------------------------------------------------------------TABLES----------------------------------------------------------------

create table orders (o\_id int primary key, cust\_id int, o\_date date, status varchar(50), p\_id int, quantity int, total\_price numeric );

create table customer (cust\_id int primary key,cust\_name varchar(50),cust\_city varchar(50), email varchar(50),ph\_no numeric,gender varchar(10));

create table catagory (cat\_id int primary key,cat\_name varchar(10));

create table product (p\_id int primary key ,cat\_id int ,p\_name varchar(50), price numeric, size varchar(10), color varchar(20),brand varchar(50), material varchar(30),stock\_quantity int, rating numeric(2,1));

**-----------------------------------------------------INSERT QUERY-------------**------------------------------------------------

insert into customer values(1,'Rohit Sharma','Ahmedabad','Rohit45@gmail',9855552533,'Male'),

(2,'Riya Shah','Mumbai','Riya22@gmail',9876543210,'Female'),

(3,'Virat Kohli','Surat','Virat18@gamil',9362581470,'Male'),

(4,'Shreyas Iyer','Pune','Shreyas19@gamil',9352581470,'Male'),

(5,'Axar Patel','Surat','Axar66@gamil',9392588470,'Male'),

(6,'Jiya Patel','Vapi','Jiya26@gmail',9876223210,'Female'),

(7'Shreya Sharma','Baroda','Riya22@gmail',9321654870,'Female'),

(8,'Kartik Aaryan','Ahmedabad','Kartik555@gamil',9214581470,'Male'),

(9,'Tara Sutaria','Surat','Tara45@gamil',9352551470,'Female'),

(10,'Shefali Verma','Baroda','Shefali75@gamil',9352542870,'Female'),

(11,'Hritik Roshan','Vapi','Hritik585@gamil',9224581000,'Male'),

(12,'Sara Tendulkar','Mumbai','Sara75@gamil',9352444470,'Female'),

(13,'Shubman Gill','Mumbai','Gill77@gamil',9224581100,'Male'),

(14,'Daya Gada','Surat','Daya664@gamil',9355874470,'Female'),

(15,'Kriti Sanon','Pune','Kriti554@gamil',7894561230,'Female');

