

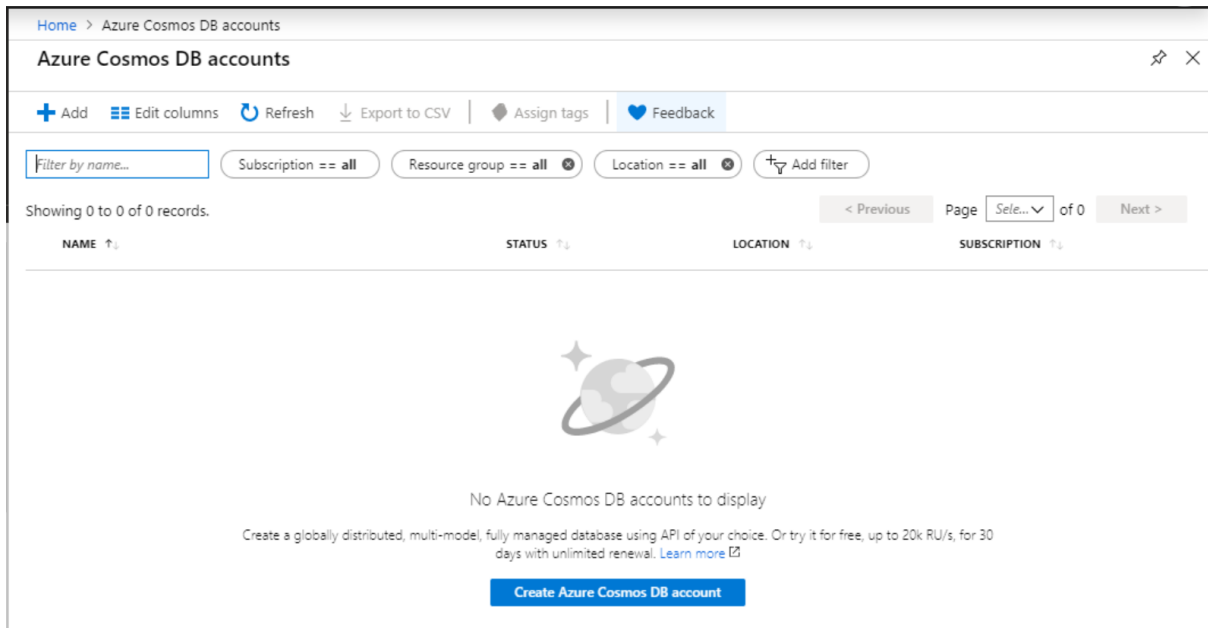
Create an Azure Cosmos DB using SQL API

In this tutorial I'm going to show how to:

- Create an Azure Cosmos DB/SQL API
- Add data to the Azure Cosmos DB
- Query Azure Cosmos DB

Step 01: Create an Azure Cosmos DB/SQL API

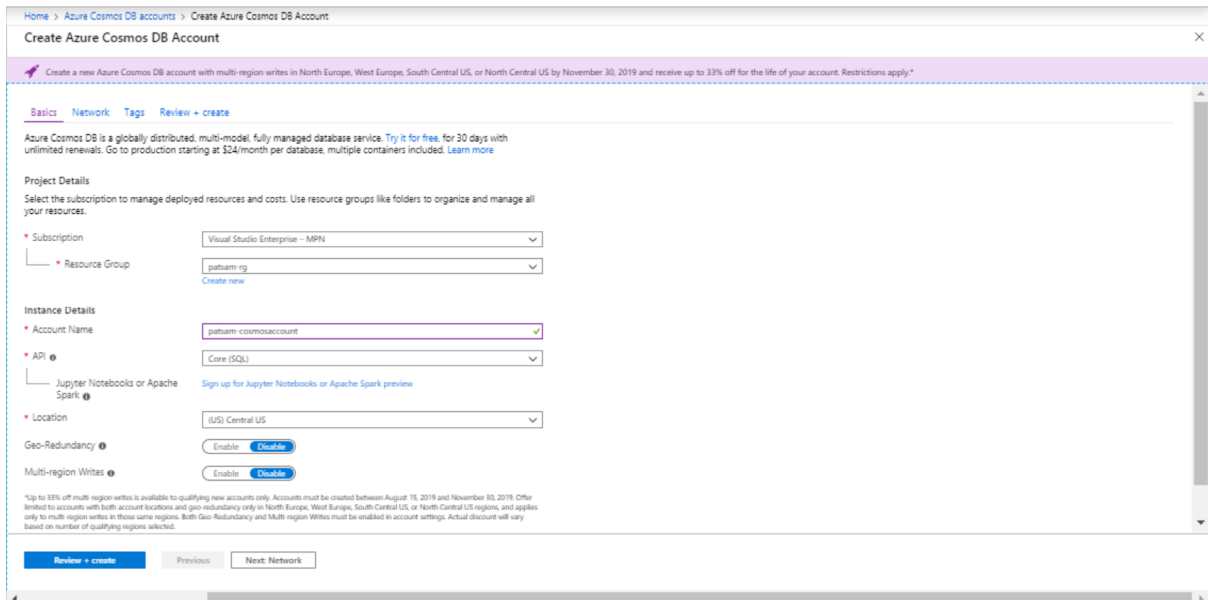
Click + Add to add new Cosmos DB Account.



The screenshot shows the 'Azure Cosmos DB accounts' page in the Azure portal. At the top, there's a breadcrumb 'Home > Azure Cosmos DB accounts'. Below the title, there are action buttons: '+ Add', 'Edit columns', 'Refresh', 'Export to CSV', 'Assign tags', and 'Feedback'. A filter bar shows 'Filter by name...' and three active filters: 'Subscription == all', 'Resource group == all', and 'Location == all'. Below the filters, it says 'Showing 0 to 0 of 0 records.' and a table with columns 'NAME', 'STATUS', 'LOCATION', and 'SUBSCRIPTION'. The table is empty, and a large graphic with the text 'No Azure Cosmos DB accounts to display' is shown. Below this, a message states: 'Create a globally distributed, multi-model, fully managed database using API of your choice. Or try it for free, up to 20k RU/s, for 30 days with unlimited renewal. [Learn more](#)'. At the bottom, there is a blue button labeled 'Create Azure Cosmos DB account'.

Provide relevant details and click Review + Create.

