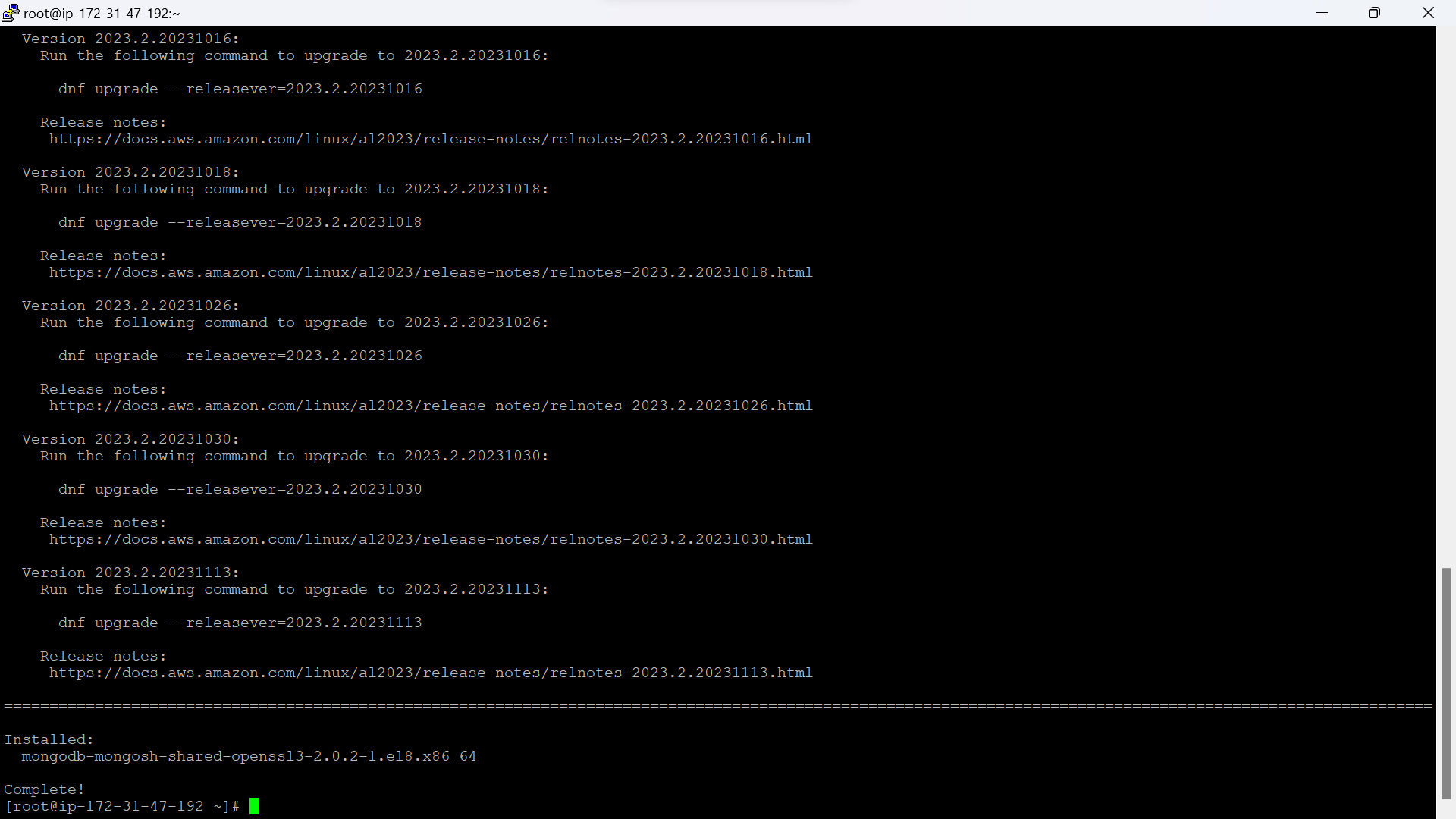
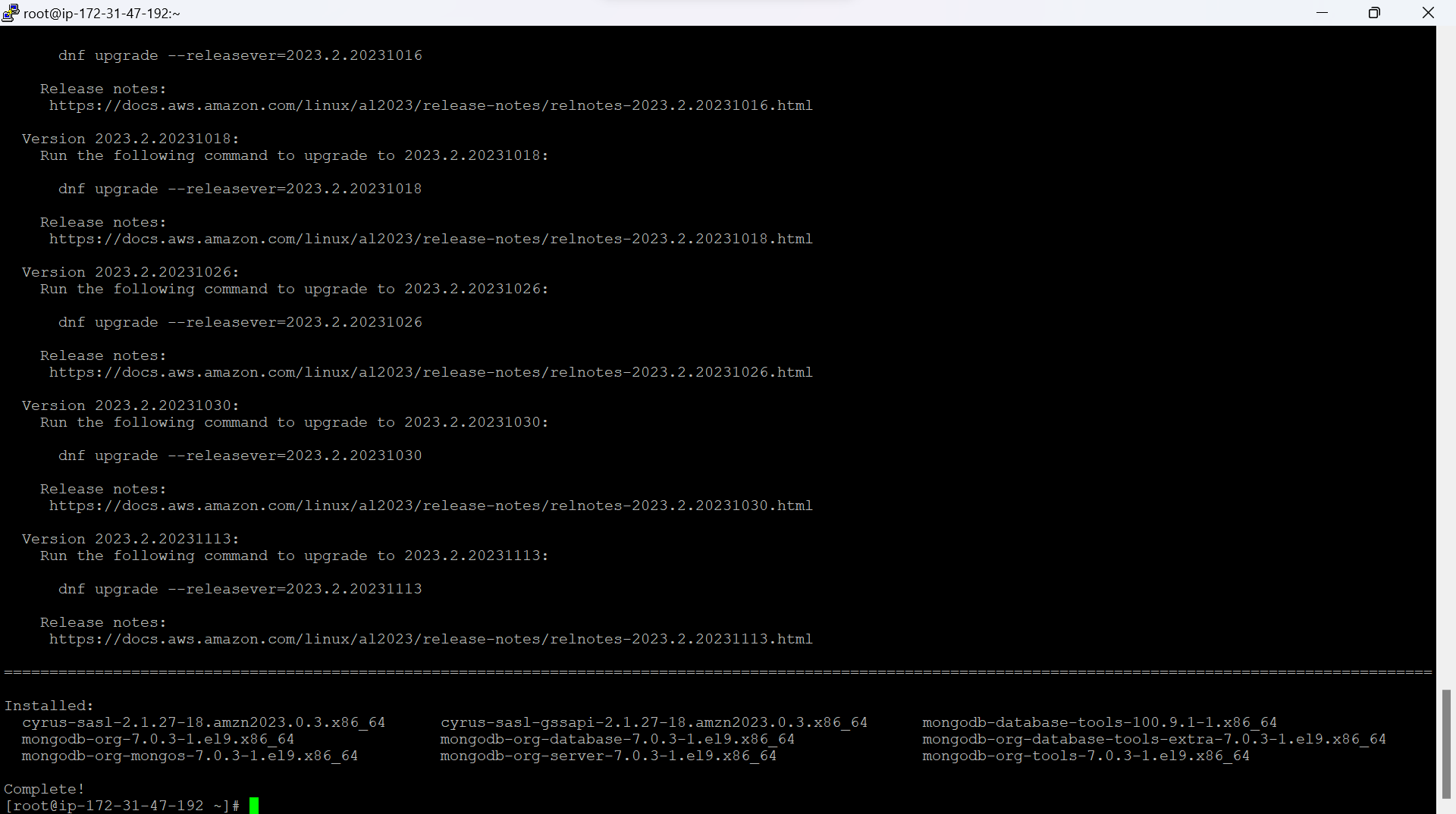
**/\* On my honor, as an Aggie, I have neither given nor received unauthorized assistance on this assignment. I further affirm that I have not and will not provide this code to any person, platform, or repository, without the express written permission of Dr. Gomillion. I understand that any violation of these standards will have serious repercussions. \*/**

