Practical No. 10

Aim: Create a simple PL/SQL program which includes declaration section executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found) ii. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

Theory:

PL/SQL(Procedural Language/Structured Query Language) is a programming language that extends SQL and provides procedural capabilities. It is used for developing stored procedures, functions, triggers, and other database objects.

Queries:

1) Write a PL/SQL block to swap two numbers.

```
  2    num1 NUMBER := 999;
  3    num2 NUMBER := 1999;
  4    temp NUMBER;
  5  BEGIN
  6    temp := num1;
  7    num1 := num2;
  8    num2 := temp;
  9
 10    DBMS_OUTPUT.PUT_LINE('After swapping:');
 11    DBMS_OUTPUT.PUT_LINE('Number 1: ' || num1);
 12    DBMS_OUTPUT.PUT_LINE('Number 2: ' || num2);
 13  END;
 14  /
After swapping:
Number 1: 1999
Number 2: 999

PL/SQL procedure successfully completed.

SQL> /* 202203103510097 */
SQL>
```

2) Write a PL/SQL block to take personal details (Enrolment Number, Name, Phone Number and Qualification) from user and display it.

```
Windows PowerShell
SQL> DECLARE
  2    enrolment_number NUMBER;
  3    name VARCHAR2(50);
  4    phone_number VARCHAR2(15);
  5    qualification VARCHAR2(50);
  6  BEGIN
  7    enrolment_number := &enrolment_number;
  8    name := '&name';
  9    phone_number := '&phone_number';
 10    qualification := '&qualification';
 11
 12    DBMS_OUTPUT.PUT_LINE('Enrolment Number: ' || enrolment_number);
 13    DBMS_OUTPUT.PUT_LINE('Name: ' || name);
 14    DBMS_OUTPUT.PUT_LINE('Phone Number: ' || phone_number);
 15    DBMS_OUTPUT.PUT_LINE('Qualification: ' || qualification);
 16  END;
 17  /
Enter value for enrolment_number: 202203103510097
old   7:   enrolment_number := &enrolment_number;
new   7:   enrolment_number := 202203103510097;
Enter value for name: Angat
old   8:   name := '&name';
new   8:   name := 'Angat';
Enter value for phone_number: 9620729683
old   9:   phone_number := '&phone_number';
new   9:   phone_number := '9620729683';
Enter value for qualification: B.Tech CSE
old  10:   qualification := '&qualification';
new  10:   qualification := 'B.Tech CSE';
Enrolment Number: 202203103510097
```

```
Windows PowerShell
 10    qualification := '&qualification';
 11
 12    DBMS_OUTPUT.PUT_LINE('Enrolment Number: ' || enrolment_number);
 13    DBMS_OUTPUT.PUT_LINE('Name: ' || name);
 14    DBMS_OUTPUT.PUT_LINE('Phone Number: ' || phone_number);
 15    DBMS_OUTPUT.PUT_LINE('Qualification: ' || qualification);
 16  END;
 17  /
Enter value for enrolment_number: 202203103510097
old   7:   enrolment_number := &enrolment_number;
new   7:   enrolment_number := 202203103510097;
Enter value for name: Angat
old   8:   name := '&name';
new   8:   name := 'Angat';
Enter value for phone_number: 9620729683
old   9:   phone_number := '&phone_number';
new   9:   phone_number := '9620729683';
Enter value for qualification: B.Tech CSE
old  10:   qualification := '&qualification';
new  10:   qualification := 'B.Tech CSE';
Enrolment Number: 202203103510097
Name: Angat
Phone Number: 9620729683
Qualification: B.Tech CSE

PL/SQL procedure successfully completed.

SQL> /* 202203103510097 */
SQL>
```

3) Write a PL/SQL block to find given year is leap year or not.

```
Windows PowerShell
SQL> DECLARE
  2    year NUMBER := 2028;
  3  BEGIN
  4    IF (MOD(year, 4) = 0 AND MOD(year, 100) != 0) OR MOD(year, 400) = 0 THEN
  5      DBMS_OUTPUT.PUT_LINE(year || ' is a leap year.');
  6    ELSE
  7      DBMS_OUTPUT.PUT_LINE(year || ' is not a leap year.');
  8    END IF;
  9  END;
 10  /
2028 is a leap year.

PL/SQL procedure successfully completed.

SQL> /* 202203103510097 */
SQL>
```

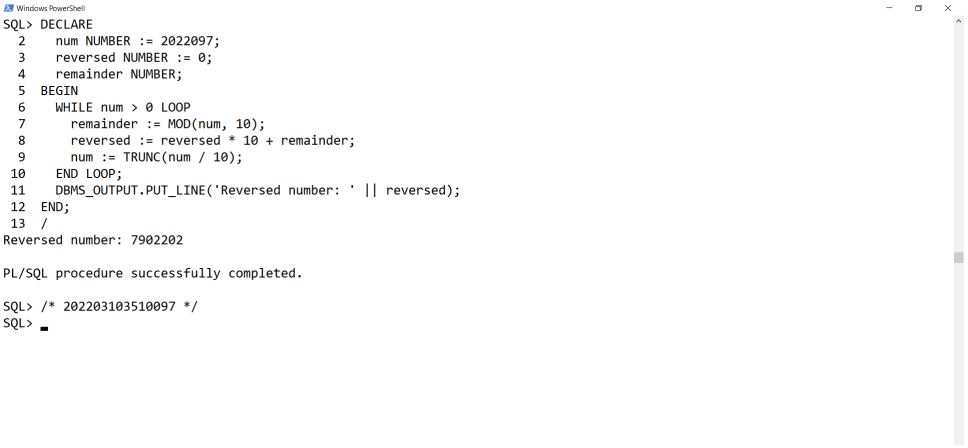
4) Write a PL/SQL block to find factorial of a given number using for loop.

```
Windows PowerShell
SQL> DECLARE
  2    num NUMBER := 7;
  3    result NUMBER := 1;
  4  BEGIN
  5    FOR i IN 1..num LOOP
  6      result := result * i;
  7    END LOOP;
  8    DBMS_OUTPUT.PUT_LINE('Factorial of ' || num || ' is ' || result);
  9  END;
 10  /
Factorial of 7 is 5040

PL/SQL procedure successfully completed.

SQL> /* 202203103510097 */
SQL>
```

5) Write a PL/SQL block to reverse a given number using while loop.

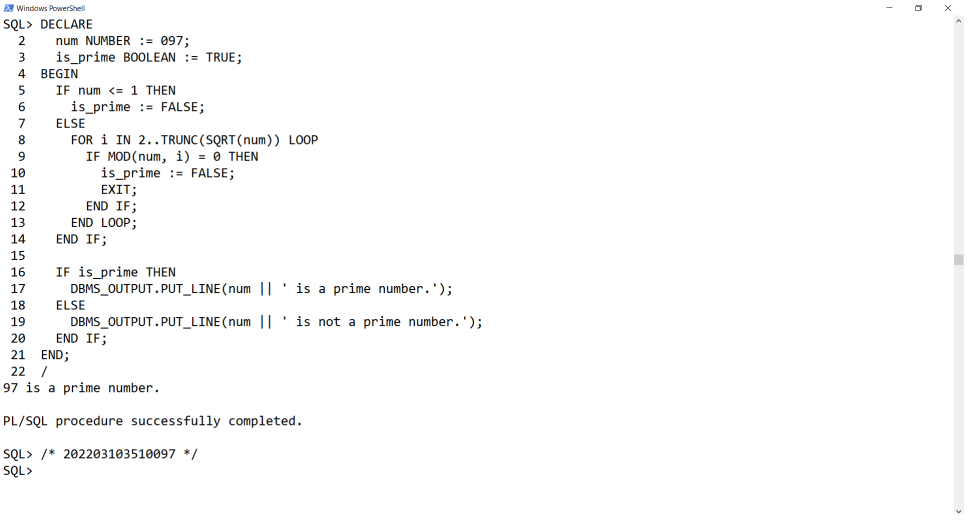


```
Windows PowerShell
SQL> DECLARE
  2    num NUMBER := 2022097;
  3    reversed NUMBER := 0;
  4    remainder NUMBER;
  5  BEGIN
  6    WHILE num > 0 LOOP
  7      remainder := MOD(num, 10);
  8      reversed := reversed * 10 + remainder;
  9      num := TRUNC(num / 10);
 10    END LOOP;
 11    DBMS_OUTPUT.PUT_LINE('Reversed number: ' || reversed);
 12  END;
 13  /
Reversed number: 7902202

PL/SQL procedure successfully completed.

SQL> /* 202203103510097 */
SQL>
```

6) Write a PL/SQL block to check whether given number is prime or not.



```
Windows PowerShell
SQL> DECLARE
  2    num NUMBER := 097;
  3    is_prime BOOLEAN := TRUE;
  4  BEGIN
  5    IF num <= 1 THEN
  6      is_prime := FALSE;
  7    ELSE
  8      FOR i IN 2..TRUNC(SQRT(num)) LOOP
  9        IF MOD(num, i) = 0 THEN
 10          is_prime := FALSE;
 11          EXIT;
 12        END IF;
 13      END LOOP;
 14    END IF;
 15
 16    IF is_prime THEN
 17      DBMS_OUTPUT.PUT_LINE(num || ' is a prime number.');
 18    ELSE
 19      DBMS_OUTPUT.PUT_LINE(num || ' is not a prime number.');
 20    END IF;
 21  END;
 22  /
97 is a prime number.

PL/SQL procedure successfully completed.

SQL> /* 202203103510097 */
SQL>
```

Conclusion: PL/SQL program demonstrates how to declare variables, retrieve and process data from a database table, and handle exceptions. It can be expanded upon to include more complex business logic and transaction management as needed for specific database operations.