**Procedures**

# Procedures

**Procedure is a block of code which is given a name and stored in the database.**

# Procedure can include

**1. SQL queries**

**2. DML,DDL, DCL, and TCL commands**

**3. Collection Types**

**4. Cursors**

**5. Loop & If Else statements**

**6. Variables**

**7. Exception Handling, etc.**

# What is the purpose of using a Procedure?

**Procedures are generally used to do things which is not possible in SQL queries.**

**Depending on the project requirement, we can bundle multiple queries into Procedures or We may build entire software logic inside Procedure which may include validation checks, processing and querying of data, and much more.**

# Syntax for Procedure Creation

## PostgresSQL

**create or replace procedure pr\_name(p\_name varchar, p\_age int)**

**language plpgsql**

**as $$ -- It is used to print the values with single quotes without using escape chars.**

**declare**

**variable**

**begin**

**procedure body - All logics**

**end;**

**$$**

## Oracle

**create or replace procedure pr\_name(p\_name varchar, p\_age int)**

**as**

**variable**

**begin**

**procedure body - All logics**

**end;**

# Simple Procedure Sample

## For every iPhone 13 Pro Max sales, modify the database tables accordingly

**create or replace procedure pr\_buy\_product()**

**language plpgsql**

**as $$**

**declare**

**v\_prod\_code varchar(20);**

**v\_price int;**

**begin**

**select prod\_code, price**

**into v\_prod\_code, v\_price**

**from p\_products**

