

ISP_Part2

This folder contains a collection of SQL scripts and Python scripts designed for managing a supply chain database. The project illustrates the process of creating and manipulating a database for supply chain management, including tasks such as database creation, schema execution, data insertion, fragmentation, and replication. It's structured to be easily understood and executed, making it ideal for educational purposes or as a base for more complex projects.

Contents

- Python Scripts:** Used for creating the database, connecting to it, executing SQL schema scripts, inserting data, performing database fragmentation, and database replication.
- SQL Scripts:** Contains the SQL code for creating tables (`main_schema.sql`) and inserting sample data (`insert_data.sql`). These tables include Products, Warehouses, Suppliers, Orders, OrderDetails, Shipments, and InventoryLevels.

Setup and Execution

Prerequisites

- Python 3.x
- PostgreSQL Database Server
- psycopg2 library (Python PostgreSQL adapter)

Installation Steps

- 1. Install psycopg2:** This Python library is essential for connecting to and interacting with PostgreSQL databases. Install it using pip:

```
pip install psycopg2
```

- 2. PostgreSQL Server:** Ensure you have PostgreSQL server installed and running on your system.

How to Use

- 1. Clone the Repository:** Clone this repository to your local machine.
- 2. Database Creation:** Run the `create_database()` function from the Python script to create a new database for the supply chain management.
- 3. Schema Execution:** Execute the `execute_schema_script()` function to create the necessary tables defined in the `main_schema.sql`.
- 4. Data Insertion:** Use the `execute_insert_script()` function to insert sample data into the database from `insert_data.sql`.

5. **Database Fragmentation:** The `perform_fragmentation()` function is used to fragment the database into multiple tables based on specific attributes like product categories and warehouse locations.
6. **Database Replication:** Finally, the `replication()` function can be used to create a backup of the database.

Database Structure

Tables

- **Products:** Stores information about the products.
- **Warehouses:** Information about warehouses.
- **Suppliers:** Details of suppliers.
- **Orders:** Customer orders.
- **OrderDetails:** Details of each order.
- **Shipments:** Information about product shipments.
- **InventoryLevels:** Inventory levels in each warehouse.

Fragmentation

Database fragmentation is a process of dividing a database into smaller, more manageable segments. This technique is particularly beneficial for large databases, where managing and querying a vast amount of data can be challenging. In this project, fragmentation is based on specific criteria such as product categories and warehouse locations, leading to more efficient data management.

Replication

Database replication involves creating a copy of the database. This copy can serve multiple purposes: it acts as a backup for recovery, aids in load balancing, and allows for running complex queries without impacting the primary database's performance.

Implementation

Purpose

- **Enhanced Performance:** Improves query performance by reducing the size of the data set being queried.
- **Efficient Management:** Simplifies management by dividing the database into logically related parts.

Implementation

- **Category-Based Fragmentation:** Products are divided into different tables based on their categories.
- **Location-Based Fragmentation:** Warehouses are segmented into distinct tables based on their locations.

Usage

Fragmentation is implemented in the Python script where specific SQL queries are used to create new tables based on the predefined criteria (categories and locations).

Replication

Purpose

- **Data Redundancy:** Ensures data availability in case of primary database failure.
- **Load Balancing:** Distributes query load across multiple database copies.
- **Analytics and Reporting:** Facilitates complex queries and reporting tasks without affecting the primary database.

