



**12\_Cities With Completed Trades**

Write a pypsark code to retrieve the top three cities that have the highest number of completed trade orders listed in descending order. Output the city name and the corresponding number of completed trade orders..

**Difficult Level :** EASY

**DataFrame:**

**# Define the schema for the trades**

**trades\_schema = StructType([**

**StructField("order\_id", IntegerType(), True),**

**StructField("user\_id", IntegerType(), True),**

**StructField("price", FloatType(), True),**

**StructField("quantity", IntegerType(), True),**

**StructField("status", StringType(), True),**

**StructField("timestamp", StringType(), True)**

**])**

**# Define the schema for the users**

**users\_schema = StructType([**

**StructField("user\_id", IntegerType(), True),**

**StructField("city", StringType(), True),**

**StructField("email", StringType(), True),**

**StructField("signup\_date", StringType(), True)**

**])**

**# Create an RDD with the data for trades**

**trades\_data = [**

**(100101, 111, 9.80, 10, 'Cancelled', '2022-08-17 12:00:00'),**

**(100102, 111, 10.00, 10, 'Completed', '2022-08-17 12:00:00'),**

**(100259, 148, 5.10, 35, 'Completed', '2022-08-25 12:00:00'),**

**(100264, 148, 4.80, 40, 'Completed', '2022-08-26 12:00:00'),**

**(100305, 300, 10.00, 15, 'Completed', '2022-09-05 12:00:00'),**

**(100400, 178, 9.90, 15, 'Completed', '2022-09-09 12:00:00'),**

**(100565, 265, 25.60, 5, 'Completed', '2022-12-19 12:00:00')**

**]**

| **# Create an RDD with the data for users**  **users\_data = [**  **(111, 'San Francisco', 'rrok10@gmail.com', '2021-08-03 12:00:00'),**  **(148, 'Boston', 'sailor9820@gmail.com', '2021-08-20 12:00:00'),**  **(178, 'San Francisco', 'harrypotterfan182@gmail.com', '2022-01-05 12:00:00'),**  **(265, 'Denver', 'shadower\_@hotmail.com', '2022-02-26 12:00:00'),**  **(300, 'San Francisco', 'houstoncowboy1122@hotmail.com', '2022-06-30 12:00:00')**  **]** |
| --- |

|  |
| --- |



**INPUT**

| **INPUT-1** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ORDER\_ID** | **USER\_ID** | **PRICE** | **QUANTITY** | **STATUS** | **TIMESTAMP** |
| 100101 | 111 | 9.8 | 10 | Cancelled | 2022-08-17 12:00:00 |
| 100102 | 111 | 10 | 10 | Completed | 2022-08-17 12:00:00 |
| 100259 | 148 | 5.1 | 35 | Completed | 2022-08-25 12:00:00 |
| 100264 | 148 | 4.8 | 40 | Completed | 2022-08-26 12:00:00 |
| 100305 | 300 | 10 | 15 | Completed | 2022-09-05 12:00:00 |
| 100400 | 178 | 9.9 | 15 | Completed | 2022-09-09 12:00:00 |
| 100565 | 265 | 25.6 | 5 | Completed | 2022-12-19 12:00:00 |

| **INPUT - 2** | | | |
| --- | --- | --- | --- |
| **USER\_ID** | **CITY** | **EMAIL** | **SIGNUP\_DATE** |
| 111 | San Francisco | rrok10@gmail.com | 2021-08-03 12:00:00 |
| 148 | Boston | sailor9820@gmail.com | 2021-08-20 12:00:00 |
| 178 | San Francisco | harrypotterfan182@gmail.com | 2022-01-05 12:00:00 |
| 265 | Denver | shadower\_@hotmail.com | 2022-02-26 12:00:00 |
| 300 | San Francisco | houstoncowboy1122@hotmail.com | 2022-06-30 12:00:00 |

**OUTPUT**

| **OUTPUT** | |
| --- | --- |
| **CITY** | **COUNT()** |
| San Francisco | 3 |
| Boston | 2 |
| Denver | 1 |



