

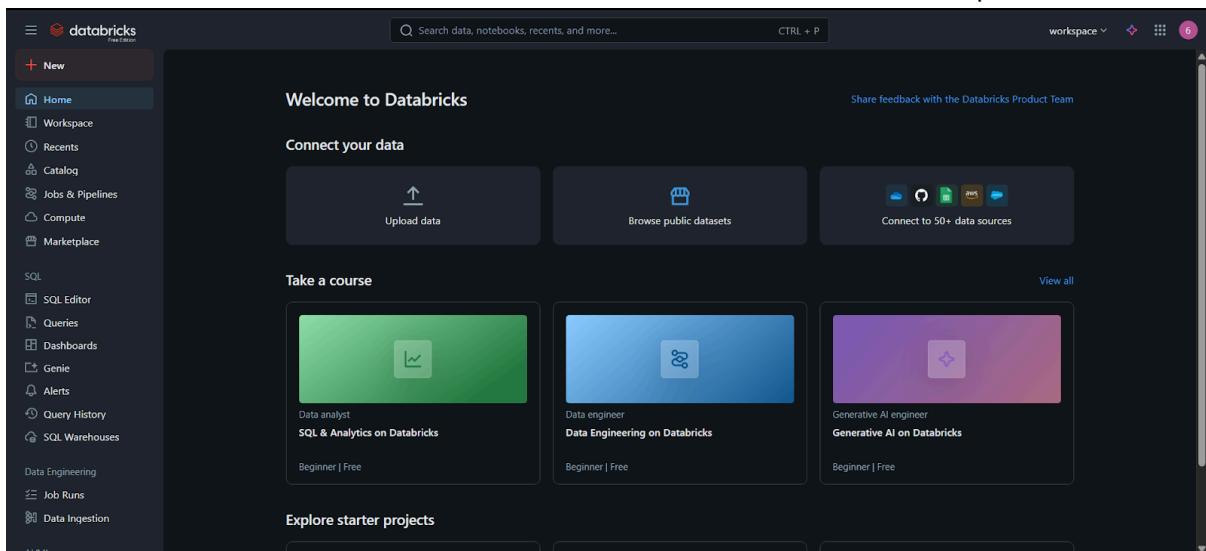
Intro

- sign up/login > ใช้การ login ด้วย google email นักศึกษา หรือจะสมัครเมลที่เราอยากใช้ก็ได้ ตรงขึ้นตอนการสมัครน่าจะไม่มีอะไรมากเลยข้ามค่ะ
- <https://docs.databricks.com/aws/en/getting-started/>

Query and visualize data

<https://docs.databricks.com/aws/en/getting-started/quick-start>

Note: ในส่วนนี้เราจะมาใช้งาน Databricks โดยเริ่มต้นจากการสร้าง Notebook ซึ่งหน้าตา Notebook จะคล้ายๆกับ colab, jupyter notebook หรือ kaggle notebook (คาดว่าน้องๆจะคุ้นเคยกันดีแล้ว)

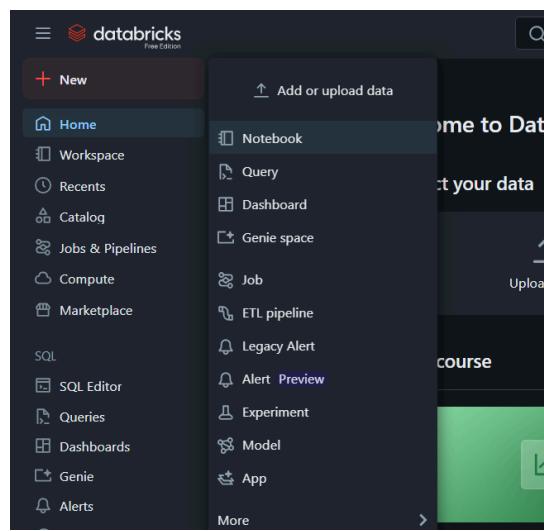


ในหน้าแรกหลังจาก login เข้ามาแล้ว จะมีหน้าตาแบบภาพข้างบน 🤝

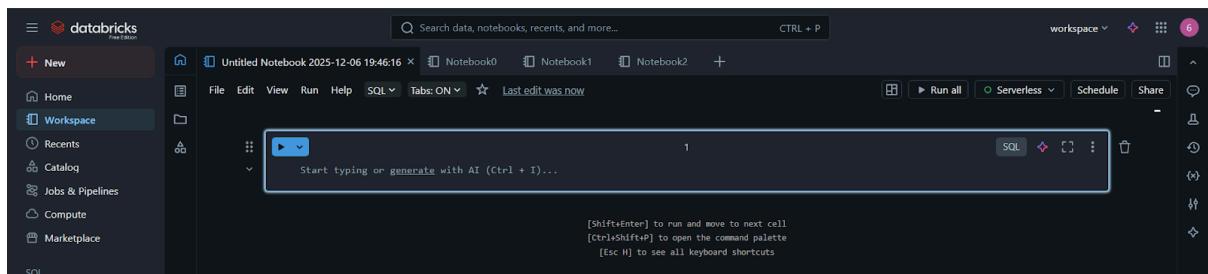
Requirements สำหรับการเริ่มต้นใน tutorial นี้ (default หลังจากเราสร้าง account มา ใน workspace ปกติเราจะมีสิ่งที่ req 2 ข้อนี้พร้อมแล้ว)