1. **Install MongoDB onto an AWS EC2 instance running on Amazon Linux 2023**

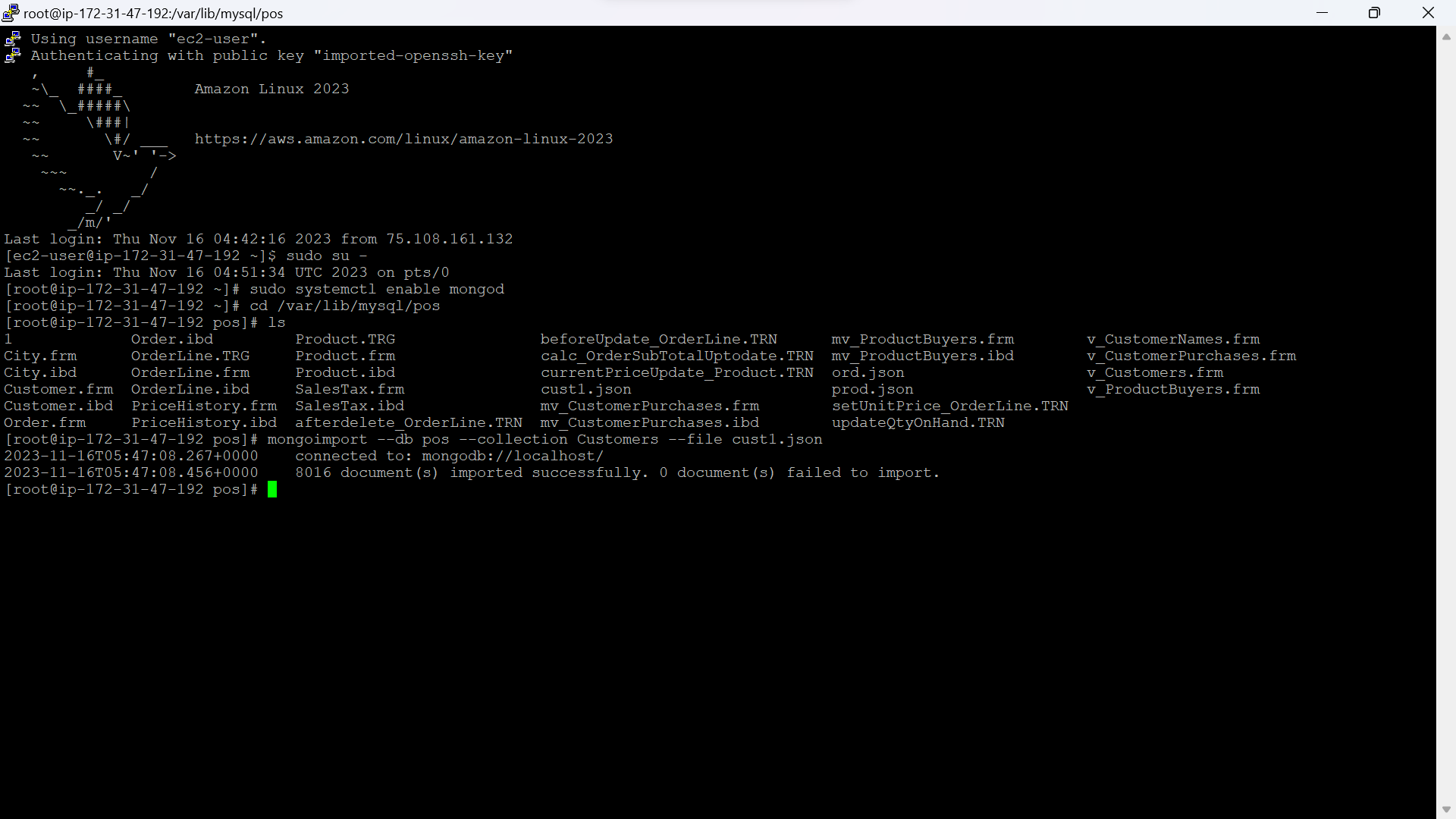




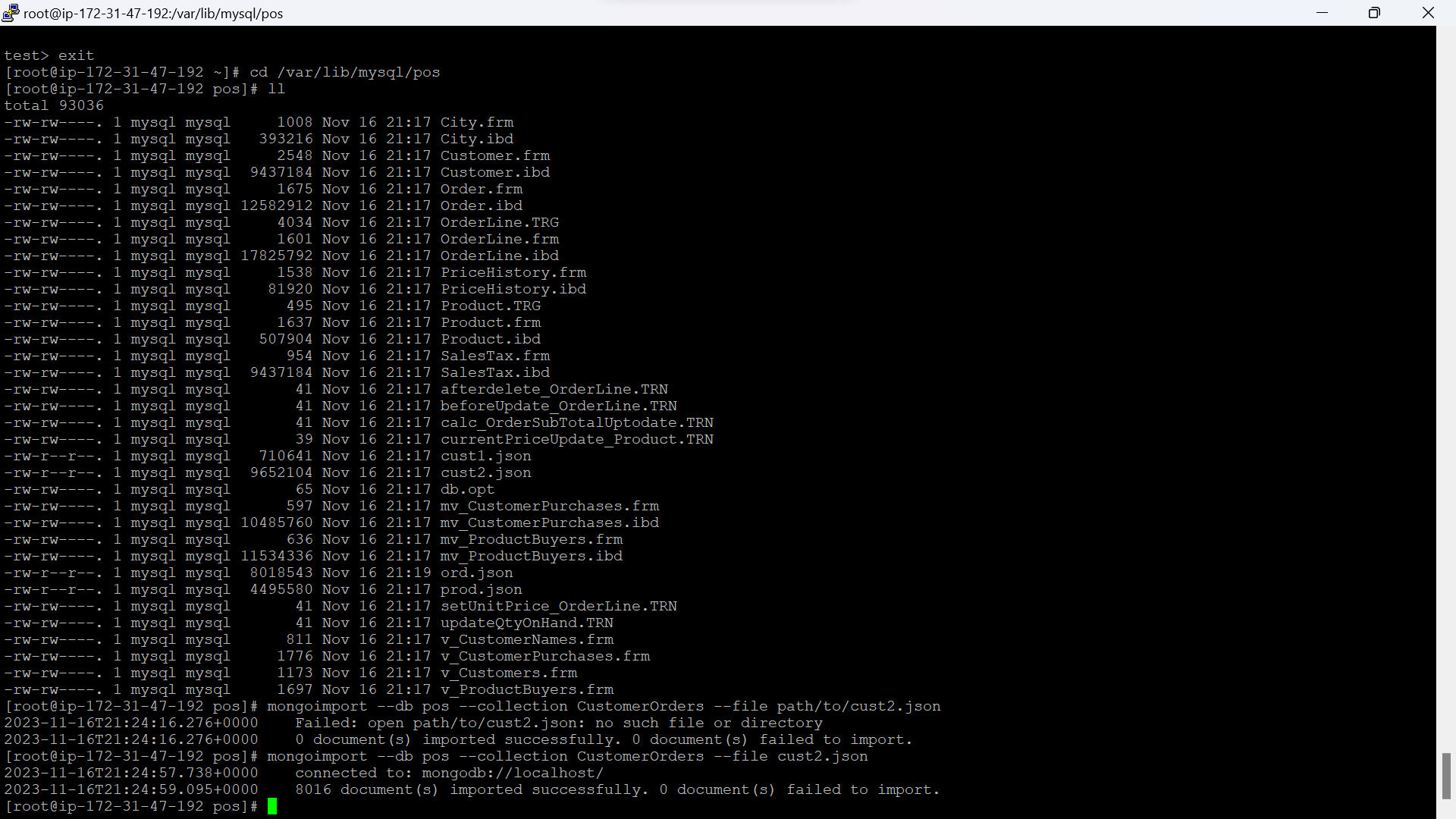
**Q) Why did you have to replace $releaseversion with 9? Explain the significance of 9 and why $releaseversion was unable to be used successfully**

$releasever is a variable which represents the release version of the operating system. We have used 9 as the release version because Amazon Linux 2023 corresponds to the Red Hat Enterprise Linux version 9. By