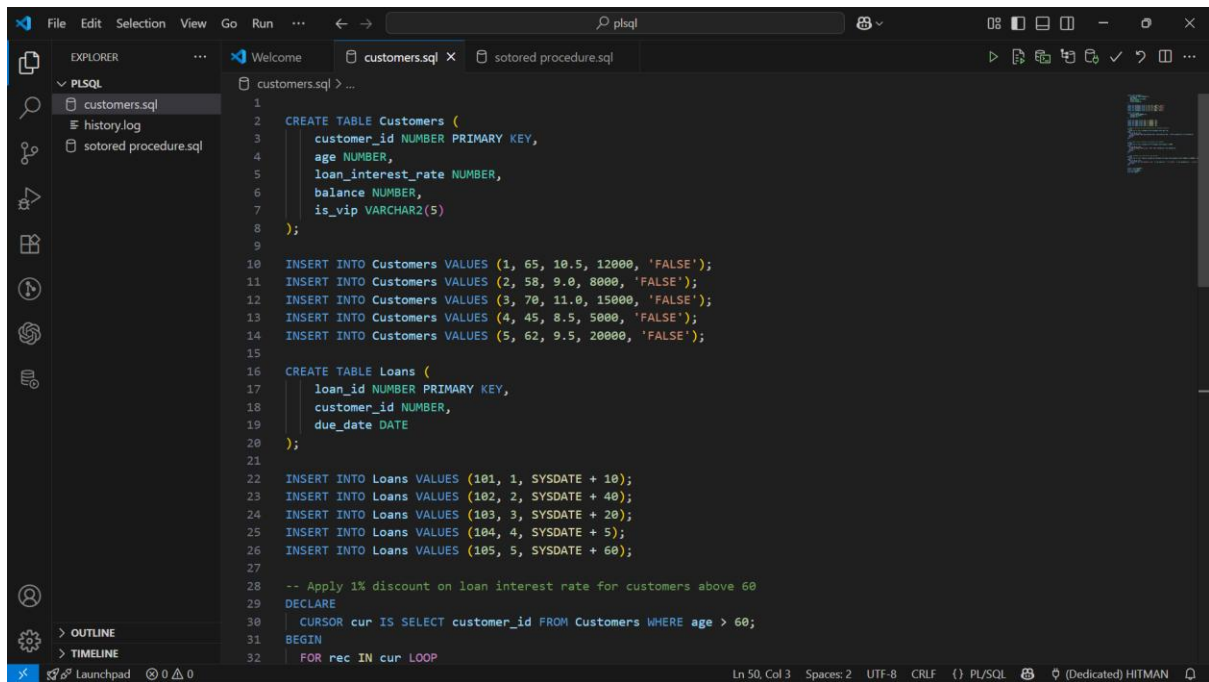
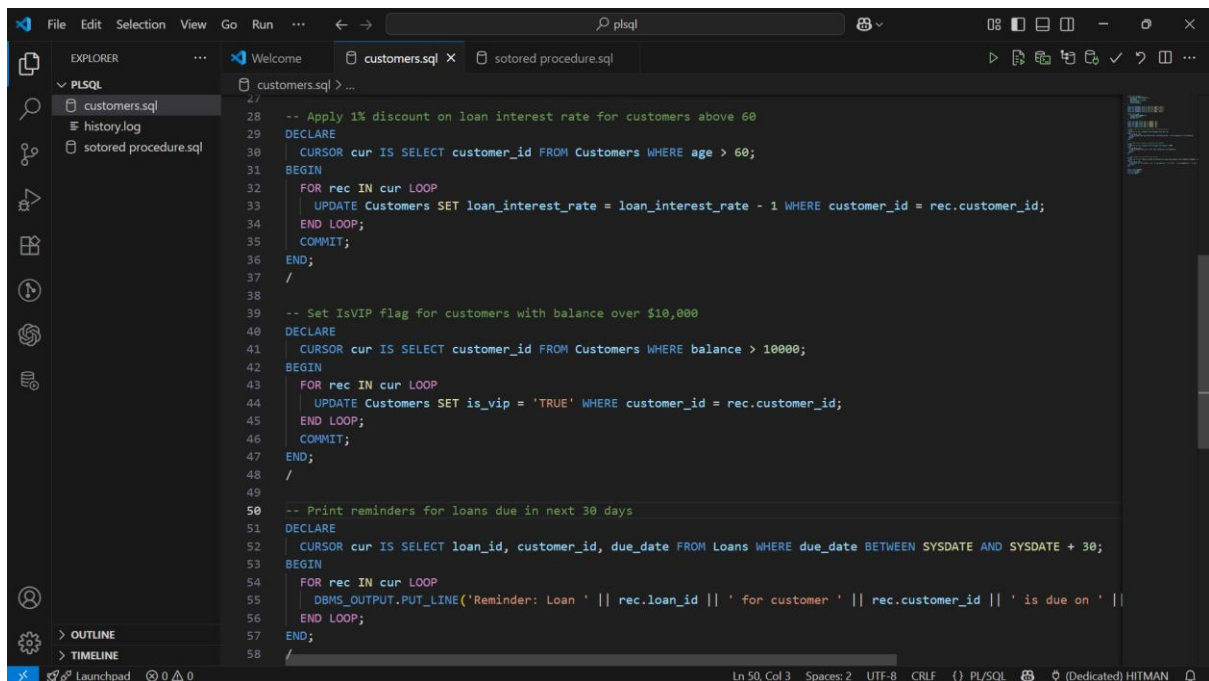