insert into product values(1,1,'Shirt',500,'S','Black','H&M','Cotton',5,4.6),

(2,1,'Shirt',500,'L','Black','H&M','Cotton',8,4.6),

(3,1,'Shirt',600,'S','Red','H&M','Cotton',4,4.6),

(4,1,'Shirt',600,'L','Red','H&M','Cotton',9,4.6),

(5,2,'Top',700,'S','Pink','Zara','Polyester',5,4.8),

(6,2,'Top',700,'L','Pink','Zara','Polyester',8,4.8),

(7,1,'Jeans',900,'S','Blue','Max','Jeans',12,3.9),

(8,1,'Jeans',900,'L','Blue','Max','Jeans',14,3.9),

(9,2,'Jeans',850,'S','Black','Ketch','Jeans',8,4.2),

(10,2,'Jeans',850,'L','Black','Ketch','Jeans',11,4.2),

(11,1,'T-shirt',450,'S','White','Zudio','Rayon',3,3.9),

(12,1,'T-shirt',450,'L','White','Zudio','Rayon',7,3.9),

(13,2,'T-shirt',650,'S','Peach','Gap','Polyester',5,4.2),

(14,2,'T-shirt',650,'L','Peach','Gap','Polyester',9,4.2),

(15,1,'Kurta',1000,'S','Orange','Anouk','Cotton',9,4.3),

(16,1,'Kurta',1000,'L','Orange','Anouk','Cotton',4,4.3),

(17,1,'Kurta',500,'S','Blue','Fubar','Cotton',15,4.5),

(18,1,'Kurta',500,'L','Blue','Fubar','Cotton',12,4.5),

(19,2,'Saree',800,'S','Magenta','Kalini','Silk',13,4.4),

(20,2,'Saree',800,'L','Magenta','Kalini','Silk',14,4.4),

(21,2,'Saree',1500,'S','Black','Taavi','Arganza',19,4.8),

(22,2,'Saree',1500,'L','Black','Taavi','Arganza',17,4.8),

(23,2,'Co-ord',950,'S','Wine','Griva','Lyara',13,3.3),

(24,2,'Co-ord',950,'L','Wine','Griva','Lyara',13,3.3),

(25,1,'Co-ord',1100,'S','Green','Bani','Rayon',4,2.3),

(26,1,'Co-ord',1100,'L','Green','Bani','Rayon',4,2.3),

(27,1,'Jacket',750,'S','Blue','Levis','Denim',7,4.7),

(28,1,'Jacket',750,'L','Blue','Levis','Denim',5,4.7),

(29,2,'Jacket',980,'S','Brown','Diwaah','Polyester',2,4.0),

(30,2,'Jacket',980,'L','Brown','Diwaah','Polyester',8,4.0);

insert into catagory values(1,'Male'),(2,'Female');

insert into orders values(1,1,'10-03-2023','Placed',7,1,900),

(2,3,'2023-10-01','Placed',25,2,2200),

(3,14,'2023-09-25','Placed',19,1,800),

(4,8,'2023-09-20','Placed',27,1,750),

(5,4,'2023-08-06','Delivered',1,2,1000),

(6,8,'2023-07-22','Delivered',12,1,450),

(7,10,'2023-06-21','Delivered',23,1,950),

(8,2,'2023-05-05','Delivered',5,2,1400);

---------------------------------------FUNCTIONS & PROCEDURES---------------------------------------------------------

create or replace procedure quantity\_lessthan (q int) as

$$

declare

a\_product product.p\_name%type;

a\_quantity int;

a\_id int;

a1 cursor for select p\_id,p\_name,stock\_quantity from product where stock\_quantity<=q;

begin

raise notice 'id name quantity';

open a1;

loop

fetch a1 into a\_id,a\_product,a\_quantity;

exit when not found;

raise notice '% , % , %',a\_id,a\_product,a\_quantity;

end loop;

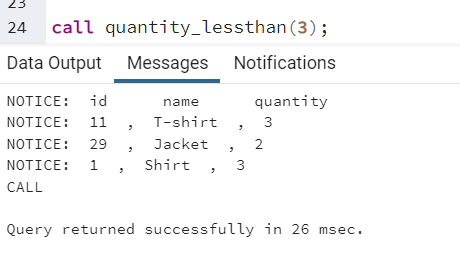
close a1;

end;

$$

language plpgsql;

call quantity\_lessthan(3);



create or replace procedure buy\_ (a\_cid int,a\_pid int,q int,a\_oid int) as

$$

declare

c1\_pid int;

a\_stock int;

a\_price numeric;

begin

select stock\_quantity,price into a\_stock,a\_price from product where p\_id=a\_pid;

if a\_stock>q then

insert into orders values (a\_oid,a\_cid, current\_date, 'placed',a\_pid,q,a\_price\*q);

update product set stock\_quantity=stock\_quantity-q where p\_id=a\_pid;

raise notice 'order succesfully placed ';

else

raise notice 'insufficient stock !!';

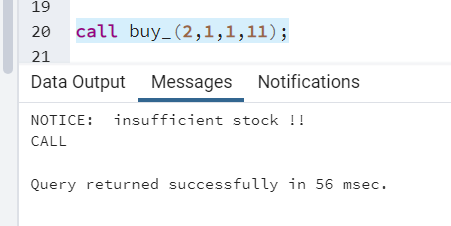
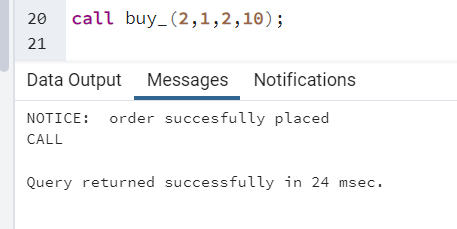
end if;

end;

$$

language plpgsql;

call buy\_(3,8,2,12);



create or replace procedure search\_by\_type\_range(p1 varchar(50), st numeric, z numeric) as

$$

declare

res product%rowtype;

c1\_id product.p\_id%type;

c1\_name product.p\_name%type;

c1\_price product.price%type;

c1\_size product.size%type;

c1\_color product.color%type;

c1\_brand product.color%type;

c1\_material product.material%type;

c1\_quantity product.stock\_quantity%type;

