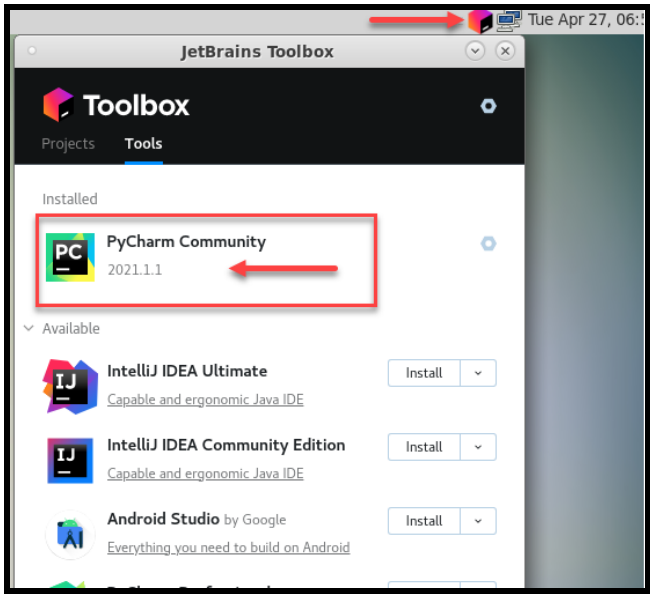
Lab: Implement Broadcast Join

Introduction

This exercise would help you to understand about the internals of the Broadcast Join in Spark. Let's explore it together.

Let's get Started

Run Pycharm using below command



In the existing **JoinDemo** project. Apply the broadcast function to the right side dataframe

```
join_expr = flight_time_df1.id == flight_time_df2.id
join_df = flight_time_df1.join(broadcast(flight_time_df2), join_expr, "inner")
```

Note: Giving a hint to spark that this dataframe is small enough, and you should consider broadcasting this table.

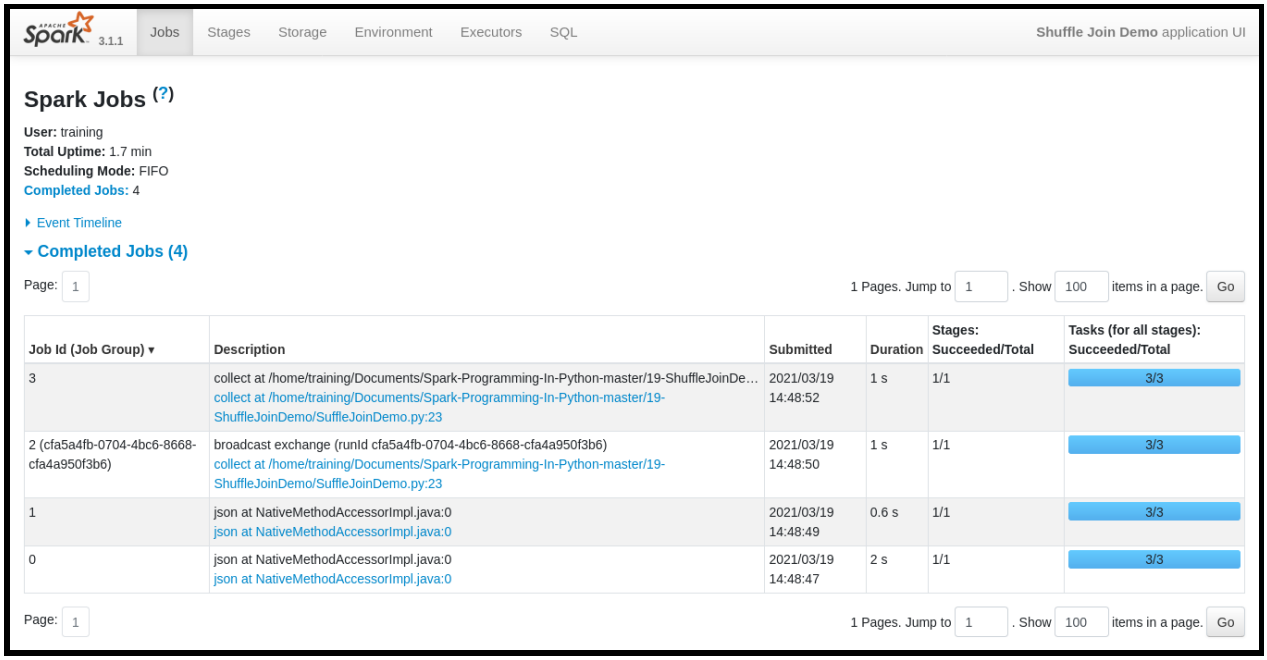
```
from pyspark.sql.functions import broadcast
```

