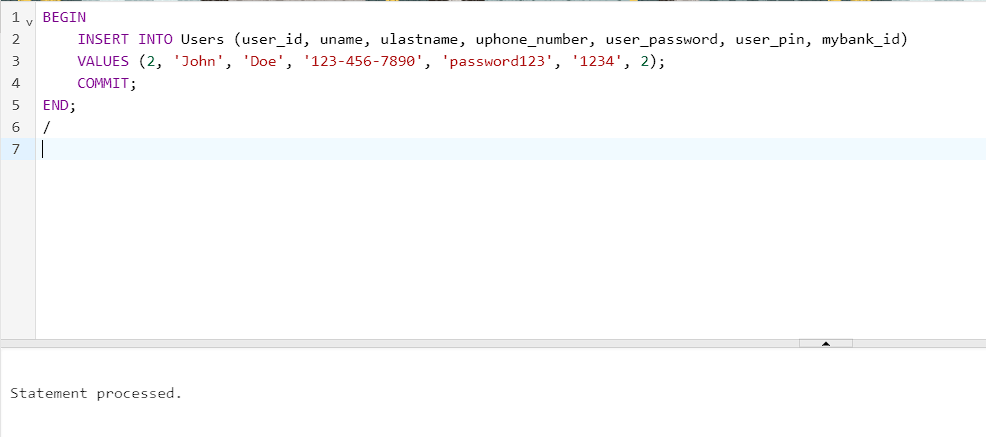
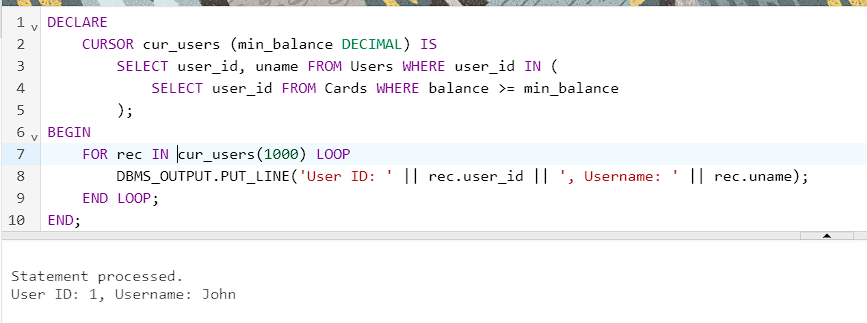
**Lab2. Creating anonymous blocks**

1. Create 6 anonymous blocks for your database (Lab 1) with the functionality according to the database domain.
2. At least three blocks must include explicit cursors.
   1. For the cursors, use the traditional way of processing them two times and the *Cursor For Loop* structure – once.
   2. One of the cursors that you create needs to be with parameters.
   3. Cursors must be used correctly, according to their functionality!!!
3. Additionally, the anonymous blocks must include:
   1. creating and application of at least one record;
   2. creating and application of at least one INDEX BY table;
   3. application of the control structures, such as IF, loops, CASE.





DECLARE

CURSOR cur\_users (min\_balance DECIMAL) IS

SELECT user\_id, uname FROM Users WHERE user\_id IN (

SELECT user\_id FROM Cards WHERE balance >= min\_balance

);

BEGIN

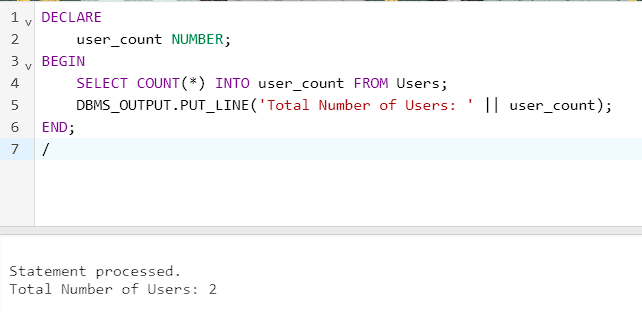
FOR rec IN cur\_users(1000) LOOP

DBMS\_OUTPUT.PUT\_LINE('User ID: ' || rec.user\_id || ', Username: ' || rec.uname);

END LOOP;

END;

/



DECLARE

user\_count NUMBER;