c1\_rating product.rating%type;

c1 cursor for select p\_id ,p\_name, price, size, color,brand, material, stock\_quantity, rating from product where p\_name=p1 and price between st and z;

begin

open c1;

loop

fetch c1 into c1\_id ,c1\_name ,c1\_price ,c1\_size ,c1\_color ,c1\_brand ,c1\_material ,c1\_quantity ,c1\_rating;

exit when not found;

raise notice ' % , % , % , % , % , % , % , % , % ',c1\_id ,c1\_name ,c1\_price ,c1\_size ,c1\_color ,c1\_brand ,c1\_material ,c1\_quantity ,c1\_rating;

end loop;

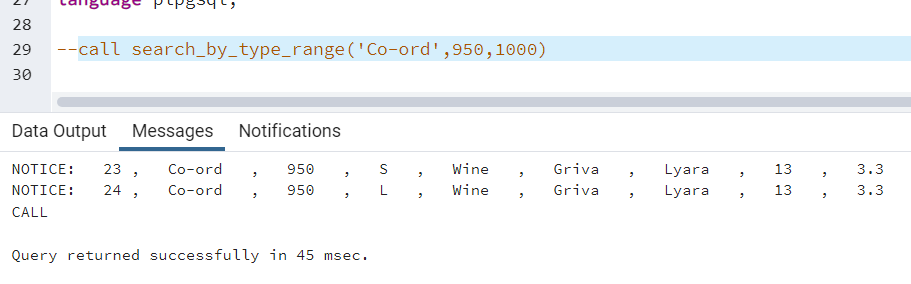
close c1;

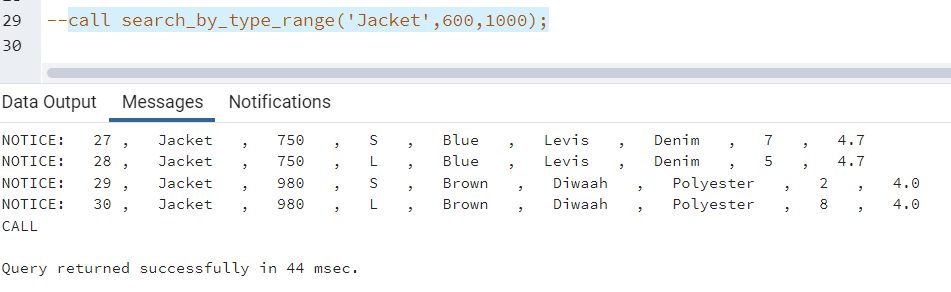
end;

$$

language plpgsql;

call search\_by\_type\_range('Co-ord',950,1000)





CREATE OR REPLACE PROCEDURE calculate\_final\_bill(id INT) AS

$$

DECLARE

amt NUMERIC;

final\_amt NUMERIC;

gst\_rate NUMERIC := 0.18; -- 18% GST rate

c1\_row orders%ROWTYPE;

c1 CURSOR FOR SELECT \* FROM orders WHERE cust\_id = id and status='placed';

BEGIN

RAISE NOTICE 'Customer ID: %', id;

RAISE NOTICE ' O\_ID Order Date Status Product ID Quantity Price';

OPEN c1;

LOOP

FETCH c1 INTO c1\_row;

EXIT WHEN NOT FOUND;

RAISE NOTICE ' % % % % % %', c1\_row.o\_id, c1\_row.o\_date, c1\_row.status, c1\_row.p\_id, c1\_row.quantity, c1\_row.total\_price;

END LOOP;

close c1;

SELECT SUM(total\_price) INTO amt FROM orders WHERE cust\_id = id and status='placed';

final\_amt = amt + (amt \* gst\_rate);

RAISE NOTICE 'GST charge: 18%%';

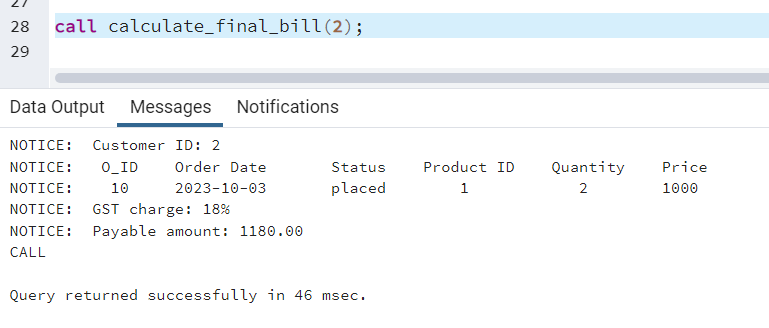
RAISE NOTICE 'Payable amount: %', final\_amt;

