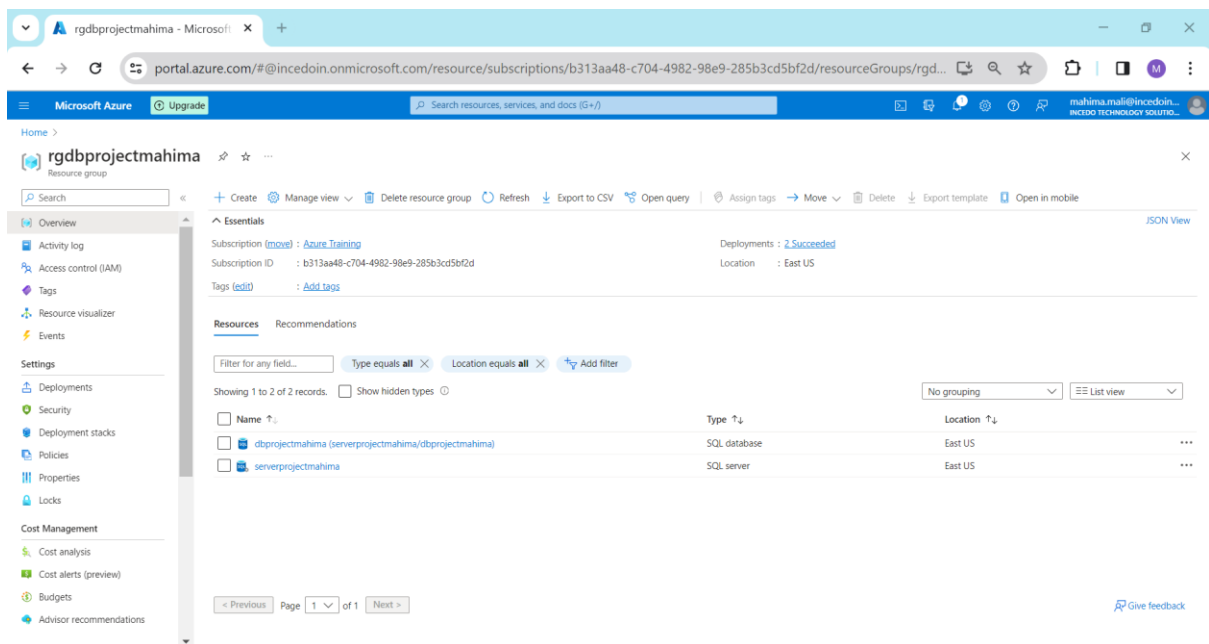


## Project 2: Project on Microsoft Azure & T-SQL

### Module 1: Seamless Data Recovery: A Step-by-Step Guide to Restoring

#### Steps:

1. Login to Azure Portal
2. Navigate to Azure SQL Database
3. Select the Target SQL Database
4. Access the "Restore" Option
5. Configure Restore Options
6. Review and Confirm
7. Validation and Restore
8. Monitor the Restore Progress
9. Verify the Restored Database
10. Connect to the Restored Database



