Data Warehousing & Integration

IE 6750

FALL 2024

[Book Sales and Inventory Management System]

Milestone 2

Group 10

Sri Sai Prabhath Reddy Gudipalli

Avani Kala

gudipalli.s@northeastern.edu

kala.a@northeastern.edu

Submission Date: 10/09/2024

DataSource:
https://github.com/bbrumm/databasestar/tree/main/sample_databases/sample_db_gravity/gravity_sqlserver

Introduction:
The purpose of this proposal is to outline the design of a data warehouse that supports the Book Sales and Inventory Management System. This design will facilitate efficient data storage, retrieval, and analysis, enabling informed decision-making and strategic insights into sales, inventory, and customer behavior.
We propose implementing a star schema as our data warehouse design. This will consist of a fact table at the center that captures transactional data, surrounded by dimension tables that provide contextual details. This structure is optimal for ROLAP as it leverages relational databases to store and query large volumes of data efficiently.

Architecture Overview:
Fact Tables:

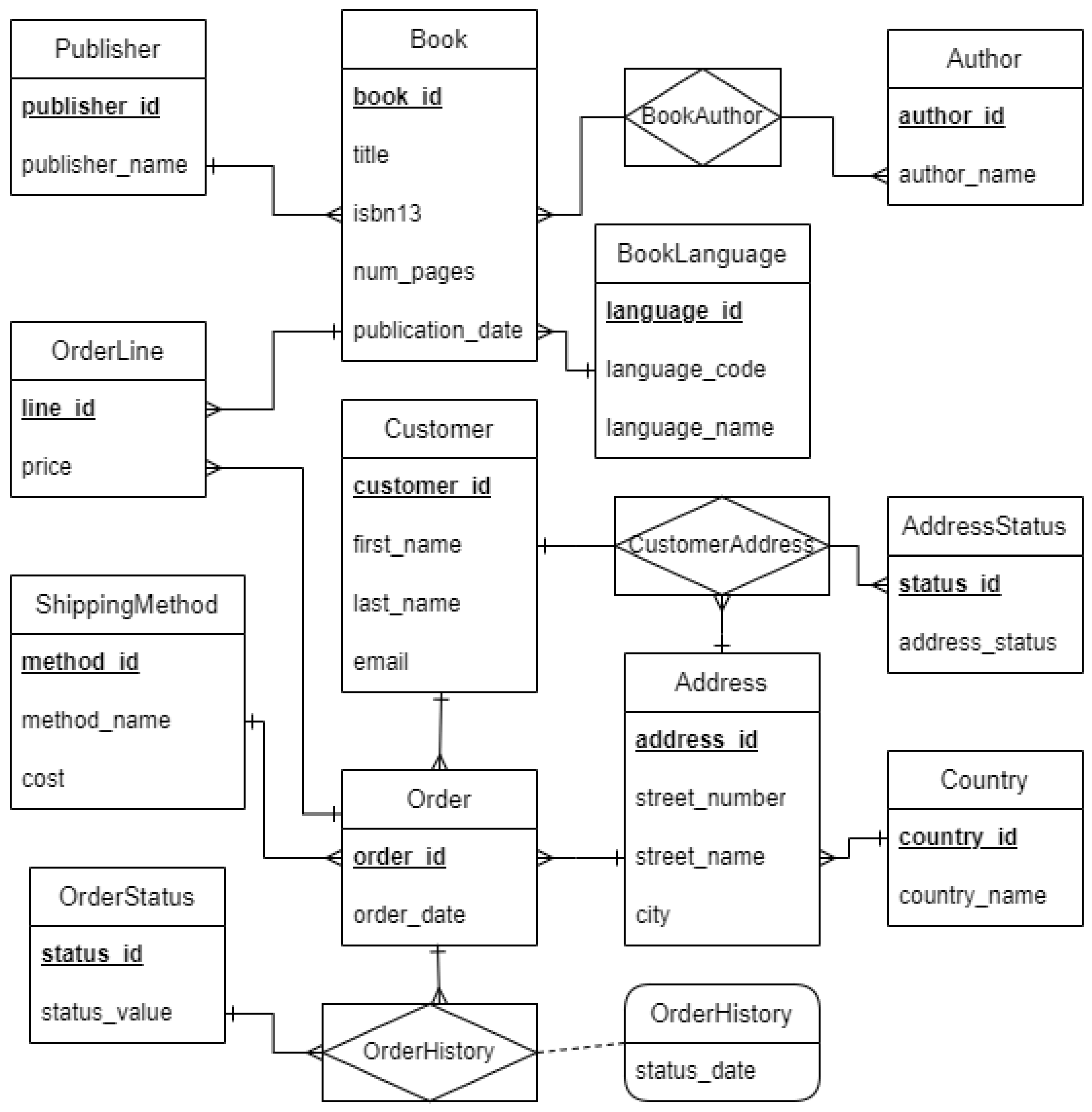
- **OrderFact:** Contains measures such as total sales, quantity sold, and order count, with foreign keys referencing dimension tables.
- **OrderLineFact:** Details each line item within an order, allowing for detailed sales analysis.

