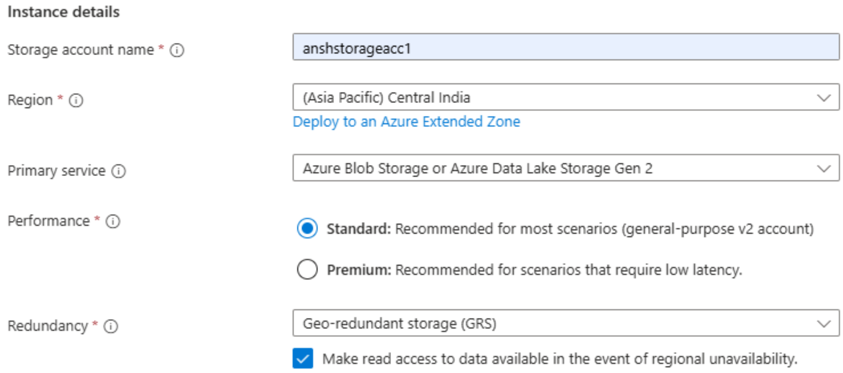
Ansh Ranjan
Azure Data

Exercise 1 – Azure Storage Options for Data

TASK 1 and 3: Creating a new Azure Storage Account and Upload a dataset

1. In Storage Accounts click on Create
2. Select subscription and resource group and give it a name and pick redundancy type



Instance details

Storage account name * ⓘ: anshstorageacc1

Region * ⓘ: (Asia Pacific) Central India

Deploy to an Azure Extended Zone

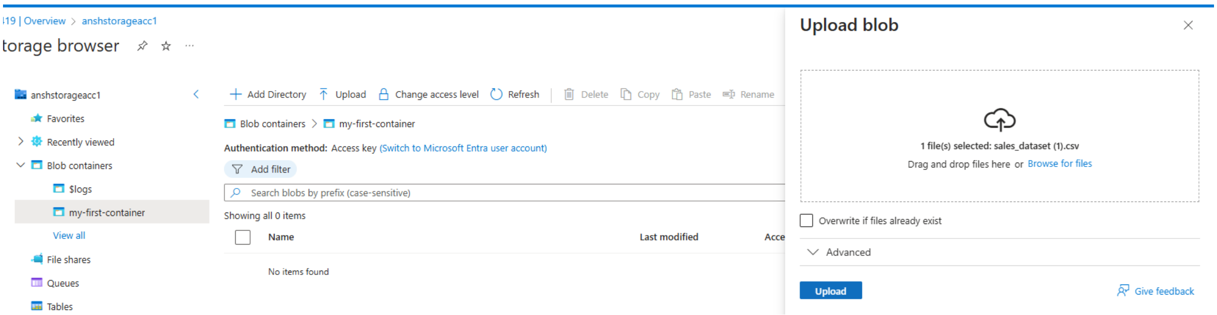
Primary service ⓘ: Azure Blob Storage or Azure Data Lake Storage Gen 2

Performance * ⓘ:
● **Standard:** Recommended for most scenarios (general-purpose v2 account)
○ **Premium:** Recommended for scenarios that require low latency.

Redundancy * ⓘ: Geo-redundant storage (GRS)

☑ Make read access to data available in the event of regional unavailability.

3. Once created, to upload a file go to your storage account > Storage Browser > Blob Containers > Add container > Enter a name > Create
4. Then navigate to your container > Upload > Browser and select the file > click Upload



Authentication method: Access key (Switch to Microsoft Entra user account)

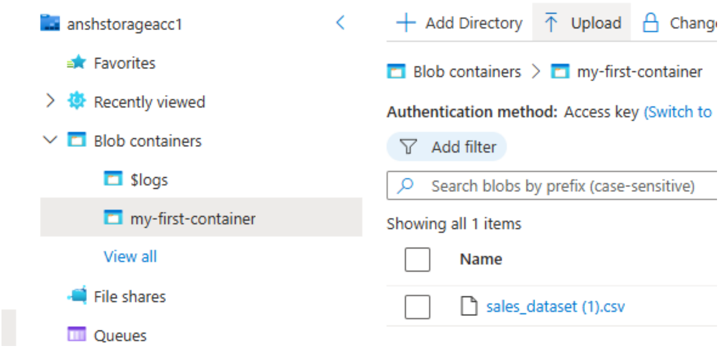
Showing all 0 items

No items found

1 file(s) selected: sales_dataset (1).csv

□ Overwrite if files already exist

5. Now your will see your dataset file in the container



Authentication method: Access key (Switch to

Showing all 1 items

□ sales_dataset (1).csv

TASK 2: Explore difference between Blob Storage, File Storage, Queue Storage, Table Storage

Feature	Blob Storage	File Storage	Queue Storage	Table Storage
Data Type	Unstructured	Structured (files)	Messages	Structured (key-value pairs)
Structure	Blobs in containers	Files in directories	Messages in queues	Entities in tables
Access Protocol	HTTP/HTTPS	SMB/NFS	HTTP/HTTPS	HTTP/HTTPS
Use Cases	Media, backups, analytics	File shares, migrations	Messaging, workflows	NoSQL database, IoT
Scalability	Highly scalable	Up to 100 TiB per share	Millions of messages	Petabytes of data
Pricing	Size and access tier	Provisioned capacity and tier	Operations and data transfer	Data stored and operations

Each storage service in Azure is optimized for specific scenarios:

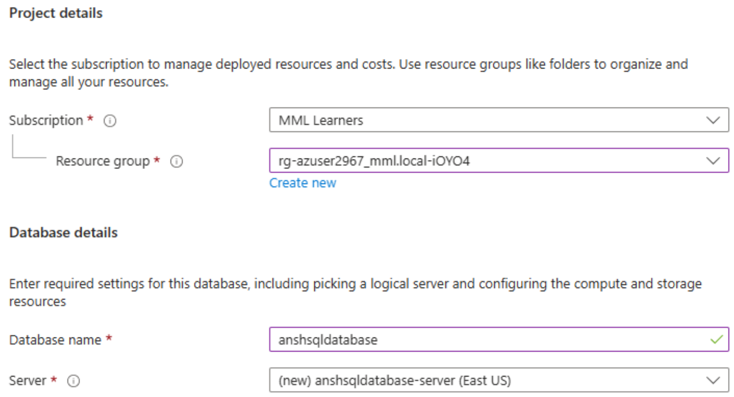
- Use **Blob Storage** for unstructured data like media files and backups.
- Use **File Storage** for shared file systems and legacy applications.
- Use **Queue Storage** for asynchronous messaging between components.
- Use **Table Storage** for structured NoSQL data with high scalability and performance.

