**Report**

DBMS EndTerm

Introduction:

The system described in this report is a e-commerce platform for selling and purchasing products online. The

platform is designed to provide a seamless shopping experience for customers and a user-friendly interface for vendors to manage

their products and sales. The system includes product categorization, search functionality, customer

and vendor account management, payment processing, order management, and shipping and delivery options.

The system is built using a database management system to store and organize information about products, customers, vendors,

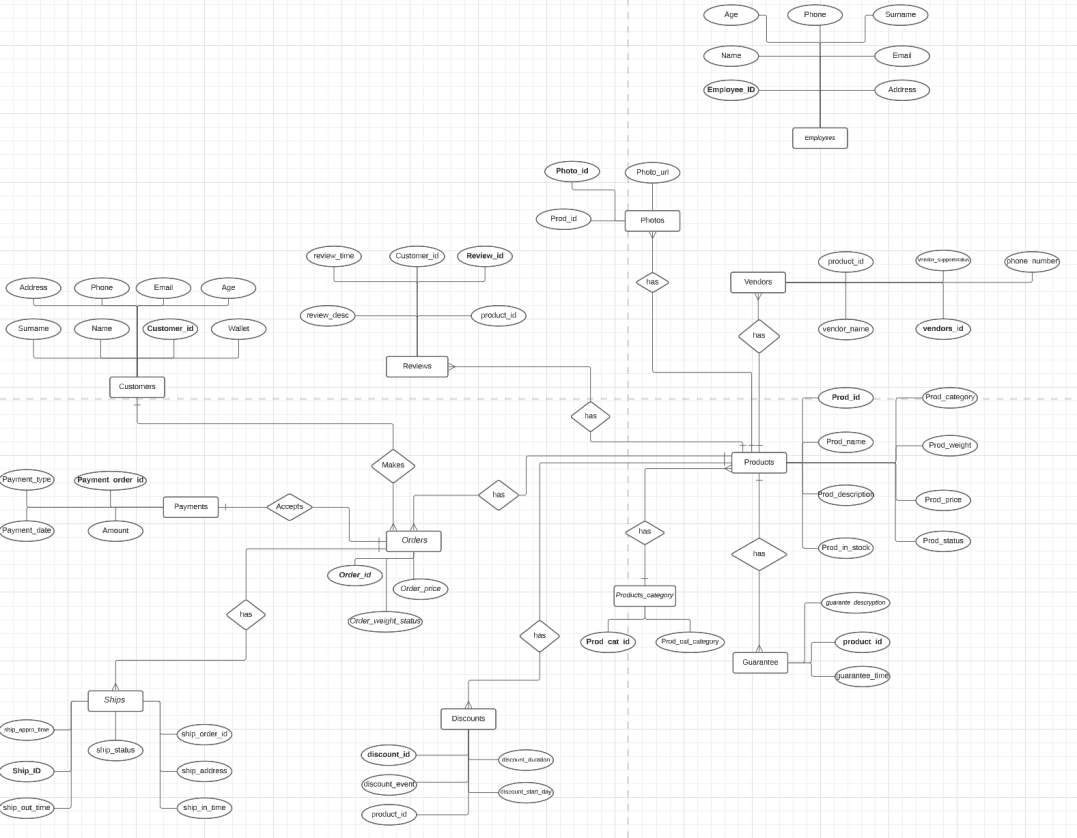
orders, discounts, reviews, photos, guarantees, and shipments. The data is structured according to the principles of relational

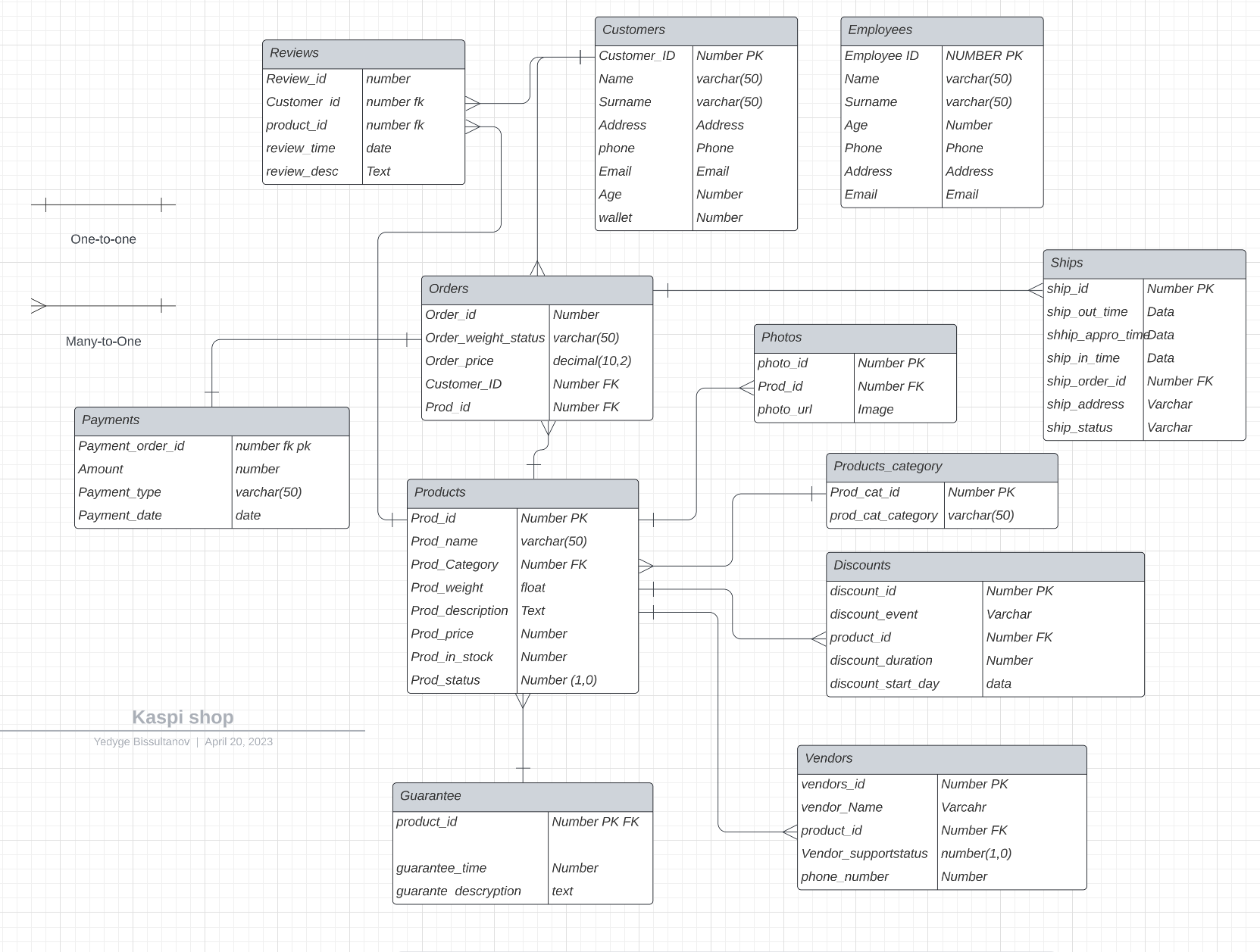
database management systems, normalized and satisfies the requirements of the third normal form.

E-commerce platform described in this report is a comprehensive system that provides a seamless and efficient shopping

experience for customers and a powerful tool for vendors to manage their products and sales.[Link to GitHub](https://github.com/B1ssultanov/Midterm_DBMS_2course)

ER Diagram:





1.Let's analyze the MP2\_Products\_category table to determine whether it satisfies the third normal form (3NF).

In the MP2\_Products\_category table, the primary key is Prod\_cat\_id, and the only non-prime attribute is prod\_cat\_category. As

can be seen, prod\_cat\_category is dependent solely on the primary key, which means that the table satisfies the 3NF.

This ensures that data redundancy is minimized, and updates, deletions, and insertions are carried out without anomalies. It

also facilitates the querying of data and the development of efficient database management systems.

2.The MP2\_Products table represents the products available for purchase on the e-commerce platform. It has the following

attributes:

The Prod\_id attribute is the primary key of the table, ensuring that each product has a unique identifier. The Prod\_Category

attribute is a foreign key that references the MP2\_Products\_category table, ensuring that each product belongs to a category

that exists in the category table.

The Prod\_name, Prod\_weight, Prod\_description, Prod\_price, Prod\_in\_stock, and Prod\_status attributes are all atomic and are not

multi-valued. Therefore, the table satisfies the first normal form.

The table does not contain any repeating groups, as each product is represented by a single row. Therefore, the table satisfies

the second normal form.

The Prod\_category attribute is functionally dependent on the Prod\_id attribute and is not dependent on any other non-key

attributes. Therefore, the table satisfies the third normal form.

3.The MP2\_Employees table stores information about the employees of the company. It contains the following attributes:

Employee\_ID, Name, Surname, Age, Phone, Address, and Email.

The Employee\_ID attribute is the primary key of the table, and it uniquely identifies each employee. The Name and Surname

attributes store the first and last name of each employee, respectively. The Age attribute stores the age of each employee. The

Phone attribute stores the phone number of each employee, while the Address attribute stores the physical address of each

employee. Finally, the Email attribute stores the email address of each employee.

This table satisfies the first normal form because it has a primary key (Employee\_ID), and all attributes are atomic, meaning

they cannot be further divided. It also satisfies the second normal form because all non-key attributes (Name, Surname, Age,

Phone, Address, and Email) depend solely on the primary key (Employee\_ID).

To demonstrate that the table satisfies the third normal form, we need to ensure that there are no transitive dependencies. A

transitive dependency occurs when a non-key attribute depends on another non-key attribute. In this table, we can see that there

are no such dependencies. All non-key attributes depend solely on the primary key (Employee\_ID). Therefore, the MP2\_Employees

table satisfies the third normal form.

4.The MP2\_Customers table contains information about customers, such as their name, address, phone number, email address, age,

and wallet. This table satisfies the 3NF because it meets the requirements of 1NF and 2NF, and there is no transitive dependency

between non-key attributes.

