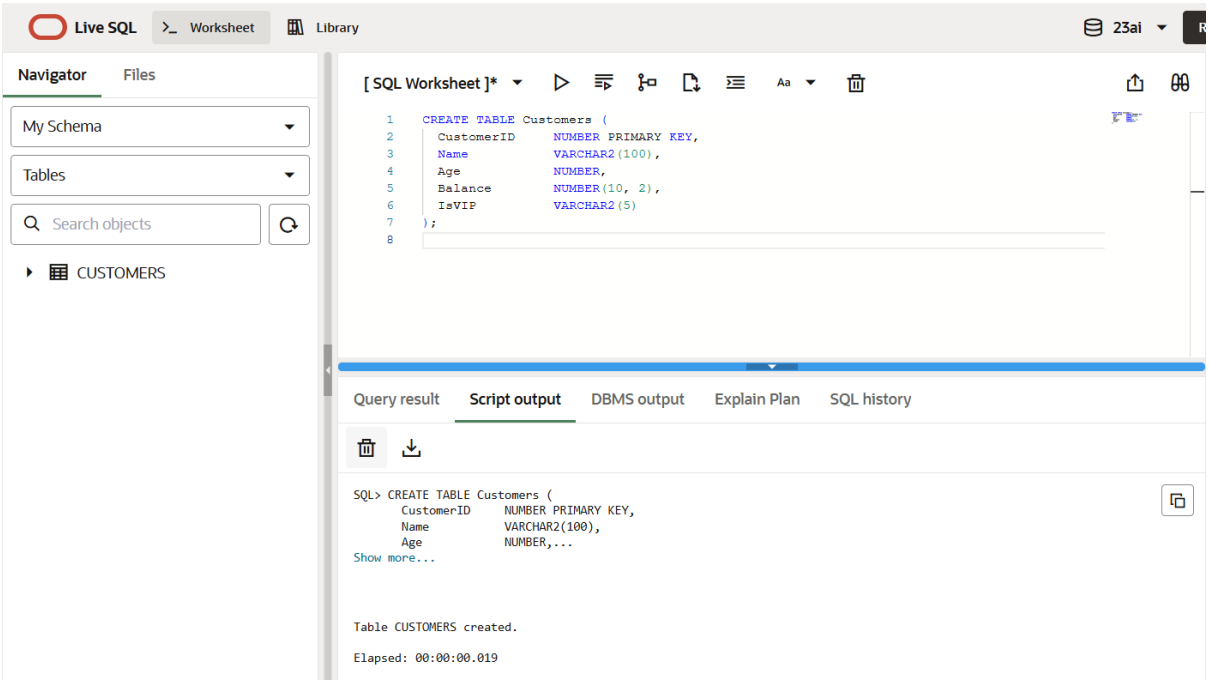
Name: SASHMITHA R

Reg No: 727722EUCD042

Exercise 1: Control Structures

Table Creation: Customers