changing the $releasever as 9 in the repo file, the mongoDB repository is configured correctly for the Amazon Linux 2023, as it aligns with the related or underlying RHEL version. Using $releasever will not be suitable if it doesn’t reflect RHEL version that Amazon Linux 2023 is based on. By mentioning 9 instead of $releasever, we are making sure that the appropriate MongoDB packages are retrieved for RHEL 9 during installation process.

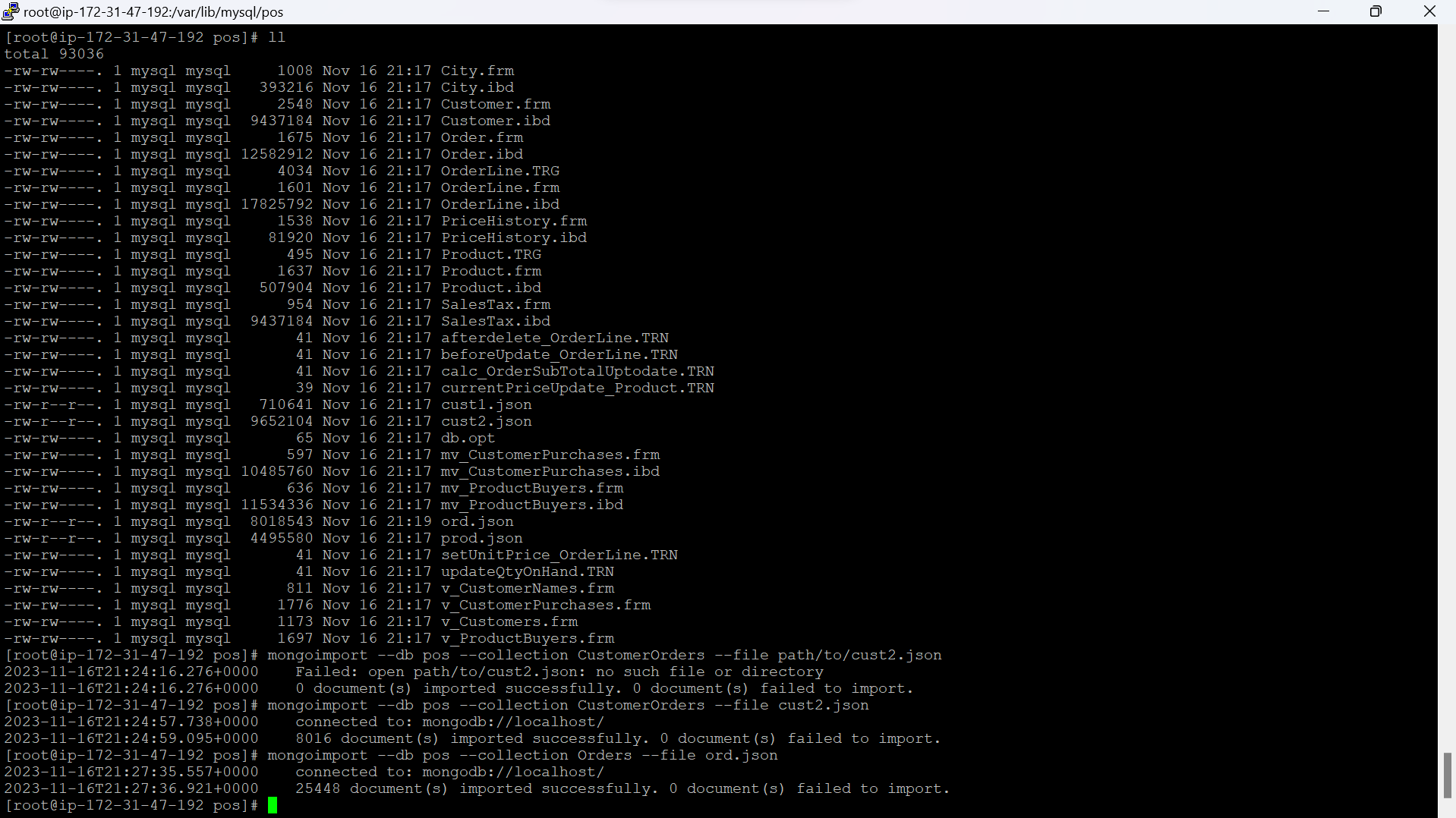
1. **Use mongoimport to import cust1.json**



