E-commerce Marketplace Database Management System

Abbas Ibrahim

With this project, I set out on the thrilling adventure of creating, putting into practice, and making use of our very own database system, which was motivated by the complexities of selling on Amazon, an e-commerce platform. By carefully examining the theoretical ideas, my goal is to build a solid database that contains all the essential features required to facilitate the selling process on an ever-changing internet marketplace. The project is guided by a set of use cases that delineate the essential functions our database needs to facilitate, ranging from sellers' generation of new products to Amazon's fulfilment of those products.

THE NAMES OF ALL ENTITIES:

- Products
- Category
- Warehouse
- Seller
- Accounts
- Product_Order
- Shipping

RELATIONSHIP BETWEEN THE ENTITIES:

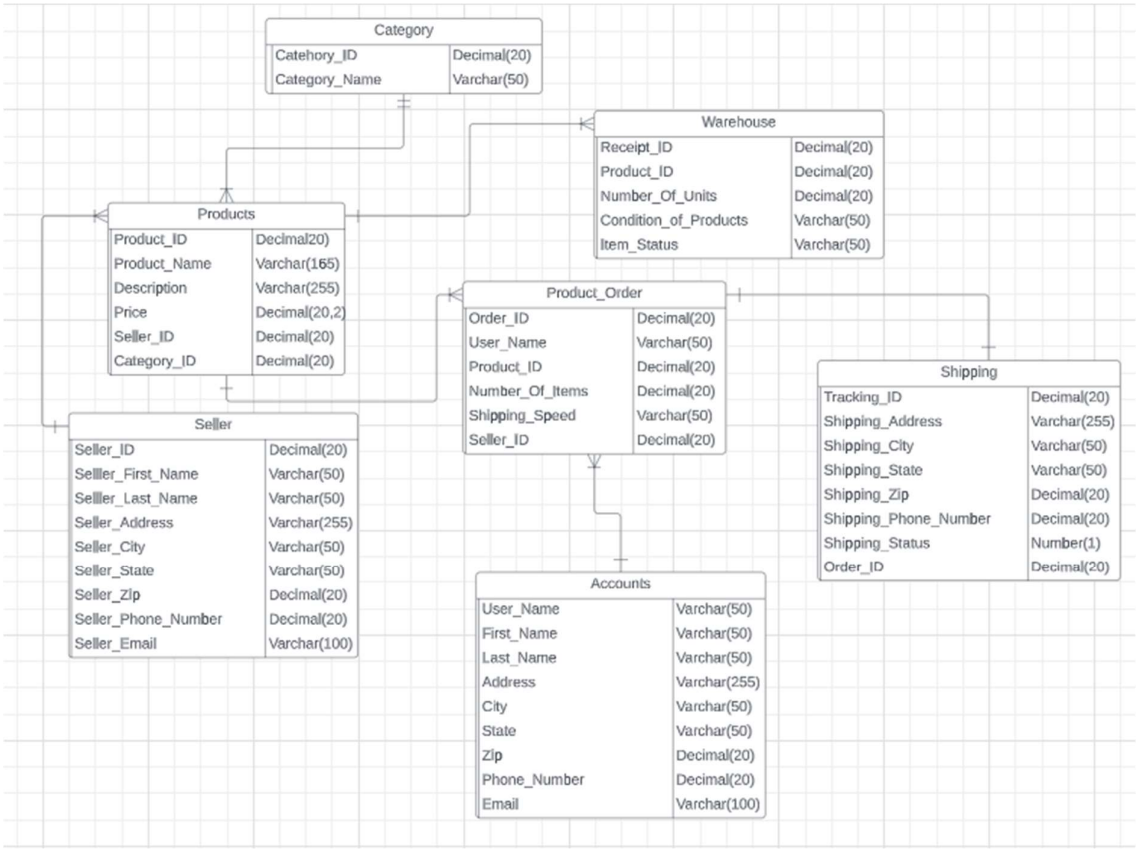
- Products to Category (one to many relationship)
- Category to Products (one to one relationship)
- Products to Seller (one to many relationship)
- Seller to Products (one to one relationship)
- Products to Warehouse (one to one relationship)
- Warehouse to Products (one to many relationship)
- Products to Product_Order (one to one relationship)
- Product_Order to Products (one to many relationship)
- Accounts to Product_Order (one to one relationship)
- Product_Order to Accounts (one to many relationship)
- Product_Order to Shipping (one to one relationship)
- Shipping to Product_Order (one to one relationship)

BUSINESS RULES:

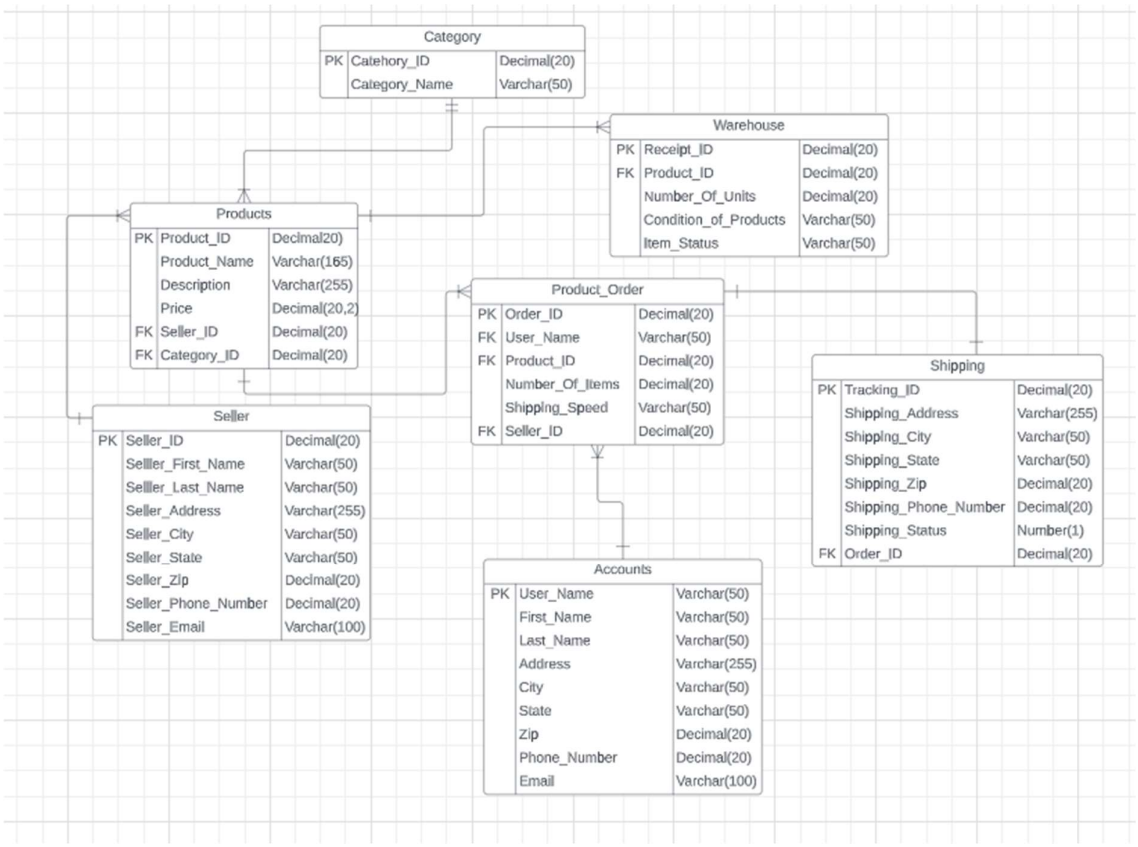
1. One or many products can fall under one category.
2. At least one product should fall under one category.
3. Each product should be listed uniquely (by using a unique identifier).
4. Sellers can list their products in relevant categories.
5. Many products can have one seller.
6. All products should have some quantity mentioned in the warehouse and their status whether the particulars stock is received or not and how many.
7. Whenever the seller ships the product the number units must be updated.

8. A seller must be associated with each product.
9. Per order should have all the product details, the number of quantities selected by the customer, shipping speed.
10. Many products can be placed in one order.
11. One order should have at least one product in it.
12. Each order should have a shipping or tracking ID.
13. Each shipping ID should have an updated shipping status.
14. User account should have all the necessary details filled, that is it cannot hold null values for the mandate columns like username, Email, Phone number, address.
15. Orders can be placed only if the users have an account or registered with an account.
16. Products can be added only if it has a seller.

CONCEPTUAL DATA MODEL:

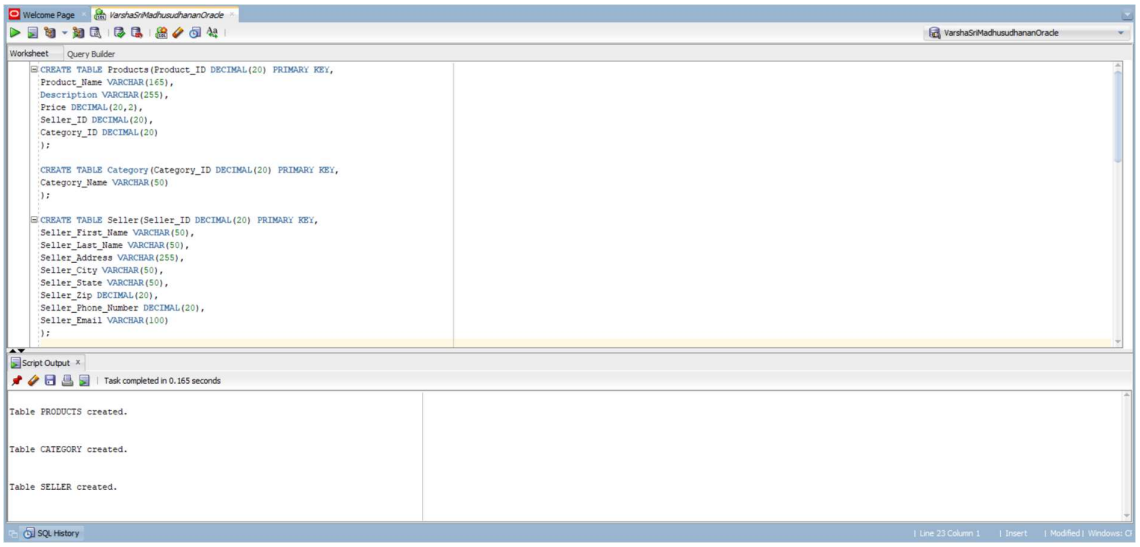


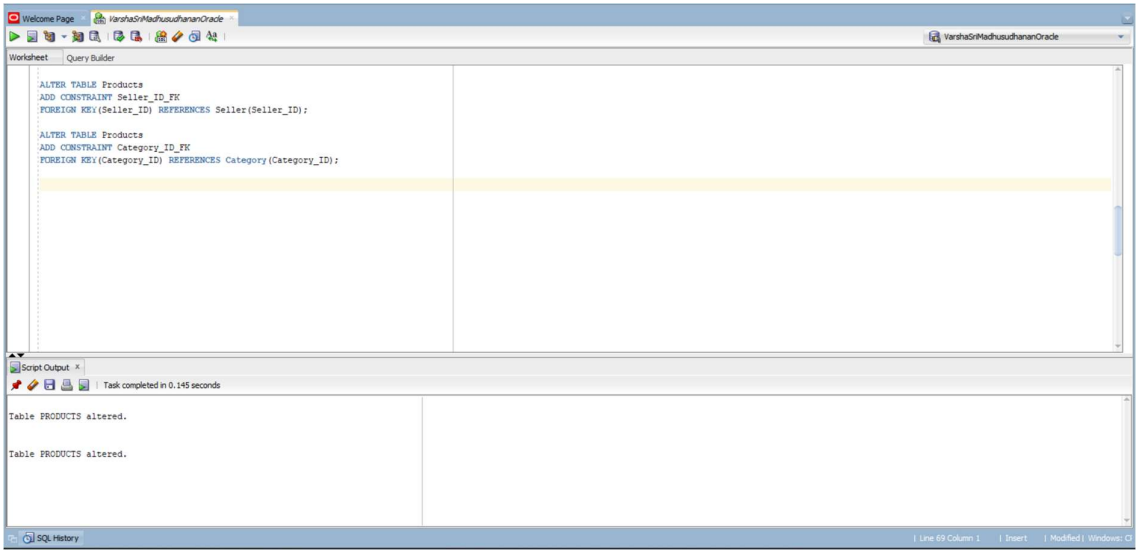
LOGICAL DATA MODEL:



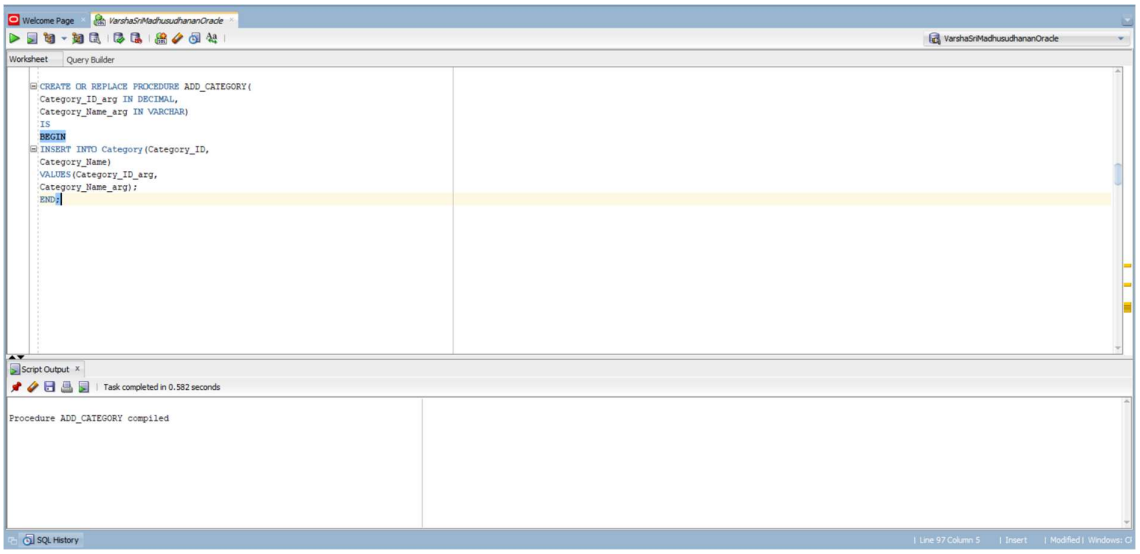
ASPECT 1: NEW PRODUCT CREATED BY SELLER

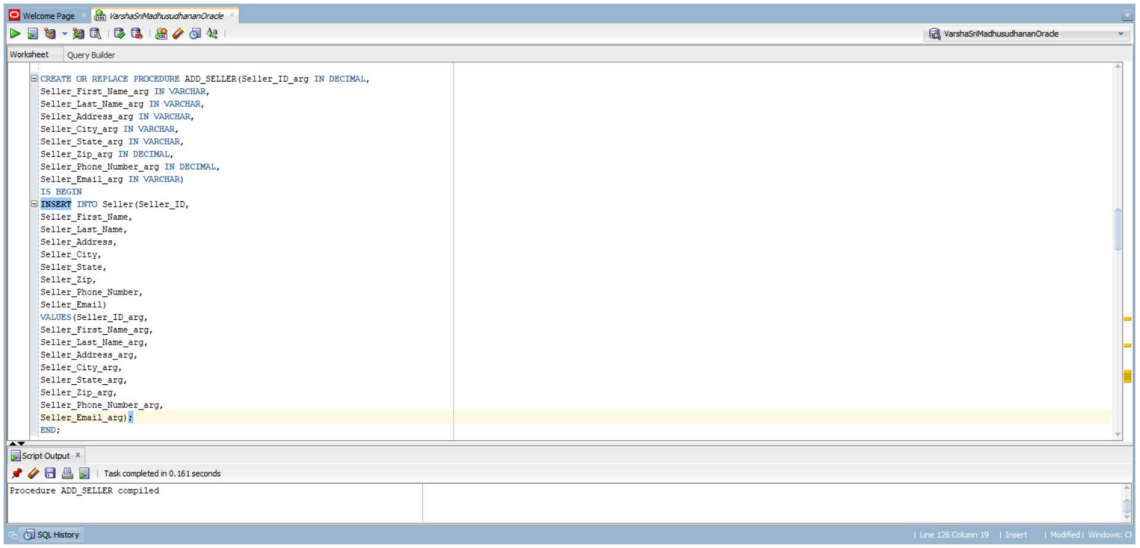
a. Creating the tables, constraints, and data needed to support new products as described in the use case.

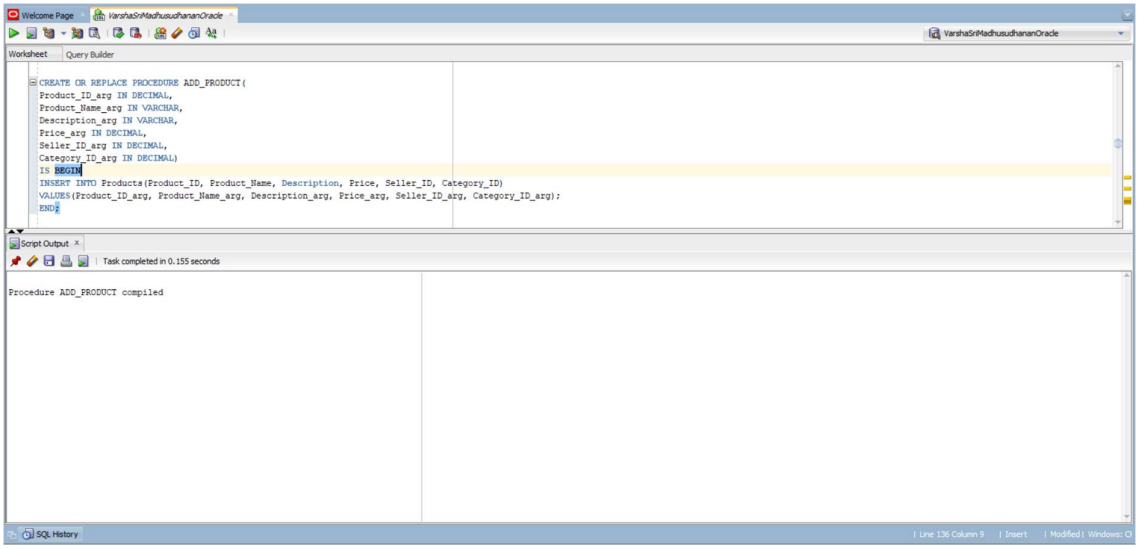




b. Developing a parameterized stored procedure that is used when a seller needs to add any new product.

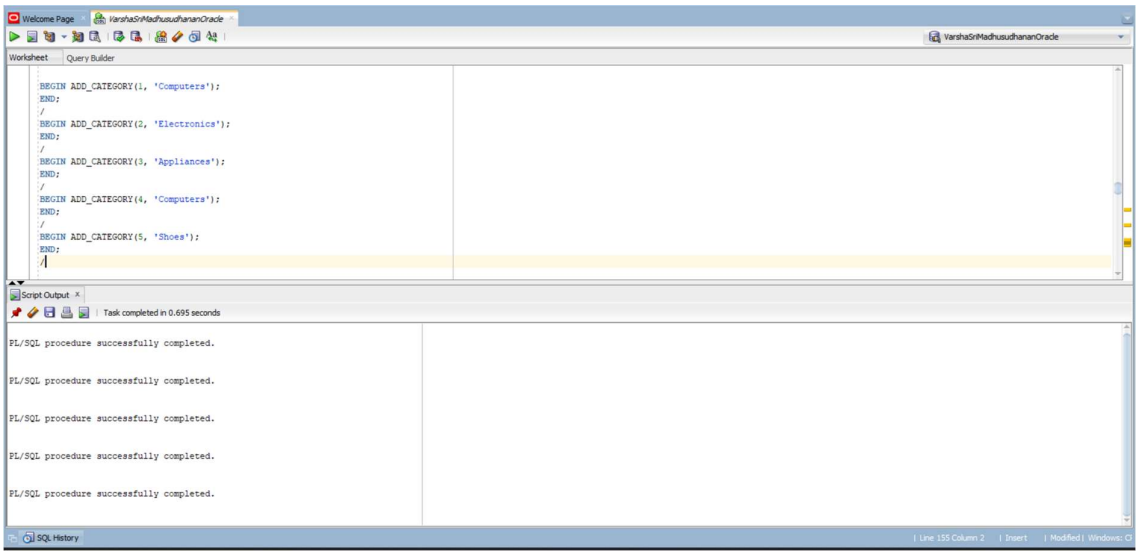






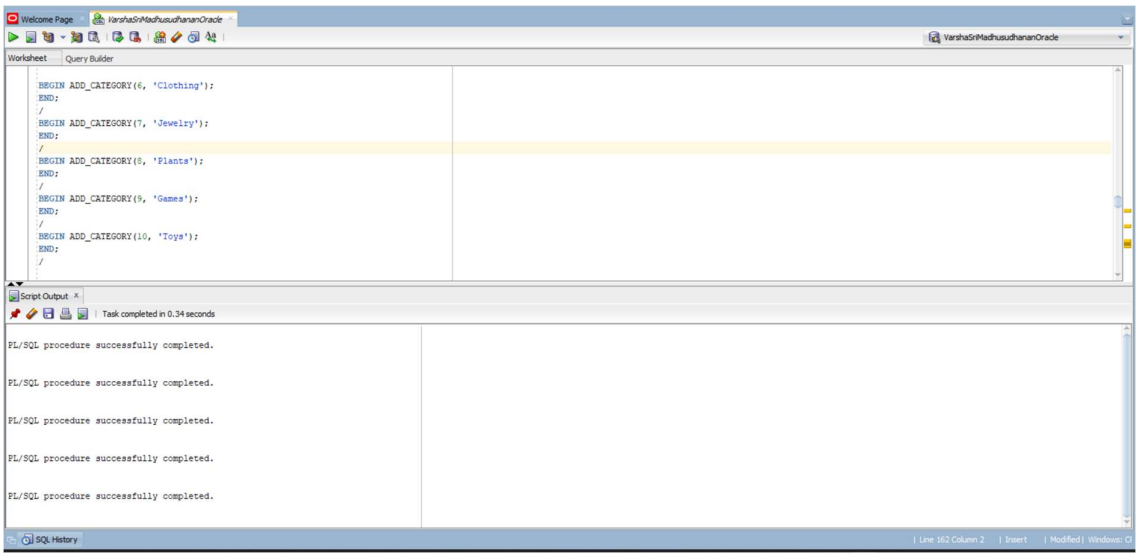
```
CREATE OR REPLACE PROCEDURE ADD_PRODUCT(
Product_ID_arg IN DECIMAL,
Product_Name_arg IN VARCHAR,
Description_arg IN VARCHAR,
Price_arg IN DECIMAL,
Seller_ID_arg IN DECIMAL,
Category_ID_arg IN DECIMAL)
IS BEGIN
INSERT INTO Products(Product_ID, Product_Name, Description, Price, Seller_ID, Category_ID)
VALUES(Product_ID_arg, Product_Name_arg, Description_arg, Price_arg, Seller_ID_arg, Category_ID_arg);
END;
```

```
Procedure ADD_PRODUCT compiled
```



