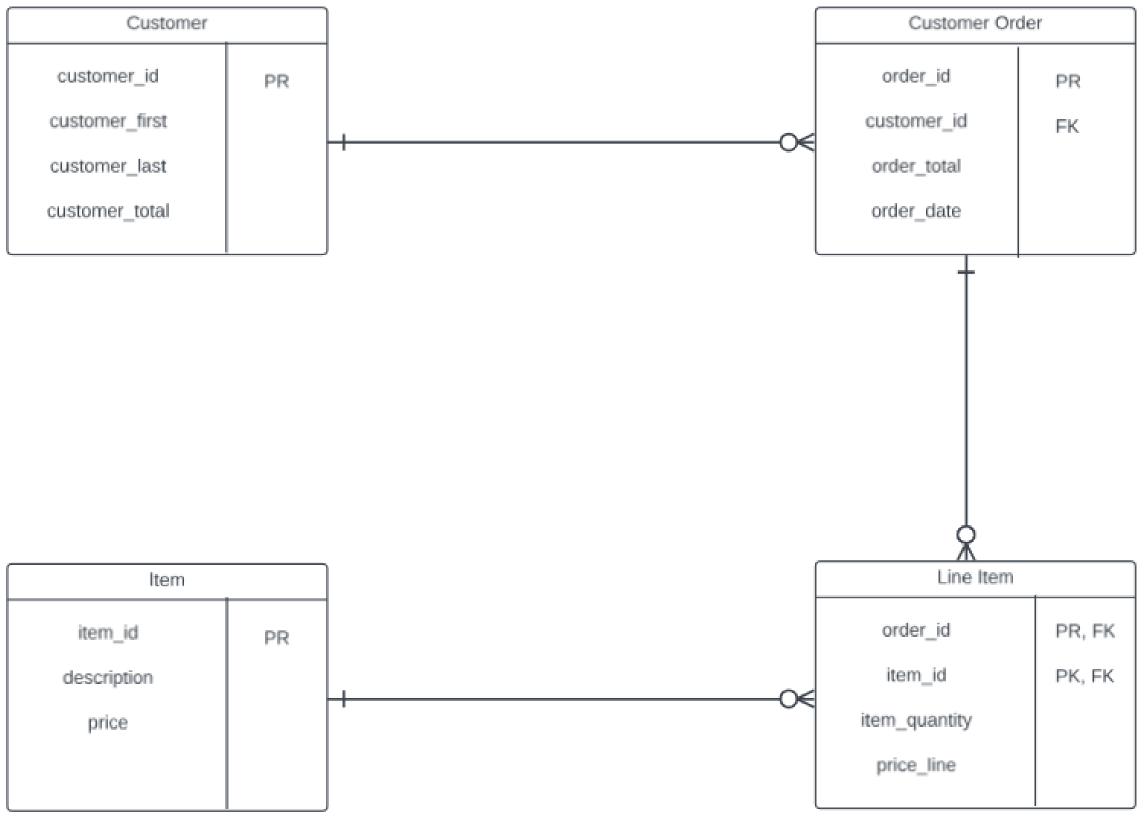
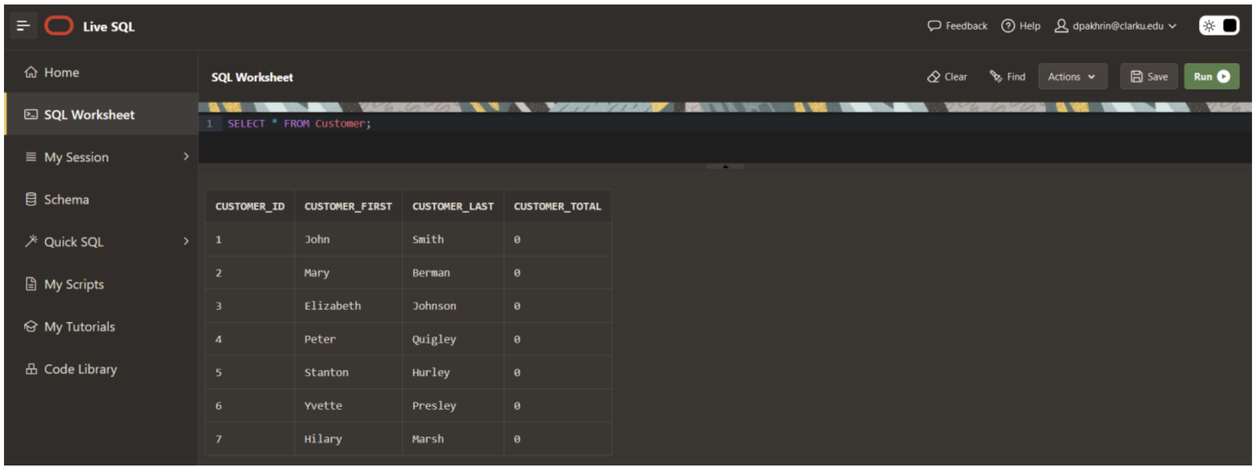
Lab 4 – Submission Sheet

Section One

1. ERD



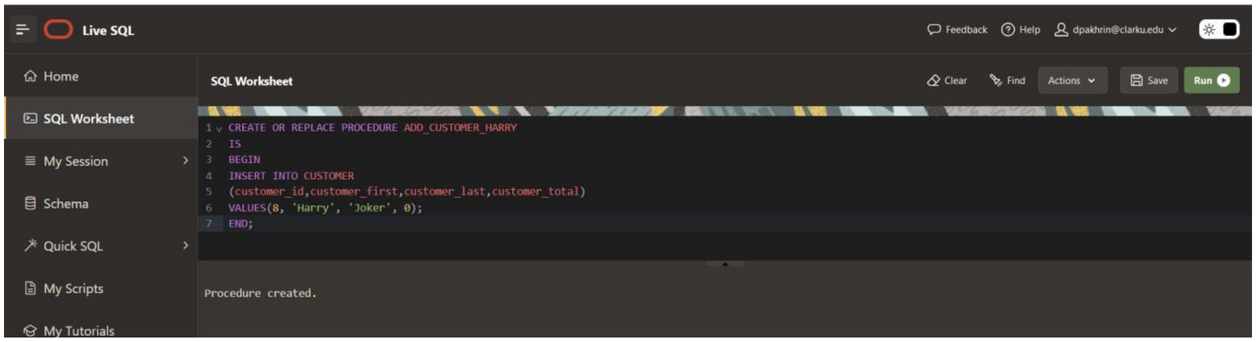
2. SELECT (Screen Shots)



```
SELECT * FROM Customer;
```

