😊 Azure Table Storage

Azure Table Storage is a service provided by Microsoft Azure that offers highly scalable, NoSQL cloud database storage. It is designed to store large amounts of structured, non-relational data. This service is ideal for storing structured, schemaless data, such as user data for web applications, address books, device information, or any other information that needs to be quickly accessed and updated.

Key Features of Azure Table Storage:

1. **Schemaless Design**: Unlike traditional relational databases, Azure Table Storage does not require a fixed schema, allowing each entity (row) to have its own set of properties (columns).
2. **Scalability**: It is designed to scale out and handle large volumes of data, making it suitable for big data scenarios.
3. **High Availability**: Azure Table Storage ensures high availability with data replicated across different geographical locations.
4. **Partitioning**: Data is automatically partitioned to optimize performance and scalability.
5. **Cost-Effective**: It offers a cost-effective storage solution, with users only paying for the storage they use.

Structure:

- **Table**: The top-level container for data, which holds a collection of entities.
- **Entity**: A set of properties, similar to a row in a relational database table. Each entity has a unique identifier made up of the PartitionKey and RowKey.
- **Property**: A name-value pair, similar to a column in a relational database.

Common Use Cases:

1. **Web Applications**: Store user data, session information, and other structured data.
2. **IoT**: Capture and store telemetry data from IoT devices.
3. **Audit Logs**: Store logs and audit data for applications.
4. **Product Catalogs**: Manage catalogs for e-commerce sites.

How It Works:

1. **Creating a Table**: Users create a table within an Azure Storage account.
2. **Adding Entities**: Entities with various properties are added to the table.
3. **Querying Data**: Data can be queried using the PartitionKey and RowKey for fast lookups.
4. **Updating and Deleting**: Entities can be updated or deleted as needed.

Benefits:

1. **Flexible Data Model**: The schemaless design allows for flexibility in data modeling.
2. **Performance**: Optimized for fast read and write operations.
3. **Global Distribution**: Ensures high availability and disaster recovery.
4. **Integration**: Easily integrates with other Azure services and tools.

Example Scenario:

Imagine you are building a web application to manage a customer database for an e-commerce site. You can use Azure Table Storage to store customer profiles, where each profile is an entity with properties such as Name, Email, Address, and OrderHistory. You can quickly query, update, and scale the data as your user base grows.

By using Azure Table Storage, you can efficiently handle large volumes of structured data, ensure high availability, and scale your application as needed without the overhead of managing a traditional relational database.

🤔 What are we doing in this lab?

In this process, we are setting up and utilizing Azure Table Storage to store and manage structured, non-relational data. Here is a concise breakdown of the steps and the end goal:

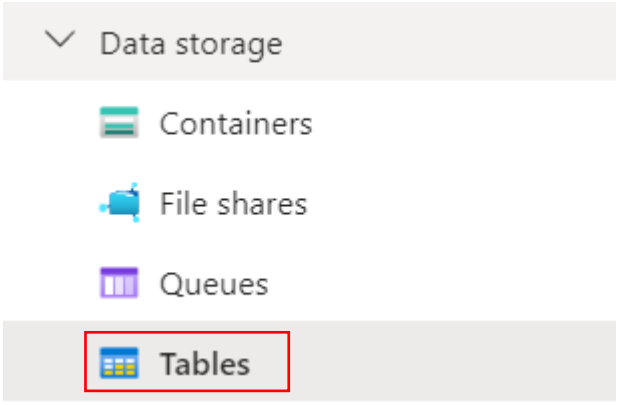
1. **Azure Portal Login**: We log in to the Azure Portal to manage our cloud resources.
2. **Creating a Storage Account**: We create a storage account, which serves as a container for our table storage.
3. **Navigating to Table Storage**: Within the storage account, we navigate to the Tables section under data storage.
4. **Creating a Table**: We create a new table by providing a name and confirming the creation.
5. **Accessing the Table**: Initially, the table can only be deleted or configured for access policies. To manage the table, we use Azure Storage Explorer or the storage browser in the portal.
6. **Adding Entities**: We add entities to the table by specifying the Partition Key and Row Key, along with other properties.
7. **Inserting Data**: We insert values into the table, demonstrating the flexibility of adding various properties with different data types.

