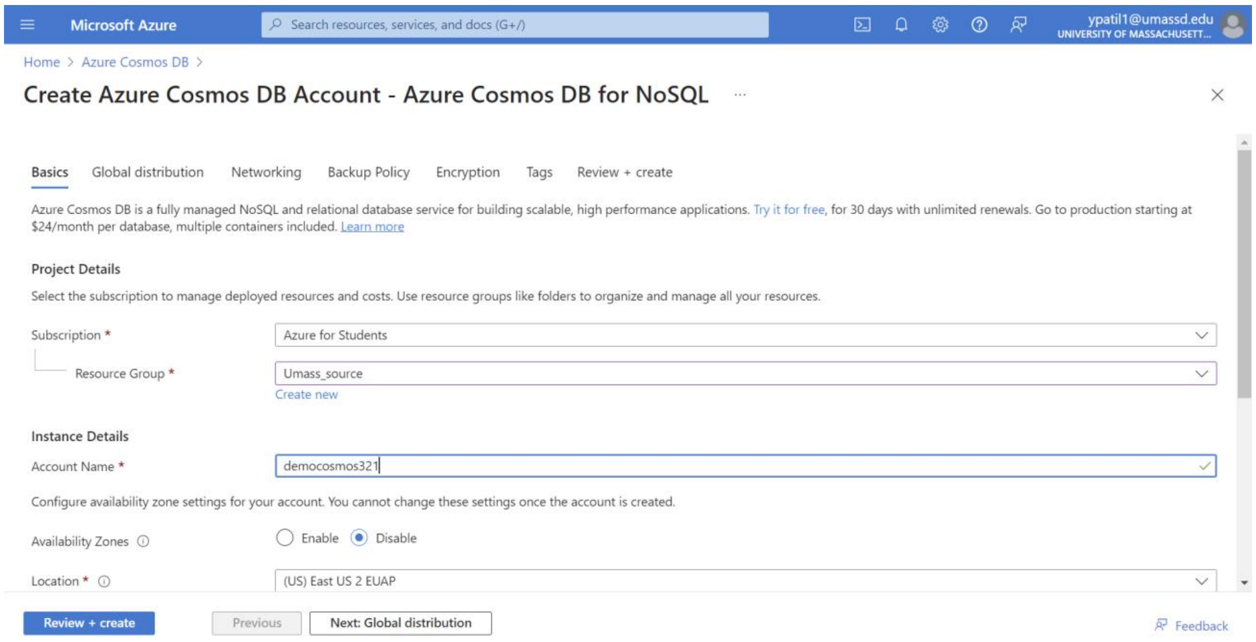
CIS 552: DATABASE DESIGN

LAB HOMEWORK 6

Submitted By – Yashika Patil

Student ID – 02115374

1. **Creating Cosmos DB Account:** In the Azure account home, search for Cosmos DB and create a cosmos DB account for NoSQL. This step involves setting up the environment required to store and manage NoSQL data using Azure Cosmos DB.


Microsoft Azure
Search resources, services, and docs (G+/)
ypatil1@umassd.edu
UNIVERSITY OF MASSACHUSETT...
Home > Azure Cosmos DB >

Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. Try it for free, for 30 days with unlimited renewals. Go to production starting at $24/month per database, multiple containers included. Learn more

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Azure for Students
Resource Group *	Umass_source
	Create new