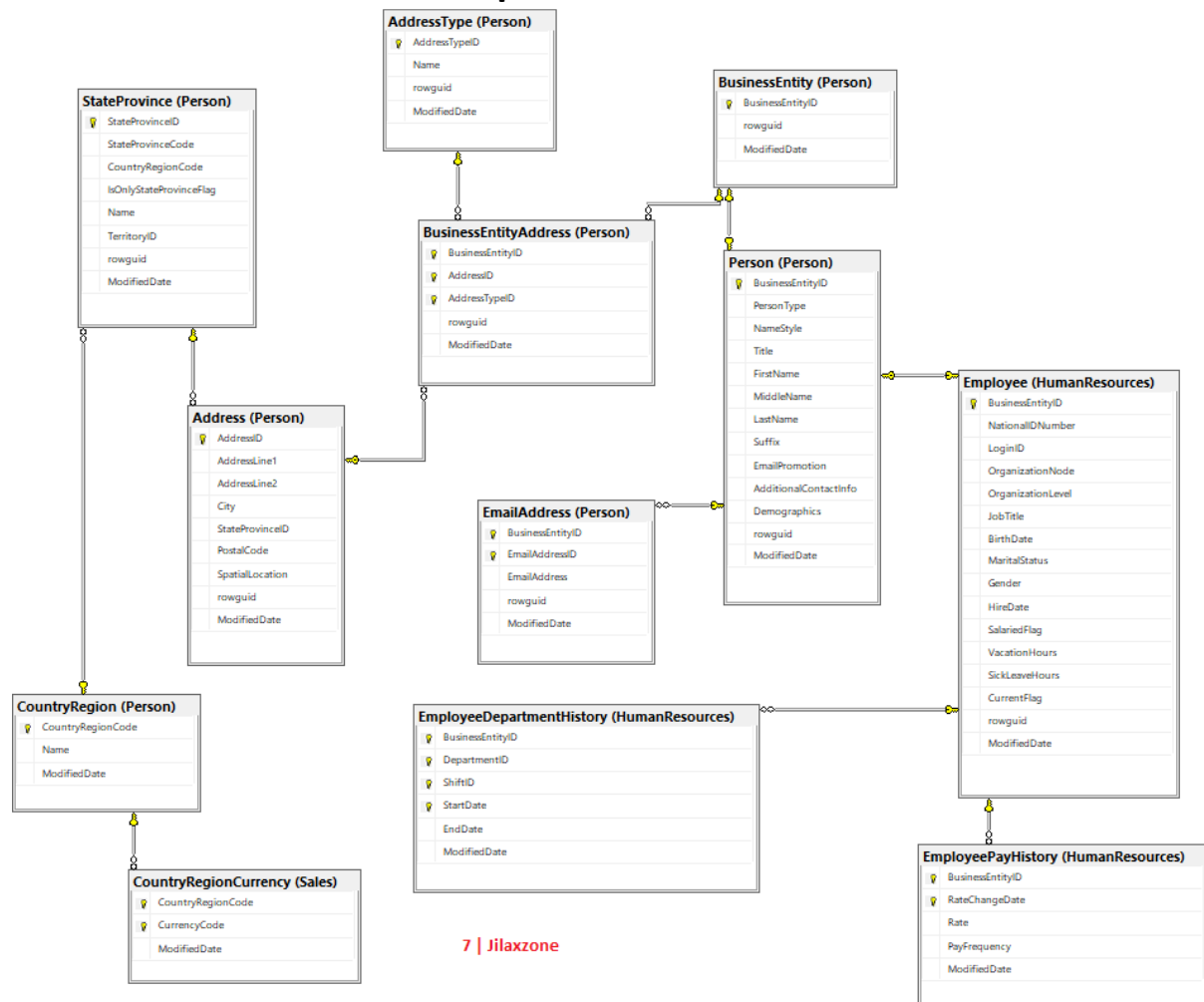
The screenshot shows the Microsoft Azure Portal interface for the resource group 'rgdbprojectmahima'. The 'Resources' tab is selected, displaying a table of resources. The table has columns for Name, Type, and Location. Two resources are listed: 'dbprojectmahima (serverprojectmahima/dbprojectmahima)' and 'serverprojectmahima', both located in 'East US'.