End Goal

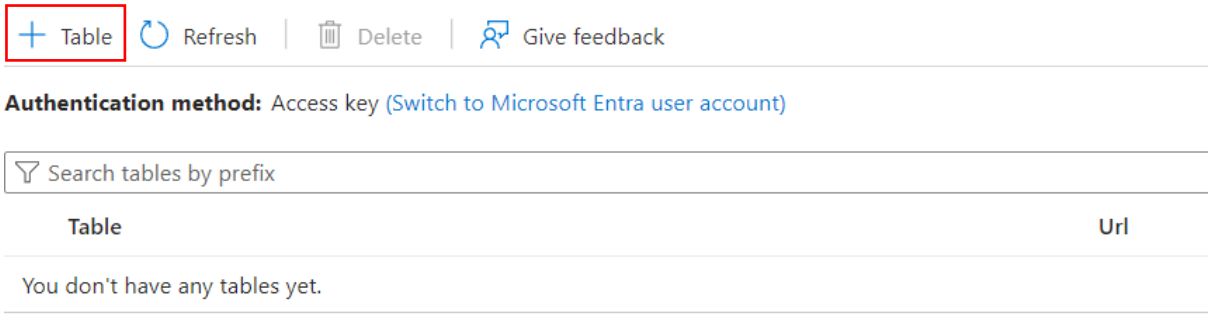
The ultimate goal is to set up a scalable, flexible, and cost-effective NoSQL database using Azure Table Storage. This allows for efficient storage and management of large volumes of structured data, suitable for applications such as user profiles, IoT data, audit logs, and more. By following these steps, we create a robust system for storing and querying structured data, enabling quick access and updates in a highly available and globally distributed environment.

😄 To begin with the Lab

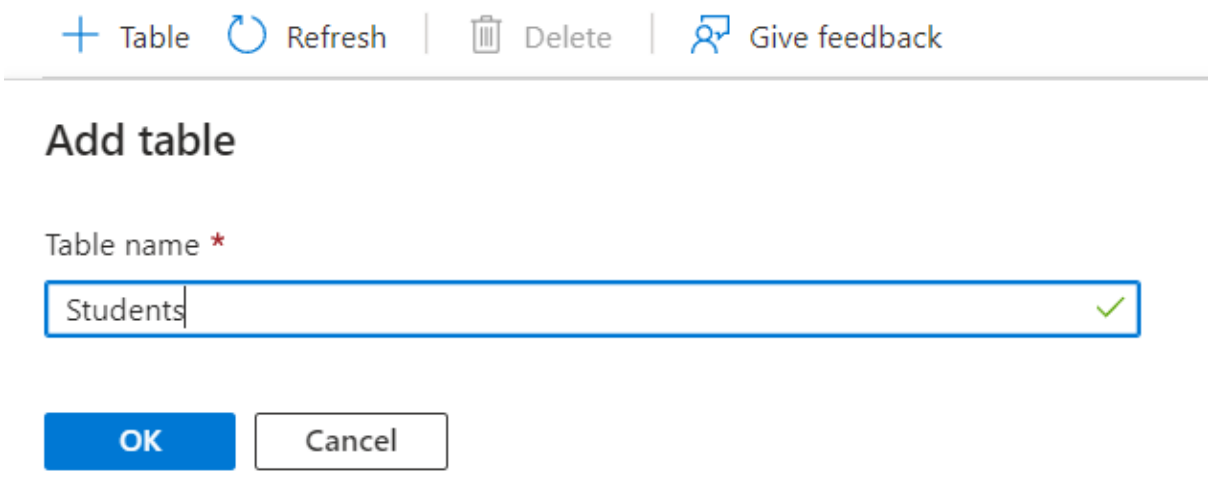
1. Now to work with Azure table storage first we need to go to Azure Portal and log in, then we need to create a storage account.
2. Once we have the storage account then we need to go into it. Then from the left pane expand data storage and go to Tables.



