

Assignment 6

NAME : Bhavik Ransubhe

CLASS: TE (B) COMP

ROLL NO : 39055

Creating tables and inserting values:

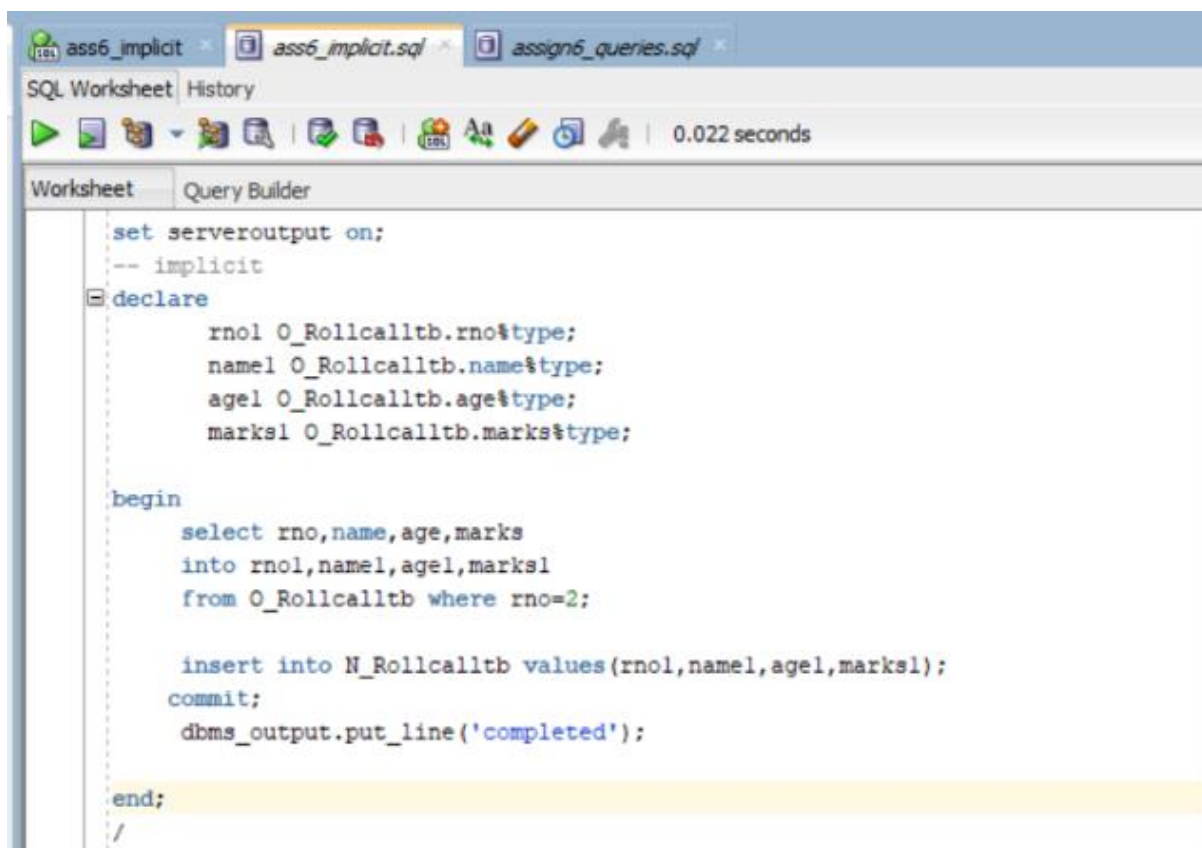
The screenshot shows a database query builder interface with two tabs: 'Worksheet' and 'Query Builder'. The 'Query Builder' tab is active, displaying the following SQL code:

```
create table O_Rollcalltb(  
  rno integer not null primary key,  
  name varchar2(20),  
  age integer,  
  marks integer);  
  
describe O_Rollcalltb;  
  
create table N_Rollcalltb(  
  rno integer not null primary key,  
  name varchar2(20),  
  age integer,  
  marks integer);  
  
describe N_Rollcalltb;  
  
INSERT INTO O_Rollcalltb values(1, 'Bhavik', 20, 99);  
INSERT INTO O_Rollcalltb values(2, 'Gaurav', 20, 98);  
INSERT INTO O_Rollcalltb values(3, 'Rithvik', 20, 94);  
INSERT INTO O_Rollcalltb values(4, 'Advait', 20, 97);  
INSERT INTO O_Rollcalltb values(5, 'Adesh', 20, 96);  
commit;  
select * from O_Rollcalltb;
```

Below the code editor, there are two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, showing the results of the SQL query. The results are displayed in a table with the following columns: RNO, NAME, AGE, and MARKS. The table contains 5 rows of data.

