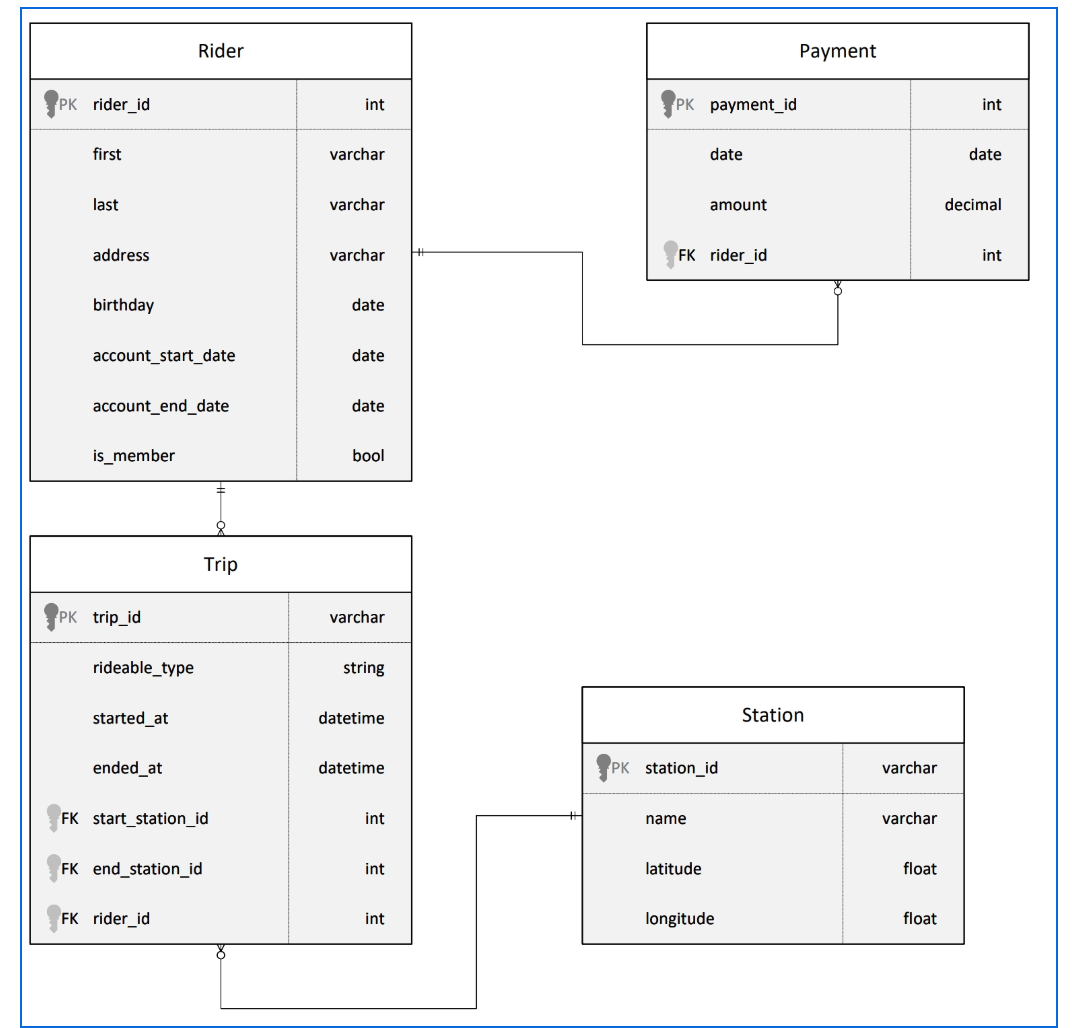
**Building Azure Data Lake for Bike Share Data**

# ****Project****

This project called "Building an Azure Data Lake for Bike Share Data Analytics" is an assignment from the UDACITY course "Data Engineering with Microsoft Azure".