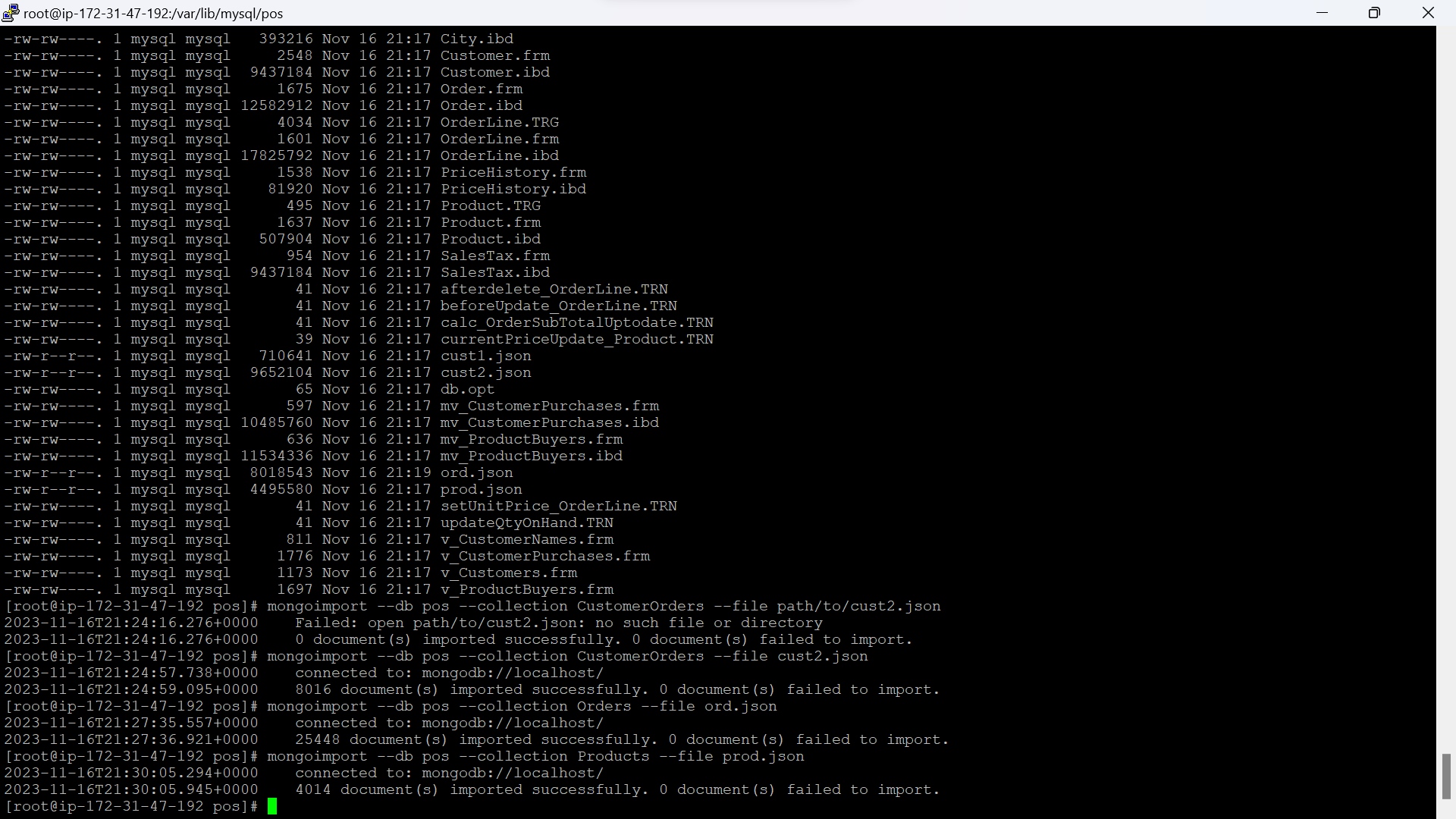
1. **Use mongoimport to import cust2.json**



