***M*ukan Yerbolat id:210103036  
Sultanov Yerkanat id:210103112**

**Turganbekov Asylkhan id:210103295**

**Melis Tileuli id:210107181**

**U-Point (e-commerce)**

INTRODUCTION

The ***U-Point*** is like an I-Point, but little-bit worse. We sell cheap gadgets from China. To do this, we have developed a **database system** to successfully earn and become super rich.

ENTITY RELATIONSHIP DIAGRAM

1. Customer: This table contains information about customers, including their names, email addresses, phone numbers, and address IDs.

2. Cart: This table stores the details of a customer's shopping cart, including the total price of the items.

3. Cart\_Item: This table contains the individual items in a customer's shopping cart, including the product ID and quantity.

4. Orders: This table contains information about customer orders, including the order date, total price, and address ID.

5. Order\_Item: This table contains the individual items in a customer's order, including the product ID and quantity.

6. Billing\_INFO: This table stores billing information for customer orders, including the bill date and credit card number.

7. Credit\_INFO: This table contains information about credit cards used for billing, including the card number and CVV.

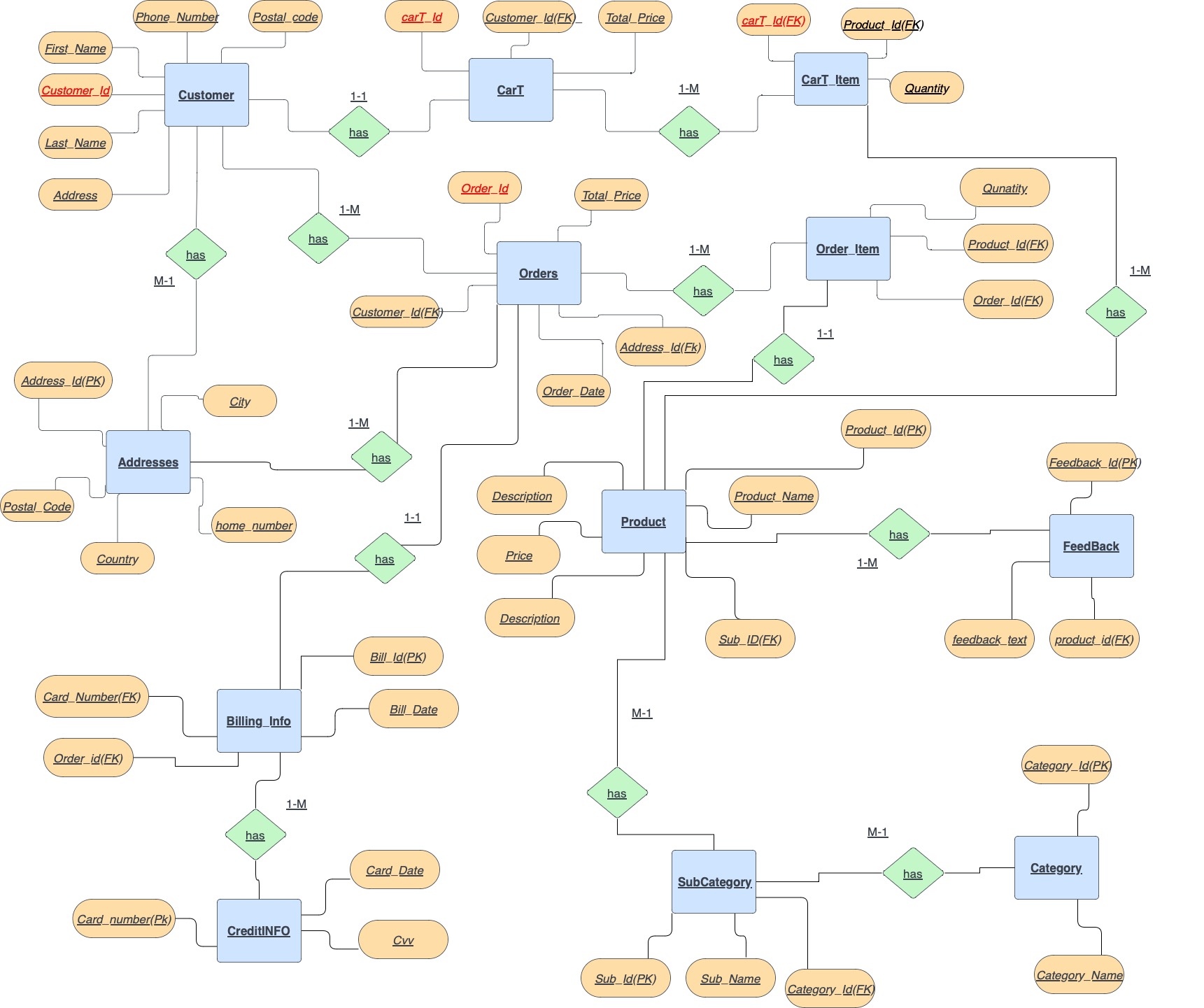
8. Product: This table contains information about products for sale, including the product name, description, price, and subcategory ID.

