**PL/SQL Fundamental Exercises**

1. Write a PL/SQL block to calculate the incentive of an employee whose ID is 110.

**DECLARE**

**v\_salary INTEGER(8,2);**

**begin**

**select salary\*0.12**

**INTO v\_salary**

**from tblemployees**

**where employee\_id=&emp\_id;**

**DBMS\_OUTPUT.put\_line('Incentive of the employee is: '||v\_salary||' rs.');**

**end;**

**/**

1. Write a PL/SQL block to show an invalid case-insensitive reference to a quoted and without quoted user-defined identifier.

**set SERVEROUTPUT ON**

**DECLARE**

**V\_LAB VARCHAR2(5):='v\_lab';**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('Case Insensitive Variable: '||v\_lab);**

**DBMS\_OUTPUT.put\_line('Case Sensitive Variable: '||"V\_LAB");**

**END;**

**/**

1. Write a PL/SQL block to show a reserved word can be used as a user-define identifier.

**set SERVEROUTPUT ON**

**DECLARE**

**"WHERE" VARCHAR2(5):='WHERE';**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('Reserve Word As Variable: '||"WHERE");**

**END;**

**/**

1. Write a PL/SQL block to show the result to neglect double quotation marks in reserved word identifier.

**set SERVEROUTPUT ON**

**DECLARE**

**WHERE VARCHAR2(5):='WHERE';**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('Reserve Word As Variable: '||"WHERE");**

**END;**

**/**

**Or**

**set SERVEROUTPUT ON**

**DECLARE**

**"WHERE" VARCHAR2(5):='WHERE';**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('Reserve Word As Variable: '||WHERE);**

**END;**

**/**

1. Write a PL/SQL block to show the result to neglect the case sensitivity of a user defined identifier which is also a reserved word.

**set SERVEROUTPUT ON**

**DECLARE**

**"WHERE" VARCHAR2(5):='WHERE';**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('Reserve Word As Variable: '||"where");**

**END;**

**/**

1. Write a PL/SQL block to explain single and multiline comments.

DECLARE

**BEGIN**

**--Single Line Comment**

**/\***

**Multi line Comment**

**\*/**

**DBMS\_OUTPUT.PUT\_LINE('Single and Multiline Comments');**

**END;**

**/**

1. Write PL/SQL blocks to show the declaration of variables.

**DECLARE**

**"IDENTIFIER" VARCHAR(10);**

**BEGIN**

**"IDENTIFIER":='IDENTIFIER';**

**END;**

**/**

1. Write PL/SQL blocks to show the scope and visibility of local and global identifiers.

**DECLARE**

**-- Global variables**

**num1 number := 95;**

**num2 number := 85;**

**BEGIN**

**dbms\_output.put\_line('Outer Variable num1: ' || num1);**

**dbms\_output.put\_line('Outer Variable num2: ' || num2);**

**DECLARE**

**-- Local variables**

**num1 number := 195;**

**num2 number := 185;**

**BEGIN**

**dbms\_output.put\_line('Inner Variable num1: ' || num1);**

**dbms\_output.put\_line('Inner Variable num2: ' || num2);**

**END;**

**END;**

**/**

1. Write a PL/SQL block to show a valid case-insensitive reference to a quoted and without quoted user-defined identifier.

**set SERVEROUTPUT ON**

**DECLARE**

**"V\_LAB" VARCHAR2(5):='v\_lab';**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('Case Insensitive Variable: '||v\_lab);**

**DBMS\_OUTPUT.put\_line('Case Sensitive Variable: '||V\_LAB);**

**END;**

**/**

1. Write a PL/SQL block to adjust the salary of the employee whose ID 122.

**DECLARE**

**v\_salary INTEGER(8,2);**

**begin**

**select salary+&increased\_sal**

**INTO v\_salary**

**from tblemployees**

**where employee\_id=&emp\_id;**

**DBMS\_OUTPUT.put\_line('Incentive of the employee is: '||v\_salary||' rs.');**

**end;**

**/**

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

**DECLARE**

**BEGIN**

**DBMS\_OUTPUT.PUT\_LINE('According To BODMOS RULE the solution of the expression [72-12/3]+(18-6)/4 is 71');**

**DBMS\_OUTPUT.PUT\_LINE('OUTPUT IS '||((72-12/3)+(18-6)/4));**

**END;**

**/**

1. Write a PL/SQL block to create a procedure using the "IS [NOT] NULL Operator" and show AND operator returns TRUE if and only if both operands are TRUE.
2. Write a PL/SQL block to create a procedure using the "IS [NOT] NULL Operator" and show OR operator returns TRUE if either operand is TRUE.
3. Write a PL/SQL block to create a procedure using the "IS [NOT] NULL Operator" and show NOT operator returns the opposite of its operand, unless the operand is NULL.
4. Write a PL/SQL block to describe the usage of NULL values in equal comparison, unequal comparison and NOT NULL equals NULL comparison.
5. Write a PL/SQL block to describe the usage of LIKE operator including wildcard characters and escape character.