WEEK 2 : CONTROL STRUCTURES

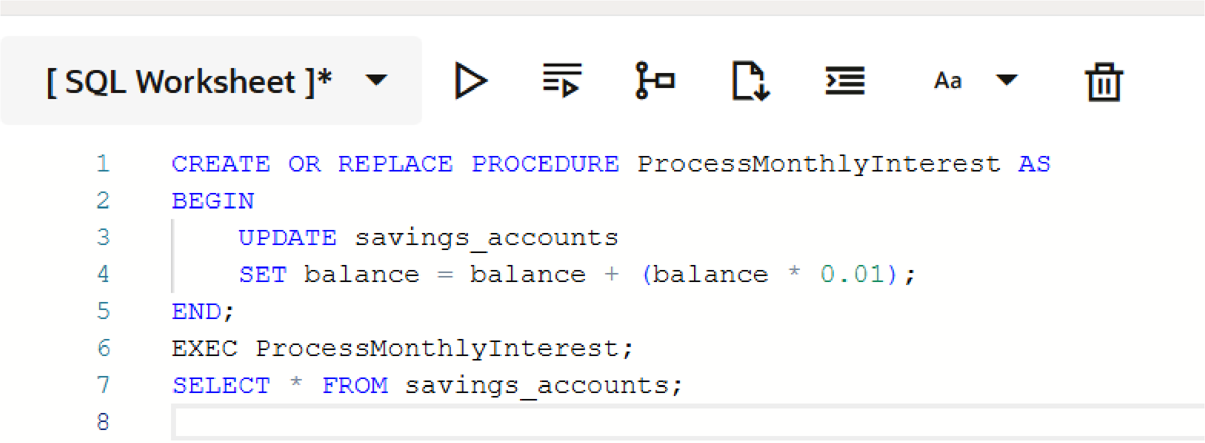
Exercise 3: Stored Procedures

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

- **Question: Write a stored procedure ProcessMonthlyInterest that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.**

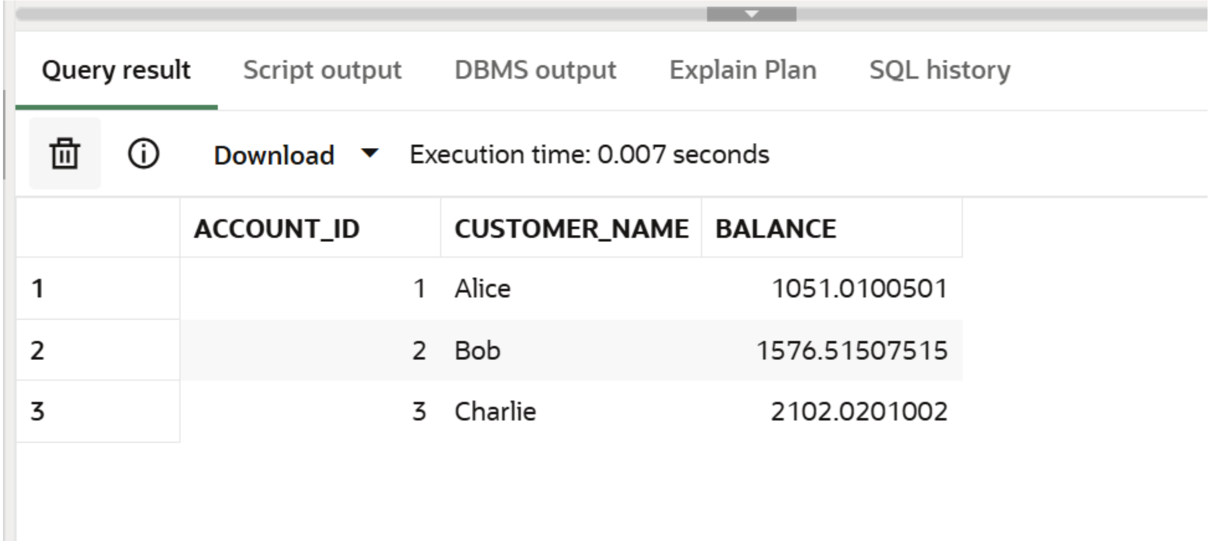
CODE :

```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

   UPDATE savings_accounts

   SET balance = balance + (balance * 0.01);

END;

EXEC ProcessMonthlyInterest;

SELECT * FROM savings_accounts;
```



```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS
BEGIN
    UPDATE savings_accounts
    SET balance = balance + (balance * 0.01);
END;
EXEC ProcessMonthlyInterest;
SELECT * FROM savings_accounts;
```