EXERCISE 2 - Introduction to Azure Databases

TASK 1: Deploy a sample database in Azure Cosmos DB and Azure SQL Database

- **SQL DATABASE**

1. Go to SQL Databases > Create > Enter details > Create a new database server if you do not have existing option.



Project details

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

Subscription * : MML Learners

Resource group * : rg-azuser2967_mml.local-iOYO4

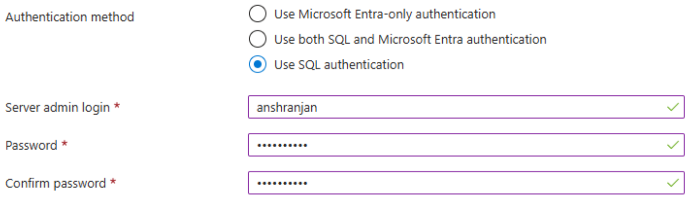
Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name * : anshsqldatabase

Server * : (new) anshsqldatabase-server (East US)

2. Under Network tab set Admin login and Password for your SQL database



Authentication method

○ Use Microsoft Entra-only authentication

○ Use both SQL and Microsoft Entra authentication

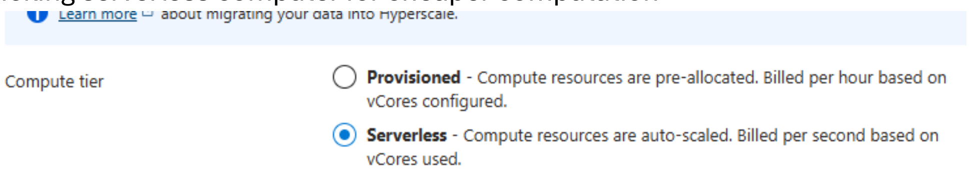
● Use SQL authentication

Server admin login * anshranjan ✓

Password * •••••••••• ✓

Confirm password * •••••••••• ✓

3. Picking serverless computer for cheaper computation



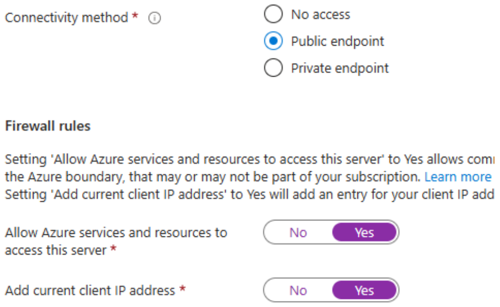
Learn more about migrating your data into Hyperscale.

Compute tier

○ **Provisioned** - Compute resources are pre-allocated. Billed per hour based on vCores configured.

● **Serverless** - Compute resources are auto-scaled. Billed per second based on vCores used.

4. Under Network settings, allow connection to database



Connectivity method *

○ No access

● Public endpoint

○ Private endpoint

Firewall rules

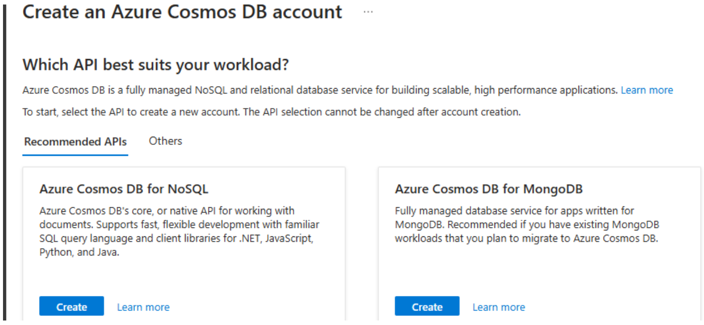
Setting 'Allow Azure services and resources to access this server' to Yes allows comi
the Azure boundary, that may or may not be part of your subscription. Learn more
Setting 'Add current client IP address' to Yes will add an entry for your client IP add

Allow Azure services and resources to access this server * No [✓ Yes]

Add current client IP address * No [✓ Yes]

- **COSMOS DB MONGO DB**

1. Open Cosmos DB > Create > Cozmos DB for MongoDB



Create an Azure Cosmos DB account ...

Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. Learn more

To start, select the API to create a new account. The API selection cannot be changed after account creation.

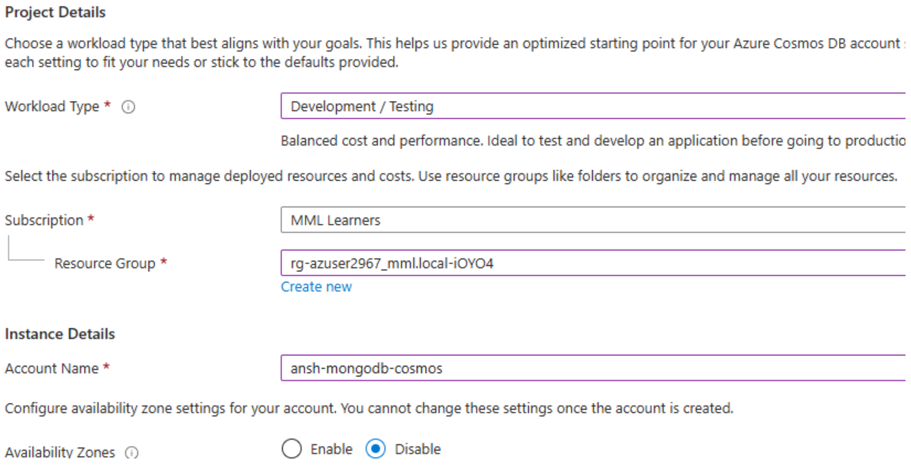
Azure Cosmos DB for NoSQL

Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.

Azure Cosmos DB for MongoDB

Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB.

2. Request Unit database account > Enter details > Review and Create



Project Details

Choose a workload type that best aligns with your goals. This helps us provide an optimized starting point for your Azure Cosmos DB account each setting to fit your needs or stick to the defaults provided.

Workload Type * Development / Testing

Balanced cost and performance. Ideal to test and develop an application before going to productio

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

Subscription * MML Learners

Resource Group * rg-azuser2967_mml.local-iOYO4

Create new

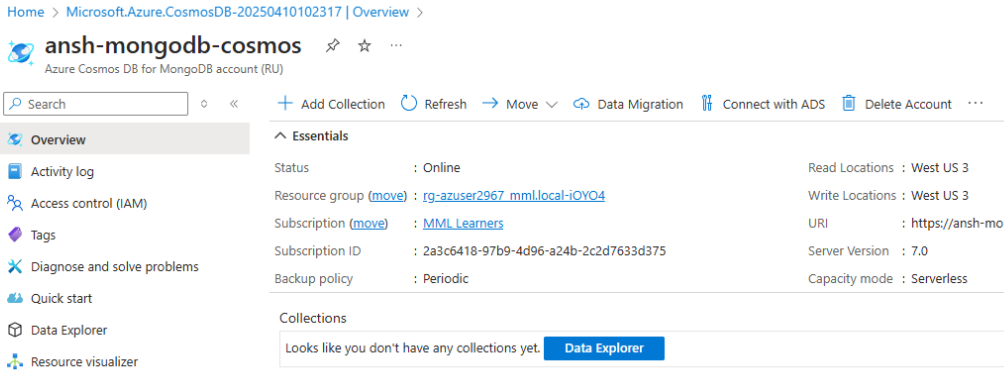
Instance Details

Account Name * ansh-mongodb-cosmos

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

Availability Zones ○ Enable ● Disable

3. Your Azure Cosmos DB API for Mongo DB will be created and running


Home > Microsoft.Azure.CosmosDB-20250410102317 | Overview >

ansh-mongodb-cosmos

Azure Cosmos DB for MongoDB account (RU)

Essentials

Status	: Online	Read Locations	: West US 3
Resource group (move)	: rg-azuser2967_mml.local-iOYO4	Write Locations	: West US 3
Subscription (move)	: MML Learners	URI	: https://ansh-mo
Subscription ID	: 2a3c6418-97b9-4d96-a24b-2c2d7633d375	Server Version	: 7.0
Backup policy	: Periodic	Capacity mode	: Serverless

Collections

Looks like you don't have any collections yet.