CUSTOMER_ID	CUSTOMER_FIRST	CUSTOMER_LAST	CUSTOMER_TOTAL
1	John	Smith	0
2	Mary	Berman	0
3	Elizabeth	Johnson	0
4	Peter	Quigley	0
5	Stanton	Hurley	0
6	Yvette	Presley	0
7	Hilary	Marsh	0

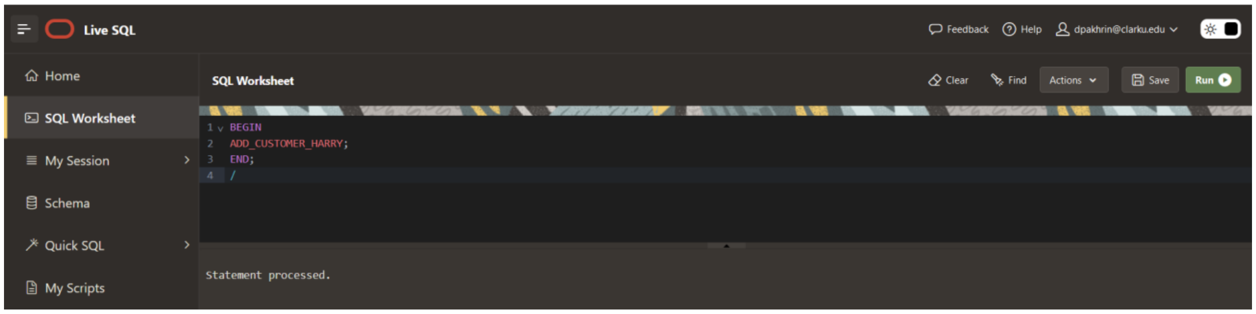
3. CREATE PROCEDURE (Screen Shot)



```
CREATE OR REPLACE PROCEDURE ADD_CUSTOMER_HARRY
IS
BEGIN
INSERT INTO CUSTOMER
(customer_id,customer_first,customer_last,customer_total)
VALUES(8, 'Harry', 'Joker', 0);
END;
```

```
Procedure created.
```

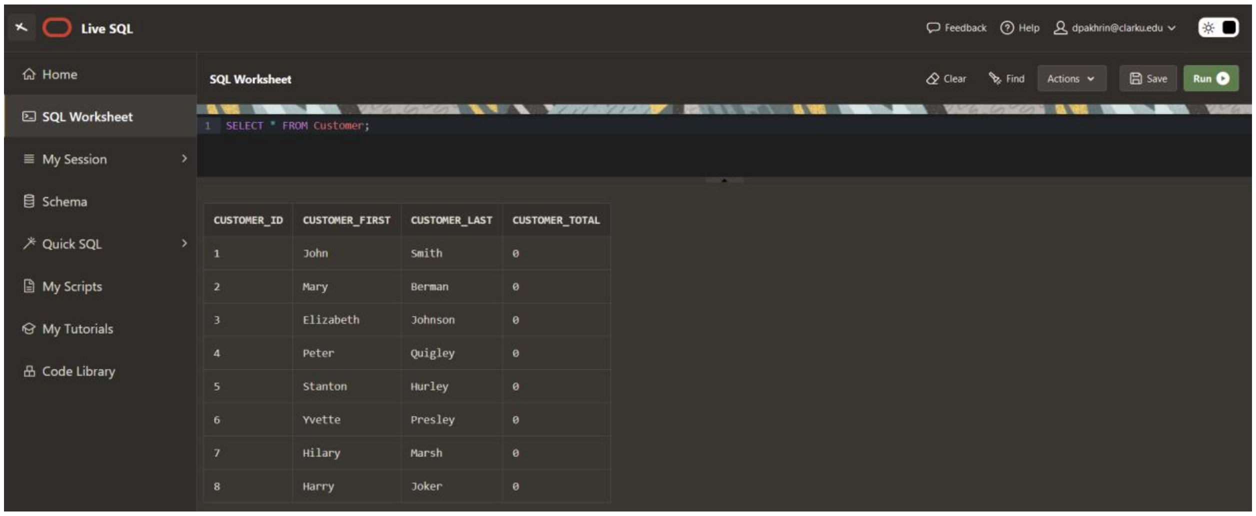
4. EXECUTE PROCEDURE (Screen Shot)



```
BEGIN
ADD_CUSTOMER_HARRY;
END;
/
```

```
Statement processed.
```

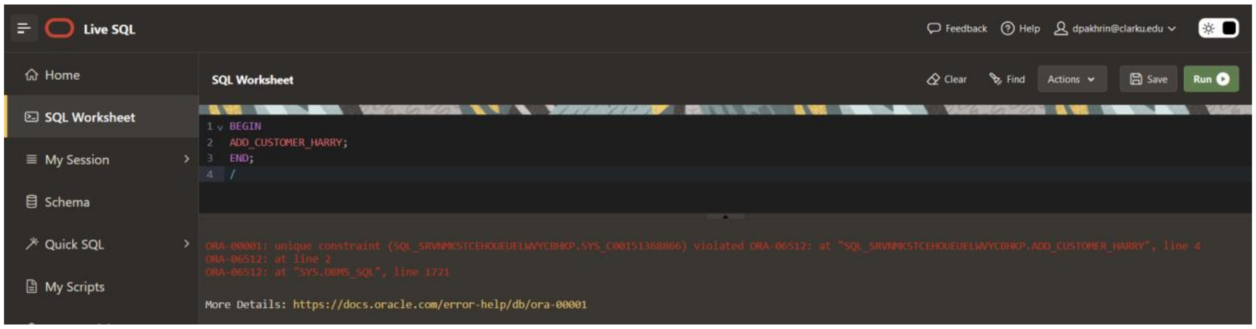
5. SELECT (Screen Shot)



```
SELECT * FROM Customer;
```

CUSTOMER_ID	CUSTOMER_FIRST	CUSTOMER_LAST	CUSTOMER_TOTAL
1	John	Smith	0
2	Mary	Berman	0
3	Elizabeth	Johnson	0
4	Peter	Quigley	0
5	Stanton	Hurley	0
6	Yvette	Presley	0
7	Hilary	Marsh	0
8	Harry	Joker	0

