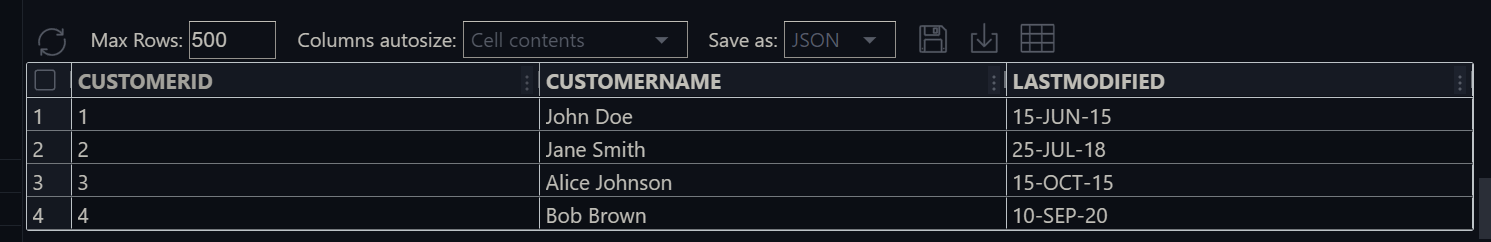
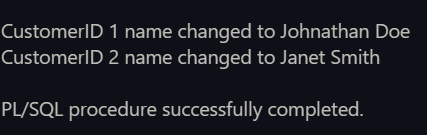
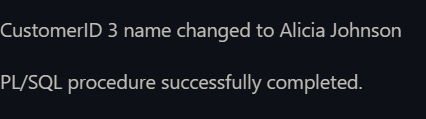
**Exercise 5: Triggers**

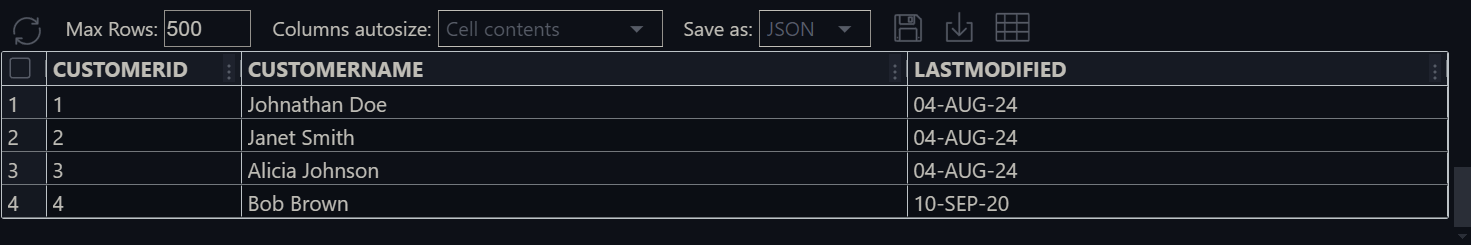
* **Scenario 1:** Automatically update the last modified date when a customer's record is updated.
* **Question:** Write a trigger ***UpdateCustomerLastModified*** that updates the *LastModified* column of the Customers table to the current date whenever a customer's record is updated.
* **Connection and Permissions**: The script connects to an Oracle database as SYSDBA to grant execute permission on DBMS\_*LOCK* to a non-SYS user and sets unlimited quota on the user schema.
* **Table Creation**: Creates a *Customer* table with columns *CustomerID, CustomerName*, and *LastModified* if it doesn't already exist.
* **Trigger Creation**: Defines a trigger *UpdateCustomerLastModified* that automatically updates the *LastModified* column to the current date whenever a record in the Customer table is updated.
* **Data Manipulation**: Inserts several customer records into the *Customer* table and then updates some of these records to observe changes in the *LastModified* column.
* **Verification**: Executes SQL statements to verify that the *LastModified* column is updated correctly after modifications, with additional *DBMS\_OUTPUT.PUT\_LINE* statements to print messages and DBMS\_LOCK.SLEEP to show the timestamp differences

