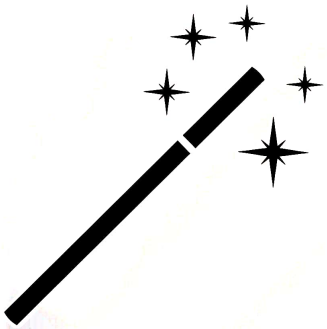

Databricks Magic Commands for Data Engineers

Databricks Magic Commands enhance productivity by simplifying tasks like file management, environment settings, and data exploration.

What Are Magic Commands?

✅ **Magic Commands** in Databricks start with % and run outside standard Spark/PySpark code.

✅ They help with **file system operations, environment settings, and SQL execution.**

✅ Common magic command categories:

- **File System Access** (%fs)
- **Execution Control** (%run, %sh)
- **SQL Queries** (%sql)
- **Python-Specific Commands** (%python)

Boost efficiency & simplify workflows with these hidden gems!

File System Operations (%fs)

Interact with Databricks **DBFS (Databricks File System)** seamlessly!

✅ **List files** in a directory:

%fs ls /mnt/data

✅ **Copy files:**

%fs cp /mnt/data/file1.csv /mnt/data/backup/

✅ **Move files:**

%fs mv /mnt/data/file1.csv /mnt/archive/

✅ **Remove files:**

%fs rm /mnt/data/temp.csv

Easily manage files without using complex Python scripts!

Running External Notebooks (%run)

Execute another Databricks notebook inside your current notebook!

✅ **Run a notebook from another workspace location:**

```
%run "/Workspace/Shared/Data_Preprocessing"
```

Great for modularizing code & reusing functions across multiple notebooks!

Running SQL Queries (%sql)

✅ **Execute SQL queries directly in a notebook:**

```
%sql
SELECT * FROM sales_data LIMIT 10;
```

✅ **Create a temporary SQL table from a DataFrame:**

```
df.createOrReplaceTempView("sales_view")
%sql
SELECT region, SUM(revenue) FROM sales_view GROUP BY
region;
```

Combine SQL & PySpark seamlessly!

Running Shell Commands (%sh)

Execute terminal commands directly from a notebook!

✅ **Check system resources:**

%sh free -m

✅ **List installed libraries:**

%sh pip list

✅ **Download a file from the internet:**

%sh wget https://example.com/data.csv -P /dbfs/tmp/

Useful for debugging & package installations!

Switching Between Languages

Run code in different languages within the same notebook!

✅ **Python Code:**

```
%python
print("Hello from Python!")
```

✅ **Scala Code:**

```
%scala
println("Hello from Scala!")
```

✅ **R Code:**

```
%r
print("Hello from R!")
```

Perfect for multi-language data engineering workflows!

Environment & Configuration Settings (%config, %env)

✅ **View Databricks environment configurations:**

%config

✅ **Check Spark settings:**

%config spark

✅ **List environment variables:**

%env

Quickly inspect runtime settings & environment variables!

Managing Mount Points in DBFS

Mount external storage (Azure Blob, ADLS, AWS S3) to Databricks File System (DBFS)

✅ **List all mounts:**

%fs mounts

✅ **Mount an external storage account:**

```
dbutils.fs.mount(
 source =
"wasbs://mycontainer@myaccount.blob.core.windows.net",
 mount_point = "/mnt/mydata",
 extra_configs =
{"fs.azure.account.key.myaccount.blob.core.windows.net": "
<access-key>"}
)
```

Seamlessly access cloud storage without manual authentication!

Clearing Cache & Optimizing Queries

✅ **Clear all cached tables** to free up memory:

%sql CLEAR CACHE;

✅ **Unpersist cached DataFrames:**

df.unpersist()

Helps manage memory usage in long-running notebooks!

Final Takeaways – Magic Commands = Efficiency!

✅ **%fs** – Manage files in DBFS

✅ **%run** – Execute external notebooks

✅ **%sql** – Run SQL queries in notebooks

✅ **%sh** – Execute shell commands

✅ **%python, %scala, %r** – Multi-language execution

✅ **%config, %env** – Inspect environment settings

✅ **%fs mounts** – Manage external storage access

Databricks Magic Commands = Faster Development & Streamlined Workflows!