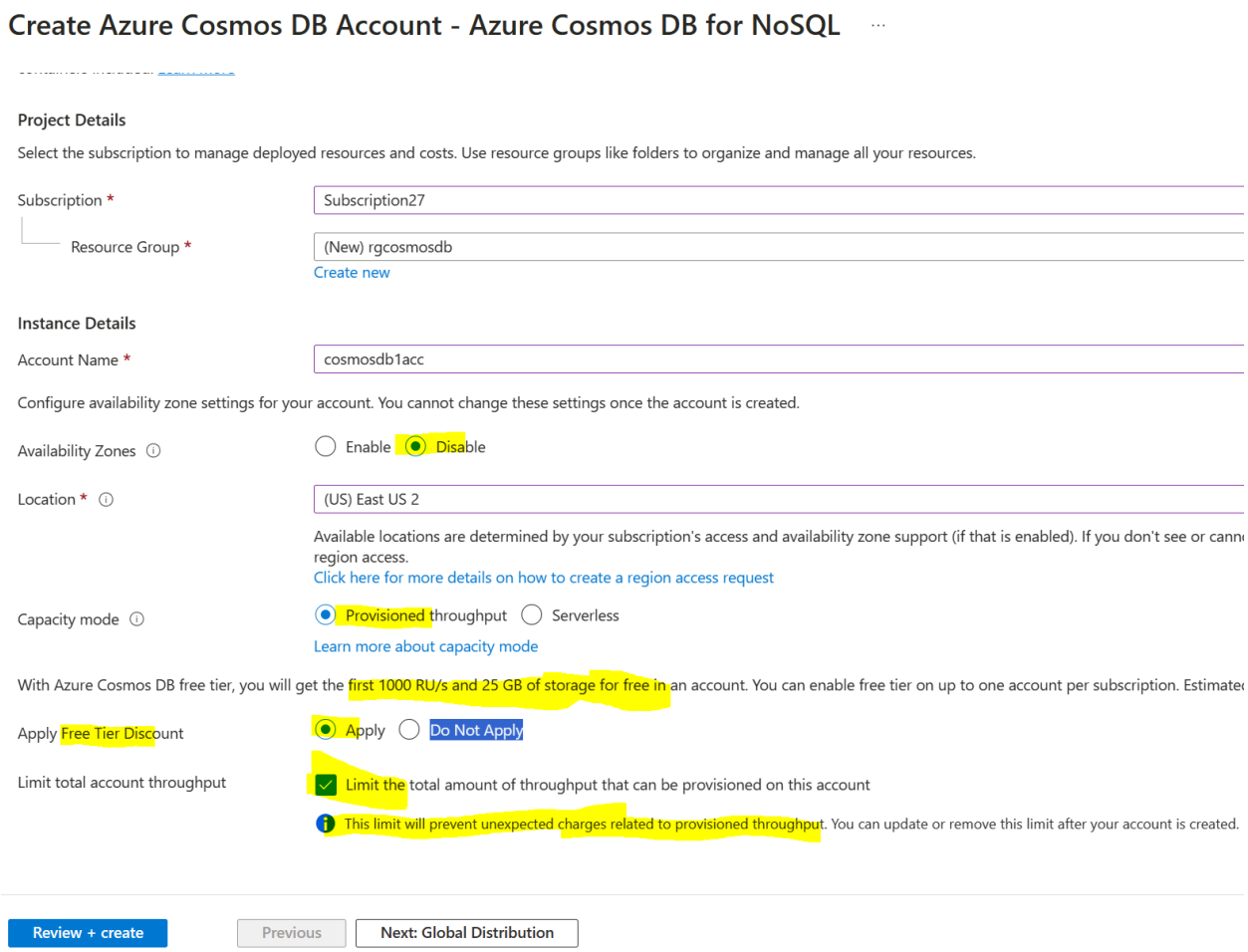
AZ 204 Certification + Cosmos DB

Contents:

Video course

Creating a CosmosDB



Next pane: Global Distribution

s

Global Distribution (Accept Default)

Geo-Redundancy Disable[Y] Multi-region Writes Disable[Y]

Next pane: Networking(Accept Default)

All public networks

Next BAckup: **(Accept Default)**

Next Encryption: **(Accept Default)**

Tags

[CREATE a COSMOS DB] OK

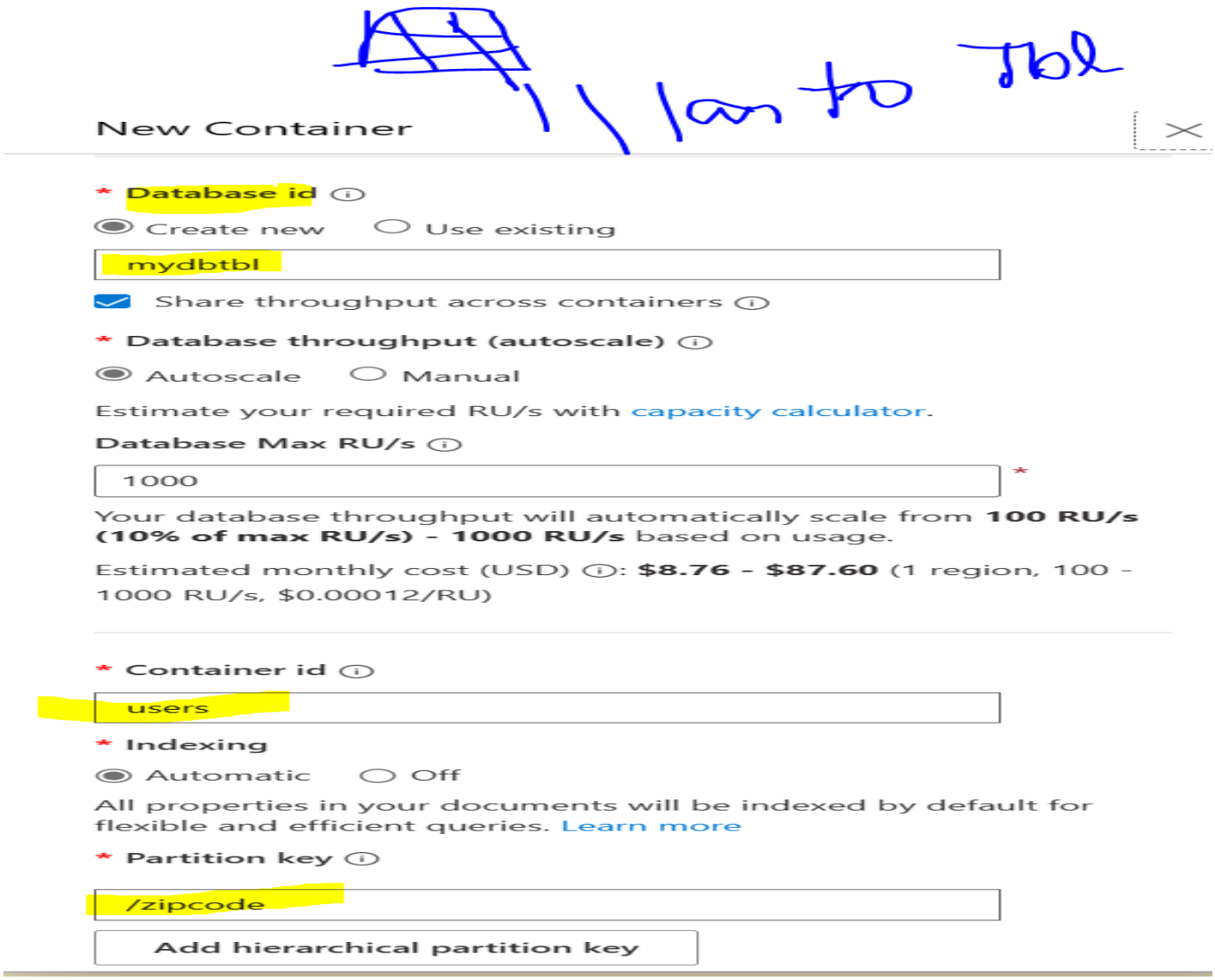
1.b.Creating a container

Container is similar to a SQL TAble

Database id Create new : mydbtbl

* Container id: users

* Partition key: /zipcode



Error & Solution:

Created a cosmos db account; soon after that there was an Error when creating a container Message: {"code":"BadRequest","message":"Message: {\"Errors\":[\"Your account is currently configured with a total throughput limit of 1000 RU\\/s. This operation failed because it would have increased the total throughput to 1400 RU\\/s. See https:\\/\\/aka.ms\\/cosmos-tp-limit for more information.\"]}\r\nActivityId: 75fda8d3-02a3-4231-9f94-75a9ffa408fc, Request URI: /apps/f0bcbc51-a121-44e1-a35a-c5de0519a069/services/c8a9aba0-aba8-4043-b02d-bee76197bcd4/partitions/e0768073-7a6d-4eb2-8b7d-23def0a00c34/replicas/133536693015853671p, RequestStats: , SDK: Microsoft.Azure.Documents.Common/2.14.0"}, Request URI: /dbs, RequestStats: , SDK: Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0 ActivityId: c7bf6f79-a5f9-41fd-a27d-a9aeb1646d5c, Microsoft.Azure.Documents.Common/2.14.0	I deleted a database called ToDoList-> Items collection After this deletion this error was gone and the container was created. 200 OK Conclusion: At a time there has to be a container to meet the limit of 1000RTU

1C.Create a new item-> new item and fill key val pair


```
{
    "id": "01",
    "name": "Deepika",
    "zipcode": "60016",
    "address": "Des Plaines"
  }
```


1D.New item with metadata


```
{
    "id": "01",
    "name": "Deepika",
    "zipcode": "60016",
    "address": "Des Plaines",
    "_rid": "XU12ANYicREBAAAAAAAAAA==",
    "_self":
"dbs/XU12AA==/colls/XU12ANYicRE=/docs/XU12ANYicREBAAAAAAAAAA==/",
    "_etag": "\"4f00268e-0000-0200-0000-662d47fa0000\"",
    "_attachments": "attachments/",
    "_ts": 1714243578
}
```