Implementation

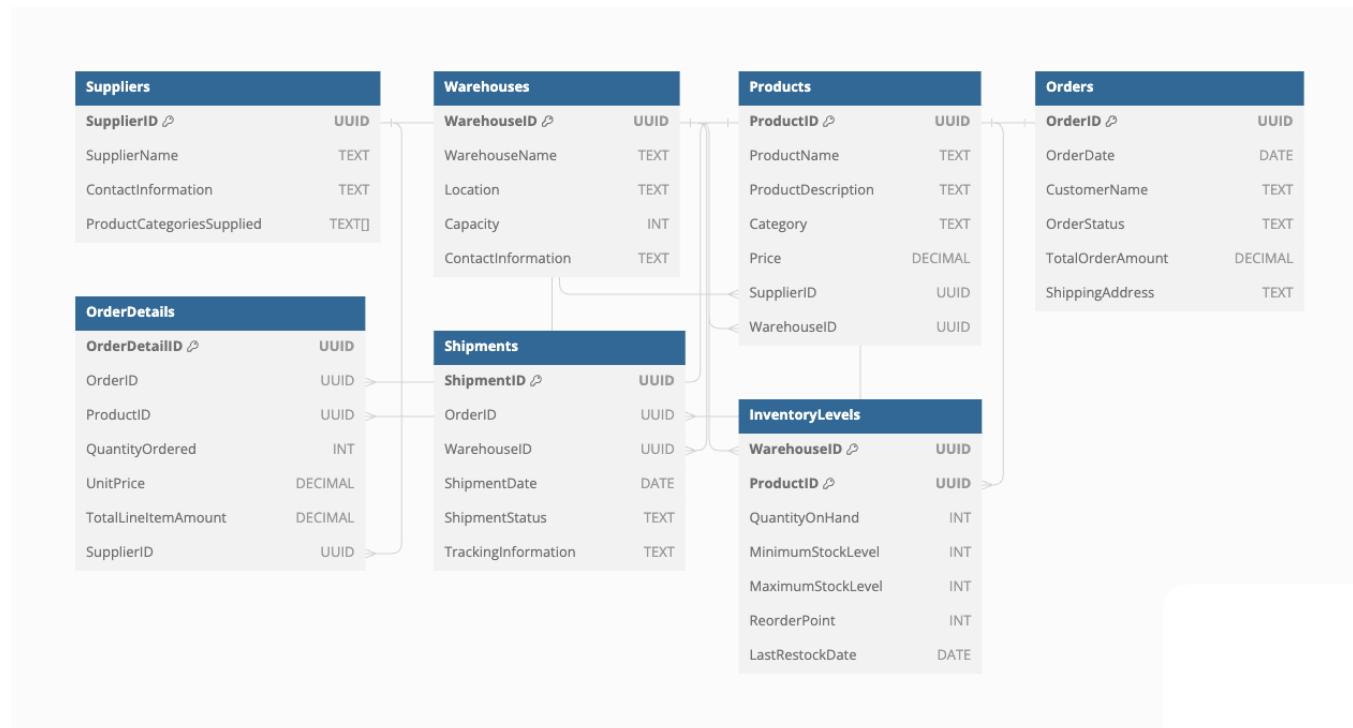
- **Timestamped Backup:** The replication process creates a timestamped backup of the database, which is essential for tracking changes and ensuring data integrity.

