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

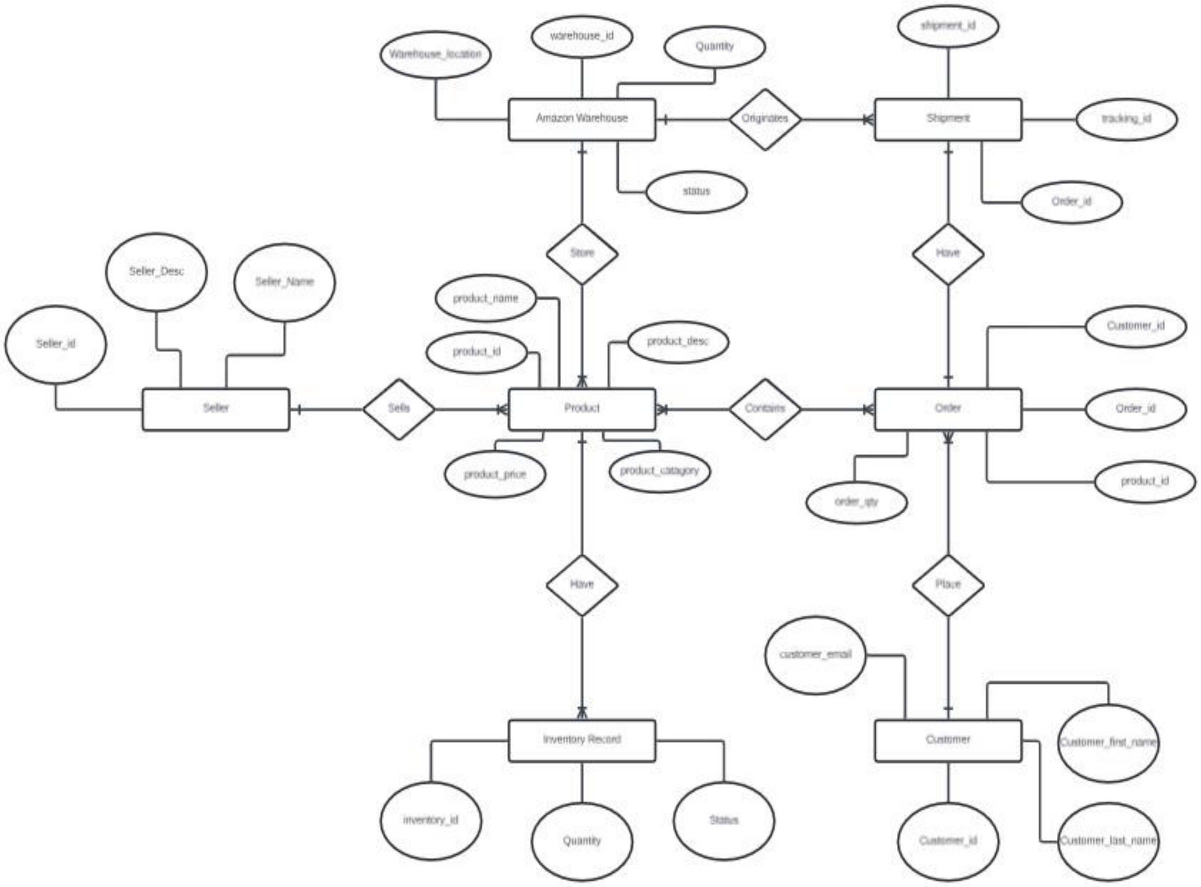


Fig: Conceptual ER-Diagram of our new business rule

3.