Explained:

```
 "_rid": ressource if
    "_self": connection to itself with r_id contained.Identify a
record from its collection and db
_self has database id, collection id, and r_id repeated again
    "_etag": for caching; for chaching the content,
    "_attachments": attachment information
    "_ts": timestamp created and update
```

1.e. Query the container in SQL Explorer

```
SELECT * FROM c where c.id='02'
```

```
[
    {
        "id": "02",
        "Name": "Niranth",
        "zipcode": "47906",
        "address": "West Lafayette",
        "_rid": "XU12ANYicRECAAAAAAAAAA==",
        "_self":
"dbs/XU12AA==/colls/XU12ANYicRE=/docs/XU12ANYicRECAAAAAAAAAA==/",
        "_etag": "\"4f00c9a4-0000-0200-0000-662d49eb0000\"",
        "_attachments": "attachments/",
        "_ts": 1714244075
    }
]
```

```
Or query using root ie select * from root 'instead of' select *
from c
```

```
select * from root where root.zipcode='60016'
```

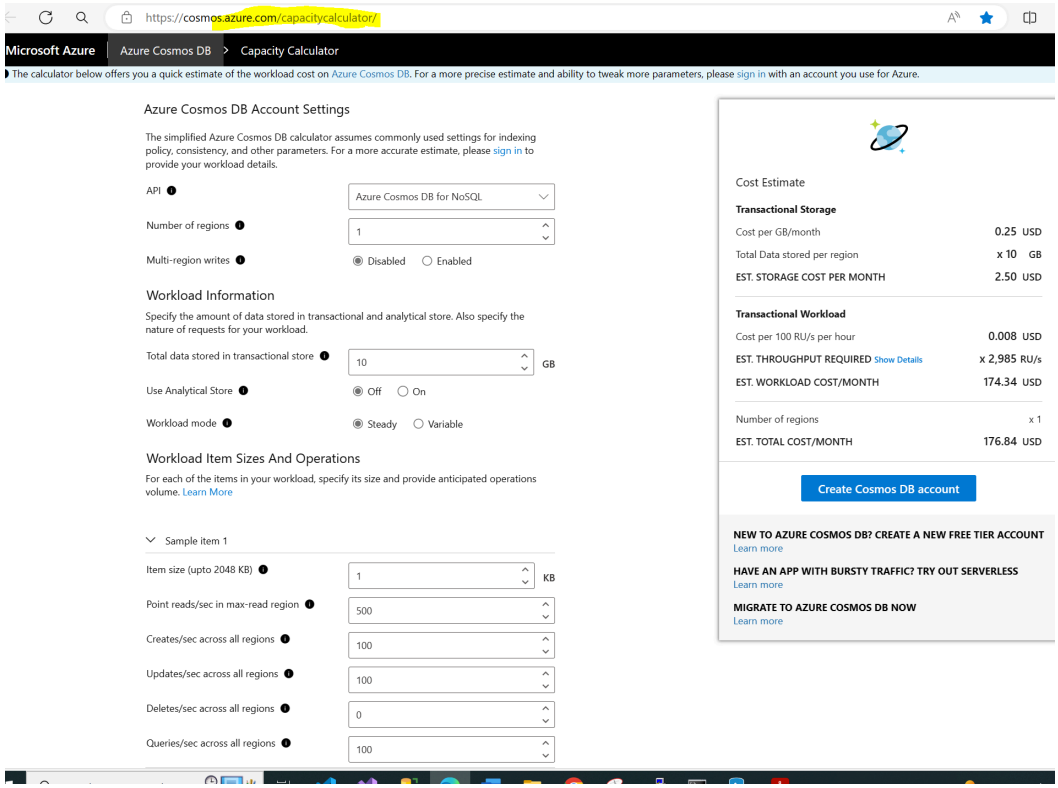
```
[
    {
        "id": "01",
        "name": "Deepika",
        "zipcode": "60016",
        "address": "Des Plaines",
        "_rid": "XU12ANYicREBAAAAAAAAAA==",
        "_self":
"dbs/XU12AA==/colls/XU12ANYicRE=/docs/XU12ANYicREBAAAAAAAAAA==/",
        "_etag": "\"4f00268e-0000-0200-0000-662d47fa0000\"",
        "_attachments": "attachments/",
        "_ts": 1714243578
    }
]
```

1.e. Calculate usage using an emulator

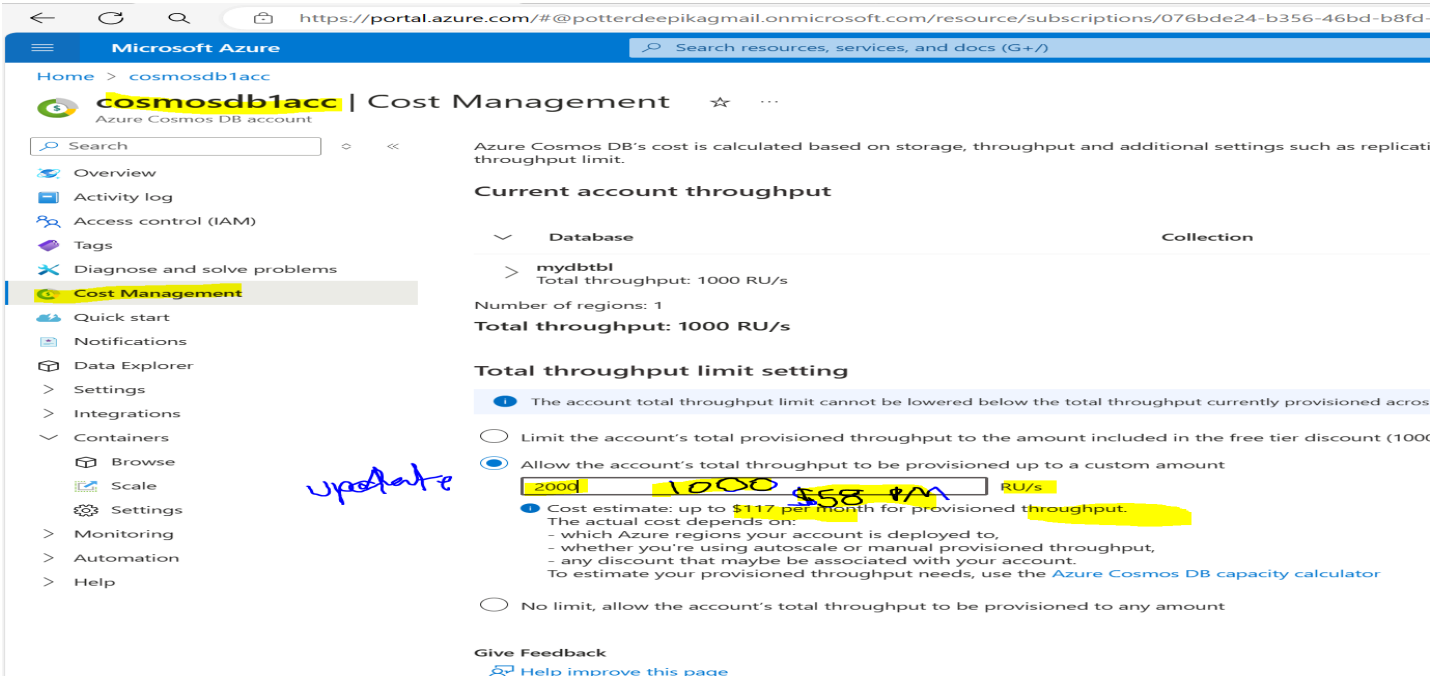
Measure your usage in cosmosdb

Azure Cosmos DB Capacity Calculator

https://cosmos.azure.com/capacitycalculator/



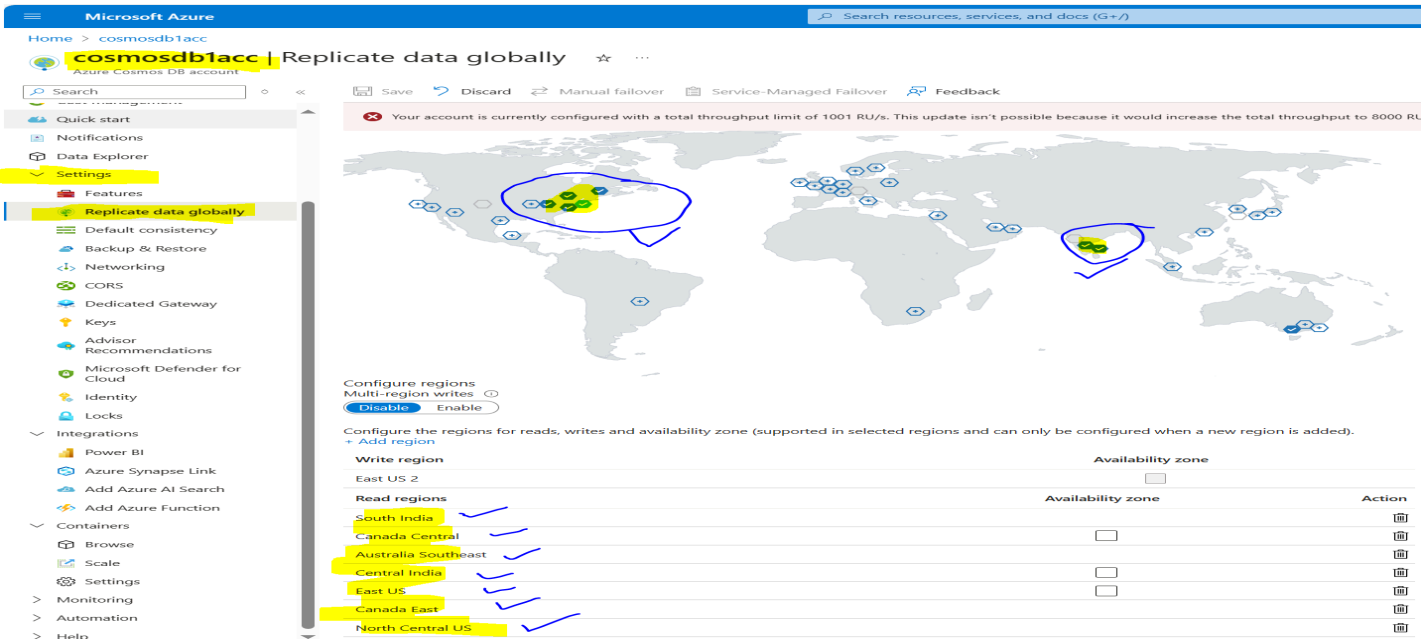
1 g Scale up Cost Management- limit cost:



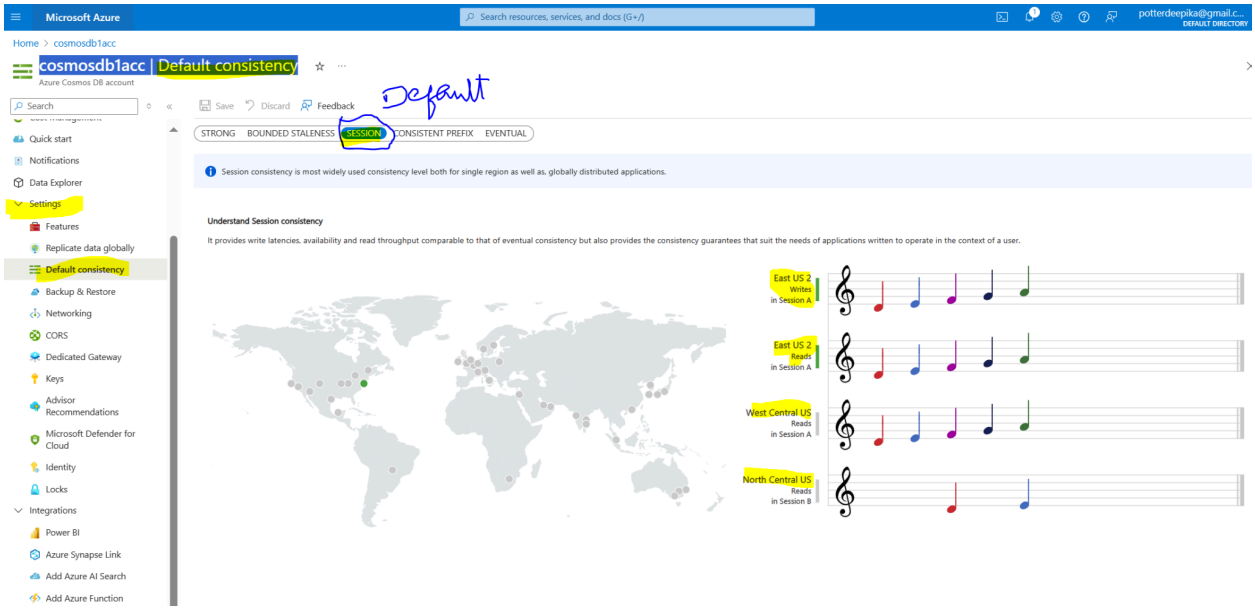
Better to limit instead of choosing no limit.

1.h.Scale out - Replicate Data Globally

Settings-> Replicate Data Globally



1. i . cosmosdb1acc | Default consistency



1.j. Stored Proc

Create a new sp

Predefined sp

```
// SAMPLE STORED PROCEDURE
function sample(prefix) {
    var collection = getContext().getCollection();

    // Query documents and take 1st item.
    var isAccepted = collection.queryDocuments(
        collection.getSelfLink(),
        'SELECT * FROM root r',
    function (err, feed, options) {
        if (err) throw err;

        // Check the feed and if empty, set the body to 'no docs found',
        // else take 1st element from feed
        if (!feed || !feed.length) {
            var response = getContext().getResponse();
            response.setBody('no docs found');
        }
        else {
            var response = getContext().getResponse();
            var body = { prefix: prefix, feed: feed[0] };
            response.setBody(JSON.stringify(body));
        }
    });

    if (!isAccepted) throw new Error('The query was not accepted by the server.');
```

```
}
```

2. Az204-lab 5.1-scholarhat-Creating a Cosmos DB Account - SQL API

Creating a Cosmos DB Account - SQL API

Mentor: Shailendra Chauhan
Type : GuidedLab
Points : 10
Duration : 00:50:00

Lab Details

Description

In this lab, you will be learning how to create an Azure Cosmos DB Account and then work with the SQL API.

Lab Objective

Upon completion of this lab, you will be able to:

- Create an Azure Cosmos DB Account.
- Create a container within Cosmos DB Account.
- Add Default data into the table.
- Querying the added data.

Prerequisites

You should be familiar with:

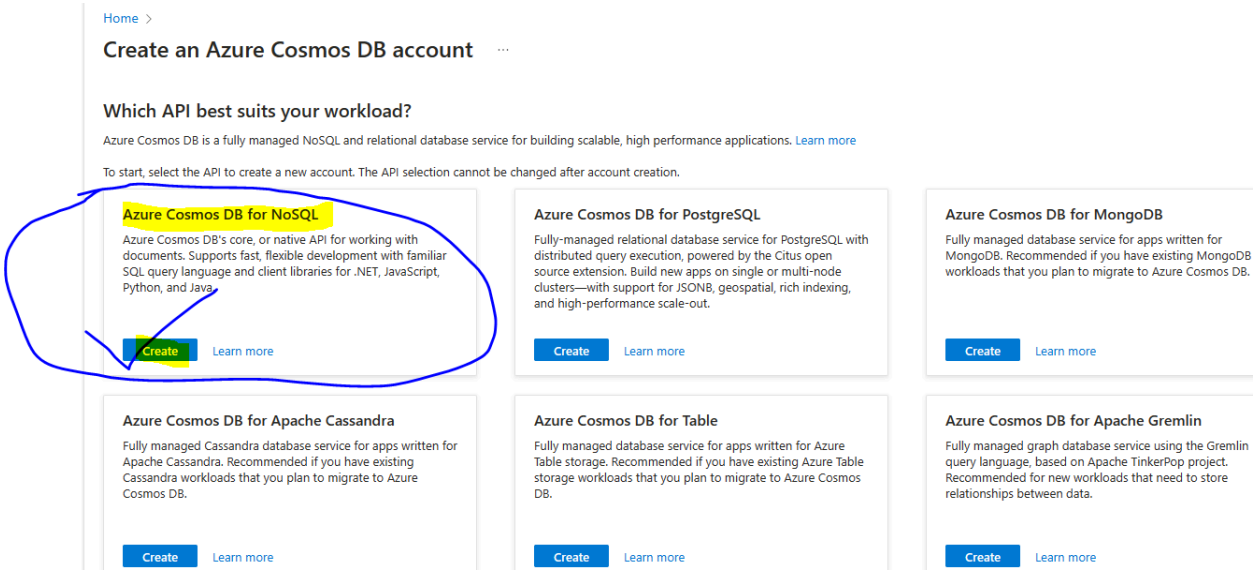
- Conceptual Understanding of Azure Cosmos DB.
- You must have the Azure Portal Access.

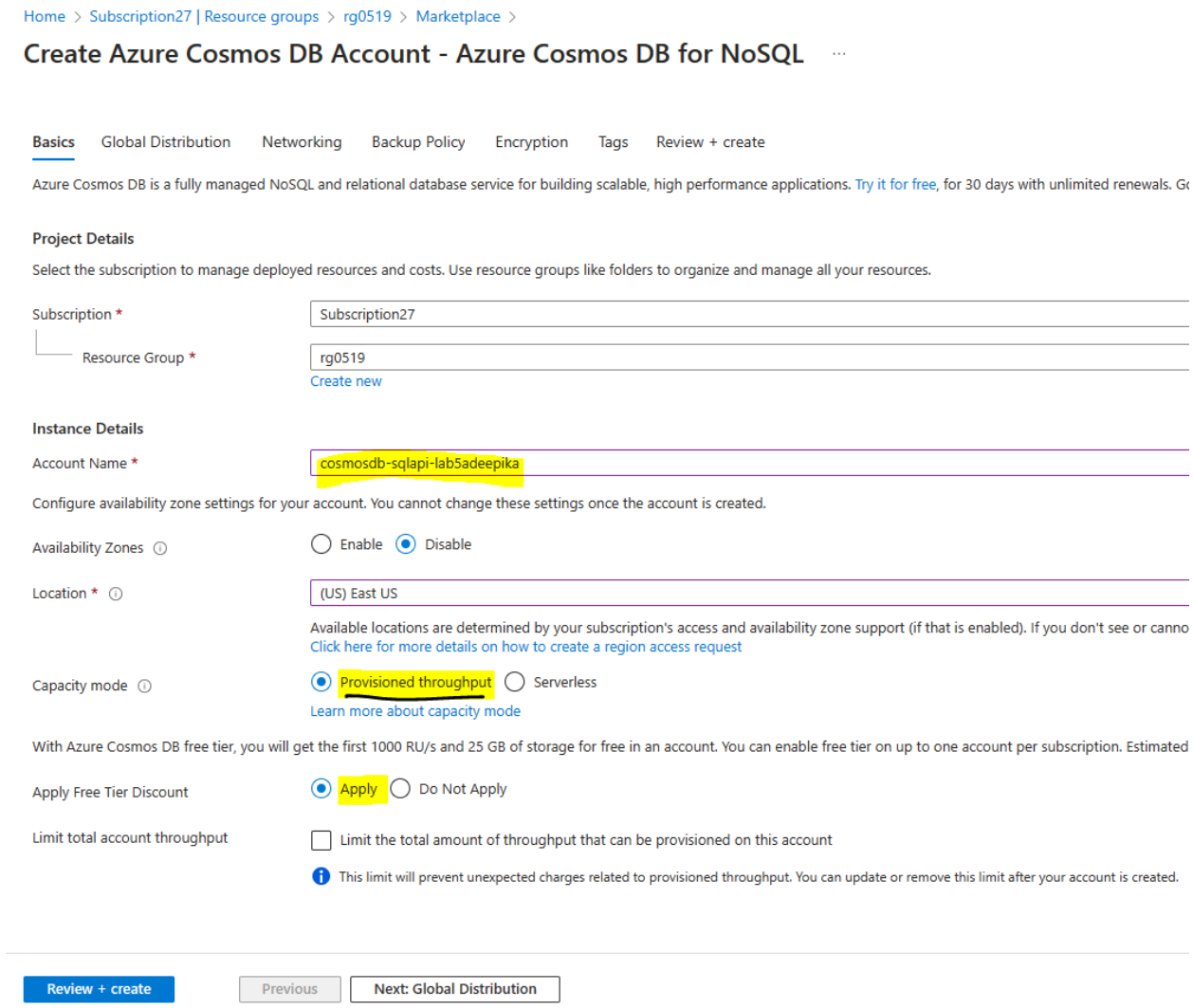
Lab Solution

Azure Cosmos DB is a globally distributed, low latency, multi-model database for managing data at large scales. It is a cloud-based NoSQL database offered as a PaaS (Platform as a Service) from Microsoft Azure. It is a highly available, high throughput, reliable database and is often called a serverless database. Cosmos database contains the Azure Document DB and is available everywhere.