3. Now to create a table you need to click on add table which is highlighted below.

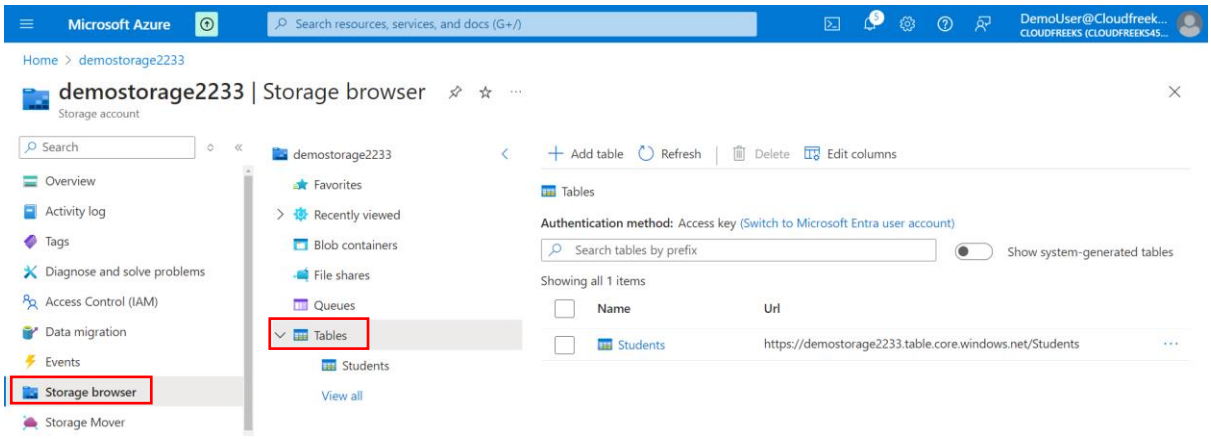


4. Now we need to give our table a name then just click on OK.

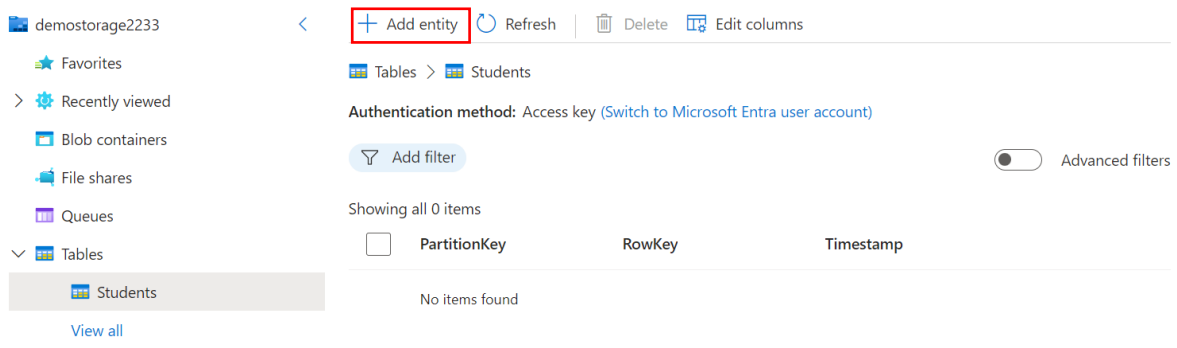


5. So, once your table has been created you will see that you cannot do anything on it. You just have the option to delete it or you can go to IAM or to the access policy.