For 1NF table has a primary key (Cust\_ID) that uniquely identifies each customer. Each attribute contains only atomic values,

and there are no repeating groups.

For 2NF All non-key attributes (Cust\_Name, Cust\_Surname, Cust\_Address, Cust\_Phone, Cust\_Email, Cust\_Age, and Cust\_Wallet) are

functionally dependent on the primary key (Cust\_ID).

For 3NF There no transitive dependencies between non-key attributes. For example, Cust\_Address is not dependent on Cust\_Phone or

Cust\_Email.

The MP2\_Customers table satisfies the 3NF, which means it is well-designed and can avoid data inconsistencies and anomalies.

5.The Discount table has the following attributes:

Discounts\_id: a unique identifier for each discount

Discount\_event: the name or description of the discount event

Product\_id: the ID of the product that the discount applies to

Discount\_start\_day: the date when the discount event starts

Discount\_duration: the duration of the discount event

This table satisfies the first normal form (1NF) because it has a primary key and there are no repeating groups or arrays. Each

attribute has a single value and each row is unique.

It satisfies the second normal form (2NF) because all non-key attributes are functionally dependent on the primary key, which is

the Discounts\_id. The Discount\_event, Product\_id, Discount\_start\_day, and Discount\_duration are all dependent on the primary key.

It satisfies the third normal form (3NF) because there are no transitive dependencies. All non-key attributes are only dependent

on the primary key, and there are no dependencies between non-key attributes. Therefore, the Discount table is in third normal

form.

6.The MP2\_Vendors table represents the vendors that supply the products in the MP2\_Products table. It contains the following

columns:

Vendors\_id: A unique identifier for each vendor.

Vendor\_Name: The name of the vendor.

Product\_id: A foreign key that references the Prod\_id column in the MP2\_Products table. It specifies which product the vendor

supplies.

Vendor\_SupportStatus: A flag that indicates whether the vendor provides support for the product.

Phone\_Number: The phone number of the vendor.

This table satisfies the first normal form because each column contains atomic values, and there are no repeating groups.

It satisfies the second normal form because it has a single-column primary key and all other columns are dependent on the

primary key. Specifically, the Product\_id column depends on the Vendors\_id column, and the remaining columns depend on both the

Vendors\_id and Product\_id columns.

It satisfies the third normal form because there are no transitive dependencies between non-key columns. The

Vendor\_SupportStatus and Phone\_Number columns are dependent only on the Vendors\_id column and not on the Product\_id column.

Thus, the table is in third normal form.

7.The MP2\_Review table is used to store product reviews submitted by customers. It has the following attributes:

Review\_id: a unique identifier for each review (primary key)

Cust\_id: a foreign key referencing the customer who submitted the review

Prod\_id: a foreign key referencing the product being reviewed

Review\_time: the date and time the review was submitted

Review\_desc: the description of the review

The table satisfies the first normal form because each attribute in each row of the table is atomic and cannot be further

decomposed.

The table satisfies the second normal form because it has a composite primary key (Review\_id, Prod\_id) that uniquely identifies

each row in the table. Additionally, all non-key attributes (Cust\_id, Review\_time, Review\_desc) are fully dependent on the

entire primary key, meaning that no partial dependency exists.

The table satisfies the third normal form because there are no transitive dependencies between non-key attributes. All non-key

attributes are dependent only on the primary key, and no other attributes in the table. Therefore, this table is also in third

normal form.

The MP2\_Review table is well-structured and satisfies the requirements of the first, second, and third normal forms.

8.The MP2\_Orders table satisfies the third normal form (3NF) as it does not have any transitive dependencies.

The table has the following columns:

order\_id (primary key)

order\_weight\_status

order\_price

customer\_id (foreign key referencing the customer\_id column in MP2\_Customers table)

product\_id (foreign key referencing the Prod\_id column in MP2\_Products table)

There are no repeating groups in the table and each column is dependent on the primary key, order\_id. There are no columns that

are dependent on non-primary key columns. Therefore, the table is in the first normal form (1NF).

The table is also in the second normal form (2NF) as there are no partial dependencies. All non-key columns depend on the

primary key, order\_id.

Finally, the table satisfies the third normal form (3NF) as there are no transitive dependencies. All non-key columns depend

only on the primary key and not on any other non-key columns.

9.The MP2\_Photos table in the database has three columns: Photo\_id, Product\_id, and Photo\_url. This table stores the URLs of

product photos and associates them with their corresponding product IDs.

To satisfy the first normal form (1NF), the MP2\_Photos table has a primary key, which is the Photo\_id column. Each row in the

table has a unique identifier, and there are no repeating groups.

To satisfy the second normal form (2NF), the MP2\_Photos table has a foreign key constraint on the Product\_id column that

references the Prod\_id column in the MP2\_Products table. This ensures that each product in the MP2\_Products table has a

corresponding photo in the MP2\_Photos table.

To satisfy the third normal form (3NF), there are no transitive dependencies in the MP2\_Photos table. All columns are

functionally dependent on the primary key, Photo\_id, and there are no other dependencies between columns. Therefore, the

MP2\_Photos table is in 3NF.

10.The Guarantee table has a primary key of Prod\_id, which is a foreign key reference to the Prod\_id column in the Products

table. This ensures that each guarantee is tied to a specific product.

The table has two other columns: Guarantee\_time and Guarantee\_description. Guarantee\_time is a timestamp indicating when the

guarantee starts, and Guarantee\_description is a description of the guarantee.

There are no repeating groups or multiple values in this table, as each guarantee is tied to a single product and has its own

unique ID. Therefore, the Guarantee table satisfies the first normal form.

There are no partial dependencies in this table, as both the Guarantee\_time and Guarantee\_description columns depend on the

entire primary key (Prod\_id). Therefore, the Guarantee table satisfies the second normal form.

Finally, there are no transitive dependencies in this table, as there are only three columns: the primary key, the guarantee

time, and the guarantee description. Therefore, the Guarantee table satisfies the third normal form.

11.The Ships table contains information about the shipping of orders. The table has a primary key column, "ship\_id", which is a

unique identifier for each shipping record. The table also has a foreign key column, "ship\_order\_id", which references the

order\_id column in the Orders table, linking each shipping record to a specific order.

The table has columns for the "ship\_out\_time", "ship\_appro\_time", and "ship\_in\_time" which represent the time when the order was

shipped out, the time when the shipping was approved, and the time when the order was received by the customer respectively.

There is also a "ship\_address" column which represents the address to which the order was shipped.

Finally, the table has a "ship\_status" column which indicates the current status of the shipment. This column can have values

like "shipped", "in transit", "delivered", etc.

The Ships table satisfies the first normal form as each column has a unique name, and each row has a unique identifier in the

form of the "ship\_id" column.

The table also satisfies the second normal form as it has a primary key, "ship\_id", which uniquely identifies each row in the

table. Additionally, all other columns are dependent only on the "ship\_id" column, and there are no partial dependencies on any

subset of columns.

Finally, the table satisfies the third normal form as there are no transitive dependencies between the columns. All columns

depend only on the "ship\_id" column, which is the primary key of the table.

ERD

1. Customers and Orders have entity relation "Places"

2. Ships and Orders have entity relation "has"

The relationship between Ships and Orders is a "has" relationship, where each order can have one or more associated shipments.

This is because an order may be fulfilled in multiple shipments, especially if the ordered items are not immediately available

or are coming from different locations.

σ Orders.OrderID = Ships.OrderID (Orders ⨝ Ships)

3. Reviews table and Products table have entity relation "has".

The relationship between the Reviews and Products tables can be expressed as "has," meaning that a product can have many

reviews, but a review can only belong to one product. This is a one-to-many relationship.

4. Discounts table and Products table have entity relation "has".

the relation between Discounts table and Products table is that a product may have multiple discounts, and a discount may apply

to multiple products. Therefore, the relation between them is "has".

5. Orders table and Products table have entity relation "has".

The relationship between the Orders table and the Products table can be expressed as "has," indicating that an order has one or

more products associated with it. This relationship can be represented in relational algebra using a join operation.

6. Guarantee table and Products table have entity relation "has".

The entity relation between Guarantee table and Products table is "has" because each product can have one or more guarantees

associated with it, and each guarantee can be associated with only one product. This is a one-to-many relationship between the

Products table and Guarantee table.

7. Products\_Category table and Products table have entity relation "has".

Products\_Category table has the attribute category\_id which is a foreign key referencing the id attribute in the Categories

table, and Products table has the attribute category\_id which is a foreign key referencing the id attribute in the

Products\_Category table.

8. Photos table and Products table have entity relation "has".

where product\_id is a foreign key that references the primary key of Products table. This relation indicates that a product can

