Final Project

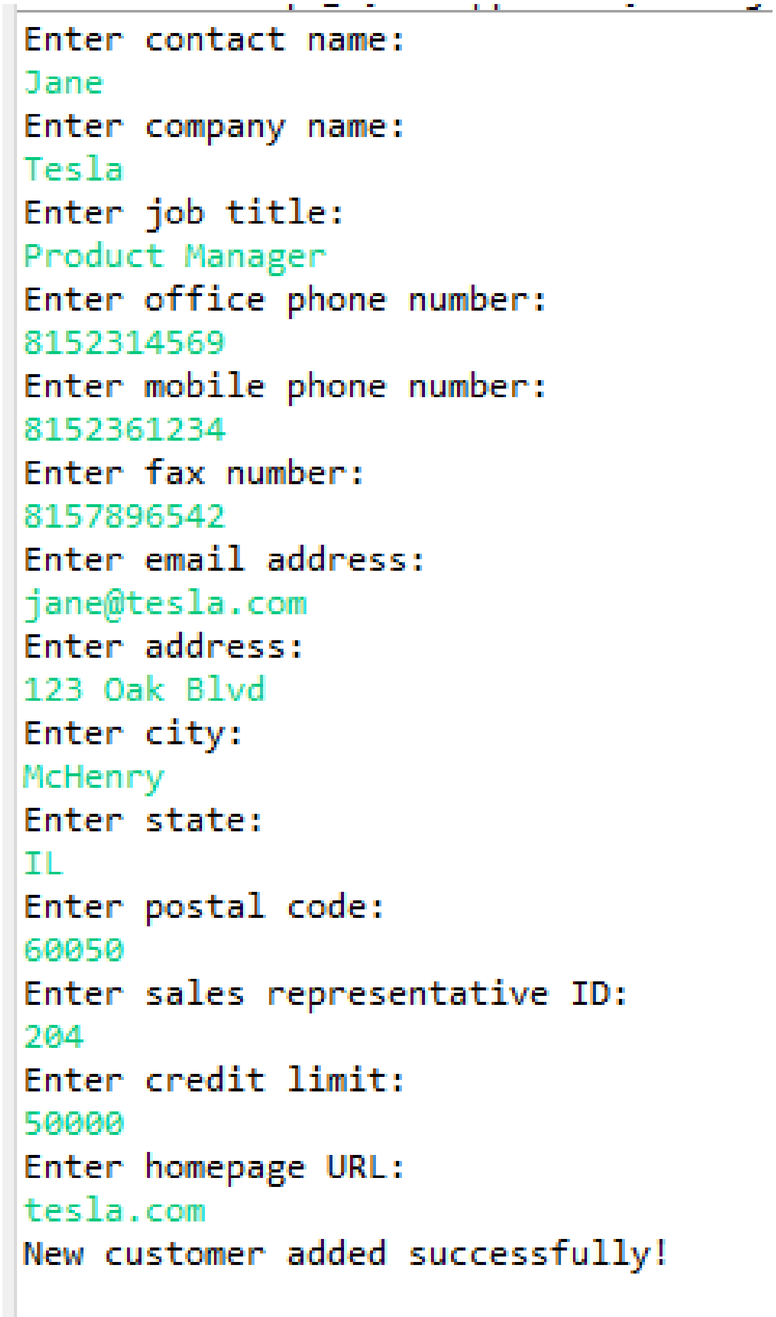
CS5200 Database Management Systems

Amber Veronese

November 27, 2024

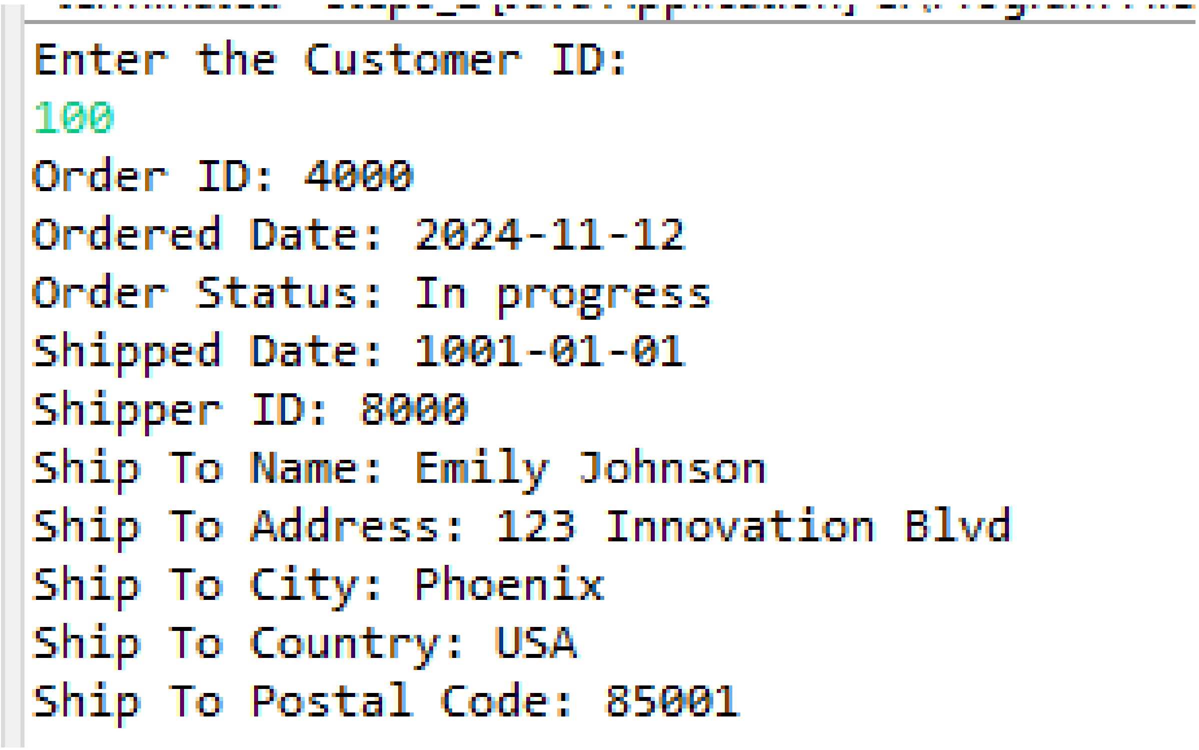
Table of Contents

Step 5_1 Screenshot: Entering a new customer into SQL database



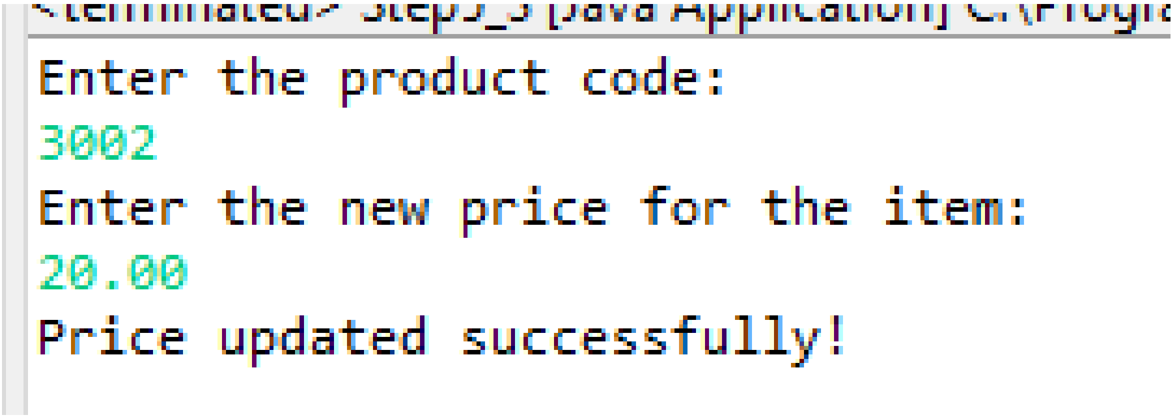
```
Enter contact name:
Jane
Enter company name:
Tesla
Enter job title:
Product Manager
Enter office phone number:
8152314569
Enter mobile phone number:
8152361234
Enter fax number:
8157896542
Enter email address:
jane@tesla.com
Enter address:
123 Oak Blvd
Enter city:
McHenry
Enter state:
IL
Enter postal code:
60050
Enter sales representative ID:
204
Enter credit limit:
50000
Enter homepage URL:
tesla.com
New customer added successfully!
```

<u>Step 5_2 Screenshot:</u> Read operation to list all orders for a specific customer



