

Term Project – Final Deliverable

Team Members: Dhiraj Pakhrin, Satkrit Bhattarai

1.

Business Rule

There is no change in our current business rules from the previous one. The updated business rules are mentioned below:

- One seller can sell multiple products, but one product is associated with only one seller.
- One product can have multiple inventory records, but each inventory record corresponds to only one product.
- One consumer can place zero or multiple orders, but one order is associated with only one consumer.
- One order can contain multiple products, and each product can be present in multiple orders.
- One order can have one shipment, and one shipment is associated with one order.
- One Amazon warehouse contains multiple products.
- One shipment originates from one Amazon warehouse, and one Amazon warehouse can have multiple shipments.

2.

Conceptual ERD

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

1. Added "product_id" as an attribute to the order table
2. Added "customer_id" as an attribute to the order table
3. Added "order_qty" as an attribute to the order table
4. Added "order_id" as a foreign key to the shipment table

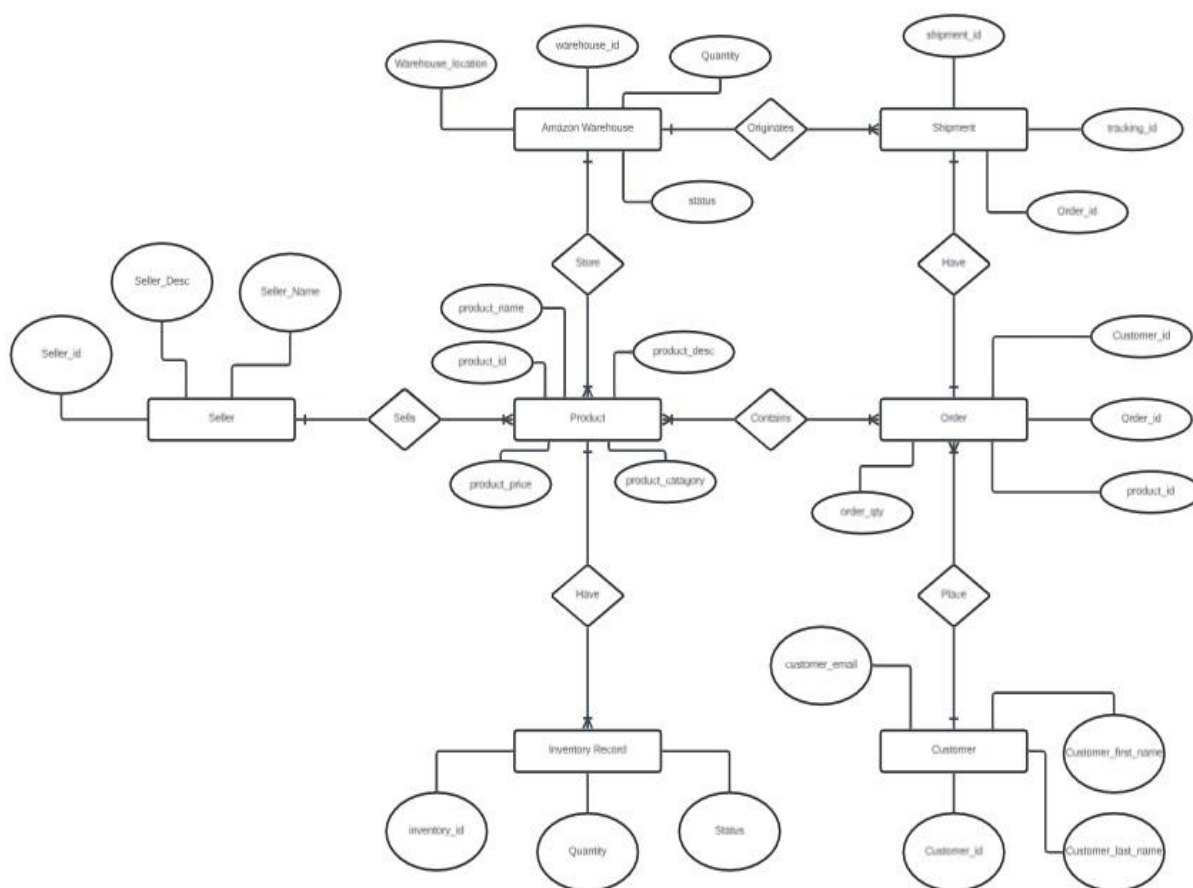


Fig: Conceptual ER-Diagram of our new business rule

3. Logical ERD

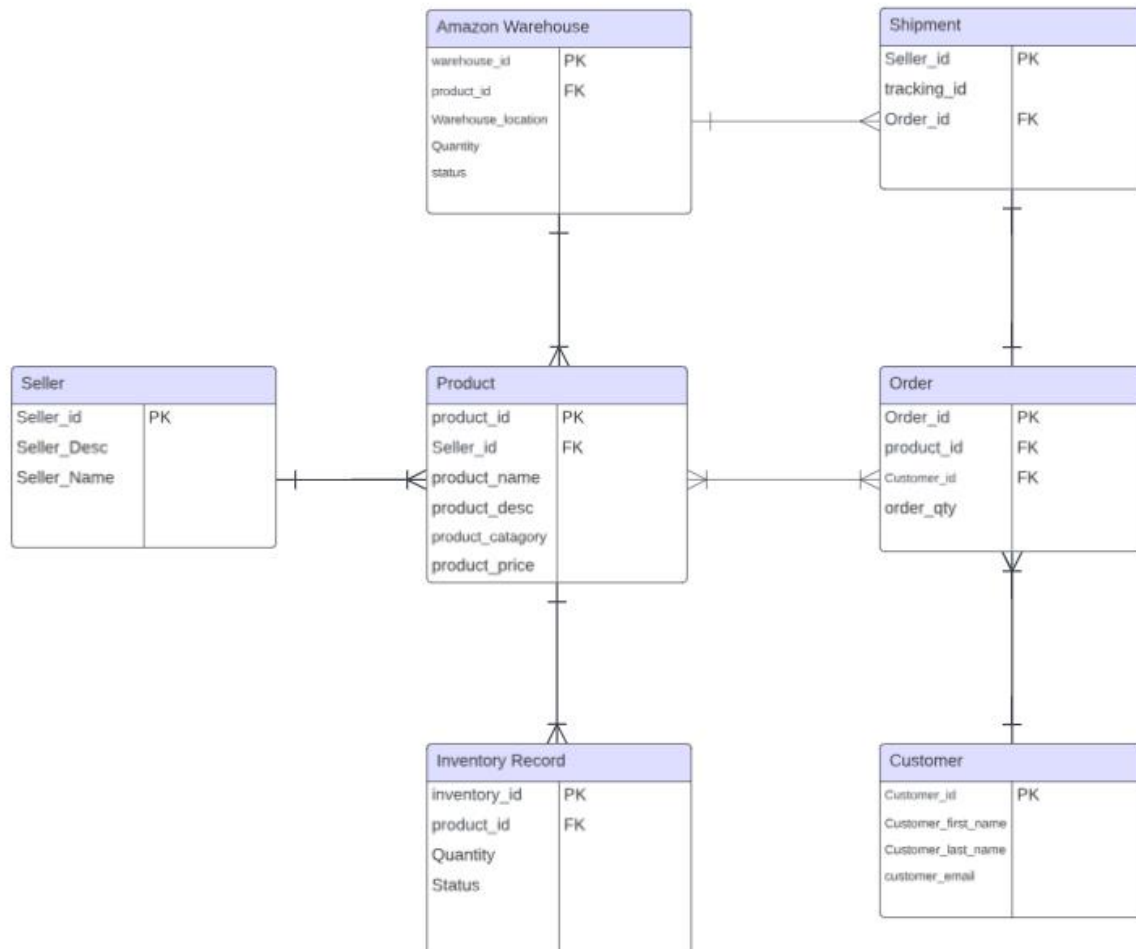


Fig: Conceptual ER-Diagram of New Business Rule

4.

Aspect 4

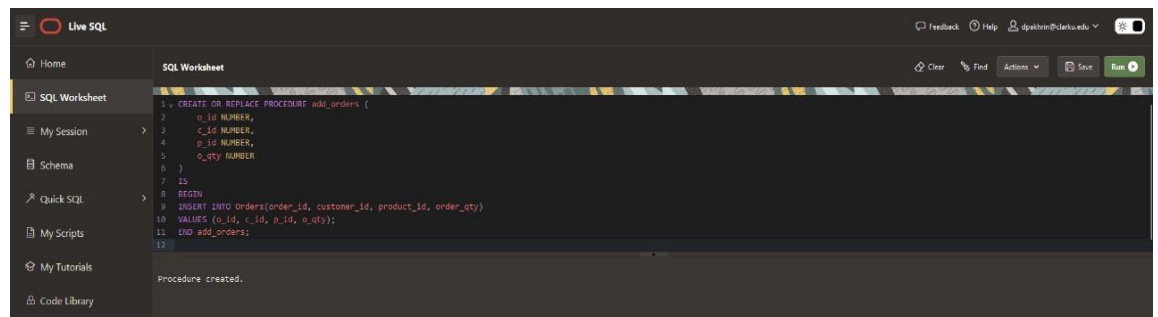
a. Creation of tables and constraints



```
1 Create Table Orders (
2   order_id NUMBER(10) PRIMARY KEY,
3   customer_id NUMBER(10),
4   product_id NUMBER(10),
5   order_qty NUMBER(10),
6   FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
7   FOREIGN KEY (product_id) REFERENCES Product(product_id)
8 );
9
```

Table created.