Dimension Tables:

- **CustomerDimension:** Stores customer details (e.g., ID, name, email).
- **BookDimension:** Contains book details (e.g., ISBN, title, author, genre).
- **PublisherDimension:** Includes publisher information (e.g., publisher ID, name).
- **TimeDimension:** Facilitates time-based analysis, storing attributes such as year, quarter, month, and day.
- **ShippingMethodDimension:** Details different shipping methods available.

- **AddressDimension**: Captures address details for customers, which can aid in geographical analysis.

ERD:



Relational:

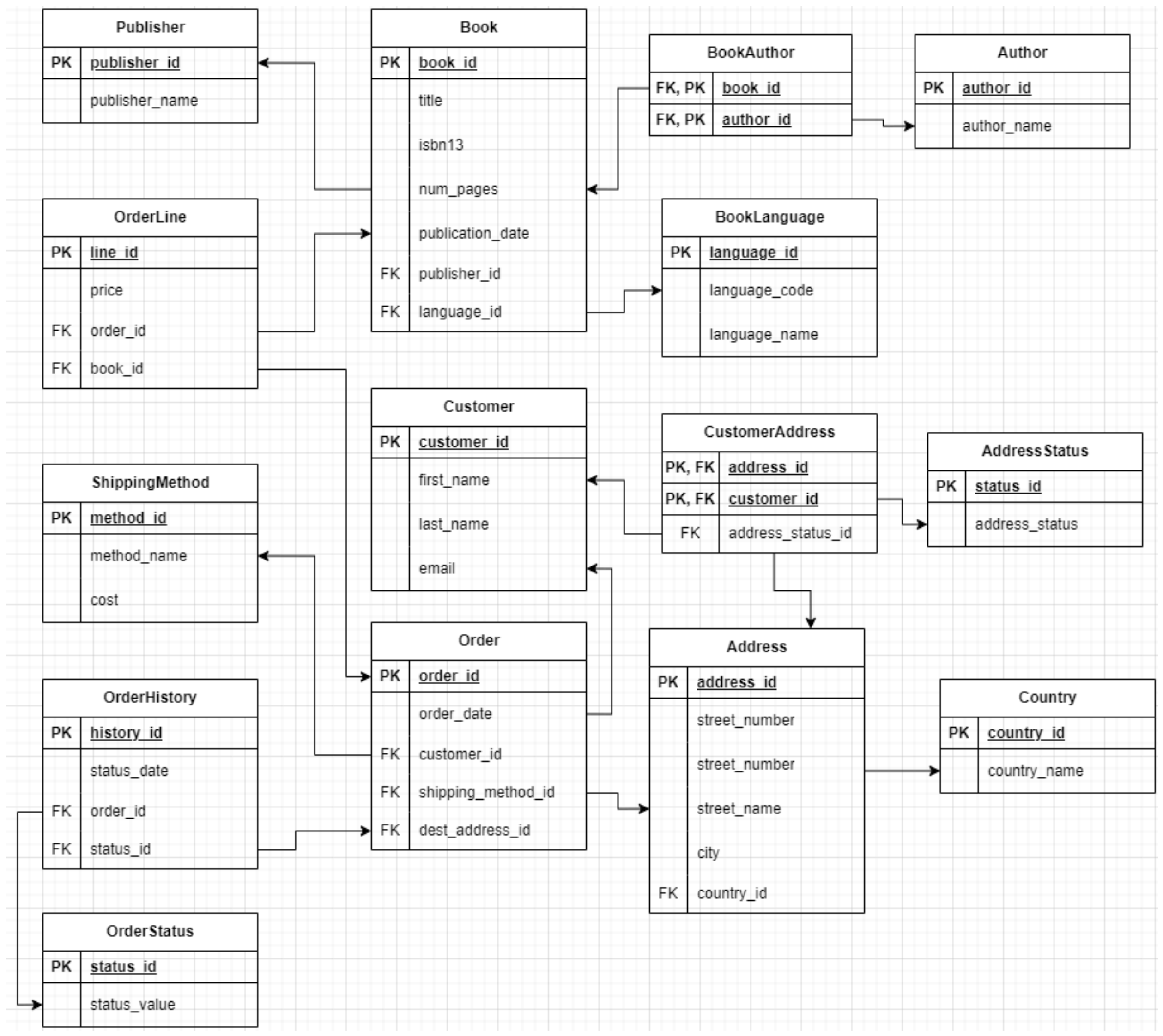
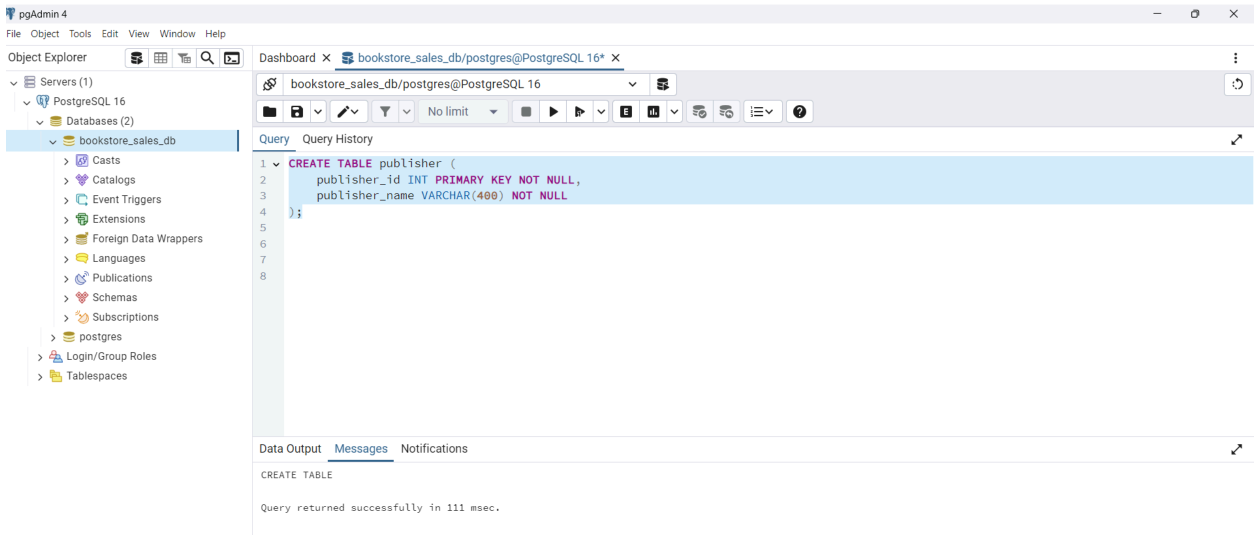