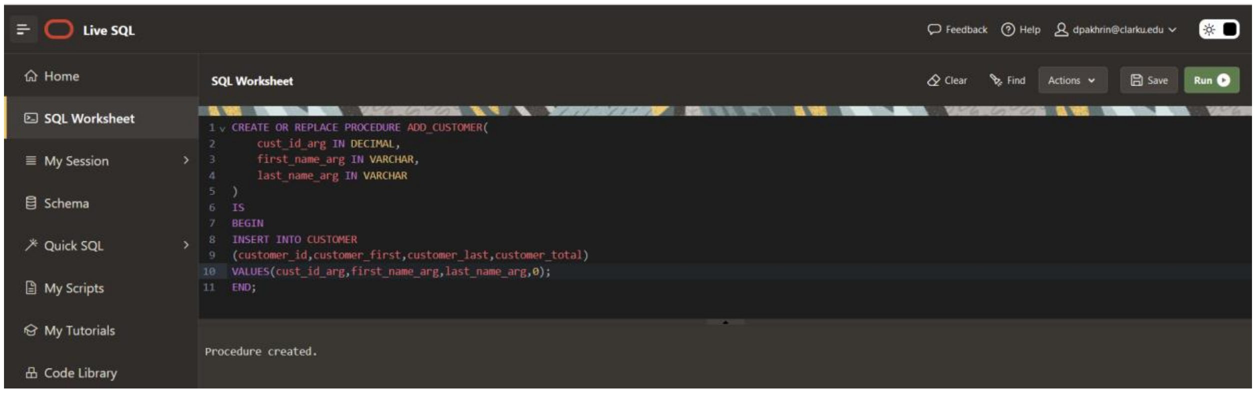
6. EXECUTE PROCEDURE (Screen Shot)



```
BEGIN
ADD_CUSTOMER_HARRY;
END;
/
```

```
ORA-00001: unique constraint (SQL_SRVNMKSTCEHOUEUELWVYCBHKP.SYS_C00151368866) violated ORA-06512: at "SQL_SRVNMKSTCEHOUEUELWVYCBHKP.ADD_CUSTOMER_HARRY", line 4
ORA-06512: at line 2
ORA-06512: at "SYS.DBMS_SQL", line 1721

More Details: https://docs.oracle.com/error-help/db/ora-00001
```

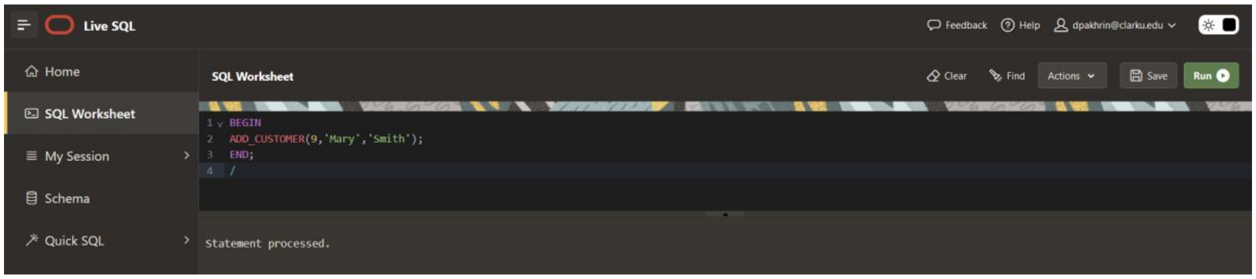
7. CREATE PROCEDURE (Screen Shot)



```
CREATE OR REPLACE PROCEDURE ADD_CUSTOMER(
    cust_id_arg IN DECIMAL,
    first_name_arg IN VARCHAR,
    last_name_arg IN VARCHAR
)
IS
BEGIN
INSERT INTO CUSTOMER
(customer_id,customer_first,customer_last,customer_total)
VALUES(cust_id_arg,first_name_arg,last_name_arg,0);
END;
```

```
Procedure created.
```

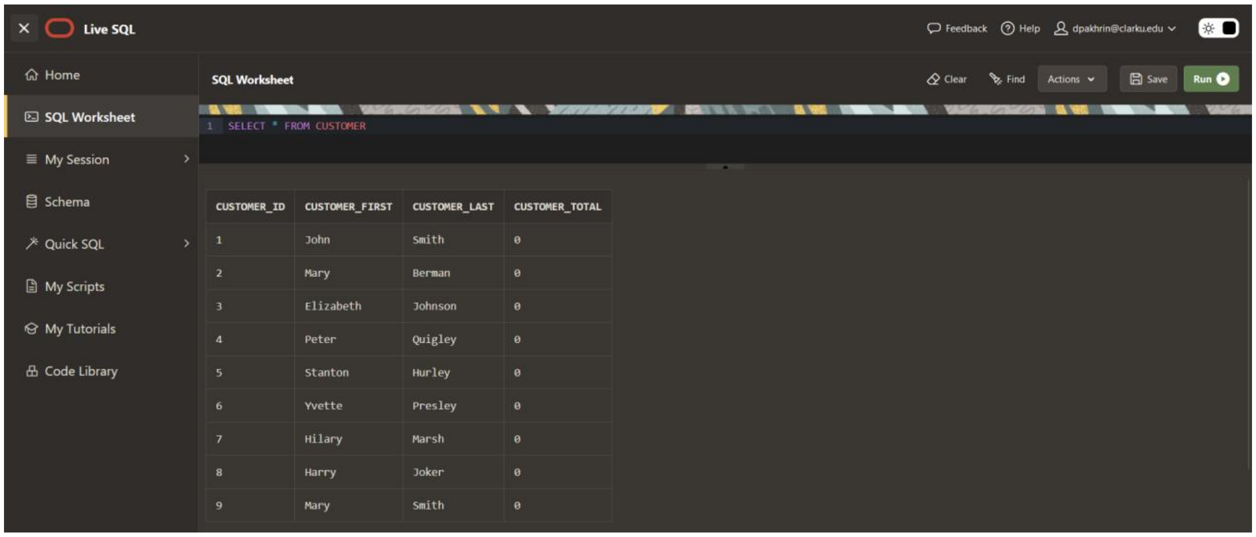
8. EXECUTE PROCEDURE (Screen Shot)



```
BEGIN
ADD_CUSTOMER(9,'Mary','Smith');
END;
/
```

```
Statement processed.
```