Usage

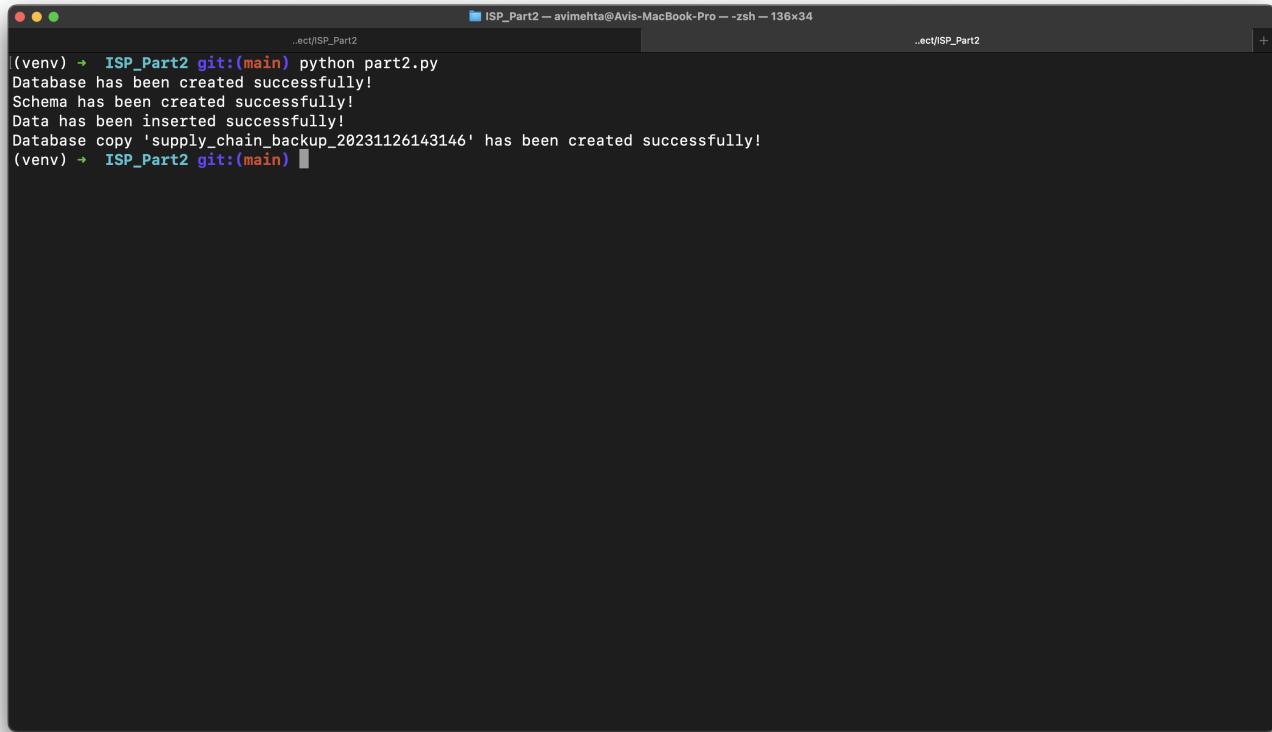
The replication process is scripted in Python, where a new database copy is created using the current database as a template. This process is timestamped to maintain version control.

