Name: Prerana Shukla

Roll. No.: 31

Title: Inventory Store management System

Sem.: 5th Branch: ICT

Subject: Database Management System

Subject Code: 01CT0502

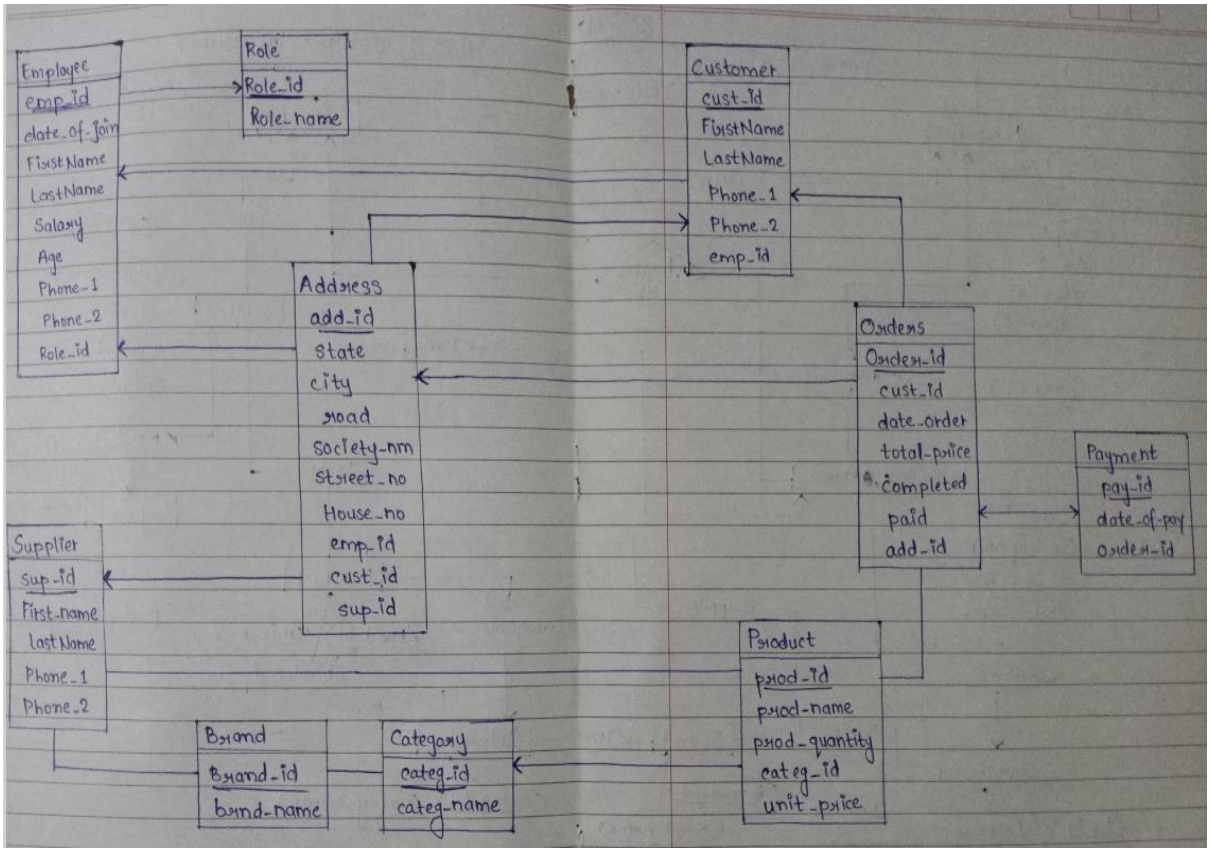
➔ Abstract:

I have the project title as Inventory Store Management System. For the project I am going to build up a database creating table to store data created keeping in mind the need for inventory store to keep record and to retrieve it easily on time. I will use PostgreSQL pgAdmin4 for making and managing the database. Database Management System is a software that is designed to define, manipulate, retrieve and manage data in database. Database helps one to store all the logically related data and by using database management system it's easy for one for updating the database as and when needed, not only this it also gives flexibility to retrieve data if required. DBMS can be said as computerized record keeping system with the advancement it also provides a wide range to store data on cloud. I will use DBMS for the project Inventory Store Management System. While going through this I will follow the basic steps required for making the database in structured manner. Making schema diagram, implementing same with ER diagram for more clarity, schema designing which helps to have a glance for writing the query for er diagram and looking for cardinality constraints. Advantages of making a DBMS is that it reduces data redundancy so that duplication of a record can be neglected. As DBMS have atomicity so there is no ratio answer which provides better understanding of it, like if there is a customer asking for any product then it can be viewed that how much is the availability of that particular product there can be no half answer like. While not only this it allows applying integrity constraints, sharing of data and authorization of user such as the employee table can be seen by owner but not by customer.

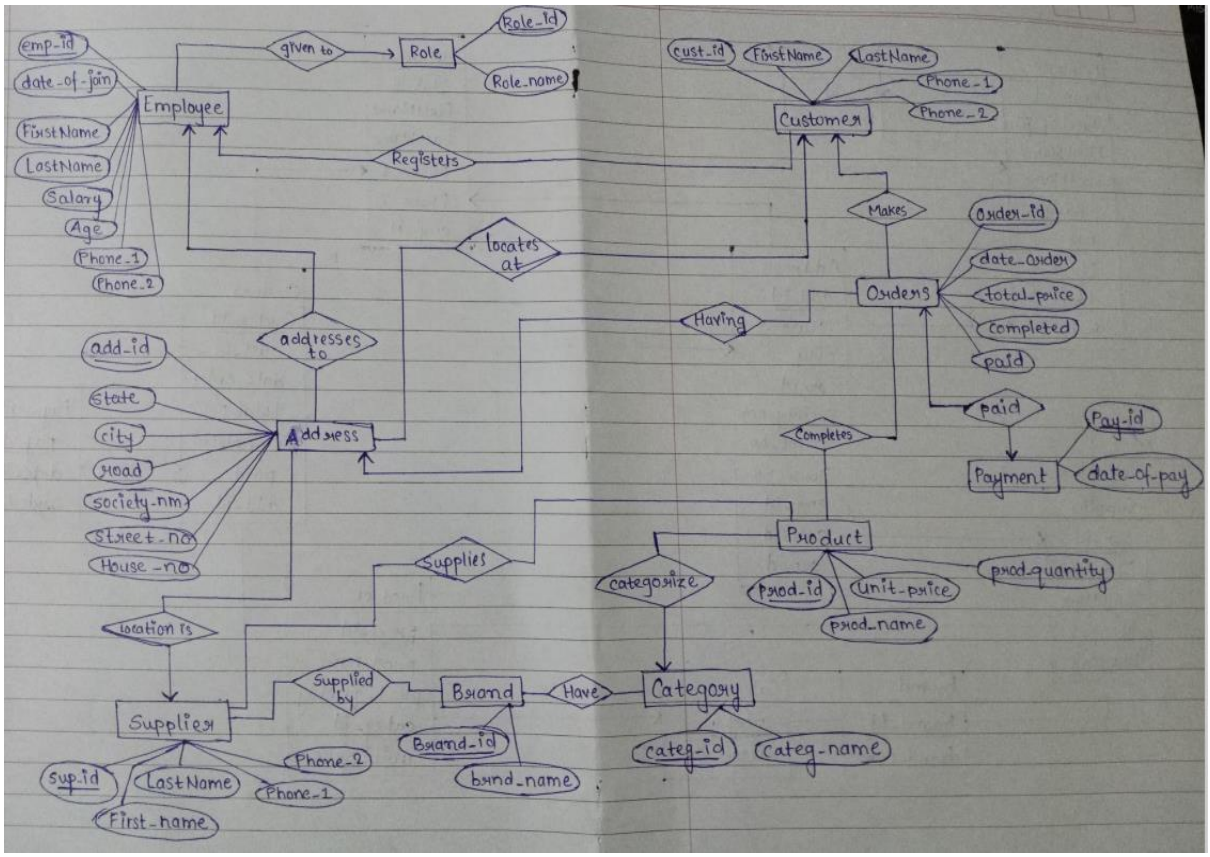
➔ Schema Diagram:

Before going through the schema diagram lets define the basic concept for the same. Database schema is a skeleton structure which represents logical view of the entire database. Database schema defines entities and relationship among them which contains descriptive detail of the database that can depicted through schema diagram. Schema is designed by database designers.

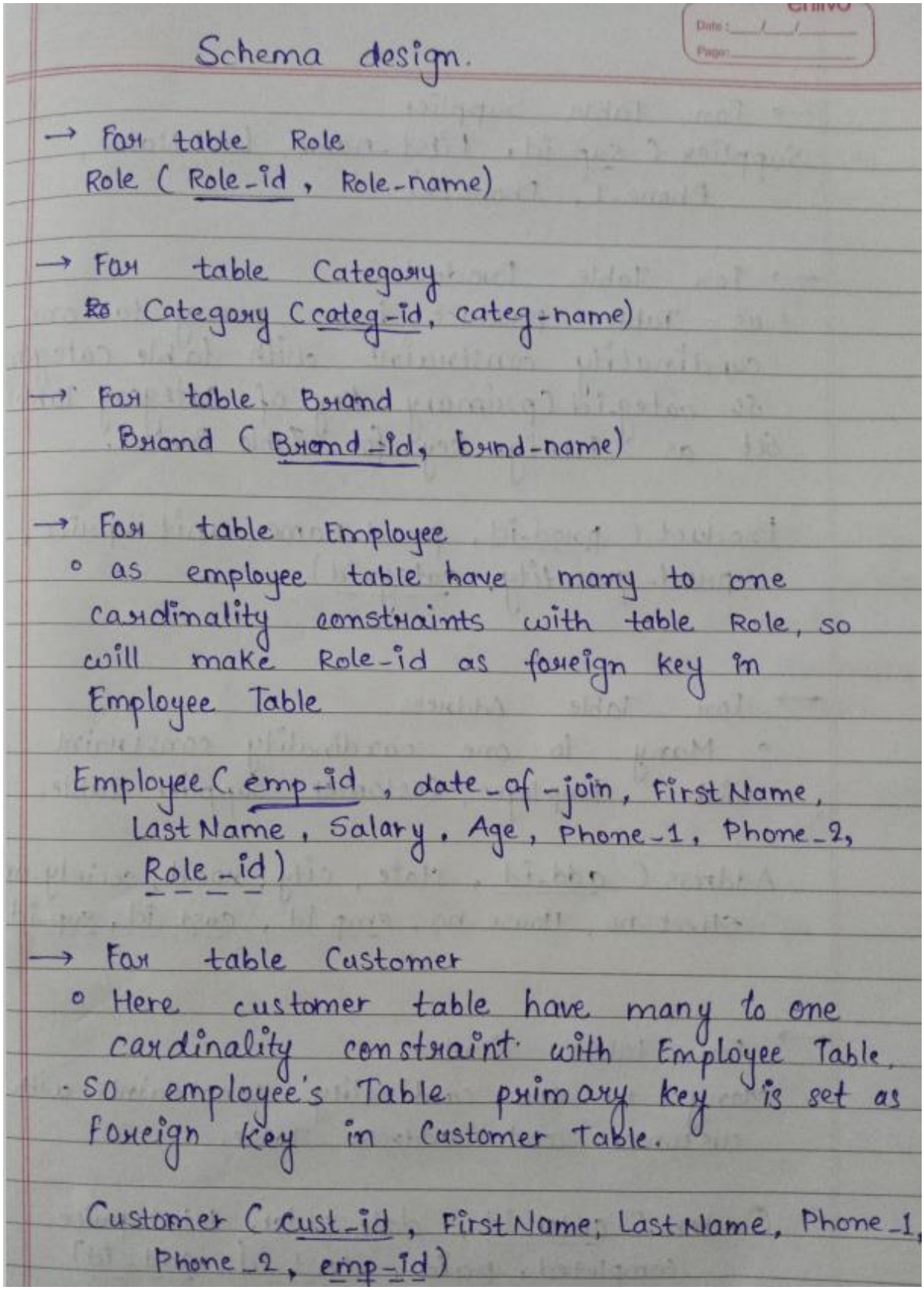
Schema diagram is a diagram which contains entities and attributes that defines the schema. Need to notice that schema diagram only shows database design not the actual data of database.


Employee
emp_id
date_of_join
FirstName
LastName
Salary
Age
Phone_1
Phone_2
Role_id
Role
Role_id
Role_name
Customer
cust_id
FirstName
LastName
Phone_1
Phone_2
emp_id
Address
add_id
State
city
road
society_nm
street_no
House_no
emp_id
cust_id
sup_id
Orders
Order_id
cust_id
date_order
total_price
completed
paid
add_id
Payment
pay_id
date_of_pay
Order_id
Supplier
sup_id
First_name
last Name
Phone_1
Phone_2
Product
prod_id
prod_name
prod_quantity
categ_id
unit_price
Brand
Brand_id
brnd_name
Category
categ_id
categ_name

➔ E-R Diagram:


emp_id
date_of_join
FirstName
LastName
Salary
Age
Phone_1
Phone_2
Employee
given to
Role
Role_id
Role_name
Registers
cust_id
FirstName
LastName
Phone_1
Phone_2
Customer
Makes
Order_id
date_order
total_price
completed
paid
Orders
Having
locates at
addresses to
add_id
State
city
road
society_nm
street_no
House_no
Address
Completes
paid
Pay_id
Payment
date_of_pay
Product
Supplies
categorize
prod_id
unit_price
prod_quantity
prod_name
location is
Supplied by
Brand
Have
Category
Brand_id
brnd_name
categ_id
categ_name
Supplier
sup_id
LastName
Phone_2
Phone_1
First_name

➔ **Schema Design:**



Schema design.

→ For table Role
Role (Role_id , Role_name)

→ For table Category
Category (categ_id, categ_name)

→ For table Brand
Brand (Brand_id, brnd_name)

→ For table Employee
- as employee table have many to one cardinality constraints with table Role, so will make Role_id as foreign key in Employee Table

Employee (emp_id , date_of_join, First Name, Last Name , Salary , Age , Phone_1 , Phone_2, Role_id)

→ For table Customer
- Here customer table have many to one cardinality constraint with Employee Table. So employee's Table primary key is set as Foreign key in Customer Table.

Customer (cust_id , First Name; Last Name, Phone_1, Phone_2, emp_id)

→ For Table Supplier

Supplier (sup_id, First_name, Last Name, Phone_1, Phone_2)

→ For Table Product

- as Table product have many to one cardinality constraint with table category. so categ_id (primary key of category Table) is set as Foreign key for Table Product.

Product (prod_id, prod_name, unit_price, prod_quantity, categ_id)

→ For Table Address

- Many to one cardinality constraint with Employee, Customer, Supplier Table.

Address (add_id, state, city, road, society_nm, Street_no, House_no, emp_id, cust_id, sup_id)

→ For table Orders

Many to one cardinality constraints with customer and address Table.

Orders (order_id, date_order, total_price, completed, paid, cust_id, add_id)

→ For table ~~Paid~~ Payment as payment and orders table have one to one cardinality constraint, I have chosen to make order-id (primary key of Table Orders) to be foreign key for Table Payment, it can also be choosen vice versa.

Payment (Pay-id , date-of-pay, order_id)

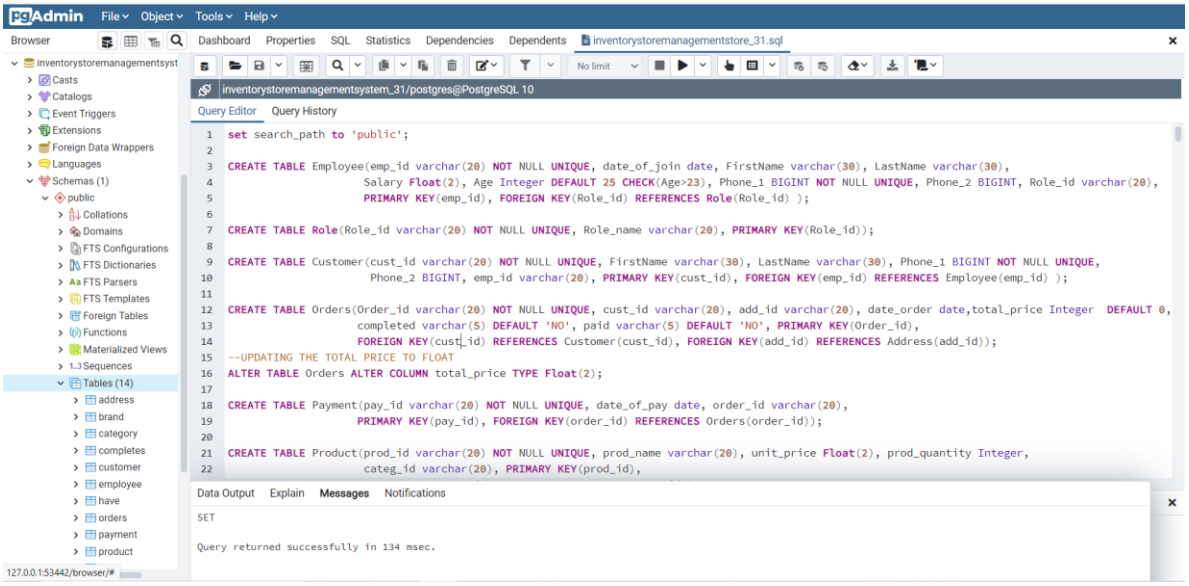
→ The Relationship between Supplier and Brand is many to many cardinality constraint, so need to make (take relationship as new Table make a unique id and take other two connected table's primary key as Foreign key for this new Table made.