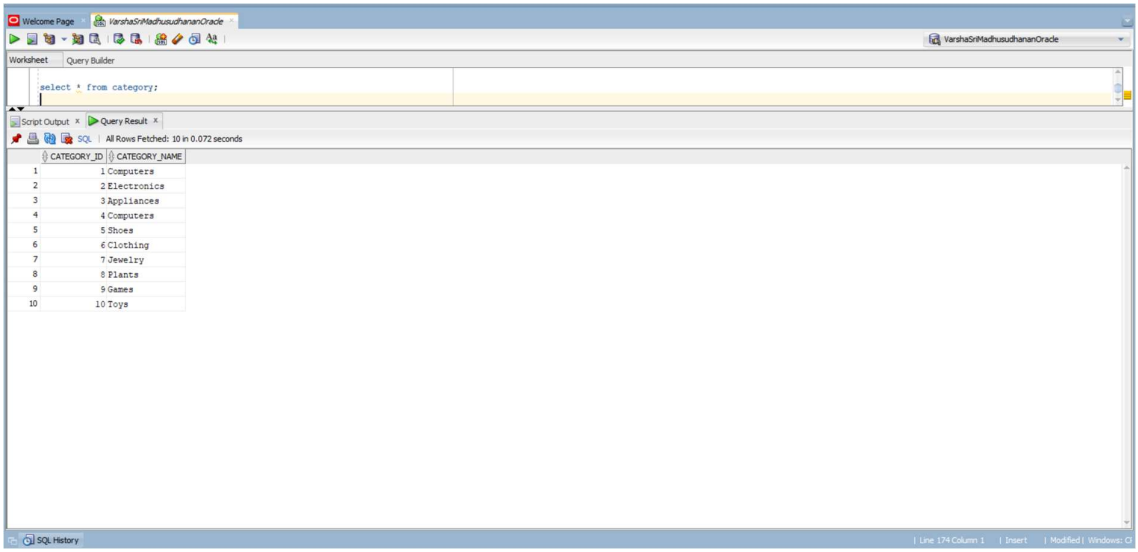
```
BEGIN ADD_CATEGORY(1, 'Computers');
END;
/
BEGIN ADD_CATEGORY(2, 'Electronics');
END;
/
BEGIN ADD_CATEGORY(3, 'Appliances');
END;
/
BEGIN ADD_CATEGORY(4, 'Computers');
END;
/
BEGIN ADD_CATEGORY(5, 'Shoes');
END;
/
```

```
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```



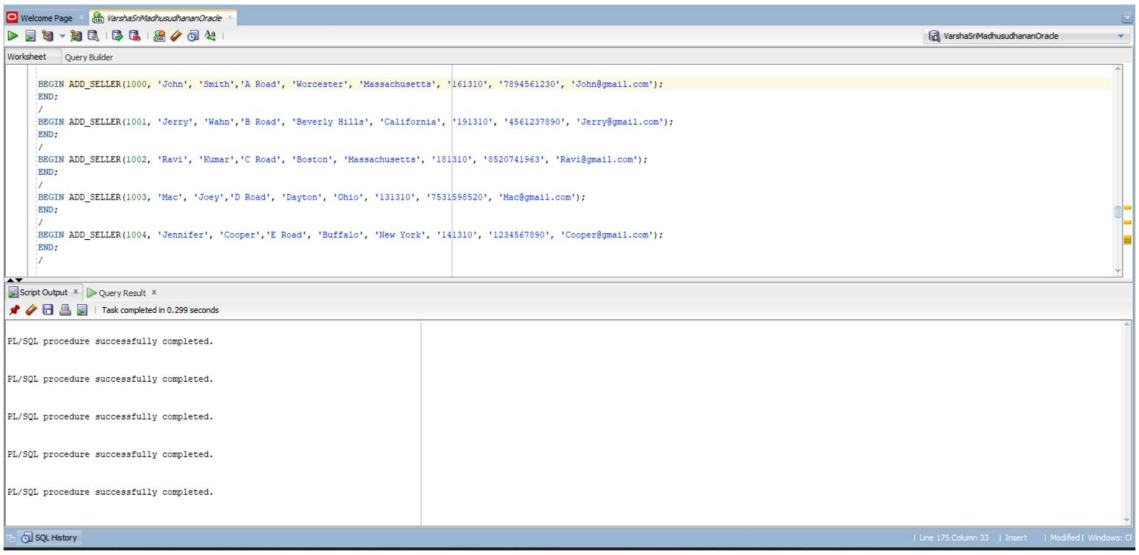
```
BEGIN ADD_CATEGORY(6, 'Clothing');
END;
/
BEGIN ADD_CATEGORY(7, 'Jewelry');
END;
/
BEGIN ADD_CATEGORY(8, 'Plants');
END;
/
BEGIN ADD_CATEGORY(9, 'Games');
END;
/
BEGIN ADD_CATEGORY(10, 'Toys');
END;
/
```

```
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```



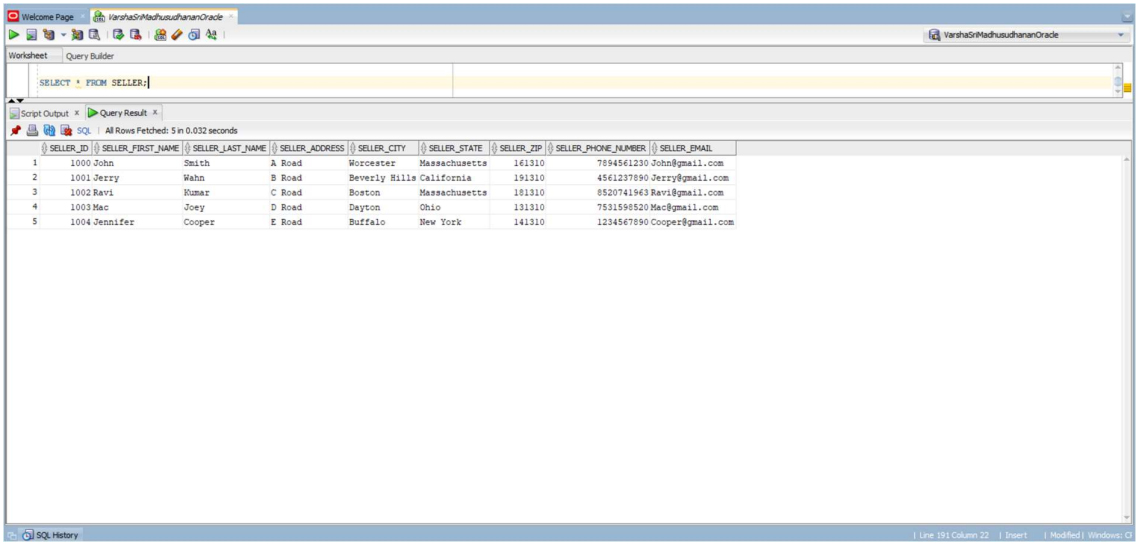
```
select * from category;
```

CATEGORY_ID	CATEGORY_NAME
1	Computers
2	Electronics
3	Appliances
4	Computers
5	Shoes
6	Clothing
7	Jewelry
8	Plants
9	Games
10	Toys



```
BEGIN ADD_SELLER(1000, 'John', 'Smith','A Road', 'Worcester', 'Massachusetts', '161310', '7894561230', 'John@gmail.com');
END;
/
BEGIN ADD_SELLER(1001, 'Jerry', 'Wahn','B Road', 'Beverly Hills', 'California', '191310', '4561237890', 'Jerry@gmail.com');
END;
/
BEGIN ADD_SELLER(1002, 'Ravi', 'Kumar','C Road', 'Boston', 'Massachusetts', '181310', '8520741963', 'Ravi@gmail.com');
END;
/
BEGIN ADD_SELLER(1003, 'Mac', 'Joey','D Road', 'Dayton', 'Ohio', '131310', '7531598520', 'Mac@gmail.com');
END;
/
BEGIN ADD_SELLER(1004, 'Jennifer', 'Cooper','E Road', 'Buffalo', 'New York', '141310', '1234567890', 'Cooper@gmail.com');
END;
/
```

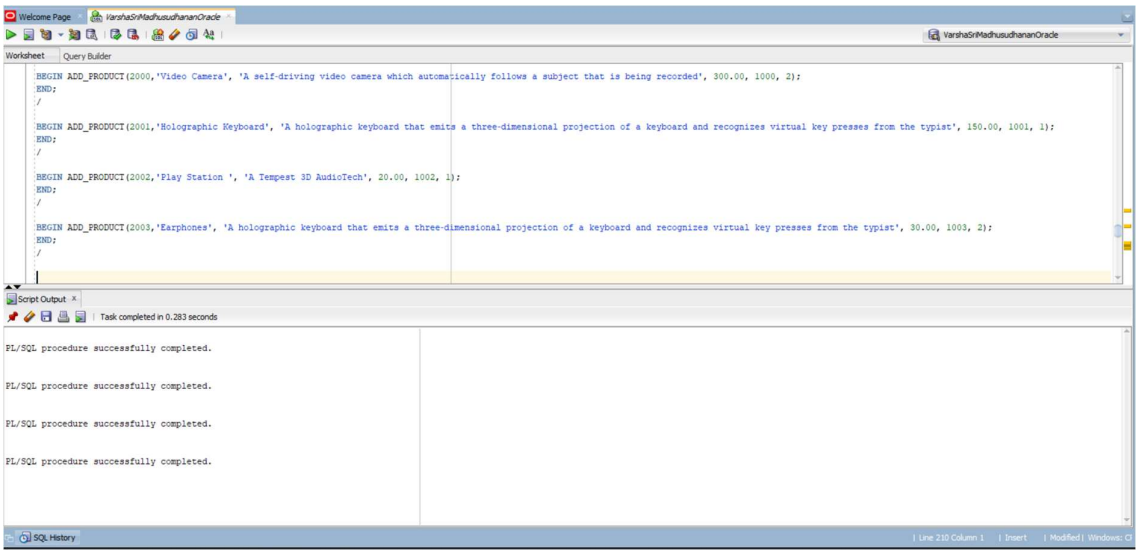
```
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```

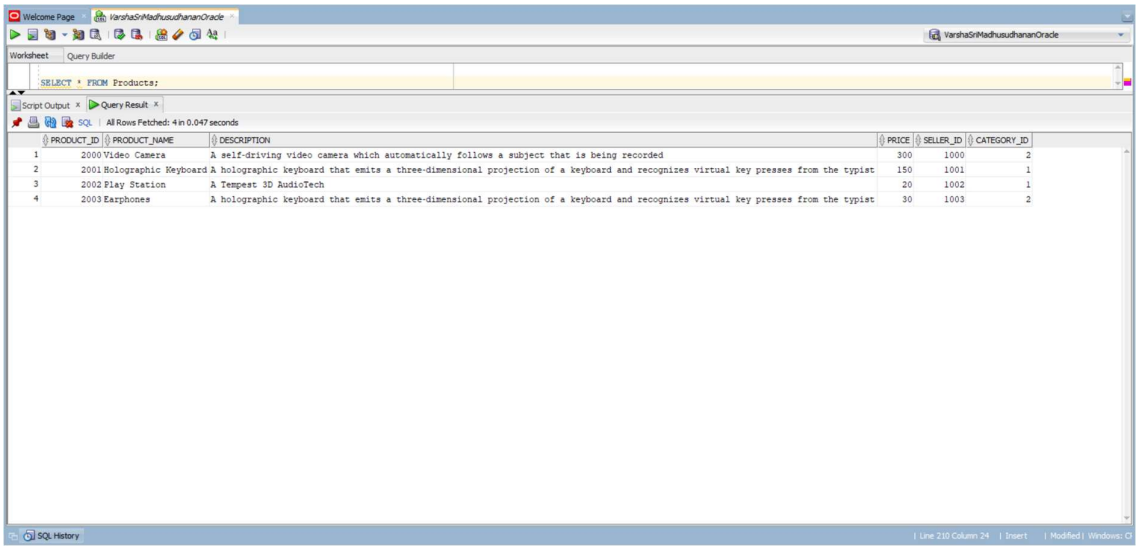