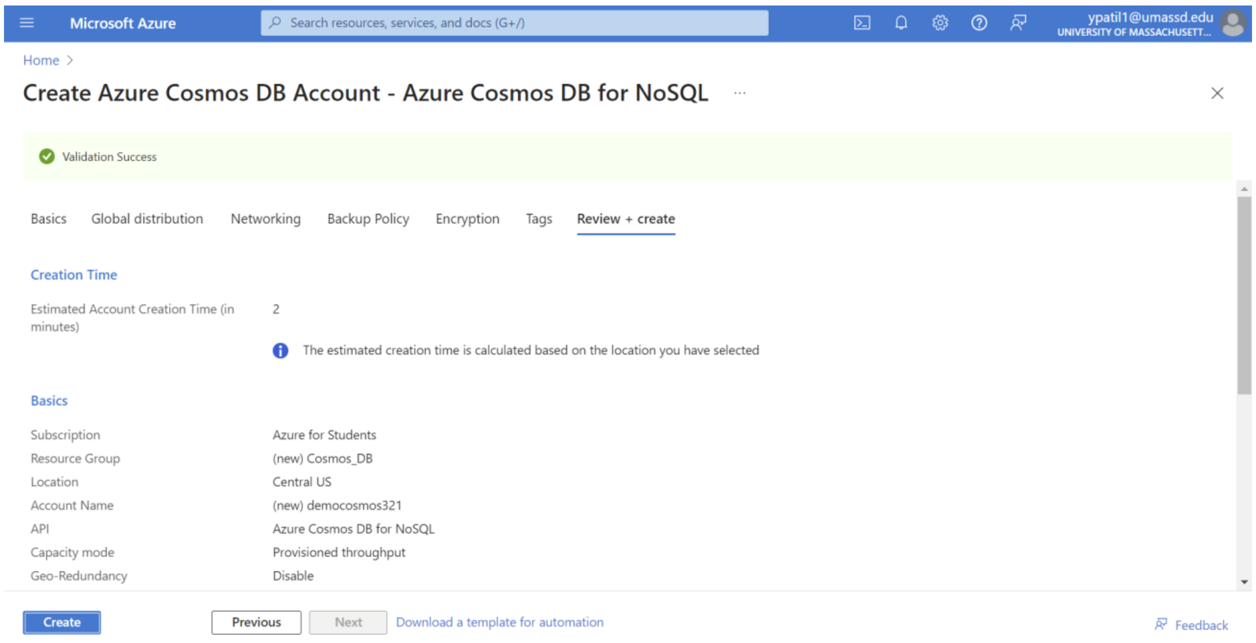
Instance Details

Account Name *	democosmos321

Configure availability zone settings for your account. You cannot change these settings once the account is created.

Availability Zones	Enable [✓ Disable]
Location *	(US) East US 2 EUAP

2. **Deployment:** Click on "Create" and wait for the deployment to complete. This step ensures the creation and deployment of the Cosmos DB account, which might take a few minutes to set up.


Microsoft Azure
Search resources, services, and docs (G+/)
ypatil1@umassd.edu
UNIVERSITY OF MASSACHUSETT...
Home >

Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL

Validation Success

Creation Time

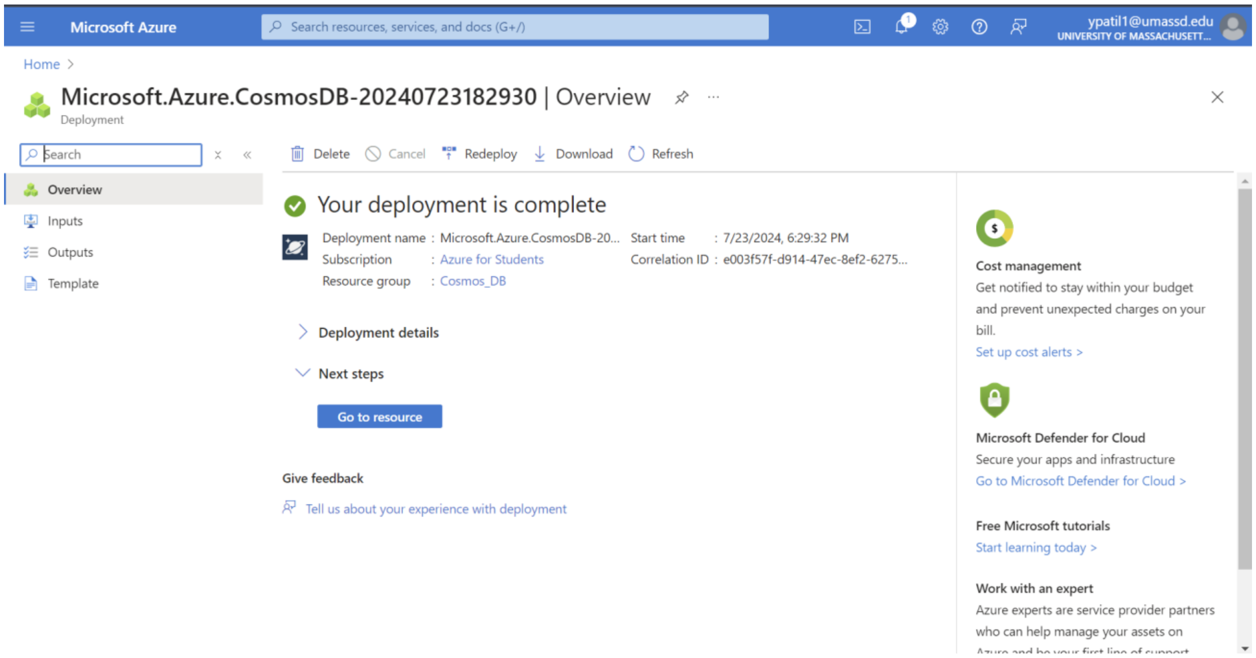
Estimated Account Creation Time (in minutes)	2

The estimated creation time is calculated based on the location you have selected

Basics

Subscription	Azure for Students
Resource Group	(new) Cosmos_DB
Location	Central US
Account Name	(new) democosmos321
API	Azure Cosmos DB for NoSQL
Capacity mode	Provisioned throughput
Geo-Redundancy	Disable

3. The deployment takes a few minutes to complete.



Microsoft Azure — Search resources, services, and docs (G+/) — ypatil1@umassd.edu UNIVERSITY OF MASSACHUSETT...

