**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.  
  
CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER,

IsVIP VARCHAR2(5),

LastModified DATE

);

INSERT INTO customers VALUES (1, 'Koushika', 25, 12000, 'FALSE', SYSDATE);

COMMIT;

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON customers

FOR EACH ROW

BEGIN

:NEW.LastModified := SYSDATE;

END;

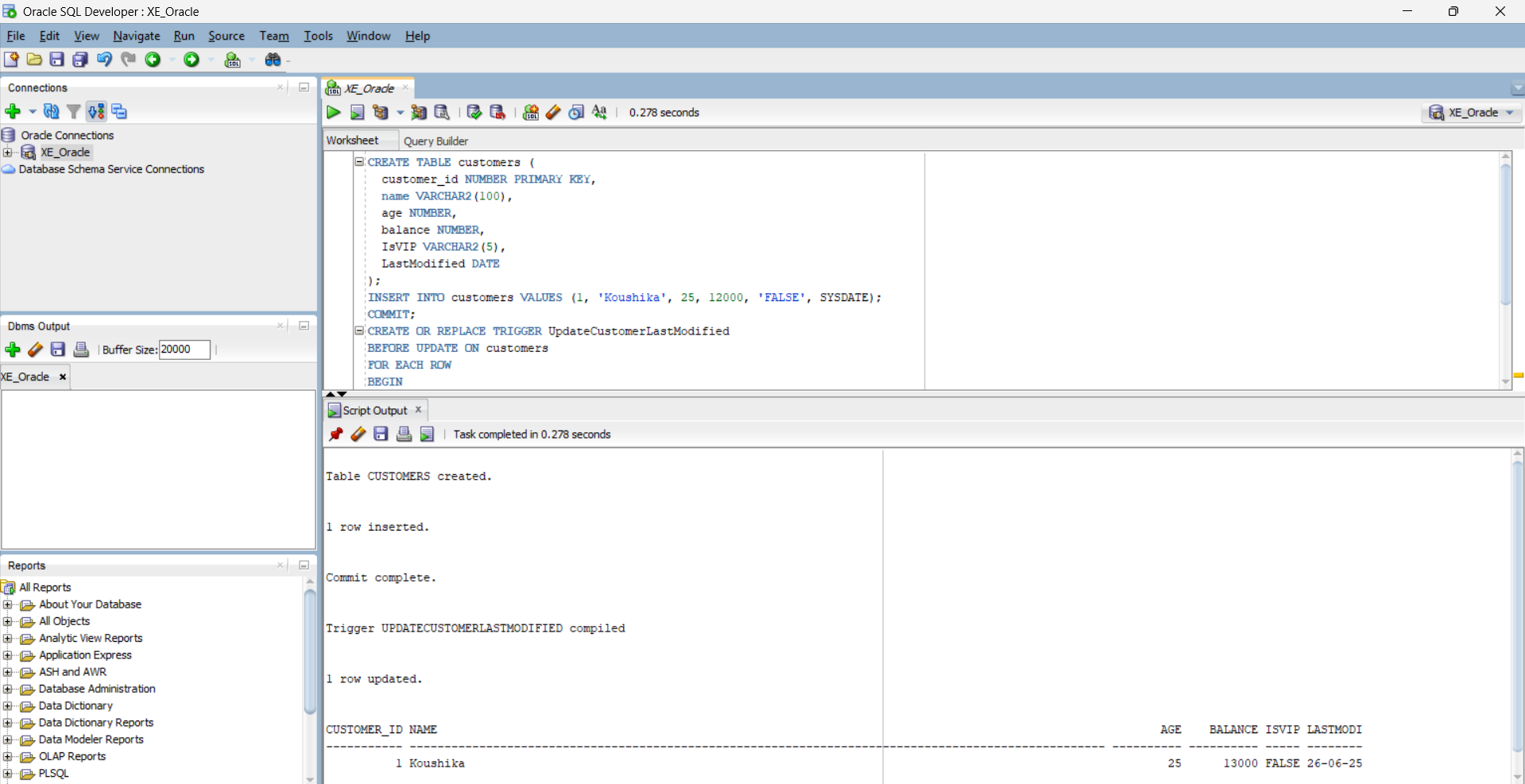
/

--example usage

UPDATE customers SET balance = balance + 1000 WHERE customer\_id = 1;

SELECT \* FROM customers WHERE customer\_id = 1;

OUTPUT:



**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.

CREATE TABLE transactions (

txn\_id NUMBER PRIMARY KEY,

acc\_no NUMBER,

txn\_type VARCHAR2(20),

amount NUMBER,

txn\_date DATE

);

CREATE TABLE audit\_log (

log\_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

txn\_id NUMBER,

log\_time DATE,

action VARCHAR2(50)

);

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON transactions

FOR EACH ROW

BEGIN

INSERT INTO audit\_log (txn\_id, log\_time, action)

VALUES (:NEW.txn\_id, SYSDATE, 'Transaction Inserted');

