

machine exercise - III rd

Q1. Differentiate between DDL and DML.

Ans:- There are many differentiate between DDL and DML.

<u>DDL</u>	<u>DML</u>
(i) It is used to create database schema and can be used to define some constraints as well.	It is used to add, retrieve or update the data.
(ii) It basically defines the column of the table.	It add or update the row of the table.
(iii) It doesn't have any further classification.	It is further classified into procedural and non-procedural DML.
iv) Basic Command present in DDL are CREATE, DROP, ALTER etc.	Basic command present in DML are update, INSERT, MERGE etc.
v) DDL does not use WHERE clause in its statement.	DML uses WHERE clause in its statement.

1

Expt. No. _____

Q2. write the Syntax for following commands.

(a) Create :- Syntax :-

`CREATE TABLE Table-name`

`(column-name1, column-name2, ...)`

`AS SELECT STATEMENT;`

(b) Insert :- Syntax :-

`INSERT INTO table-name`

`(column-name1, column-name2, ...)`

`VALUES`

`(column-value1, column-value2, ...);`

(c) Delete :- Syntax :-

`DELETE FROM table-name`

`[WHERE CONDITION] ;`



(d) Select :- Syntax :-

`SELECT column-name FROM Table-name ;`

or,

`SELECT * FROM table-name ;`

(e) Update :- Syntax :-

`UPDATE table-name`

`SET column-name = <expression>, [column-name2 = <expression>]`

`[WHERE condition] ;`

Teacher's Signature _____

Date _____

Page No. _____

Expt. No. _____

Q3. what is PL/SQL block? what are its various types?

Ans:- PL/SQL blocks can include variables, SQL statements, loops, constants, conditional statements and exception handling. Blocks can also build a function or a procedure or a package.

There are two various types of PL/SQL.

(i) Anonymous blocks:- In PL/SQL, that's blocks which is not have header are known as anonymous blocks. These blocks do not form the body of a function or triggers or procedure.

(ii) Named blocks:- that's PL/SQL blocks which having header or labels are known as named blocks. These blocks can either be subprograms like functions, procedures, packages or triggers.

Eg:- Declare

Begin



Exception

End;

Teacher's Signature _____

D4. what is an anonymous block? what is its structure? How can we declare and initialize a variable in PL/SQL?

Ans:- PL/SQL is a block-structured language whose code is organized into blocks. A PL/SQL block consists of three sections: declaration, executable and executable-handling section. An anonymous block is not saved in the Oracle Database Server, so it is just for one-time use.

Its structure of PL/SQL block.

- (i) Declaration Section:-
- (ii) Executable section.
- (iii) Exception-handling section.



(i) Declaration section:- A PL/SQL block has a declaration section where you "declare variables," allocate memory for "cursors," and define data types.

(ii) Executable section:- A PL/SQL block has an executable section. An executable section starts with the keyword "BEGIN" and ends with keyword "END".

(iii) Exception-handling section :- A PL/SQL block has an exception handling section that starts with the keyword "exception".

Declaring and initializing variables :-

- Forward references are not allowed. i.e.- Variable must be declared before referencing in another statement.
- Variables of same datatype cannot be declared in same statement.
- Variable name should not exceed 30 characters.
- PL/SQL is case sensitive.

Syntax of declaring variable :-

variable_name [CONSTANT] datatype [NOT NULL] [:] DEFAULT
[initial_value]

Ex:-

DECLARE

a integer := 10;
b integer := 20;
c integer;

Q5. Explain various operators in PL/SQL along with their hierarchy.

Ans:- The various categories of operators are :-

- (i) Arithmetic operators.
- (ii) Comparison operators.
- (iii) Logical operators.
- (iv) Concatenation operators.
- (v) Set operators.
- (vi) Other in-built operators.

(i) Arithmetic operators:- The arithmetic operators are used in an expression to add, subtract, multiply, negate, divide numeric values. The result of the operation is also a numeric value. The plus(+) and minus(-) are also used in date arithmetic.

(ii) Comparison operators:- The Comparison operators are used to compare one expression with other. It's operators are $>$, $<$, \geq , \leq , $=$, \neq , IN, BETWEEN, LIKE etc.

(iii) Logical operators:- The logical operators are used to combine the individual logical expressions into more complex conditions are TRUE or FALSE. Its operators are AND, OR, NOT etc.

Date _____

Expt. No. _____

Page No. _____

(iv) CONCATENATION OPERATOR(//) :- The concatenation operator is used to concatenate two or more strings.

(v) SET OPERATORS :- The set operators Combines the result of two Component queries into a Single result. It's operators are UNION, INTERSECTION etc.

(vi) OTHER-IN-BUILT OPERATORS :- Some other in-built operators include (+) which used in outer Join operations, PRIOR used in tree structure queries.

✓

Teacher's Signature _____

Ques No. _____

Ques. What are the basic Control Statement in PL/SQL? write their syntax with examples.