Table Description:

- **Publisher**: Details of publishers for books.
- **Book**: Details of books available in the store.
- **BookAuthor**: Stores the authors for each book, which is a many-to-many relationship.
- **Author**: Details of all authors.

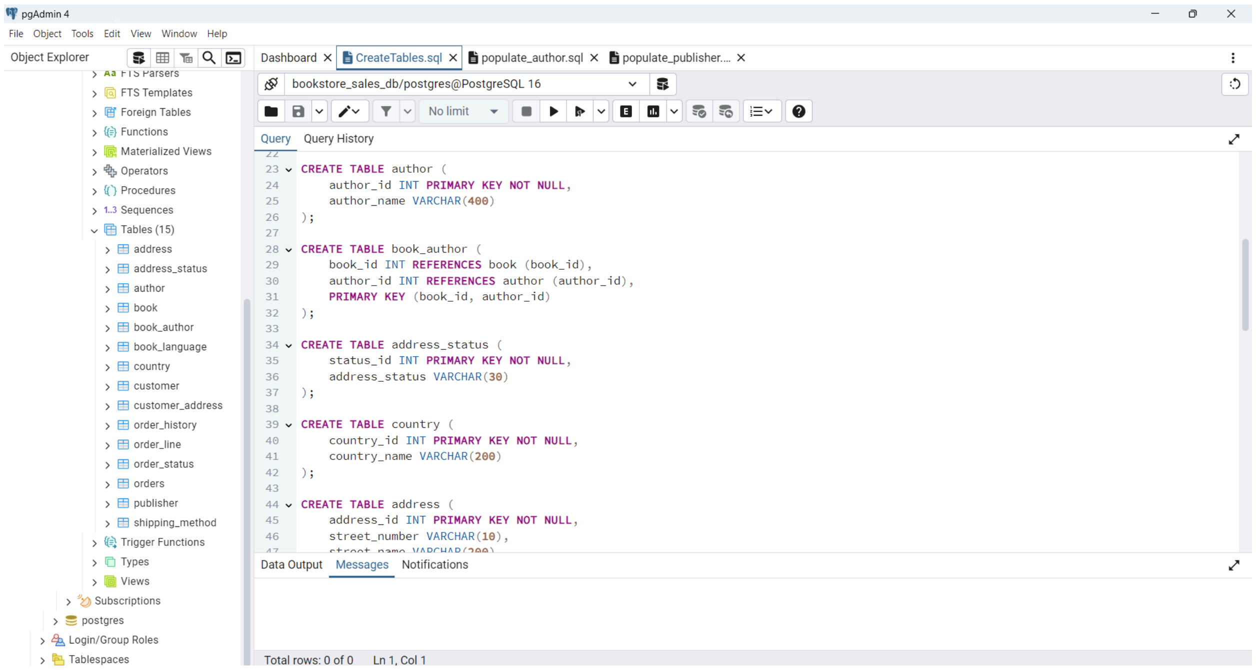
- **BookLanguage**: A list of possible languages of books.
- **Customer**: A list of the customers of the Bookstore.
- **CustomerAddress**: A list of addresses for customers. A customer can have more than one address, and an address can have more than one customer.
- **AddressStatus**: A list of statuses for an address. Addresses can be Active or Inactive
- **Address**: Details of addresses in the system.
- **Country**: A list of countries that addresses are in.
- **Order**: A list of orders placed by customers.
- **OrderLine**: A list of books that are a part of each order.
- **ShippingMethod**: Details of shipping methods for an order (Priority, Standard, Express).
- **OrderHistory**: History of an order, such as ordered, shipped, delivered.
- **OrderStatus**: The possible status of an order (Order Received, Delivered, Canceled)

