**PL/SQL Fundamental Exercises**

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

SET SERVEROUTPUT ON;

DECLARE

e\_incentive number(10);

BEGIN

select salary\*0.4

into e\_incentive

from employees

where employee\_id = 110;

DBMS\_OUTPUT.PUT\_LINE('Incentive'||e\_incentive);

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 varchar(15):='hello world';

BEGIN

DBMS\_OUTPUT.PUT\_LINE(''|| "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

"From" varchar(15):= 'hello world';

BEGIN

DBMS\_OUTPUT.PUT\_LINE('SENTENCE IS'||"From");

end;

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

set SERVEROUTPUT ON;

DECLARE

From varchar(15):= 'hello world';

BEGIN

DBMS\_OUTPUT.PUT\_LINE('SENTENCE IS'||From);

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

"FROM" varchar(15):= 'hello world';

begin

DBMS\_OUTPUT.PUT\_LINE('Sentence is'||"from");

end;

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

set SERVEROUTPUT ON;

DECLARE

"FROM" varchar(15):= 'hello world';--declaring variable

begin

DBMS\_OUTPUT.PUT\_LINE('Sentence is'||"from");

/\*display

output\*/

end;

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

set SERVEROUTPUT ON;

DECLARE

v\_string varchar(15):= 'hello world';

v\_num number(10):= 100;

begin

DBMS\_OUTPUT.PUT\_LINE('Sentence is '||v\_string);

DBMS\_OUTPUT.PUT\_LINE('no. is '||v\_num);

end;

1. Write PL/SQL blocks to show the scope and visibility of local and global identifiers.
2. 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 varchar(15):='hello world';

BEGIN

DBMS\_OUTPUT.PUT\_LINE(''|| v\_lab);

end;

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

set SERVEROUTPUT ON;

DECLARE

v\_adjust number(10);

BEGIN

select salary+1000 into v\_adjust from employees where employee\_id = 122;

DBMS\_OUTPUT.PUT\_LINE('adjusted salary is '|| v\_adjust);

end;

1. Write a PL/SQL block to show the operator precedence and parentheses in several more complex expressions.
2. 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.
3. 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.
4. 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.
5. Write a PL/SQL block to describe the usage of NULL values in equal comparison, unequal comparison and NOT NULL equals NULL comparison.
6. Write a PL/SQL block to describe the usage of LIKE operator including wildcard characters and escape character.