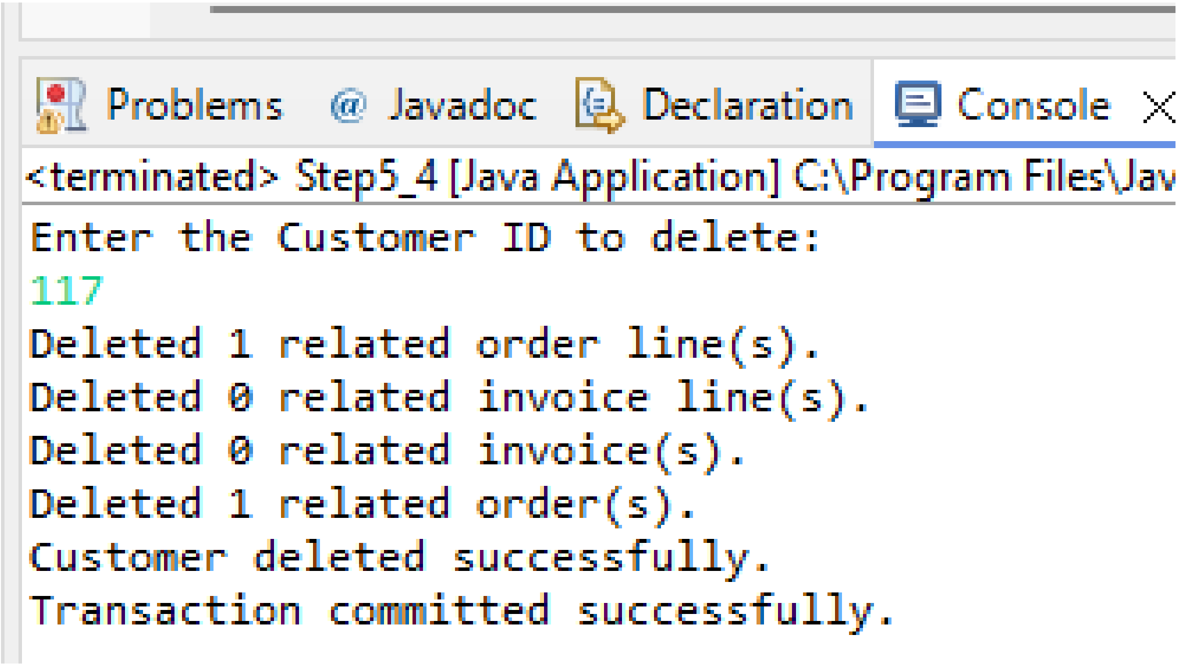
```
Enter the Customer ID:
100
Order ID: 4000
Ordered Date: 2024-11-12
Order Status: In progress
Shipped Date: 1001-01-01
Shipper ID: 8000
Ship To Name: Emily Johnson
Ship To Address: 123 Innovation Blvd
Ship To City: Phoenix
Ship To Country: USA
Ship To Postal Code: 85001
```

Step 5_3 Screenshot: Updating the price of an item with a given product code



```
<terminated> Step5_3 [Java Application] C:\Progra
Enter the product code:
3002
Enter the new price for the item:
20.00
Price updated successfully!
```

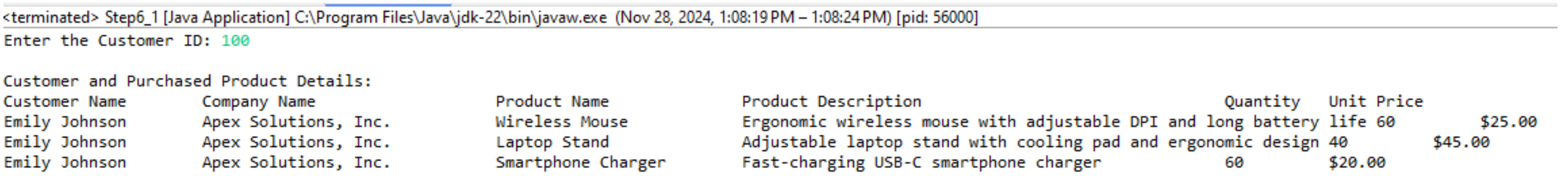
<u>Step 5_4 Screenshot:</u> Updating records: deleting one customer given the customer ID