- Resource group: patsam-rg
- Account name: patsam-cosmosaccount
- API: Core (SQL)
- Location: Central US

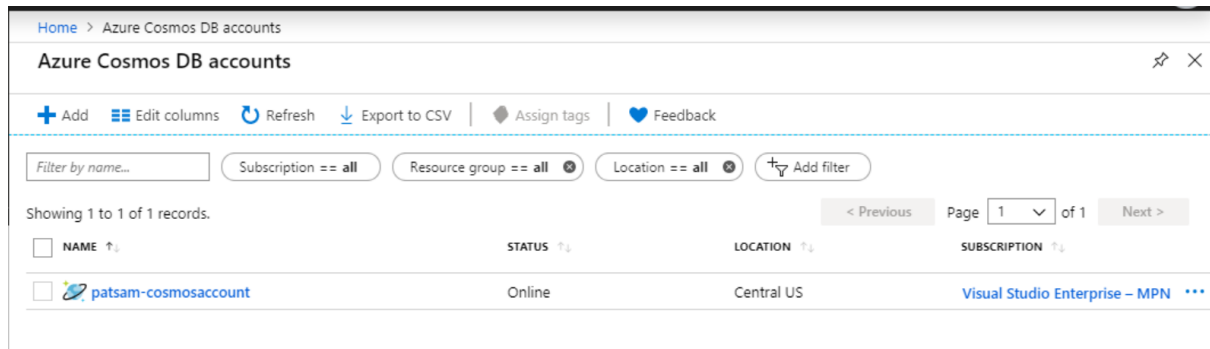


The screenshot shows the 'Create Azure Cosmos DB Account' wizard in the Azure portal. The 'Basics' tab is selected, showing the following details:

- Project Details:** Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.
- Subscription:** Visual Studio Enterprise - MPN
- Resource Group:** patsam-rg (with a 'Create new' link)
- Instance Details:**
 - Account Name:** patsam-cosmosaccount (with a checkmark)
 - API:** Core (SQL)
 - Location:** (US) Central US
 - Geo-Redundancy:** Enable (with a 'Disable' link)
 - Multi-region Writes:** Enable (with a 'Disable' link)

At the bottom, there are buttons for 'Review + create', 'Previous', and 'Next: Network'. A small footnote at the bottom left states: '*Up to 10% off multi-region writes is available to qualifying new accounts only. Accounts must be created between August 15, 2019 and November 30, 2019. Offer limited to accounts with both account locations and geo-redundancy only in North Europe, West Europe, South Central US, or North Central US regions, and applies only to multi-region writes in those same regions. Both Geo-Redundancy and Multi-region Writes must be enabled in account settings. Actual discount will vary based on number of qualifying regions selected.'

Once the deployment finishes, we can see the newly created Cosmos DB Account.



Home > Azure Cosmos DB accounts

Azure Cosmos DB accounts

+ Add Edit columns Refresh Export to CSV Assign tags Feedback

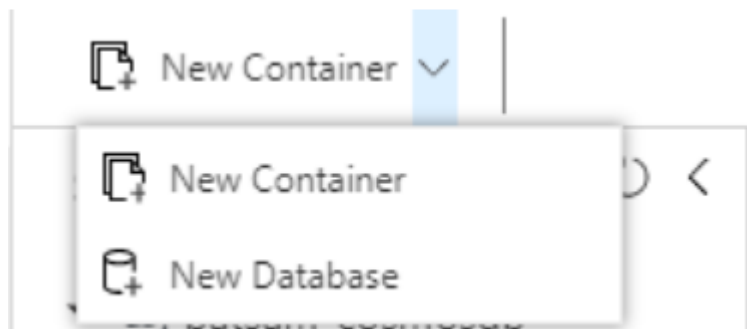
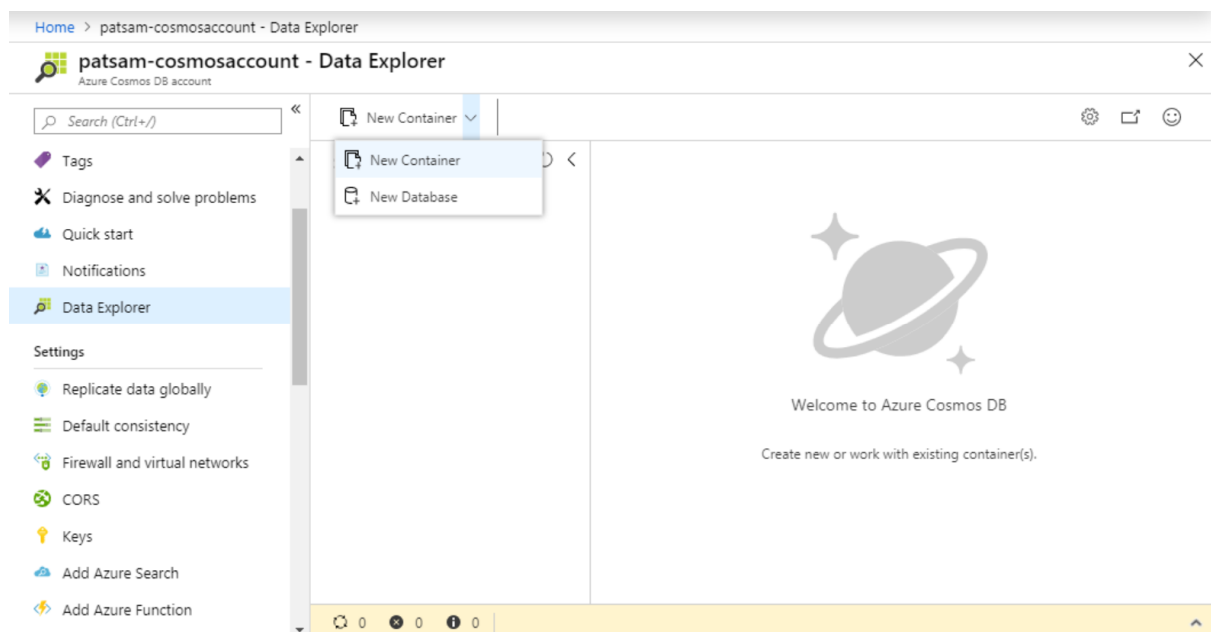
Filter by name... Subscription == all Resource group == all Location == all Add filter

Showing 1 to 1 of 1 records. < Previous Page 1 of 1 Next >

NAME	STATUS	LOCATION	SUBSCRIPTION
patsam-cosmosaccount	Online	Central US	Visual Studio Enterprise – MPN

Next, we need to add a Database. Click add New Database.


Data Explorer → New Database



Provide relevant details and Click OK.

- Database id: patsam-cosmosdb

New Database

 Start at \$24/mo per database, multiple containers included
[More details](#)

* Database id ⓘ

☒ Provision throughput ⓘ

* Throughput (400 - 100,000 RU/s) ⓘ

+

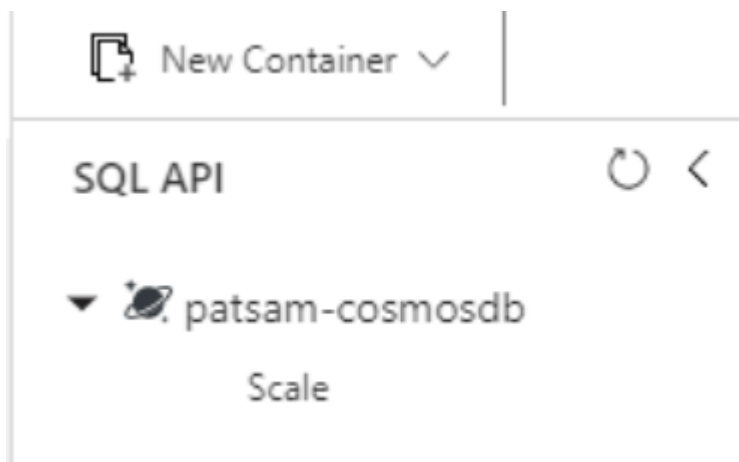
-

Estimated spend (USD): **\$0.032 hourly / \$0.77 daily** (1 region,
1000RU/s \$0.00008/RU/s)

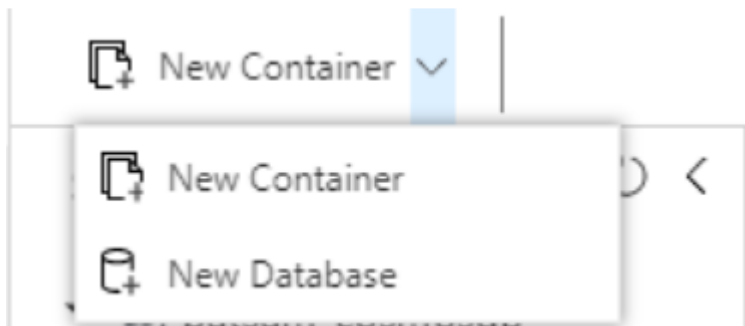
OK

Patrick Sameera Jayalath

We can see the newly created Database.

