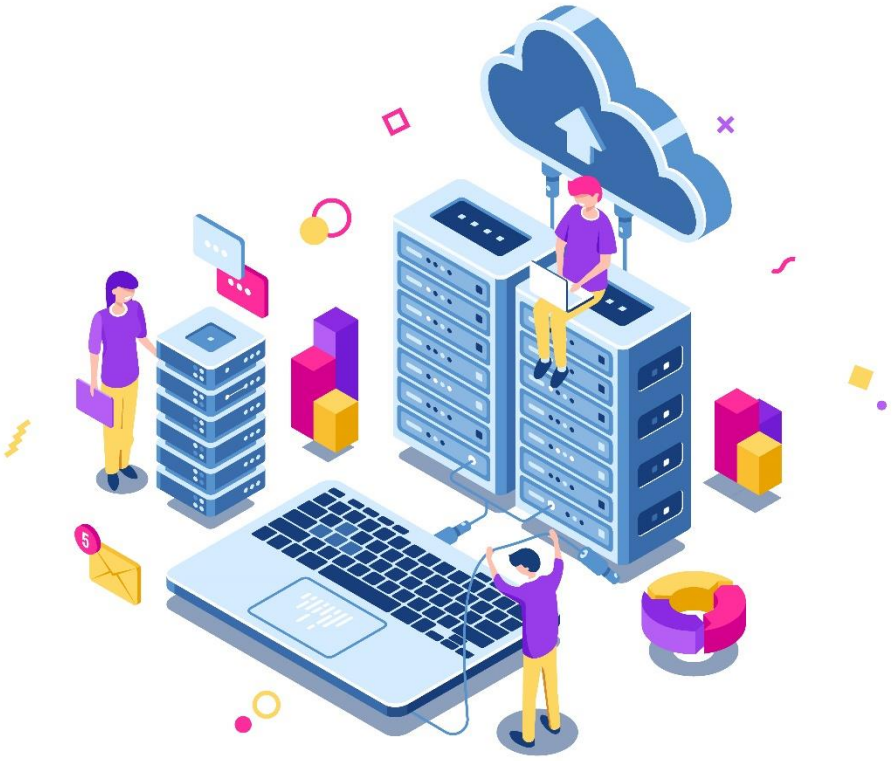


Lab 4: Cosmos DB and ASP.NET.

Akshar Panchani ID- 202101522

IT457 Cloud Computing

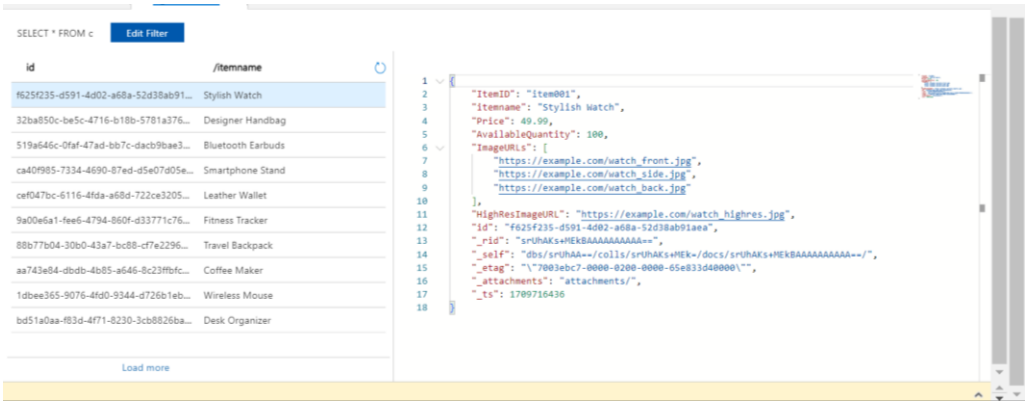
3/2/24

Question – 1

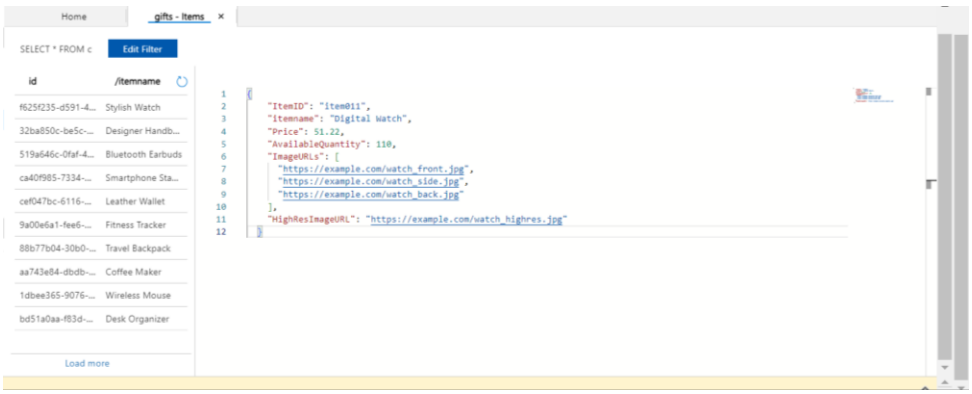
1. Create NoSQL Azure Table storage design for the following problems and implement crud operations using your opted language SDK to validate the working of the design.