# ****Project Overview****

In this project, you'll build a data lake solution for Divvy bikeshare. Divvy is a bike sharing program in Chicago, Illinois USA that allows riders to purchase a pass at a kiosk or use a mobile application to unlock a bike at stations around the city and use the bike for a specified amount of time. The bikes can be returned to the same station or to another station. The City of Chicago makes the anonymized bike trip data publicly available for projects like this where we can analyze the data. Since the data from Divvy are anonymous, we have generated fake rider and account profiles along with fake payment data to go along with the data from Divvy. The dataset looks like this:



**Objective**

The goal of this project is to develop a data lake solution using Azure Databricks using a lake house architecture. The project involve the following steps:

* Design a star schema based on the business outcomes listed below;
* Import the data into Azure Databricks using Delta Lake to create a Bronze data store;
* Create a gold data store in Delta Lake tables;
* Transform the data into the star schema for a Gold data store.

**Business outcomes**

1. Analyze how much time is spent per ride

* Based on date and time factors such as day of week and time of day
* Based on which station is the starting and / or ending station
* Based on age of the rider at time of the ride
* Based on whether the rider is a member or a casual rider

1. Analyze how much money is spent

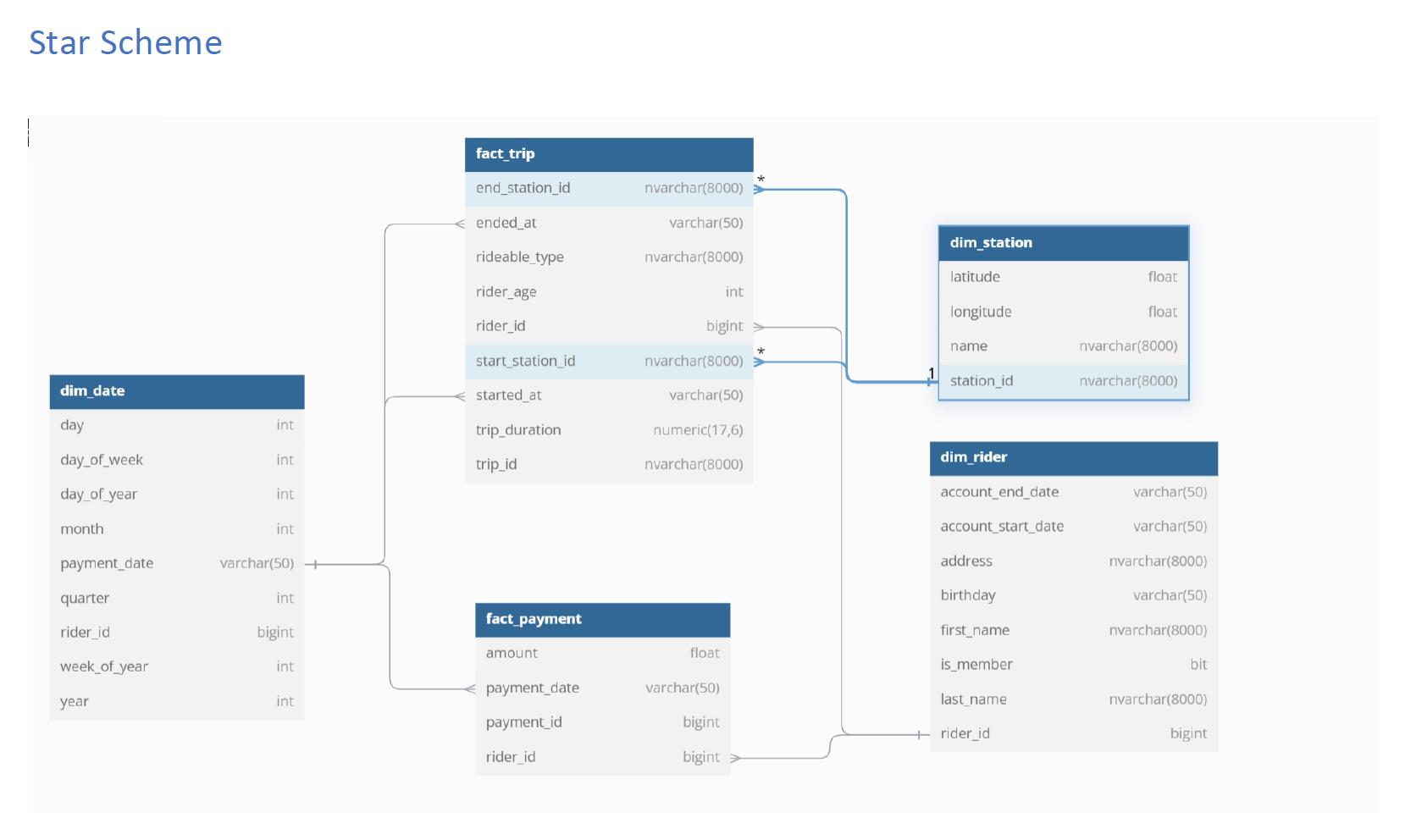
* Per month, quarter, year
* Per member, based on the age of the rider at account start

