

# INTRODUCCIÓN A OBJETOS EN ORACLE

---

# Introducción a objetos

---

- ❑ Oracle es una base de datos de tipo relacional y por lo tanto no es una base de datos de tipo objeto.
- ❑ Sin embargo, Oracle incorpora funcionalidades que permiten trabajar con algunas de las características de las bases de datos orientadas a objeto.
- ❑ Hay que tener en cuenta que este tipo de base de datos representa la información mediante objetos, que son similares a los que se utilizan en los lenguajes de desarrollo orientados a objeto

# Introducción a objetos

---

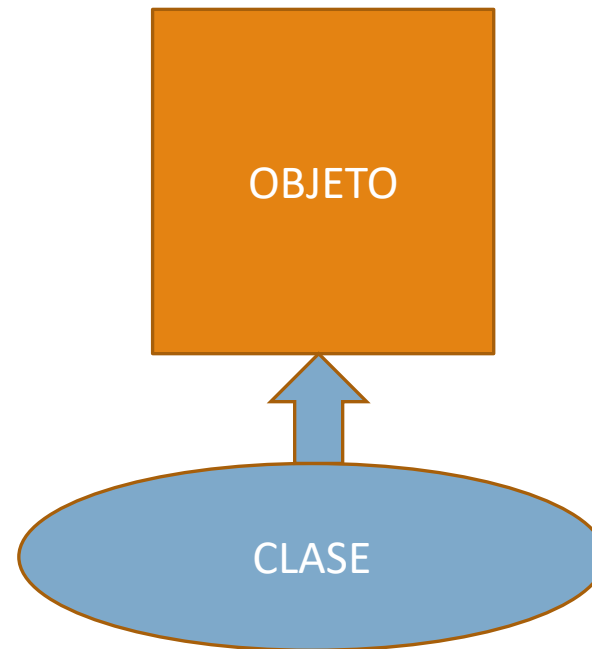
- ❑ Por supuesto una base de datos de este tipo permite implementar objetos complejos y se guía a través de la programación orientada a objetos



# Introducción a objetos

---

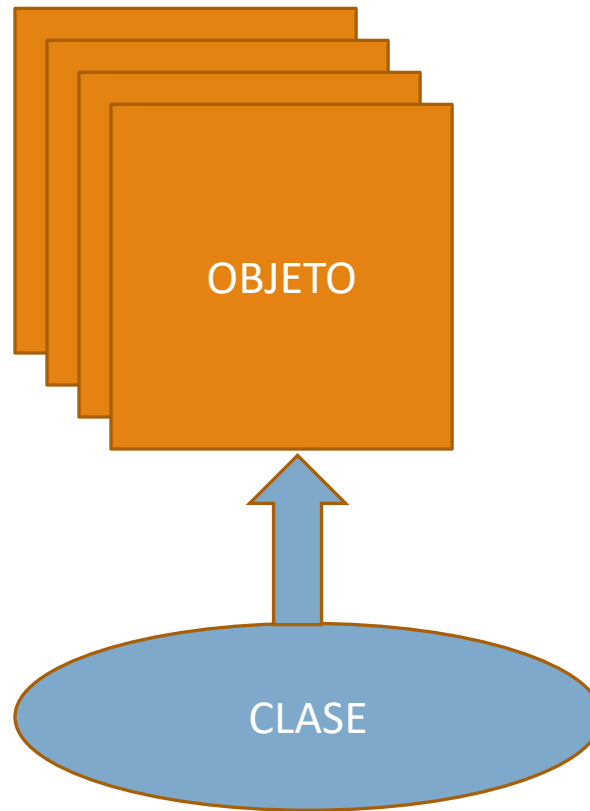
- Un objeto se crea a través de una clase (plantilla) que define sus características