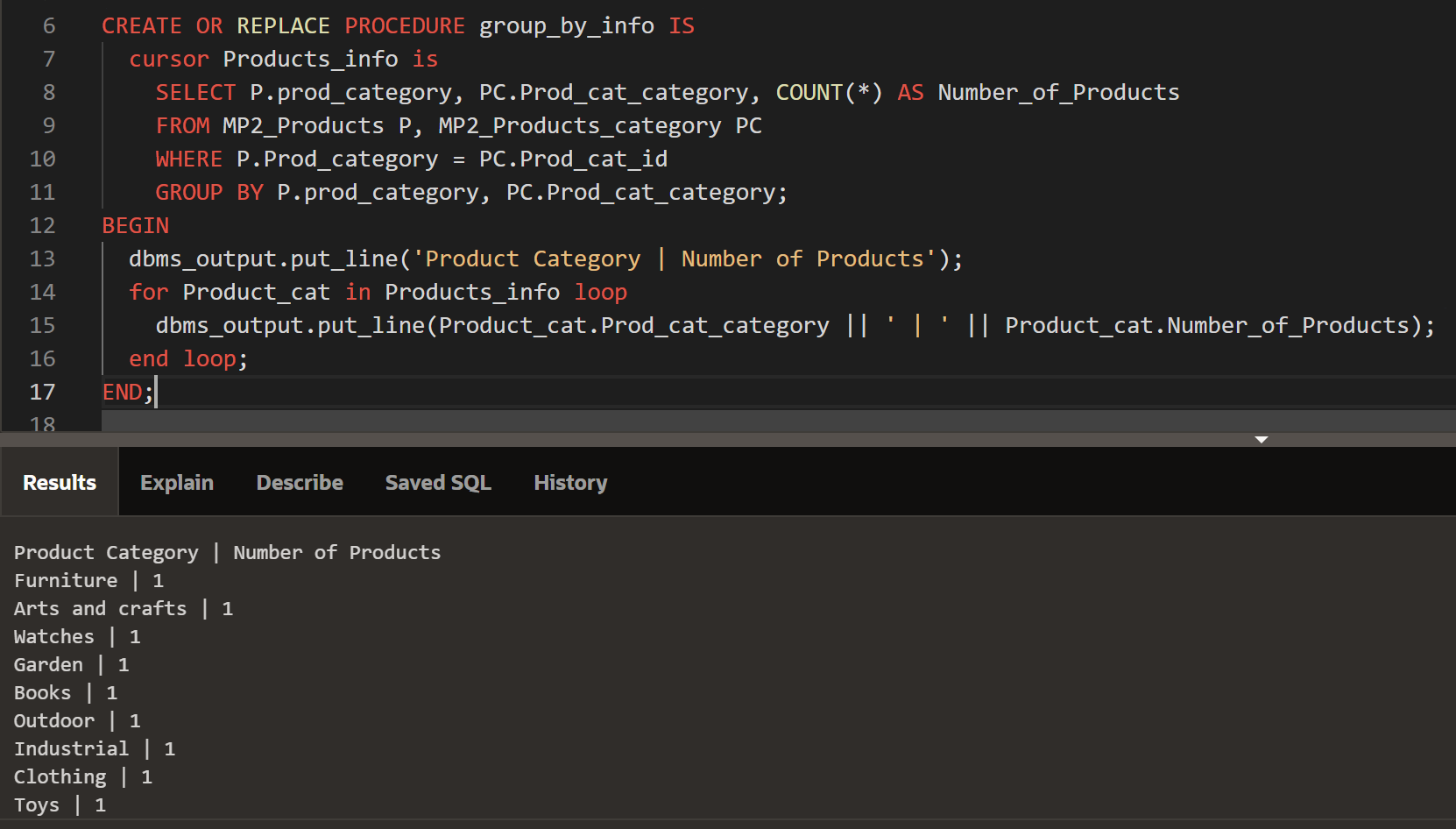
have multiple photos associated with it.

9.Vendors table and Products table have entity relation "has".

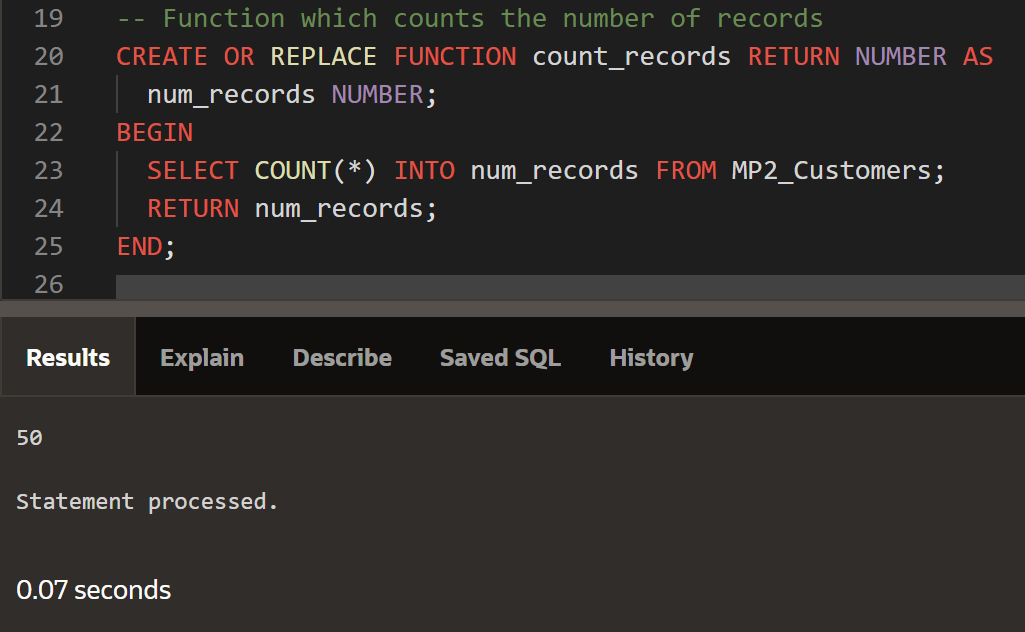
vendor can have many products, and a product can belong to only one vendor. Therefore, we can represent this relationship using

the foreign key vendor\_id in the Products table that references the primary key id in the Vendors table.

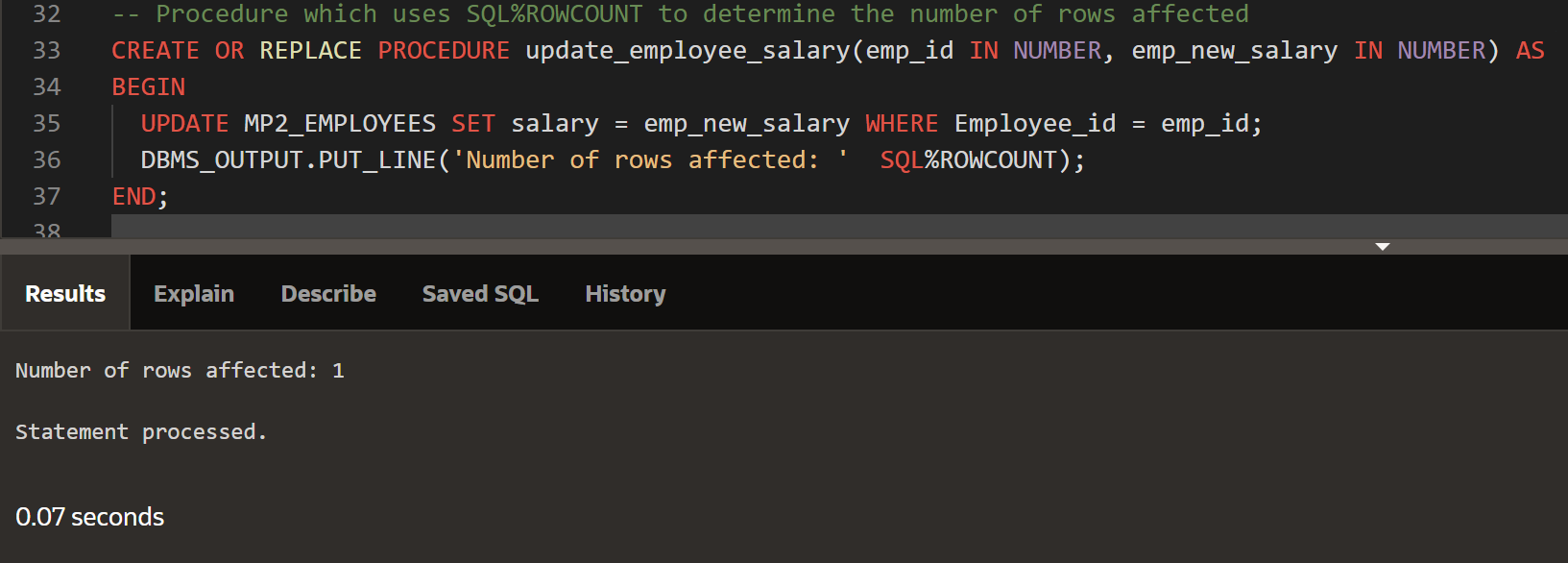
- Procedure which does group by information



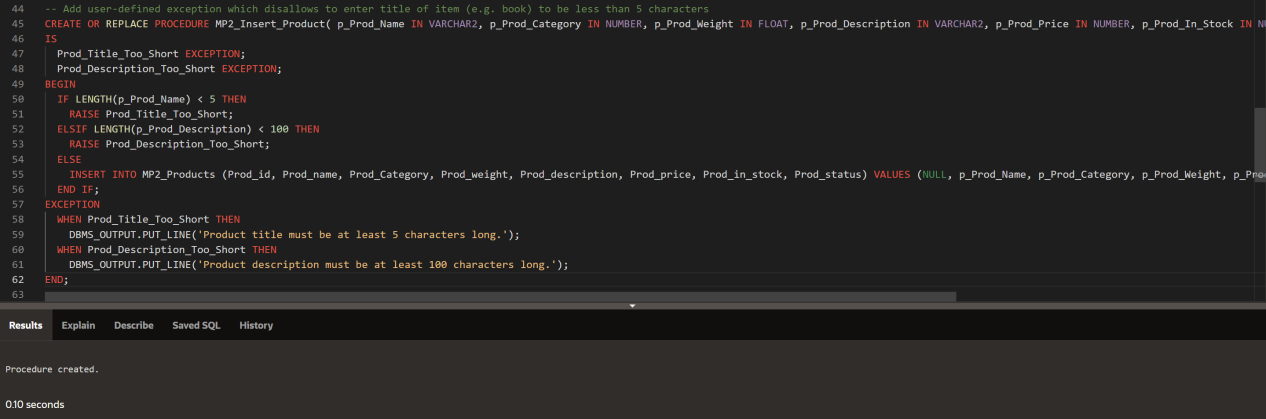
- Function which counts the number of records



