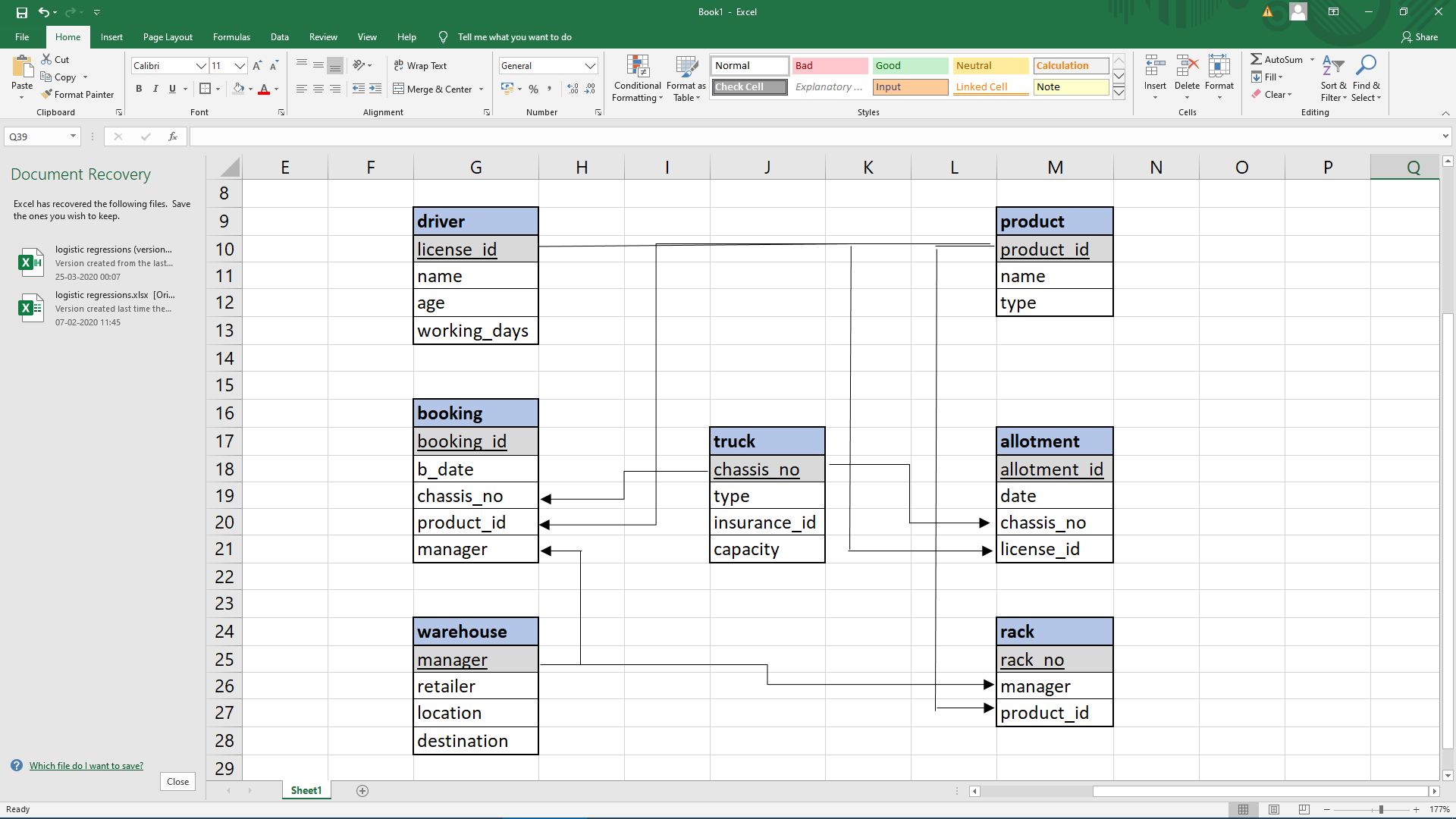
INTERCITY SUPPLY CHAIN PORTER SYSTEM

In the current supply chain market, the warehouse managers find it difficult to arrange a transport service for their goods because of the increased number of distribution centers and warehouses across the prime hubs. Even if they do, it becomes difficult to coordinate with the truck services to get the appropriate truck for their goods and then to get a driver for the same.

We aim to develop an integrated system that will keep a track of the trucks that were hired by the particular warehouse manager and the driver allotted to each truck. It will avoid the discrepancies and at the same time make sure that the correct product type gets the appropriate truck when ordered by the warehouse manager.