Home >

Microsoft.Azure.CosmosDB-20240723182930 | Overview

Deployment

Delete | Cancel | Redeploy | Download | Refresh

Overview, Inputs, Outputs, Template

Your deployment is complete

Deployment name : Microsoft.Azure.CosmosDB-20...
Subscription : Azure for Students
Resource group : Cosmos_DB
Start time : 7/23/2024, 6:29:32 PM
Correlation ID : e003f57f-d914-47ec-8ef2-6275...

Deployment details

Next steps

Go to resource

Give feedback

Tell us about your experience with deployment

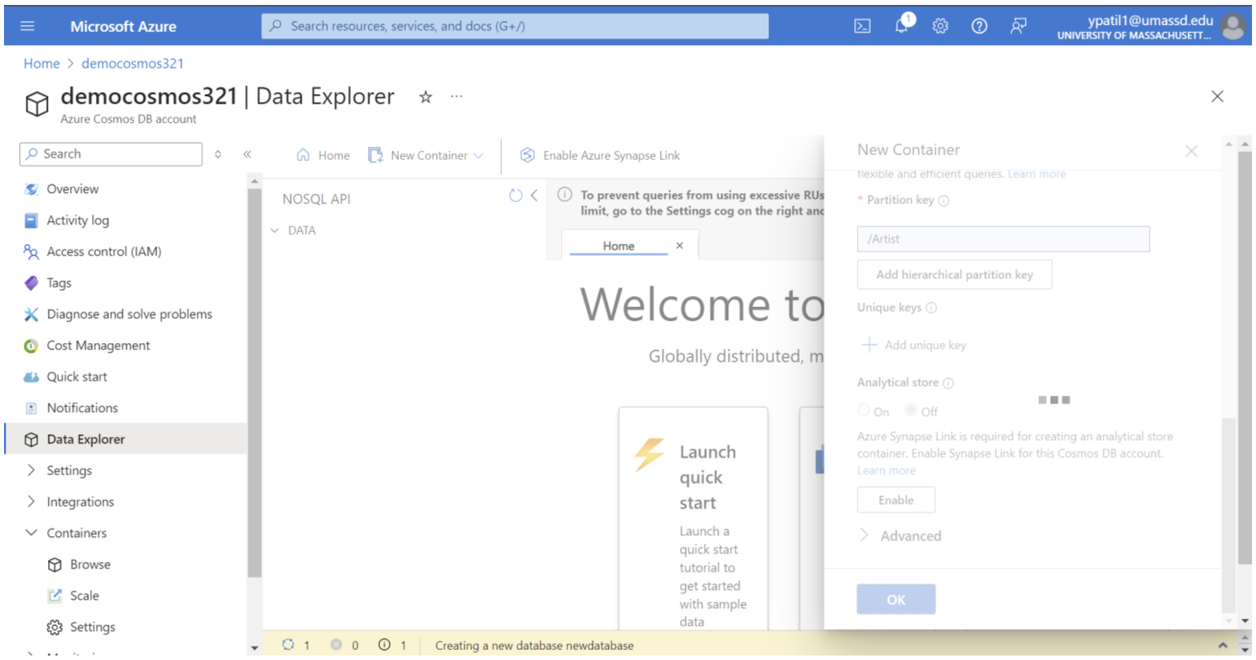
Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alerts >

Microsoft Defender for Cloud
Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on

4. **Creating a New Container:** Now, go to 'Data Explorer' and create a new container. Give a database name and create container with the name 'Music' and partition key 'Artist'.



Microsoft Azure — Search resources, services, and docs (G+/) — ypatil1@umassd.edu UNIVERSITY OF MASSACHUSETT...

Home > democosmos321

democosmos321 | Data Explorer

Azure Cosmos DB account

Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Cost Management, Quick start, Notifications, Data Explorer, Settings, Integrations, Containers, Browse, Scale, Settings

Home | New Container | Enable Azure Synapse Link

NOSQL API — DATA

To prevent queries from using excessive RUs limit, go to the Settings cog on the right and

Welcome to

Globally distributed, m

Launch quick start — Launch a quick start tutorial to get started with sample data