TASK 2: Document key features and use cases for each

Azure Synapse Analytics

Key Features:

- Unified platform for big data and data warehousing.
- Massively Parallel Processing (MPP) for large datasets.
- Integrated pipelines for ETL/ELT with Azure Data Factory.
- Synapse Studio for data exploration and analytics.
- Scalable, secure, and supports machine learning.

Use Cases:

- Data warehousing and big data analytics.
- Business intelligence with Power BI integration.
- Advanced analytics and IoT data processing.

Azure SQL Database

Key Features:

- Fully managed relational database service.
- High availability, scalability, and automated maintenance.
- Elastic pools for resource sharing.
- Advanced security and geo-replication.
- Seamless integration with other Azure services.

Use Cases:

- Transactional workloads (OLTP).
- Backend for web/mobile apps and e-commerce systems.
- ERP/CRM databases and lightweight analytics.

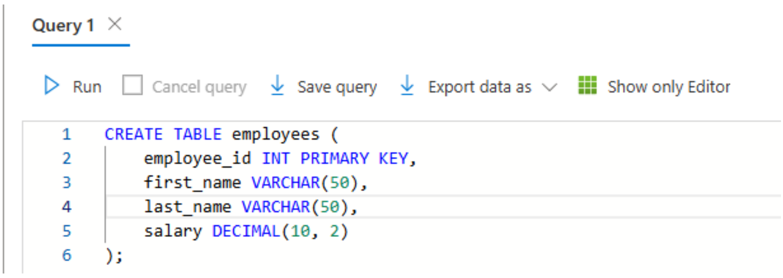
Comparison of Use Cases

Feature/Use Case	Azure Synapse Analytics	Azure SQL Database
Primary Focus	Analytical workloads (OLAP)	Transactional workloads (OLTP)
Data Volume	Petabytes of data	Gigabytes to terabytes of data
Scalability	Massively parallel processing	Elastic scaling for transactional data
Integration	Big data tools, Power BI, Data Lake	Web apps, mobile apps, and business apps
Machine Learning	Advanced analytics and AI workloads	Limited to lightweight analytics
Use Case Examples	Data warehousing, predictive analytics	E-commerce systems, ERP/CRM databases

TASK 3: Perform basic CRUD operations

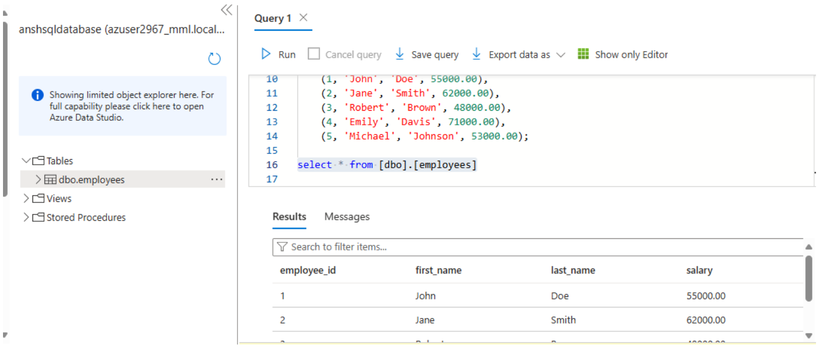
- **SQL DATABASE**

1. Go to your DB > Query Editor > Login with admin ID and password
2. You will be presented with query page. Write a query to create a table in your database



```sql
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    salary DECIMAL(10, 2)
);
```

3. Insert records into your database table and read them


anshsqldatabase (azuser2967_mml.local...
Showing limited object explorer here. For full capability please click here to open Azure Data Studio.
Tables
dbo.employees
Views
Stored Procedures
Query 1
Run
Cancel query
Save query
Export data as
Show only Editor
Results
Messages
Search to filter items...

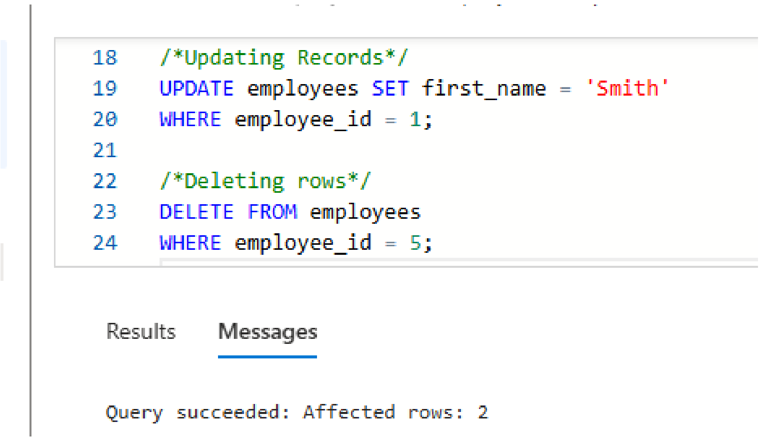
```sql
    (1, 'John', 'Doe', 55000.00),
    (2, 'Jane', 'Smith', 62000.00),
    (3, 'Robert', 'Brown', 48000.00),
    (4, 'Emily', 'Davis', 71000.00),
    (5, 'Michael', 'Johnson', 53000.00);

select * from [dbo].[employees]

```

employee_id	first_name	last_name	salary
1	John	Doe	55000.00
2	Jane	Smith	62000.00

