**Data Provided and Inputs:**

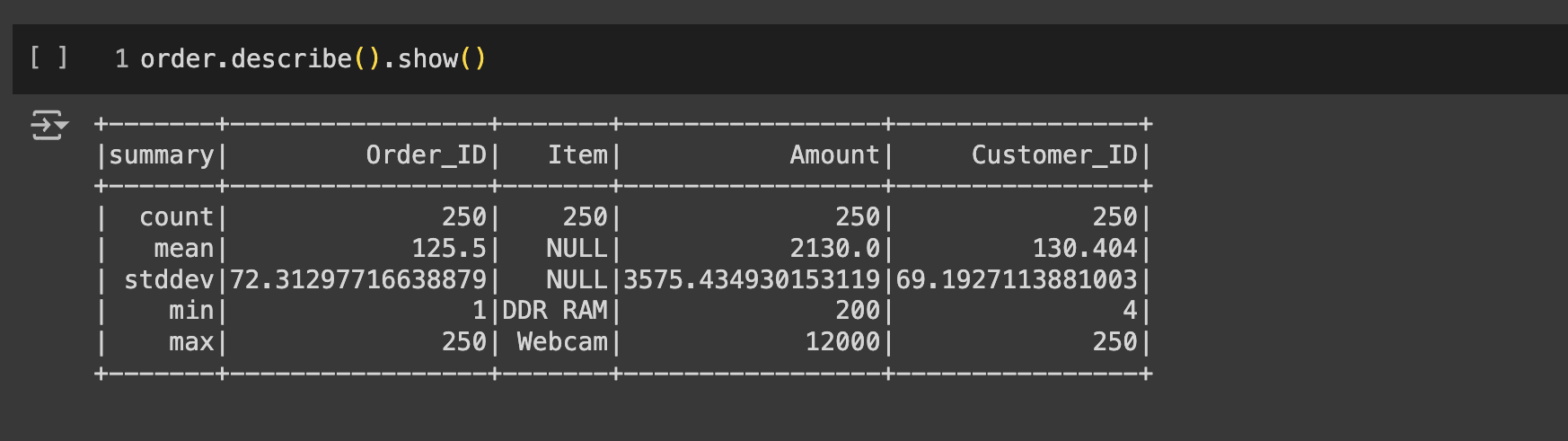
The data given consisted of 3 tables

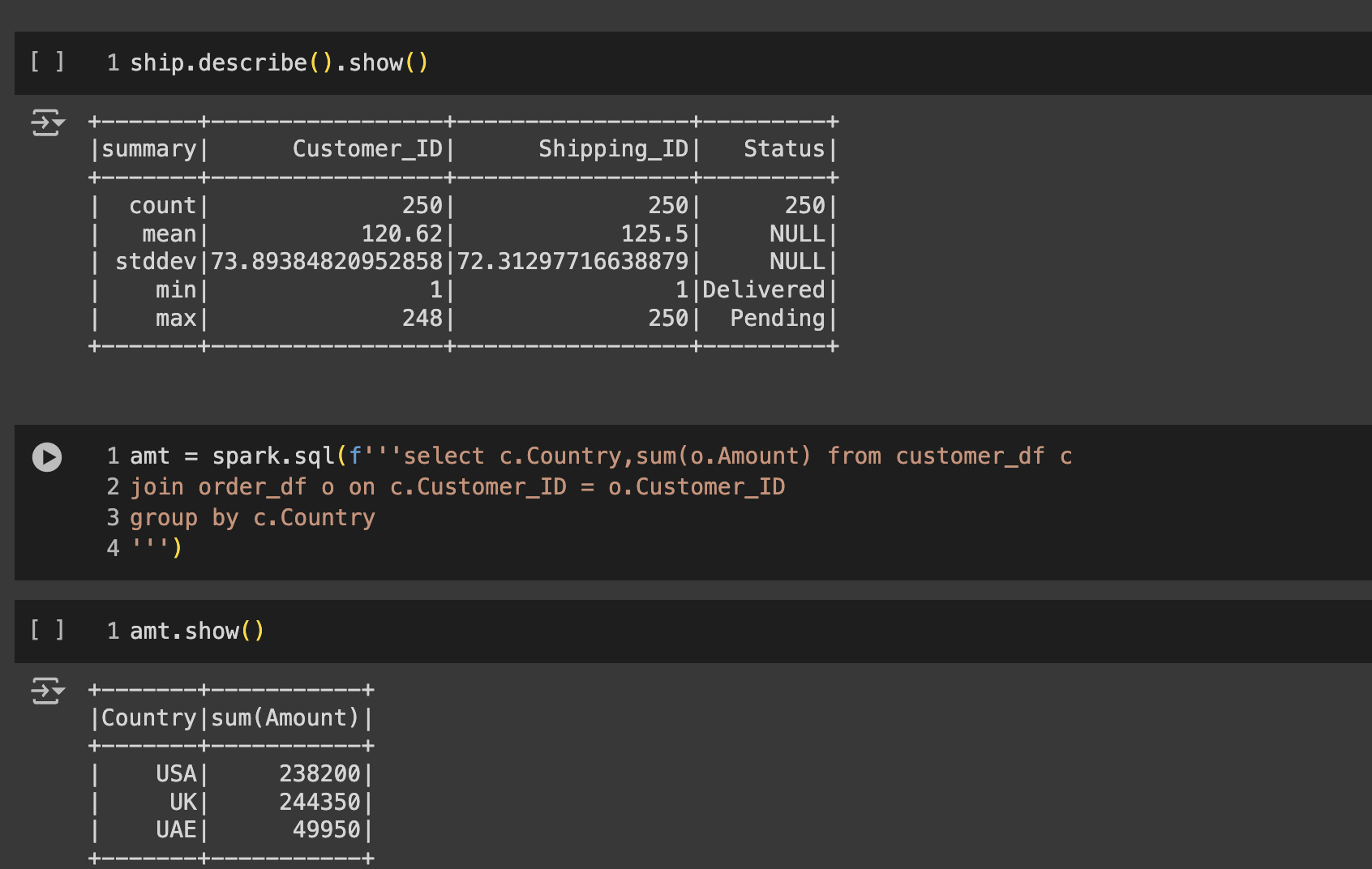
* **Customers** - Giving information on the customers - their Name, Age, and Country along with the ID
* **Orders** - Giving information on the items purchased by the Customers along with Order IDs and Amount spent
* **Shipping** - Giving information on the status for the orders and the shipping ID

**Data Summary:**

In terms of completeness and accuracy of the data the data looks ***complete*** and there are ***no null values or missing data*** in any of the 3 data sets.





