**Basic PL/SQL**

**NOTE:** We are going to use **HR database**. You need to start SQL Developer for this.

***PL/SQL Basics***

Today we are going to do some simple exercises involving PL/SQL. It is a programming language for Oracle Database that allows you to do much more complex things than SQL itself.

A PL/SQL blocks looks like following.

[<Block header>]

[**declare**

<Constants>

<Variables>

<Cursors>

<User defined exceptions>]

**begin**

<PL/SQL statements>

[**exception**

<Exception handling>]

**end;**

The block header specifies whether the PL/SQL block is a procedure, a function, or a package. If no header is specified, the block is said to be an anonymous PL/SQL block. The scope of declared variables (i.e., the part of the program in which one can refer to the variable) is analogous to the scope of variables in programming languages such as C or Pascal.

To do this tutorial, your online module will be helpful.

We are going to use HR database.

**Exercise 1: Syntax Checks**

Which of the following Pl/SQL blocks execute successfully?Why?

1. BEGIN

END;

1. DECLARE

V\_amountINTEGER(10);

END;

1. DECLARE

BEGIN

END;

1. DECLARE

V\_AMOUNT INTEGER(10);

BEGIN

DBMS\_OUTPUT.PUT\_LINE(V\_AMOUNT);

END;

**Exercise 2: Print**

Create and execute a simple anonymous block that outputs “Hello World”.

SET SERVEROUTPUT ON

BEGIN

DBMS\_OUTPUT.PUT\_LINE(‘Hello World’);

END;

What it outputs? Write another PL/SQL block which prints “Hello *YOURNAME*”.

**Exercise 3: Variable Declaration**

What is the output of following:

Set Serveroutput on

Declare

Fname VARCHAR (20);

LnameVARCHAR(15) DEFAULT ‘fernandez’;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(FNAME|| ‘ ‘ || lname);

End;

DECLARE

today DATE:=SYSDATE;

tomorrowtoday%TYPE;

BEGIN

Tomorrow:=today+1;

DBMS\_OUTPUT.PUT\_LINE(‘Hello World’);

DBMS\_OUTPUT.PUT\_LINE(‘today is:’ || today);

DBMS\_OUTPUT.PUT\_LINE(‘tomorrow is’ || tomorrow);

End ;

**Exercise 4: Variable Names**

Identify valid and invalid name of variables( aka. Identifier) in following:

* 1. Today
  2. Last\_name
  3. Today’s\_date
  4. Number\_of\_days
  5. #number
  6. Number#
  7. Number1to7

**Exercise 5: Variable Declarations**

Identify valid and invalid variable declaration and initialization

1. Number\_compies PLS\_INTEGER;
2. PRINTER\_NAME constant VARCHAR2(10);
3. Deliver\_to VARCHAR2(10):=Johnson;
4. By\_when DATE:CURRENT\_DATE+1;

**Exercise 6: Input at Runtime**

Try to run following:

DECLARE

my\_var VARCHAR2(30);

BEGIN

my\_var := '&input';

dbms\_output.put\_line('Hello'|| my\_var );

END;

/

What do you observe?What is reason for this output?

Modify above code to take input at the run time your name and your student number, and print out “Hello, *YOURNAME, YOURID”*

**DECLARE**

**my\_name VARCHAR2(30);**

**my\_id VARCHAR2(30);**

**BEGIN**

**my\_name := '&input';**

**my\_id := '&input1';**

**dbms\_output.put\_line('Hello'|| my\_name ||my\_id );**

**END;**

**/**

**Exercise 7: IF-Then Else Statements**

Modify the code given in previous exercise, so that it will only print when input is other than ‘No’.

**DECLARE**

**my\_var VARCHAR2(30);**

**BEGIN**

**my\_var := '&input';**

**if(my\_var='No') then**

**dbms\_output.put\_line('Hello Cannot print the name' );**

**else**

**dbms\_output.put\_line('Hello'|| my\_var );**

**end if;**

**END;**

**/**

**Exercise 8: Reading SQL data**

Within PL/SQL block we can also write our regular SQL statement and read the output in a variable. Execute following PL/SQL block

DECLARE

F\_name varchar(20);

BEGIN

SELECT first\_name INTO f\_name from employees where employee\_id=100;

End;

/

Modify above PL/SQL block to print out the first name of the employee whose id is 100;

**Set serveroutput on**

**DECLARE**

**F\_name varchar(20);**

**BEGIN**

**SELECT first\_name INTO f\_name from employees where employee\_id=100;**

**dbms\_output.put\_line(f\_name);**

**End;**

Modify above PL/SQL block to print out both first name and last name of the employee whose id is 100;

**setserveroutput on**

**DECLARE**

**F\_name varchar(20);**

**l\_name varchar(20);**

**BEGIN**

**SELECT first\_name, last\_name INTO f\_name, l\_name from employees where employee\_id=100;**

**dbms\_output.put\_line(f\_name||’ ‘||l\_name );**

**End;**

**HINT: declare two variables for first name and last name. then modify select statement to also retrieve last name of the employee.**