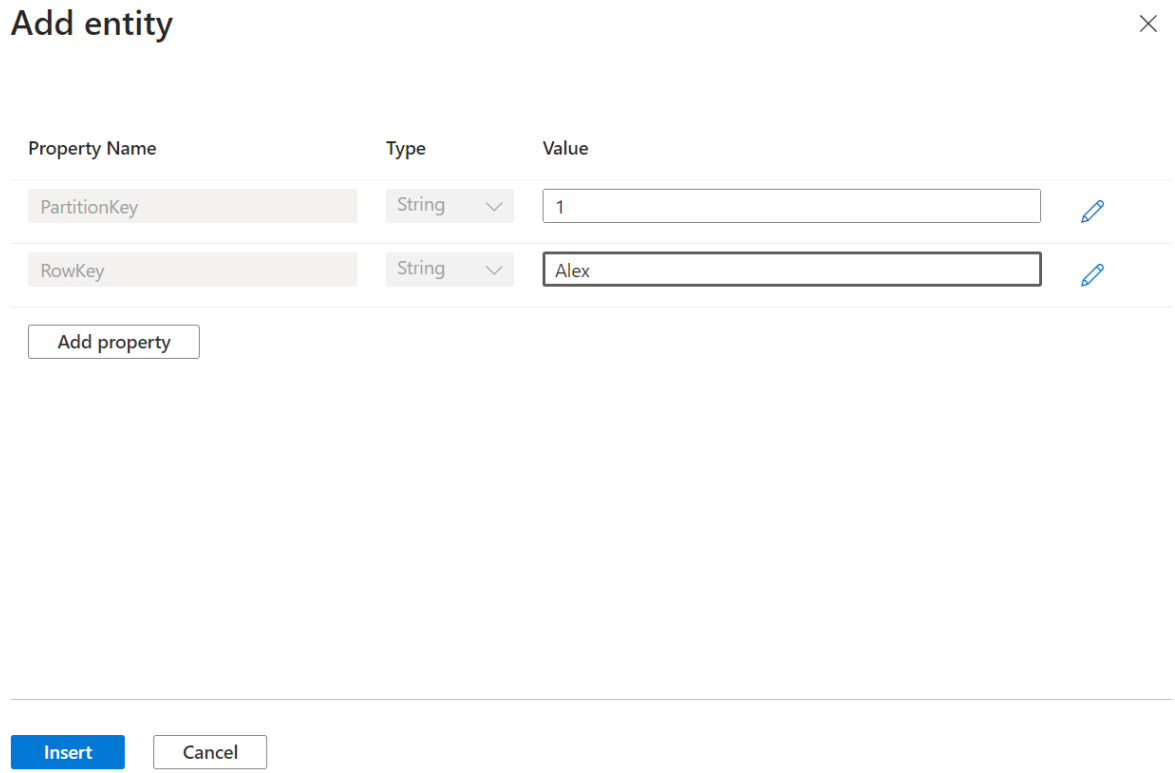
6. So, to access your table you can log in to Azure Storage Explorer or in the portal from the left pane go to the storage browser. Then go to tables here you will see your table.
7. Now click on it.



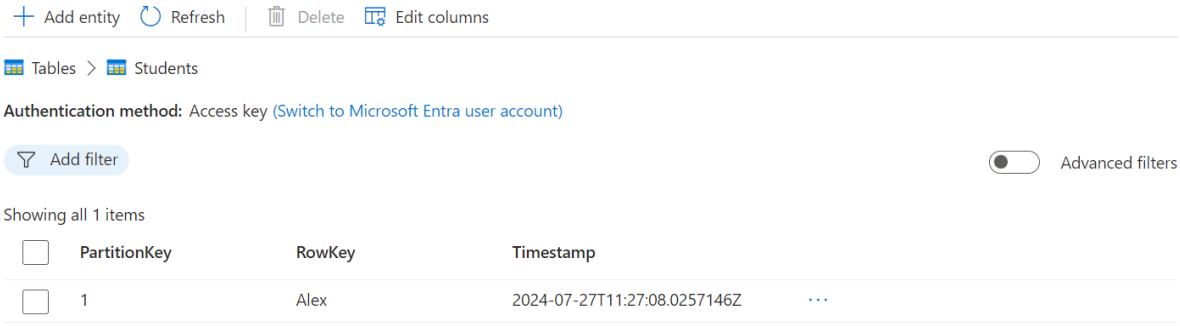
8. Here you will see that your table is empty. Now to add something in your you need to click on add entity.



