

Working with Azure Cosmos DB - SQL (Core) API



Reza Salehi

CLOUD CONSULTANT

@zaalion [linkedin.com/in/rezasalehi2008](https://www.linkedin.com/in/rezasalehi2008)



Overview



Containers or collections?

Use Data Explorer to manage your data from the portal

Work with your data using SQL syntax

Azure Cosmos DB .NET SDK

- Version 3.x targeting .NET Standard

Demo: Working with the SQL API

Summary



Cosmos DB Containers

An Azure Cosmos container is the unit of scalability for throughput and storage

The container items and the throughput are distributed across a set of logical partitions

Logical partitions are created based on partition keys

An Azure Cosmos container can scale elastically



Containers in Each API

Azure Cosmos API

Specialized Entity

SQL API

Collection

Table API

Table

MongoDB API

Collection

Cassandra API

Table

Gremlin API

Graph



Data Explorer

The screenshot displays the Azure Data Explorer interface for an account named 'ps-demo'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, Notifications, Data Explorer (selected), and Settings. The Settings section includes options for Replicate data globally, Default consistency, Firewall and virtual networks, CORS, Keys, and Add Azure Search.

The main area shows the 'SQL API' view for a database named 'db01'. The 'users' collection is selected, and the 'Items' tab is active. A SQL query 'SELECT * FROM c' is entered in the query editor, with an 'Edit Filter' button. The results are displayed in a table with two columns: 'id' and '/email'.

id	/email
34343434	Reza@test.com
1243565	john@test.com

A 'Load more' button is visible below the table. To the right of the table, a JSON document is shown, representing the structure of the data items.

```
1 {
2   "id": "34343434",
3   "name": "Reza",
4   "email": "Reza@test.com",
5   "_rid": "6IYHAIrNXEBAAAAAAAAA==",
6   "_self": "dbs/6IYHAA==/colls/6IYHAIrNXE=/docs/6IYHAIrNXE",
7   "_etag": "\"0300da70-0000-0100-0000-5d50ad070000\"",
8   "_attachments": "attachments/",
9   "_ts": 1565568263
10 }
```



Azure Cosmos DB SQL API
supports querying
JSON items using SQL.



SQL and JSON

```
{  
  "givenname": "John",  
  "lastname": "Smith",  
  "address": {  
    "streetAddress": "1 sample street",  
    "city": "Boston",  
    "country": "USA"  
  },  
  "email": "john@test.com",  
  "phone": "+1 123 4567891"  
}
```

```
SELECT *  
FROM People p  
WHERE p.givenname = "John"
```

```
SELECT *  
FROM People p  
WHERE p.address.city = "Boston"
```

```
SELECT {"Name":p.id, "City":p.address.city} AS Person  
FROM People p  
WHERE p.address.city = "Boston"
```



```
using Microsoft.Azure.Cosmos;
```

Azure Cosmos DB Client Library 3.x for .NET



Demo



Using Data Explorer to add data

- Partition keys & unique constraints

Using Cosmos DB data migration tool

Query data using the SQL syntax



Demo



Working with the Cosmos DB client library 3.x for .NET



Summary



Cosmos DB containers

- Containers or collections?

Data Explorer enables you to manage your data from the portal

Use familiar SQL syntax to interact with your Cosmos DB data

Azure Cosmos DB .NET SDK 3.x

Demo: Working with the SQL API

