Installing and Loading with Northwind Database / PostgreSQL

Didem B. Aykurt

Colorado State University Global

MIS541; Data Warehousing in Enterprise Environments

Dr.Peter Salemi

March 26, 2023

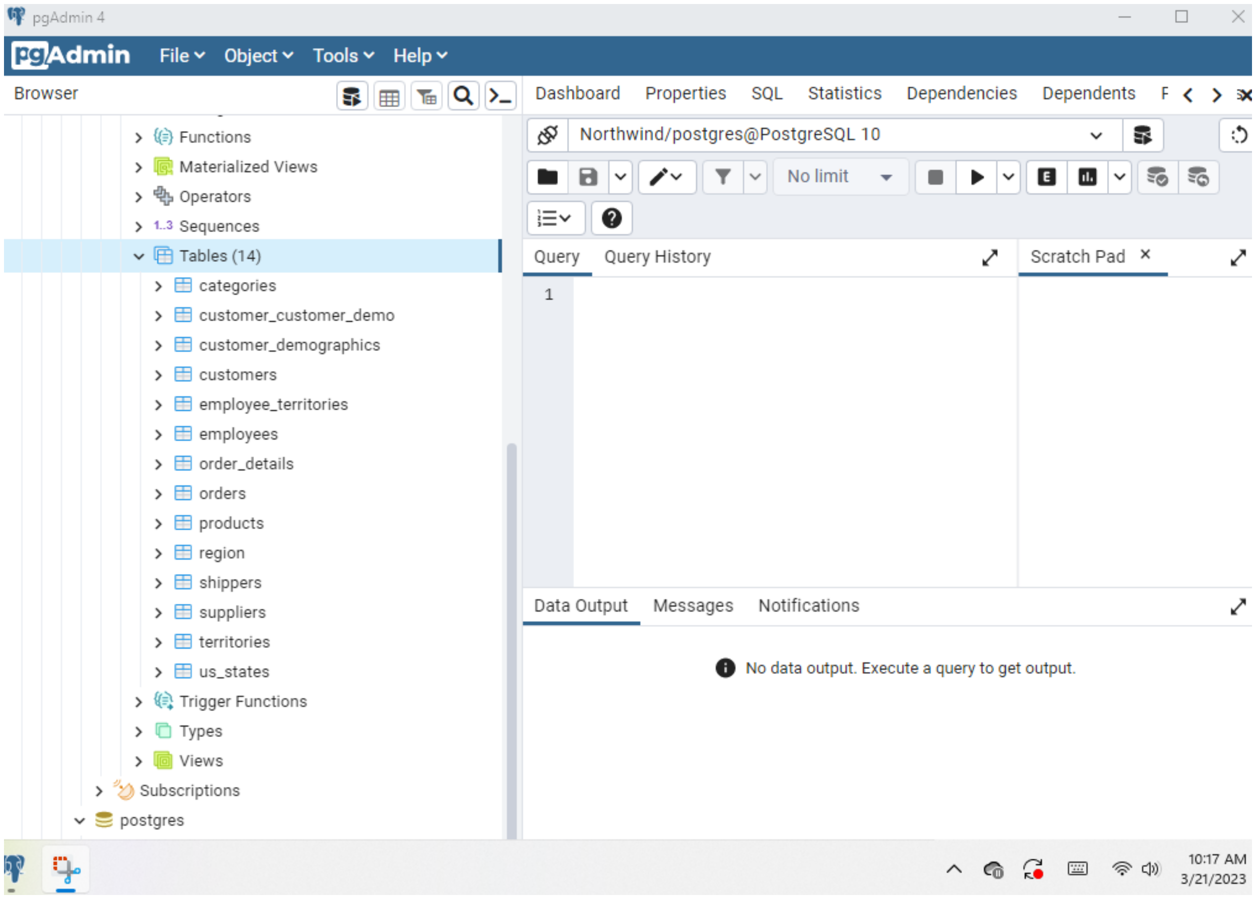
Installing and Loading the Northwind Database into Postgres

Part 1 & Part 2

When do I start to learn new knowledge that comes up with why I need that tool? Where will I use it? What kind of problem would I solve? Why do most companies work with them for the PostgreSQL program? First, I would like to give some minor information about all my questions. The best reason to use PostgreSQL is that it is robust enough to help developers create an open-source object-relational database system for proven architecture, reliability, data integrity, scalability, and building applications and administrators to protect data integrity. Additionally, it is free, open source, and highly extensible. Also, safety and security are the reasons all companies need them. Look at my work; PostgreSQL has many different properties as data type; primitive: integer, string, Boolean, numeric; structure: array, UUID, date/time, range; data integrity; Unique, not null, primary keys, foreign keys, exclusion constraints, and explicit locks, advisor locks. It has an excellent console to manage code, data, and results. I downloaded the Postgres program and then the SQL software language. Open Postgres; right-click on Database 1, New database name Northwind, go to query tool, then a new working page pops up, copy past Script from the code file.