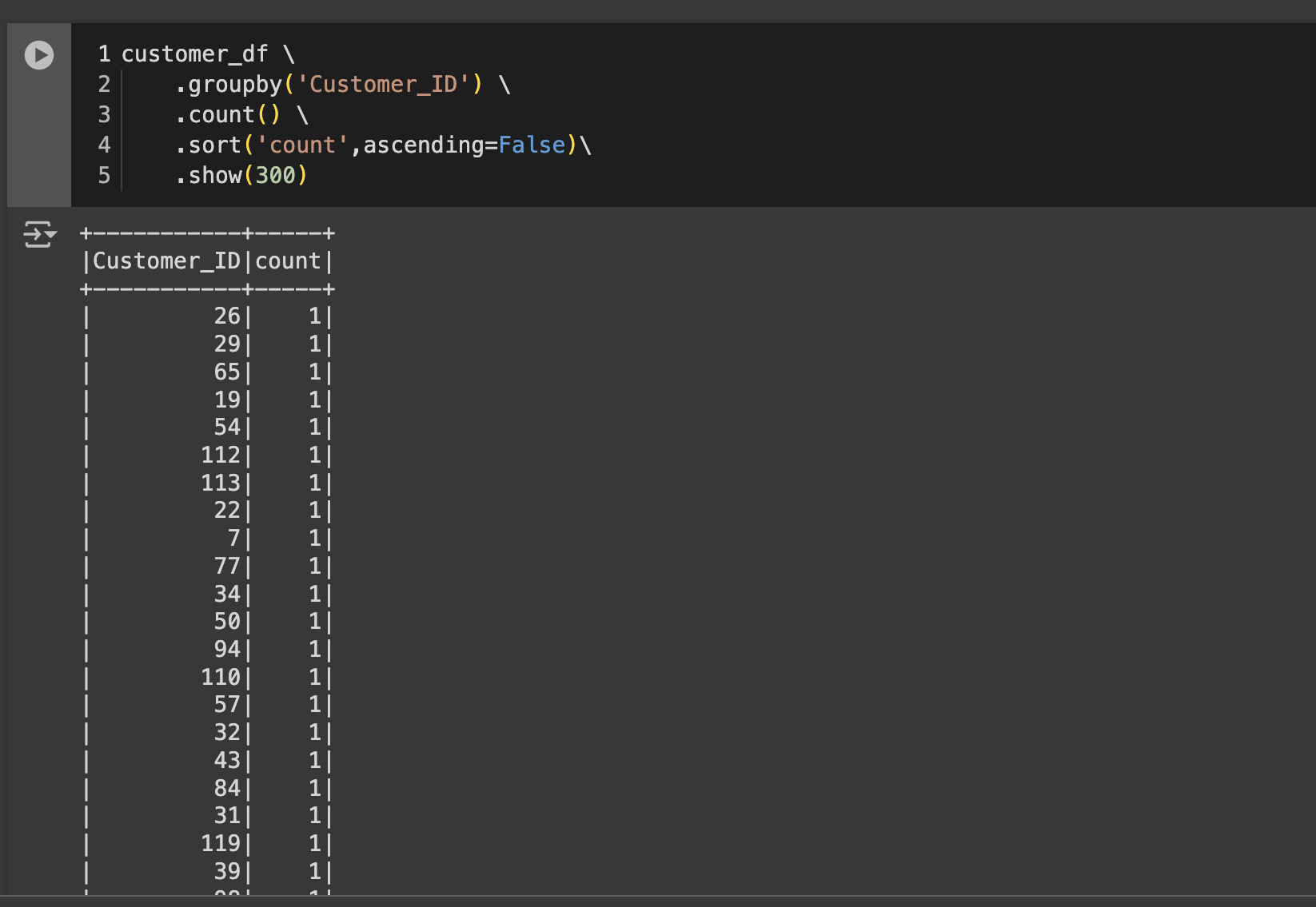


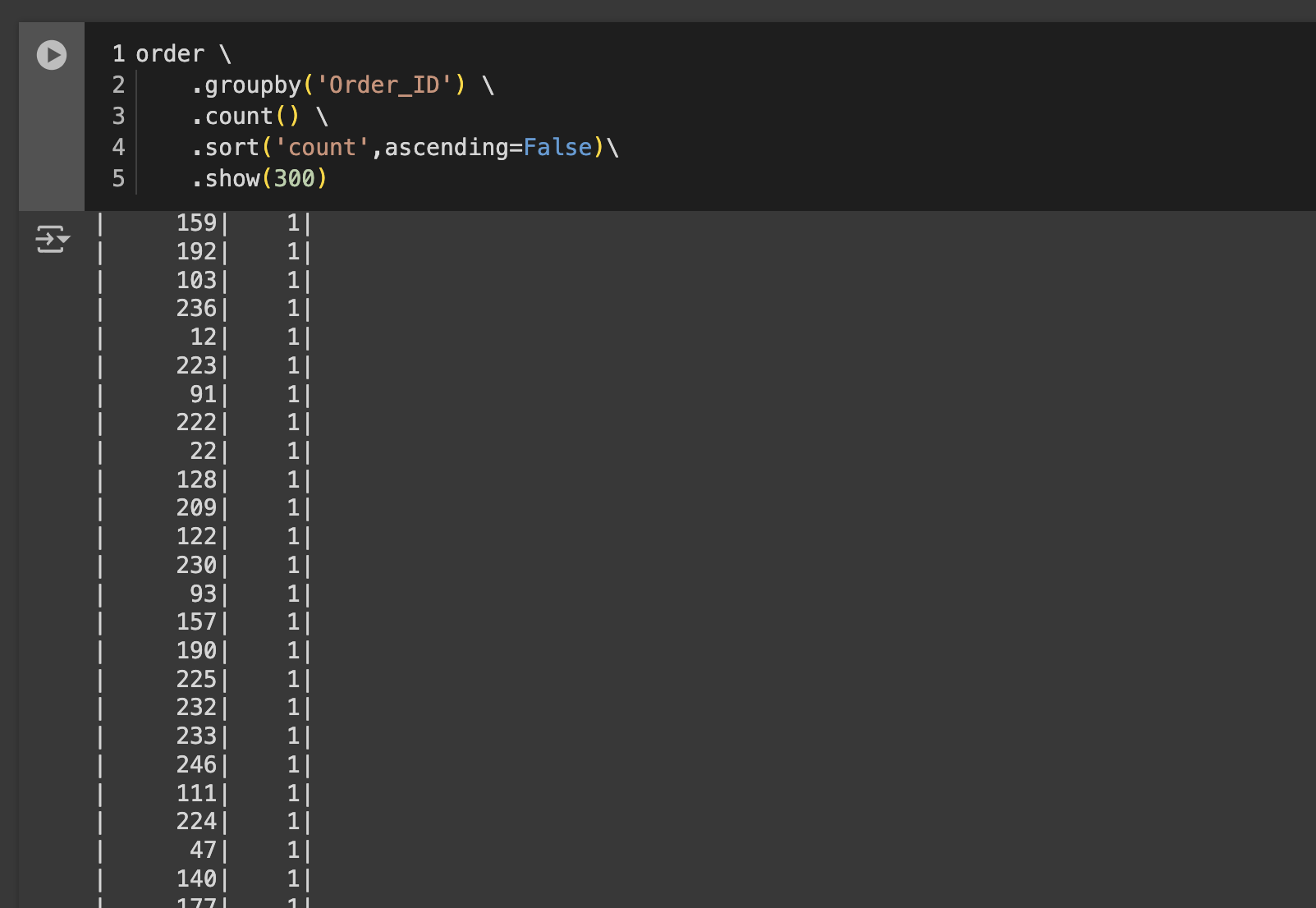
Customers table can be connected to both Orders and Shipping Table by using the Customer ID