OUTPUTS :



Execution time: 0.007 seconds

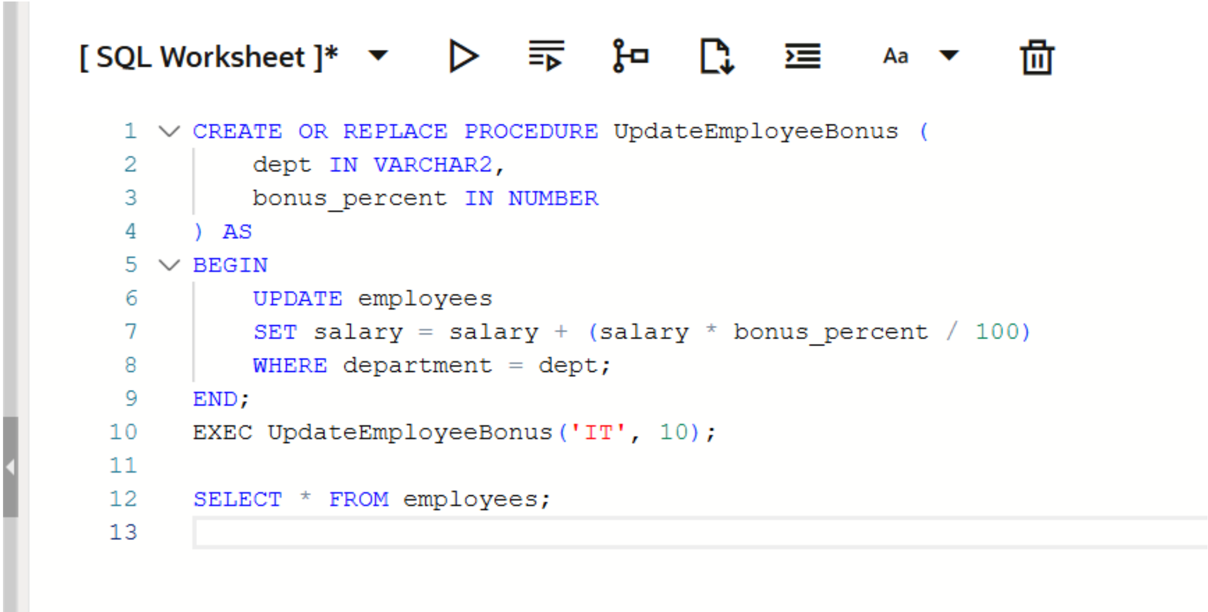
	ACCOUNT_ID	CUSTOMER_NAME	BALANCE
1	1	Alice	1051.0100501
2	2	Bob	1576.51507515
3	3	Charlie	2102.0201002

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

- **Question: Write a stored procedure UpdateEmployeeBonus that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.**

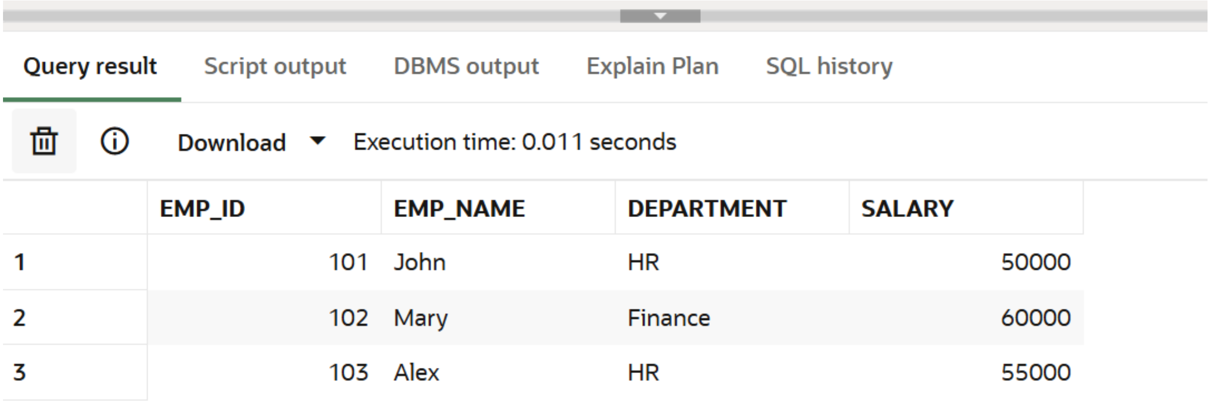
CODE :