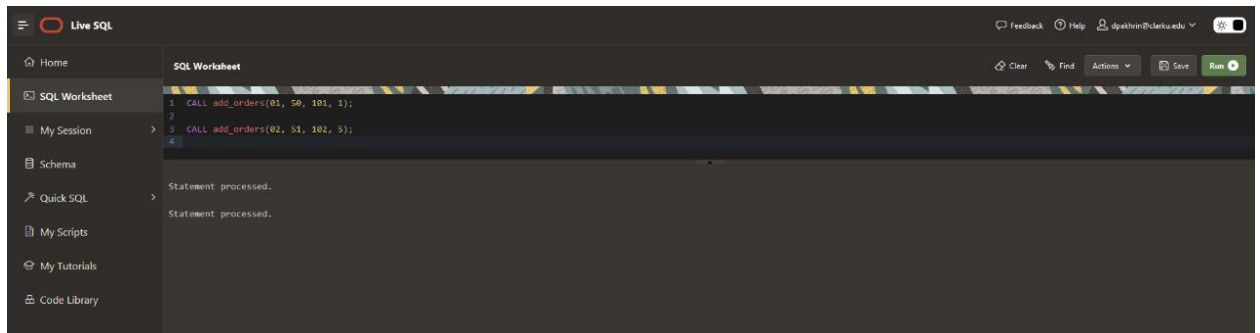
b. Creation of reusable stored procedure



```
1 CREATE OR REPLACE PROCEDURE add_orders (
2   o_id NUMBER,
3   c_id NUMBER,
4   p_id NUMBER,
5   o_qty NUMBER
6 )
7 IS
8 BEGIN
9   INSERT INTO Orders(order_id, customer_id, product_id, order_qty)
10  VALUES (o_id, c_id, p_id, o_qty);
11 END add_orders;
12
```

Procedure created.

c. Use of the stored procedure

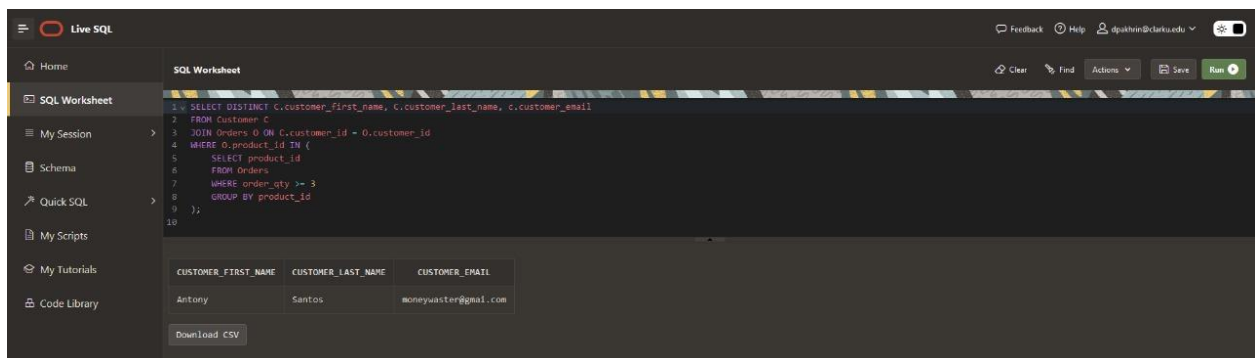


The screenshot shows the Live SQL interface with the following SQL code in the editor:

```
1 CALL add_orders(01, 50, 101, 1);
2
3 CALL add_orders(02, 51, 102, 5);
4
```

Below the code, the output area shows two messages: "Statement processed." and "Statement processed."

d. SQL query



The screenshot shows the Live SQL interface with the following SQL query in the editor:

```
1 SELECT DISTINCT C.customer_first_name, C.customer_last_name, C.customer_email
2 FROM Customer C
3 JOIN Orders O ON C.customer_id = O.customer_id
4 WHERE O.product_id IN (
5   SELECT product_id
6   FROM Orders
7   WHERE order_qty >= 3
8   GROUP BY product_id
9 );
10
```