Inserting values in Customers table

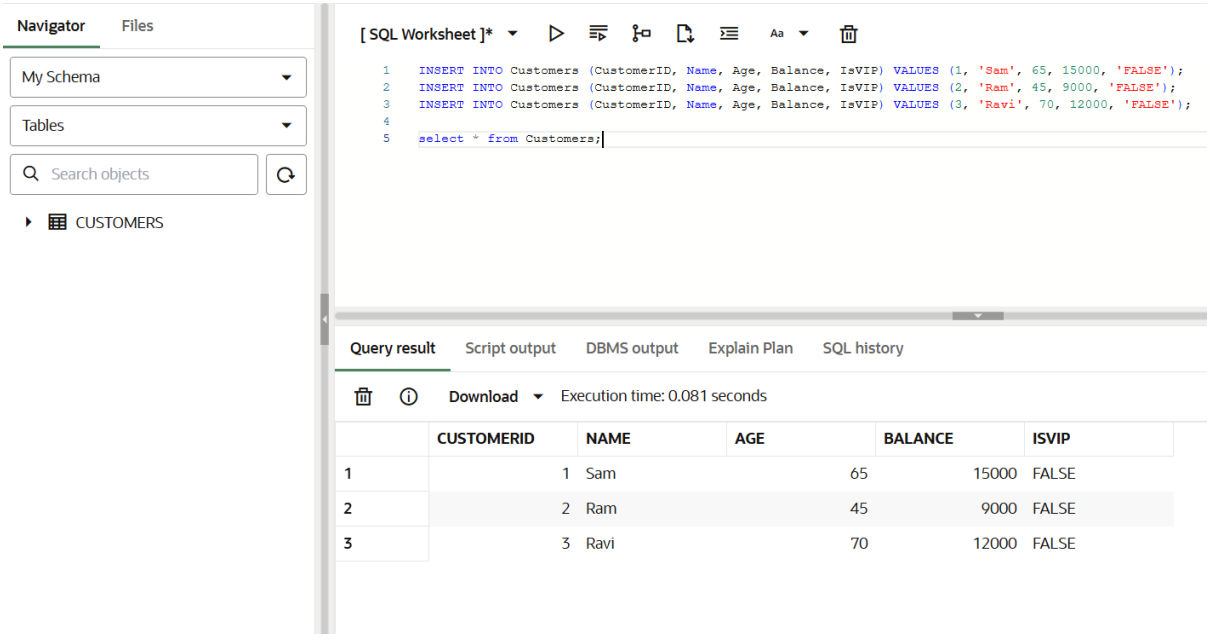
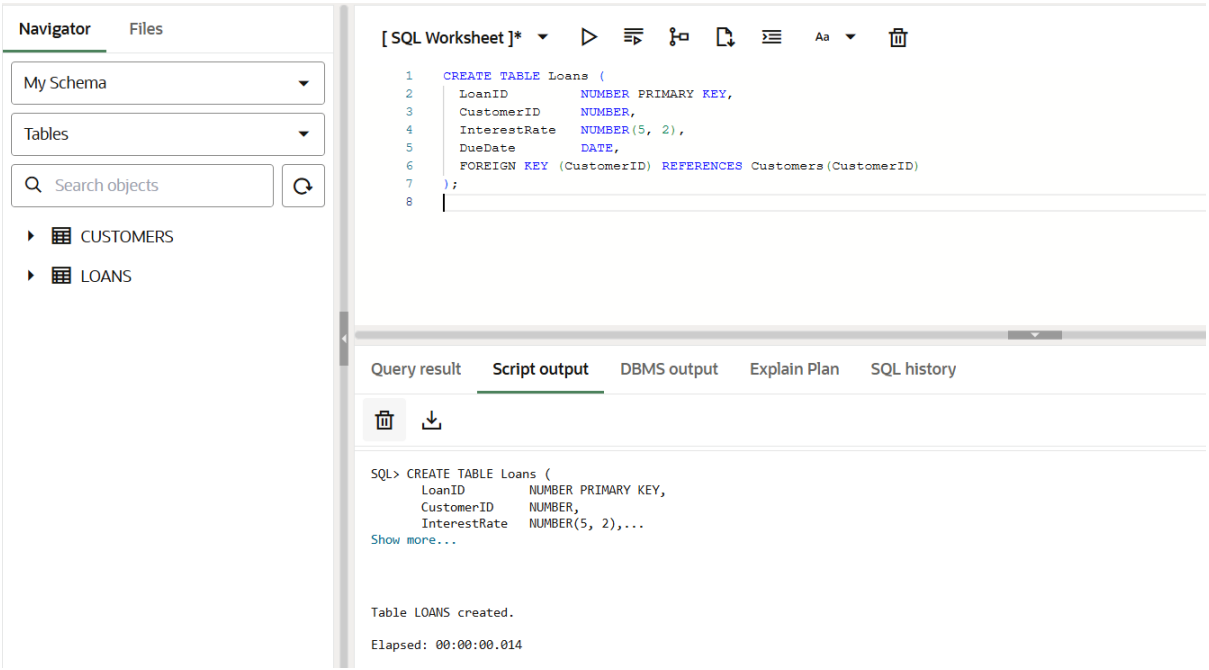
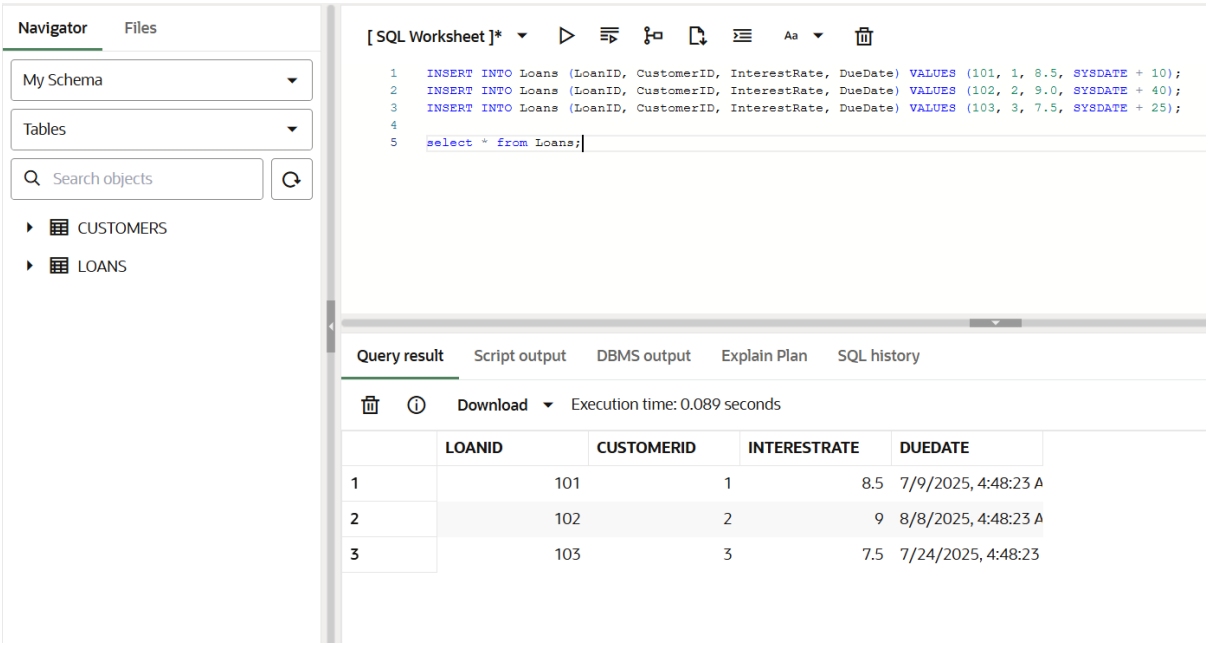