Creating tables in PostgresSQL (bookstore_sales_db)



```sql
CREATE TABLE publisher (
    publisher_id INT PRIMARY KEY NOT NULL,
    publisher_name VARCHAR(400) NOT NULL
);
```

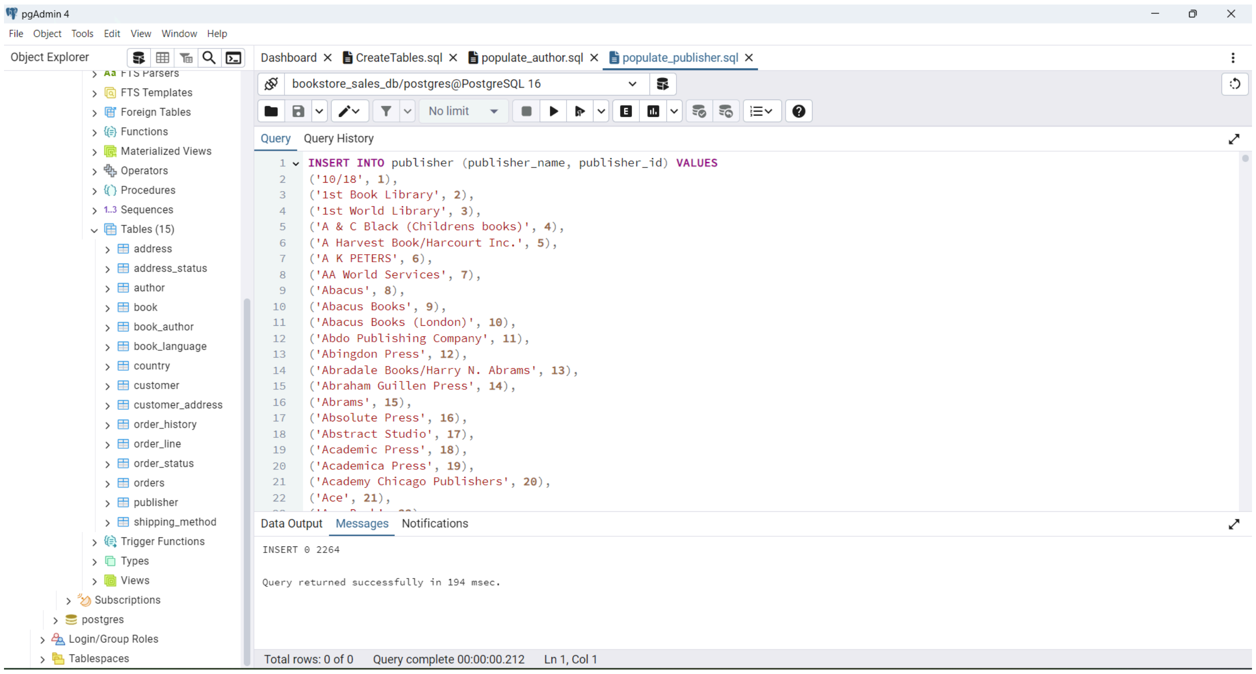
```
CREATE TABLE

Query returned successfully in 111 msec.
```



```sql
CREATE TABLE author (
    author_id INT PRIMARY KEY NOT NULL,
    author_name VARCHAR(400)
);

CREATE TABLE book_author (
    book_id INT REFERENCES book (book_id),
    author_id INT REFERENCES author (author_id),
    PRIMARY KEY (book_id, author_id)
);

CREATE TABLE address_status (
    status_id INT PRIMARY KEY NOT NULL,
    address_status VARCHAR(30)
);

CREATE TABLE country (
    country_id INT PRIMARY KEY NOT NULL,
    country_name VARCHAR(200)
);

CREATE TABLE address (
    address_id INT PRIMARY KEY NOT NULL,
    street_number VARCHAR(10),
```