```
SELECT * FROM SELLER;
```

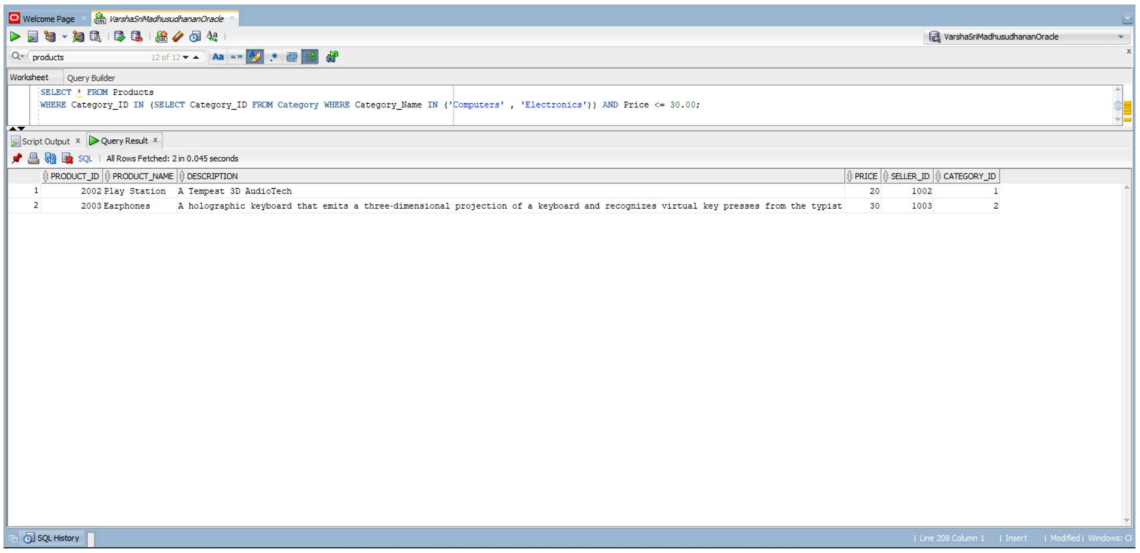
SELLER_ID	SELLER_FIRST_NAME	SELLER_LAST_NAME	SELLER_ADDRESS	SELLER_CITY	SELLER_STATE	SELLER_ZIP	SELLER_PHONE_NUMBER	SELLER_EMAIL
1	1000 John	Smith	A Road	Worcester	Massachusetts	161310	7894561230	John@gmail.com
2	1001 Jerry	Wahn	B Road	Beverly Hills	California	191310	4561237890	Jerry@gmail.com
3	1002 Ravi	Kumar	C Road	Boston	Massachusetts	181310	8520741963	Ravi@gmail.com
4	1003 Mac	Joey	D Road	Dayton	Ohio	131310	7531598520	Mac@gmail.com
5	1004 Jennifer	Cooper	E Road	Buffalo	New York	141310	1234567890	Cooper@gmail.com

c. A seller adds two new products. The first is a self-driving video camera which automatically follows a subject that is being recorded. The second is a holographic keyboard that emits a three-dimensional projection of a keyboard and recognizes virtual key presses from the typist. Invoking the stored procedure twice to add these products, which have at a minimum a name, description, price, and category in its attributes.



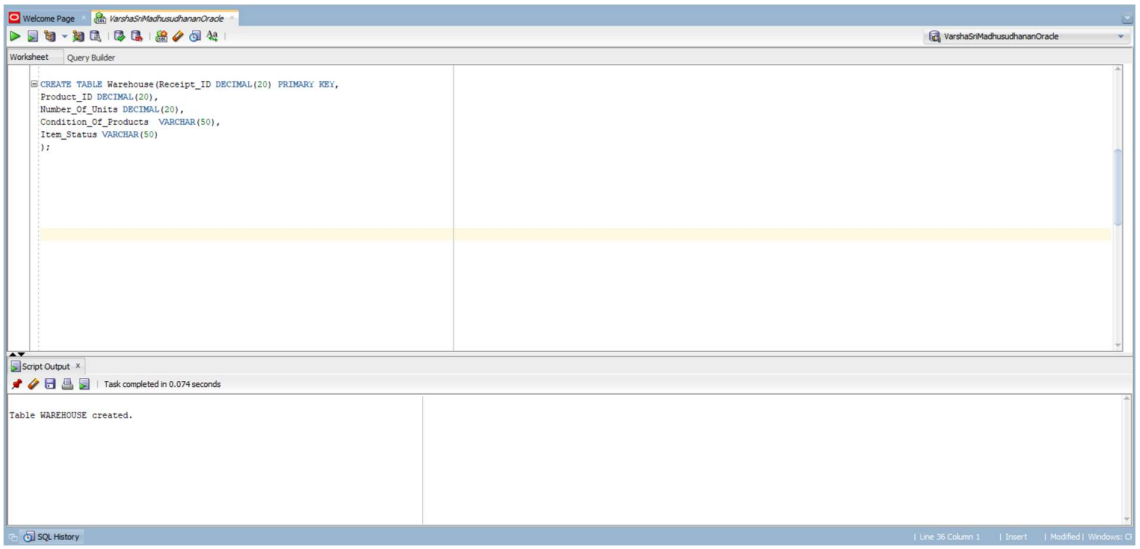


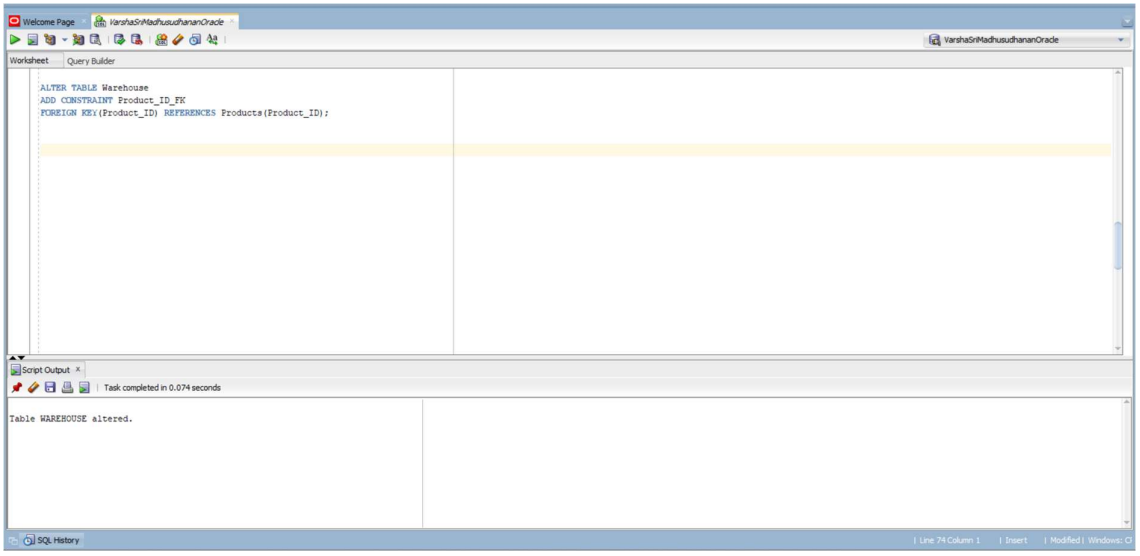
d. A seller is considering developing a new electronic product and requests a list of existing products in the “Computers” or “Electronics” categories that cost $30 or less.



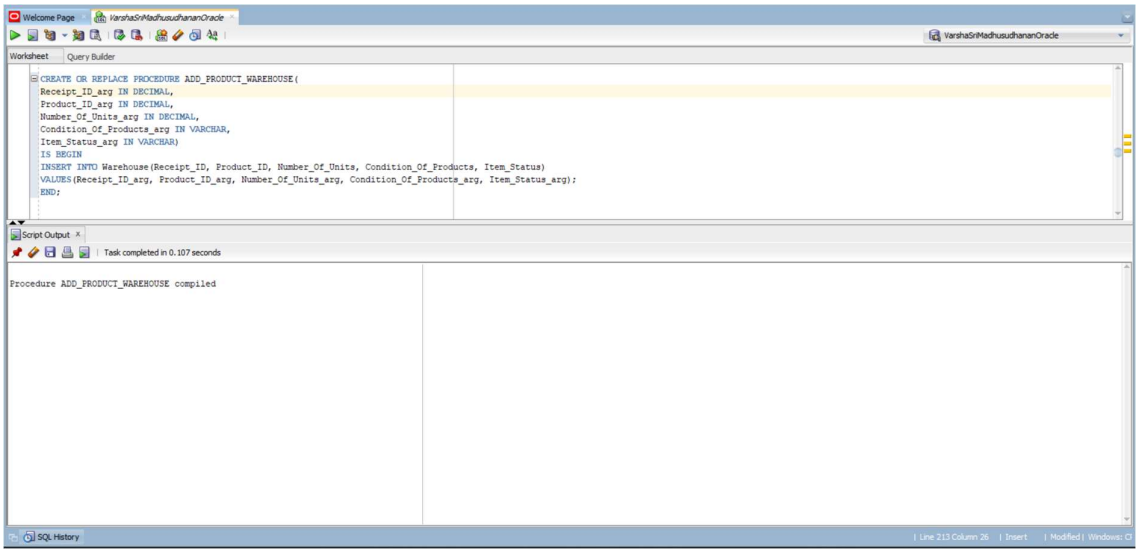
ASPECT 2: AMAZON RECEIPT OF PRODUCT FROM SELLER

a. Creating the tables, constraints, and data needed to support product delivery.

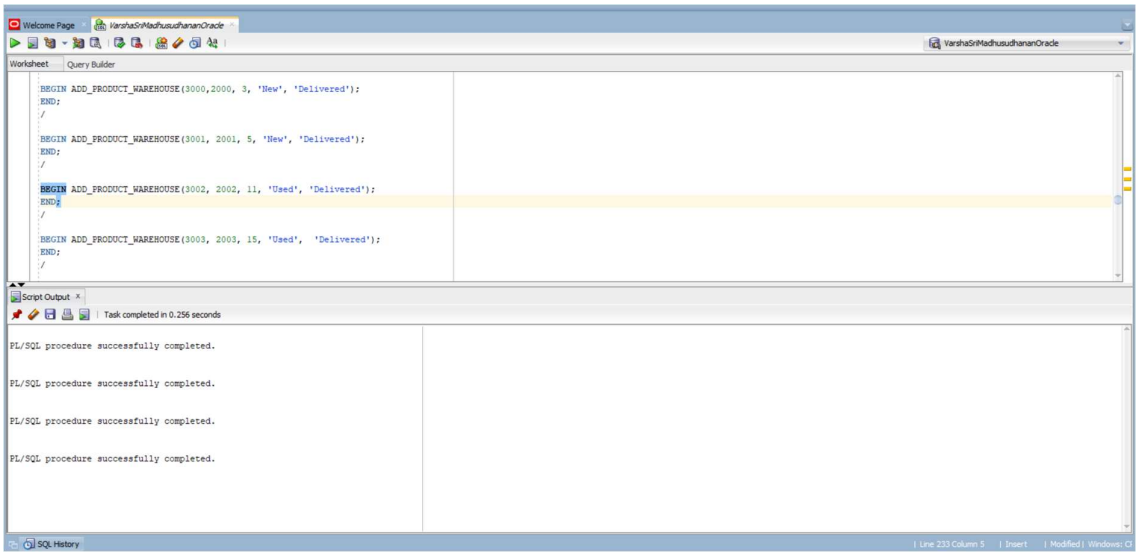


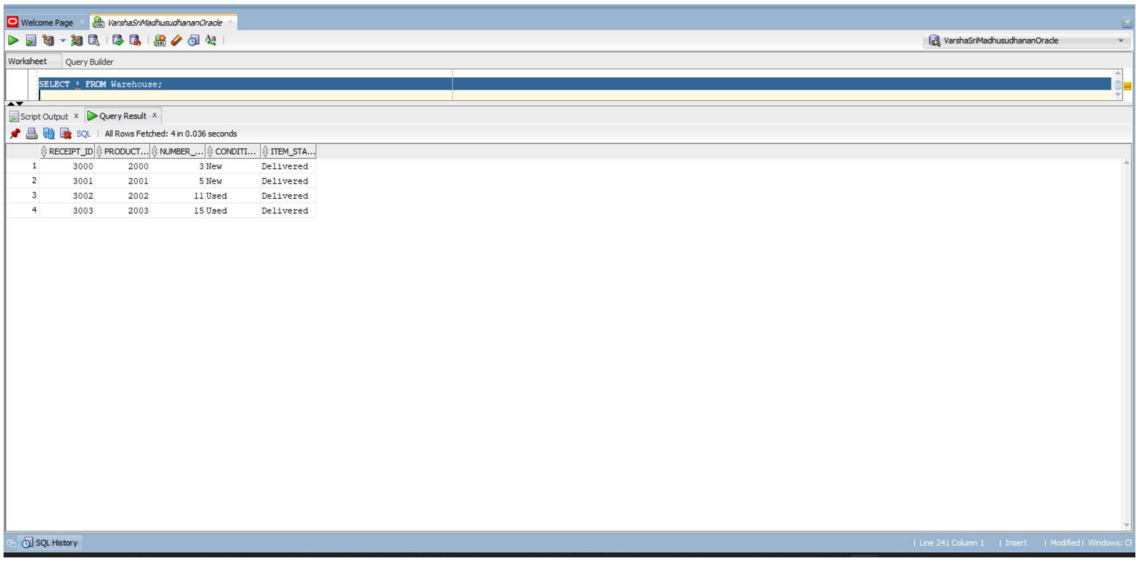


