

Term Project – Deliverable 4

The changes that I have made to the business rule are as follows:

1. Added " product_catagory " as an attribute in the Product table
2. Added "product_price" as an attribute in the Product table

Creating tables and introducing some test values into the table

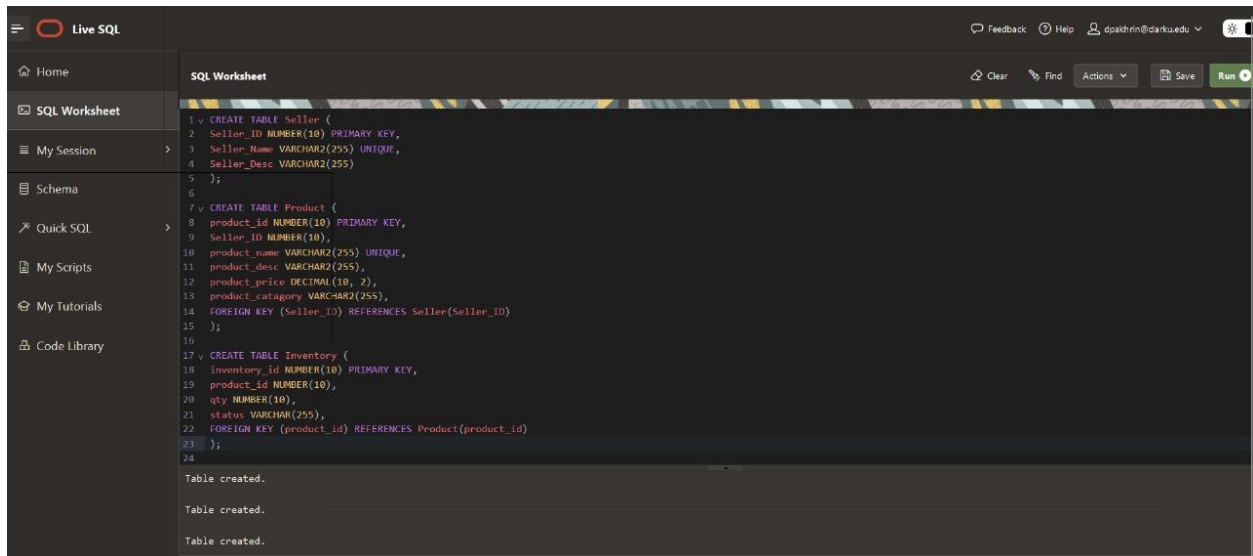
- Creating the tables

```
CREATE TABLE Seller (  
  Seller_ID NUMBER(10) PRIMARY KEY,  
  Seller_Name VARCHAR2(255) UNIQUE,  
  Seller_Desc VARCHAR2(255)  
);
```

```
CREATE TABLE Product (  
  product_id NUMBER(10) PRIMARY KEY,  
  Seller_ID NUMBER(10),  
  product_name VARCHAR2(255) UNIQUE,  
  product_desc VARCHAR2(255),  
  product_price DECIMAL(10, 2),  
  product_catagory VARCHAR2(255),  
  FOREIGN KEY (Seller_ID) REFERENCES Seller(Seller_ID)  
);
```

```
CREATE TABLE Inventory (  
  inventory_id NUMBER(10) PRIMARY KEY,  
  product_id NUMBER(10),
```

```
qty NUMBER(10),  
status VARCHAR(255),  
FOREIGN KEY (product_id) REFERENCES Product(product_id)  
);
```