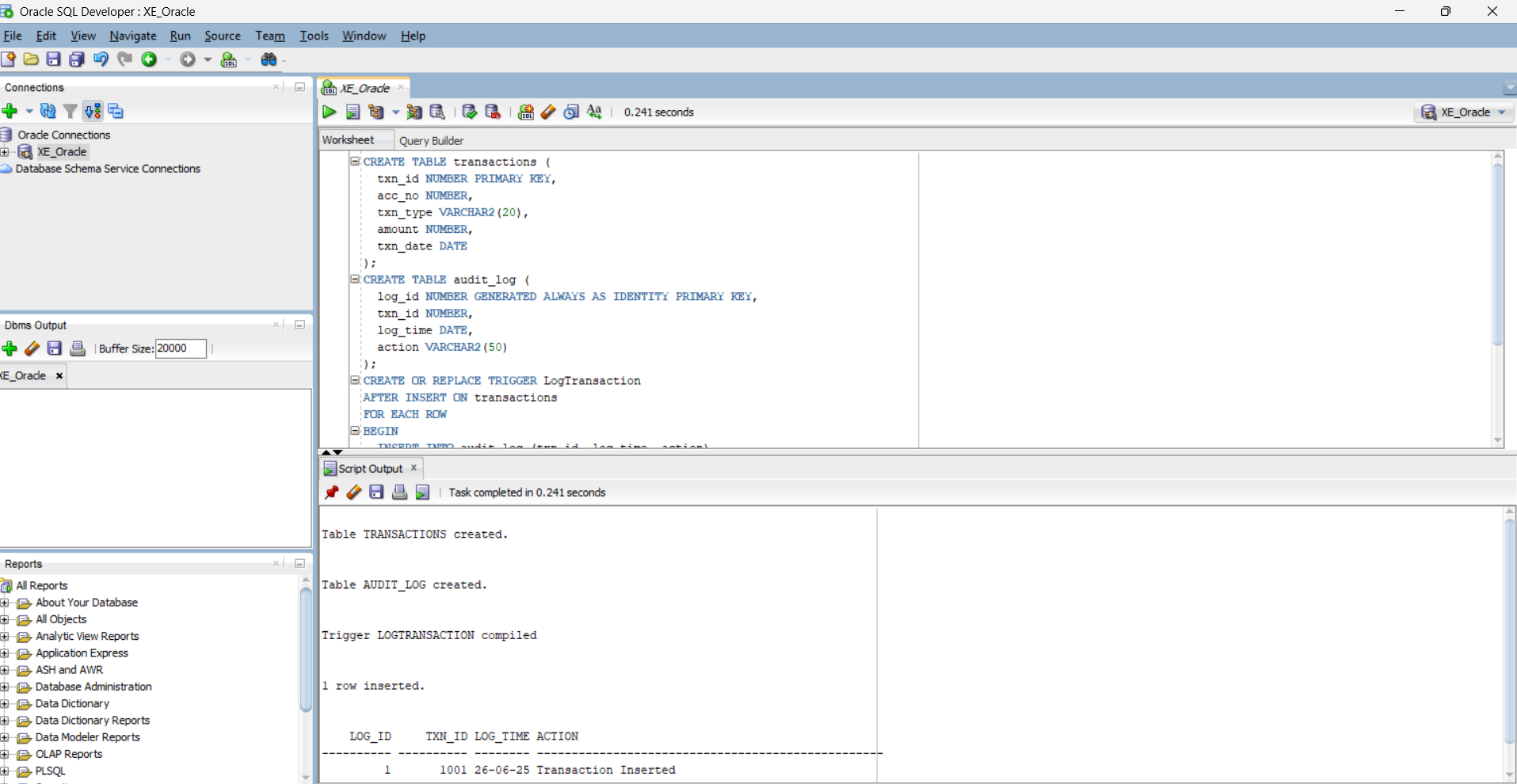
END;

/

--example usage

INSERT INTO transactions VALUES (1001, 401, 'deposit', 2000, SYSDATE);

SELECT \* FROM audit\_log;

OUTPUT:  


**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.  
  
CREATE TABLE accounts (

acc\_no NUMBER PRIMARY KEY,

holder\_name VARCHAR2(100),

balance NUMBER

);

CREATE TABLE transactions (

txn\_id NUMBER PRIMARY KEY,

acc\_no NUMBER,

txn\_type VARCHAR2(20),

amount NUMBER,

txn\_date DATE

);

INSERT INTO accounts VALUES (401, 'Koushika', 8000);

COMMIT;

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON transactions

FOR EACH ROW

DECLARE

acc\_balance NUMBER;

BEGIN

SELECT balance INTO acc\_balance

FROM accounts

WHERE acc\_no = :NEW.acc\_no;

IF :NEW.txn\_type = 'withdraw' AND :NEW.amount > acc\_balance THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Withdrawal exceeds current balance.');

END IF;

IF :NEW.txn\_type = 'deposit' AND :NEW.amount <= 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Deposit amount must be positive.');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Account not found.');

END;

/

--example usage

INSERT INTO transactions VALUES (1001, 401, 'deposit', 1000, SYSDATE);

OUTPUT:

