


PySpark
Learning Hub | Practice Problem



Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

Step - 1 : Problem Statement

12_Cities With Completed Trades

Write a pyspark 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')
]
```

PYSPARK LEARNING HUB : DAY - 12

```
# 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')
]
```



PYSPARK LEARNING HUB : DAY - 12

Step - 2 : Identifying The Input Data And Expected

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 |

PYSPARK LEARNING HUB : DAY - 12

| | |
|--------|---|
| Boston | 2 |
| Denver | 1 |

Step - 3 : Writing the pyspark code to solve

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),
```

PYSPARK LEARNING HUB : DAY - 12

```
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()
```

PYSPARK LEARNING HUB : DAY - 12

```
+-----+-----+-----+-----+-----+-----+
|order_id|user_id|price|quantity|    status|          timestamp|
+-----+-----+-----+-----+-----+-----+
|  100101|   111|  9.8|      10|Cancelled|2022-08-17 12:00:00|
|  100102|   111| 10.0|      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| 10.0|      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|
+-----+-----+-----+-----+-----+-----+
```

```
+-----+-----+-----+-----+-----+-----+
|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...|2022-01-05 12:00:00|
|   265|      Denver|shadower_@hotmail...|2022-02-26 12:00:00|
|   300|San Francisco|houstoncowboy1122...|2022-06-30 12:00:00|
+-----+-----+-----+-----+-----+-----+
```

```
join_df=Trade_df.join(User_df,Trade_df['user_id']==User_df['user_id'],
                        "inner")
join_df.show()
```

PYSPARK LEARNING HUB : DAY - 12

| order_id | user_id | price | quantity | status | timestamp | user_id | city | email | signup_date |
|----------|---------|-------|----------|-----------|---------------------|---------|---------------|----------------------|---------------------|
| 100259 | 148 | 5.1 | 35 | Completed | 2022-08-25 12:00:00 | 148 | Boston | sailor9820@gmail.com | 2021-08-20 12:00:00 |
| 100264 | 148 | 4.8 | 40 | Completed | 2022-08-26 12:00:00 | 148 | Boston | sailor9820@gmail.com | 2021-08-20 12:00:00 |
| 100305 | 300 | 10.0 | 15 | Completed | 2022-09-05 12:00:00 | 300 | San Francisco | houstoncowboy1122... | 2022-06-30 12:00:00 |
| 100101 | 111 | 9.8 | 10 | Cancelled | 2022-08-17 12:00:00 | 111 | San Francisco | rrrok10@gmail.com | 2021-08-03 12:00:00 |
| 100102 | 111 | 10.0 | 10 | Completed | 2022-08-17 12:00:00 | 111 | San Francisco | rrrok10@gmail.com | 2021-08-03 12:00:00 |
| 100400 | 178 | 9.9 | 15 | Completed | 2022-09-09 12:00:00 | 178 | San Francisco | harrypotterfan182... | 2022-01-05 12:00:00 |
| 100565 | 265 | 25.6 | 5 | Completed | 2022-12-19 12:00:00 | 265 | Denver | shadower_@hotmail... | 2022-02-26 12:00:00 |

```
join_df.filter(join_df['status'] ==  
'Completed').groupby(join_df['city']).count()
```

```
: join_df.filter(join_df['status'] == 'Completed').groupby(join_df['city']).count()
```

```
:      city  count
```

```
San Francisco    3
```

```
Denver           1
```

```
Boston           2
```




Save

**Was it
helpful?**
follow for more!



Akash Mahindrakar

Data Engineer

akashsjce8050@gmail.com



Comment

**SHARE YOUR THOUGHTS
IN COMMENT BELOW**



Share