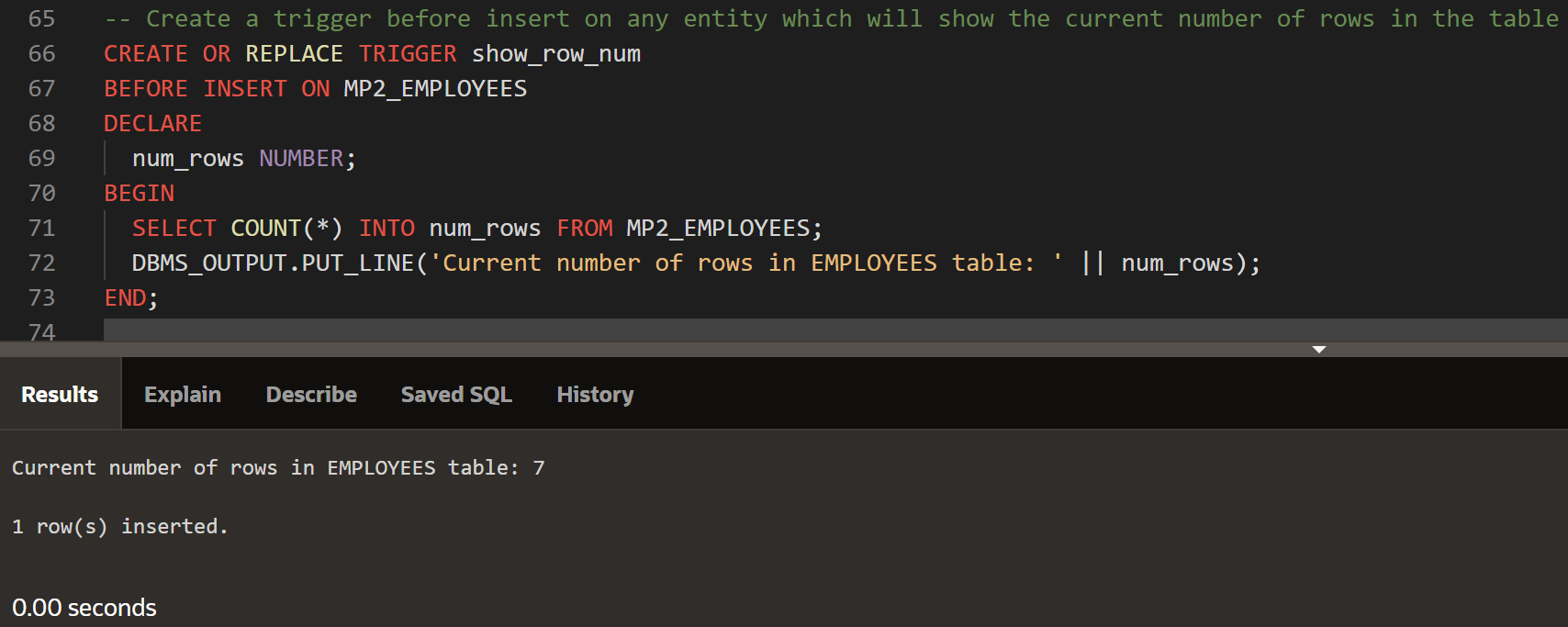
- Procedure which uses SQL%ROWCOUNT to determine the number of rows affected



- Add user-defined exception which disallows to enter title of item (e.g. book) to be less than 5 characters

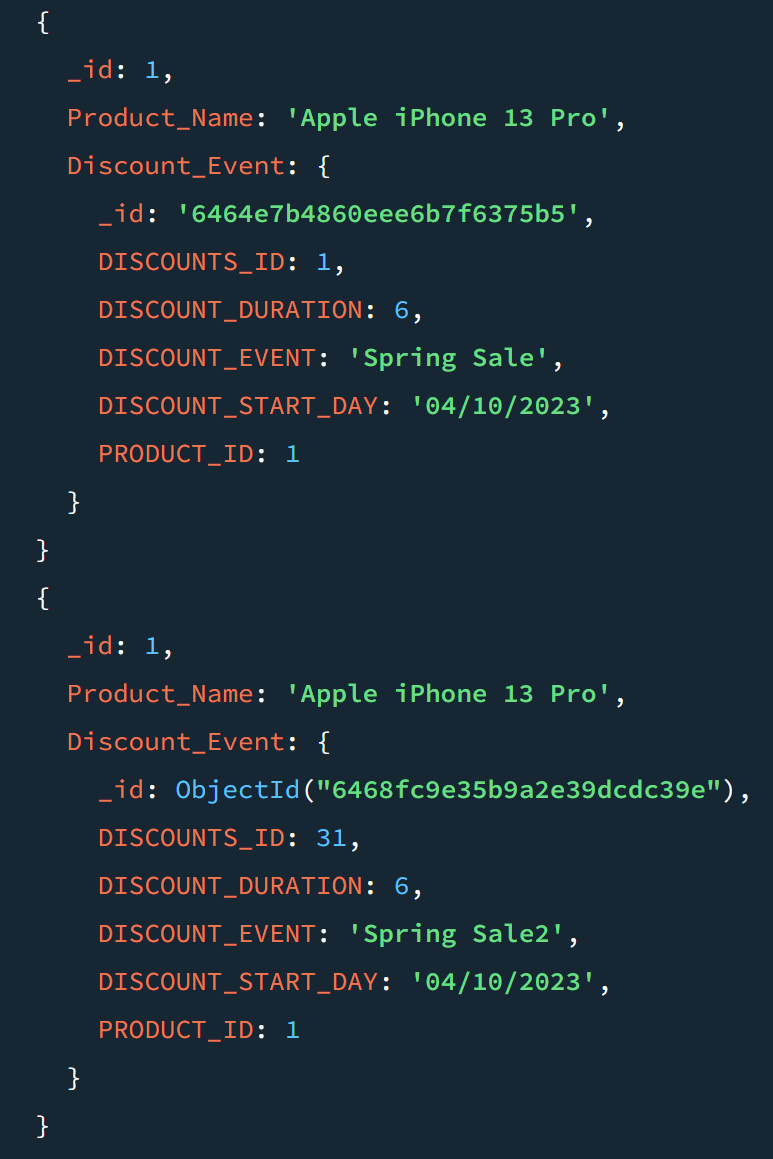
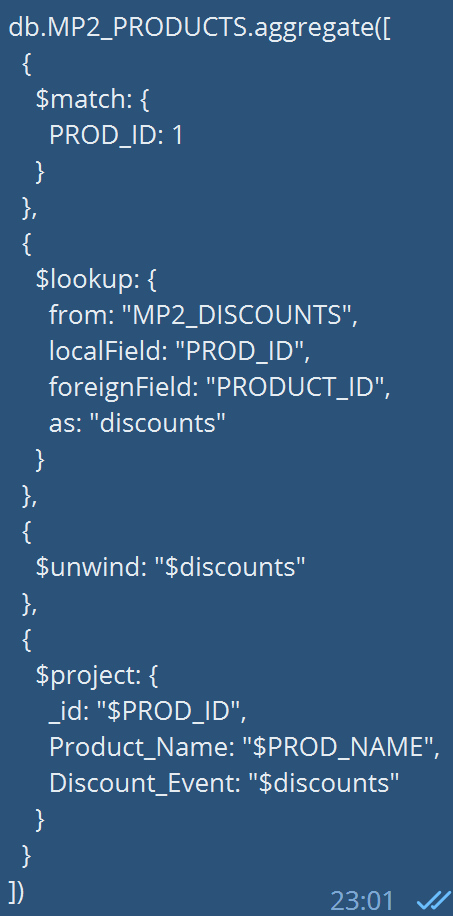


- Create a trigger before insert on any entity which will show the current number of rows in the table

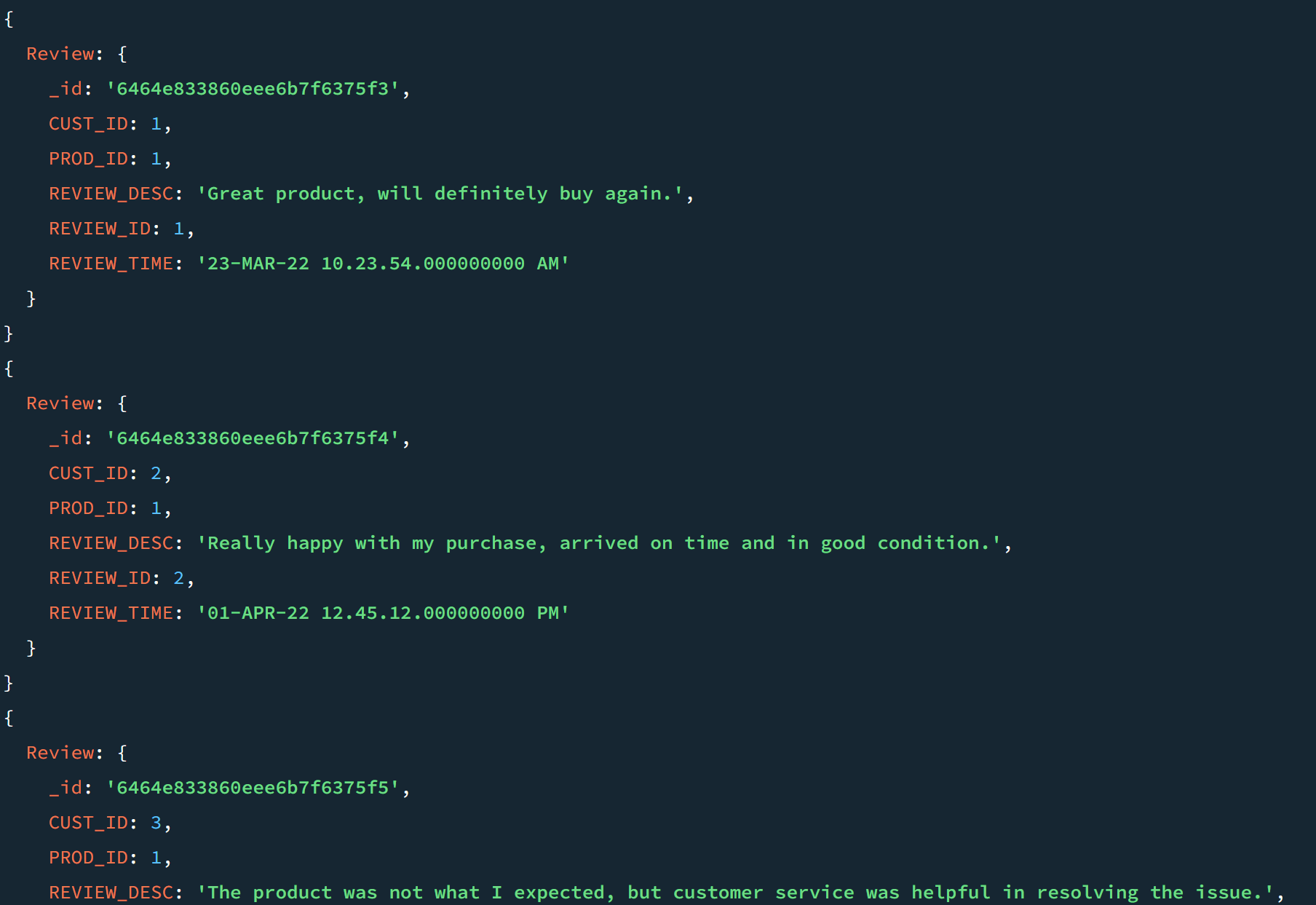
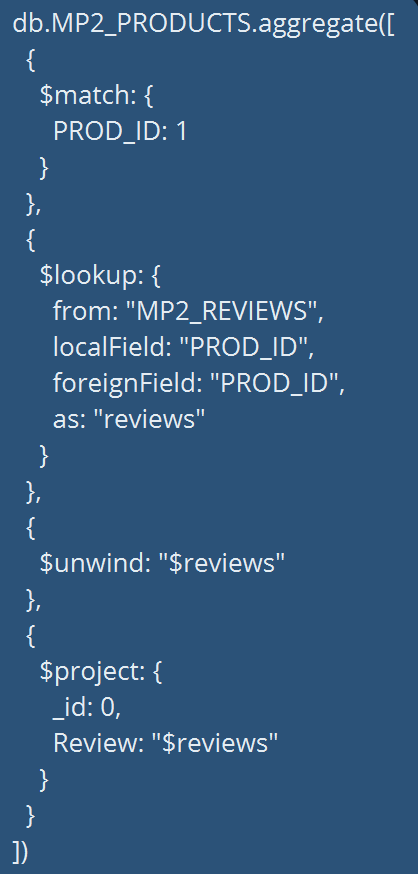