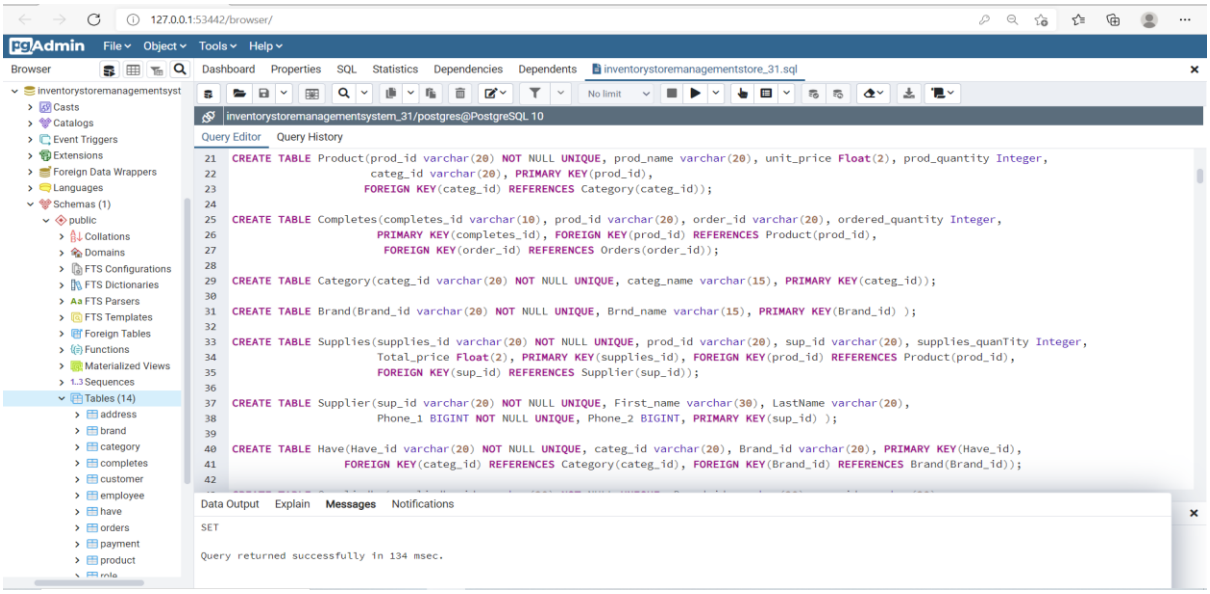
Suppliedby (suppliedby_id , brand_id, sup_id)

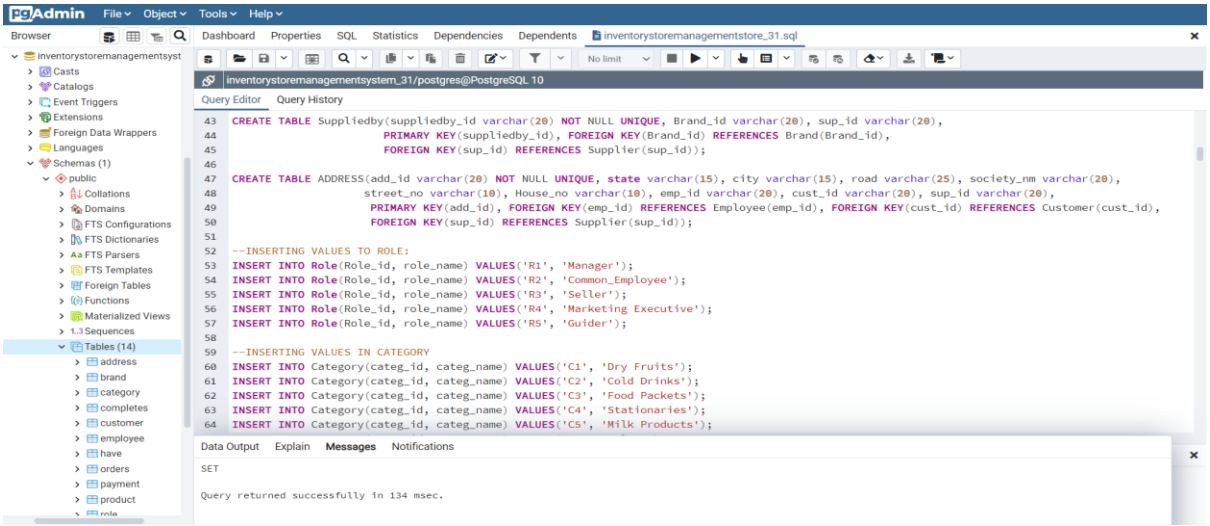
→ Same way for other N:N Relationships

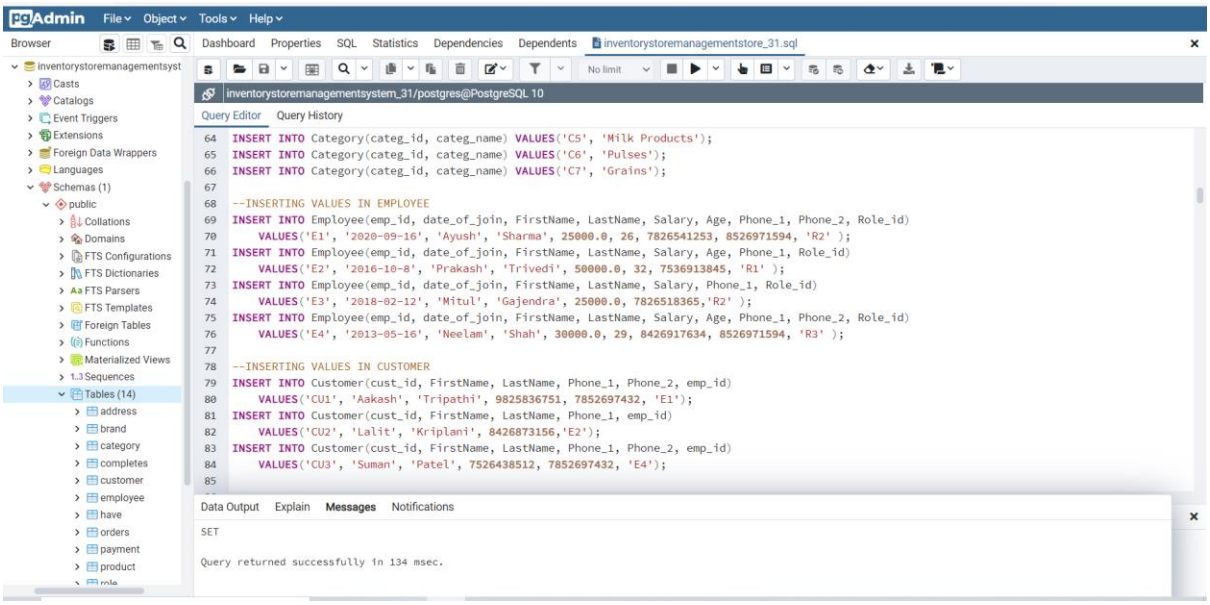
- Supplies (supplies_id, prod_id , sup_id , supplies - quantity , total - price)
- Have (Have_id, categ_id , brand_id)
- Completes (completes_id , prod_id , order_id , ordered - quantity)

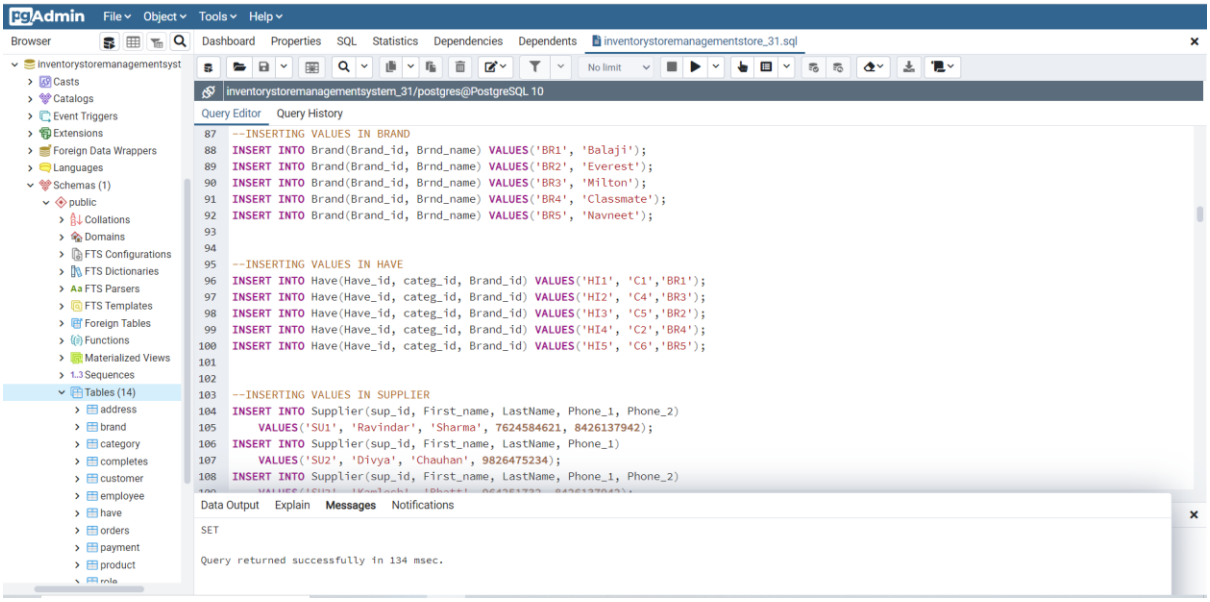
➔ Snapshots:

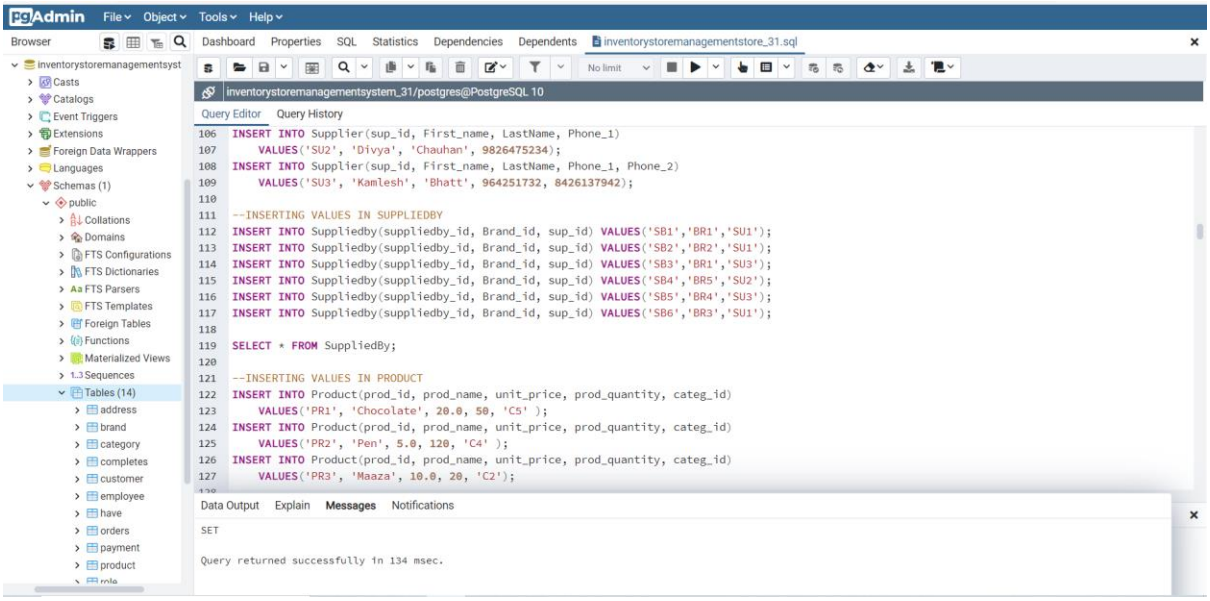

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
set search_path to 'public';
CREATE TABLE Employee(emp_id varchar(20) NOT NULL UNIQUE, date_of_join date, FirstName varchar(30), LastName varchar(30),
Salary Float(2), Age Integer DEFAULT 25 CHECK(Age>23), Phone_1 BIGINT NOT NULL UNIQUE, Phone_2 BIGINT, Role_id varchar(20),
PRIMARY KEY(emp_id), FOREIGN KEY(Role_id) REFERENCES Role(Role_id));
CREATE TABLE Role(Role_id varchar(20) NOT NULL UNIQUE, Role_name varchar(20), PRIMARY KEY(Role_id));
CREATE TABLE Customer(cust_id varchar(20) NOT NULL UNIQUE, FirstName varchar(30), LastName varchar(30), Phone_1 BIGINT NOT NULL UNIQUE,
Phone_2 BIGINT, emp_id varchar(20), PRIMARY KEY(cust_id), FOREIGN KEY(emp_id) REFERENCES Employee(emp_id));
CREATE TABLE Orders(Order_id varchar(20) NOT NULL UNIQUE, cust_id varchar(20), add_id varchar(20), date_order date,total_price Integer DEFAULT 0,
completed varchar(5) DEFAULT 'NO', paid varchar(5) DEFAULT 'NO', PRIMARY KEY(Order_id),
FOREIGN KEY(cust_id) REFERENCES Customer(cust_id), FOREIGN KEY(add_id) REFERENCES Address(add_id));
--UPDATING THE TOTAL PRICE TO FLOAT
ALTER TABLE Orders ALTER COLUMN total_price TYPE Float(2);
CREATE TABLE Payment(pay_id varchar(20) NOT NULL UNIQUE, date_of_pay date, order_id varchar(20),
PRIMARY KEY(pay_id), FOREIGN KEY(order_id) REFERENCES Orders(order_id));
CREATE TABLE Product(prod_id varchar(20) NOT NULL UNIQUE, prod_name varchar(20), unit_price Float(2), prod_quantity Integer,
categ_id varchar(20), PRIMARY KEY(prod_id),
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

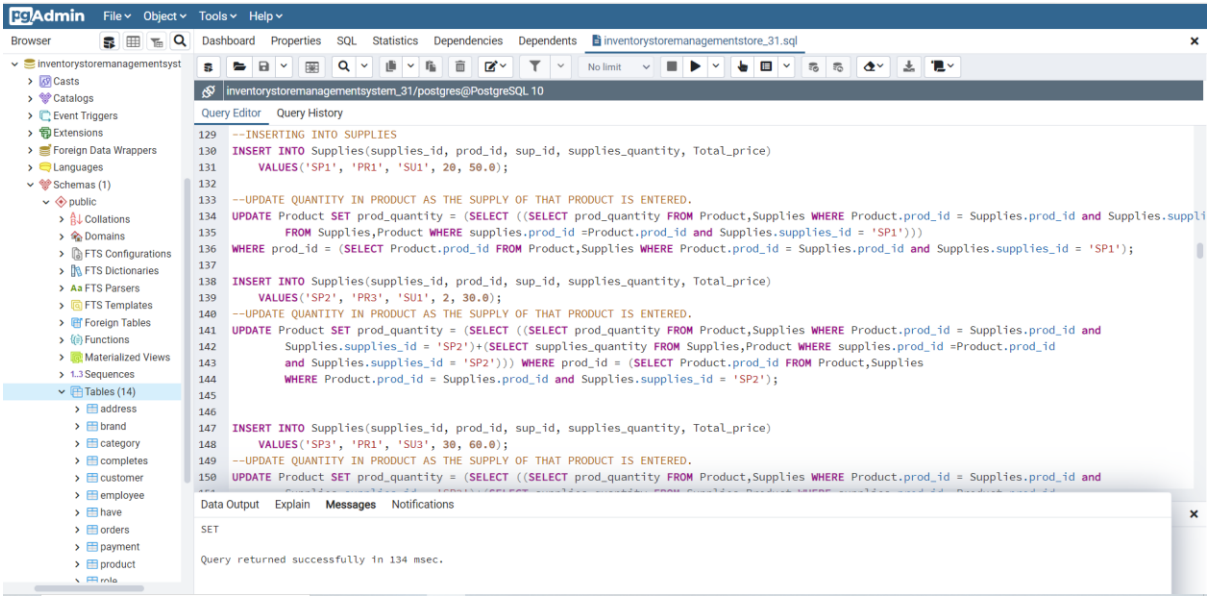

127.0.0.1:53442/browser/
pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
CREATE TABLE Product(prod_id varchar(20) NOT NULL UNIQUE, prod_name varchar(20), unit_price Float(2), prod_quantity Integer,
categ_id varchar(20), PRIMARY KEY(prod_id),
FOREIGN KEY(categ_id) REFERENCES Category(categ_id));
CREATE TABLE Completes(completes_id varchar(10), prod_id varchar(20), order_id varchar(20), ordered_quantity Integer,
PRIMARY KEY(completes_id), FOREIGN KEY(prod_id) REFERENCES Product(prod_id),
FOREIGN KEY(order_id) REFERENCES Orders(order_id));
CREATE TABLE Category(categ_id varchar(20) NOT NULL UNIQUE, categ_name varchar(15), PRIMARY KEY(categ_id));
CREATE TABLE Brand(Brand_id varchar(20) NOT NULL UNIQUE, Brnd_name varchar(15), PRIMARY KEY(Brand_id));
CREATE TABLE Supplies(supplies_id varchar(20) NOT NULL UNIQUE, prod_id varchar(20), sup_id varchar(20), supplies_quanTity Integer,
Total_price Float(2), PRIMARY KEY(supplies_id), FOREIGN KEY(prod_id) REFERENCES Product(prod_id),
FOREIGN KEY(sup_id) REFERENCES Supplier(sup_id));
CREATE TABLE Supplier(sup_id varchar(20) NOT NULL UNIQUE, First_name varchar(30), LastName varchar(20),
Phone_1 BIGINT NOT NULL UNIQUE, Phone_2 BIGINT, PRIMARY KEY(sup_id));
CREATE TABLE Have(Have_id varchar(20) NOT NULL UNIQUE, categ_id varchar(20), Brand_id varchar(20), PRIMARY KEY(Have_id),
FOREIGN KEY(categ_id) REFERENCES Category(categ_id), FOREIGN KEY(Brand_id) REFERENCES Brand(Brand_id));
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

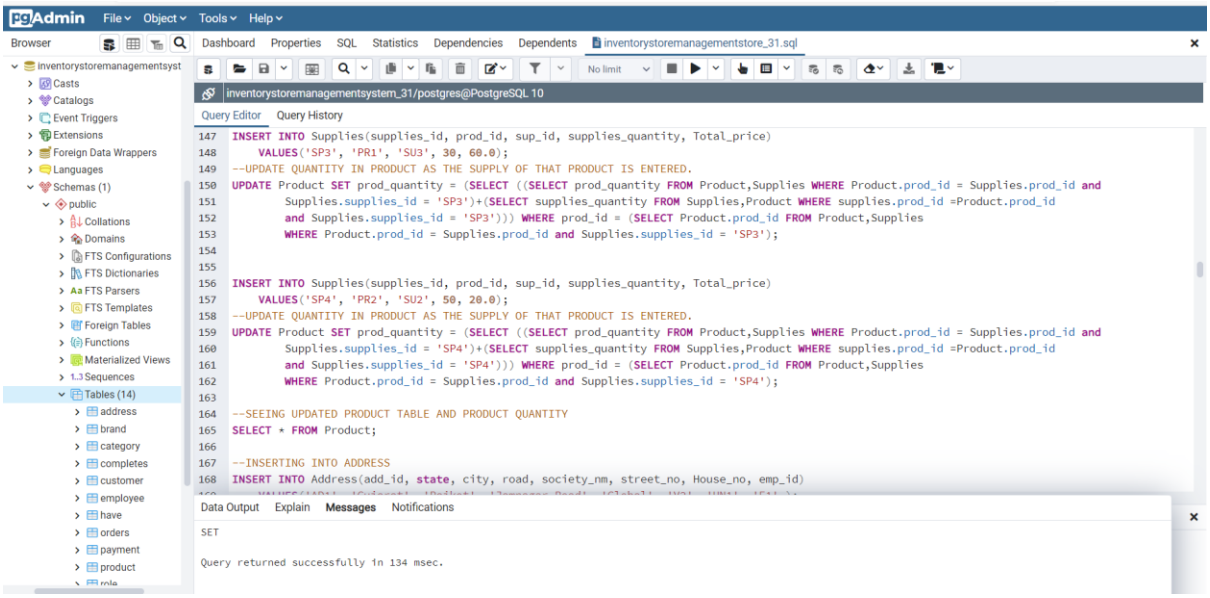

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
CREATE TABLE Suppliedby(suppliedby_id varchar(20) NOT NULL UNIQUE, Brand_id varchar(20), sup_id varchar(20),
PRIMARY KEY(suppliedby_id), FOREIGN KEY(Brand_id) REFERENCES Brand(Brand_id),
FOREIGN KEY(sup_id) REFERENCES Supplier(sup_id));
CREATE TABLE ADDRESS(add_id varchar(20) NOT NULL UNIQUE, state varchar(15), city varchar(15), road varchar(25), society_nm varchar(20),
street_no varchar(10), House_no varchar(10), emp_id varchar(20), cust_id varchar(20), sup_id varchar(20),
PRIMARY KEY(add_id), FOREIGN KEY(emp_id) REFERENCES Employee(emp_id), FOREIGN KEY(cust_id) REFERENCES Customer(cust_id),
FOREIGN KEY(sup_id) REFERENCES Supplier(sup_id));
--INSERTING VALUES TO ROLE:
INSERT INTO Role(Role_id, role_name) VALUES('R1', 'Manager');
INSERT INTO Role(Role_id, role_name) VALUES('R2', 'Common_Employee');
INSERT INTO Role(Role_id, role_name) VALUES('R3', 'Seller');
INSERT INTO Role(Role_id, role_name) VALUES('R4', 'Marketing Executive');
INSERT INTO Role(Role_id, role_name) VALUES('R5', 'Guider');
--INSERTING VALUES IN CATEGORY
INSERT INTO Category(categ_id, categ_name) VALUES('C1', 'Dry Fruits');
INSERT INTO Category(categ_id, categ_name) VALUES('C2', 'Cold Drinks');
INSERT INTO Category(categ_id, categ_name) VALUES('C3', 'Food Packets');
INSERT INTO Category(categ_id, categ_name) VALUES('C4', 'Stationaries');
INSERT INTO Category(categ_id, categ_name) VALUES('C5', 'Milk Products');
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

