

## Unit - 2

### Managing Cursor

classmate

Date \_\_\_\_\_  
Page \_\_\_\_\_

#### Program - 1 :-

-- Implicit cursor and use  
of %rowcount attribute  
declare

n number;

begin

update emp1 set sal = sal + 500;

n := SQL%rowcount;

dbms\_output.put\_line('n' || n);

'row updated');

end;

/

#### Program - 2 :-

-- Implicit cursor program  
using %found and %notfound  
attribute

begin

delete From emp1 where eid =  
'e007';

IF SQL%found then

dbms\_output.put\_line('Record  
Found');

ELSIF SQL%notfound then

```

dbms_output.put_line('Record
not Found');  

END IF;  

end;
/

```

eid	ename	addr	sal
e01	A	Junagadh	50,000
e02	B	Mumbai	20,000
e03	C	Morbi	5,000
e04	D	Mumbai	10,000
e05	E	Baroda	16,000
e06	F	Mumbai	19,000
e07	G	Ahmedabad	10,000

### Program - 3 :-

-- Using implicit cursor update salary of employee by rs 500. who are staying in mumbai print appropriate message about updation of number of rows.

declare

  lno number;

begin

  update emp1 set sal = sal + 500

open cursor;

dbms\_output.put\_line('cannot

else

records:,'||size||row count);'

dbms\_output.put\_line('Total

END LOOP;

sum);'||d1||a1||b1||c1||

dbms\_output.put\_line('Line C'||1||

END IF;--add if record found

exit;

IF size not found Then

Fetch size into sum;

loop

IF size is open then

open size

decrease

use of size after putting

-- execute cursor with the

Program 5:

-----

end;

close c1;

END LOOP;

'||addr);'

dbms - output - line empid 11  
exit when %softfound?  
fetch cur into empid, add;  
loop  
open cur for insert operation  
begin  
from emp1 update by sad;  
cursor cur is select eid,addr  
add emp1.add%type;  
empid emp1.eid%type;  
declose

- explicit cursor demo

Program - 4 ~ DBMS  
END IF;  
updated;  
dbms - output - line C, Record  
ELSE SQL%RowCount;  
A := SQL%RowCount;  
updated;  
dbms - output - line C, Record  
IF SQL%found Then  
where Addr = 'Mumbai';  
dbms - output - line C, Record  
ELSIF SQL%found Then  
A := SQL%RowCount;  
updated;

for i in 1  
begin  
cursor c1 is select eid, addr  
from emp1 order by sal;  
dbms\_output.put\_line(c1%rowcount);  
loop

## Program - 7 ~

001	DBMS ALexis Leaon	MCGrawHill	600	ispa	Title Author	Publisher	Price
002	Tava K Swami Naithna	Person	900				
003	Sata Mitala Rabba	Naukreeet	550				
004	DS UdeSh Patel	Arihant	700				

Ex! + when C50% not found?  
where title = 'C Programming'  
update book set price = 100  
fetch cs into m-title, m-price;  
END LOOP;

loop

bag in

from book

choose for classmate select title price

m - publicasher book publisher % type

m - author book author % type

m - title book title % type

m - isbn book isbn % type

decimals

explicit classmate

- updating records with

book

Program = 6 loops

for i = 1 to 6 do

input book

Rank	Name	Age	Course	Classmate	Nisha	19	BBA
14	Chetan	20	BCA				
13	Arpit	18	BCA				
12	Asma	21	BCOM				
11	Arun	20	BSC				
10	Rao	Age	Course				

end;

close s1;

END IF;

END Loop;

end;

/

cursor for loop example

loop cursor for loop example

### Program - 8 :-

-- cursor For Loop Demo - 2

declare

cursor s1 is select rno, name

From estudt where course='BCA';

begin

dbms\_output.put\_line(c.i.rno||

'||i.name);

END Loop;

end;

/

cursor for loop example

loop cursor for loop example

### Program - 9 :-

-- explicit cursor example

with user define valuedb

type

declare

m\_isbn book.isbn%type; i

m\_title book.title%type;

m\_author book.author%type;

m\_publisher book.publisher%type;

m\_price book.price%type;

cursor c5 is select isbn, title,

author, publisher, price from

book;

begin

if publisher = '&publisher' then

open c5;

loop

Fetch c5 into m\_isbn, m\_title,

m\_author, m\_publisher, m\_price;

Exit when c5%notfound;

IF m\_publisher = c\_publisher then

dbms\_output.put\_line(m\_isbn||

m\_title||m\_author||m\_publisher||

m\_price);

END IF;

end loop;

close c5;

end;

/

exit;

end if;

end if;

end if;

