

```
drop table triangulos;
drop type tipo_triangulo;
```

```
create or replace type tipo_triangulo as object(
base number,
altura number,
member function area return number);
```

```
/
```

```
create or replace type body tipo_triangulo as
member function area return number is
```

```
a number;
```

```
begin
```

```
a:=(base*altura)/2;
```

```
return a;
```

```
end;
```

```
end;
```

```
/
```

```
create table triangulos (id number, triangulo tipo_triangulo);
```

```
insert into triangulos values(1, tipo_triangulo(5,6));
```

```
insert into triangulos values(2, tipo_triangulo(10,10));
```

```
set serveroutput on;
```

```
SQL> cle@ejercicio3.sql
```

```
Table dropped.
```

```
Type dropped.
```

```
Type created.
```

```
Type body created.
```

```
Table created.
```

```
1 row created.
```

```
1 row created.
```

```
1 row created.
```

```
/*Recorrer la tabla con for */
```

```
declare
```

```
t tipo_triangulo;
```

```
begin
```

```
for i in (select * from triangulos) loop
```

```
t:=i.triangulo;
```

```
dbms_output.put_line('EL TRIANGULO CON ID: ' || i.id);
```

```
dbms_output.put_line('CON BASE: ' || t.base);
```

```
dbms_output.put_line('CON ALTURA: ' || t.altura);
```

```
dbms_output.put_line('TIENE UN AREA DE: ' || t.area);
```

```
END LOOP;
```

```
END;
```

```
/
```

```
/*Recorrer la tabla con cursor y while*/
```

```

declare

cursor c1 is select id,t.triangulo.base base, t.triangulo.altura altura, t.triangulo.area() area from
triangulos t;

v_cur c1%rowtype;

begin

open c1;

fetch c1 into v_cur;

while c1%found loop

dbms_output.put_line('EL TRIANGULO CON ID: '||v_cur.id);

dbms_output.put_line('CON BASE: '||v_cur.base);

dbms_output.put_line('CON ALTURA: '||v_cur.altura);

dbms_output.put_line('TIENE UN AREA DE: '||v_cur.area);

fetch c1 into v_cur;

end loop;

close c1;

end;

/

/*Recorrer la tabla con cursor y for */

declare

cursor c1 is select * from triangulos;

begin

for v_triangulo in c1 loop

dbms_output.put_line('EL TRIANGULO CON ID: '||v_triangulo.id);

dbms_output.put_line('CON BASE: '||v_triangulo.triangulo.base);

dbms_output.put_line('CON ALTURA: '||v_triangulo.triangulo.altura);

dbms_output.put_line('TIENE UN AREA DE: '||v_triangulo.triangulo.area);

end loop;

end;

/

```

```
EL TRIANGULO CON ID: 1  
CON BASE: 5  
CON ALTURA: 6  
TIENE UN AREA DE: 15  
EL TRIANGULO CON ID: 2  
CON BASE: 10  
CON ALTURA: 10  
TIENE UN AREA DE: 50  
EL TRIANGULO CON ID: 2  
CON BASE: 10  
CON ALTURA: 10  
TIENE UN AREA DE: 50
```

PL/SQL procedure successfully completed.

```
EL TRIANGULO CON ID: 1  
CON BASE: 5  
CON ALTURA: 6  
TIENE UN AREA DE: 15  
EL TRIANGULO CON ID: 2  
CON BASE: 10  
CON ALTURA: 10  
TIENE UN AREA DE: 50  
EL TRIANGULO CON ID: 2  
CON BASE: 10  
CON ALTURA: 10  
TIENE UN AREA DE: 50
```

PL/SQL procedure successfully completed.

```
EL TRIANGULO CON ID: 1  
CON BASE: 5  
CON ALTURA: 6  
TIENE UN AREA DE: 15  
EL TRIANGULO CON ID: 2  
CON BASE: 10  
CON ALTURA: 10  
TIENE UN AREA DE: 50  
EL TRIANGULO CON ID: 2  
CON BASE: 10  
CON ALTURA: 10  
TIENE UN AREA DE: 50
```

PL/SQL procedure successfully completed.