WEEK 2

CONTROL STRUCTURES

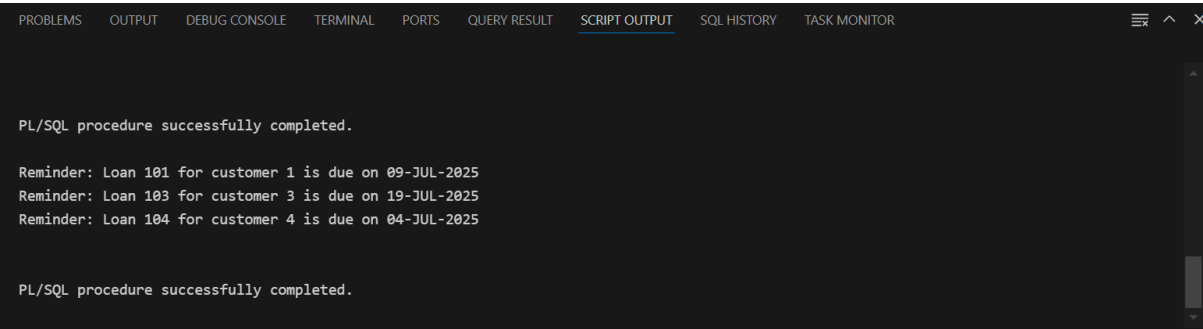
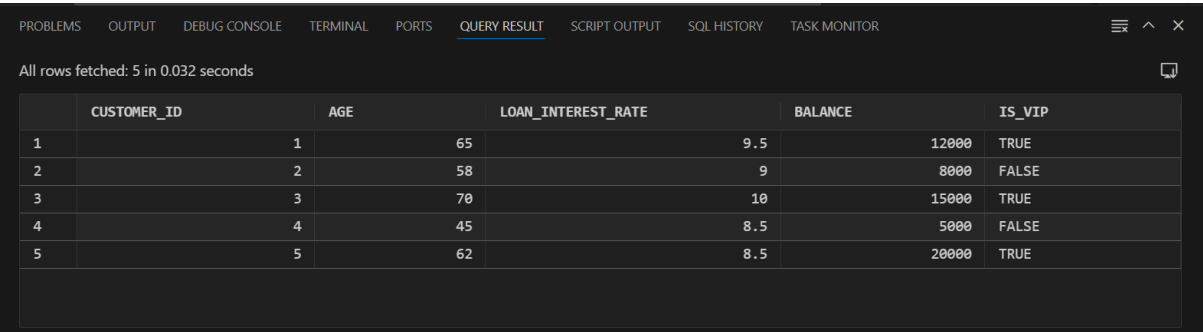
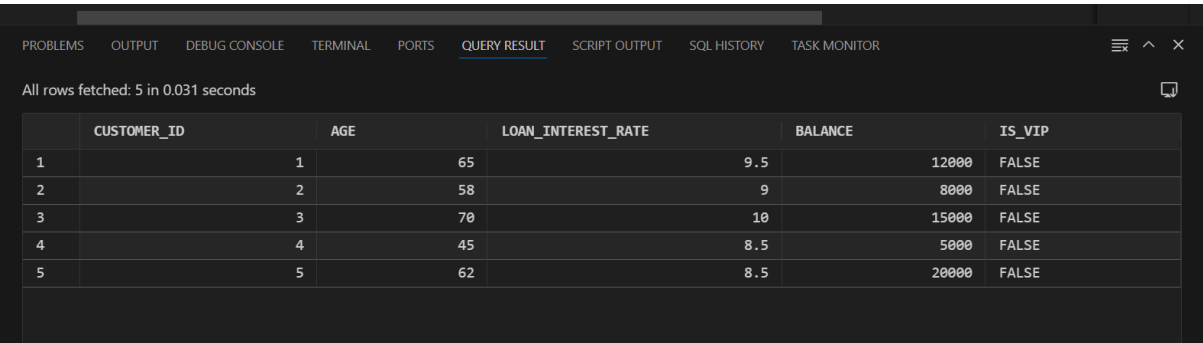
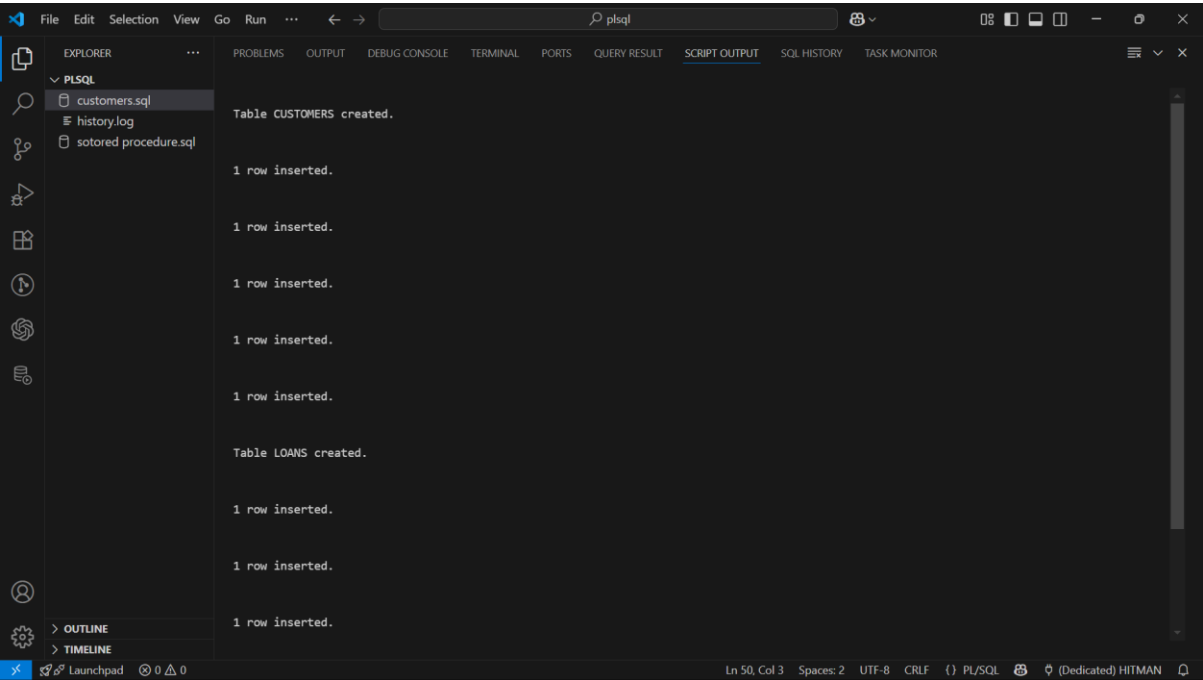


```
1
2 CREATE TABLE Customers (
3     customer_id NUMBER PRIMARY KEY,
4     age NUMBER,
5     loan_interest_rate NUMBER,
6     balance NUMBER,
7     is_vip VARCHAR2(5)
8 );
9
10 INSERT INTO Customers VALUES (1, 65, 10.5, 12000, 'FALSE');
11 INSERT INTO Customers VALUES (2, 58, 9.0, 8000, 'FALSE');
12 INSERT INTO Customers VALUES (3, 70, 11.0, 15000, 'FALSE');
13 INSERT INTO Customers VALUES (4, 45, 8.5, 5000, 'FALSE');
14 INSERT INTO Customers VALUES (5, 62, 9.5, 20000, 'FALSE');
15
16 CREATE TABLE Loans (
17     loan_id NUMBER PRIMARY KEY,
18     customer_id NUMBER,
19     due_date DATE
20 );
21
22 INSERT INTO Loans VALUES (101, 1, SYSDATE + 10);
23 INSERT INTO Loans VALUES (102, 2, SYSDATE + 40);
24 INSERT INTO Loans VALUES (103, 3, SYSDATE + 20);
25 INSERT INTO Loans VALUES (104, 4, SYSDATE + 5);
26 INSERT INTO Loans VALUES (105, 5, SYSDATE + 60);
27
28 -- Apply 1% discount on loan interest rate for customers above 60
29 DECLARE
30     CURSOR cur IS SELECT customer_id FROM Customers WHERE age > 60;
31 BEGIN
32     FOR rec IN cur LOOP
```