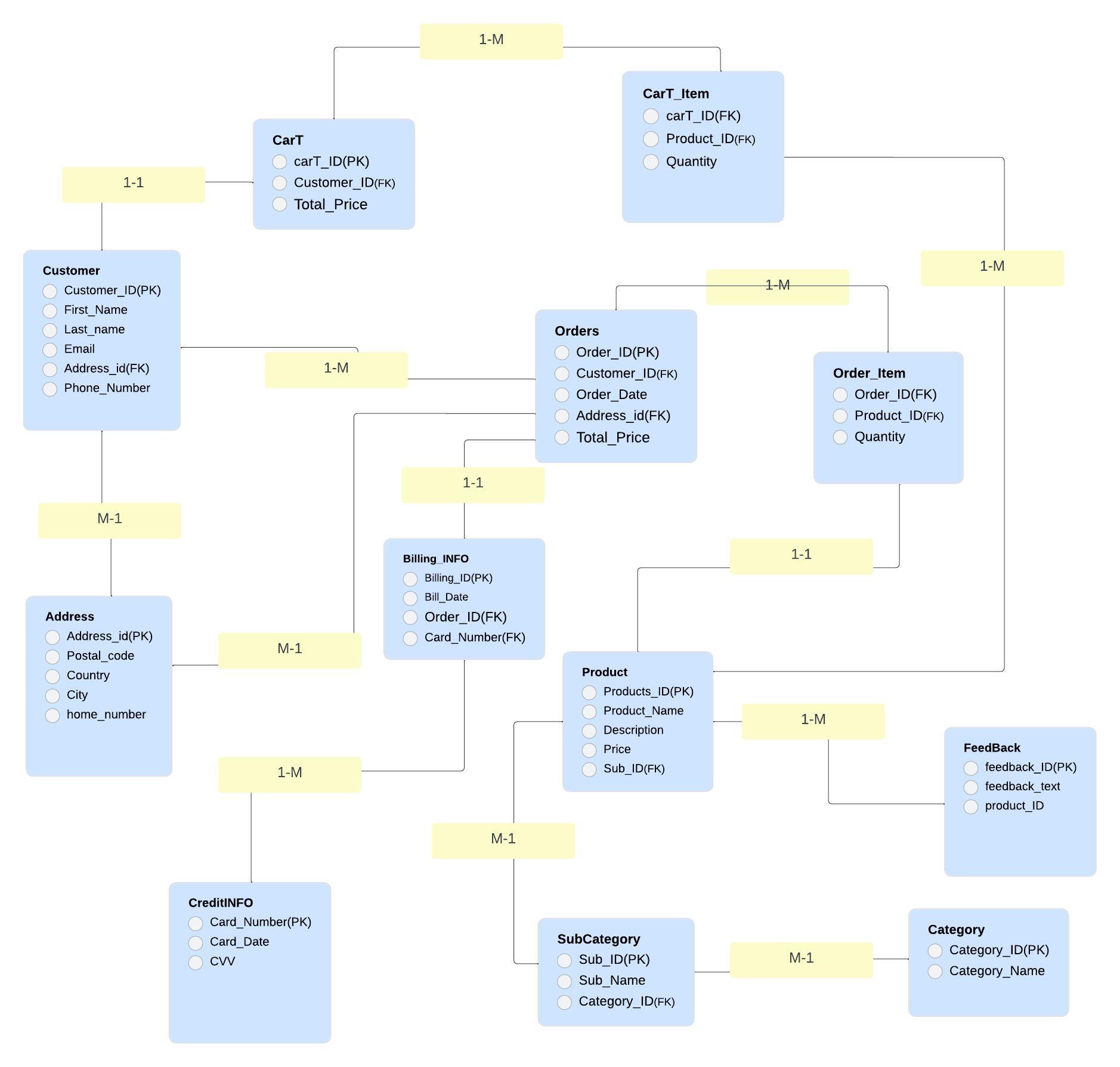
9. SubCategory: This table stores the subcategories to which products belong, including the subcategory name and category ID.

10. Category: This table stores the categories to which subcategories belong, including the category name.

11. Feedback: This table contains feedback from customers about specific products, including the feedback text and product ID.

12. Addresses: This table stores the addresses of customers, including the postal code, country, city, and home number.





NORMAL FORMS

*Customer Table*

Customer\_ID is a primary key, and thus satisfies 1NF and 2NF.

First\_Name and Last\_Name are atomic, and thus satisfy 1NF.

Email is atomic and unique, and thus satisfies 2NF.

Address\_id is a foreign key that references Addresses, and thus satisfies 2NF.

Phone\_Number is atomic, and thus satisfies 1NF.

The Customer table appears to be in 3NF.

*CartT Table*

CartT\_ID is a primary key, and thus satisfies 1NF and 2NF.

Customer\_ID is a foreign key that references Customer, and thus satisfies 2NF.

Total\_Price is atomic, and thus satisfies 1NF.

The CartT table appears to be in 3NF.

*Cart\_Item Table*

id is a primary key, and thus satisfies 1NF and 2NF.

CartT\_ID is a foreign key that references CartT, and thus satisfies 2NF.

Product\_ID is a foreign key that references Product, and thus satisfies 2NF.

Quantity is atomic, and thus satisfies 1NF.

The Cart\_Item table appears to be in 3NF.

*Orders Table*

Order\_ID is a primary key, and thus satisfies 1NF and 2NF.

Customer\_ID is a foreign key that references Customer, and thus satisfies 2NF.

Order\_Date is atomic, and thus satisfies 1NF.

Total\_Price is atomic, and thus satisfies 1NF.

Address\_id is a foreign key that references Addresses, and thus satisfies 2NF.

The Orders table appears to be in 3NF.

*Order\_Item Table*

id is a primary key, and thus satisfies 1NF and 2NF.

Order\_ID is a foreign key that references Orders, and thus satisfies 2NF.

Product\_ID is a foreign key that references Product, and thus satisfies 2NF.

Quantity is atomic, and thus satisfies 1NF.

The Order\_Item table appears to be in 3NF.

*Billing\_Info Table*

Billing\_ID is a primary key, and thus satisfies 1NF and 2NF.

Bill\_Date is atomic, and thus satisfies 1NF.