Table Creation: Loans



Inserting values in Loans table

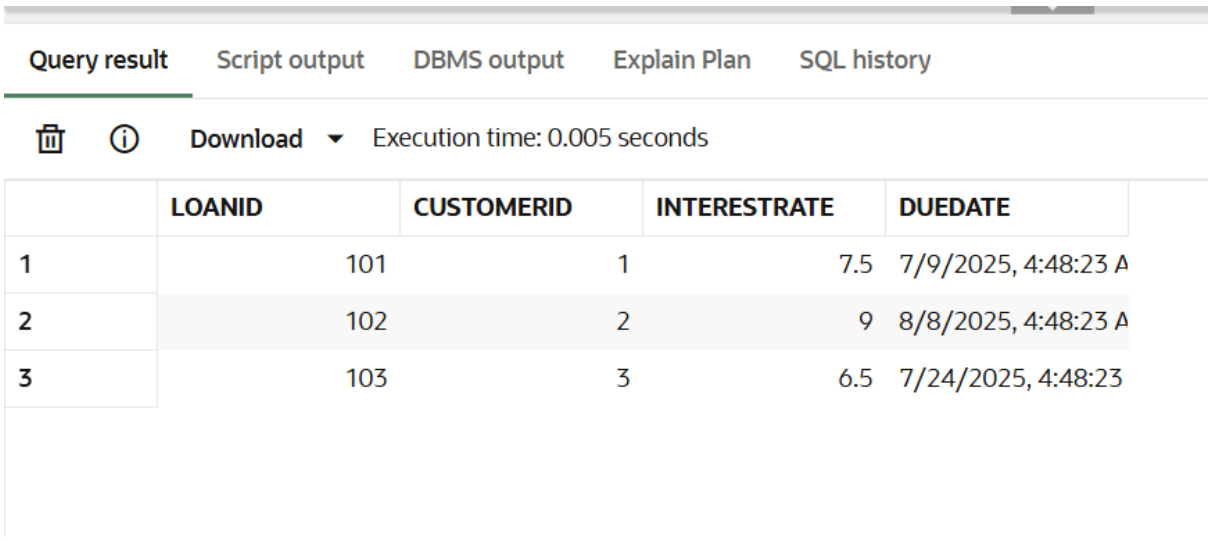


Scenario 1: Apply 1% discount to loan interest rates for customers above 60 years old

```
BEGIN
  FOR rec IN (SELECT c.CustomerID, l.LoanID, l.InterestRate
              FROM Customers c
              JOIN Loans l ON c.CustomerID = l.CustomerID
              WHERE c.Age > 60)
  LOOP
    UPDATE Loans
    SET InterestRate = InterestRate - 1
    WHERE LoanID = rec.LoanID;
  END LOOP;
  COMMIT;
END;

SELECT * FROM Loans;
```

Output :



Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.005 seconds

	LOANID	CUSTOMERID	INTERESTRATE	DUEDATE
1	101	1	7.5	7/9/2025, 4:48:23 A
2	102	2	9	8/8/2025, 4:48:23 A
3	103	3	6.5	7/24/2025, 4:48:23

Scenario 2: A customer can be promoted to VIP status based on their balance.

```
BEGIN
  FOR rec IN (SELECT CustomerID
              FROM Customers
              WHERE Balance > 10000)
  LOOP
    UPDATE Customers
    SET IsVIP = 'TRUE'
    WHERE CustomerID = rec.CustomerID;
  END LOOP;
  COMMIT;
END;

SELECT * FROM Customers;
```

Output :

Query result | Script output | DBMS output | Explain Plan | SQL history

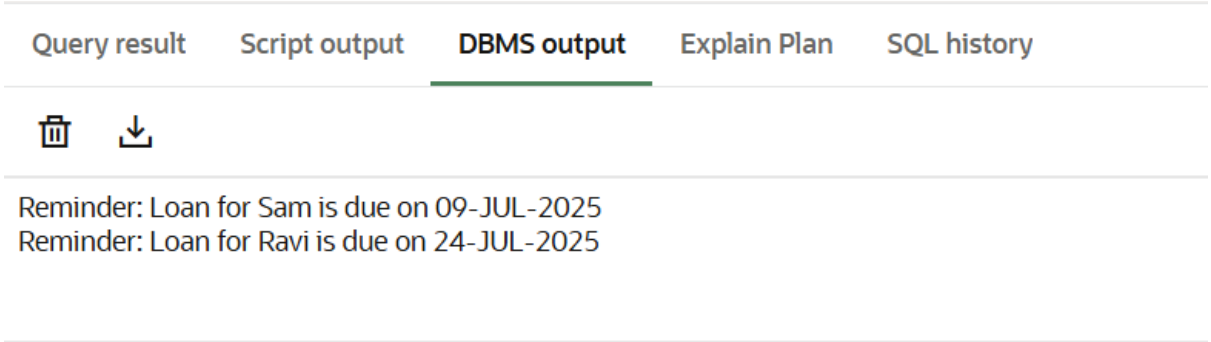
Download ▾ Execution time: 0.001 seconds

	CUSTOMERID	NAME	AGE	BALANCE	ISVIP
1	1	Sam	65	15000	TRUE
2	2	Ram	45	9000	FALSE
3	3	Ravi	70	12000	TRUE

Scenario 3: The bank wants to send reminders to customers whose loans are due within the next 30 days.

