**SQL Practical Answer Documents**

**Advanced Subqueries and Set Operations**

**1) Querying Data by Using Subqueries**

SELECT Name FROM Customers  
WHERE CustomerID IN (SELECT CustomerID FROM Orders WHERE Quantity > 5);

*This retrieves customer names who have placed orders with quantity greater than 5.*

**2) Using the EXISTS Keyword**

SELECT Name FROM Customers c  
WHERE EXISTS (  
 SELECT 1 FROM Orders o WHERE o.CustomerID = c.CustomerID  
);

*Checks for the existence of at least one order per customer.*

**3) Using the ANY Keyword**

SELECT Name FROM Products  
WHERE Price > ANY (  
 SELECT Price FROM Products WHERE Category = 'Electronics'  
);

*Returns products more expensive than at least one in the Electronics category.*

**4) Using the ALL Keyword**

SELECT Name FROM Products  
WHERE Price > ALL (  
 SELECT Price FROM Products WHERE Category = 'Accessories'  
);

*Lists products priced higher than all products in the Accessories category.*

**5) Using Nested Subqueries**

SELECT Name FROM Customers  
WHERE CustomerID IN (  
 SELECT CustomerID FROM Orders  
 WHERE ProductID IN (  
 SELECT ProductID FROM Products WHERE Price > 1000  
 )  
);

*Selects customers who ordered products with price > 1000.*

**6) Using Correlated Subqueries**

SELECT \* FROM Orders o1  
WHERE Quantity > (  
 SELECT AVG(Quantity) FROM Orders o2 WHERE o1.CustomerID = o2.CustomerID  
);

*Returns orders having quantity greater than average quantity by that customer.*

**7) Using UNION**

SELECT Name FROM SegmentA  
UNION  
SELECT Name FROM SegmentB;

*Combines unique names from both segments.*

**8) Using INTERSECT (simulated)**

SELECT Name FROM SegmentA  
WHERE Name IN (SELECT Name FROM SegmentB);

*Finds common names between two segments.*

**9) Using EXCEPT (simulated)**

SELECT Name FROM SegmentA  
WHERE Name NOT IN (SELECT Name FROM SegmentB);

*Returns names from SegmentA not found in SegmentB.*

**10) Using MERGE (Simulated with UPDATE + INSERT)**

-- Update if exists  
UPDATE Customers SET Email = 'updated@domain.com'  
WHERE CustomerID = 1001;  
  
-- Insert if not exists  
INSERT INTO Customers (CustomerID, Name, Email)  
SELECT 1001, 'New Customer', 'updated@domain.com'  
WHERE NOT EXISTS (  
 SELECT 1 FROM Customers WHERE CustomerID = 1001  
);

*Implements upsert (update or insert) behavior similar to MERGE.*