- Unity Catalog ต้องเปิดใช้งานอยู่
<https://docs.databricks.com/aws/en/data-governance/unity-catalog/get-started>
- ต้องมีสิทธิ์ในการใช้งาน/สร้างทรัพยากรประมวลผล

ที่ sidebar ด้านซ้าย คลิกปุ่ม '+ New' > Notebook

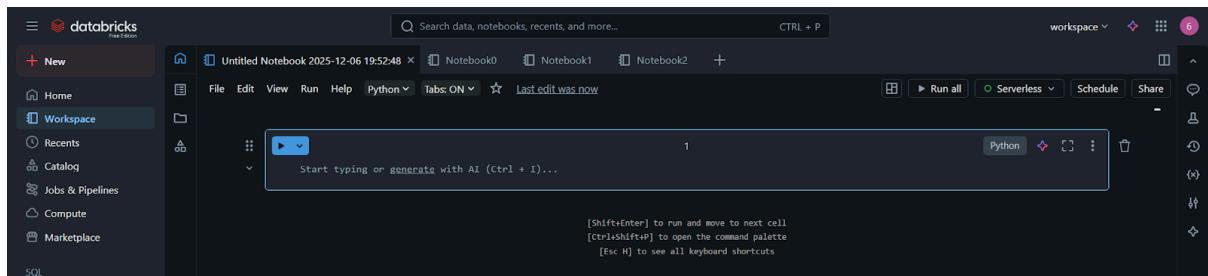


ตัวอย่างหลังจากกดสร้าง Notebook มาแล้ว (default เดิมจะเป็น python ค่ะ)



The screenshot shows the Databricks workspace interface. On the left, there's a sidebar with options like Home, Workspace, Recents, Catalog, Jobs & Pipelines, Compute, and Marketplace. The main area is titled "Untitled Notebook 2025-12-06 19:46:16" and has tabs for Notebook0, Notebook1, Notebook2, and a "+" button. The notebook editor is in "SQL" mode, indicated by the "SQL" tab at the top right of the code cell. The cell itself contains the placeholder text "Start typing or generate with AI (Ctrl + I)...".

default python ver.



This screenshot is similar to the previous one, but the notebook editor is set to "Python" mode, as shown by the "Python" tab at the top right of the code cell. The rest of the interface, including the sidebar and the notebook title, remains the same.

เรารสามารถคลิกที่ชื่อ Notebook เพื่อเปลี่ยนชื่อด้วย (จะเห็นว่ามี Notebook 0, 1, 2 ที่ลองเล่นมาก่อนหน้านี้ด้วย)

ต่อไป เราจะ query เอาข้อมูลตัวอย่างมาลง visualize ดู โดยเริ่มจากการนำเข้าข้อมูลตัวอย่างมาก่อน

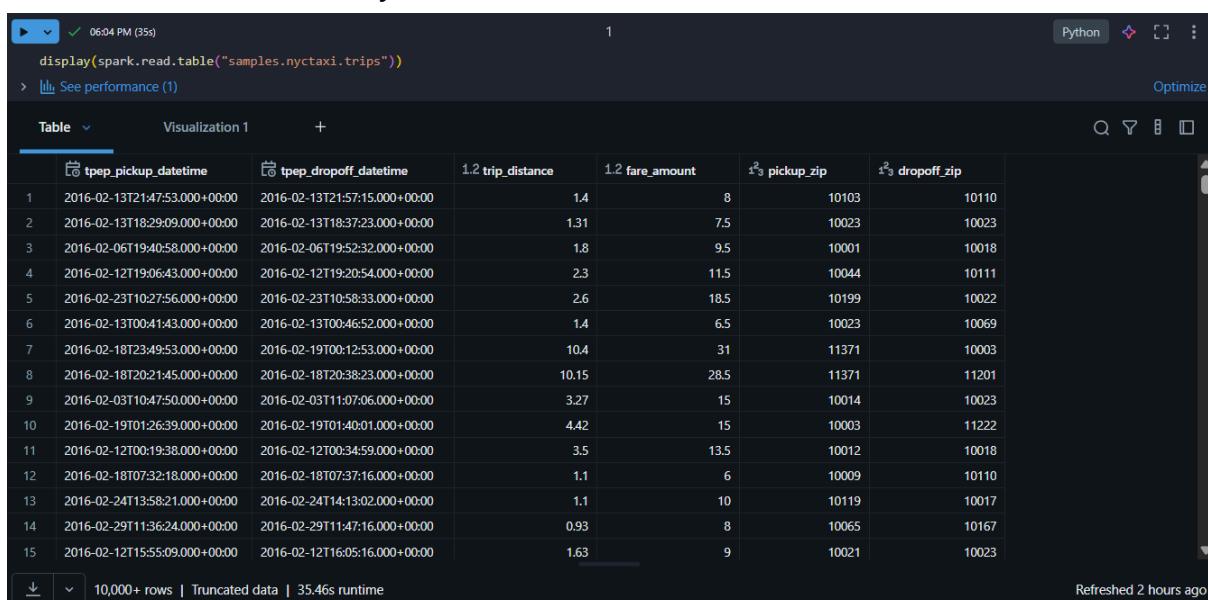
- sql : SELECT * FROM samples.nyctaxi.trips
- python : display(spark.read.table("samples.nyctaxi.trips"))

