

Assignment -7

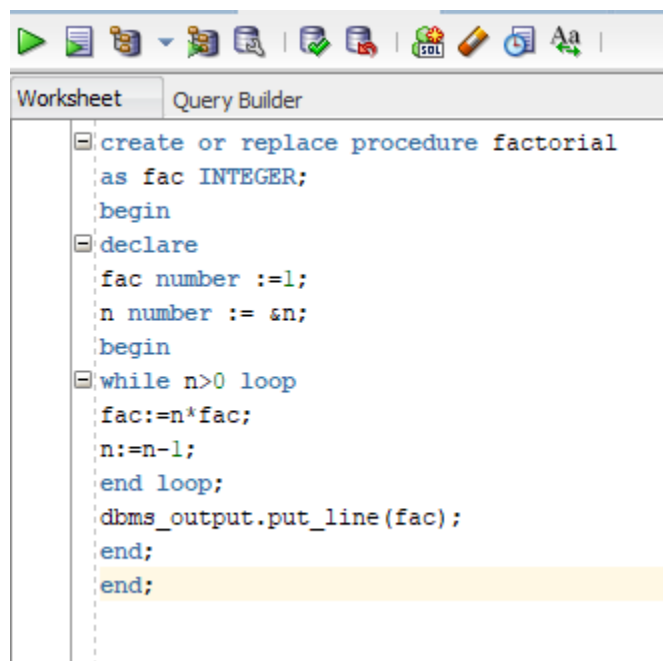
PL/SQL

Name-Ashish Goyal

Id- 2016ucp1100

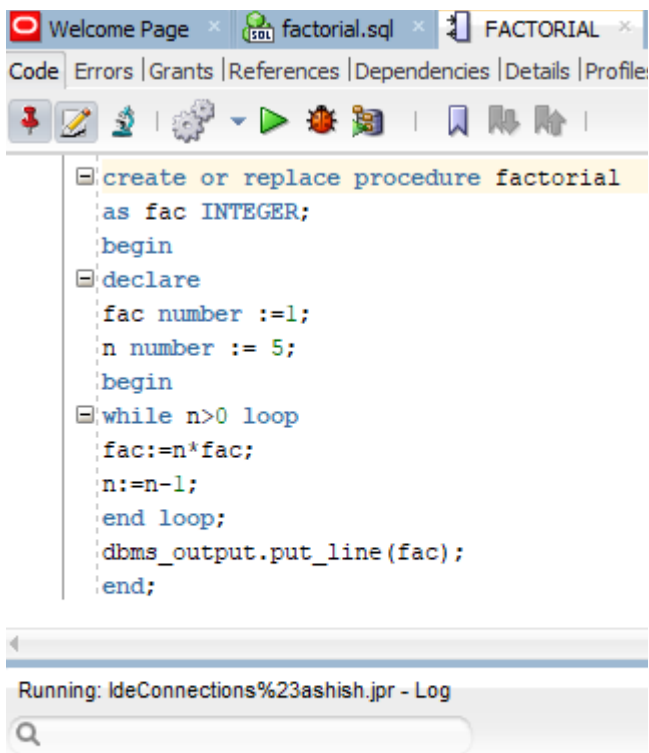
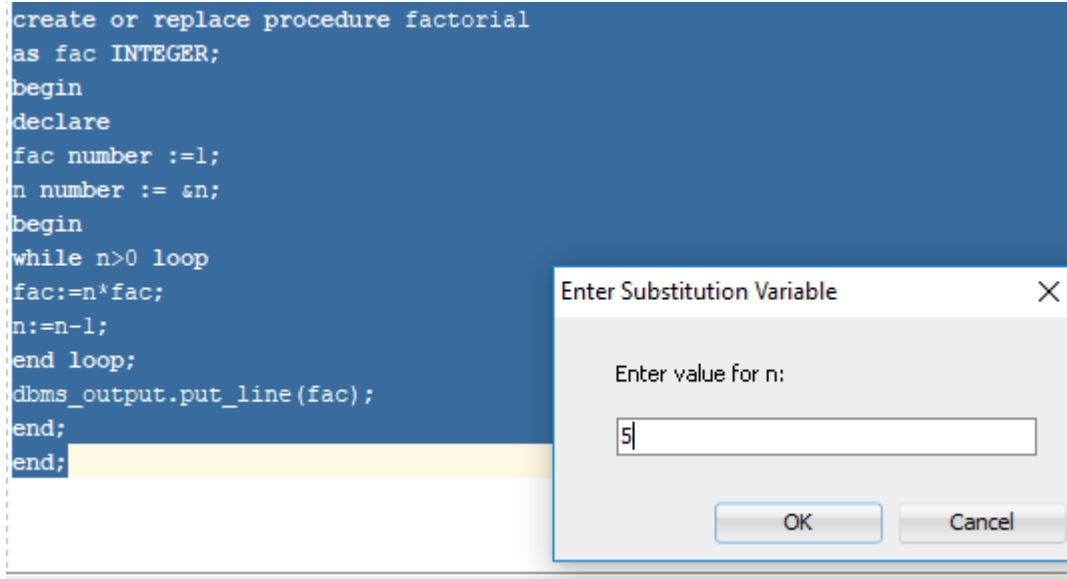
Batch-A(1,2)

1-Create a procedure to find factorial of number provided by user.