New Container

flexible and efficient queries. Learn more

* Partition key

/Artist

Add hierarchical partition key

Unique keys

Add unique key

Analytical store

On Off

Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. Learn more

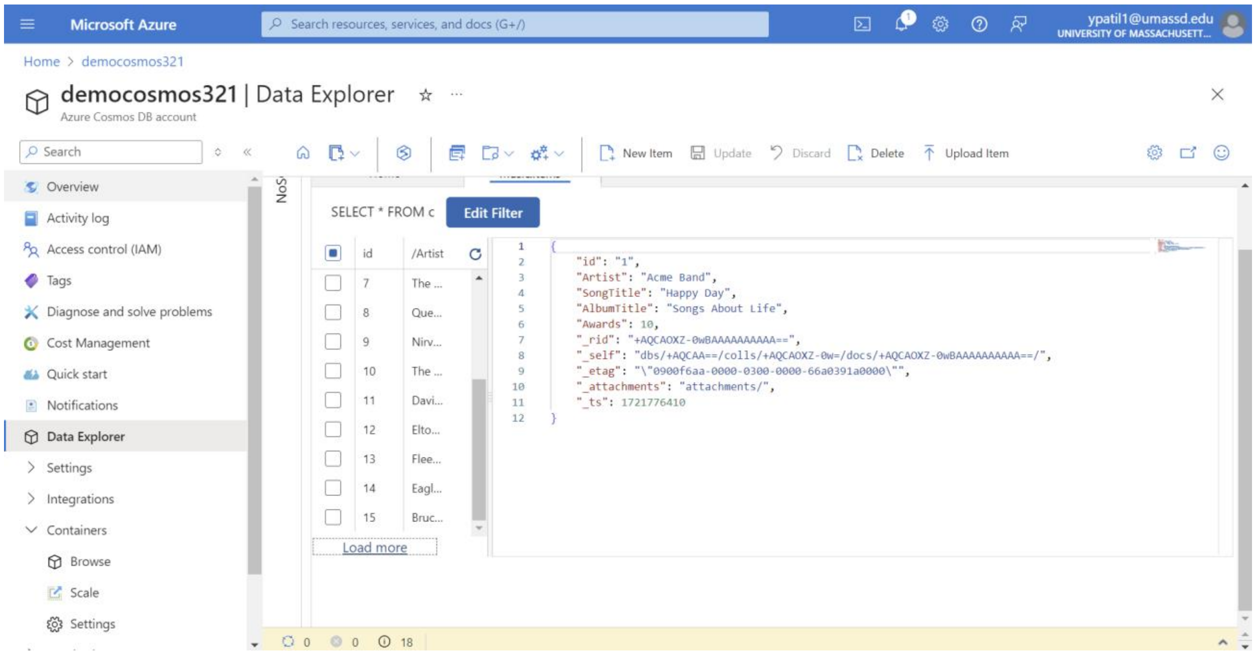
Enable

Advanced

OK

1 0 1 Creating a new database newdatabase

5. **Adding Items:** Add 15 such items with song name, artist name, and other details. This step involves populating the container with sample data that can be queried and manipulated.