Logical ERD

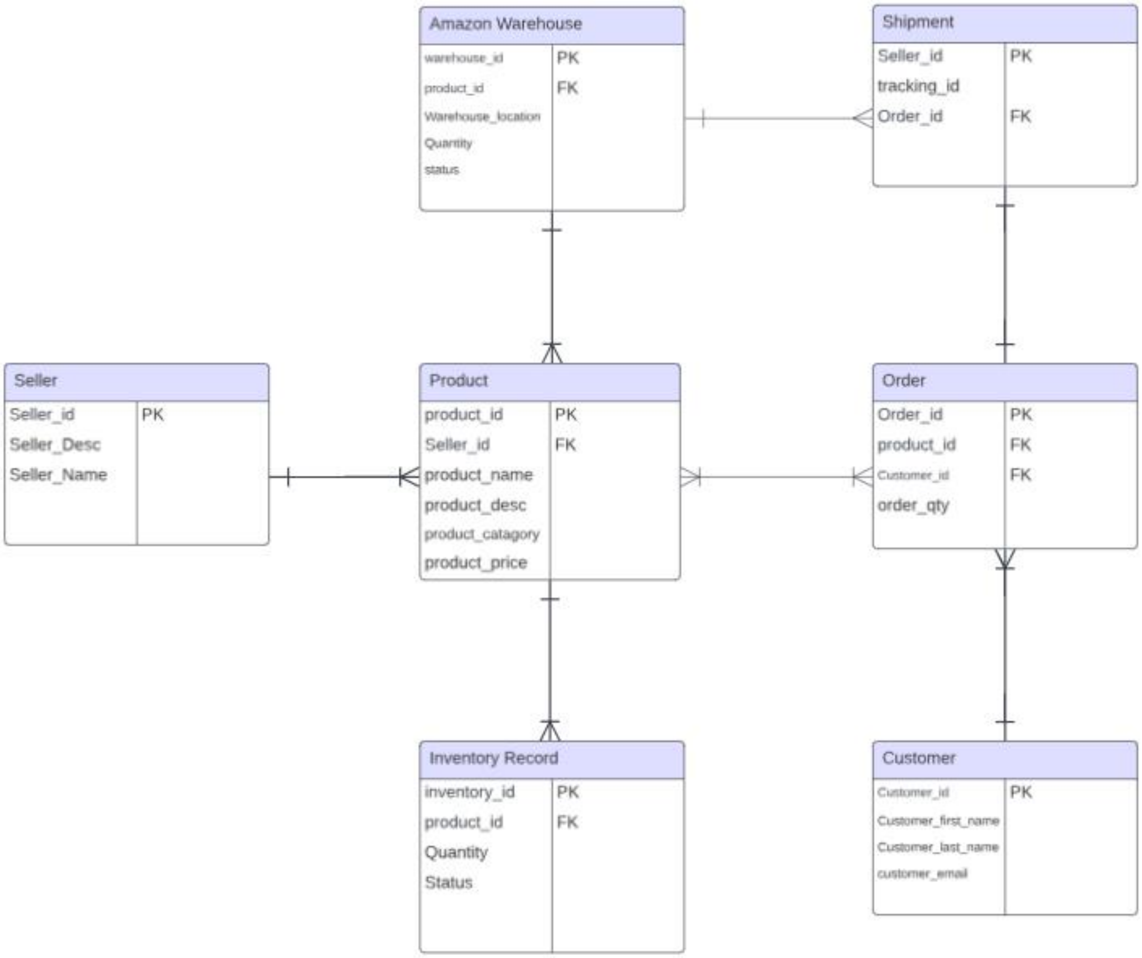
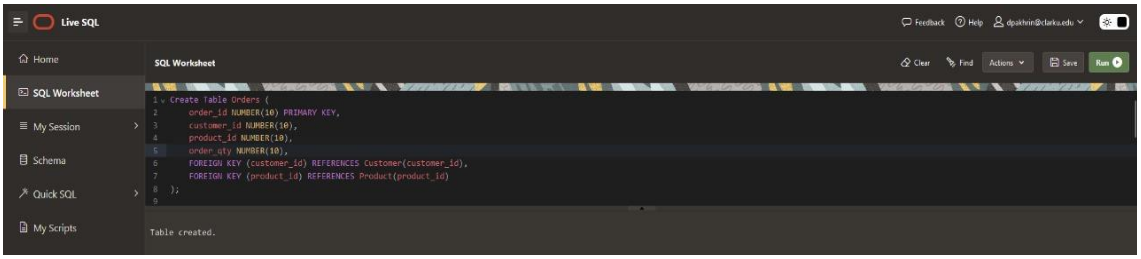


Fig: Conceptual ER-Diagram of New Business Rule

4.

