Output:

Scenario 1: Handle exceptions during fund transfers between accounts.

Before:

```
67    /*
68    select * from Accounts;
69    |
70    /*
71    Scenario 3: Update the interest rate for all loans based on a new policy.
72    Question: Write a PL/SQL block using an explicit cursor UpdateLoanInterestRates that fetches all loans and updates their interest rates ba
```

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

All rows fetched: 10 in 0.022 seconds

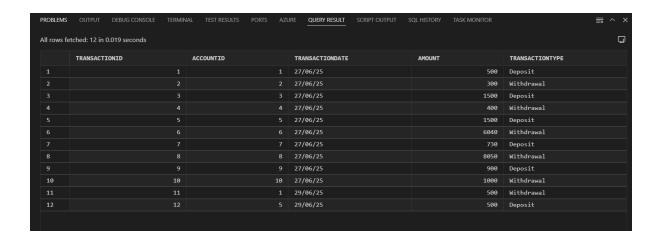|	ACCOUNTID	CUSTOMERID	ACCOUNTTYPE	BALANCE	LASTMODIFIED
1	1	1	Savings	910	29/06/25
2	2	2	Checking	14900	29/06/25
3	3	3	Savings	20400	29/06/25
4	4	4	Checking	24900	29/06/25
5	5	5	Savings	32220	29/06/25
6	6	6	Checking	34900	29/06/25
7	7	7	Savings	40370.7	29/06/25
8	8	8	Checking	11900	29/06/25
9	9	9	Savings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

PROBLEMS 10   OUTPUT   DEBUG CONSOLE   TERMINAL   TEST RESULTS   PORTS   AZURE   **QUERY RESULT**   SCRIPT OUTPUT   SQL HISTORY   TASK MONITOR

All rows fetched: 10 in 0.017 seconds

	TRANSACTIONID	ACCOUNTID	TRANSACTIONDATE	AMOUNT	TRANSACTIONTYPE
1	1	1	27/06/25	500	Deposit
2	2	2	27/06/25	300	Withdrawal
3	3	3	27/06/25	1500	Deposit
4	4	4	27/06/25	400	Withdrawal
5	5	5	27/06/25	1500	Deposit
6	6	6	27/06/25	6040	Withdrawal
7	7	7	27/06/25	730	Deposit
8	8	8	27/06/25	8050	Withdrawal
9	9	9	27/06/25	900	Deposit
10	10	10	27/06/25	1000	Withdrawal

After: SafeTransferFunds(1, 5, 500);

All rows fetched: 10 in 0.017 seconds

	ACCOUNTID	CUSTOMERID	ACCOUNTTYPE	BALANCE	LASTMODIFIED
1	1	1	Savings	410	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	32720	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

	TRANSACTIONID	ACCOUNTID	TRANSACTIONDATE	AMOUNT	TRANSACTIONTYPE
1	1	1	27/06/25	500	Deposit
2	2	2	27/06/25	300	Withdrawal
3	3	3	27/06/25	1500	Deposit
4	4	4	27/06/25	400	Withdrawal
5	5	5	27/06/25	1500	Deposit
6	6	6	27/06/25	6040	Withdrawal
7	7	7	27/06/25	730	Deposit
8	8	8	27/06/25	8050	Withdrawal
9	9	9	27/06/25	900	Deposit
10	10	10	27/06/25	1000	Withdrawal
11	11	1	29/06/25	500	Withdrawal
12	12	5	29/06/25	500	Deposit

For: SafeTransferFunds(1, 2, 1000);

```
49  BEGIN
50      SafeTransferFunds(1, 2, 1000);
51  END;
52
53  SELECT * FROM Transactions;
54  SELECT * FROM Accounts;
55
```

```
From Account New Balance: 34900


PL/SQL procedure successfully completed.

Error during fund transfer: ORA-20001: Insufficient funds in source account.


PL/SQL procedure successfully completed.
```

Scenario 2: Manage errors when updating employee salaries.

Before:

```
102  SELECT * FROM Employees;
103
104  -- Call procedure to update employee details
105  BEGIN
106      EmployeeManagement.UpdateEmployee(101, 85000, 'Senior Manager');
107  END;
108  /
```

	EMPLOYEEID	NAME	POSITION	SALARY	DEPARTMENT	HIREDATE
1	1	Amit Sharma	Manager	70000	HR	15/06/15
2	2	Priya Singh	Developer	66000	IT	20/03/17
3	3	Rahul Verma	Analyst	55000	Finance	10/01/18
4	4	Sneha Patel	Designer	50000	Marketing	25/05/19
5	5	Vikram Rao	Sales	65000	Sales	15/02/20
6	6	Anjali Nair	Support	48000	Customer Service	30/04/21
7	7	Karan Mehta	HR Assistant	40000	HR	05/07/22
8	8	Pooja Desai	Marketing Specialist	45000	Marketing	15/01/23
9	9	Suresh Reddy	IT Support	46200	IT	10/03/23
10	10	Neha Gupta	Finance Assistant	43000	Finance	20/05/23
11	101	Alice	Manager	80000	HR	29/06/25