The screenshot shows a web-based SQL editor interface titled "Live SQL". On the left is a sidebar with navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet" and contains the following SQL code:

```
1 v CREATE TABLE Seller (  
2   Seller_ID NUMBER(10) PRIMARY KEY,  
3   Seller_Name VARCHAR2(255) UNIQUE,  
4   Seller_Desc VARCHAR2(255)  
5 );  
6  
7 v CREATE TABLE Product (  
8   product_id NUMBER(10) PRIMARY KEY,  
9   Seller_ID NUMBER(10),  
10  product_name VARCHAR2(255) UNIQUE,  
11  product_desc VARCHAR2(255),  
12  product_price DECIMAL(10, 2),  
13  product_catagory VARCHAR2(255),  
14  FOREIGN KEY (Seller_ID) REFERENCES Seller(Seller_ID)  
15 );  
16  
17 v CREATE TABLE Inventory (  
18  inventory_id NUMBER(10) PRIMARY KEY,  
19  product_id NUMBER(10),  
20  qty NUMBER(10),  
21  status VARCHAR(255),  
22  FOREIGN KEY (product_id) REFERENCES Product(product_id)  
23 );  
24
```

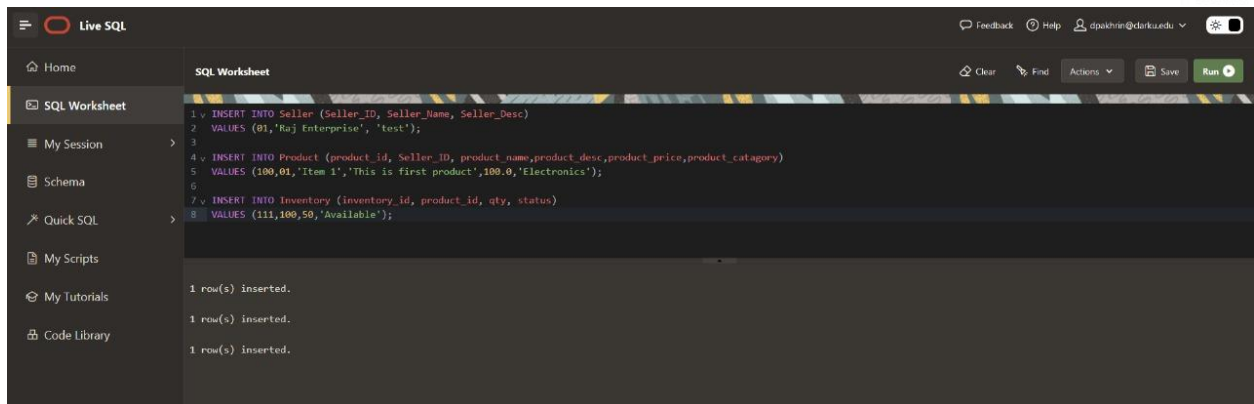
Below the code, the execution results are shown as "Table created." for each of the three tables.

- Insert test Values into the table:

```
INSERT INTO Seller (Seller_ID, Seller_Name, Seller_Desc)  
VALUES (01,'Raj Enterprise', 'test');
```

```
INSERT INTO Product (product_id, Seller_ID,  
product_name,product_desc,product_price,product_catagory)  
VALUES (100,01,'Item 1','This is first product',100.0,'Electronics');
```

```
INSERT INTO Inventory (inventory_id, product_id, qty, status)  
VALUES (111,100,50,'Available');
```



Creating Stored Procedures and introducing values using these proceduers:

- Creating storage procedures for inserting products into the product table and also inserting into the inventory table

CREATE OR REPLACE PROCEDURE add_product_to_seller_and_inventory (

p_id NUMBER,

s_id NUMBER,

p_name VARCHAR,

p_desc VARCHAR,

p_price DECIMAL,

p_catagory VARCHAR,

i_id NUMBER,

i_qty NUMBER,

i_status VARCHAR

)

