SHIVRAJ ROMAN

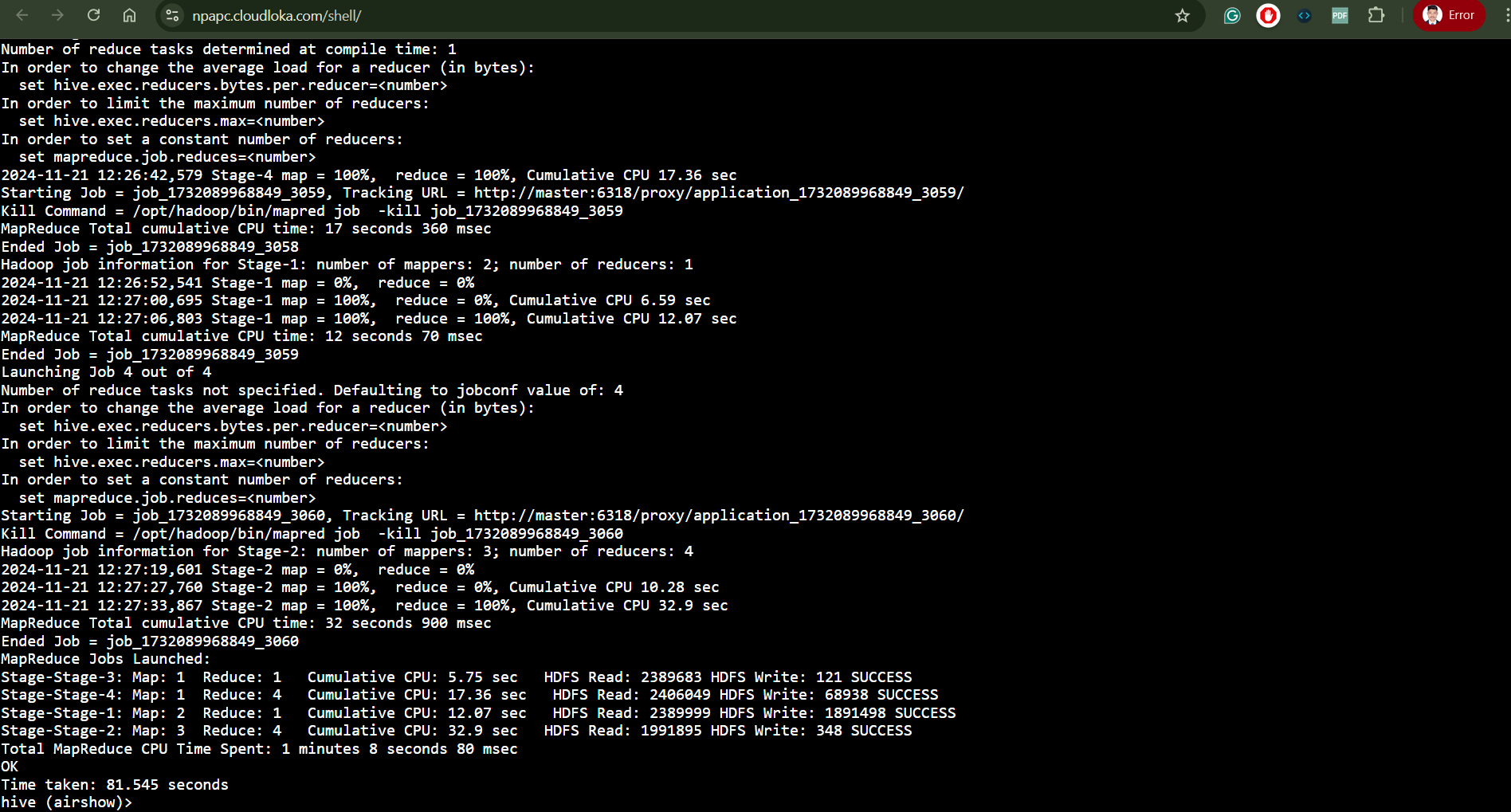
240840325056

—------------------------------------------------------------------------------------------------------------------------------ **HIVE**

Q1)

