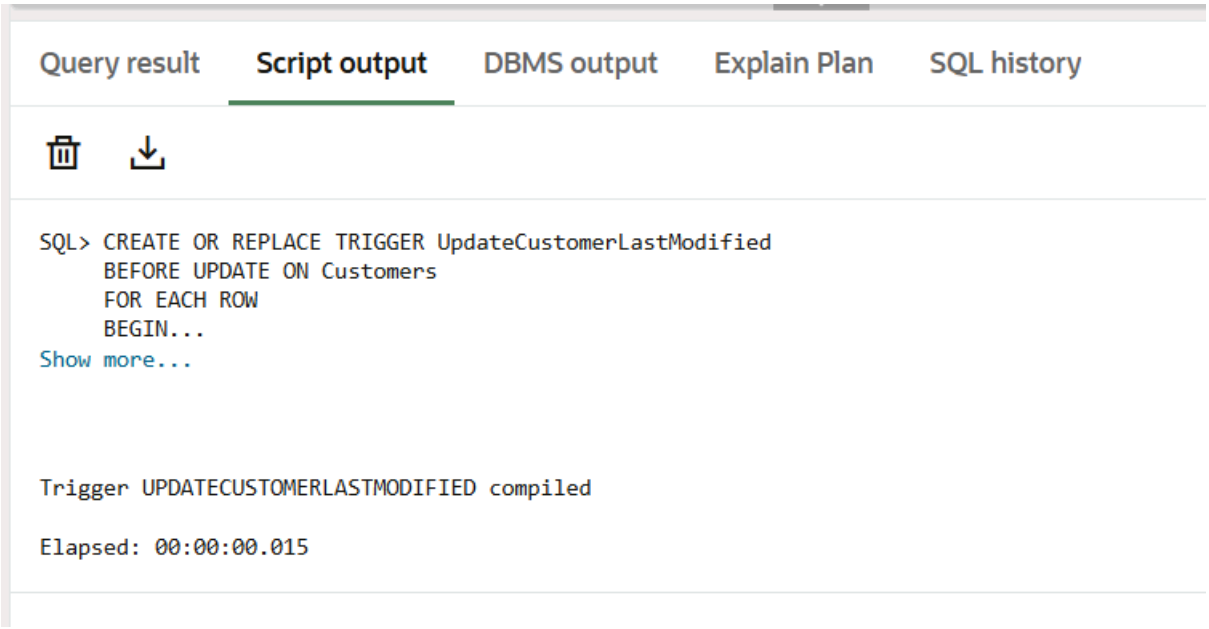
Exercise 5: Triggers

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.

Solution:

```
CREATE OR REPLACE TRIGGER UpdateCustomerLastModified
BEFORE UPDATE ON Customers
FOR EACH ROW
BEGIN
   :NEW.LastModified := SYSDATE;
END;
/
```



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.

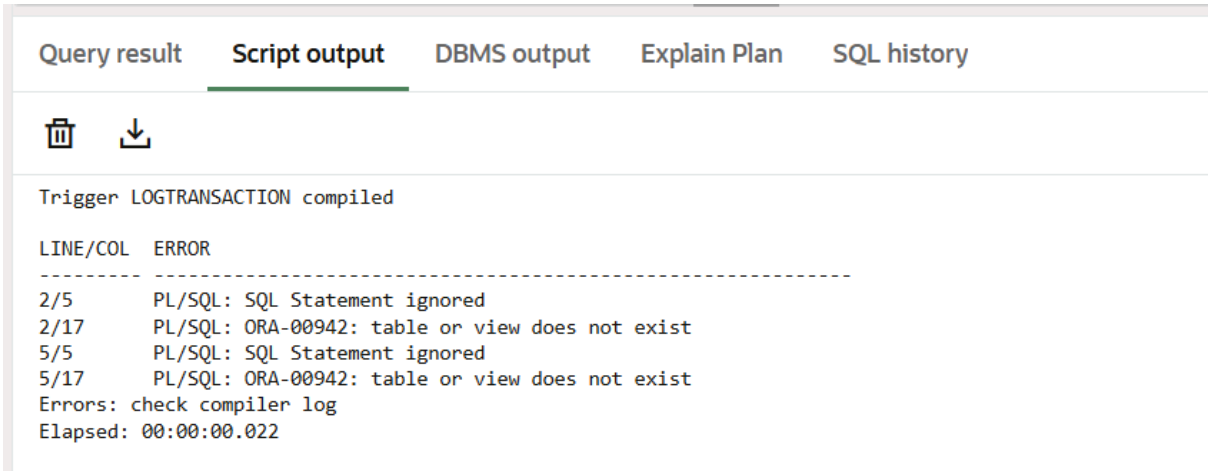
Solution:

```
CREATE TABLE AuditLog (
   LogID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
```

```
    TransactionID NUMBER,
    AccountID NUMBER,
    TransactionDate DATE,
    Amount NUMBER,
    TransactionType VARCHAR2(10),
    LoggedAt DATE DEFAULT SYSDATE
);

CREATE OR REPLACE TRIGGER LogTransaction
AFTER INSERT ON Transactions
FOR EACH ROW
BEGIN
    INSERT INTO AuditLog (TransactionID, AccountID, TransactionDate, Amount,
TransactionType)
    VALUES (:NEW.TransactionID, :NEW.AccountID, :NEW.TransactionDate,
:NEW.Amount, :NEW.TransactionType);

    INSERT INTO TriggerLog (LogMessage)
    VALUES ('Audit log created for Transaction ID: ' || :NEW.TransactionID);
END;
/
```



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.

Solution:

```
CREATE OR REPLACE TRIGGER CheckTransactionRules
BEFORE INSERT ON Transactions
FOR EACH ROW
DECLARE
   current_balance NUMBER;
BEGIN
   -- Get the current balance of the account
   SELECT Balance INTO current_balance
   FROM Accounts
   WHERE AccountID = :NEW.AccountID;

   -- Validate deposit
   IF :NEW.TransactionType = 'Deposit' THEN
      IF :NEW.Amount <= 0 THEN
         RAISE_APPLICATION_ERROR(-20001, 'Deposit amount must be positive');
      END IF;

   -- Validate withdrawal
   ELSIF :NEW.TransactionType = 'Withdrawal' THEN
      IF :NEW.Amount > current_balance THEN
         RAISE_APPLICATION_ERROR(-20002, 'Insufficient funds for withdrawal');
      END IF;
   END IF;
END;
/
```