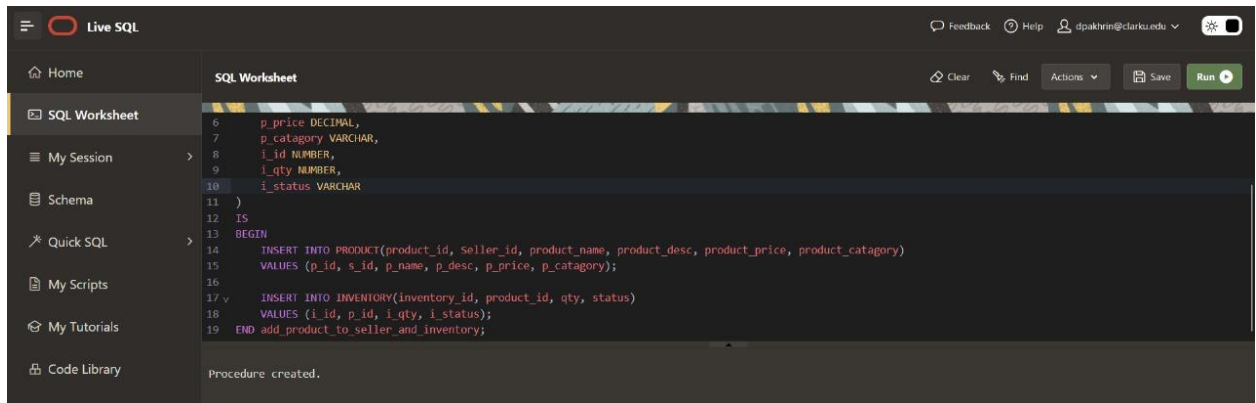
IS

BEGIN

INSERT INTO PRODUCT(product_id, Seller_id, product_name, product_desc,
product_price, product_catagory)

VALUES (p_id, s_id, p_name, p_desc, p_price, p_catagory);

```
INSERT INTO INVENTORY(inventory_id, product_id, qty, status)
VALUES (i_id, p_id, i_qty, i_status);
END add_product_to_seller_and_inventory;
```



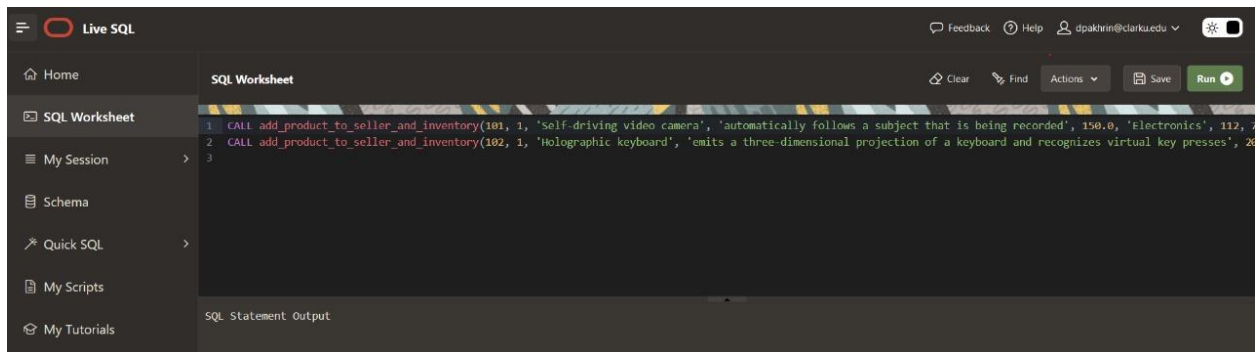
The screenshot shows the 'Live SQL' web application. On the left is a sidebar with navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains the following SQL code:

```
6  p_price DECIMAL,
7  p_category VARCHAR,
8  i_id NUMBER,
9  i_qty NUMBER,
10 i_status VARCHAR
11 )
12 IS
13 BEGIN
14     INSERT INTO PRODUCT(product_id, seller_id, product_name, product_desc, product_price, product_category)
15     VALUES (p_id, s_id, p_name, p_desc, p_price, p_category);
16
17     INSERT INTO INVENTORY(inventory_id, product_id, qty, status)
18     VALUES (i_id, p_id, i_qty, i_status);
19 END add_product_to_seller_and_inventory;
```

Below the code, the status 'Procedure created.' is displayed. At the top right of the interface, there are links for Feedback, Help, and a user profile (dpakhrin@clarku.edu).