4. Updating and Deleting records



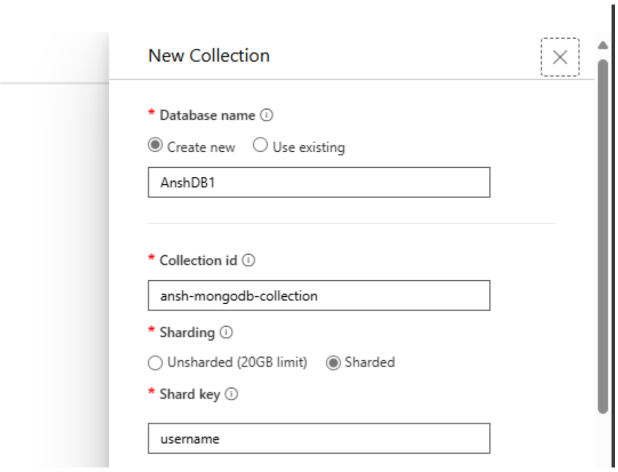
```
/*Updating Records*/
UPDATE employees SET first_name = 'Smith'
WHERE employee_id = 1;

/*Deleting rows*/
DELETE FROM employees
WHERE employee_id = 5;
```

```
Query succeeded: Affected rows: 2
```

- **COSMOS DB MONGO DB**

1. Go to Data Explorer > Create Database > New Collection > Select Database or create new, give collection id, shard key



New Collection

* Database name

● Create new ○ Use existing

AnshDB1

* Collection id

ansh-mongodb-collection

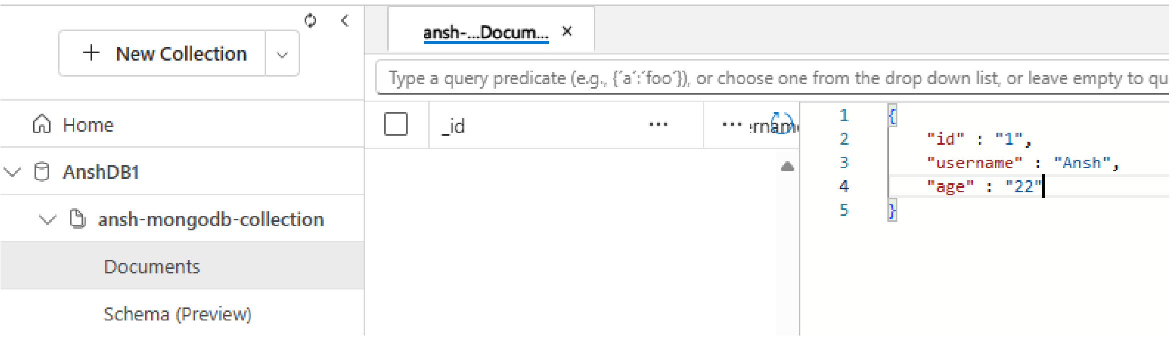
* Sharding

○ Unsharded (20GB limit) ● Sharded

* Shard key

username

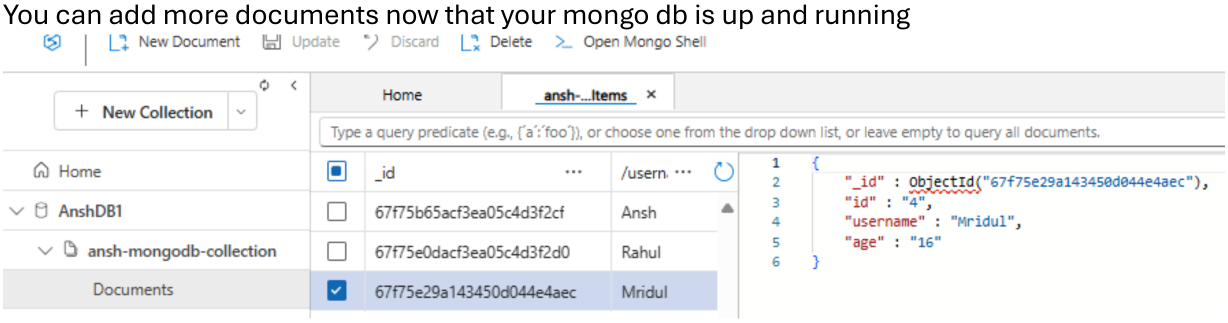
2. Once your collection is created > click New Document and enter data in document > Save



New Collection
Home
AnshDB1
ansh-mongodb-collection
Documents
Schema (Preview)
ansh-...Docum...
Type a query predicate (e.g., {´a´:´foo´}), or choose one from the drop down list, or leave empty to qu
_id
rname

```
{
    "id" : "1",
    "username" : "Ansh",
    "age" : "22"
}
```

3. You can add more documents now that your mongo db is up and running



New Document
Update
Discard
Delete
Open Mongo Shell
New Collection
Home
AnshDB1
ansh-mongodb-collection
Documents
Home
ansh-...Items
Type a query predicate (e.g., {´a´:´foo´}), or choose one from the drop down list, or leave empty to query all documents.

	_id	/usern
☐	67f75b65acf3ea05c4d3f2cf	Ansh
☐	67f75e0dacf3ea05c4d3f2d0	Rahul
☑	67f75e29a143450d044e4aec	Mridul

```
{
    "_id" : ObjectId("67f75e29a143450d044e4aec"),
    "id" : "4",
    "username" : "Mridul",
    "age" : "16"
}
```

EXERCISE 3 - Data Security and Compliance in Azure

1. Research and Document Azure's Data Encryption Methods

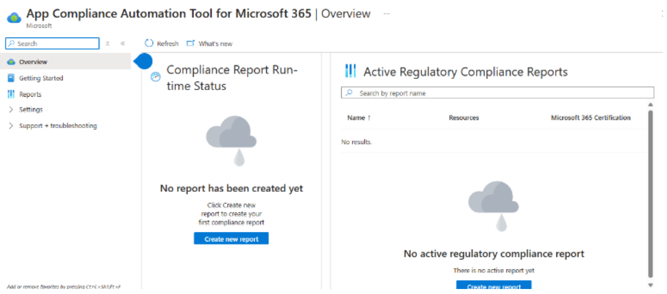
- **What to do**:
  - Research how Azure secures data through encryption. Include details on encryption at rest, encryption in transit, and customer-managed keys.
  - Cover services like Azure Storage encryption, SQL Database encryption (TDE), and Azure Key Vault.
  - Summarize your findings in **1 page** and save it as a **PDF document**.
- **How to do it**:
  - Use Azure documentation (https://learn.microsoft.com/en-us/azure/) for research.
  - Tools like Word or Google Docs can help you write and export your document as a PDF.

2. Enable Encryption for a Sample Azure Storage Account

- **What to do**:
  - Create or use an existing Azure Storage Account.
  - Enable encryption for the storage account (Azure encrypts all storage accounts at rest by default).
  - Optionally, configure **customer-managed keys** for encryption using Azure Key Vault.
- **How to do it**:
  - Log in to the Azure Portal (https://portal.azure.com).
  - Navigate to your storage account.
  - Under **Settings**, go to **Encryption** and verify or enable encryption.
  - If using customer-managed keys, integrate Azure Key Vault.

3. Explore Azure Compliance Manager for Data Regulation

- **What to do**:
  - Use Azure Compliance Manager to explore how Azure helps meet regulatory requirements (e.g., GDPR, HIPAA).
  - Review the compliance score and understand how to improve it.
- **How to do it**:
  - Log in to the Azure Portal.
  - Search for **Compliance Manager** in the portal.
  - Explore the dashboard, review assessments, and understand the regulatory frameworks Azure supports.