b. Developing a parameterized stored procedure that is used when any seller delivers any product to Amazon's warehouse.

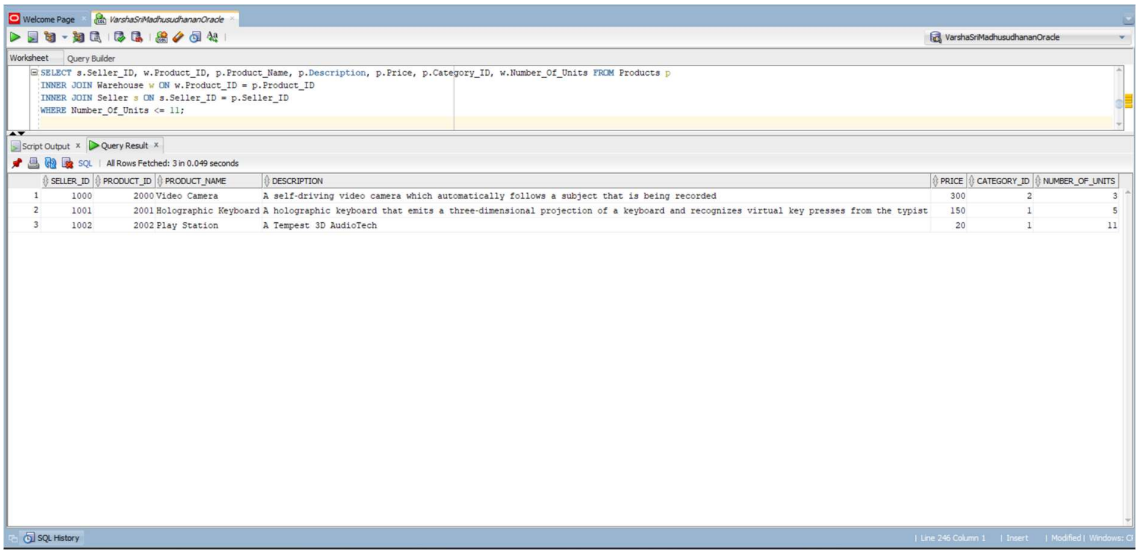


c. A seller delivers four of each of the two new products added in Aspect 1 (the self-driving video camera and the holographic keyboard). Invoking the stored procedure twice to update the inventory of these products for a seller.



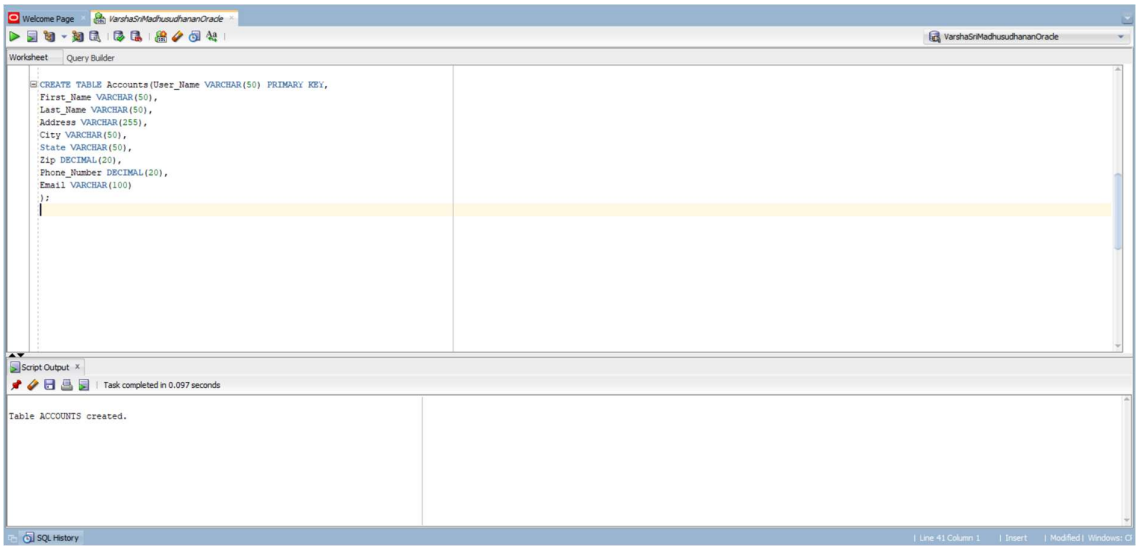


d. The seller from b above requests a listing of all of its products that have an inventory of 11 or less. (the self-driving video camera and holographic keyboard are included in the listed).

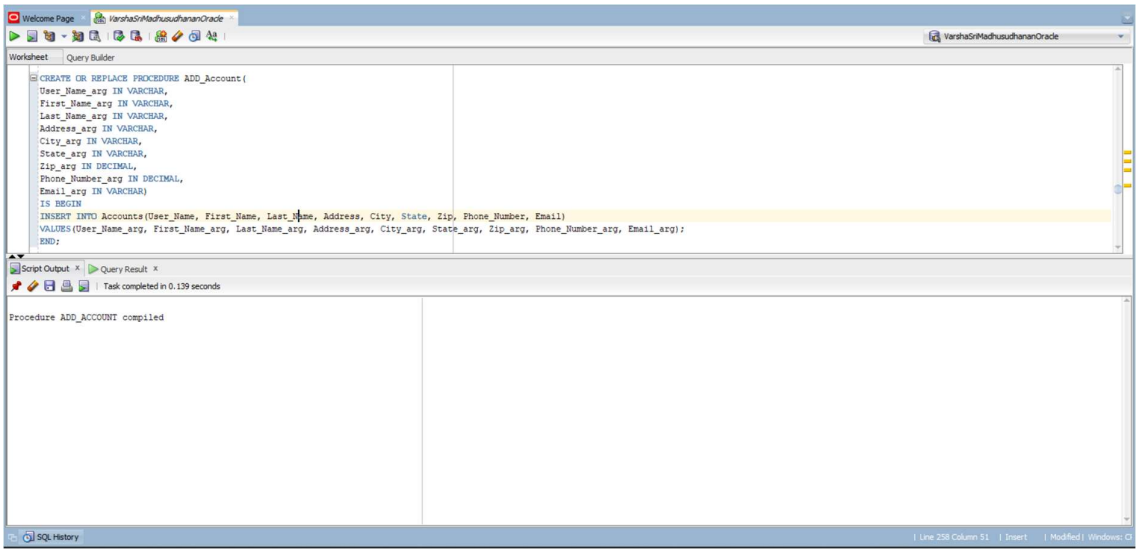


ASPECT 3: NEW CONSUMER ACCOUNT

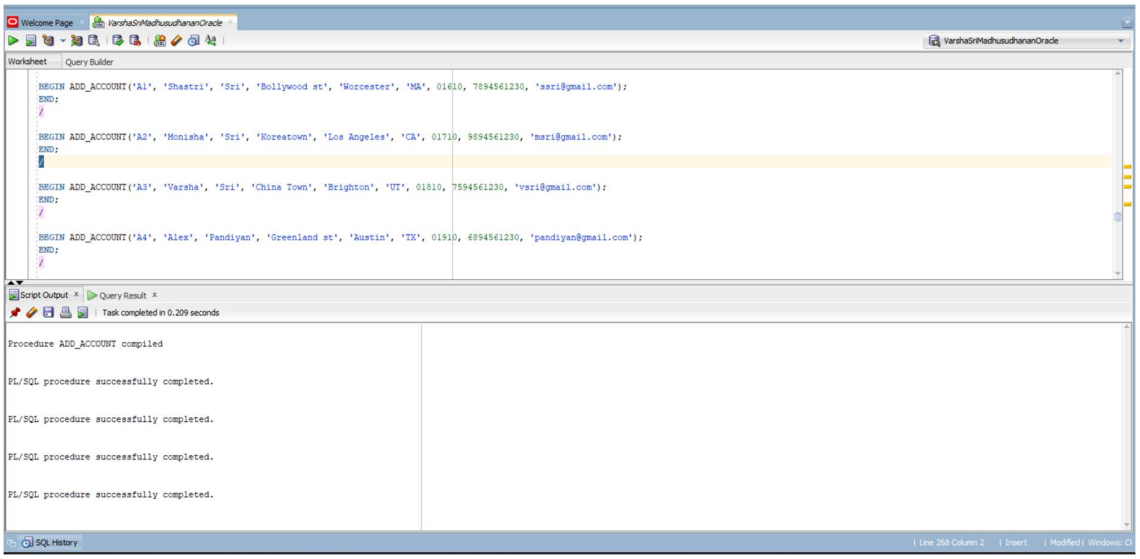
a. Creating the tables, constraints, and data needed to support customer accounts.

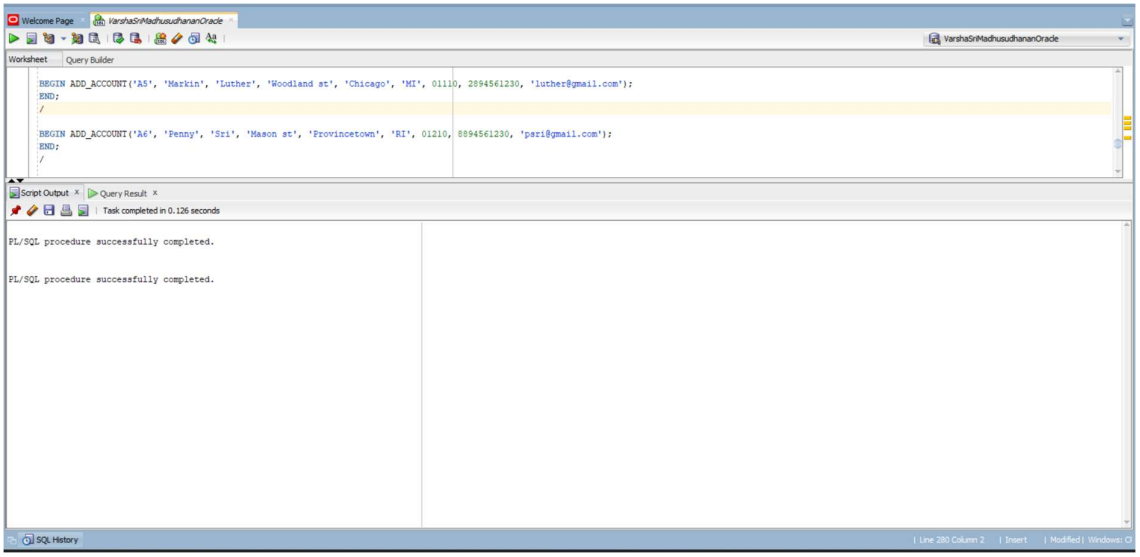


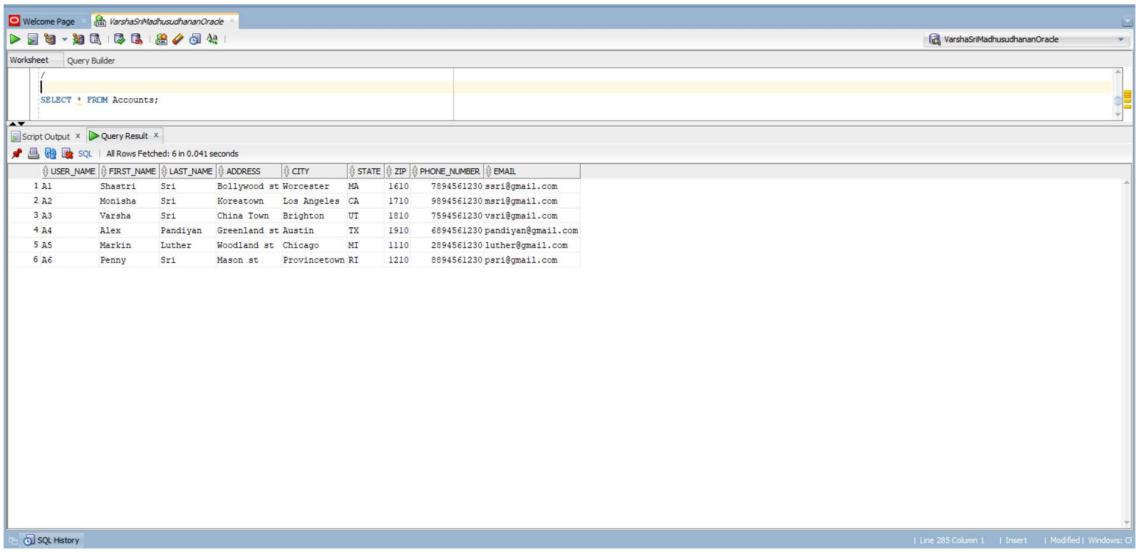
b. Developing a parameterized stored procedure that is used when any new customer signs up for a new account on Amazon.



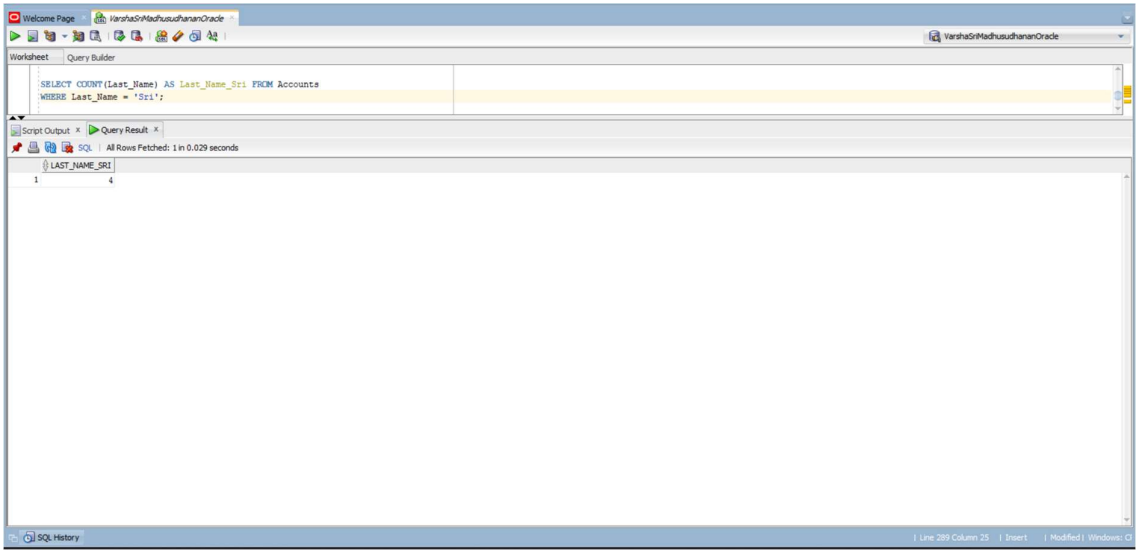
c. You and your facilitator sign up for new accounts on Amazon. Invoking the stored procedure twice to add me and my facilitator as customers.




Welcome Page
VarshaSriMadhusudhananOracle
VarshaSriMadhusudhananOracle
Worksheet
Query Builder
BEGIN ADD_ACCOUNT('A5', 'Markin', 'Luther', 'Woodland st', 'Chicago', 'MI', 01110, 2894561230, 'luther@gmail.com');
END;
/
BEGIN ADD_ACCOUNT('A6', 'Penny', 'Sri', 'Mason st', 'Provincetown', 'RI', 01210, 8894561230, 'psri@gmail.com');
END;
/
Script Output
Query Result
Task completed in 0.126 seconds