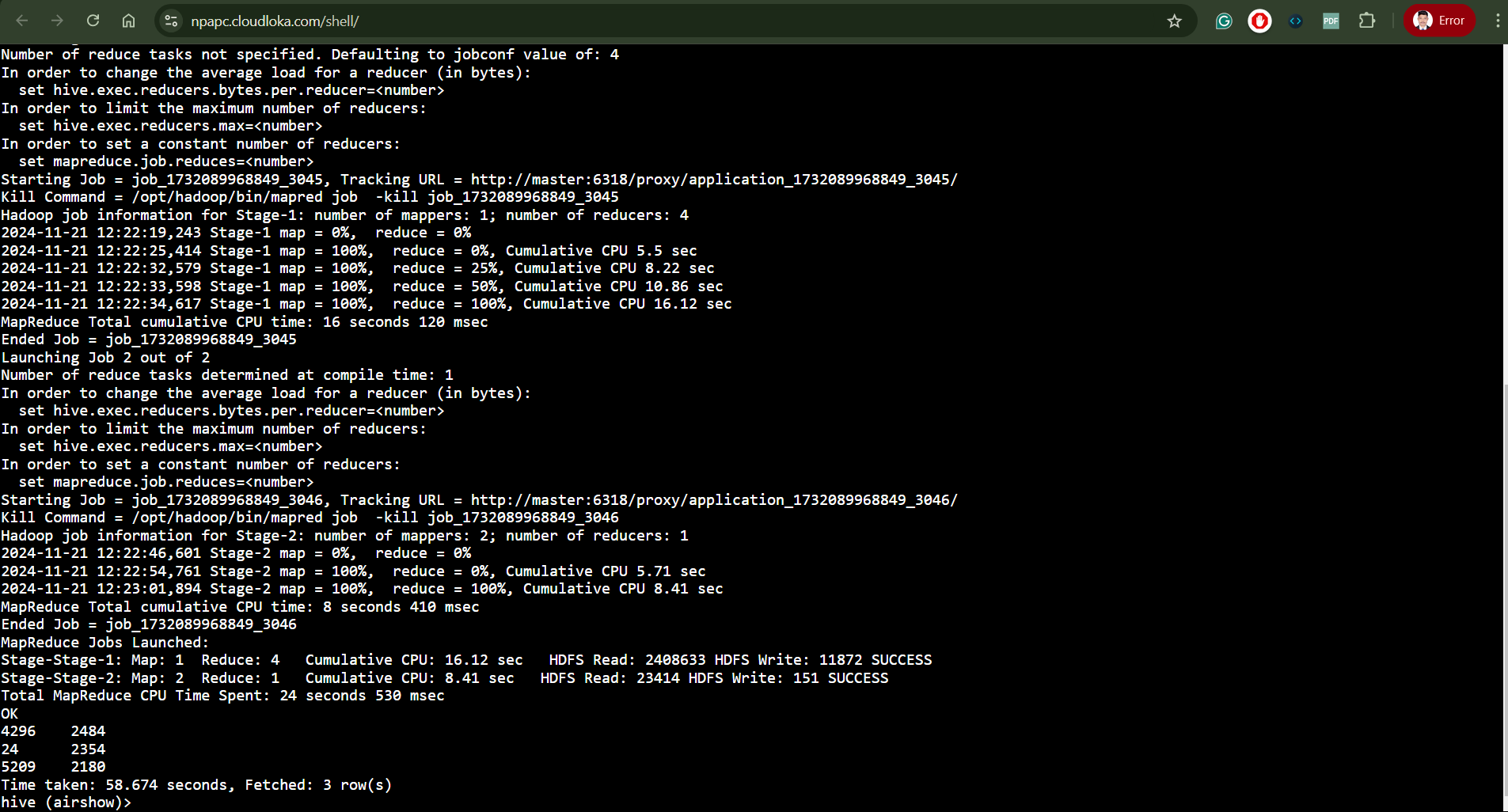
1. Find the airports that are only listed as sources but not as destinations in the routes table.

select src\_airport\_id from routes where src\_airport\_id not in (select dest\_airport\_id from routes) limit 5;