App Compliance Automation Tool for Microsoft 365 | Overview

Microsoft

Compliance Report Run-time Status

No report has been created yet

Click Create new report to create your first compliance report

Active Regulatory Compliance Reports

Search by report name

Name ↑	Resources	Microsoft 365 Certification
No results.		

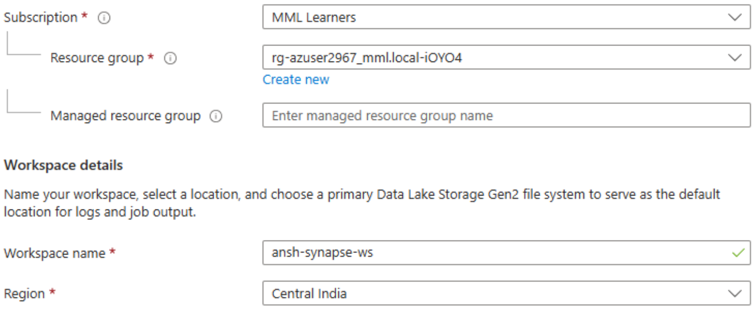
No active regulatory compliance report

There is no active report yet

EXERCISE 4: Azure Synapse Analytics

TASK 1: Deploy a sample Azure Synapse Analytics workspace

1. Go to Azure Synapse Analytics > Create > Enter Details



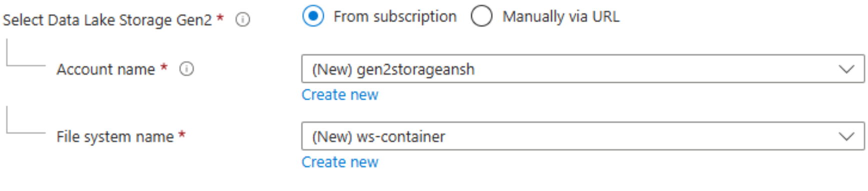
Subscription *	MML Learners
Resource group *	rg-azuser2967_mml.local-iOYO4
Managed resource group	Enter managed resource group name

Workspace details

Name your workspace, select a location, and choose a primary Data Lake Storage Gen2 file system to serve as the default location for logs and job output.

Workspace name *	ansh-synapse-ws
Region *	Central India

2. Select or create a Dala Lake Gen 2 storage



Select Data Lake Storage Gen2 * ● From subscription ○ Manually via URL

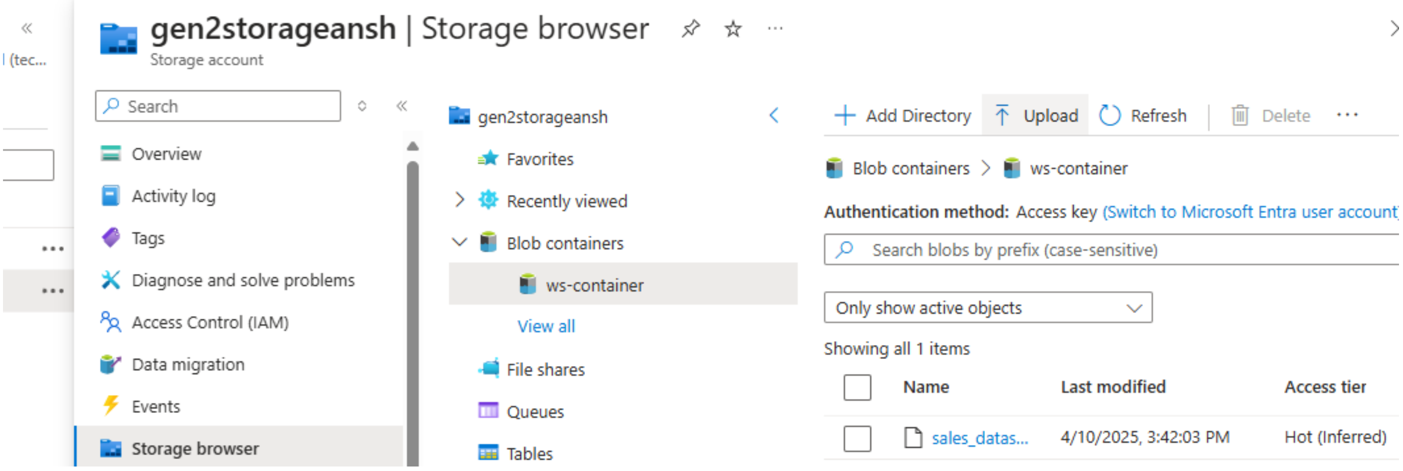
Account name *	(New) gen2storageansh
File system name *	(New) ws-container

3. Once again set a sql server admin login and password

4. Click on Review and Create

TASK 2: Load a sample dataset and perform basic queries

1. Upload a dataset in your container created



gen2storageansh | Storage browser

Storage account

Authentication method: Access key (Switch to Microsoft Entra user account

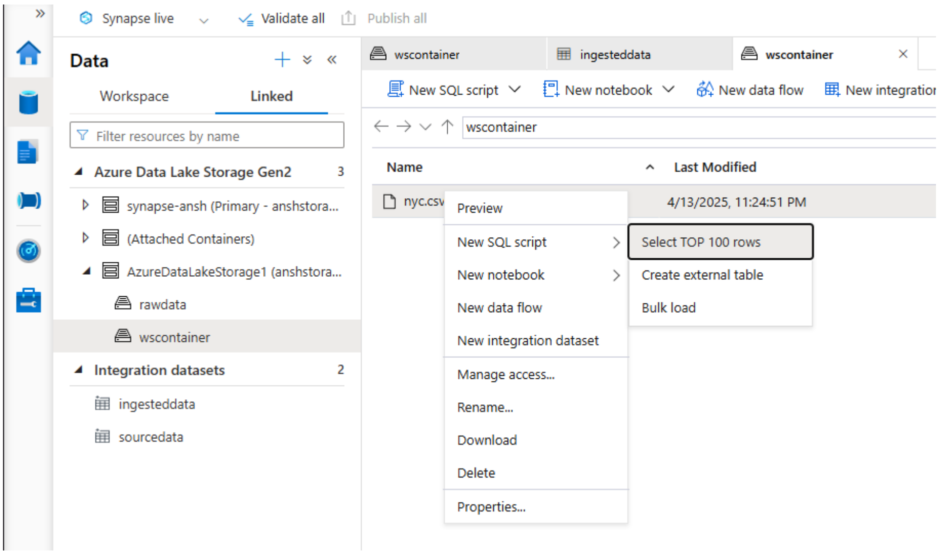
Search blobs by prefix (case-sensitive)