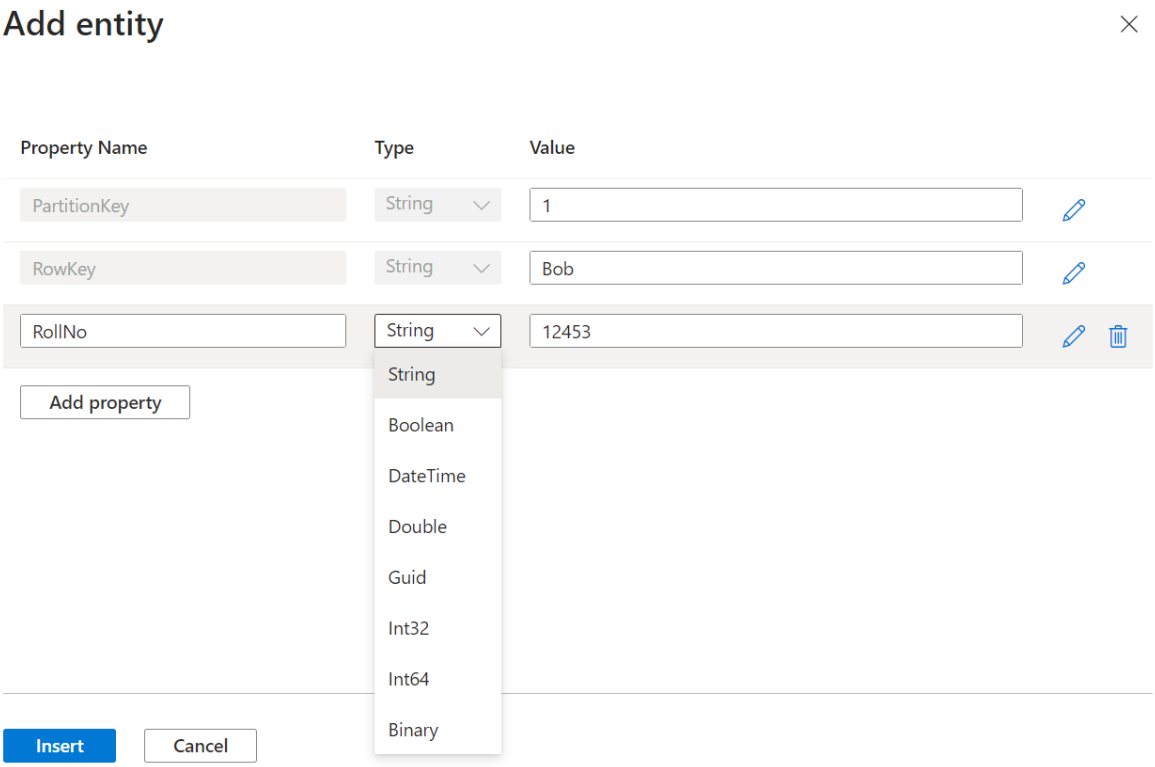
9. You can see that you have two properties Partition Key and Row Key both of type string. So, here we are just providing our value as you can see below. Then you just need to click on insert.



10. Below you can see that you value has been inserted.



11. Now if you click on add entity again, now we will try to add a new property, also when you add a new property you will see that you have so many types.



12. This is how you can work with table storage.