# Introducción a objetos

---

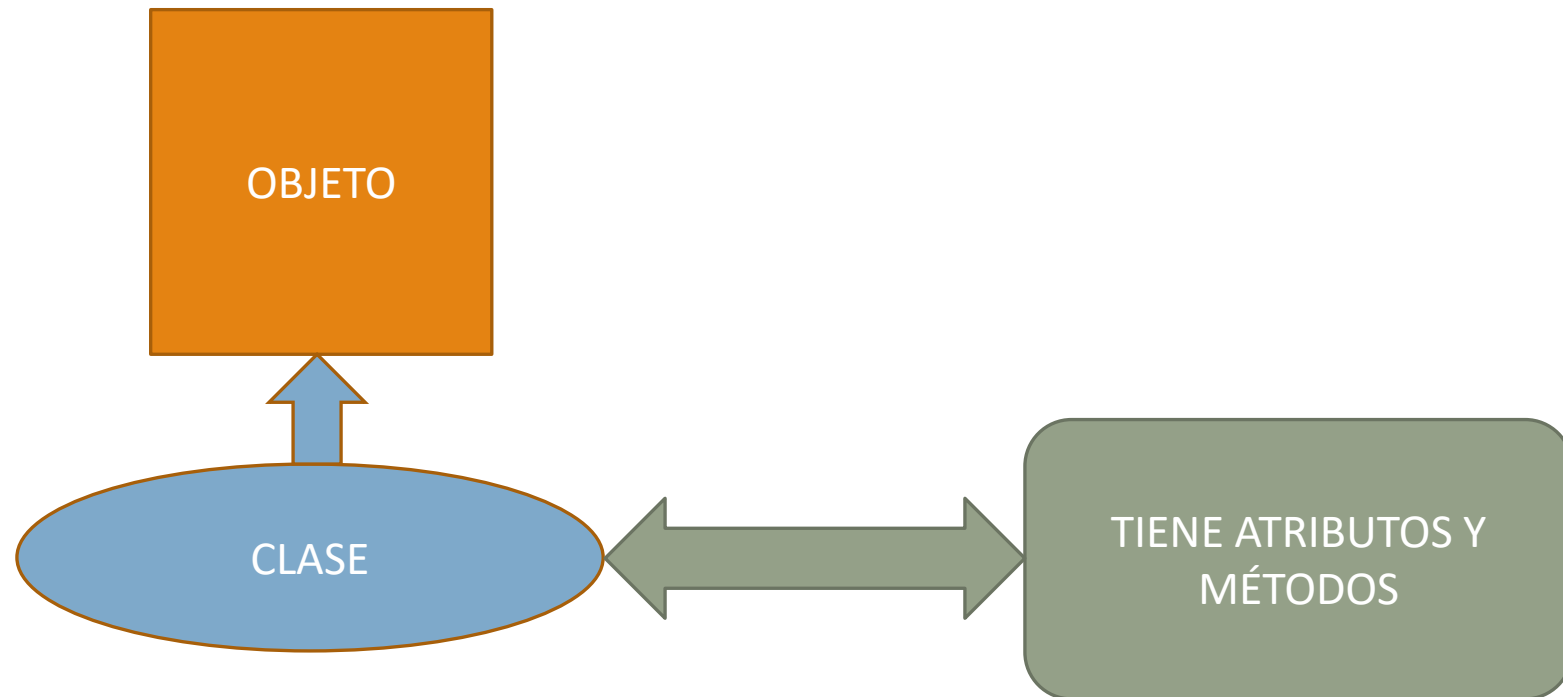
- A partir de la clase podemos crear (instanciar) múltiples objetos



