Unit - 6

PL / SQL (PL+SQL)

procedural Langubage.

PL/SQL is a database oriented programming language which combines features of SQL ~~and~~ with procedural Capabilities

- SQL doesnot provide programming techniques looping, branching, Condition checking etc

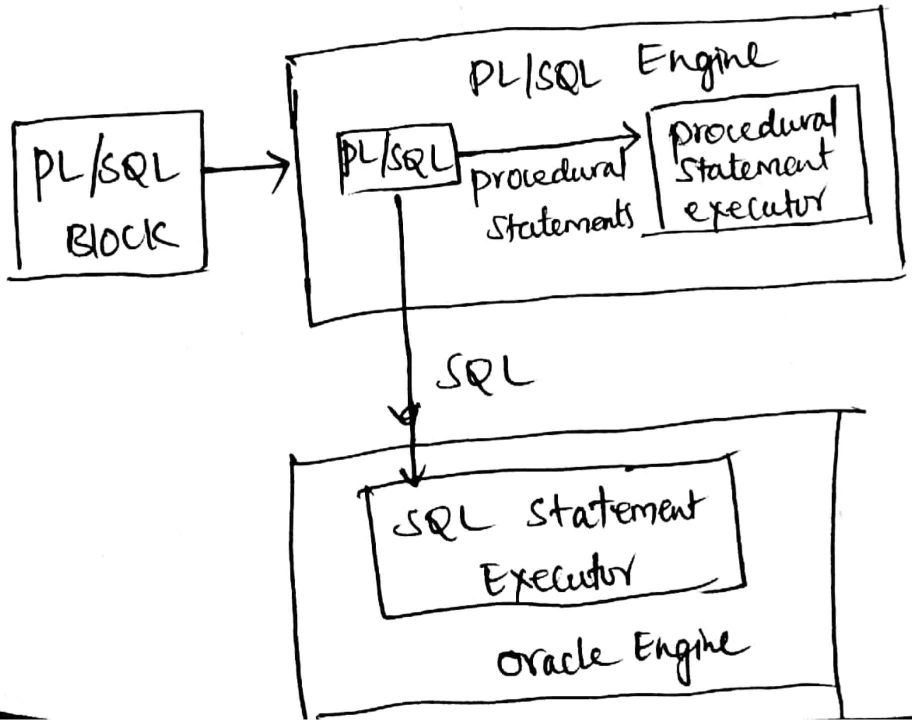
- SQL queries are passed one by one to the oracle engine for processing. this increases traffic on the network and reduces speed of processing

- SQL doesnot provide users to generate their own error messages

Advantages of PL/SQL :-

- looping, Branching is applicable
- processor Speed is fast
- User friendly error messages are applicable
- PL/SQL is portable

Architecture of PL/SQL :-



PL/SQL Block:-

Declare (optional)
Begin (compulsory)
Exception (optional)
END (compulsory)

① write pl/sql code to print "hello world".

Syntax:

```
Begin
dbms_output.put_line("Hello world")
      ↓                  ↓
   package      pre-defined function
End;
```

② write PL/SQL code to add two numbrs.

```
Declare
    a int;
    b int;
    c int;
Begin
    a:= 10;
    b:= 20;
    c:=a+b;
dbms_output.put_line(c);
End;
```

user input

```
a:= &a;
 (or)
a int:= &a;
```

Q③ Create table Student

Rno	CA1	CA2	Total
1	10	20	
2	15	30	

PL/SQL Code

```
Delare
   a int;
   b int;      // Same data type from table
   c int;

Begin
  Select  CA1, CA2 into a,b from Student where
                                        Rno=1;

     c:=a+b;
  Update student set total=c where Rno=1;

  End;
```

output
Total of rno 1 is updated to sum of CA1 and CA2

21/11/2017 INT-306

% Type:- It gives datatype of a column in a table to a variable in PL/SQL

Syntax:-

```
var-name Tablename.colname%type;
```

%rowtype:-

It stores the data type of all the columns of a row.

Syntax:-

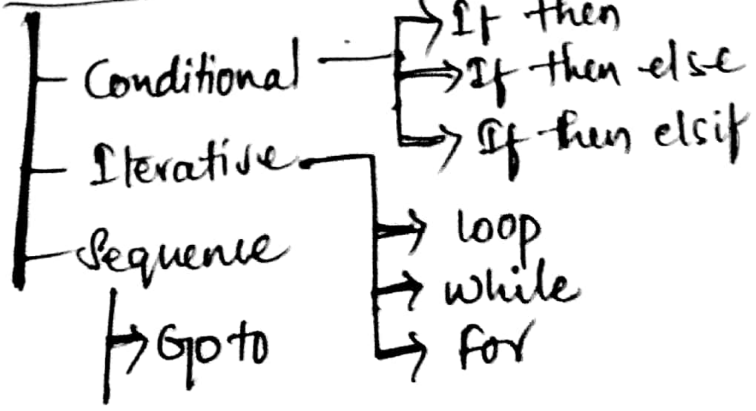
```
var-name Tablename%rowtype;
```

Ex:- for before table | Declare a

```
Begin
Select * into a from student where
        rno=1;
    a.Total= a.CA1 + a.CA2
Opdate student set total=a.total
        where Rno=1;

END
```