Schema and Output Screenshots

Database Schema

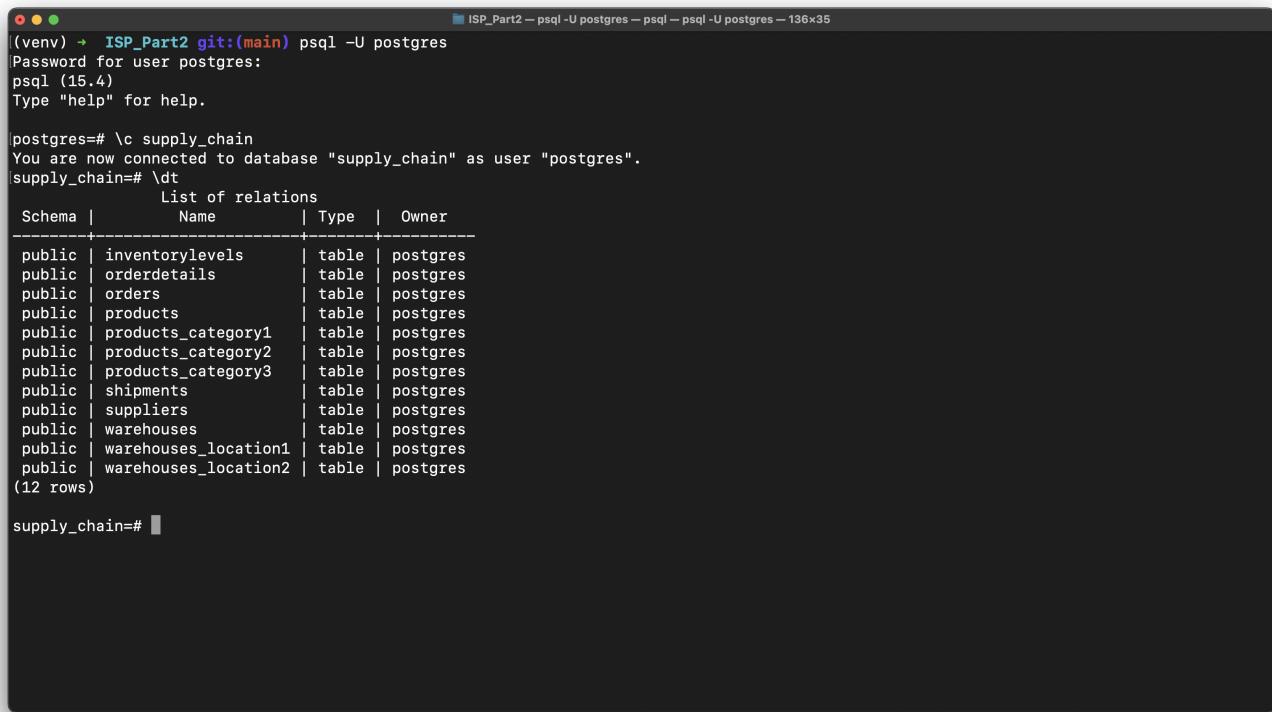


Python Script



```
(venv) + ISP_Part2 git:(main) python part2.py
Database has been created successfully!
Schema has been created successfully!
Data has been inserted successfully!
Database copy 'supply_chain_backup_20231126143146' has been created successfully!
(venv) + ISP_Part2 git:(main)
```

PostgreSQL Connection



```
(venv) + ISP_Part2 git:(main) psql -U postgres
Password for user postgres:
psql (15.4)
Type "help" for help.

postgres=# \c supply_chain
You are now connected to database "supply_chain" as user "postgres".
supply_chain=# \dt
      List of relations
 Schema |        Name         | Type  | Owner
-----+---------------------+-----+
 public | inventorylevels   | table | postgres
 public | orderdetails      | table | postgres
 public | orders            | table | postgres
 public | products          | table | postgres
 public | products_category1| table | postgres
 public | products_category2| table | postgres
 public | products_category3| table | postgres
 public | shipments         | table | postgres
 public | suppliers         | table | postgres
 public | warehouses        | table | postgres
 public | warehouses_location1| table | postgres
 public | warehouses_location2| table | postgres
(12 rows)

supply_chain=#
```

Schema Execution and Fragmentation

```
ISP_Part2 - psql -U postgres - psql - psql -U postgres -- 136x35

|supply_chain=# SELECT * FROM PRODUCTS;
+-----+-----+-----+-----+-----+-----+
| productid | productname | productdescription | category | price | supplierid |
+-----+-----+-----+-----+-----+-----+
| 550e8400-e29b-41d4-a716-446655440001 | Product1 | Description1 | Category1 | 20.5 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
| 9d8a6ad0-4ca6-49a3-83d2-096b01abdc91 | Product2 | Description2 | Category2 | 30.75 | baa74302-7c5e-4174-8364-9d885e8d3ef8
| 8fcfc45f0-5e58-44f6-96e2-4fb6b3d79d29 | Product3 | Description3 | Category1 | 15.0 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
(3 rows)

|supply_chain=# SELECT * FROM ORDERS;
+-----+-----+-----+-----+-----+-----+
| orderid | orderdate | customername | orderstatus | totalorderamount | shippingaddress |
+-----+-----+-----+-----+-----+-----+
| fd8e7ed2-8cf8-4cd8-836f-774d3557d26b | 2023-11-25 | Customer1 | Shipped | 100.0 | Address1
| 48f67e3b-981d-4e2b-b373-303f3423e1d0 | 2023-11-26 | Customer2 | Processing | 75.5 | Address2
(2 rows)

|supply_chain=# SELECT * FROM PRODUCTS_CATEGORY1;
+-----+-----+-----+-----+-----+-----+
| productid | productname | productdescription | category | price | supplierid |
+-----+-----+-----+-----+-----+-----+
| 550e8400-e29b-41d4-a716-446655440001 | Product1 | Description1 | Category1 | 20.5 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
| 8fcfc45f0-5e58-44f6-96e2-4fb6b3d79d29 | Product3 | Description3 | Category1 | 15.0 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
(2 rows)

|supply_chain=# SELECT * FROM PRODUCTS_CATEGORY2;
+-----+-----+-----+-----+-----+-----+
| productid | productname | productdescription | category | price | supplierid |
+-----+-----+-----+-----+-----+-----+
| 9d8a6ad0-4ca6-49a3-83d2-096b01abdc91 | Product2 | Description2 | Category2 | 30.75 | baa74302-7c5e-4174-8364-9d885e8d3ef8
(1 row)

|supply_chain=# SELECT * FROM WAREHOUSES;
+-----+-----+-----+-----+-----+
| warehouseid | warehousename | location | capacity | contactinformation |
+-----+-----+-----+-----+-----+
| b8b17a8d-64c0-49a2-9c56-4c6c3c1142fe | Warehouse1 | Location1 | 1000 | Contact1
| e8266422-cc0d-42a9-9d82-dbdcc3fb7b1d | Warehouse2 | Location2 | 1500 | Contact2
(2 rows)
```

