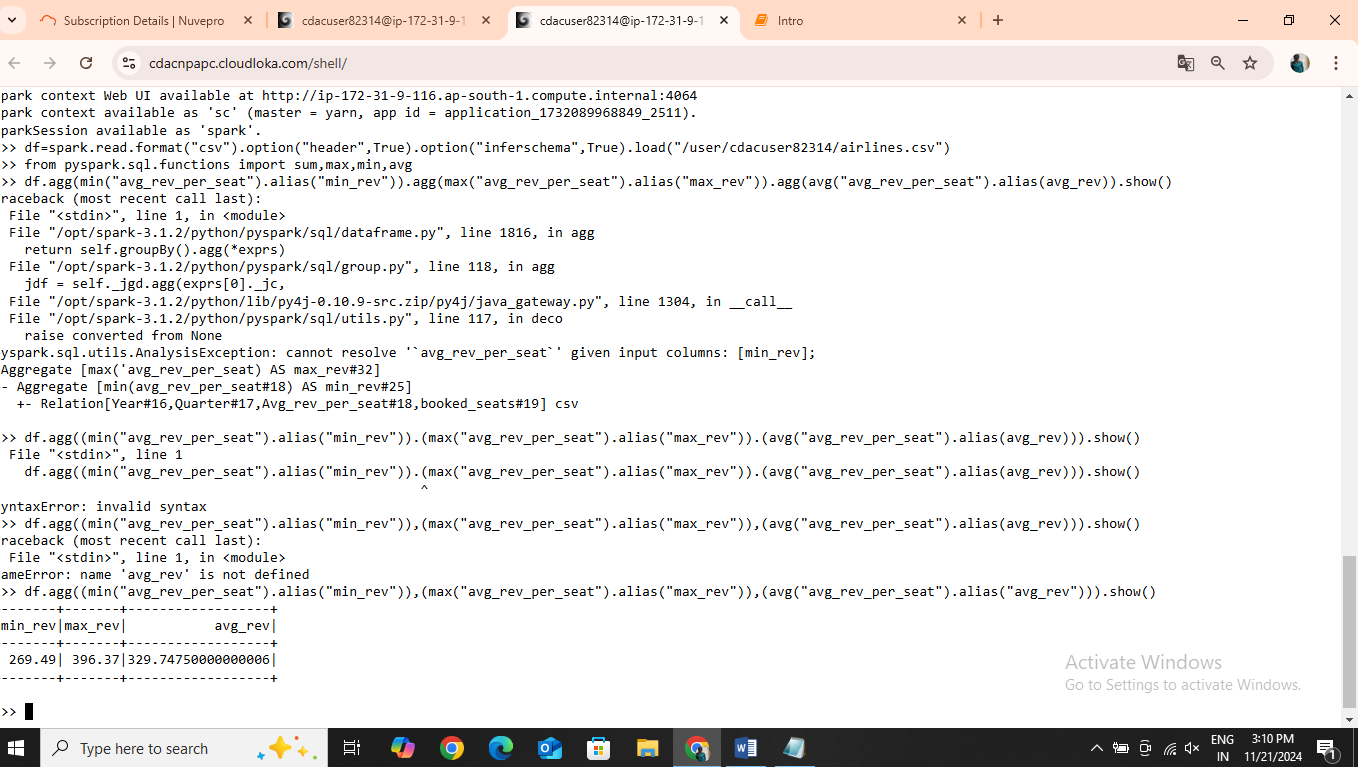
**Spark**

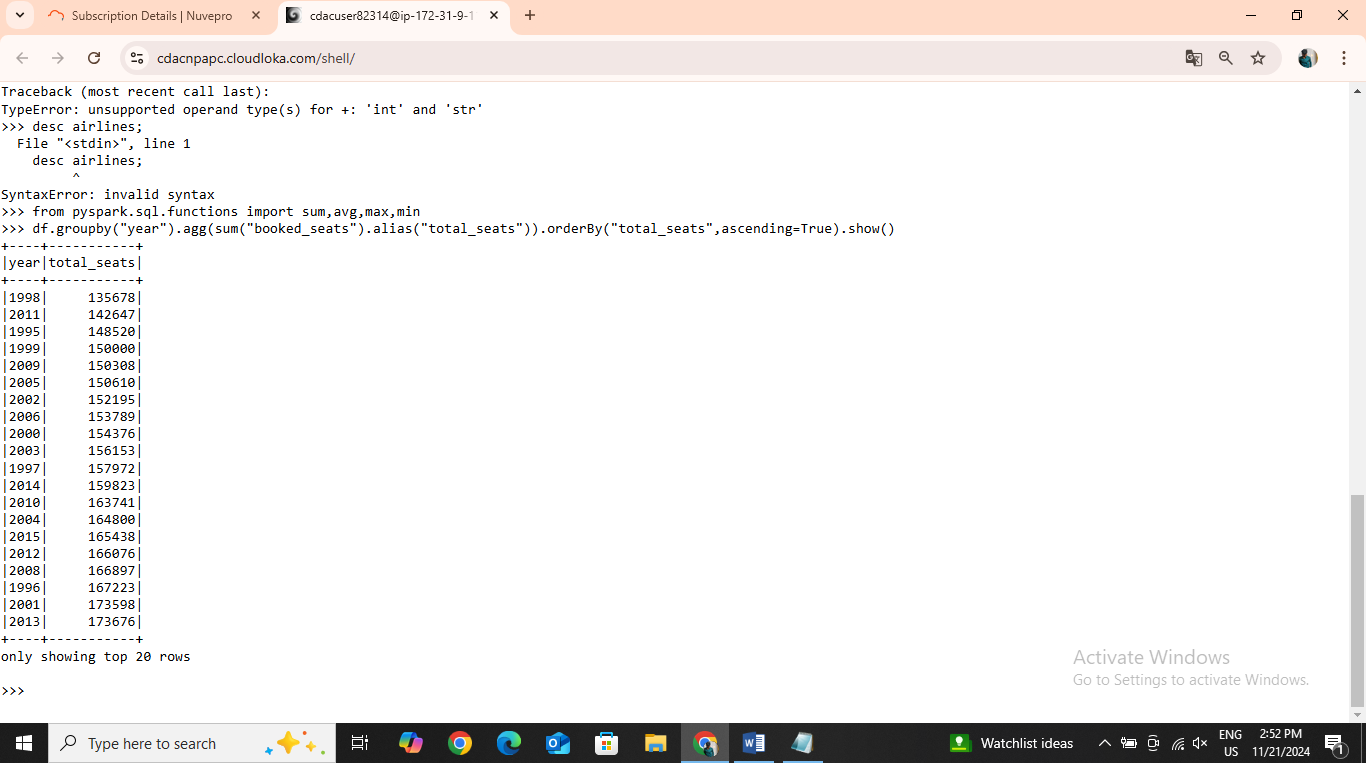
**Q-2.1**

df.agg((min("avg\_rev\_per\_seat").alias("min\_rev")),(max("avg\_rev\_per\_seat").alias("max\_rev")),(avg("avg\_rev\_per\_seat").alias("avg\_rev"))).show()