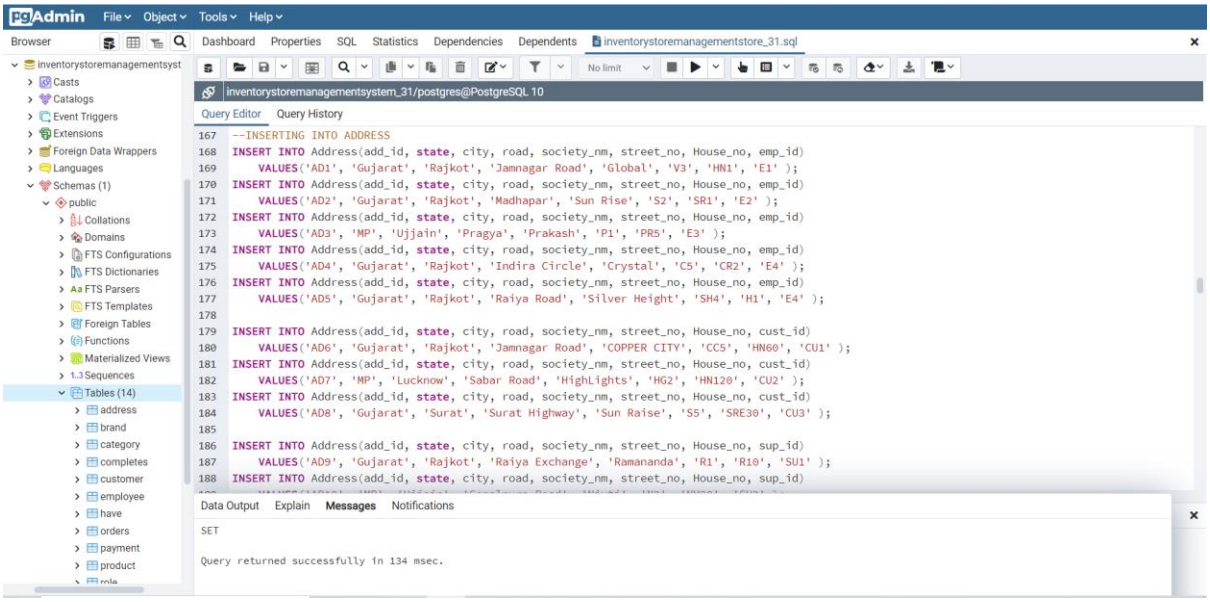

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
64 INSERT INTO Category(categ_id, categ_name) VALUES('C5', 'Milk Products');
65 INSERT INTO Category(categ_id, categ_name) VALUES('C6', 'Pulses');
66 INSERT INTO Category(categ_id, categ_name) VALUES('C7', 'Grains');
67
68 --INSERTING VALUES IN EMPLOYEE
69 INSERT INTO Employee(emp_id, date_of_join, FirstName, LastName, Salary, Age, Phone_1, Phone_2, Role_id)
70 VALUES('E1', '2020-09-16', 'Ayush', 'Sharma', 25000.0, 26, 7826541253, 8526971594, 'R2');
71 INSERT INTO Employee(emp_id, date_of_join, FirstName, LastName, Salary, Age, Phone_1, Role_id)
72 VALUES('E2', '2016-10-8', 'Prakash', 'Trivedi', 50000.0, 32, 7536913845, 'R1');
73 INSERT INTO Employee(emp_id, date_of_join, FirstName, LastName, Salary, Phone_1, Role_id)
74 VALUES('E3', '2018-02-12', 'Mitul', 'Gajendra', 25000.0, 7826518365,'R2');
75 INSERT INTO Employee(emp_id, date_of_join, FirstName, LastName, Salary, Age, Phone_1, Phone_2, Role_id)
76 VALUES('E4', '2013-05-16', 'Neelam', 'Shah', 30000.0, 29, 8426917634, 8526971594, 'R3');
77
78 --INSERTING VALUES IN CUSTOMER
79 INSERT INTO Customer(cust_id, FirstName, LastName, Phone_1, Phone_2, emp_id)
80 VALUES('CU1', 'Aakash', 'Tripathi', 9825836751, 7852697432, 'E1');
81 INSERT INTO Customer(cust_id, FirstName, LastName, Phone_1, emp_id)
82 VALUES('CU2', 'Lalit', 'Kriplani', 8426873156,'E2');
83 INSERT INTO Customer(cust_id, FirstName, LastName, Phone_1, Phone_2, emp_id)
84 VALUES('CU3', 'Suman', 'Patel', 7526438512, 7852697432, 'E4');
85
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

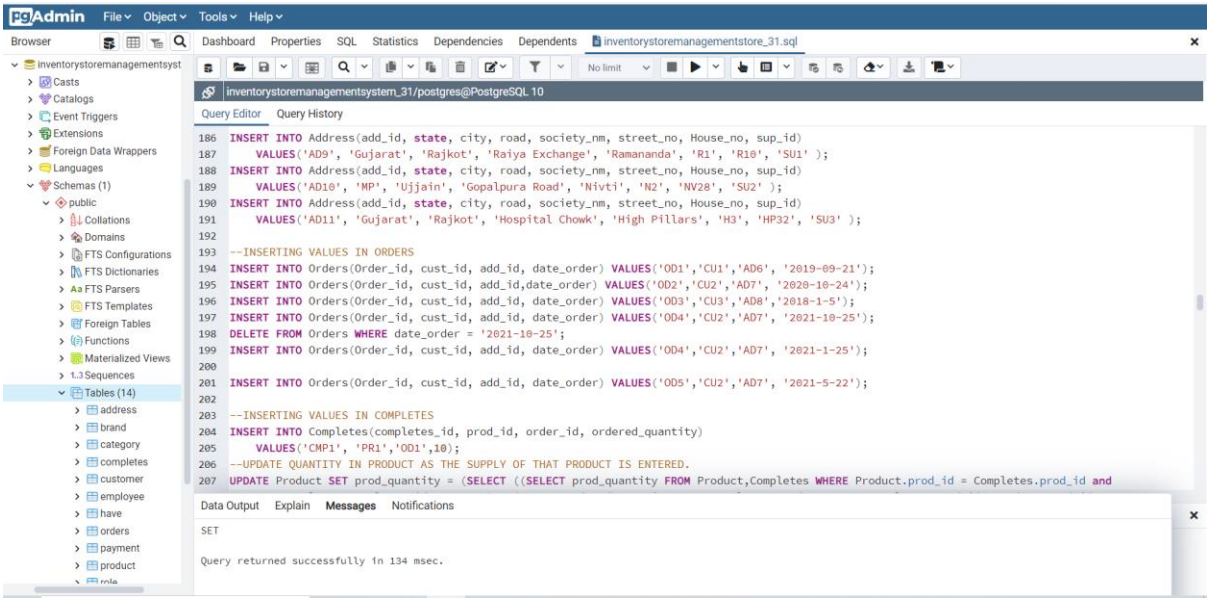

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
87 --INSERTING VALUES IN BRAND
88 INSERT INTO Brand(Brand_id, Brnd_name) VALUES('BR1', 'Balaji');
89 INSERT INTO Brand(Brand_id, Brnd_name) VALUES('BR2', 'Everest');
90 INSERT INTO Brand(Brand_id, Brnd_name) VALUES('BR3', 'Milton');
91 INSERT INTO Brand(Brand_id, Brnd_name) VALUES('BR4', 'Classmate');
92 INSERT INTO Brand(Brand_id, Brnd_name) VALUES('BR5', 'Navneet');
93
94
95 --INSERTING VALUES IN HAVE
96 INSERT INTO Have(Have_id, categ_id, Brand_id) VALUES('HI1', 'C1','BR1');
97 INSERT INTO Have(Have_id, categ_id, Brand_id) VALUES('HI2', 'C4','BR3');
98 INSERT INTO Have(Have_id, categ_id, Brand_id) VALUES('HI3', 'C5','BR2');
99 INSERT INTO Have(Have_id, categ_id, Brand_id) VALUES('HI4', 'C2','BR4');
100 INSERT INTO Have(Have_id, categ_id, Brand_id) VALUES('HI5', 'C6','BR5');
101
102
103 --INSERTING VALUES IN SUPPLIER
104 INSERT INTO Supplier(sup_id, First_name, LastName, Phone_1, Phone_2)
105 VALUES('SU1', 'Ravindar', 'Sharma', 7624584621, 8426137942);
106 INSERT INTO Supplier(sup_id, First_name, LastName, Phone_1)
107 VALUES('SU2', 'Divya', 'Chauhan', 9826475234);
108 INSERT INTO Supplier(sup_id, First_name, LastName, Phone_1, Phone_2)
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

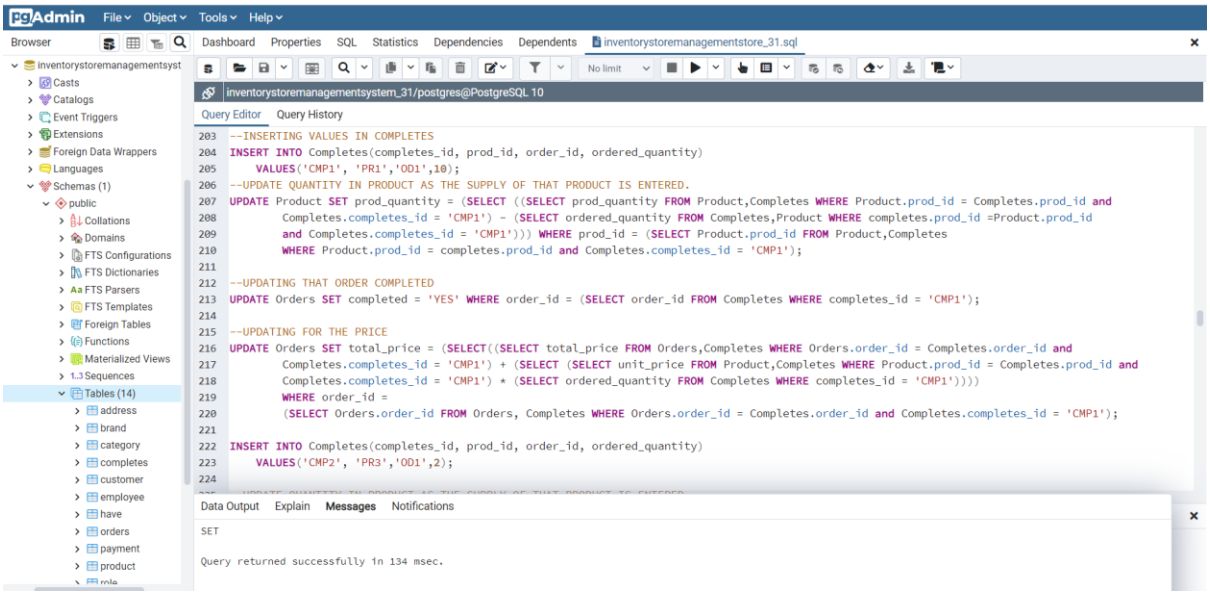

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
106 INSERT INTO Supplier(sup_id, First_name, LastName, Phone_1)
107 VALUES('SU2', 'Divya', 'Chauhan', 9826475234);
108 INSERT INTO Supplier(sup_id, First_name, LastName, Phone_1, Phone_2)
109 VALUES('SU3', 'Kamlesh', 'Bhatt', 964251732, 8426137942);
110
111 --INSERTING VALUES IN SUPPLIEDBY
112 INSERT INTO Suppliedby(suppliedby_id, Brand_id, sup_id) VALUES('SB1','BR1','SU1');
113 INSERT INTO Suppliedby(suppliedby_id, Brand_id, sup_id) VALUES('SB2','BR2','SU1');
114 INSERT INTO Suppliedby(suppliedby_id, Brand_id, sup_id) VALUES('SB3','BR1','SU3');
115 INSERT INTO Suppliedby(suppliedby_id, Brand_id, sup_id) VALUES('SB4','BR5','SU2');
116 INSERT INTO Suppliedby(suppliedby_id, Brand_id, sup_id) VALUES('SB5','BR4','SU3');
117 INSERT INTO Suppliedby(suppliedby_id, Brand_id, sup_id) VALUES('SB6','BR3','SU1');
118
119 SELECT * FROM SuppliedBy;
120
121 --INSERTING VALUES IN PRODUCT
122 INSERT INTO Product(prod_id, prod_name, unit_price, prod_quantity, categ_id)
123 VALUES('PR1', 'Chocolate', 20.0, 50, 'C5');
124 INSERT INTO Product(prod_id, prod_name, unit_price, prod_quantity, categ_id)
125 VALUES('PR2', 'Pen', 5.0, 120, 'C4');
126 INSERT INTO Product(prod_id, prod_name, unit_price, prod_quantity, categ_id)
127 VALUES('PR3', 'Maaza', 10.0, 20, 'C2');
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

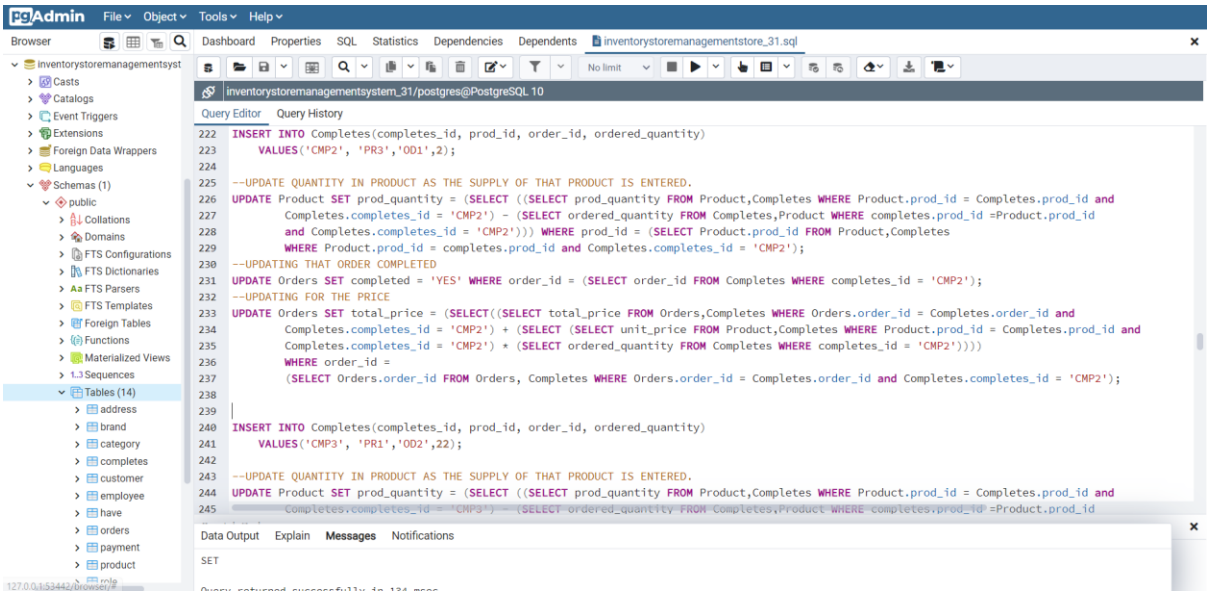

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--INSERTING INTO SUPPLIES
INSERT INTO Supplies(supplies_id, prod_id, sup_id, supplies_quantity, Total_price)
VALUES('SP1', 'PR1', 'SU1', 20, 50.0);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Supplies WHERE Product.prod_id = Supplies.prod_id and Supplies.suppli
FROM Supplies,Product WHERE supplies.prod_id =Product.prod_id and Supplies.supplies_id = 'SP1')))
WHERE prod_id = (SELECT Product.prod_id FROM Product,Supplies WHERE Product.prod_id = Supplies.prod_id and Supplies.supplies_id = 'SP1');
INSERT INTO Supplies(supplies_id, prod_id, sup_id, supplies_quantity, Total_price)
VALUES('SP2', 'PR3', 'SU1', 2, 30.0);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Supplies WHERE Product.prod_id = Supplies.prod_id and
Supplies.supplies_id = 'SP2')+(SELECT supplies_quantity FROM Supplies,Product WHERE supplies.prod_id =Product.prod_id
and Supplies.supplies_id = 'SP2'))) WHERE prod_id = (SELECT Product.prod_id FROM Product,Supplies
WHERE Product.prod_id = Supplies.prod_id and Supplies.supplies_id = 'SP2');
INSERT INTO Supplies(supplies_id, prod_id, sup_id, supplies_quantity, Total_price)
VALUES('SP3', 'PR1', 'SU3', 30, 60.0);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Supplies WHERE Product.prod_id = Supplies.prod_id and
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

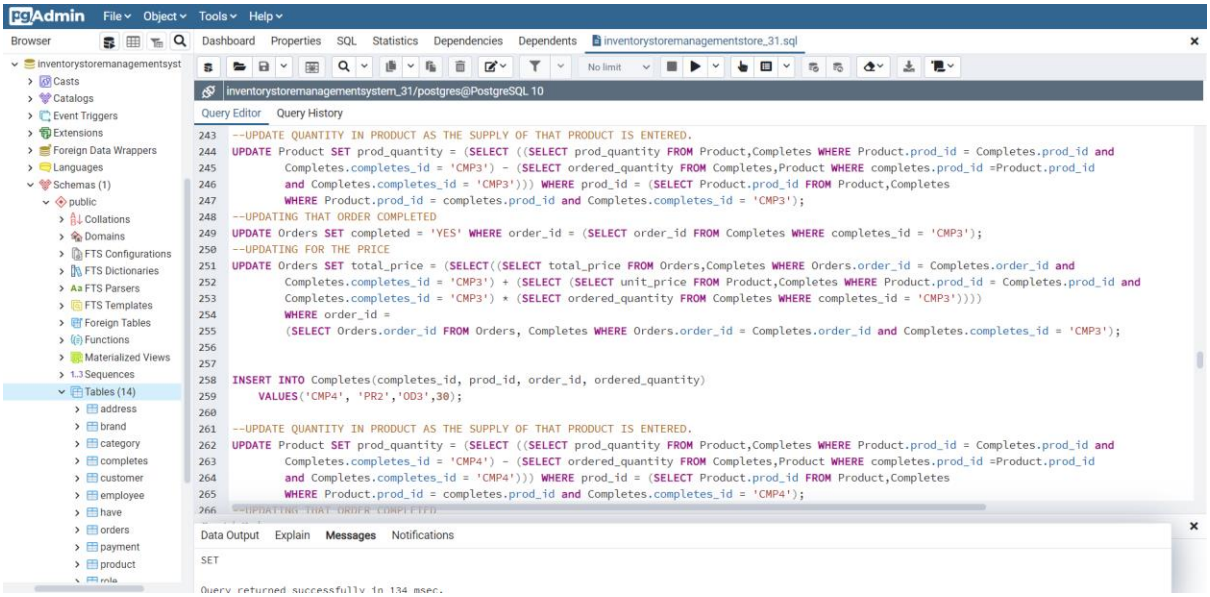

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
INSERT INTO Supplies(supplies_id, prod_id, sup_id, supplies_quantity, Total_price)
VALUES('SP3', 'PR1', 'SU3', 30, 60.0);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Supplies WHERE Product.prod_id = Supplies.prod_id and
Supplies.supplies_id = 'SP3')+(SELECT supplies_quantity FROM Supplies,Product WHERE supplies.prod_id =Product.prod_id
and Supplies.supplies_id = 'SP3'))) WHERE prod_id = (SELECT Product.prod_id FROM Product,Supplies
WHERE Product.prod_id = Supplies.prod_id and Supplies.supplies_id = 'SP3');
INSERT INTO Supplies(supplies_id, prod_id, sup_id, supplies_quantity, Total_price)
VALUES('SP4', 'PR2', 'SU2', 50, 20.0);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Supplies WHERE Product.prod_id = Supplies.prod_id and
Supplies.supplies_id = 'SP4')+(SELECT supplies_quantity FROM Supplies,Product WHERE supplies.prod_id =Product.prod_id
and Supplies.supplies_id = 'SP4'))) WHERE prod_id = (SELECT Product.prod_id FROM Product,Supplies
WHERE Product.prod_id = Supplies.prod_id and Supplies.supplies_id = 'SP4');
--SEEING UPDATED PRODUCT TABLE AND PRODUCT QUANTITY
SELECT * FROM Product;
--INSERTING INTO ADDRESS
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, emp_id)
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

