

Name: Parth A Patel

B00811471

Course: CSCI 4140

Submission: Assignment 1

Note: This assignment is done on a Platform that supports/uses PostgreSQL.

## Table of Contents

<b><i>ER Model</i></b> .....	<b>3</b>
<b><i>Screenshots</i></b> .....	<b>7</b>
Insert Client .....	7
Insert Part .....	8
Insert Order .....	9
Insert Line .....	10
Stored Procedure .....	11
Trigger .....	12

## ER Model

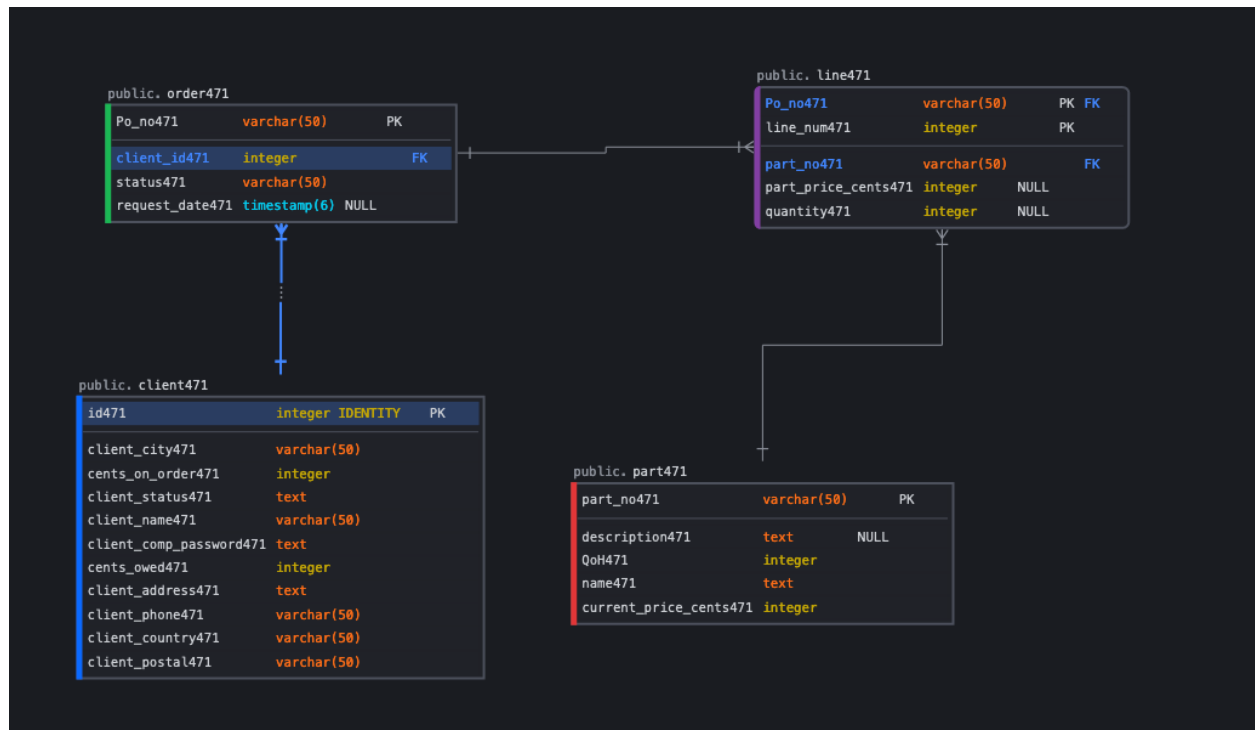


Figure 1 Shows complete ER Diagram

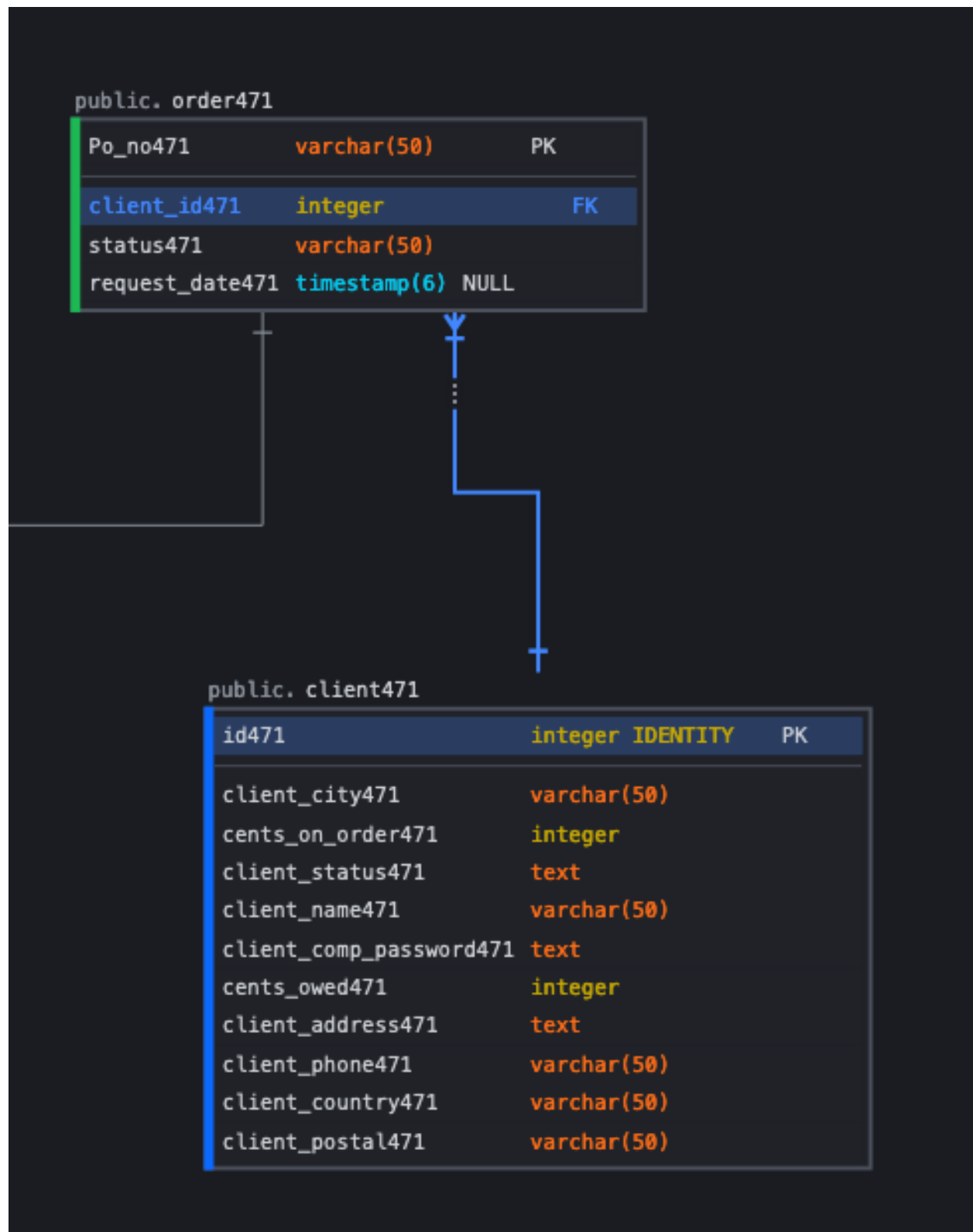


Figure 2 Depicts relation between client and order

public. part471

part_no471	varchar(50)	PK
description471	text	NULL
QoH471	integer	
name471	text	
current_price_cents471	integer	

public. line471

Po_no471	varchar(50)	PK	FK
line_num471	integer	PK	
part_no471	varchar(50)		FK
part_price_cents471	integer	NULL	
quantity471	integer	NULL	

Figure 3 Depicts relation between line and part



Figure 4 Depicts relation between order and line

# Screenshots

## Insert Client

```
1 set schema 'public';
2
3 INSERT INTO client471 (client_city471, cents_on_order471, client_status471,client_name471, client_comp_password471,cents_owed471,client_address471,client_phone471,client_country471,