<terminated> Step5_4 [Java Application] C:\Program Files\Jav

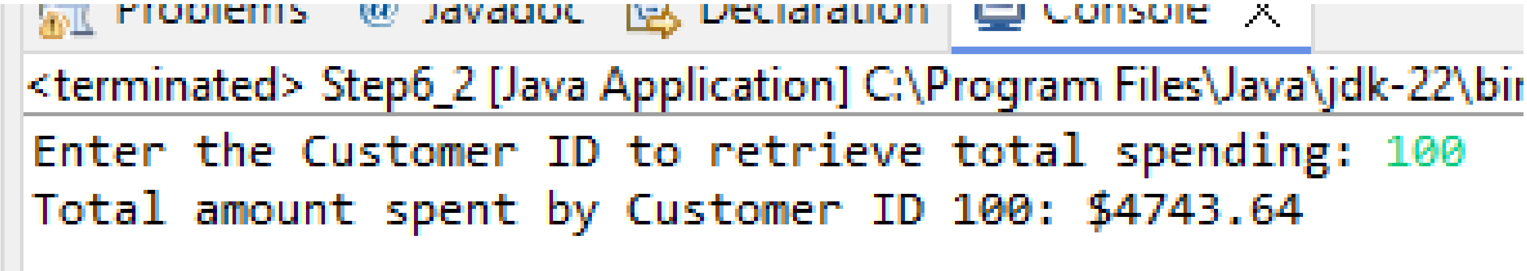
```
Enter the Customer ID to delete:
117
Deleted 1 related order line(s).
Deleted 0 related invoice line(s).
Deleted 0 related invoice(s).
Deleted 1 related order(s).
Customer deleted successfully.
Transaction committed successfully.
```

Step 6_1 Screenshot: Using Joins to retrieve data about a customer and any products they may have ordered.



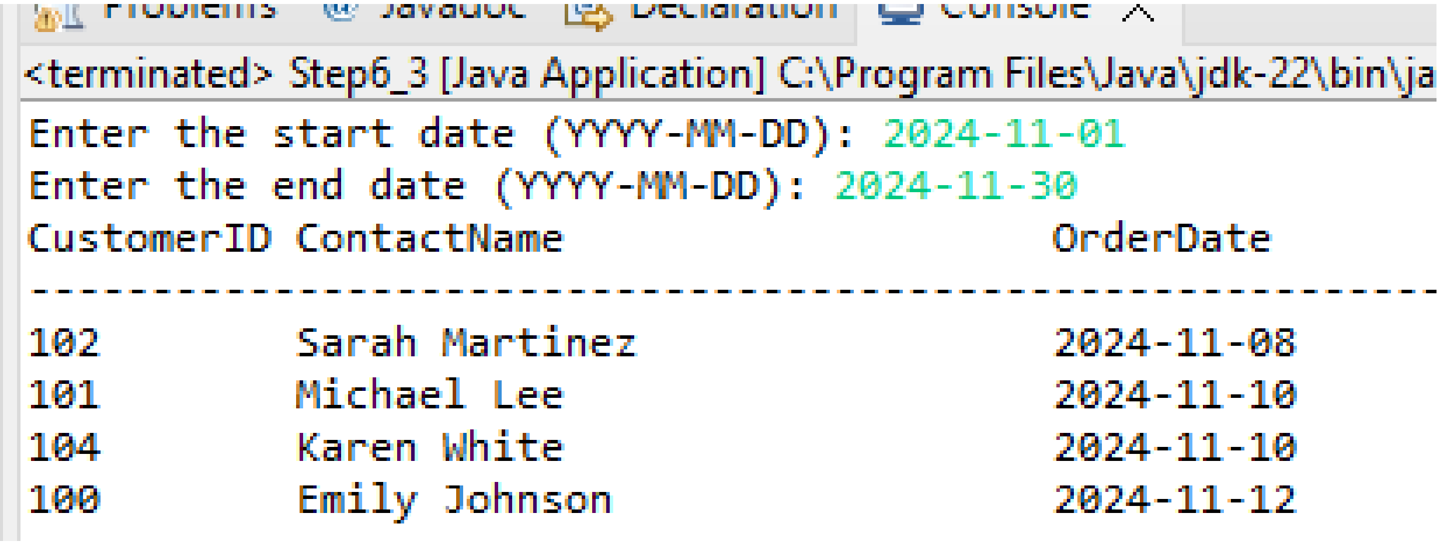
```
<terminated> Step6_1 [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (Nov 28, 2024, 1:08:19 PM – 1:08:24 PM) [pid: 56000]
Enter the Customer ID: 100

Customer and Purchased Product Details:
Customer Name        Company Name                   Product Name              Product Description                                  Quantity   Unit Price
Emily Johnson        Apex Solutions, Inc.           Wireless Mouse            Ergonomic wireless mouse with adjustable DPI and long battery life 60         $25.00
Emily Johnson        Apex Solutions, Inc.           Laptop Stand              Adjustable laptop stand with cooling pad and ergonomic design 40         $45.00
Emily Johnson        Apex Solutions, Inc.           Smartphone Charger        Fast-charging USB-C smartphone charger             60         $20.00
```