Control statements:-



Conditional → If then, If then else, If then elsif

Iterative → loop, while, for

Sequence → Goto

If then Syntax

```
If condition then statements;
    End if;
```

If then else Syntax:

```
If condition then
  statements ;
else
  statements;
Endif)
```

If then elsif

```
If conditions then
  Statements;
elsif conditions then
  Statements;
else
  statements;
End if;
```

① write PL/SQL code to find greatest of two number.

② write PL/SQL code to check whether the number is even or odd.

①

```
Declare
a:=&a;
b:=&b;

Begin
If  a>b  then
  dbms-output.~~Ba~~put-line ("a is greatest');

ElsE
  dbms_output.put_line('b is greatest');

Endif;

END;
```

```
② Declare
      a int:= &a;
     a:= &a;

   Begin
     If (a/2) = 0 then
        dbms_output.put_line('a is even');
      else
        dbms_output.put_line('a is odd');
       Endif;
     END;
```

2/11/2017

INT-306 PL/SQL

③ write PL/SQL code to find the greatest of three numbers

```
  Declare
     a int := &a;
     b int:= &b;
     c int:= &c;
  Begin
     if a>b and a>c then
        dbms_output.put_line('a is greather');
     Elsif b>a and b>c then
        dbms_output.put_line('b is greater');
     Else
        dbms_output.put_line('c is greater');
      ENDif
     END
```

Iterative:-

- Loop
- while loop
- For loop

Syntax:-

```
loop
Statements;
Exit when condition;
 End loop.
```

(4) write PL/SQL to print 100 numbers.

```
Declare
 a int;
Begin     a:=0;
   loop
   dbms_output.put_line(a);
  a:=a+1;
 EXIT when   a≥100;
  END loop;
 END;
```

Syntax for while loop:-

```
while condition.
   Loop
   Statements;
   End loop;
```

(5) write PL/SQL code to print square of first 10 numbers

```
Declare
  a int:=1;
   c int;
Begin
 while (a<=10)
  loop   c:=a*a;
  dbms_output.put_line(c);
       a:a+1;
    End loop;   End;
```

Syntax for 'for' loop:-

(6) write PL/SQL Code to print sum of first ten numbers.

```
declare
    a int := 0;
     b int;
 begin
     while (a <=10),
        loop
           b:= a+1
           b:= a+b
       Endloop;
       dbms_output.put_line (b);

       ~~Endloop;~~
     End;
```

'for' Loop

```
 for  Counter variable In lowerbound.. upperbound.

      loop
    Statements ;
      End loop;
```

(7) write PL/SQL code to print 5 table

```
declare
    a int := 5;
 begin
    for i In 1..10
      loop  a:= a*i;
    dbms_output.put_line('5*'||i||'='||a);
 endloop;
end;
```

8/11/2017

INT-306

Cursors:-

The space or working ~~space~~ area where the result of SQL query is stored. The data stored in cursor is called 'Active data set'.

1. Implicit cursor
2. Explicit cursor (user-defined cursors)

→ It is work area which is declared open and closed internally by the oracle engine the user is not involved in the process of managing cursor

2. Explicit- cursor:- It is the work area which is declared open and closed by user. It is also called "User-defined cursors"

Steps for managing cursor:-

1) Declare cursor
2) open cursor
3) fetch cursor data
4) close cursor

Cursor Attributes:-

1) %ISOPEN 3) %NOTFOUND
2) %FOUND 4) %ROWCOUNT

Above cursor attributes, In implicit it is written as SQL%'Attributename'. In case of Explicit cursorname%'Attributename'.

① write a pl/SQL to display where a message whether a row is updated or not using implicit cursor

```
Begin
 update  T1  Set  Name='D' where  rno= 1;

  If   SQL%FOUND   then
     dbms_output.put_line ('Raw updated');
  Else
      dbms_output.put_line('Not updated');
    ENDIF;
  END;
```

Explicit cursor:-

① Declare syntax:-

```
cursor   cursorname   Is   Select statement;
```

② open syntax:-

```
   open  cursorname;
```

③ Fetch

```
  fetch  cursor  into  variables;
```

④ .close

```
        close  cursorname;
```

② write PL/SQL code to display the name of students from CSE department with the use of explicit cursor method

```
  Declare
  cursor   c is  select name from Tablename(student)
                    where  dept='CSE';

         a  student.name%type;
```

```
           Begin
           loop
fetch    c into   ~~variables~~   a;

         Exit   when   c%NOT FOUND;

         dbms_output.put-line (a);
           END LOOP;
          END;
```

Scanned by CamScanner