Exercise 2: Error Handling

Scenario 1: Handle exceptions during fund transfers between accounts.

oQuestion: Write a stored procedure SafeTransferFunds that transfers funds between two accounts. Ensure that if any error occurs (e.g., insufficient funds), an appropriate error message is logged and the transaction is rolled back.

CREATE OR REPLACE PROCEDURE SafeTransferFunds(acc1 NUMBER, acc2 NUMBER, amt NUMBER)

IS

amt\_insufficient EXCEPTION;

acc\_not\_found EXCEPTION;

bal NUMBER;

BEGIN

SELECT balance INTO bal

FROM accounts

WHERE accountid = acc1;

IF bal < amt THEN

RAISE amt\_insufficient;

ELSE

UPDATE accounts SET balance = balance + amt

WHERE accountid = acc2;

IF SQL%NOTFOUND THEN

RAISE acc\_not\_found;

END IF;

UPDATE accounts SET balance = balance - amt

WHERE accountid = acc1;

END IF;

EXCEPTION

WHEN acc\_not\_found THEN

DBMS\_OUTPUT.PUT\_LINE('Account no. '|| acc2 || ' not found.');

WHEN amt\_insufficient THEN

DBMS\_OUTPUT.PUT\_LINE('Account does not have enough balance.');

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Account no. '|| acc1 || ' not found.');

END;

/

Scenario 2: Manage errors when updating employee salaries.

oQuestion: Write a stored procedure UpdateSalary that increases the salary of an employee by a given percentage. If the employee ID does not exist, handle the exception and log an error message.

CREATE OR REPLACE PROCEDURE UpdateSalary(emp\_no NUMBER, percent NUMBER)

IS

emp\_not\_found EXCEPTION;

BEGIN

UPDATE employees SET salary = salary \* (1+(percent/100))

WHERE employeeid = emp\_no;

IF SQL%NOTFOUND THEN

RAISE emp\_not\_found;

END IF;

EXCEPTION

WHEN emp\_not\_found THEN

DBMS\_OUTPUT.PUT\_LINE('EmployeeID '|| emp\_no || ' not found.');

END;

/

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

oQuestion: Write a stored procedure AddNewCustomer that inserts a new customer into the Customers table. If a customer with the same ID already exists, handle the exception by logging an error and preventing the insertion.

CREATE OR REPLACE PROCEDURE AddNewCustomer(cust\_id NUMBER, c\_name VARCHAR2,c\_dob DATE, c\_bal NUMBER, c\_lm DATE)

IS

cust\_exists EXCEPTION;

PRAGMA EXCEPTION\_INIT(cust\_exists, -00001);

BEGIN

INSERT INTO customers VALUES(cust\_id, c\_name,c\_dob, c\_bal, c\_lm);

EXCEPTION

WHEN cust\_exists THEN

DBMS\_OUTPUT.PUT\_LINE('CustomerID '|| cust\_id || ' already exists in table.');

END;

/