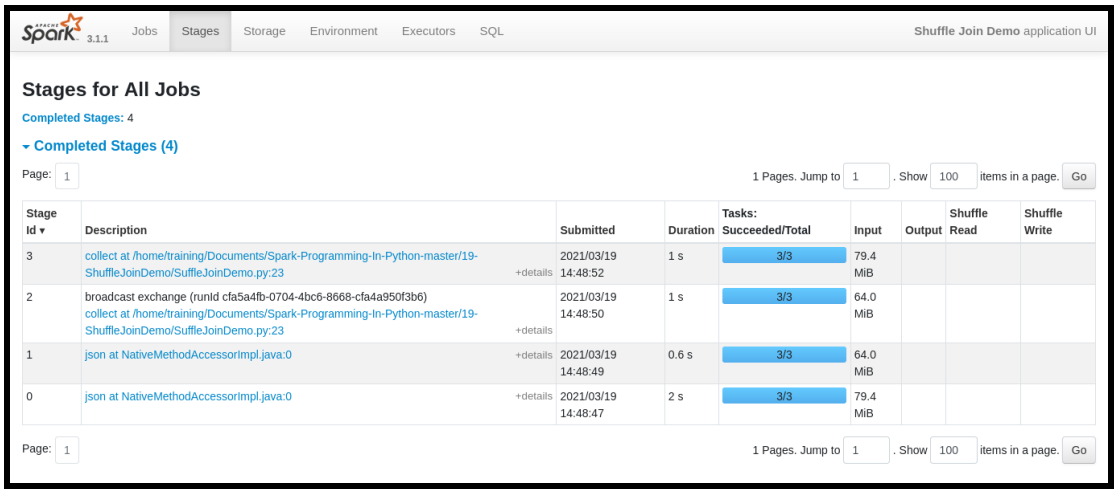
JoinDemo.py

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import broadcast

from lib.logger import Log4j

if __name__ == "__main__":
    spark = SparkSession \
        .builder \
        .appName("Join Demo") \
        .master("local[3]") \
        .getOrCreate()

    logger = Log4j(spark)

    flight_time_df1 = spark.read.json("data/d1/")
    flight_time_df2 = spark.read.json("data/d2/")

    spark.conf.set("spark.sql.shuffle.partitions", 3)

