

[Before you start](#)

[Create a Cosmos DB account](#)

[Create a sample database](#)

[View and create items](#)

[Query the database](#)

Explore Azure Cosmos DB

In this exercise you'll provision an Azure Cosmos DB database in your Azure subscription, and explore the various ways you can use it to store non-relational data.

This lab will take approximately **15** minutes to complete.

Before you start

You'll need an [Azure subscription](#) in which you have administrative-level access.

Create a Cosmos DB account

To use Cosmos DB, you must provision a Cosmos DB account in your Azure subscription. In this exercise, you'll provision a Cosmos DB account that uses Azure Cosmos DB for NoSQL.

1. In the Azure portal, select **+ Create a resource** at the top left, and search for *Azure Cosmos DB*. In the results, select **Azure Cosmos DB** and select **Create**.
2. In the **Azure Cosmos DB for NoSQL** tile, select **Create**.
3. Enter the following details, and then select **Review + Create**:
 - **Subscription**: If you're using a sandbox, select *Concierge Subscription*. Otherwise, select your Azure subscription.
 - **Resource group**: If you're using a sandbox, select the existing resource group (which will have a name like *learn-xxxx...*). Otherwise, create a new resource group with a name of your choice.
 - **Account Name**: Enter a unique name
 - **Location**: Choose any recommended location
 - **Capacity mode**: Provisioned throughput
 - **Apply Free-Tier Discount**: Select Apply if available
 - **Limit total account throughput**: Unselected
4. When the configuration has been validated, select **Create**.
5. Wait for deployment to complete. Then go to the deployed resource.

Create a sample database

Throughout this procedure, close any tips that are displayed in the portal.

1. On the page for your new Cosmos DB account, in the pane on the left, select **Data Explorer**.
2. In the **Data Explorer** page, select **Launch quick start**.
3. In the **New container** tab, review the pre-populated settings for the sample database, and then select **OK**.
4. Observe the status in the panel at the bottom of the screen until the **SampleDB** database and its **SampleContainer** container has been created (which may take a minute or so).

View and create items

1. In the Data Explorer page, expand the **SampleDB** database and the **SampleContainer** container, and select **Items** to see a list of items in the container. The items represent product data, each with a unique id and other properties.
2. Select any of the items in the list to see a JSON representation of the item data.
3. At the top of the page, select **New Item** to create a new blank item.
4. Modify the JSON for the new item as follows, and then select **Save**.

Code


 Copy

```
{
  "name": "Road Helmet,45",
  "id": "123456789",
  "categoryID": "123456789",
  "SKU": "AB-1234-56",
  "description": "The product called \"Road Helmet,45\" ",
  "price": 48.74
}
```

5. After saving the new item, notice that additional metadata properties are added automatically.


Query the database

1. In the **Data Explorer** page, select the **New SQL Query** icon.
2. In the SQL Query editor, review the default query (`SELECT * FROM c`) and use the **Execute Query** button to run it.
3. Review the results, which includes the full JSON representation of all items.
4. Modify the query as follows:

Sql	 Copy
<pre>SELECT * FROM c WHERE CONTAINS(c.name, "Helmet")</pre>	

5. Use the **Execute Query** button to run the revised query and review the results, which includes JSON entities for any items with a **name** field containing the text "Helmet".
6. Close the SQL Query editor, discarding your changes.

You've seen how to create and query JSON entities in a Cosmos DB database by using the data explorer interface in the Azure portal. In a real scenario, an application developer would use one of the many programming language specific software development kits (SDKs) to call the NoSQL API and work with data in the database.

 **Tip:** If you've finished exploring Azure Cosmos DB, you can delete the resource group that you created in this exercise.