RNO	NAME	AGE	MARKS
1	1 Bhavik	20	99
2	2 Gaurav	20	98
3	3 Rithvik	20	94
4	4 Advait	20	97
5	5 Adesh	20	96

1.IMPLICIT CURSOR



The screenshot shows a SQL Worksheet interface with three tabs: 'ass6_implicit', 'ass6_implicit.sql', and 'assign6_queries.sql'. The 'ass6_implicit.sql' tab is active, displaying a PL/SQL block. The block starts with 'set serveroutput on;' and a comment '-- implicit'. It then declares four variables: 'rno1', 'name1', 'age1', and 'marks1', each assigned a value from the 'O_Rollcalltb' table. The 'begin' block contains a 'select' statement that inserts data into 'N_Rollcalltb' based on the values of the declared variables. The block ends with 'end;' and a forward slash '/'. The execution time is shown as 0.022 seconds.

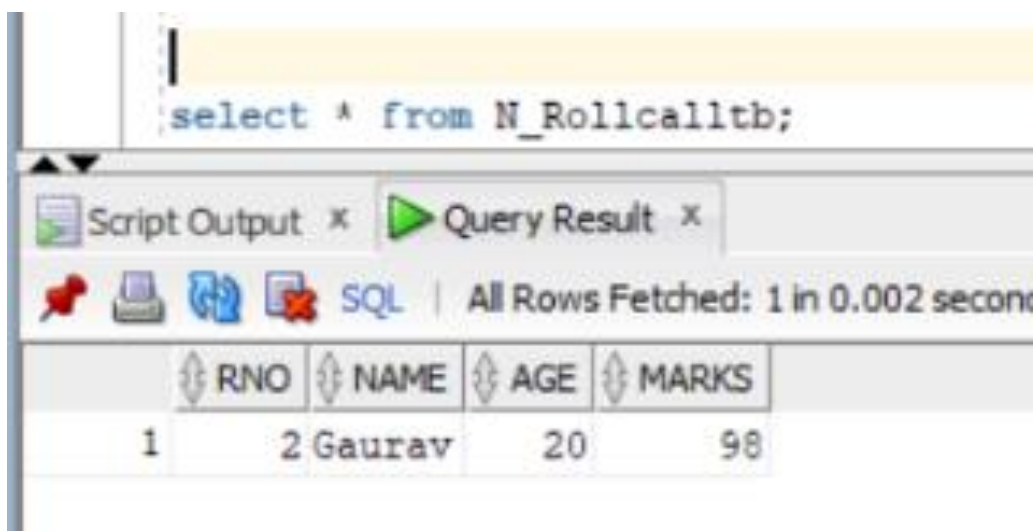
```
set serveroutput on;
-- implicit
declare
    rno1 O_Rollcalltb.rno%type;
    name1 O_Rollcalltb.name%type;
    age1 O_Rollcalltb.age%type;
    marks1 O_Rollcalltb.marks%type;

begin
    select rno,name,age,marks
    into rno1,name1,age1,marks1
    from O_Rollcalltb where rno=2;

    insert into N_Rollcalltb values(rno1,name1,age1,marks1);
    commit;
    dbms_output.put_line('completed');

end;
/
```

OUTPUT:



The screenshot shows the 'Query Result' window of the SQL Worksheet. It displays the results of the query 'select * from N_Rollcalltb;'. The results are shown in a table with four columns: 'RNO', 'NAME', 'AGE', and 'MARKS'. There is one row of data with the values 1, 2 Gaurav, 20, and 98.

RNO	NAME	AGE	MARKS
1	2 Gaurav	20	98

2.EXPLICIT CURSOR:

The screenshot displays the SQL Developer interface. The top pane shows a PL/SQL block with an explicit cursor. The code declares variables for rno, name, age, and marks, then opens a cursor named 'expli_cur' to select from 'O_Rollicalltb'. It enters a loop to fetch each row, insert it into 'N_Rollicalltb', and output the message 'inserted record'.

```
set serveroutput on;
--explicit
declare
    rno1 O_Rollicalltb.rno%type;
    name1 O_Rollicalltb.name%type;
    age1 O_Rollicalltb.age%type;
    marks1 O_Rollicalltb.marks%type;

    Cursor expli_cur is
        select rno,name,age,marks from O_Rollicalltb;

begin
    open expli_cur;
    loop
        fetch expli_cur into rno1,name1,age1, marks1;

        exit when expli_cur%notfound;

        insert into N_Rollicalltb values(rno1,name1,age1,marks1);
        commit;

    end loop;
    dbms_output.put_line('inserted record');
    close expli_cur;

end;
```