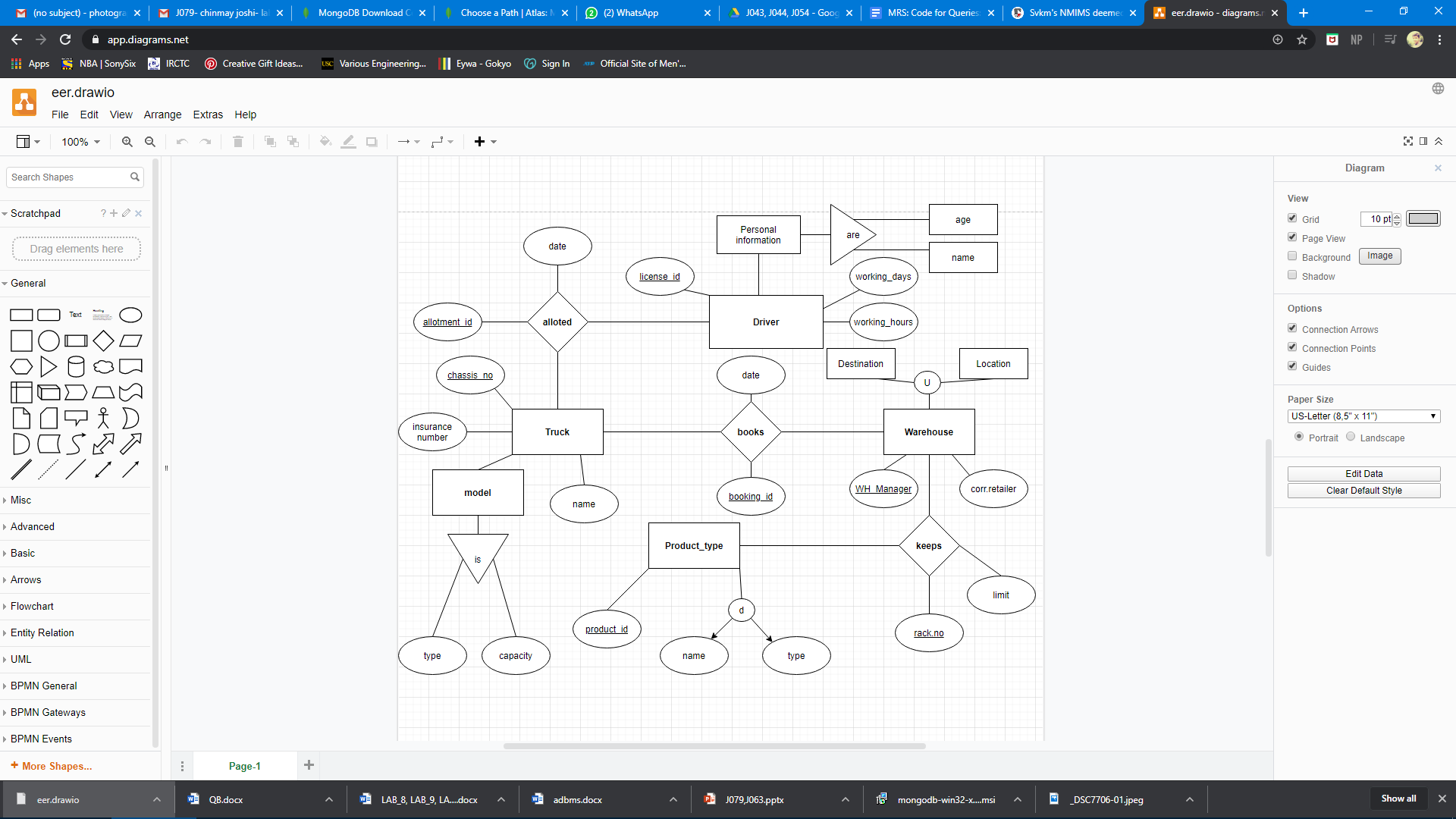
UML diagram



It consists of 4 major entities, i.e. driver, product, truck, warehouse and are related by allotment, rack and warehouse. The manager **books** (booking) a truck; a driver is **allotted** (allotment) to a truck.

At the same time, a product is **kept** in the rack according to the productid and the same product is loaded in the truck.

