



**Ads Performance**

Write an SQL query to find the ctr of each Ad.Round ctr to 2 decimal points. Order the result table by ctr in descending order and by ad\_id in ascending order in case of a tie.

Ctr=Clicked**/(**Clicked**+**Viewed**)**

**Difficult Level :** EASY

**DataFrame:**

**# Define the schema for the Ads table**

**schema=StructType([**

**StructField('AD\_ID',IntegerType(),True)**

**,StructField('USER\_ID',IntegerType(),True)**

**,StructField('ACTION',StringType(),True)**

**])**

**# Define the data for the Ads table**

**data = [**

**(1, 1, 'Clicked'),**

**(2, 2, 'Clicked'),**

**(3, 3, 'Viewed'),**

**(5, 5, 'Ignored'),**

**(1, 7, 'Ignored'),**

**(2, 7, 'Viewed'),**

**(3, 5, 'Clicked'),**

**(1, 4, 'Viewed'),**

**(2, 11, 'Viewed'),**

**(1, 2, 'Clicked')**

**]**



**INPUT**

| **INPUT** | | |
| --- | --- | --- |
| **AD\_ID** | **USER\_ID** | **ACTION** |
| **1** | **1** | **Clicked** |
| **2** | **2** | **Clicked** |
| **3** | **3** | **Viewed** |
| **5** | **5** | **Ignored** |
| **1** | **7** | **Ignored** |
| **2** | **7** | **Viewed** |
| **3** | **5** | **Clicked** |
| **1** | **4** | **Viewed** |
| **2** | **11** | **Viewed** |
| **1** | **2** | **Clicked** |

**OUTPUT**

| **OUTPUT** | |
| --- | --- |
| **AD\_ID** | **CTR** |
| **1** | **0.67** |
| **3** | **0.5** |
| **2** | **0.33** |
| **5** | **0** |



**# Creating Spark Session**

**from pyspark.sql import SparkSession**

**from pyspark.sql.types import StructType,StructField,IntegerType,StringType**

**from pyspark.sql.functions import when**

**from pyspark.sql import functions as F**

**from pyspark.sql.window import Window**

**#creating spark session**

**spark = SparkSession. \**

**builder. \**

**config('spark.shuffle.useOldFetchProtocol', 'true'). \**

**config('spark.ui.port','0'). \**

**config("spark.sql.warehouse.dir", "/user/itv008042/warehouse"). \**

**enableHiveSupport(). \**

**master('yarn'). \**

**getOrCreate()**

**# Define the schema for the Ads table**

**schema=StructType([**

**StructField('AD\_ID',IntegerType(),True)**

**,StructField('USER\_ID',IntegerType(),True)**

**,StructField('ACTION',StringType(),True)**

**])**

**# Define the data for the Ads table**

**data = [**

**(1, 1, 'Clicked'),**

**(2, 2, 'Clicked'),**

**(3, 3, 'Viewed'),**

**(5, 5, 'Ignored'),**

**(1, 7, 'Ignored'),**

**(2, 7, 'Viewed'),**

**(3, 5, 'Clicked'),**

**(1, 4, 'Viewed'),**

**(2, 11, 'Viewed'),**

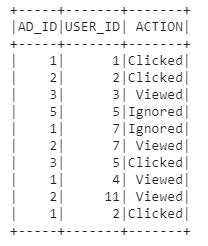
**(1, 2, 'Clicked')**

**]**

**# Create a PySpark DataFrame**

**df=spark.createDataFrame(data,schema)**

**df.show()**

****

**ctr\_df = (**

**ads\_df.groupBy("ad\_id")**

**.agg(**

**F.sum(F.when(ads\_df["action"] == "Clicked", 1).otherwise(0)).alias("click\_count"),**

**F.sum(F.when(ads\_df["action"] == "Viewed", 1).otherwise(0)).alias("view\_count")**

**)**

**.withColumn("ctr", F.round(F.col("click\_count") / (F.col("click\_count") + F.col("view\_count")), 2))**

**)**

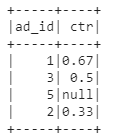
**# Order the result table by CTR in descending order and by ad\_id in ascending order**

**window\_spec = Window.orderBy(F.col("ctr").desc(), F.col("ad\_id").asc())**

**result\_df = ctr\_df.withColumn("rank", F.rank().over(window\_spec))**

**# Show the result DataFrame**

**result\_df.select('ad\_id','ctr').show()**

****