**where prod\_name = 'Iphone 13 Max Pro';**

**insert into p\_sales(order\_date, prod\_code, qty\_ordered, sale\_price)**

**values (current\_date, v\_prod\_code, 1, (v\_price \* 1));**

**update p\_products**

**set qty\_remaining = (qty\_remaining - 1),**

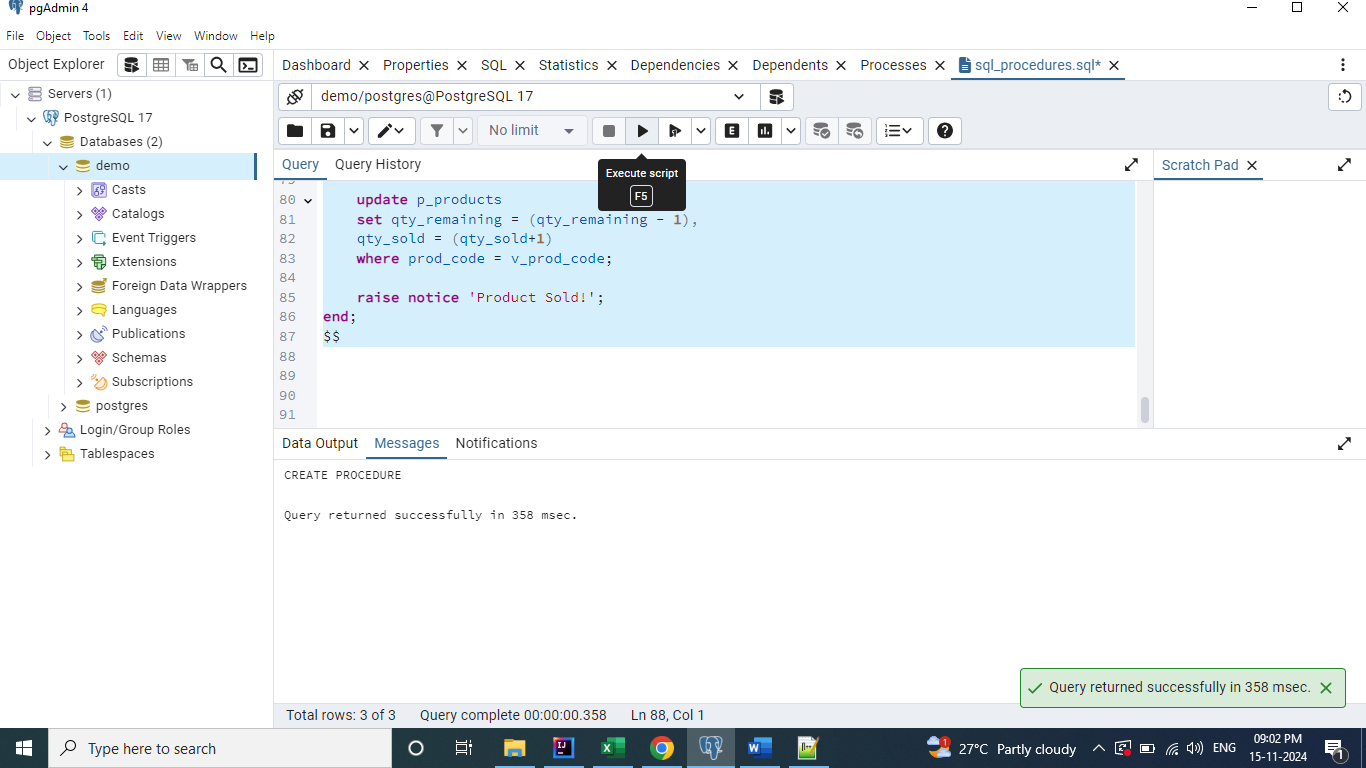
**qty\_sold = (qty\_sold+1)**

**where prod\_code = v\_prod\_code;**

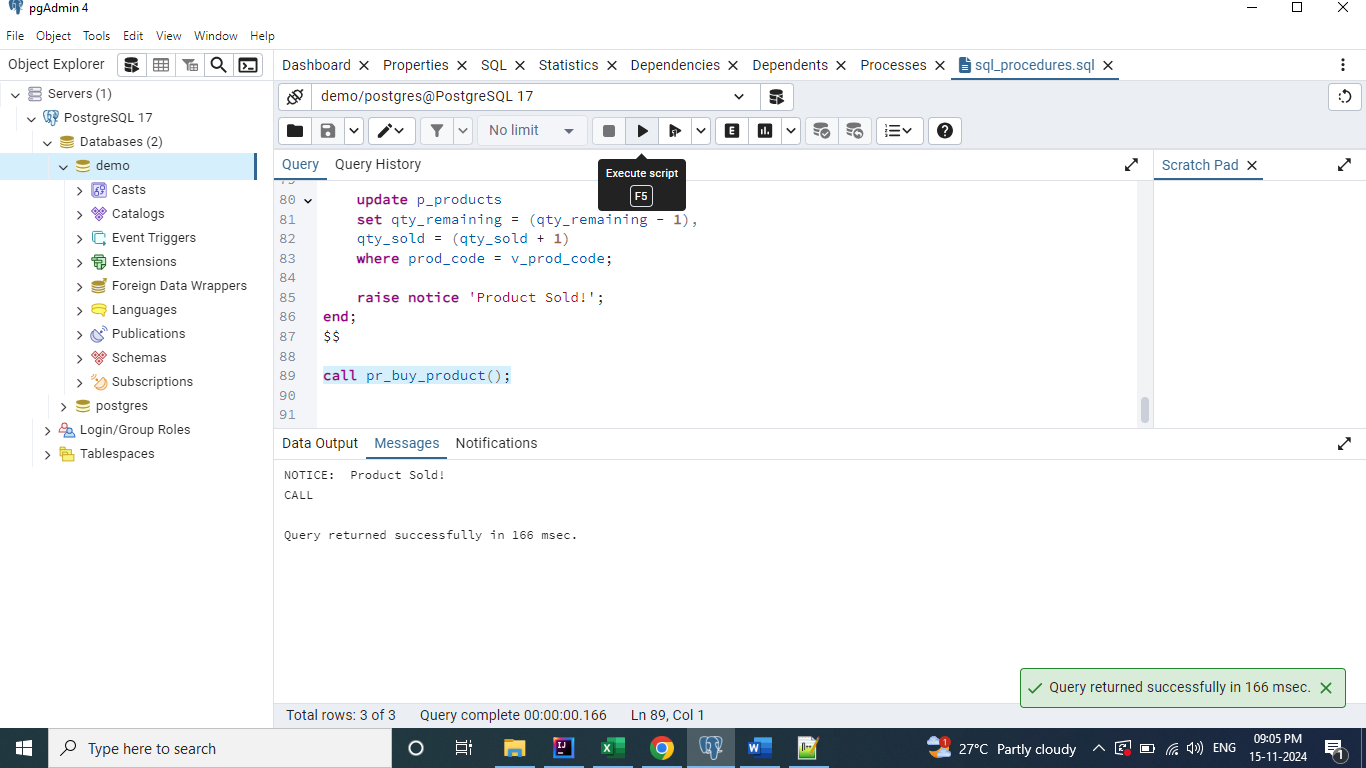
**raise notice 'Product Sold!';**

**end;**

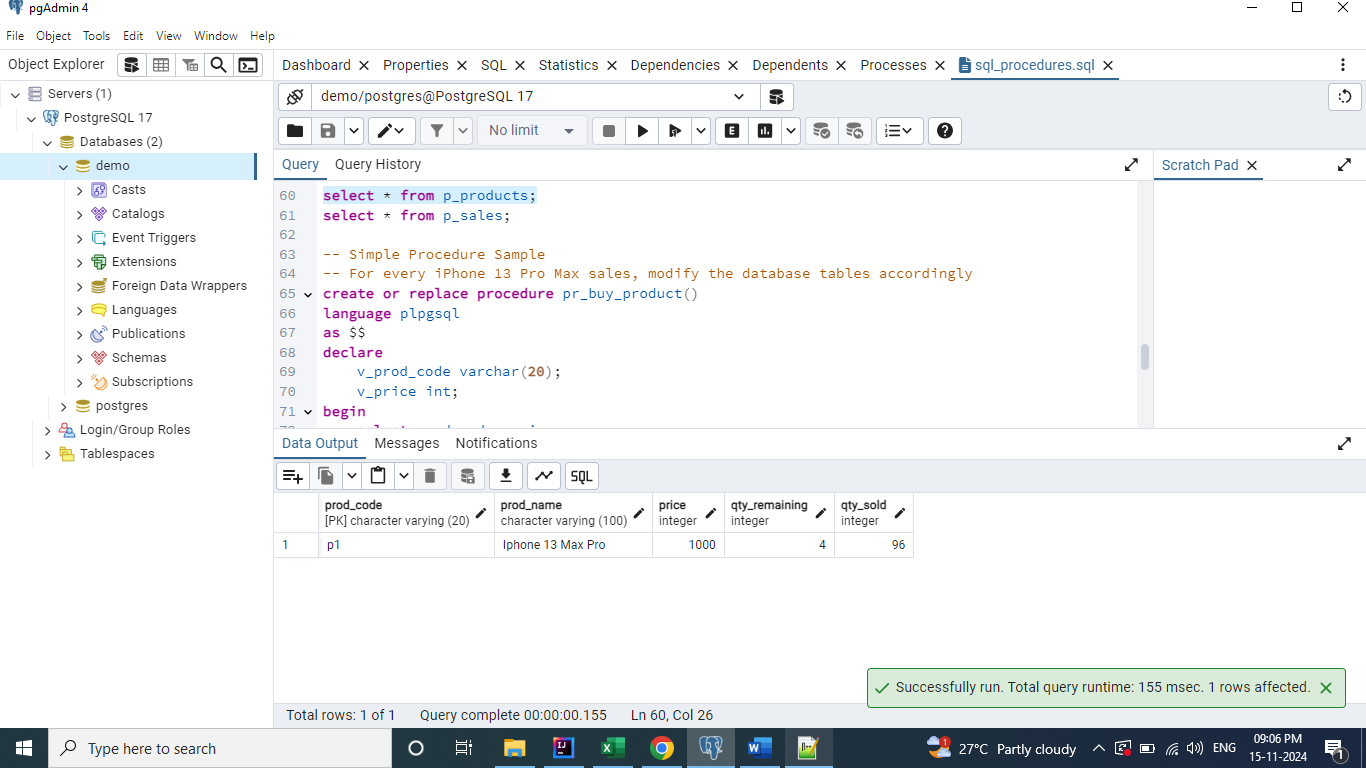
**$$**

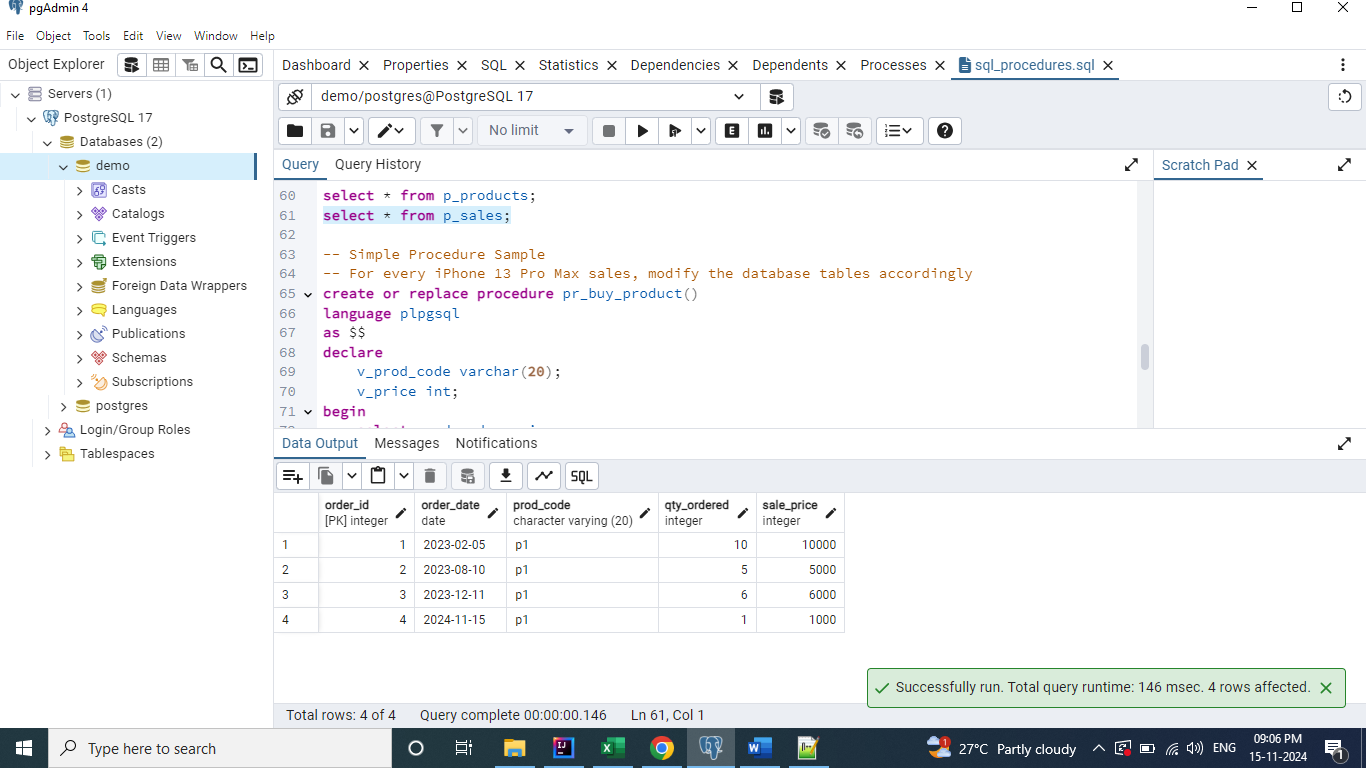


**call pr\_buy\_product();**



**select \* from p\_products;**



**select \* from p\_sales;**

## Same Procedure in Oracle DB format

**create or replace procedure pr\_buy\_product**

**as**

**v\_prod\_code varchar(20);**

**v\_price int;**

**begin**

**select prod\_code, price**

**into v\_prod\_code, v\_price**

**from p\_products**

**where prod\_name = 'Iphone 13 Max Pro';**

**insert into p\_sales(order\_date, prod\_code, qty\_ordered, sale\_price)**

**values (current\_date, v\_prod\_code, 1, (v\_price \* 1));**

**update p\_products**

**set qty\_remaining = (qty\_remaining - 1),**

**qty\_sold = (qty\_sold + 1)**

**where prod\_code = v\_prod\_code;**

**dbms\_output.put\_line ('Product Sold!');**

**end;**

**To Call Procedure in Oracle**

**exec pr\_buy\_products;**

# Note

By default, if we don't mention the type of input, then it will be treated as IN parameter. If it’s OUT parameter, then it has to be mentioned explicitly.