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
SQL History

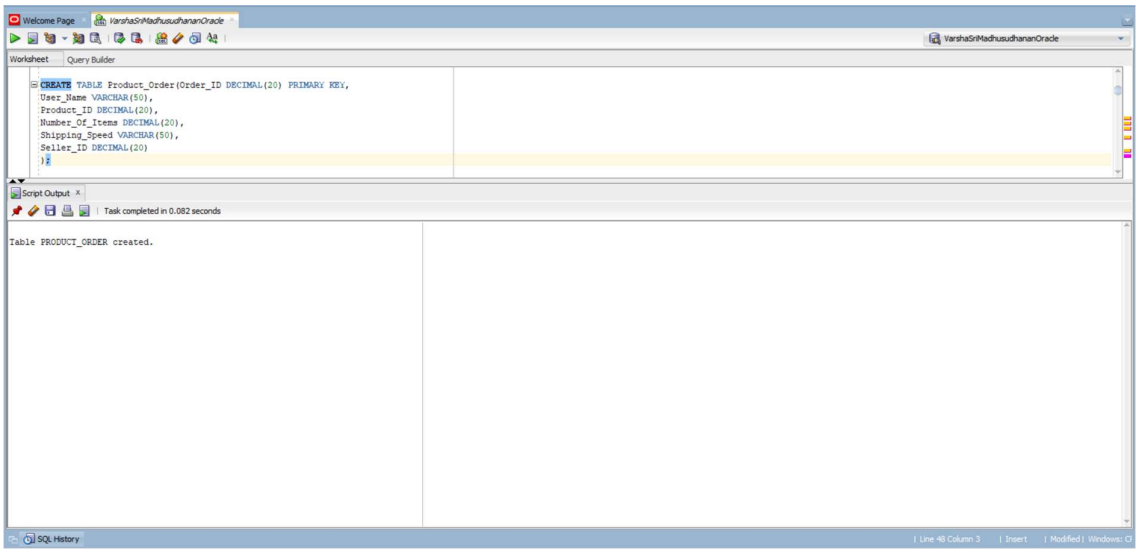

Welcome Page
VarshaSriMadhusudhananOracle
VarshaSriMadhusudhananOracle
Worksheet
Query Builder
/
SELECT * FROM Accounts;
Script Output
Query Result
SQL
All Rows Fetched: 6 in 0.041 seconds
USER_NAME FIRST_NAME LAST_NAME ADDRESS CITY STATE ZIP PHONE_NUMBER EMAIL
1 A1 Shastri Sri Bollywood st Worcester MA 1610 7894561230 ssri@gmail.com
2 A2 Monisha Sri Koreatown Los Angeles CA 1710 9894561230 msri@gmail.com
3 A3 Varsha Sri China Town Brighton UT 1810 7594561230 vsri@gmail.com
4 A4 Alex Pandiyan Greenland st Austin TX 1910 6894561230 pandiyan@gmail.com
5 A5 Markin Luther Woodland st Chicago MI 1110 2894561230 luther@gmail.com
6 A6 Penny Sri Mason st Provincetown RI 1210 8894561230 psri@gmail.com
SQL History

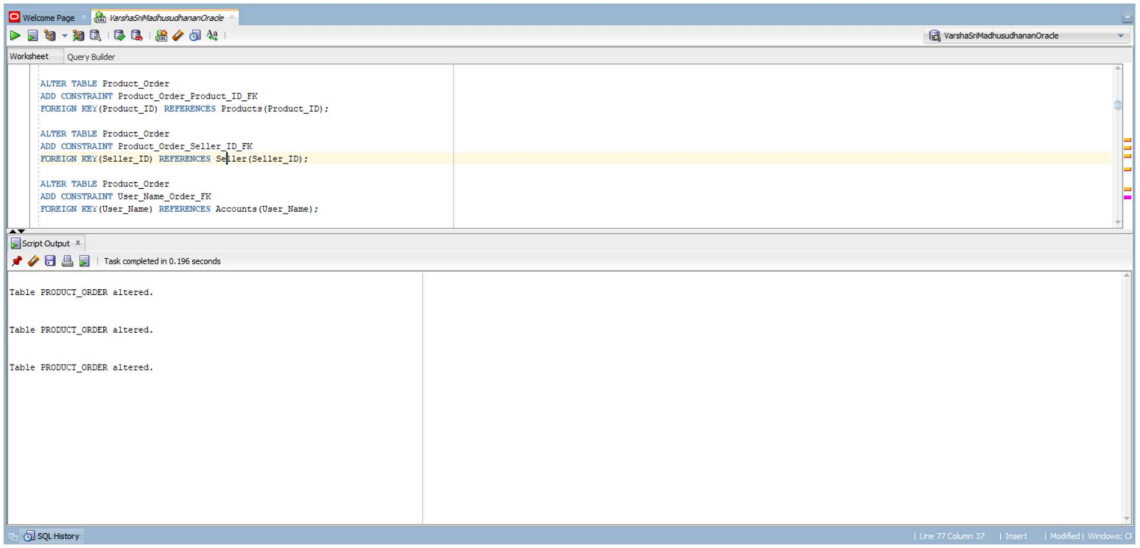
d. For research purposes, Amazon requests the last names of consumers where there are at least 4 accounts associated with the last name. Amazon would like to see the actual number of accounts associated with those last names. Developing and executing a single query that helps the above request.



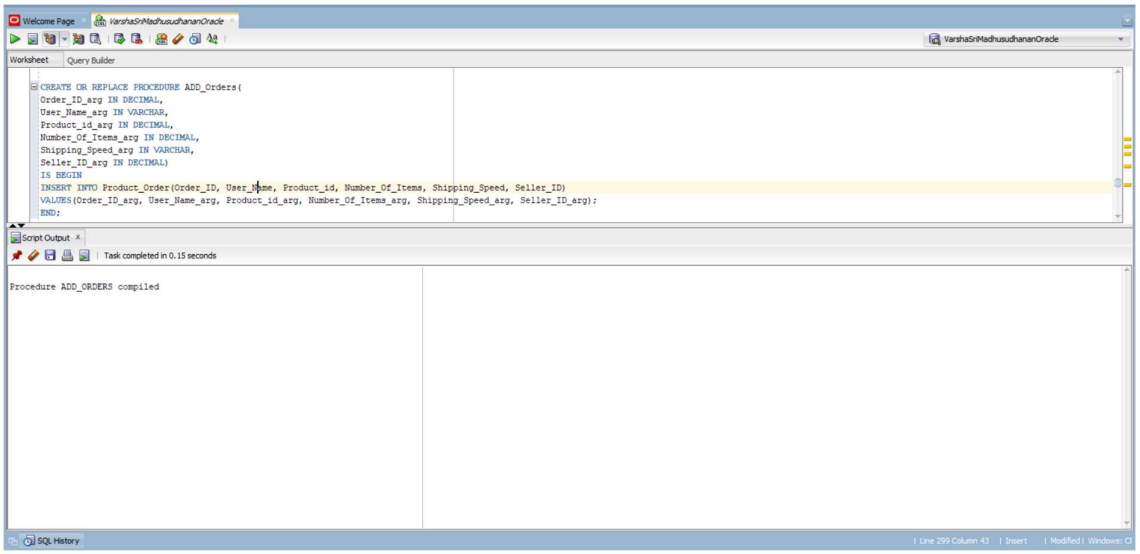
ASPECT 4: PRODUCT PURCHASE BY CONSUMER

a. Creating the tables, constraints, and data needed to support product purchases.

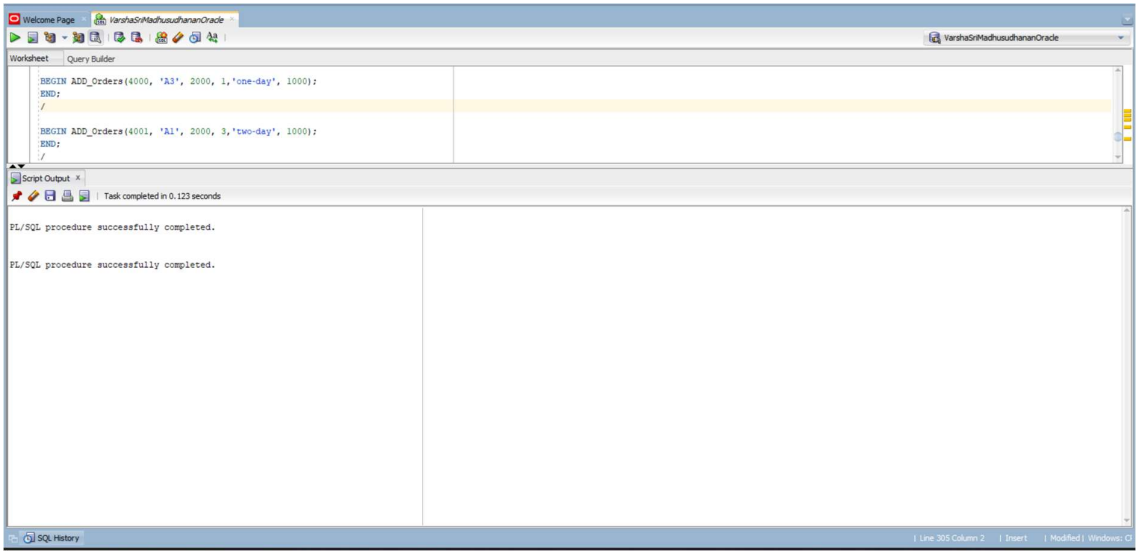




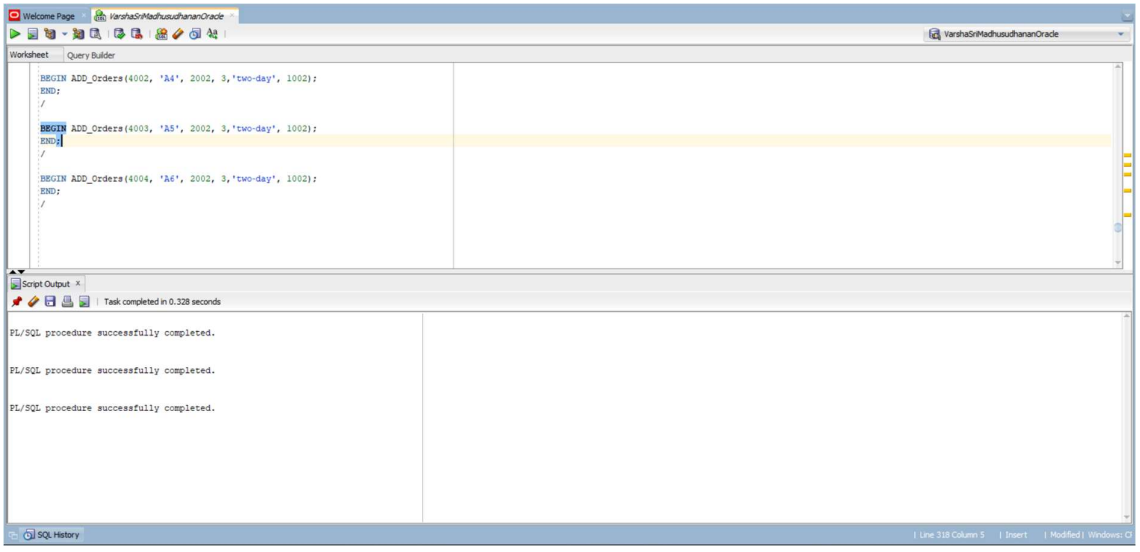
b. Developing a parameterized stored procedure that is used when any customer purchases any product.

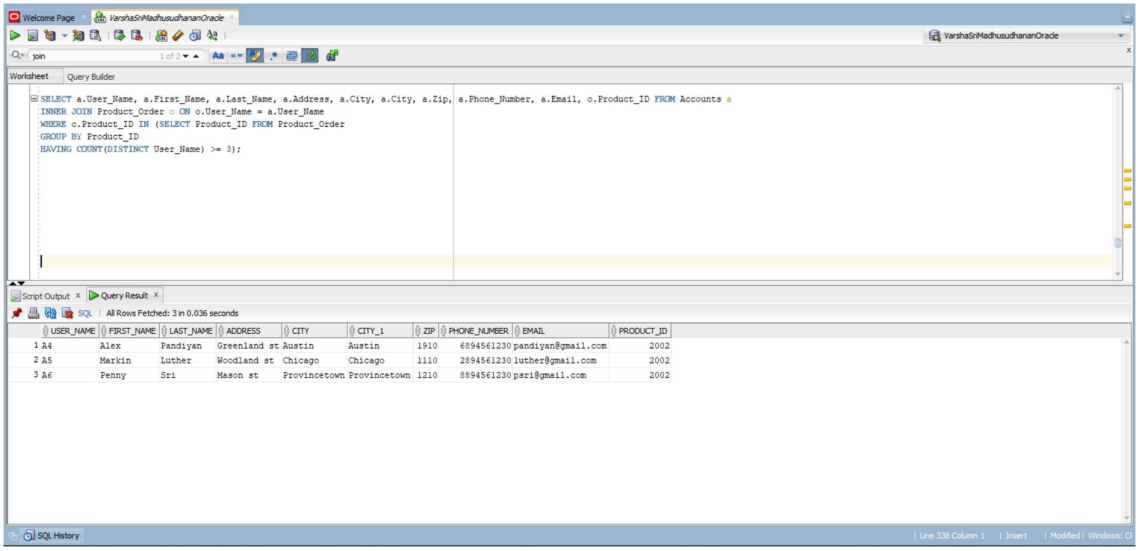