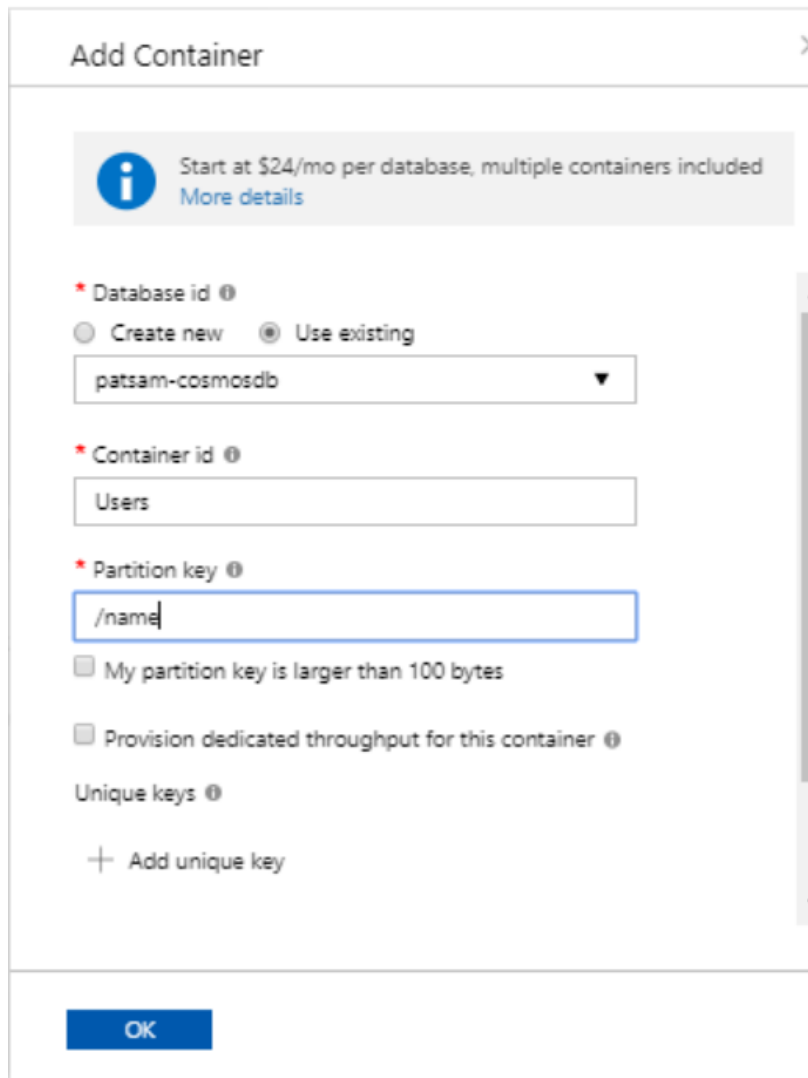


Next, we need to add a Container. Click on New Container.



Provide relevant data and Click OK.

- Database id: patsam-cosmosdb (select the Database we created earlier)
- Container id: Users
- Partition key: name

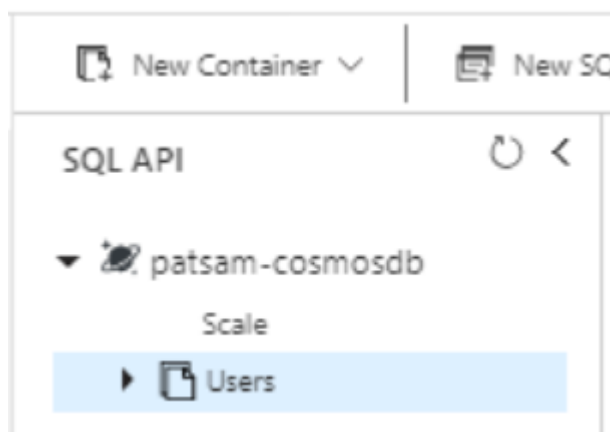


The screenshot shows the 'Add Container' dialog box. At the top, there is a close button (X) and a title bar. Below the title bar, there is an information banner that says 'Start at \$24/mo per database, multiple containers included' with a link to 'More details'. The main form area contains the following fields and options:

- * Database id**: A dropdown menu with 'patsam-cosmosdb' selected. Above the dropdown are radio buttons for 'Create new' and 'Use existing' (which is selected).
- * Container id**: A text input field containing 'Users'.
- * Partition key**: A text input field containing '/name'.
- Below the partition key field are two checkboxes:
 - ☐ My partition key is larger than 100 bytes
 - ☐ Provision dedicated throughput for this container
- Unique keys**: A section with a plus icon and the text 'Add unique key'.

At the bottom of the dialog is a blue 'OK' button.

We can see the newly created Container.



Same way add another Container called Address.

Add Container

Start at \$24/mo per database, multiple containers included
[More details](#)