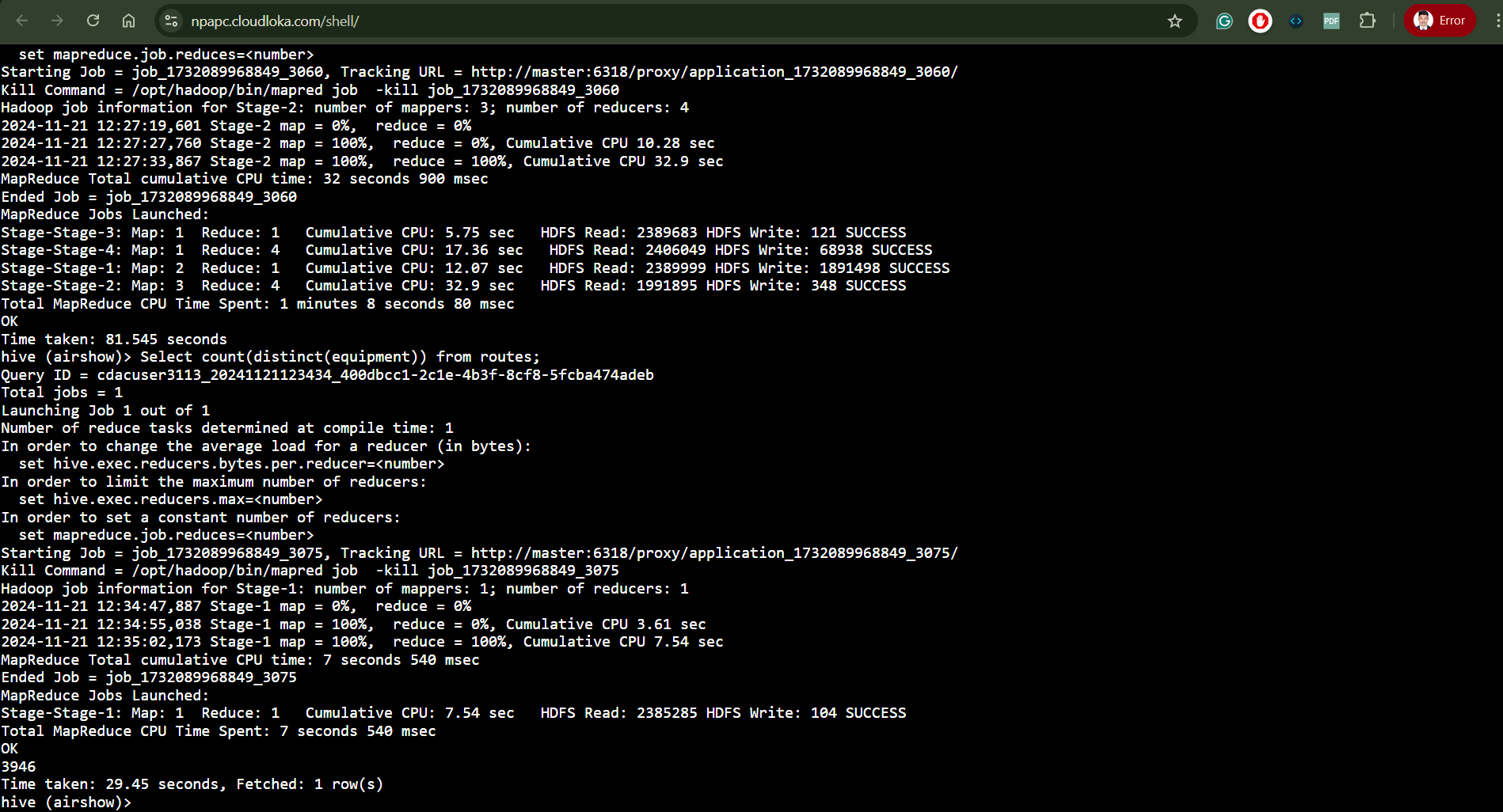
Q 2. Top 3 airlines that operates on the highest number of distinct routes.

Query - select airline\_id, count(\*) from routes group by airline\_id order by count(\*) desc limit 3;



Q.3 Find the total num of distinct aircraft types used in the routes table.