a. A gift shop wants to start a portal which allows users to view various items present in its store. The items details like name, price, available quantity etc. are shown. Along with that, pictures of item from different angles are also available. There's a high-res image also available for every item:

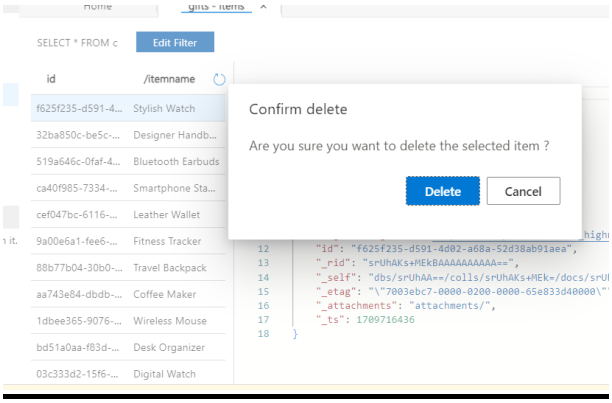
Items to upload in gifts container:



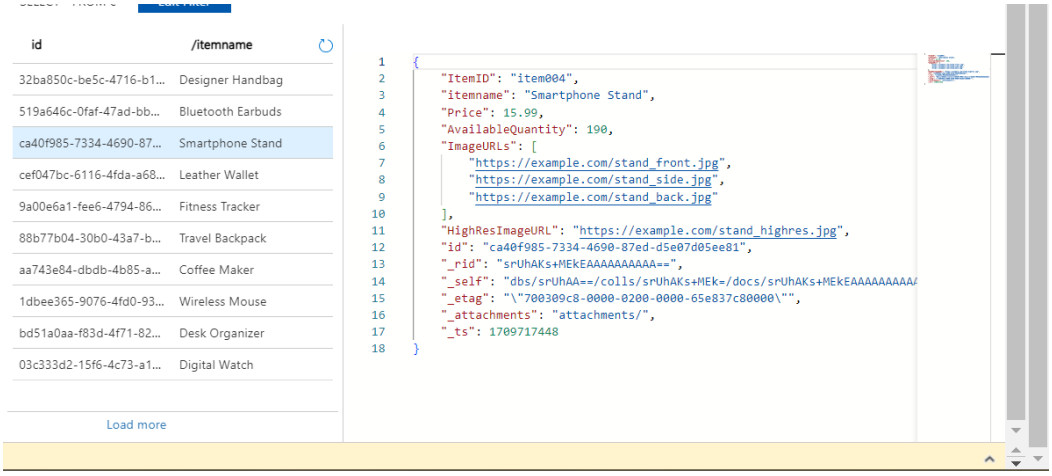
Inserting Item and inserted list: -



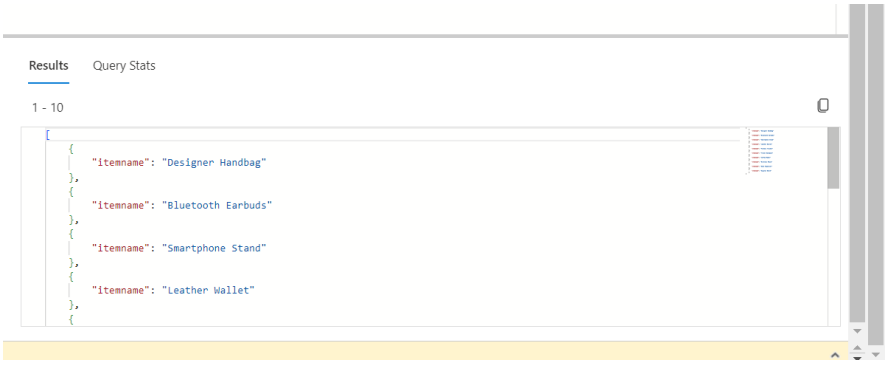
Deleting item and deleted items: -