The screenshot shows a software interface with a toolbar at the top containing icons for execution, saving, and editing. Below the toolbar is a tabbed interface with 'Worksheet' and 'Query Builder' tabs. The 'Query Builder' tab is active, displaying a PL/SQL procedure in a text editor. The code is as follows:

```
create or replace procedure factorial
as fac INTEGER;
begin
declare
fac number :=1;
n number := &n;
begin
while n>0 loop
fac:=n*fac;
n:=n-1;
end loop;
dbms_output.put_line(fac);
end;
end;
```



2-Procedure for reversing a number provided by user.

```
Worksheet | Query Builder
--
create or replace procedure rnum
as revnum varchar2(10);
begin
declare
n varchar2(10) := ' ' || &n;
len number;
revnum varchar2(10);
begin
len := length(n);
for i in reverse 1.. len
loop
revnum := revnum || substr(n,i,1);
end loop;
dbms_output.put_line('given number =' || n);
dbms_output.put_line('reverse number =' || revnum);
end;
end;
```

Welcome Page | reverse.sql | factorial.sql | RNUM | EMPLOYEE

SQL Worksheet | History

Worksheet | Query Builder

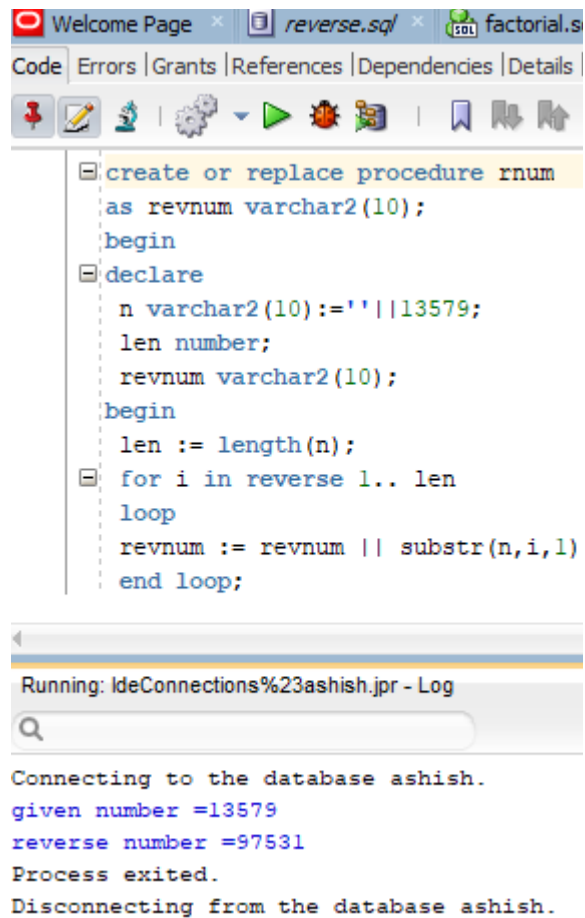
```
--
create or replace procedure rnum
as revnum varchar2(10);
begin
declare
n varchar2(10) := ' ' || &n;
len number;
revnum varchar2(10);
begin
len := length(n);
for i in reverse 1.. len
loop
revnum := revnum || substr(n,i,1);
end loop;
dbms_output.put_line('given number =' || n);
dbms_output.put_line('reverse number =' || revnum);
end;
end;
```

Enter Substitution Variable

Enter value for n:

13579

OK Cancel



The screenshot shows an IDE window with a tab for 'reverse.sql'. The code editor contains a PL/SQL procedure named 'rnum' that takes a string 'n' and returns its reverse. The procedure uses a loop to build the reversed string character by character. Below the code editor is a console window titled 'Running: IdeConnections%23ashish.jpr - Log' which shows the execution output.

```
create or replace procedure rnum
as revnum varchar2(10);
begin
declare
n varchar2(10):='13579';
len number;
revnum varchar2(10);
begin
len := length(n);
for i in reverse 1.. len
loop
revnum := revnum || substr(n,i,1)
end loop;
```

Running: IdeConnections%23ashish.jpr - Log

Connecting to the database ashish.
given number =13579
reverse number =97531
Process exited.
Disconnecting from the database ashish.

