1 Sept 2023:

AZURE QUERIES:

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 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

LEFT 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';

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;

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;

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);

SELECT \* FROM Products WHERE ProductName = 'Product A';

SELECT \* FROM Products WHERE ProductID = 2;

SELECT \* FROM Products WHERE Category = 'Electronics';

SELECT ProductName FROM Products WHERE Price > 100;

GALLERY QUERIES:

create table galleries1(id int primary key, city varchar(50));

insert into galleries1 values(1,'Jaipur');

insert into galleries1 values(2,'Kolkata');

insert into galleries1 values(3,'Madhubani');

create table paintings1(id int primary key, name varchar(50), gallery\_id int,

FOREIGN KEY (gallery\_id) REFERENCES galleries1(id),

 price int);

insert into paintings1 values(1,'Patterns',3,5000);

insert into paintings1 values(2,'Ringer',1,4500);

insert into paintings1 values(3,'Gift',1,3200);

insert into paintings1 values(4,'Violin Lessons',2,6700);

insert into paintings1 values(5,'Curiosity',2,9800);

create table sales\_agents(id int primary key, last\_name varchar(50), first\_name varchar(50), gallery\_id int,

FOREIGN KEY (gallery\_id) REFERENCES galleries1(id),

 agency\_fee int);

insert into sales\_agents values(1,'Brown','Denis',2,2250);

insert into sales\_agents values(2,'White','Kate',3,3120);

insert into sales\_agents values(3,'Black','Sarah',2,1640);

insert into sales\_agents values(4,'Smith','Helen',1,4500);

insert into sales\_agents values(5,'Stewart','Tom',3,2130);

create table managers(id int primary key, gallery\_id int,FOREIGN KEY (gallery\_id) REFERENCES galleries1(id));

insert into managers values(1,2);

insert into managers values(2,3);

insert into managers values(4,1);

select name, price,(select avg(price) from paintings1) AvgPrice from paintings1;

select \* from  paintings1 where price>(select avg(price) from paintings1);

select avg(agency\_fee) from sales\_agents;

select avg(agency\_fee) from sales\_agents where id not in (select id from managers);

select id,name,price from paintings1 where(concat(name,price))in

(select concat(name,min(price)) from paintings1

group by name

);

select city,(select count(\*) from paintings1 p where g.id=p.gallery\_id) count\_painting from galleries1 g;

select last\_name, first\_name,agency\_fee from sales\_agents sal where sal.agency\_fee>=(select avg(agency\_fee) from sales\_agents sal2 where sal2.gallery\_id=sal.gallery\_id);

select name as painting, price,(Select avg(price) from paintings1 where price in (select price from paintings1 where price>=5000)) as avg\_price from paintings1;

create view vw\_painting as

select name as painting, price,

(Select avg(price) from paintings1

where price in (select price from paintings1 where price>=5000))

as avg\_price from paintings1;

select \* from vw\_painting;