Query - select count(distinct(equipment)) from routes;



Question 2)

Query to create a partition table - (Shell not working)

create table routes\_partitioned

(airline\_iata string, airline\_id int, src\_airport\_iata string,

src\_airport\_id int, dest\_airport\_iata string,

dest\_airport\_id int, codeshare string, stops int, equipment int)

partitioned by (dest\_airport\_iata) row format delimited

fields terminated by "," stored as textfile;

**—-----------------------------------------------------------------------------------------------------------------------------**

**SPARK**

**Question 1**. Solve with RDD only

1)

Code:

Step\_1 Loading file

datardd = sc.textFile("user/cdacuser/training/airlines.csv")

Step\_2 Checking for header

datardd.take(5)

Step\_3 Eliminating the header

head = datardd.first()

eliminate = datardd.filter(lambda a: a!=head)

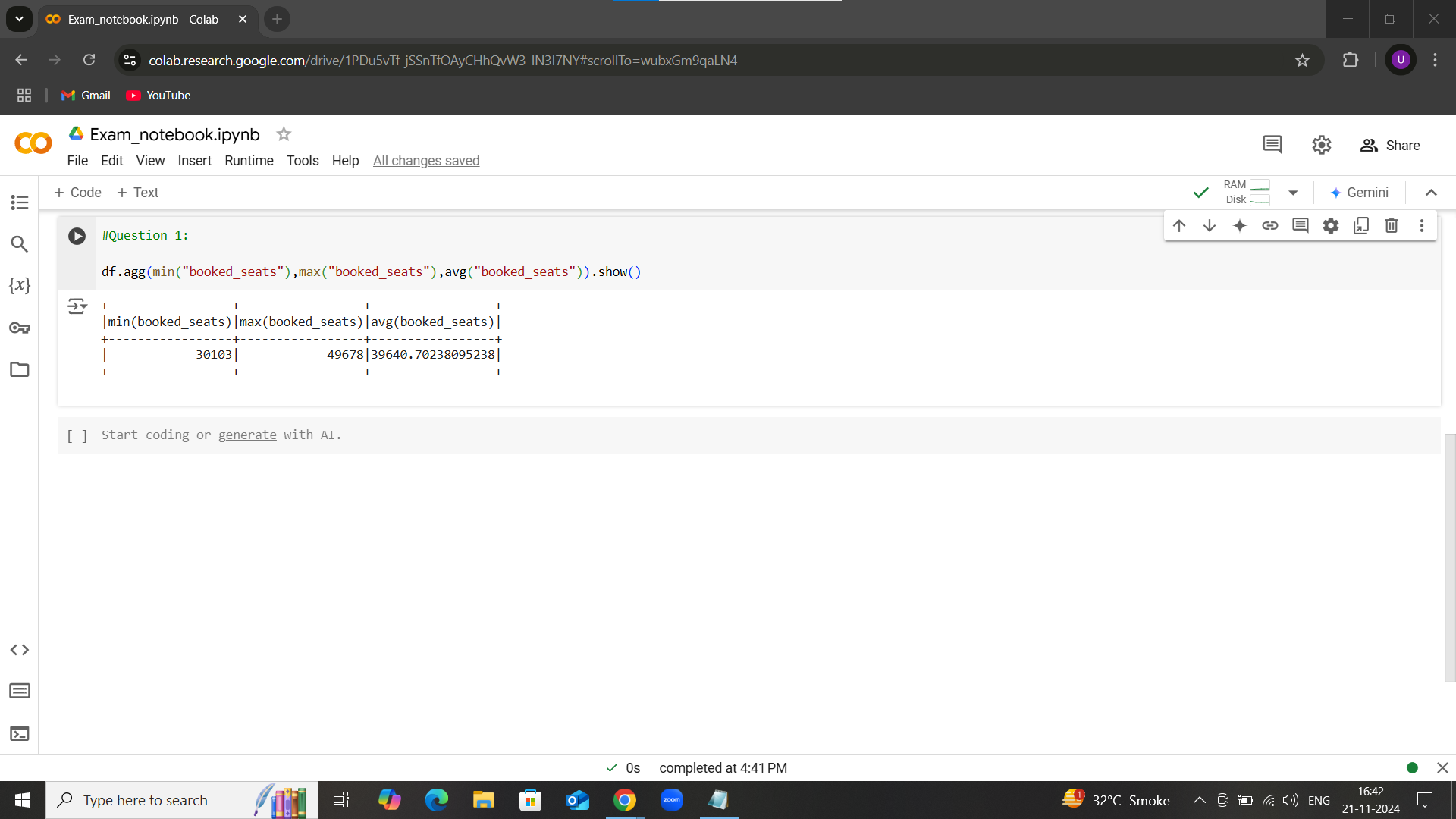
Comment - Spark was not working

**Question. 2**. RDD transformations

**USING RDD OR DATAFRAME: (Shell was not working properly so used Colab and Jupyter notebook for some questions)**

1)