****

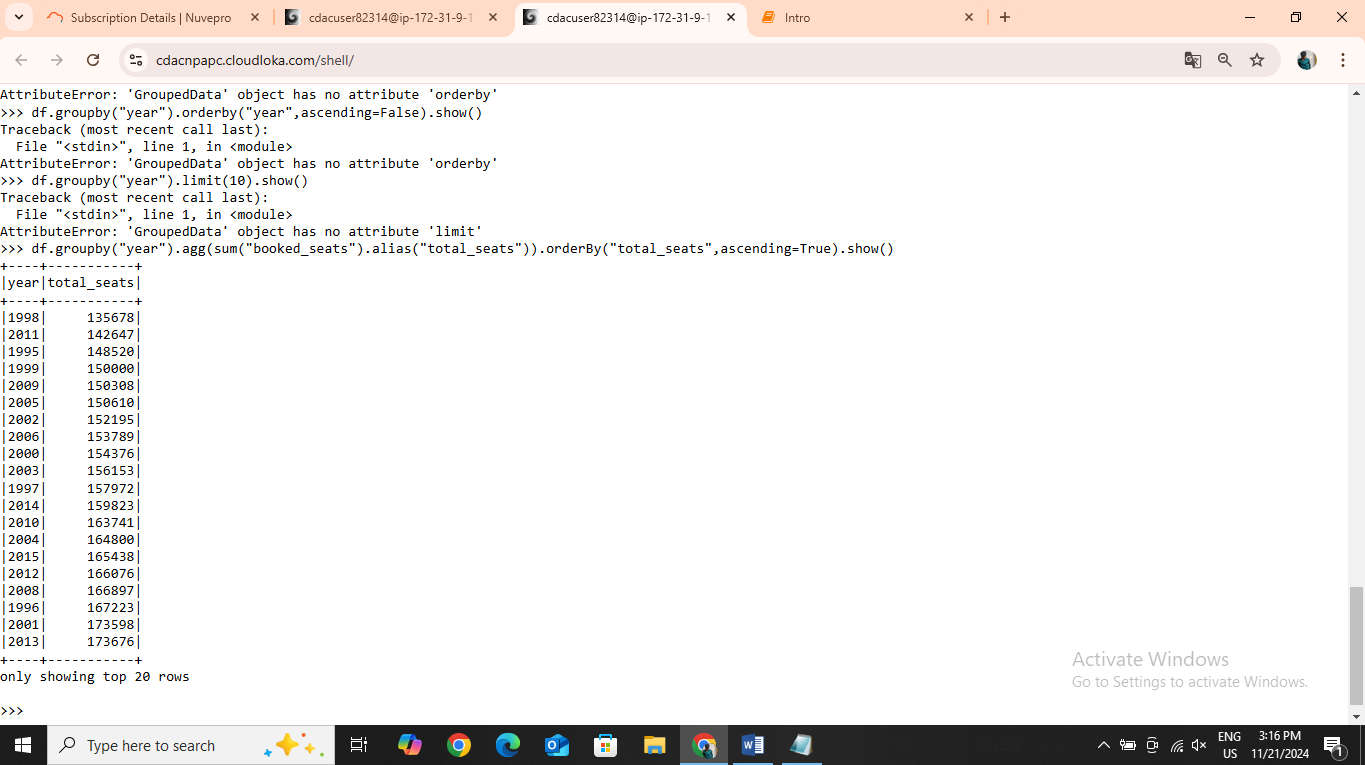
**Q-2.3**

df.groupby("year").agg(sum("booked\_seats").alias("total\_seats")).orderBy("total\_seats",ascending=True).show()