Order\_ID is a foreign key that references Orders, and thus satisfies 2NF.

Credit\_Number is a foreign key that references Credit\_Info, and thus satisfies 2NF.

The Billing\_Info table appears to be in 3NF.

*Credit\_Info Table*

Card\_Number is a primary key, and thus satisfies 1NF and 2NF.

Card\_Date is atomic, and thus satisfies 1NF.

CVV is atomic, and thus satisfies 1NF.

The Credit\_Info table appears to be in 3NF.

*Product Table*

Products\_ID is a primary key, and thus satisfies 1NF and 2NF.

Product\_Name is atomic, and thus satisfies 1NF.

Description is atomic, and thus satisfies 1NF.

Price is atomic, and thus satisfies 1NF.

Sub\_ID is a foreign key referencing the primary key of SubCategory table, and thus satisfies 2NF and 3NF.

The Product table appears to be in 3NF.

*Subcategory Table*

Sub\_ID is a primary key, and thus satisfies 1NF and 2NF.

Sub\_Name is atomic, and thus satisfies 1NF.

Category\_ID is a foreign key referencing the primary key of Category table, and thus satisfies 2NF and 3NF.

The Subcategory table appears to be in 3NF.

*Category Table*

Category\_ID is a primary key, and thus satisfies 1NF and 2NF.

Category\_Name is atomic, and thus satisfies 1NF.

The Category table appears to be in 3NF.

*Feedback Table*

Feedback\_ID is a primary key, and thus satisfies 1NF and 2NF.

Feedback\_Text is atomic, and thus satisfies 1NF.

Product\_ID is a foreign key referencing the primary key of Product table, and thus satisfies 2NF and 3NF.

The Feedback table appears to be in 3NF.

*Addresses Table*

Address\_ID is a primary key, and thus satisfies 1NF and 2NF.

Postal\_Code is atomic, and thus satisfies 1NF.

Country is atomic, and thus satisfies 1NF.

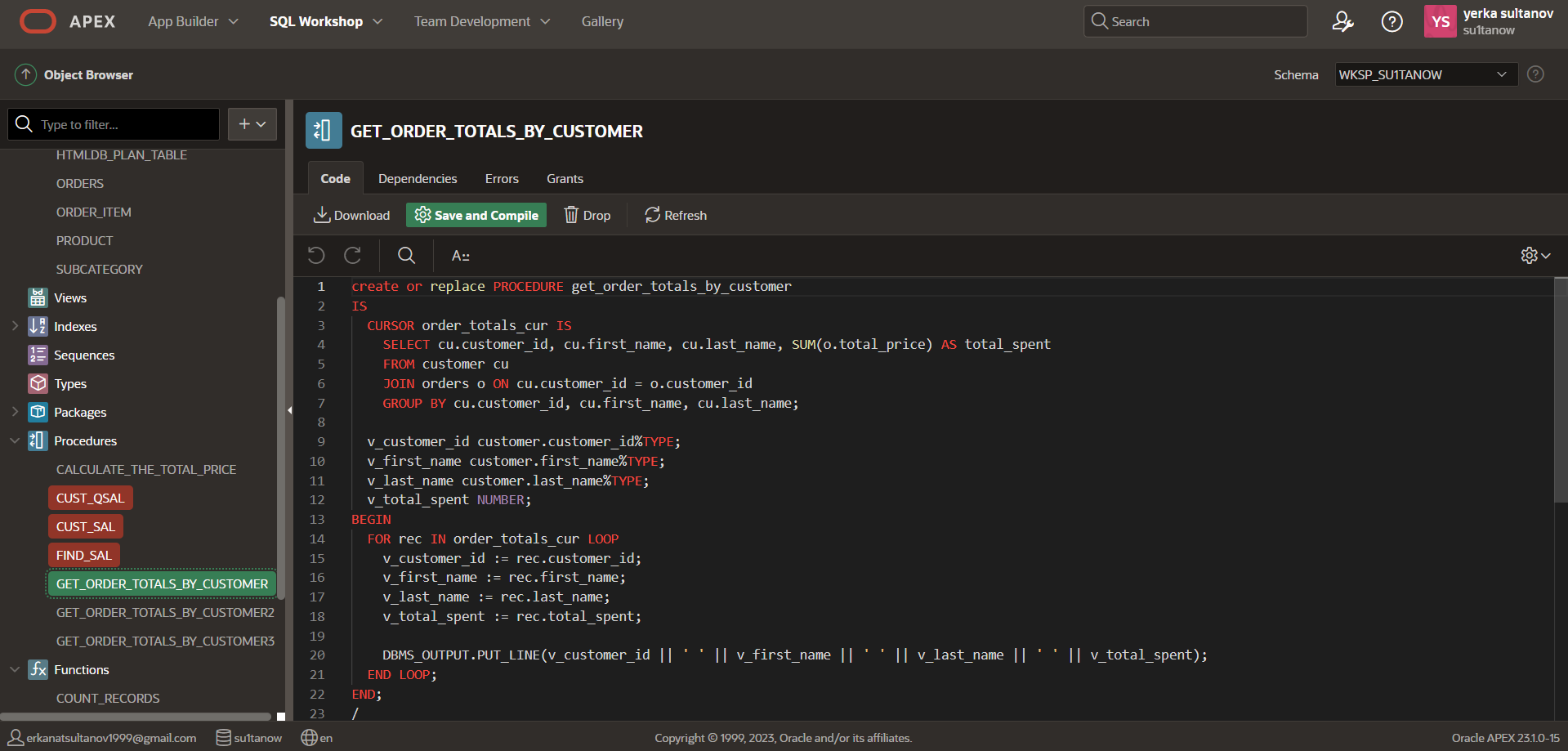
City is atomic, and thus satisfies 1NF.