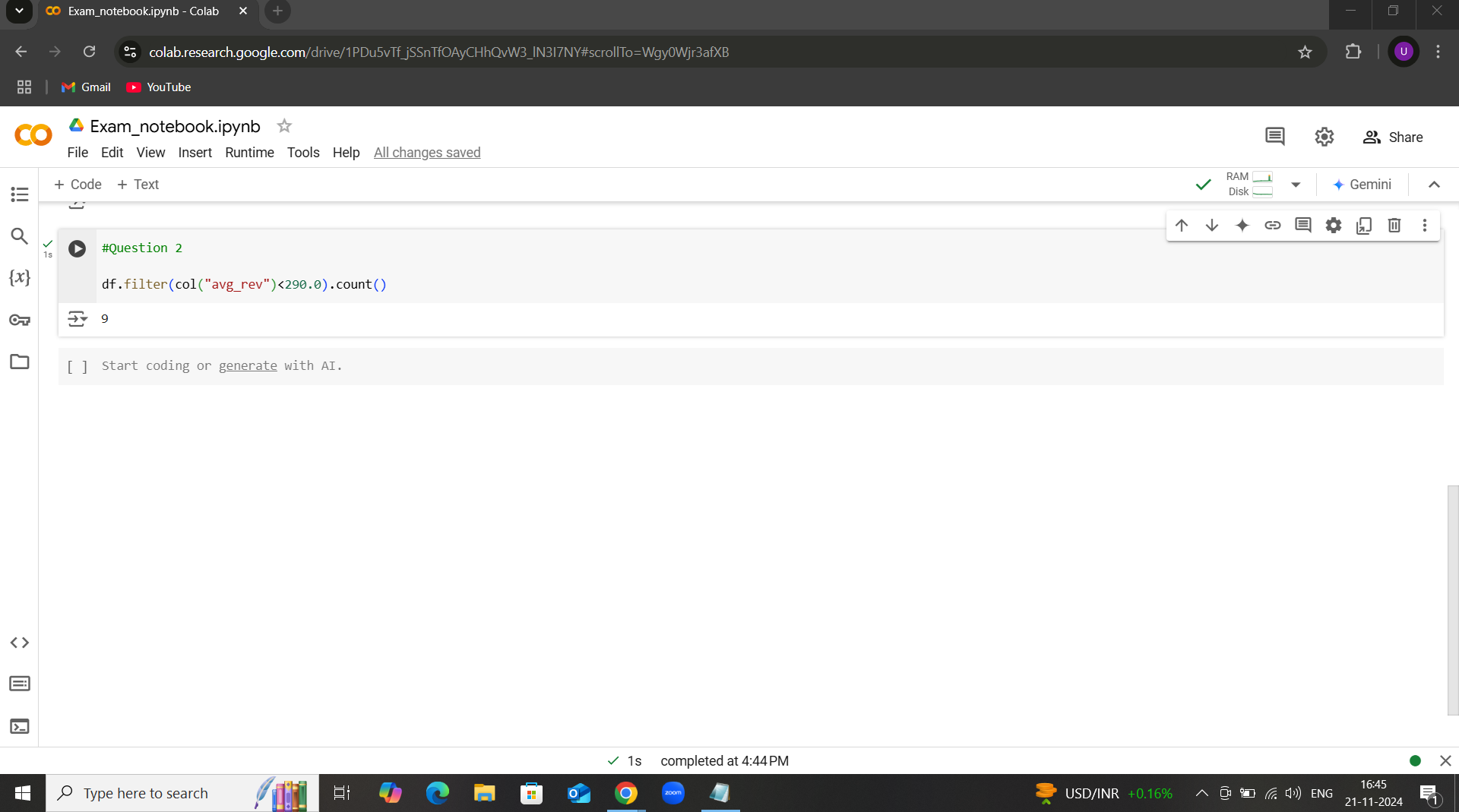
df.agg(min(“booked\_seats”), max(“booked\_seats”), avg(“booked\_seats”)).show()