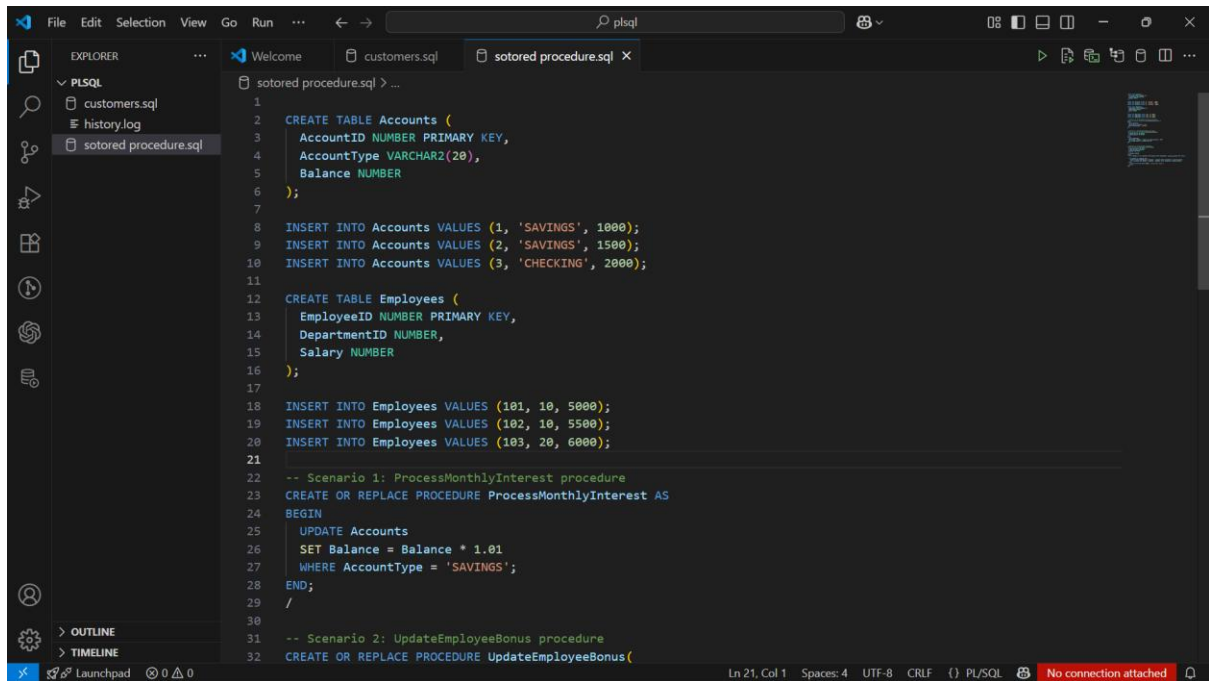


```
33         UPDATE Customers SET loan_interest_rate = loan_interest_rate - 1 WHERE customer_id = rec.customer_id;
34     END LOOP;
35     COMMIT;
36 END;
37 /
38
39 -- Set IsVIP flag for customers with balance over $10,000
40 DECLARE
41     CURSOR cur IS SELECT customer_id FROM Customers WHERE balance > 10000;
42 BEGIN
43     FOR rec IN cur LOOP
44         UPDATE Customers SET is_vip = 'TRUE' WHERE customer_id = rec.customer_id;
45     END LOOP;
46     COMMIT;
47 END;
48 /
49
50 -- Print reminders for loans due in next 30 days
51 DECLARE
52     CURSOR cur IS SELECT loan_id, customer_id, due_date FROM Loans WHERE due_date BETWEEN SYSDATE AND SYSDATE + 30;
53 BEGIN
54     FOR rec IN cur LOOP
55         DBMS_OUTPUT.PUT_LINE('Reminder: Loan ' || rec.loan_id || ' for customer ' || rec.customer_id || ' is due on ' ||
56     END LOOP;
57 END;
58 /
```