Name	Type	Location
dbprojectmahima (serverprojectmahima/dbprojectmahima)	SQL database	East US
serverprojectmahima	SQL server	East US

The screenshot shows the Microsoft Azure portal interface. The browser address bar displays the URL: `portal.azure.com/#@incedoin.onmicrosoft.com/resource/subscriptions/b313aa48-c704-4982-98e9-285b3cd5bf2d/resourceGroups/rgd...`. The page title is "dbprojectmahima (serverprojectmahima/dbprojectmahima)". The left sidebar contains 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 Power Platform. The main content area is titled "Getting started" and includes a "JSON View" button. It displays essential information about the database: Resource group (rgdprojectmahima), Status (Online), Location (East US), Subscription (Azure Training), and Subscription ID (b313aa48-c704-4982-98e9-285b3cd5bf2d). It also shows server details: Server name (serverprojectmahima.database.windows.net), Connection strings (Show database connection strings), Pricing tier (General Purpose - Serverless: Gen5, 1 vCore), Auto-pause delay (1 hour), and Earliest restore point (No restore point available). Below this, there are three main action cards: "Configure access" (Configure network access to your SQL server), "Connect to application" (Use connection strings to connect to your SQL database), and "Start developing" (Work in your database by using tools to add, modify and query data). Each card has a corresponding button: "Configure", "See connection strings", and "Open Azure Data Studio" (with a link to "Open in Visual Studio").

The screenshot shows the Microsoft Azure portal interface, specifically the "Query editor (preview)" page for the SQL database "dbprojectmahima". The browser address bar displays the URL: `portal.azure.com/#@incedoin.onmicrosoft.com/resource/subscriptions/b313aa48-c704-4982-98e9-285b3cd5bf2d/resourceGroup...`. The page title is "dbprojectmahima (serverprojectmahima/dbprojectmahima) | Query editor (preview)". The left sidebar is identical to the previous screenshot. The main content area is titled "Query 1" and includes a "Run" button, "Cancel query", "Save query", "Export data as", and "Show only Editor". Below the query editor, there is a "Results" section with a search bar to filter items. The "Results" section is currently empty, showing a "Ready" status at the bottom. The "Query editor" area contains a list of database objects: Tables (dbo.BuildVersion, dbo.ErrorLog, SalesLT.Address, SalesLT.Customer, SalesLT.CustomerAddress, SalesLT.Product, SalesLT.ProductCategory, SalesLT.ProductDescription, SalesLT.ProductModel, SalesLT.ProductModelProduct, SalesLT.SalesOrderDetail, SalesLT.SalesOrderHeader), Views (SalesLT.vGetAllCategories, SalesLT.vProductAndDescription, SalesLT.vProductModelCatalog, sys.database\_firewall\_rules), and Stored Procedures (System Stored Procedures, dbo.uspLogError, dbo.uspPrintError).

## Module 2: Perform T-SQL Operations on Restored Database



### Basic

#### Joins:

1. Retrieve a list of customers along with their total order amounts.

```

SELECT
    c.CustomerID,
    c.FirstName,
    c.LastName,
    SUM(o.TotalDue) AS TotalOrderAmount
FROM
    SalesLT.Customer c
JOIN
    SalesLT.SalesOrderHeader o ON c.CustomerID = o.CustomerID
GROUP BY
    c.CustomerID, c.FirstName, c.LastName
ORDER BY
    c.CustomerID;
  
```

Results Messages

Search to filter items...			
CustomerID	FirstName	LastName	TotalOrderAmount
29485	Catherine	Abel	43962.7901
29531	Cory	Booth	7330.8972
29546	Christopher	Beck	98138.2131
29568	Donald	Blanton	2669.3183
29584	Walter	Brian	272.6468

2. Display product information along with the number of units sold for each product.

```

SELECT
    p.ProductID,
    p.Name AS ProductName,
    p.ProductNumber,
    p.Color,
    SUM(od.OrderQty) AS TotalUnitsSold
FROM
    SalesLT.Product p
JOIN
    SalesLT.SalesOrderDetail od ON p.ProductID = od.ProductID
JOIN
    SalesLT.SalesOrderHeader oh ON od.SalesOrderID = oh.SalesOrderID
GROUP BY
    p.ProductID, p.Name, p.ProductNumber, p.Color
ORDER BY
    p.ProductID;