Question 2:

Query-

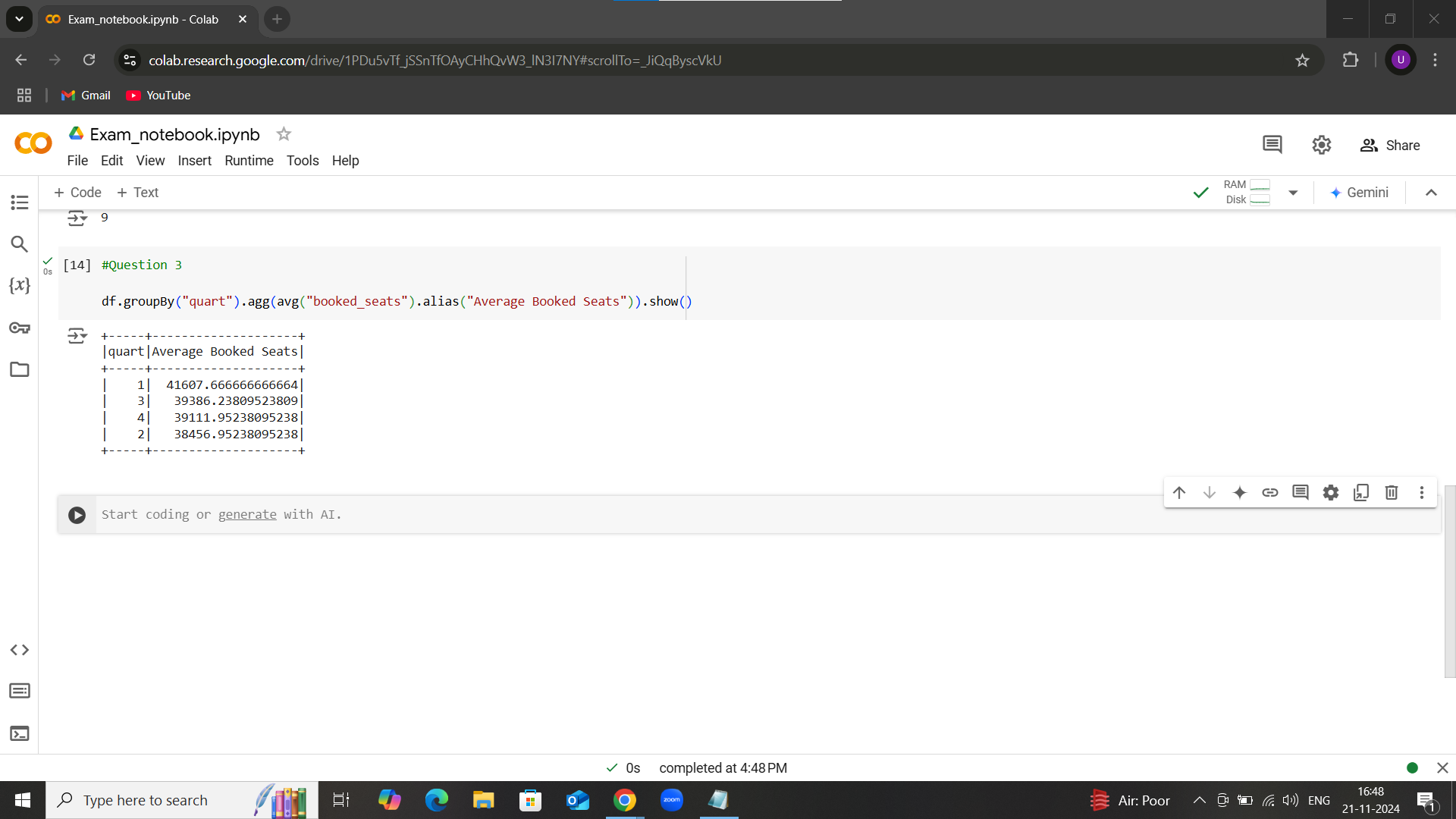
df.filter(col(“avg\_rev”)<290.0)count()



Question 3:

Query -

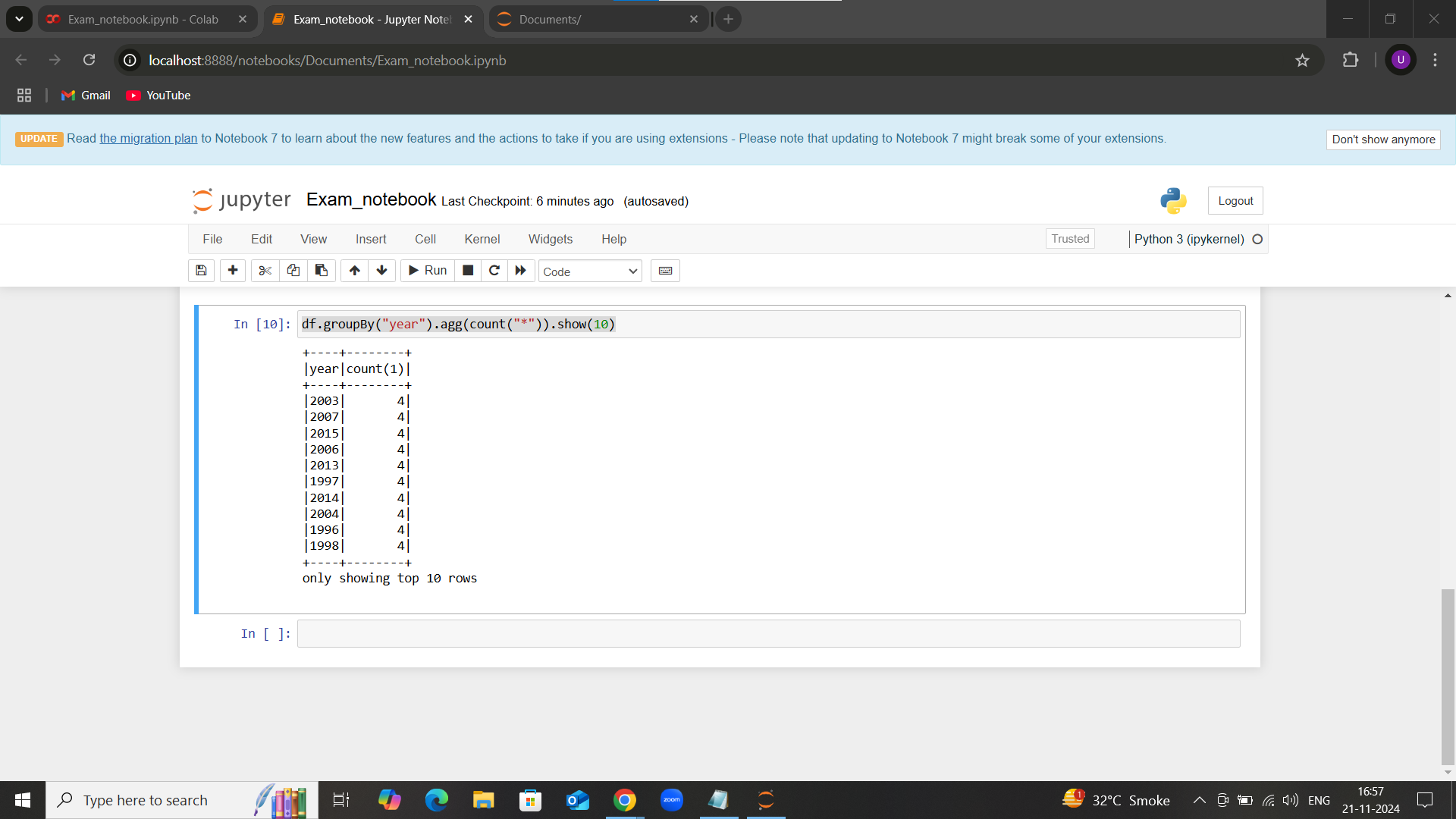
df.groupBy(“quert”).aff(avg(“booked\_seats”).alias(“Average Booked Seats”)).show()



Question 4:

Query -

df.groupBy("year").agg(count("\*")).show(10)



Question 5:

Query -

max\_rev = df.agg(max("avg\_rev")).collect()[0][0]

result = df.filter(col("avg\_rev")==max\_rev)

result.select("quart","avg\_rev").show()

Comments- Webshell is not working