Enhanced ER diagram



Data insertion

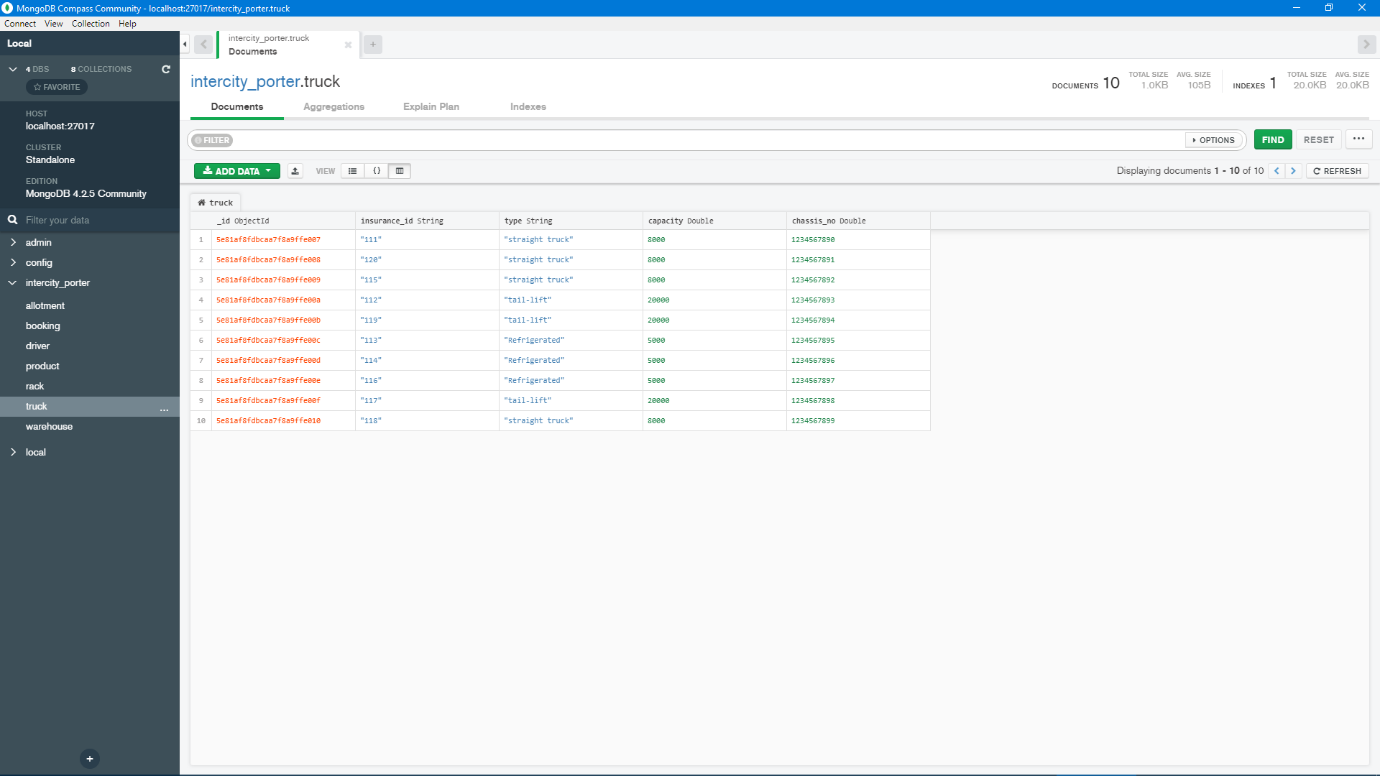
Software used: MongoDB (mongoDB compass and command prompt)

Initially collections were directly created in compass and later data was inserted using the following code:

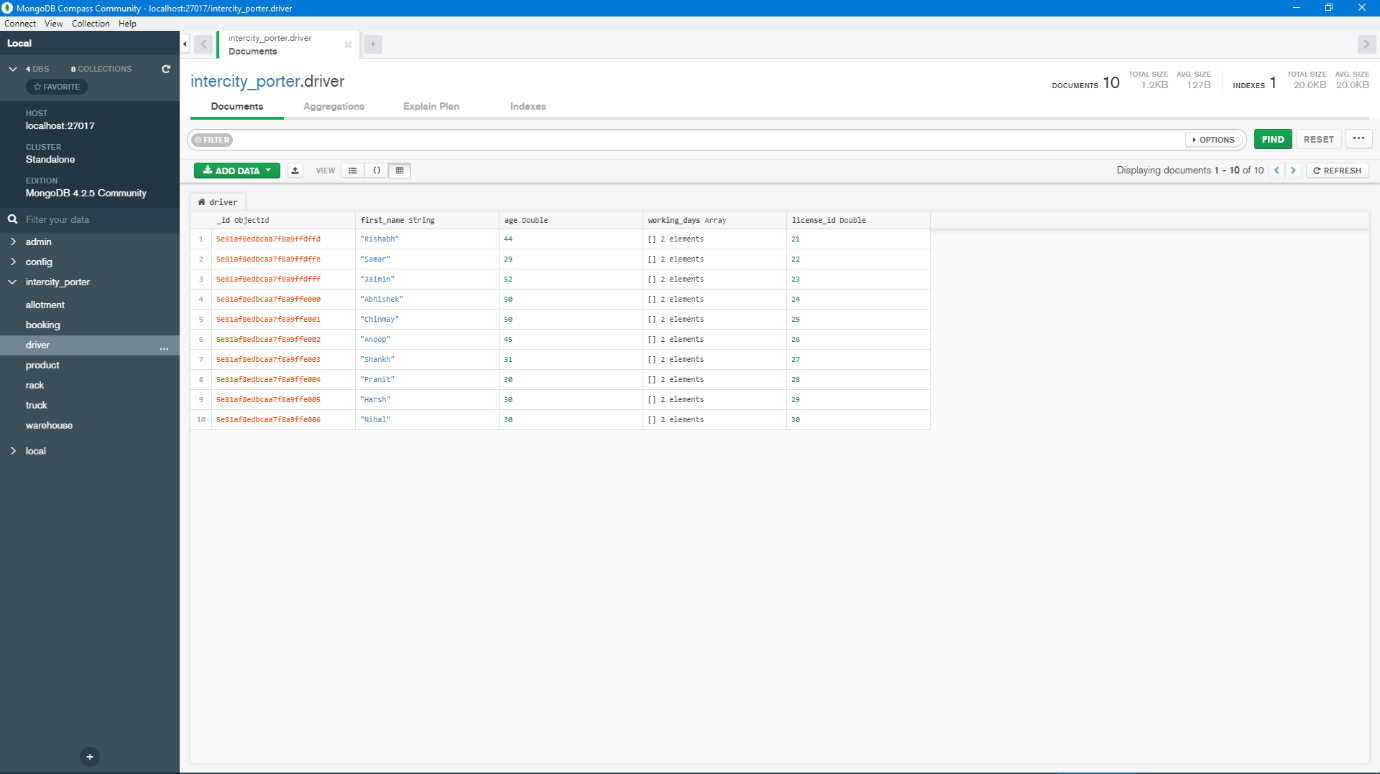
<https://docs.google.com/document/d/1u6CyyO6xInyybG1Uq_3hbg6R0asgKoyTalYG2MrIBiQ/edit?usp=sharing>