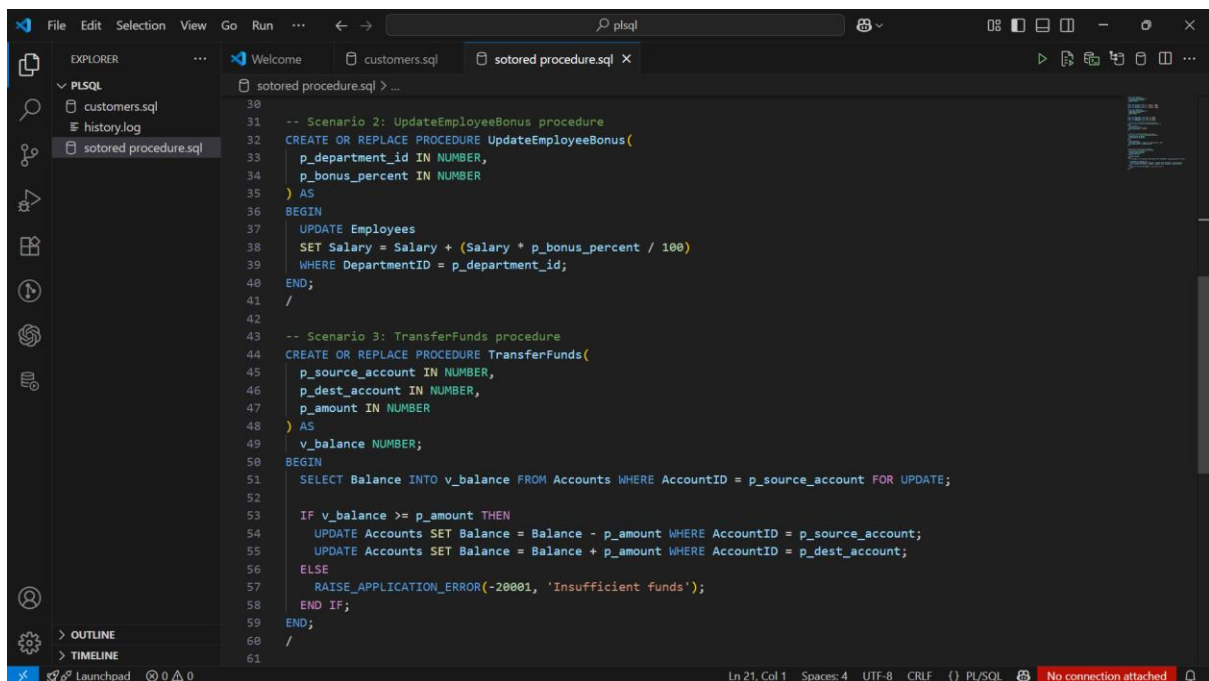
OUTPUT



STORED PROCEDURE

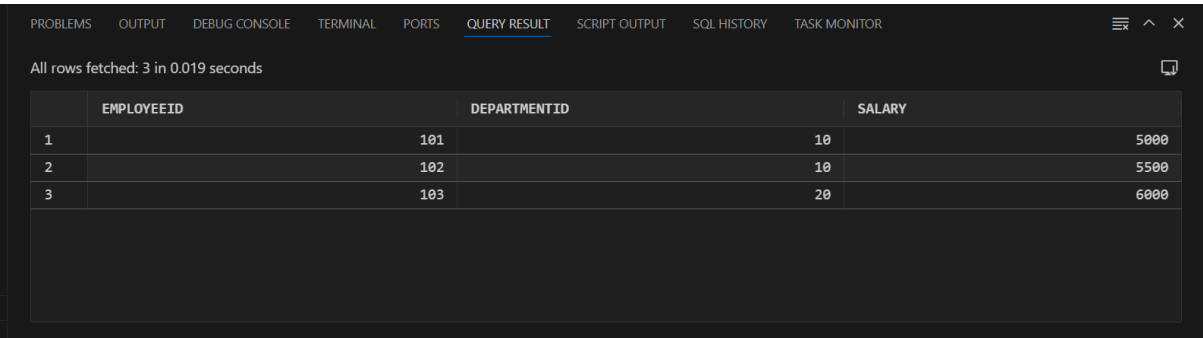
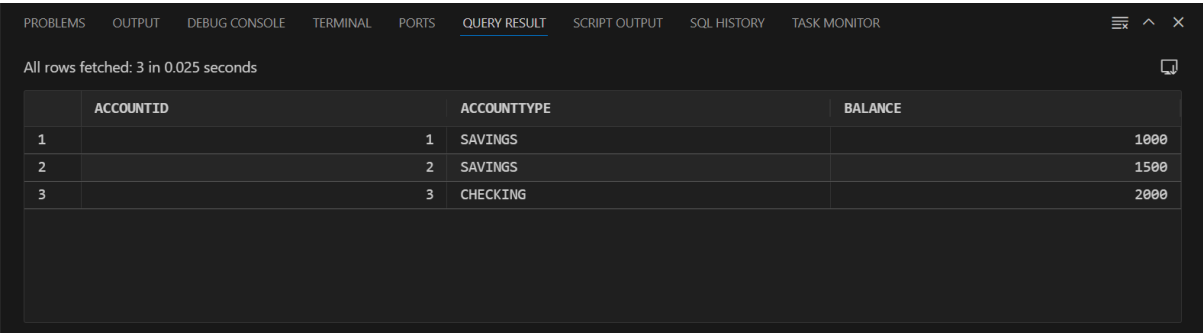
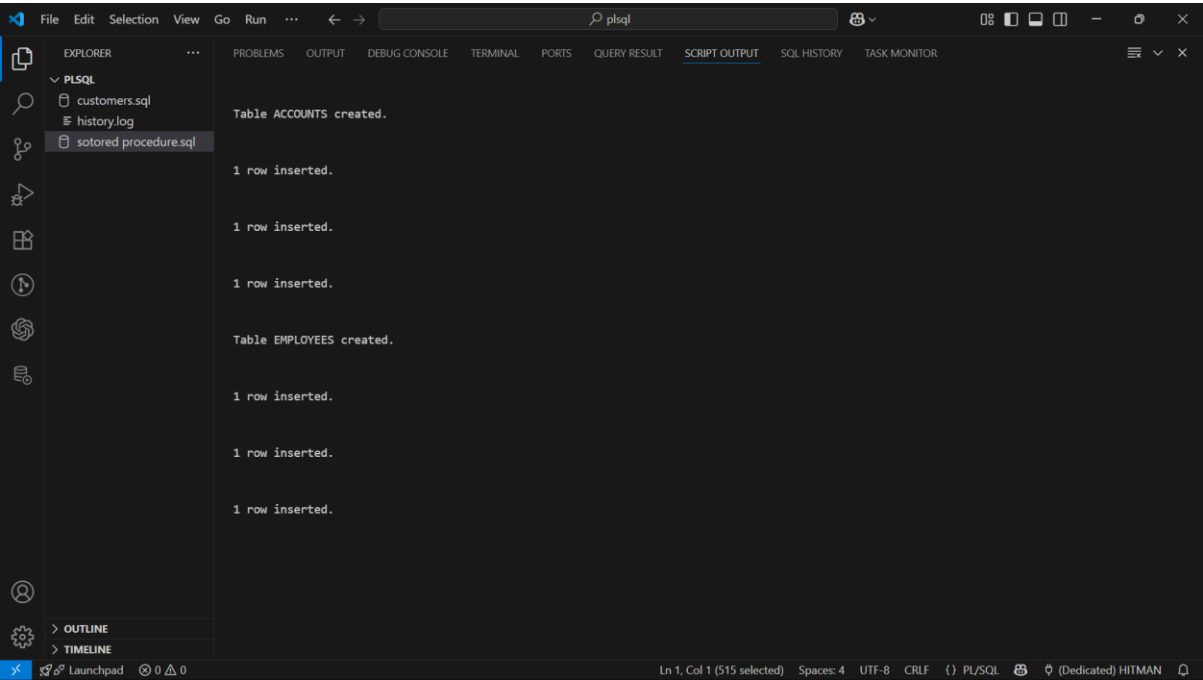


```
1 CREATE TABLE Accounts (  
2   AccountID NUMBER PRIMARY KEY,  
3   AccountType VARCHAR2(20),  
4   Balance NUMBER  
5 );  
6  
7  
8 INSERT INTO Accounts VALUES (1, 'SAVINGS', 1000);  
9 INSERT INTO Accounts VALUES (2, 'SAVINGS', 1500);  
10 INSERT INTO Accounts VALUES (3, 'CHECKING', 2000);  
11  
12 CREATE TABLE Employees (  
13   EmployeeID NUMBER PRIMARY KEY,  
14   DepartmentID NUMBER,  
15   Salary NUMBER  
16 );  
17  
18 INSERT INTO Employees VALUES (101, 10, 5000);  
19 INSERT INTO Employees VALUES (102, 10, 5500);  
20 INSERT INTO Employees VALUES (103, 20, 6000);  
21  
22 -- Scenario 1: ProcessMonthlyInterest procedure  
23 CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS  
24 BEGIN  
25   UPDATE Accounts  
26     SET Balance = Balance * 1.01  
27     WHERE AccountType = 'SAVINGS';  
28 END;  
29 /  
30  
