

Outputs:

Scenario 1: Generate monthly statements for all customers.

And all scenarios script output

PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS AZURE QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR

```
--- Monthly Statement ---
Customer: John Doe | Date: 27/06/25 | Type: Deposit | Amount: 500
Customer: Jane Smith | Date: 27/06/25 | Type: Withdrawal | Amount: 300
Customer: Alice Johnson | Date: 27/06/25 | Type: Deposit | Amount: 1500
Customer: Bob Brown | Date: 27/06/25 | Type: Withdrawal | Amount: 400
Customer: Charlie White | Date: 27/06/25 | Type: Deposit | Amount: 1500
Customer: Diana Green | Date: 27/06/25 | Type: Withdrawal | Amount: 6040
Customer: Ethan Black | Date: 27/06/25 | Type: Deposit | Amount: 730
Customer: Fiona Blue | Date: 27/06/25 | Type: Withdrawal | Amount: 8050
Customer: George Yellow | Date: 27/06/25 | Type: Deposit | Amount: 900
Customer: Hannah Purple | Date: 27/06/25 | Type: Withdrawal | Amount: 1000

PL/SQL procedure successfully completed.

Annual fee applied to Account ID: 1
Annual fee applied to Account ID: 2
Annual fee applied to Account ID: 3
Annual fee applied to Account ID: 4
Annual fee applied to Account ID: 5
Annual fee applied to Account ID: 6
Annual fee applied to Account ID: 7
Annual fee applied to Account ID: 8
Annual fee applied to Account ID: 9
Annual fee applied to Account ID: 10

PL/SQL procedure successfully completed.

Loan ID: 1 updated.
Loan ID: 2 updated.
Loan ID: 3 updated.
Loan ID: 4 updated.
Loan ID: 5 updated.
Loan ID: 6 updated.
Loan ID: 7 updated.
Loan ID: 8 updated.
Loan ID: 9 updated.
Loan ID: 10 updated.
```

Scenario 2: Apply annual fee to all accounts.

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS AZURE QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR

All rows fetched: 10 in 0.025 seconds

	ACCOUNTID	CUSTOMERID	ACCOUNTTYPE	BALANCE	LASTMODIFIED
1	1	1	Savings	1010	28/06/25
2	2	2	Checking	15300	27/06/25
3	3	3	Savings	20200	28/06/25
4	4	4	Checking	25000	27/06/25
5	5	5	Savings	32320	28/06/25
6	6	6	Checking	35000	27/06/25
7	7	7	Savings	40470.7	28/06/25
8	8	8	Checking	12000	27/06/25
9	9	9	Savings	5050	28/06/25
10	10	10	Checking	5500	27/06/25

67
68
69
70
71
72

```
--  
select * from Accounts;  
/*  
Scenario 3: Update the interest rate for all loans based on a new policy.  
Question: Write a PL/SQL block using an explicit cursor UpdateLoanInterestRates that fetches all loans and updates their interest rates ba
```

PROBLEMS24

OUTPUT

DEBUG CONSOLE

TERMINAL

TEST RESULTS

PORTS

AZURE

QUERY RESULT

SCRIPT OUTPUT

SQL HISTORY

TASK MONITOR

All rows fetched: 10 in 0.022 seconds

	ACCOUNTID	CUSTOMERID	ACCOUNTTYPE	BALANCE	LASTMODIFIED
1	1	1	Savings	910	29/06/25
2	2	2	Checking	14900	29/06/25
3	3	3	Savings	20400	29/06/25
4	4	4	Checking	24900	29/06/25
5	5	5	Savings	32220	29/06/25
6	6	6	Checking	34900	29/06/25
7	7	7	Savings	40370.7	29/06/25
8	8	8	Checking	11900	29/06/25
9	9	9	Savings	4950	29/06/25
10	10	10	Checking	5400	29/06/25

Scenario 3: Update the interest rate for all loans based on a new policy.

PROBLEMS10

OUTPUT

DEBUG CONSOLE

TERMINAL

TEST RESULTS

PORTS

AZURE

QUERY RESULT

SCRIPT OUTPUT

SQL HISTORY

TASK MONITOR

All rows fetched: 10 in 0.016 seconds

	LOANID	CUSTOMERID	LOANAMOUNT	INTERESTRATE	STARTDATE	ENDDATE
1	1	1	5000	5	27/06/25	27/05/28
2	2	2	10000	4.5	27/06/25	27/04/26
3	3	3	15000	6	27/06/25	27/09/25
4	4	4	20000	5.5	27/06/25	27/02/26
5	5	5	25000	4.8	27/06/25	27/06/30
6	6	6	30000	5.2	27/06/25	07/07/25
7	7	7	35000	4.9	27/06/25	22/07/25
8	8	8	40000	5.1	27/06/25	02/07/25
9	9	9	45000	4.7	27/06/25	27/06/26
10	10	10	50000	5.3	27/06/25	27/11/25

107
108
109

```
-- Check updated interest rates  
SELECT * FROM Loans;  
--
```

PROBLEMS24

OUTPUT

DEBUG CONSOLE

TERMINAL

TEST RESULTS

PORTS

AZURE

QUERY RESULT

SCRIPT OUTPUT

SQL HISTORY

TASK MONITOR

All rows fetched: 10 in 0.022 seconds

	LOANID	CUSTOMERID	LOANAMOUNT	INTERESTRATE	STARTDATE	ENDDATE
1	1	1	5000	5.5	26/06/25	26/06/30
2	2	2	10000	4.75	26/06/25	01/07/25
3	3	3	15000	5.25	26/06/25	26/06/30
4	4	4	20000	4.75	26/06/25	26/06/30
5	5	5	25000	4.05	26/06/25	21/07/25
6	6	6	30000	4.45	26/06/25	26/06/30
7	7	7	35000	5.15	26/06/25	26/06/30
8	8	8	40000	5.35	26/06/25	21/07/25
9	9	9	45000	4.95	26/06/25	26/06/30
10	10	10	50000	5.55	26/06/25	06/07/25