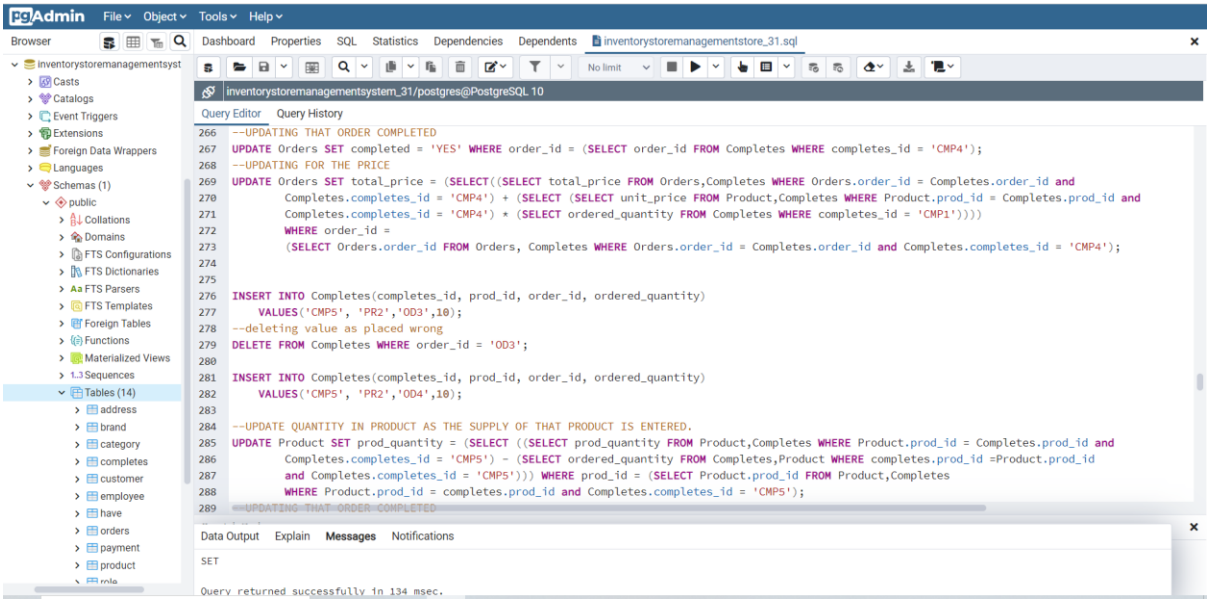

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--INSERTING INTO ADDRESS
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, emp_id)
VALUES('AD1', 'Gujarat', 'Rajkot', 'Jamnagar Road', 'Global', 'V3', 'HN1', 'E1');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, emp_id)
VALUES('AD2', 'Gujarat', 'Rajkot', 'Madhapar', 'Sun Rise', 'S2', 'SR1', 'E2');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, emp_id)
VALUES('AD3', 'MP', 'Ujjain', 'Pragya', 'Prakash', 'P1', 'PR5', 'E3');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, emp_id)
VALUES('AD4', 'Gujarat', 'Rajkot', 'Indira Circle', 'Crystal', 'C5', 'CR2', 'E4');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, emp_id)
VALUES('AD5', 'Gujarat', 'Rajkot', 'Raiya Road', 'Silver Height', 'SH4', 'H1', 'E4');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, cust_id)
VALUES('AD6', 'Gujarat', 'Rajkot', 'Jamnagar Road', 'COPPER CITY', 'CC5', 'HN60', 'CU1');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, cust_id)
VALUES('AD7', 'MP', 'Lucknow', 'Sabar Road', 'HighLights', 'HG2', 'HN120', 'CU2');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, cust_id)
VALUES('AD8', 'Gujarat', 'Surat', 'Surat Highway', 'Sun Raise', 'S5', 'SRE30', 'CU3');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, sup_id)
VALUES('AD9', 'Gujarat', 'Rajkot', 'Raiya Exchange', 'Ramananda', 'R1', 'R10', 'SU1');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, sup_id)
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

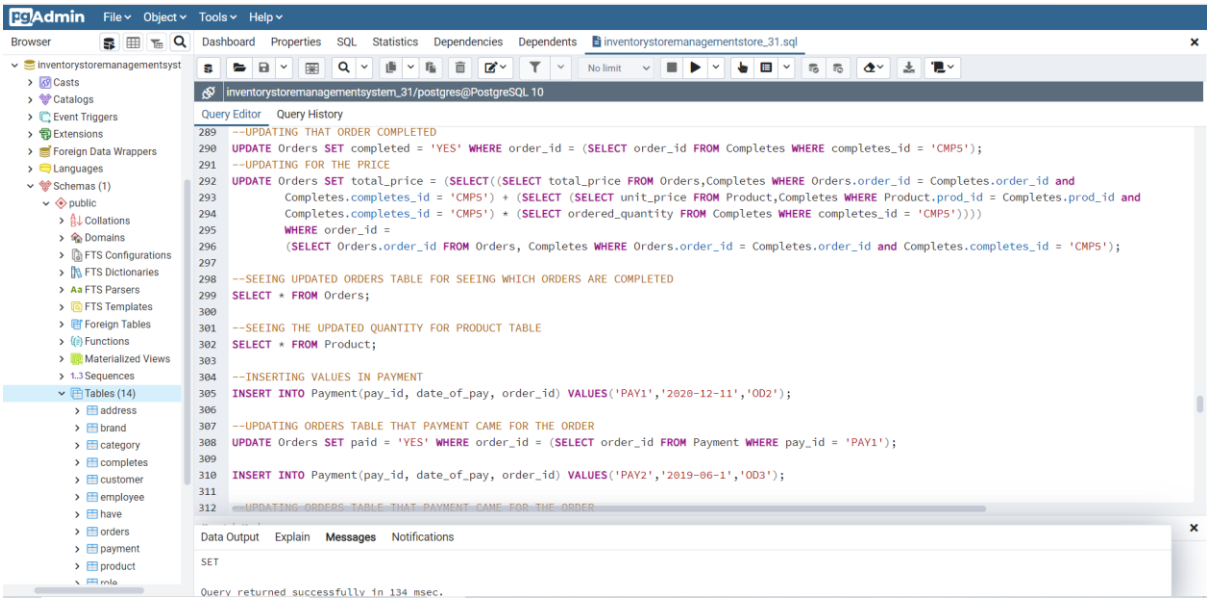

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, sup_id)
VALUES('AD9', 'Gujarat', 'Rajkot', 'Raiya Exchange', 'Ramananda', 'R1', 'R10', 'SU1');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, sup_id)
VALUES('AD10', 'MP', 'Ujjain', 'Gopalpura Road', 'Nivti', 'N2', 'NV28', 'SU2');
INSERT INTO Address(add_id, state, city, road, society_nm, street_no, House_no, sup_id)
VALUES('AD11', 'Gujarat', 'Rajkot', 'Hospital Chowk', 'High Pillars', 'H3', 'HP32', 'SU3');
--INSERTING VALUES IN ORDERS
INSERT INTO Orders(Order_id, cust_id, add_id, date_order) VALUES('OD1','CU1','AD6', '2019-09-21');
INSERT INTO Orders(Order_id, cust_id, add_id,date_order) VALUES('OD2','CU2','AD7', '2020-10-24');
INSERT INTO Orders(Order_id, cust_id, add_id, date_order) VALUES('OD3','CU3','AD8','2018-1-5');
INSERT INTO Orders(Order_id, cust_id, add_id, date_order) VALUES('OD4','CU2','AD7', '2021-10-25');
DELETE FROM Orders WHERE date_order = '2021-10-25';
INSERT INTO Orders(Order_id, cust_id, add_id, date_order) VALUES('OD4','CU2','AD7', '2021-1-25');
INSERT INTO Orders(Order_id, cust_id, add_id, date_order) VALUES('OD5','CU2','AD7', '2021-5-22');
--INSERTING VALUES IN COMPLETES
INSERT INTO Completes(completes_id, prod_id, order_id, ordered_quantity)
VALUES('CMP1', 'PR1','OD1',10);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

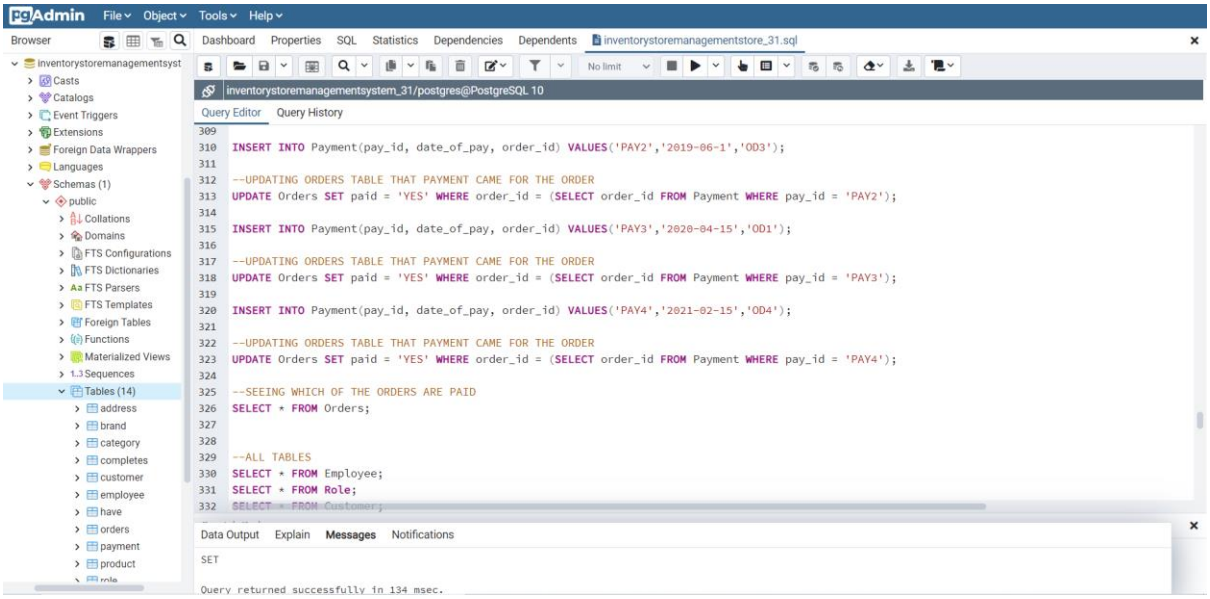

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--INSERTING VALUES IN COMPLETES
INSERT INTO Completes(completes_id, prod_id, order_id, ordered_quantity)
VALUES('CMP1', 'PR1','OD1',10);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP1') - (SELECT ordered_quantity FROM Completes,Product WHERE completes.prod_id =Product.prod_id
and Completes.completes_id = 'CMP1'))) WHERE prod_id = (SELECT Product.prod_id FROM Product,Completes
WHERE Product.prod_id = completes.prod_id and Completes.completes_id = 'CMP1');
--UPDATING THAT ORDER COMPLETED
UPDATE Orders SET completed = 'YES' WHERE order_id = (SELECT order_id FROM Completes WHERE completes_id = 'CMP1');
--UPDATING FOR THE PRICE
UPDATE Orders SET total_price = (SELECT((SELECT total_price FROM Orders,Completes WHERE Orders.order_id = Completes.order_id and
Completes.completes_id = 'CMP1') + (SELECT (SELECT unit_price FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP1') * (SELECT ordered_quantity FROM Completes WHERE completes_id = 'CMP1'))))
WHERE order_id =
(SELECT Orders.order_id FROM Orders, Completes WHERE Orders.order_id = Completes.order_id and Completes.completes_id = 'CMP1');
INSERT INTO Completes(completes_id, prod_id, order_id, ordered_quantity)
VALUES('CMP2', 'PR3','OD1',2);
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.


pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
INSERT INTO Completes(completes_id, prod_id, order_id, ordered_quantity)
VALUES('CMP2', 'PR3','OD1',2);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP2') - (SELECT ordered_quantity FROM Completes,Product WHERE completes.prod_id =Product.prod_id
and Completes.completes_id = 'CMP2'))) WHERE prod_id = (SELECT Product.prod_id FROM Product,Completes
WHERE Product.prod_id = completes.prod_id and Completes.completes_id = 'CMP2');
--UPDATING THAT ORDER COMPLETED
UPDATE Orders SET completed = 'YES' WHERE order_id = (SELECT order_id FROM Completes WHERE completes_id = 'CMP2');
--UPDATING FOR THE PRICE
UPDATE Orders SET total_price = (SELECT((SELECT total_price FROM Orders,Completes WHERE Orders.order_id = Completes.order_id and
Completes.completes_id = 'CMP2') + (SELECT (SELECT unit_price FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP2') * (SELECT ordered_quantity FROM Completes WHERE completes_id = 'CMP2'))))
WHERE order_id =
(SELECT Orders.order_id FROM Orders, Completes WHERE Orders.order_id = Completes.order_id and Completes.completes_id = 'CMP2');
INSERT INTO Completes(completes_id, prod_id, order_id, ordered_quantity)
VALUES('CMP3', 'PR1','OD2',22);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Data Output Explain Messages Notifications
SET

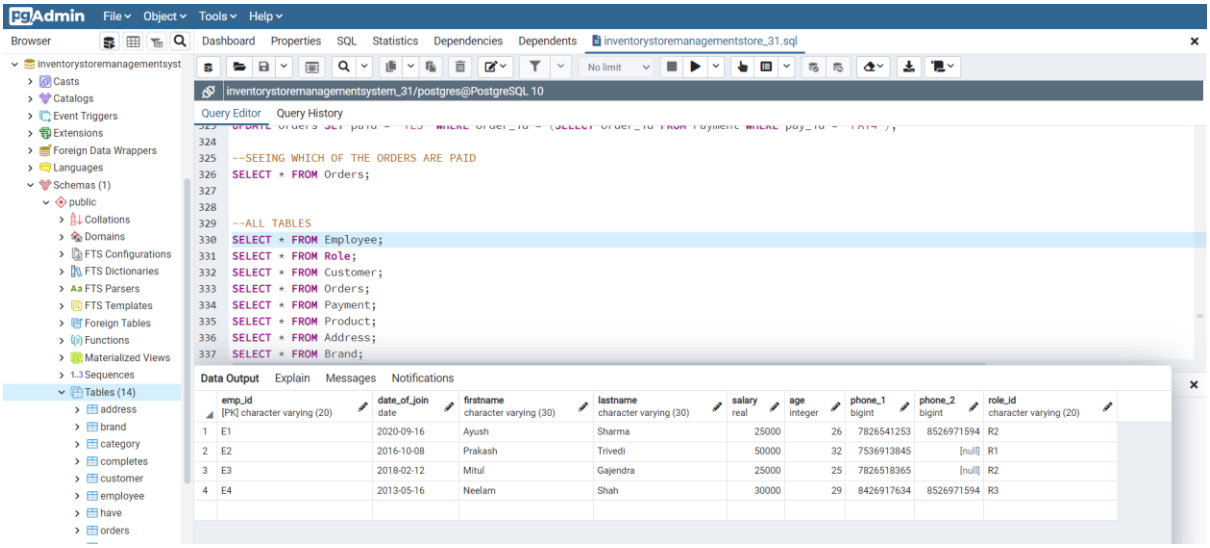

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP3') - (SELECT ordered_quantity FROM Completes,Product WHERE completes.prod_id =Product.prod_id
and Completes.completes_id = 'CMP3'))) WHERE prod_id = (SELECT Product.prod_id FROM Product,Completes
WHERE Product.prod_id = completes.prod_id and Completes.completes_id = 'CMP3');
--UPDATING THAT ORDER COMPLETED
UPDATE Orders SET completed = 'YES' WHERE order_id = (SELECT order_id FROM Completes WHERE completes_id = 'CMP3');
--UPDATING FOR THE PRICE
UPDATE Orders SET total_price = (SELECT((SELECT total_price FROM Orders,Completes WHERE Orders.order_id = Completes.order_id and
Completes.completes_id = 'CMP3') + (SELECT (SELECT unit_price FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP3') * (SELECT ordered_quantity FROM Completes WHERE completes_id = 'CMP3'))))
WHERE order_id =
(SELECT Orders.order_id FROM Orders, Completes WHERE Orders.order_id = Completes.order_id and Completes.completes_id = 'CMP3');
INSERT INTO Completes(completes_id, prod_id, order_id, ordered_quantity)
VALUES('CMP4', 'PR2','OD3',30);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP4') - (SELECT ordered_quantity FROM Completes,Product WHERE completes.prod_id =Product.prod_id
and Completes.completes_id = 'CMP4'))) WHERE prod_id = (SELECT Product.prod_id FROM Product,Completes
WHERE Product.prod_id = completes.prod_id and Completes.completes_id = 'CMP4');
--UPDATING THAT ORDER COMPLETED
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

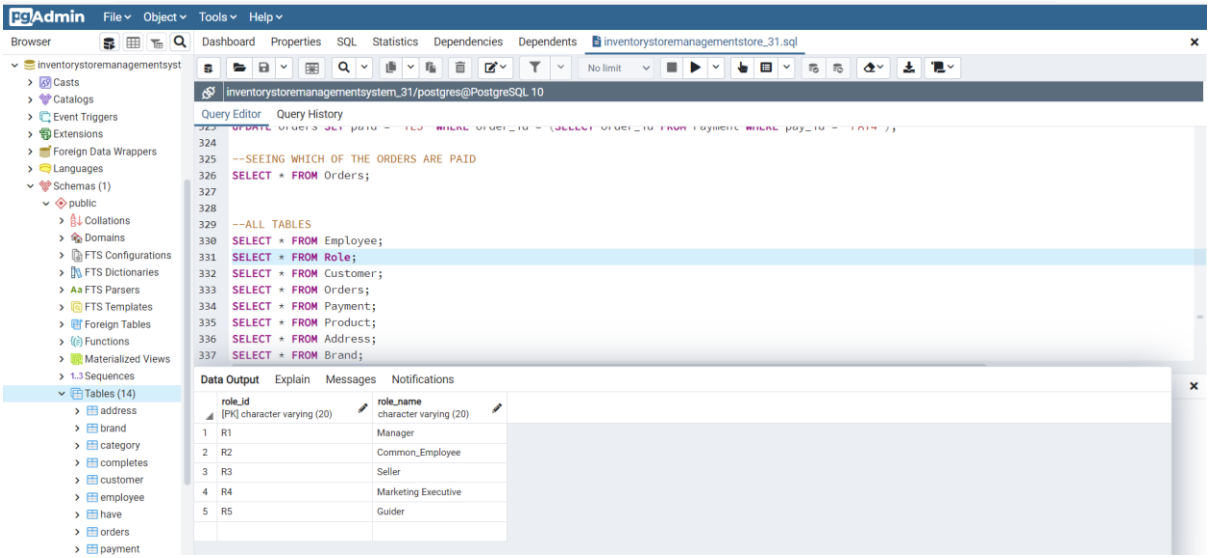

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--UPDATING THAT ORDER COMPLETED
UPDATE Orders SET completed = 'YES' WHERE order_id = (SELECT order_id FROM Completes WHERE completes_id = 'CMP4');
--UPDATING FOR THE PRICE
UPDATE Orders SET total_price = (SELECT((SELECT total_price FROM Orders,Completes WHERE Orders.order_id = Completes.order_id and
Completes.completes_id = 'CMP4') + (SELECT (SELECT unit_price FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP4') * (SELECT ordered_quantity FROM Completes WHERE completes_id = 'CMP1'))))
WHERE order_id =
(SELECT Orders.order_id FROM Orders, Completes WHERE Orders.order_id = Completes.order_id and Completes.completes_id = 'CMP4');
INSERT INTO Completes(completes_id, prod_id, order_id, ordered_quantity)
VALUES('CMP5', 'PR2','OD3',10);
--deleting value as placed wrong
DELETE FROM Completes WHERE order_id = 'OD3';
INSERT INTO Completes(completes_id, prod_id, order_id, ordered_quantity)
VALUES('CMP5', 'PR2','OD4',10);
--UPDATE QUANTITY IN PRODUCT AS THE SUPPLY OF THAT PRODUCT IS ENTERED.
UPDATE Product SET prod_quantity = (SELECT ((SELECT prod_quantity FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP5') - (SELECT ordered_quantity FROM Completes,Product WHERE completes.prod_id =Product.prod_id
and Completes.completes_id = 'CMP5'))) WHERE prod_id = (SELECT Product.prod_id FROM Product,Completes
WHERE Product.prod_id = completes.prod_id and Completes.completes_id = 'CMP5');
--UPDATING THAT ORDER COMPLETED
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