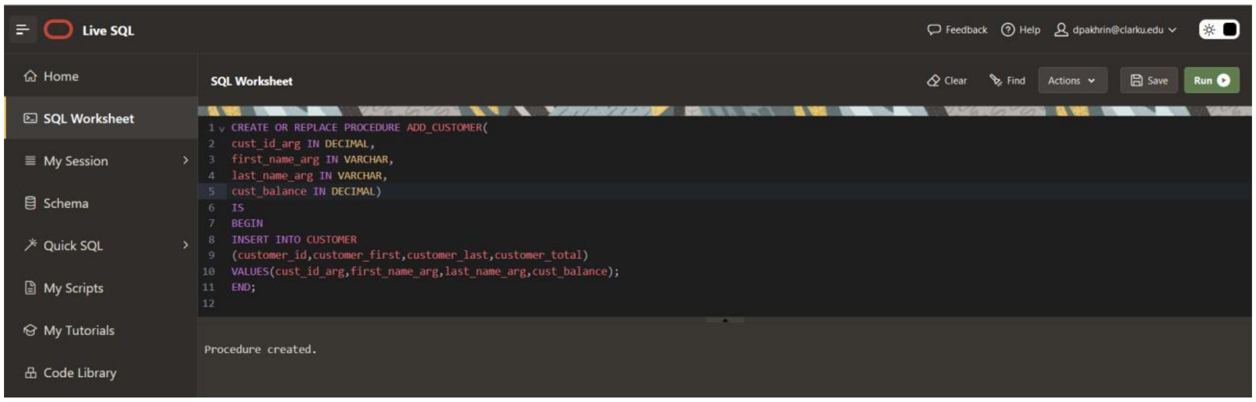
9. SELECT (Screen Shot)



```
SELECT * FROM CUSTOMER
```

CUSTOMER_ID	CUSTOMER_FIRST	CUSTOMER_LAST	CUSTOMER_TOTAL
1	John	Smith	0
2	Mary	Berman	0
3	Elizabeth	Johnson	0
4	Peter	Quigley	0
5	Stanton	Hurley	0
6	Yvette	Presley	0
7	Hilary	Marsh	0
8	Harry	Joker	0
9	Mary	Smith	0

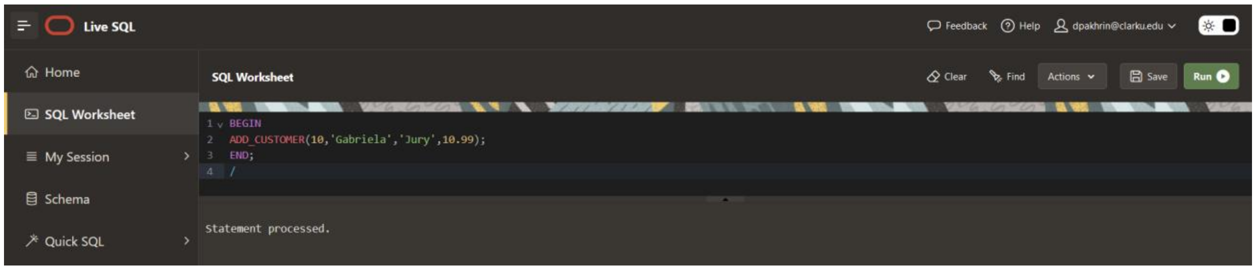
10. CREATE PROCEDURE (Screen Shot)



```
CREATE OR REPLACE PROCEDURE ADD_CUSTOMER(
cust_id_arg IN DECIMAL,
first_name_arg IN VARCHAR,
last_name_arg IN VARCHAR,
cust_balance IN DECIMAL)
IS
BEGIN
INSERT INTO CUSTOMER
(customer_id,customer_first,customer_last,customer_total)
VALUES(cust_id_arg,first_name_arg,last_name_arg,cust_balance);
END;

```

```
Procedure created.
```

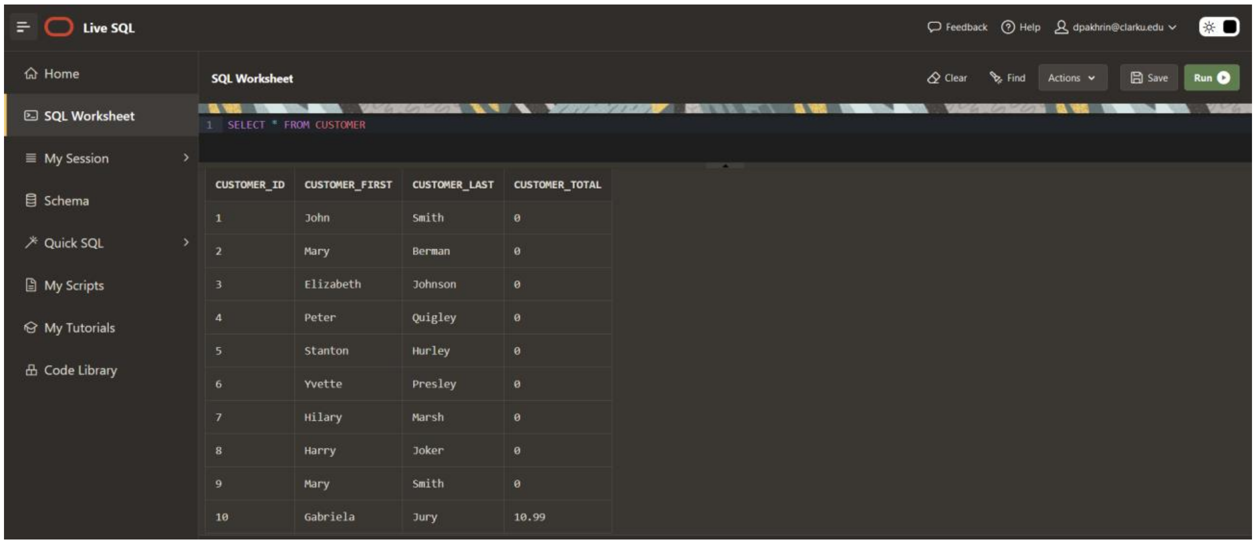
11. EXECUTE PROCEDURE (Screen Shot)



```
BEGIN
ADD_CUSTOMER(10,'Gabriela','Jury',10.99);
END;
/
```

```
Statement processed.
```

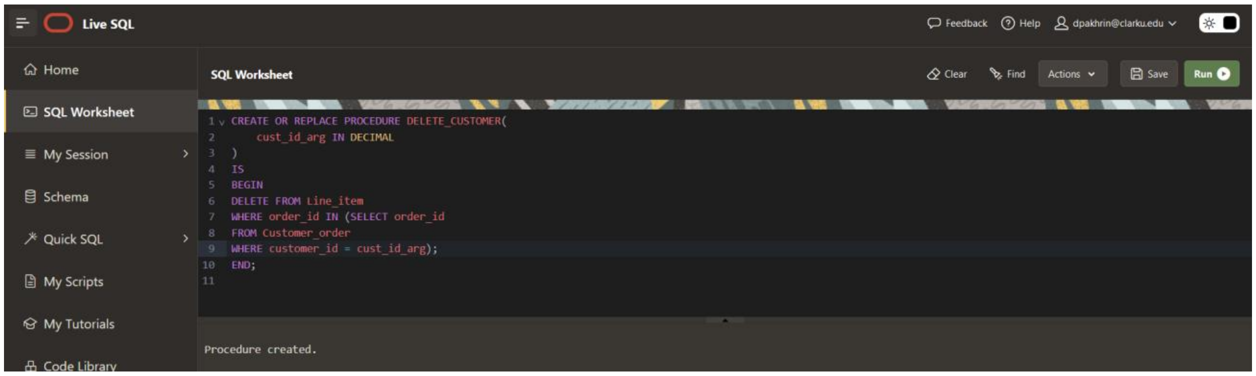
12. SELECT (Screen Shot)



```
SELECT * FROM CUSTOMER
```