## Program - 10 ~

-- insertion of record from one table to another table using explicitly cursor

```
declare
    m_addr emp1.addr%type;
    m_eid emp1.eid%type;
    m_ename emp1.ename%type;
    m_sal emp1.sal%type;
cursor c3 is select eid, ename,
    addr, sal from emp1 order by
    sal desc;
begin
    open c3;
    for lp in 1..5
    loop
        fetch c3 into m_eid, m_ename,
            m_addr, m_sal;
        insert into top5.sal values
            (m_eid, m_ename, m_addr, m_sal);
        exit when c3%notfound;
    end loop;
    if SQL%found then
        dbms_output.put_line('Record
            inserted');
    else
```

```
dbms_output.put_line('not
inserted');  
END IF;  
close(c3);  
end;
```

/

### Program - 11 :-

-- Example of Parameterize cursor

declare

e1 emp1%rowtype;

dno emp1.deptno%type;

salary emp1.Sal%type;

cursor (c1(dno number)) IS Select

\* from emp1 where deptno=dno;

cursor (c2(dno number)) IS Select

deptno, avg(sal) From emp1

group by deptno having deptno  
= dno;

begin

-- Show Department wise salary  
of all employee

```
dbms_output.put_line('Department
wise employee salary');
```

```
open (c1(20));
```

Loop until c1 not found

Fetch c1 into ei; begin;

Exit when c1%notfound;

dbms\_output.put\_line('e1.deptno  
|| ':'|| e1.sal);

END Loop;

close c1;

-- Show Department wise average

Salary

dbms\_output.put\_line('Department  
wise average salary'); --

open c2(c20);

Loop

Fetch c2 into dno, salary;

Exit when c2%notfound;

dbms\_output.put\_line('dno ||

'|| salary);

END Loop;

close c2; close pvt notab;

end; end antab pd option;

/

Unit-3Error HandlingProgram-1

-- No Data found exception and  
Invalid\_number exception  
declare

salary number;

begin

select sal into salary From emp1  
where eid = 100;

dbms\_output.put\_line('The  
Salary is : '||salary);

Exception

when no\_data\_found Then

dbms\_output.put\_line('No  
Records are Found for empno 100');

when invalid\_number Then

dbms\_output.put\_line('Invalid  
value');

end;

/

## Program - 2

```
-- No_Data_Found exception
declare
    id emp1.eid%type;
    name emp1.ename%type;
begin
    select ename into name from
        emp1 where eid = '&id';
    dbms_output.put_line('The
        name is : '||name);
exception
    when no_data_found then
        dbms_output.put_line('No
            records are Found');
end;
```

## Program - 3 :-

-- too many records

declare

salary emp1.sal%type;

dno emp1.deptno%type;

begin

select sal into salary from emp1

where deptno = &dno;

dbms\_output.put\_line(

'salary is : || salary);

exception

when too\_many\_rows then

dbms\_output.put\_line('more

than one rows are selected

for department.');

when no\_data\_found then

dbms\_output.put\_line('No

records are Found.');

end;

/

## • Program - 4 ~

```
-- invalid number exception
declare
    no number(5);
begin
    select to_number('0001') into
    no from dual;
    dbms_output.put_line(cno);
exception
    when invalid_number then
        dbms_output.put_line('wrong
data type / data conversion');
end;
```

## Program - 5

-- write a program that explains the use of zero divide exception.

declare

n number(3);

begin

select 100/0 into n From Dual;

dbms\_output.put\_line(n);

exception

when zero\_divide then

dbms\_output.put\_line('you can not divide the number by 0');

end;

/

## Program - 6%

-- others exception (unnamed  
System exception)

begin

dbms\_output.put\_line('To\_'  
DATE('2010 10 10 44:55:66',  
'yyys'));

exception

when others then  
dbms\_output.put\_line('Bad  
data Format')||SQLERRM);

end;

## Program - 7

```
-- cursor related exceptions
declare
  m_ename emp1.ename%type;
  m_sal emp1.sal%type;
  cursor c3 is select ename,
  sal from emp1;
begin
  open c3;
  loop
    fetch c3 into m_ename, m_sal;
    dbms_output.put_line(m_ename
    || ' ' || m_sal);
  close c3;
  exit when c3%notfound;
end loop;
close c3;
exception
  when cursor_ALREADY_OPEN then
    dbms_output.put_line('cursor
  already open');
  when INVALID_CURSOR then
    dbms_output.put_line('no such
  cursor');
end;
```

## Program - 8 :-

-- duplicate value exception

declare

eno number := 125;

begin

insert into emp(ceid) values

(eno);

dbms\_output.put\_line('

Insertion successful');

exception

when dup\_val\_on\_index then

dbms\_output.put\_line('Error:11  
SQLERRM);

end;

## Program - 9

-- case not found or others exception.

```
declare
v_value number := &v_value;
begin
case v_value
when 1 then
dbms_output.put_line('Branch 1');
when 2 then
dbms_output.put_line('Branch 2');
when 3 then
dbms_output.put_line('Branch 3');
end case;
```

Exception

```
when case_not_found then
dbms_output.put_line('No
Branch matches the value.');
when others then
dbms_output.put_line('An
Error' || SQLERRM);
end;
```