After:

```
82     -- Example usage of the UpdateSalary procedure
83   ∨ BEGIN
84       ✦  UpdateSalary(1, 10);
85          UpdateSalary(132, 10);
86     END;
87     SELECT * FROM Employees;
88
89
90     /*Scenario 3: Ensure data integrity when adding a new customer.
```

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

```
PL/SQL procedure successfully completed.


Procedure UPDATESALARY compiled

Updated salary for employee ID 1 to 77000
Employee ID 132 not found.


PL/SQL procedure successfully completed.
```

```
82     -- Example usage of the UpdateSalary procedure
83     BEGIN
84         UpdateSalary(1, 10);
85         UpdateSalary(132, 10);
86     END;
87     SELECT * FROM Employees;
88
89
90     /*Scenario 3: Ensure data integrity when adding a new customer.
91
```

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

All rows fetched: 11 in 0.015 seconds

	EMPLOYEEID	NAME	POSITION	SALARY	DEPARTMENT	HIREDATE
1	1	Amit Sharma	Manager	77000	HR	15/06/15
2	2	Priya Singh	Developer	66000	IT	20/03/17
3	3	Rahul Verma	Analyst	55000	Finance	10/01/18
4	4	Sneha Patel	Designer	50000	Marketing	25/05/19
5	5	Vikram Rao	Sales	65000	Sales	15/02/20
6	6	Anjali Nair	Support	48000	Customer Service	30/04/21
7	7	Karan Mehta	HR Assistant	40000	HR	05/07/22
8	8	Pooja Desai	Marketing Specialist	45000	Marketing	15/01/23
9	9	Suresh Reddy	IT Support	46200	IT	10/03/23
10	10	Neha Gupta	Finance Assistant	43000	Finance	20/05/23
11	101	Alice	Senior Manager	85000	HR	29/06/25

Scenario 3: Ensure data integrity when adding a new customer.

Before:

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

All rows fetched: 12 in 0.038 seconds

	CUSTOMERID	NAME	DOB	BALANCE	LASTMODIFIED	ISVIP
1	2	Jane Smith	20/07/90	15300	19/06/25	TRUE
2	3	Alice Johnson	10/03/50	20000	27/06/25	TRUE
3	4	Bob Brown	25/11/58	25000	27/06/25	TRUE
4	5	Charlie White	30/01/60	32000	27/06/25	TRUE
5	6	Diana Green	05/09/63	35000	27/06/25	TRUE
6	7	Ethan Black	15/12/92	40070	19/06/25	TRUE
7	8	Fiona Blue	22/04/80	12000	19/06/25	TRUE
8	9	George Yellow	18/08/87	5000	19/06/25	FALSE
9	10	Hannah Purple	30/06/93	5500	19/06/25	FALSE
10	1	John Doe	15/05/85	1000	19/06/25	FALSE
11	11	Ravi Kumar	15/05/80	2000	28/06/25	(null)
12	12	smith	01/01/90	5000	29/06/25	(null)

After:

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


Procedure ADDNEWCUSTOMER compiled

Customer with ID 11 already exists.
Customer with ID 1 already exists.


PL/SQL procedure successfully completed.


Procedure ADDNEWCUSTOMER compiled

New customer added with ID: 13
Customer with ID 1 already exists.


PL/SQL procedure successfully completed.
```

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

All rows fetched: 13 in 0.083 seconds

	CUSTOMERID	NAME	DOB	BALANCE	LASTMODIFIED	ISVIP
1	2	Jane Smith	20/07/90	15300	19/06/25	TRUE
2	3	Alice Johnson	10/03/50	20000	27/06/25	TRUE
3	4	Bob Brown	25/11/58	25000	27/06/25	TRUE
4	5	Charlie White	30/01/60	32000	27/06/25	TRUE
5	6	Diana Green	05/09/63	35000	27/06/25	TRUE
6	7	Ethan Black	15/12/92	40070	19/06/25	TRUE
7	8	Fiona Blue	22/04/80	12000	19/06/25	TRUE
8	9	George Yellow	18/08/87	5000	19/06/25	FALSE
9	10	Hannah Purple	30/06/93	5500	19/06/25	FALSE
10	1	John Doe	15/05/85	1000	19/06/25	FALSE
11	11	Ravi Kumar	15/05/80	2000	28/06/25	(null)
12	12	smith	01/01/90	5000	29/06/25	(null)
13	13	bharadwaj	21/05/86	12000	29/06/25	(null)