**# Creating Spark Session**

**from pyspark.sql import SparkSession**

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

**#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 trades**

**trades\_schema = StructType([**

**StructField("order\_id", IntegerType(), True),**

**StructField("user\_id", IntegerType(), True),**

**StructField("price", FloatType(), True),**

**StructField("quantity", IntegerType(), True),**

**StructField("status", StringType(), True),**

**StructField("timestamp", StringType(), True)**

**])**

**# Define the schema for the users**

**users\_schema = StructType([**

**StructField("user\_id", IntegerType(), True),**

**StructField("city", StringType(), True),**

**StructField("email", StringType(), True),**

**StructField("signup\_date", StringType(), True)**

**])**

**# Create an RDD with the data for trades**

**trades\_data = [**

**(100101, 111, 9.80, 10, 'Cancelled', '2022-08-17 12:00:00'),**

**(100102, 111, 10.00, 10, 'Completed', '2022-08-17 12:00:00'),**

**(100259, 148, 5.10, 35, 'Completed', '2022-08-25 12:00:00'),**

**(100264, 148, 4.80, 40, 'Completed', '2022-08-26 12:00:00'),**

**(100305, 300, 10.00, 15, 'Completed', '2022-09-05 12:00:00'),**

**(100400, 178, 9.90, 15, 'Completed', '2022-09-09 12:00:00'),**

**(100565, 265, 25.60, 5, 'Completed', '2022-12-19 12:00:00')**

**]**

**# Create an RDD with the data for users**

**users\_data = [**

**(111, 'San Francisco', 'rrok10@gmail.com', '2021-08-03 12:00:00'),**

**(148, 'Boston', 'sailor9820@gmail.com', '2021-08-20 12:00:00'),**

**(178, 'San Francisco', 'harrypotterfan182@gmail.com', '2022-01-05 12:00:00'),**

**(265, 'Denver', 'shadower\_@hotmail.com', '2022-02-26 12:00:00'),**

**(300, 'San Francisco', 'houstoncowboy1122@hotmail.com', '2022-06-30 12:00:00')**

**]**

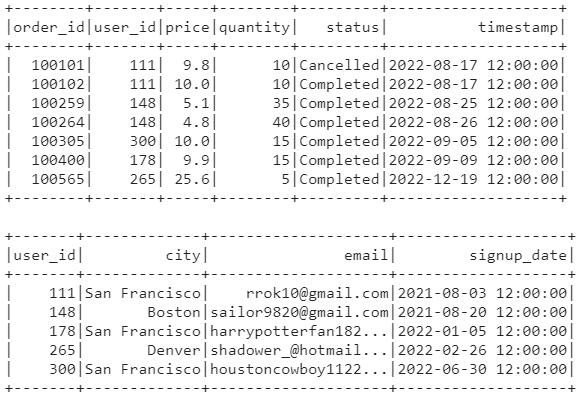
**Trade\_df=spark.createDataFrame(trades\_data,trades\_schema)**

**User\_df=spark.createDataFrame(users\_data,users\_schema)**

**Trade\_df.show()**

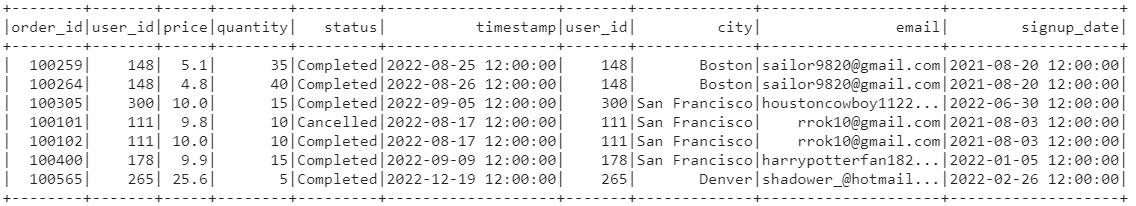
**User\_df.show()**

|  |
| --- |

****

**join\_df=Trade\_df.join(User\_df,Trade\_df['user\_id']==User\_df['user\_id'],"inner")**

**join\_df.show()**

****

**join\_df.filter(join\_df['status'] == 'Completed').groupby(join\_df['city']).count()**