Ans:- Control statements are conditional statement and loop constructs which help you to build the code that gets executed in PL/SQL.

The various control statement include -

- (i) Conditional control statement
- (ii) Iterative control statement
- (iii) Sequential control statement

(i) Conditional control statement:- PL/SQL allows the use of an IF statement to control the execution of a block of code. In PL/SQL, the IF-THEN, ELSE, IF-THEN can IF construct in code blocks allow specifying certain conditions under which a specific block of code should be executed.

Syntax:-

IF < condition > THEN

< Action >

ELSE < condition > THEN

< Action >

ELSE < Action >

END IF;



Date _____

Page No. _____

Expt. No. _____

Example :- Declare

n integer;

BEGIN

n := &n;

If n > 0 Then

If n mod 2 = 0 Then

dbms_output.put_line (n || ' is even');

ELSE

dbms_output.put_line (n || ' is odd');

END If;

ELSE

dbms_output.put_line ('Number is zero');

END If;

END;

1

(ii) ITERATIVE CONTROL STATEMENTS :- The Iterative Control statements in PL/SQL are used to execute the same code repeatedly. i.e.- It enables you to execute a sequence of statements multiple times. There are three basic iterative control statements used in PL/SQL. simple loop statement, while loop statement, for loop statement.

Syntax :-

. Loop

< sequence of statements >

End loop;

Teacher's Signature _____

OUTPUT:- Enter value for n: 5

old 4:n:=~~zn~~;

new 4:n = 5;

5 is odd

PL/SQL procedure successfully completed.

Example :-

```

    Declare
        n number := 0;
    Begin
        Loop
            n := n + 1;
        Exit WHEN n > 3;
        dbms_output.put_line('Loop executes ' || n || ' times');
    END Loop;
    END;
    /

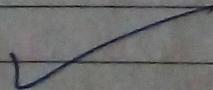
```

(iii) SEQUENTIAL CONTROL STATEMENT:- By default all PL/SQL blocks are executed in a top-down sequential process. The process begins with a BEGIN statement and terminates with an END statement.

Syntax:-

GOTO jump;

<<jump>>



Example:- DECLARE

n number := 2^n;

Begin

IF N mod 2 = 0 THEN

dbms_output.put_line('Number is Even');

ELSE

NULL;

END IF;

END;

OUTPUT :-

Loop executes 1 times

Loop executes 2 times

Loop executes 3 times

PL/SQL procedure successfully completed.

Date _____

Expt. No. _____

Page No. _____

Q7. write an anonymous block to display details of a specific employee.

Ans:- declare

eno emp.empno% type;

en emp.ename% type;

dept emp.deptno% type;

begin

Select empno, ename, deptno into eno, en,

dept from emp

where empno = & eno;

dbms_output.put_line(eno||'/'||en||'/'||dept);

exception

when no_data_found then

dbms_output.put_line('NO employee found');

end;



Teacher's Signature _____

Q8. write an anonymous block to check whether a given number is even or odd.

SOLⁿ:- Declare

n integer;

BEGIN

N:=&N;

if n mod 2=0 then

dbms_output.put_line (n||' is EVEN');

ELSE

dbms_output.put_line (n||' is ODD');

END IF;

END;

/

Q9. write an anonymous block to increase the salary of a given employee by a specific amount.

SOLⁿ:-

declare

eno emp.empno%type;

amt emp.sal%type;

begin

eno:=&employeeeno;

amt:=&amount;

..



08.

OUTPUT:- Enter value for n: 7

old 4:N:=2N;

new 4:N:= 7;

7 is ODD

PL/SQL Procedure Successfully Completed.

Date _____

Expt. No. _____

Page No. _____

update emp set sal = sal + amt where empno = empno;
exception
when no_date_found then
dbms_output.put_line ("NO employee found");
end;

✓

Teacher's Signature _____

Q10. write an anonymous block to check whether a given number is prime or not.

Sol:- declare

```
n number;
i number;
flag number;
begin
i:=2;
flag:=1;
n:=2;
```

for i in 2..n/2

loop

if mod(n,i)=0

then

flag:=0;

exit;

end if;

end loop;

if flag=1

then

dbms_output.put_line('Prime');

else

dbms_output.put_line('not prime');

end if;

end;

/



Q 10.

Output:- Enter value for n:12

old g:n:=2n;

new g:n:=12

not prime.

Q11. what do you understand by %TYPE and %ROWTYPE attributes. Give example to illustrate.