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (
  dept IN VARCHAR2,
  bonus_percent IN NUMBER
) AS
BEGIN
  UPDATE employees
  SET salary = salary + (salary * bonus_percent / 100)
  WHERE department = dept;
END;
EXEC UpdateEmployeeBonus('IT', 10);
SELECT * FROM employees;
```



```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (
    dept IN VARCHAR2,
    bonus_percent IN NUMBER
) AS
BEGIN
    UPDATE employees
    SET salary = salary + (salary * bonus_percent / 100)
    WHERE department = dept;
END;
EXEC UpdateEmployeeBonus('IT', 10);

SELECT * FROM employees;
```

OUTPUT :



Execution time: 0.011 seconds

	EMP_ID	EMP_NAME	DEPARTMENT	SALARY
1	101	John	HR	50000
2	102	Mary	Finance	60000
3	103	Alex	HR	55000

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

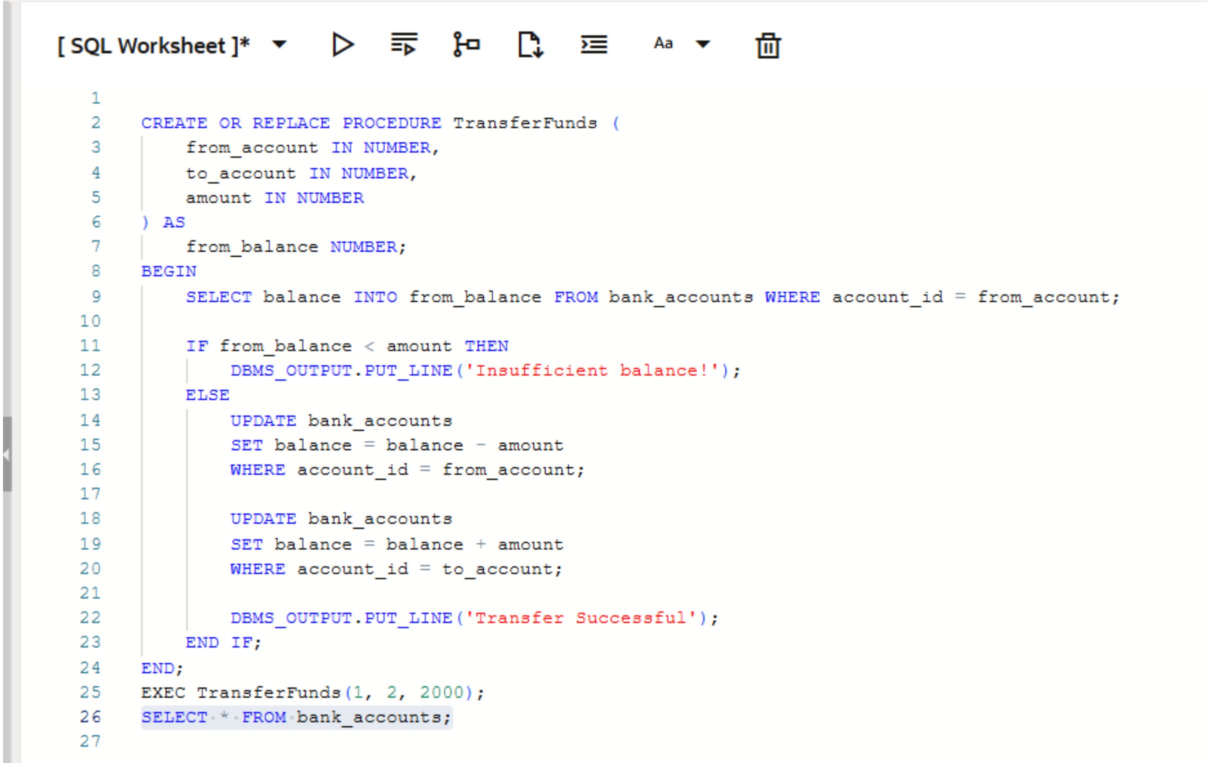
- **Question: Write a stored procedure TransferFunds that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.**

CODE :

CREATE OR REPLACE PROCEDURE TransferFunds (

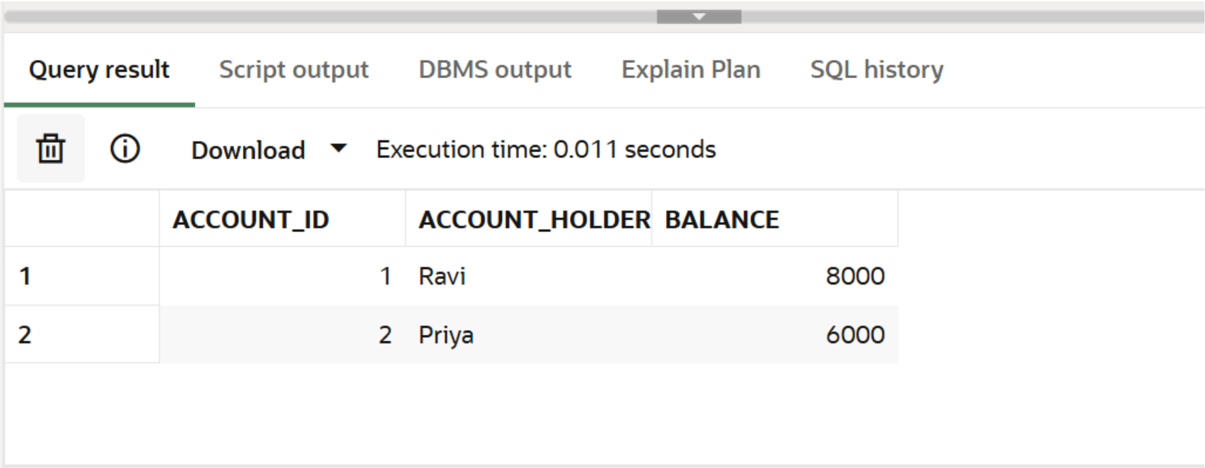
from_account IN NUMBER,