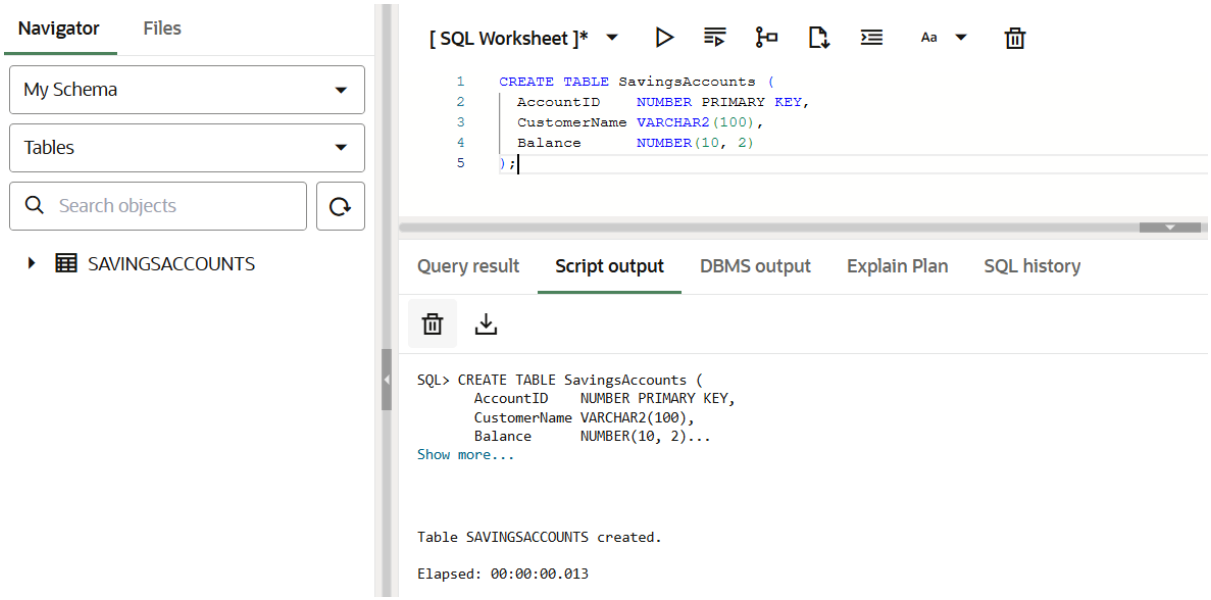
```
DECLARE
  v_due_date Loans.DueDate%TYPE;
  v_name Customers.Name%TYPE;
BEGIN
  FOR rec IN (
    SELECT c.Name, l.DueDate
    FROM Loans l
    JOIN Customers c ON c.CustomerID = l.CustomerID
    WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30
  )
  LOOP
    DBMS_OUTPUT.PUT_LINE('Reminder: Loan for ' || rec.Name || ' is due on ' || TO_CHAR(rec.DueDate, 'DD-MON-YYYY'));
  END LOOP;
END;
```

Output :