<u>Step 6_2 Screenshot:</u> Using SUM to count the total amount a customer has spent



```
<terminated> Step6_2 [Java Application] C:\Program Files\Java\jdk-22\bin
Enter the Customer ID to retrieve total spending: 100
Total amount spent by Customer ID 100: $4743.64
```

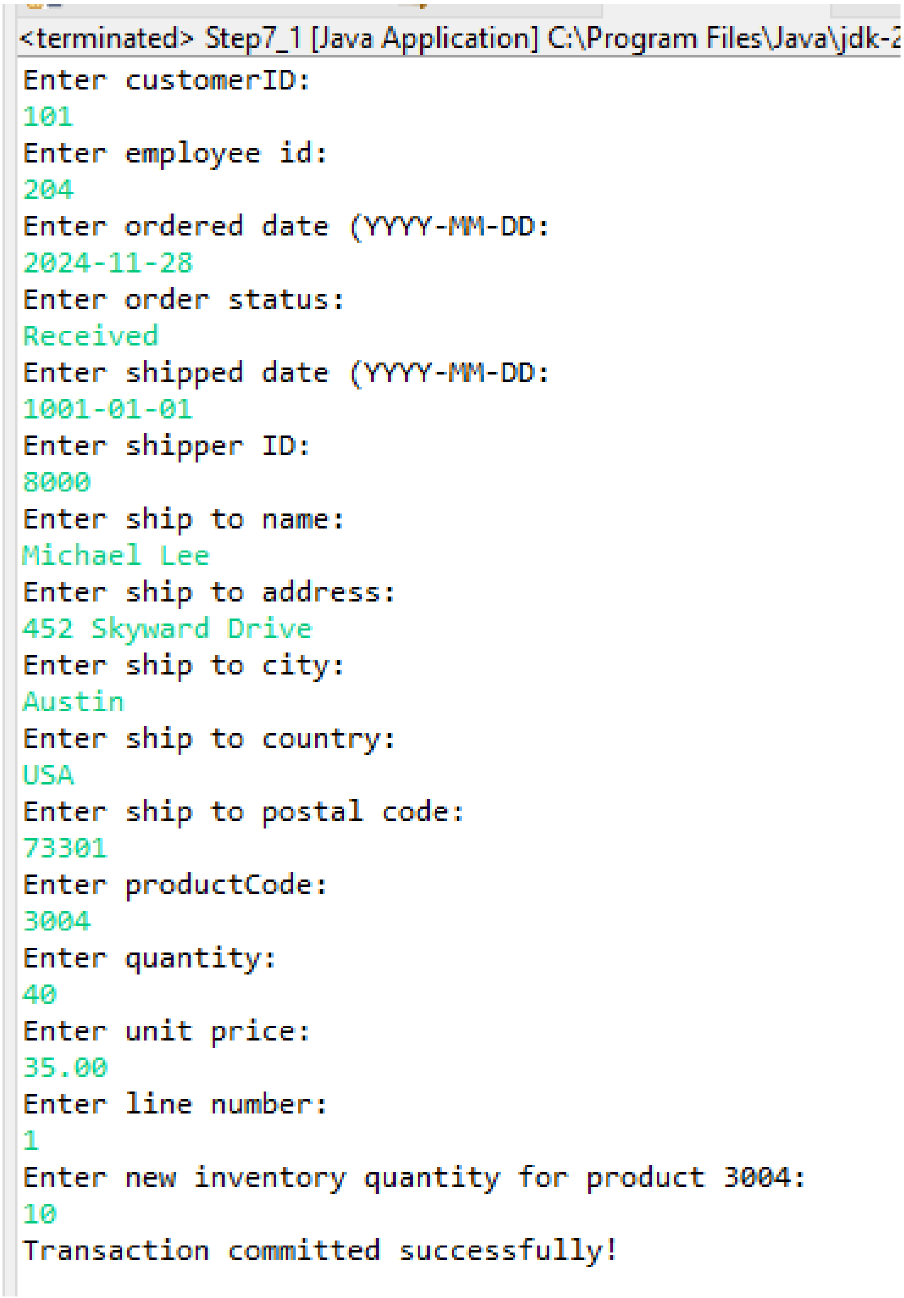
Step 6_3 Screenshot: Creating a query that includes filtering a sorting, listing all customers who have placed an order in November, sorted by order date



```
<terminated> Step6_3 [Java Application] C:\Program Files\Java\jdk-22\bin\ja
Enter the start date (YYYY-MM-DD): 2024-11-01
Enter the end date (YYYY-MM-DD): 2024-11-30
CustomerID ContactName                    OrderDate
----------------------------------------------------------
102        Sarah Martinez                 2024-11-08
101        Michael Lee                    2024-11-10
104        Karen White                    2024-11-10
100        Emily Johnson                  2024-11-12
```