* Database id ⓘ

☐ Create new

☒ Use existing

patsam-cosmosdb

▼

* Container id ⓘ

Address

* Partition key ⓘ

/address

☐ My partition key is larger than 100 bytes

☐ Provision dedicated throughput for this container ⓘ

Unique keys ⓘ

+ Add unique key

OK

New Container ▼

New SQL Query

SQL API

↺ <

▼ patsam-cosmosdb

Scale

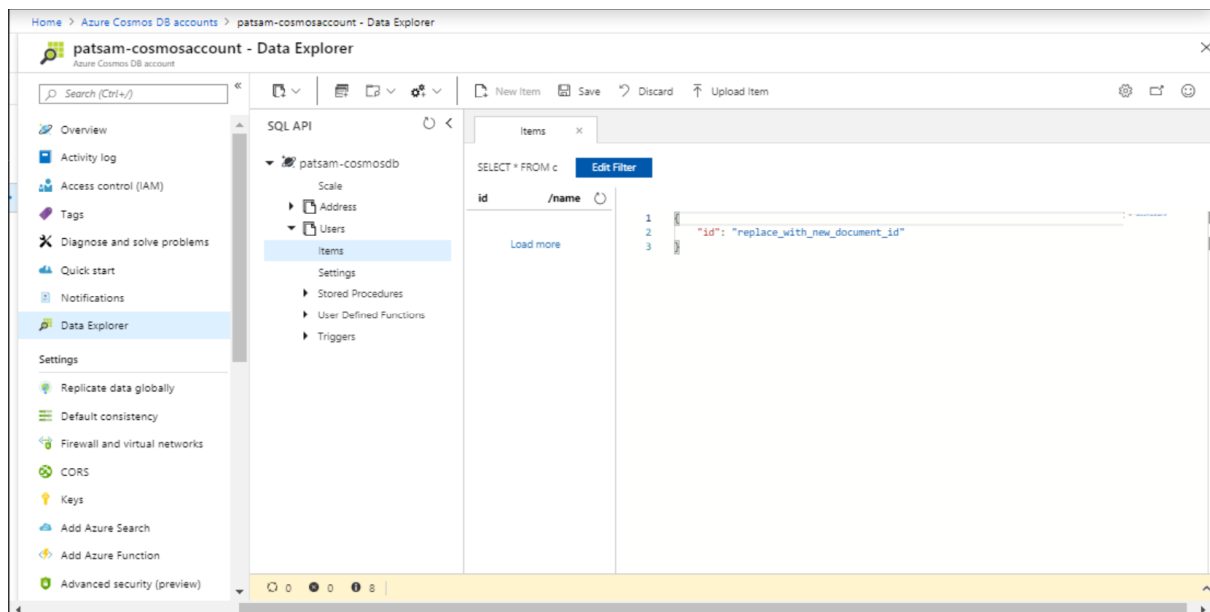
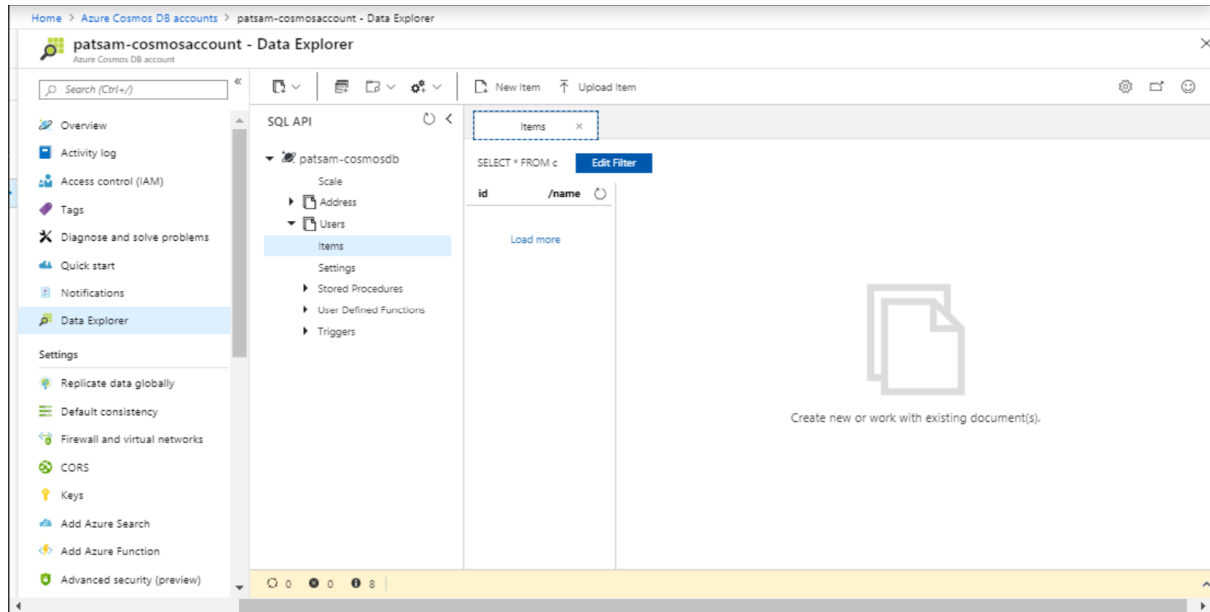
▶ Address

▶ Users

Step 02: Add data to the Azure Cosmos DB

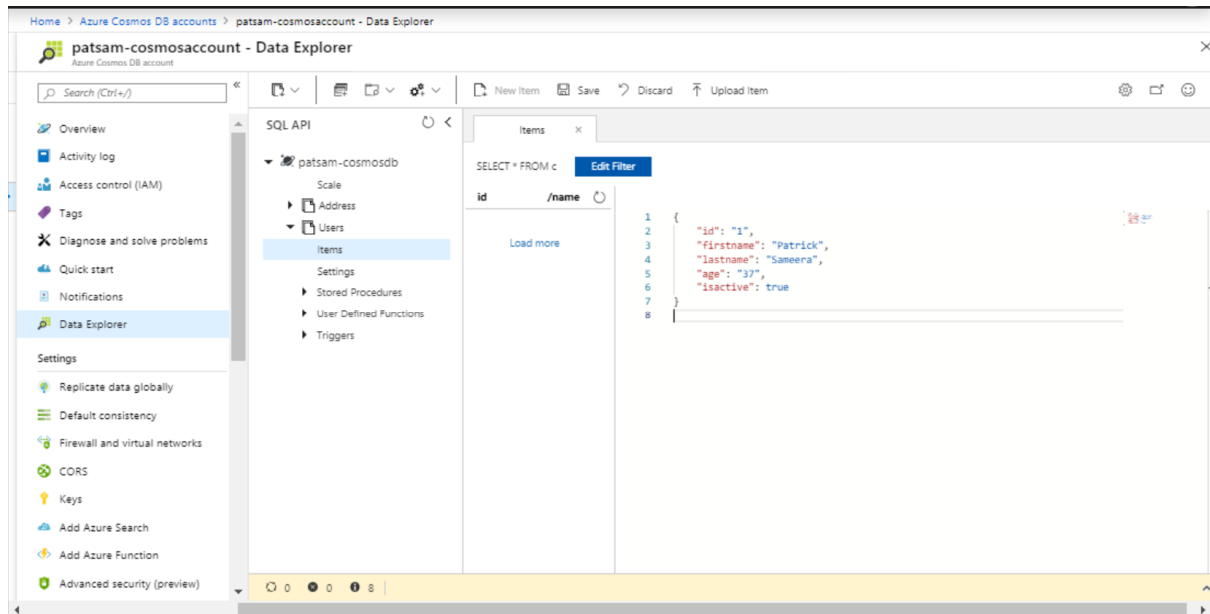
In Data Explorer, expand the patsam-cosmosdb Database, and expand the Users Container.

Next, select Items, and then select New Item.