Exercise 3: Stored Procedures

Table Creation: SavingsAccounts



Inserting values in SavingsAccounts table

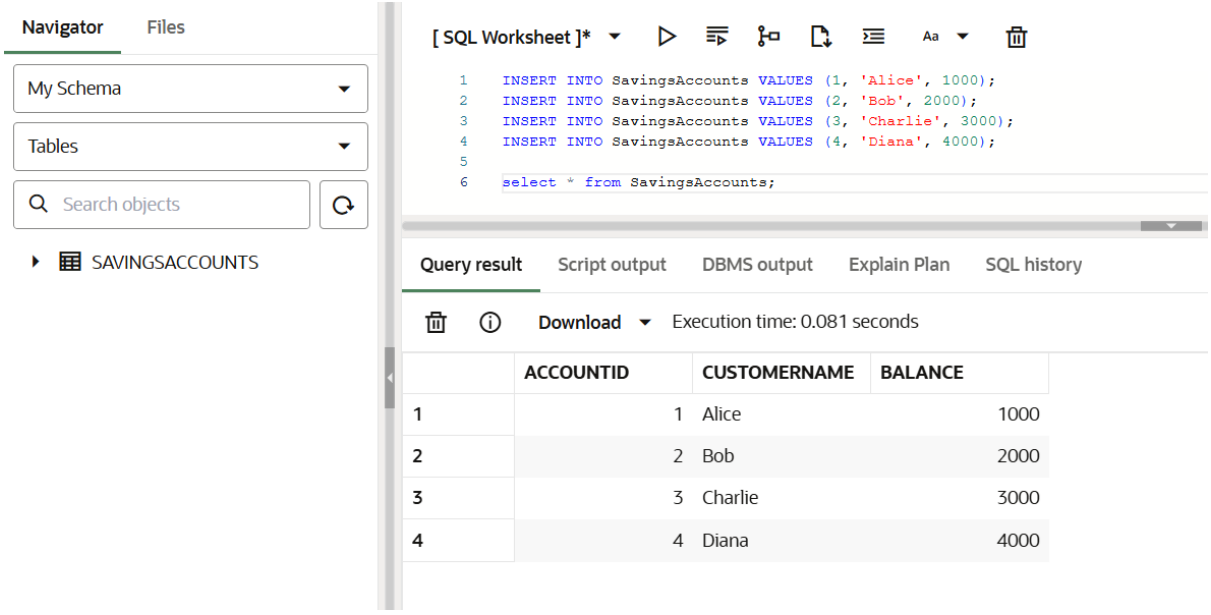
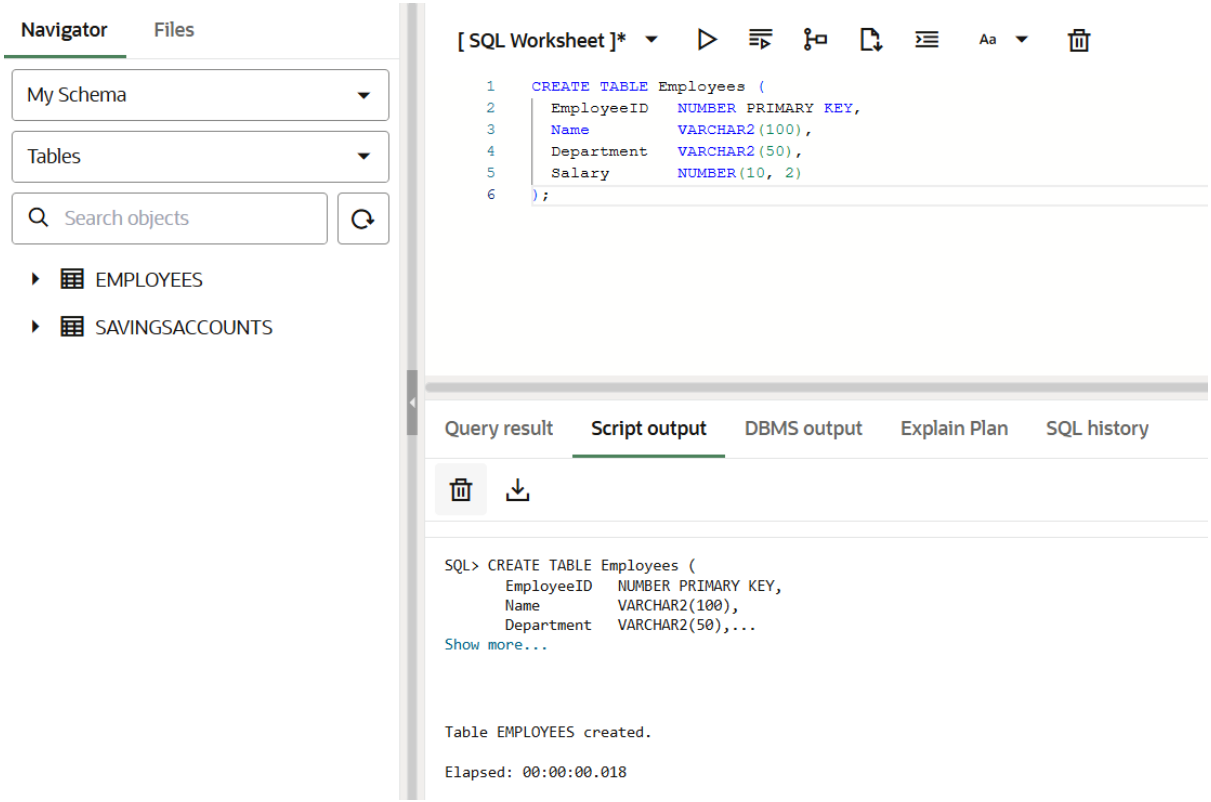


