GALGOTIAS UNIVERSITY

School of Computing Science and Engineering

Program: BCA - IOP

Course Code: BCAS3031

Course Name: PL/SQL & Cursors and Triggers

Dr. T. Poongodi

Associate Professor

Course Outcomes :

Course outcomes (COs)	
CO1	Understand the basic concepts of PL/SQL
CO2	Enumerate PL/SQL code constructs using looping structures
CO3	Solve database problems using the concept of cursors and triggers
CO4	Distinguish the structure of packages, collections and records
CO5	Construct programming skills for handling exceptions.
CO6	Acquire hands-on experience with case studies

Syllabus

Unit I **Introduction to PL/SQL** **8 hours**

List the different Types of Identifiers in a PL/SQL subprogram, Usage of the Declarative Section to Define Identifiers, Use of variables to store data, Scalar Data Types, %TYPE Attribute, Bind Variables, Sequences in PL/SQL Expressions, Basic PL/SQL Block Syntax Guidelines, SQL Functions in PL/SQL, Data Type Conversion, Nested Blocks, Operators in PL/SQL, SELECT Statements in PL/SQL to Retrieve data, Data Manipulation in the Server Using PL/SQL, How to save and discard transactions

Unit II **Conditional Statements and Procedures** **8 hours**

Conditional processing Using IF Statements, Conditional processing Using CASE Statements, Simple Loop Statement, While Loop Statement, For Loop Statement, The Continue Statement, Composite Data Types, PL/SQL Records, The %ROWTYPE Attribute, Insert and Update with PL/SQL Records, Associative Arrays (INDEX BY Tables), INDEX BY Table Methods, INDEX BY Table of Records, Procedure, Create a Simple Procedure; Create a Simple Procedure with IN parameter, Function, Create a Simple Function.

Unit III **Cursor and Trigger** **8 hours**

Cursor, Declare the Cursor, Cursor Attributes, Cursor Variables, Cursor Expressions, Fetching data from the Cursor, Cursor FOR loop, Explicit Cursors, Explicit Cursor Attributes, FOR UPDATE Clause and WHERE CURRENT Clause, Triggers, Trigger Architecture, Trigger types, Sample trigger with example, Trigger concept.

Unit IV **Packages, Collections and Records** **8 hours**

Packages, Package Architecture, Package Specifications, %TYPE and %ROWTYPE, Advantages of Package, Overview of Collections and Records: Structure of Varray and Nested Table, Initializing, Referencing and Comparing Collections, Operations on Collections and Varray, Manipulation in nested Table and Varray, PL/SQL Table.

UNIT V **Exception Handling** **6 hours**

Exception Handling, Handle Exceptions with PL/SQL, User-Defined Exception, RAISE_APPLICATION_ERROR, Exceptions raised in Declarations, SQLCODE and SQLERRM.

UNIT VI **Case Studies** **6 hours**

Payroll Processing System, Gas booking and delivering, Conducting online Quiz, Bank transaction, Library information system, Electricity bill processing, Material requirement processing, Railway reservation system, Web based user identification system

- PL/SQL is designed for seamless processing of SQL statements enhancing the security, portability, and robustness of the database.
- It gives more control to the programmers by the use of loops, conditions and object-oriented concepts.
- PL/SQL is "Procedural Language extensions to SQL".

Difference between SQL and PL/SQL

•SQL is a single query that is used to perform DML and DDL operations.	•PL/SQL is a block of codes that used to write the entire program blocks/ procedure/ function, etc.
•It is declarative, that defines what need to be done, rather than how things need to be done.	•PL/SQL is procedural that defines how the things needs to be done.
•Execute as a single statement.	•Execute as a whole block.
•Mainly used to manipulate data.	•Mainly used to create an application.
•Interaction with a Database server.	•No interaction with the database server.
•Cannot contain PL/SQL code in it.	•It is an extension of SQL, so that it can contain SQL inside it.

Declare Variable, Identifiers, Naming Conventions in PL/SQL

What is Identifiers?

- Identifiers are names given to a PL/SQL object. The object could be constant, variable, exception, cursors, procedures, function, package, trigger, object type, reserved word or label.

Properties of Identifiers

- Must start with a letter
- Maximum size is limited to 30 letters
- Cannot contain whitespace characters
- Can contain dollar sign ('$'), underscore ('_') and hash sign ('#')
- Is case-insensitive

Naming Conventions of Identifiers in PL/SQL

- The first letter should be used to specify the declared level of the variable. The below point give the different first letters and their declarative level
  - 'P' – Variable is declared at the parameter level
  - 'L' – Variable is declared at the local block
  - 'G' – Variable is declared at the global level

- The second letter specifies the type of identifier. Below are the commonly used identifier types and their naming code.
  - 'C' – Cursor Identifier
  - 'V' – Varchar and char datatype
  - 'N' – Number datatype
  - 'R' – Record type
  - 'T' – Table type

Some of the examples of proper naming conventions

- Lv_name – local level variable of varchar/char datatype
- Pc_num – parameter level cursor identifier
- Gn_user_id – Global level variable of numerical data type

- The identifiers consist of a letter optionally followed by more letters, numerals, dollar signs, underscores, and number signs and should not exceed 30 characters.
- By default, **identifiers are not case-sensitive**. So you can use **integer** or **INTEGER** to represent a numeric value. You cannot use a reserved keyword as an identifier.

- The PL/SQL supports single-line and multi-line comments. All characters available inside any comment are ignored by the PL/SQL compiler.
- The PL/SQL single-line comments start with the delimiter -- (double hyphen) and multi-line comments are enclosed by /* and */.

```
DECLARE
   -- variable declaration
   message  varchar2(20):= 'Hello, World!';
BEGIN
   /*
   *  PL/SQL executable statement(s)
   */
   dbms_output.put_line(message);
END;
/
```

Variables – An Identifier

- Variable is the basic identifier which is used more frequently, where the user can store the value.
- This variable needs to be associated with some valid PL/SQL datatype before using them.
- The datatype will define the storage and processing method for these variables.

Declaration of Variables

- Variables are mainly used to store data during the data manipulation or data processing.
- They need to be declared before using them inside the program. This declaration needs to be done in the declarative section of the PL/SQL blocks.
- Declaration of variables is a process of assigning the name with a valid datatype.

Syntax

<variable name> <datatype>;

<variable_name> <datatype> := <default_value>;

The above syntax shows how to declare the variable and assign value in the declarative section.

<Variable_name> <datatype>; <variable name> := <value>;

```
DECLARE
lv_name VARCHAR2(50);
lv_name_2 VARCHAR2(50) := 'PL/SQL';
BEGIN
lv_name := lv_name_2;
dbms_output .put_line(lv_name);
END:
```

Code Explanation:

- **Code line 2**: Declaring the variable 'lv_name' of VARCHAR2 with size 50.
- **Code line 3**: Declaring the variable 'lv_name_2' of VARCHAR2 with size 50 and assigned the default value using literal ‘PL/SQL'.
- **Code line 5**: Value for variable 'lv_name' has been assigned from the variable 'lv_name_2'.
- **Code line 6**: Printing the stored value of variable 'lv_name'.
- When the above code is executed, the output will be,

Output:

PL/SQL


Thank You