

Experiment No. 12

AIM: You can speed up PL/SQL procedures by compiling them into native code residing in shared libraries. The procedures are translated into C code, then compiled with your usual C compiler and linked into the Oracle Database process. Create a procedure “employee_details” which gives the details of the employee.

Theory:

Procedure: a stored procedure in PL/SQL is nothing but a series of declarative SQL statements which can be stored in the database catalogue. Procedure can be invoked through triggers, other procedures, or applications on Java, PHP etc.

Syntax:

Create or replace procedure procedure_name(parameters if any)

Is

Variable declaration(if any)

Begin

Procedure_body

End;

E.g 1 IN and OUT parameters

create or replace procedure big(x IN number,Y IN number,Z OUT number)

is

c number(4);

Begin

if x>y then

c:=x;

else

c:=y;

end if;

end;

Declare

a number(4);

b number(4);

c number(4);

Begin

a:=20;

b:=45;

Roll no. :1610991565

Name: Nidhi Mittal

```
big(a,b,c);
dbms_output.put_line(c);
end;

create or replace procedure big(x IN number,Y IN number,Z OUT number)
is
c number(4);
Begin
if x>y then
c:=x;
else
c:=y;
end if;
end;
```

Results Explain Describe Saved SQL History

Procedure created.

```
Declare
a number(4);
b number(4);
z number(4);
Begin
a:=20;
b:=45;
big(a,b,z);
dbms_output.put_line('Bigger number = '||z);
end;
```

Results Explain Describe Saved SQL History

Bigger number = 45

E.g 2 INOUT parameters

create or replace procedure square(x IN OUT number) is

Begin

x:=x*x;

end;

Declare

a number(4);

Begin

a:=5;

square(a);

dbms_output.put_line('Square = '||a);

end;

```
create or replace procedure square(x IN OUT number) is
Begin
x:=x*x;
end;
```

Results Explain Describe Saved SQL History

Procedure created.

```
Declare
a number(4);
Begin
a:=5;
square(a);
dbms_output.put_line('Square = '||a);
end;
```

Results Explain Describe Saved SQL His

Square = 25

Statement processed.

E.g 3

```
create or replace procedure p1(dno IN number)
is
cursor c1 is select ename,sal,deptno from emp where deptno=dno;
name emp.ename%type;
salary emp.sal%type;
dept emp.deptno%type;
Begin
open c1;
loop
fetch c1 into name,salary,dept;
exit when c1%notfound;
dbms_output.put_line(name || ' ' || salary || ' ' ||dept);
end loop;
close c1;
end;
Declare
Begin
pro(10);
```

end;

```
create or replace procedure pro(dno IN number)
is
cursor c1 is select ename,sal,deptno from emp where deptno=dno;
name emp.ename%type;
salary emp.sal%type;
dept emp.deptno%type;
Begin
open c1;
loop
fetch c1 into name,salary,dept;
exit when c1%notfound;
dbms_output.put_line(name || ' ' || salary || ' ' ||dept);
end loop;
close c1;
end;
```

```
Declare
a number(5);
Begin
a:=10;
pro(a);
end;
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

```
KING 16288.94 10
CLARK 3990.81 10
MILLER 2117.58 10
```

Statement processed.

Experiment No. 13

AIM: How to bundle related PL/SQL code and data into package. The package might include a set of procedures that form an API, or a pool of type definitions and variables declarations. The package is compiled and stored in the database, where its contents can be shared by many applications.

Theory:

Package: these are schema objects that group logically related PL/SQL types, variables, and subprograms. Package has two mandatory parts-

- 1) Package specification
- 2) package body

E.g package contains procedure to find sq of number and function to reverse a number

package specification create or replace package pack is

procedure square(x IN OUT number);

function reverse(a number)

return number;

end;

package body create or replace package body pack as

function reverse(a number) //function

return number

is

rev number(5):=0;

rem number(5):=0;

n number(5):=0;

Begin

n:=a;

while (n>0) loop

rem:=mod(n,10);

rev:=(rev*10)+rem;

n:=trunc(n/10);

end loop;

return rev;

end reverse;

procedure square(x IN OUT number) is //procedure

Begin

x:=x*x;

end square;

end

Roll no. :1610991565

Name: Nidhi Mittal

```
Declare // calling  
a number(5);  
b number(5);  
Begin  
a:=pack.reverse(123);  
b:=5;  
dbms_output.put_line('Reverse = '||a);  
pack.square(b);  
dbms_output.put_line('Square = '||b);  
end;  
  
create or replace package pack is  
    procedure square(x IN OUT number);  
    function reverse(a number)  
        return number;  
end;
```

Results Explain Describe Saved SQL History

Package created.

Roll no. :1610991565

Name: Nidhi Mittal

```
create or replace package body pack as
function reverse(a number)
return number
is
rev number(5):=0;
rem number(5):=0;
n number(5):=0;
Begin
n:=a;
while (n>0) loop
rem:=mod(n,10);
rev:=(rev*10)+rem;
n:=trunc(n/10);
end loop;
return rev;
end reverse;
```

```
procedure square(x IN OUT number) is
Begin
x:=x*x;
end square;
end;
```

Results	Explain	Describe	Saved SQL	History
----------------	---------	----------	-----------	---------

Package Body created.

```
Declare
a number(5);
b number(5);
Begin
a:=pack.reverse(123);
b:=5;
dbms_output.put_line('Reverse = '||a);
pack.square(b);
dbms_output.put_line('Square = '||b);
end;
```

Results	Explain	Describe	Saved SQL	History
----------------	---------	----------	-----------	---------

```
Reverse = 321
Square = 25
```

Statement processed.

Experiment No. 14

AIM: To create a PL/SQL block structure which is fired when DML statements like insert, delete, update is executed on a database table. A trigger automatically when an associated DML statement.

Theory:

TRIGGERS: - They are the stored programs that are automatically executed or fired when some event occurs. Triggers are written to be executed in response to DML, DDL or any database operation. A trigger can include SQL, PL/SQL statements to execute as a unit. Trigger is automatically executed without any actions required by user whereas stored procedure needs to be explicitly invoked.

Types of Triggers: -

Row level trigger: - Execute one for each row in a transaction. Commands of row level triggers are executed on all rows that are affected by the command.

Statement level trigger: - They are triggered only once for each transaction. These are default type of triggers created by create trigger command.

SYNTAX:-

Create or replace trigger trigger_name

[before/after/instead of]

[insert or /update or /delete]of coloumn_name

On table_name

Referencing New as N Old as O

For each row

When [condition]

Declare

Declaration_statement ;

Begin

Roll no. :1610991565

Name: Nidhi Mittal



Executable_statement ;

Exception

Exception_statement;

End;
