# Hexaware SQL Assignment – Part A

## Database and Table Setup

-- Create Database  
CREATE DATABASE IF NOT EXISTS HexawareDB;  
USE HexawareDB;  
  
-- Create Tables  
CREATE TABLE IF NOT EXISTS Customers (  
 CustomerID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 City VARCHAR(100)  
);  
  
CREATE TABLE IF NOT EXISTS Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 Amount DECIMAL(10,2),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
  
CREATE TABLE IF NOT EXISTS Products (  
 ProductID INT PRIMARY KEY,  
 ProductName VARCHAR(100),  
 Price DECIMAL(10,2)  
);  
  
CREATE TABLE IF NOT EXISTS OrderDetails (  
 OrderDetailID INT PRIMARY KEY,  
 OrderID INT,  
 ProductID INT,  
 Quantity INT,  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
 FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);

## Part A – Subqueries (20 Marks)

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

SELECT c.Name  
FROM Customers c  
WHERE NOT EXISTS (  
 SELECT DISTINCT MONTH(OrderDate) AS month\_num  
 FROM Orders  
 WHERE YEAR(OrderDate) = YEAR(CURDATE())  
 EXCEPT  
 SELECT DISTINCT MONTH(o.OrderDate)  
 FROM Orders o  
 WHERE c.CustomerID = o.CustomerID AND YEAR(o.OrderDate) = YEAR(CURDATE())  
);

### 2. Products ordered more than average quantity

SELECT p.ProductName  
FROM Products p  
JOIN OrderDetails od ON p.ProductID = od.ProductID  
GROUP BY p.ProductID, p.ProductName  
HAVING SUM(od.Quantity) > (  
 SELECT AVG(total\_qty) FROM (  
 SELECT SUM(Quantity) AS total\_qty  
 FROM OrderDetails  
 GROUP BY ProductID  
 ) AS avg\_sub  
);

### 3. Customers who never ordered product priced above ₹1000

SELECT Name  
FROM Customers  
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 OrderDetails  
 GROUP BY ProductID  
 ORDER BY SUM(Quantity \* (SELECT Price FROM Products p WHERE p.ProductID = OrderDetails.ProductID)) DESC  
 LIMIT 3  
);

### 5. Orders that contain only one product

SELECT o.OrderID  
FROM Orders o  
WHERE (  
 SELECT COUNT(\*)  
 FROM OrderDetails od  
 WHERE od.OrderID = o.OrderID  
) = 1;