1. EXTRA CREDIT - Analyze how much money is spent per member

* Based on how many rides the rider averages per month
* Based on how many minutes the rider spends on a bike per month

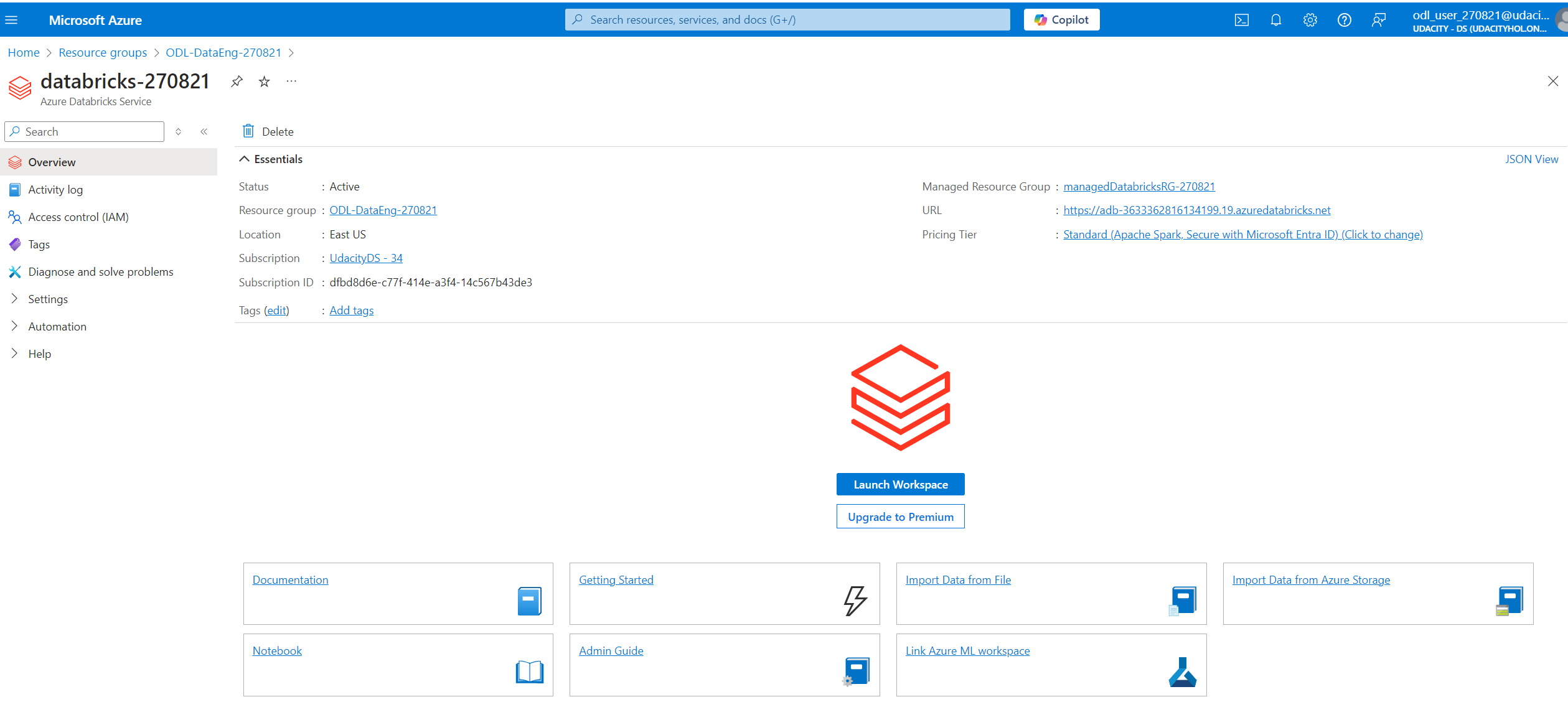
# TASK NEED TO COMPLETE

## Task 1: Design Bikeshare Star Schema. Below is our star schema:

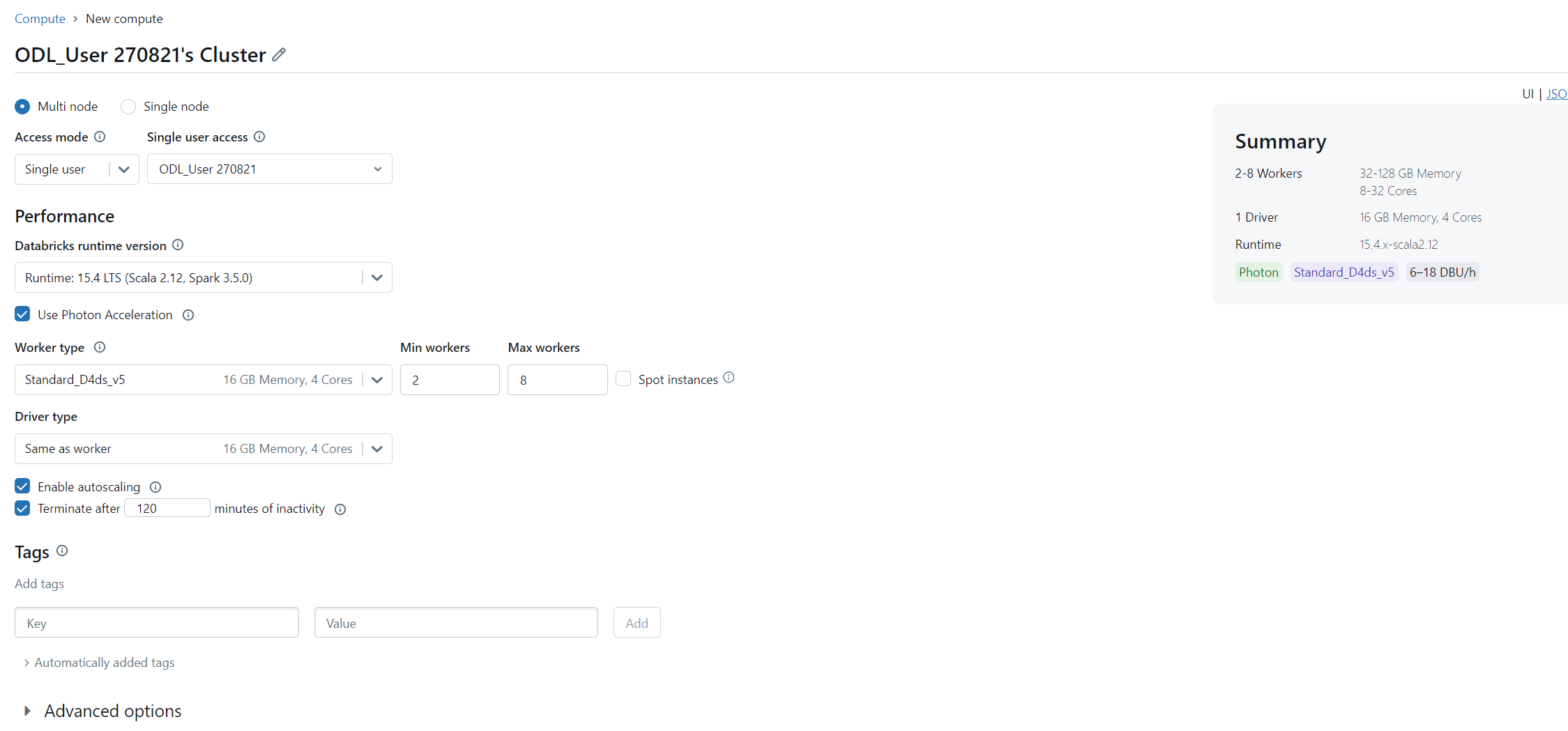


## Task 2: Import the data into Azure Databricks using Delta Lake to create a Bronze data store

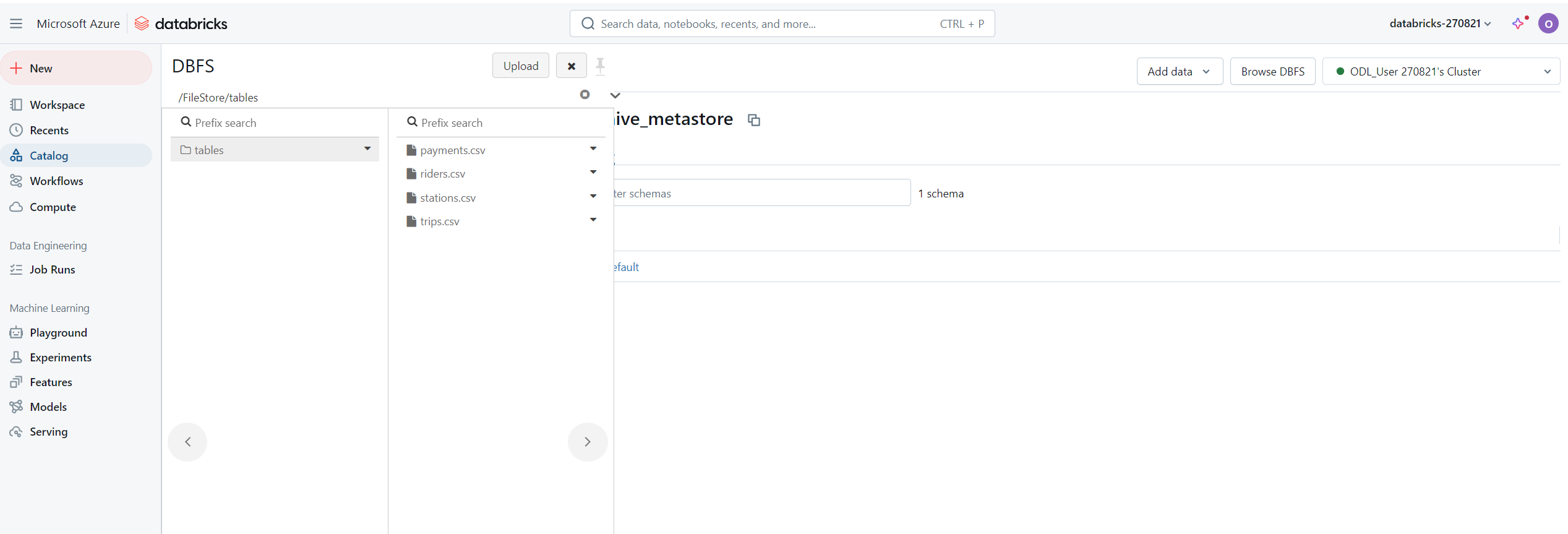
1. Create Azure Databricks Workspace (use the resource group already created



1. Create Databricks Cluster



1. Import data to DBFS path **/FileStore/tables/**



3.2 Extract: Load data into Databricks as dataframes and Write data into Bronze data lake

Load CSV files

payments\_df = spark.read.format("csv") \

    .option("inferSchema", "true") \

    .option("header", "false") \

    .option("sep", ",") \

    .load("dbfs:/FileStore/tables/payments.csv") \

    .toDF("payment\_id","payment\_date","amount","rider\_id")

# riders extraction

riders\_df = spark.read.format("csv") \

    .option("inferSchema", "true") \

    .option("header", "false") \

    .option("sep", ",") \

    .load("dbfs:/FileStore/tables/riders.csv") \

    .toDF("rider\_id", "first\_name", "last\_name", "address", "birthday", "account\_start\_date", "account\_end\_date", "is\_member")