<u>Step 7_1 Screenshot:</u> Creating a transactional query where the user can input a new order into the database system, update the order lines table, and then update the product quantity per the orders. The transaction is committed and rolledback if any errors occur.

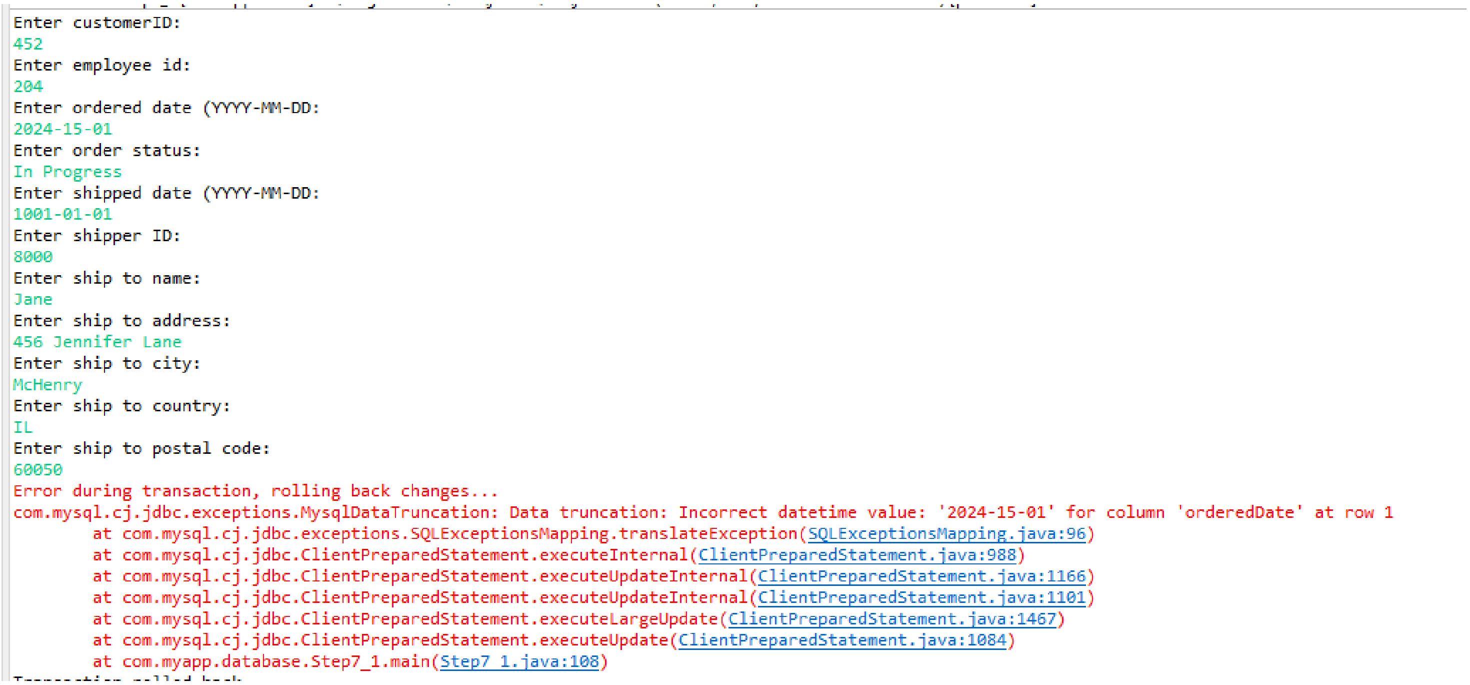
Successful Transaction



```
<terminated> Step7_1 [Java Application] C:\Program Files\Java\jdk-2
Enter customerID:
101
Enter employee id:
204
Enter ordered date (YYYY-MM-DD:
2024-11-28
Enter order status:
Received
Enter shipped date (YYYY-MM-DD:
1001-01-01
Enter shipper ID:
8000
Enter ship to name:
Michael Lee
Enter ship to address:
452 Skyward Drive
Enter ship to city:
Austin
Enter ship to country:
USA
Enter ship to postal code:
73301
Enter productCode:
3004
Enter quantity:
40
Enter unit price:
35.00
Enter line number:
1
Enter new inventory quantity for product 3004:
10
Transaction committed successfully!
```