**The Output of the code:**

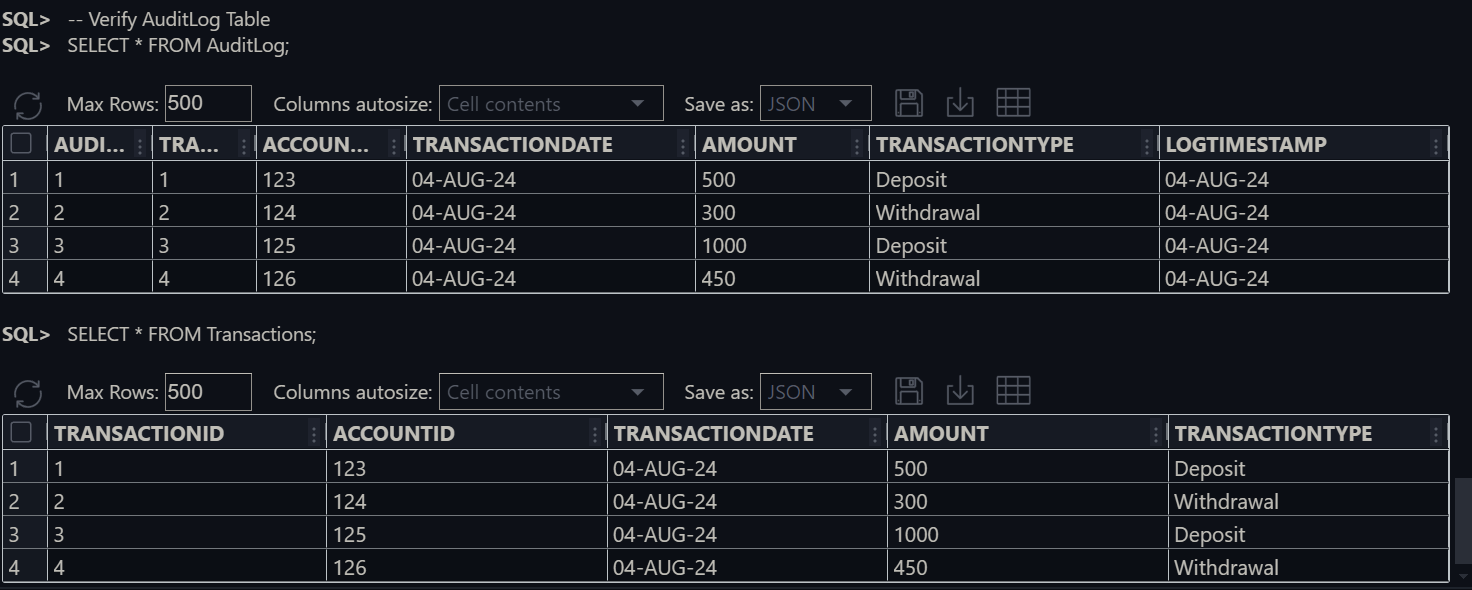
* Before modified the Customer table



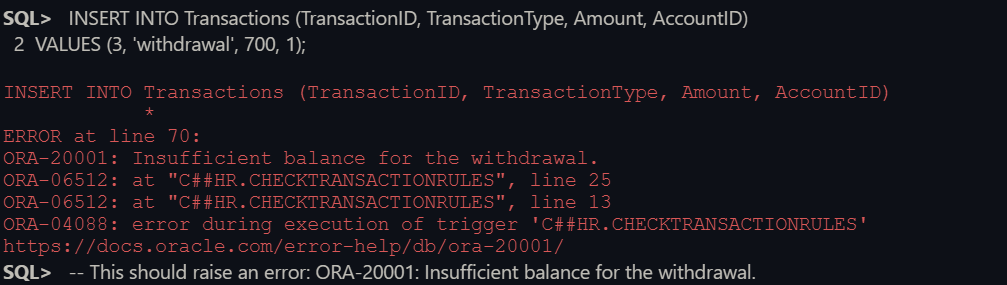
* Modify the *CustomerID* 1, 2, 3



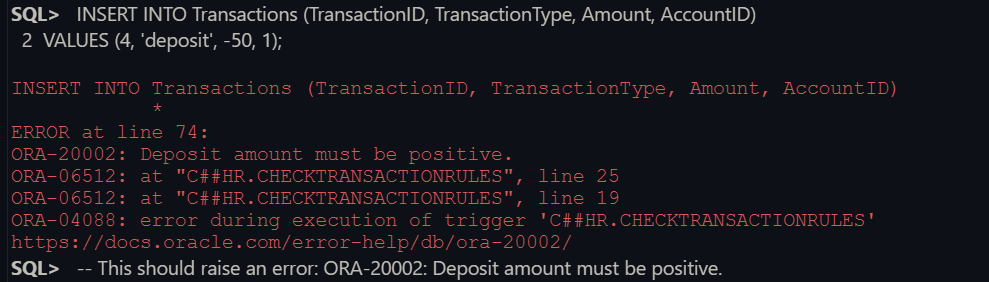
* After Modification the Customer table
* **Scenario 2:** Maintain an audit log for all transactions.
* **Question:** Write a trigger ***LogTransaction*** that inserts a record into an *AuditLog* table whenever a transaction is inserted into the Transactions table.
* **Connection and Quota**: Connects to the Oracle database as SYSDBA, grants unlimited quota on the USERS tablespace to the &&DB\_USER, and then reconnects as &&DB\_USER.
* **Table Creation**: Creates two tables, *AuditLog* and Transactions, to store transaction records and audit logs.
* **Sequence Creation**: Defines a sequence, *AuditLog*\_seq, to generate unique *AuditID* values for the *AuditLog* table.
* **Trigger Definition**: Creates a trigger, *LogTransaction*, which automatically inserts a record into *AuditLog* whenever a new row is added to the Transactions table.
* **Data Insertion and Verification**: Inserts sample transaction data into the Transactions table and then selects all records from both *AuditLog* and Transactions to verify the trigger’s operation.

The output of the code : The *Transactions* table and *AuditLog* table

* **Scenario 3:** Enforce business rules on deposits and withdrawals.
* **Question:** Write a trigger ***CheckTransactionRules*** that ensures withdrawals do not exceed the balance and deposits are positive before inserting a record into the Transactions table.
* **Database Connection and Permissions**: Connects to the database as SYSDBA to grant execute permissions on *DBMS*\_*LOCK* to DB\_*USER* and provides unlimited quota on the *USERS* tablespace for *DB*\_USER.
* **Table Creation**: Creates two tables—*Accounts* for storing account information and *Transactions* for recording transactions, with foreign key constraints linking *Transactions* to *Accounts*.
* **Insert Sample Data**: Inserts sample records into the *Accounts* table, and displays the contents of the table.
* **Trigger Creation**: Defines a trigger *CheckTransactionRules* that enforces rules on transactions before they are inserted. It checks for sufficient balance for withdrawals and positive deposit amounts, raising errors if rules are violated.
* **Insert Transactions and Validation**: Attempts to insert transactions into the *Transactions* table, which triggers error messages for invalid operations (e.g., insufficient balance and non-positive deposit amounts), followed by displaying the contents of both tables.

**The output of the code**:

The error is shown because of insufficient balance for the withdrawal



Here the error message is shown for the negative Deposit number



Here is the Account and Transactions table