Aspect 4

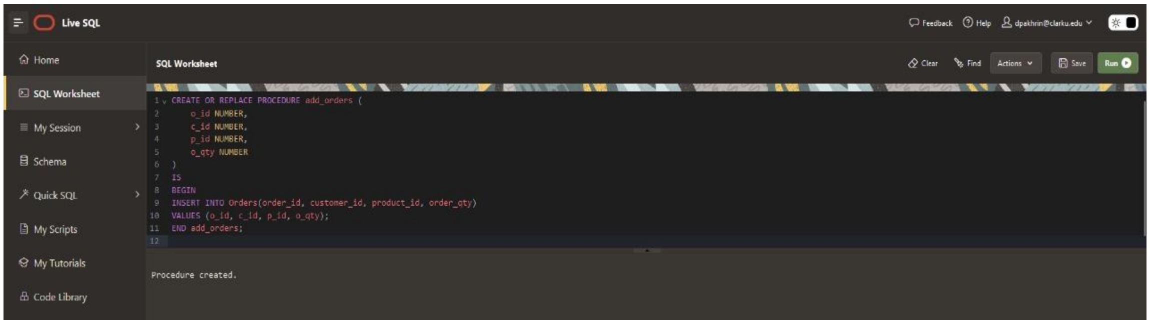
a. Creation of tables and constraints



```
Create Table Orders (
    order_id NUMBER(10) PRIMARY KEY,
    customer_id NUMBER(10),
    product_id NUMBER(10),
    order_qty NUMBER(10),
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
    FOREIGN KEY (product_id) REFERENCES Product(product_id)
);
```

```
Table created.
```

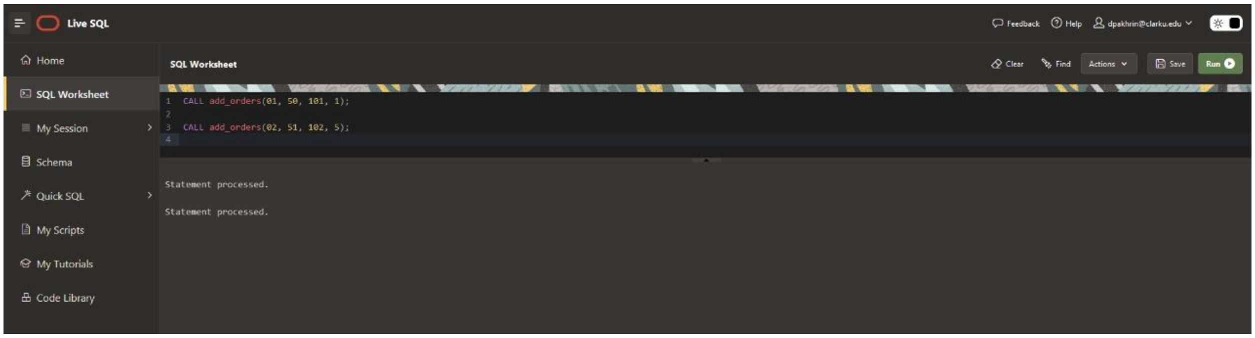
b. Creation of reusable stored procedure



```
CREATE OR REPLACE PROCEDURE add_orders (
    o_id NUMBER,
    c_id NUMBER,
    p_id NUMBER,
    o_qty NUMBER
)
IS
BEGIN
INSERT INTO Orders(order_id, customer_id, product_id, order_qty)
VALUES (o_id, c_id, p_id, o_qty);
END add_orders;
```

```
Procedure created.
```

c. Use of the stored procedure



SQL Worksheet

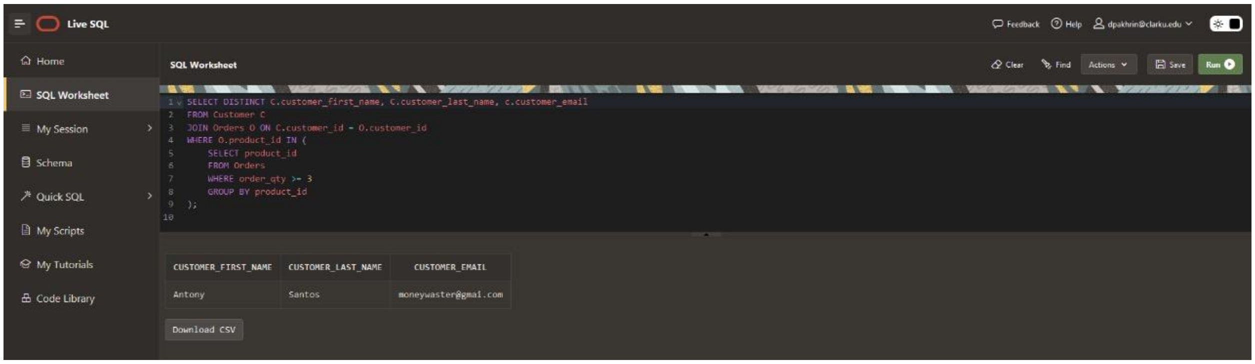
```
CALL add_orders(01, 50, 101, 1);

CALL add_orders(02, 51, 102, 5);
```

```
Statement processed.

Statement processed.
```

