|  |  |
| --- | --- |
| **A picture containing logo  Description automatically generated** | **DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING**  **MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY, JAMSHORO**  **Database Management Systems (4th Semester) 18CS**  **Lab Experiment 11** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Roll No:** |  | **Date of Conduct:** |  |
| **Submission Date:** |  | **Grade Obtained:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem Recognition (0.3)** | **Completeness & accuracy (0.4)** | **Timeliness (0.3)** | **Score (1.0)** |
|  |  |  |  |

**Objective**: **To understand and practice PL/SQL block structure, Control Structures and Data Types.**

**Tools: MySql, Oracle.**

**Introduction:**

**PL/SQL Block Structure**

In PL/SQL, as in most other procedural languages, the smallest meaningful grouping of code is known as a block. A block is a unit of code that provides execution and scoping boundaries for variable declarations and exception handling. PL/SQL allows you to create anonymous blocks (blocks of code that have no name) and named blocks, which may be packages, procedures, functions, triggers, or object types..

**A PL/SQL block has up to four different sections, only one of which is mandatory:**

**Header**

Used only for named blocks. The header determines the way the named block or program must be called. Optional.

**Declaration section**

Identifies variables, cursors, and subblocks that are referenced in the execution and exception sections. Optional.

**Execution section**

Statements the PL/SQL runtime engine will execute at runtime. Mandatory.

**Exception section**

Handles exceptions to normal processing (warnings and error conditions). Optional

**Basic Syntax structure**

**Declare**

Declaration of variable, constants

**Begin**

Execute statements in pl/sql

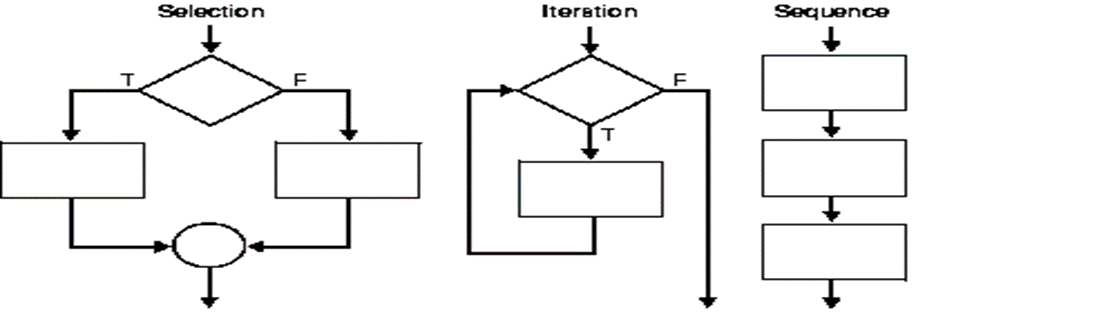
**Exception**

Exception handlers in pl/sql

**END;**

**PL/SQL Control Structures**

The selection structure tests a condition, then executes one sequence of statements instead of another, depending on whether the condition is true or false. A condition is any variable or expression that returns a Boolean value (TRUE or FALSE). The iteration structure executes a sequence of statements repeatedly as long as a condition holds true. The sequence structure simply executes a sequence of statements in the order in which they occur.

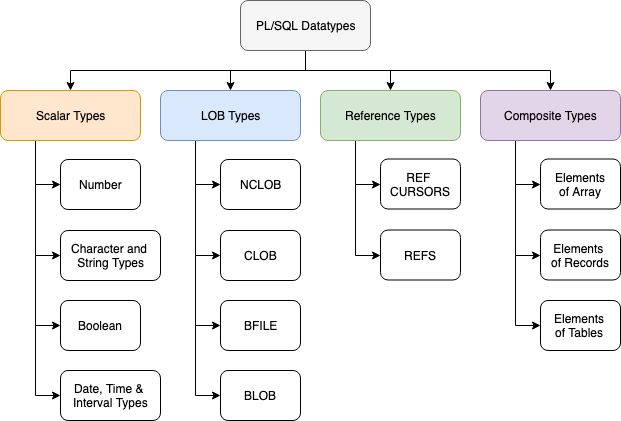


**PL/SQL Data Types.**

PL/SQL datatypes are not just limited to writing SQL queries but they are used in the PL/SQL block as well, just like any other programming language.

Provising a datatype specifies how any data will be stored and processed by Oracle when any PL/SQL code block is executed.

