

# Assignment-10

## PL/sql-3

Name-Ashish Goyal

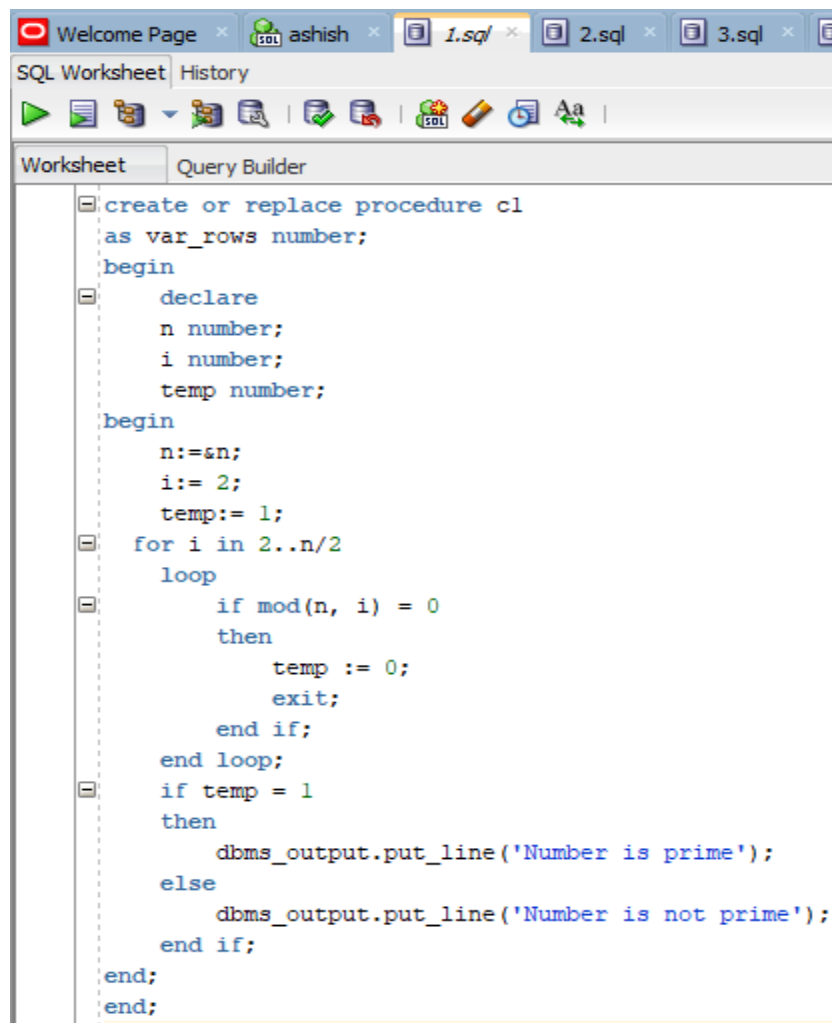
Id-2016ucp1100

Batch-A (1, 2)

---

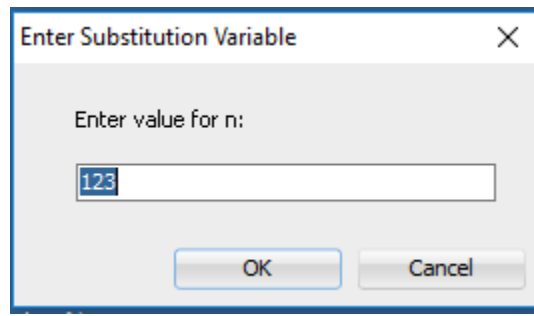
1- Write procedure to check whether a number is prime or not.