1. **Use mongoimport to import ord.json**



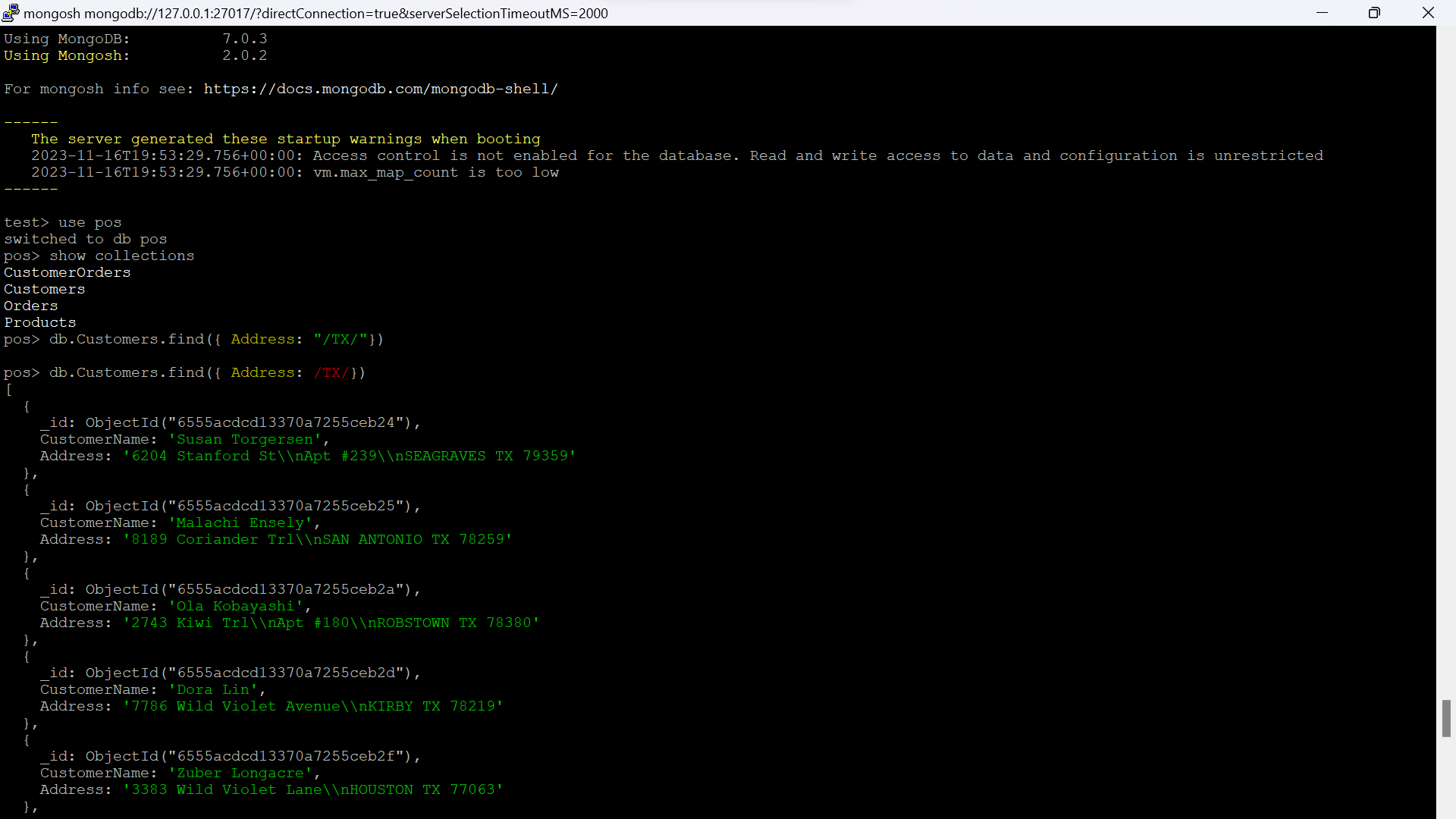
1. **Use mongoimport to import prod.json**



**Q) What is the purpose of mongoimport, and why did we format the files the way that we did in the last milestone?**

Mongoimport is a command which is used to import data from the files formatted as JSON, CSV and TSV files into collections of MongoDB. MongoDB uses BSON(Binary JSON) as its native data format, so in the last milestone we have formatted the files in the JSON format so that we can import data from the files into the collections of MongoDB easily without any data losses.

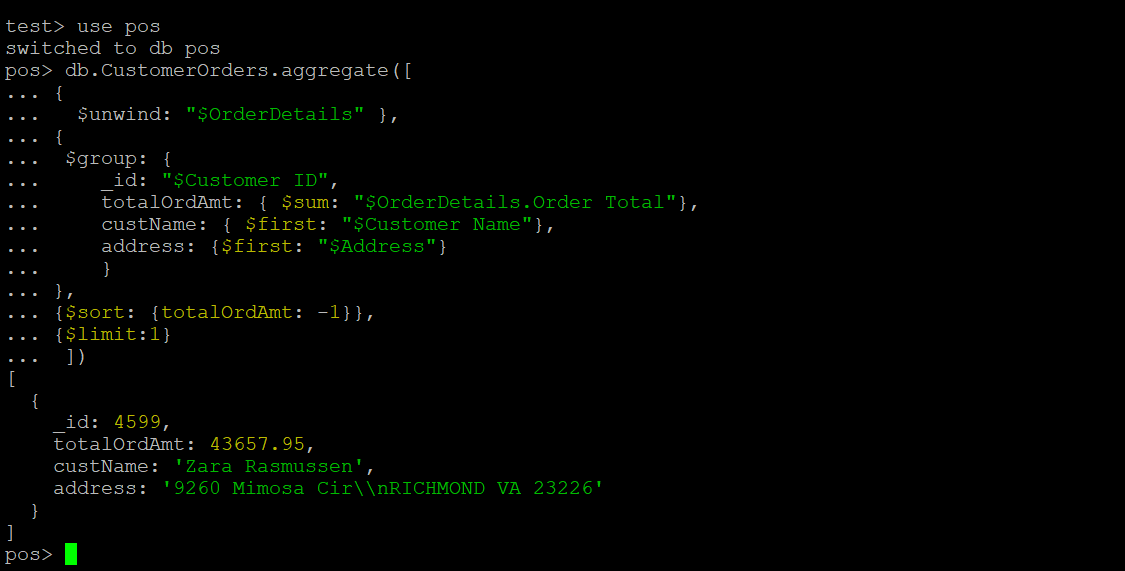
1. **Who are my customers that live in Texas?**

****

**Q) Explain your query and each part of it, including why you chose the collection you used.**

In this query, I’m trying to find the customers who live in Texas. Customer data was imported into the Customers collection. I have selected customers collection to get the customers list easily, as it contains 2 fields (1 – customer name, 2- Address contains city, state, zip etc.). So I have used find function to retrieve the customers whose location is Texas. To find the location, I have used Address field. Find({Address: /TX/}) – this is going to retrieve the customers whose address field contains TX value.

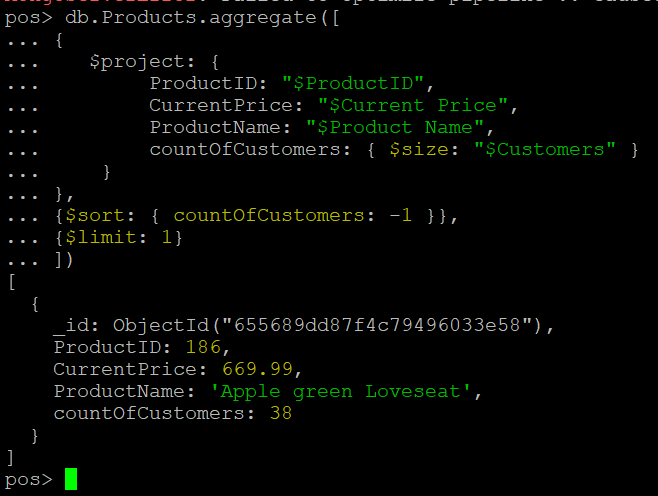
1. **Who is my best customer?**

****

**Q) Explain your query and each part of it, including why you chose the collection you used.**

I have selected CustomerOrders collection to find the best customer because it contains customer information and orders information of each customer. My parameter to find out the best customer is order total. In my opinion, If there are 2 customer – A,B. A has 2 orders and B has 3 orders. If the sum of the order total of 2 orders of Customer A > Sum of the order total of 3 orders of Customer B, then Customer A is considered as best customer. According to that, first I’m unwinding the Order details array of the CustomerOrders collection. This will create a new document for each element in the array. And then I’m grouping the documents by unique values of the field ‘Customer ID’. For each group, it’s going to calculate the sum of the order totals using $sum. Using the sort statement to sort the results based on the totalOrdAmt in descending order. Using limit statement to return the first document which has highest total order amount.