Below the code, the results are displayed in a table:

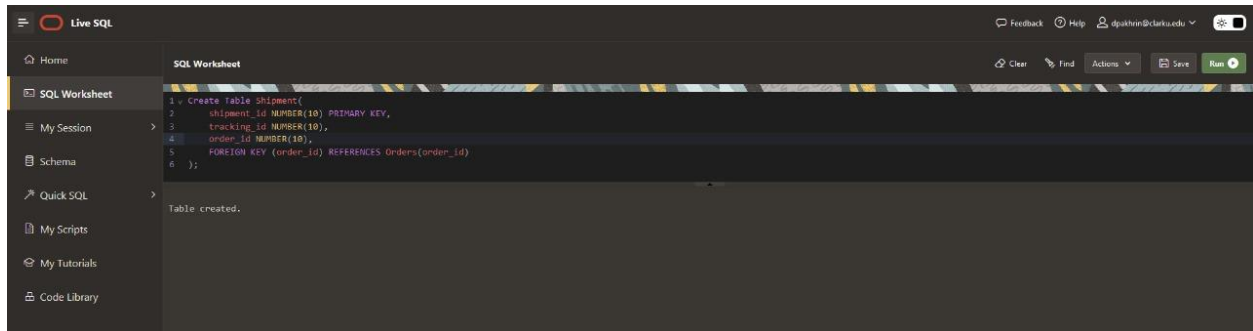
CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	CUSTOMER_EMAIL
Antony	Santos	moneywaster@gmail.com

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

Note: There is no address found for this deliverable because we do not have that attribute for the customer.

Aspect 5

a. Creation of tables and constraints

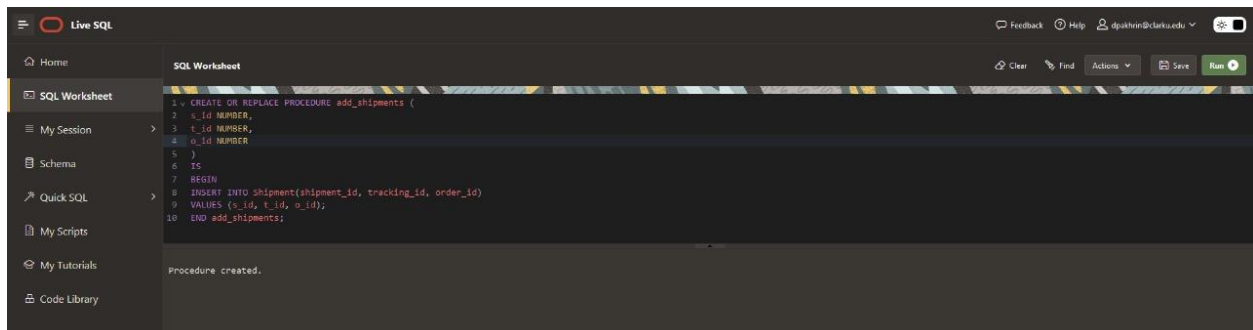


The screenshot shows the Live SQL interface with the following SQL code in the editor:

```
1. Create Table shipment(
2   shipment_id NUMBER(10) PRIMARY KEY,
3   tracking_id NUMBER(10),
4   order_id NUMBER(10),
5   FOREIGN KEY (order_id) REFERENCES Orders(order_id)
6 );
```

The output area below the editor displays the message: "Table created."