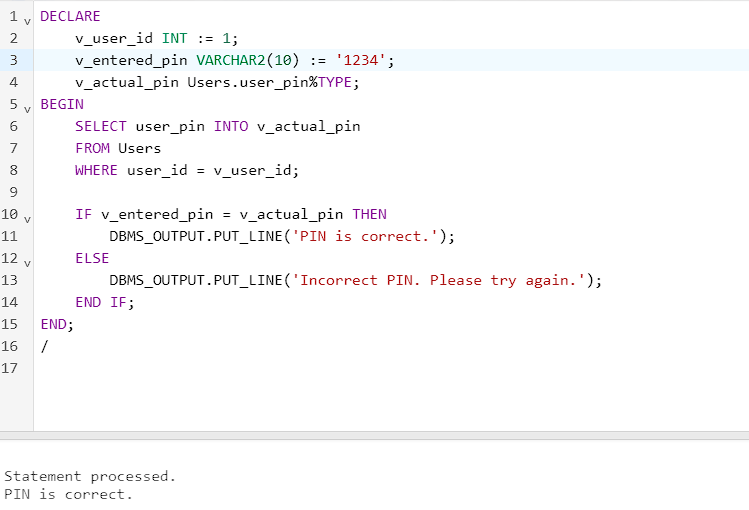
BEGIN

SELECT COUNT(\*) INTO user\_count FROM Users;

DBMS\_OUTPUT.PUT\_LINE('Total Number of Users: ' || user\_count);

END;

/



DECLARE

v\_user\_id INT := 1;

v\_entered\_pin VARCHAR2(10) := '1234';

v\_actual\_pin Users.user\_pin%TYPE;

BEGIN

SELECT user\_pin INTO v\_actual\_pin

FROM Users

WHERE user\_id = v\_user\_id;

IF v\_entered\_pin = v\_actual\_pin THEN

DBMS\_OUTPUT.PUT\_LINE('PIN is correct.');

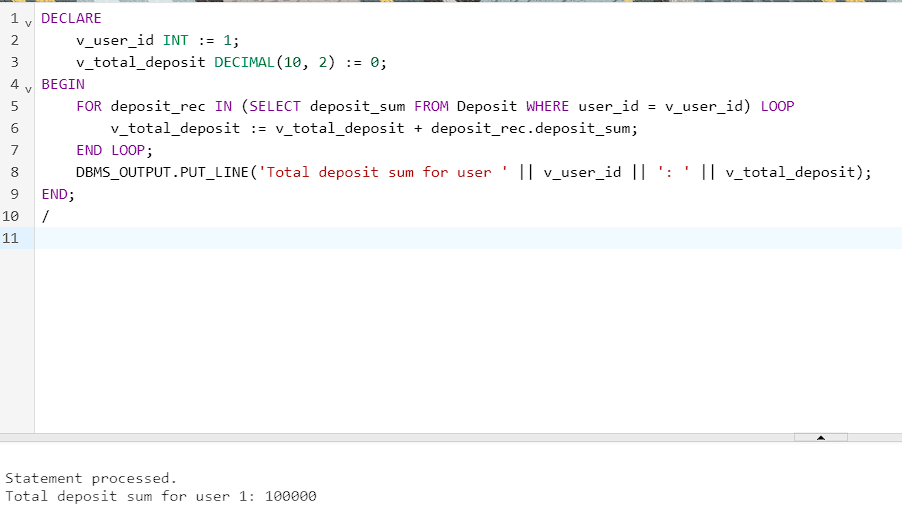
ELSE

DBMS\_OUTPUT.PUT\_LINE('Incorrect PIN. Please try again.');

END IF;

END;

/



DECLARE

v\_user\_id INT := 1;

v\_total\_deposit DECIMAL(10, 2) := 0;

BEGIN

FOR deposit\_rec IN (SELECT deposit\_sum FROM Deposit WHERE user\_id = v\_user\_id) LOOP