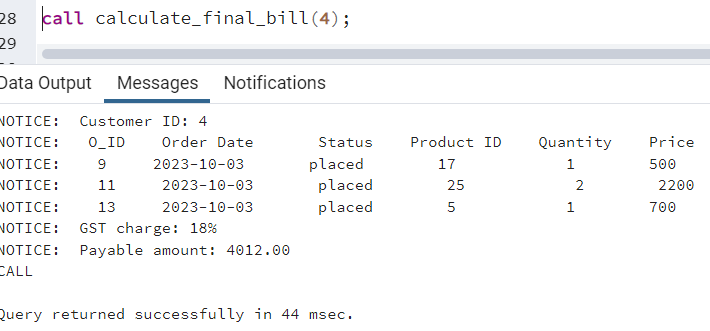
END;

$$

LANGUAGE plpgsql;

call calculate\_final\_bill(2);





create or replace function change\_price (a\_name varchar(50),inc int) returns varchar as

$$

declare

a int;

p1\_name product.p\_name%type;

p1\_price product.price%type;

p1 cursor for select p\_name,price from product;

begin

open p1;

loop

fetch p1 into p1\_name,p1\_price;

exit when not found;

if p1\_name=a\_name then

update product set price=price+inc where p\_name=a\_name;

return 'price updated !!';

end if;

end loop;

close p1;

get diagnostics a=row\_count;

if a=0 then

return 'no product available';

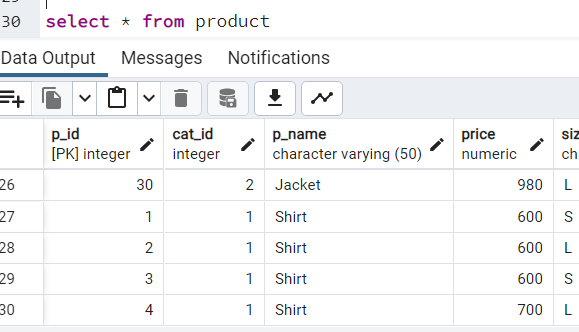
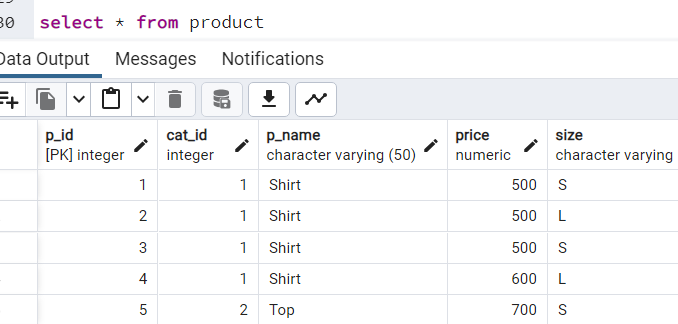
end if;

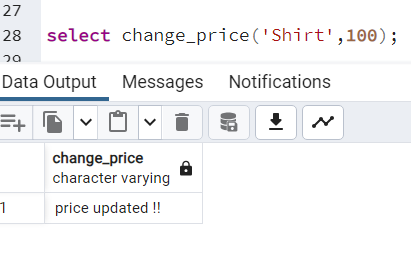
end;

$$

language plpgsql;

select change\_price('Shirt',100);





----------------------------------------------------SQL QUERIES----------------------------------------------------------------

select \* from orders where cust\_id=5;

select \* from product where p\_name='kurta' and size='s';

select \* from product inner join catagory on product.cat\_id=catagory.cat\_id where cat\_name='Female' and p\_name='Top';

select cust\_id, sum(total\_price) from orders group by cust\_id order by sum(total\_price) desc limit 1;

select \* from orders where status='delivered';

select p\_id, p\_name from product where stock\_quantity between 1 and 5;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_