

2. PLSQL

Page No.	
Date	

* PLSQL:- Procedural language/structured query language

SQL is a popular language for both querying and updating data in the RDBMS.

PLSQL Adds many procedural construct to SQL language to overcome some limitations of SQL.

PLSQL means instructing the compiler 'what to do' through SQL and a 'How to do' through its procedural way.

mp
marks

PLSQL Block structure:-

Declare :-
// declaration statements;

Begin :-
// Execution statements;

Exceptions:-
// Exception handling statements;
// Error handling statements;

End :-
The end;

Control Statements

① IF

In 'c'

if(condition)

{

Statements;

—

}

In 'PLSQL'

if condition then

Statements;

end if;

② if-else

In 'c'

if(condition)

{

statements;

}

else

{

statements;

}

In 'PLSQL'

if condition then

Statements;

else

Statements;

end if;

③ Nested if-... else

In 'c'

if(condition)

{

statements;

}

else if (condition)

{

statements;

}

else if (condition)

{

statements;

}

else { statements;

}

In 'PLSQL'

if condition then

Statements;

elsif condition then

Statements;

elsif condition then

Statements;

else

statements;

end if;

④ while loop PLSol
 In 'c' (Initialization): initialization;
 while(condition) while condition loop
 { Statements;
 // statements; increment/decrement
 } Increment/decrement; end loop;
 }

⑤ For loop Increment
 for(initialization; cond? inc/dec) For counter-Variable IN
 { initial value .. ending-value
 Statements; loop
 } statements;
 end loop;

decrement (reverse loop)
 For counter-Variable IN reverse
 initial-value .. ending-value loop
 statements;
 end loop;

⑥ Simple loop

loop
 statements;
 end loop;

1. Q. Write a PL/SQL block to check the given number is maximum or minimum

```

→ declare
a int;
b int;
begin
a := 10;
b := 30;
if a > b then
dbms_output.put_line('a is greater');
else
dbms_output.put_line('b is greater');
end if;
end;

```

Q. 2 Write a program to print 1st 10 numbers

Q. 3 write a program to Find Factorial of a number.

```

→ ② declare
n int;
begin
n := 1;
while n <= 10 loop
dbms_output.put_line(n);
n := n + 1;
end loop;
end;

```

```

③ declare
  n int;
  i int;
  fact int;
begin
  i:=1;
  fact:=1;
  while (i<=n) loop
    fact=fact*i;
    i++;
  end loop;
  dbms_output.putline("The fact");
end;

```

```

for loop
declare
  n int;
begin
  for n in 1..11 loop
    dbms_output.putline(n);
  end loop;
end;

```

Maximum between 3 number.

```

declare
  a int;
  b int;
  c int;
begin
  a:=9;
  b:=12;
  c:=11;

```

run runtime sathi value access for user through.

```

  a:=&a;
  b:=&b;
  c:=&c;

```

```
if a > b and a > c
dbms_output.put_line("a is greater");
else if b > a and b > c
dbms_output.put_line("b is greater");
else
dbms_output.put_line("c is greater");
end if;
end;
```

Imp * Exception Handling

Exception:- An exception is an error which disturbs the normal flow of program instruction.

② PL/SQL provides the exception block which raises the exception thus helping the programmer to find out the fault and resolve it.

③ There are two types of exceptions defined in PL/SQL

① Pre-define exception:- These exceptions are pre-defined in PL/SQL which get raised when certain database rule is violated.

It is a pre-defined exception which is already created.

② User define exception:-

① Users can create their own exceptions according to the need and to raise these exceptions explicitly, the raise command is used.

syntax:

when exception then
statement;

syntax with PL/SQL block:

declare

// declaration statements

begin

// execution statements

exception

when exception1 then
statement 1;

when exception2 then
statement 2;

[when others then]

default exception handling statement;

end;

Raise Statement

① IF you explicitly needs to raise an error then raise statement is used.

② you have to declare an exception variable in declare section.

2) raise_application_error

This statement takes two input parameters the error number and error message.

The error number must be between -20001 to -20999

(-) ममाचनस

Example.

** raise - application - error (-20009, 'error msg')

* Adding

declare

n int;

negative-number exception;

begin

n:=&n;

if n < 0 then

raise negative-number;

else

dbms-output.put-line(n);

end if;

exception

when negative-number then

dbms-output.put-line("The number is negative");

when others then

dbms-output.put-line('default error');

end;

Pre define exceptions:-

- ① No-data-found
- ② too-many-rows
- ③ zero-divide
- ④ storage-error
- ⑤ invalid-number
- ⑥ others
- ⑦ Program-error
- ⑧ invalid-cursor
- ⑨ cursor-already-open.