**1 September 2023 Queries**

**Joins/ Union/ Union ALL/ Stored Procedure/Views/ Exercise –**

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100)

);

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

TotalAmount DECIMAL(10, 2),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO Customers (CustomerID, FirstName, LastName, Email)

VALUES

(1, 'John', 'Doe', 'john.doe@example.com'),

(2, 'Jane', 'Smith', 'jane.smith@example.com')

-- Insert data into the Orders table

INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)

VALUES

(1, 1, '2023-08-01', 50.00),

(2, 2, '2023-08-15', 75.00)

SELECT Customers.\*, Orders.\*

FROM Customers

INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

SELECT Customers.\*, Orders.\*

FROM Customers

RIGHT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

SELECT CustomerID FROM Customers

UNION

SELECT CustomerID FROM Orders;

SELECT CustomerID FROM Customers

UNION ALL

SELECT CustomerID FROM Orders;

SELECT CustomerID FROM Customers

INTERSECT

SELECT CustomerID FROM Orders;

SELECT \* FROM Orders WHERE CustomerID IN

(SELECT CustomerID FROM Customers WHERE Email = 'john.doe@example.com');

SELECT \* FROM Customers WHERE CustomerID IN (SELECT CustomerID FROM Orders);

CREATE PROCEDURE GetOrdersByEmail(

@email VARCHAR(100))

AS

SELECT Orders.\*

FROM Customers

INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

WHERE Customers.Email = email;\

Exec GetOrdersByEmail @email = 'john.doe@example.com';

--- Functions ---

CREATE FUNCTION CalculateTotalAmount

(

@OrderID INT

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @TotalAmount DECIMAL(10, 2);

SELECT @TotalAmount = TotalAmount

FROM Orders

WHERE OrderID = @OrderID;

RETURN @TotalAmount;

END;

DECLARE @OrderTotal DECIMAL(10, 2);

SET @OrderTotal = dbo.CalculateTotalAmount(1);

SELECT @OrderTotal AS TotalAmount;

--- Views ---

CREATE VIEW BasicCustomerView AS

SELECT CustomerID, FirstName, LastName

FROM Customers;

CREATE VIEW CustomerOrderView AS

SELECT C.CustomerID, C.FirstName, C.LastName, O.OrderID, O.OrderDate, O.TotalAmount

FROM Customers C

INNER JOIN Orders O ON C.CustomerID = O.CustomerID;

CREATE VIEW CustomerTotalSalesView AS

SELECT C.CustomerID, C.FirstName, C.LastName, SUM(O.TotalAmount) AS TotalSales

FROM Customers C

LEFT JOIN Orders O ON C.CustomerID = O.CustomerID

GROUP BY C.CustomerID, C.FirstName, C.LastName;

--- Indexes ---

CREATE TABLE Products (

ProductID INT,

ProductName VARCHAR(100),

Category VARCHAR(50),

Price DECIMAL(10, 2),

StockQuantity INT

);

INSERT INTO Products (ProductID, ProductName, Category, Price, StockQuantity)

VALUES

(1, 'Product A', 'Electronics', 499.99, 100),

(2, 'Product B', 'Clothing', 39.99, 250),

(3, 'Product C', 'Electronics', 899.99, 50);

CREATE CLUSTERED INDEX IX\_ProductID ON Products (ProductID);

CREATE NONCLUSTERED INDEX IX\_Category ON Products (Category);

CREATE NONCLUSTERED COLUMNSTORE INDEX CS\_Price ON Products (Price);

PTO

**Subqueries Exercise –**

CREATE TABLE Galleries(ID int primary key,City varchar(20));

CREATE TABLE Paintings(ID int primary key,Names varchar(20),Gallery\_ID int, Price int);

INSERT INTO Galleries VALUES

    (1,'Jaipur'),

    (2,'Kolkata'),

    (3,'Madhubani');

INSERT INTO Paintings VALUES

    (1,'Patterns',3,5000),

    (2,'Ringer',1,4500),

    (3,'Gift',1,3200),

    (4,'Violin Lessons',2,6700),

    (5,'Curiosity',2,9800);

CREATE TABLE sales\_agents(ID int,last\_name varchar(20),first\_name varchar(20),gallery\_id int,agency\_fee int);

INSERT INTO sales\_agents VALUES

(1,'Brown','Denis',2,2250),

(2,'White','Kate',3,3120),

(3,'Black ','Sarah',2,1640),

(4,'Smith','Helen',1,4500),

(5,'Stewart','Tom',3,2130);

CREATE TABLE Managers(ID int,gallery\_id int);

INSERT INTO Managers VALUES

    (1,2),

    (2,3),

    (4,1);

Select id,names,price

from paintings

where concat(names,price)

in (Select concat(names,min(price)) from paintings group by names);

Select Galleries.City,C.Cnt

From Galleries,(Select count(\*) as Cnt,gallery\_id from paintings group by gallery\_id) C

Where Galleries.ID=C.gallery\_id;

Select concat(S.first\_name,' ',S.last\_name) Full\_name,S.agency\_fee

From sales\_agents S

Where S.agency\_fee >= (Select avg(agency\_fee)

                        From sales\_agents S2

                        Where S2.gallery\_id = S.gallery\_id);