/

## Program - 10

-- user defined exception

declare

no number := &no;

negative\_val Exception;

begin

If no<0 Then

Raise negative\_val;

End If;

dbms\_output.put\_line('

value is '||no||);

Exception

when negative\_val Then

dbms\_output.put\_line('

negative values not allowed');

end;

/

## Program - 11 :-

-- write a program to using implicit cursor to display the commission of given Empno, also raise a user defined exception null\_commission when no value (null) is available for commission

declare

vcommission emp1.comm%otype;  
null\_comission Exception;

begin

select comm Into vcommission  
From emp1 when eid = '&eid';  
dbms\_output.put\_line('The  
commission value is :- '||  
vcommission);

IF vcommission is null then  
Raise null\_comission

End IF;

Exception when null\_comission Then  
dbms\_output.put\_line('No  
comission found for given  
Employee');  
end;

1

## Program - 12 :-

-- write a PL/SQL block to display the salary of that employee whose age is 45 year otherwise display appropriate message using exception handling (with system exception).

declare

age emp1.age%type;

salary emp1.sal%type;

begin

select age, sal into age,  
salary from emp1 where

age = 45;

dbms\_output.put\_line (age

|| salary);

exception

when no\_data\_found then

dbms\_output.put\_line ('no  
data found');

when too\_many\_rows then

dbms\_output.put\_line ('  
more number of records found');

end;

/

## Program - 13 :-

-- write a PL/SQL block using a cursor to insert the records of employee in emp\_backup table for given department number, also raise a user defined exception.

declare

```
I number(3);  
veno emp1.eid%type;  
vename emp1.ename%type;  
vsal emp1.sal%type;  
vdno emp1.deptno%type;  
cursor cur_emp is select  
eid,ename,sal from emp1  
where deptno = &deptno;  
no_dept_found Exception;
```

begin

```
I := 0;
```

```
Open cur_emp;
```

Loop

Fetch cur\_emp into veno,  
vename, vsal;

Exit when cur\_emp%not  
found;

```
I := I + 1;
```

```
dbms_output.put_line(vno
||' '||vename||' '||vsal);
Insert into emp_backup values
(vno, vename, vsal);
End Loop;
IF cur_emp%rowcount = 0 then
close cur_emp;
Raise no_dept_found;
End IF;
dbms_output.put_line('Total
no. of Records into emp_backup
Table are :-'||I);
Else cur_emp; exception
when no_dept_found then
dbms_output.put_lines('No
Records are found for Given
Dept no');
when others then
dbms_output.put_line('PL/
SQL code Encountered an
Error');
end;
/
```

## Program - 14 8~

-- user defined exception to check employee's salary or below range

declare

below\_salary\_range Exception;

above\_salary\_range Exception;

n\_salary emp1.sal%o type;

n\_emp\_id emp1.eid%o type;

begin

Select sal into n\_salary from emp1 where eid = 'In.emp\_id';

IF n\_salary < 5000 then

raise below\_salary\_range;

Elsif n\_salary > 5000 then

raise above\_salary\_range;

End IF;

dbms\_output.put\_line ('

employee' || n\_emp\_id || ' has  
salary Rs.' || n\_salary);

exception

when below\_salary\_range  
then

dbms\_output.put\_line ('

Employee' || n\_emp\_id || ' has  
salary below the salary

range');  
when above\_salary\_range  
then  
dbms\_output.put\_line('Employee'||n.emp\_id||' has  
salary below the salary  
range');  
when no\_data\_found then  
dbms\_output.put\_line('employee'||n.emp\_id||' not  
found');  
end;

## Program - 15 :-

-- user defined exceptions while  
entering data into the table  
declare

eid emp1.eid%type := '&eid';

ename emp1.ename%type := '&ename';

addr emp1.addr%type := '&addr';

sal emp1.sal%type := &sal;

dno emp1.deptno%type := &dno;

comm emp1.deptno%type :=  
&comm;

age emp1.age%type := &age;

invalid - value Exception;

begin

IF (sal > 100000) Then

Raise invalid\_value;

Elsif (comm > sal) Then

Raise invalid\_value;

Elsif (age < 18) Then

Raise invalid\_value;

End IF;

insert into emp1 value (eid,  
ename, addr, sal, dno, comm,  
age);

IF SQL%FOUND THEN  
DBMS\_OUTPUT.PUT\_LINE('Record  
Inserted');  
END IF;  
exception  
when invalid\_value THEN  
DBMS\_OUTPUT.PUT\_LINE('Enter  
data properly'||SQLERRM);  
end;

## Program - 16 :-

-- write a program that explains the use of zero\_divide exception with Raise\_application error procedure

declare

n number(3);

vd number(3);

begin

vd := &vd;

IF vd = 0 Then

Raise\_application\_error(-20230, 'you can not divide any number with zero');

Else

select 100/vd into n from dual;

dbms\_output.put\_line(n);

End IF;

end;

/