Hands-on SQL Exercises with PostgreSQL

Emmanuel BRUNO
Université de Toulon, LIS UMR CNRS 7020
emmanuel.bruno@univ-tln.fr

2026-10-05

Table des matières

1 Hands-on Practice 1
1.1 Objectives 1
1.2 Additional Tips 1
1.3 Example Tables 1
1.3.a Customers Table 1
1.3.b Products Table 1
1.3.c Orders Table 1
1.3.d Categories Table 2
1.3.e ProductCategories Table 2
1.4 Diagram 2
1.5 DDL Queries to Create and Populate the Database 2
1.6 Hands-on Practice Queries 3
1.7 Additional Query Ideas 4
2 Move to Your Project 4

1 Hands-on Practice

1.1 Objectives

- Understand how to create and manage tables in PostgreSQL.
- Learn to insert, update, delete, and query data.
- Practice using joins and other advanced SQL features.

1.2 Additional Tips

- Install and use a real PostgreSQL Database with a GUI like pgAdmin.
- Use an online test tool like SQL Fiddle.

1.3 Example Tables

1.3.a Customers Table

customer_id	name	email	phone_number	country
1	John Doe	john@example.com	123-456-7890	USA
2	Jane Smith	jane@example.com	987-654-3210	USA
3	Pierre Dupont	pierre@example.fr	555-987-6543	France

1.3.b Products Table

product_id	product_name	price
101	Laptop	999.99
102	Smartphone	499.99

1.3.c Orders Table

order_id	customer_id	product_id	quantity	status
1001	1	101	2	Pending
1002	2	102	1	Shipped

