

Experiment 9:

Write a procedure to check whether the given number is Armstrong or not.

```
SQL> set serveroutput on;
SQL> declare
    n number(3);
    s number(3):=0;
    t number(3);
    begin
        n:=&n;
        t:=n;
        while t>0 loop
            s:=s+power((t mod 10),3);
            t:=trunc(t/10);
        end loop;
        if(s=n) then
            dbms_output.put_line('the given number' || n || 'is an armstrong number');
        else
            dbms_output.put_line('the given number' || n || 'is not an armstrong number');
        end if;
    end;
/
```

Enter value for n: 158

old 6: n:=&n;

new 6: n:=158;

the given number158is not an armstrong number

PL/SQL procedure successfully completed.

```
SQL> /
```

Enter value for n: 153

old 6: n:=&n;

new 6: n:=153;

the given number153is an armstrong number

PL/SQL procedure successfully completed.

Experiment 10:

Write a procedure which accept the account number of a customer and retrieve the balance.

```
Sql>create table customer(acc int, name varchar(20), bal int);
```

```
Sql>insert into customer values(1,'sagar',1050)
```

```
Sql>insert into customer values(2,'ram',150)
```

```
Sql>insert into customer values(3,'bhim',100)
```

```
Sql>insert into customer values(4,'srk',105)
```

```
Sql>insert into customer values(5,'sai',175)
```

```
SQL> select * from customer;
```

ACC NAME	BAL
1 sagar	1050
2 ram	150
3 bhim	100
4 srk	105
5 sai	155