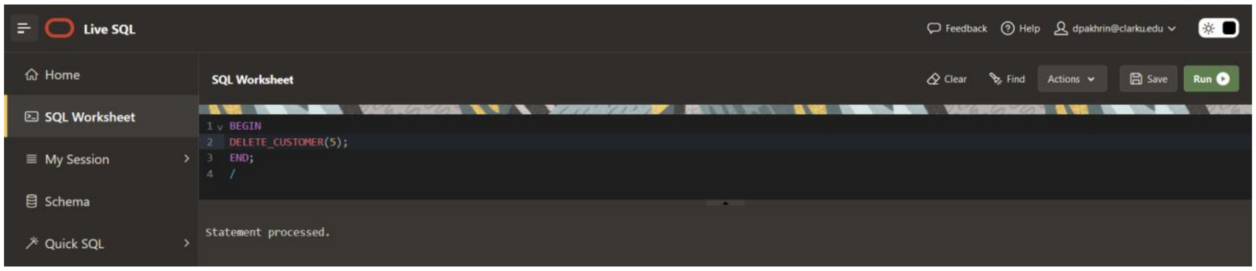
CUSTOMER_ID	CUSTOMER_FIRST	CUSTOMER_LAST	CUSTOMER_TOTAL
1	John	Smith	0
2	Mary	Berman	0
3	Elizabeth	Johnson	0
4	Peter	Quigley	0
5	Stanton	Hurley	0
6	Yvette	Presley	0
7	Hilary	Marsh	0
8	Harry	Joker	0
9	Mary	Smith	0
10	Gabriela	Jury	10.99

13. CREATE & EXECUTE PROCEDURE, VERIFY RESULTS (Screen Shots)



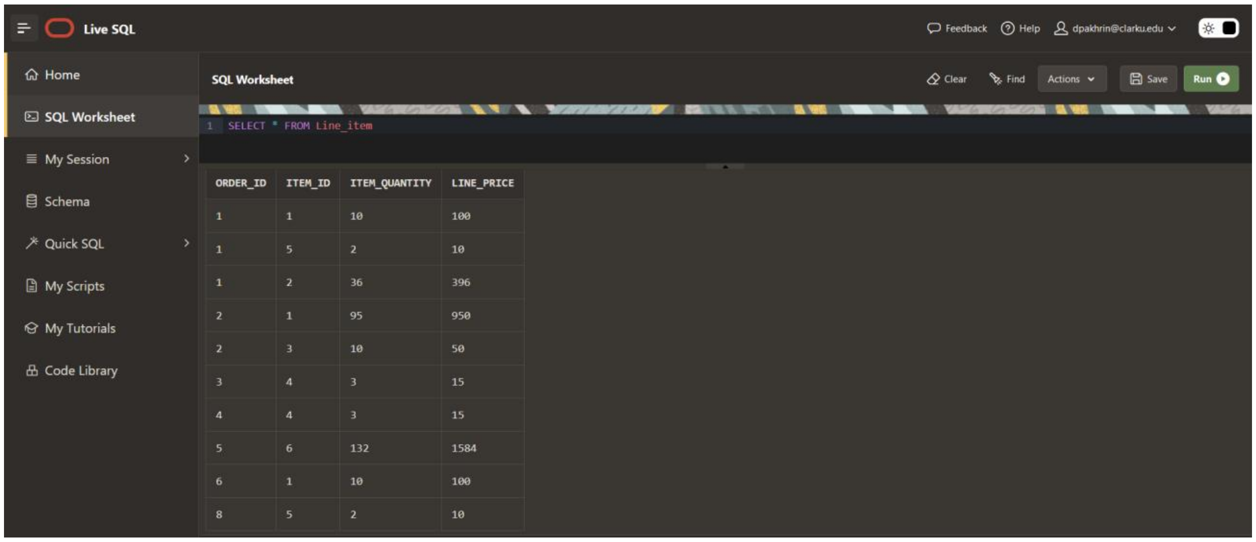
```
CREATE OR REPLACE PROCEDURE DELETE_CUSTOMER(
    cust_id_arg IN DECIMAL
)
IS
BEGIN
DELETE FROM Line_item
WHERE order_id IN (SELECT order_id
FROM Customer_order
WHERE customer_id = cust_id_arg);
END;
```

```
Procedure created.
```



```
BEGIN
DELETE_CUSTOMER(5);
END;
/
```

```
Statement processed.
```

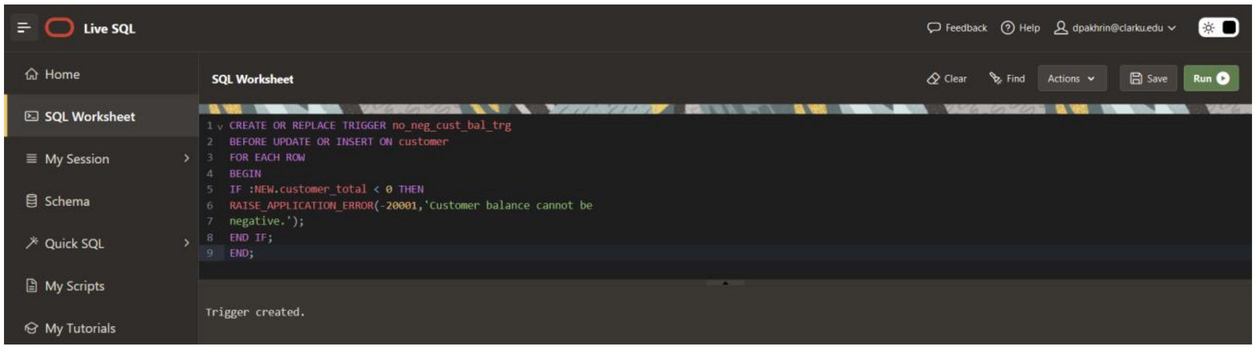


```
SELECT * FROM Line_item
```

ORDER_ID	ITEM_ID	ITEM_QUANTITY	LINE_PRICE
1	1	10	100
1	5	2	10
1	2	36	396
2	1	95	950
2	3	10	50
3	4	3	15
4	4	3	15
5	6	132	1584
6	1	10	100
8	5	2	10