(เลือกมา 2 แบบ เพราะว่าในองค์กรจะคุ้น 2 ตัวนี้มากที่สุด ทั้งนี้ มีภาษา R, scala ให้ใช้งานด้วย ดูเพิ่มเติมใน docs ได้)

ในที่นี้ เลือกใช้แบบ python นะค่ะ

Shift + Enter > run this cell and move to next cell

Ctrl + Enter > run this cell only



The screenshot shows the result of running the Python code "display(spark.read.table("samples.nyctaxi.trips"))". It displays a table visualization of taxi trip data. The table has columns: tpep_pickup_datetime, tpep_dropoff_datetime, 1.2 trip_distance, 1.2 fare_amount, 1.2 pickup_zip, and 1.2 dropoff_zip. The data consists of 15 rows of taxi trip information. At the bottom, it says "10,000+ rows | Truncated data | 35.46s runtime" and "Refreshed 2 hours ago".

	tpep_pickup_datetime	tpep_dropoff_datetime	1.2 trip_distance	1.2 fare_amount	1.2 pickup_zip	1.2 dropoff_zip
1	2016-02-13T21:47:53.000+00:00	2016-02-13T21:57:15.000+00:00	1.4	8	10103	10110
2	2016-02-13T18:29:09.000+00:00	2016-02-13T18:37:23.000+00:00	1.31	7.5	10023	10023
3	2016-02-06T19:40:58.000+00:00	2016-02-06T19:52:32.000+00:00	1.8	9.5	10001	10018
4	2016-02-12T19:06:43.000+00:00	2016-02-12T19:20:54.000+00:00	2.3	11.5	10044	10111
5	2016-02-23T10:27:56.000+00:00	2016-02-23T10:58:33.000+00:00	2.6	18.5	10199	10022
6	2016-02-13T00:41:43.000+00:00	2016-02-13T00:46:52.000+00:00	1.4	6.5	10023	10069
7	2016-02-18T23:49:53.000+00:00	2016-02-19T00:12:53.000+00:00	10.4	31	11371	10003
8	2016-02-18T20:21:45.000+00:00	2016-02-18T20:38:23.000+00:00	10.15	28.5	11371	11201
9	2016-02-03T10:47:50.000+00:00	2016-02-03T11:07:06.000+00:00	3.27	15	10014	10023
10	2016-02-19T01:26:39.000+00:00	2016-02-19T01:40:01.000+00:00	4.42	15	10003	11222
11	2016-02-12T00:19:38.000+00:00	2016-02-12T00:34:59.000+00:00	3.5	13.5	10012	10018
12	2016-02-18T07:32:18.000+00:00	2016-02-18T07:37:16.000+00:00	1.1	6	10009	10110
13	2016-02-24T13:58:21.000+00:00	2016-02-24T14:13:02.000+00:00	1.1	10	10119	10017
14	2016-02-29T11:36:24.000+00:00	2016-02-29T11:47:16.000+00:00	0.93	8	10065	10167
15	2016-02-12T15:55:09.000+00:00	2016-02-12T16:05:16.000+00:00	1.63	9	10021	10023

ตัวอย่างหลังรันแล้ว จะได้ข้อมูลประมาณนี้

ต่อไป ลอง visualize แบบง่ายๆ คลิก ‘+’ ข้าง Table เลือก Visualization

A screenshot of a Jupyter Notebook cell. At the top, there is a code block:

```
display(spark.read.table("samples.nyctaxi.trips"))
```

. Below it is a table with three rows. The first row has a blue background and contains the column name `tpep_pickup_datetime`. The second row shows the date `2016-02-23`. The third row shows the date `2016-02-13`. To the right of the table, a context menu is open with three options: `Table`, `+`, and `Visualization`. The `Visualization` option is highlighted. A preview of the visualization is shown on the right, displaying the `datetime` column with values `033.000+00:00` and `52.000+00:00`.