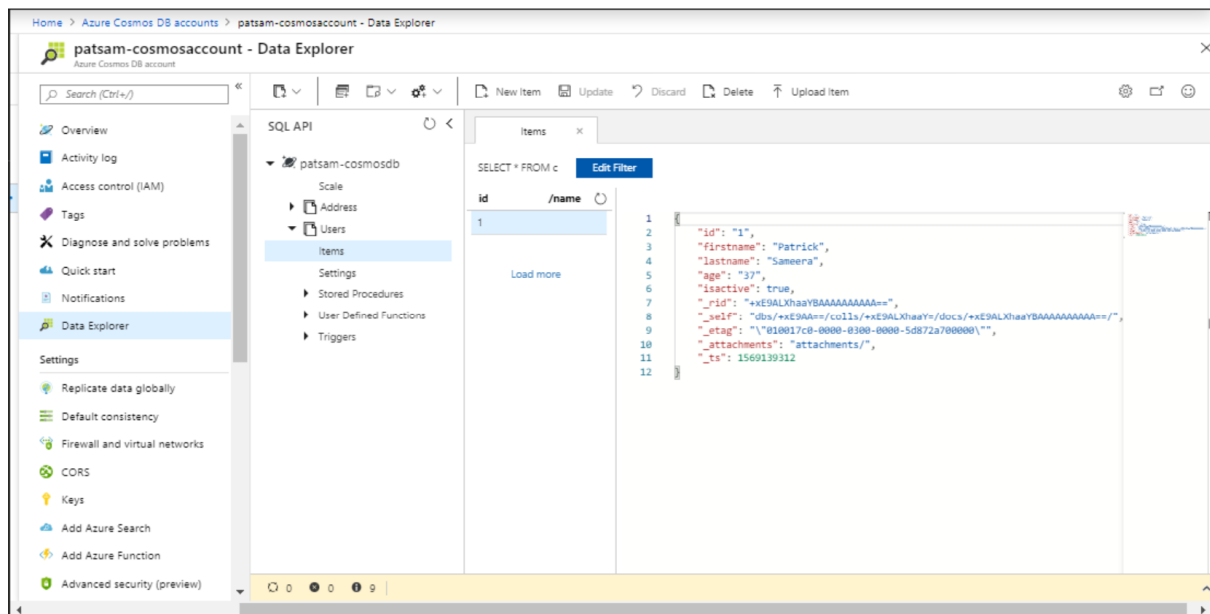


Add following structure and Click Save.

```
{  
  "id": "1",  
  "firstname": "Patrick",  
  "lastname": "Sameera",  
  "age": "37",  
  "isactive": true  
}
```

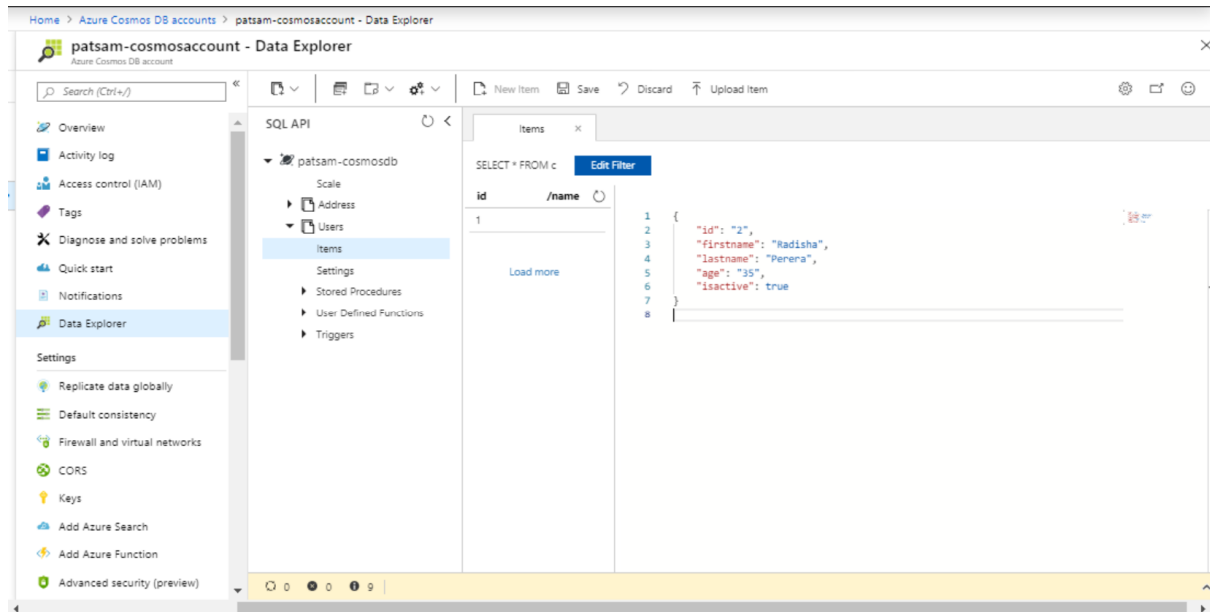


Notice when Click Save it adds some additional JSON properties.

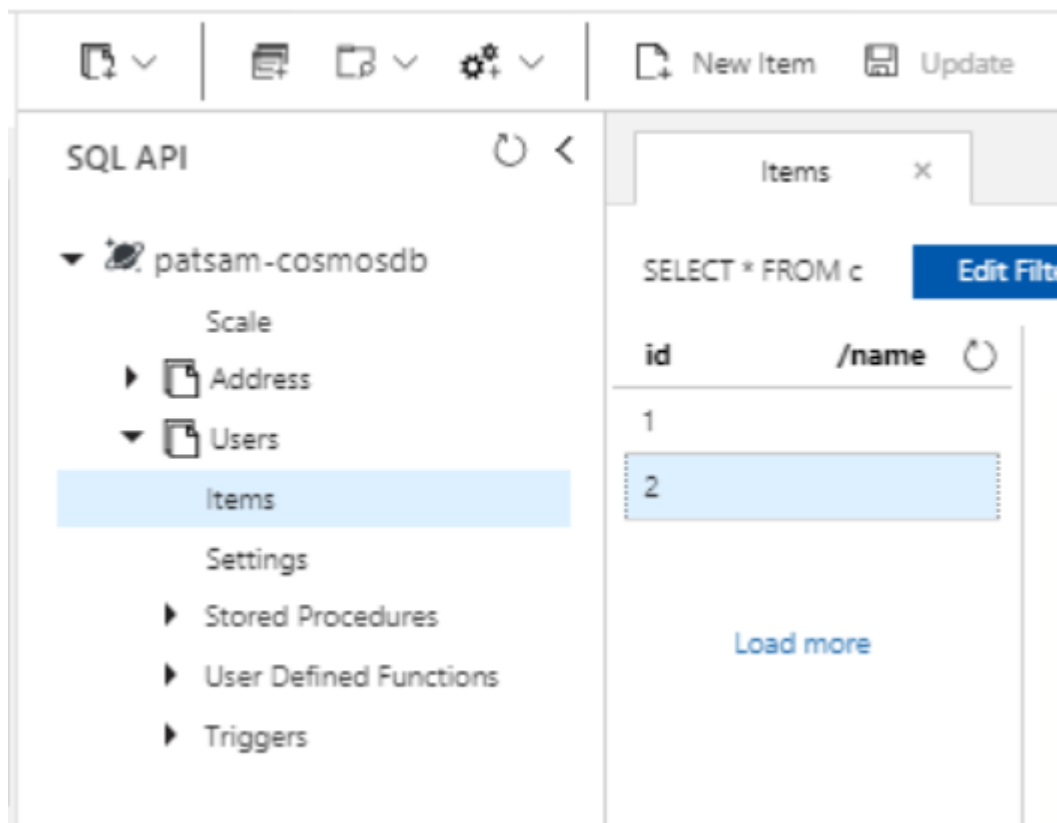


Same way adds another New Item with a unique id and Click Save.

```
{  
  "id": "2",  
  "firstname": "Radisha",  
  "lastname": "Perera",  
  "age": "35",  
  "isactive": true  
}
```



We can see the 2 items we added.

