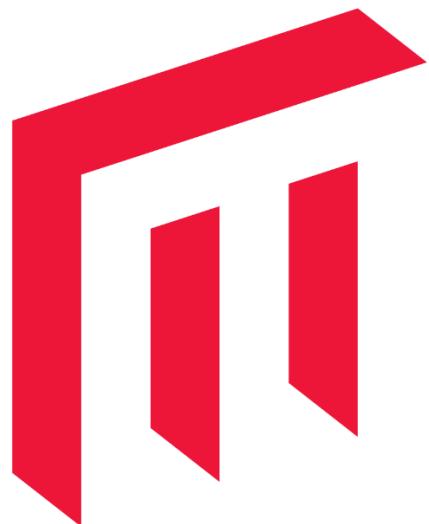


# Lab 4



**MALMÖ  
UNIVERSITY**

**Big Data Analytics on Cloud  
Computing Infrastructures**

**Azad Shokrollahi, Mahtab Jamali**



## 1. What is Databricks?

Databricks is a cloud platform built for big-data processing and machine learning with Apache Spark. It provides:

- Spark clusters for distributed computing
- Notebooks (Python, SQL, Scala, R)
- Machine learning with MLlib and MLflow
- Streaming and ETL pipelines
- Lakehouse architecture

### **Important point:**

- Databricks does **not** have its own cloud.
- It always runs on one of these clouds:
  - Microsoft Azure
  - Amazon AWS
  - Google GCP

Databricks is a platform that sits on top of these clouds and manages Spark for you.

---

## 2. Two Ways to Run Databricks (World 1 vs World 2)

Databricks can be used in two different ways, depending on where you start:

- Cloud provider users (Azure/AWS/GCP) → World 1
- Databricks website users → World 2

This creates two access paths.

---

### WORLD 1 — Cloud-First Databricks

This is when you start from the cloud portal, not from [databricks.com](#).

How you use it:

- Go to Azure Portal, AWS Console, or GCP Console
- Search for “Databricks”
- Create a Databricks workspace
- The cloud provider creates all resources:
  - Virtual machines
  - Networking
  - Storage
  - Spark clusters

What you get:

- Full Spark cluster support
- PySpark
- MLlib
- Streaming
- MLflow
- Workflows and jobs
- All enterprise features

Why it exists:

- Cloud providers want Databricks integrated into their ecosystem.

Best for:

- Students with Azure for Students credits
- University labs
- Companies already using Azure, AWS, or GCP

In World 1, you create Databricks inside the cloud, and Spark runs on your cloud account.

---

## **WORLD 2 — Databricks-First (Upgrade Free Edition)**

This starts from the Databricks website, not from the cloud portal.

How you use it:

- Go to databricks.com and sign in
- You get the Free Edition
- Click “Upgrade”
- Databricks asks: “Choose your cloud — Azure, AWS, or GCP?”
- Databricks deploys your workspace into that cloud
- You attach your cloud account for billing

Why it exists:

- Many companies buy Databricks directly from Databricks
- They need to choose a cloud during setup
- This is Databricks' direct onboarding path

What you get after upgrading:

- Full Spark clusters
- Full compute power
- MLLib
- MLflow
- Jobs, workflows, and all features

In World 2, you start from the Databricks website, upgrade, choose a cloud, and Databricks creates your workspace in that cloud.

---

### 3. World 2 History: Community Edition → Free Edition

Before 2024, World 2 included Community Edition.

Old: Community Edition (free Spark)

- Free Spark cluster
- PySpark support
- MLlib machine learning
- Real distributed Spark jobs
- No cloud account needed
- Good for learning and teaching
- Completely free

Why Community Edition was removed:

- Too expensive to give free Spark to everyone
- Heavy workloads caused problems
- Databricks wanted a lightweight free tier

New: Free Edition (no Spark)

Free Edition provides:

- A small serverless SQL warehouse
- Ability to run SQL queries
- Ability to explore the Databricks interface

But it does NOT provide:

- Spark clusters
- PySpark
- MLlib
- Streaming
- Machine learning
- Workflows or jobs
- Any cluster creation

Big idea:

Free Edition = SQL only.