Section Two

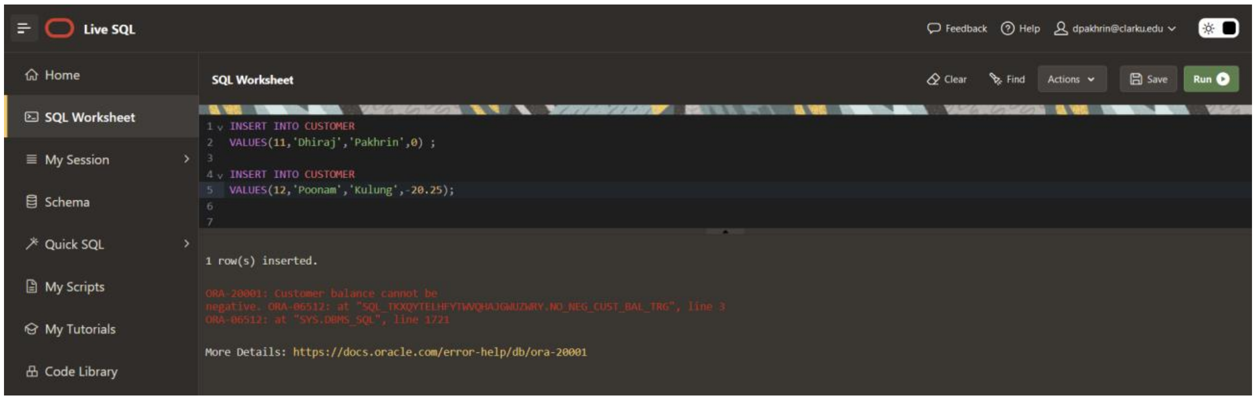
14. CREATE TRIGGER (Screen Shot)



```
CREATE OR REPLACE TRIGGER no_neg_cust_bal_trg
BEFORE UPDATE OR INSERT ON customer
FOR EACH ROW
BEGIN
IF :NEW.customer_total < 0 THEN
RAISE_APPLICATION_ERROR(-20001,'Customer balance cannot be
negative.');
END IF;
END;
```

```
Trigger created.
```

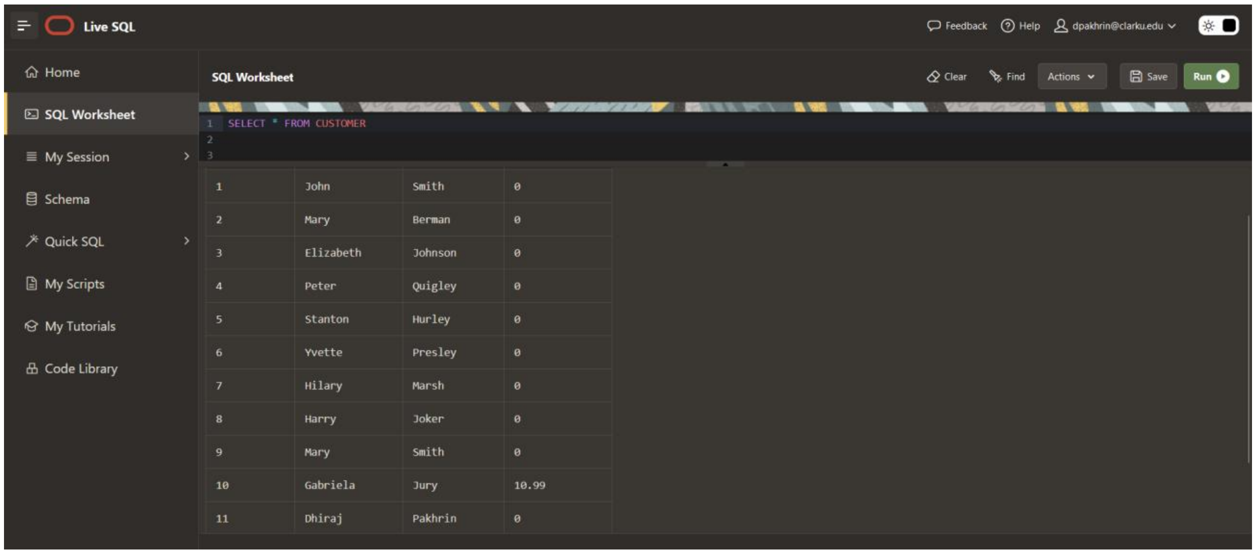
15. INSERT (Screen Shot)



```
INSERT INTO CUSTOMER
VALUES(11,'Dhiraj','Pakhrin',0) ;

INSERT INTO CUSTOMER
VALUES(12,'Poonam','Kulung',-20.25);
```

```
1 row(s) inserted.

ORA-20001: Customer balance cannot be
negative. ORA-06512: at "SQL_TKXQYTELHFYTWVQHAJGWUZWRY.NO_NEG_CUST_BAL_TRG", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721

More Details: https://docs.oracle.com/error-help/db/ora-20001
```

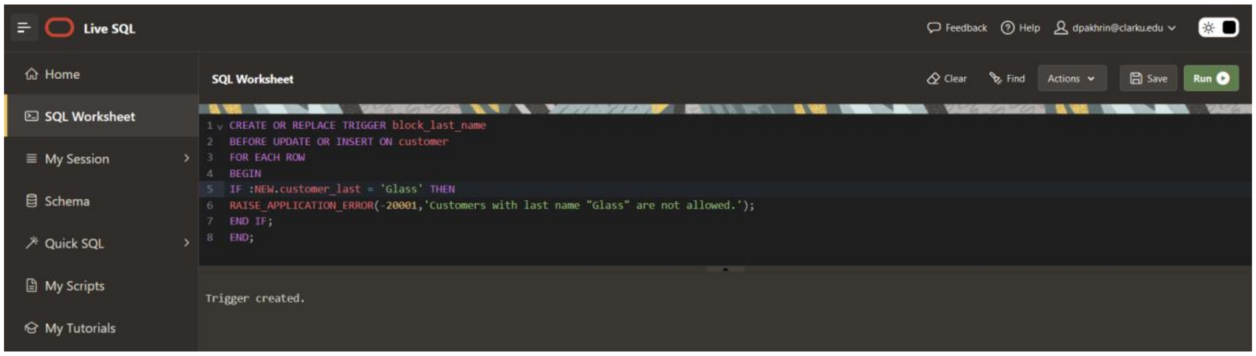
16. SELECT (Screen Shot)



```
SELECT * FROM CUSTOMER
```