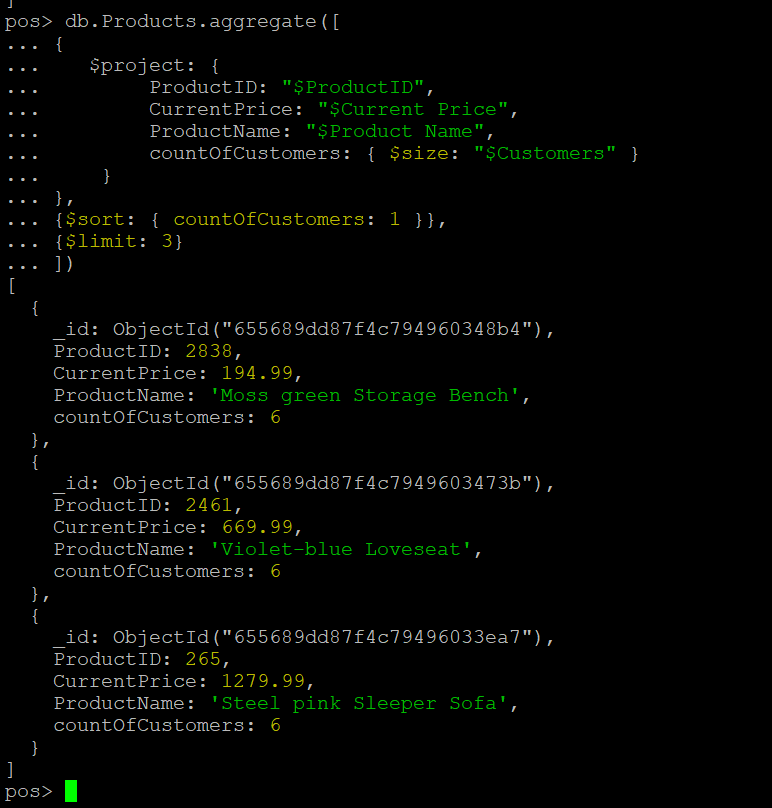
1. **What is my best product?**

****

**Q) Explain your query and each part of it, including why you chose the collection you used.**

I have selected Products collection because the products collection contains products information and the customer details who bought that product. In my opinion whichever product was bought by the most number of customers is considered as the best product. According to that I have used $project to project a new set of fields for each document along with that I’m calculating the count of customers using the $size command with respect to the customers array of each document. I’m using the $sort statement to sort the documents based on the count of Customers in descending order. Using $limit to the return the first document in the sorted order which has highest number of customers.

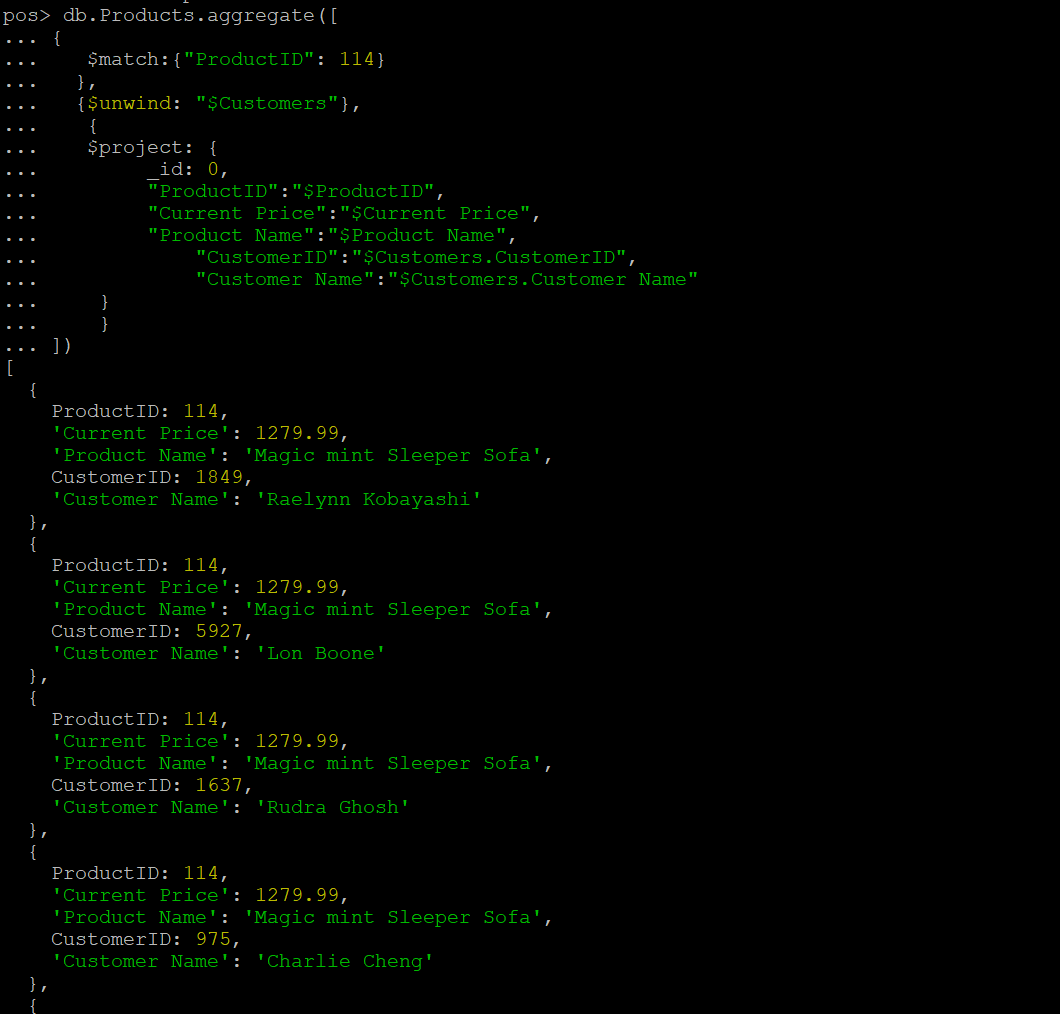
1. **Which products should I consider no longer carrying?**