Table Creation: Employees



Inserting values in Employees table

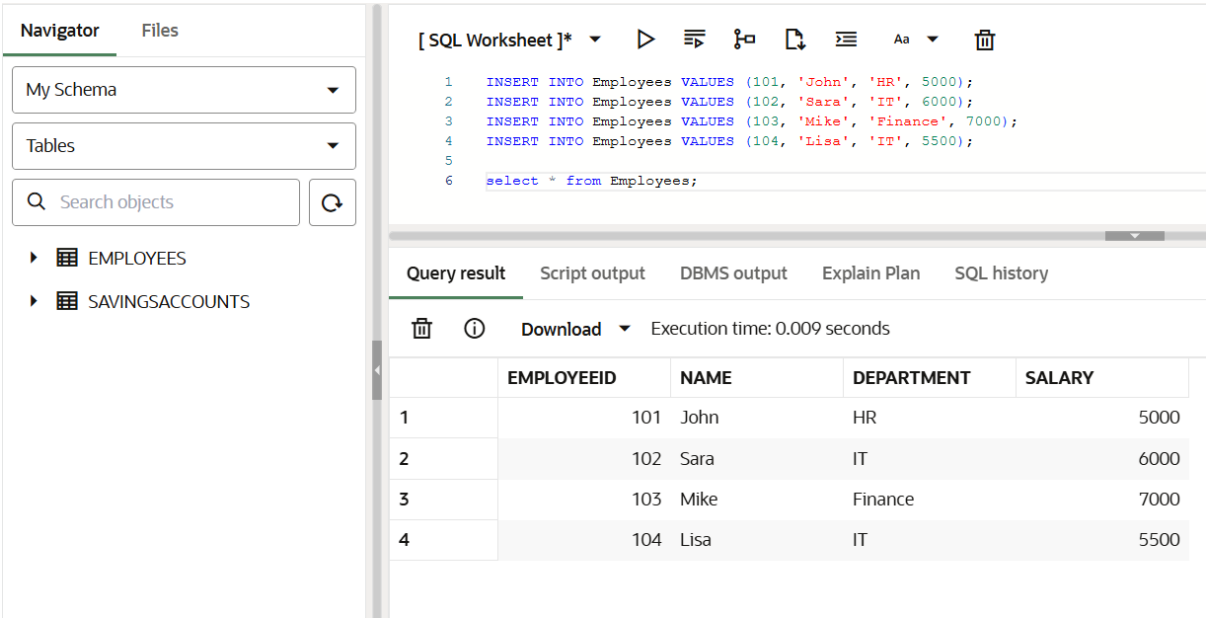
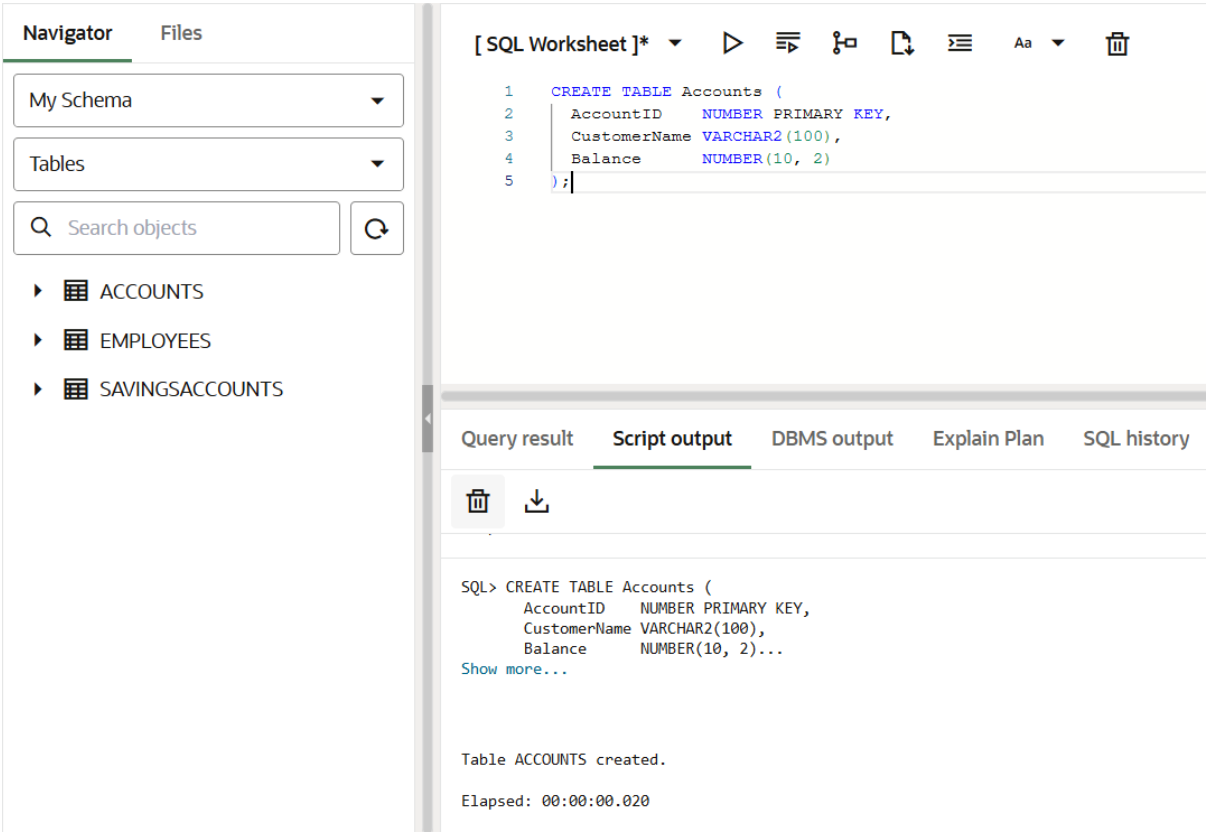
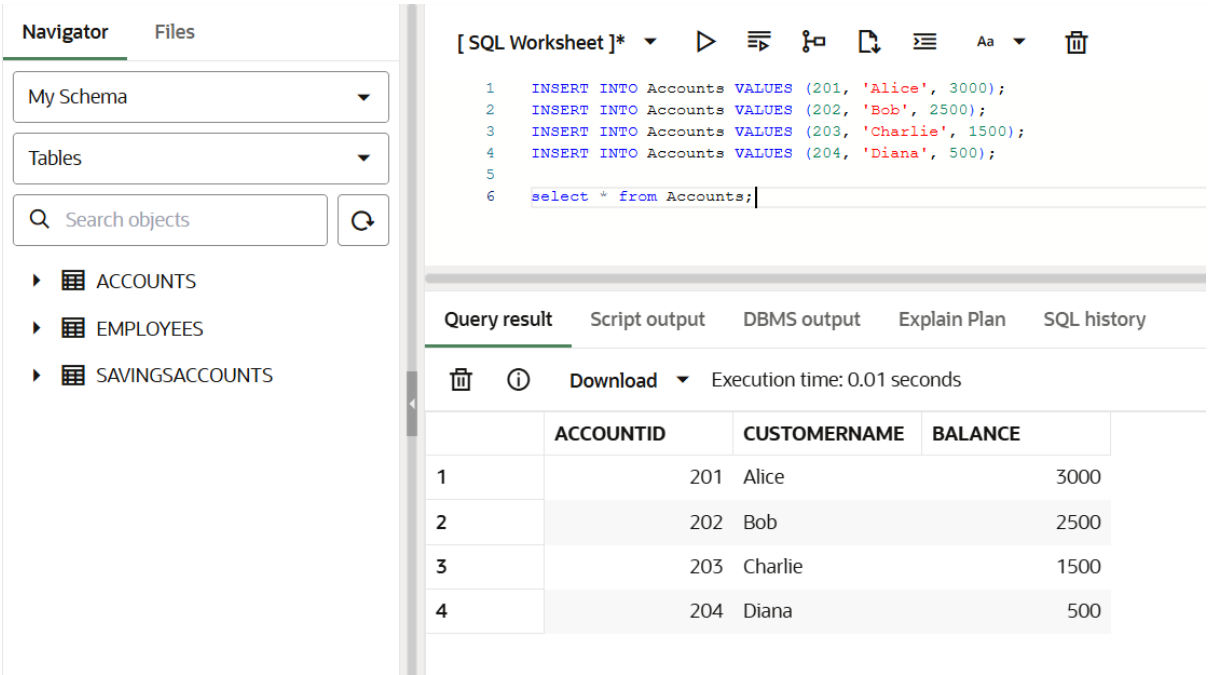