Home\_Number is atomic, and thus satisfies 1NF.

The Addresses table appears to be in 3NF.

CODE PART

1. Procedure which does group by information

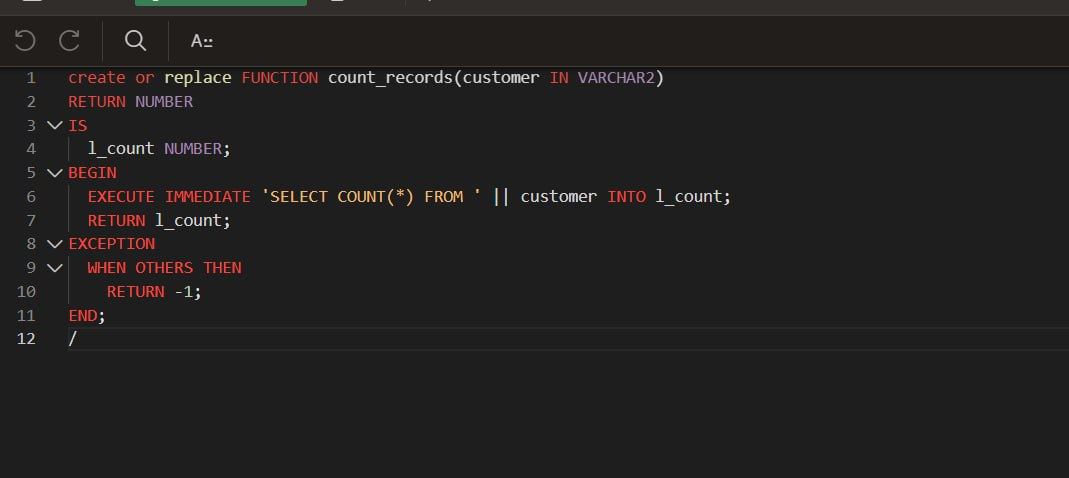


for checking:   
 BEGIN

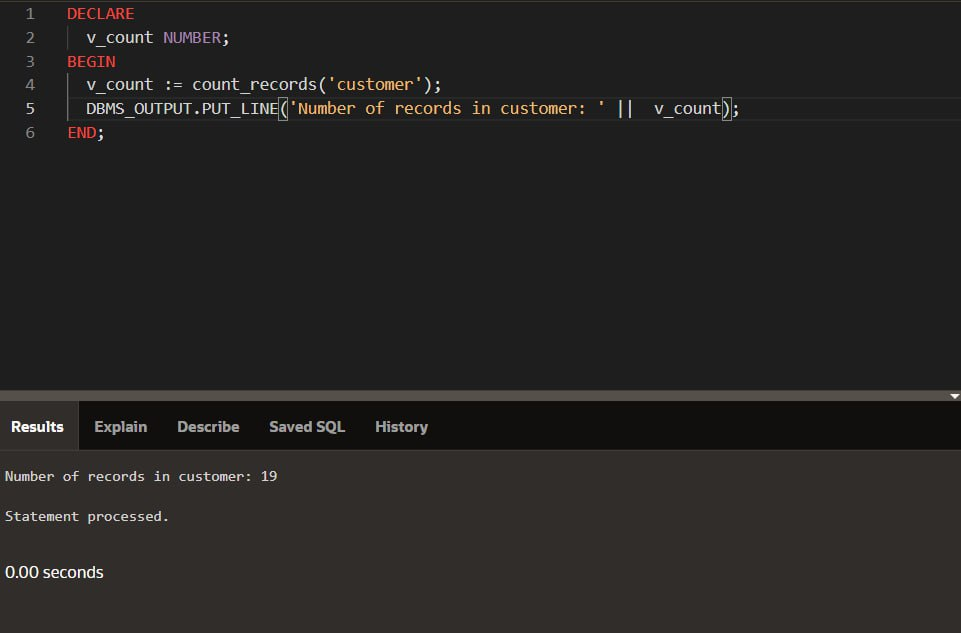
get\_order\_totals\_by\_customer;

END;

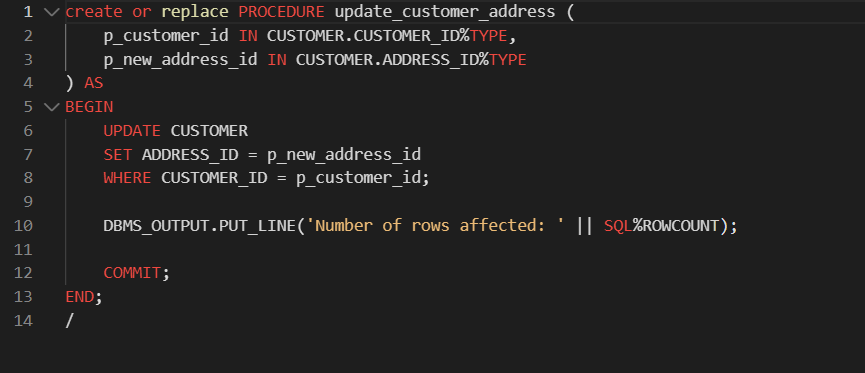
1. Function which counts the number of records

