**Oracle Lab 6: PL/SQL (1)**

***Assoc. Prof. Dr. Nguyen Thai Nghe***

Refer to this link <http://plsql-tutorial.com/> then make a report (***StudentID-lab6***.docx) including examples (***StudentID-lab6***.sql).

1. **Write a report about the basics in PL/SQL as the following:**
2. Introduction to PL/SQL

**What is PL/SQL?**

- PL/SQL stands for Procedural Language extension of SQL.

- PL/SQL is a combination of SQL along with the procedural features of programming languages.

- It was developed by Oracle Corporation in the early 90’s to enhance the capabilities of SQL.

**The PL/SQL Engine:**

Oracle uses a PL/SQL engine to processes the PL/SQL statements. A PL/SQL language code can be stored in the client system (client-side) or in the database (server-side).

1. Anonymous Block

**PL/SQL Anonymous Block:**

- PL/SQL is a block-structured language whose code is organized into blocks.   
- A PL/SQL block consists of three sections: declaration, executable, and exception-handling sections. In a block, the executable section is mandatory while the declaration and exception-handling sections are optional.

- A PL/SQL block has a name. Functions or Procedures is an example of a named block. A named block is stored into the Oracle Database server and can be reused later.

- A block without a name is an anonymous block. An anonymous block is not saved in the Oracle Database server, so it is just for one-time use. However, PL/SQL anonymous blocks can be useful for testing purposes.

1. [PL/SQL Variables](http://plsql-tutorial.com/plsql-variables.htm)

**PL/SQL Placeholders**

Placeholders are temporary storage area. PL/SQL Placeholders can be any of Variables, Constants and Records. Oracle defines placeholders to store data temporarily, which are used to manipulate data during the execution of a PL SQL block.

**Define PL/SQL Placeholders**

Depending on the kind of data you want to store, you can define placeholders with a name and a datatype. Few of the datatypes used to define placeholders are as given below. Number (n,m) , Char (n) , Varchar2 (n) , Date , Long , Long raw, Raw, Blob, Clob, Nclob, Bfile

**PL/SQL Variables**

These are placeholders that store the values that can change through the PL/SQL Block.

**General Syntax to declare a variable is**

variable\_name datatype [NOT NULL := value ];

* variable\_name is the name of the variable.
* datatype is a valid PL/SQL datatype.
* NOT NULL is an optional specification on the variable.
* value or DEFAULT value is also an optional specification, where you can initialize a variable.
* Each variable declaration is a separate statement and must be terminated by a semicolon.

1. [PL/SQL Constants](http://plsql-tutorial.com/plsql-constants.htm)

**PL/SQL Constants**

As the name implies a constant is a value used in a PL/SQL Block that remains unchanged throughout the program. A constant is a user-defined literal value. You can declare a constant and use it instead of actual value.

For example: If you want to write a program which will increase the salary of the employees by 25%, you can declare a constant and use it throughout the program. Next time when you want to increase the salary again you can change the value of the constant which will be easier than changing the actual value throughout the program.

**General Syntax to declare a constant is:**

constant\_name CONSTANT datatype := VALUE;

* constant\_name is the name of the constant i.e. similar to a variable name.
* The word CONSTANT is a reserved word and ensures that the value does not change.
* VALUE - It is a value which must be assigned to a constant when it is declared. You cannot assign a value later.

1. [PL/SQL Conditional Statements](http://plsql-tutorial.com/plsql-conditional-statements.htm)

**Conditional Statements in PL/SQL**

As the name implies, PL/SQL supports programming language features like conditional statements, iterative statements.

The programming constructs are similar to how you use them in programming languages like Java and C++.

**Examples:**

1/ IF condition1

THEN

statement1;

statement2;

ELSIF condtion2 THEN

statement3;

ELSE

statement4;

END IF;

2/ IF condition1 THEN

ELSE

IF condition2 THEN

statement1;

END IF;

ELSIF condition3 THEN

statement2;

END IF;

1. [PL/SQL Iterative Statements](http://plsql-tutorial.com/plsql-iterative-statements.htm)

**Iterative Statements in PL/SQL**

Iterative control Statements are used when we want to repeat the execution of one or more statements for a specified number of times.

**There are three types of loops in PL/SQL:**

**1) Simple Loop**

A Simple Loop is used when a set of statements is to be executed at least once before the loop terminates. An EXIT condition must be specified in the loop, otherwise the loop will get into an infinite number of iterations. When the EXIT condition is satisfied the process exits from the loop.

**General Syntax:**

LOOP

statements;

EXIT;

{or EXIT WHEN condition;}

END LOOP;

**2) While Loop**

A While Loop is used when a set of statements has to be executed as long as a condition is true. The condition is evaluated at the beginning of each iteration. The iteration continues until the condition becomes false.

**General Syntax:**

WHILE <condition>

LOOP statements;

END LOOP;

**3) FOR Loop**

A For Loop is used to execute a set of statements for a predetermined number of times. Iteration occurs between the start and end integer values given. The counter is always incremented by 1. The loop exits when the counter reaches the value of the end integer.

**General Syntax:**

FOR counter IN val1..val2

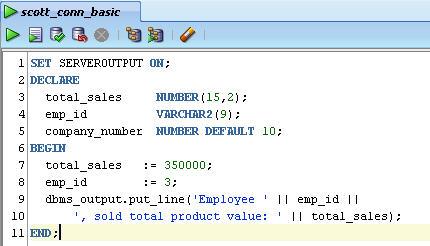
LOOP statements;

END LOOP;

val1 - Start integer value.