The final tables created were:

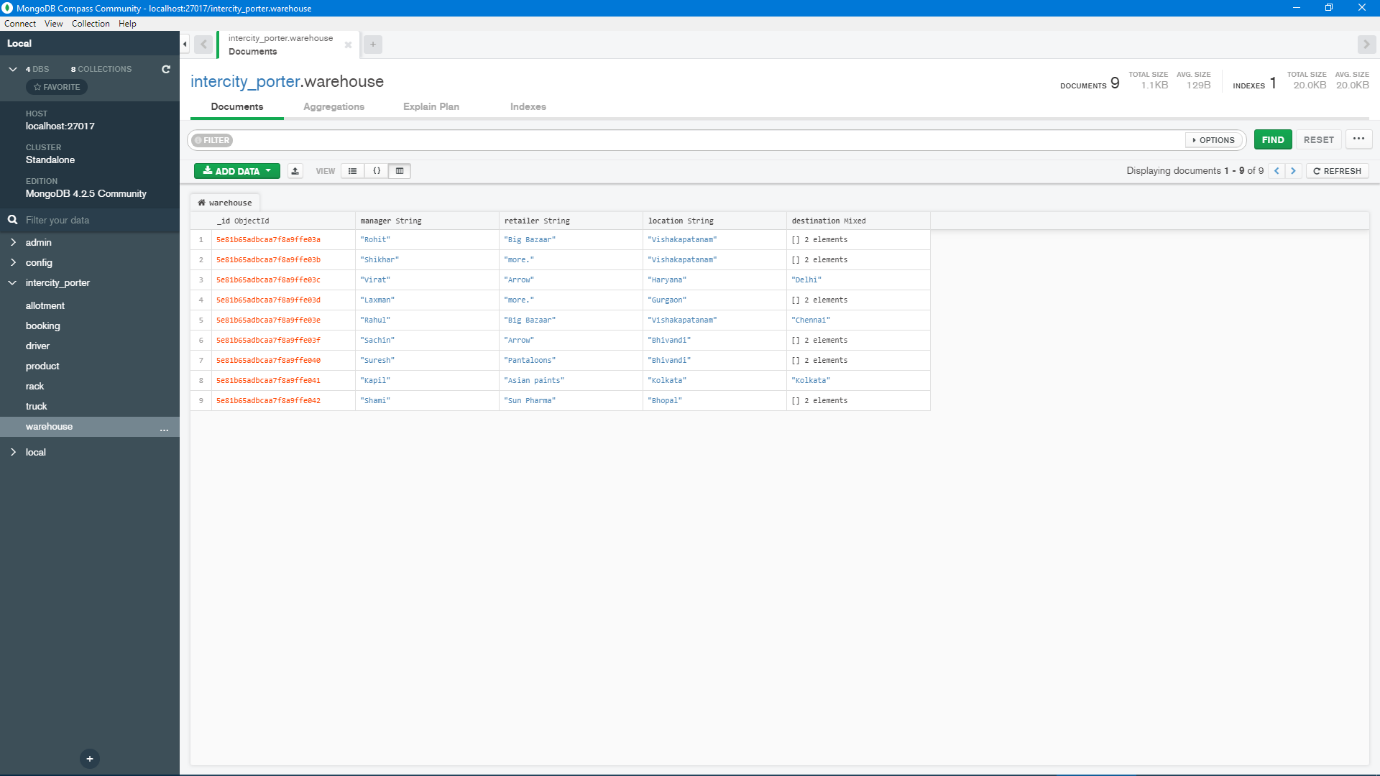
**Truck:**



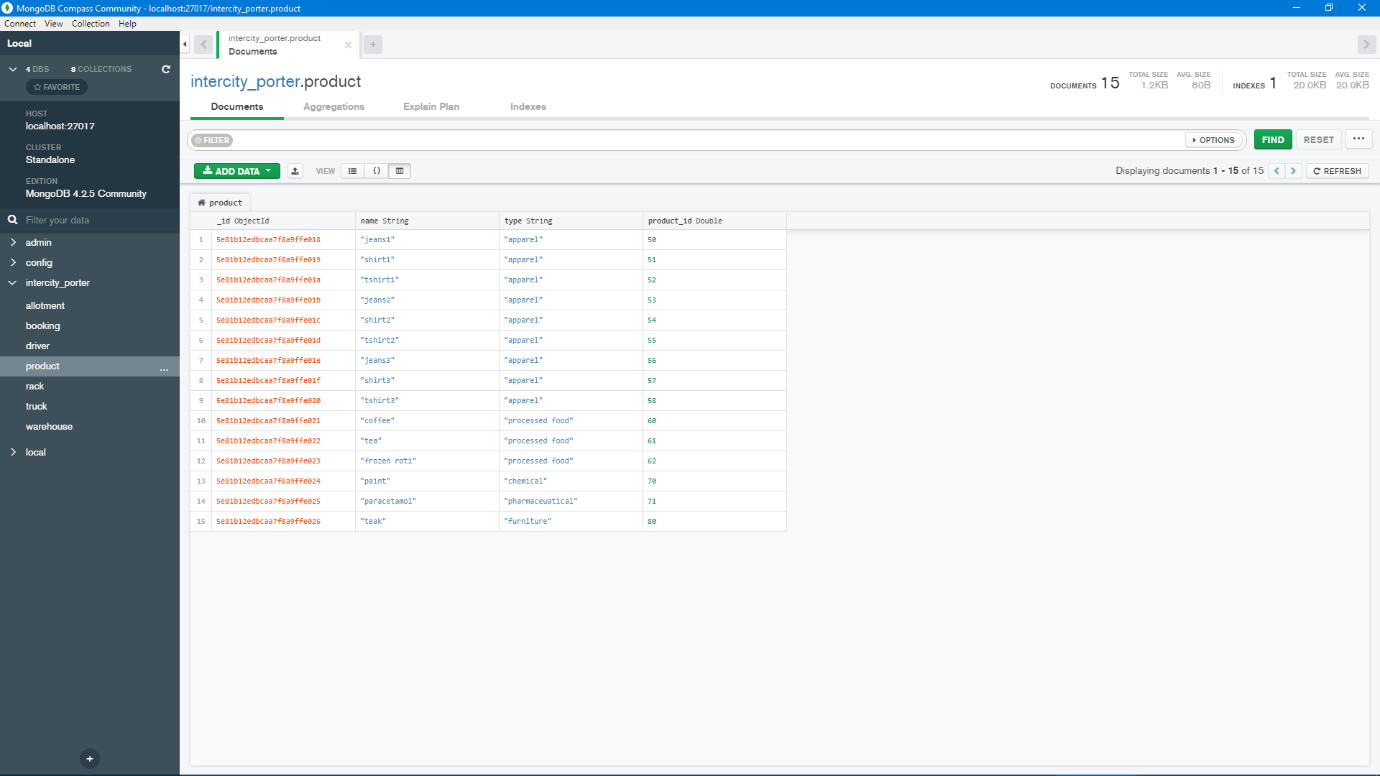
**Driver:**



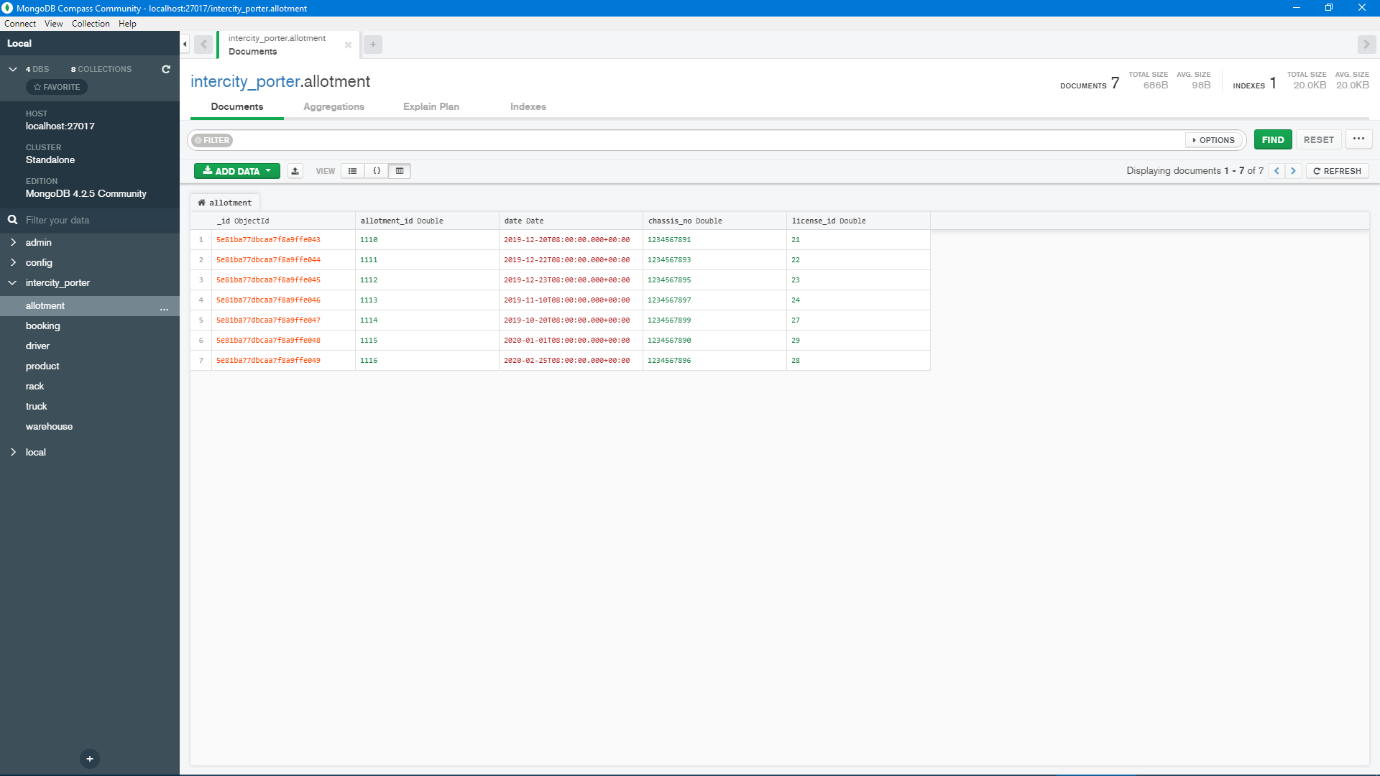
**warehouse:**

****

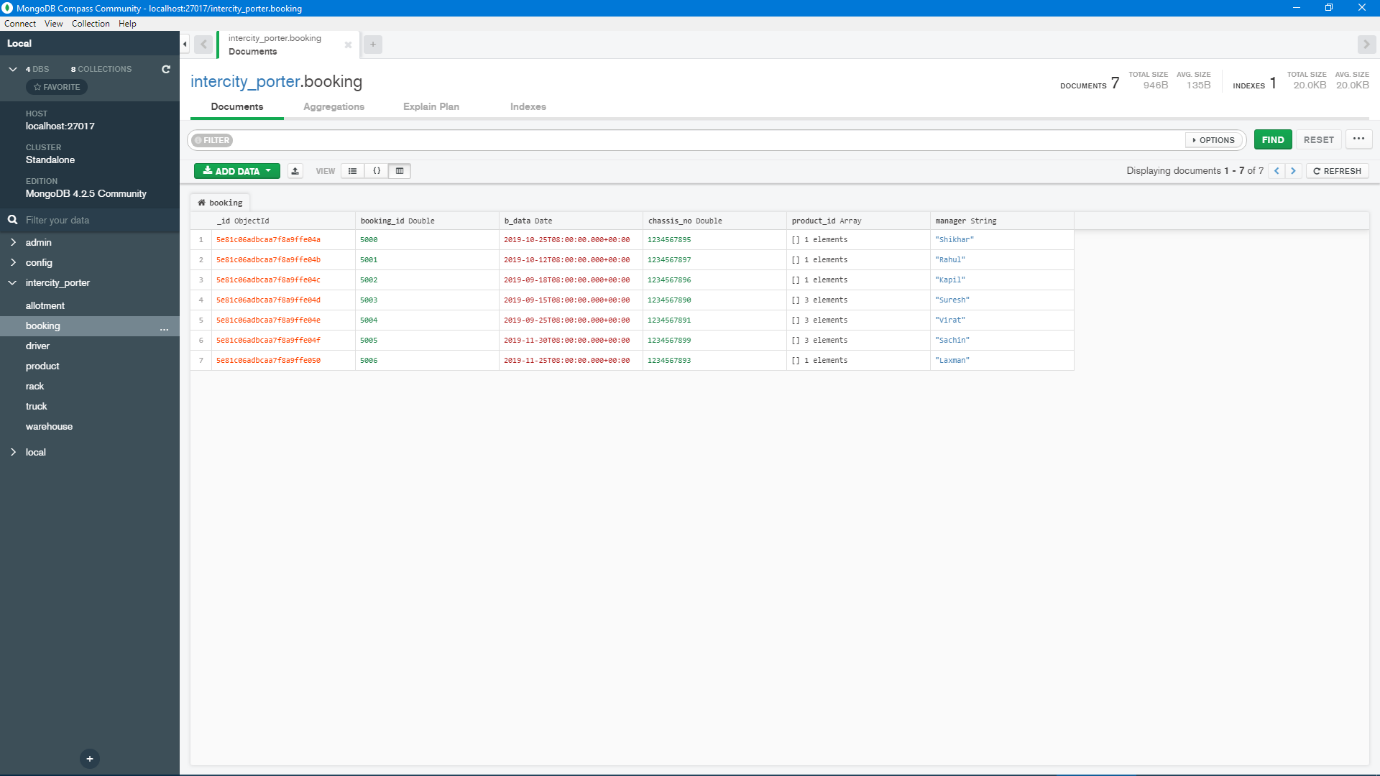