Table Creation: Accounts

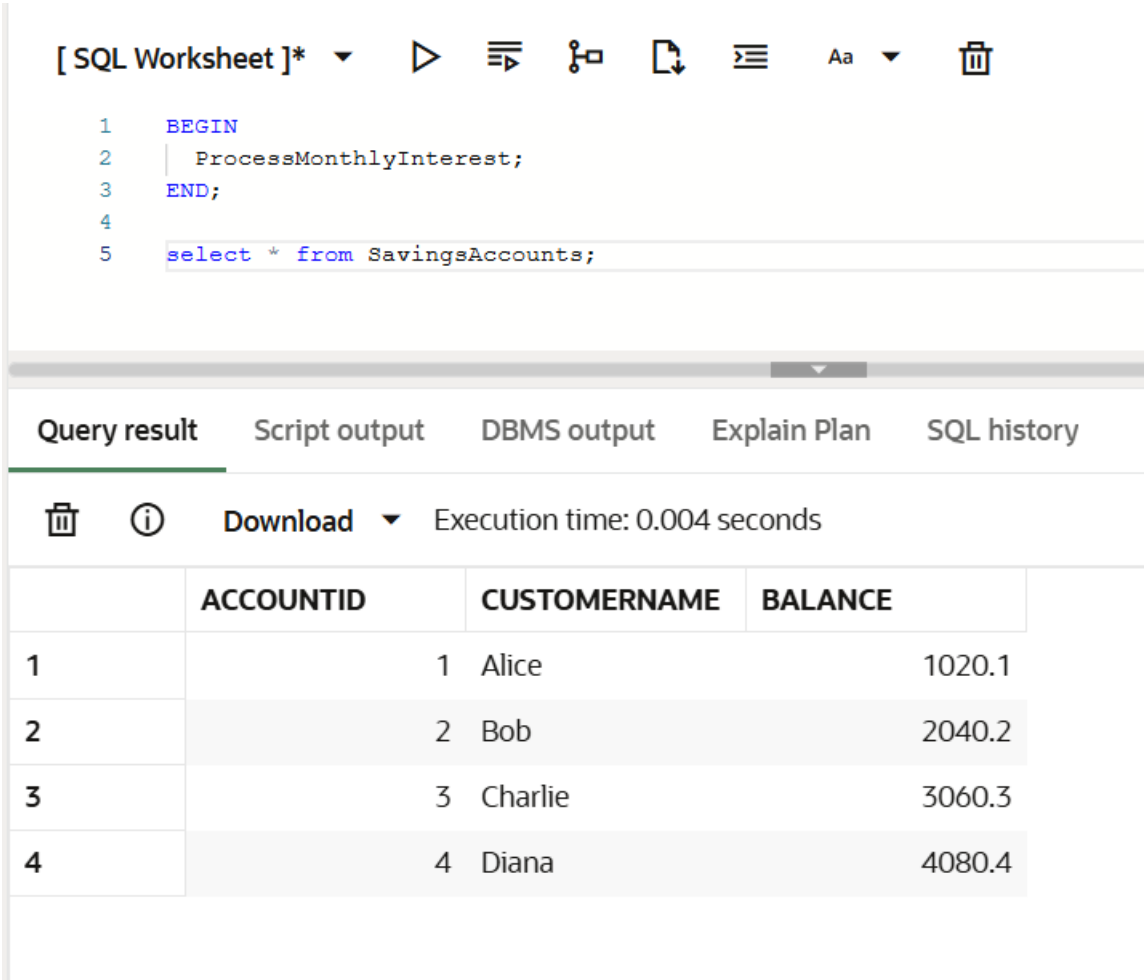


Inserting values in Accounts table



Scenario 1: The bank needs to process monthly interest for all savings accounts.

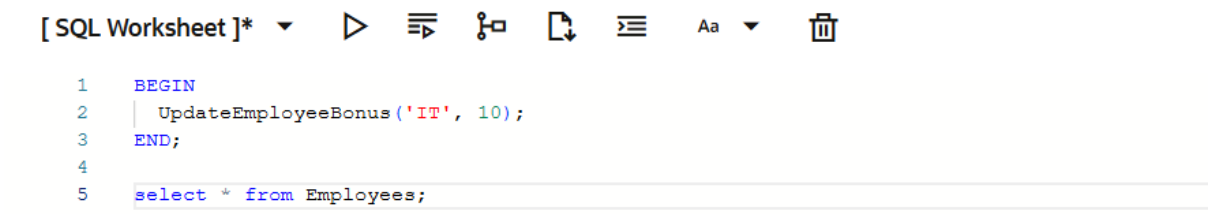
```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS
BEGIN
  FOR rec IN (SELECT AccountID, Balance FROM SavingsAccounts) LOOP
    UPDATE SavingsAccounts
    SET Balance = Balance + (Balance * 0.01)
    WHERE AccountID = rec.AccountID;
  END LOOP;
  COMMIT;
END;
```



	ACCOUNTID	CUSTOMERNAME	BALANCE
1	1	Alice	1020.1
2	2	Bob	2040.2
3	3	Charlie	3060.3
4	4	Diana	4080.4

Scenario 2: The bank wants to implement a bonus scheme for employees based on their performance.