Datatype defines the type of data being used, whether it is a number or a word (string) or a single character etc. Following datatypes can be used in PL/SQL depending upon the type of data required:



**So we have 4 broader categories of datatypes and they are:**

**1. Scalar Types:** These are basic datatypes which generally holds a single value like a number or a string of characters. Scalar types have 4 different categories which are listed in the diagram above, namely Number Types, Character and String, Boolean Types and Date and Time etc.

2. **LOB Types**: This datatype deals with large objects and is used to specify location of these large objects like text files, images etc which are generally not stored outside the database.

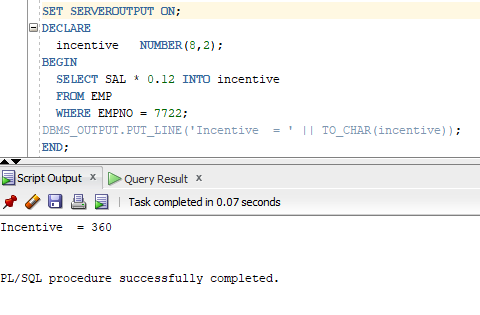
3. **Reference Types:** This datatype is used to hold pointer values which generally stores address of other program items.

4. **Composite Types**: Last but not the least, as the name suggests this type of data is a composition of individual data which can be manipulated/processed separatel as well.

**LAB TASK**

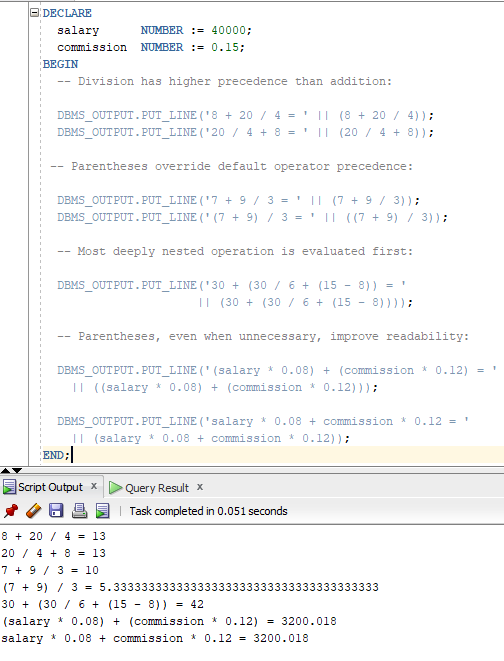
1. **Write a PL/SQL block to calculate the annual salary of an employee whose ID is 7722.**

**Task:**

****

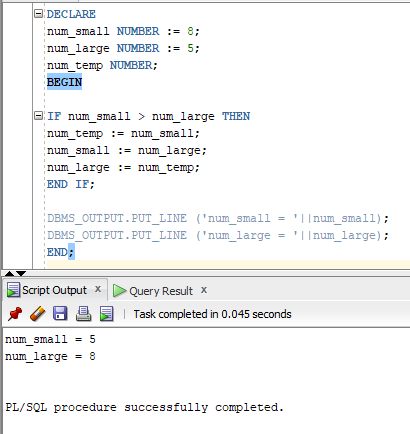
1. **Write a PL/SQL block to show the operator precedence and parentheses in 5 or more complex expressions.**

**Task:**

****

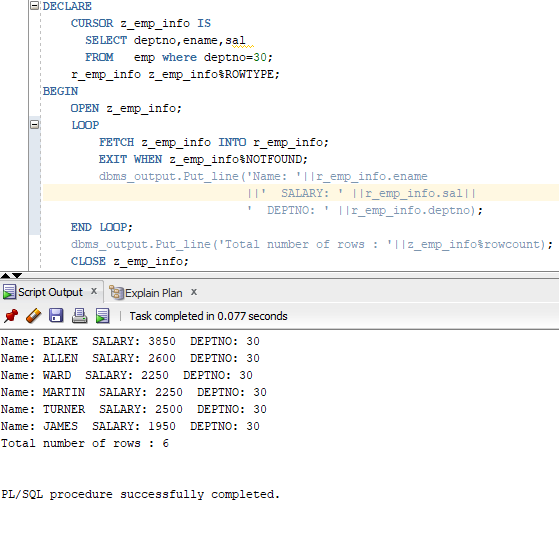
1. **Write a PL/SQL program to arrange the number of two variable in such a way that the small number will store in num\_small variable and large number will store in num\_large variable.**

**Task:**

****

1. **Write a PL/SQL program to count number of employees in department 30 and check whether this department have any vacancies or not. There are total 45 posts in this department.**

**Task:**

****

1. **Write a program in PL/SQL to check whether a number is prime or not using for loop.**

**Task:**

