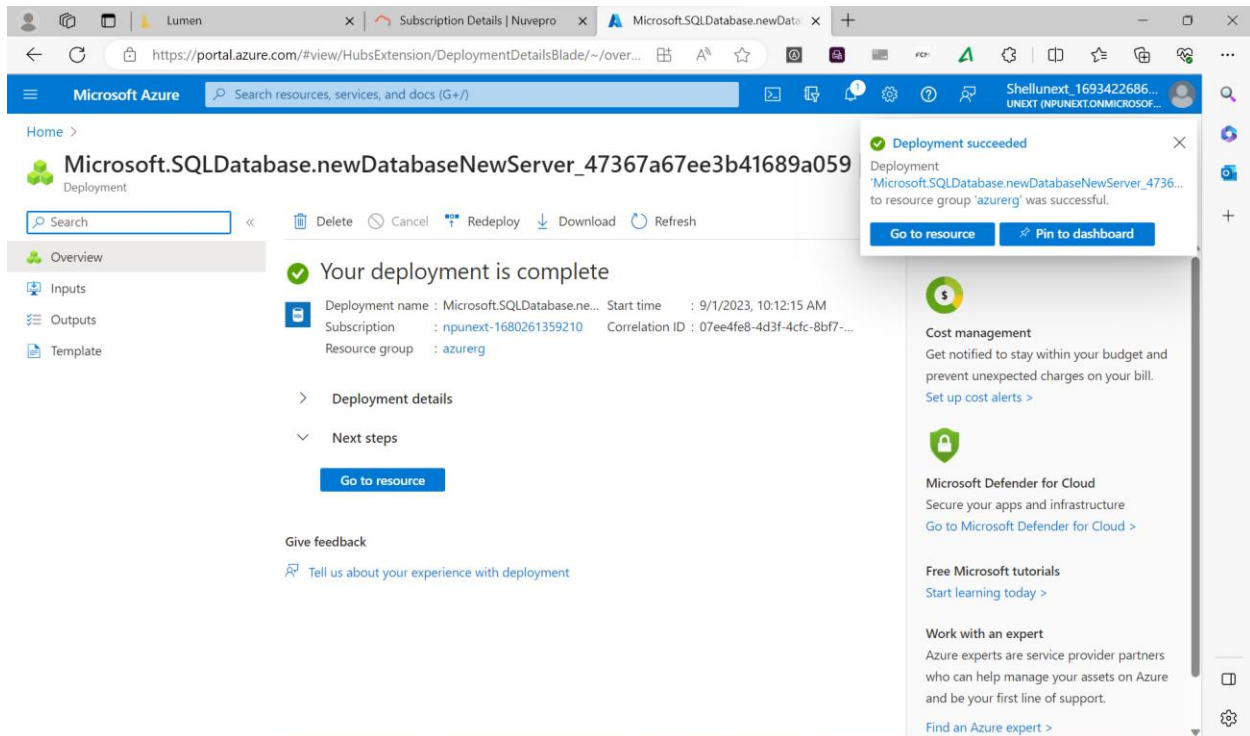


Prerna Singh

DBMS Day-4

Emp Code- 654870

## AZURE SQL



```
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)
```

---Inner Join---

```
SELECT Customers.*, Orders.*
FROM Customers
INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

---Left Join---

```
SELECT Customers.*, Orders.*
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

The screenshot shows the Microsoft Azure portal interface for a SQL database named 'premnadb'. The query editor is open, displaying two SQL queries. The first query is an Inner Join, and the second is a Left Join. The results of the Inner Join query are displayed in a table with columns: CustomerID, FirstName, LastName, Email, OrderID, and OrderDate. The results show two rows: one for John Doe (OrderID 1, OrderDate 2023) and one for Jane Smith (OrderID 2, OrderDate 2023). The status bar at the bottom indicates 'Query succeeded | 0s'.