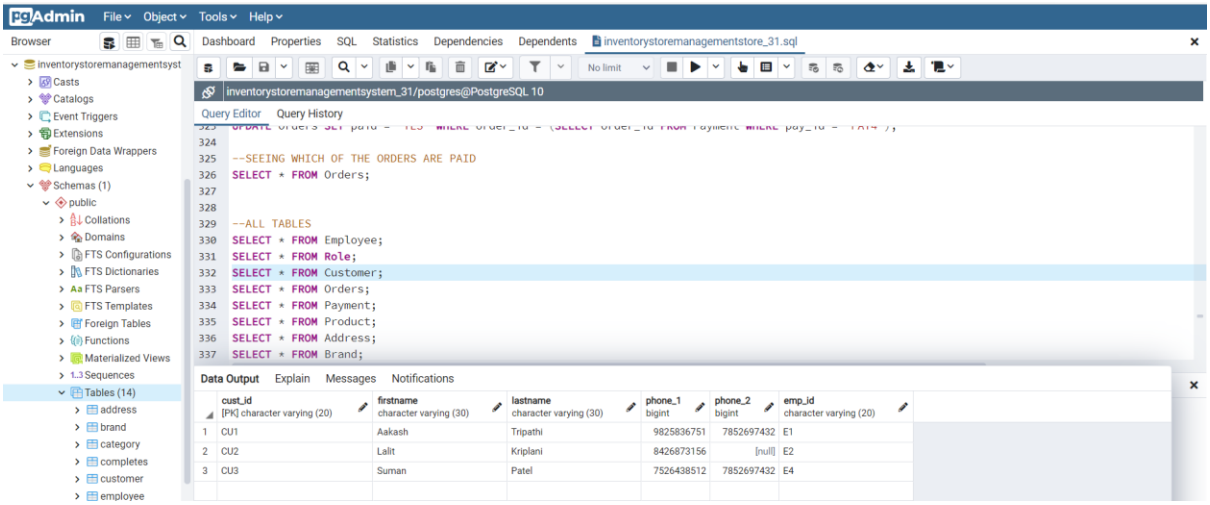

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--UPDATING THAT ORDER COMPLETED
UPDATE Orders SET completed = 'YES' WHERE order_id = (SELECT order_id FROM Completes WHERE completes_id = 'CMP5');
--UPDATING FOR THE PRICE
UPDATE Orders SET total_price = (SELECT((SELECT total_price FROM Orders,Completes WHERE Orders.order_id = Completes.order_id and
Completes.completes_id = 'CMP5') + (SELECT (SELECT unit_price FROM Product,Completes WHERE Product.prod_id = Completes.prod_id and
Completes.completes_id = 'CMP5') * (SELECT ordered_quantity FROM Completes WHERE completes_id = 'CMP5'))))
WHERE order_id =
(SELECT Orders.order_id FROM Orders, Completes WHERE Orders.order_id = Completes.order_id and Completes.completes_id = 'CMP5');
--SEEING UPDATED ORDERS TABLE FOR SEEING WHICH ORDERS ARE COMPLETED
SELECT * FROM Orders;
--SEEING THE UPDATED QUANTITY FOR PRODUCT TABLE
SELECT * FROM Product;
--INSERTING VALUES IN PAYMENT
INSERT INTO Payment(pay_id, date_of_pay, order_id) VALUES('PAY1','2020-12-11','OD2');
--UPDATING ORDERS TABLE THAT PAYMENT CAME FOR THE ORDER
UPDATE Orders SET paid = 'YES' WHERE order_id = (SELECT order_id FROM Payment WHERE pay_id = 'PAY1');
INSERT INTO Payment(pay_id, date_of_pay, order_id) VALUES('PAY2','2019-06-1','OD3');
--UPDATING ORDERS TABLE THAT PAYMENT CAME FOR THE ORDER
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

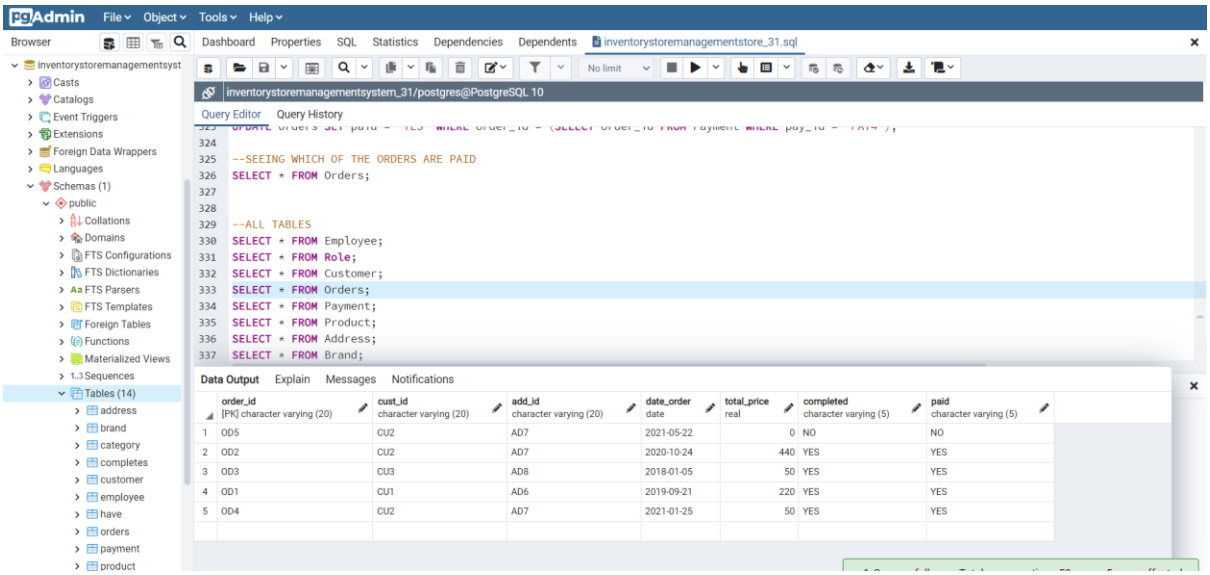

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
309
310 INSERT INTO Payment(pay_id, date_of_pay, order_id) VALUES('PAY2','2019-06-1','OD3');
311
312 --UPDATING ORDERS TABLE THAT PAYMENT CAME FOR THE ORDER
313 UPDATE Orders SET paid = 'YES' WHERE order_id = (SELECT order_id FROM Payment WHERE pay_id = 'PAY2');
314
315 INSERT INTO Payment(pay_id, date_of_pay, order_id) VALUES('PAY3','2020-04-15','OD1');
316
317 --UPDATING ORDERS TABLE THAT PAYMENT CAME FOR THE ORDER
318 UPDATE Orders SET paid = 'YES' WHERE order_id = (SELECT order_id FROM Payment WHERE pay_id = 'PAY3');
319
320 INSERT INTO Payment(pay_id, date_of_pay, order_id) VALUES('PAY4','2021-02-15','OD4');
321
322 --UPDATING ORDERS TABLE THAT PAYMENT CAME FOR THE ORDER
323 UPDATE Orders SET paid = 'YES' WHERE order_id = (SELECT order_id FROM Payment WHERE pay_id = 'PAY4');
324
325 --SEEING WHICH OF THE ORDERS ARE PAID
326 SELECT * FROM Orders;
327
328
329 --ALL TABLES
330 SELECT * FROM Employee;
331 SELECT * FROM Role;
332 SELECT * FROM Customer;
Data Output Explain Messages Notifications
SET
Query returned successfully in 134 msec.

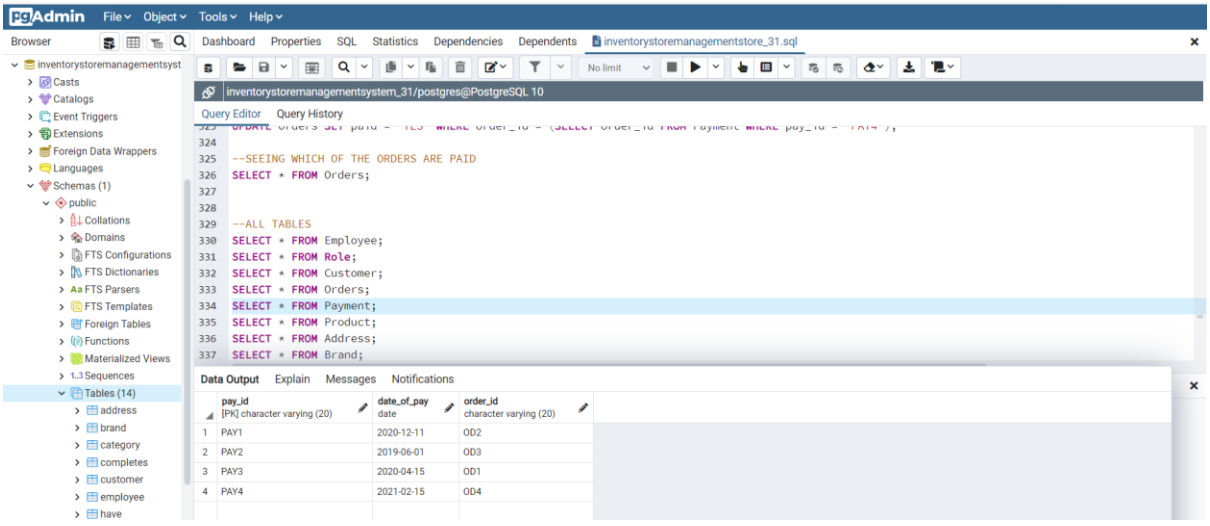
TABLES:

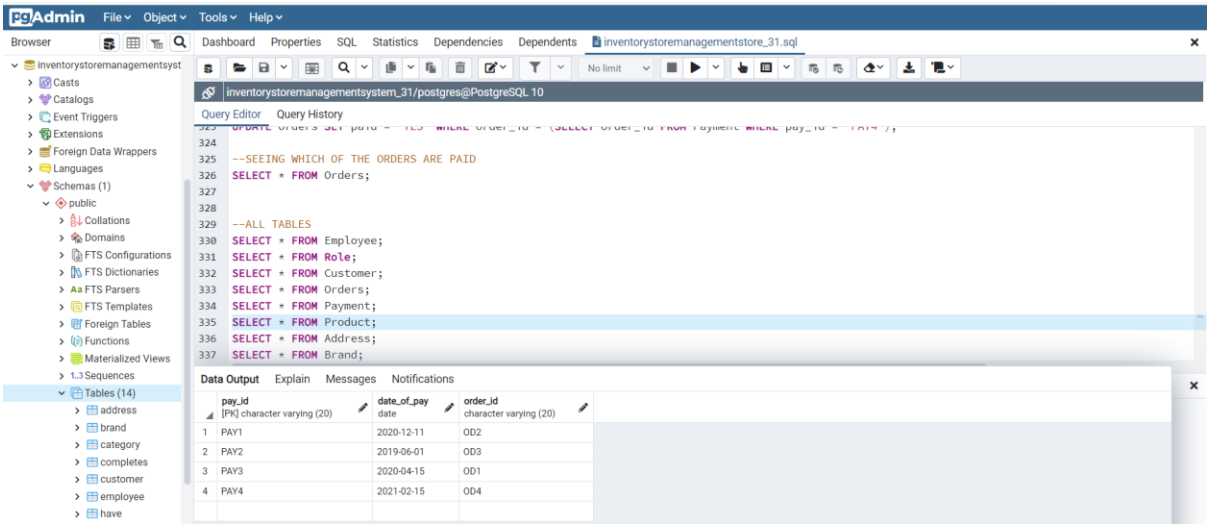

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
324
325 --SEEING WHICH OF THE ORDERS ARE PAID
326 SELECT * FROM Orders;
327
328
329 --ALL TABLES
330 SELECT * FROM Employee;
331 SELECT * FROM Role;
332 SELECT * FROM Customer;
333 SELECT * FROM Orders;
334 SELECT * FROM Payment;
335 SELECT * FROM Product;
336 SELECT * FROM Address;
337 SELECT * FROM Brand;
Data Output Explain Messages Notifications
emp_id [PK] character varying (20) | date_of_join date | firstname character varying (30) | lastname character varying (30) | salary real | age integer | phone_1 bigint | phone_2 bigint | role_id character varying (20)
1 E1 2020-09-16 Ayush Sharma 25000 26 7826541253 8526971594 R2
2 E2 2016-10-08 Prakash Trivedi 50000 32 7536913845 [null] R1
3 E3 2018-02-12 Mitul Gajendra 25000 25 7826518365 [null] R2
4 E4 2013-05-16 Neelam Shah 30000 29 8426917634 8526971594 R3

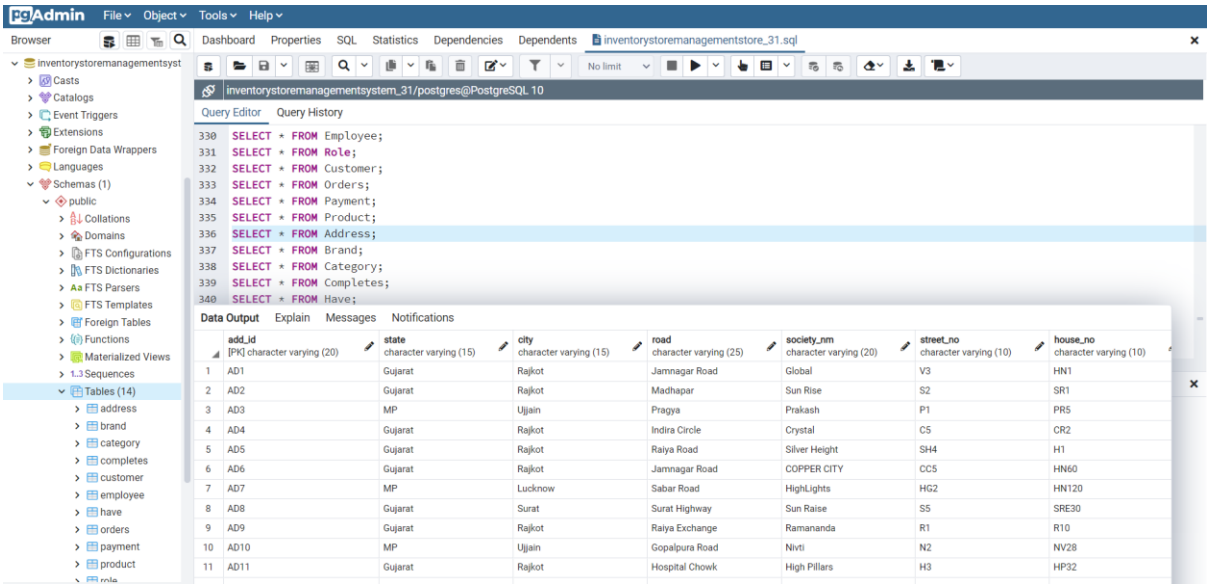

pgAdmin File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
324
325 --SEEING WHICH OF THE ORDERS ARE PAID
326 SELECT * FROM Orders;
327
328
329 --ALL TABLES
330 SELECT * FROM Employee;
331 SELECT * FROM Role;
332 SELECT * FROM Customer;
333 SELECT * FROM Orders;
334 SELECT * FROM Payment;
335 SELECT * FROM Product;
336 SELECT * FROM Address;
337 SELECT * FROM Brand;
Data Output Explain Messages Notifications
role_id [PK] character varying (20) | role_name character varying (20)
1 R1 Manager
2 R2 Common_Employee
3 R3 Seller
4 R4 Marketing Executive
5 R5 Guider

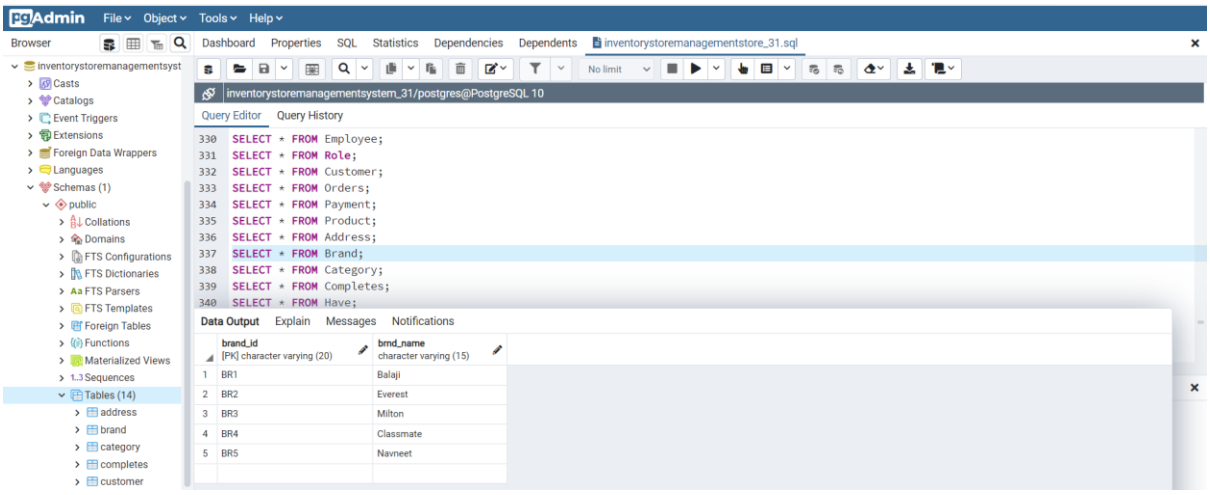

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--SEEING WHICH OF THE ORDERS ARE PAID
SELECT * FROM Orders;
--ALL TABLES
SELECT * FROM Employee;
SELECT * FROM Role;
SELECT * FROM Customer;
SELECT * FROM Orders;
SELECT * FROM Payment;
SELECT * FROM Product;
SELECT * FROM Address;
SELECT * FROM Brand;
Data Output Explain Messages Notifications
cust_id [PK] character varying (20)
firstname character varying (30)
lastname character varying (30)
phone_1 bigint
phone_2 bigint
emp_id character varying (20)
1 CU1 Aakash Tripathi 9825836751 7852697432 E1
2 CU2 Lalit Kriplani 8426873156 [null] E2
3 CU3 Suman Patel 7526438512 7852697432 E4

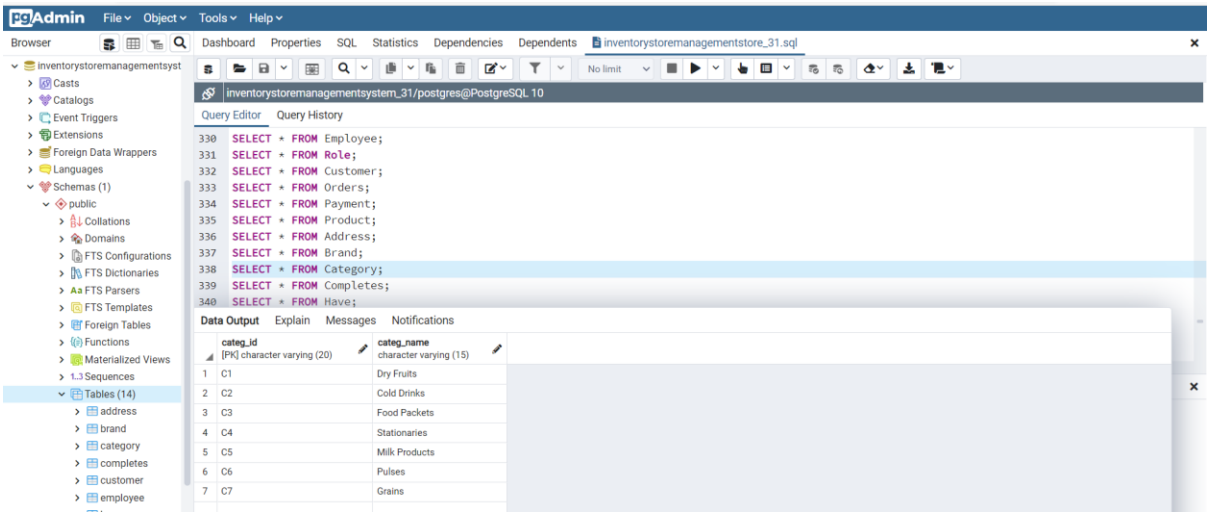

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--SEEING WHICH OF THE ORDERS ARE PAID
SELECT * FROM Orders;
--ALL TABLES
SELECT * FROM Employee;
SELECT * FROM Role;
SELECT * FROM Customer;
SELECT * FROM Orders;
SELECT * FROM Payment;
SELECT * FROM Product;
SELECT * FROM Address;
SELECT * FROM Brand;
Data Output Explain Messages Notifications
order_id [PK] character varying (20)
cust_id character varying (20)
add_id character varying (20)
date_order date
total_price real
completed character varying (5)
paid character varying (5)
1 OD5 CU2 AD7 2021-05-22 0 NO NO
2 OD2 CU2 AD7 2020-10-24 440 YES YES
3 OD3 CU3 AD8 2018-01-05 50 YES YES
4 OD1 CU1 AD6 2019-09-21 220 YES YES
5 OD4 CU2 AD7 2021-01-25 50 YES YES