Yedyge:

1. Куери который показывает все скидки имеющиеся на каком то продукте



2.Запрос который показывает все Отзывы на определенный продукт



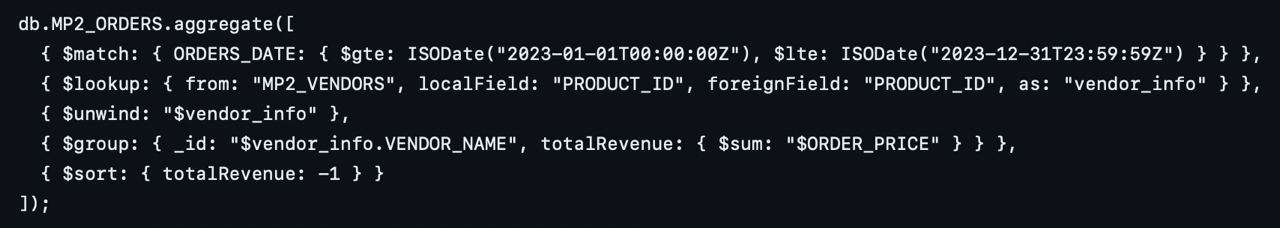
3.

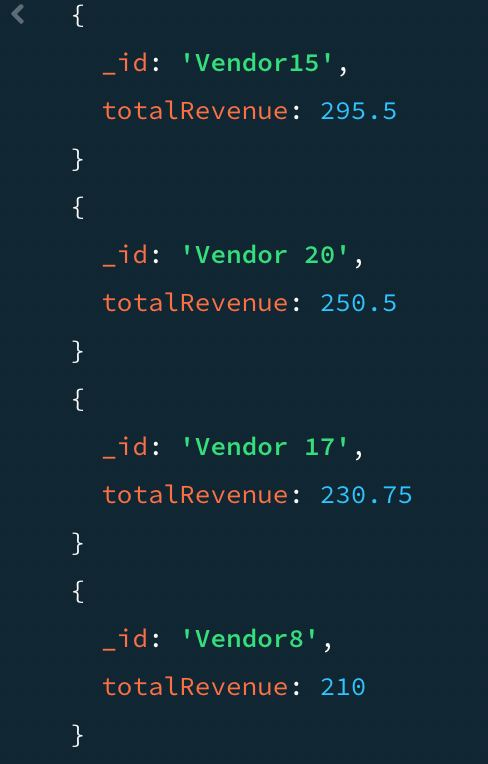
Этот запрос выдает все фотки для данного продукта



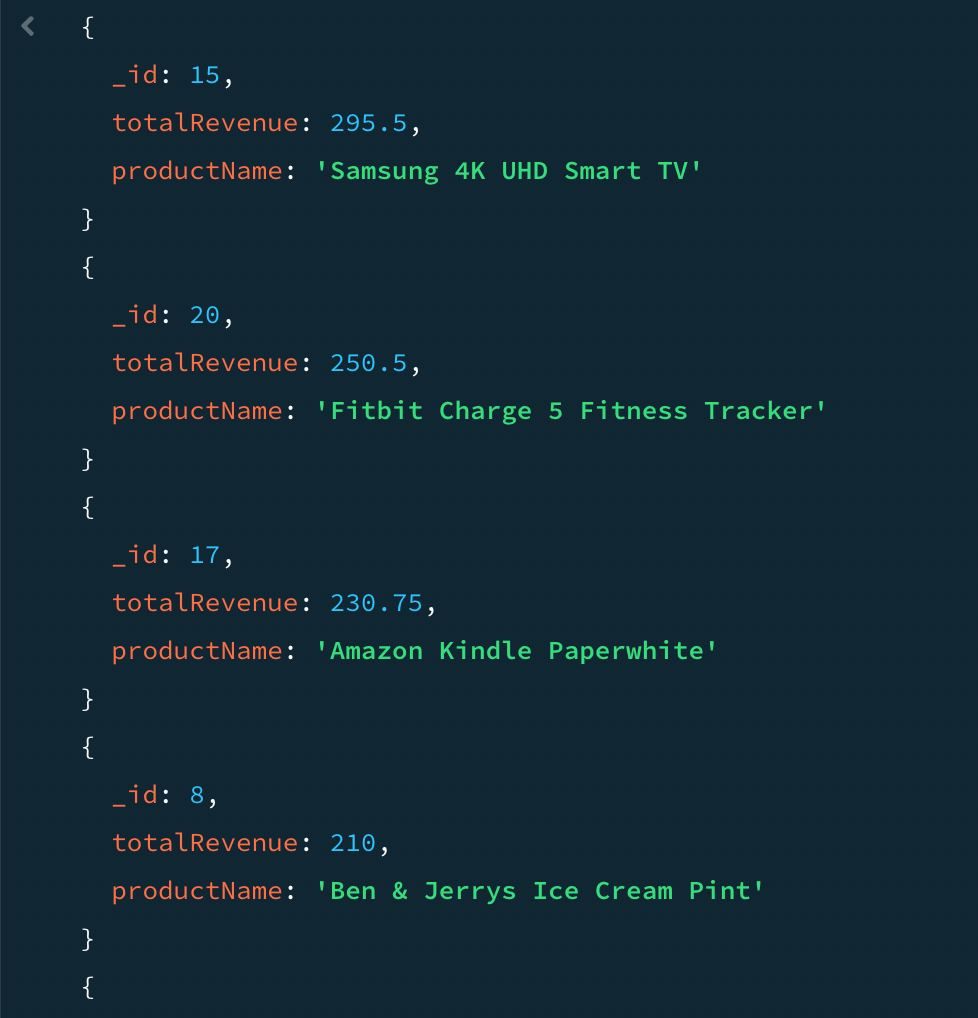
Rassul:

1. Calculates the total revenue of each supplier over a given period of time: This query can be used to determine the most profitable suppliers, which can be useful for purchasing decisions.





