orders=spark.read.option("header",True).option("inferschema",True).csv("orders.csv")

orders.createOrReplaceTempView("order")

orders.show()

A screenshot of a computer

Description automatically generated

order\_items=spark.read.option("header",True).option("inferschema",True).csv("order\_items.csv")

order\_items.createOrReplaceTempView("order1")

order\_items.show()

A screenshot of a computer

Description automatically generated

products=spark.read.option("header",True).option("inferschema",True).csv("products.csv")

products.createOrReplaceTempView("product1")

products.show()

**A screenshot of a table

Description automatically generated**

1. spark.sql("SELECT order\_status, count(order\_id) from order group by order\_status").show()

**A screenshot of a computer code

Description automatically generated**



spark.sql("SELECT order\_status, order\_id,order\_date, order\_customer\_id from order where order\_status = 'CLOSED' or order\_status = '

COMPLETE'").show()

A screenshot of a computer

Description automatically generated



spark.sql("SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from order o join order1 ot on o.order\_id =

ot.order\_item\_order\_id join product1 p on p.product\_id = ot.order\_item\_product\_id where o.order\_status in ('CLOSED','COMPLETE') group b

y o.order\_date, p.product\_name").show()

A screenshot of a computer

Description automatically generated



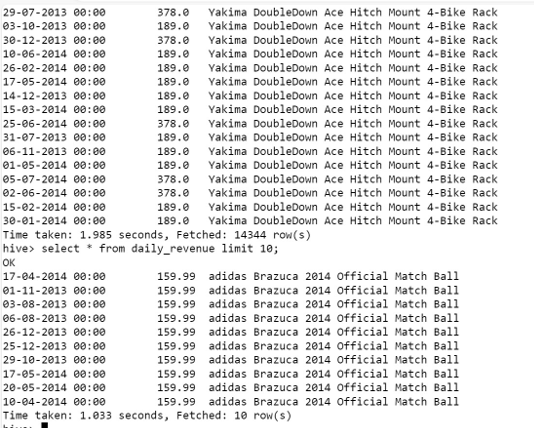
spark.sql("SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from order o join order1 ot on o.order\_id =

ot.order\_item\_order\_id join product1 p on p.product\_id = ot.order\_item\_product\_id group by o.order\_date, p.product\_name").show()

A screenshot of a computer

Description automatically generated





spark=SparkSession.builder.appName("Spark Hive").enableHiveSupport().config("spark.sql.warehouse.dir","/user/hive/warehouse").getOr

Create()

spark.sparkContext.setLogLevel("ERROR")

df = spark.sql("SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from order o join order1 ot on o.order\_

id = ot.order\_item\_order\_id join product1 p on p.product\_id = ot.order\_item\_product\_id group by o.order\_date, p.product\_name")

df.show()

df.write.partitionBy("product\_name").mode("overwrite").saveAsTable("airline\_db.daily\_revenue")