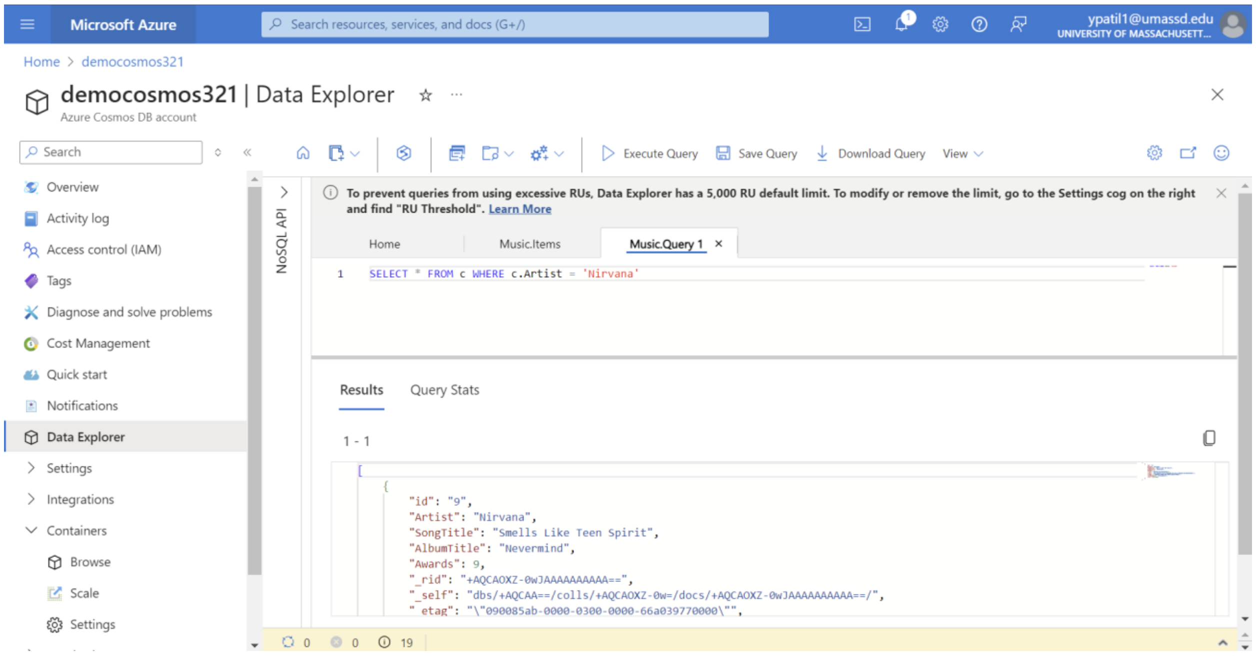


SELECT * FROM c

id	/Artist
7	The ...
8	Que...
9	Nirv...
10	The ...
11	Davi...
12	Elto...
13	Flee...
14	Eagl...
15	Bruc...

```
{
    "id": "1",
    "Artist": "Acme Band",
    "SongTitle": "Happy Day",
    "AlbumTitle": "Songs About Life",
    "Awards": 10,
    "_rid": "+AQCAOXZ-0wBAAAAAAAAAA==",
    "_self": "dbs/+AQCAA==/colls/+AQCAOXZ-0w=/docs/+AQCAOXZ-0wBAAAAAAAAAA==/",
    "_etag": "\"0900f6aa-0000-0300-0000-66a0391a0000\"",
    "_attachments": "attachments/",
    "_ts": 1721776410
}
```

6. **Querying Data:** Execute a query to retrieve details where the artist is "Nirvana."



To prevent queries from using excessive RUs, Data Explorer has a 5,000 RU default limit. To modify or remove the limit, go to the Settings cog on the right and find "RU Threshold". Learn More

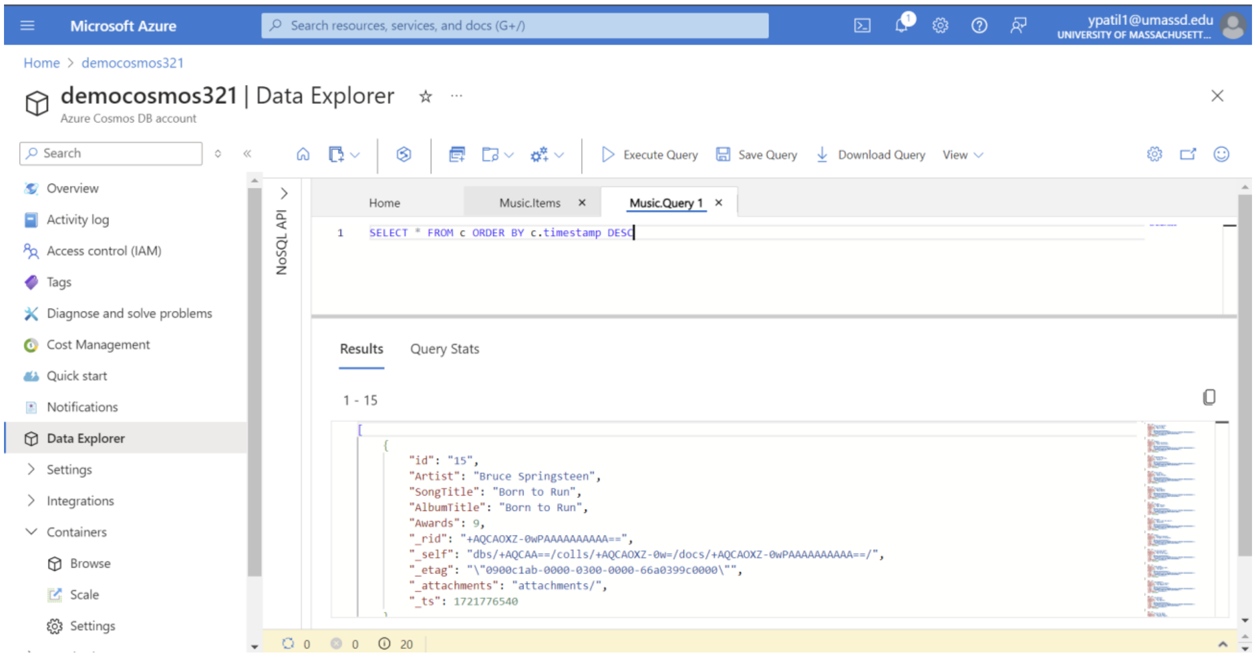
```
SELECT * FROM c WHERE c.Artist = 'Nirvana'
```

Results

1 - 1

```
[
    {
        "id": "9",
        "Artist": "Nirvana",
        "SongTitle": "Smells Like Teen Spirit",
        "AlbumTitle": "Nevermind",
        "Awards": 9,
        "_rid": "+AQCAOXZ-0wJAAAAAAAAAA==",
        "_self": "dbs/+AQCAA==/colls/+AQCAOXZ-0w=/docs/+AQCAOXZ-0wJAAAAAAAAAA==/",
        "_etag": "\"090085ab-0000-0300-0000-66a039770000\"",
```

7. **Ordering Items in Descending Order:** Execute a query to order the items in descending order.