Fragmentation Tables

```
ISP_Part2 - psql -U postgres - psql - psql -U postgres -- 136x35

(2 rows)

|supply_chain=# SELECT * FROM PRODUCTS_CATEGORY1;
+-----+-----+-----+-----+-----+-----+
| productid | productname | productdescription | category | price | supplierid |
+-----+-----+-----+-----+-----+-----+
| 550e8400-e29b-41d4-a716-446655440001 | Product1 | Description1 | Category1 | 20.5 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
| 8fcfc45f0-5e58-44f6-96e2-4fb6b3d79d29 | Product3 | Description3 | Category1 | 15.0 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
(2 rows)

|supply_chain=# SELECT * FROM PRODUCTS_CATEGORY2;
+-----+-----+-----+-----+-----+-----+
| productid | productname | productdescription | category | price | supplierid |
+-----+-----+-----+-----+-----+-----+
| 9d8a6ad0-4ca6-49a3-83d2-096b01abdc91 | Product2 | Description2 | Category2 | 30.75 | baa74302-7c5e-4174-8364-9d885e8d3ef8
(1 row)

|supply_chain=# SELECT * FROM WAREHOUSES;
+-----+-----+-----+-----+-----+
| warehouseid | warehousename | location | capacity | contactinformation |
+-----+-----+-----+-----+-----+
| b8b17a8d-64c0-49a2-9c56-4c6c3c1142fe | Warehouse1 | Location1 | 1000 | Contact1
| e8266422-cc0d-42a9-9d82-dbdcc3fb7b1d | Warehouse2 | Location2 | 1500 | Contact2
(2 rows)

|supply_chain=# SELECT * FROM WAREHOUSES_LOCATION1;
+-----+-----+-----+-----+-----+
| warehouseid | warehousename | location | capacity | contactinformation |
+-----+-----+-----+-----+-----+
| b8b17a8d-64c0-49a2-9c56-4c6c3c1142fe | Warehouse1 | Location1 | 1000 | Contact1
(1 row)

|supply_chain=# SELECT * FROM WAREHOUSES_LOCATION2;
+-----+-----+-----+-----+-----+
| warehouseid | warehousename | location | capacity | contactinformation |
+-----+-----+-----+-----+-----+
| e8266422-cc0d-42a9-9d82-dbdcc3fb7b1d | Warehouse2 | Location2 | 1500 | Contact2
(1 row)

supply_chain=#
```

Replication and Backup

```
ISP_Part2 - psql -U postgres - psql - psql -U postgres -- 136x35

public | inventorylevels      | table | postgres
public | orderdetails         | table | postgres
public | orders                | table | postgres
public | products              | table | postgres
public | products_category1   | table | postgres
public | products_category2   | table | postgres
public | products_category3   | table | postgres
public | shipments             | table | postgres
public | suppliers             | table | postgres
public | warehouses            | table | postgres
public | warehouses_location1 | table | postgres
public | warehouses_location2 | table | postgres
(12 rows)

|supply_chain=# \l
                                         List of databases
   Name    | Owner | Encoding | Collate | Ctype | ICU Locale | Locale Provider | Access privileges
---+-----+-----+-----+-----+-----+-----+-----+
midterm | postgres | UTF8     | C       | C     |           | libc
postgres | postgres | UTF8     | C       | C     |           | libc
supply_chain | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231125213753 | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231125213806 | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231125214043 | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231125214046 | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231126143146 | postgres | UTF8     | C       | C     |           | libc
template0 | postgres | UTF8     | C       | C     |           | libc
                                         +c/postgres
                                         +postgres=CTc/postgres
template1 | postgres | UTF8     | C       | C     |           | libc
                                         +c/postgres
                                         +postgres=CTc/postgres
test    | postgres | UTF8     | C       | C     |           | libc
(11 rows)

supply_chain=# 
```