Total rows: 0 of 0   Ln 1, Col 1

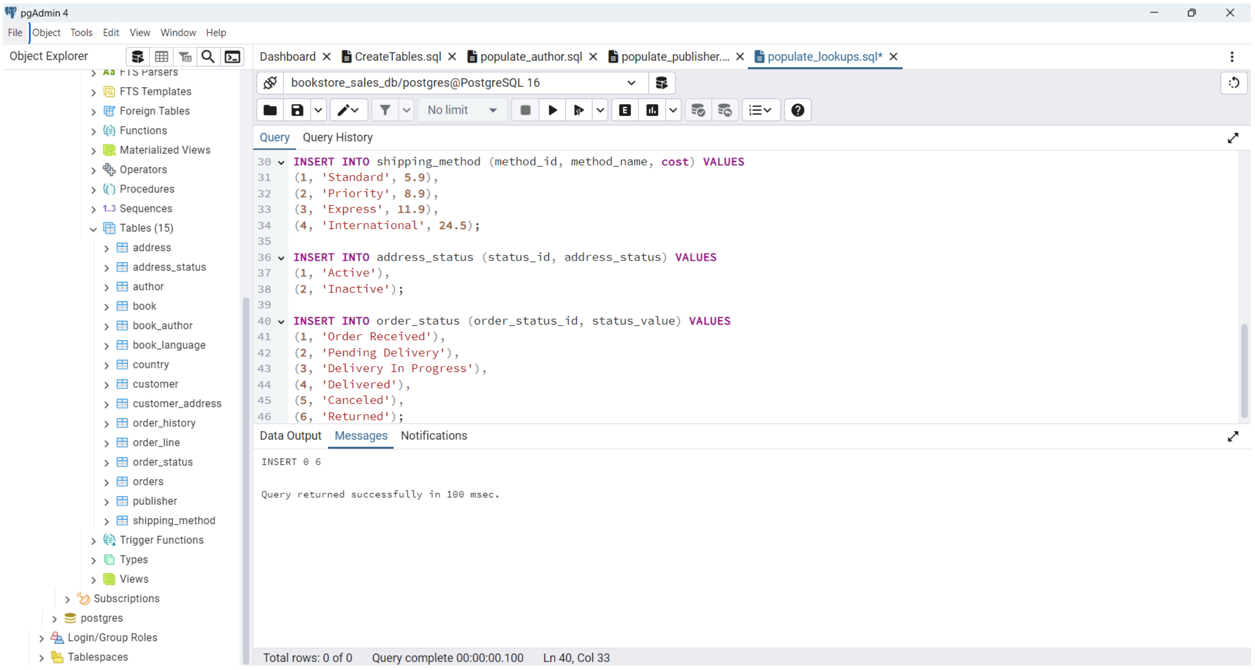
Inserting data into the tables in bookstore_sales_db:



```sql
INSERT INTO publisher (publisher_name, publisher_id) VALUES
('10/18', 1),
('1st Book Library', 2),
('1st World Library', 3),
('A & C Black (Childrens books)', 4),
('A Harvest Book/Harcourt Inc.', 5),
('A K PETERS', 6),
('AA World Services', 7),
('Abacus', 8),
('Abacus Books', 9),
('Abacus Books (London)', 10),
('Abdo Publishing Company', 11),
('Abingdon Press', 12),
('Abradale Books/Harry N. Abrams', 13),
('Abraham Guillen Press', 14),
('Abrams', 15),
('Absolute Press', 16),
('Abstract Studio', 17),
('Academic Press', 18),
('Academica Press', 19),
('Academy Chicago Publishers', 20),
('Ace', 21),
```

```
INSERT 0 2264

Query returned successfully in 194 msec.
```