d. SQL query



SQL Worksheet

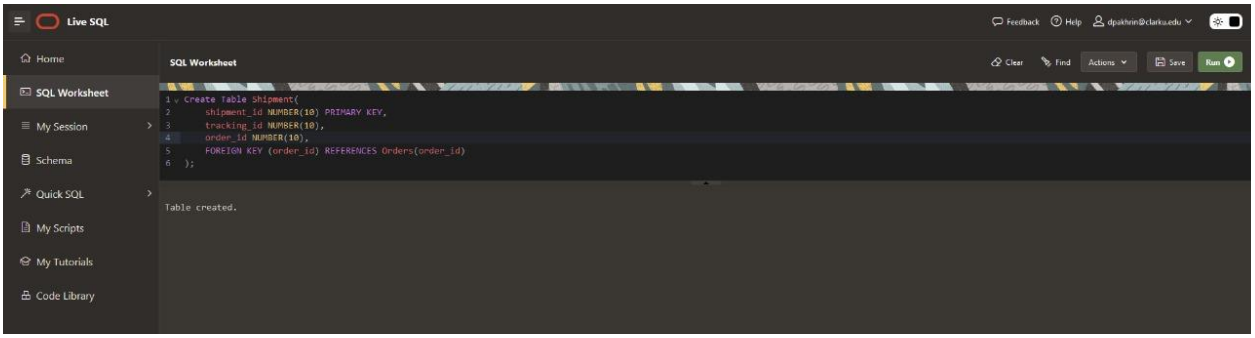
```
SELECT DISTINCT C.customer_first_name, C.customer_last_name, c.customer_email
FROM Customer C
JOIN Orders O ON C.customer_id = O.customer_id
WHERE O.product_id IN (
    SELECT product_id
    FROM Orders
    WHERE order_qty >= 3
    GROUP BY product_id
);
```

CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	CUSTOMER_EMAIL
Antony	Santos	moneywaster@gmai.com

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

Aspect 5

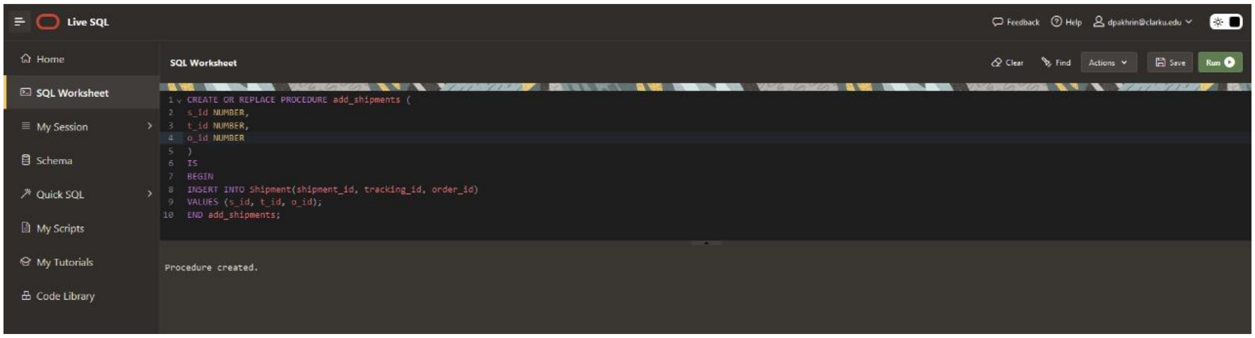
a. Creation of tables and constraints



```
Create Table Shipment(
    shipment_id NUMBER(10) PRIMARY KEY,
    tracking_id NUMBER(10),
    order_id NUMBER(10),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)
);
```

```
Table created.
```