1	John	Smith	0
2	Mary	Berman	0
3	Elizabeth	Johnson	0
4	Peter	Quigley	0
5	Stanton	Hurley	0
6	Yvette	Presley	0
7	Hilary	Marsh	0
8	Harry	Joker	0
9	Mary	Smith	0
10	Gabriela	Jury	10.99
11	Dhiraj	Pakhrin	0

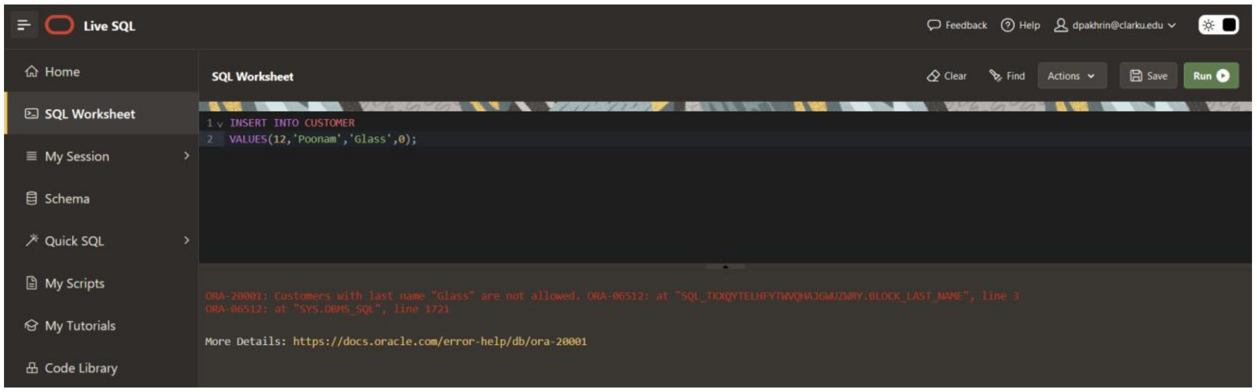
17. CREATE TRIGGER (Screen Shot)



```
CREATE OR REPLACE TRIGGER block_last_name
BEFORE UPDATE OR INSERT ON customer
FOR EACH ROW
BEGIN
IF :NEW.customer_last = 'Glass' THEN
RAISE_APPLICATION_ERROR(-20001,'Customers with last name "Glass" are not allowed.');
END IF;
END;
```

```
Trigger created.
```

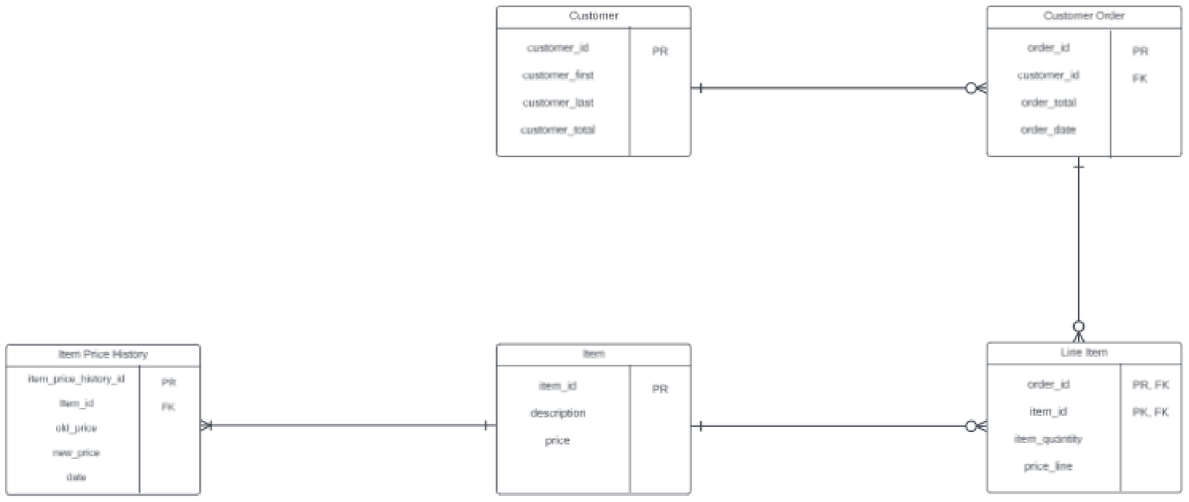
18. INSERT (Screen Shot)



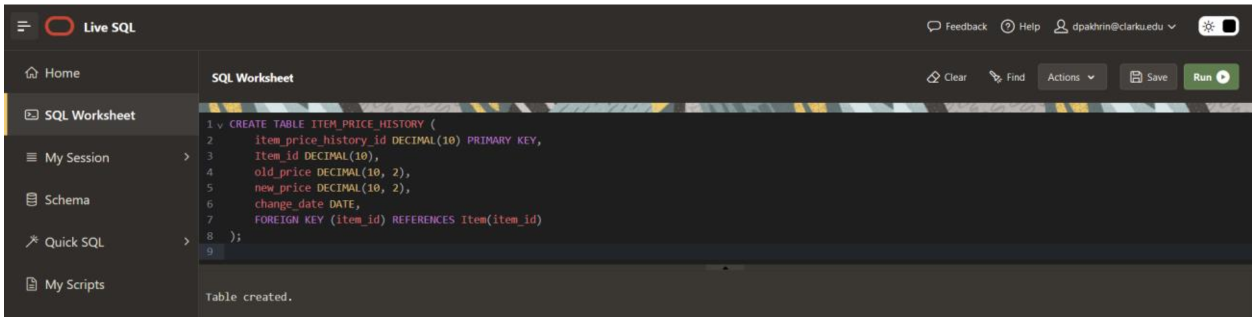
```
INSERT INTO CUSTOMER
VALUES(12,'Poonam','Glass',0);
```

```
ORA-20001: Customers with last name "Glass" are not allowed. ORA-06512: at "SQL_TKXQYTELHFYTWVQHAJGWUZWRY.BLOCK_LAST_NAME", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721

More Details: https://docs.oracle.com/error-help/db/ora-20001
```