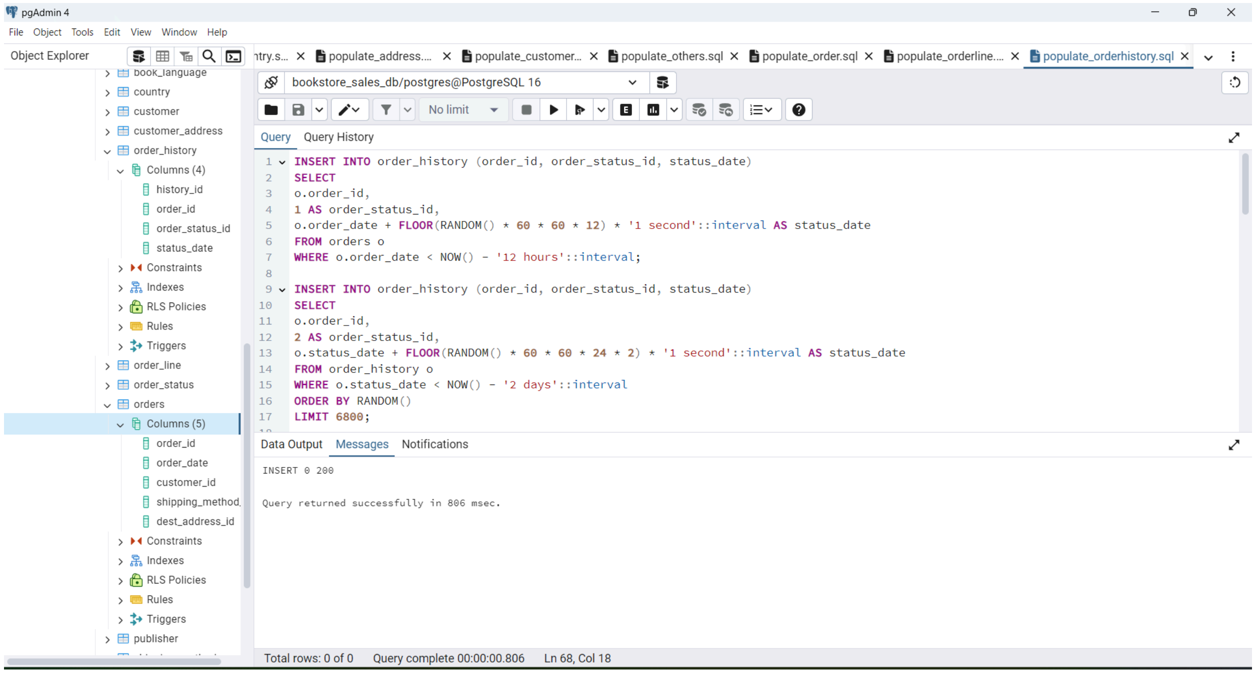
Total rows: 0 of 0   Query complete 00:00:00.212   Ln 1, Col 1



```sql
INSERT INTO shipping_method (method_id, method_name, cost) VALUES
(1, 'Standard', 5.9),
(2, 'Priority', 8.9),
(3, 'Express', 11.9),
(4, 'International', 24.5);

INSERT INTO address_status (status_id, address_status) VALUES
(1, 'Active'),
(2, 'Inactive');

INSERT INTO order_status (order_status_id, status_value) VALUES
(1, 'Order Received'),
(2, 'Pending Delivery'),
(3, 'Delivery In Progress'),
(4, 'Delivered'),
(5, 'Canceled'),
(6, 'Returned');
```

```
INSERT 0 6

Query returned successfully in 100 msec.
```