ในหน้าต่าง Visualization Editor

A screenshot of the `Visualization Editor` window. On the left, there is a sidebar with various configuration options: `General`, `X axis`, `Y axis`, `Series`, `Colors`, and `Data labels`. Under `General`, the `Visualization type` is set to `Bar`. In the center, there is a large icon of a box with a minus sign and the text `No Data`. Below it, a message says `Please choose at least one Y column to create a chart.`. At the bottom right, there are `Cancel` and `Save` buttons.

เลือก

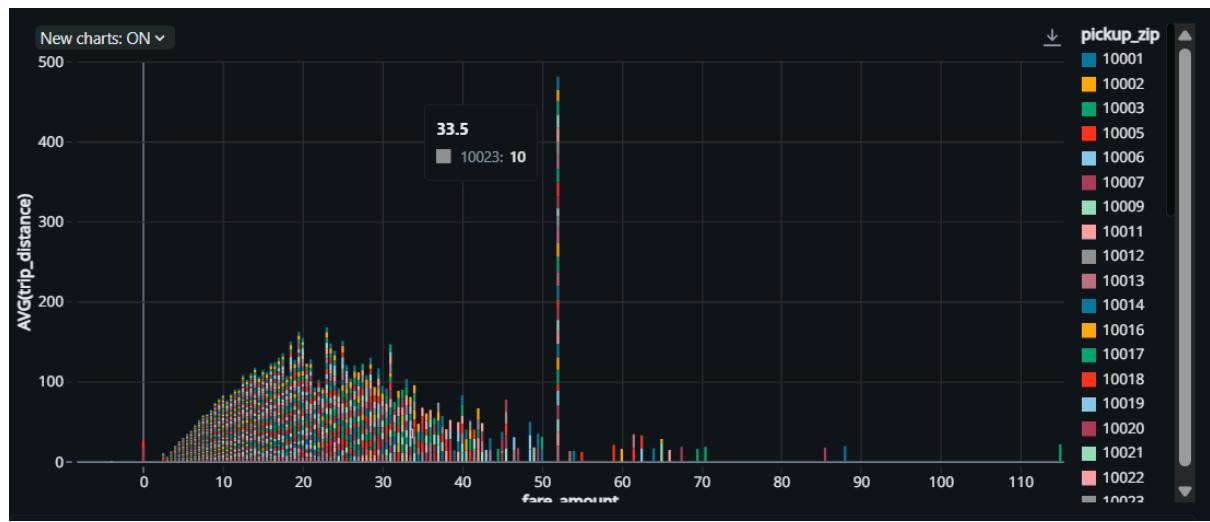
- `Visualization type : Bar`
- `X column : fare_amount`
- `Y column : trip_distance`
- ที่ `Y column` หลังจากเลือก col แล้ว จะมี `aggregation type` ให้เลือกด้วย หลังเลือกคอลัมน์นี้ไป แล้ว เดิมจะมี type เป็น `Sum` ให้เปลี่ยนเป็น `Average` แบบนี้

A screenshot of the `X column` and `Y columns` settings in the `Visualization Editor`. The `X column` dropdown contains `fare_amount`. The `Y columns` dropdown contains `trip_distance`. To the right of the `Y columns` dropdown, there is a `Aggregation type` dropdown set to `Average`. There is also a `-` button to remove the selected column.

- `Group by : pickup_zip`

คลิก Save

(หน้าตาที่ได้) 



เอาเม้าส์ซึ่งขึ้น Tooltip ให้ดูได้ สามารถปรับเล่นกราฟอื่นๆ ได้ตามต้องการ การปรับแต่งอาจจะต้องใช้ sql มาช่วยด้วย เช่น เลือกแค่ top 5 และค่อยมา visualize



Import and visualize CSV data from a notebook

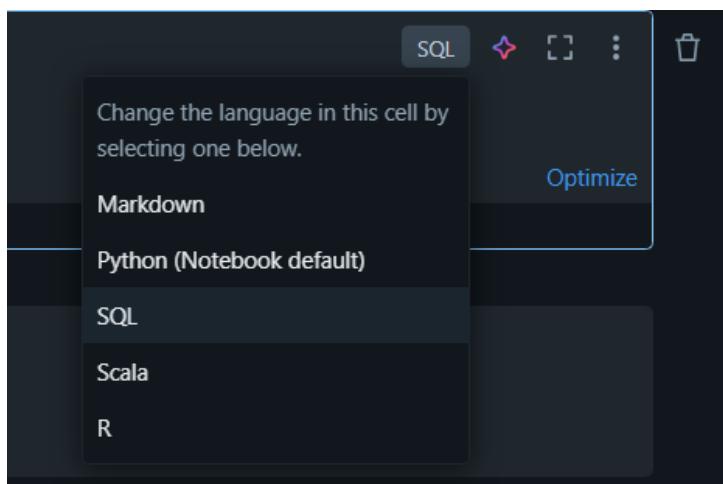
<https://docs.databricks.com/aws/en/getting-started/import-visualize-data>