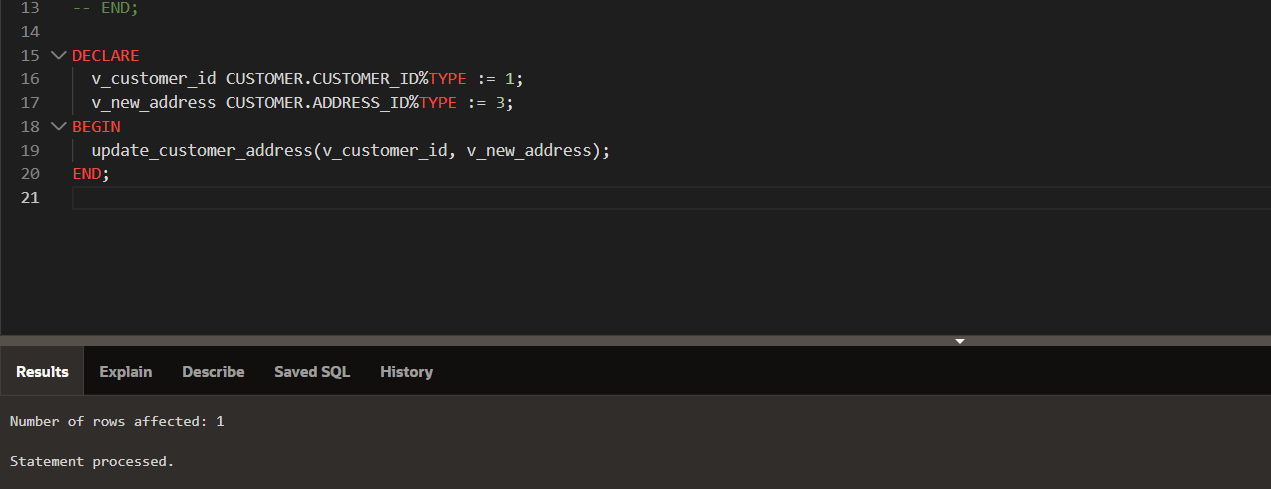


running:

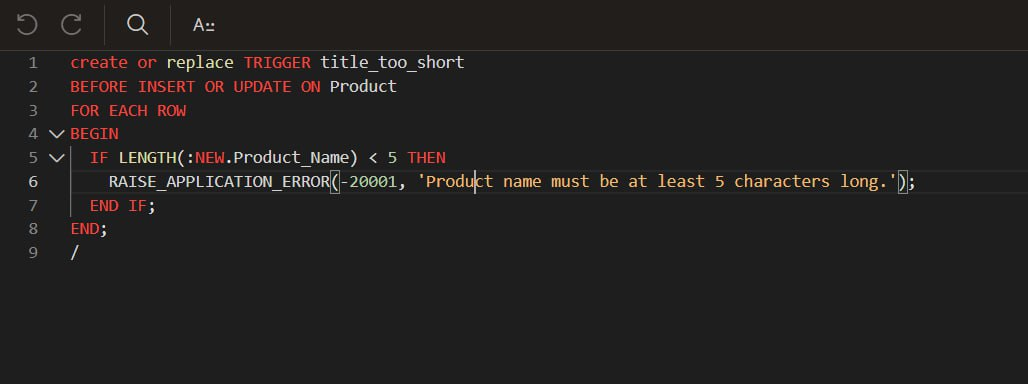


1. Procedure which uses SQL%ROWCOUNT to determine the number of rows affected

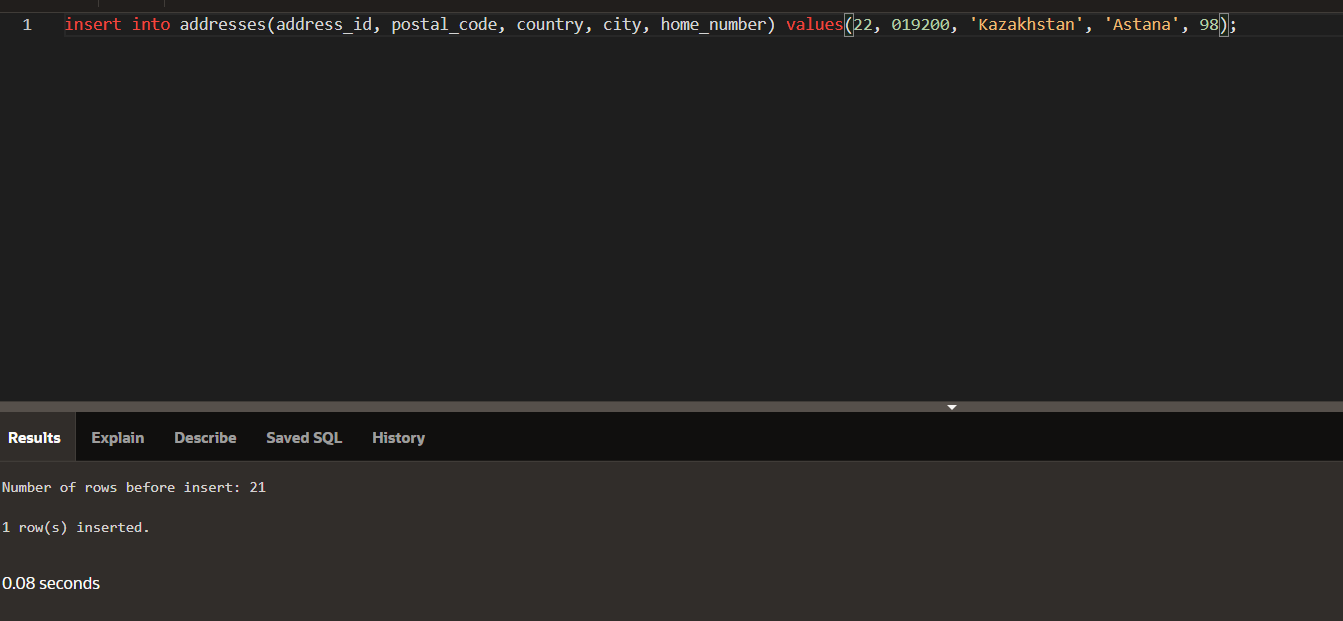
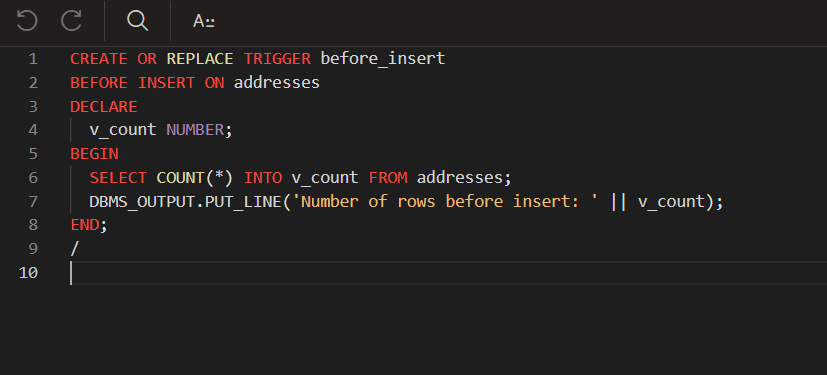