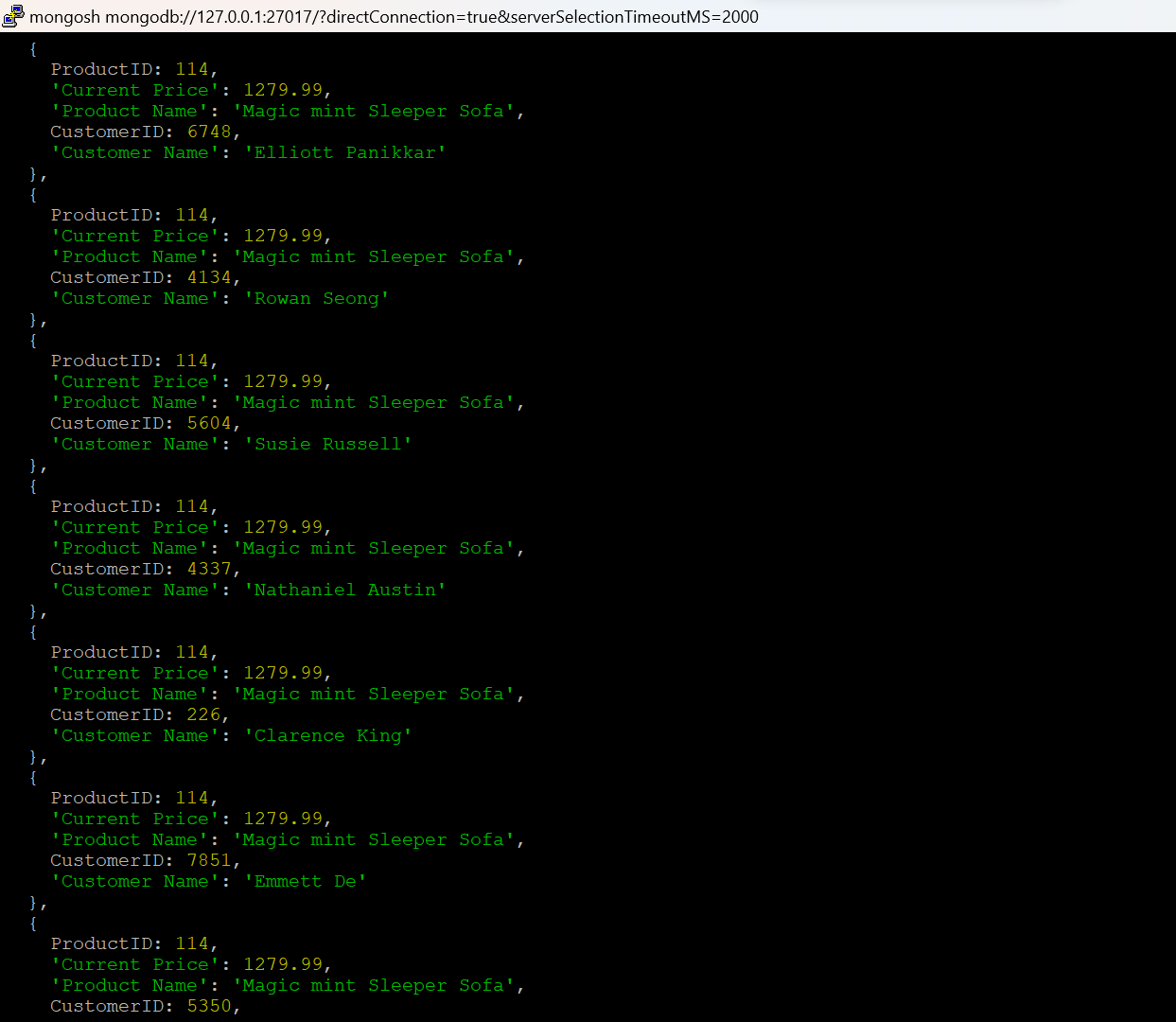
****

**Q) Explain your query and each part of it, including why you chose the collection you used.**

I have selected Products collection because the products collection contains products information and the customer details who bought that product. In my opinion whichever product has bought by the less number of customers is considered as the no longer carrying / out of date product. According to that I have used $project to project a new set of fields for each document along with that I’m calculating the count of customers using the $size operator with respect to the customers array of each document. I’m using the $sort statement to sort the documents based on the count of Customers in ascending order. Using $limit to the return the top three documents in the sorted order which has lowest number of customers.

1. **How will I find all customers that purchased a product in case of a recall?**