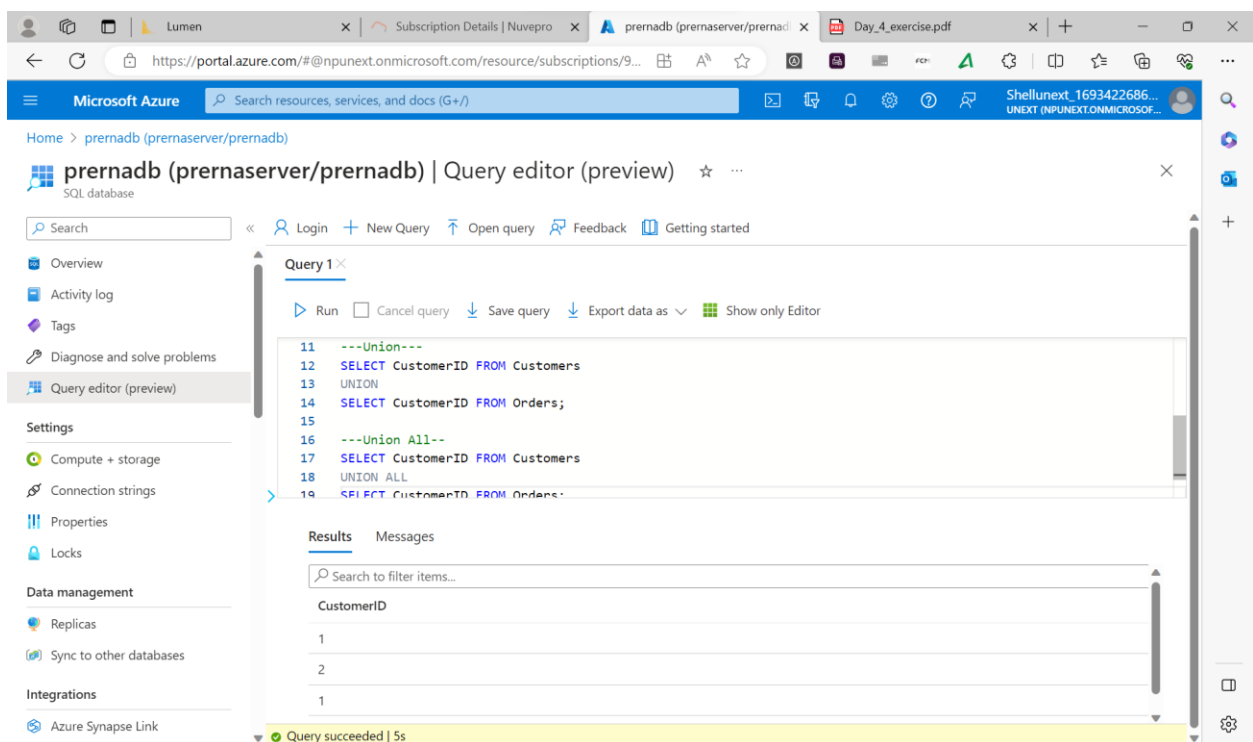
CustomerID	FirstName	LastName	Email	OrderID	OrderDate
1	John	Doe	john.doe@example.c...	1	2023
2	Jane	Smith	jane.smith@example...	2	2023

---Union---

```
SELECT CustomerID FROM Customers
UNION
SELECT CustomerID FROM Orders;
```

---Union All--

```
SELECT CustomerID FROM Customers
UNION ALL
SELECT CustomerID FROM Orders;
```



The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo and a search bar. The main content area is titled 'premnadb (premnaserver/premnadb) | Query editor (preview)'. The left sidebar contains various navigation options: Overview, Activity log, Tags, Diagnose and solve problems, Query editor (preview), Settings, Compute + storage, Connection strings, Properties, Locks, Data management, Replicas, Sync to other databases, Integrations, and Azure Synapse Link. The central pane shows the SQL query editor with the following code:

```
11 ---Union---
12 SELECT CustomerID FROM Customers
13 UNION
14 SELECT CustomerID FROM Orders;
15
16 ---Union All--
17 SELECT CustomerID FROM Customers
18 UNION ALL
19 SELECT CustomerID FROM Orders;
```

Below the query editor, the 'Results' pane is visible, showing a table with the following data:

CustomerID
1
2
1

A status bar at the bottom indicates 'Query succeeded | 5s'.

---Intersect---

```
SELECT CustomerID FROM Customers
SELECT CustomerID FROM Orders;
```

---Subqueries--

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

Microsoft Azure portal interface showing the Query editor for a SQL database named 'premnadb'. The query editor displays a T-SQL query using an INTERSECT operator to find customers who have placed orders. The query is:

```

21 ---Intersect---
22 SELECT CustomerID FROM Customers
23 SELECT CustomerID FROM Orders;
24
25 ---Subqueries---
26 SELECT * FROM Orders WHERE CustomerID IN
27 (SELECT CustomerID FROM Customers WHERE Email = 'john.doe@example.com');
28

```

The results pane shows a single row with the following data:

OrderID	CustomerID	OrderDate	TotalAmount
1	1	2023-08-01T00:00:00.0000000	50.00

The status bar indicates 'Query succeeded | 0s'.

```

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
(
@emailID 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);
```

### **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);