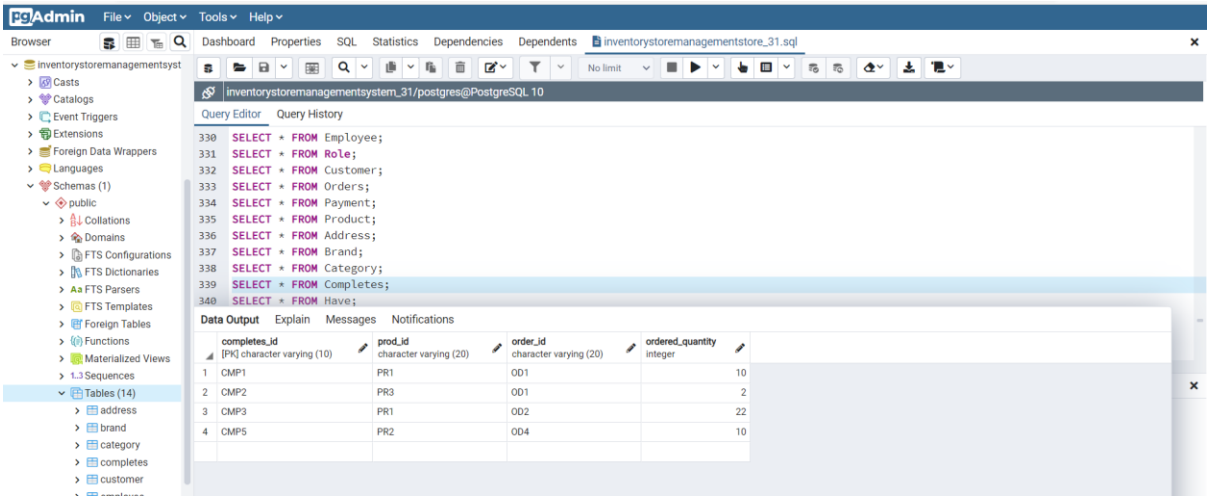

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--SEEING WHICH OF THE ORDERS ARE PAID
SELECT * FROM Orders;
--ALL TABLES
SELECT * FROM Employee;
SELECT * FROM Role;
SELECT * FROM Customer;
SELECT * FROM Orders;
SELECT * FROM Payment;
SELECT * FROM Product;
SELECT * FROM Address;
SELECT * FROM Brand;
Data Output Explain Messages Notifications
pay_id [PK] character varying (20)
date_of_pay date
order_id character varying (20)
1 PAY1 2020-12-11 OD2
2 PAY2 2019-06-01 OD3
3 PAY3 2020-04-15 OD1
4 PAY4 2021-02-15 OD4

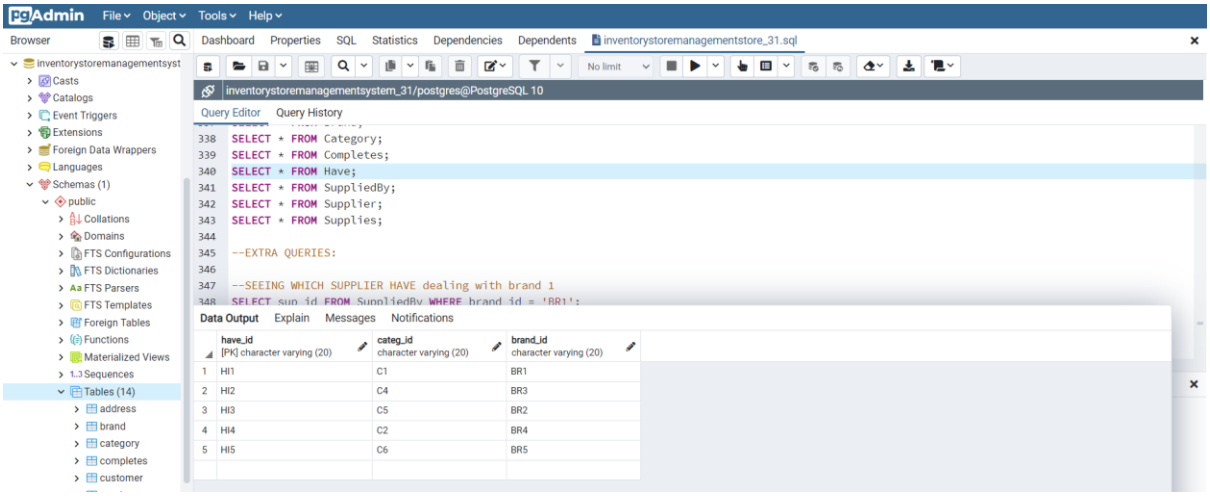

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--SEEING WHICH OF THE ORDERS ARE PAID
SELECT * FROM Orders;
--ALL TABLES
SELECT * FROM Employee;
SELECT * FROM Role;
SELECT * FROM Customer;
SELECT * FROM Orders;
SELECT * FROM Payment;
SELECT * FROM Product;
SELECT * FROM Address;
SELECT * FROM Brand;
Data Output Explain Messages Notifications
pay_id [PK] character varying (20)
date_of_pay date
order_id character varying (20)
1 PAY1 2020-12-11 OD2
2 PAY2 2019-06-01 OD3
3 PAY3 2020-04-15 OD1
4 PAY4 2021-02-15 OD4

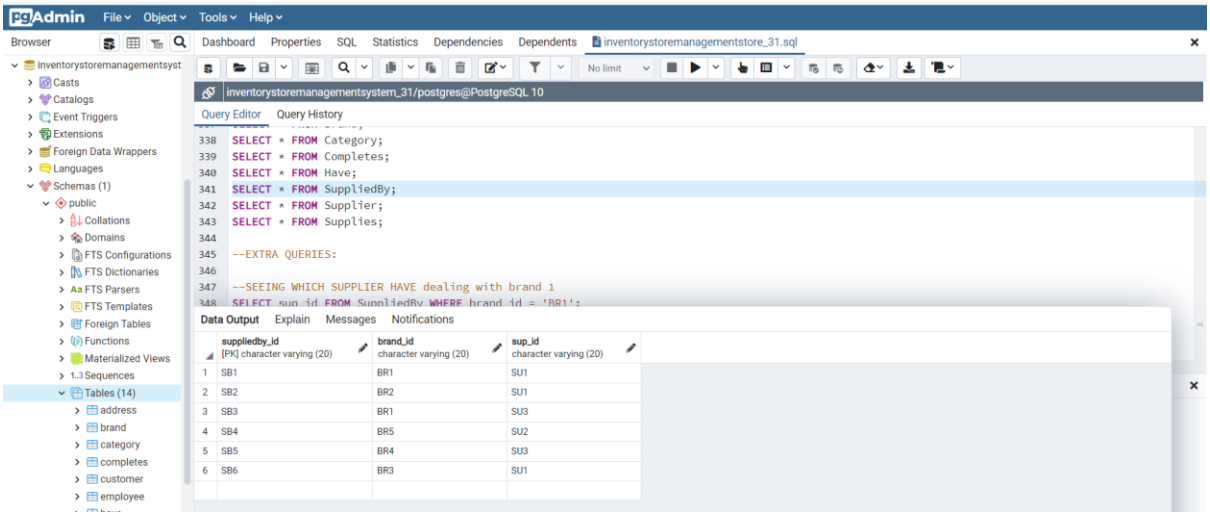

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
SELECT * FROM Employee;
SELECT * FROM Role;
SELECT * FROM Customer;
SELECT * FROM Orders;
SELECT * FROM Payment;
SELECT * FROM Product;
SELECT * FROM Address;
SELECT * FROM Brand;
SELECT * FROM Category;
SELECT * FROM Completes;
SELECT * FROM Have;
Data Output Explain Messages Notifications
add_id [PK] character varying (20)
state character varying (15)
city character varying (15)
road character varying (25)
society_nm character varying (20)
street_no character varying (10)
house_no character varying (10)
1 AD1 Gujarat Rajkot Jamnagar Road Global V3 HN1
2 AD2 Gujarat Rajkot Madhapar Sun Rise S2 SR1
3 AD3 MP Ujjain Pragya Prakash P1 PR5
4 AD4 Gujarat Rajkot Indira Circle Crystal C5 CR2
5 AD5 Gujarat Rajkot Raiya Road Silver Height SH4 H1
6 AD6 Gujarat Rajkot Jamnagar Road COPPER CITY CC5 HN60
7 AD7 MP Lucknow Sabar Road HighLights HG2 HN120
8 AD8 Gujarat Surat Surat Highway Sun Raise S5 SRE30
9 AD9 Gujarat Rajkot Raiya Exchange Ramananda R1 R10
10 AD10 MP Ujjain Gopalpura Road Nivti N2 NV28
11 AD11 Gujarat Rajkot Hospital Chowk High Pillars H3 HP32

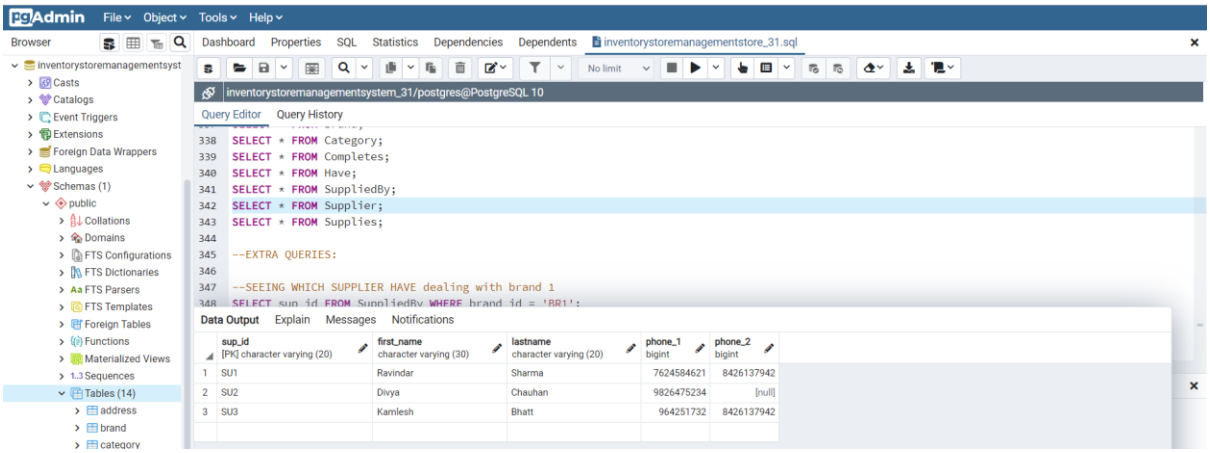

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
SELECT * FROM Employee;
SELECT * FROM Role;
SELECT * FROM Customer;
SELECT * FROM Orders;
SELECT * FROM Payment;
SELECT * FROM Product;
SELECT * FROM Address;
SELECT * FROM Brand;
SELECT * FROM Category;
SELECT * FROM Completes;
SELECT * FROM Have;
Data Output Explain Messages Notifications
brand_id [PK] character varying (20)
brnd_name character varying (15)
1 BR1 Balaji
2 BR2 Everest
3 BR3 Milton
4 BR4 Classmate
5 BR5 Navneet

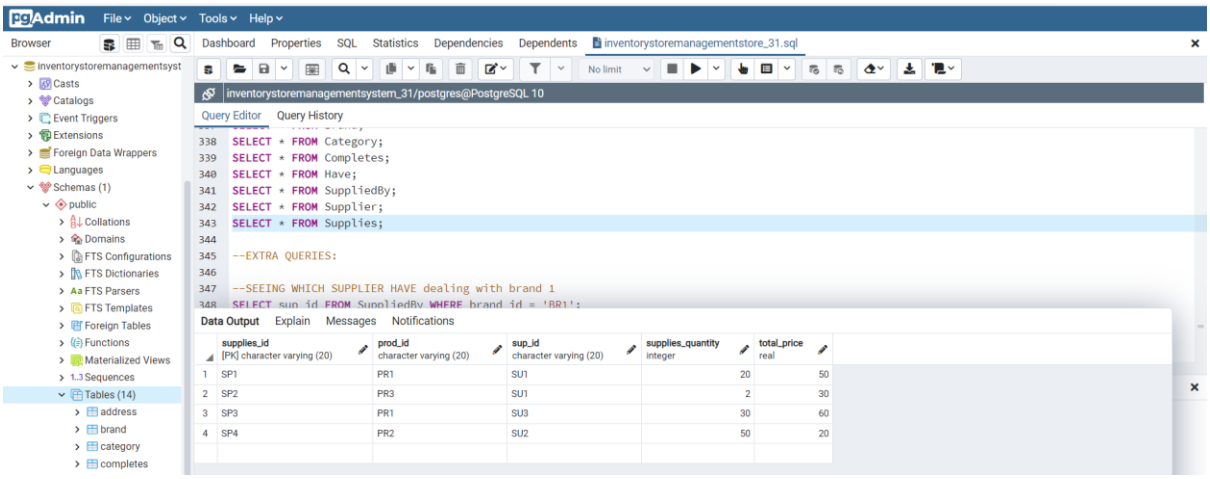

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsyst
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (14)
address
brand
category
completes
customer
employee
No limit
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
330 SELECT * FROM Employee;
331 SELECT * FROM Role;
332 SELECT * FROM Customer;
333 SELECT * FROM Orders;
334 SELECT * FROM Payment;
335 SELECT * FROM Product;
336 SELECT * FROM Address;
337 SELECT * FROM Brand;
338 SELECT * FROM Category;
339 SELECT * FROM Completes;
340 SELECT * FROM Have;
Data Output Explain Messages Notifications
categ_id [PK] character varying (20)
categ_name character varying (15)
1 C1 Dry Fruits
2 C2 Cold Drinks
3 C3 Food Packets
4 C4 Stationaries
5 C5 Milk Products
6 C6 Pulses
7 C7 Grains


pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsyst
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (14)
address
brand
category
completes
customer
No limit
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
330 SELECT * FROM Employee;
331 SELECT * FROM Role;
332 SELECT * FROM Customer;
333 SELECT * FROM Orders;
334 SELECT * FROM Payment;
335 SELECT * FROM Product;
336 SELECT * FROM Address;
337 SELECT * FROM Brand;
338 SELECT * FROM Category;
339 SELECT * FROM Completes;
340 SELECT * FROM Have;
Data Output Explain Messages Notifications
completes_id [PK] character varying (10)
prod_id character varying (20)
order_id character varying (20)
ordered_quantity integer
1 CMP1 PR1 OD1 10
2 CMP2 PR3 OD1 2
3 CMP3 PR1 OD2 22
4 CMP5 PR2 OD4 10

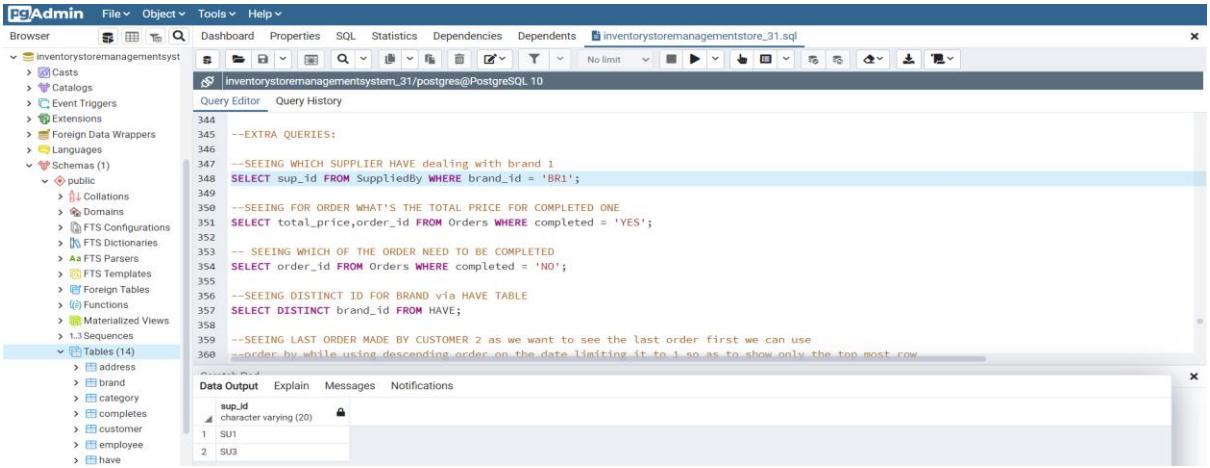

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsyst
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (14)
address
brand
category
completes
customer
No limit
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
338 SELECT * FROM Category;
339 SELECT * FROM Completes;
340 SELECT * FROM Have;
341 SELECT * FROM SuppliedBy;
342 SELECT * FROM Supplier;
343 SELECT * FROM Supplies;
344
345 --EXTRA QUERIES:
346
347 --SEEING WHICH SUPPLIER HAVE dealing with brand 1
348 SELECT sup_id FROM SuppliedBy WHERE brand_id = 'BR1';
Data Output Explain Messages Notifications
have_id [PK] character varying (20)
categ_id character varying (20)
brand_id character varying (20)
1 HI1 C1 BR1
2 HI2 C4 BR3
3 HI3 C5 BR2
4 HI4 C2 BR4
5 HI5 C6 BR5

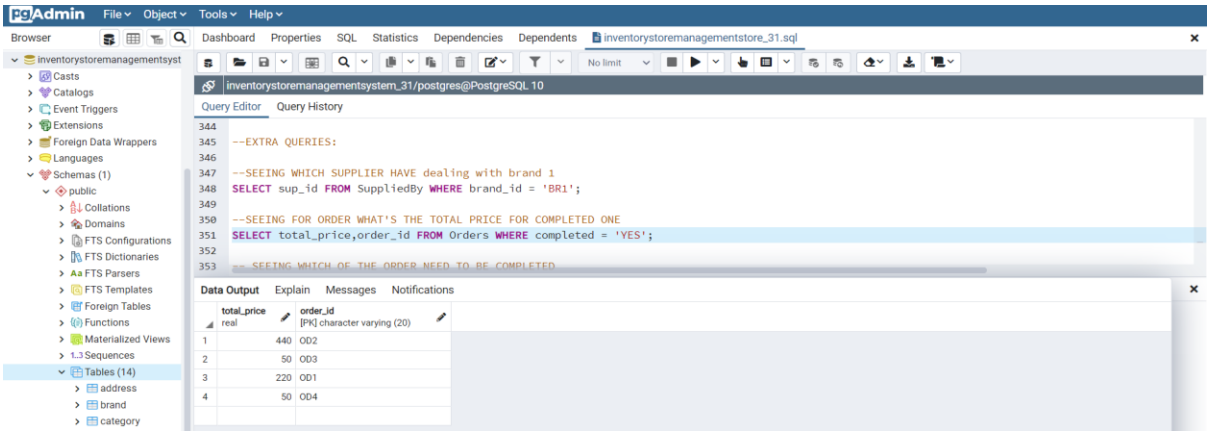

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsyst
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (14)
address
brand
category
completes
customer
employee
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
338 SELECT * FROM Category;
339 SELECT * FROM Completes;
340 SELECT * FROM Have;
341 SELECT * FROM SuppliedBy;
342 SELECT * FROM Supplier;
343 SELECT * FROM Supplies;
344
345 --EXTRA QUERIES:
346
347 --SEEING WHICH SUPPLIER HAVE dealing with brand 1
Data Output Explain Messages Notifications
suppliedby_id [PK] character varying (20)
brand_id character varying (20)
sup_id character varying (20)
1 SB1 BR1 SU1
2 SB2 BR2 SU1
3 SB3 BR1 SU3
4 SB4 BR5 SU2
5 SB5 BR4 SU3
6 SB6 BR3 SU1