b. Creation of reusable stored procedure

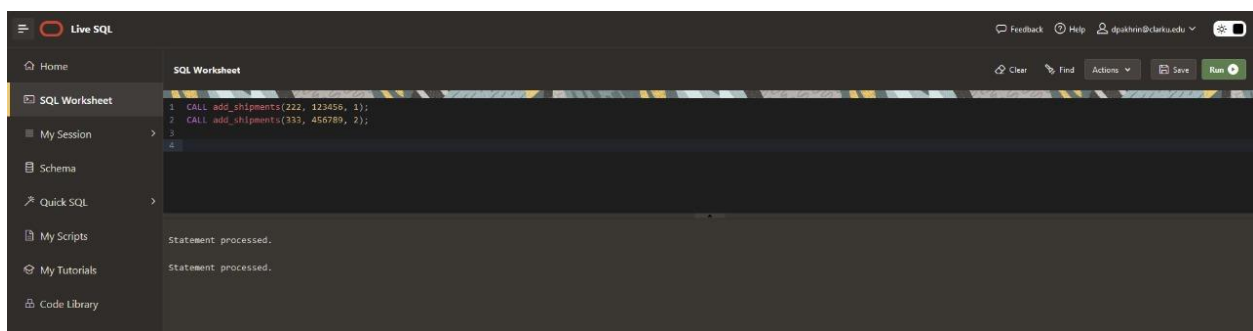


The screenshot shows the Live SQL interface with the following SQL code in the editor:

```
1. CREATE OR REPLACE PROCEDURE add_shipments (
2   s_id NUMBER,
3   t_id NUMBER,
4   o_id NUMBER
5 )
6 IS
7 BEGIN
8   INSERT INTO shipment(shipment_id, tracking_id, order_id)
9   VALUES (s_id, t_id, o_id);
10 END add_shipments;
```

The output area below the editor displays the message: "Procedure created."

c. Use of the stored procedure

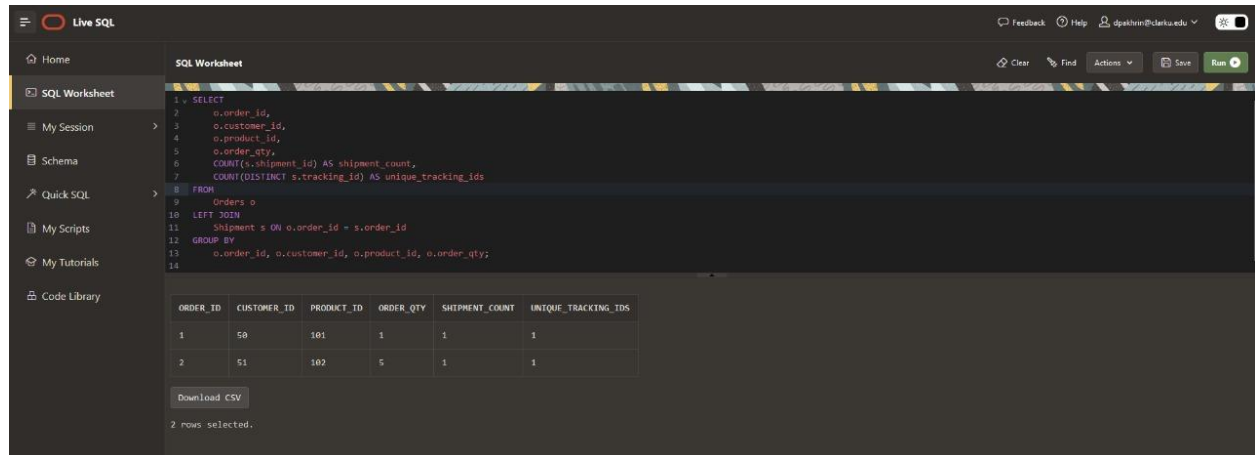


The screenshot shows the Live SQL interface with the following SQL code in the editor:

```
1 CALL add_shipments(222, 123456, 1);
2 CALL add_shipments(333, 456789, 2);
3
4
```

The output area below the editor displays the message: "Statement processed."

d. SQL query



The screenshot shows a web-based SQL editor interface titled "Live SQL". 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 an SQL query. The query is as follows:

```
1. SELECT
2   o.order_id,
3   o.customer_id,
4   o.product_id,
5   o.order_qty,
6   COUNT(s.shipment_id) AS shipment_count,
7   COUNT(DISTINCT s.tracking_id) AS unique_tracking_ids
8. FROM
9   Orders o
10  LEFT JOIN
11    Shipment s ON o.order_id = s.order_id
12  GROUP BY
13    o.order_id, o.customer_id, o.product_id, o.order_qty;
```

Below the query, the results are displayed in a table with 6 columns: ORDER_ID, CUSTOMER_ID, PRODUCT_ID, ORDER_QTY, SHIPMENT_COUNT, and UNIQUE_TRACKING_IDS. There are 2 rows of data.

ORDER_ID	CUSTOMER_ID	PRODUCT_ID	ORDER_QTY	SHIPMENT_COUNT	UNIQUE_TRACKING_IDS
1	50	101	1	1	1
2	51	102	5	1	1

Below the table, there is a "Download CSV" button and a status message: "2 rows selected."

Aspect 4	4
Creation of reusable stored procedure	4
Creation of tables and constraints	4
SQL query	4
Use of the stored procedure	4
Aspect 5	6
Creation of reusable stored procedure	6

Creation of tables and constraints	6
SQL query	6
Use of the stored procedure	6
Business Rule	1
Conceptual ERD	2
Logical ERD	3