Community Edition = free Spark (now removed).

---

## 4. Comparison Table

Feature	Community Edition (OLD)	Free Edition (NEW)
<b>Spark cluster</b>	Yes	No
<b>PySpark</b>	Yes	No
<b>MLlib</b>	Yes	No
<b>SQL support</b>	Yes	Yes
<b>Distributed compute</b>	Yes	No
<b>Real cluster attach</b>	No (internal only)	No
<b>Cloud usage</b>	Databricks internal	Databricks internal
<b>Purpose</b>	Learning Spark	Exploring UI + SQL
<b>Status</b>	Removed	Active

---

## 5. How to Get Real Spark Today

Because Free Edition has no Spark, you must use one of these options:

- Azure Databricks (recommended for students, free Azure credits available)
- AWS Databricks free trial (14 days)
- GCP Databricks free trial
- Paid Databricks workspace linked to your cloud account

All of these options provide real Spark clusters.

# Free Edition

<https://www.databricks.com/learn/free-edition>

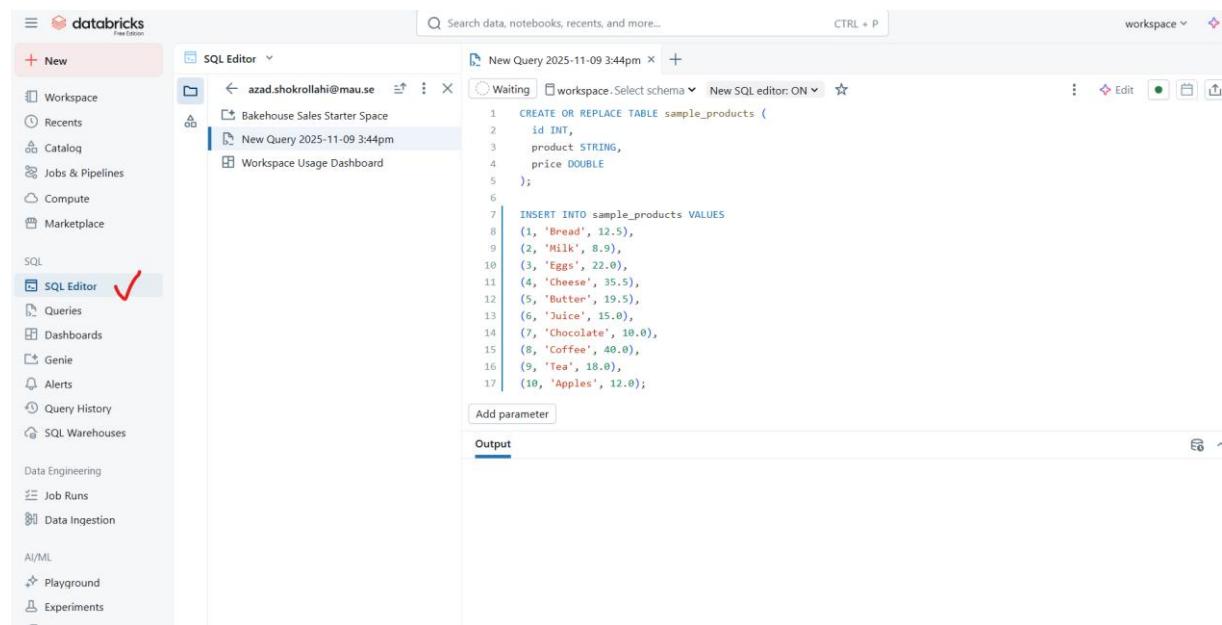
The screenshot shows the Databricks Free Edition homepage. The left sidebar contains navigation links for Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Marketplace, SQL, and various AI/ML features like Genie, Alerts, Query History, and Data Warehouses. The main content area features a "Welcome to Databricks" header and a "Connect your data" section with three cards: "Upload data", "Browse public datasets", and "Connect to 50+ data sources". Below this is a "Explore sample projects" section with three cards: "Explore your first notebook", "Converse with data using natural language", and "Build a GenAI chatbot". At the bottom, there's a "Take a course" section and a "View all" link.