    join_expr = flight_time_df1.id == flight_time_df2.id
    join_df = flight_time_df1.join(broadcast(flight_time_df2), join_expr, "inner")
```

Now, Run the program:

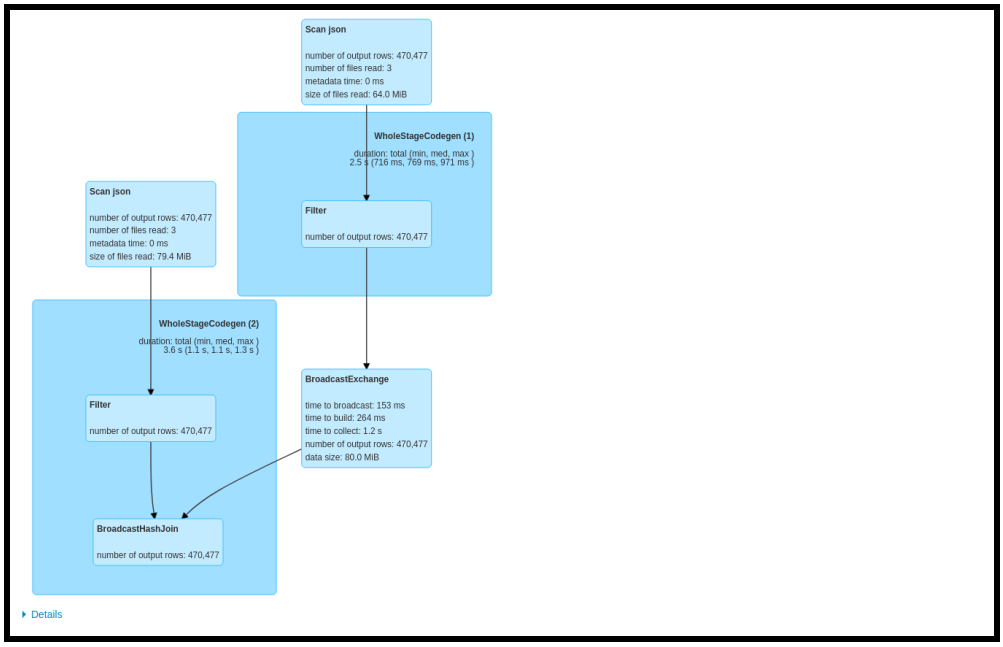
Let's go to the **Spark UI** by visiting the http://localhost:4040



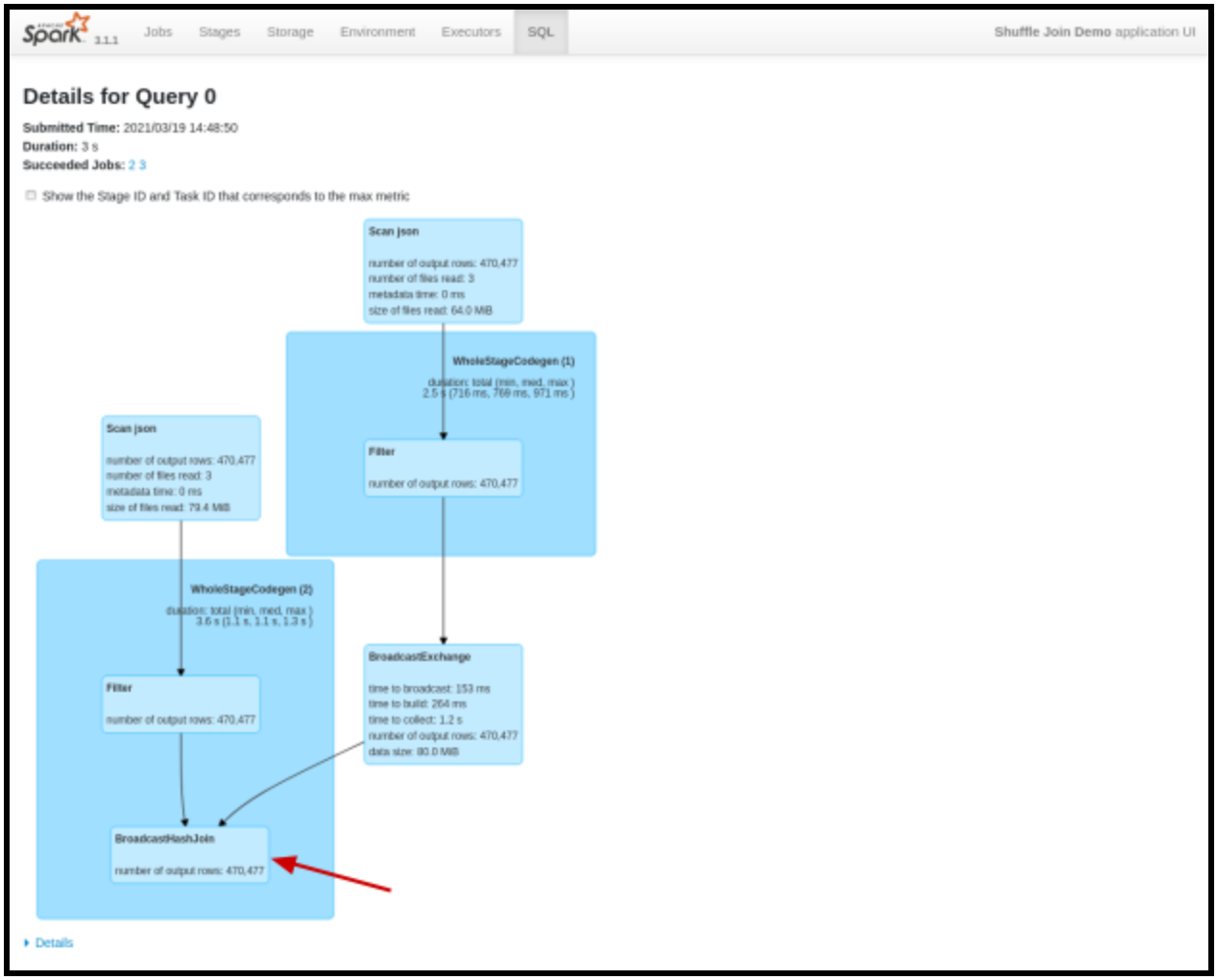


Move to the **SQL tab**

You see the broadcast exchange



And here you can see that, we have a broadcast hash join.



Great! You have learned about the broadcast join and how to implement it.

Voila!! We have successfully completed this exercise.