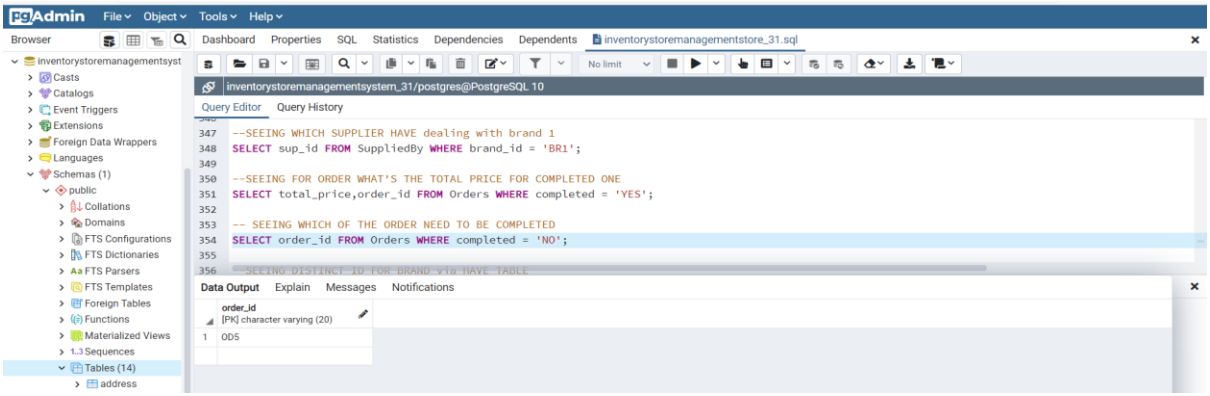

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsyst
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (14)
address
brand
category
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
338 SELECT * FROM Category;
339 SELECT * FROM Completes;
340 SELECT * FROM Have;
341 SELECT * FROM SuppliedBy;
342 SELECT * FROM Supplier;
343 SELECT * FROM Supplies;
344
345 --EXTRA QUERIES:
346
347 --SEEING WHICH SUPPLIER HAVE dealing with brand 1
Data Output Explain Messages Notifications
sup_id [PK] character varying (20)
first_name character varying (30)
lastname character varying (20)
phone_1 bigint
phone_2 bigint
1 SU1 Ravindar Sharma 7624584621 8426137942
2 SU2 Divya Chauhan 9826475234 [null]
3 SU3 Kamlesh Bhatt 964251732 8426137942

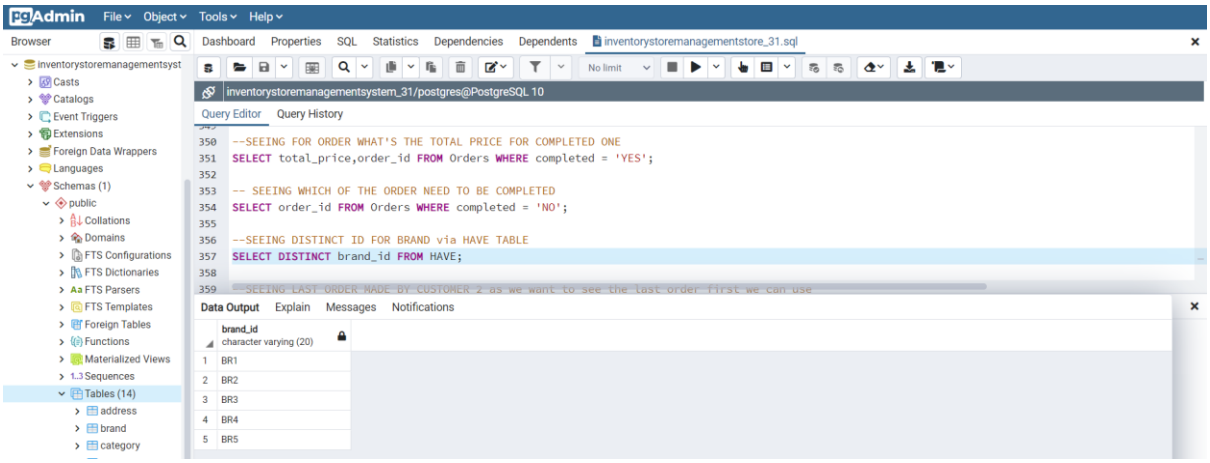

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsyst
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Sequences
Tables (14)
address
brand
category
completes
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
338 SELECT * FROM Category;
339 SELECT * FROM Completes;
340 SELECT * FROM Have;
341 SELECT * FROM SuppliedBy;
342 SELECT * FROM Supplier;
343 SELECT * FROM Supplies;
344
345 --EXTRA QUERIES:
346
347 --SEEING WHICH SUPPLIER HAVE dealing with brand 1
Data Output Explain Messages Notifications
supplies_id [PK] character varying (20)
prod_id character varying (20)
sup_id character varying (20)
supplies_quantity integer
total_price real
1 SP1 PR1 SU1 20 50
2 SP2 PR3 SU1 2 30
3 SP3 PR1 SU3 30 60
4 SP4 PR2 SU2 50 20

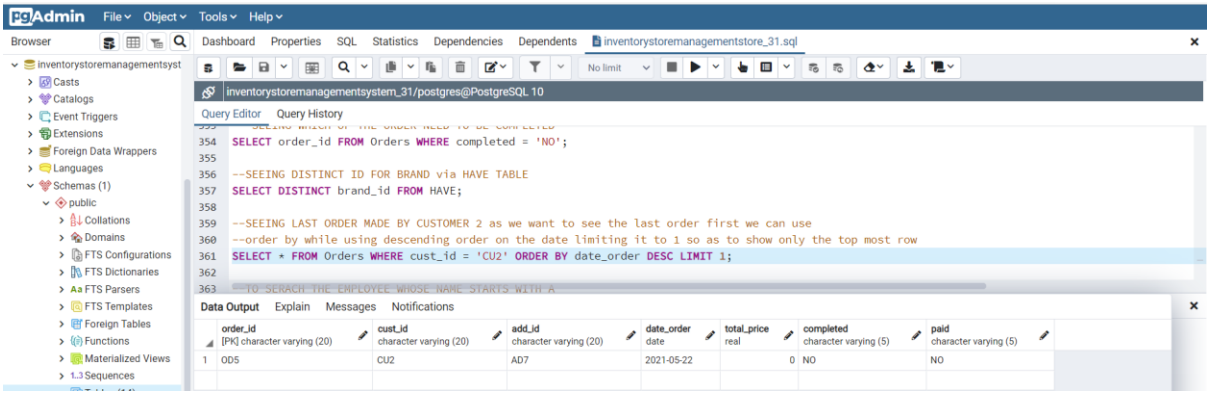
QUERIES:

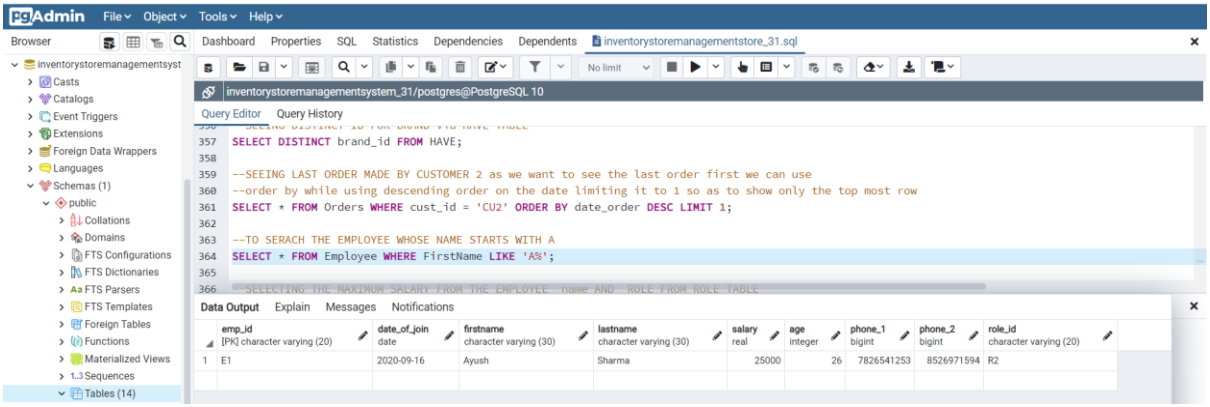

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--EXTRA QUERIES:
--SEEING WHICH SUPPLIER HAVE dealing with brand 1
SELECT sup_id FROM SuppliedBy WHERE brand_id = 'BR1';
--SEEING FOR ORDER WHAT'S THE TOTAL PRICE FOR COMPLETED ONE
SELECT total_price,order_id FROM Orders WHERE completed = 'YES';
-- SEEING WHICH OF THE ORDER NEED TO BE COMPLETED
SELECT order_id FROM Orders WHERE completed = 'NO';
--SEEING DISTINCT ID FOR BRAND via HAVE TABLE
SELECT DISTINCT brand_id FROM HAVE;
--SEEING LAST ORDER MADE BY CUSTOMER 2 as we want to see the last order first we can use
--order by while using descending order on the date limiting it to 1 so as to show only the top most row
Data Output Explain Messages Notifications
sup_id character varying (20)
SU1
SU3

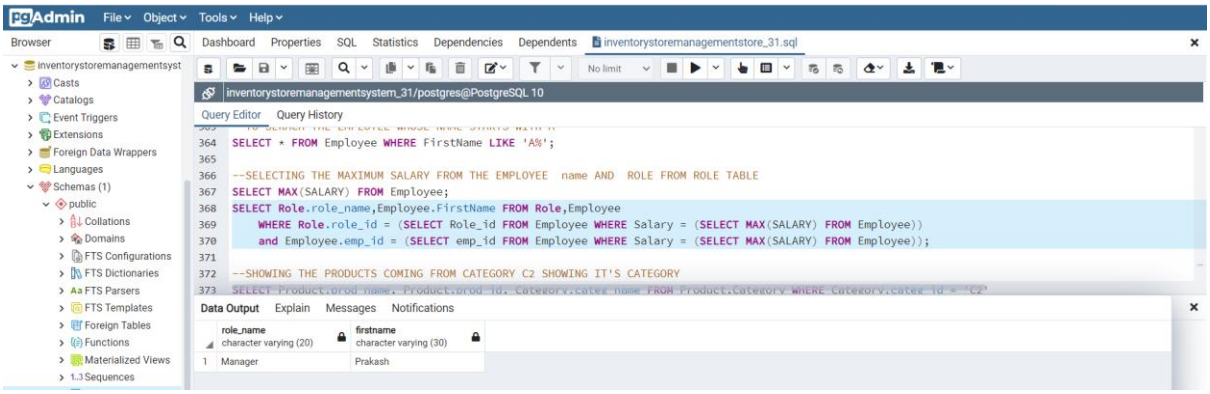

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--EXTRA QUERIES:
--SEEING WHICH SUPPLIER HAVE dealing with brand 1
SELECT sup_id FROM SuppliedBy WHERE brand_id = 'BR1';
--SEEING FOR ORDER WHAT'S THE TOTAL PRICE FOR COMPLETED ONE
SELECT total_price,order_id FROM Orders WHERE completed = 'YES';
-- SEEING WHICH OF THE ORDER NEED TO BE COMPLETED
Data Output Explain Messages Notifications
total_price real
order_id [PK] character varying (20)
440 OD2
50 OD3
220 OD1
50 OD4

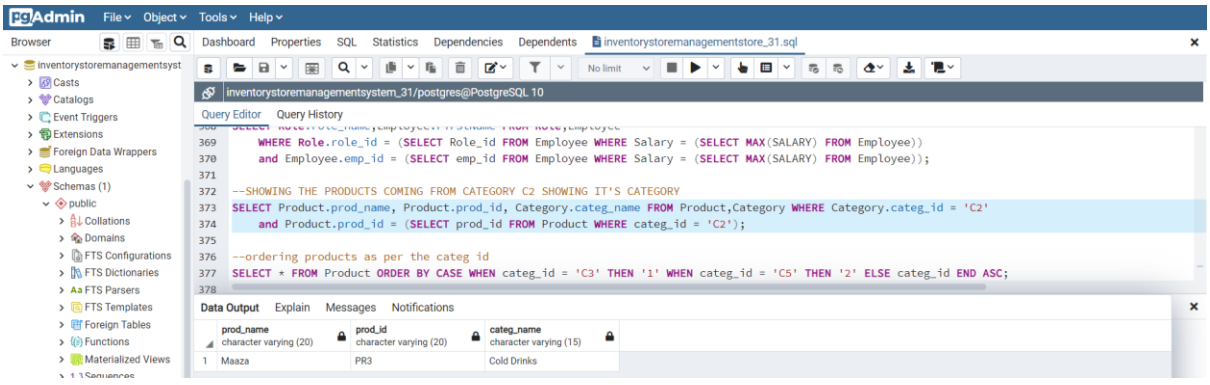

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--SEEING WHICH SUPPLIER HAVE dealing with brand 1
SELECT sup_id FROM SuppliedBy WHERE brand_id = 'BR1';
--SEEING FOR ORDER WHAT'S THE TOTAL PRICE FOR COMPLETED ONE
SELECT total_price,order_id FROM Orders WHERE completed = 'YES';
-- SEEING WHICH OF THE ORDER NEED TO BE COMPLETED
SELECT order_id FROM Orders WHERE completed = 'NO';
--SEEING DISTINCT ID FOR BRAND via HAVE TABLE
Data Output Explain Messages Notifications
order_id [PK] character varying (20)
OD5

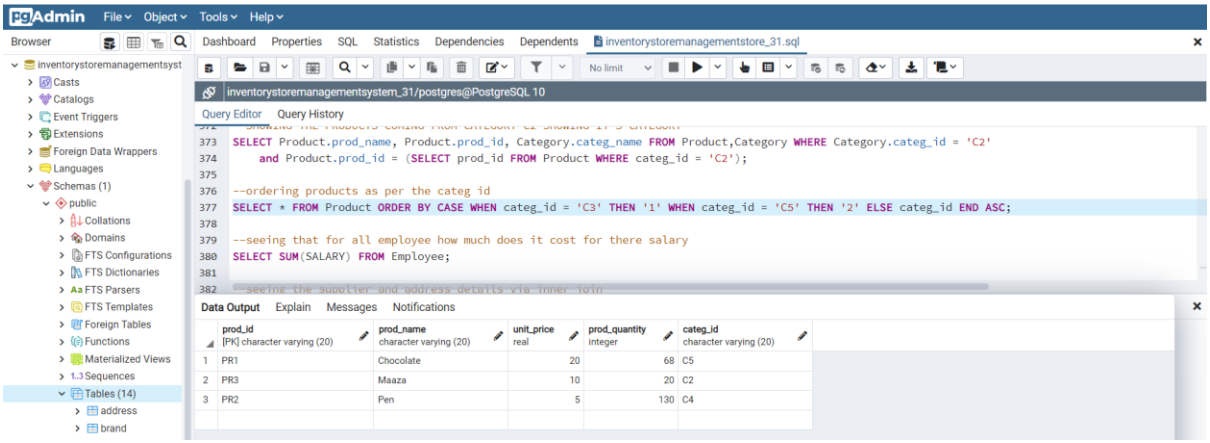

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--SEEING FOR ORDER WHAT'S THE TOTAL PRICE FOR COMPLETED ONE
SELECT total_price,order_id FROM Orders WHERE completed = 'YES';
-- SEEING WHICH OF THE ORDER NEED TO BE COMPLETED
SELECT order_id FROM Orders WHERE completed = 'NO';
--SEEING DISTINCT ID FOR BRAND via HAVE TABLE
SELECT DISTINCT brand_id FROM HAVE;
--SEEING LAST ORDER MADE BY CUSTOMER 2 as we want to see the last order first we can use
Data Output Explain Messages Notifications
brand_id character varying (20)
BR1
BR2
BR3
BR4
BR5

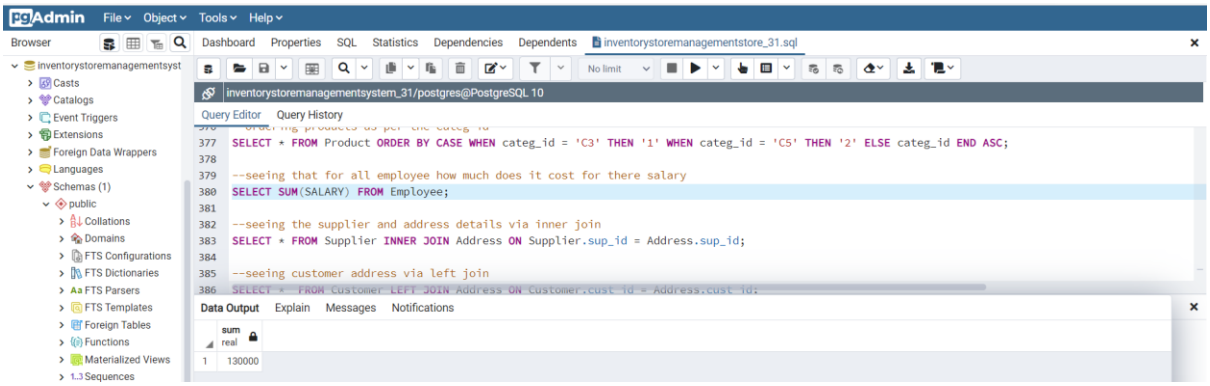

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
354 SELECT order_id FROM Orders WHERE completed = 'NO';
355
356 --SEEING DISTINCT ID FOR BRAND via HAVE TABLE
357 SELECT DISTINCT brand_id FROM HAVE;
358
359 --SEEING LAST ORDER MADE BY CUSTOMER 2 as we want to see the last order first we can use
360 --order by while using descending order on the date limiting it to 1 so as to show only the top most row
361 SELECT * FROM Orders WHERE cust_id = 'CU2' ORDER BY date_order DESC LIMIT 1;
362
363 --TO SERACH THE EMPLOYEE WHOSE NAME STARTS WITH A
Data Output Explain Messages Notifications
order_id [PK] character varying (20)
cust_id character varying (20)
add_id character varying (20)
date_order date
total_price real
completed character varying (5)
paid character varying (5)
1 OD5 CU2 AD7 2021-05-22 0 NO NO

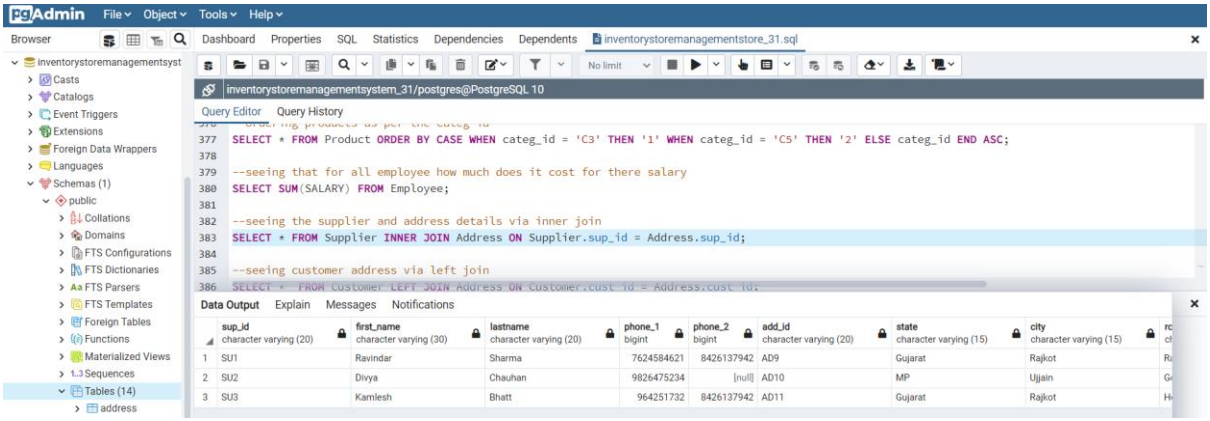

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
357 SELECT DISTINCT brand_id FROM HAVE;
358
359 --SEEING LAST ORDER MADE BY CUSTOMER 2 as we want to see the last order first we can use
360 --order by while using descending order on the date limiting it to 1 so as to show only the top most row
361 SELECT * FROM Orders WHERE cust_id = 'CU2' ORDER BY date_order DESC LIMIT 1;
362
363 --TO SERACH THE EMPLOYEE WHOSE NAME STARTS WITH A
364 SELECT * FROM Employee WHERE FirstName LIKE 'A%';
365
366 --SELECTING THE MAXIMUM SALARY FROM THE EMPLOYEE name AND ROLE FROM ROLE TABLE
Data Output Explain Messages Notifications
emp_id [PK] character varying (20)
date_of_join date
firstname character varying (30)
lastname character varying (30)
salary real
age integer
phone_1 bigint
phone_2 bigint
role_id character varying (20)
1 E1 2020-09-16 Ayush Sharma 25000 26 7826541253 8526971594 R2