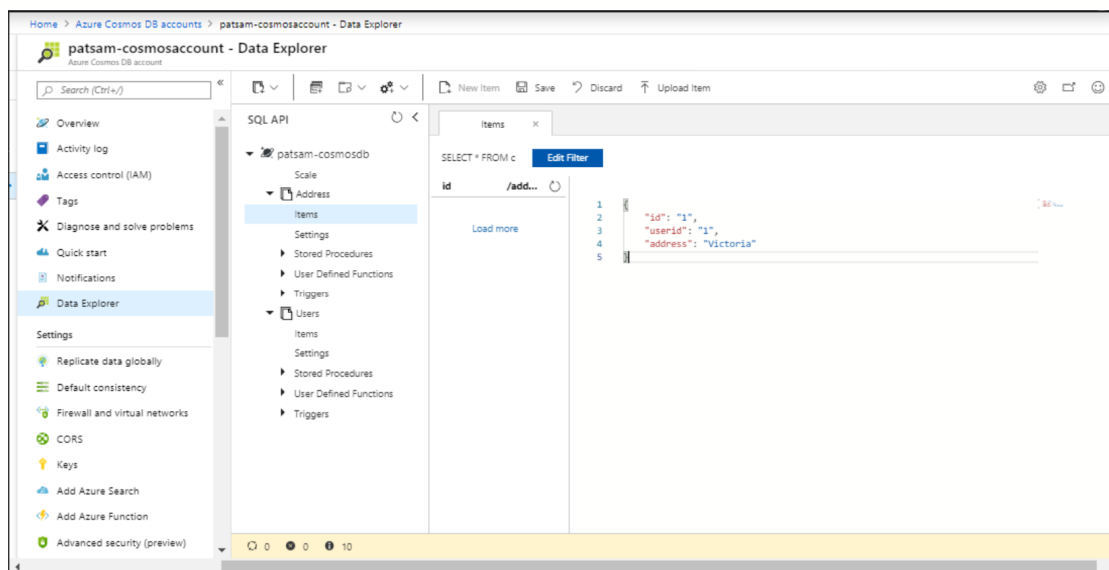


Same way add address details.

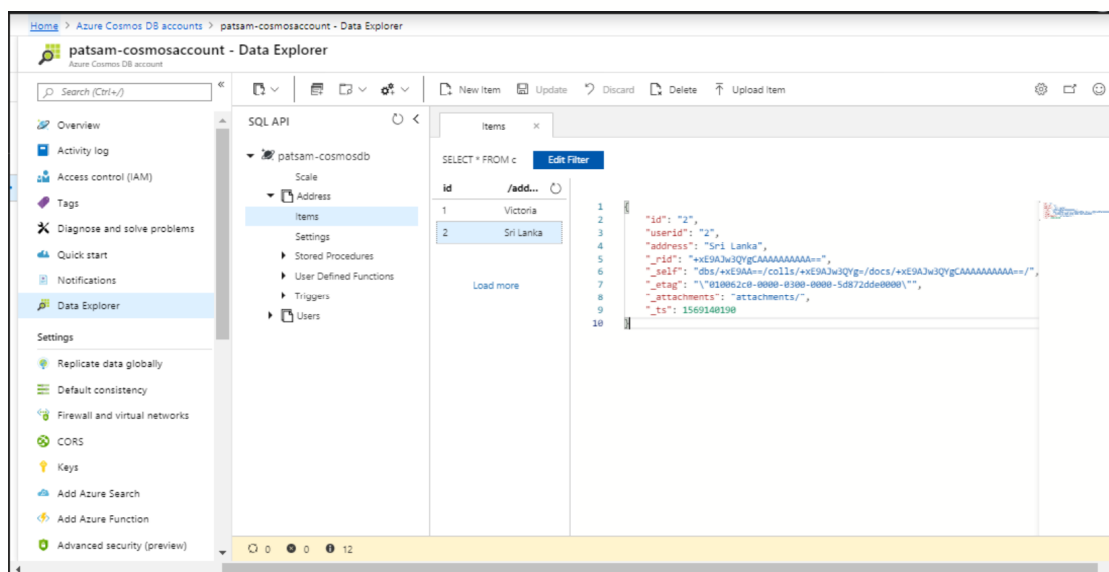
Expand the Address Container.

Next, select Items, and then select New Item.

```
{  
  "id": "1",  
  "userid": "1",  
  "address": "Victoria"  
}
```

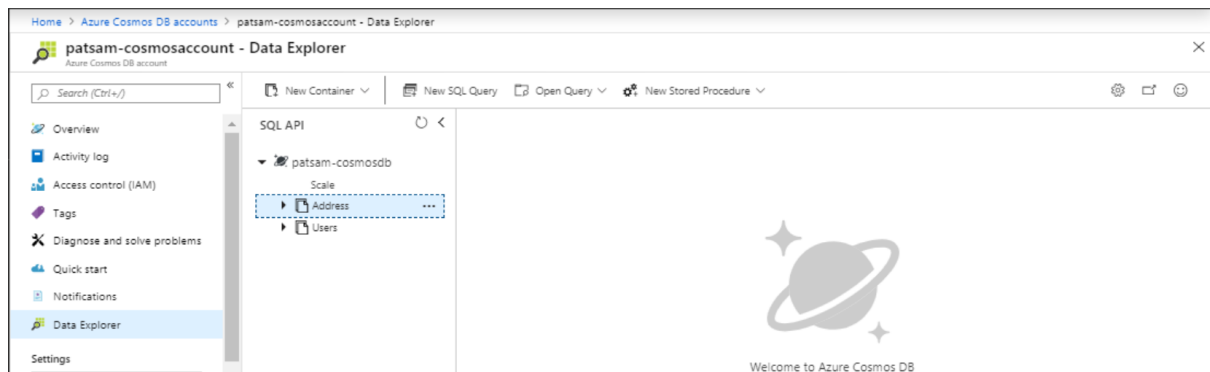


```
{  
  "id": "2",  
  "userid": "2",  
  "address": "Sri Lanka"  
}
```