31 -- Scenario 2: UpdateEmployeeBonus procedure  
32 CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(  
33   p_department_id IN NUMBER,  
34   p_bonus_percent IN NUMBER  
35 ) AS  
36 BEGIN  
37   UPDATE Employees  
38     SET Salary = Salary + (Salary * p_bonus_percent / 100)  
39     WHERE DepartmentID = p_department_id;  
40 END;  
41 /  
42  
43 -- Scenario 3: TransferFunds procedure  
44 CREATE OR REPLACE PROCEDURE TransferFunds(  
45   p_source_account IN NUMBER,  
46   p_dest_account IN NUMBER,  
47   p_amount IN NUMBER  
48 ) AS  
49   v_balance NUMBER;  
50 BEGIN  
51   SELECT Balance INTO v_balance FROM Accounts WHERE AccountID = p_source_account FOR UPDATE;  
52  
53   IF v_balance >= p_amount THEN  
54     UPDATE Accounts SET Balance = Balance - p_amount WHERE AccountID = p_source_account;  
55     UPDATE Accounts SET Balance = Balance + p_amount WHERE AccountID = p_dest_account;  
56   ELSE  
57     RAISE_APPLICATION_ERROR(-20001, 'Insufficient funds');  
58   END IF;  
59 END;  
60 /  
61
```



```