**product:**



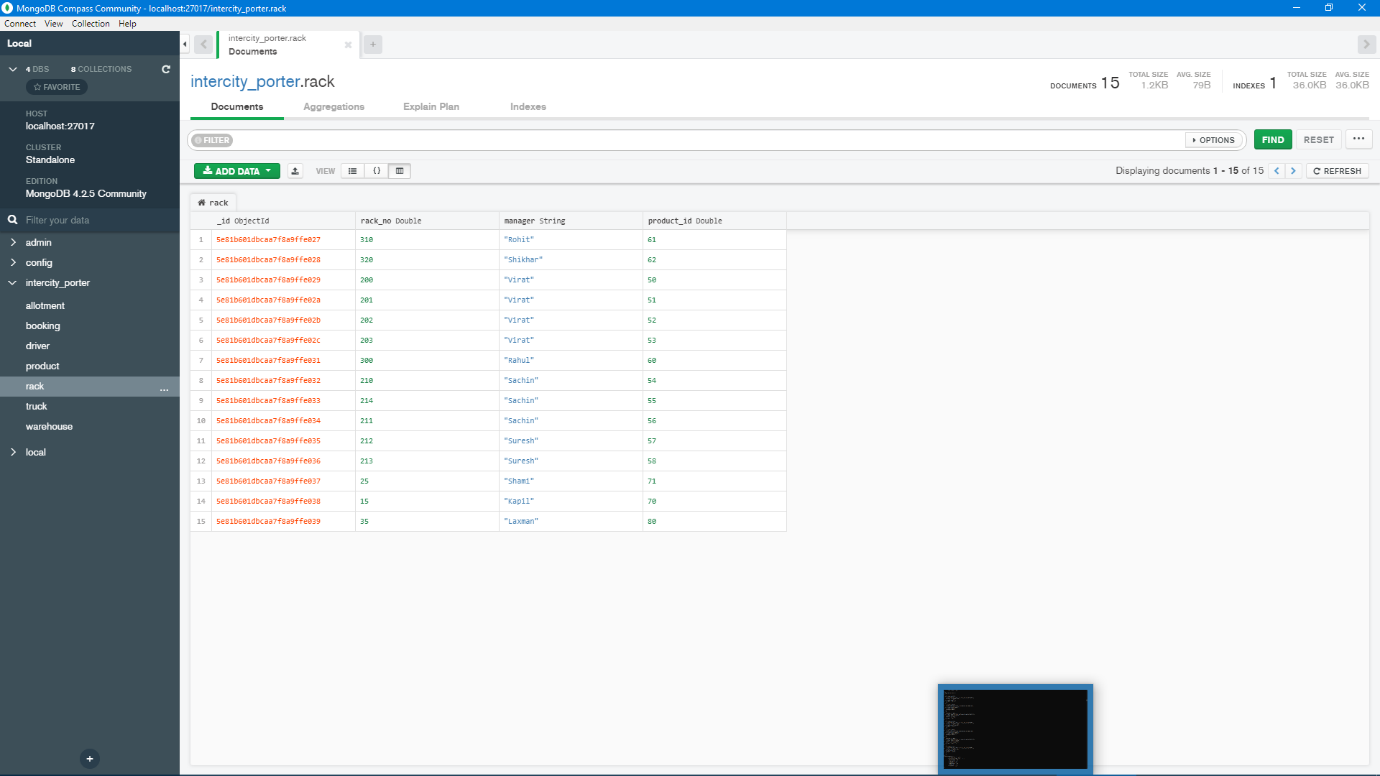
**Allotment:**



**Booking:**



**Rack:**



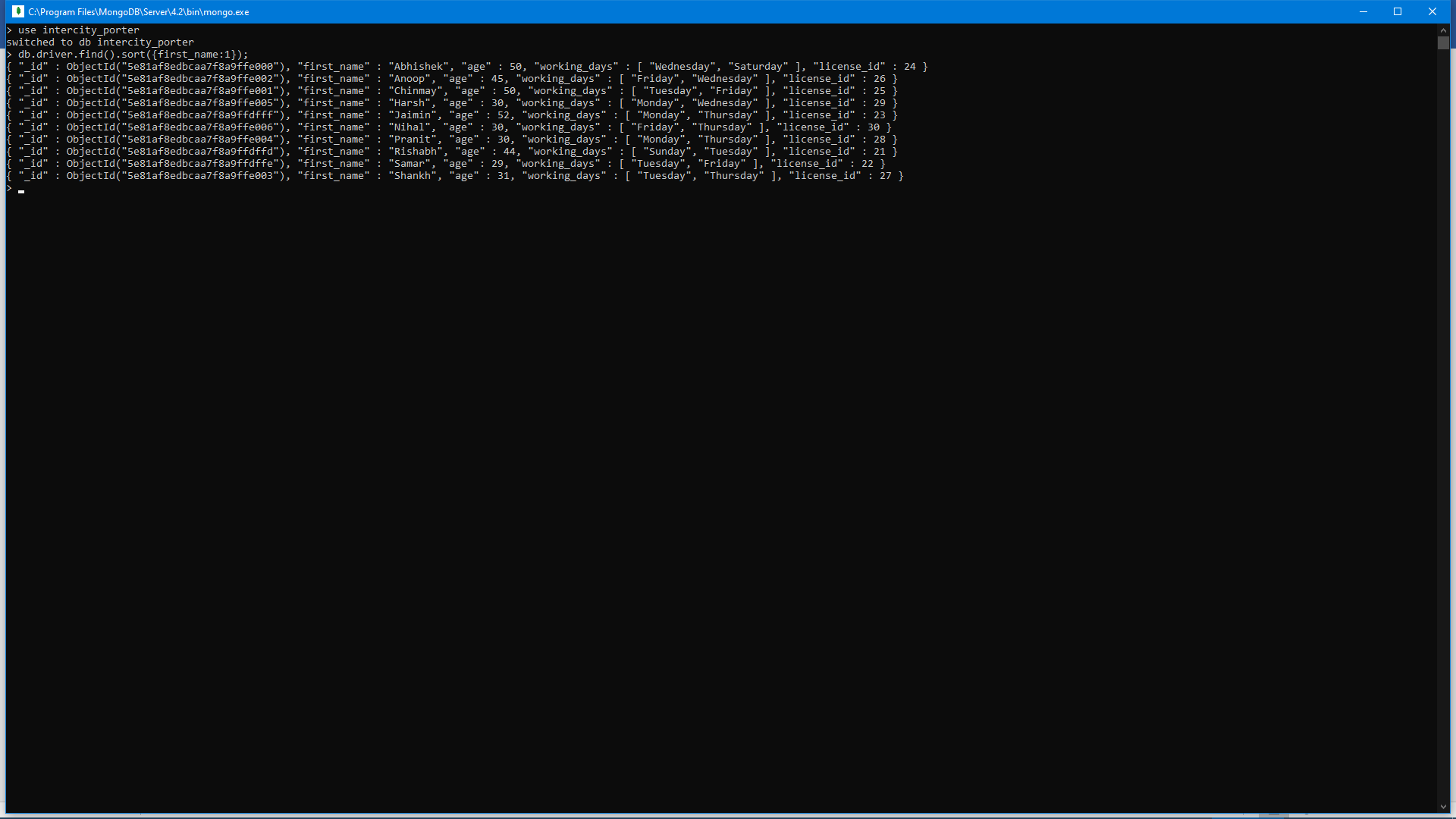
Query implementation

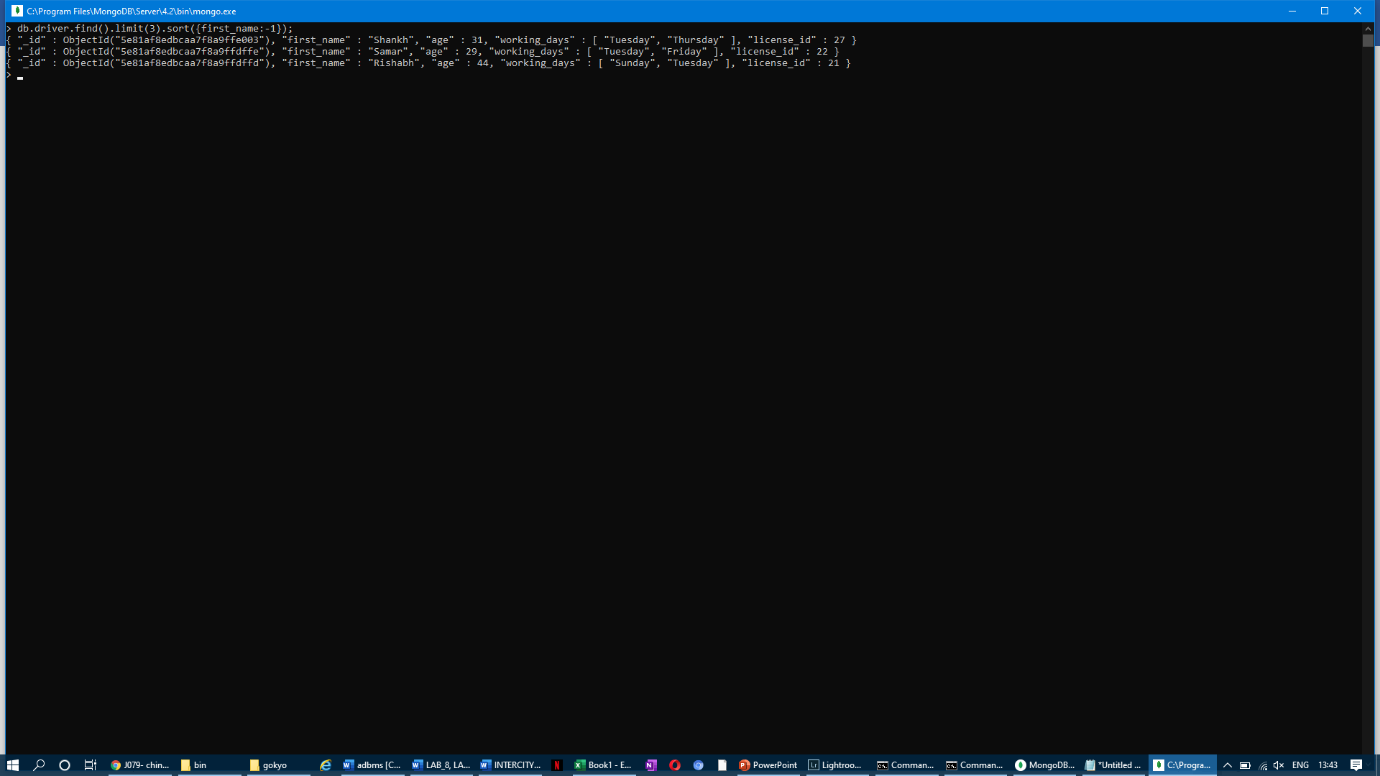
**Query 1:**

db.driver.find().sort({first\_name:1});

This query will sort the driver’s data in ascending order. If we use -1 instead of 1, it will provide us the same data in descending order. We can also limit the number of entries. The following code will explain these two variations:

db.driver.find().limit(3).sort({first\_name:-1});