Total rows: 0 of 0 Query complete 00:00:00.100 Ln 40, Col 33

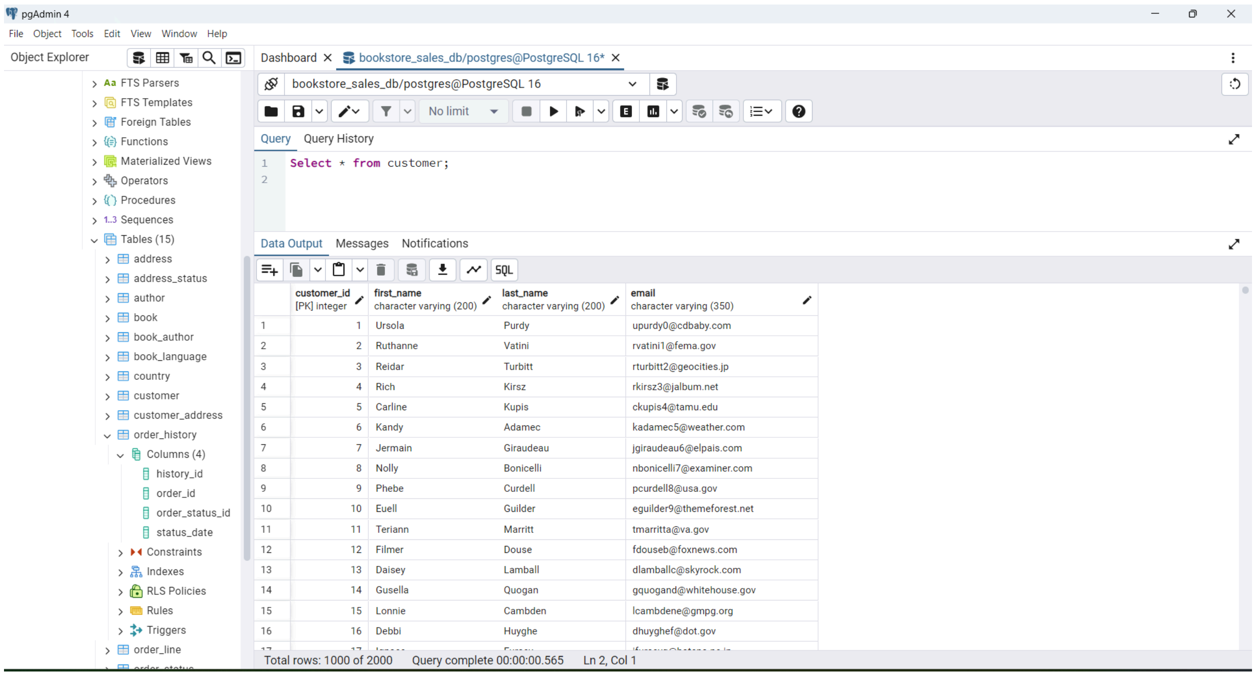


```sql
INSERT INTO order_history (order_id, order_status_id, status_date)
SELECT
o.order_id,
1 AS order_status_id,
o.order_date + FLOOR(RANDOM() * 60 * 60 * 12) * '1 second'::interval AS status_date
FROM orders o
WHERE o.order_date < NOW() - '12 hours'::interval;

INSERT INTO order_history (order_id, order_status_id, status_date)
SELECT
o.order_id,
2 AS order_status_id,
o.status_date + FLOOR(RANDOM() * 60 * 60 * 24 * 2) * '1 second'::interval AS status_date
FROM order_history o
WHERE o.status_date < NOW() - '2 days'::interval
ORDER BY RANDOM()
LIMIT 6800;
```

```
INSERT 0 200

Query returned successfully in 806 msec.
```

Total rows: 0 of 0 Query complete 00:00:00.806 Ln 68, Col 18

Viewing data in customer table:



```sql
Select * from customer;
```

	customer_id [PK] integer	first_name character varying (200)	last_name character varying (200)	email character varying (350)
1	1	Ursola	Purdy	upurdy0@cdbaby.com
2	2	Ruthanne	Vatini	rvatini1@fema.gov
3	3	Reidar	Turbitt	rturbitt2@geocities.jp
4	4	Rich	Kirsz	rkirsz3@jalbum.net
5	5	Carline	Kupis	ckupis4@tamu.edu
6	6	Kandy	Adamec	kadamec5@weather.com
7	7	Jermain	Giraudeau	jgiraudeau6@elpais.com
8	8	Nolly	Bonicelli	nbonicelli7@examiner.com
9	9	Phebe	Curdell	pcurdell8@usa.gov
10	10	Euell	Guilder	eguilder9@themeforest.net
11	11	Teriann	Marritt	tmarritta@va.gov
12	12	Filmer	Douse	fdouseb@foxnews.com
13	13	Daisey	Lamball	dlamballc@skyrock.com
14	14	Gusella	Quogan	gquogand@whitehouse.gov
15	15	Lonnie	Cambden	lcambdene@gmpg.org
16	16	Debbi	Huyghe	dhuyghef@dot.gov

Total rows: 1000 of 2000   Query complete 00:00:00.565   Ln 2, Col 1

In this project, we have categorized the data into dimensional data, fact data, and reference data based on their roles within the system. Here is how we have labeled and organized them:

1. Dimensional Data (Slowly and Not-So-Slowly Changing Data)

These tables contain descriptive information about entities, and some of these attributes may change over time. We'll apply slowly changing dimension (SCD) strategies to track historical changes where necessary.

- **Publisher**: Stores details of publishers for books. This dimension may change over time (e.g., publisher mergers or name changes), making it a slowly-changing dimension if historical data needs to be retained.

- **Book**: Contains details of books available in the store, including ISBN, title, and other metadata. Since book details like editions or prices could change over time, we treat this as a slowly-changing dimension.

