I Silas Davis declare that I have completed this assignment completely and entirely on my own, without any consultation with others. I understand that any breach of the UAB Academic Honor Code may result in severe penalties.

Silas Davis

```
sdavis08=> SELECT * FROM Items;
 itemid | available_qty
--------+---------------
(0 rows)

sdavis08=> SELECT * FROM Sell_transactions;
 sell_trans_id | date_of_sale
---------------+--------------
(0 rows)

sdavis08=> SELECT * FROM Sell_items;
 sell_trans_id | itemid | qty | sell_price
---------------+--------+-----+------------
(0 rows)

sdavis08=> SELECT * FROM Return_transactions;
 return_trans_id | date_of_return | return_amount
-----------------+----------------+---------------
(0 rows)

sdavis08=> SELECT * FROM Return_items;
 return_trans_id | sell_trans_id | itemid | return_qty
-----------------+---------------+--------+------------
(0 rows)
```

```
sdavis08=> CREATE TABLE Return_Items(
Return_trans_ID INT REFERENCES Return_transactions(Return_trans_ID), Sell_trans_ID INT REFERE
NCES Sell_transactions(Sell_trans_ID), ItemID INT REFERENCES Items(ItemID),
PRIMARY KEY(Sell_trans_ID, ItemID),
return_qty INT);
```

```
sdavis08=> CREATE TABLE Return_transactions(
sdavis08(> Return_trans_ID INT PRIMARY KEY,
sdavis08(> date_of_return DATE,
sdavis08(> return_amount INT DEFAULT 0);
```

```
sdavis08=> CREATE TABLE Sell_items(
Sell_trans_ID INT REFERENCES Sell_transactions(Sell_trans_ID),
ItemID INT REFERENCES Items(ItemID),
qty INT,
sell_price INT);
```

```
sdavis08=> CREATE TABLE Sell_transactions(
Sell_trans_ID INT PRIMARY KEY,
date_of_sale DATE);
```

```
sdavis08=> CREATE TABLE Items(
sdavis08(> ItemID INT PRIMARY KEY,
sdavis08(> available_qty INT);
```

```
sdavis08=> CREATE OR REPLACE FUNCTION public.return1()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
BEGIN
UPDATE Items
SET available_qty = available_qty + NEW.return_qty WHERE ItemID = NEW.ItemID;
RETURN NULL;
END
$function$
sdavis08-> ;
CREATE FUNCTION
sdavis08=> INSERT INTO Return_items(Return_trans_ID, Sell_trans_ID, ItemID, return_qty) VALUES (01,001,1,1);
INSERT 0 1
sdavis08=> SELECT * FROM return_items;
 return_trans_id | sell_trans_id | itemid | return_qty
-----------------+---------------+--------+------------
               1 |             1 |      1 |          1
(1 row)

sdavis08=> SELECT * FROM items;
 itemid | available_qty
--------+---------------
      2 |             1
      1 |             5
(2 rows)

sdavis08=> INSERT INTO Return_items(Return_trans_ID, Sell_trans_ID, ItemID, return_qty) VALUES (01,001,2,12);
INSERT 0 1
sdavis08=> SELECT * FROM return_items;
 return_trans_id | sell_trans_id | itemid | return_qty
-----------------+---------------+--------+------------
               1 |             1 |      1 |          1
               1 |             1 |      2 |         12
(2 rows)

sdavis08=> SELECT * FROM items;
 itemid | available_qty
--------+---------------
      1 |             5
      2 |            13
(2 rows)
```

```
sdavis08=> CREATE FUNCTION return2() RETURNS trigger AS $return2$
BEGIN
UPDATE Return_transactions
SET return_amount = ((SELECT sell_price FROM Sell_items WHERE Sell_trans_ID = NEW.Sell_trans_ID) * NEW.return_qty) WHERE Return_trans_ID = NEW.Return_trans_ID;
END
$return2$ LANGUAGE plpgsql;
CREATE FUNCTION
sdavis08=> CREATE TRIGGER return2 AFTER INSERT ON Return_items
FOR EACH ROW EXECUTE PROCEDURE return2();
CREATE TRIGGER
```

```
sdavis08=> INSERT INTO Return_items(Return_trans_ID, Sell_trans_ID, ItemID, return_qty) VALUES (01,002,1,1);
INSERT 0 1
sdavis08=> SELECT * FROM Items;
 itemid | available_qty
--------+---------------
      2 |            13
      1 |             6
(2 rows)

sdavis08=> SELECT * FROM Return_transactions;
 return_trans_id | date_of_return | return_amount
-----------------+----------------+---------------
               1 | 2022-11-20     |             2
```

Sdavis08web

Francis!23$<><>