I worked with the Northwind database. Hence, let's dive deep into the Northwind database with the SQL programming language with Postgres. The Northwind database contains detailed company information that helps understand company customers and employees and their performance to measure financial and operational performance, such as what company is selling to whom at what price.

Figure 1: Northwind database tables list on Postgres.



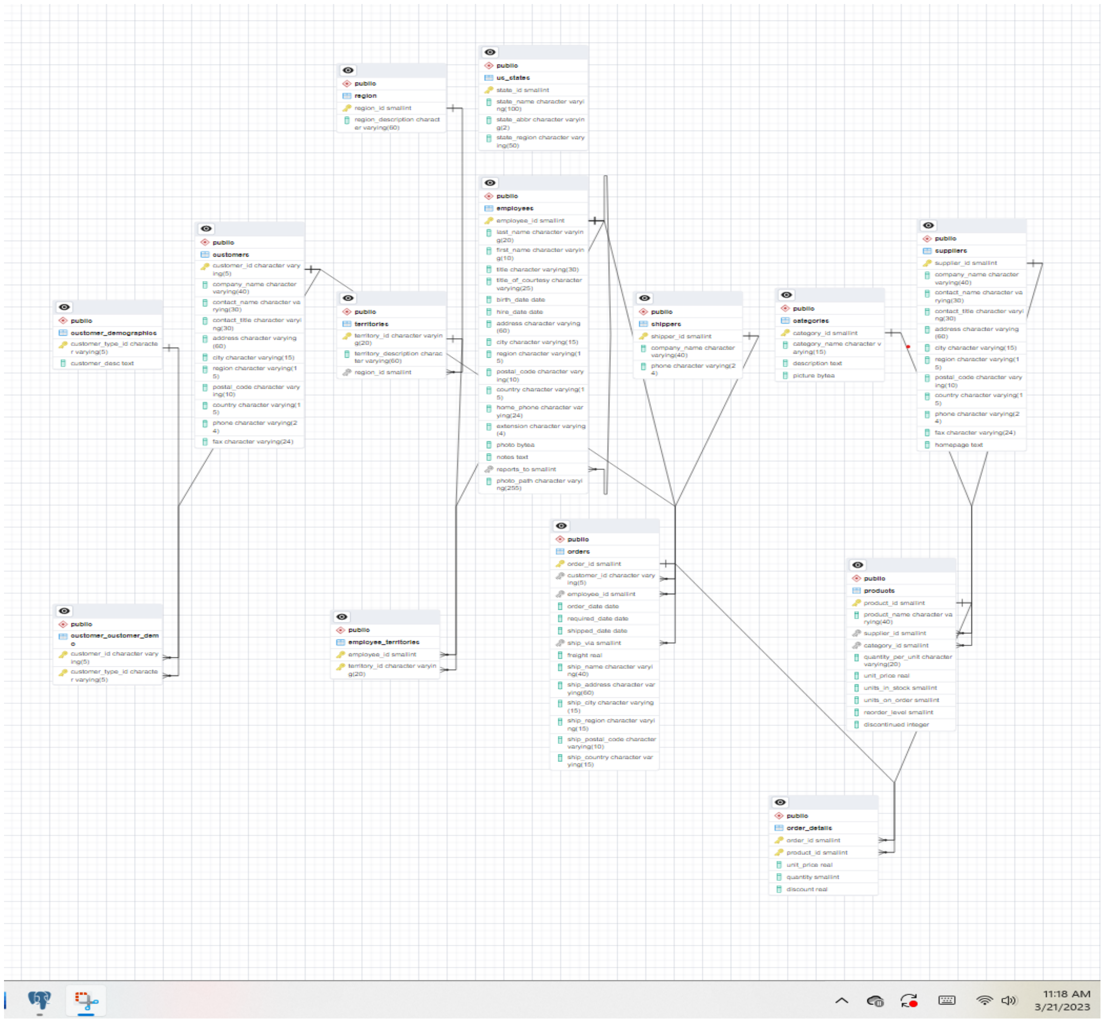
No data output. Execute a query to get output.