Only show active objects

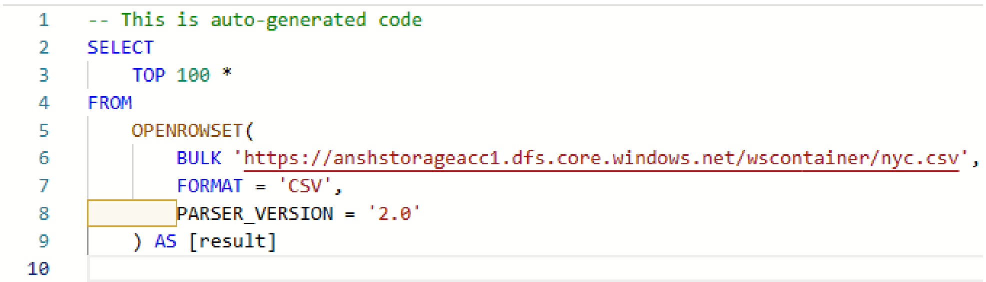
Showing all 1 items

□	Name	Last modified	Access tier
□	sales_datas...	4/10/2025, 3:42:03 PM	Hot (Inferred)

2. Go to Data side tab > browse to your container > right click on ingested data > Select top 100 rows



3. This will write a sql script for you to display first 100 rows of your ingested dataset.



```sql
-- This is auto-generated code
SELECT
    TOP 100 *
FROM
    OPENROWSET(
        BULK 'https://anshstorageacc1.dfs.core.windows.net/wscontainer/nyc.csv',
        FORMAT = 'CSV',
        PARSER_VERSION = '2.0'
    ) AS [result]

```

4. You can execute the query using the serverless sql pool provided while creating the workspace.
5. However due to firewall limitations we are unable to use either serverless sql pools of dedicated sql pools in this azure account.

TASK 3: Document Azure Synapse Key Benefits and Use cases

Azure Synapse Key Benefits:

1. **Unified Analytics Platform**: Combines big data and data warehousing into a single platform, allowing for streamlined data analysis.
2. **Scalability**: Offers on-demand scalability, enabling you to handle large datasets and workloads without managing infrastructure.
3. **Integrated AI and Machine Learning**: Built-in integration with Azure Machine Learning and cognitive services to run advanced analytics and AI models.
4. **Real-time Analytics**: Supports real-time data streaming and analytics, providing insights with minimal delay.
5. **End-to-End Security**: Features robust security with encryption, firewalls, threat protection, and compliance with industry standards.
6. **Optimized Query Performance**: Leverages in-memory processing, parallel query execution, and caching for faster query performance.
7. **Serverless and Provisioned Models**: Offers both serverless querying and provisioned resources, allowing cost flexibility.
8. **Easy Integration with Other Azure Services**: Seamlessly integrates with tools like Power BI, Azure Data Factory, and Azure Databricks for enhanced data processing and visualization.

Azure Synapse Use Cases:

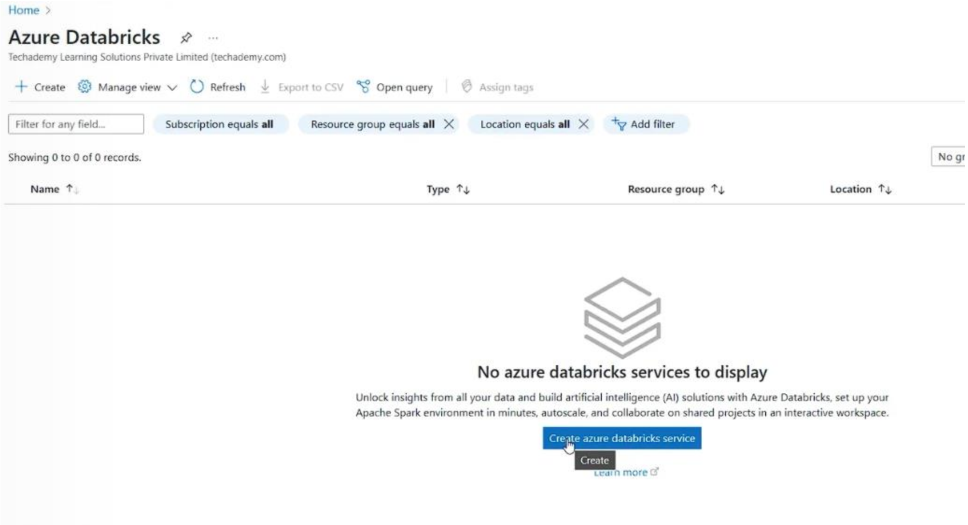
1. **Data Warehousing**: Store, manage, and analyze large datasets with high-performance query capabilities.

2. **Real-Time Analytics**: Process and analyze real-time streaming data for quick decision-making.
3. **Big Data Processing**: Work with massive amounts of unstructured and structured data using Spark and other big data tools.
4. **Business Intelligence**: Integrate with Power BI for advanced reporting, dashboards, and visual analytics.
5. **Advanced Analytics and AI**: Run predictive analytics and machine learning models directly within the platform.
6. **ETL/ELT Pipelines**: Build data pipelines using Azure Data Factory to move, transform, and load data for further analysis.
7. **Data Lakes**: Store raw, unstructured data in Azure Data Lake and process it using Synapse's integrated tools.
8. **Cost-Effective Storage**: Archive historical data with minimal cost by utilizing the platform's tiered storage model.

EXERCISE 5: DataBricks for Data Engineering

TASK 1: Deploy a DataBricks workspace

1. **Sign in to the Azure Portal**:
   - Go to Azure Portal and log in with your credentials.
2. **Create a New Resource**:
   - Click **Create a resource** on the left-hand menu.
   - Search for **Azure Databricks** in the search bar.
   - Select **Azure Databricks** from the results and click **Create**.



Azure Databricks

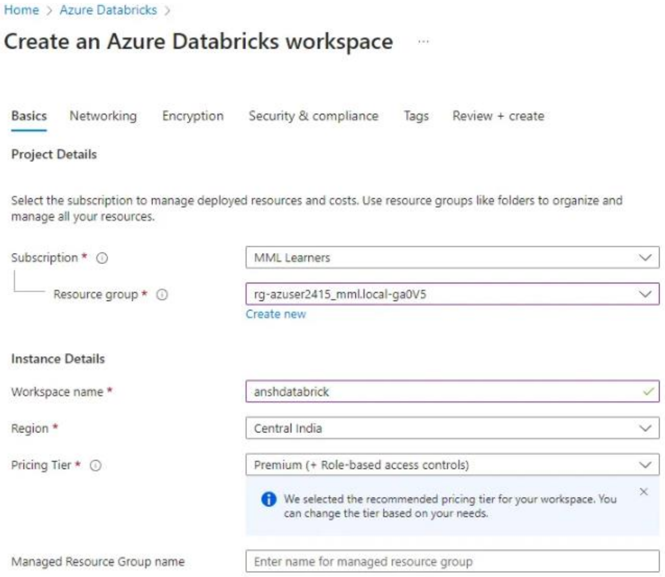
Techademy Learning Solutions Private Limited (techademy.com)

Showing 0 to 0 of 0 records.