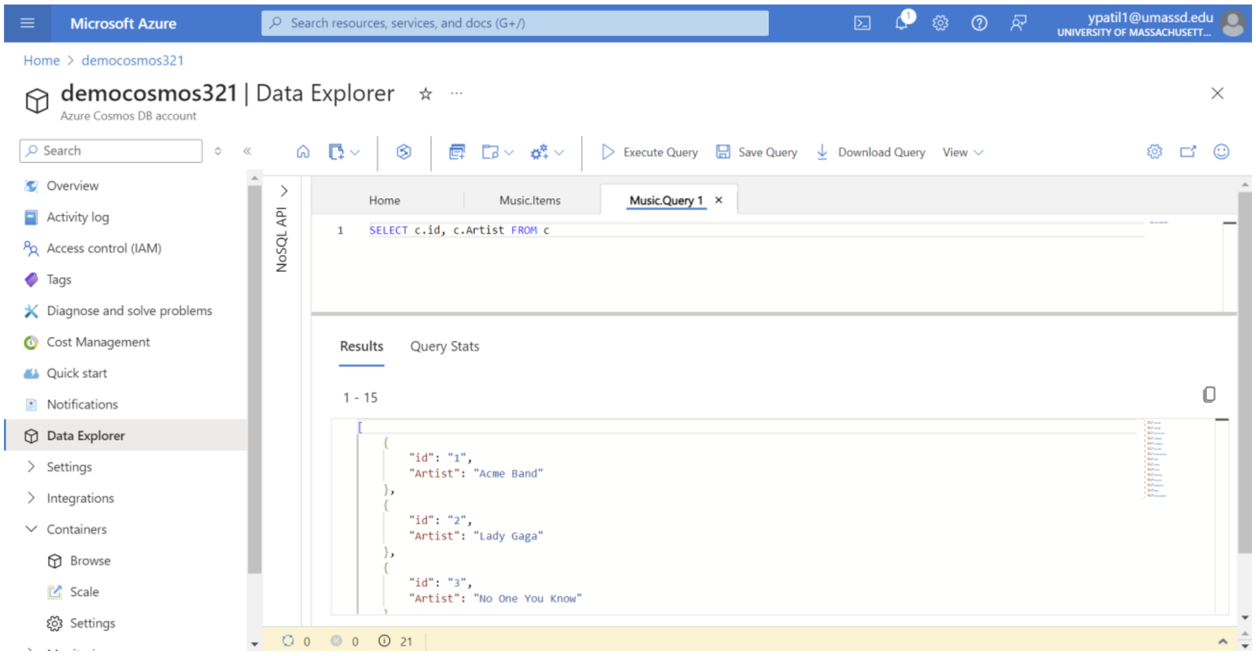


```
SELECT * FROM c ORDER BY c.timestamp DESC
```

Results 1 - 15

```
[
    {
        "id": "15",
        "Artist": "Bruce Springsteen",
        "SongTitle": "Born to Run",
        "AlbumTitle": "Born to Run",
        "Awards": 9,
        "_rid": "+AQCAOXZ-0wPAAAAAAAAAA==",
        "_self": "dbs/+AQCAA==/colls/+AQCAOXZ-0w=/docs/+AQCAOXZ-0wPAAAAAAAAAA==/",
        "_etag": "\"0900c1ab-0000-0300-0000-66a0399c0000\"",
        "_attachments": "attachments/",
        "_ts": 1721776540
```