ตอนนี้เรารู้วิธีสร้าง Notebook และ เราจะไปกันต่อ กับการลงนำ CSV มาเล่นดู

เพื่อความไม่ยุ่งเหยิง เราจะสร้าง Notebook ใหม่ ซึ่งวิธีการสร้าง Notebook ใหม่จะเหมือนเดิม (ใช้ Notebook เดิมได้)

ก่อนที่เราจะไปต่อ Databricks จะมี Catalog และ Schema ที่เป็น default ไว้ให้ แต่ Volume เพื่อเก็บข้อมูลจะยังไม่มี เราจะต้องสร้าง Volume เพื่อรับรับมันก่อน

ด้านขวาของ cell ให้เลือก SQL เพื่อให้ง่ายต่อการสร้าง volume



Query : `CREATE VOLUME IF NOT EXISTS <name>;`

แทน <name> ด้วยชื่อที่เราอยากรักษา ในที่นี้ สมมติคือ test1

`CREATE VOLUME IF NOT EXISTS test1;`

(ถ้าสร้างสำเร็จเรียบร้อย จะขึ้นผลลัพธ์เป็น OK)

ต่อจากนี้เราจะกลับมาใช้ python เมื่อเดิม

ตั้งค่าตัวแปรก่อน โดยใช้โค้ดนี้ 

```
catalog = "workspace"
schema = "default"
volume = "test1"

download_url = "https://health.data.ny.gov/api/views/jxy9-yhdk/rows.csv"
file_name = "baby_names.csv"
table_name = "baby_names"
```

```

path_volume = "/Volumes/" + catalog + "/" + schema + "/" + volume
path_table = catalog + "." + schema
print(path_table) # Show the complete path
print(path_volume) # Show the complete path

```

```

catalog = "workspace"
schema = "default"
volume = "test1"
download_url = "https://health.data.ny.gov/api/views/jxy9-yhdk/rows.csv"
file_name = "baby_names.csv"
table_name = "baby_names"
path_volume = "/Volumes/" + catalog + "/" + schema + "/" + volume
path_table = catalog + "." + schema
print(path_table) # Show the complete path
print(path_volume) # Show the complete path

workspace.default
/Volumes/workspace/default/test1

```

หลังจากกำหนดค่าตัวแปรและ path เรียบร้อยแล้ว ใน cell ใหม่ เราจะ import csv file จาก

<https://health.data.ny.gov/api/views/jxy9-yhdk/rows.csv>

โดยใช้ **dbutils** ([Databricks dbutils command](#))

Code : dbutils.fs.cp(f"{download_url}", f"{path_volume}" + "/" + f"{file_name}")

```

dbutils.fs.cp(f"{download_url}", f"{path_volume}" + "/" + f"{file_name}")
> See performance (1)
True

```

เมื่อนำเข้ามาแล้ว ขั้นต่อไปเป็นการ load CSV file ดังกล่าวไปสู่ DataFrame

ใน cell ใหม่

Code :

```

df = spark.read.csv(f"{path_volume}/{file_name}",
header=True,
inferSchema=True,
sep=",")

```

เมื่อรันเรียบร้อยแล้ว ลอง display df ออกมา จะได้ตัวอย่างข้อมูลดังภาพ 👇

▶ ✓ 1 minute ago (9s)

display(df)

> [See performance \(1\)](#)

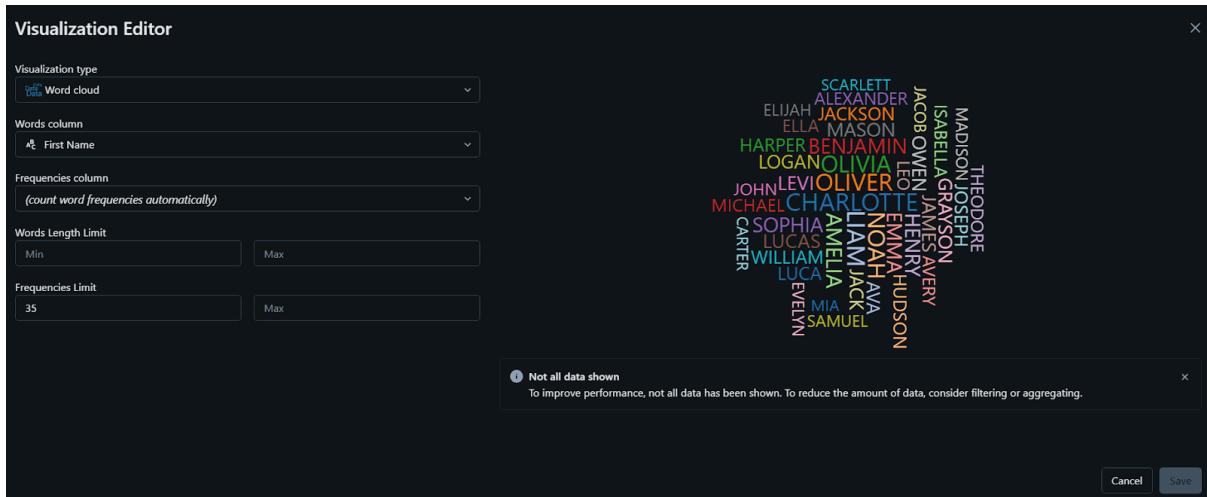
Table		Visualization 1		+	
#	Year	First Name	County	Sex	Count
1	2022	OLIVIA	Albany	F	16
2	2022	AMELIA	Albany	F	15
3	2022	AVERY	Albany	F	12
4	2022	EMMA	Albany	F	11
5	2022	CHARLOTTE	Albany	F	11
6	2022	CHLOE	Albany	F	11
7	2022	SOPHIA	Albany	F	8
8	2022	CORA	Albany	F	8
9	2022	MIA	Albany	F	7
10	2022	LUNA	Albany	F	7