c. Purchasing a self-driving video camera (from Aspect 1), and , my facilitator purchases three holographic keyboards. Invoking the stored procedure twice, once for each purchase.



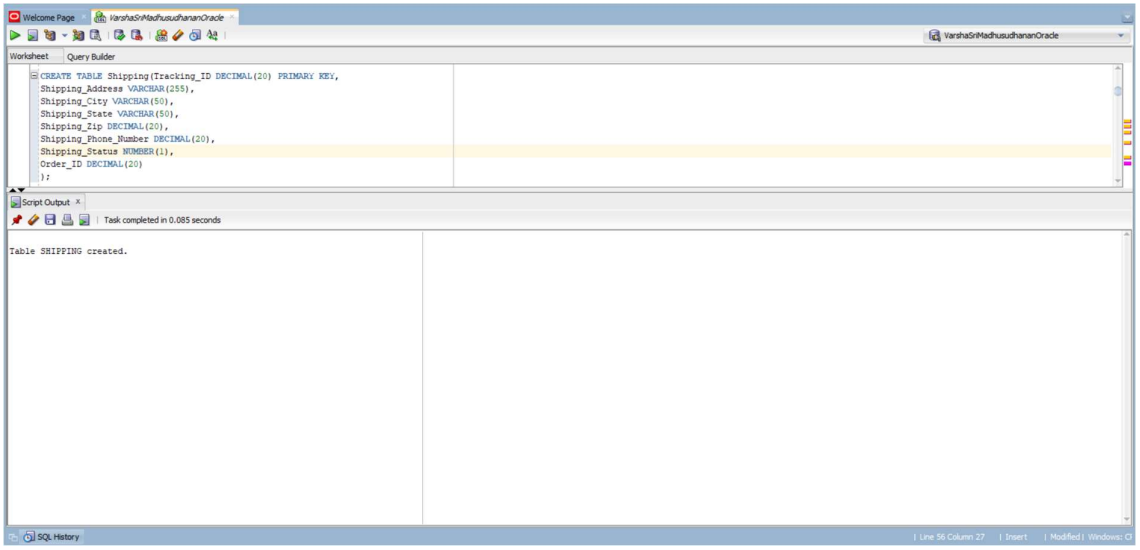
d. The marketing department at Amazon wants to reach out to consumers who buy popular products. The department requests the names and addresses of all consumers who bought any product that was purchased by at least three different people.

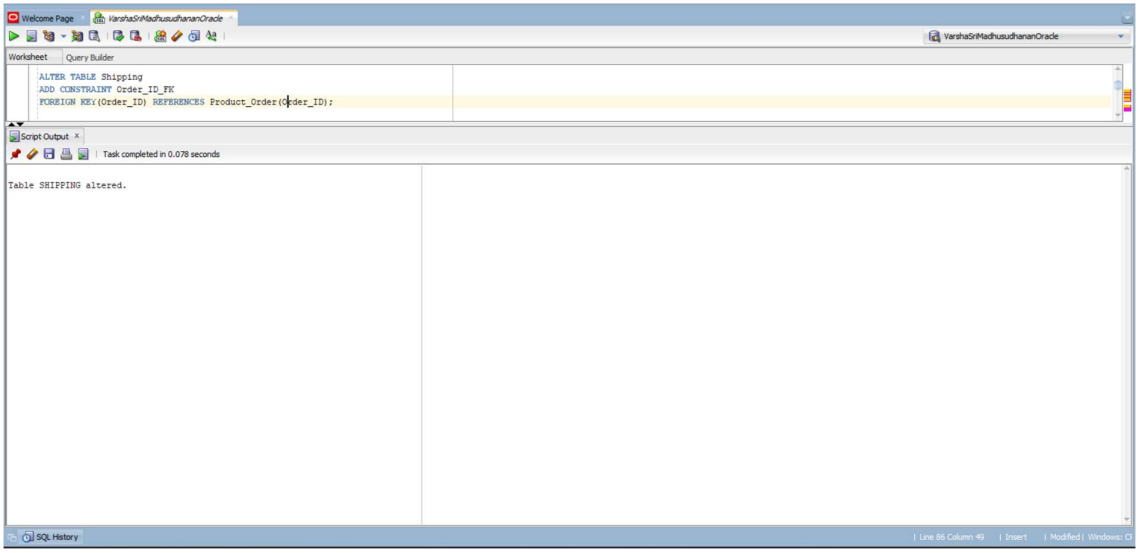




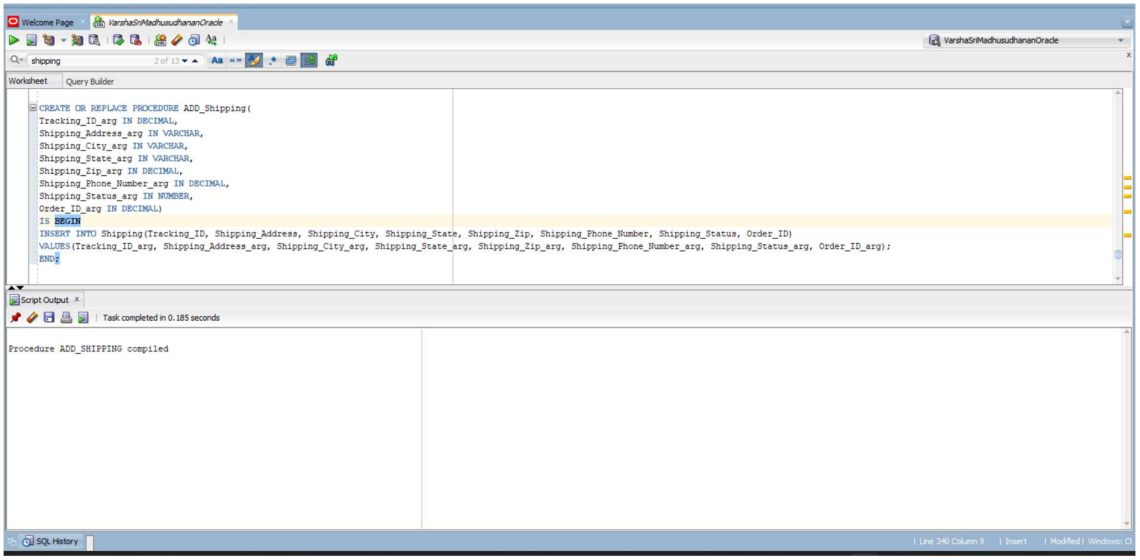
ASPECT 5: PRODUCT SHIPMENT BY AMAZON

a. Creating the tables, constraints, and data needed to support product shipments.

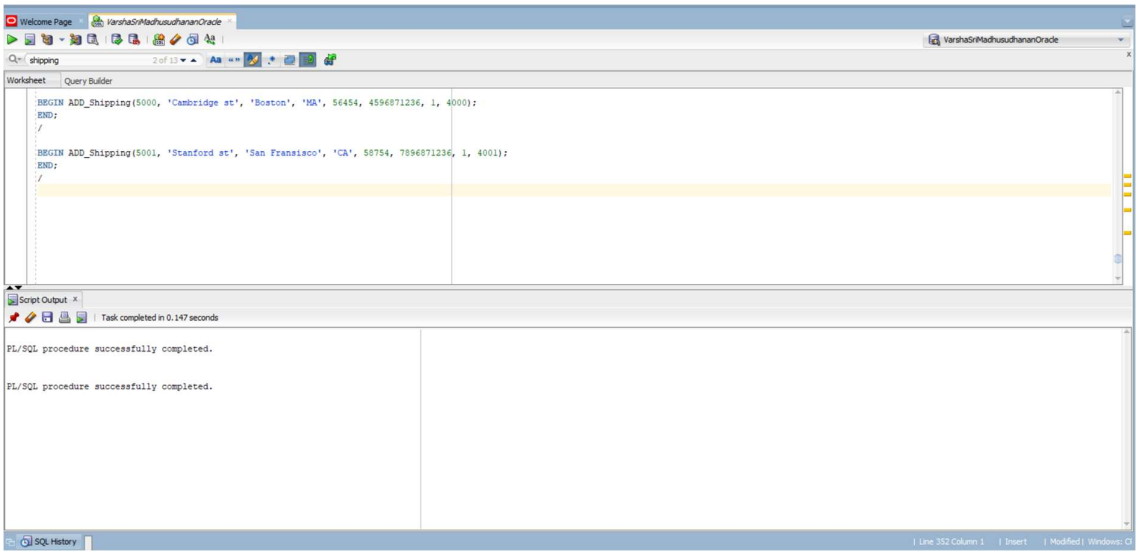


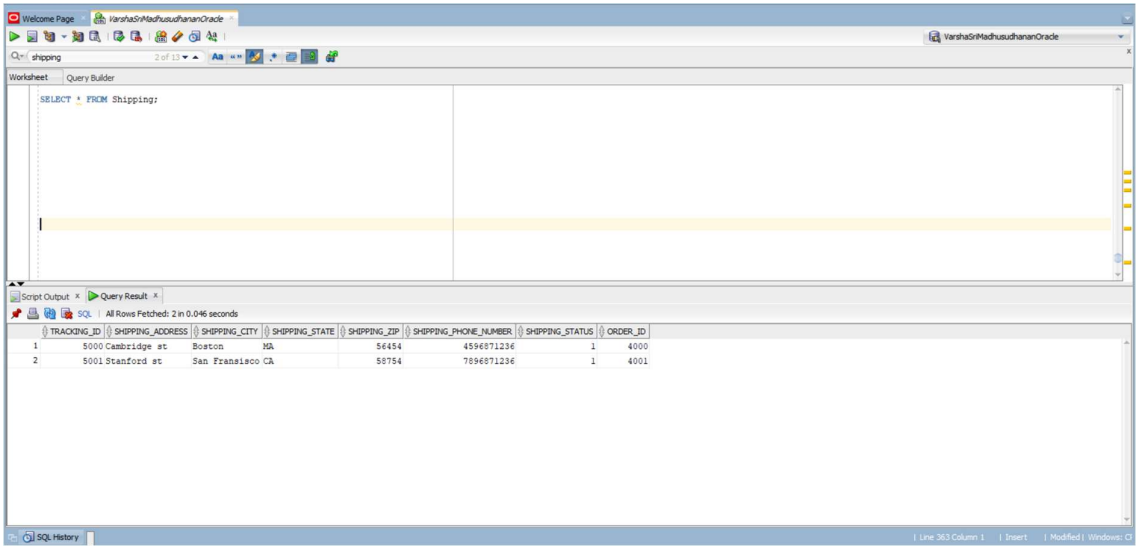


b. Developing a parameterized stored procedure that is used when Amazon ships any order.

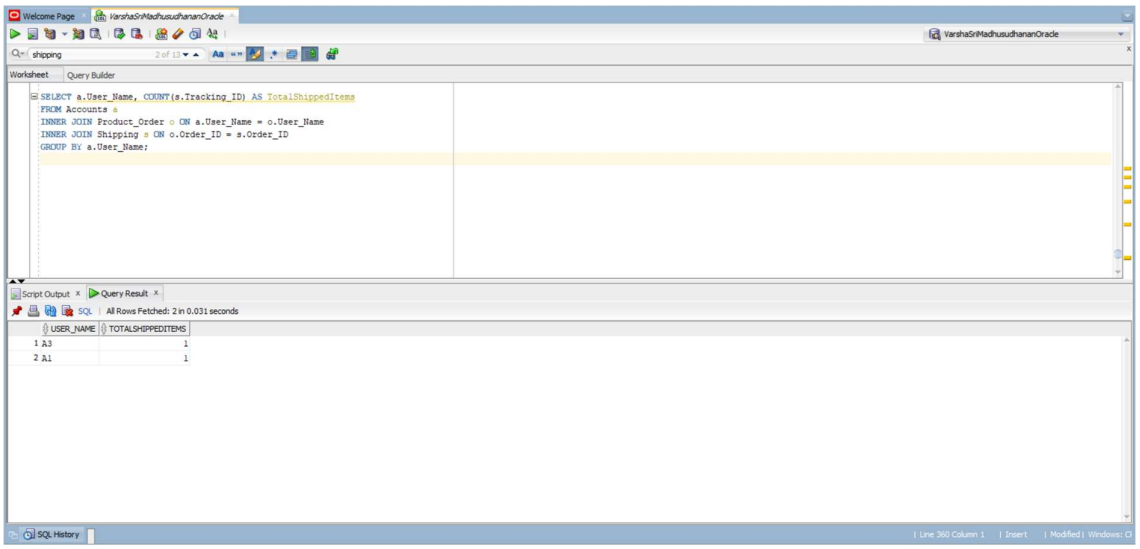


c. Amazon ships the orders listed in Aspect 4, one to you and the other to your facilitator. Invoking the stored procedure twice, once for each order.

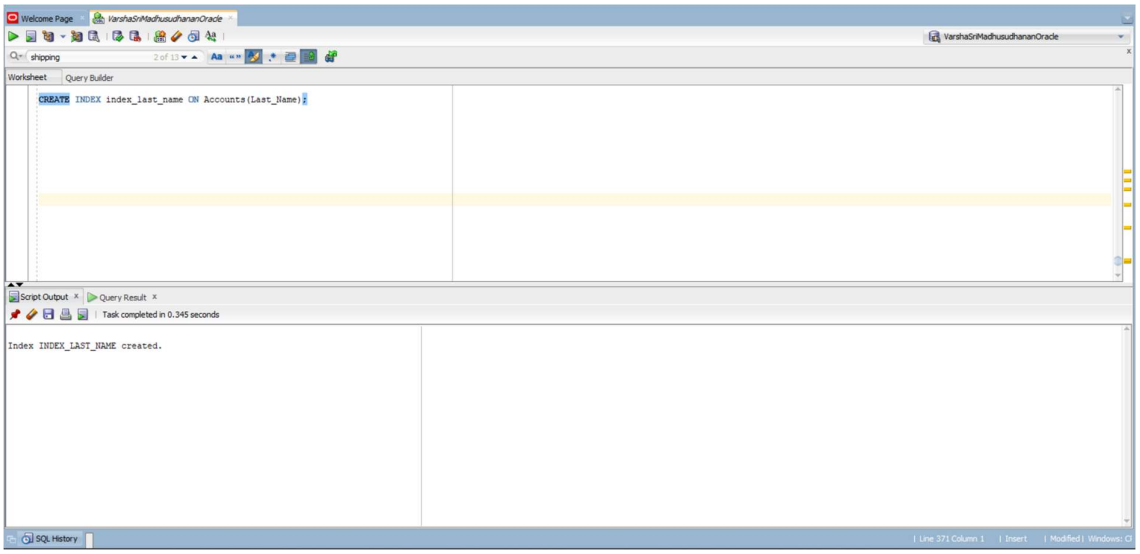


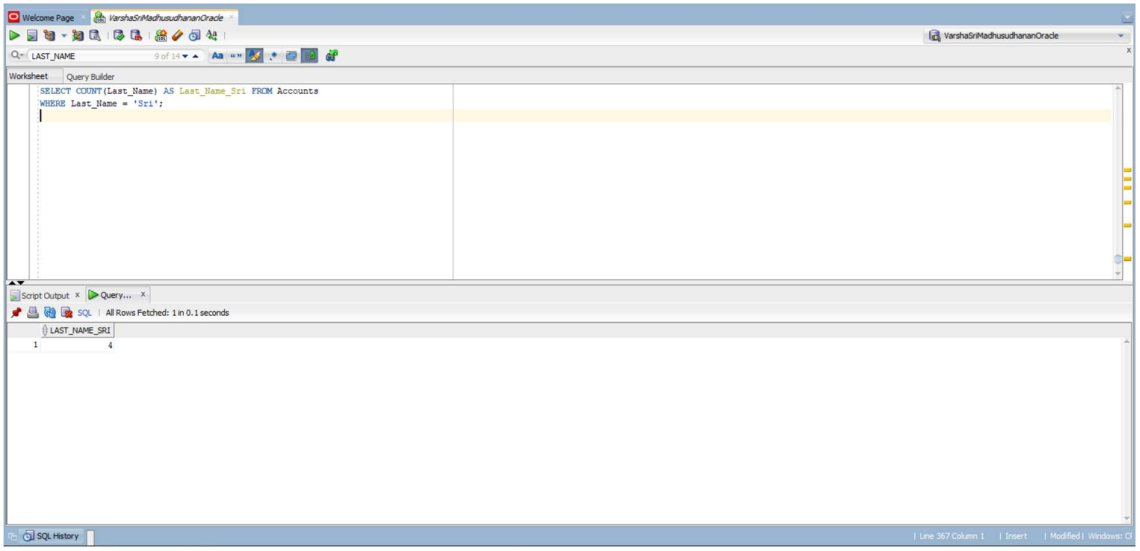


d. Calculating total shipped items for each user.



4: Index Justification and Creation



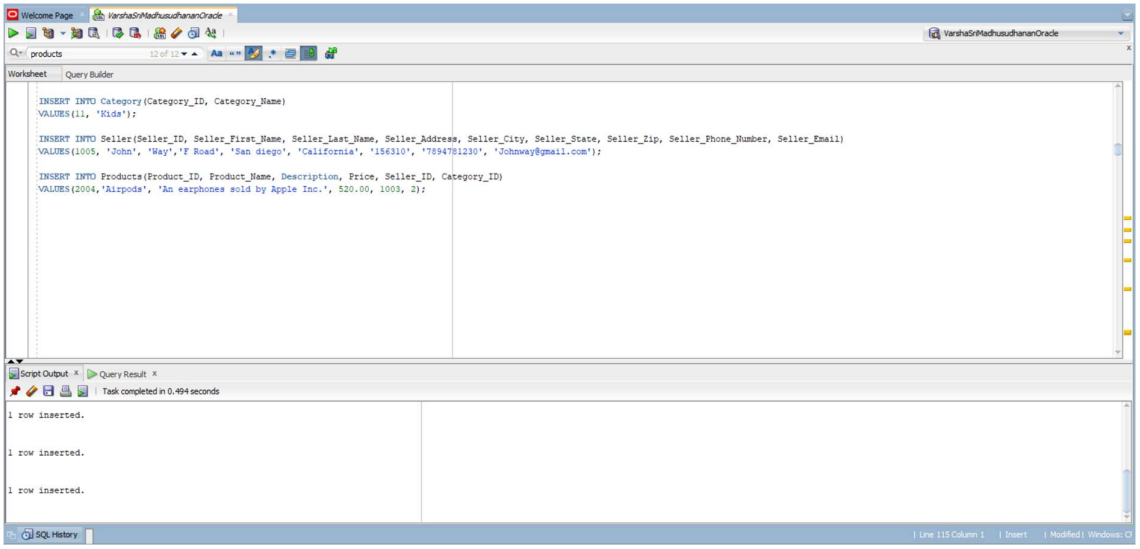


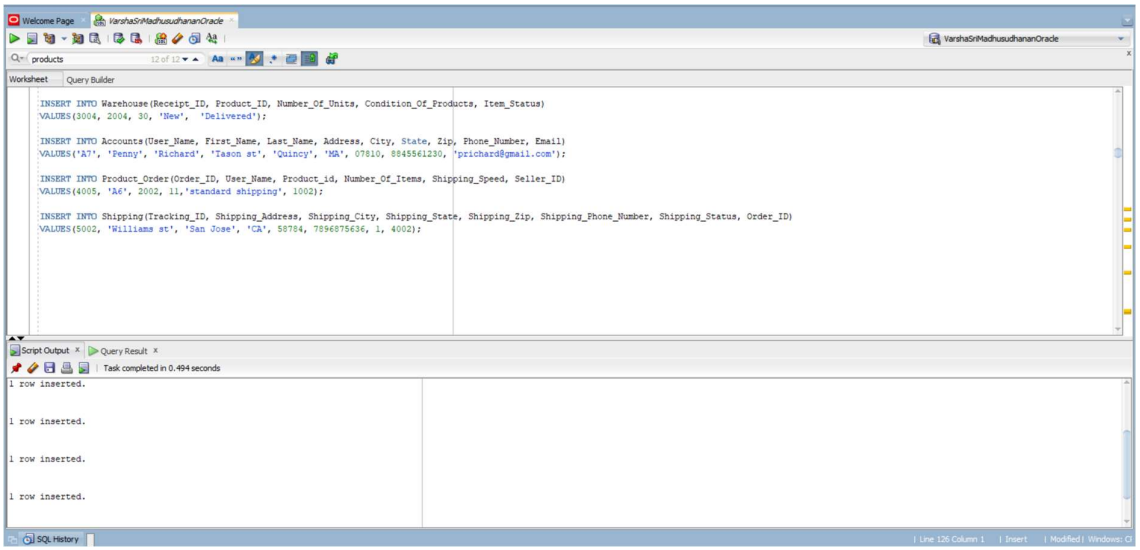
EXPLANATION:

Here, I have created an index for Last_Name in the Accounts table. An index is used to access the values from the table quickly and to increase the performance speed while executing. In the above scenario, the index can improve the performance of the search based on the user's last name in the query. When searching for a specific user (identified by last name), having an index on the Last_Name column allows the database engine to quickly locate the relevant rows in the Accounts table.