Azure Cosmos DB SQL API accounts provide support for querying items using the Structured Query Language (SQL), one of the most familiar and popular query languages, as a JSON query language.

- Step 1: Open the Azure management portal and log in to https://portal.azure.com
- Step 2: Click on “Create a Resource”.

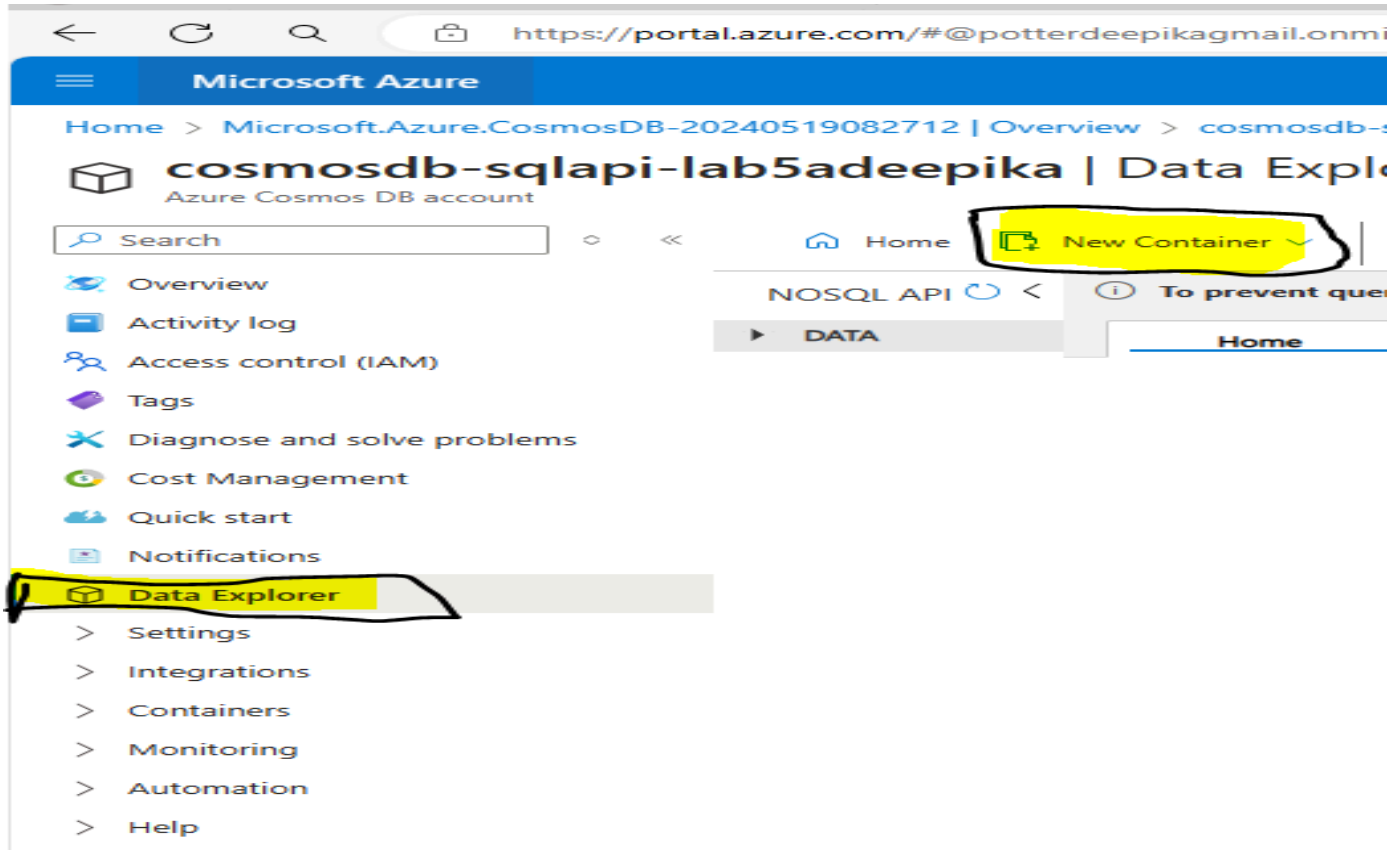




Step 3: In the search bar, search for “Azure Cosmos DB” and select it from the search results.
Step 4: Click on the “Create” button.
Step 5: Click on the “Create” button under Core (SQL) - Recommended as shown below.
Step 6: After clicking create button, Fill in the required details:

1. Subscription: Choose a suitable subscription.
2. Resource group: Resource group is a container that holds related resources for an Azure solution. You can keep the resource group as it is or you can also create one by clicking “Create New”.
3. Account name: Mention your account name: “cosmosdb-sqlapi-lab5adeepika”

Step 7: Click on “Review + Create”. Azure will authenticate the details you’ve filled so far and will show a message as “Validation Success”. Afterwards, click on the “Create” button.



Step 8: After clicking the CREATE button. A message will be displayed – “Your deployment is complete”. Then click on “Go to resource”.

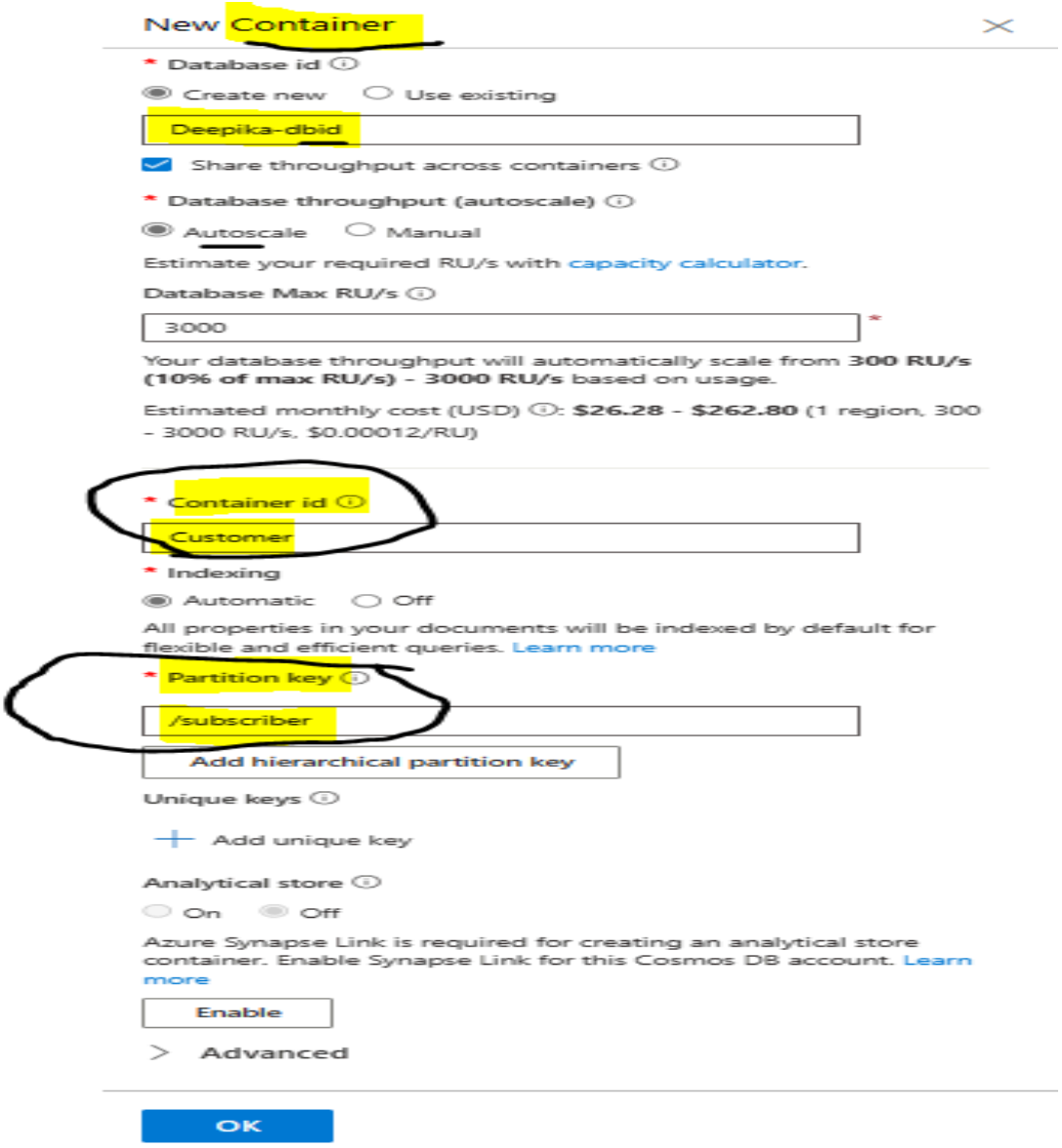
Step 9: After Clicking on “Go to resource”, select “Data Explorer” from the left pane. Then, click on “New Container”.