Replicated Database

```
ISP_Part2 - psql -U postgres - psql - psql -U postgres -- 136x35

postgres | postgres | UTF8     | C       | C     |           | libc
supply_chain | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231125213753 | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231125213806 | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231125214043 | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231125214046 | postgres | UTF8     | C       | C     |           | libc
supply_chain_backup_20231126143146 | postgres | UTF8     | C       | C     |           | libc
template0 | postgres | UTF8     | C       | C     |           | libc
                                         +c/postgres
                                         +postgres=CTc/postgres
template1 | postgres | UTF8     | C       | C     |           | libc
                                         +c/postgres
                                         +postgres=CTc/postgres
test    | postgres | UTF8     | C       | C     |           | libc
(11 rows)

supply_chain=# \c supply_chain_backup_20231126143146
You are now connected to database "supply_chain_backup_20231126143146" as user "postgres".
|supply_chain_backup_20231126143146=# \dt
          List of relations
 Schema |        Name        | Type | Owner
---+-----+-----+-----+
public | inventorylevels | table | postgres
public | orderdetails   | table | postgres
public | orders          | table | postgres
public | products         | table | postgres
public | products_category1 | table | postgres
public | products_category2 | table | postgres
public | products_category3 | table | postgres
public | shipments        | table | postgres
public | suppliers         | table | postgres
public | warehouses        | table | postgres
public | warehouses_location1 | table | postgres
public | warehouses_location2 | table | postgres
(12 rows)

supply_chain_backup_20231126143146=# 
```

Replicated Tables

```
ISP_Part2 - psql -U postgres -psql -psql -U postgres - 136x35
You are now connected to database "supply_chain_backup_20231126143146" as user "postgres".
[supply_chain_backup_20231126143146=# \dt
      List of relations
 Schema |        Name         | Type  | Owner
-----+---------------------+-----+
 public | inventorylevels   | table | postgres
 public | orderdetails     | table | postgres
 public | orders           | table | postgres
 public | products          | table | postgres
 public | products_category1 | table | postgres
 public | products_category2 | table | postgres
 public | products_category3 | table | postgres
 public | shipments         | table | postgres
 public | suppliers         | table | postgres
 public | warehouses        | table | postgres
 public | warehouses_location1 | table | postgres
 public | warehouses_location2 | table | postgres
(12 rows)

[supply_chain_backup_20231126143146=# SELECT * FROM PRODUCTS;
      productid       | productname | productdescription | category | price | supplierid
-----+-----+-----+-----+-----+-----+
 550e8400-e29b-41d4-a716-44665540001 | Product1    | Description1      | Category1 | 20.5 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
 9d8a6ad0-4ca6-49a3-83d2-096b01abd91 | Product2    | Description2      | Category2 | 30.75 | baa74302-7c5e-4174-8364-9d885e8d3ef8
 8fcf45f0-5e58-44f6-96e2-4fb6b3d79d29 | Product3    | Description3      | Category1 | 15.0 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
(3 rows)

[supply_chain_backup_20231126143146=# SELECT * FROM PRODUCTS_CATEGORY1;
      productid       | productname | productdescription | category | price | supplierid
-----+-----+-----+-----+-----+-----+
 550e8400-e29b-41d4-a716-44665540001 | Product1    | Description1      | Category1 | 20.5 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
 8fcf45f0-5e58-44f6-96e2-4fb6b3d79d29 | Product3    | Description3      | Category1 | 15.0 | 8a3e5d56-7b70-4a6f-a9c0-8b0c0bb1c063
(2 rows)

[supply_chain_backup_20231126143146=# ]
```