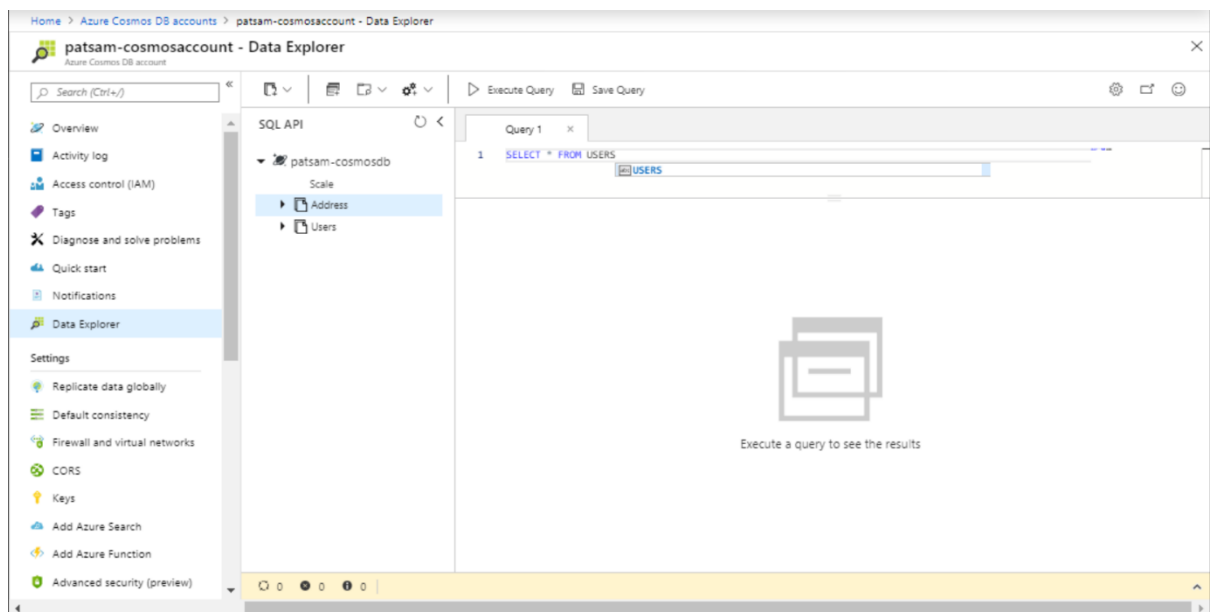
Step 03: Query Azure Cosmos DB

Click on one of the Containers and you will get to see New SQL Query option.



Click on New SQL Query option.

Now we can run queries against the selected Container.



Run bellow query to view data in the Address Container. Click Execute Query.

```
SELECT * FROM c
```

The screenshot shows the SQL API interface with the 'Address' container selected in the left sidebar. The query editor displays the query `SELECT * FROM c`. The results pane shows two documents, labeled '1 - 2'.

```
{
  "id": "1",
  "userid": "1",
  "address": "Victoria",
  "_rid": "+xE9AJw3QYgBAAAAAAAAA==",
  "_self": "dbs/+xE9AA==/colls/+xE9AJw3QYg=/docs/+xE9AJw3QYgBAAAAAAAAA==/",
  "_etag": "\"01005cc0-0000-0300-0000-Sd872d960000\"",
  "_attachments": "attachments/",
  "_ts": 1569140118
},
{
  "id": "2",
  "userid": "2",
  "address": "Sri Lanka",
  "_rid": "+xE9AJw3QYgCAAAAAAAAAA==",
  "_self": "dbs/+xE9AA==/colls/+xE9AJw3QYg=/docs/+xE9AJw3QYgCAAAAAAAAAA==/",
  "_etag": "\"010062c0-0000-0300-0000-Sd872d960000\"",
  "_attachments": "attachments/",
  "_ts": 1569140190
}
```

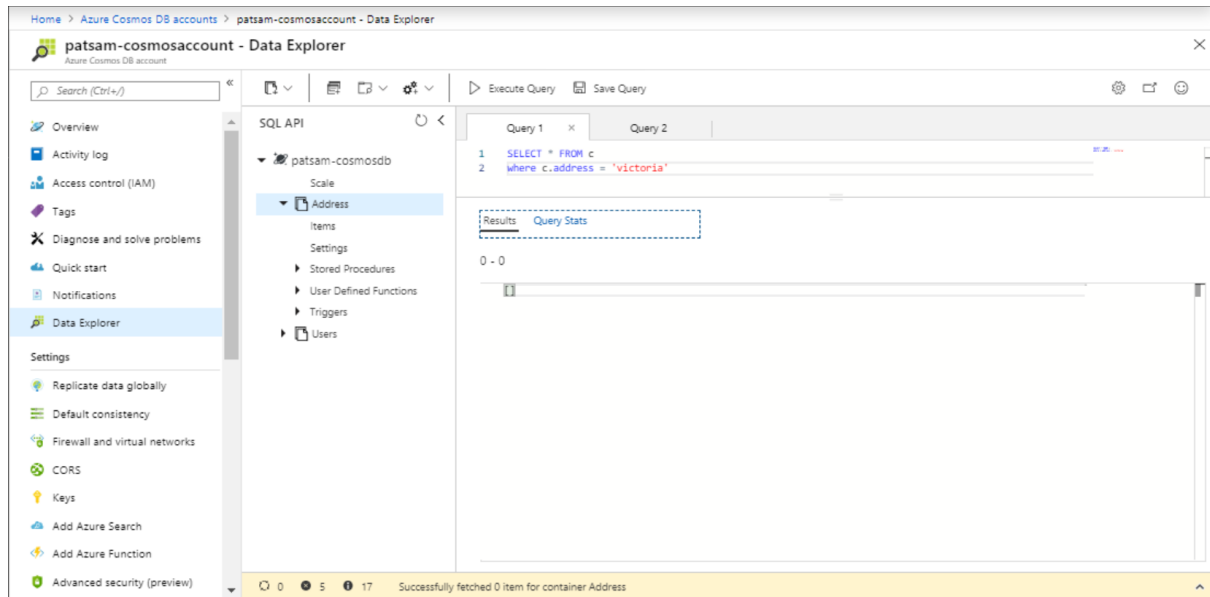
```
SELECT * FROM c
where c.id = '1'
```

The screenshot shows the SQL API interface with the 'Address' container selected in the left sidebar. The query editor displays the query `SELECT * FROM c where c.id = '1'`. The results pane shows one document, labeled '1 - 1'.

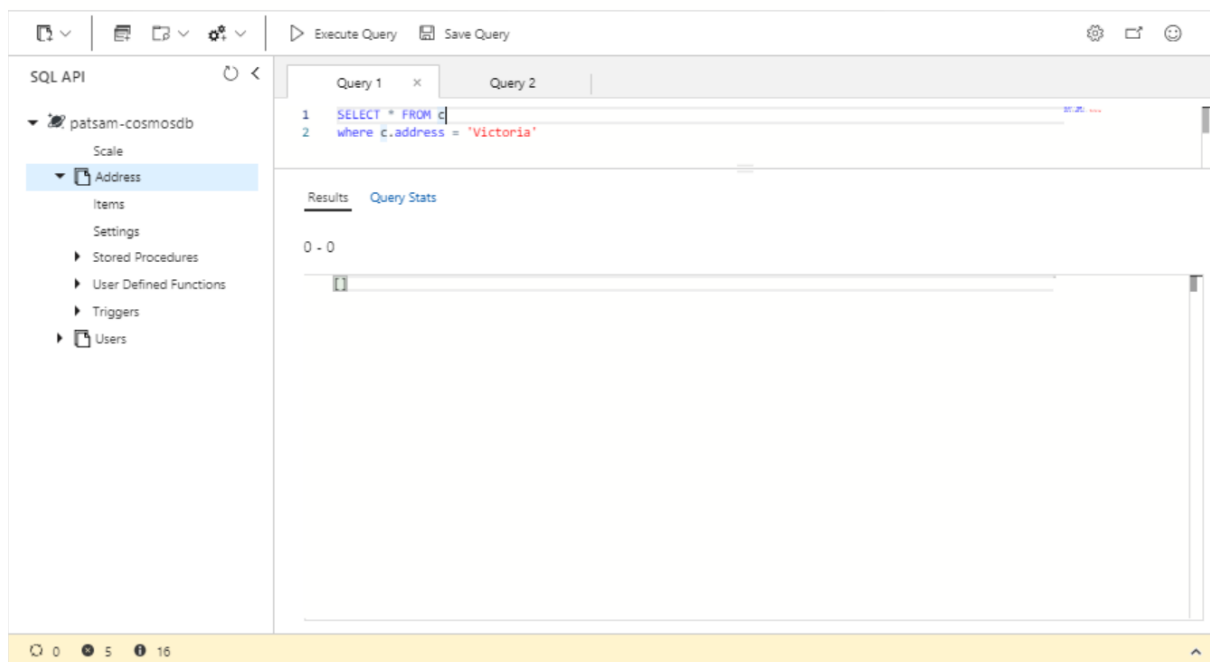
```
{
  "id": "1",
  "userid": "1",
  "address": "Victoria",
  "_rid": "+xE9AJw3QYgBAAAAAAAAA==",
  "_self": "dbs/+xE9AA==/colls/+xE9AJw3QYg=/docs/+xE9AJw3QYgBAAAAAAAAA==/",
  "_etag": "\"01005cc0-0000-0300-0000-Sd872d960000\"",
  "_attachments": "attachments/",
  "_ts": 1569140118
}
```

```
SELECT * FROM c
where c.address = 'victoria'
```