```

ProductID	ProductName	ProductNumber	Color	TotalUnitsSold
707	Sport-100 Helmet, Red	HL-U509-R	Red	35
708	Sport-100 Helmet, Black	HL-U509	Black	51
711	Sport-100 Helmet, Blue	HL-U509-B	Blue	38
712	AWC Logo Cap	CA-1098	Multi	52
714	Long-Sleeve Logo Jersey, M	LJ-0192-M	Multi	26

3. Find employees who have the same manager.

Unavailable data

4. List all customers who have never placed an order.

```

SELECT
    c.CustomerID,
    c.FirstName,
    c.LastName
FROM
    SalesLT.Customer c
LEFT JOIN
    SalesLT.SalesOrderHeader o ON c.CustomerID = o.CustomerID
WHERE
    o.CustomerID IS NULL;

```

CustomerID	FirstName	LastName
1	Orlando	Gee
2	Keith	Harris
3	Donna	Carreras
4	Janet	Gates
5	Lucy	Harrington

5. Retrieve the total sales amount for each product category.

```

SELECT
    pc.ProductCategoryID,
    pc.Name AS CategoryName,
    SUM(od.OrderQty * od.UnitPrice) AS TotalSalesAmount

```

```

FROM
    SalesLT.ProductCategory pc
JOIN
    SalesLT.Product p ON pc.ProductCategoryID = p.ProductCategoryID
JOIN
    SalesLT.SalesOrderDetail od ON p.ProductID = od.ProductID
JOIN
    SalesLT.SalesOrderHeader oh ON od.SalesOrderID = oh.SalesOrderID
GROUP BY
    pc.ProductCategoryID, pc.Name
ORDER BY
    pc.ProductCategoryID;

```

ProductCategoryID	CategoryName	TotalSalesAmount
5	Mountain Bikes	173085.8460
6	Road Bikes	185513.0436
7	Touring Bikes	221081.9622
8	Handlebars	1192.9680
9	Bottom Brackets	1320.1680

6. Display the names of employees and their direct managers.

Unavailable Data

7. Show the order details with product names for a specific customer.

```

SELECT
    oh.SalesOrderID,
    od.ProductID,
    p.Name AS ProductName,
    od.OrderQty,
    od.UnitPrice,
    od.LineTotal
FROM
    SalesLT.Customer c
JOIN
    SalesLT.SalesOrderHeader oh ON c.CustomerID = oh.CustomerID
JOIN
    SalesLT.SalesOrderDetail od ON oh.SalesOrderID = od.SalesOrderID
JOIN
    SalesLT.Product p ON od.ProductID = p.ProductID
WHERE
    c.CustomerID = 29485;

```

SalesOrderID	ProductID	ProductName	OrderQty	UnitPrice	LineTotal
71782	714	Long-Sleeve Logo Jersey, M	3	29.9940	89.982000
71782	956	Touring-1000 Yellow, 54	3	1430.4420	4291.326000
71782	954	Touring-1000 Yellow, 46	1	1430.4420	1430.442000
71782	712	AWC Logo Cap	10	5.3940	53.940000
71782	877	Bike Wash - Dissolver	10	4.7700	47.700000

8. List customers who have made purchases in the last 30 days.

```

SELECT DISTINCT
    c.CustomerID,
    c.FirstName,
    c.LastName
FROM
    SalesLT.Customer c
JOIN
    SalesLT.SalesOrderHeader oh ON c.CustomerID = oh.CustomerID
WHERE
    oh.OrderDate >= DATEADD(day, -30, GETDATE());

```

Query succeeded: Affected rows: 0

9. Find employees who do not have any direct reports.

Unavailable Data

10. Retrieve all products along with their average selling prices.

```
SELECT
  p.ProductID,
  p.Name AS ProductName,
  AVG(od.UnitPrice) AS AverageSellingPrice
FROM
  SalesLT.Product p
JOIN
  SalesLT.SalesOrderDetail od ON p.ProductID = od.ProductID
GROUP BY
  p.ProductID, p.Name
ORDER BY
  p.ProductID;
```