มา visualize กันต่อ

(ขั้นตอน visualize เมื่อันเดิม ด้านขวาของ Table คลิก + > Visualization)

เลือก

- Visualization type : Word cloud
 - Words column : First Name
 - Frequencies limit : 35



Save

หลังจากลอง Viz เล่นดแล้ว ตอมา เราจะ Save ตัว DataFrame เปนตาราง

> แก้ชื่อคอลัมน์ก่อน เนื่องจากคอลัมน์ที่มี Space (และอักษรพิเศษ) ไม่สามารถเป็นชื่อคอลัมน์ได้ โดยใช้ WithColumnRenamed()

```
Code : df = df.withColumnRenamed("First Name", "First_Name")
        df.printSchema
```

> Save DataFrame เป็นตาราง โดยใช้ตัวแปรที่เรา defined ไปตั้งแต่ตอนต้น

Code : df.write.mode("overwrite").saveAsTable(f"{path_table}" + "." + f"{table_name}")

The screenshot shows two code execution steps in a Databricks notebook:

Step 7:

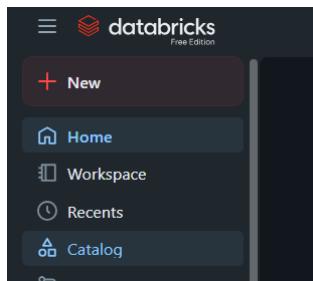
```
df = df.withColumnRenamed("First Name", "First_Name")
df.printSchema
```

Step 8:

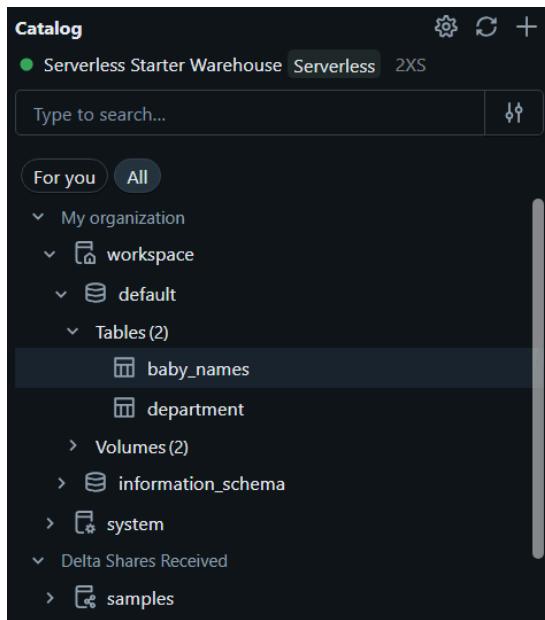
```
df.write.mode("overwrite").saveAsTable(f"{path_table}" + "." + f"{table_name}")
```

Below step 8, there is a link "See performance (1)" and an "Optimize" button.

เพื่อความชัวร์ เราจะไปเช็คตารางกันต่อ
ตรง Sidebar ให้เลือก 'Catalog'



ที่ workspace > default จะมี Tables จากที่เราสร้างไว้ปรากฏขึ้นมา ถ้าไม่มี ลองคลิก Refresh ใกล้ๆรูป
พื้นเพื่องดูก่อน (ในรูปนี้มี 2 อัน เพราะได้ลองทำ tutorials ตาม docs อื่นด้วย แต่ถ้าน้องๆเริ่มใหม่จะมีแค่
อันเดียวโผล่ขึ้นมา คือ baby_names)



คลิกตาราง baby_names และดู Overview

- เลือก Sample Data เพื่อดู 100 แถวแรกของตาราง
(ตอนเลือกดู Sample Data ครั้งแรก จะยังไม่สามารถดูได้ให้คลิกเลือก Severless Starter Warehouse ก่อน ซึ่งปุ่มให้เลือกชื่อ ‘Select compute’ อยู่ตรงกลางใต้ Ask question เลย)

The screenshot shows the Azure Data Explorer interface for the 'department' table. The 'Sample Data' tab is active. At the top, there are tabs for Overview, Sample Data, Details, Permissions, Policies, History, Lineage, Insights, and Quality. Below the tabs is a search bar with placeholder text 'Ask your question about the sample data...'. To the right of the search bar are 'Preview' and 'Reset' buttons. Below the search bar are three suggested questions: 'How many departments are in each location?', 'Are there any departments with missing deptnames?', and 'What are the most common departm'. A large cloud icon is centered on the page. A message at the bottom states 'Sample data is not available without an active SQL warehouse or cluster'. At the bottom right is a 'Select compute' button.