it doesn't return any data even though we have "Victoria" as an address. That is because it's case sensitive.



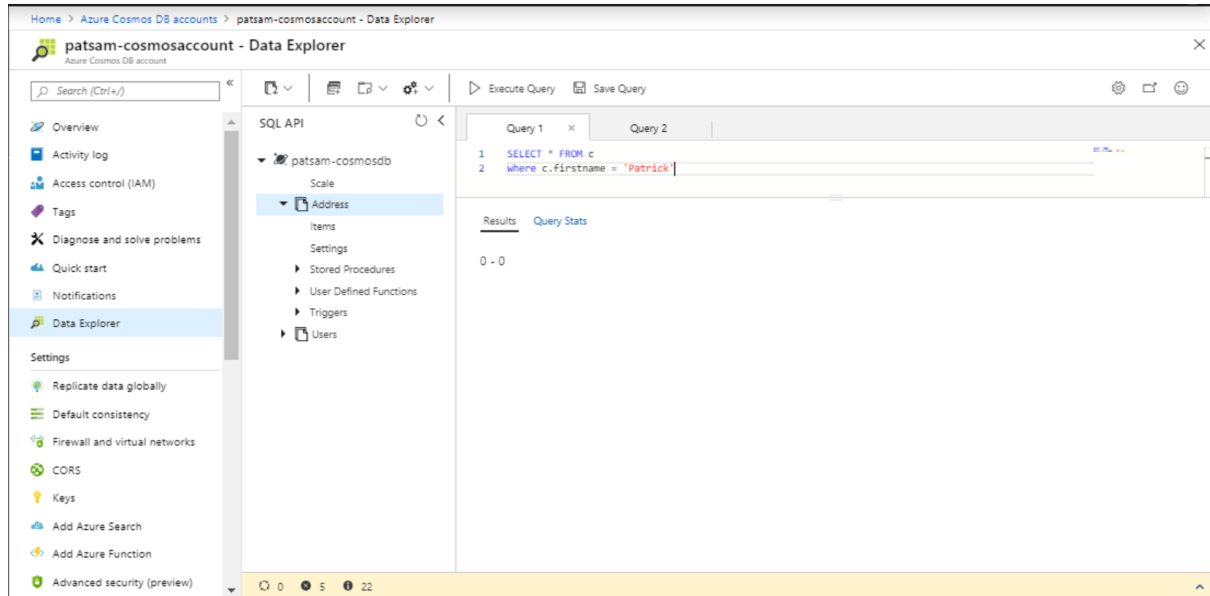
```
SELECT * FROM c
where c.address = 'Victoria'
```



Run below query while Address Container is selected.

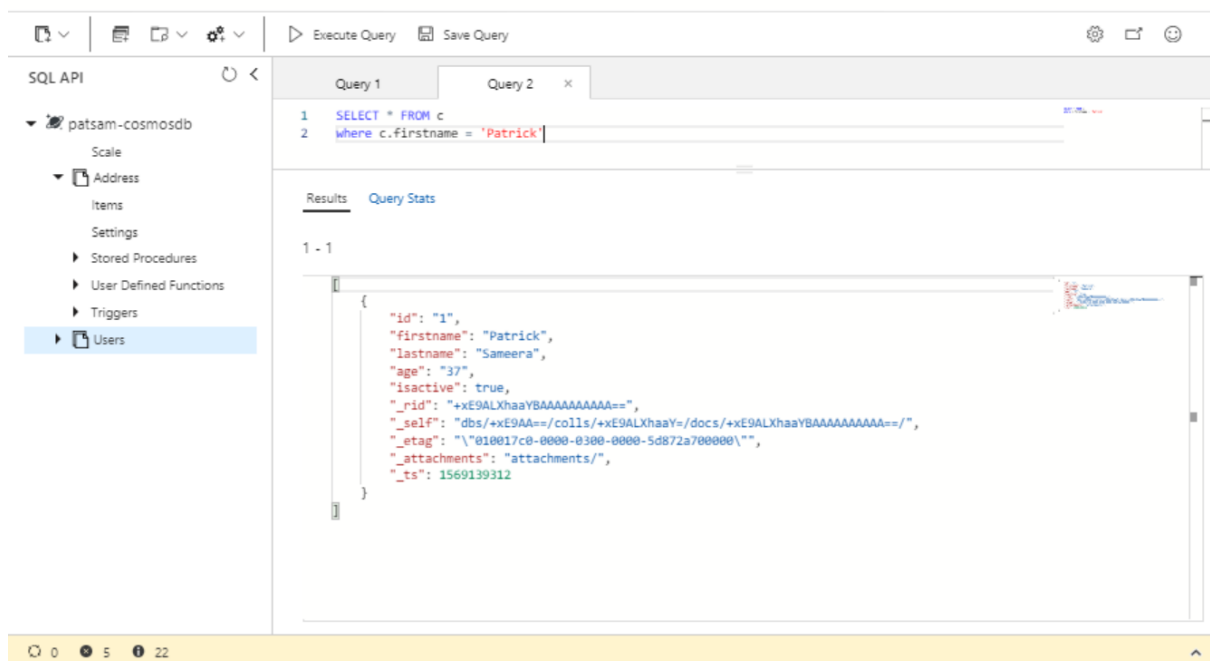
```
SELECT * FROM c
where c.firstname = 'Patrick'
```

It won't return any data because we are running the query against the Address Container.



So, if want to run that query, we need to select Users Container.

```
SELECT * FROM c
where c.firstname = 'Patrick'
```



If you want to learn about more query techniques on DocumentDB SQL, have a look below URL:

https://www.tutorialspoint.com/documentdb_sql/index.htm