The screenshot shows the Databricks Free Edition Compute page. The left sidebar has a "Compute" section highlighted with a red checkmark, and a "SQL Warehouses" link is also highlighted. The main content area displays a table of SQL warehouses. The columns are Status, Name, Created by, Size, Active / Max, and Type. There is one entry: "Serverless Starter Warehouse" created by azad.shokrollahi@mau.se, 2X-Small size, 0 / 1 active, and Serverless type. A "Create SQL" button is located at the top right of the table.

Status	Name	Created by	Size	Active / Max	Type
Ø	Serverless Starter Warehouse	azad.shokrollahi@mau.se	2X-Small	0 / 1	Serverless

This screen shows the only compute available in Databricks Free Edition—a small serverless SQL warehouse that runs SQL queries but cannot run Spark or machine-learning code.

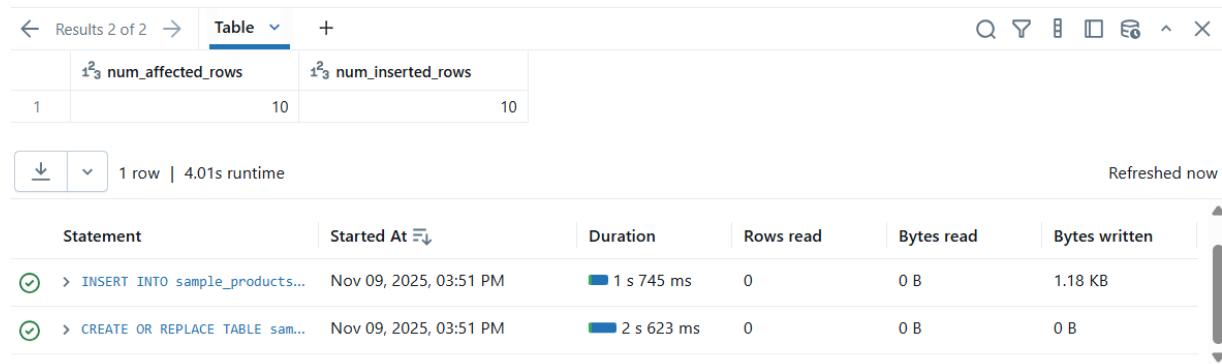
Databricks replaced Community Edition with Free Edition to limit free compute costs by removing Spark clusters and allowing only a small serverless SQL warehouse for basic SQL use.



The screenshot shows the Databricks Free Edition interface. The left sidebar includes sections for Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Marketplace, SQL (with SQL Editor selected), Data Engineering, AI/ML, and more. The main area is titled "SQL Editor" and shows a query titled "New Query 2025-11-09 3:44pm". The query code is:

```
1 CREATE OR REPLACE TABLE sample_products (
2   id INT,
3   product STRING,
4   price DOUBLE
5 );
6
7 INSERT INTO sample_products VALUES
8   (1, 'Bread', 12.5),
9   (2, 'Milk', 8.9),
10  (3, 'Eggs', 22.0),
11  (4, 'Cheese', 35.5),
12  (5, 'Butter', 19.5),
13  (6, 'Juice', 15.0),
14  (7, 'Chocolate', 18.0),
15  (8, 'Coffee', 40.0),
16  (9, 'Tea', 18.0),
17  (10, 'Apples', 12.0);
```

The "Output" tab is selected at the bottom.



The screenshot shows the Databricks Free Edition interface with two tabs: "Results" and "History".

**Results Tab:**

	1 num_affected_rows	1 num_inserted_rows
1	10	10

1 row | 4.01s runtime  
Refreshed now

**History Tab:**

Statement	Started At	Duration	Rows read	Bytes read	Bytes written
> INSERT INTO sample_products...	Nov 09, 2025, 03:51 PM	1 s 745 ms	0	0 B	1.18 KB
> CREATE OR REPLACE TABLE sam...	Nov 09, 2025, 03:51 PM	2 s 623 ms	0	0 B	0 B