30  
31 -- Scenario 2: UpdateEmployeeBonus procedure  
32 CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(  
33   p_department_id IN NUMBER,  
34   p_bonus_percent IN NUMBER  
35 ) AS  
36 BEGIN  
37   UPDATE Employees  
38     SET Salary = Salary + (Salary * p_bonus_percent / 100)  
39     WHERE DepartmentID = p_department_id;  
40 END;  
41 /  
42  
43 -- Scenario 3: TransferFunds procedure  
44 CREATE OR REPLACE PROCEDURE TransferFunds(  
45   p_source_account IN NUMBER,  
46   p_dest_account IN NUMBER,  
47   p_amount IN NUMBER  
48 ) AS  
49   v_balance NUMBER;  
50 BEGIN  
51   SELECT Balance INTO v_balance FROM Accounts WHERE AccountID = p_source_account FOR UPDATE;  
52  
53   IF v_balance >= p_amount THEN  
54     UPDATE Accounts SET Balance = Balance - p_amount WHERE AccountID = p_source_account;  
55     UPDATE Accounts SET Balance = Balance + p_amount WHERE AccountID = p_dest_account;  
56   ELSE  
57     RAISE_APPLICATION_ERROR(-20001, 'Insufficient funds');  
58   END IF;  
59 END;  
60 /  
61
```