Updating item: -

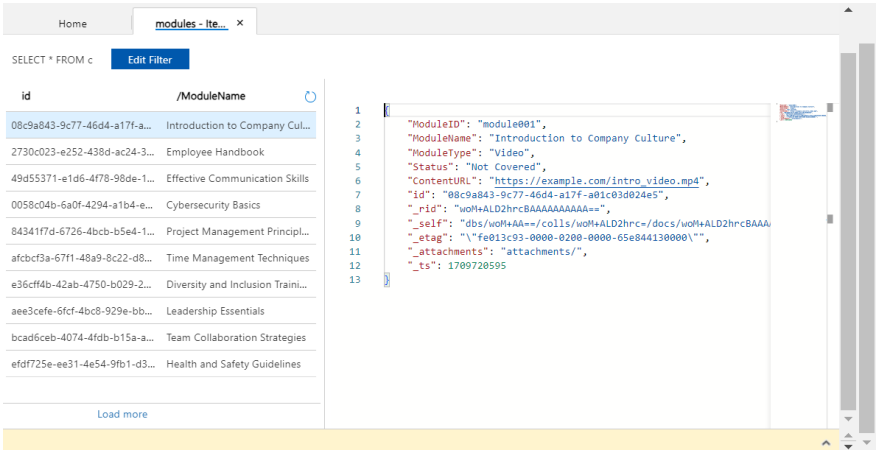


Query Execution: -

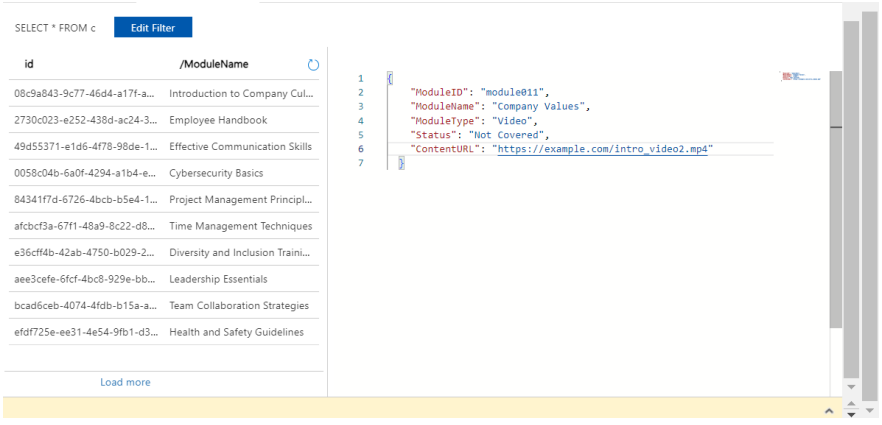


A company has training period for employees. It has around 500 training modules that an employee needs to cover over the period of 6 months. A module could be a video, document, e-book or other form or content. The requirement is to track the status of modules covered by employee over the training period:

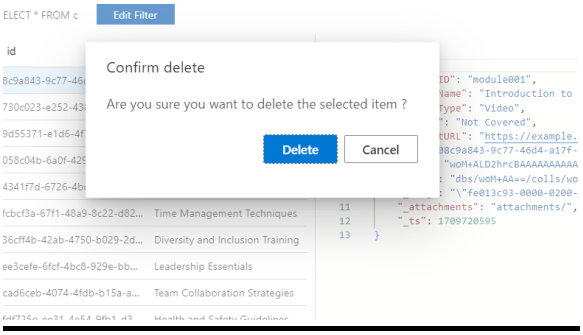
Items uploaded in modules container:



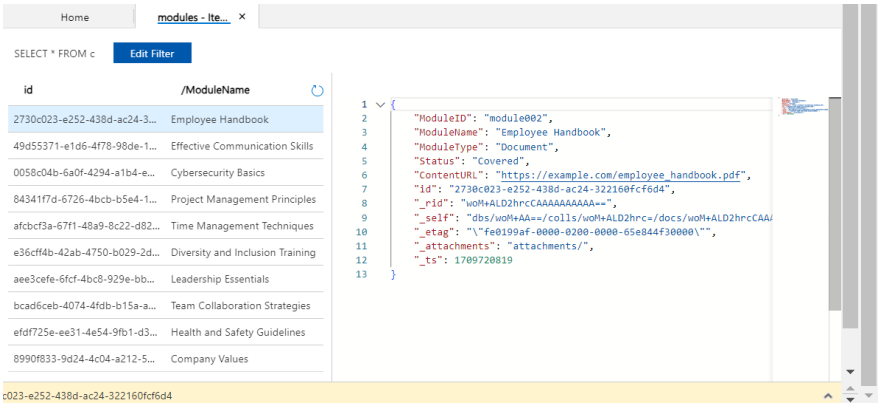
Inserting Item and inserted item: -



Deleting item: -



Item deleted and Updating item: -

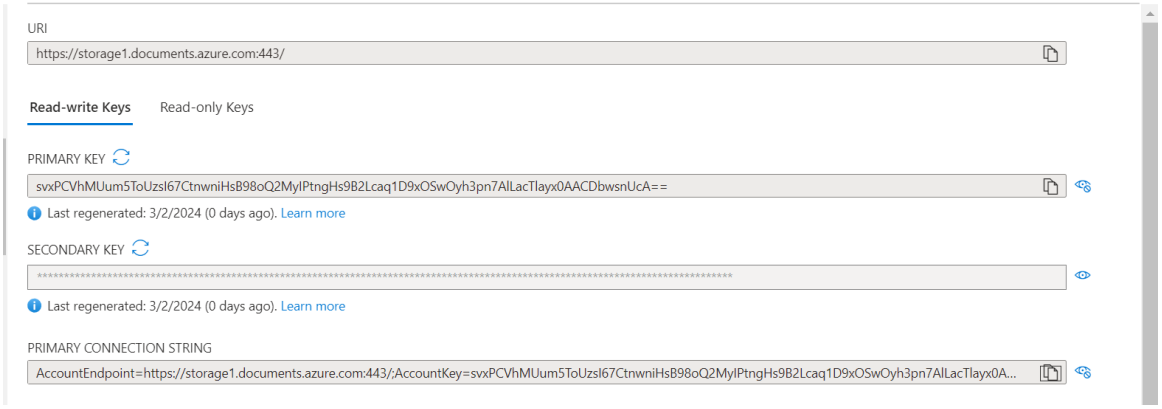


Query Execution: -

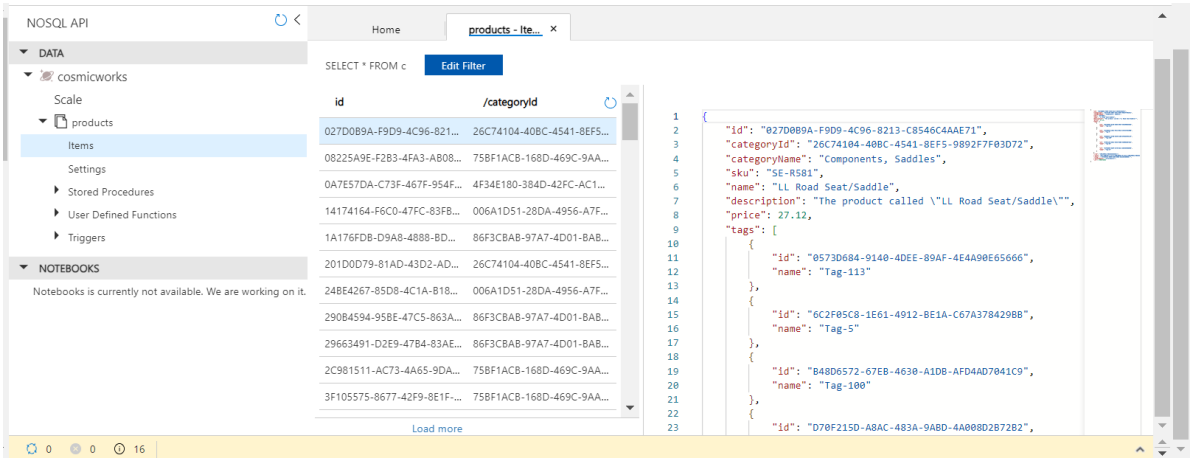
Question - 2

Develop an ASP.NET web application with Azure Cosmos DB for NoSQL

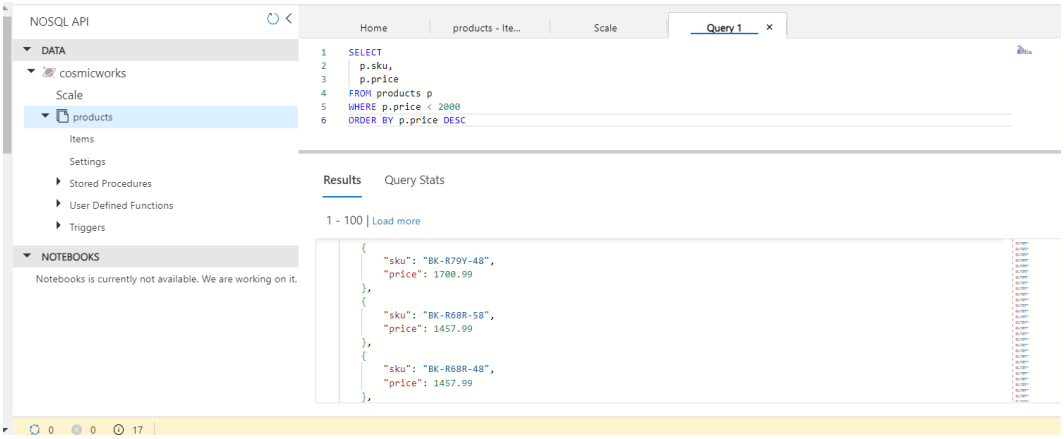
Keys in Cosmos DB account: -

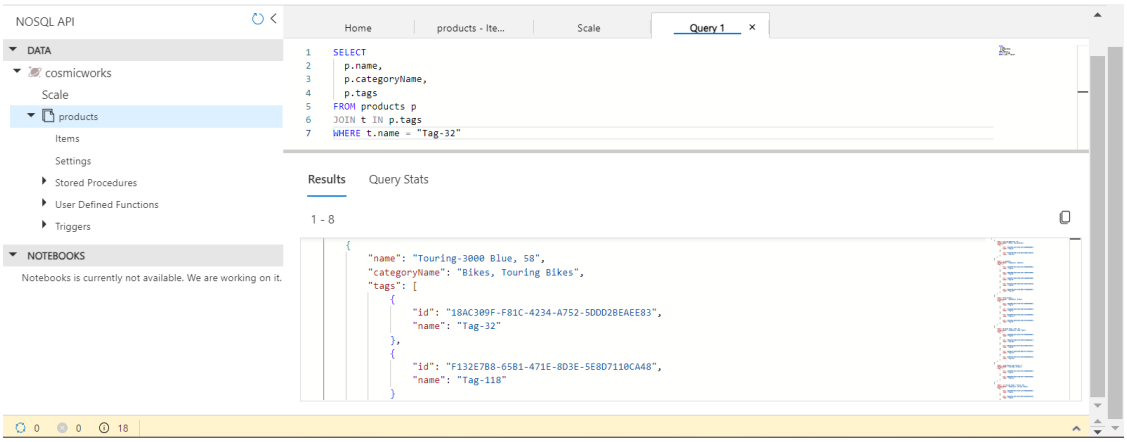


Data Population: -



Query Executions: -

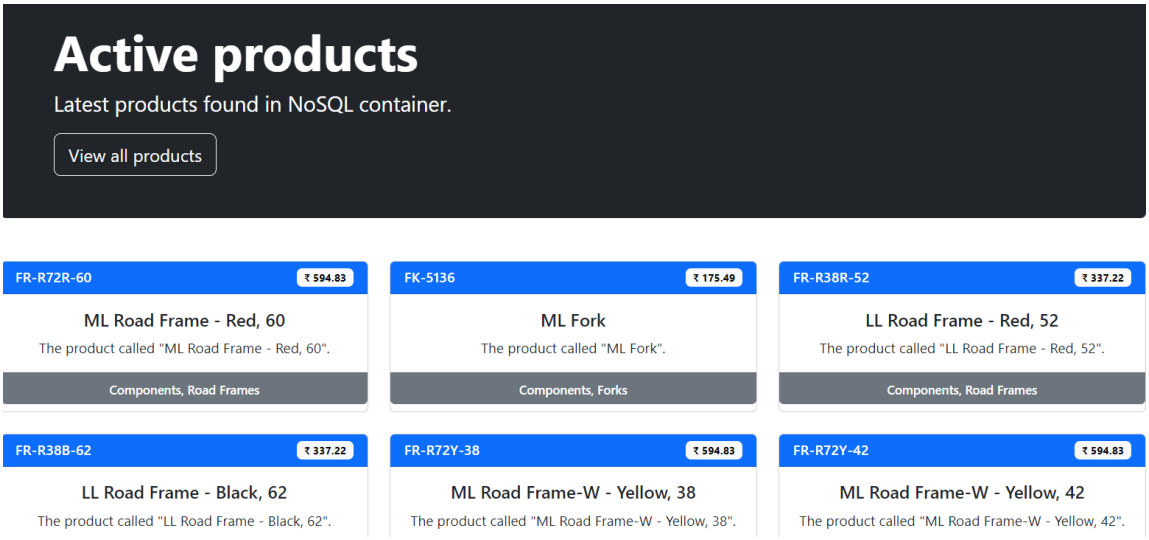




Building and running application:-

```
E:\Acads\Sem 6\IT457 Cloud Computing\Labs\Lab4\Code>dotnet new cosmosdbnosql-webapp
The template "Azure Cosmos DB for NoSQL Tutorial - ASP.NET project template" was created successfully.


E:\Acads\Sem 6\IT457 Cloud Computing\Labs\Lab4\Code>dotnet run
info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[63]
      User profile is available. Using 'C:\Users\Akshar2\AppData\Local\ASP.NET\DataProtection-Keys' as key
 Windows DPAPI to encrypt keys at rest.
info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[58]
      Creating key {9e3dcb9c-e30d-4a82-88dd-b25648d2b7b1} with creation date 2024-03-02 11:14:08Z, activati