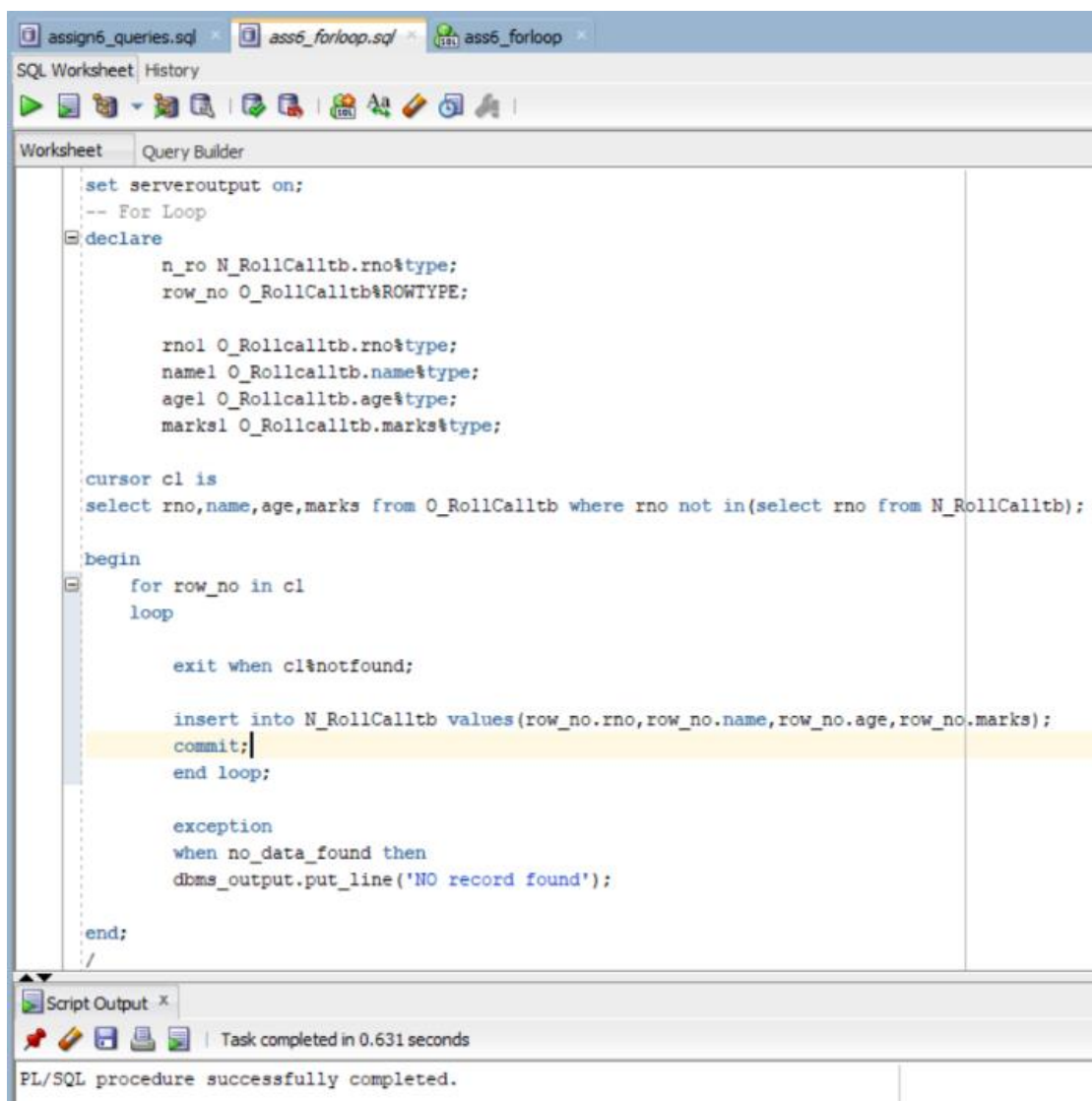
The bottom pane shows the 'Script Output' window with the message: 'Task completed in 0.161 seconds' and 'PL/SQL procedure successfully completed.' followed by the output 'inserted record'.

OUTPUT:

The screenshot shows the 'Query Result' window. The query 'select * from N_Rollicalltb;' has been executed, and the results are displayed in a table with 5 rows. The columns are RNO, NAME, AGE, and MARKS.