****

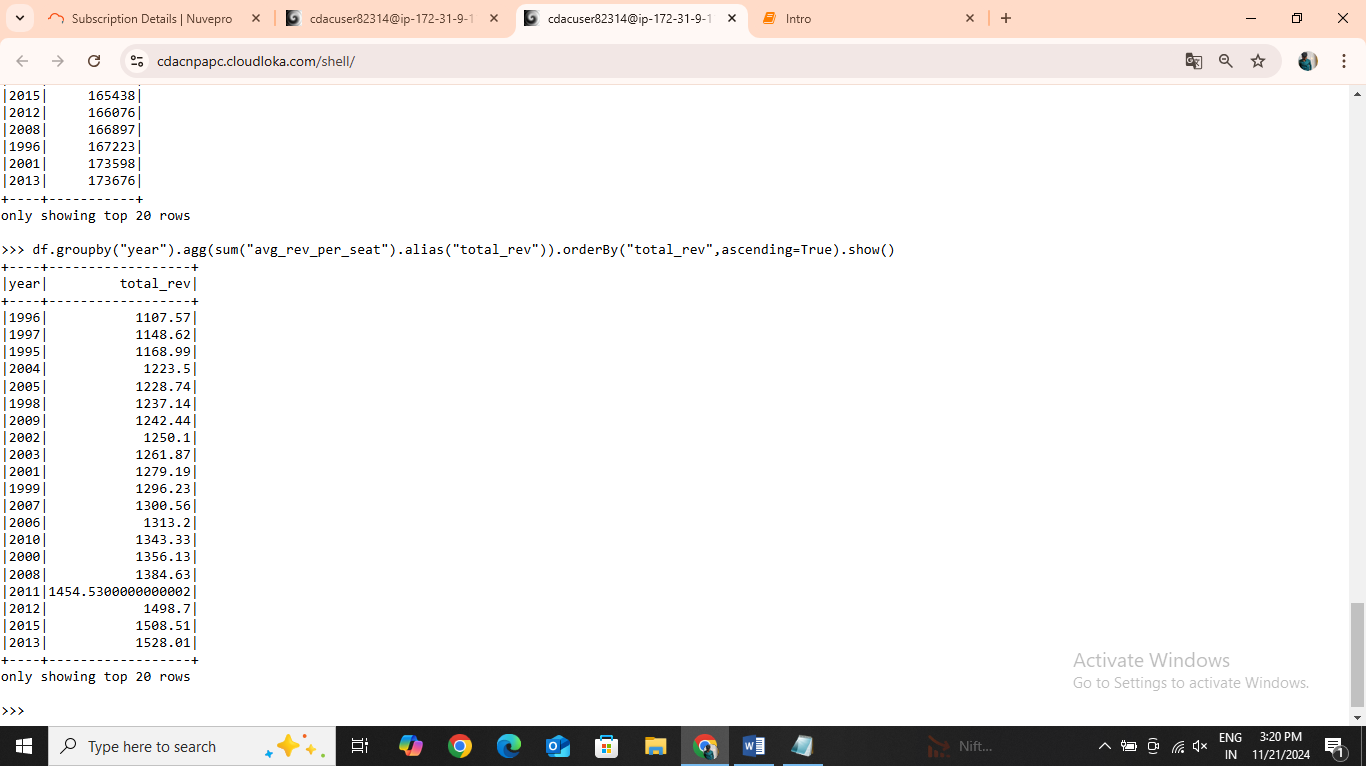
**Q-2.4**

df.groupby("year").agg(sum("booked\_seats").alias("total\_seats")).orderBy("total\_seats",ascending=True).show()