Procedure-



```
create or replace procedure c1
as var_rows number;
begin
  declare
    n number;
    i number;
    temp number;
  begin
    n:=&n;
    i:= 2;
    temp:= 1;
    for i in 2..n/2
      loop
        if mod(n, i) = 0
          then
            temp := 0;
            exit;
          end if;
        end loop;
    if temp = 1
      then
        dbms_output.put_line('Number is prime');
      else
        dbms_output.put_line('Number is not prime');
      end if;
    end;
  end;
```

## Input-



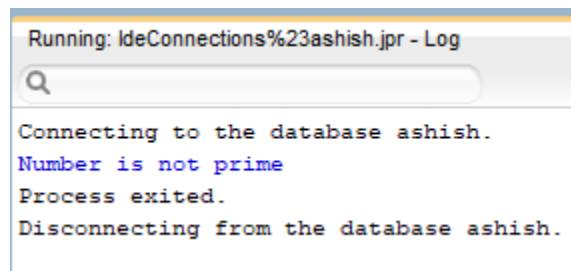
Enter Substitution Variable

Enter value for n:

123

OK Cancel

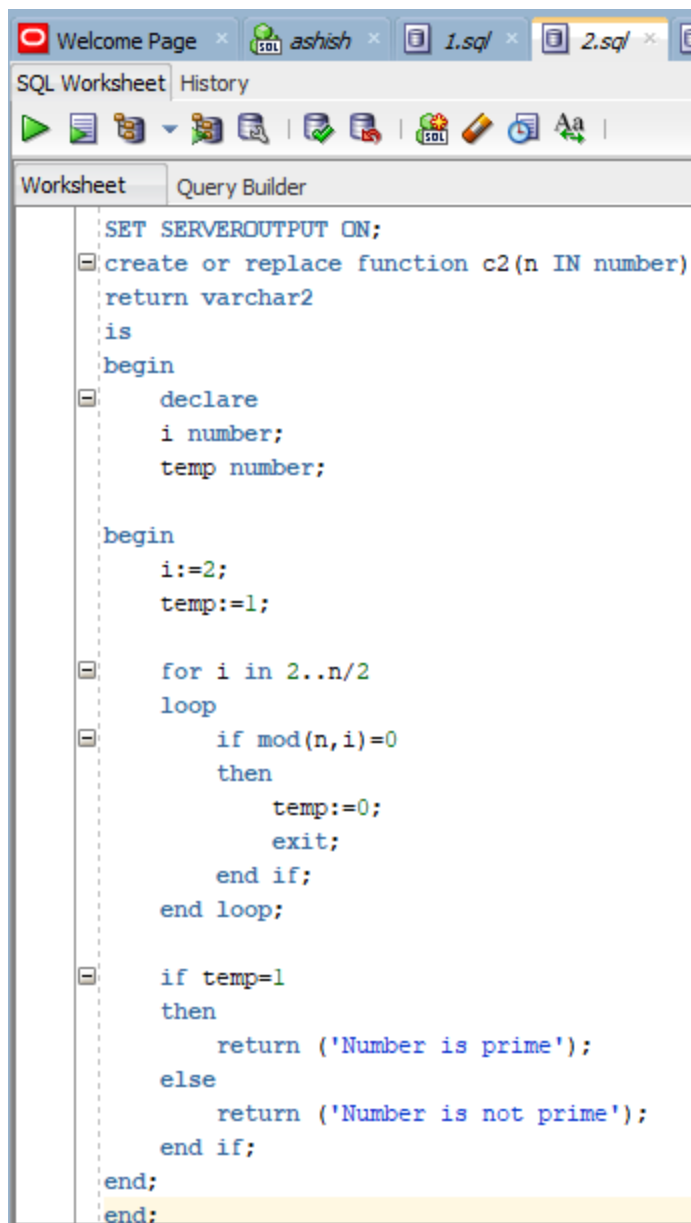
## Result-



```
Running: IdeConnections%23ashish.jpr - Log
Connecting to the database ashish.
Number is not prime
Process exited.
Disconnecting from the database ashish.
```

2- Write a function to check whether a number is prime or not.

## Function-



The screenshot shows an SQL Worksheet window with a title bar containing 'Welcome Page', 'ashish', '1.sql', and '2.sql'. The window has a menu bar with 'SQL Worksheet' and 'History', and a toolbar with various icons. Below the toolbar are two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying a PL/SQL function definition for a prime number checker. The code is as follows:

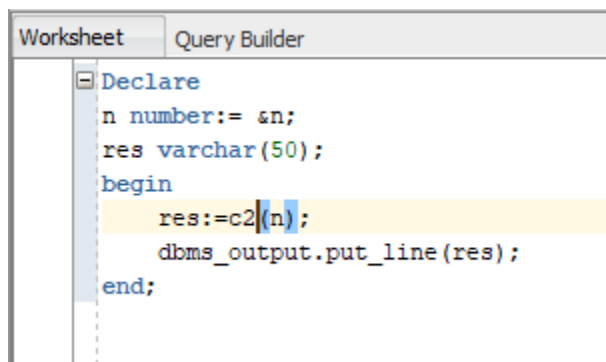
```
SET SERVEROUTPUT ON;
create or replace function c2(n IN number)
return varchar2
is
begin
    declare
        i number;
        temp number;

    begin
        i:=2;
        temp:=1;

        for i in 2..n/2
        loop
            if mod(n,i)=0
            then
                temp:=0;
                exit;
            end if;
        end loop;

        if temp=1
        then
            return ('Number is prime');
        else
            return ('Number is not prime');
        end if;
    end;
end;
```

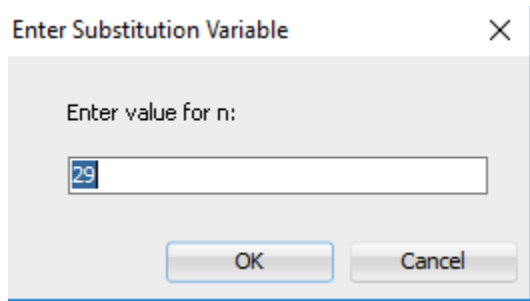
## Function Call-



The screenshot shows an SQL Worksheet window with a title bar containing 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying a PL/SQL block that calls the function 'c2' defined in the previous screenshot. The code is as follows:

```
Declare
n number:= &n;
res varchar(50);
begin
    res:=c2(n);
    dbms_output.put_line(res);
end;
```

## Input-



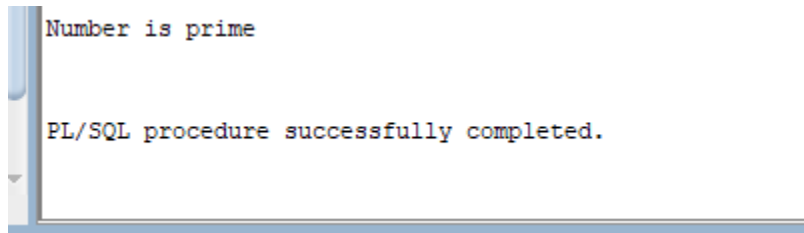
Enter Substitution Variable

Enter value for n:

29

OK Cancel

## Result-

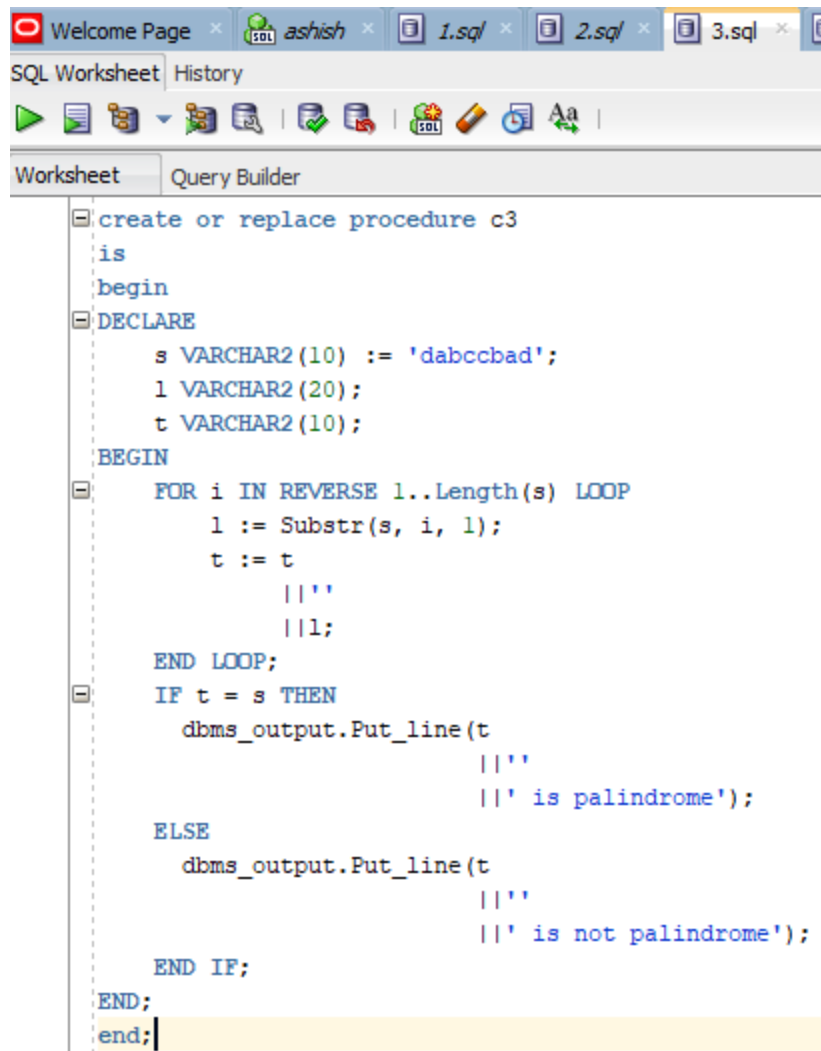


```
Number is prime

PL/SQL procedure successfully completed.
```