# Procedures with Parameters

## For every given product and the quantity,

## 1) Check if product is available based on the required quantity.

## 2) If available then modify the database tables accordingly.

**create or replace procedure pr\_buy\_products(p\_prod\_name varchar, p\_quantity int)**

**language plpgsql**

**as $$**

**declare**

**v\_prod\_code varchar(20);**

**v\_price int;**

**v\_count int;**

**begin**

**select count(1)**

**into v\_count**

**from p\_products**

**where prod\_name = p\_prod\_name**

**and qty\_remaining >= p\_quantity;**

**if v\_count > 0 then**

**select prod\_code, price**

**into v\_prod\_code, v\_price**

**from p\_products**

**where prod\_name = p\_prod\_name;**

**insert into p\_sales(order\_date, prod\_code, qty\_ordered, sale\_price)**

**values (current\_date, v\_prod\_code, p\_quantity, (v\_price \* p\_quantity));**

**update p\_products**

**set qty\_remaining = (qty\_remaining - p\_quantity),**

**qty\_sold = (qty\_sold + p\_quantity)**

**where prod\_code = v\_prod\_code;**

**raise notice 'Product Sold!';**

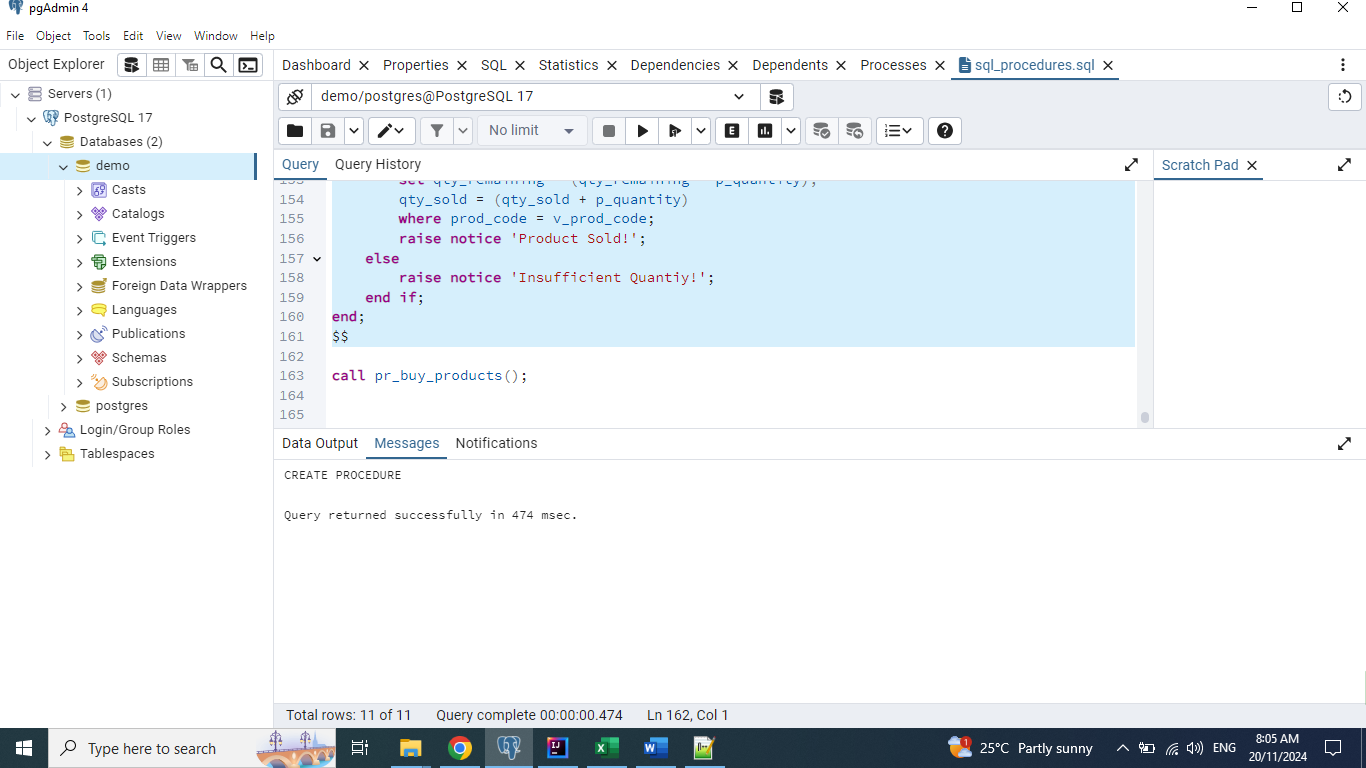
