PL/SQL Hands-on Assignment: Dynamic SQL, Error Handling, and Bulk Processing

# 1. Table Creation

-- Sales tables  
CREATE TABLE SALES\_2022 (  
 id NUMBER PRIMARY KEY,  
 sale\_date DATE,  
 amount NUMBER  
);  
  
CREATE TABLE SALES\_2023 (  
 id NUMBER PRIMARY KEY,  
 sale\_date DATE,  
 amount NUMBER  
);  
  
-- Archive table  
CREATE TABLE ARCHIVE\_SALES (  
 id NUMBER PRIMARY KEY,  
 sale\_date DATE,  
 amount NUMBER  
);  
  
-- Error log table  
CREATE TABLE ERROR\_LOG (  
 error\_time TIMESTAMP DEFAULT SYSTIMESTAMP,  
 error\_msg VARCHAR2(4000),  
 procedure\_name VARCHAR2(100)  
);

# 2. Insert Dummy Data

BEGIN  
 FOR i IN 1..50 LOOP  
 INSERT INTO SALES\_2022 VALUES (i, DATE '2022-05-01' + MOD(i, 365), ROUND(DBMS\_RANDOM.VALUE(100, 1000), 2));  
 INSERT INTO SALES\_2023 VALUES (i, DATE '2023-03-01' + MOD(i, 365), ROUND(DBMS\_RANDOM.VALUE(100, 1000), 2));  
 END LOOP;  
 COMMIT;  
END;  
/

# 3. Archive Procedure

CREATE OR REPLACE PROCEDURE archive\_old\_sales IS  
 TYPE sale\_rec\_type IS TABLE OF ARCHIVE\_SALES%ROWTYPE;  
 v\_sales\_tables SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('SALES\_2022', 'SALES\_2023');  
 v\_sql VARCHAR2(1000);  
 v\_records sale\_rec\_type;  
 v\_year VARCHAR2(4);  
 v\_table VARCHAR2(30);  
BEGIN  
 FOR i IN 1 .. v\_sales\_tables.COUNT LOOP  
 v\_table := v\_sales\_tables(i);  
 BEGIN  
 -- Check table existence  
 SELECT 1 INTO v\_year  
 FROM all\_tables  
 WHERE table\_name = UPPER(v\_table)  
 AND owner = USER;  
  
 -- Dynamic SQL to fetch old records  
 v\_sql := 'SELECT id, sale\_date, amount FROM ' || v\_table ||  
 ' WHERE sale\_date < ADD\_MONTHS(SYSDATE, -12)';  
  
 EXECUTE IMMEDIATE v\_sql BULK COLLECT INTO v\_records;  
  
 -- Insert into ARCHIVE\_SALES  
 FORALL j IN 1 .. v\_records.COUNT  
 INSERT INTO ARCHIVE\_SALES VALUES v\_records(j);  
  
 -- Delete from source table  
 v\_sql := 'DELETE FROM ' || v\_table ||   
 ' WHERE sale\_date < ADD\_MONTHS(SYSDATE, -12)';  
 EXECUTE IMMEDIATE v\_sql;  
  
 COMMIT;  
 EXCEPTION  
 WHEN OTHERS THEN  
 INSERT INTO ERROR\_LOG(error\_msg, procedure\_name)  
 VALUES(SQLERRM, 'archive\_old\_sales');  
 COMMIT;  
 END;  
 END LOOP;  
END;  
/

# 4. Output & Testing

## Output: ARCHIVE\_SALES

Sample rows archived:

ID | SALE\_DATE | AMOUNT

1 | 2022-05-02 | 723.45

2 | 2022-05-03 | 581.32

... | ... | ...

50 | 2023-03-15 | 856.77

## Output: SALES\_2022

0 rows (all data archived)

## Output: SALES\_2023

Only recent rows (within the last 12 months) remain

## Output: ERROR\_LOG

ERROR\_TIME | ERROR\_MSG | PROCEDURE\_NAME

04-JUL-25 05:22:31.000 PM | ORA-00942: table or view does not exist | archive\_old\_sales

\*Note: Only appears if error occurs\*