8. Execute query to retrieve only the name and ID fields of all items.

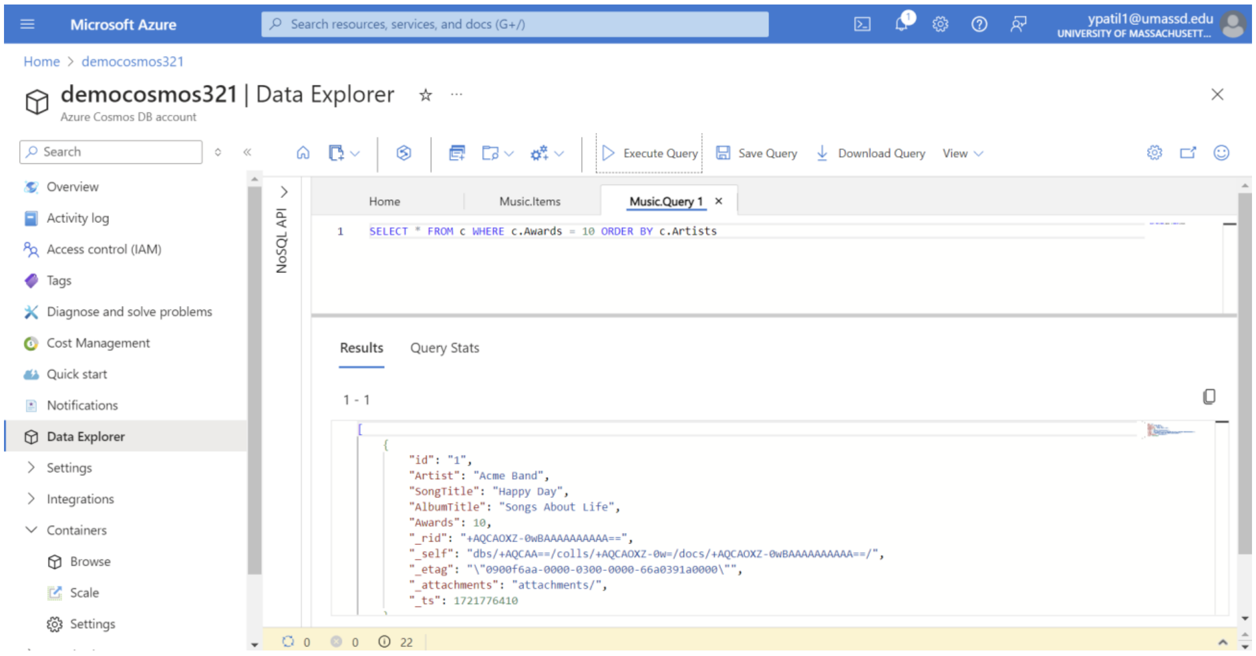


```
SELECT c.id, c.Artist FROM c
```

Results 1 - 15

```
[
    {
        "id": "1",
        "Artist": "Acme Band"
    },
    {
        "id": "2",
        "Artist": "Lady Gaga"
    },
    {
        "id": "3",
        "Artist": "No One You Know"
```

9. **Filtering and Sorting Data (Awards = 10):** To retrieve records of the songs with awards = 10 and sort them by the Artist’s names:

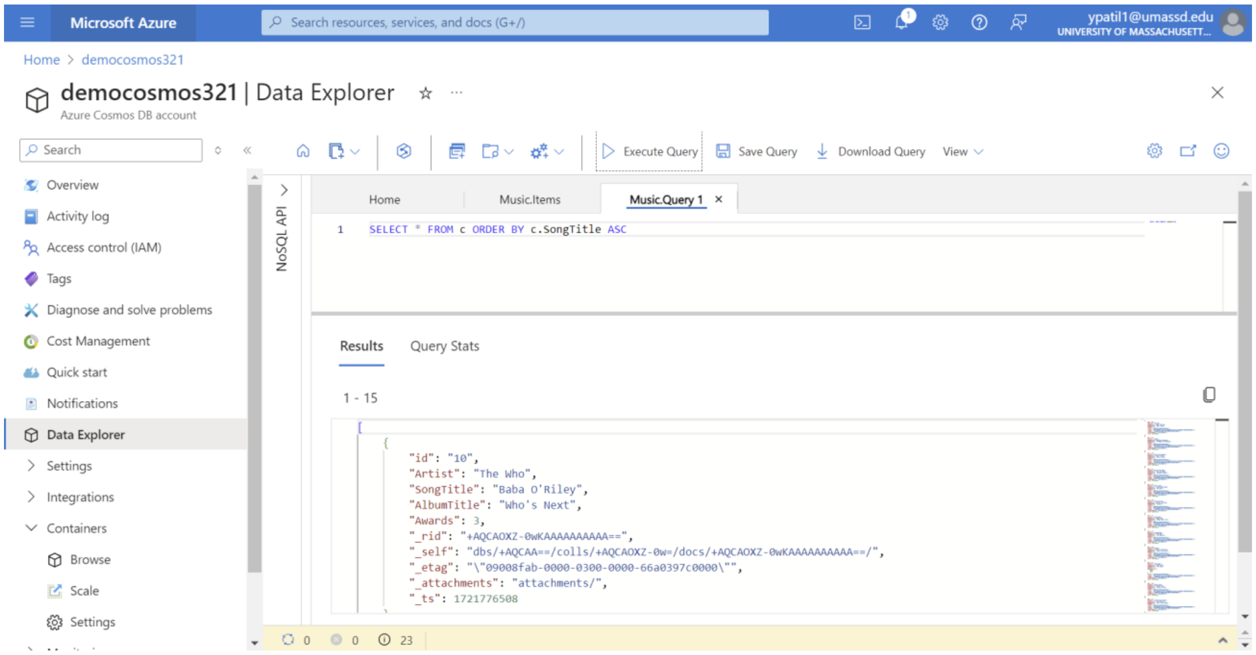


```
SELECT * FROM c WHERE c.Awards = 10 ORDER BY c.Artists
```

1 - 1

```
[
    {
        "id": "1",
        "Artist": "Acme Band",
        "SongTitle": "Happy Day",
        "AlbumTitle": "Songs About Life",
        "Awards": 10,
        "_rid": "+AQCAOXZ-0wBAAAAAAAAAA==",
        "_self": "dbs/+AQCAA==/colls/+AQCAOXZ-0w=/docs/+AQCAOXZ-0wBAAAAAAAAAA==/",
        "_etag": "\"0900f6aa-0000-0300-0000-66a0391a0000\"",
        "_attachments": "attachments/",
        "_ts": 1721776410
```

10. **Alphabetical Order:** To retrieve records for songs with titles in alphabetical order.