4 client_postal471)
5 VALUES('Boston', 0, 'Active', 'TimHortons','passwordxyz',0,'Some St','13853456782','USA','02118');
6
```

```
1 select * from public.client471;
```

Results

id471	client_city471	cents_on_order	client_status4	client_name471	client_comp_pa	cents_owed471	client_address	client_phone47	client_country	client_postal4
1	"Boston"	0	"Active"	"TimHortons"	"passwordxyz"	0	"Some St"	"13853456782"	"USA"	"02118"

## Insert Part

```
set schema 'public';  
  
INSERT INTO  
part471 (part_no471, description471, name471, current_price_cents471, qoh471)  
VALUES('QD2-00350', 'A can of coffee', 'Nescafe Gold Blend Coffee 7oz', 3499, 10);
```

```
1 select * from public.part471;
```

Results ▾

part_no471	description471	qoh471	name471	current_price_
"QD2-00350"	"A can of coffee"	10	"Nescafe Gold Blend Coffee 7oz"	3499



## Insert Order

```
1  INSERT INTO order471(po_no471, client_id471, status471, request_date471) VALUES ('P0001', 1, 'Active', now());
```

Results ▾

Success. No rows returned

```
1  select * from public.order471;
```

Results ▾

po_no471	client_id471	status471	request_date471
"P0001"	1	"Active"	"2022-09-27 02:31:36.66807"