<u>Step 7_1 Screenshot:</u> Creating a transactional query where the user can input a new order into the database system, update the order lines table, and then update the product quantity per the orders. The transaction is committed and rolledback if any errors occur.

Example of Rolled Back Transaction: Invalid information entered



```
Enter customerID:
452
Enter employee id:
204
Enter ordered date (YYYY-MM-DD:
2024-15-01
Enter order status:
In Progress
Enter shipped date (YYYY-MM-DD:
1001-01-01
Enter shipper ID:
8000
Enter ship to name:
Jane
Enter ship to address:
456 Jennifer Lane
Enter ship to city:
McHenry
Enter ship to country:
IL
Enter ship to postal code:
60050
Error during transaction, rolling back changes...
com.mysql.cj.jdbc.exceptions.MysqlDataTruncation: Data truncation: Incorrect datetime value: '2024-15-01' for column 'orderedDate' at row 1
	at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:96)
	at com.mysql.cj.jdbc.ClientPreparedStatement.executeInternal(ClientPreparedStatement.java:988)
	at com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1166)
	at com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1101)
	at com.mysql.cj.jdbc.ClientPreparedStatement.executeLargeUpdate(ClientPreparedStatement.java:1467)
	at com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdate(ClientPreparedStatement.java:1084)
	at com.myapp.database.Step7_1.main(Step7_1.java:108)
Transaction rolled back
```

READ ME FILE STEP 5_1:

Overview and General Purpose Statement:

This Java program was created in conjunction with database management systems through SQL using MySQL Workbench. This program creates generated fields for a user to enter a new customer into the project database in the Customers table.

Instructions:

When running this customer creation program, the program will ask for the following information about the customer you are inserting into the database:

- Contact Name
- Company name
- Contact Job Title
- Office Phone
- Mobile Phone
- Fax Number
- Email Address
- Address
- City
- State
- Postal code
- Sales Rep ID assigned to customer
- Credit Limit
- Homepage from website

Enter all required information into the database fields. If the customer is added successfully, a successful completion message will be revealed on the screen. If an error occurs, the program will reveal an error code on the screen.

- The error statement: "*Foreign key constraint failed: The salesRepID doesn't exist*." means that there was an error with the information entered into the system that relates to the foreign key.

- The foreign key for this table is the Sales Rep ID field. The error message “*Failed to add customer*” means that there was an error when entering the customer into the system.
- The error message “*Database connection error*” means that there was an error when connecting to the database management system in MySQL workbench.

READ ME FILE STEP 5_2:

Overview and General Purpose Statement:

This Java program was created in conjunction with database management systems through SQL using MySQL Workbench. This program was created to show all orders for a certain customer based on the user's input of a Customer ID that corresponds to a certain customer and their orders.

Instructions:

When running this customer creation program, the program will ask for the following information about the customer you are looking to view the orders for:

- Customer ID

Enter the Customer ID when prompted. After which, the program will list order information for the selected customer. The order fields will include:

- Order ID
- Ordered Date
- Order Status
- Shipped Date
- Shipper ID
- Ship To Name
- Ship To Address
- Ship To City
- Ship To Country
- Ship To Postal Code

READ ME FILE STEP 5_3:

Overview and General Purpose Statement:

This Java program was created in conjunction with database management systems through SQL using MySQL Workbench. This program was created to allow users to update the price of the products within the database.

Instructions:

When running this price update program, the program will ask for the following information about the product and price you are updating:

- Product Code
- New price for the item

Enter all required information into the database fields. If the price is updated successfully, a successful completion message will be revealed on the screen. If an error occurs, the program will reveal an error code on the screen.

- The error statement: "*No product found with the given product code*." means that there was an error with the information entered into the system for the product code.
- The error message "*Error updating the product price*" means that there was an error when entering the customer the product price into the system.