running : 

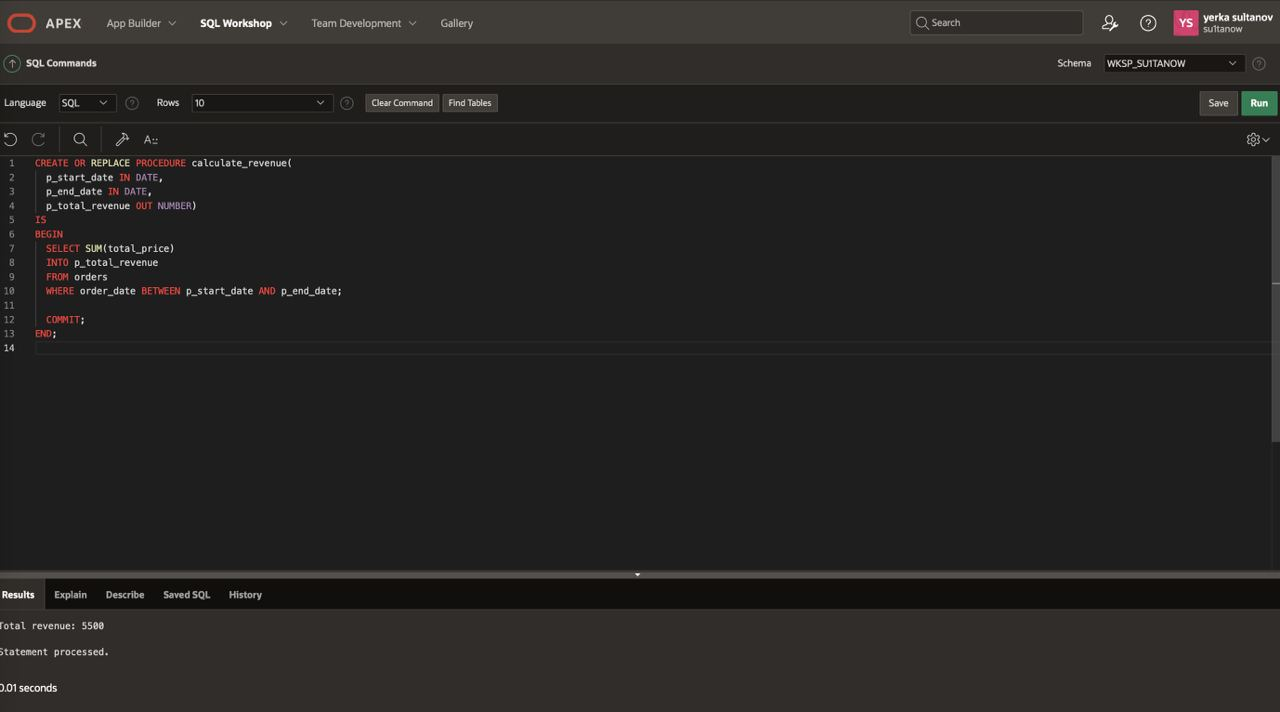
1. Add user-defined exception which disallows to enter title of item (e.g. book) to be less than 5 characters