**3- Write procedure to check whether a string is palindrome or not.**

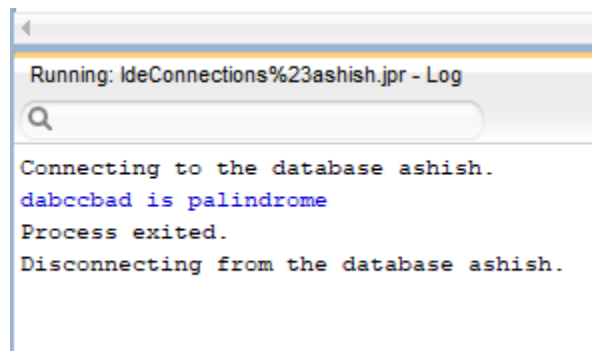
## Procedure



The screenshot shows an SQL Worksheet window with multiple tabs: 'Welcome Page', 'ashish', '1.sql', '2.sql', and '3.sql'. The 'ashish' tab is active. Below the tabs is a toolbar with various icons for running, saving, and editing. The main area is divided into 'Worksheet' and 'Query Builder' tabs, with 'Worksheet' selected. The code in the worksheet is as follows:

```
create or replace procedure c3
is
begin
DECLARE
    s VARCHAR2(10) := 'dabccbad';
    l VARCHAR2(20);
    t VARCHAR2(10);
BEGIN
    FOR i IN REVERSE 1..Length(s) LOOP
        l := Substr(s, i, 1);
        t := t || l;
    END LOOP;
    IF t = s THEN
        dbms_output.Put_line(t || ' is palindrome');
    ELSE
        dbms_output.Put_line(t || ' is not palindrome');
    END IF;
END;
end;
```

## Result-



The screenshot shows a log window titled 'Running: IdeConnections%23ashish.jpr - Log'. It contains the following text:

```
Connecting to the database ashish.
dabccbad is palindrome
Process exited.
Disconnecting from the database ashish.
```

4- Write function to check whether string is palindrome or not.

## Function-

The screenshot shows an SQL Worksheet application with a menu bar (Welcome Page, ashish, 1.sql, 2.sql, 3.sql, 4.s) and a toolbar. The 'Worksheet' tab is active, displaying a PL/SQL function definition. The function, named c4, takes a string s as input and returns a varchar2. It uses a loop to reverse the string and an if statement to check if the original string is a palindrome. The code is as follows:

```
create or replace function c4(s in varchar2)
return varchar2
is
begin
DECLARE
  l VARCHAR2(20);
  t VARCHAR2(10);
BEGIN
  FOR i IN REVERSE 1..Length(s) LOOP
    l := Substr(s, i, 1);
    t := t
        || l;
  END LOOP;
  IF t = s THEN
    return(t
        || ' is palindrome');
  ELSE
    return(t
        || ' is not palindrome');
  END IF;
END;
end;
```

## Function Call-

The screenshot shows the same SQL Worksheet application with the 'Worksheet' tab active. It displays a PL/SQL block that declares a variable s with the value 'abcd', calls the function c4, and outputs the result using dbms\_output.put\_line. The code is as follows:

```
Declare
s varchar2(30):='abcd';
res varchar(50);
begin
  res:=c4(s);
  dbms_output.put_line(res);
end;
```