# stations extraction

stations\_df = spark.read.format("csv") \

    .option("inferSchema", "true") \

    .option("header", "false") \

    .option("sep", ",") \

    .load("dbfs:/FileStore/tables/stations.csv") \

    .toDF("station\_id", "name", "latitude", "longitude")

# stations extraction

stations\_df = spark.read.format("csv") \

    .option("inferSchema", "true") \

    .option("header", "false") \

    .option("sep", ",") \

    .load("dbfs:/FileStore/tables/stations.csv") \

    .toDF("station\_id", "name", "latitude", "longitude")

# trips extraction

trips\_df = spark.read.format("csv") \

    .option("inferSchema", "true") \

    .option("header", "false") \

    .option("sep", ",") \

    .load("dbfs:/FileStore/tables/trips.csv") \

    .toDF("trip\_id", "rideable\_type", "start\_at", "ended\_at", "start\_station\_id", "end\_station\_id", "rider\_id")

Write into Dataframes Delta Lakes tables

# Write data to delta location

payments\_df.write.format("delta") \

    .mode("overwrite") \

    .save("/delta/payments")

# Write data to delta location

riders\_df.write.format("delta") \

    .mode("overwrite") \

    .save("/delta/riders")

# Write data to delta location

stations\_df.write.format("delta") \

    .mode("overwrite") \

    .save("/delta/stations")

# Write data to delta location

trips\_df.write.format("delta") \

    .mode("overwrite") \

    .save("/delta/trips")

Write Datagrams into bronze data lake

# write dataframes into bronze data lake

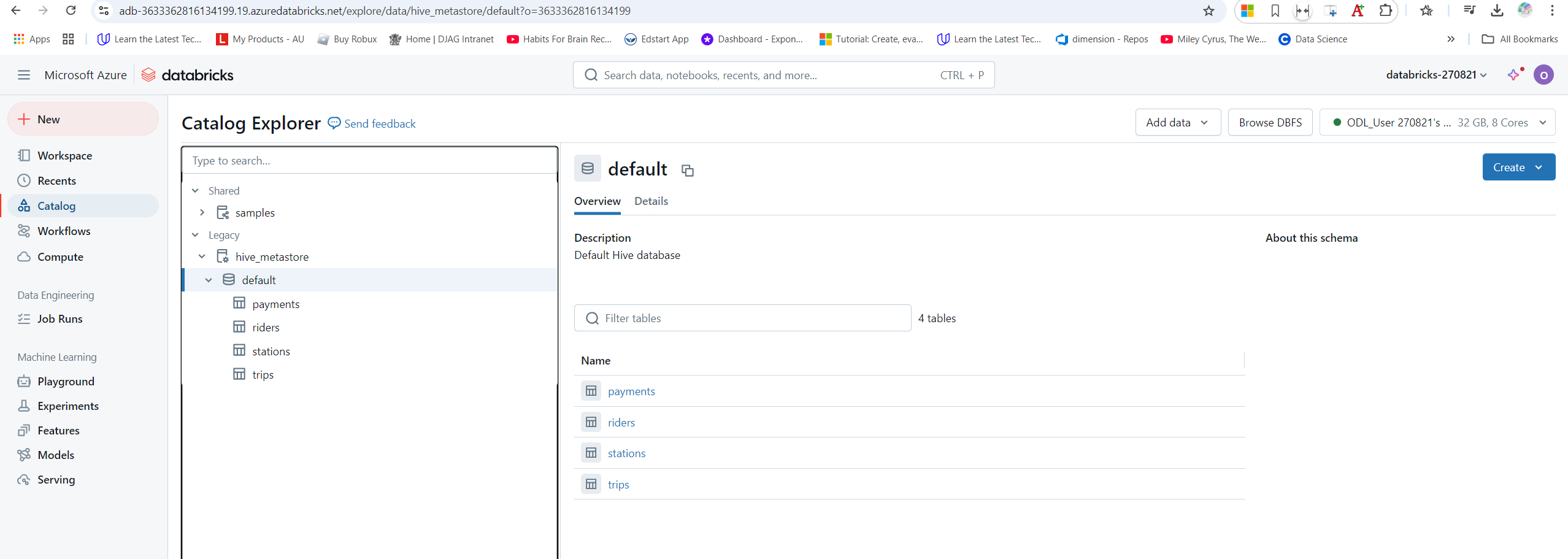
delta\_path = "/bronze"

df\_payments.write.format("delta").mode("overwrite").save(f"{delta\_path}/payments")