The screenshot shows the Databricks interface with the SQL Editor tab selected. In the Catalog sidebar, the 'sample\_products' table under the 'default' schema is selected. A red checkmark is placed next to the table name. The main area displays the SQL code for creating the table and inserting 10 rows of sample data. Below the code, the results of the run are shown in a table with two columns: 'num\_affected\_rows' and 'num\_inserted\_rows', both with a value of 10. The run details table at the bottom shows two operations: 'INSERT INTO sample\_products...' and 'CREATE OR REPLACE TABLE ...', both completed successfully with 0 rows read and 0 B bytes written.

```

CREATE OR REPLACE TABLE sample_products (
  id INT,
  product STRING,
  price DOUBLE
);

INSERT INTO sample_products VALUES
(1, 'Bread', 12.5),
(2, 'Milk', 8.9),
(3, 'Eggs', 22.0),
(4, 'Cheese', 35.5),
(5, 'Butter', 19.5),
(6, 'Juice', 15.0),
(7, 'Chocolate', 10.0),
(8, 'Coffee', 40.0),
(9, 'Tea', 18.0),
(10, 'Apples', 12.0);

```

	Table
1	num_affected_rows
1	num_inserted_rows

Statement	Started At	Duration	Rows read	Bytes read	Bytes written
> INSERT INTO sample_products...	Nov 09, 2025, 03:51 PM	1 s 745 ms	0	0 B	1.18 KB
> CREATE OR REPLACE TABLE sam...	Nov 09, 2025, 03:51 PM	2 s 623 ms	0	0 B	0 B

## Create notebook

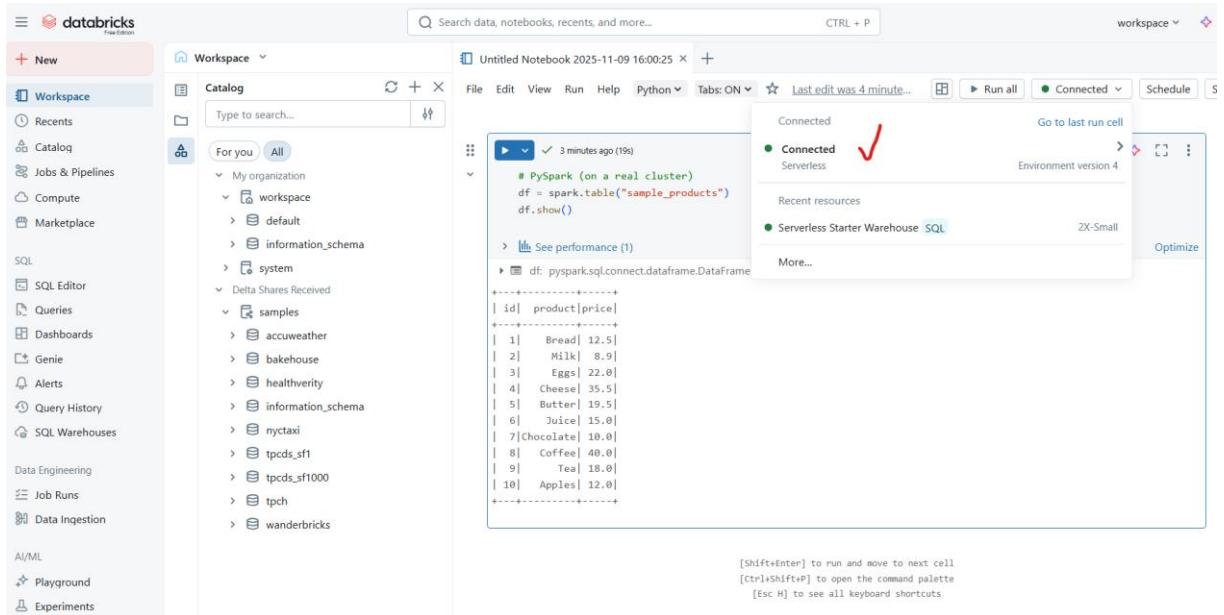
The screenshot shows the Databricks interface with the Workspace tab selected. In the Catalog sidebar, the 'sample\_products' table under the 'default' schema is selected. The main area shows a Python notebook cell containing PySpark code to read the 'sample\_products' table and display its contents. The output of the cell shows the 10 rows of sample data as a DataFrame.

```

# PySpark (on a real cluster)
df = spark.table("sample_products")      # or spark.read.table("sample_products")
df.show()