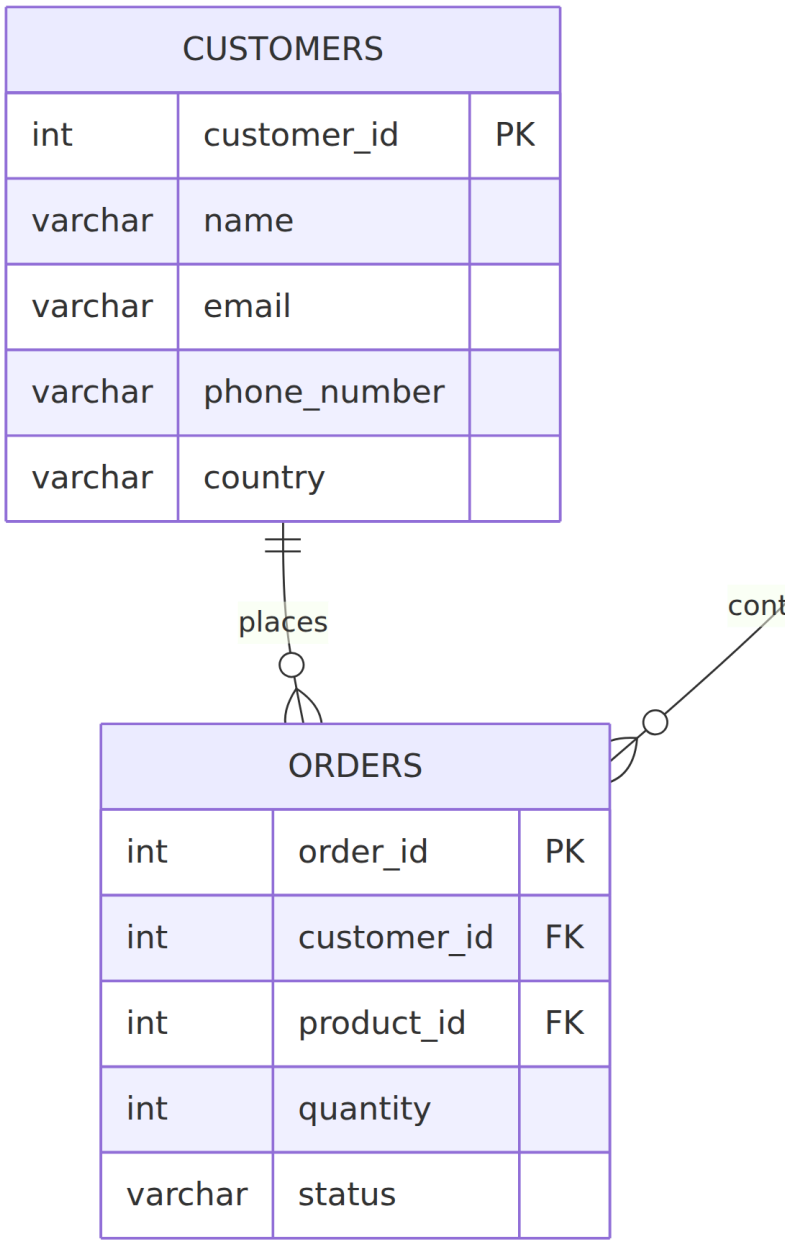
1.3.d Categories Table

category_id	category_name
1	Electronics
2	Furniture

1.3.e ProductCategories Table

product_id	category_id
101	1
102	1

1.4 Diagram



1.5 DDL Queries to Create and Populate the Database

```
-- Create Customers Table
CREATE TABLE customers (
    customer_id SERIAL PRIMARY KEY,
    name VARCHAR(100),
```

```
    email VARCHAR(100),
    phone_number VARCHAR(15),
    country VARCHAR(50)
);

-- Create Products Table
CREATE TABLE products (
    -- Uncomment to use generated Id
    -- product_id SERIAL PRIMARY KEY,
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    price DECIMAL(10, 2)
);

-- Create Orders Table
CREATE TABLE orders (
    order_id SERIAL PRIMARY KEY,
        customer_id INT REFERENCES
customers(customer_id),
        product_id INT REFERENCES
products(product_id),
    quantity INT,
    status VARCHAR(50)
);

-- Create Categories Table
CREATE TABLE categories (
    category_id SERIAL PRIMARY KEY,
    category_name VARCHAR(100)
);

-- Create ProductCategories Table
CREATE TABLE product_categories (
        product_id INT REFERENCES
products(product_id),
        category_id INT REFERENCES
categories(category_id),
        PRIMARY KEY (product_id,
category_id)
);

-- Insert Data into Customers Table
INSERT INTO customers (name, email,
phone_number, country) VALUES
('John Doe', 'john@example.com',
'123-456-7890', 'USA'),
('Jane Smith', 'jane@example.com',
'987-654-3210', 'USA'),
('Pierre Dupont',
'pierre@example.fr', '555-987-6543',
'France');

-- Insert Data into Products Table
INSERT INTO products (product_id,
product_name, price) VALUES
(101, 'Laptop', 999.99),
(102, 'Smartphone', 499.99);

-- Insert Data into Orders Table
INSERT INTO orders (customer_id,
product_id, quantity, status) VALUES
(1, 101, 2, 'Pending'),
(2, 102, 1, 'Shipped');

-- Insert Data into Categories Table
INSERT INTO categories
(category_name) VALUES
('Electronics'),
('Furniture');

-- Insert Data into ProductCategories
Table
INSERT INTO product_categories
(product_id, category_id) VALUES
(101, 1),
(102, 1);
```

1.6 Hands-on Practice Queries

- **Querying Product Information**:
  - Write a query to retrieve all products with their prices.

```
#| code-fold: true
#| code-summary: "Show the code"
SELECT product_name, price FROM
products;
```

- **Adding a New Order**:
  - Write a query to add a new order for a customer.

```
#| code-fold: true
#| code-summary: "Show the code"
INSERT INTO orders (customer_id,
product_id, quantity, status)
VALUES (1, 101, 2, 'Pending');
```

- **Updating Order Status**:
  - Write a query to update the status of an existing order.

```
#| code-fold: true
#| code-summary: "Show the code"
SELECT * FROM orders;
UPDATE orders SET status = 'Shipped'
WHERE order_id = 1001;
SELECT * FROM orders;
```

- **Deleting a Customer Record**:
  - Write a query to delete a customer record from the database.
  - What is the problem ? How to solve (two solutions) ?

```
#| code-fold: true
#| code-summary: "Show the code"
DELETE FROM customers WHERE
customer_id = 1;
```

- **Querying Products by Category**:
  - Write a query to retrieve all products along with their categories.

```
#| code-fold: true
#| code-summary: "Show the code"
SELECT
p.product_name, c.category_name
FROM products p
JOIN product_categories pc ON
p.product_id = pc.product_id
JOIN categories c ON pc.category_id
= c.category_id;
```

1.7 Additional Query Ideas

- **Querying Customers by Country**:
  - Write a query to retrieve all customers from a specific country.
- **Calculating Total Order Value**:
  - Write a query to calculate the total value of all orders for a specific customer.
- **Finding Products with Low Stock**:
  - Write a query to find all products with a stock quantity below a certain threshold.
- **Listing Orders with Customer Information**:
  - Write a query to list all orders along with the customer information for each order.
- **Updating Product Prices**:
  - Write a query to increase the price of all products in a specific category by a certain percentage.

2 Move to Your Project

Now, apply this to your Team Project. Refer to the Final Project and Review for more details. Create a new database for your project, add two or more tables to store some of the data (e.g., products, customers, etc.), and include some useful queries.