19. UPDATED ERD

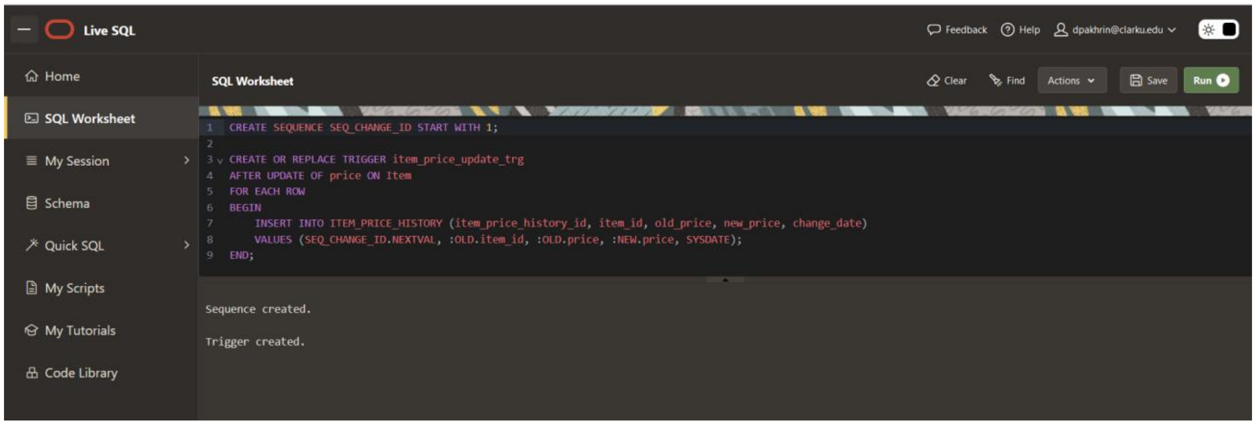


20. CREATE TRIGGER (Screen Shot and Logic Explanation)



```
CREATE TABLE ITEM_PRICE_HISTORY (
    item_price_history_id DECIMAL(10) PRIMARY KEY,
    Item_id DECIMAL(10),
    old_price DECIMAL(10, 2),
    new_price DECIMAL(10, 2),
    change_date DATE,
    FOREIGN KEY (item_id) REFERENCES Item(item_id)
);
```

```
Table created.
```

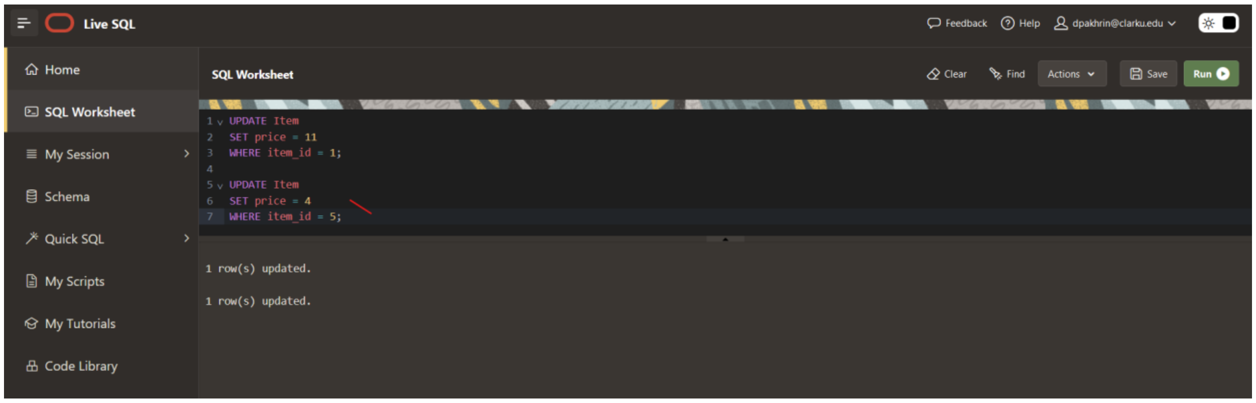


```
CREATE SEQUENCE SEQ_CHANGE_ID START WITH 1;

CREATE OR REPLACE TRIGGER item_price_update_trg
AFTER UPDATE OF price ON Item
FOR EACH ROW
BEGIN
    INSERT INTO ITEM_PRICE_HISTORY (item_price_history_id, item_id, old_price, new_price, change_date)
    VALUES (SEQ_CHANGE_ID.NEXTVAL, :OLD.item_id, :OLD.price, :NEW.price, SYSDATE);
END;
```

```
Sequence created.

Trigger created.
```

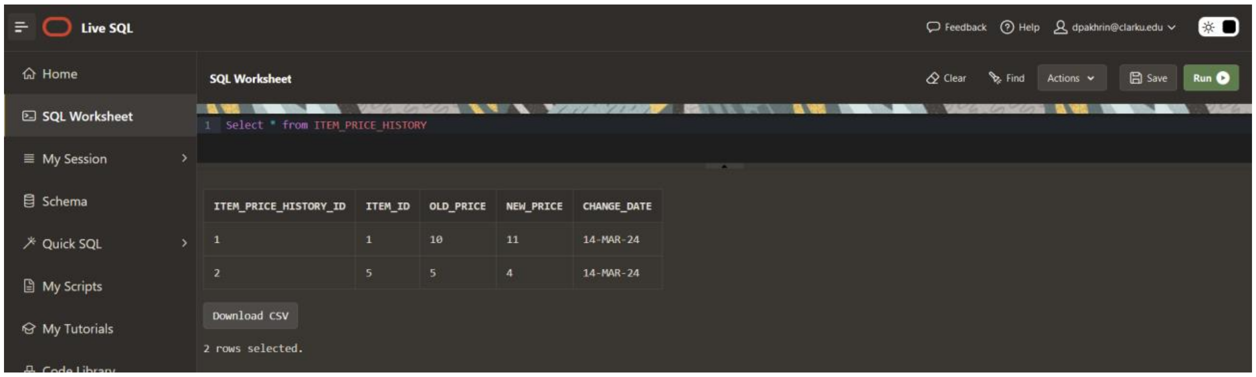
Firstly, we create a new table named item_price_history to store value of price being changed in the item table. After that, we define the trigger named item_price_update trg which is called after the price of the items in the item table is updated. It inserts the record to the item_price_history table whenever it item_price_update_trg is triggered.

21. Price Change and Item_price_history



```
UPDATE Item
SET price = 11
WHERE item_id = 1;

UPDATE Item
SET price = 4
WHERE item_id = 5;
```

```
1 row(s) updated.

1 row(s) updated.
```



```
Select * from ITEM_PRICE_HISTORY
```

ITEM_PRICE_HISTORY_ID	ITEM_ID	OLD_PRICE	NEW_PRICE	CHANGE_DATE
1	1	10	11	14-MAR-24
2	5	5	4	14-MAR-24

```
2 rows selected.
```

Here, the price of the plate and spoon is updated to the new value 11 and 4 respectively. Since it is updated, it calls the trigger item_price_update_trg which is defined in problem 20. Inside the block of this trigger, insert query is defined which records the item_price_history_id, item_id, old_price, new_price, and change_date of the items which is being displayed in using the select query.