****

**Q-2.5**

df.groupby("year").agg(sum("avg\_rev\_per\_seat").alias("total\_rev")).orderBy("total\_rev",ascending=True).show()

****

**Hive :**

**Q-1.1**

select distinct(name) from airport as a

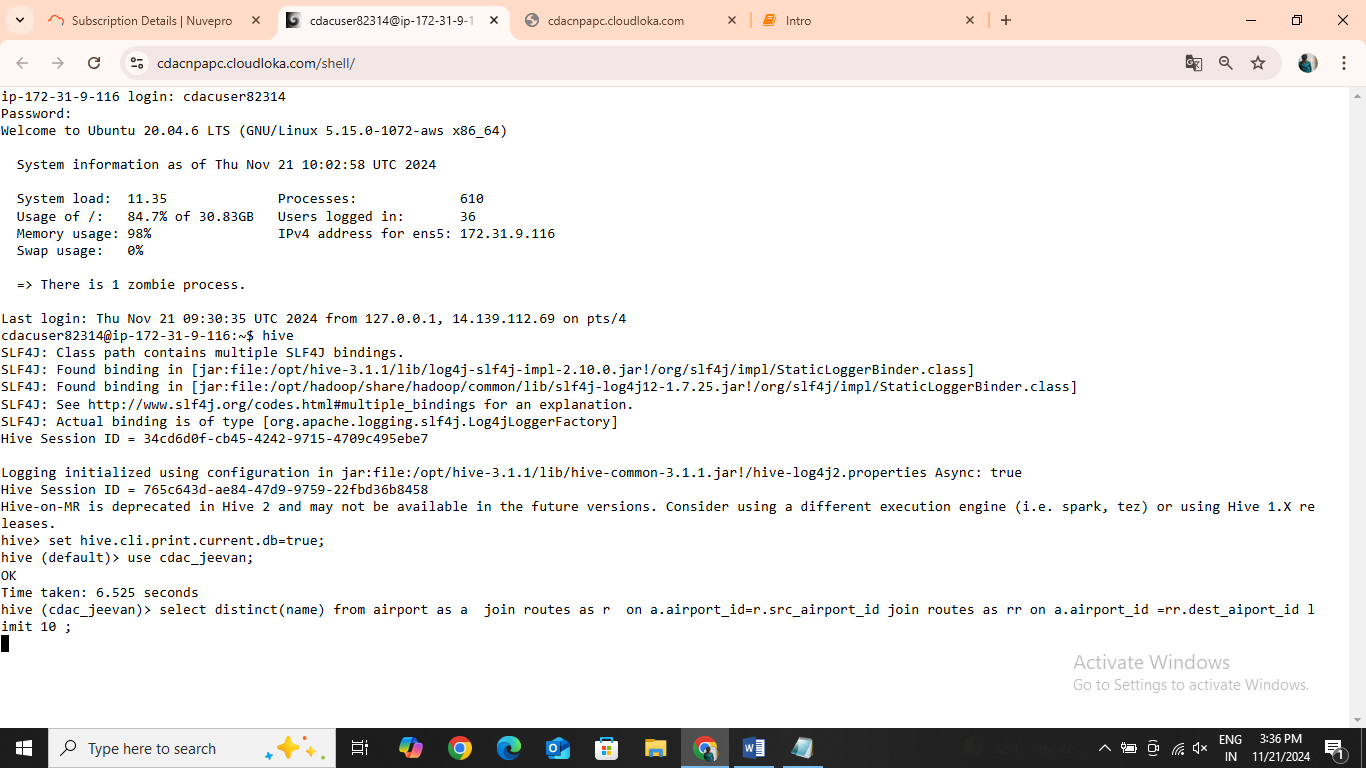
join routes as r

on a.airport\_id=r.src\_airport\_id

join routes as rr

on a.airport\_id =rr.dest\_aiport\_id

limit 10 ;



**Q2.1**

create table routes\_partition (airline\_iata string , airline\_id int , src\_airport\_iata string , src\_airport\_id int , dest\_airport\_iata string , dest\_airport\_id

int , codeshare string , stop int , equipment string ) partitioned by(src\_airport\_id) row format delimited fields terminated by "," stored as textfile;