Northwind database includes 14 tables containing information about customers, products, employees, and order details. Each table has a primary key containing a unique identifier for rows, such as category_id in the categories table or customer_id in the customer table. Each table has a foreign key containing specific values in a column that must match the values appearing in some row of another table, such as the customer_id foreign key constraint in the orders table that references the customers' table. Below, the list shows each table's primary and foreign keys and relational table. Also, figure 2 shows the Northwind database tree:

Table Name	Number of Variables	Primary Key	Foreign Key	Relation table
categories	4	category_id		products

customer_customer_demo	2	customer_id customer_type_id		customer_demographics customers
customer_demographics	2	Customer_type_id		customer_customer_demo
customers	11	customer_id		customer_customer_demo orders
employee_territories	2	employee_id territory_id		region and employees
employee	18	employee_id	reports_to	employee_territories orders,
Order_detail	5	order_id product_id		Orders, products
orders	14	order_id	customer_id, employee_id, ship_via	Categories, customers employees
products	10	product_id	supplier_id category_id	Orders_details, suppliers, categories
region	2	region_id		territories
Shippers	3	shipper_id		orders
Suppliers	12	supplier_id		products
territories	3	territory_id	region_id	Region, employee_territories
us_states	4	state_id		

Figure 2: Northwind database tree (right-click on Northwind and click Generate ERD).


public
region
region_id smallint
region_description character varying(60)
public
us_states
state_id smallint
state_name character varying(100)
state_abbr character varying(2)
state_region character varying(50)
public
employees
employee_id smallint
last_name character varying(20)
first_name character varying(10)
title character varying(30)
title_of_courtesy character varying(25)
birth_date date
hire_date date
address character varying(60)
city character varying(15)
region character varying(15)
postal_code character varying(10)
country character varying(15)
home_phone character varying(24)
extension character varying(4)
photo bytea
notes text
reports_to smallint
photo_path character varying(255)
public
customers
customer_id character varying(5)
company_name character varying(40)
contact_name character varying(30)
contact_title character varying(30)
address character varying(60)
city character varying(15)
region character varying(15)
postal_code character varying(10)
country character varying(15)
phone character varying(24)
fax character varying(24)
public
customer_demographics
customer_type_id character varying(5)
customer_desc text
public
territories
territory_id character varying(20)
territory_description character varying(60)
region_id smallint
public
shippers
shipper_id smallint
company_name character varying(40)
phone character varying(24)
public
categories
category_id smallint
category_name character varying(15)
description text
picture bytea
public
suppliers
supplier_id smallint
company_name character varying(40)
contact_name character varying(30)
contact_title character varying(30)
address character varying(60)
city character varying(15)
region character varying(15)
postal_code character varying(10)
country character varying(15)
phone character varying(24)
fax character varying(24)
homepage text
public
orders
order_id smallint
customer_id character varying(5)
employee_id smallint
order_date date
required_date date
shipped_date date
ship_via smallint
freight real
ship_name character varying(40)
ship_address character varying(60)
ship_city character varying(15)
ship_region character varying(15)
ship_postal_code character varying(10)
ship_country character varying(15)
public
products
product_id smallint
product_name character varying(40)
supplier_id smallint
category_id smallint
quantity_per_unit character varying(20)
unit_price real
units_in_stock smallint
units_on_order smallint
reorder_level smallint
discontinued integer
public
customer_customer_demo
customer_id character varying(5)
customer_type_id character varying(5)
public
employee_territories
employee_id smallint
territory_id character varying(20)
public
order_details
order_id smallint
product_id smallint
unit_price real
quantity smallint
discount real
11:18 AM
3/21/2023

Let's look at where data comes into the relational database platform or which model or process helps the industry get all transaction details into the relational database. Online transactional processing (OLTP) is correct in all data processing between online banking and ATM transactions, in-store purchases and e-commerce, and hotel and airline bookings that implement large numbers of database transactions by large numbers of orders and people daily. OLTP can insertions, deletions, and update financial and non-financial database transactions like exchanges such as password changes and text messages. All the transaction backups are available at any moment.

Now that we know where data comes from next, all the delivered data should be understandable to business users and provide fast query performance as dimensional modeling describes all the query performance, business user understandability, and resilience to change. Dimensional models are efficiently accurate in the multidimensional database. Particular activities are referred to as online analytical processing (OLAP) cubes.

Thus, I will work on the Northwind database with dimensional modeling to answer my business question: what company is selling to whom at what price?

What is the recommended business process and grain to use as a starting point for a dimensional model?

The order_details table contains the most atomic information, namely one row per order/product combination (the recommended grain to use as a starting point), so we get more information than the individual orders (i.e., how much each product was purchased in each order). Thus, I recommend starting with the order_details table for the fact table. The order table has one row per order, so the product information is missing. Also, the order_details table contains values we would want to aggregate in BI reports (e.g., unit price and quantity).

References

PostgreSQL: The World’s Most Advanced Open-Source Relational Database. https://www.postgresql.org/

What is OLTP? On the IBM website. https://www.ibm.com/topics/oltp

Ralph Kimball & Margy Ross, 2013. The Data Warehouse Toolkit The Definitive Guide to Dimensional Modeling. 3rd edition.