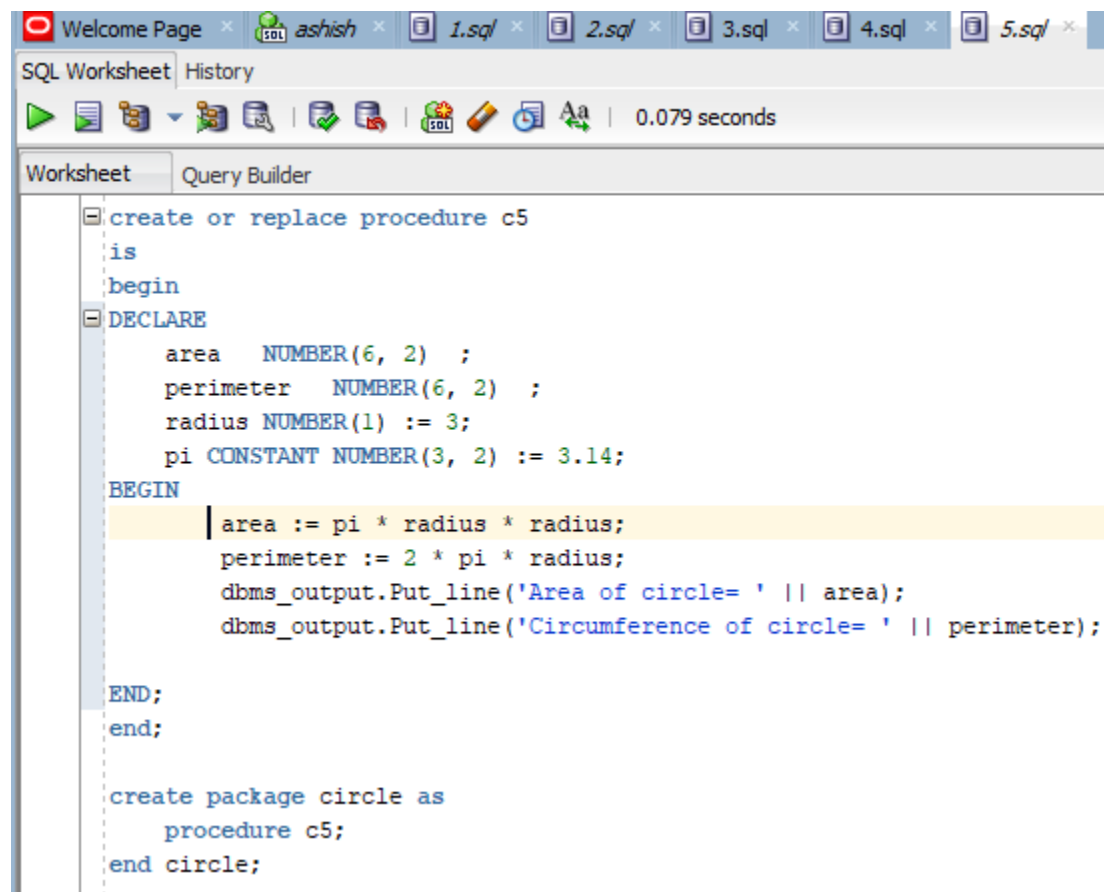
## Result-

```
dcba is not palindrome
```

```
PL/SQL procedure successfully completed.
```