OUTPUT



61
62 SELECT * FROM Accounts;
63 EXECUTE ProcessMonthlyInterest;
64 SELECT * FROM Accounts;
65 SELECT * FROM Employees;

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR

All rows fetched: 3 in 0.026 seconds

	ACCOUNTID	ACCOUNTTYPE	BALANCE
1	1	SAVINGS	1010
2	2	SAVINGS	1515
3	3	CHECKING	2000

62 SELECT * FROM Accounts;
63 EXECUTE ProcessMonthlyInterest;
64 EXECUTE UpdateEmployeeBonus(10, 10);
65 SELECT * FROM Employees;

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR

All rows fetched: 3 in 0.023 seconds

	EMPLOYEEID	DEPARTMENTID	SALARY
1	101	10	5500
2	102	10	6050
3	103	20	6000

62 SELECT * FROM Accounts;
63 EXECUTE ProcessMonthlyInterest;
64 EXECUTE UpdateEmployeeBonus(10, 10);
65 EXECUTE TransferFunds(1, 2, 100);
66 SELECT * FROM Employees;

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR

All rows fetched: 3 in 0.023 seconds

	ACCOUNTID	ACCOUNTTYPE	BALANCE
1	1	SAVINGS	910
2	2	SAVINGS	1615
3	3	CHECKING	2000