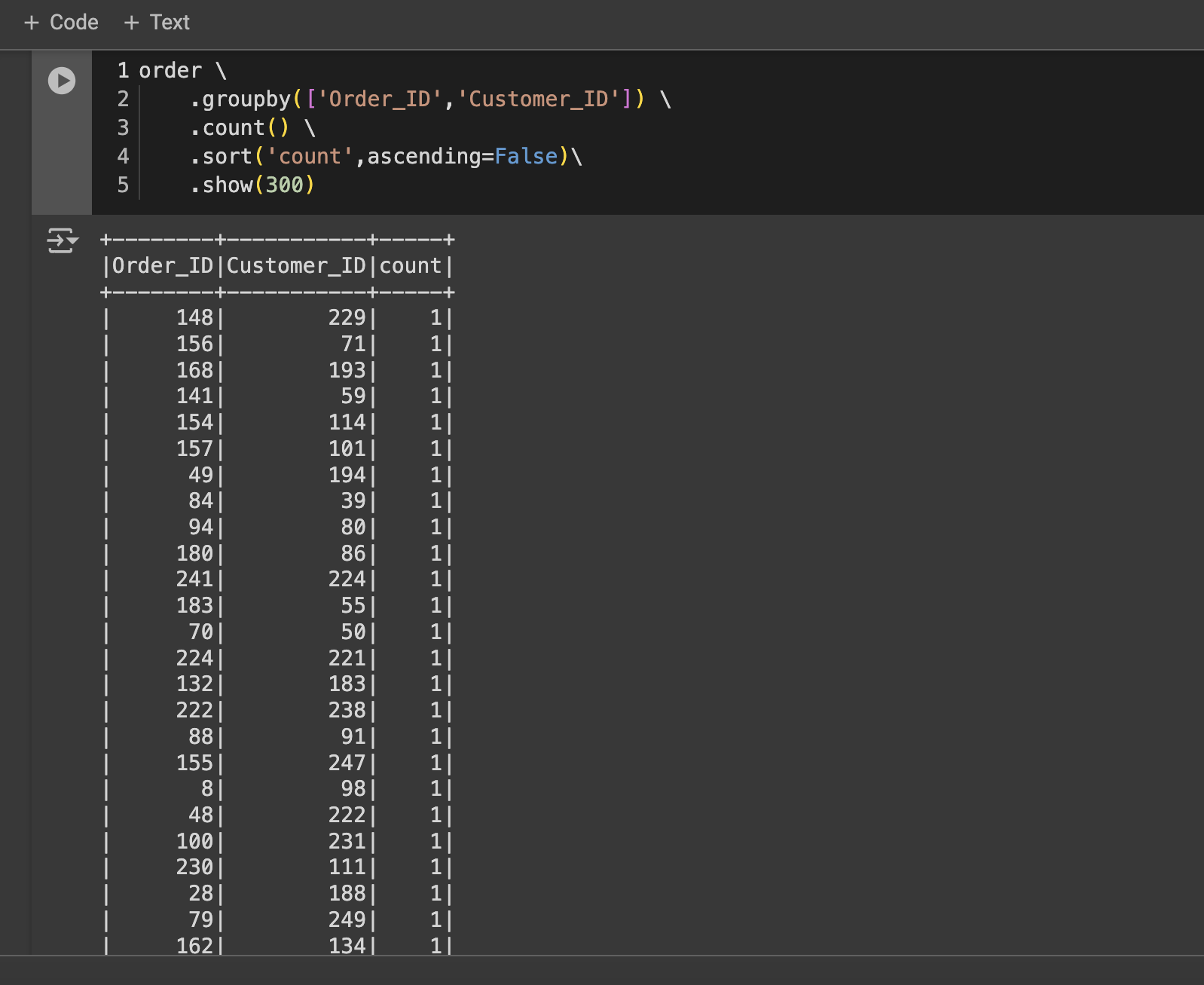
Customer Id is primary key for Customers table and foreign key for both Orders and Shipping table

Order ID is primary key (unique records) in Orders table

Shipping ID has unique values for Shipping Table







There seems to be a 1:1 mapping between customer and orders on the basis of customer\_id and 1:1 mapping between customer and shipping on the basis of customer\_id

The data model is available in this [document](https://docs.google.com/document/d/1GWQI93RGPHBu5U_2BmaGkqKeqHP-WLIAUcPSf5xIHH4/edit) below which explains the relationship between the 3 tables.

The 3 data sets are small datasets and thus we see 1:1 mapping but we could have a relationship like 1 customer has placed multiple orders across different time periods and then the data model would require more nuances to be addressed. Similarly we could have the orders being shipped at different time periods for different customers.

The data engineer has to keep in mind the primary keys and the foreign keys while creating the pipeline and the data types of the keys so that the joins can be done without any issues. The data engineer needs to create 3 data sets and make sure the columns are as specified in the data model. For combining all the 3 tables together, first the engineer can join the customers and orders table and then with this combined table the shipping table can be joined. However since shipping data is available in json it needs to be first converted (I have done in Pyspark). This will give one data set where we have all the information together.

**Insights:**

* The country with the maximum amount spent is the UK, closely followed by the USA. In terms of deliveries which are pending, again UK has the maximum amount of items purchased
* 5 customers have placed 4 orders each comprising of 3 or 2 products and 1 of the customers (Id - 96) have purchased 4 different items
* In terms of amount spent customer ID 166 has the maximum amount spent followed by 129 and 123
* Customer 166 has purchased 3 products with a monitor of the highest value
* For the products purchased at max by Country we have: Keyboard for UAE and Mousepad for both USA and UAE
* In terms of age group people with an age less than 30 prefer ***Mousepad*** followed by ***Keyboard***. While people in the age group of greater than equal to 30 prefer ***Keyboard***, followed by ***Mousepad***
* The country with minimum transactions and sales amount is UAE