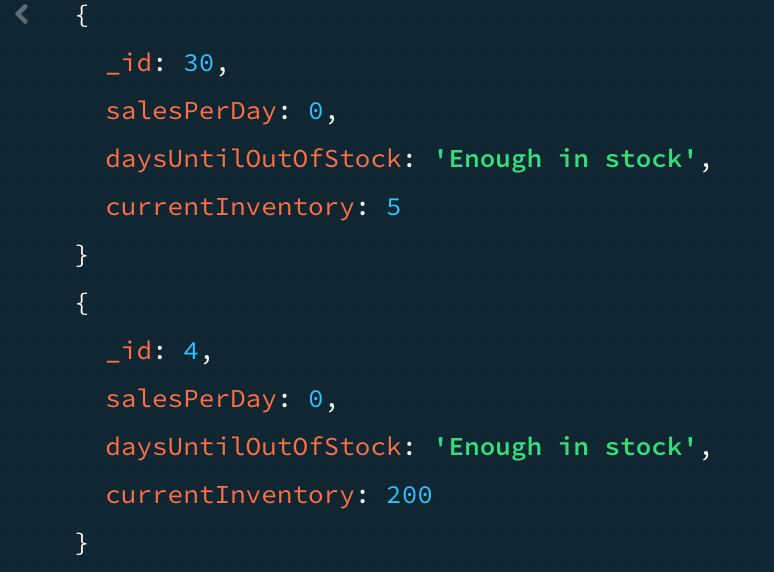
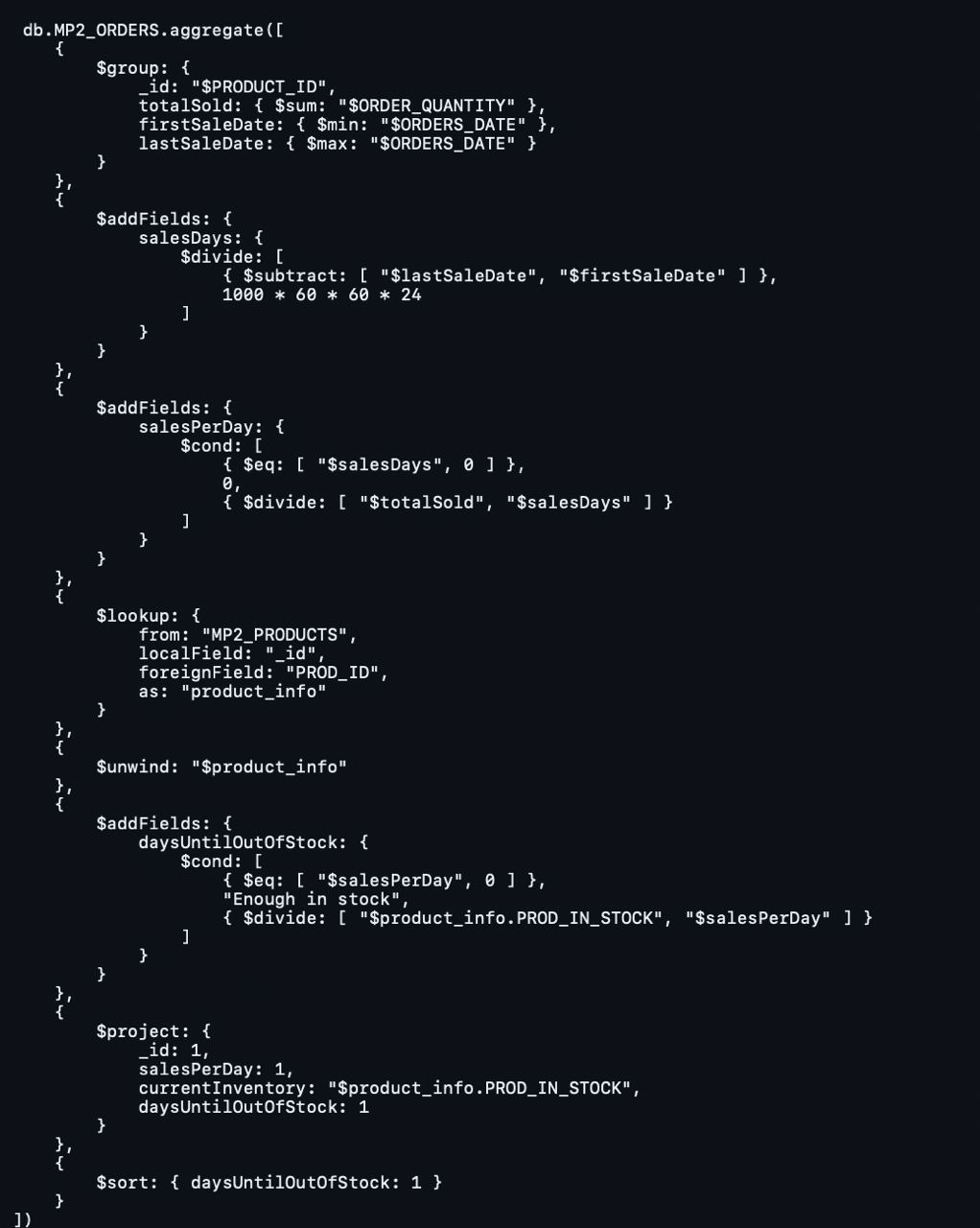
2. Top products generating the highest revenue: The convenience of using this query lies in identifying the best-selling or most profitable products, allowing for a greater focus on promoting these products.



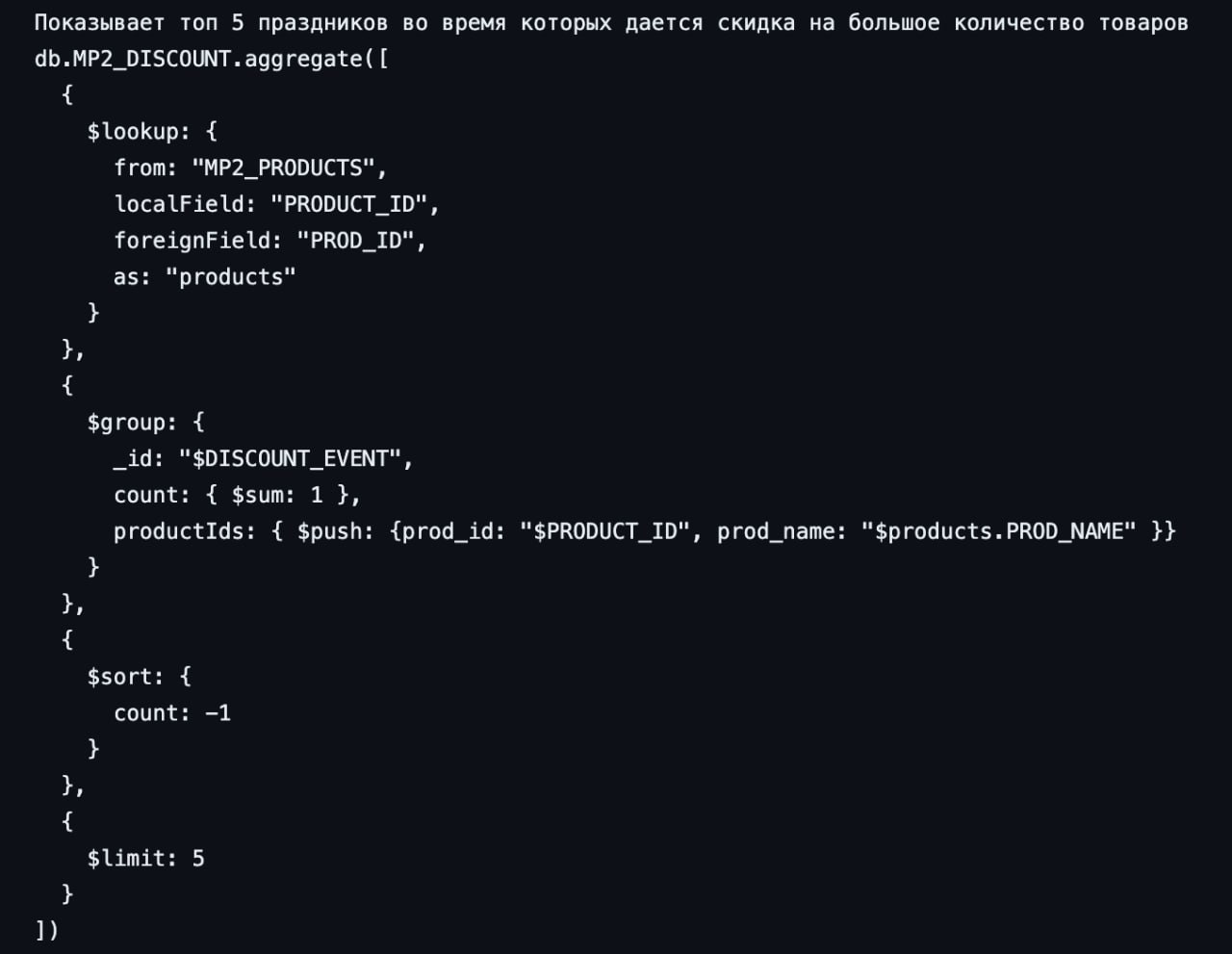
3. Customer Lifetime Value Calculation: This is a forecast of the net profit associated with all future customer relationships. It is crucial for marketing decisions, such as determining how much money you can afford to spend on acquiring a new customer



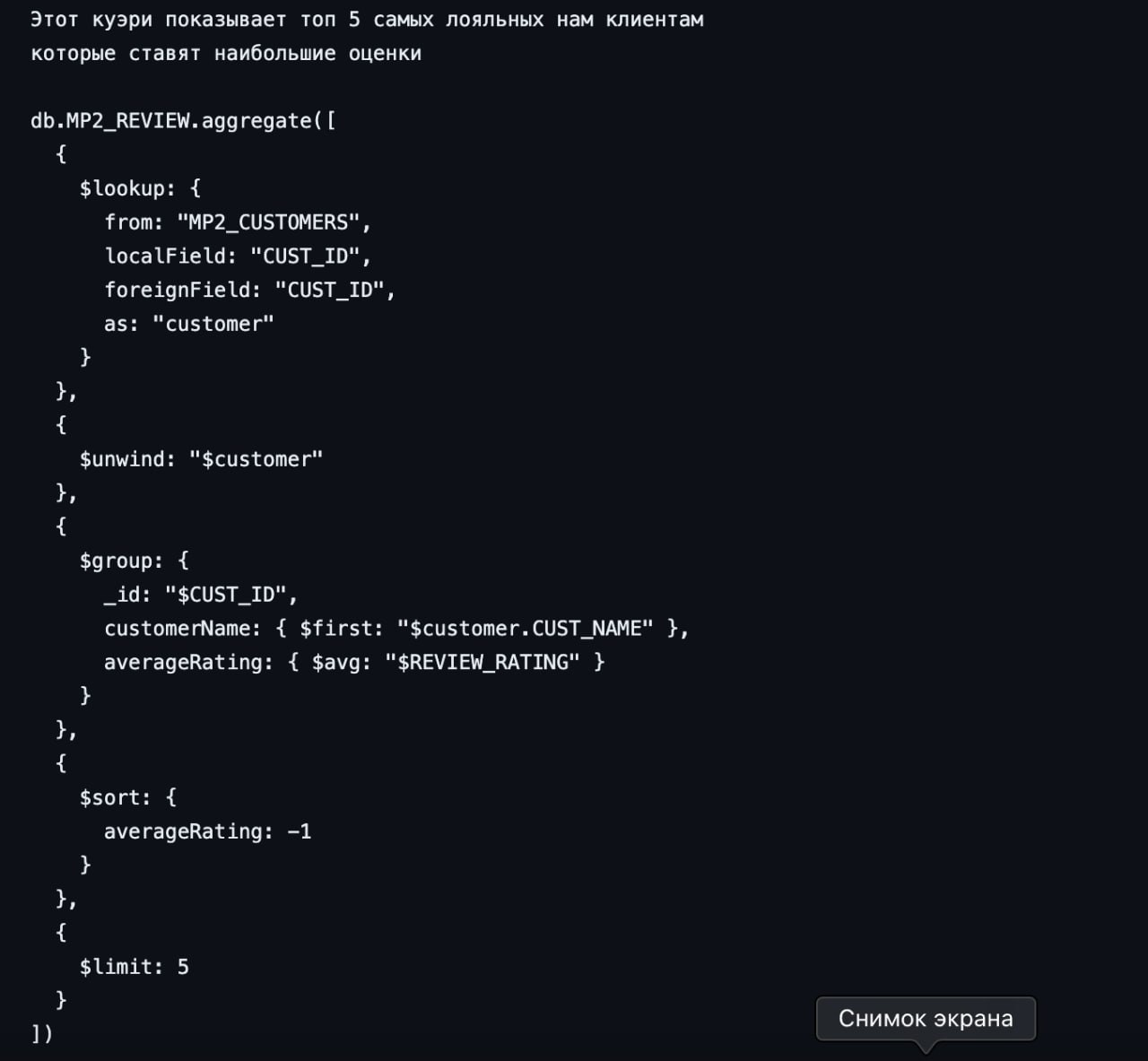
1. Inventory Forecasting: This is crucial for inventory management as stock depletion results in missed sales opportunities, while excessive inventory leads to increased storage costs.