-02 11:14:08Z, and expiration date 2024-05-31 11:14:08Z.
info: Microsoft.AspNetCore.DataProtection.Repositories.FileSystemXmlRepository[39]
      Writing data to file 'C:\Users\Akshar2\AppData\Local\ASP.NET\DataProtection-Keys\key-9e3dcb9c-e30d-4a
d2b7b1.xml'.
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: E:\Acads\Sem 6\IT457 Cloud Computing\Labs\Lab4\Code\
```



Adding code and running the application: -

```
public class CosmosService : ICosmosService
    public CosmosService()
    {
        _client = new CosmosClient(
            connectionString: "AccountEndpoint=https://storage1.documents.azure.com:443/;AccountKey=jo
        );
    }
    2 references
    private Container container
    {
        get => _client.GetDatabase("cosmicworks").GetContainer("products");
    }
    2 references
    public async Task<IEnumerable<Product>> RetrieveAllProductsAsync()
    {
        var queryable = container.GetItemLinqQueryable<Product>();
        using FeedIterator<Product> feed = queryable
            .Where(p => p.price < 2000m)
            .OrderByDescending(p => p.price)
            .ToFeedIterator();
        List<Product> results = new();
        while (feed.HasMoreResults)
        {
            var response = await feed.ReadNextAsync();
            foreach (Product item in response)
            {
                results.Add(item);
            }
        }
        return results;
    }
```

```
ROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS  AZURE

    Executed an implicit handler method, returned result Microsoft.AspNetCore.Mvc.RazorPages.PageResult.
nfo: Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker[4]
    Executed page /Index in 1506.6176ms
nfo: Microsoft.AspNetCore.Routing.EndpointMiddleware[1]
nfo: Microsoft.AspNetCore.Hosting.Diagnostics[2]
    Request finished HTTP/1.1 GET http://localhost:5000/ - - - 200 - text/html;+charset=utf-8 1507.2470ms
nfo: Microsoft.AspNetCore.Hosting.Diagnostics[1]
    Request starting HTTP/1.1 GET http://localhost:5000/favicon.ico - -
nfo: Microsoft.AspNetCore.StaticFiles.StaticFileMiddleware[6]
    The file /favicon.ico was not modified
om
```

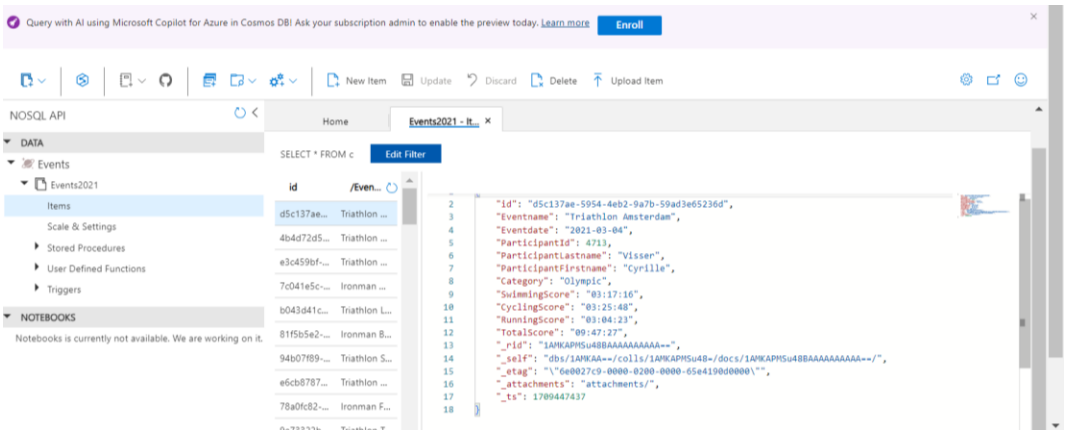
Updated list of products from the Azure Cosmos DB database with prices descending

Legend
Standard price
High price [> ₹ 1,000.00]
Low price [< ₹ 100.00]

