

Statements used to create trigger and associated function :

```
CREATE OR REPLACE FUNCTION addtototalfunc()
RETURNS TRIGGER AS $addtototalfunc$
BEGIN
IF (TG_OP = 'INSERT') THEN
UPDATE Total_Expense
SET total_amount = total_amount + NEW.amount;
ELSEIF (TG_OP = 'DELETE') THEN
UPDATE Total_Expense
SET total_amount = total_amount - OLD.amount;
ELSE
UPDATE Total_Expense
SET total_amount = total_amount - OLD.amount + NEW.amount;
END IF;
RETURN NULL;
END;
$addtototalfunc$ LANGUAGE plpgsql;

CREATE TRIGGER AddToTotal
AFTER INSERT OR DELETE OR UPDATE ON Purchases
FOR EACH ROW
EXECUTE PROCEDURE AddToTotalfunc();
```

TESTING :

Before and after insertion of two values into the Purchases table :

```
justony7=> SELECT * FROM Purchases;
pid | amount
-----+-----
(0 rows)

justony7=> SELECT * FROM Total_Expense;
total_amount
-----
0
(1 row)

justony7=> INSERT INTO Purchases VALUES (1, 2.0), (2, 3.0);
INSERT 0 2
justony7=> SELECT * FROM Purchases;
pid | amount
-----+-----
1 | 2
2 | 3
(2 rows)

justony7=> SELECT * FROM Total_Expense;
total_amount
-----
5
(1 row)
```

Deletion of the value with PID = 2 :

```
justony7=> DELETE FROM Purchases WHERE pid = 2;
DELETE 1
justony7=> SELECT * FROM Purchases;
pid | amount
-----+-----
1 | 2
(1 row)

justony7=> SELECT * FROM Total_Expense;
total_amount
-----
2
(1 row)
```

Updating the amount to 4 for the Purchases tuple with PID = 1 :

```
justony7=> UPDATE Purchases
justony7-> SET amount = 4
justony7-> WHERE pid = 1;
UPDATE 1
justony7=> SELECT * FROM Purchases;
  pid | amount
-----+-----
    1 |      4
(1 row)

justony7=> SELECT * FROM Total_Expense;
  total_amount
-----
            4
(1 row)
```

I, Justin Anthony Timberlake, 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.