ProductID	ProductName	AverageSellingPrice
707	Sport-100 Helmet, Red	20.9940
708	Sport-100 Helmet, Black	20.6441
711	Sport-100 Helmet, Blue	20.7440
712	AWC Logo Cap	5.3740
714	Long-Sleeve Logo Jersey, M	29.9940

## Subqueries:

11. Find the order with the highest total amount.

```
SELECT TOP 1
  oh.SalesOrderID,
  oh.OrderDate,
  SUM(od.LineTotal) AS TotalAmount
FROM
  SalesLT.SalesOrderHeader oh
JOIN
  SalesLT.SalesOrderDetail od ON oh.SalesOrderID = od.SalesOrderID
GROUP BY
  oh.SalesOrderID, oh.OrderDate
ORDER BY
  TotalAmount DESC;
```

SalesOrderID	OrderDate	TotalAmount
71784	2008-06-01T00:00:00.0000000	89869.276314

12. Display customers who have placed orders with a total amount greater than the average.

```
WITH CustomerOrderTotals AS (
  SELECT
    c.CustomerID,
```

```

        c.FirstName,
        c.LastName,
        SUM(od.LineTotal) AS TotalAmount
FROM
    SalesLT.Customer c
JOIN
    SalesLT.SalesOrderHeader oh ON c.CustomerID = oh.CustomerID
JOIN
    SalesLT.SalesOrderDetail od ON oh.SalesOrderID = od.SalesOrderID
GROUP BY
    c.CustomerID, c.FirstName, c.LastName
)

SELECT
    CustomerID,
    FirstName,
    LastName,
    TotalAmount
FROM
    CustomerOrderTotals
WHERE
    TotalAmount > (SELECT AVG(TotalAmount) FROM CustomerOrderTotals);

```

CustomerID	FirstName	LastName	TotalAmount
29485	Catherine	Abel	33319.986000
29546	Christopher	Beck	74160.228000
29660	Anthony	Chor	47848.026000
29736	Terry	Eminhizer	89869.276314
29796	Jon	Grande	65123.463418

13. List products with prices higher than the average product price.

```

WITH ProductPrices AS (
    SELECT
        ProductID,
        Name AS ProductName,
        ListPrice
    FROM
        SalesLT.Product
)

SELECT
    ProductID,
    ProductName,
    ListPrice
FROM
    ProductPrices
WHERE
    ListPrice > (SELECT AVG(ListPrice) FROM ProductPrices);

```

ProductID	ProductName	ListPrice
680	HL Road Frame - Black, 58	1431.5000
706	HL Road Frame - Red, 58	1431.5000
717	HL Road Frame - Red, 62	1431.5000
718	HL Road Frame - Red, 44	1431.5000
719	HL Road Frame - Red, 48	1431.5000

14. Retrieve orders placed by employees who have a specific job title.

Data Unavailable

15. Display customers who have placed orders for a specific product category.

```
SELECT
    c.CustomerID,
    c.FirstName,
    c.LastName
FROM
    SalesLT.Customer c
JOIN
    SalesLT.SalesOrderHeader oh ON c.CustomerID = oh.CustomerID
JOIN
    SalesLT.SalesOrderDetail od ON oh.SalesOrderID = od.SalesOrderID
JOIN
    SalesLT.Product p ON od.ProductID = p.ProductID
JOIN
    SalesLT.ProductCategory pc ON p.ProductCategoryID = pc.ProductCategoryID
WHERE
    pc.ProductCategoryID = 22;
```