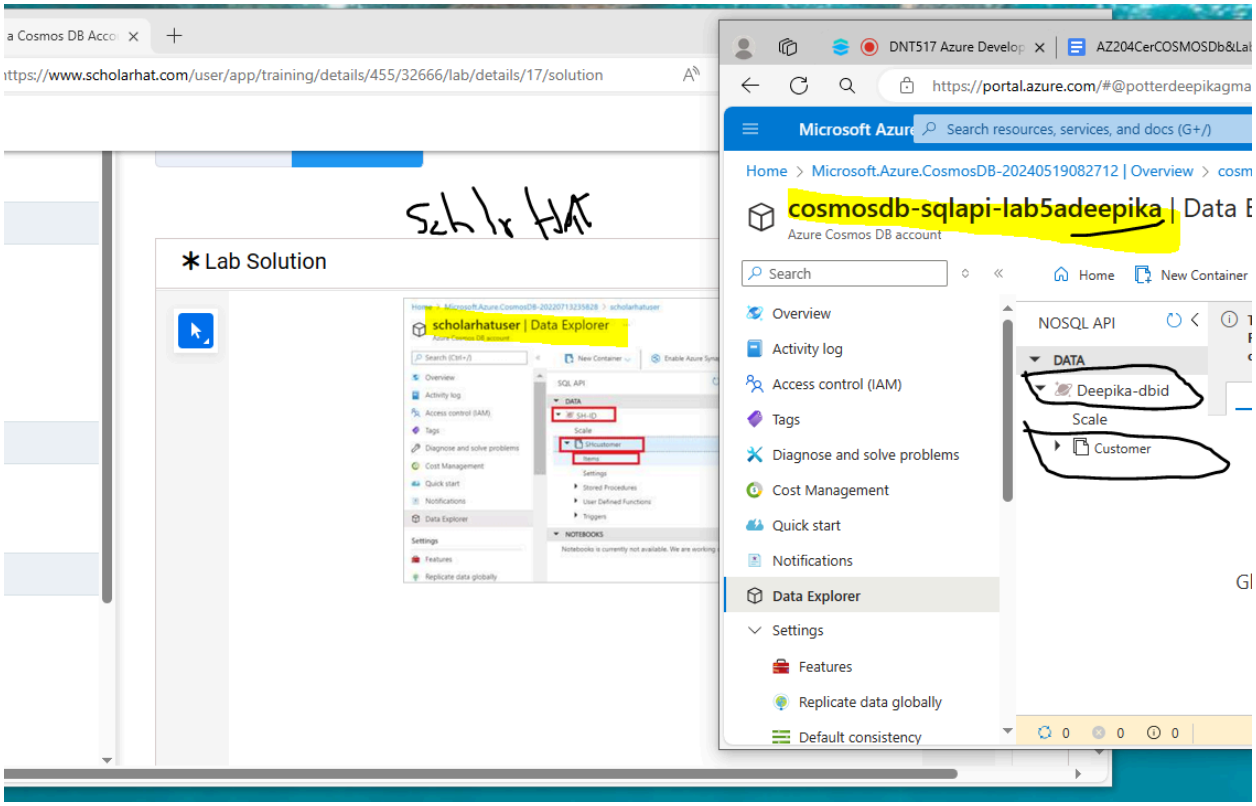
Step 10: A window will be displayed on your right. Enter the “Database Id”, “Customer Id” and “/partition key”. Then, click on the “OK” button.

“Database Id” : Deepika-dbid

“Customer Id” : Customer

“/partition key” : /subscriber





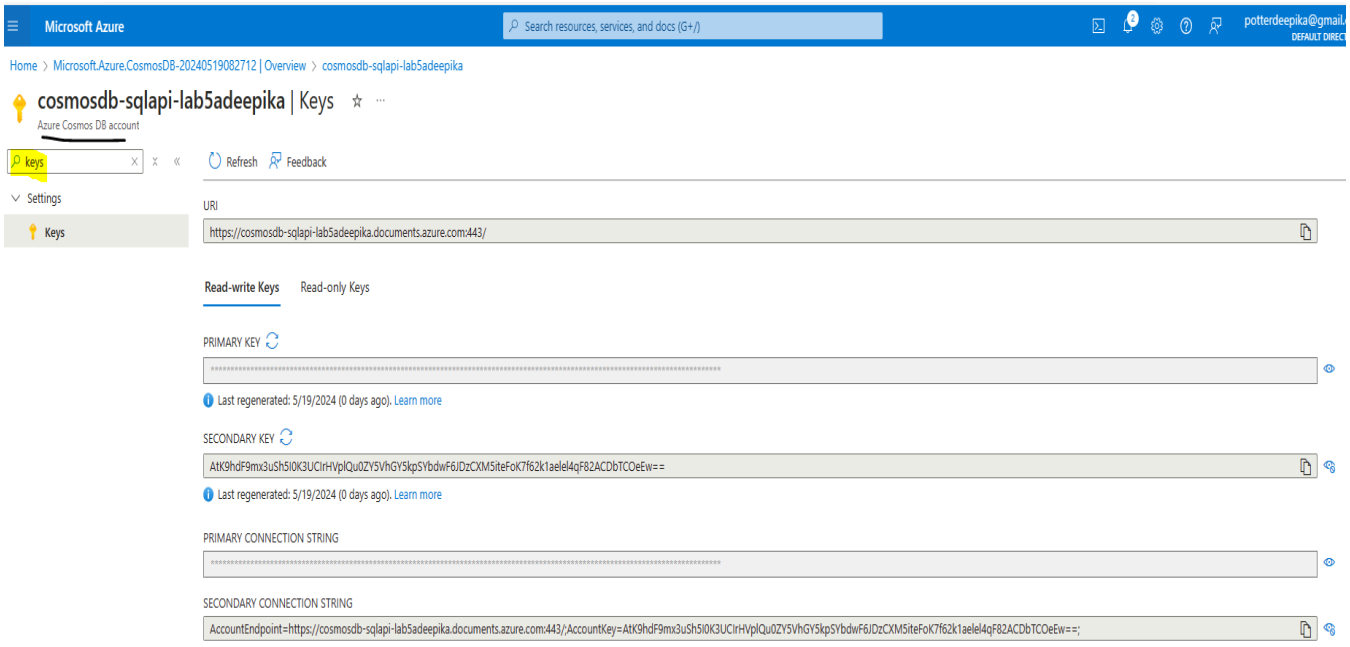
Step 11: Go to keys menu from left hand menus and copy the primary connection string

Step 12: Select “Data Explorer” from the left pane. Then, Click on SH-ID (your database Id) > SHcustomer > Items. The menu will expand.

“Database Id” : Deepika-dbid

“Customer Id” : Customer

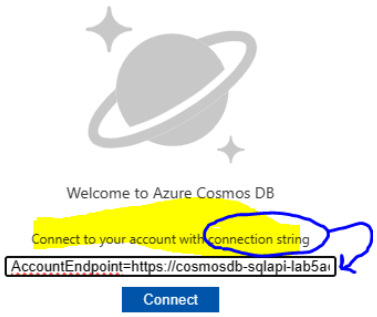
“/partition key” : /subscriber



SECONDARY CONNECTION STRING

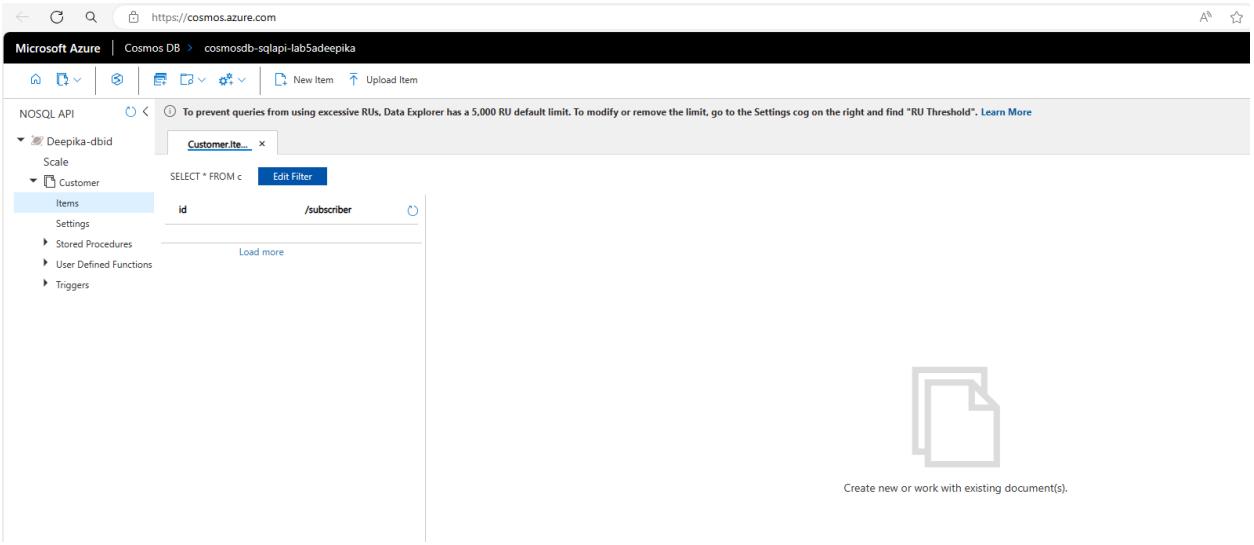
AccountEndpoint=https://cosmosdb-sqlapi-lab5adeepika.documents.azure.com:443/;AccountKey=AtK9hdF9mx3uSh5l0K3UCIrHVplQu0ZY5VhGY5kpSYbdwF6JDzCXM5iteFoK7f62k1aelel4qF82ACDbTCOeEw==;

Using connection string connect it for full string



Step 13: Click on the "Open Full Screen" option in the top right corner.
Step 14: Click on the "Open" button under Read and Write.
Step 15: New tab will open,paste the copied key in above steps and click on connect button.
Step 16: As soon as you'll click on "connect", it will be redirected to a new tab in your browser.
Step 17: Now, in the new tab, you'll see "Deepika-dbid" in your left pane. Click on Deepika-dbid > Customer > Items
"Database Id" : Deepika-dbid
"Customer Id" : Customer
"/partition key" : /subscriber

Now in Maximize mode



Step 18: Select the “New Item” option in the top pane.