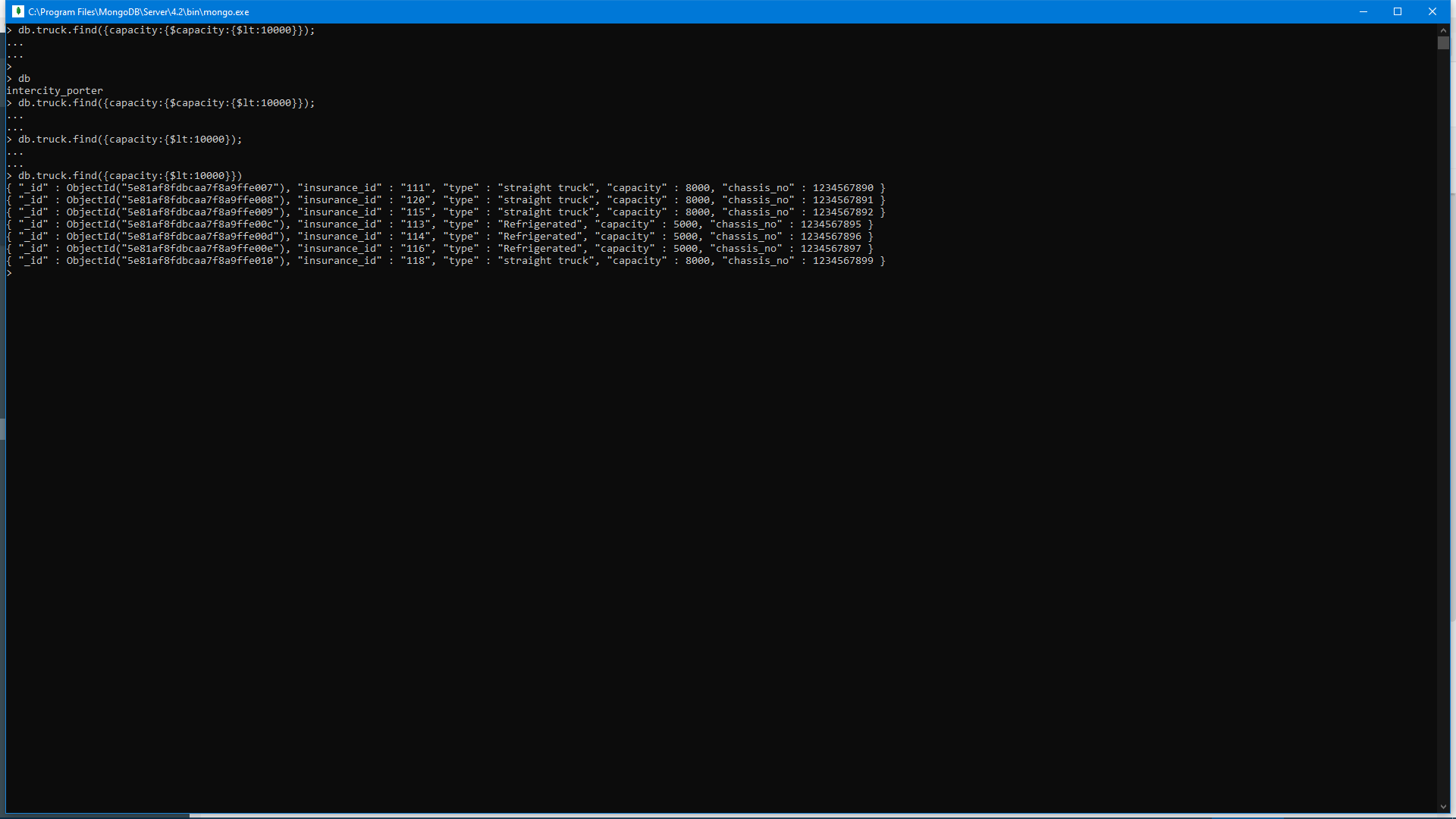


**­**

**Query 2:**

db.truck.find({capacity:{$lt:10000})

The above query iterates through the capacity field of the truck collection and tells us which truck has a capacity less than 10000. It will be useful when we have a total product load of certain kg, and we have to check which truck will be the most suitable.



**Query 3:**

db.allotment.aggregate(

[

{

$lookup:

{

from:"driver",

localField:"license\_id",

foreignField:"license\_id",

as:"lid"

}

},

{

$unwind:

{

path:"$lid",

preserveNullAndEmptyArrays:false

}

},

{

$sort:

{

"lid.license\_id":-1

}

}

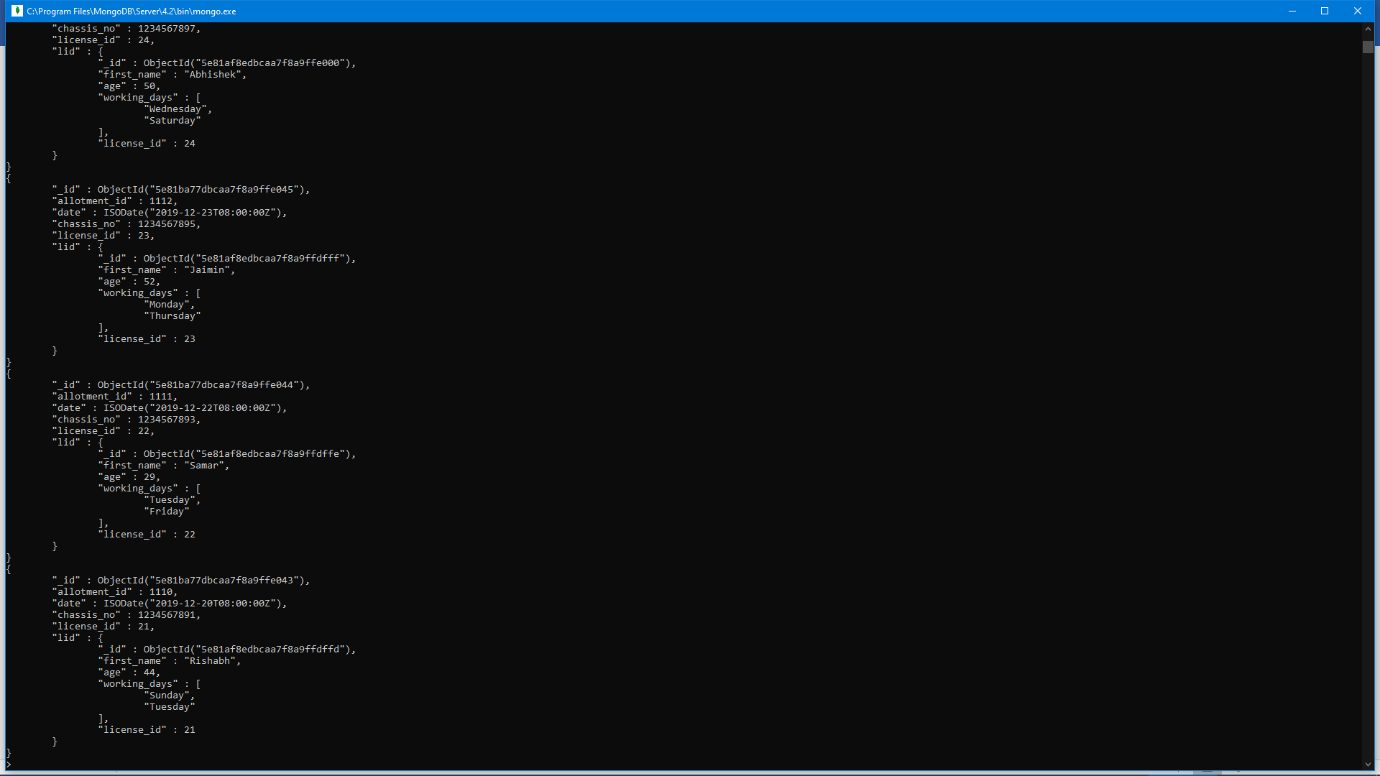
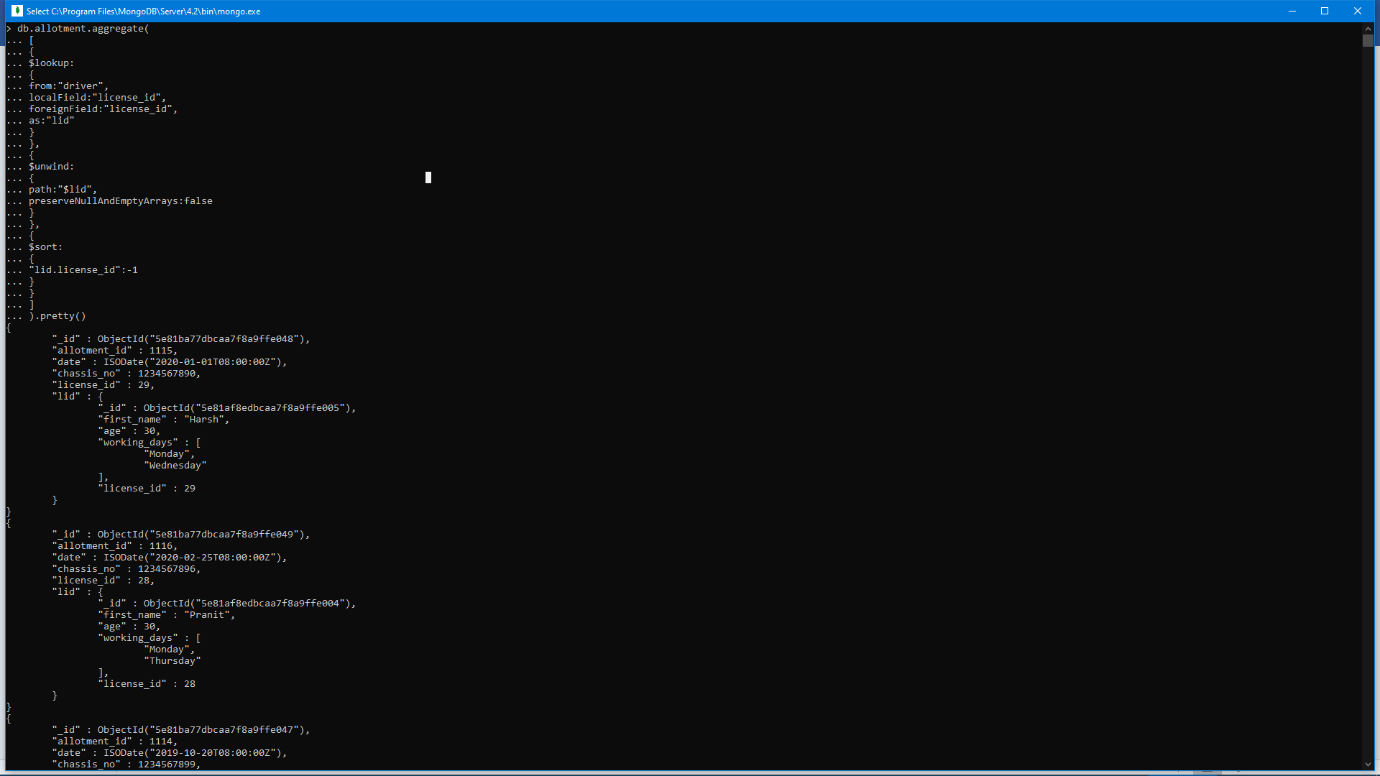
]

