

# Managing User Permissions for Different Roles in Bookstore

Here are the different roles:

- **Manager:** Has full access (SELECT, INSERT, UPDATE, DELETE) to all tables ( `books` , `authors` , `customers` , `orders` ).
- **Sales Staff:** Can view books and customer information (SELECT) but cannot modify or delete data.
- **Inventory Staff:** Can add new books (INSERT) and update existing book information (UPDATE) but cannot access customer or order information.

```
CREATE TABLE authors (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE customers (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(255) NOT NULL,  
  email VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE books (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  title VARCHAR(255) NOT NULL,  
  author_id INT NOT NULL,  
  FOREIGN KEY (author_id) REFERENCES authors(id)  
);
```

```
CREATE TABLE orders (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  customer_id INT NOT NULL,
```

```
book_id INT NOT NULL,  
FOREIGN KEY (customer_id) REFERENCES customers(id),  
FOREIGN KEY (book_id) REFERENCES books(id)  
);
```

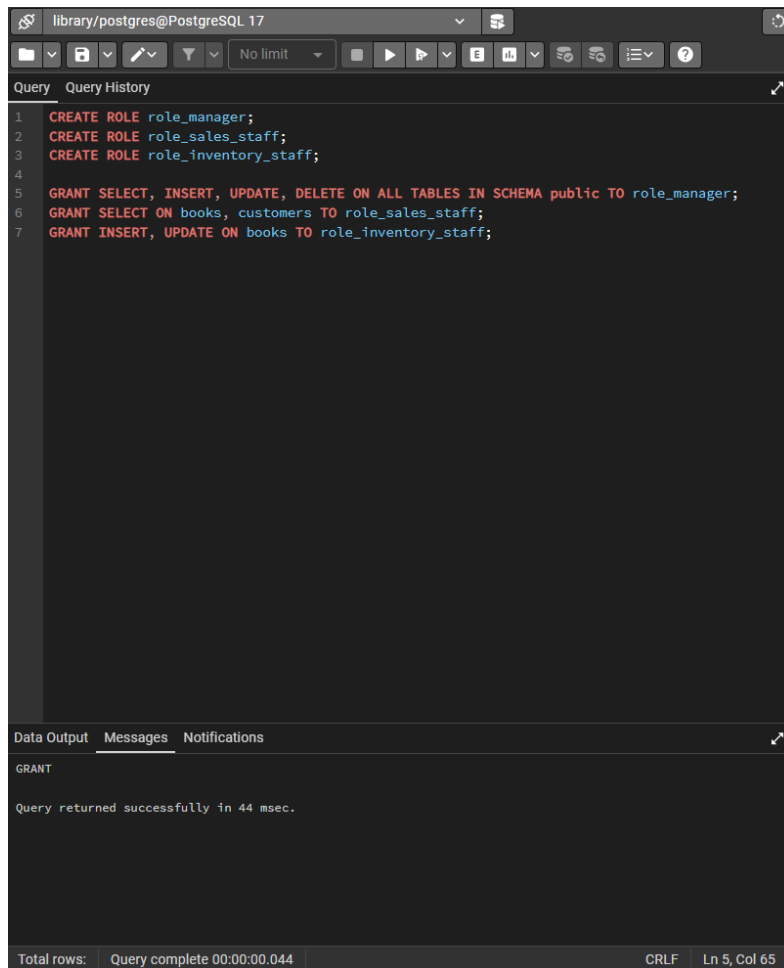
### **Creating Roles:**

```
CREATE ROLE role_manager;  
CREATE ROLE role_sales_staff;  
CREATE ROLE role_inventory_staff;
```

### **Granting Privileges to Roles:/code**

```
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES TO role_manager;  
GRANT SELECT ON books, customers TO role_sales_staff;  
GRANT INSERT, UPDATE ON books TO role_inventory_staff;
```

### **Output:**



The screenshot shows a PostgreSQL query editor window titled 'library/postgres@PostgreSQL 17'. The editor has a toolbar with icons for file operations, query execution, and settings. Below the toolbar, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL code:

```
1 CREATE ROLE role_manager;  
2 CREATE ROLE role_sales_staff;  
3 CREATE ROLE role_inventory_staff;  
4  
5 GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO role_manager;  
6 GRANT SELECT ON books, customers TO role_sales_staff;  
7 GRANT INSERT, UPDATE ON books TO role_inventory_staff;
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the following output:

```
GRANT  
  
Query returned successfully in 44 msec.
```

At the bottom of the window, there is a status bar showing 'Total rows: Query complete 00:00:00.044', 'CRLF', and 'Ln 5, Col 65'.

## Create Users:

```
1 CREATE ROLE fatima WITH LOGIN PASSWORD '1234';
2 CREATE ROLE sara WITH LOGIN PASSWORD '1234';
3 CREATE ROLE norah WITH LOGIN PASSWORD '1234';
```

Query returned successfully in 74 msec.

Total rows: Query complete 00:00:00.074 CRLF Ln 3, Col 24

## Assigning Users to Roles:

The screenshot shows a PostgreSQL client window titled 'library/postgres@PostgreSQL 17'. The main query editor contains three lines of SQL: `1 GRANT role_manager TO fatima;`, `2 GRANT role_sales_staff TO sara;`, and `3 GRANT role_inventory_staff TO norah;`. A tooltip 'Execute script' with the 'F5' key icon is visible over the first line. Below the editor, the 'Data Output' tab is active, displaying the text 'GRANT ROLE' and 'Query returned successfully in 42 msec.'. A green status bar at the bottom right confirms 'Query returned successfully in 42 msec.' with a close button. The bottom status bar shows 'Total rows: Query complete 00:00:00.042' and 'CRLF Ln 3, Col 36'.

## Explanation:

- `fatima` (manager) has full access to all tables through the `role_manager` role.
- `sara` (sales staff) can only view the information in `books` and `customer` tables due to `role_sales_staff` permissions.
- `norah` (inventory staff) can add new books and update existing ones but cannot access customer or order information.