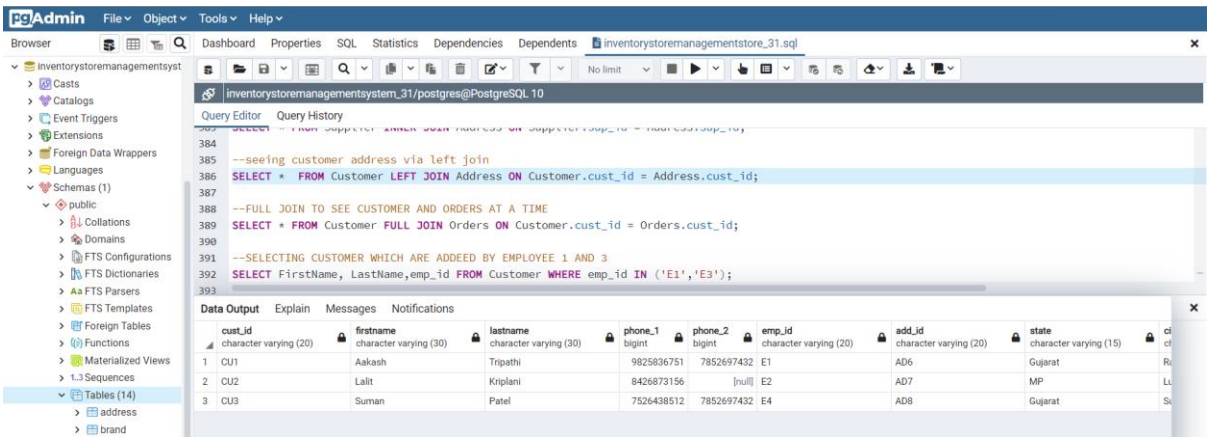

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
364 SELECT * FROM Employee WHERE FirstName LIKE 'A%';
365
366 --SELECTING THE MAXIMUM SALARY FROM THE EMPLOYEE name AND ROLE FROM ROLE TABLE
367 SELECT MAX(SALARY) FROM Employee;
368 SELECT Role.role_name,Employee.FirstName FROM Role,Employee
369 WHERE Role.role_id = (SELECT Role_id FROM Employee WHERE Salary = (SELECT MAX(SALARY) FROM Employee))
370 and Employee.emp_id = (SELECT emp_id FROM Employee WHERE Salary = (SELECT MAX(SALARY) FROM Employee));
371
372 --SHOWING THE PRODUCTS COMING FROM CATEGORY C2 SHOWING IT'S CATEGORY
Data Output Explain Messages Notifications
role_name character varying (20)
firstname character varying (30)
1 Manager Prakash

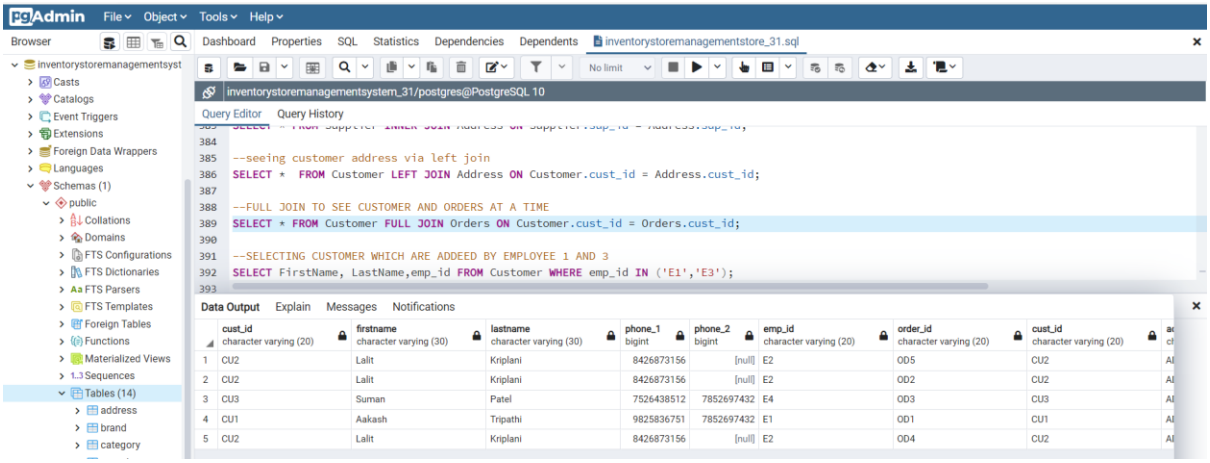

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
369 WHERE Role.role_id = (SELECT Role_id FROM Employee WHERE Salary = (SELECT MAX(SALARY) FROM Employee))
370 and Employee.emp_id = (SELECT emp_id FROM Employee WHERE Salary = (SELECT MAX(SALARY) FROM Employee));
371
372 --SHOWING THE PRODUCTS COMING FROM CATEGORY C2 SHOWING IT'S CATEGORY
373 SELECT Product.prod_name, Product.prod_id, Category.categ_name FROM Product,Category WHERE Category.categ_id = 'C2'
374 and Product.prod_id = (SELECT prod_id FROM Product WHERE categ_id = 'C2');
375
376 --ordering products as per the categ id
377 SELECT * FROM Product ORDER BY CASE WHEN categ_id = 'C3' THEN '1' WHEN categ_id = 'C5' THEN '2' ELSE categ_id END ASC;
378
Data Output Explain Messages Notifications
prod_name character varying (20)
prod_id character varying (20)
categ_name character varying (15)
1 Maaza PR3 Cold Drinks

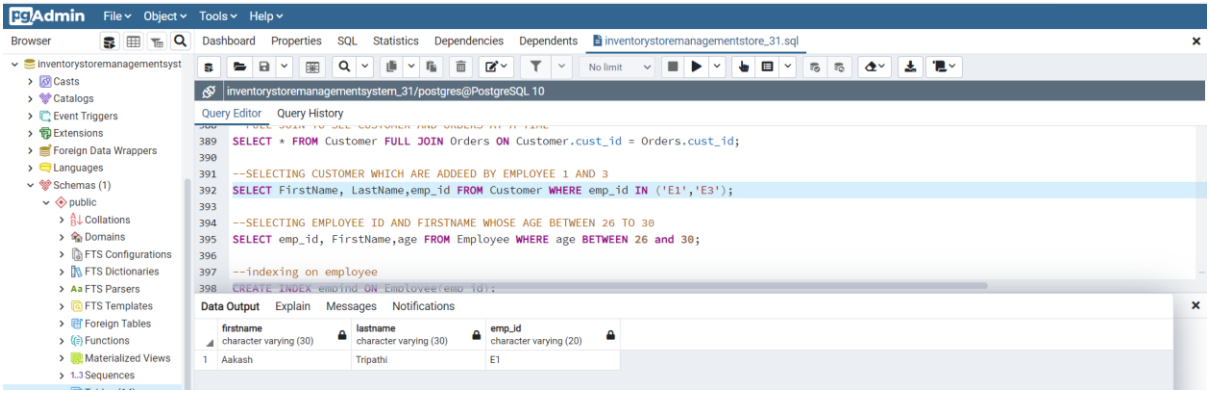

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
373 SELECT Product.prod_name, Product.prod_id, Category.categ_name FROM Product,Category WHERE Category.categ_id = 'C2'
374 and Product.prod_id = (SELECT prod_id FROM Product WHERE categ_id = 'C2');
375
376 --ordering products as per the categ id
377 SELECT * FROM Product ORDER BY CASE WHEN categ_id = 'C3' THEN '1' WHEN categ_id = 'C5' THEN '2' ELSE categ_id END ASC;
378
379 --seeing that for all employee how much does it cost for there salary
380 SELECT SUM(SALARY) FROM Employee;
381
Data Output Explain Messages Notifications
prod_id [PK] character varying (20)
prod_name character varying (20)
unit_price real
prod_quantity integer
categ_id character varying (20)
1 PR1 Chocolate 20 68 C5
2 PR3 Maaza 10 20 C2
3 PR2 Pen 5 130 C4

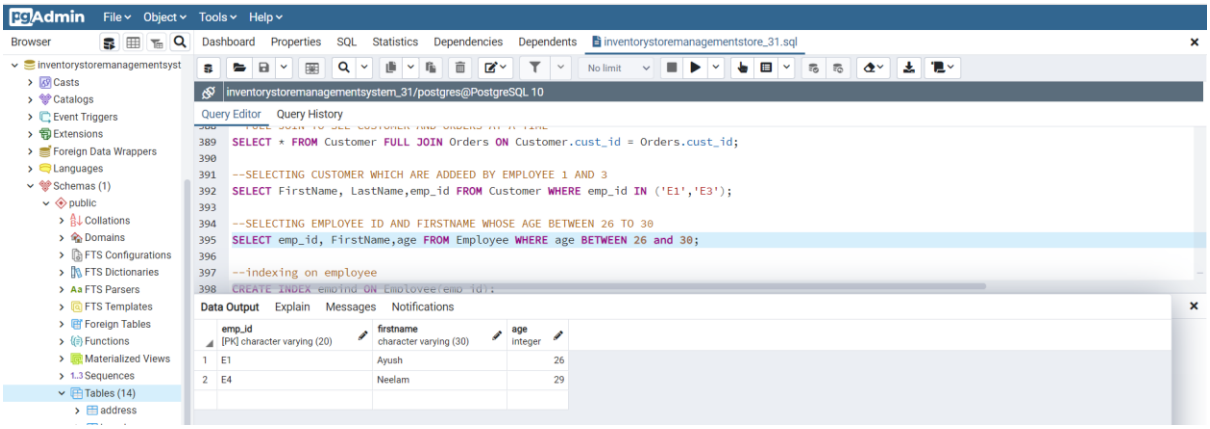

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
377 SELECT * FROM Product ORDER BY CASE WHEN categ_id = 'C3' THEN '1' WHEN categ_id = 'C5' THEN '2' ELSE categ_id END ASC;
378
379 --seeing that for all employee how much does it cost for there salary
380 SELECT SUM(SALARY) FROM Employee;
381
382 --seeing the supplier and address details via inner join
383 SELECT * FROM Supplier INNER JOIN Address ON Supplier.sup_id = Address.sup_id;
384
385 --seeing customer address via left join
Data Output Explain Messages Notifications
sum real
1 130000

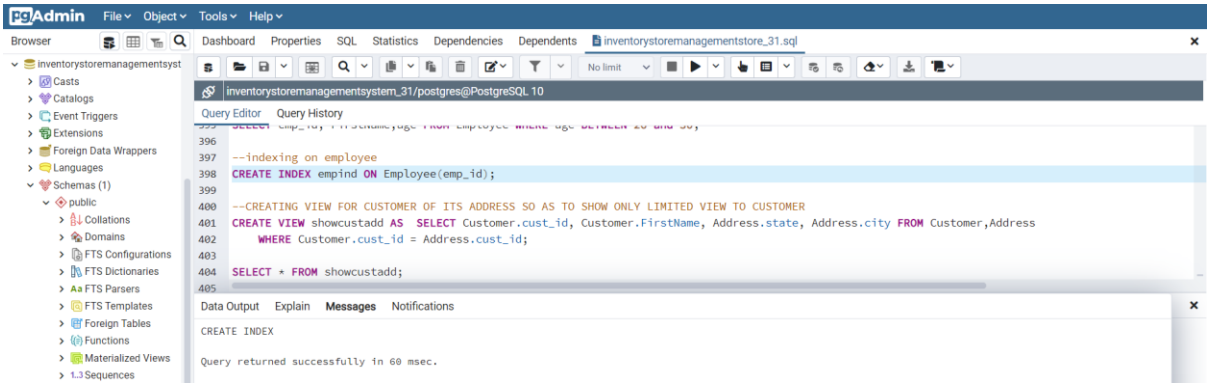

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
377 SELECT * FROM Product ORDER BY CASE WHEN categ_id = 'C3' THEN '1' WHEN categ_id = 'C5' THEN '2' ELSE categ_id END ASC;
378
379 --seeing that for all employee how much does it cost for there salary
380 SELECT SUM(SALARY) FROM Employee;
381
382 --seeing the supplier and address details via inner join
383 SELECT * FROM Supplier INNER JOIN Address ON Supplier.sup_id = Address.sup_id;
384
385 --seeing customer address via left join
Data Output Explain Messages Notifications
sup_id character varying (20)
first_name character varying (30)
lastname character varying (20)
phone_1 bigint
phone_2 bigint
add_id character varying (20)
state character varying (15)
city character varying (15)
1 SU1 Ravindar Sharma 7624584621 8426137942 AD9 Gujarat Rajkot
2 SU2 Divya Chauhan 9826475234 [null] AD10 MP Ujjain
3 SU3 Kamlesh Bhatt 964251732 8426137942 AD11 Gujarat Rajkot

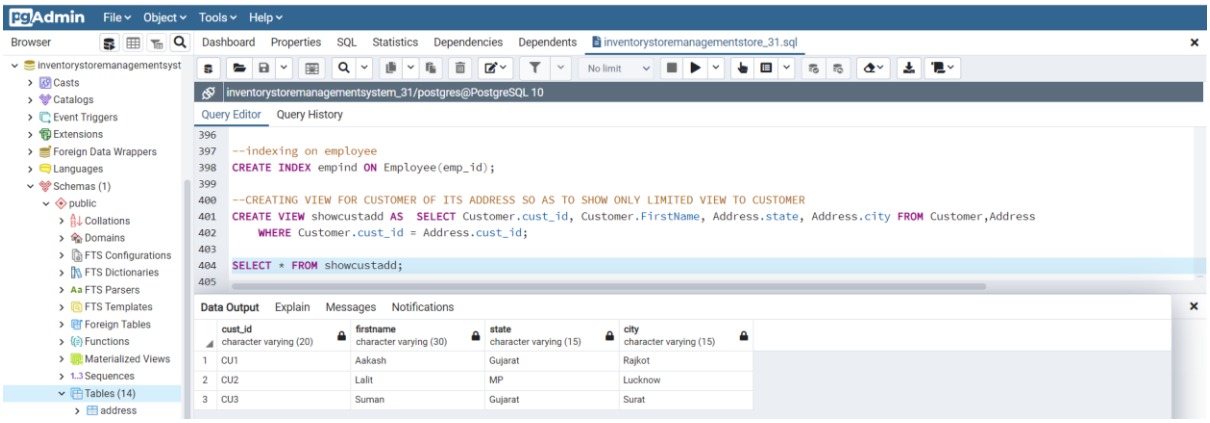

pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
384
385 --seeing customer address via left join
386 SELECT * FROM Customer LEFT JOIN Address ON Customer.cust_id = Address.cust_id;
387
388 --FULL JOIN TO SEE CUSTOMER AND ORDERS AT A TIME
389 SELECT * FROM Customer FULL JOIN Orders ON Customer.cust_id = Orders.cust_id;
390
391 --SELECTING CUSTOMER WHICH ARE ADDEED BY EMPLOYEE 1 AND 3
392 SELECT FirstName, LastName,emp_id FROM Customer WHERE emp_id IN ('E1','E3');
Data Output Explain Messages Notifications
cust_id character varying (20)
firstname character varying (30)
lastname character varying (30)
phone_1 bigint
phone_2 bigint
emp_id character varying (20)
add_id character varying (20)
state character varying (15)
1 CU1 Aakash Tripathi 9825836751 7852697432 E1 AD6 Gujarat
2 CU2 Lalit Kriplani 8426873156 [null] E2 AD7 MP
3 CU3 Suman Patel 7526438512 7852697432 E4 AD8 Gujarat


pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--seeing customer address via left join
SELECT * FROM Customer LEFT JOIN Address ON Customer.cust_id = Address.cust_id;
--FULL JOIN TO SEE CUSTOMER AND ORDERS AT A TIME
SELECT * FROM Customer FULL JOIN Orders ON Customer.cust_id = Orders.cust_id;
--SELECTING CUSTOMER WHICH ARE ADDEED BY EMPLOYEE 1 AND 3
SELECT FirstName, LastName,emp_id FROM Customer WHERE emp_id IN ('E1','E3');
Data Output Explain Messages Notifications
cust_id character varying (20)
firstname character varying (30)
lastname character varying (30)
phone_1 bigint
phone_2 bigint
emp_id character varying (20)
order_id character varying (20)
cust_id character varying (20)
1 CU2 Lalit Kriplani 8426873156 [null] E2 OD5 CU2
2 CU2 Lalit Kriplani 8426873156 [null] E2 OD2 CU2
3 CU3 Suman Patel 7526438512 7852697432 E4 OD3 CU3
4 CU1 Aakash Tripathi 9825836751 7852697432 E1 OD1 CU1
5 CU2 Lalit Kriplani 8426873156 [null] E2 OD4 CU2


pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
SELECT * FROM Customer FULL JOIN Orders ON Customer.cust_id = Orders.cust_id;
--SELECTING CUSTOMER WHICH ARE ADDEED BY EMPLOYEE 1 AND 3
SELECT FirstName, LastName,emp_id FROM Customer WHERE emp_id IN ('E1','E3');
--SELECTING EMPLOYEE ID AND FIRSTNAME WHOSE AGE BETWEEN 26 TO 30
SELECT emp_id, FirstName,age FROM Employee WHERE age BETWEEN 26 and 30;
--indexing on employee
CREATE INDEX empind ON Employee(emp_id);
Data Output Explain Messages Notifications
firstname character varying (30)
lastname character varying (30)
emp_id character varying (20)
1 Aakash Tripathi E1


pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
SELECT * FROM Customer FULL JOIN Orders ON Customer.cust_id = Orders.cust_id;
--SELECTING CUSTOMER WHICH ARE ADDEED BY EMPLOYEE 1 AND 3
SELECT FirstName, LastName,emp_id FROM Customer WHERE emp_id IN ('E1','E3');
--SELECTING EMPLOYEE ID AND FIRSTNAME WHOSE AGE BETWEEN 26 TO 30
SELECT emp_id, FirstName,age FROM Employee WHERE age BETWEEN 26 and 30;
--indexing on employee
CREATE INDEX empind ON Employee(emp_id);
Data Output Explain Messages Notifications
emp_id [PK] character varying (20)
firstname character varying (30)
age integer
1 E1 Ayush 26
2 E4 Neelam 29


pgAdmin
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents inventorystoremanagementstore_31.sql
inventorystoremanagementsystem_31/postgres@PostgreSQL 10
Query Editor Query History
--indexing on employee
CREATE INDEX empind ON Employee(emp_id);
--CREATING VIEW FOR CUSTOMER OF ITS ADDRESS SO AS TO SHOW ONLY LIMITED VIEW TO CUSTOMER
CREATE VIEW showcustadd AS SELECT Customer.cust_id, Customer.FirstName, Address.state, Address.city FROM Customer,Address
WHERE Customer.cust_id = Address.cust_id;
SELECT * FROM showcustadd;
Data Output Explain Messages Notifications
CREATE INDEX
Query returned successfully in 60 msec.



➔ Conclusion:

From the above project I did for the inventory store management system I began with the understanding of inventory store, what may be the general requirement, how can be things be differentiated and all the basic root cause. I have tried meeting every aspect as much as possibility. I learnt that how database can be designed which are the different queries that runs behind when we do some activity. I have written some the queries such that manually no one has to each and every time update the table as the product quantity is supplied, instead of that I have wrote a query which only takes the supply id and will fetch the information that which product is supplied and in which quantity and will increase in total product quantity. Same way I have learnt many other queries running behind while giving order, making payment, observing if the order is completed or not and so on. By the above project I leant how to approach the project, build step by step the ideation and how to implement and update as per requirement.