The screenshot shows a modal dialog titled 'Attach to an existing compute resource'. It contains a dropdown menu with 'Serverless Starter Warehouse' selected. At the bottom right is a 'Close' button.

ภาพรวมจะประมาณนี้

The screenshot shows the Catalog Explorer interface for the 'baby_names' table. On the left, there's a sidebar with 'Catalog' and a list of databases: 'Serverless Starter Warehouse' (selected), 'workspace', 'default', and 'Tables (3)'. Under 'Tables (3)', 'baby_names' is selected. The main area shows the table schema with columns: Year, First_Name, County, Sex, and Count. Below the schema is a sample data grid with 9 rows of data. The top navigation bar includes 'Open in a dashboard', 'Share', and 'Create' buttons.

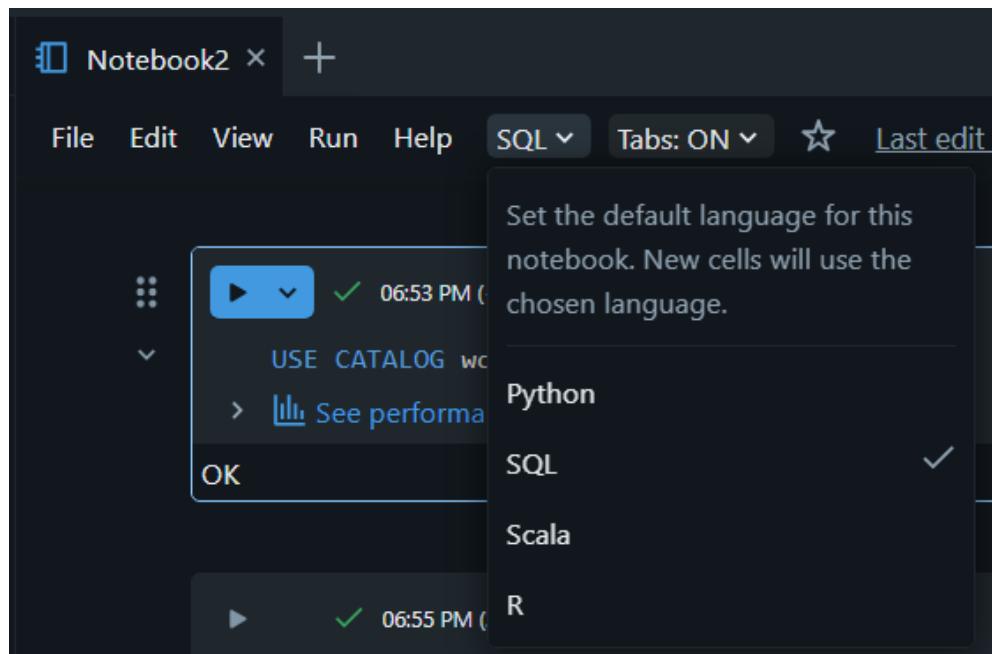
	Year	First_Name	County	Sex	Count
1	2022	OLIVIA	Albany	F	16
2	2022	AMELIA	Albany	F	15
3	2022	AVERY	Albany	F	12
4	2022	EMMA	Albany	F	11
5	2022	CHARLOTTE	Albany	F	11
6	2022	CHLOE	Albany	F	11
7	2022	SOPHIA	Albany	F	8
8	2022	CORA	Albany	F	8
9	2022	MIA	Albany	F	7

Create a table

<https://docs.databricks.com/aws/en/getting-started/create-table>

แนะนำสร้าง Notebook ใหม่ เพราะ Notebook นี้ใช้ SQL

โดยหลังจากสร้าง Notebook ใหม่แล้ว ให้ปรับตรงนี้เลย (อยู่ข้างๆ Help) ทุก cell ใหม่จะใช้ SQL หมด
ง่ายต่อการทำางานต่อในหัวข้อนี้มากกว่าปรับ cell ทีละอัน



เขียน query

- เลือก catalog : USE CATALOG workspace
- สร้างตาราง :

```
CREATE TABLE IF NOT EXISTS default.department (
    deptcode INT,
    deptname STRING,
    location STRING
);
```
- เพิ่มข้อมูล :

```
INSERT INTO default.department VALUES
(10, 'FINANCE', 'EDINBURGH'),
(20, 'SOFTWARE', 'PADDINGTON');
```

ไปดูตารางที่เพิ่มเมื่อนหัวข้อที่แล้ว โดยเลือก Catalog > workspace > default > Tables > ตาราง department