No azure databricks services to display

Unlock insights from all your data and build artificial intelligence (AI) solutions with Azure Databricks, set up your Apache Spark environment in minutes, autoscale, and collaborate on shared projects in an interactive workspace.

3. **Configure the Workspace**:
   - **Resource Group**: Choose an existing resource group or create a new one.
   - **Workspace Name**: Provide a unique name for your Databricks workspace.
   - **Region**: Select the Azure region closest to your users or data source for better performance.
   - **Pricing Tier**: Select a pricing tier based on your requirements (Standard, Premium, or Trial).



Create an Azure Databricks workspace

Project Details

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

Subscription *: MML Learners

Resource group *: rg-azuser2415_mml.local-ga0V5

Instance Details

Workspace name *: anshdatabrick

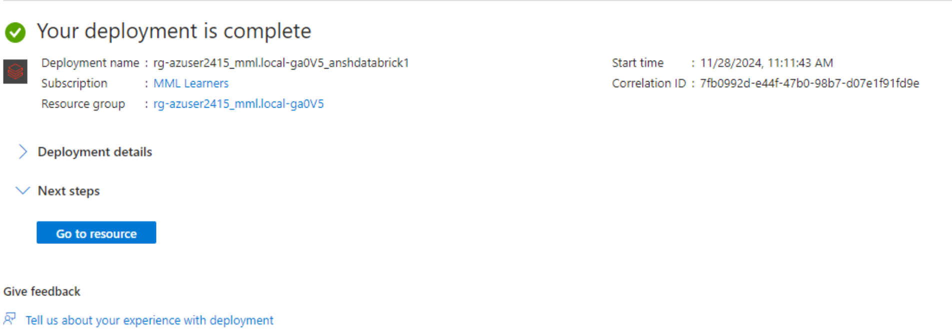
Region *: Central India

Pricing Tier *: Premium (+ Role-based access controls)

We selected the recommended pricing tier for your workspace. You can change the tier based on your needs.

Managed Resource Group name: Enter name for managed resource group

4. **Networking (Optional)**:
   - Configure network settings if required, such as deploying the workspace in a Virtual Network (VNet).
5. **Review + Create**:
   - Click **Review + Create** to validate your configuration.
   - Once validation is successful, click **Create** to deploy the workspace.
6. **Wait for Deployment**:
   - The deployment process might take a few minutes. Monitor the progress in the **Notifications** section of the portal.



Your deployment is complete

Deployment name : rg-azuser2415_mml.local-ga0V5_anshdatabrick1

Subscription : MML Learners

Resource group : rg-azuser2415_mml.local-ga0V5

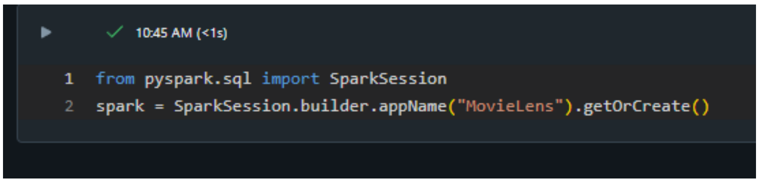
Start time : 11/28/2024, 11:11:43 AM

Correlation ID : 7fb0992d-e44f-47b0-98b7-d07e1f91fd9e

7. **Access the Workspace**:
   - Once deployment is complete, go to the **Resource** to access the Databricks workspace.
   - Click the **Launch Workspace** button to open the Azure Databricks environment.

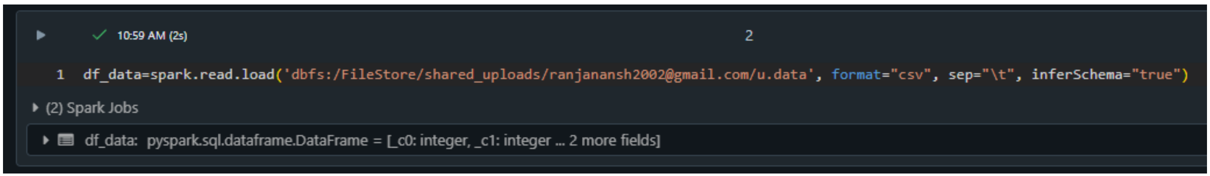
TASK 2: Process a big sample dataset

1. Go to your DataBricks Workspace > Create > New Notebook and initialize your spark session. You will need a cluster to run queries



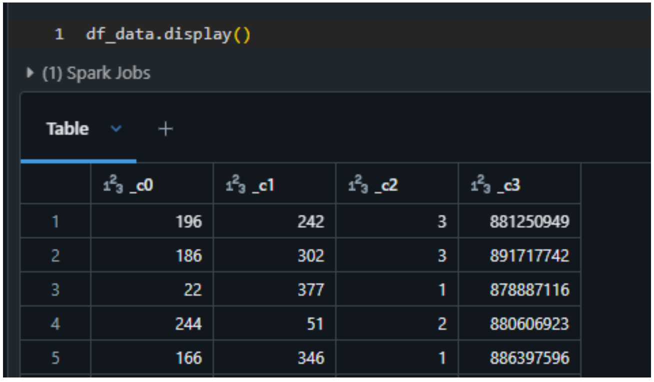
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("MovieLens").getOrCreate()
```

2. Extraction



```
df_data=spark.read.load('dbfs:/FileStore/shared_uploads/ranjanansh2002@gmail.com/u.data', format="csv", sep="\t", inferSchema="true")
```

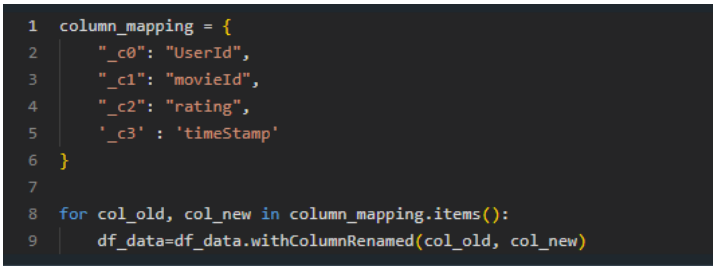
df_data: pyspark.sql.dataframe.DataFrame = [_c0: integer, _c1: integer ... 2 more fields]



```
df_data.display()
```

	_c0	_c1	_c2	_c3
1	196	242	3	881250949
2	186	302	3	891717742
3	22	377	1	878887116
4	244	51	2	880606923
5	166	346	1	886397596

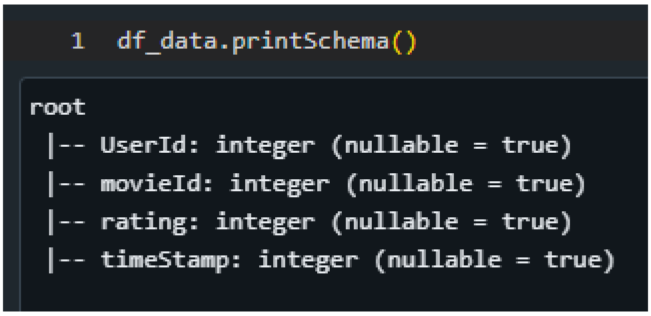
5. Since no column names are provided, we need to add them manually



