**16-06-2025**

**HEXAWARE PRACTICE QUESTION PAPER**

**Part A – Subqueries (20 marks)**

1. Customers who have placed orders in every month of the current year

SELECT Name

FROM Customers c

WHERE NOT EXISTS (

SELECT DISTINCT MONTH(OrderDate)

FROM Orders

WHERE YEAR(OrderDate) = YEAR(CURDATE())

EXCEPT

SELECT DISTINCT MONTH(o.OrderDate)

FROM Orders o

WHERE o.CustomerID = c.CustomerID AND YEAR(o.OrderDate) = YEAR(CURDATE())

);

2. Products ordered more than average quantity

SELECT ProductName

FROM Products p

WHERE ProductID IN (

SELECT ProductID

FROM OrderDetails

GROUP BY ProductID

HAVING SUM(Quantity) > (

SELECT AVG(total\_qty)

FROM (

SELECT SUM(Quantity) AS total\_qty

FROM OrderDetails

GROUP BY ProductID

) AS avg\_qty

)

);

3. Customers who never ordered product priced above ₹1000

SELECT Name

FROM Customers c

WHERE CustomerID NOT IN (

SELECT DISTINCT o.CustomerID

FROM Orders o

JOIN OrderDetails od ON o.OrderID = od.OrderID

JOIN Products p ON od.ProductID = p.ProductID

WHERE p.Price > 1000

);

4. Top 3 products by total revenue

SELECT ProductName

FROM Products

WHERE ProductID IN (

SELECT ProductID

FROM (

SELECT ProductID, SUM(Quantity \* Price) AS Revenue

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY ProductID

ORDER BY Revenue DESC

LIMIT 3

) AS top\_products

);

5. Orders that contain only one product

SELECT OrderID

FROM Orders o

WHERE 1 = (

SELECT COUNT(\*)

FROM OrderDetails od

WHERE od.OrderID = o.OrderID

);

**Part B – Correlated & Nested Subqueries (25 marks)**

6. Customers who placed order on the same date as 'John'

SELECT DISTINCT c.Name

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

WHERE o.OrderDate IN (

SELECT o2.OrderDate

FROM Orders o2

JOIN Customers c2 ON o2.CustomerID = c2.CustomerID

WHERE c2.Name = 'John'

);

7. Customer who placed the most recent order

SELECT Name

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

WHERE o.OrderDate = (

SELECT MAX(OrderDate) FROM Orders

);

8. Product with the second lowest price

SELECT ProductName

FROM Products

WHERE Price = (

SELECT MIN(Price)

FROM Products

WHERE Price > (

SELECT MIN(Price) FROM Products

)

);

9. Customers who spent more than double average spending

SELECT Name

FROM Customers c

WHERE CustomerID IN (

SELECT o.CustomerID

FROM Orders o

GROUP BY o.CustomerID

HAVING SUM(o.Amount) > 2 \* (

SELECT AVG(total)

FROM (

SELECT SUM(Amount) AS total FROM Orders GROUP BY CustomerID

) AS avg\_amt

)

);

10. Customers with total amount > any customer from Delhi

SELECT Name

FROM Customers c

WHERE CustomerID IN (

SELECT CustomerID

FROM Orders o

GROUP BY CustomerID

HAVING SUM(Amount) > (

SELECT MAX(total)

FROM (

SELECT SUM(o.Amount) AS total

FROM Orders o

JOIN Customers c2 ON o.CustomerID = c2.CustomerID

WHERE c2.City = 'Delhi'

GROUP BY c2.CustomerID

) AS delhi\_customers

)

);

**Part C – Join + Subquery Mix (30 marks)**

11. Customers who placed more orders than the average

SELECT Name

FROM Customers c

WHERE (

SELECT COUNT(\*) FROM Orders o WHERE o.CustomerID = c.CustomerID

) > (

SELECT AVG(order\_count)

FROM (

SELECT COUNT(\*) AS order\_count FROM Orders GROUP BY CustomerID

) AS avg\_orders

);

12. Products with total sales quantity > average per product

SELECT ProductName

FROM Products p

WHERE ProductID IN (

SELECT ProductID

FROM OrderDetails

GROUP BY ProductID

HAVING SUM(Quantity) > (

SELECT AVG(total\_qty)

FROM (

SELECT SUM(Quantity) AS total\_qty

FROM OrderDetails

GROUP BY ProductID

) AS avg\_qty

)

);

13. Customers who ordered unique products

SELECT DISTINCT c.Name

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

JOIN OrderDetails od ON o.OrderID = od.OrderID

WHERE od.ProductID IN (

SELECT ProductID

FROM OrderDetails

GROUP BY ProductID

HAVING COUNT(DISTINCT OrderID) = 1

);

14. Orders with total = customer's max

SELECT o.OrderID

FROM Orders o

WHERE o.Amount = (

SELECT MAX(Amount)

FROM Orders

WHERE CustomerID = o.CustomerID

);

15. Customers who never ordered quantity > 5

SELECT Name

FROM Customers c

WHERE CustomerID NOT IN (

SELECT DISTINCT o.CustomerID

FROM Orders o

JOIN OrderDetails od ON o.OrderID = od.OrderID

WHERE od.Quantity > 5

);

**Part D – Joins & Set Operations (25 marks)**

16. Top 5 customers by total spending

SELECT Name

FROM Customers

WHERE CustomerID IN (

SELECT CustomerID

FROM Orders

GROUP BY CustomerID

ORDER BY SUM(Amount) DESC

LIMIT 5

);

17. Customers who ordered only one unique product

SELECT Name

FROM Customers c

WHERE CustomerID IN (

SELECT o.CustomerID

FROM Orders o

JOIN OrderDetails od ON o.OrderID = od.OrderID

GROUP BY o.CustomerID

HAVING COUNT(DISTINCT od.ProductID) = 1

);

18. Orders not in top 10 highest amounts

SELECT \*

FROM Orders

WHERE Amount NOT IN (

SELECT Amount

FROM Orders

ORDER BY Amount DESC

LIMIT 10

);

19. Customers who ordered in last 7 days but not in previous 30

SELECT DISTINCT c.Name

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

WHERE o.OrderDate BETWEEN CURDATE() - INTERVAL 7 DAY AND CURDATE()

AND c.CustomerID NOT IN (

SELECT CustomerID

FROM Orders

WHERE OrderDate BETWEEN CURDATE() - INTERVAL 37 DAY AND CURDATE() - INTERVAL 8 DAY

);

20. Products ordered in highest number of distinct orders

SELECT ProductName

FROM Products

WHERE ProductID IN (

SELECT ProductID

FROM OrderDetails

GROUP BY ProductID

HAVING COUNT(DISTINCT OrderID) = (

SELECT MAX(order\_count)

FROM (

SELECT COUNT(DISTINCT OrderID) AS order\_count

FROM OrderDetails

GROUP BY ProductID

) AS counts

)

);