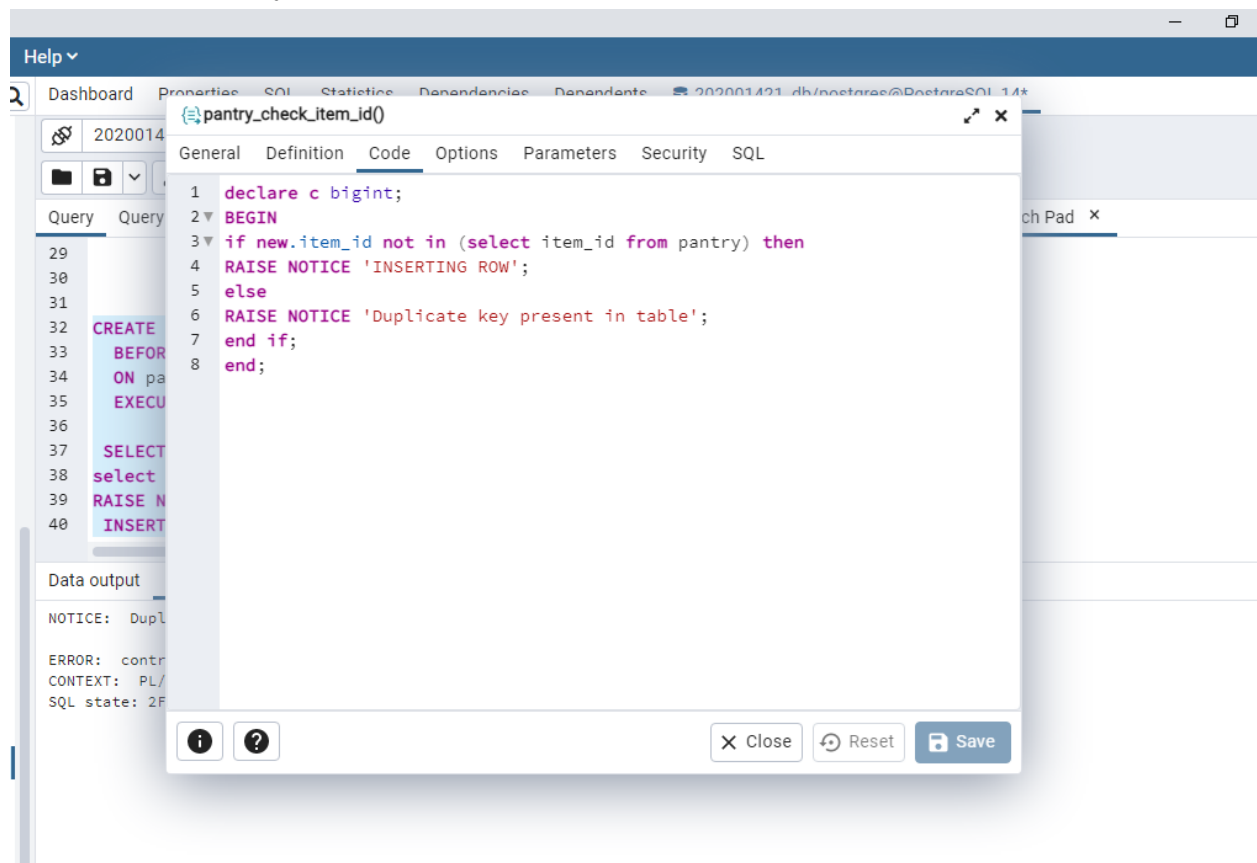


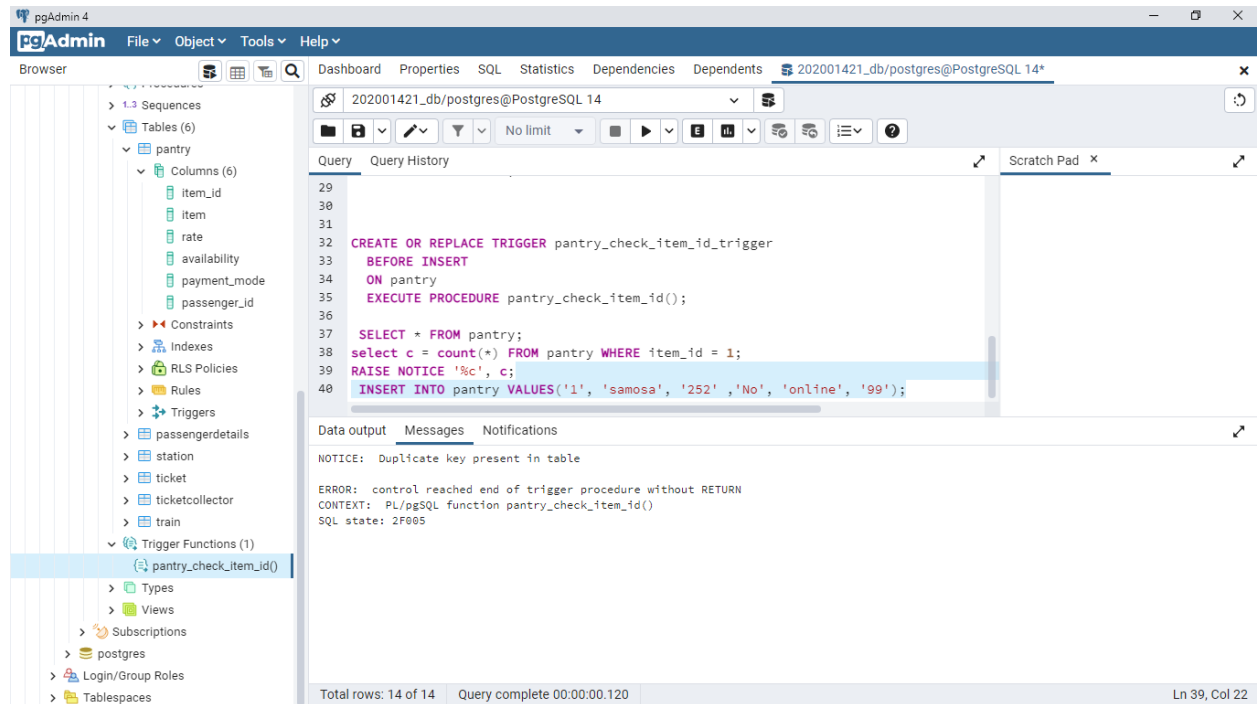
202001421 Lab 3

1. Create a trigger on the Table of your choice to check if the Primary key ID already exists or not before inserting a new record. & Send a custom reply instead of an error message.

```
CREATE OR REPLACE TRIGGER pantry_check_item_id_trigger
BEFORE INSERT
ON pantry
EXECUTE PROCEDURE pantry_check_item_id();
```

```
SELECT * FROM pantry;
select c = count(*) FROM pantry WHERE item_id = 1;
RAISE NOTICE '%c', c;
INSERT INTO pantry VALUES('1', 'samosa', '252', 'No', 'online', '99');
```





2. Create a trigger on the Table of your choice to check if the Foreign key ID already exists or not before inserting a new record. & Send a custom reply instead of an error message.

1. Find the ticket id for an amount greater than 2000. Return a temp table with ticket id, amount and status in the result table. Make sure below both records are visible in results.

```

CREATE OR REPLACE function "rail_db"."ticket_amt_gt_2000"()
RETURNS TABLE(a integer, b integer, c character(30))
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN
RETURN QUERY EXECUTE format ('SELECT ticket_id, amount, status FROM "rail_db".ticket
where amount > 2000');
END;
$BODY$

```

```
SELECT * from ticket;
```

```
select "rail_db"."ticket_amt_gt_2000"();
```

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays a tree view of the database structure, including 'Sequences', 'Tables (6)', 'Columns (6)', 'Constraints', 'Indexes', 'RLS Policies', 'Rules', 'Triggers', 'passengerdetails', 'station', 'ticket', 'ticketcollector', 'train', and 'Trigger Functions (1)'. The 'Trigger Functions (1)' folder is expanded, showing 'pantry_check_item_id()'. The main pane displays a SQL query in the 'Query' tab, which is a PL/pgSQL function named 'ticket_amt_gt_2000'. The query is as follows:

```

1 CREATE OR REPLACE function "rail_db"."ticket_amt_gt_2000"()
2 RETURNS TABLE(a integer, b integer, c character(30))
3 LANGUAGE 'plpgsql'
4 AS $BODY$
5 BEGIN