## 5- Create package to find area and circumference of circle with given radius.

### Procedure and Package-



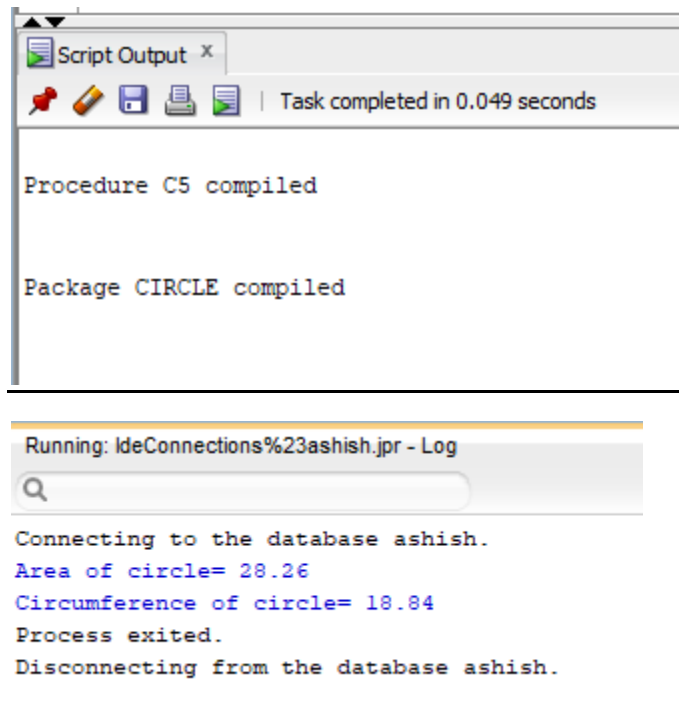
The screenshot shows a SQL Worksheet interface with a toolbar and a code editor. The code editor contains the following PL/SQL code:

```
create or replace procedure c5
is
begin
DECLARE
    area    NUMBER(6, 2) ;
    perimeter NUMBER(6, 2) ;
    radius  NUMBER(1) := 3;
    pi CONSTANT NUMBER(3, 2) := 3.14;
BEGIN
    area := pi * radius * radius;
    perimeter := 2 * pi * radius;
    dbms_output.Put_line('Area of circle= ' || area);
    dbms_output.Put_line('Circumference of circle= ' || perimeter);

END;
end;

create package circle as
    procedure c5;
end circle;
```

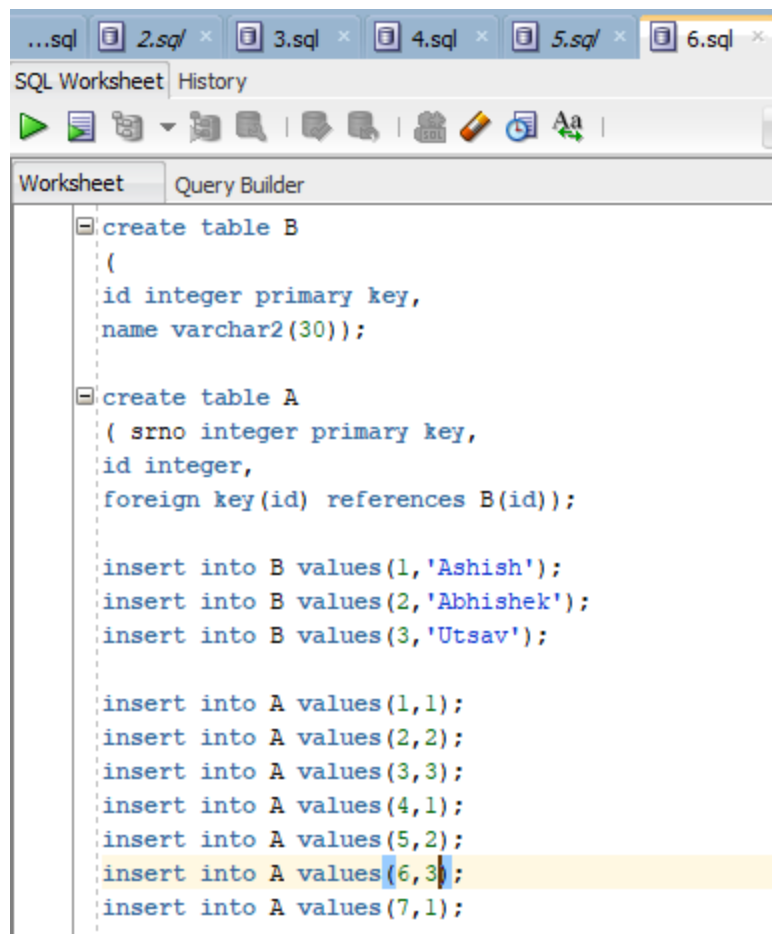
### Result-



**6- Create a trigger to delete tuple from table A even if foreign key violation is there by first deleting tuples from table B.**