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (
  p_department IN VARCHAR2,
  p_bonus_percent IN NUMBER
) IS
BEGIN
  UPDATE Employees
  SET Salary = Salary + (Salary * p_bonus_percent / 100)
  WHERE Department = p_department;
  COMMIT;
END;
```



Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.001 seconds

	EMPLOYEEID	NAME	DEPARTMENT	SALARY
1	101	John	HR	5000
2	102	Sara	IT	6600
3	103	Mike	Finance	7000
4	104	Lisa	IT	6050

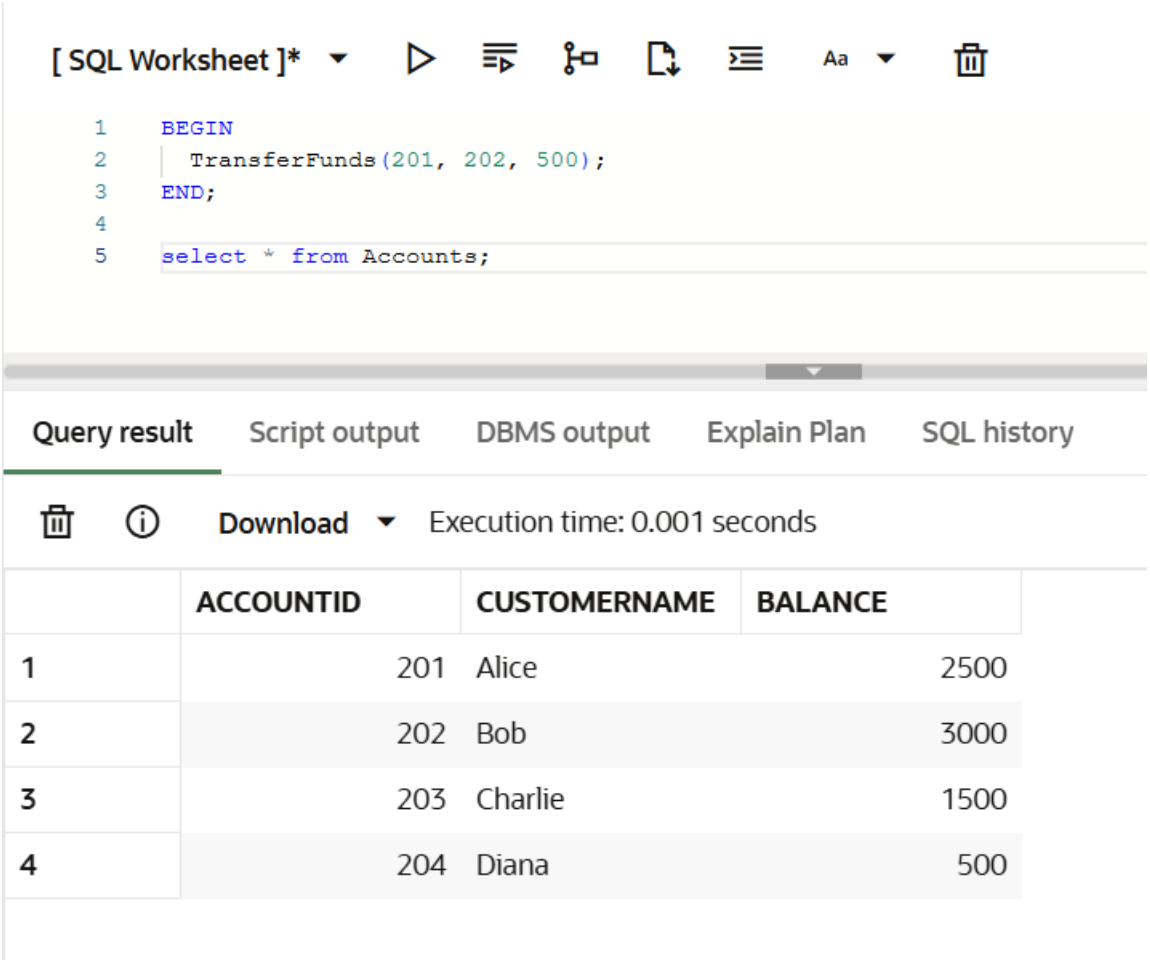
Scenario 3: Customers should be able to transfer funds between their accounts.

```
CREATE OR REPLACE PROCEDURE TransferFunds (
  p_from_account IN NUMBER,  p_to_account   IN NUMBER,
  p_amount      IN NUMBER )  IS  v_balance NUMBER;
BEGIN
  SELECT Balance INTO v_balance FROM Accounts WHERE AccountID = p_from_account;

IF v_balance < p_amount THEN
    RAISE_APPLICATION_ERROR(-20001, 'Insufficient balance in source account');
END IF;

UPDATE Accounts  SET Balance = Balance - p_amount  WHERE AccountID = p_from_account;

UPDATE Accounts  SET Balance = Balance + p_amount  WHERE AccountID = p_to_account;
COMMIT;
```



	ACCOUNTID	CUSTOMERNAME	BALANCE
1	201	Alice	2500
2	202	Bob	3000
3	203	Charlie	1500
4	204	Diana	500