```
column_mapping = {
    "_c0": "UserId",
    "_c1": "movieId",
    "_c2": "rating",
    '_c3' : 'timeStamp'
}

for col_old, col_new in column_mapping.items():
    df_data=df_data.withColumnRenamed(col_old, col_new)
```

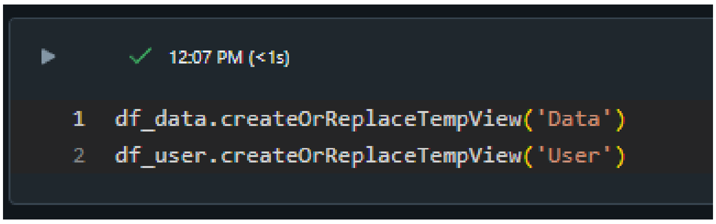
6. Getting Schema of the dataframe



```
df_data.printSchema()
```

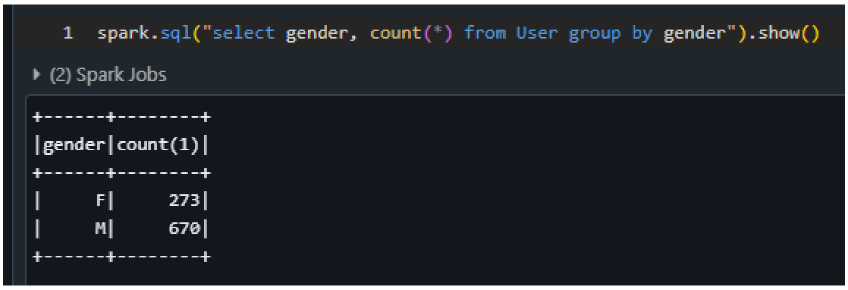
```
root
 |-- UserId: integer (nullable = true)
 |-- movieId: integer (nullable = true)
 |-- rating: integer (nullable = true)
 |-- timeStamp: integer (nullable = true)
```

7. Repeating same steps for other dataframe and creating Views



```
df_data.createOrReplaceTempView('Data')
df_user.createOrReplaceTempView('User')
```

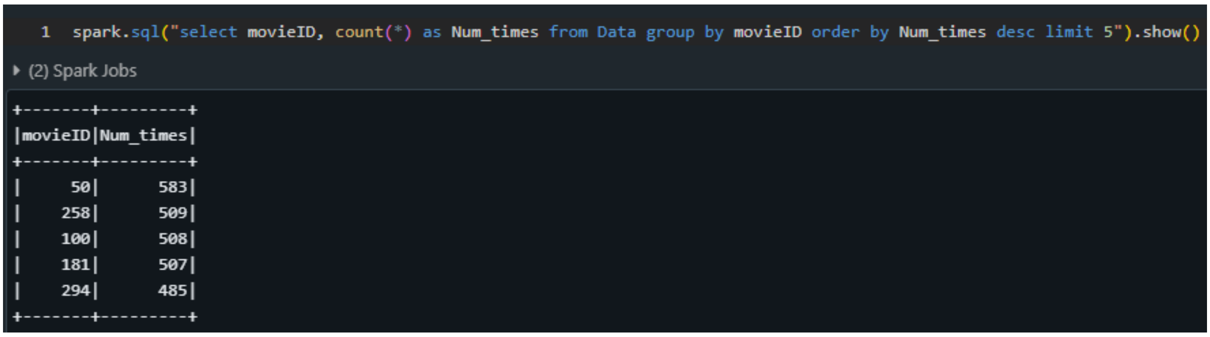
8. Deriving Analytics from the data
   Gender wise user breakdown



```
spark.sql("select gender, count(*) from User group by gender").show()
```

```
+------+--------+
|gender|count(1)|
+------+--------+
|     F|     273|
|     M|     670|
+------+--------+
```

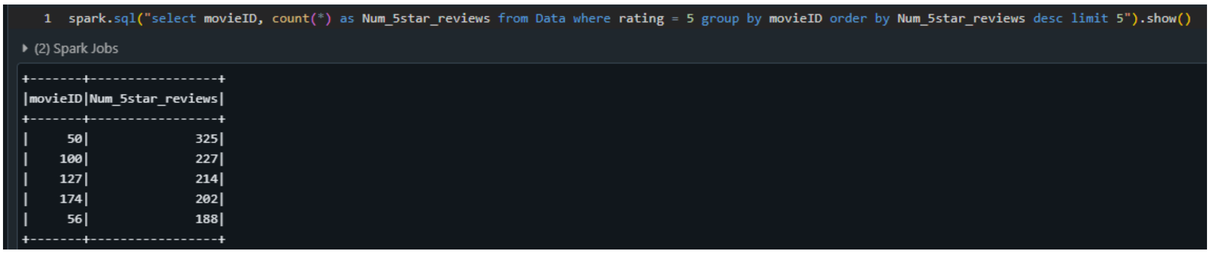
Give the top 5 movies which are reviewed maximum number of times



```
spark.sql("select movieID, count(*) as Num_times from Data group by movieID order by Num_times desc limit 5").show()
```

```
+-------+---------+
|movieID|Num_times|
+-------+---------+
|     50|      583|
|    258|      509|
|    100|      508|
|    181|      507|
|    294|      485|
+-------+---------+
```

List the top 10 movies which received highest number of 5 star ratings



```
spark.sql("select movieID, count(*) as Num_5star_reviews from Data where rating = 5 group by movieID order by Num_5star_reviews desc limit 5").show()
```

```
+-------+-----------------+
|movieID|Num_5star_reviews|
+-------+-----------------+
|     50|              325|
|    100|              227|
|    127|              214|
|    174|              202|
|     56|              188|
+-------+-----------------+
```

TASK 3: DataBricks Key Features and use cases

Key Features of Databricks

- Unified platform for **data engineering, analytics, and ML**
- Built on **Apache Spark** for fast, distributed processing
- Supports **Delta Lake** with ACID transactions and schema enforcement
- Collaborative **notebooks** with multi-language support (Python, SQL, etc.)
- Built-in **visualizations** and **MLflow** for ML lifecycle
- **Auto-scaling clusters**, CI/CD, and cloud integration (Azure, AWS, GCP)

Use Cases

- **ETL & Data Pipelines**: Ingest, transform, and clean large datasets
- **Data Lakehouse**: Unified storage and analytics using Delta Lake
- **Machine Learning**: Build, train, and deploy ML models
- **Real-Time Analytics**: Process streaming data
- **Business Intelligence**: Connect with Power BI/Tableau for reporting