**Creating Tables-**





The screenshot shows an SQL Worksheet interface with a toolbar and a code editor. The code editor contains the following SQL statements:

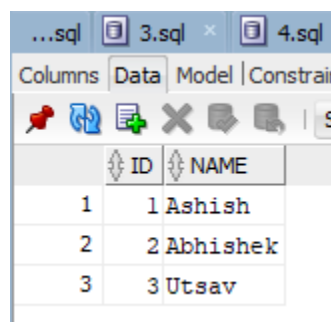
```
create table B
(
  id integer primary key,
  name varchar2(30));

create table A
( srno integer primary key,
  id integer,
  foreign key(id) references B(id));

insert into B values(1,'Ashish');
insert into B values(2,'Abhishek');
insert into B values(3,'Utsav');

insert into A values(1,1);
insert into A values(2,2);
insert into A values(3,3);
insert into A values(4,1);
insert into A values(5,2);
insert into A values(6,3);
insert into A values(7,1);
```

## Show Tables-



The screenshot shows a database table view with columns ID and NAME. The data is as follows:

ID	NAME
1	1 Ashish
2	2 Abhishek
3	3 Utsav

	SRNO	ID
1	1	1
2	2	2
3	3	3
4	4	1
5	5	2
6	6	3
7	7	1

## Trigger-

0.049 seconds

Worksheet Query Builder

```

create or replace trigger c6
after delete on B
for each row
declare
num integer;
BEGIN
    num:=:old.id;
    delete from A where id=num;
END;

```

Script Output x

Task completed in 0.049 seconds

Trigger C6 compiled

## Delete Query-

0.079 seconds

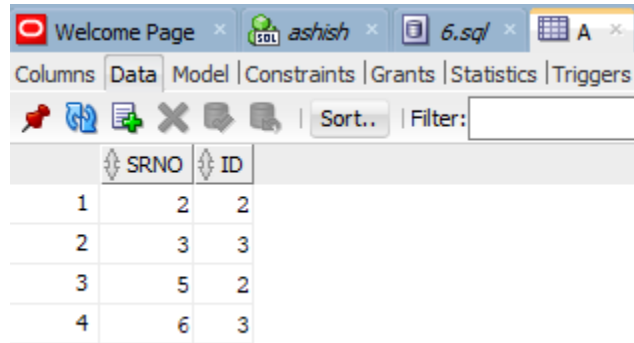
Worksheet Query Builder

```

delete from B where id=1;

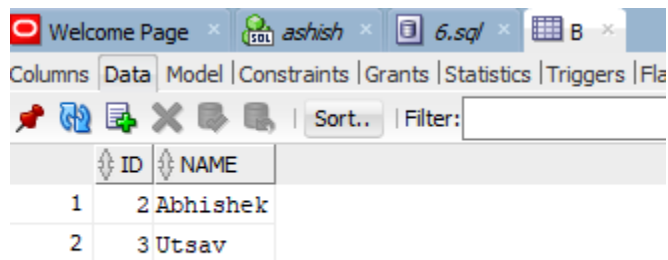
```

## Final Result-



The screenshot shows a database management tool interface. The top bar includes tabs for 'Welcome Page', 'ashish', '6.sql', and 'A'. Below the tabs are tabs for 'Columns', 'Data', 'Model', 'Constraints', 'Grants', 'Statistics', and 'Triggers'. The 'Data' tab is selected. Below the tabs are icons for adding, deleting, and refreshing data, along with 'Sort..' and 'Filter:' buttons. The table has two columns: 'SRNO' and 'ID'. The data is as follows:

	SRNO	ID
1	2	2
2	3	3
3	5	2
4	6	3



The screenshot shows a database management tool interface. The top bar includes tabs for 'Welcome Page', 'ashish', '6.sql', and 'B'. Below the tabs are tabs for 'Columns', 'Data', 'Model', 'Constraints', 'Grants', 'Statistics', 'Triggers', and 'Fla'. The 'Data' tab is selected. Below the tabs are icons for adding, deleting, and refreshing data, along with 'Sort..' and 'Filter:' buttons. The table has two columns: 'ID' and 'NAME'. The data is as follows:

	ID	NAME
1	2	Abhishek
2	3	Utsav