- **Author**: Contains information about the authors of books. This is also a dimensional table since author details might occasionally change, such as name updates.

- **BookAuthor**: Manages the many-to-many relationship between books and authors. Changes to author assignments are not so frequent, so this is a slowly-changing dimension.

- **Customer**: A list of bookstore customers. Since customer data can change (e.g., name, contact info), we treat this as a slowly-changing dimension.

- **CustomerAddress**: Stores the addresses of customers. Customers may update their addresses over time, making this a slowly-changing dimension.

- **Address**: Contains details of addresses in the system, which may change over time (e.g., updates to city or postal codes). This is also treated as a slowly-changing dimension.

- **OrderHistory**: Tracks the status changes of an order, from being received to being shipped and delivered. This would be a not-so-slowly-changing dimension, as order status updates occur frequently but still need tracking.

2. Fact Data (Transactional Data)
These tables represent transactions and capture specific events. They are often updated frequently and contain references to dimensional data.

- **Order**: Represents individual customer orders. Each order is a transactional record capturing details like the date and customer associated with it.

- **OrderLine**: Contains the books in each order. This is a fact table that links orders to the books being purchased. The data here is transactional and updated with each purchase.

3. Reference Data (Static or Slowly Changing)
These tables hold static or near-static data that is referenced across the system but doesn't change frequently. They are often used for validation or categorization.

- **BookLanguage**: A static list of possible languages of books. This is reference data, as language options rarely change over time.

- **AddressStatus**: Contains statuses for addresses, such as "Active" or "Inactive." This reference data doesn’t change frequently, but it's used for managing addresses.

- **Country**: Stores a list of countries that addresses belong to. Countries rarely change, so this is static reference data.

- **ShippingMethod**: Defines the available shipping methods (e.g., Priority, Standard, Express). This is reference data that may occasionally expand but changes infrequently.

- **OrderStatus**: Tracks the various stages of an order (Order Received, Delivered, Canceled). This data is static but frequently referenced in order processing.

This Book Sales and Inventory Management System is designed to ensure a reasonable volume of transactional data. The Order and OrderLine tables capture key transactional data, including every purchase made by customers and the details of the books in each order.

By tracking each customer order and the specific items within those orders, we generate a robust dataset for analysis. Additionally, the OrderHistory table logs the status changes of each order (e.g., "Order Received," "Shipped," "Delivered"), further expanding the volume of transactional data. This allows for comprehensive reporting on customer purchases, order fulfillment times, and sales trends over time.

Dimensions and Hierarchies

This project contains multiple dimensions, hierarchies, and key measures that can be analyzed.

Potential Dimensions:

Dimensions represent entities or attributes that provide context to the data. Here are some key dimensions from our system:

1. **Customer Dimension**: Contains customer details such as name, email, and customer ID.
2. **Book Dimension**: Contains details about the books, such as title, genre, ISBN, and publication year.
3. **Publisher Dimension**: Stores publisher information, which can also be used to analyze sales by publisher.
4. **Order Dimension**: Contains order details, such as order date, order ID, and shipping method.
5. **Time Dimension**: Implicit in the Order table via the order date, allowing for time-based analysis (e.g., sales over months, quarters, or years).
6. **Address Dimension**: Stores customer address details, which could be used for geographical analysis of sales by region.

Hierarchies:

Hierarchies allow us to drill down into the data at different levels of granularity for more detailed analysis.

1. **Time Hierarchy**:
   - Year → Quarter → Month → Day → Order Date
   - This allows analysis of sales trends over time, from high-level yearly views to detailed daily trends.
2. **Geographical Hierarchy**:

    - Country → State/Province → City → Postal Code
    - This enables analysis of sales or customer distribution across geographic regions.
3. **Product Hierarchy**:
    - Publisher → Book Title → Edition
    - This hierarchy can help analyze book sales by publisher, then by title, and then by edition.

Measures:

Measures are quantitative data points that can be aggregated or analyzed. We have several important measures:

1. **Total Sales**:
    - Sum of the sales amount in the OrderLine table. This is a primary measure to evaluate store performance and can be analyzed by dimensions like Time, Customer, or Book.
2. **Quantity Sold**:
    - The number of books sold, which can be aggregated from the OrderLine table. This can be analyzed across different dimensions like Book or Publisher.
3. **Order Count**:
    - The number of orders placed, which can be counted in the Order table and analyzed by Customer, Time, or Geographical dimensions.
4. **Average Order Value**:
    - Calculated by dividing total sales by the number of orders. This can be used to analyze customer purchasing behavior and trends over time.

These measures, combined with the dimensions and hierarchies, provide a framework for analysis.