).pretty()

The above query is a complex query in which we have used 3 function, i.e. lookup, unwind and sort. The lookup function is used to get the data from two different collection. unwind deconstructs the arrays and the sort (-1) sorts the complete data from two collections in descending order.

We have also used a pretty() function which makes the output more aesthetic. The above query could be useful when the admin needs to know which driver was allotted to which truck, as it contains the driver’s data as well as the allotment data which includes the chassis number.

The use of $project in addition to the previous query could help in inclusion and exclusion of certain fields.



**Query 4:**

db.driver.find(

{

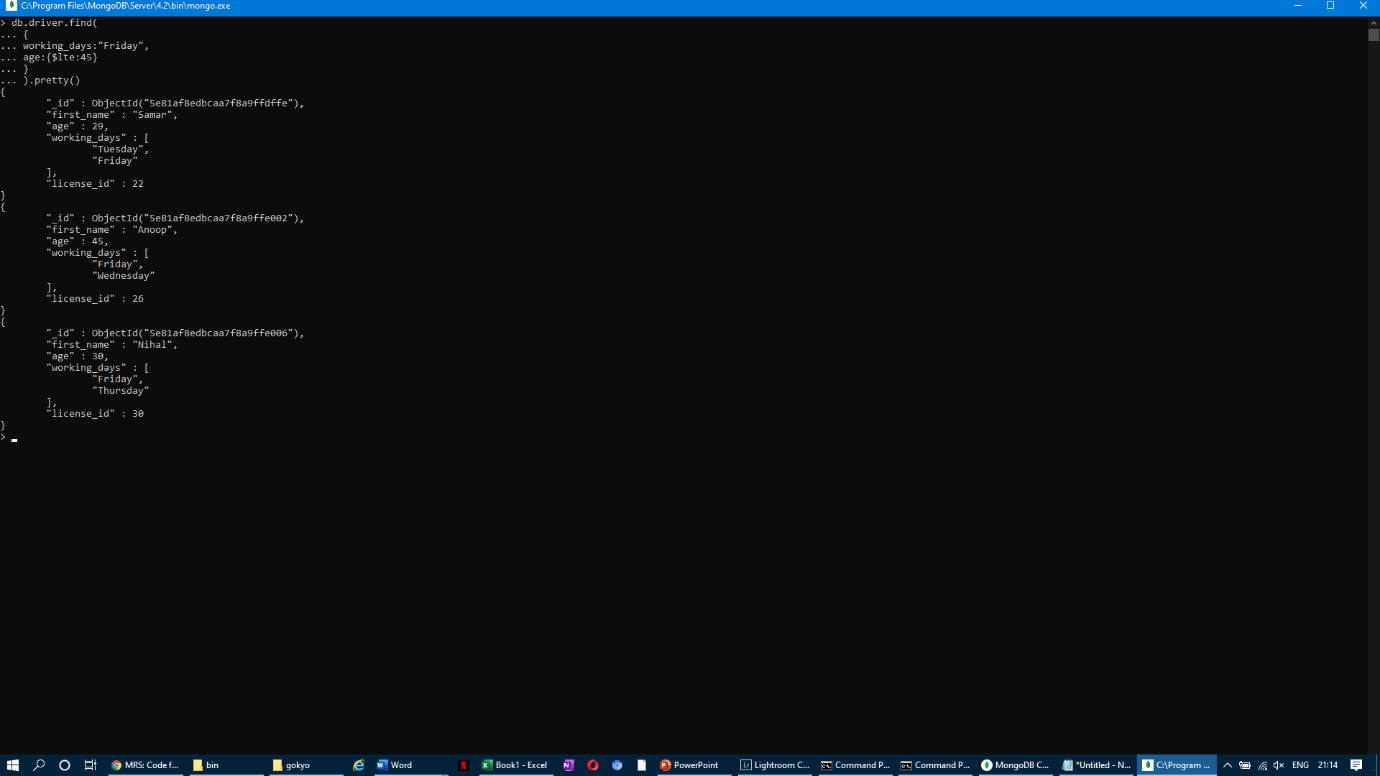
working\_days:"Friday",

age:{$lte:45}

}

).pretty()

The above query finds data mentioned in the parenthesis. It is capable of finding two data based on two different fields. Like in this case, suppose the admin wants to know which driver under the age of 45 will be available on a Friday, he or she can run this query.



# Future scope

In the complete supply chain, an integrated transport/porter network is a huge is boon. It makes it easy for the supply chain head to track the movements of the products as well as find out the flaws when anything goes wrong. It counts to a small but a very vital part of planning and executing the smooth transport of the desired products from the warehouse to the stores.

Right now, supply chain industry lacks a complete system which connects the warehouse, retailer, driver and the truck. This project provides us with an integrated system that gives us all the desired data by firing simple queries.

# Conclusion

Hence, we have successfully developed an intercity supply chain porter system which connects and tracks the warehouse (manager), truck services and driver services. With firing simple queries, we can extract relevant data regarding the above entities helping us to track and solve any problems regarding the same.

Regards,

Anup Mishra: +918369559773

Chinmay Joshi: +919619823358

THANK YOU