Remember: scholarhat scholarhat(SH-ID > SHcustomer > Items.)

Step 19: Delete whatever is written on the screen. Copy the below command and modify it according to your database id and partition key that you’ve entered above.

```
{
"id": "1",
"Name": "ScholarhatUser1",
"scholarhatsubscriber": "New York"
}
```

“Database Id” : Deepika-dbid
“Customer Id” : Customer
“/partition key” : /subscriber

Derived:Deepika


```
{
"id": "1",
"Name": "ScholarhatUser1",
"subscriber": "New York"
}
```


Step 20: Now, again click on “New Item”.

● Step 21: Perform step 16 again to enter the second record into the database. Delete whatever is written and paste that command. Remember to do the changes according to your needs. Click on “Save” to save the data.

Step 22: Now, you are going to query the added data. For that, Click on the “New SQL query” icon as highlighted in the below image.

Step 23: You’ll see that there’s a default query written. In case it’s not showing, you can enter the below command to display all the records that you’ve stored above.

SELECT * FROM c Click on the “Execute Query” option in the top pane.

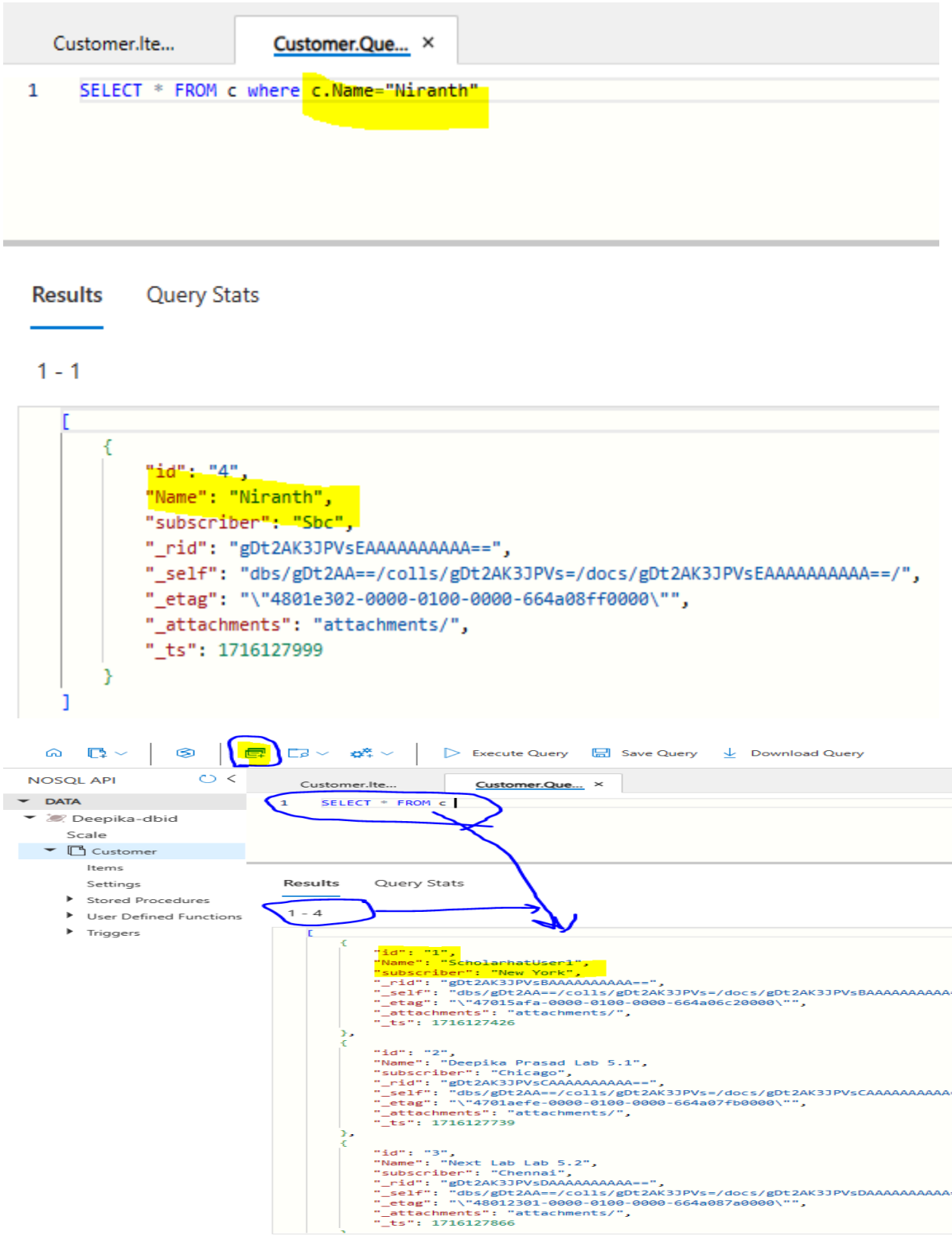

```
{
"id": "2",
"Name": "Deepika Prasad Lab 5.1",
"subscriber": "Chicago"
}
{
"id": "3",
"Name": "Next Lab Lab 5.2",
"subscriber": "Chennai"
}
{
```

"id": "4",
"Name": "Niranth",
"subscriber": "Sbc"
}

Filter records with c.Name or c.id or c.subscriber
Input: select * from c where c.Name=”Niranth”

Step 24: As soon as you’ll click “Execute Query”, All the records will be displayed as you scroll down the
page.

Success output:



------***End of lab 5.1- Cosmos DB- SQL API***------

3. Az204-lab 5.2-scholarhat-Creating a Cosmos DB account - Table API

Creating a Cosmos DB account - Table API

Level : Beginner

Mentor: Shailendra Chauhan

Type : GuidedLab

Points : 10

Duration : 00:50:00

Lab Problem Lab Solution

Lab Details
Description

In this lab, you will be learning how to create an Azure Cosmos DB Account and then work with the Table API.

Lab Objective

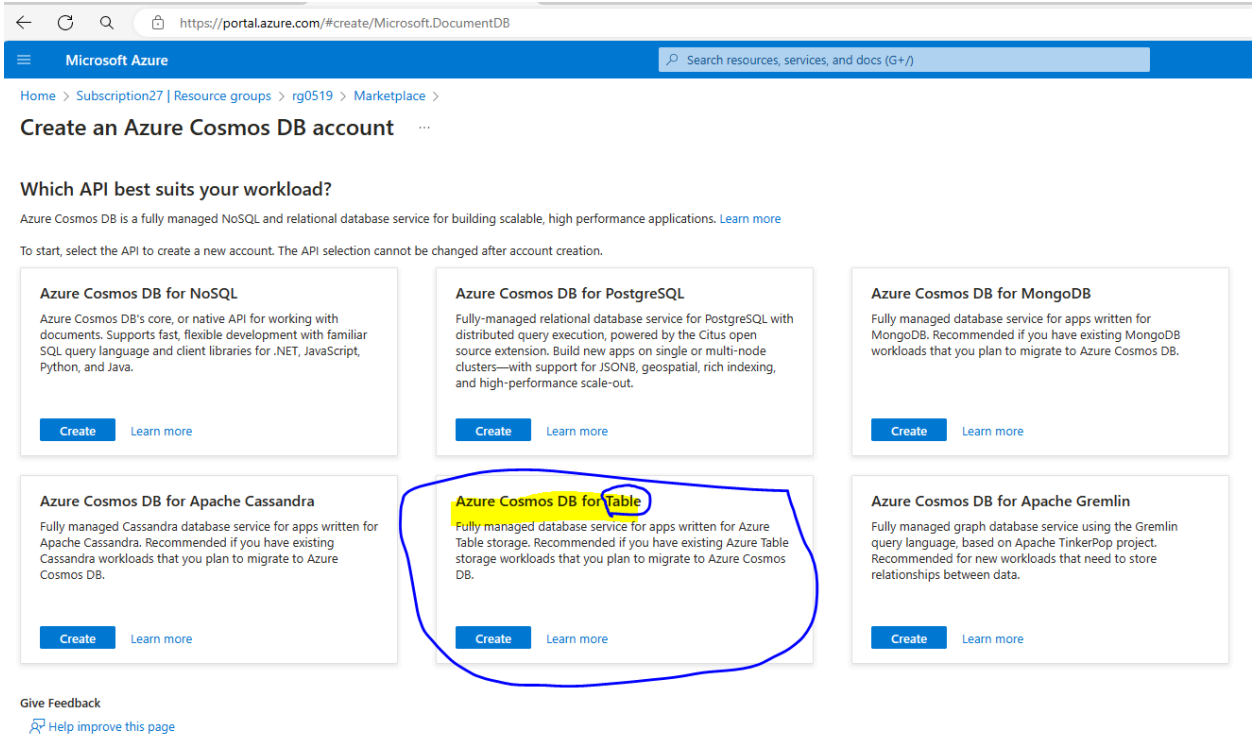
Upon completion of this lab, you will be able to:

- Create an Azure Cosmos DB Account.
- Create a new table within Cosmos DB Account.
- Add Default data into the table.