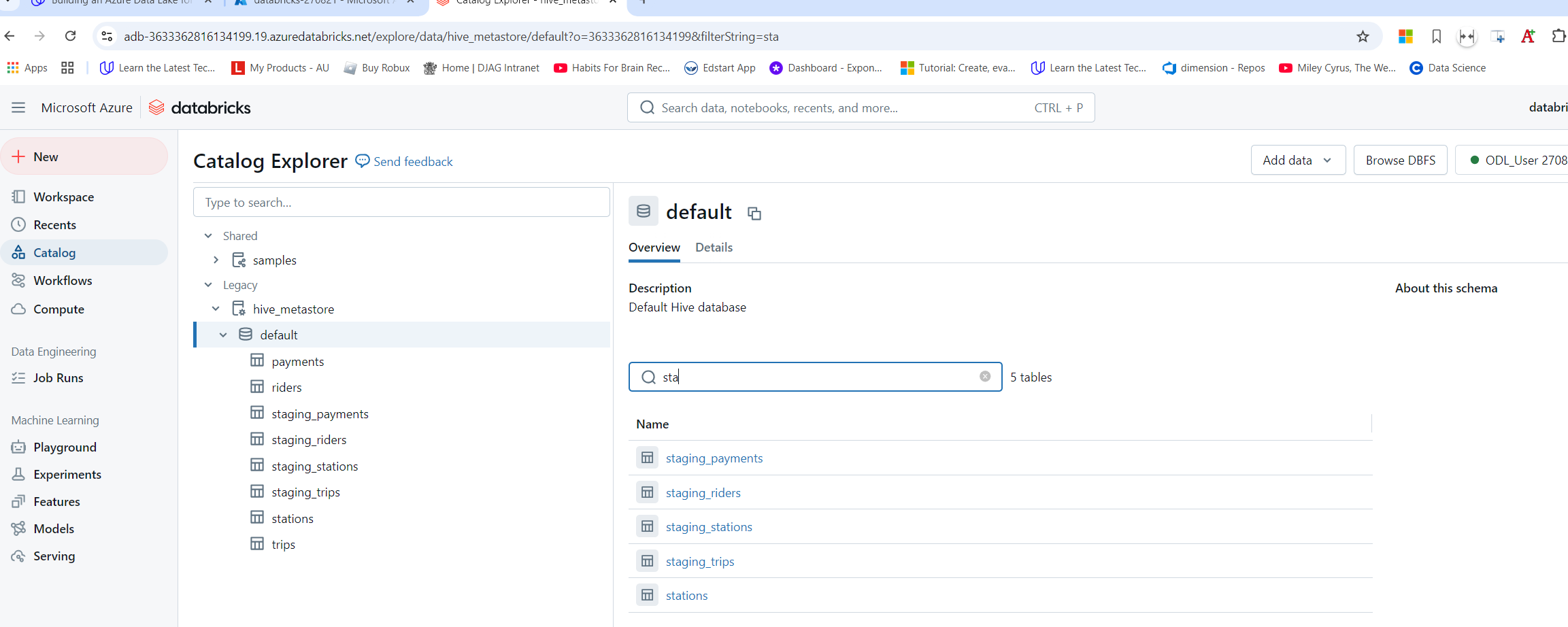
df\_riders.write.format("delta").mode("overwrite").save(f"{delta\_path}/riders")

df\_stations.write.format("delta").mode("overwrite").save(f"{delta\_path}/stations")

df\_trips.write.format("delta").mode("overwrite").save(f"{delta\_path}/trips")



1. Load Spark SQL from Delta tables



1. Transform Delta source tables into designed star schema (dim/fact tables)