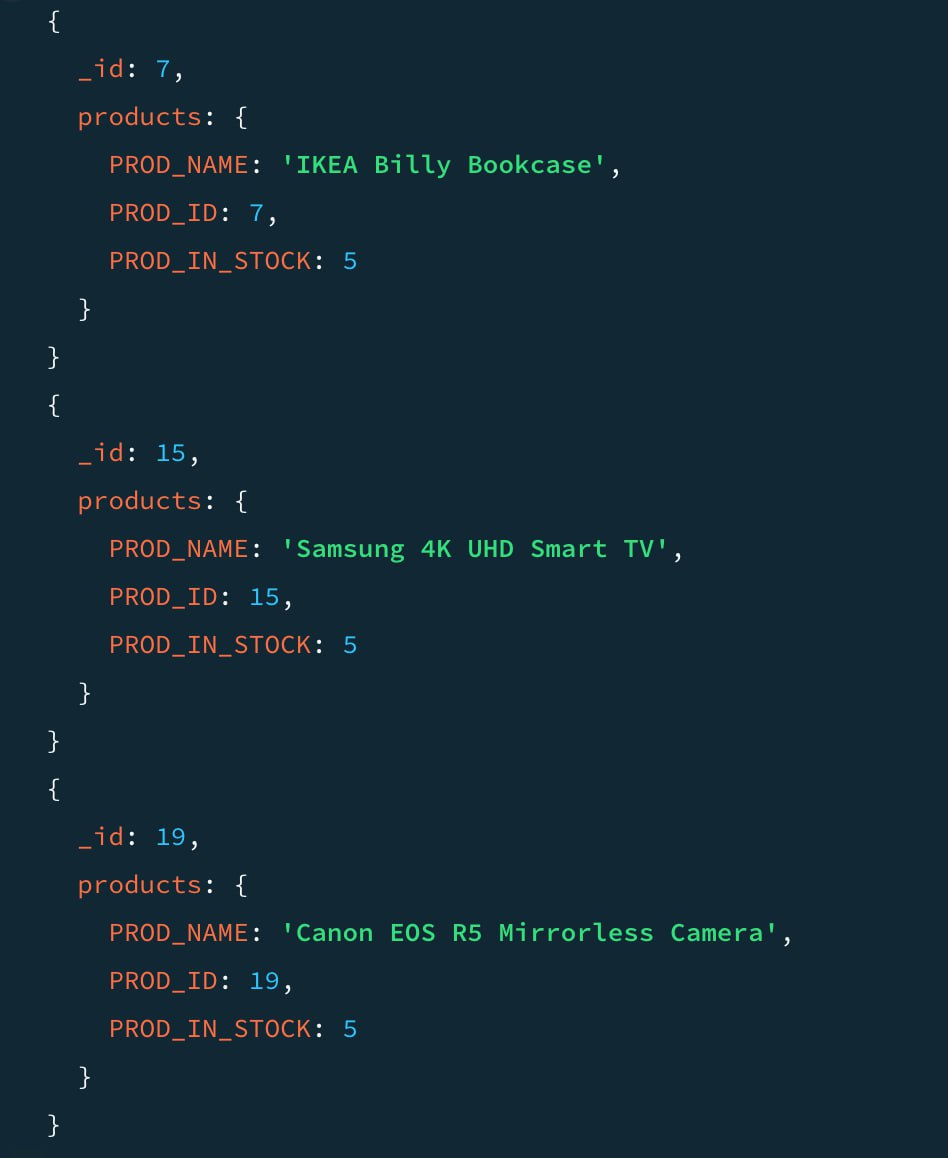
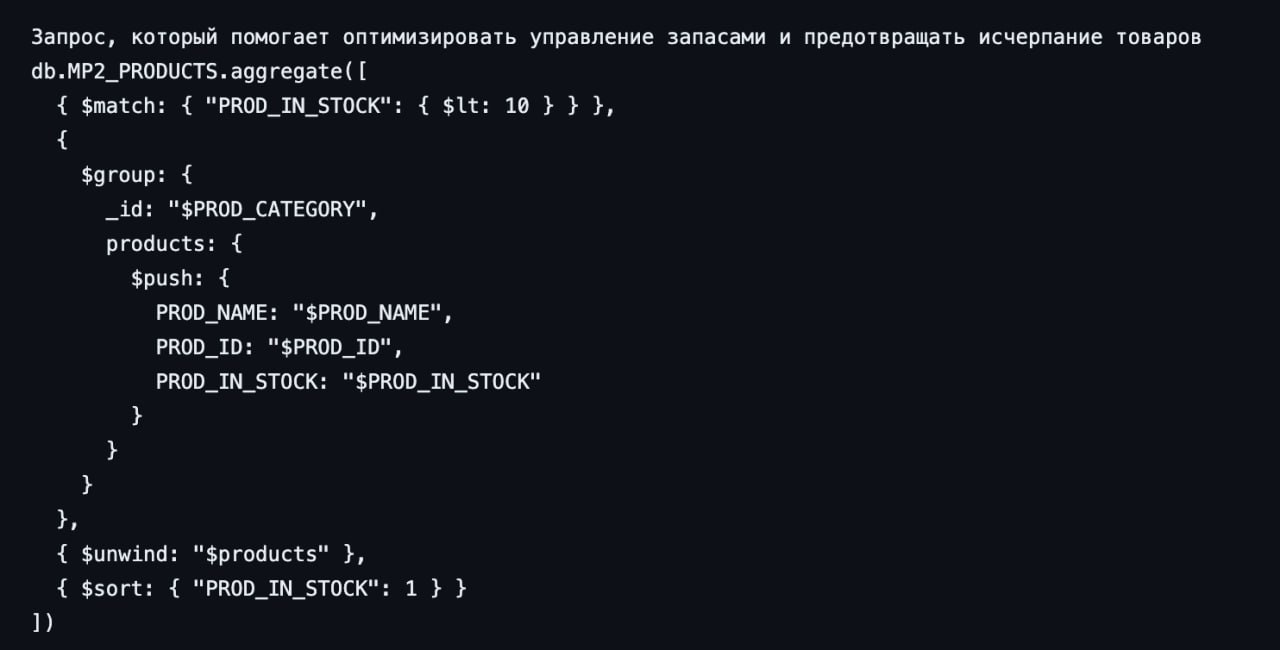


Abylay:



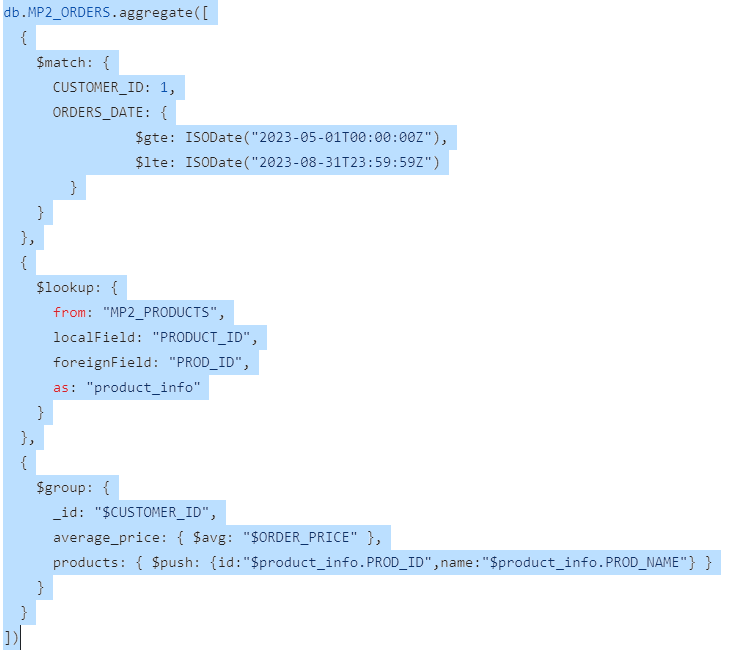
2.