## Insert Line

```
1  INSERT INTO line471(part_no471, part_price_cents471, quantity471, po_no471) VALUES ('QD2-00350', 5530, 2, 'P0001');
```

Results ▾

Success. No rows returned

```
1  select * from line471
```

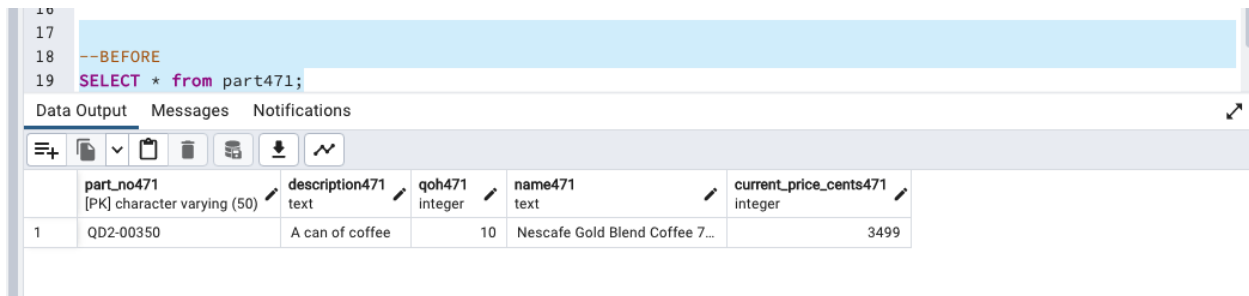
Results ▾

po_no471	line_num471	part_no471	part_price_cen	quantity471
"P0001"	3	"QD2-00350"	5530	2

## Stored Procedure

Steps Taken to test:

1. Select all Parts (To see what Parts exists on the table)



The screenshot shows a PostgreSQL query editor with the following SQL statement:

```
--BEFORE
SELECT * from part471;
```

Below the query, there are tabs for "Data Output", "Messages", and "Notifications". The "Data Output" tab is active, displaying a table with the following columns and data:

	part_no471 [PK] character varying (50)	description471 text	qoh471 integer	name471 text	current_price_cents471 integer
1	QD2-00350	A can of coffee	10	Nescafe Gold Blend Coffee 7...	3499

2. Create and Call Procedure. It should return 3499 when provided with the part\_no above.

Note:

In the script below RAISE NOTICE logs output to the terminal in PostgreSQL.



The screenshot shows a PostgreSQL query editor with the following SQL script:

```
1 CREATE OR REPLACE PROCEDURE get_part_price(IN part_number471 varchar, OUT part_price_cents471 numeric)
2 AS $$
3 BEGIN
4     SELECT current_price_cents471 INTO $2 from part471 WHERE part471.part_no471 = $1;
5 END;
6 $$ LANGUAGE plpgsql;
7
8
9 DO $$
10 DECLARE output_var471 numeric;
11 BEGIN
12     CALL get_part_price('QD2-00350', output_var471);
13     RAISE NOTICE 'Procedure Output = %', output_var471;
14 END;
15 $$;
16
```

Below the script, there are tabs for "Data Output", "Messages", and "Notifications". The "Messages" tab is active, displaying the following output:

```
NOTICE: Procedure Output = 3499
DO

Query returned successfully in 61 msec.
```

## Trigger

### Steps Taken to test:

#### 1. Insert new client

```
13 INSERT INTO client471 (client_city471, cents_on_order471, client_status471, client_name471, client_comp_password471, cents_owed471, client_address471, client_phone471)
14 VALUES('Florida', 0, 'Active', 'Test', 'passwordxyz', 0, 'Some St', '13053456782', 'USA', '09323');
15
16
```

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 62 msec.

#### 2. Verify client insertion

```
15
16 Select * from client471;
```

Data Output Messages Notifications

	id471 [PK] integer	client_city471 character varying (50)	cents_on_order471 integer	client_status471 text	client_name471 character varying (50)	client_comp_password471 text	cents_owed471 integer	client_address471 text	client_phone471 character varying (50)	client_country471 character varying (50)	client_p charact
1	1	Boston	0	Active	TimHortons	passwordxyz	22120	Some St	13053456782	USA	02118
2	2	Florida	0	Active	Test	passwordxyz	0	Some St	13053456782	USA	09323

Note: On the screenshot above client 2 has cents\_owed471 = 0

#### 3. Insert new Part

```
18
19 INSERT INTO
20 part471 (part_no471, description471, name471, current_price_cents471, qoh471)
21 VALUES('QD2-00001', 'Pen', 'Blue Pen', 20, 5);
22
```

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 103 msec.