Query succeeded: Affected rows: 0

16. Find employees with salaries greater than the average salary in their department.

Data unavailable

17. List customers who have placed orders before a specific date.

```
SELECT DISTINCT
    c.CustomerID,
    c.FirstName,
    c.LastName
FROM
    SalesLT.Customer c
JOIN
    SalesLT.SalesOrderHeader oh ON c.CustomerID = oh.CustomerID
WHERE
    oh.OrderDate < 2008-07-02;
```

Query succeeded: Affected rows: 0

18. Retrieve the order with the highest quantity of a specific product.

```
SELECT TOP 1
    od.SalesOrderID,
    od.ProductID,
    p.Name AS ProductName,
    SUM(od.OrderQty) AS TotalQuantity
FROM
    SalesLT.SalesOrderDetail od
JOIN
    SalesLT.Product p ON od.ProductID = p.ProductID
WHERE
    od.ProductID = 714
GROUP BY
    od.SalesOrderID, od.ProductID, p.Name
ORDER BY
    TotalQuantity DESC;
```

SalesOrderID	ProductID	ProductName	TotalQuantity
71784	714	Long-Sleeve Logo Jersey, ...	9



19. Display products with prices lower than the lowest product price in a specific category.

```
WITH ProductPrices AS (
    SELECT
        p.ProductID,
        p.Name AS ProductName,
        p.ListPrice,
        pc.ProductCategoryID
    FROM
        SalesLT.Product p
    JOIN
        SalesLT.ProductCategory pc ON p.ProductCategoryID = pc.ProductCategoryID
    WHERE
        pc.ProductCategoryID = 11
)

SELECT
    ProductID,
    ProductName,
    ListPrice
FROM
    ProductPrices
WHERE
    ListPrice < (SELECT MIN(ListPrice) FROM ProductPrices);
```

Query succeeded: Affected rows: 0

20. Find employees who have the same job title as their manager.  
Data unavailable

## Unions:

21. Combine results from two queries to get a list of unique customer and employee names.  
Data Unavailable

22. Retrieve product names that are common in two different product categories.  
Data unavailable

```
SELECT
    p.Name AS ProductName
FROM
    SalesLT.Product p
JOIN
    SalesLT.ProductCategory pc1 ON p.ProductCategoryID = pc1.ProductCategoryID
JOIN
    SalesLT.ProductCategory pc2 ON p.ProductCategoryID = pc2.ProductCategoryID
WHERE
    pc1.ProductCategoryID <> pc2.ProductCategoryID;
```

Query succeeded: Affected rows: 0

23. Display the names of employees and customers in a single result set.  
Data unavailable

24. List products that are in stock or have been discontinued.  
Data unavailable

25. Combine the results of two queries to find unique products ordered by a specific customer.

```

SELECT DISTINCT
    p.ProductID,
    p.Name AS ProductName
FROM
    SalesLT.Product p
JOIN
    SalesLT.SalesOrderDetail od ON p.ProductID = od.ProductID
JOIN
    SalesLT.SalesOrderHeader oh ON od.SalesOrderID = oh.SalesOrderID
WHERE
    oh.CustomerID = 29485

UNION

SELECT DISTINCT
    p.ProductID,
    p.Name AS ProductName
FROM
    SalesLT.Product p
JOIN
    SalesLT.SalesOrderDetail od ON p.ProductID = od.ProductID

ORDER BY
    ProductID;

```

ProductID	ProductName
707	Sport-100 Helmet, Red
708	Sport-100 Helmet, Black
711	Sport-100 Helmet, Blue
712	AWC Logo Cap
714	Long-Sleeve Logo Jersey, M

26. Retrieve orders placed by customers and employees in a single result set.  
Data unavailable
27. Display products that are either in a specific category or have a specific safety stock level.  
Data unavailable
28. List customers who have placed orders and employees who have direct reports in a single result set.  
Data unavailable
29. Retrieve products that are in stock in one location and out of stock in another.  
Data unavailable
30. Combine information about employees who are managers and employees who have managers.  
Data unavailable