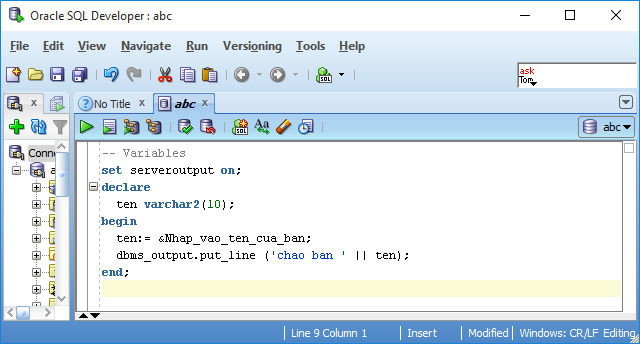
val2 - End integer value

1. **Try the following blocks and explain the results**



* The value of **total\_sales** is 350000.00 \*number(15,2)\*
* The value of **emp\_id** is ‘3’ \*varchar2(9)\*

=> System printed: *Employee 3, sold total product value: 350000.00*

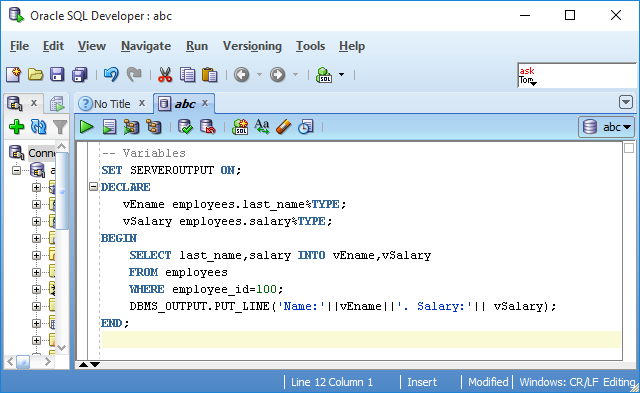


* Input **ten** as ‘Moo Moo’ \*varchar2(10)\*

=> System printed: *chao ban Moo Moo*

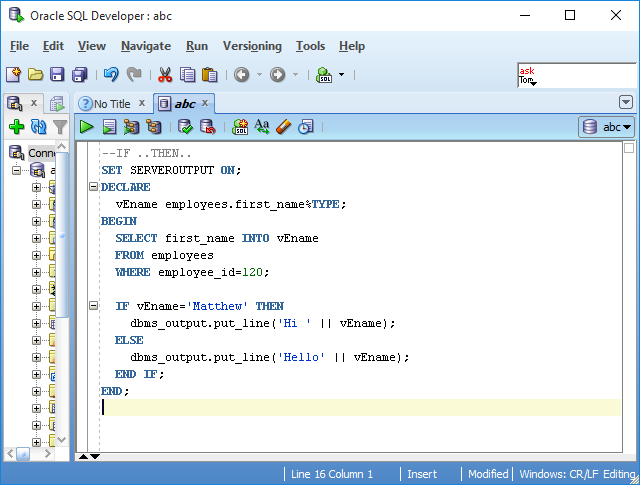
Install HR schema:

<https://matthiashoys.wordpress.com/2020/07/09/installation-of-hr-sample-schema-on-free-oracle-atp-cloud-database/>



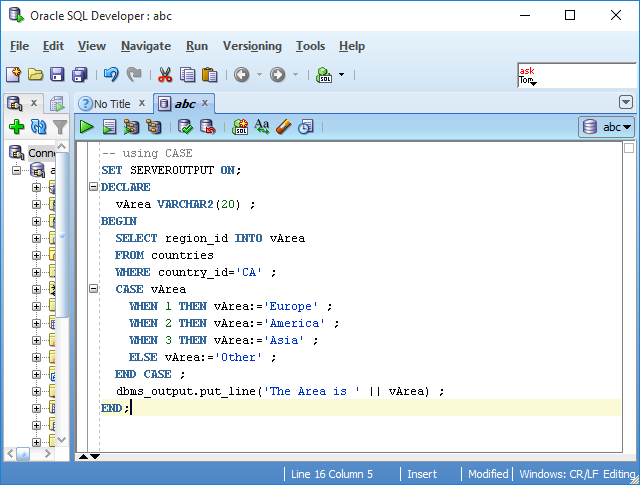
With employee\_id=100, we have **vEname** = last\_name = ‘King’, **vSalary** = salary =24000

=> System printed: *Name: King. Salary: 24000*



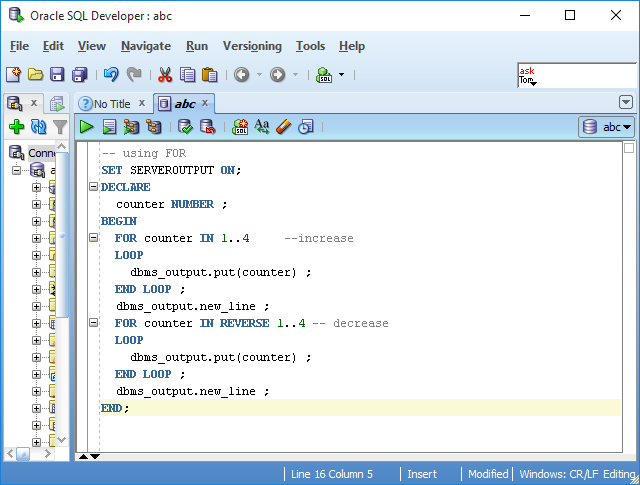
* **vEname** = employee.first\_name = ‘Matthew’ (employee\_id=120)

=> System printed: *Hi Matthew*



* country\_id= ‘CA’ => Canada => region\_id = 2 => vArea = America

=> System printed: *The Area is America*

*Hi Matthew*

* The first loop will printed: *1234*
* The next line will be printed in a new line
* The second loop will printed: *4321*
* The next line will be printed in a new line

=> System printed: *1234  
 4321*