#### 4. Verify part insertion

```
22  
23 Select * from part471;  
24
```

Data Output Messages Notifications

	part_no471 [PK] character varying (50)	description471 text	qoh471 integer	name471 text	current_price_cents471 integer
1	QD2-00350	A can of coffee	10	Nescafe ...	3499
2	QD2-00001	Pen	5	Blue Pen	20

Realized I put the current\_price\_cents471 for part\_no 2 as 20 instead of 2000.

Going to update the value

```
29 Select * from line471;  
30  
31 Update part471 set current_price_cents471 = 2000 where part_no471 = 'QD2-00001';  
32
```

Data Output Messages Notifications

UPDATE 1

Query returned successfully in 71 msec.

```
32 select * from part471  
33
```

Data Output Messages Notifications

	part_no471 [PK] character varying (50)	description471 text	qoh471 integer	name471 text	current_price_cents471 integer
1	QD2-00350	A can of coffee	10	Nescafe ...	3499
2	QD2-00001	Pen	5	Blue Pen	2000

## 5. Insert new Purchase Order

```
24  
25  
26 INSERT INTO order471(po_no471, client_id471, status471, request_date471) VALUES ('P0002', 2, 'Active', now());
```

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 137 msec.

## 6. Verify Purchase Order Insertion

```
24  
25 Select * from order471;  
26
```

Data Output Messages Notifications

	po_no471 [PK] character varying (50)	client_id471 integer	status471 character varying (50)	request_date471 timestamp without time zone
1	P0001	1	Active	2022-09-27 02:31:36.66807
2	P0002	2	Active	2022-09-27 13:05:57.485188

## 7. Insert a line item for the new order and part inserted above.

```
27 -- Using the same price as during insertion  
28 INSERT INTO line471(part_no471, part_price_cents471, quantity471, po_no471) VALUES ('QD2-00001', 2000, 4, 'P0002');  
29  
30 Select * from line471;
```

Data Output Messages Notifications

NOTICE: Begin  
NOTICE: Client ID: 2  
NOTICE: Previous Cents Owed: 0  
NOTICE: New Cents Owed: 8000  
NOTICE: Done  
INSERT 0 1

Query returned successfully in 61 msec.

## 8. Verify line Insertion

```
29
30 Select * from line471;
31
```

Data Output Messages Notifications



	po_no471 [PK] character varying (50)	line_num471 [PK] integer	part_no471 character varying (50)	part_price_cents471 integer	quantity471 integer
1	P0001	2	QD2-00350	5530	2
2	P0001	3	QD2-00350	5530	2
3	P0002	4	QD2-00001	2000	4

## 9. Now we can check whether the cents\_owed471 for the inserted client increased by 8000. (part\_price \* quantity)

Note: The client of interest is client 2 that we inserted above.

```
8
9
10 select * from public.client471;
11
```

Data Output Messages Notifications



	id471 [PK] integer	client_city471 character varying (50)	cents_on_order471 integer	client_status471 text	client_name471 character varying (50)	client_comp_password471 text	cents_owed471 integer	client_address471 text	client_phone471 character varying (50)
1	1	Boston	0	Active	TimHortons	passwordxyz	22120	Some St	13053456782
2	2	Florida	0	Active	Test	passwordxyz	8000	Some St	13053456782

We see that the trigger functions as intended as the cents\_owed471 was incremented when a line was inserted.

## 10. Further we can add another line on the purchase order with the same details as above. The cents\_owed471 should be incremented by 8000.

```
30
31 -- Using the same price as during insertion
32 INSERT INTO line471(part_no471, part_price_cents471, quantity471, po_no471) VALUES ('QD2-00001', 2000, 4, 'P0002');
33
```

Data Output Messages Notifications

```
NOTICE: Begin
NOTICE: Client ID: 2
NOTICE: Previous Cents Owed: 8000
NOTICE: New Cents Owed: 8000
NOTICE: Done
INSERT 0 1

Query returned successfully in 111 msec.
```

```
33
34 Select * from line471;
35
```

Data Output Messages Notifications

	po_no471 [PK] character varying (50)	line_num471 [PK] integer	part_no471 character varying (50)	part_price_cents471 integer	quantity471 integer
1	P0001	2	QD2-00350	5530	2
2	P0001	3	QD2-00350	5530	2
3	P0002	4	QD2-00001	2000	4
4	P0002	5	QD2-00001	2000	4

```
9
10 select * from public.client471;
11
12
```

Data Output Messages Notifications

	id471 [PK] integer	client_city471 character varying (50)	cents_on_order471 integer	client_status471 text	client_name471 character varying (50)	client_comp_password471 text	cents_owed471 integer	client_address471 text	client_phone471 character varying (50)
1	1	Boston	0	Active	TimHortons	passwordxyz	22120	Some St	13053456782
2	2	Florida	0	Active	Test	passwordxyz	16000	Some St	13053456782

We can see from the screenshot above that the cents\_owed471 was incremented again by 8000 resulting in 16000.