6 RETURN QUERY EXECUTE format ('SELECT ticket_id, amount, status FROM "rail_db".ticket
7 END;
8 $BODY$
9
10 SELECT * from ticket;
11
12 select "rail_db"."ticket_amt_gt_2000"();

```

The 'Data output' tab shows the results of the query, which is a table with 3 columns (a, b, c) and 123 rows. The first 7 rows are visible:

a	b	c
1000001	4592	con...
1000002	4733	con...
1000003	2620	con...
1000004	3946	con...
1000006	3103	wal...
1000007	2979	wal...
1000008	3044	wal...

The status bar at the bottom indicates 'Total rows: 123 of 123' and 'Query complete 00:00:00.140'.

123 rows

2. Create a new column, "total_amount" in the ticket table. Call the function to calculate the total amount with the formula amount+0.12*amount. Display the updated table.

CREATE OR REPLACE function "rail_db"."calc_tot_amt"()

returns integer

LANGUAGE 'plpgsql'

AS \$BODY\$

BEGIN

UPDATE "rail_db".ticket set total_amount = amount + 0.12*amount;

return 0;

END;

\$BODY\$

select "rail_db"."calc_tot_amt"();

SELECT * FROM ticket;

pgAdmin 4

File Object Tools Help

202001421_db/postgres@PostgreSQL 14*

Query

```

17 CREATE OR REPLACE function "rail_db"."calc_tot_amt"()
18 returns integer
19 LANGUAGE 'plpgsql'
20 AS $BODY$
21 BEGIN
22 UPDATE "rail_db"."ticket set total_amount = amount + 0.12*amount;
23 return 0;
24 END;
25 $BODY$
26
27 select "rail_db"."calc_tot_amt"();
28 SELECT * FROM ticket;

```

Data output

	ticket_id [PK] integer	passenger_id integer	status character (30)	amount integer	train_id integer	total_amount double precision
1	1000040	138	confirm	2474	140	2770.88
2	1000041	139	waiting	4242	141	4751.04
3	1000042	140	waiting	3273	142	3665.76
4	1000043	141	waiting	4799	143	5374.88
5	1000044	142	confirm	4292	144	4807.04
6	1000045	143	confirm	2512	145	2813.44
7	1000046	144	waiting	4335	146	4855.2
8	1000047	145	confirm	2354	147	2626.48

Total rows: 150 of 150 Query complete 00:00:02.597 Ln 25, Col 7

1. Create a column “state” in the station table. Create a trigger to put the default value 1 for every new entry if nothing is given by the user. Insert new records and check the functionality of Triggers.