```
  to_account IN NUMBER,
  amount IN NUMBER
) AS
  from_balance NUMBER;
BEGIN
  SELECT balance INTO from_balance FROM bank_accounts WHERE account_id = from_account;

  IF from_balance < amount THEN
    DBMS_OUTPUT.PUT_LINE('Insufficient balance!');
  ELSE
    UPDATE bank_accounts
    SET balance = balance - amount
    WHERE account_id = from_account;

    UPDATE bank_accounts
    SET balance = balance + amount
    WHERE account_id = to_account;

    DBMS_OUTPUT.PUT_LINE('Transfer Successful');
  END IF;
END;
EXEC TransferFunds(1, 2, 2000);
SELECT * FROM bank_accounts;
```



```
CREATE OR REPLACE PROCEDURE TransferFunds (
    from_account IN NUMBER,
    to_account IN NUMBER,
    amount IN NUMBER
) AS
    from_balance NUMBER;
BEGIN
    SELECT balance INTO from_balance FROM bank_accounts WHERE account_id = from_account;

    IF from_balance < amount THEN
        DBMS_OUTPUT.PUT_LINE('Insufficient balance!');
    ELSE
        UPDATE bank_accounts
        SET balance = balance - amount
        WHERE account_id = from_account;

        UPDATE bank_accounts
        SET balance = balance + amount
        WHERE account_id = to_account;

        DBMS_OUTPUT.PUT_LINE('Transfer Successful');
    END IF;
END;
EXEC TransferFunds(1, 2, 2000);
SELECT * FROM bank_accounts;
```

OUTPUT:



Execution time: 0.011 seconds

	ACCOUNT_ID	ACCOUNT_HOLDER	BALANCE
1	1	Ravi	8000
2	2	Priya	6000