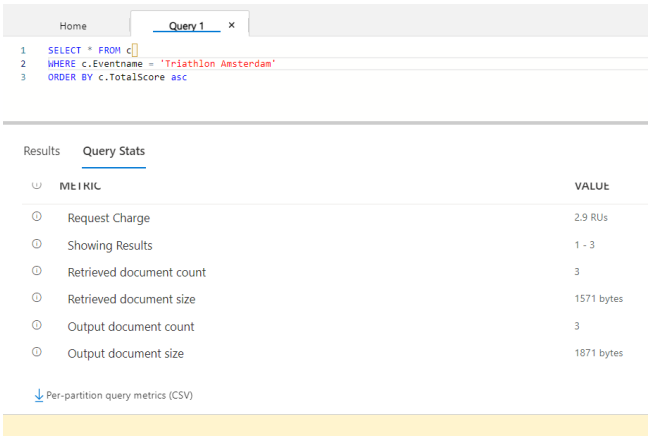
ID	Name	SKU	Category	Description	Price
C461038A-6DB6-4EC7-924F-ECA906259A6E	Road-350-W Yellow, 42	BK-R79Y-42	Bikes, Road Bikes	The product called "Road-350-W Yellow, 42"	₹ 1,700.99
9E5C74FD-F685-45AE-A799-D67EFB5C28A1	Road-350-W Yellow, 40	BK-R79Y-40	Bikes, Road Bikes	The product called "Road-350-W Yellow, 40"	₹ 1,700.99
4F9FC42A-F43F-4C13-92FC-ADF701F48C36	Road-350-W Yellow, 44	BK-R79Y-44	Bikes, Road Bikes	The product called "Road-350-W Yellow, 44"	₹ 1,700.99
063F1A00-8CA1-4DB9-8298-BEAC4B8CC238	Road-350-W Yellow, 48	BK-R79Y-48	Bikes, Road Bikes	The product called "Road-350-W Yellow, 48"	₹ 1,700.99
F1AA8B6D-4CF2-4DB2-BB17-997C2BD1A6AC	Road-450 Red, 58	BK-R68R-58	Bikes, Road Bikes	The product called "Road-450 Red, 58"	₹ 1,457.99
E2FD2420-B084-4764-8BC4-94574DFF1AC6	Road-450 Red, 48	BK-R68R-48	Bikes, Road Bikes	The product called "Road-450 Red, 48"	₹ 1,457.99
D616598D-3159-4616-BF9D-	Road-450 Red, 60	BK-R68R-	Bikes, Road Bikes	The product called "Road-450 Red, 60"	₹

Azure Cosmos DB – end to end example:

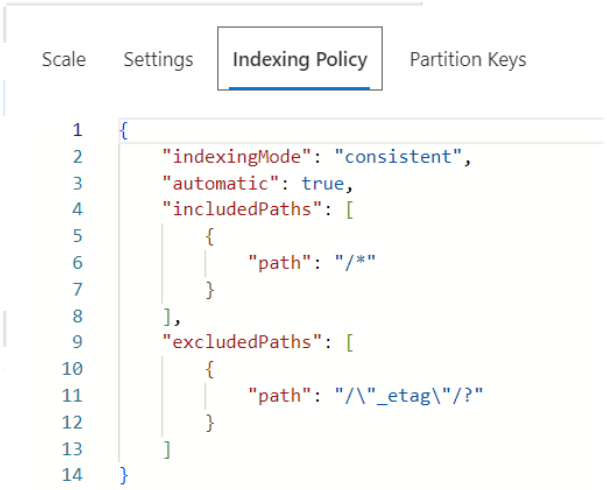
Storage account, database, container, and items: -



Executing query: -

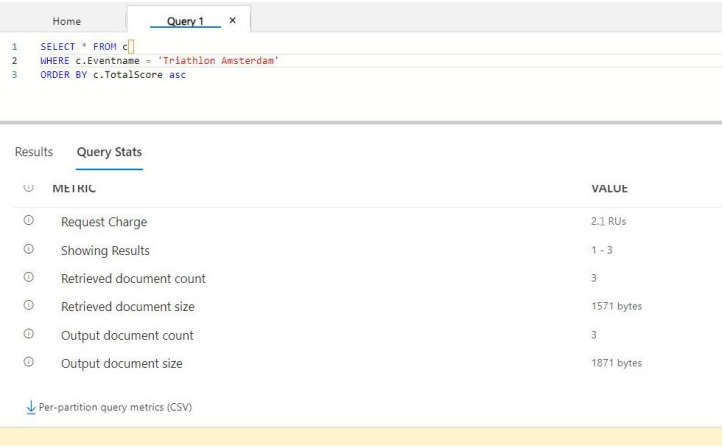


Old indexing policy: -



New indexing policy: -

Optimised querying: -



Azure Cosmos DB for Table for .NET

Creating Cosmos DB account using Azure PowerShell:

I

```
PS /home/akshar2 > # Variable for resource group name
PS /home/akshar2 > $RESOURCE_GROUP_NAME = "msdocs-cosmos-quickstart-rg"
PS /home/akshar2 > $LOCATION = "West US"
PS /home/akshar2 >
PS /home/akshar2 > # Variable for account name with a randomnly generated suffix
PS /home/akshar2 > $SUFFIX = Get-Random
PS /home/akshar2 > $ACCOUNT_NAME = "msdocs-$SUFFIX"
PS /home/akshar2 > $parameters = @{
>>     Name = $RESOURCE_GROUP_NAME
>>     Location = $LOCATION
>> }
PS /home/akshar2 > New-AzResourceGroup @parameters

ResourceGroupName : msdocs-cosmos-quickstart-rg
Location          : westus
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/97e43b90-2dab-44c3-a90a-fa73529dceac/resourceGroups/msdocs-cosmos-quickstart-rg
```

```
PS /home/akshar2 > New-AzCosmosDBAccount @parameters

Id                                : /subscriptions/97e43b90-2dab-44c3-a90a-fa73529dceac/resourceGroups/msdocs-cosmos-quickstart-rg/providers/Microsoft.Do
Name                              : msdocs-1162849875
Location                          : West US
Tags                              : {}
EnableCassandraConnector          :
EnableMultipleWriteLocations      : False
VirtualNetworkRules               : {}
FailoverPolicies                  : {msdocs-1162849875-westus}
Locations                         : {msdocs-1162849875-westus}
ReadLocations                     : {msdocs-1162849875-westus}
WriteLocations                    : {msdocs-1162849875-westus}
Capabilities                      : {EnableTable}
ConsistencyPolicy                 : Microsoft.Azure.Management.CosmosDB.Models.ConsistencyPolicy
EnableAutomaticFailover           : False
IsVirtualNetworkFilterEnabled     : False
IpRules                           : {}
DatabaseAccountOfferType          : Standard
DocumentEndpoint                  : https://msdocs-1162849875.documents.azure.com:443/
ProvisioningState                 : Succeeded
Kind                              : GlobalDocumentDB
ConnectorOffer                    :
DisableKeyBasedMetadataWriteAccess : False
PublicNetworkAccess               : Enabled
KeyVaultKeyUri                    :
PrivateEndpointConnections        :
EnableFreeTier                    : False
ApiProperties                     : Microsoft.Azure.Commands.CosmosDB.Models.PSApiProperties
EnableAnalyticalStorage           : False
EnableBurstCapacity               : False
```

Getting document endpoints: -

```
PS /home/akshar2> $parameters = @{
>>     ResourceGroupName = $RESOURCE_GROUP_NAME
>>     Name = $ACCOUNT_NAME
>>     Type = "ConnectionStrings"
>> }
PS /home/akshar2> Get-AzCosmosDBAccountKey @parameters | Select-Object -Property "Primary Table Connection String"

Primary Table Connection String
-------------------------------
DefaultEndpointsProtocol=https;AccountName=msdocs-1162849875;AccountKey=rcsCkdV0P0Jjk8MFqH8qMiMACXqNM1Dd8kx5qpsIu6JpuN45QwWZEnRAHxMwehoGNIj
```

Code: -

```
var product = await tableClient.GetEntityAsync<Product>(
    rowKey: "68719518388",
    partitionKey: "gear-surf-surfboards"
);
Console.WriteLine("Single product:");
Console.WriteLine(product.Value.Name);

// Read multiple items from container
var prod2 = new Product()
{
    RowKey = "68719518390",
    PartitionKey = "gear-surf-surfboards",
    Name = "Sand Surfboard",
    Quantity = 5,
    Sale = false
};

await tableClient.AddEntityAsync<Product>(prod2);

var products = tableClient.Query<Product>(x => x.PartitionKey == "gear-surf-surfboards");

Console.WriteLine("Multiple products:");
foreach (var item in products)
{
    Console.WriteLine(item.Name);
}
```

Output: -

```
PS E:\Acads\Sem 6\IT457 Cloud Computing\Labs\Lab4\Code\myApp> dotnet run
Single product:
Ocean Surfboard
Multiple products:
Ocean Surfboard
Sand Surfboard
```

Output on Azure portal: -