Catalog Explorer > workspace > default > department

Overview Sample Data Details Permissions Policies History Lineage Insights Quality

Description

Column Type Comment Tags Column masking

- deptcode int
- deptname string
- location string

About this table

Owner	65070213@kmit.ac.th
Type	Managed
Data source	Delta
Popularity	0.00
Last updated	2 hours ago
Size	1.1KiB, 1 file

ในฐานะเจ้าของตาราง เราสามารถให้สิทธิ์ User ในการ อ่าน เขียน ตารางของเราราได้
ทำได้ 2 วิธี

- ใช้ UI ของ Databricks : ไปที่ Permissions คลิก 'Grant' ด้านซ้าย

Catalog Explorer > workspace > default > department

Overview Sample Data Details Permissions Policies History Lineage Insights Quality

Grant Revoke Privileges Inherited Type to filter by principal

จะมีหน้าต่างนี้ขึ้นมา 👇

Grant on workspace.default.department

Principals

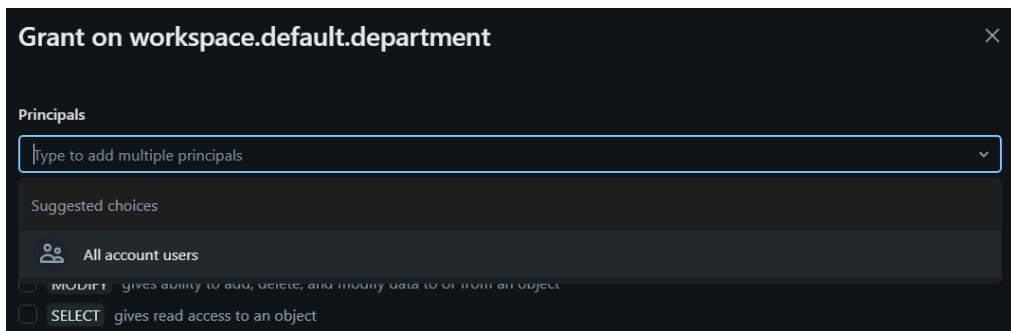
Type to add multiple principals

Privileges

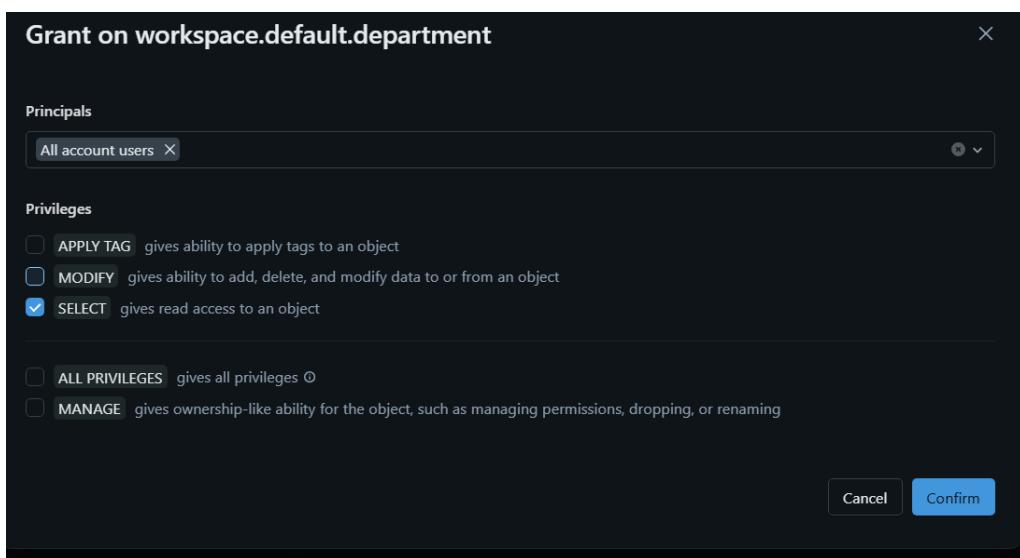
- APPLY TAG** gives ability to apply tags to an object
- MODIFY** gives ability to add, delete, and modify data to or from an object
- SELECT** gives read access to an object
- ALL PRIVILEGES** gives all privileges
- MANAGE** gives ownership-like ability for the object, such as managing permissions, dropping, or renaming

Cancel Confirm

ในที่นี่ยกตัวอย่างเป็น All account users



ให้สิทธิเป็น SELECT (read) > Confirm



- ใช้ SQL :

Query ในการให้สิทธิ์ด้วย SQL

GRANT SELECT ON default.department TO `data-consumers`;

ตามตัวอย่างใช้ `data-consumers` ในที่นี่เราจะให้สิทธิ์ทุกคน ให้ใช้เป็น `account users`

GRANT SELECT ON default.department TO `account users`;

ระวัง : เครื่องหมาย ` (Backtick) ไม่ใช่เครื่องหมายคำพูด