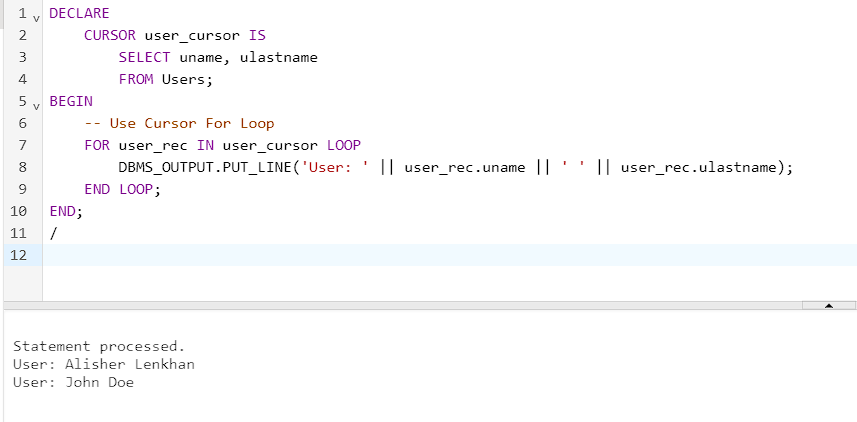
v\_total\_deposit := v\_total\_deposit + deposit\_rec.deposit\_sum;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Total deposit sum for user ' || v\_user\_id || ': ' || v\_total\_deposit);

END;

/



DECLARE

CURSOR user\_cursor IS

SELECT uname, ulastname

FROM Users;

BEGIN

-- Use Cursor For Loop

FOR user\_rec IN user\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('User: ' || user\_rec.uname || ' ' || user\_rec.ulastname);

END LOOP;

END;

/

Questions:

1. **Describe the structure of an anonymous block**

Anonymous block in PL/SQL consists of three sections: Declaration, Execution, Exception Handling.

1. **What types of blocks exist in PL/SQL?**

Anonymous Blocks: These are unnamed and standalone PL/SQL blocks that can be executed directly.

Named Blocks (Procedures and Functions): These are named and reusable blocks of code that can be called from other parts of the program.

1. **The rules to declare variables**

Variable names must start with a letter and can contain letters, numbers, and underscores.

Variable names are case insensitive.

Variables must have a data type specified.

Variables can be declared in the Declaration section of a block

1. **What types of composite data types do you know?**

Records: Used to hold related data items of different data types.

Nested Tables: Variable-size arrays.

VARRAYs (Variable Size Arrays): Fixed-size arrays.

1. **What is the main difference between records and INDEX BY tables?**

Records: Contain a fixed number of fields of different data types.

INDEX BY Tables (Associative Arrays): Are collections where data is stored as key-value pairs, and the size is dynamic.

1. **Show how to create records (2 ways)**

DECLARE

emp\_rec RECORD\_TYPE;

BEGIN

emp\_rec.field1 := 'Value1';

emp\_rec.field2 := 123;

END;

DECLARE

emp\_rec RECORD\_TYPE := RECORD\_TYPE('Value1', 123);

BEGIN

emp\_rec.field1

emp\_rec.field2

END;

1. **Show how to create INDEX BY table**

DECLARE

TYPE index\_by\_table IS TABLE OF VARCHAR2(50) INDEX BY PLS\_INTEGER;

my\_table index\_by\_table;

BEGIN

my\_table(1) := 'Value1';

my\_table(2) := 'Value2';

END;

1. **The use of %TYPE attribute**

emp\_name employees.first\_name%TYPE;

1. **The use of %ROWTYPE attribute**

emp\_record employees%ROWTYPE;

1. **The types of loops**
2. **The types of CASE**
3. **Implicit cursors and their attributes**

%FOUND, %NOTFOUND, %ROWCOUNT, %ISOPEN.

1. **Explicit cursors**

Explicit cursors are user-defined and must be declared, opened, fetched, and closed explicitly.

1. **The attributes of explicit cursors**

%FOUND, %NOTFOUND, %ROWCOUNT, %ISOPEN, and others

1. **What happens when we declare an explicit cursor?**

When declaring an explicit cursor, you specify the SQL query that the cursor will execute. The cursor is not yet associated with any result set.

1. **What happens when we open the explicit cursor?**

Opening an explicit cursor associates it with a specific result set based on the SQL query. It prepares the cursor for fetching rows.

1. **What happens when we fetch the explicit cursor?**

Fetching retrieves rows one at a time from the result set associated with the cursor. The fetched data can be processed or displayed.

1. **What happens when we close the explicit cursor?**

Closing a cursor releases resources and frees it from the associated result set. After closing, no more fetch operations can be performed on the cursor.