READ ME FILE STEP 5_4:

Overview and General Purpose Statement:

This Java program was created in conjunction with database management systems through SQL using MySQL Workbench. This program was created to allow users to delete customers when needed.

Instructions:

When running this customer deletion program, the program will ask for the following information about the customer you are trying to delete from the database:

- Customer ID

Enter all required information into the database fields. If the customer is deleted successfully, a successful completion message will be revealed on the screen. If an error occurs, the program will reveal an error code on the screen.

- The error statement: "*No customer found with the provided Customer ID*." means that there was an error with the information entered into the system for the Customer ID.
- The error message "*Transaction rolled back due to an error*" means that there was an error when running the program and the change was not executed.

READ ME FILE STEP 6_1:

Overview and General Purpose Statement:

This Java program was created in conjunction with database management systems through SQL using MySQL Workbench. This program was created to allow users to insert a customer name and find any information about that customer as well as information about any products they may have ordered.

Instructions:

When running this information retrieval program, the program will ask for the following information about the customer you are trying to delete from the database:

- Customer ID

Enter all required information into the database fields. If the query is successful, the customer and product details will be shown on the screen. If an error occurs, the program will reveal an error message on the screen.

- The error statement: "*No records found for the Customer ID [customerID]*" means that there was an error with the information entered into the system for the Customer ID.
- The error message "*Database error*" means that there was a database error.
- The error statement "*Invalid input. Please enter a valid numeric customer ID*" means that there was an error with the way the user input the customer ID, perhaps placing string values instead of numeric values.

READ ME FILE STEP 6_2:

Overview and General Purpose Statement:

This Java program was created in conjunction with database management systems through SQL using MySQL Workbench. This program was created to allow users to calculate the total amount spent by any customer that user inputs into the program.

Instructions:

When running this information retrieval program, the program will ask for the following information about the customer you are trying to delete from the database:

- Customer ID

Enter all required information into the database fields. If the query is successful, the total amount the customer spent will be shown on the screen. If an error occurs, the program will reveal an error message on the screen.

- The error statement: "*No spending records found for the provided customer ID*" means that this customer has not purchased any products in the database system.
- The error message "*Database error*" means that there was a database error.
- The error statement: "*No data found for the provided customer ID*" means that there was an error with the information entered into the system for the Customer ID.

READ ME FILE STEP 6_3:

Overview and General Purpose Statement:

This Java program was created in conjunction with database management systems through SQL using MySQL Workbench. This program was created to allow users to show customers who have sent orders within a certain time frame that are input by the user.

Instructions:

When running this information retrieval program, the program will ask for the following information about the customer you are trying to delete from the database:

- Start date to find orders (finding orders between x date and x date)
- End date to find orders (finding orders between x date and x date)

Enter all required information into the database fields. If the query is successful, the orders that were sent between those specified dates will be shown on the screen. Included in the information will be the customer ID, contact name, and order date.

- The error statement: "*No customers found with orders in the specified date range*" means that no customers have purchased orders within that time frame that was input by the user.
- The error message "*Database connection or query failed!*" means that there was a database error.

READ ME FILE STEP 7_1:

Overview and General Purpose Statement:

This Java program was created in conjunction with database management systems through SQL using MySQL Workbench. This program was created to allow users to input an order which updates the orders table, input order information which updates the order lines table, and manually update product quantity from the products table.

Instructions:

When running this information retrieval program, the program will ask for the following information about the customer and order details:

- Information that updates within the "Orders" Table
    - Customer ID
    - Employee ID
    - Ordered Date
    - Order Status
    - Ship to Name
    - Ship to Address
    - Ship to City
    - Ship to Postal Code
- Information that updates within the "Order Lines" table
    - Product Code
    - Quantity
    - Unit Price
    - Line Number
- Information that updates within the "Products" Table
    - New inventory quantity for product [x]

Enter all required information into the database fields. If the query is successful, the order will be sent and updated in the orders table, order lines table, and products table. If an error occurs, the program will reveal an error message on the screen.

- The error statement: "*Error during transaction, rolling back changes*" means that there was an error handling the transaction
- The error message "*Transaction rolled back"* means that there was an error and the transaction will be rolled back and not committed.
- The error message "*Error during rollback"* means that there was an error during the rollback process.
- The error message "*Error closing resources"* means that there was an error during the closing process of executing the query.
- The error message "*Invalid input. Please enter a valid integer."* means that there was an error when entering an integer statement.
- The error message "*Invalid input. Please enter a valid number."* means that there was an error when entering a number statement.