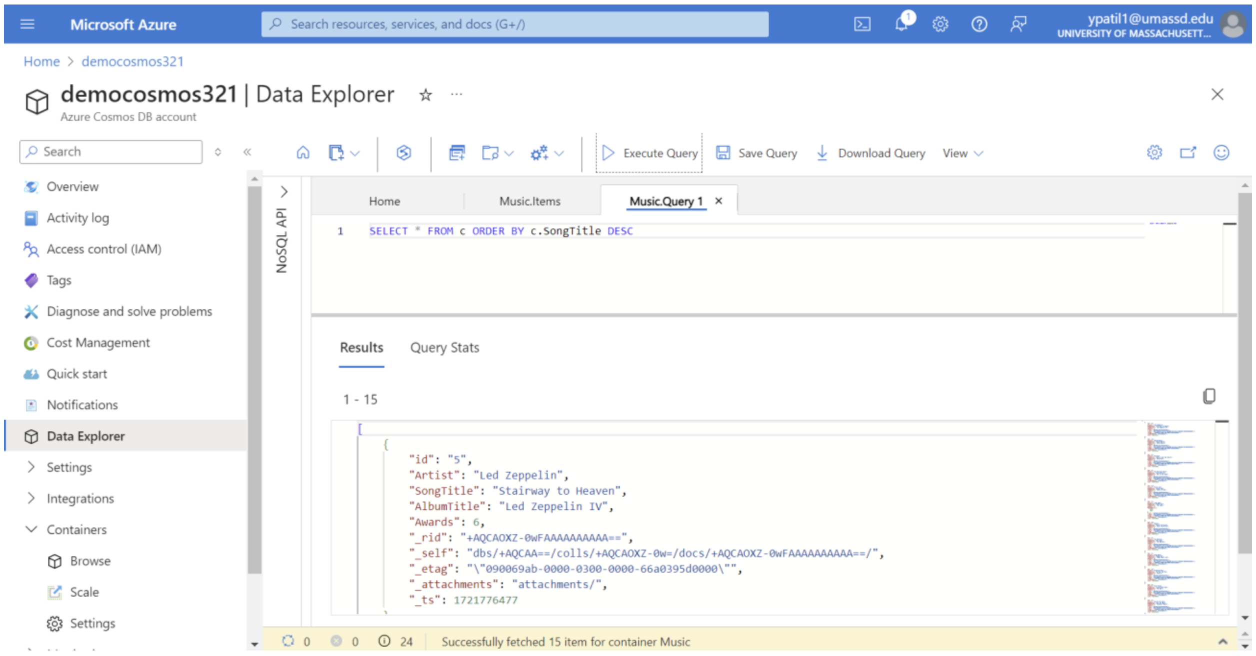


```
SELECT * FROM c ORDER BY c.SongTitle ASC
```

1 - 15

```
[
    {
        "id": "10",
        "Artist": "The Who",
        "SongTitle": "Baba O'Riley",
        "AlbumTitle": "Who's Next",
        "Awards": 3,
        "_rid": "+AQCAOXZ-0wKAAAAAAAAAA==",
        "_self": "dbs/+AQCAA==/colls/+AQCAOXZ-0w=/docs/+AQCAOXZ-0wKAAAAAAAAAA==/",
        "_etag": "\"09008fab-0000-0300-0000-66a0397c0000\"",
        "_attachments": "attachments/",
        "_ts": 1721776508
```

11. **Reverse Alphabetical Order:** To retrieve records for songs with titles in reverse alphabetical order or descending order.



```
SELECT * FROM c ORDER BY c.SongTitle DESC
```

```
[
    {
        "id": "5",
        "Artist": "Led Zeppelin",
        "SongTitle": "Stairway to Heaven",
        "AlbumTitle": "Led Zeppelin IV",
        "Awards": 6,
        "_rid": "+AQCAOXZ-0wFAAAAAAAAAA==",
        "_self": "dbs/+AQCAA==/colls/+AQCAOXZ-0w=/docs/+AQCAOXZ-0wFAAAAAAAAAA==/",
        "_etag": "\"090069ab-0000-0300-0000-66a0395d0000\"",
        "_attachments": "attachments/",
        "_ts": 1721776477
```

0 0 24 Successfully fetched 15 item for container Music

CONCLUSION:

In this homework, I learned to set up and use Azure Cosmos DB for managing NoSQL data. The steps involved creating a Cosmos DB account, setting up a container, adding data, and executing various queries to manipulate and retrieve data in JSON format. I started by creating a new Cosmos DB account for NoSQL, then used Data Explorer to create a "Music" container with "Artist" as the partition key.

I added 15 items with details such as song title, artist name, album title, and awards. I executed queries to filter and sort data, such as retrieving details for the artist "Nirvana" and ordering items in descending order. I also extracted specific fields like name and ID and sorted records based on awards and song titles in alphabetical and reverse alphabetical order.

This exercise improved my understanding of data management in Azure Cosmos DB and provided practical experience in executing queries to handle NoSQL data. Filtering data based on various conditions and sorting it in both ascending and descending order helped me understand the importance of proper data organization and retrieval techniques in a cloud environment.