****

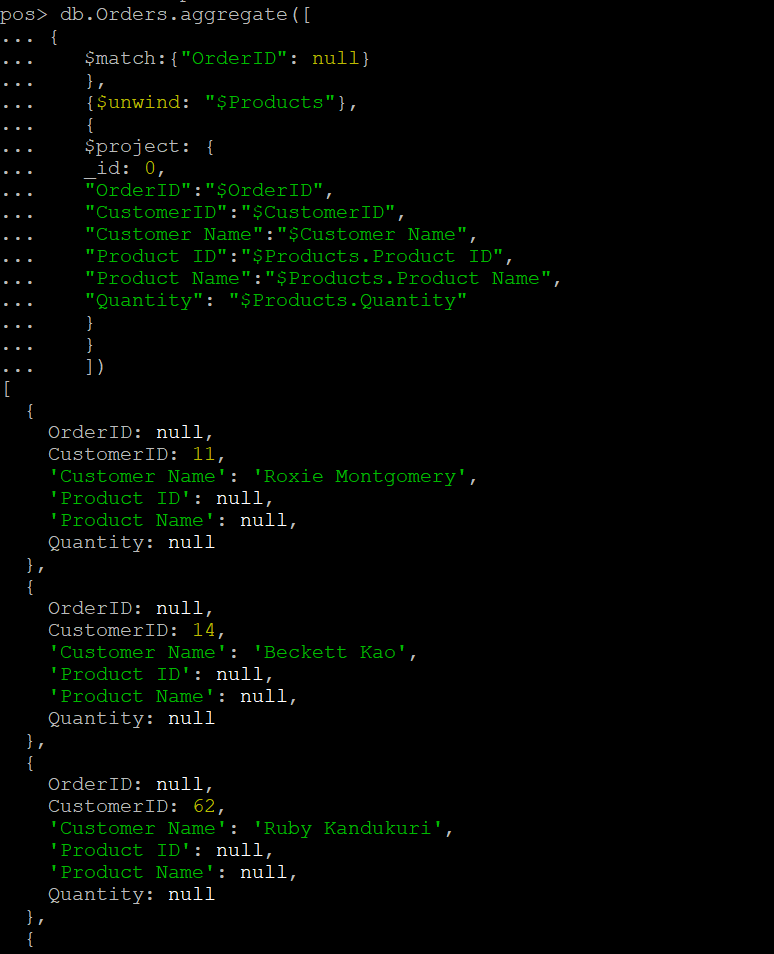


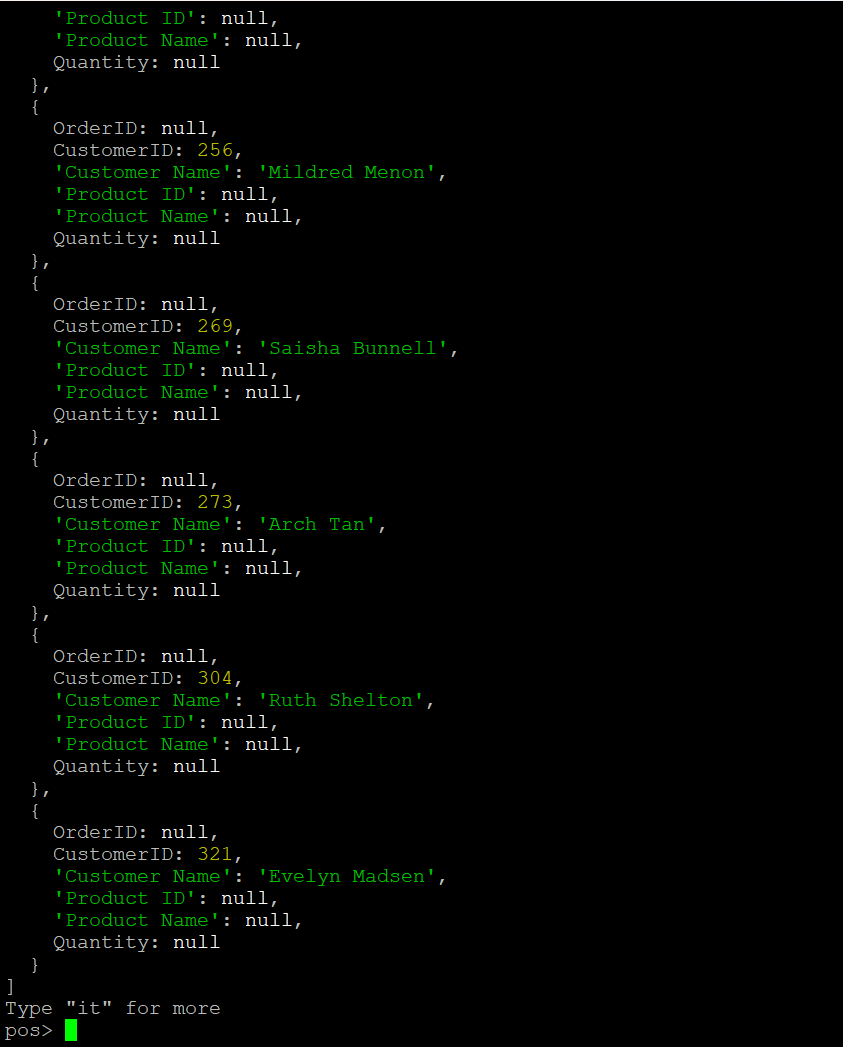
**Q) Explain your query and each part of it, including why you chose the collection you used.**

I have selected Products collection because the products collection contains products information and the customer details who bought those products. It will be easy to recall / list down the details of the customers who bought a particular product. I have used $match to filter documents based on a specified condition. In this scenario, I’m recalling all the customers who bought product – 114. So in the match condition I have given the ProductID as 114 to list down the customers who purchased the product.

I have used $unwind to deconstruct the Customers array and create separate document for each element in the array. I have used $project to project a new set of fields for each document. In this case I’m displaying the ProductID, Name, Current Price, Customer Name and Customer ID information for each document.

1. **Do any of your orders look fraudulent?**

****

****

**Q) Explain your query and each part of it, including why you chose the collection you used**

I have selected Orders collection because it contains Order, customer and the products details of that particular order purchased by customer. So it will be easy for me to check which orders are fraudulent. In my opinion, if any order does not contain OrderID and the products information of that particular order then those will be considered as fraudulent orders. According to that I have written a query. I have used $match to filter documents based on a specific condition. In this case, it filters documents whose OrderID is null. I have used $unwind to deconstruct the Products array and to create separate document for each element in the array. After that I have used $project to project the document in the following format, in this scenario it projects OrderID, CustomerID, Customer Name, Product ID, Product Name and the Quantity fields information.

The pattern that I observed is that if a particular order’s OrderID is null then it doesn’t contain any products in it. So these orders look fraudulent to me for a particular customer.

1. **Answer the following question:**
2. **Which is easier, SQL or MongoDB, and why?**

SQL databases are good for structured data with well-defined relationships between tables whereas mongodb is designed for semi-structured or unstructured data in JSON like format. Both of them have their own pros and cons. For me, mongodb looks easier. For instance, to get the data from multiple tables using SQL was a bit difficult due to the relationships between tables and using JOINs to travel across tables. But mongodb looks flexible in storing the information as we can store multiple records using array function inside of a particular collection and can retrieve the data very simply. For example, we can store all customers information who bought a particular product inside Products collection. But when it comes to the Query language, SQL looks easy compared to MongoDB as SQL includes straight forward, user friendly key words to perform operations.

1. **How is processing data in MongoDB different from SQL?**

Processing data in MongoDB is different from SQL due to the difference in the data models and quey languages. In SQL (Relational Model), the data is organized into tables with rows and columns and relationships are established through foreign keys. Whereas in MongoDB (NoSQL, Document-Oriented), data is organized into collections of documents in JSON format. Relationships can be embedded within the documents or established using references. In MongoDB, schema is flexible, allowing dynamic documents within a collection.

1. **What is the role of de-normalization in this database? Use specific examples.**

De-normalization played a key role in this database. It really improved the performance of the reads in the database. Some examples for the de-normalization in the database are

1. Customer information is available in all collections. Such as CustomerID, Customer Name and Address is available in Customers, CustomerOrders, Orders and Products Collections.
2. Products information is available in multiple collections. Such as Orders, Products, CustomerOrders collections.
3. Orders information such as OrderID is available in multiple collections. Orders and CustomerOrders collections contains the OrderID information.

Having duplicate data in multiple collections can occupy / take more space on disk but it really improves the Query performance. Due to this de-normalization, I’m able to create queries or solve multiple business related problems easily.