For example, the above-given query is used in Aspect 3 to retrieve the Last_Name of the users whose name is repeated more than 3 times, compiler acts quickly while the index is added. The previous query in Aspect 3 it takes 4 milliseconds after adding index it only takes 1 or 2 milliseconds.

INSERTING VALUES:
Note: these are not part of the aspects, if wanted to insert additional values these statements can be used or even the stored procedures can be invoked again.




Welcome Page
VarshaSriMadhusudhananOracle
products
Worksheet
Query Builder
INSERT INTO Warehouse(Receipt_ID, Product_ID, Number_Of_Units, Condition_Of_Products, Item_Status)
VALUES(3004, 2004, 30, 'New', 'Delivered');
INSERT INTO Accounts(User_Name, First_Name, Last_Name, Address, City, State, Zip, Phone_Number, Email)
VALUES('A7', 'Penny', 'Richard', 'Tason st', 'Quincy', 'MA', 07810, 8845561230, 'prichard@gmail.com');
INSERT INTO Product_Order(Order_ID, User_Name, Product_id, Number_Of_Items, Shipping_Speed, Seller_ID)
VALUES(4005, 'A6', 2002, 11,'standard shipping', 1002);
INSERT INTO Shipping(Tracking_ID, Shipping_Address, Shipping_City, Shipping_State, Shipping_Zip, Shipping_Phone_Number, Shipping_Status, Order_ID)
VALUES(5002, 'Williams st', 'San Jose', 'CA', 58784, 7896875636, 1, 4002);
Script Output
Query Result
Task completed in 0.494 seconds
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
SQL History