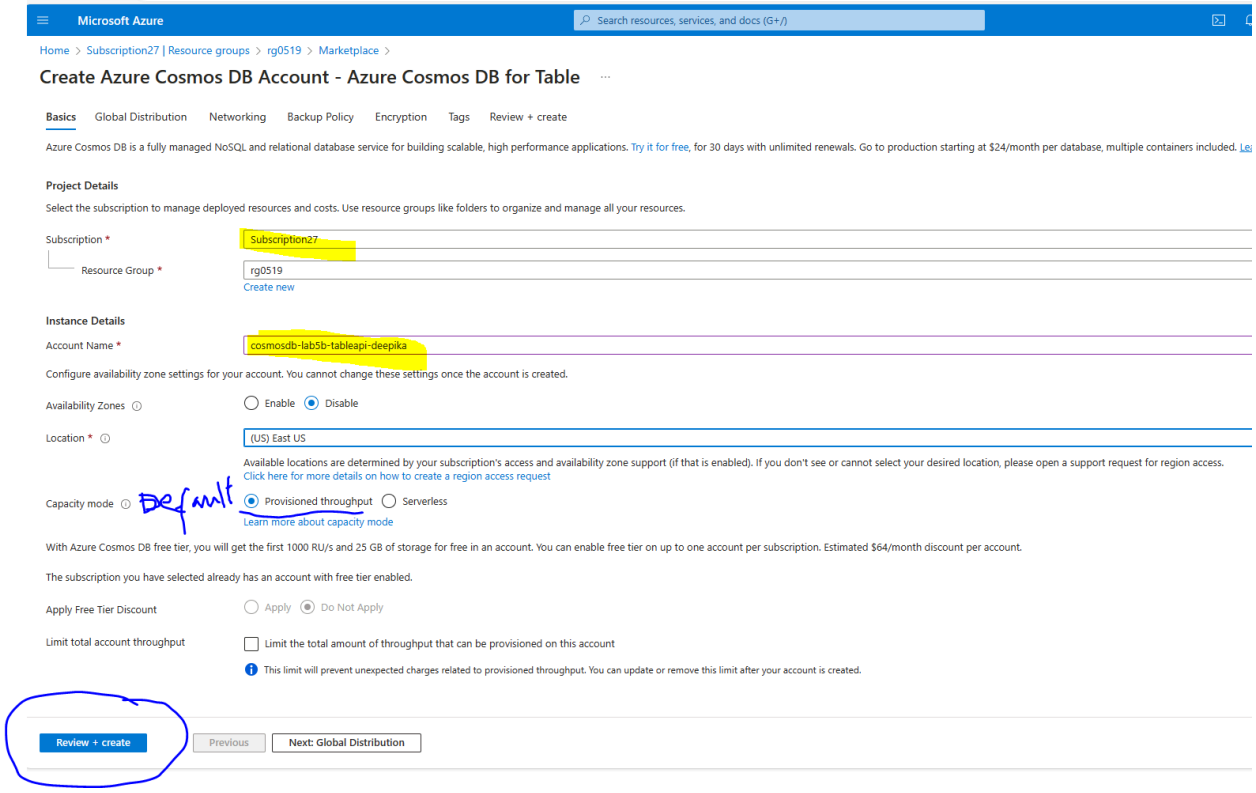
Prerequisites

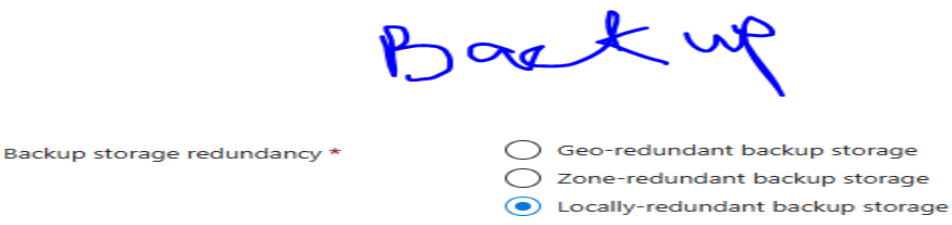
You should be familiar with:

- Conceptual Understanding of Azure Cosmos DB.
- You must have the Azure Portal Access.

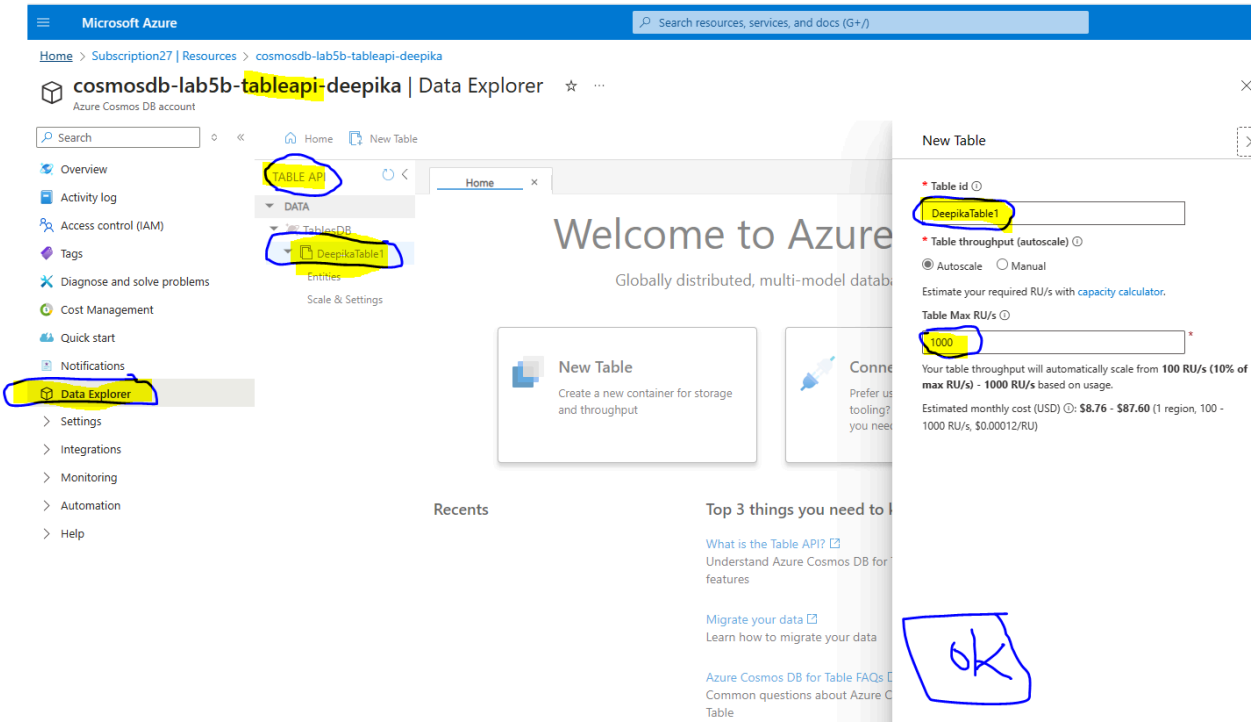


Create a table api

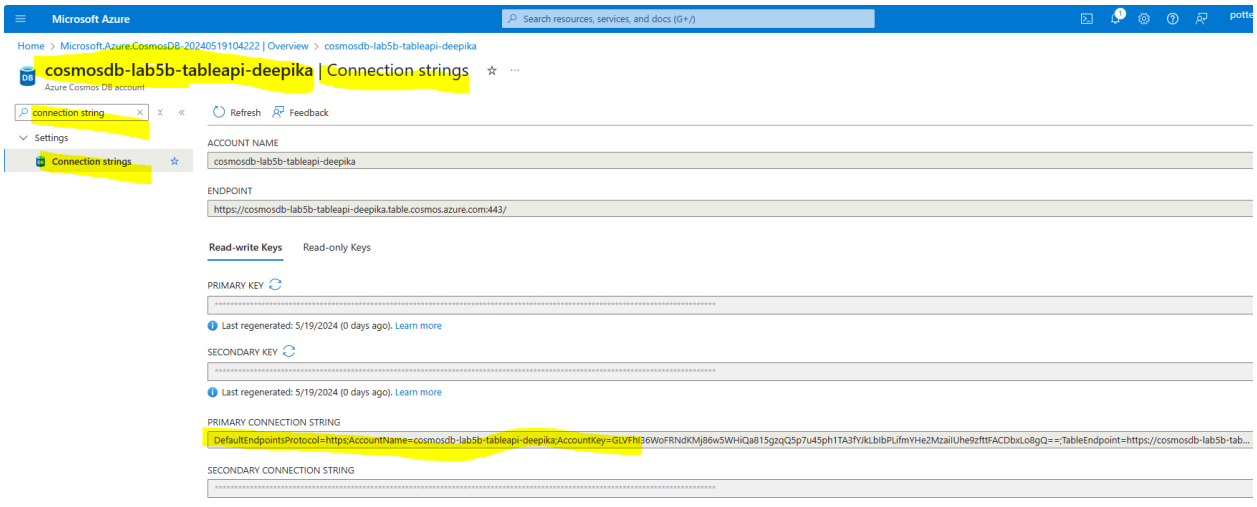




After Clicking on “Go to resource”, select “Data Explorer” from the left pane. Then, click on “New Table”.



On the left pane, type connection string and get the connection strings.



Step 11: Now, you are going to add default data to the table. For that, click on the “Open Full Screen” icon on the top right corner of the window.