Employee(Eno,ename,dno,salary,job)

Worksheet	Query Builder
<pre> create table Employee (Eno int, ename varchar(20), dno int, salary int, job varchar(20)); insert into Employee values(13,'Ashu',3,10000,'tester'); insert into Employee values(11,'Goyal',1,20000,'coder'); insert into Employee values(12,'shambhu',2,15000,'maintainer'); insert into Employee values(10,'Ashish',2,27000,'coder'); insert into Employee values(9,'Abhishek',1,17000,'tester'); insert into Employee values(8,'swaraj',3,17000,'clerk'); insert into Employee values(6,'Utsav',7,28000,'manager'); insert into Employee values(1,'Chinmey',1,21000,'tester'); </pre>	

	ENO	ENAME	DNO	SALARY	JOB
1	13	Ashu	3	10000	tester
2	11	Goyal	1	20000	coder
3	12	shambhu	2	15000	maintainer
4	10	Ashish	2	27000	coder
5	9	Abhishek	1	17000	tester
6	8	swaraj	3	17000	clerk
7	6	Utsav	7	28000	manager
8	1	Chinmey	1	21000	tester

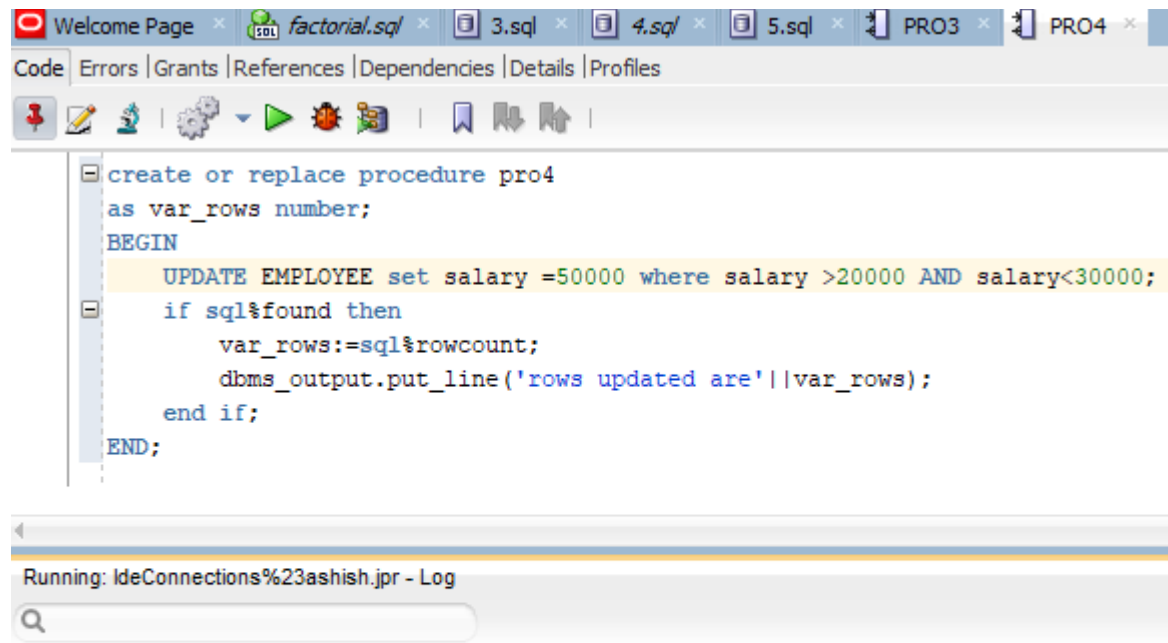
3-Delete tuples for employee where eid is greater than 10 and display number of rows affected.

```
create or replace procedure pro3
as var_rows number;
BEGIN
    delete from EMPLOYEE where eno > 10;
    if sql%found then
        var_rows:=sql%rowcount;
        dbms_output.put_line('rows deleted are'||var_rows);
    end if;
END;
```

Running: IdeConnections%23ashish.jpr - Log

Connecting to the database ashish.
rows deleted are3
Process exited.
Disconnecting from the database ashish.

4-Update record in table set salary =50000 where salary is greater than 20000 and less than 30000 And display number of rows affected.



The screenshot shows the Oracle SQL Developer IDE. The top toolbar includes icons for running, debugging, and other development tasks. The main editor displays a PL/SQL procedure named 'pro4' that updates employee salaries and counts the rows updated. The procedure is as follows:

```
create or replace procedure pro4
as var_rows number;
BEGIN
    UPDATE EMPLOYEE set salary =50000 where salary >20000 AND salary<30000;
    if sql%found then
        var_rows:=sql%rowcount;
        dbms_output.put_line('rows updated are'||var_rows);
    end if;
END;
```

Below the editor, the 'Log' window shows the execution output:

```
Running: IdeConnections%23ashish.jpr - Log
Connecting to the database ashish.
rows updated are3
Process exited.
Disconnecting from the database ashish.
```

5-Display the employee number and number of rows selected having salary equal to 17000.

```

create or replace procedure pro5
as var_rows number;
BEGIN
    declare
        rnumber number:=0;
        CURSOR curse is
            Select eno from employee where salary=17000;
        begin
            for i in curse loop
                dbms_output.put_line('e_number '||i.eno);
                rnumber:=rnumber+1;
            end loop;
            dbms_output.put_line('rows selected are-'||rnumber);
        end;
    END;

```

Running: IdeConnections%23ashish.jpr - Log

Connecting to the database ashish.
 e_number 9
 e_number 8
 rows selected are-2
 Process exited.
 Disconnecting from the database ashish.