Inserting Values using the stored procedure:

- CALL add_product_to_seller_and_inventory(101, 1, 'Self-driving video camera', 'automatically follows a subject that is being recorded', 150.0, 'Electronics', 112, 70, 'Available');
- CALL add_product_to_seller_and_inventory(102, 1, 'Holographic keyboard', 'emits a three-dimensional projection of a keyboard and recognizes virtual key presses', 20.0, 'Computer', 113, 50, 'Available');



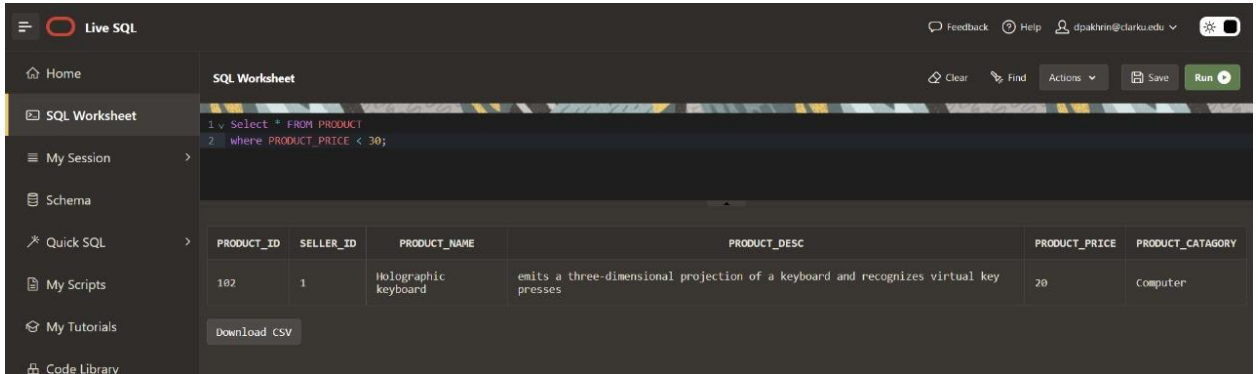
The screenshot shows the 'Live SQL' web application with the same sidebar as the previous image. The 'SQL Worksheet' area now contains two lines of SQL code:

```
1 CALL add_product_to_seller_and_inventory(101, 1, 'Self-driving video camera', 'automatically follows a subject that is being recorded', 150.0, 'Electronics', 112, 70, 'Available');
2 CALL add_product_to_seller_and_inventory(102, 1, 'Holographic keyboard', 'emits a three-dimensional projection of a keyboard and recognizes virtual key presses', 20.0, 'Computer', 113, 50, 'Available');
```

Below the code, the status 'SQL Statement Output' is displayed. The top right of the interface remains the same.

SQL Query to find the item in the list of existing products in the “Computers” or “Electronics” categories that cost \$30 or less.

- Select * FROM PRODUCT
where PRODUCT_PRICE < 30;



The screenshot shows the 'Live SQL' web application interface. On the left is a sidebar with navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains the following SQL query:

```
1. Select * FROM PRODUCT
2. where PRODUCT_PRICE < 30;
```

Below the query editor, the results are displayed in a table with the following columns: PRODUCT_ID, SELLER_ID, PRODUCT_NAME, PRODUCT_DESC, PRODUCT_PRICE, and PRODUCT_CATEGORY. One result is shown:

| PRODUCT_ID | SELLER_ID | PRODUCT_NAME | PRODUCT_DESC | PRODUCT_PRICE | PRODUCT_CATEGORY |
|------------|-----------|----------------------|---|---------------|------------------|
| 102 | 1 | Holographic keyboard | emits a three-dimensional projection of a keyboard and recognizes virtual key presses | 20 | Computer |

A 'Download CSV' button is located below the table.