```

id   product   price
1   Bread   12.5
2   Milk   8.9
3   Eggs   22.0
4   Cheese   35.5
5   Butter   19.5
6   Juice   15.0
7   Chocolate   10.0
8   Coffee   40.0
9   Tea   18.0
10   Apples   12.0



# Databricks Lab

## Student Tasks

### Public Dataset

download the dataset from this link:

#### Tips Dataset (CSV)

<https://raw.githubusercontent.com/mwaskom/seaborn-data/master/tips.csv>

Save it as: **tips.csv**

This dataset contains the following columns:

total\_bill, tip, sex, smoker, day, time, size

### Tasks

## Task 1 — Start the SQL Warehouse

- Go to **Compute → SQL Warehouses**
  - Start your **Serverless Starter Warehouse**
  - Take a screenshot showing it is *Connected*
- 

## Task 2 — Upload the Public Dataset

- Go to **Catalog → default**
  - Click **Create → Add data → Upload file**
  - Upload **tips.csv**
  - Name the table **tips\_data**
  - Select your SQL warehouse
  - Create the table
  - Provide a screenshot showing the table under the **default** catalog
- 

## Task 3 — Explore the Dataset (SQL)

Write and run SQL queries to answer the following.

Provide a screenshot for each result.

1. Display the first **20 rows** of the dataset
  2. Count the **total number of rows**
  3. Find the **average total bill**
  4. Show the **top 5 highest total bills**
  5. Calculate the **total tip amount for each day**
  6. Compare the **average bill of smokers vs non-smokers**
  7. Compare the **average tip amount for lunch vs dinner**
- 

## Task 4 — Create a New Calculated Column

Create a column named **tip\_percentage**  
(tip divided by total\_bill times 100)

Then display the following for all rows:

- total\_bill
- tip
- tip\_percentage

Screenshot required.

---

### Task 5 — Create a New Table

Create a new table named **high\_spenders** containing only rows where:

**total\_bill > 30**

Then:

- Display the contents of **high\_spenders**
  - Show how many rows this table contains
  - Provide screenshots
- 

### Task 6 — Read the Tables in a Python Notebook

Open a Python notebook and:

- Read the **tips\_data** table using `spark.table()`
- Read the **high\_spenders** table
- Display both DataFrames using `.show()`

Provide screenshots of the outputs.

---

### Task 7 — Export Results

Download a CSV export of the **high\_spenders** table using the Databricks download option.

Provide a screenshot showing that the download completed.

---

### Task 8 — Reflection

Explore at least **five** items from the left navigation menu in Databricks.

For **each** of the five items:

- Take **one screenshot**, and
  - Write **2–5 sentences** explaining what you observed.
- 

## Optional Task (For Students With Cloud Access)

*This task is optional and only for students who have Azure, AWS, or GCP credits.*

Students who have access to a cloud account (Azure for Students credit, AWS free trial, or GCP trial) may complete the following optional task to run **real Spark and machine learning using MLLib**.

### Optional: Run Machine Learning on Spark (Cloud Databricks)

If you choose to complete this task:

1. Create a Databricks workspace on one of the cloud providers:
  - Azure Databricks
  - AWS Databricks
  - GCP Databricks
2. Create a **Spark cluster** inside your cloud workspace
3. Create a new **Python notebook**
4. Upload or load the **tips.csv** dataset
5. Using PySpark, complete the following steps:
  - Load the CSV into a Spark DataFrame
  - Select appropriate features and label
  - Assemble features using **VectorAssembler**
  - Train a simple model using Spark MLLib, such as:
    - Logistic Regression
    - Decision Tree
    - Linear Regression
  - Show predictions

- Evaluate the model using accuracy or RMSE

6. Take screenshots showing:

- The ML model training
- The prediction output

### **Optional Deliverable**

A short summary including:

- Which cloud you used
- Your Spark cluster settings
- The ML algorithm you ran
- One screenshot of your model output