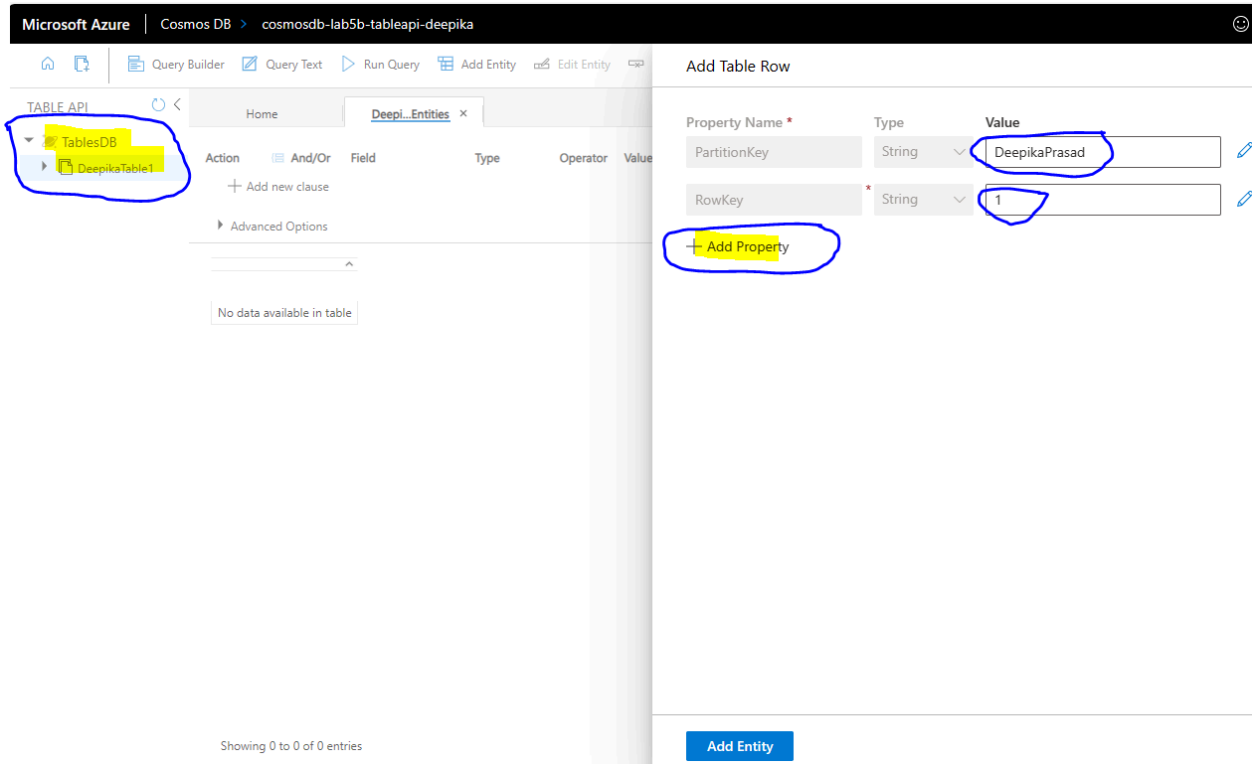
Welcome to Azure Cosmos DB

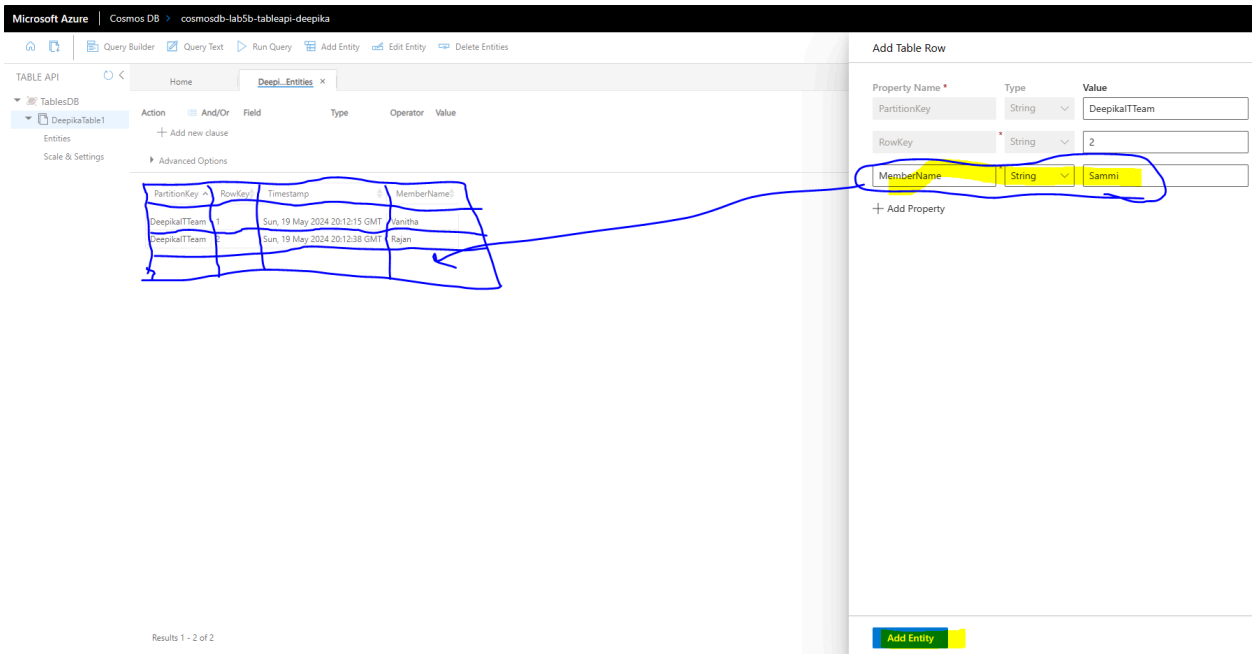
Connect to your account with connection string

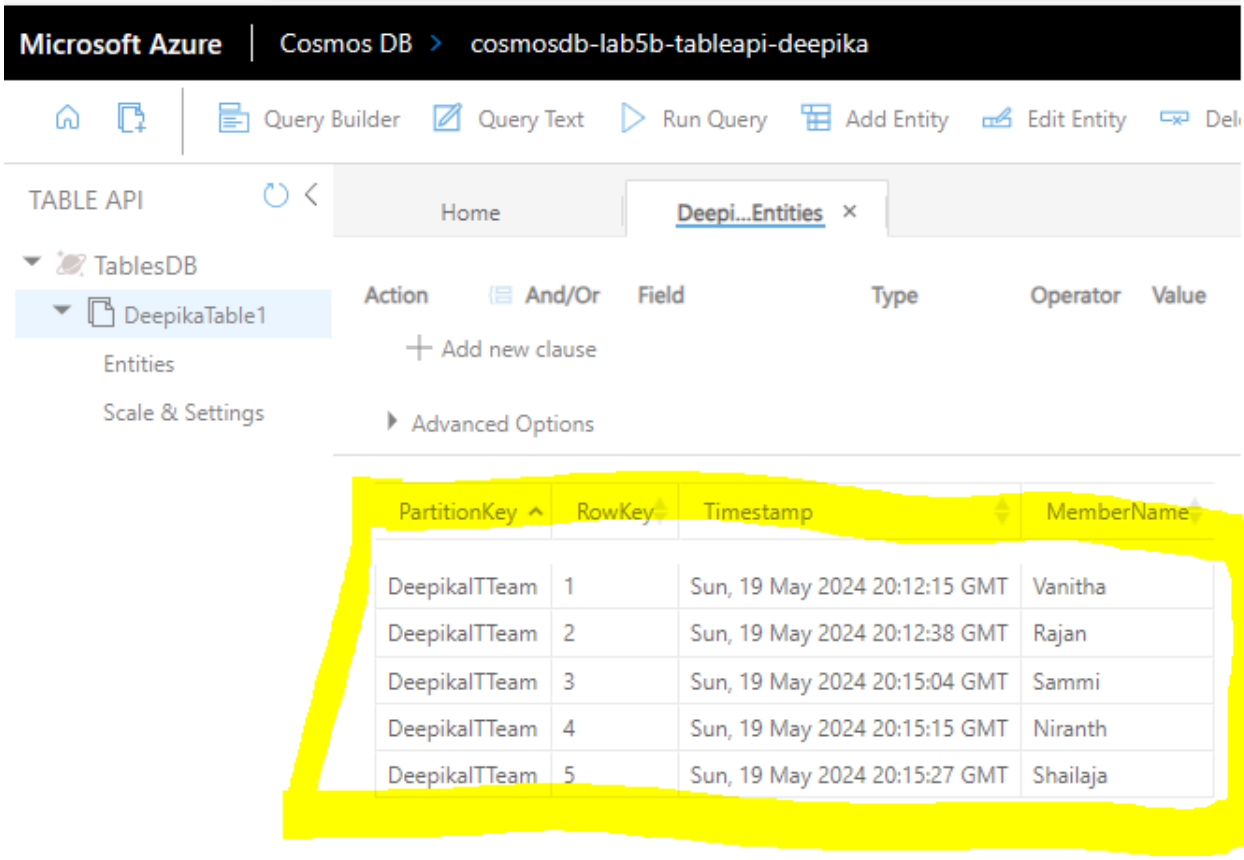
DefaultEndpointsProtocol=https;AccountName=c

Connect

Sign In with Azure Account


Microsoft Azure
Cosmos DB
cosmosdb-lab5b-tableapi-deepika
Query Builder
Query Text
Run Query
Add Entity
Edit Entity
TABLE API
TablesDB
DeepikaTable1
Home
Deepi...Entities
Action
And/Or
Field
Type
Operator
Value
Add new clause
Advanced Options
No data available in table
Showing 0 to 0 of 0 entries
Add Table Row
Property Name *
Type
Value
PartitionKey
String
DeepikaPrasad
RowKey
String
1
Add Property
Add Entity


Microsoft Azure
Cosmos DB
cosmosdb-lab5b-tableapi-deepika
Query Builder
Query Text
Run Query
Add Entity
Edit Entity
Delete Entities
TABLE API
TablesDB
DeepikaTable1
Entities
Scale & Settings
Home
Deepi...Entities
Action
And/Or
Field
Type
Operator
Value
Add new clause
Advanced Options
PartitionKey
RowKey
Timestamp
MemberName
DeepikaITTeam
1
Sun, 19 May 2024 20:12:15 GMT
Vanitha
DeepikaITTeam
2
Sun, 19 May 2024 20:12:38 GMT
Rajan
Results 1 - 2 of 2
Add Table Row
Property Name *
Type
Value
PartitionKey
String
DeepikaITTeam
RowKey
String
2
MemberName
String
Sammi
Add Property
Add Entity


Microsoft Azure
Cosmos DB
cosmosdb-lab5b-tableapi-deepika
Query Builder
Query Text
Run Query
Add Entity
Edit Entity
TABLE API
TablesDB
DeepikaTable1
Entities
Scale & Settings
Home
Deepi...Entities
Action
And/Or
Field
Type
Operator
Value
Add new clause
Advanced Options
PartitionKey
RowKey
Timestamp
MemberName
DeepikaITTeam 1 Sun, 19 May 2024 20:12:15 GMT Vanitha
DeepikaITTeam 2 Sun, 19 May 2024 20:12:38 GMT Rajan
DeepikaITTeam 3 Sun, 19 May 2024 20:15:04 GMT Sammi
DeepikaITTeam 4 Sun, 19 May 2024 20:15:15 GMT Niranth
DeepikaITTeam 5 Sun, 19 May 2024 20:15:27 GMT Shailaja

—--**End of lab 5.2****—---**