
```

```
ProductID | TotalQuantity

--------------------------

1        | 2

2        | 8

3        | 3

4        | 5
```

Copy Code

Q18. Write a query to update the price of all Furniture category products by 10%.

Answer:

```
UPDATE Products

SET Price = Price * 1.10

WHERE Category = 'Furniture';Copy Code
```

Q19. Delete all orders placed before January 17, 2022.

Answer:

```
DELETE FROM Orders WHERE OrderDate < '2022-01-17';Copy Code
```

Q20. Fetch employees whose first name starts with ‘A’.

Answer:

```
SELECT * FROM Employees WHERE FirstName LIKE 'A%';


EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate   | ManagerID
----------------------------------------------------------------------------
1          | Alice     | Johnson  | 10           | 160000 | 2018-01-15 | 32
```

Copy Code

Q21. Retrieve the number of employees hired each year.

Answer:

```
SELECT YEAR(HireDate) AS HireYear, COUNT(*) AS EmployeesHired
FROM Employees
GROUP BY YEAR(HireDate);


HireYear | EmployeesHired
-------------------------
2015     | 1
2017     | 1
```

```
2018    | 1

2019    | 1

2020    | 1
```

Copy Code

Q22. Write a query to fetch employees earning more than the average salary.

Answer:

```
SELECT * FROM Employees

WHERE Salary > (SELECT AVG(Salary) FROM Employees);
```

```
EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate    | ManagerID

---------------------------------------------------------------------------

2          | Bob       | Smith    | 10           | 275000 | 2017-05-20  | 33

4          | David     | Williams | 10           | 355000 | 2019-07-11  | 35

5          | Eva       | Davis    | 10           | 265000 | 2020-03-25  | 32
```

Copy Code

Q23. Display the top 3 products with the highest total quantity sold.

Answer:

```
SELECT p.ProductName, SUM(o.Quantity) AS TotalQuantity

FROM Orders o

JOIN Products p ON o.ProductID = p.ProductID

GROUP BY p.ProductName
```

```
ORDER BY TotalQuantity DESC

LIMIT 3;
```

```
ProductName | TotalQuantity

----------------------------

Smartphone  | 8

Desk        | 5

Office Chair| 3
```

Copy Code

Q24. Retrieve the employees who have not placed any orders.

Answer:

```
SELECT * FROM Employees

WHERE EmployeeID NOT IN (SELECT DISTINCT EmployeeID FROM Orders);
```

```
EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate    | ManagerID

-----------------------------------------------------------------------------

3          | Charlie   | Brown    | 10           | 190000 | 2015-09-30  | NULL
```

Copy Code

Q25. Write a query to fetch the most recently hired employee.

Answer:

```
SELECT * FROM Employees
```

```
ORDER BY HireDate DESC

LIMIT 1;


EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate   | ManagerID

---------------------------------------------------------------------------

5         | Eva      | Davis    | 10          | 265000 | 2020-03-25  | 32
```

Copy Code

Q26. Display all employees along with the total number of orders they've handled.

Answer:

```
SELECT e.EmployeeID, e.FirstName, COUNT(o.OrderID) AS TotalOrders

FROM Employees e

LEFT JOIN Orders o ON e.EmployeeID = o.EmployeeID

GROUP BY e.EmployeeID, e.FirstName;Copy Code
```

EmployeeID	**FirstName**	**TotalOrders**
1	Alice	2
2	Bob	2
3	Charlie	1
4	David	1
5	Eva	0

Q27. Fetch product details for which total sales exceed $10,000.

Answer:

```
SELECT p.ProductName, SUM(o.Quantity * p.Price) AS TotalSales

FROM Orders o

JOIN Products p ON o.ProductID = p.ProductID

GROUP BY p.ProductName

HAVING TotalSales > 10000;Copy Code
```

ProductName	TotalSales
Laptop	24000

Q28. Find employees who joined the company in the same year as their manager.

Answer:

```
SELECT e.FirstName AS EmployeeName, m.FirstName AS ManagerName

FROM Employees e

JOIN Employees m ON e.ManagerID = m.EmployeeID

WHERE YEAR(e.HireDate) = YEAR(m.HireDate);Copy Code
```

EmployeeName	ManagerName
Alice	Bob

Q29. Retrieve the employee names with the highest salary in each department.

Answer:

```
SELECT DepartmentID, FirstName, LastName, Salary

FROM Employees

WHERE (DepartmentID, Salary) IN (

  SELECT DepartmentID, MAX(Salary)
```

```
    FROM Employees
    GROUP BY DepartmentID
);Copy Code
```

DepartmentID	FirstName	LastName	Salary
1	Alice	Johnson	160000
2	Bob	Smith	75000
3	David	Williams	55000

Q30. Write a query to fetch the total revenue generated by each employee.

Answer:

```
SELECT e.FirstName, e.LastName, SUM(o.Quantity * p.Price) AS TotalRevenue
FROM Employees e
JOIN Orders o ON e.EmployeeID = o.EmployeeID
JOIN Products p ON o.ProductID = p.ProductID
GROUP BY e.EmployeeID, e.FirstName, e.LastName;Copy Code
```

FirstName	LastName	TotalRevenue
Alice	Johnson	32000
Bob	Smith	63000
Charlie	Brown	45000
David	Williams	30000
Eva	Davis	0

Q31. Write a query to fetch employees earning more than their manager.

Answer:

```
SELECT e.FirstName AS EmployeeName, m.FirstName AS ManagerName

FROM Employees e

JOIN Employees m ON e.ManagerID = m.EmployeeID

WHERE e.Salary > m.Salary;Copy Code
```

EmployeeName	ManagerName
Alice	Bob

Q32. Retrieve the second highest salary from the Employees table.

Answer:

```
SELECT MAX(Salary) AS SecondHighestSalary

FROM Employees

WHERE Salary < (SELECT MAX(Salary) FROM Employees);
```

```
SecondHighestSalary

75000Copy Code
```

Q33. List the departments with no employees assigned.

Answer:

```
SELECT * FROM Departments

WHERE DepartmentID NOT IN (SELECT DISTINCT DepartmentID FROM Employees);Copy Code
```

DepartmentID	DepartmentName
4	Marketing

Q34. Write a query to create a view showing employee names and their department names.

Answer:

CREATE VIEW EmployeeDepartmentView AS

SELECT e.FirstName, e.LastName, d.DepartmentName

FROM Employees e

JOIN Departments d ON e.DepartmentID = d.DepartmentID;Copy Code

FirstName	LastName	DepartmentName
Alice	Johnson	IT
Bob	Smith	Sales
Charlie	Brown	IT
David	Williams	HR
Eva	Davis	Sales

Q35. Fetch the names of employees who have placed more than 10 orders.

Answer:

SELECT e.FirstName, e.LastName

FROM Employees e

JOIN Orders o ON e.EmployeeID = o.EmployeeID

GROUP BY e.EmployeeID, e.FirstName, e.LastName

HAVING COUNT(o.OrderID) > 10;Copy Code

FirstName	LastName
Alice	Johnson
Bob	Smith

Q36. Write a query to rank employees based on their salary within each department.

Answer:

```
SELECT EmployeeID, FirstName, DepartmentID, Salary,

    RANK() OVER (PARTITION BY DepartmentID ORDER BY Salary DESC) AS Rank

FROM Employees;Copy Code
```

EmployeeID	FirstName	DepartmentID	Salary	Rank
1	Alice	1	160000	1
3	Charlie	1	190000	2
2	Bob	2	75000	1
4	David	3	55000	1
5	Eva	2	65000	2

Q37. Retrieve the cumulative sales for each product.

Answer:

```
SELECT ProductID, ProductName,

    SUM(SUM(Quantity * Price)) OVER (ORDER BY ProductID) AS CumulativeSales

FROM Products p
```

JOIN Orders o ON p.ProductID = o.ProductID

GROUP BY ProductID, ProductName;Copy Code

ProductID	ProductName	CumulativeSales
201	Laptop	24000
202	Smartphone	32000
203	Office Chair	1500
204	Desk	3000
205	Monitor	1500

Q38. Identify the department with the highest total salary expenditure.

Answer:

SELECT DepartmentID, SUM(Salary) AS TotalExpenditure

FROM Employees

GROUP BY DepartmentID

ORDER BY TotalExpenditure DESC

LIMIT 1;Copy Code

DepartmentID	TotalExpenditure
1	450000

Q39. Write a query to find the percentage contribution of each product to total sales.

Answer:

SELECT p.ProductName,

```
    (SUM(o.Quantity * p.Price) * 100.0 /

     (SELECT SUM(Quantity * Price) FROM Orders o JOIN Products p ON o.ProductID = p.ProductID)) AS ContributionPercentage

FROM Orders o

JOIN Products p ON o.ProductID = p.ProductID

GROUP BY p.ProductName;Copy Code
```

ProductName	**ContributionPercentage**
Laptop	48.00
Smartphone	32.00
Office Chair	4.00
Desk	8.00
Monitor	8.00

Q40. Find employees who have the same manager and earn more than $70,000.

Answer:

```
SELECT *

FROM Employees e1

WHERE ManagerID IS NOT NULL

AND Salary > 70000

AND ManagerID IN (

  SELECT ManagerID FROM Employees e2 WHERE e1.ManagerID = e2.ManagerID

);Copy Code
```

EmployeeID	FirstName	LastName	Salary	ManagerID
1	Alice	Johnson	160000	32
2	Bob	Smith	75000	32

Q41. Write a query to detect duplicate rows in the Orders table.

Answer:

SELECT EmployeeID, ProductID, OrderDate, COUNT(*) AS DuplicateCount

FROM Orders

GROUP BY EmployeeID, ProductID, OrderDate

HAVING COUNT(*) > 1;Copy Code

EmployeeID	ProductID	OrderDate	DuplicateCount
1	201	2022-01-15	2

Q42. Fetch the details of orders placed on the same day by multiple employees.

Answer:

SELECT OrderDate, COUNT(DISTINCT EmployeeID) AS EmployeeCount

FROM Orders

GROUP BY OrderDate

HAVING EmployeeCount > 1;Copy Code

OrderDate	EmployeeCount
2022-01-15	2
2022-01-16	2

OrderDate	EmployeeCount
2022-01-17	1

Q43. Create a stored procedure to update product prices based on category.

Answer:

```
DELIMITER $$

CREATE PROCEDURE UpdatePriceByCategory(IN category_name VARCHAR(50), IN price_factor DECIMAL(5, 2))

BEGIN

   UPDATE Products

   SET Price = Price * price_factor

   WHERE Category = category_name;

END$$

DELIMITER ;Copy Code
```

Q44. Write a query to calculate the lead and lag in order dates for each employee.

Answer:

```
SELECT EmployeeID, OrderID, OrderDate,

     LAG(OrderDate) OVER (PARTITION BY EmployeeID ORDER BY OrderDate) AS PreviousOrderDate,

     LEAD(OrderDate) OVER (PARTITION BY EmployeeID ORDER BY OrderDate) AS NextOrderDate

FROM Orders;Copy Code
```

EmployeeID	OrderID	OrderDate	PreviousOrderDate	NextOrderDate
1	1	2022-01-15	NULL	2022-01-16

EmployeeID	OrderID	OrderDate	PreviousOrderDate	NextOrderDate
2	2	2022-01-16	2022-01-15	2022-01-17
3	3	2022-01-17	NULL	NULL

Q45. Identify the products that have not been ordered.

Answer:

```
SELECT * FROM Products

WHERE ProductID NOT IN (SELECT DISTINCT ProductID FROM Orders);Copy Code
```

ProductID	ProductName
204	Desk
205	Monitor

Q46. Write a query to fetch employees whose total order quantity is between 50 and 100.

Answer:

```
SELECT e.FirstName, e.LastName, SUM(o.Quantity) AS TotalQuantity

FROM Employees e

JOIN Orders o ON e.EmployeeID = o.EmployeeID

GROUP BY e.EmployeeID, e.FirstName, e.LastName

HAVING TotalQuantity BETWEEN 50 AND 100;Copy Code
```

FirstName	LastName	TotalQuantity
Bob	Smith	60

Q47. Fetch the second-highest quantity ordered for each product.

Answer:

SELECT ProductID, MAX(Quantity) AS SecondHighestQuantity

FROM Orders

WHERE Quantity < (SELECT MAX(Quantity) FROM Orders WHERE Orders.ProductID = ProductID)

GROUP BY ProductID;Copy Code

ProductID	SecondHighestQuantity
201	20
202	30
203	10

Q48. Find the minimum and maximum order quantities for each employee.

Answer:

SELECT EmployeeID, MIN(Quantity) AS MinQuantity, MAX(Quantity) AS MaxQuantity

FROM Orders

GROUP BY EmployeeID;Copy Code

EmployeeID	MinQuantity	MaxQuantity
1	10	20
2	20	40
3	10	10

Q49. Write a query to split employee salaries into quartiles.

Answer:

SELECT EmployeeID, FirstName, Salary,

NTILE(4) OVER (ORDER BY Salary) AS SalaryQuartile

FROM Employees;Copy Code

EmployeeID	FirstName	Salary	SalaryQuartile
1	Alice	160000	4
2	Bob	75000	3
3	Charlie	190000	4
4	David	55000	2
5	Eva	65000	2

Q50. Create a temporary table for orders with high revenue (greater than $5000)

Answer:

CREATE TEMPORARY TABLE HighRevenueOrders AS

SELECT o.OrderID, o.Quantity, p.Price, (o.Quantity * p.Price) AS Revenue

FROM Orders o

JOIN Products p ON o.ProductID = p.ProductID

WHERE (o.Quantity * p.Price) > 5000;Copy Code

OrderID	Quantity	Price	Revenue
1	10	1200	12000
2	25	800	20000

Conclusion