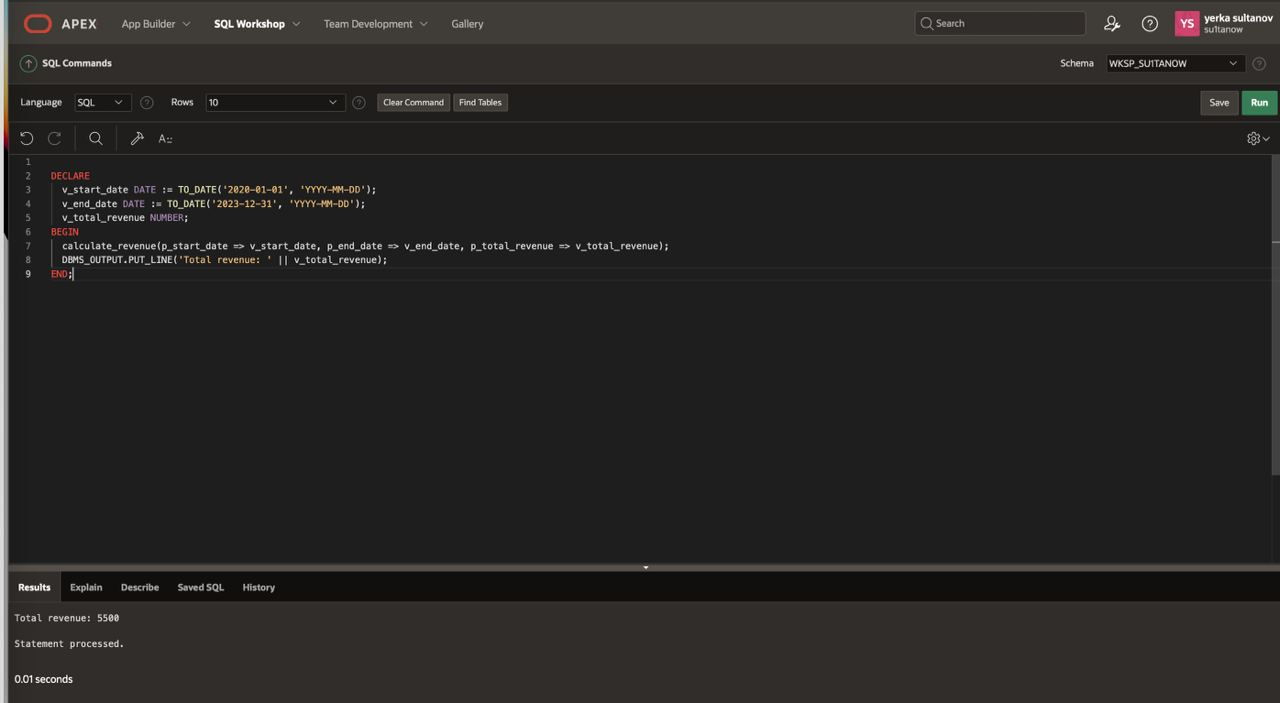


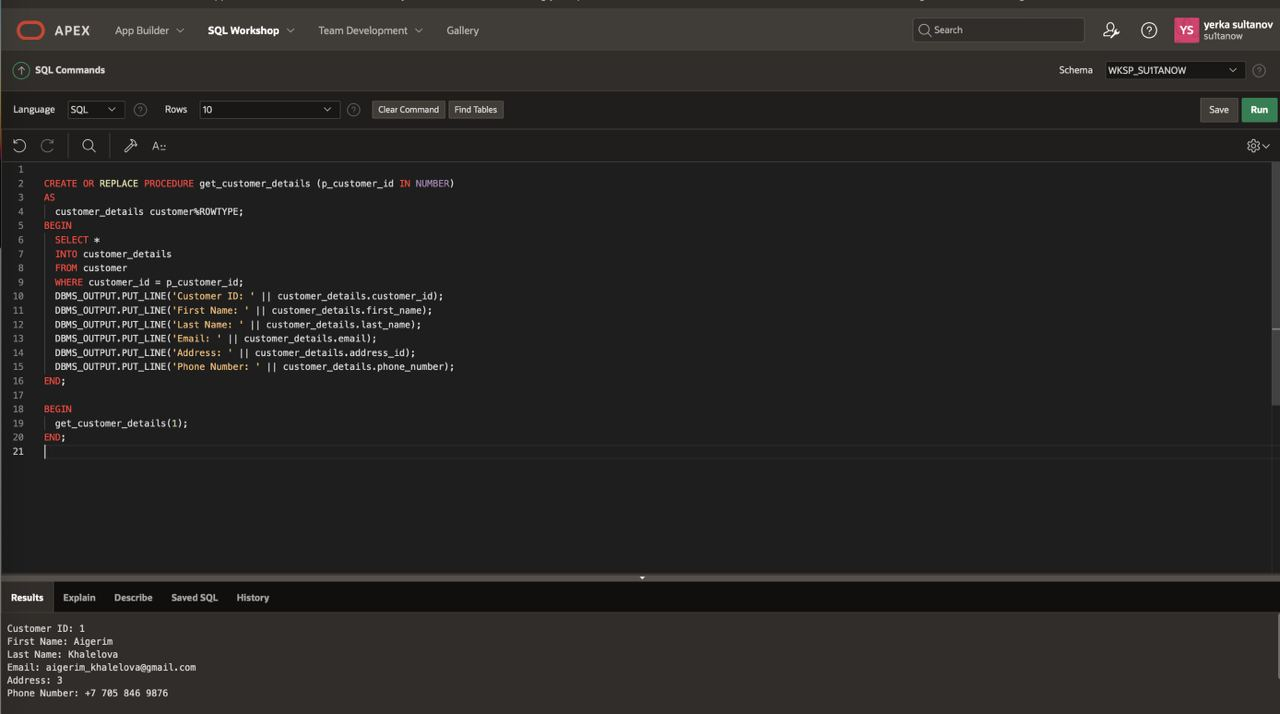
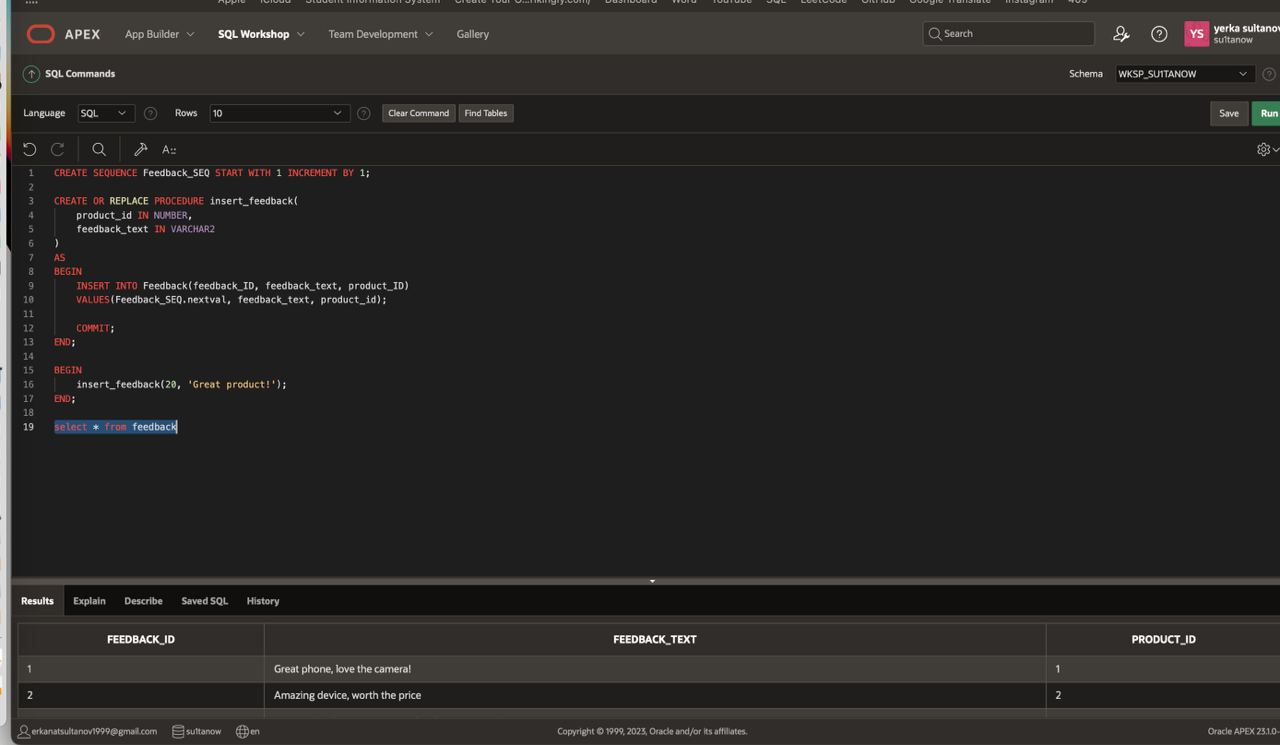
1. Create a trigger before insert on any entity which will show the current number of rows in the table