RNO	NAME	AGE	MARKS
1	1 Bhavik	20	99
2	2 Gaurav	20	98
3	3 Rithvik	20	94
4	4 Advait	20	97
5	5 Adesh	20	96

3.FOR LOOP :



```
set serveroutput on;
-- For Loop
declare
    n_no N_RollCalltb.rno%type;
    row_no O_RollCalltb%ROWTYPE;

    rno1 O_Rollcalltb.rno%type;
    name1 O_Rollcalltb.name%type;
    age1 O_Rollcalltb.age%type;
    marks1 O_Rollcalltb.marks%type;

    cursor c1 is
    select rno,name,age,marks from O_RollCalltb where rno not in(select rno from N_RollCalltb);

begin
    for row_no in c1
    loop

        exit when c1%notfound;

        insert into N_RollCalltb values(row_no.rno,row_no.name,row_no.age,row_no.marks);
        commit;
    end loop;

    exception
    when no_data_found then
        dbms_output.put_line('NO record found');

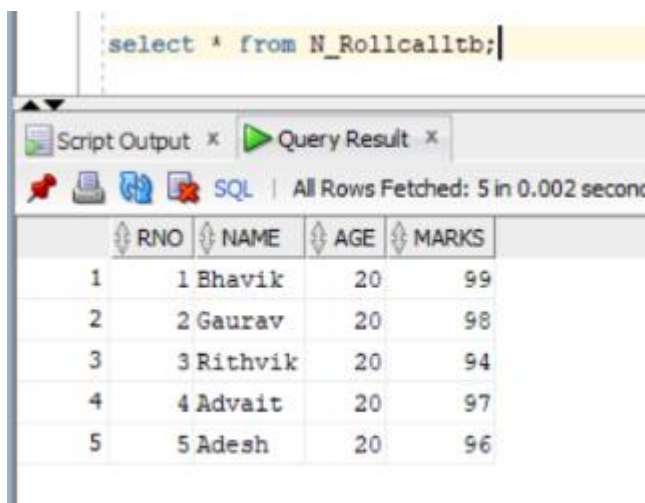
end;
/
```

Script Output x

Task completed in 0.631 seconds

PL/SQL procedure successfully completed.

OUTPUT:



```
select * from N_Rollcalltb;
```

	RNO	NAME	AGE	MARKS
1	1	Bhavik	20	99
2	2	Gaurav	20	98
3	3	Rithvik	20	94
4	4	Advait	20	97
5	5	Adesh	20	96

4:PARAMETRIZED CURSOR:

The screenshot shows the SQL Developer interface with two tabs: 'assign6_queries.sql' and 'ass6_parametrized.sql'. The 'Query Builder' tab is active, displaying a PL/SQL procedure. The procedure sets serveroutput on, declares a cursor 'c1' that selects from 'O_RollCalltb' based on a parameter 'r', and then loops through the cursor to insert data into 'N_RollCalltb'. An exception is handled for 'no_data_found' by outputting 'NO record found'. The procedure ends with 'end;' and a slash. The 'Script Output' window at the bottom shows the execution results, indicating that the PL/SQL procedure was successfully completed.

```
set serveroutput on;
-- parametrized

declare
cursor c1(r int) is
select rno,name,age,marks from O_RollCalltb where rno=r and rno not in(select rno from N_RollCalltb);

begin
  for row_no in c1(&r)
  loop

    exit when c1%notfound;

    insert into N_RollCalltb values(row_no.rno,row_no.name,row_no.age,row_no.marks);

  end loop;

  exception
  when no_data_found then
    dbms_output.put_line('NO record found');

end;
/
```

Script Output x

Task completed in 0.04 seconds

```
exception
when no_data_found then
dbms_output.put_line('NO record found');

end;

PL/SQL procedure successfully completed.
```

OUTPUT:

The screenshot shows a dialog box titled 'Enter Substitution Variable'. It prompts the user to 'Enter value for r:' and has a text input field containing the value '1'. There are 'OK' and 'Cancel' buttons at the bottom.

The screenshot shows the SQL Developer interface with the 'Query Result' tab active. It displays the SQL query 'select * from N_Rollcalltb;' and the results of the query. The results are shown in a table with columns 'RNO', 'NAME', 'AGE', and 'MARKS'. The first row contains the values '1', '1 Bhavik', '20', and '99'.

RNO	NAME	AGE	MARKS
1	1 Bhavik	20	99