**else**

**raise notice 'Insufficient Quantity!';**

**end if;**

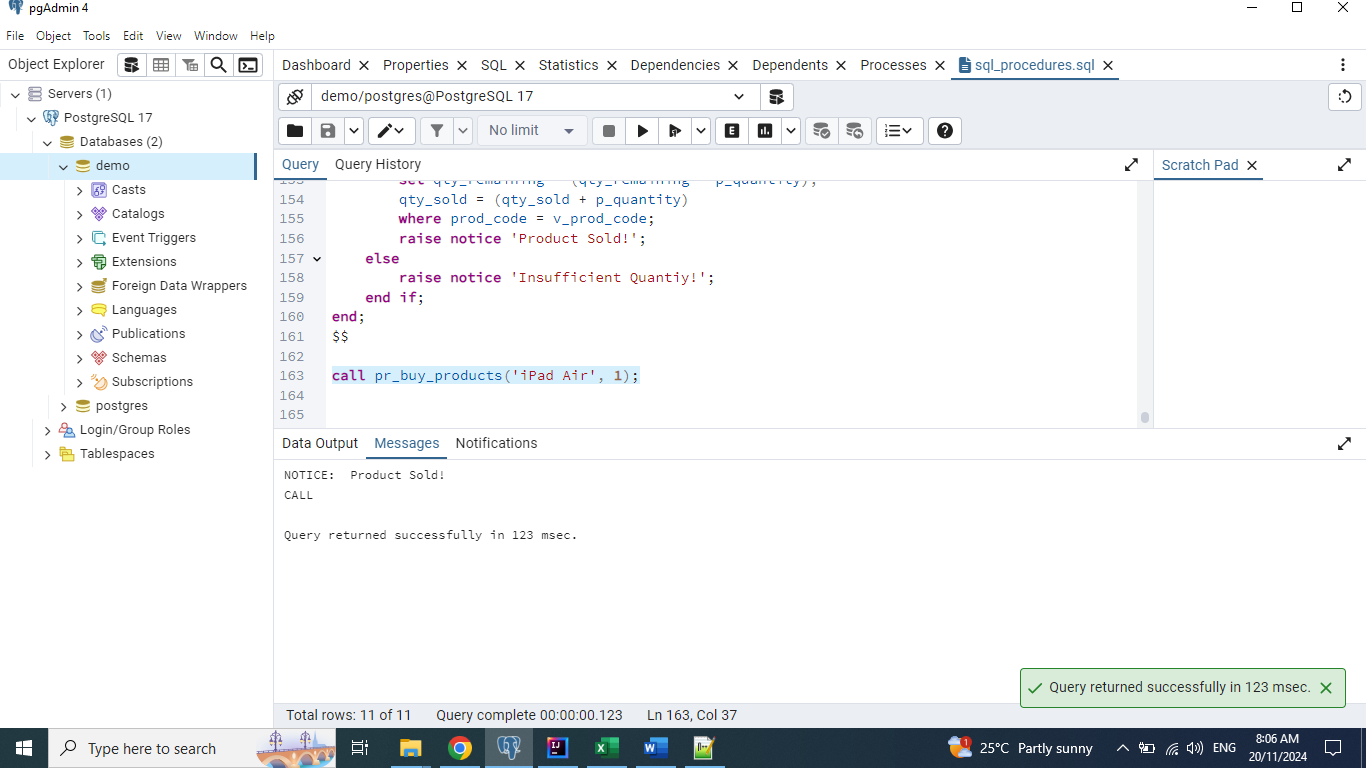
**end;**

**$$**



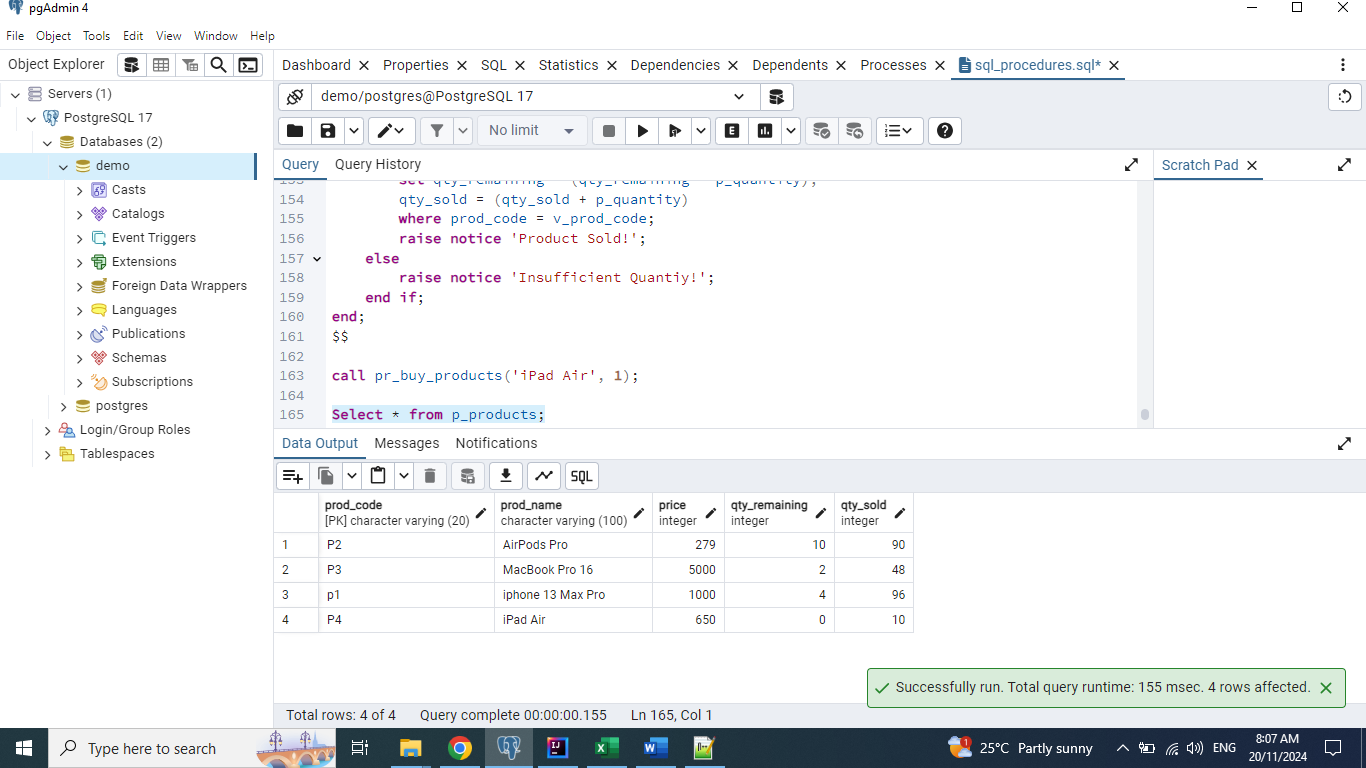
### If Scenario

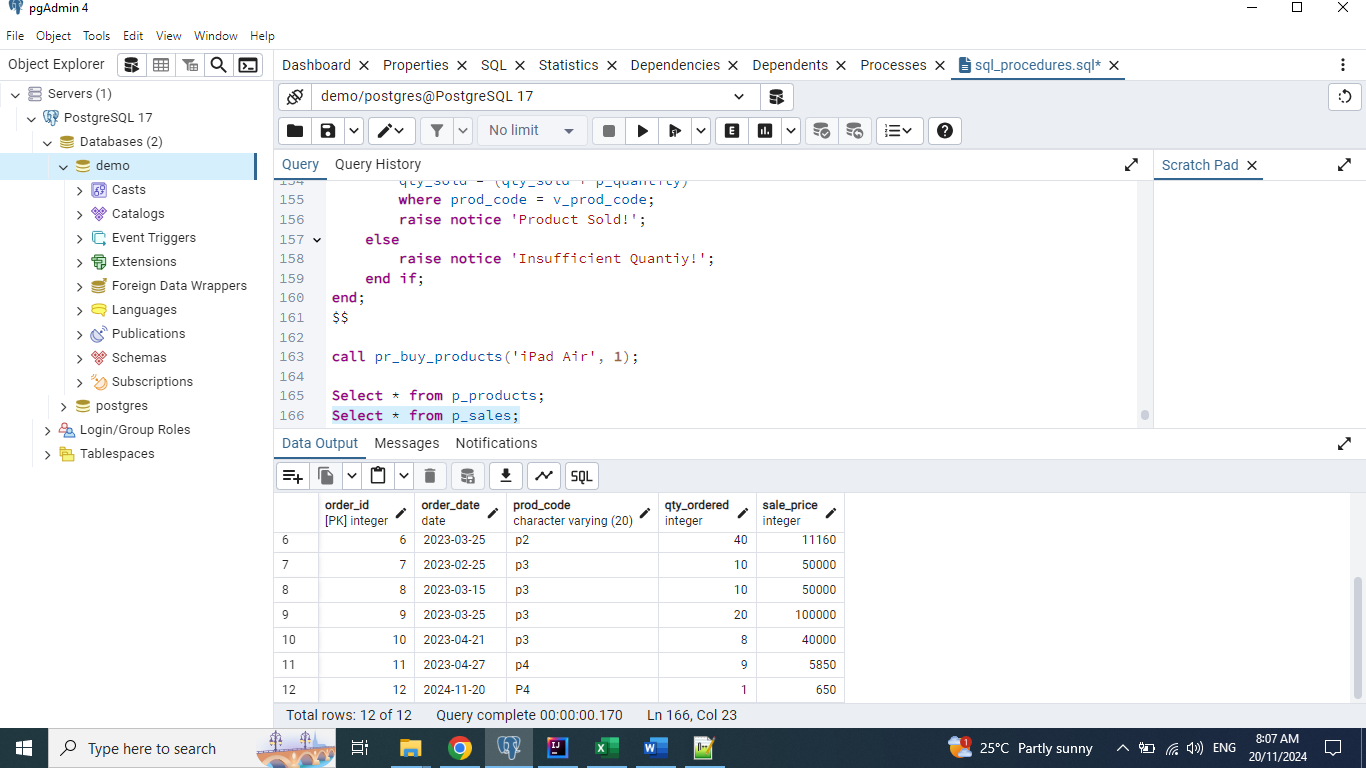
**call pr\_buy\_products('iPad Air', 1);**



**Select \* from p\_products;**

**Select \* from p\_sales;**





### Else Scenario

**call pr\_buy\_products('iPad Air', 1);**