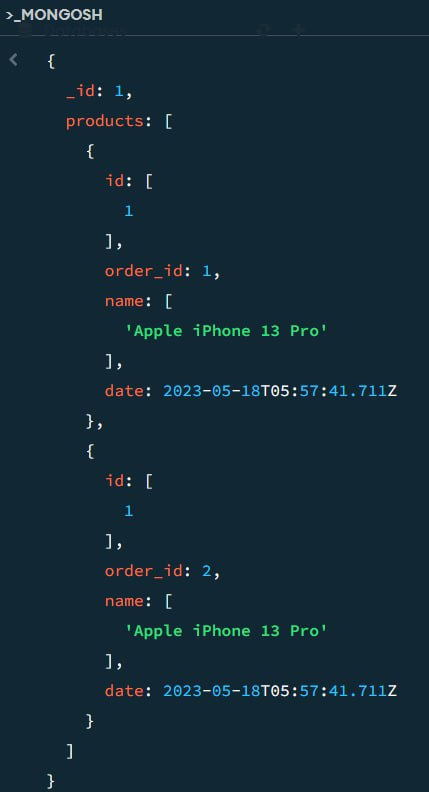
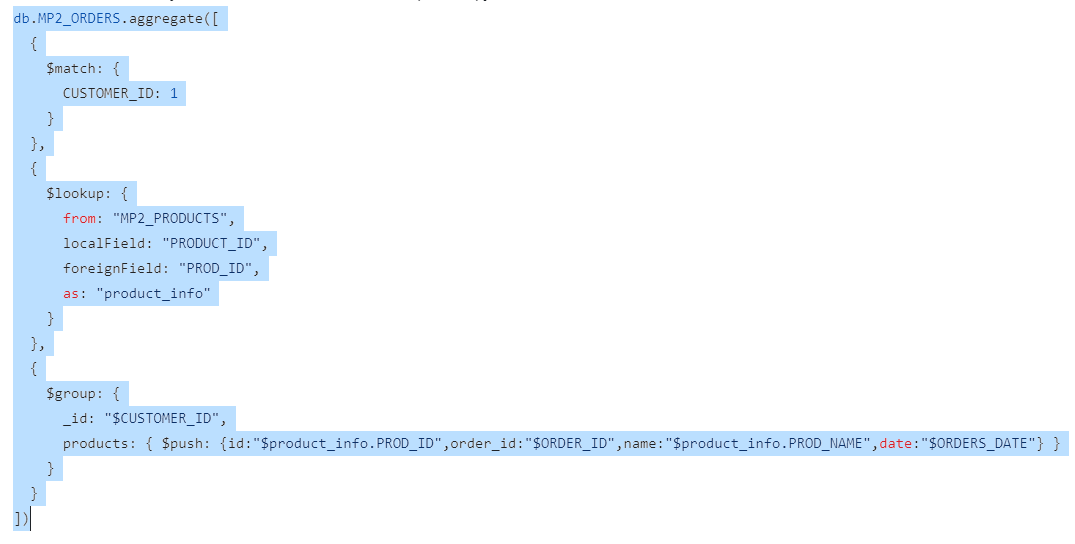


Ramazan:

1. В промежутке времени у заказшика показывает сколько стоят вешь которую он покупает



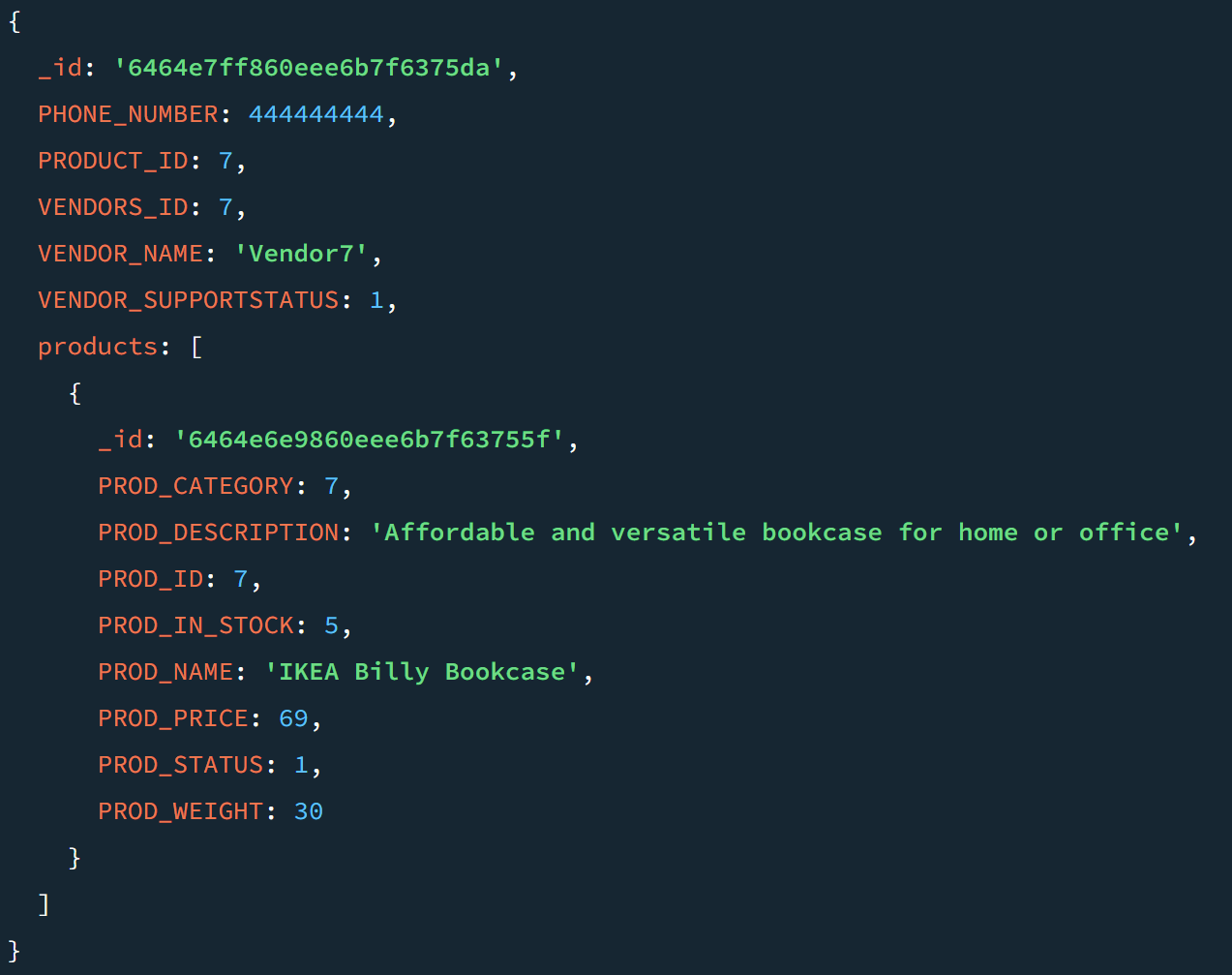
1. выводит по заказшику все его заказы и название берет с другой коллекции

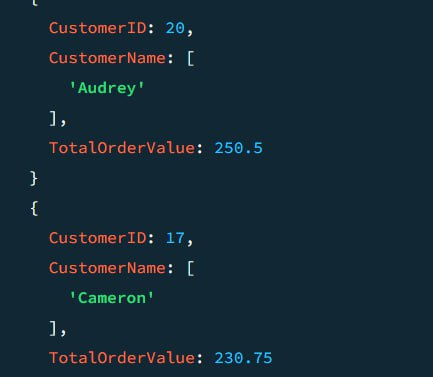
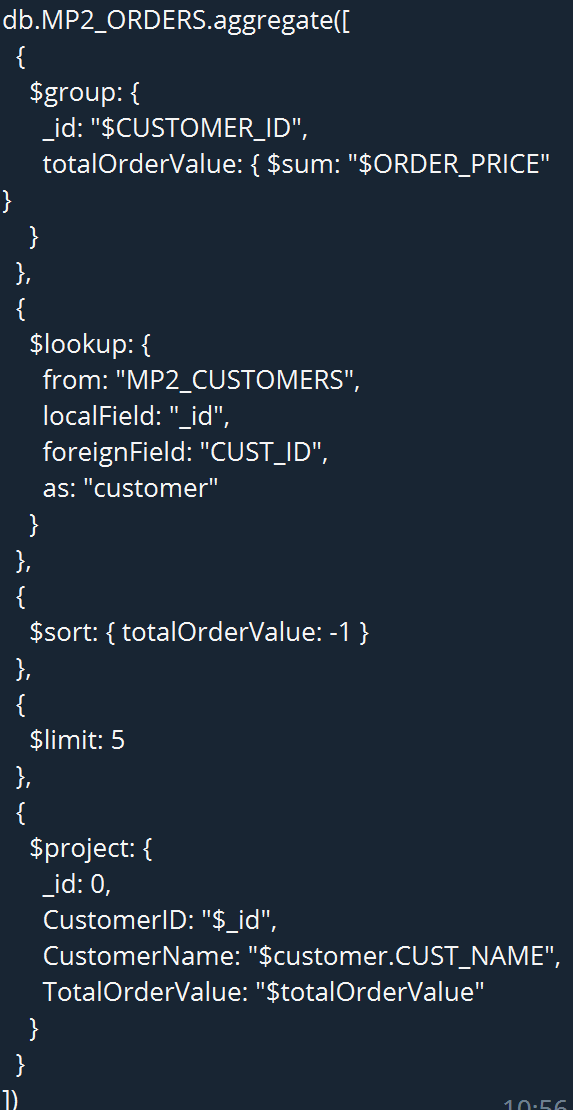


3.если продукты с браком их отзывают выводит имя и место доставки 3 колектион колданады

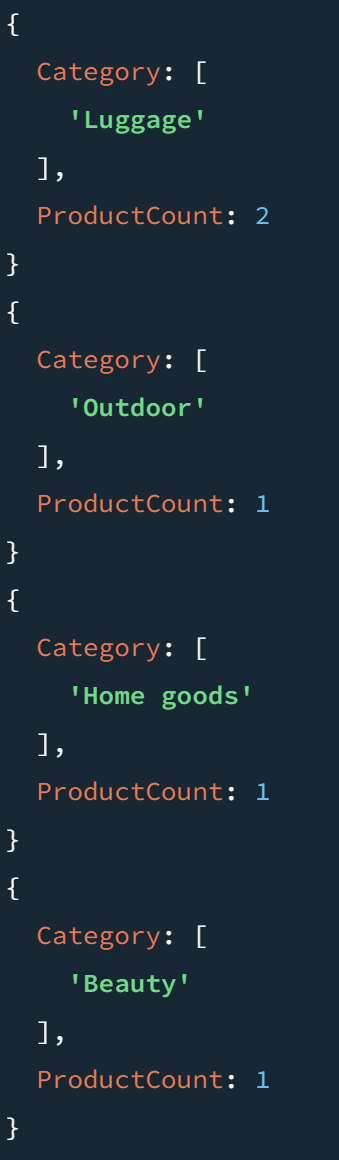


Amir:





3.



Made By:

Bissultanov Yedyge

Abylay Akhmetov

Ramazan Khakimzhan

Chsheriyazdanov Rassul

Amirkhan Mamytbekov