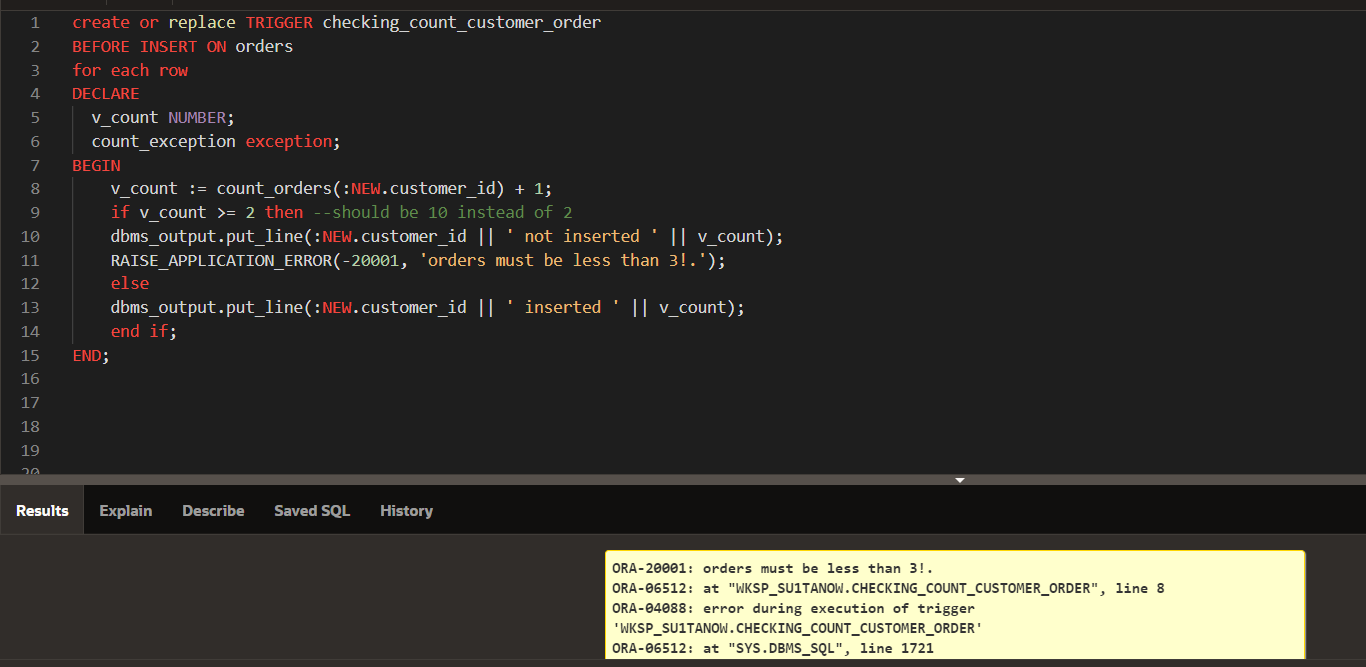


1. Calculate revenue,in some interval





1. Procedure to get customer details by customer ID:  
   
2. Procedure to insert a new feedback for a product:  
   
3. trigger that limit orders for one customer



insert into orders (order\_id, customer\_id, order\_date, total\_price, address\_id) values (13, 2, TO\_DATE('04/01/2020', 'MM/DD/YYYY'), 100, 1);

1. update cart total price