Ans :- %TYPE:- The %Type attribute is used to declare variable according to the already declared variable or database column. It is used when you are declaring individual variable, not a record. The

Syntax :-

<var-name><tab-name>.<column-name>%TYPE;

Example :- Declare

SALARY emp.SAL%TYPE;

EPCODE emp.EMPNO%TYPE;

BEGIN

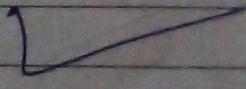
EPCODE := &EPCODE;

Select SAL INTO SALARY FROM EMP WHERE EMPNO = EPCODE;

DBMS_OUTPUT.PUT_LINE('Salary of' || EPCODE || 'is =' || SALARY);

END;

/



%ROWTYPE :- %ROWTYPE attribute describe a record type that represents a row in the table. The record can store an entire row or some specific data selected from the table.

Syntax :- <var-name><tab-name>%ROWTYPE;

Q.11.

./TYPE

OUTPUT :- Enter value for ecode : 7499

Salary of 7499 is = 1600

PL/SQL procedure successfully completed.

Date No. _____

Example:- declare

EMPLOYEE EMP% ROWTYPE;

BEGIN

EMPLOYEE.EMPNO:=2092;

EMPLOYEE.ENAME:='Sanju';

Insert into EMP(EMPNO, ENAME)

VALUES (employee.empno, employee.ename);

DBMS_OUTPUT.PUT_LINE('Row Inserted');

END;

/

Q12. what do you understand by a cursor for loop.
 Give syntax and examples.

Ans:- A Cursor is a type of pointer built into PL/SQL for querying the database, retrieving a set of records and allowing a developer to access the active data set, a row at a time.
 Two types of the most common Cursor used in oracle are -

(i) Implicit Cursors.

(ii) Explicit Cursors.

Syntax:- using loop statement;

Teacher's Signature _____

%ROWTYPE :-

OUTPUT :-

Row Inserted.

PL/SQL procedure successfully completed.

Syntax:- using loop statement.

DECLARE

CURSOR <Cursor-name> IS
<SELECT statement>;

BEGIN

FOR I IN <cursor-name>
Loop

END LOOP;

END;

Example:- declare

cursor emp-data is

select empno,ename,sal from emp where deptno=20
ecode emp.empno%type;
name emp.ename%type;
salary emp.salary%type;

begin

open emp-data;

loop

fetch emp-data into ecode,ename,salary;

exit when emp-data%not found;

dbms_output.put_line(ecode||' '||ename||' '||salary);

end loop;

close emp-data;

end;

Teacher's Signature _____

Q12.

OUTPUT :-

Enter value for deptno: 20

old 3: Select empno,ename,sal from emp where deptno=&deptno;

new 3: select empno,ename,sal from emp where deptno= 20;

PL/SQL procedure successfully completed.

Q13. write a PL/SQL block to update the Commission of a specific employee by RS 500 if his Commission is NULL, otherwise increase his Commission by 25%.

SOL3 - declare

eno emp.empno% type;

c emp.comm% type;

BEGIN

eno := empno;

select comm into c first emp where empno=eno;
if c is null then

update emp set comm = comm + 500 where
empno=eno;

else

update emp set comm = comm * 1.25
where empno=eno;

end if;

end;



Q14. Define a cursor in PL/SQL along with its type. What is the need for a cursor? Explain the various operations associated with an explicit cursor?

Ans:- A cursor is a type of pointer built into PL/SQL for querying the database, retrieving a set of records and allowing a developer access the active data set, a row at a time.

There are two types of cursor in PL/SQL.

(i) Implicit Cursor.

(ii) Explicit Cursor.

(i) Implicit Cursor:- Cursors those are created automatically by Oracle where SQL statement is executed, when there is no explicit cursor for statement.

(ii) Explicit Cursor:- Cursors those are created by users themselves to retrieve multiple rows at same time.

Need for Cursor :- To execute a multi-row query. Oracle opens an unsaved work area that stores processed information. A cursor lets you name the work area accessing the information & process row individually.

There are operation associated with explicit cursor as below —

(i) Declaring an explicit cursor:-

Syntax:-

cursor Cursor-name

is

select statement;

(ii) Opening / operating an explicit cursor:-

Syntax:-

open Cursor name;

(iii) Fetching values from explicit cursor:-

Syntax:-

fetch Cursor.name into <var-lst>;

(iv) closing an explicit cursor:-

Syntax:-

close Cursorname;

Eg:- To display details of employee from specific department;

declare

x emp.empno%type;

y emp.ename%type;

Cursor emp is Select empno,ename from emp where deptno = <dept>;

Begin

open emp;

loop

fetch emp into x,y;

if emp%found then

dbms_output.put_line(x||'/'||y);

else

exit;

end if;

end loop;

end;



Q15. create a cursor that displays the names and salaries of all employees who were hired in a specific month. use a parameterized cursor.

SOLN - declare

name emp.ename%type;

s emp.sal%type;

cursor c1 (month number)

is

select ename, sal from emp

where to_char(hiredate,'mm')=month;

BEGIN

open c1(4);

loop

fetch c1 into name s,

exit when c1%not found;

dbms_output.put_line(name||' '||s);

end loop;

close c1;

end;

/