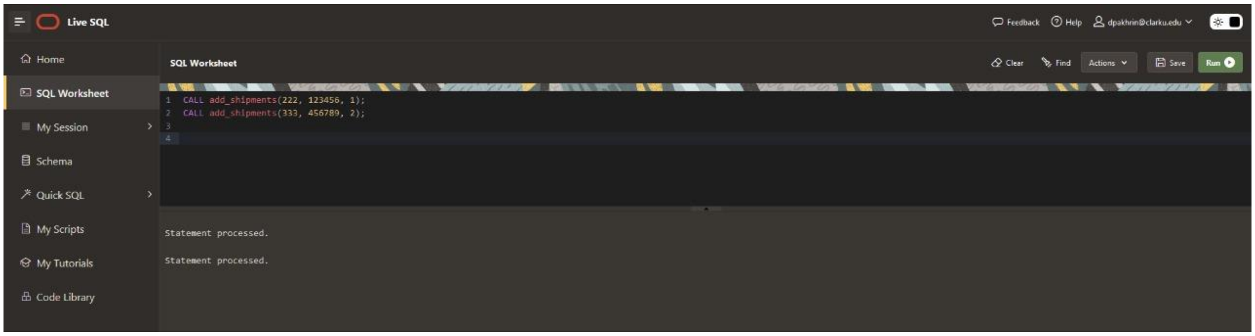
b. Creation of reusable stored procedure



```
CREATE OR REPLACE PROCEDURE add_shipments (
s_id NUMBER,
t_id NUMBER,
o_id NUMBER
)
IS
BEGIN
INSERT INTO Shipment(shipment_id, tracking_id, order_id)
VALUES (s_id, t_id, o_id);
END add_shipments;
```

```
Procedure created.
```

c. Use of the stored procedure

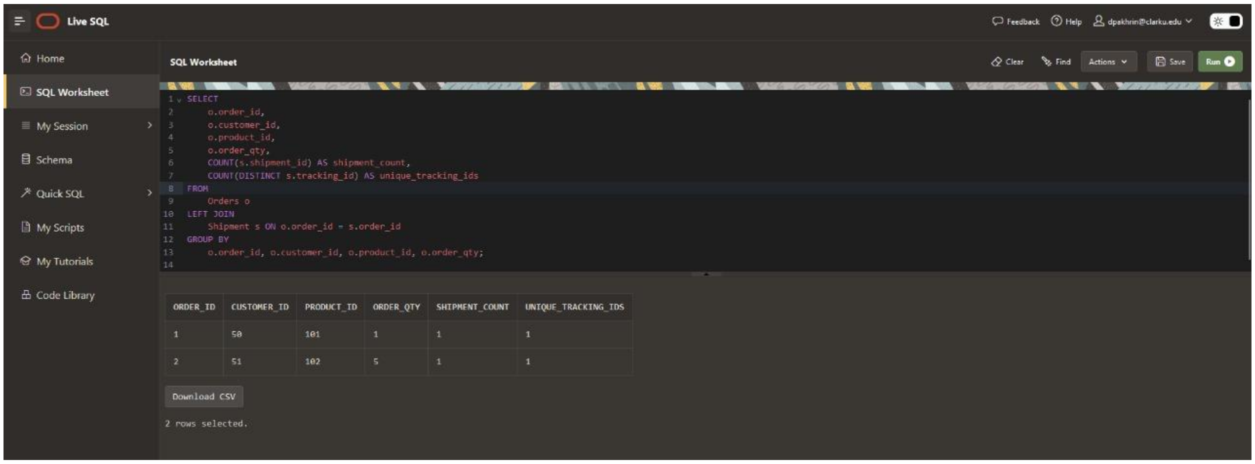


```
CALL add_shipments(222, 123456, 1);
CALL add_shipments(333, 456789, 2);
```

```
Statement processed.

Statement processed.
```

d. SQL query



```sql
SELECT
    o.order_id,
    o.customer_id,
    o.product_id,
    o.order_qty,
    COUNT(s.shipment_id) AS shipment_count,
    COUNT(DISTINCT s.tracking_id) AS unique_tracking_ids
FROM
    Orders o
LEFT JOIN
    Shipment s ON o.order_id = s.order_id
GROUP BY
    o.order_id, o.customer_id, o.product_id, o.order_qty;
```

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

```
2 rows selected.
```