# Introducción a objetos

---

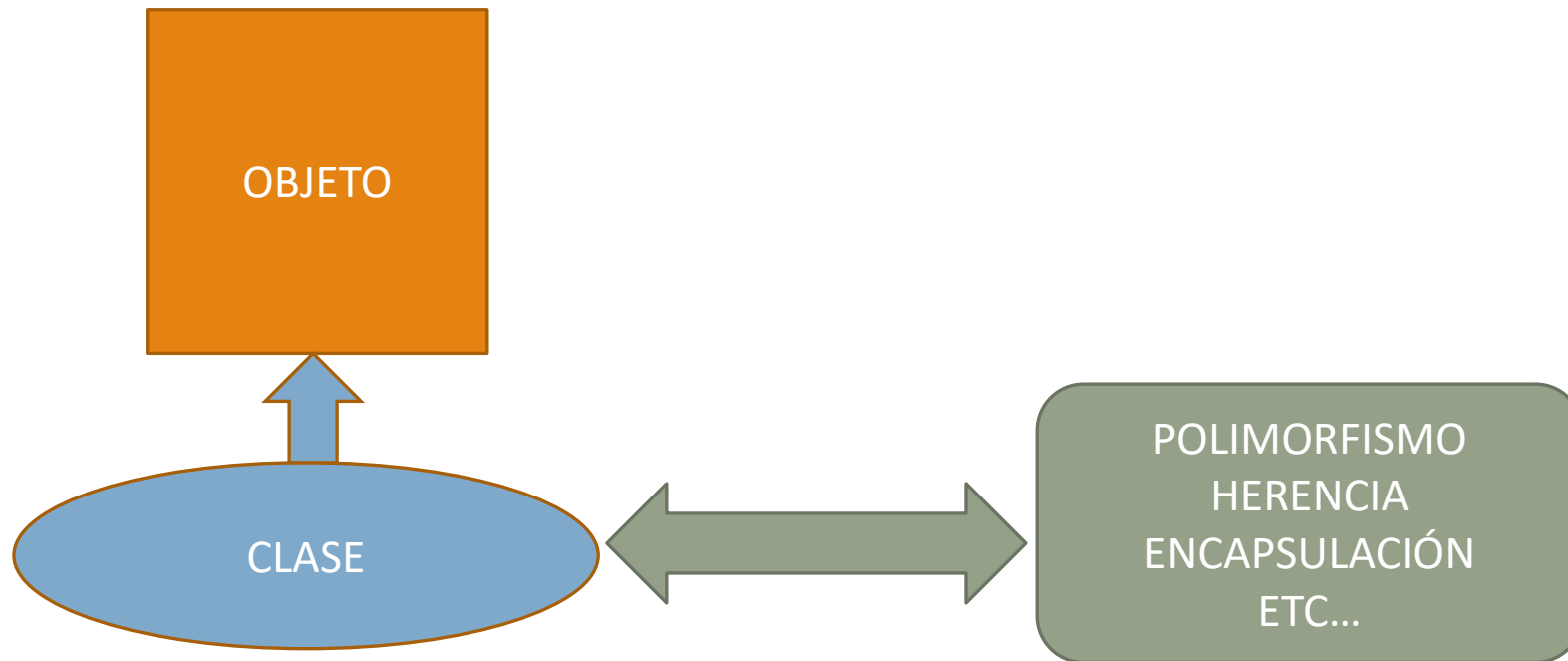
- Una clase tiene atributos y métodos



# Introducción a objetos

---

□ Y otras características de la orientación a objetos



# Introducción a objetos

---

- ❑ En oracle podemos crear una clase (plantilla) de esta forma:
- ❑ Primero creamos la definición, como cuando creamos la cabecera de un paquetes

```
CREATE OR REPLACE TYPE empleado AS OBJECT (  
  --Atributos  
  nombre varchar2(100),  
  apellidos varchar2(100),  
  Edad number,  
  --Métodos  
  MEMBER FUNCTION ver_nombre RETURN varchar2,  
  MEMBER FUNCTION ver_edad RETURN NUMBER  
)
```



# Introducción a objetos

---

□ Luego creamos la especificación o body del objeto

```
CREATE OR REPLACE TYPE body empleado as
MEMBER FUNCTION ver_nombre RETURN varchar2
IS
BEGIN
    RETURN NOMBRE;
END;
MEMBER FUNCTION ver_edad RETURN NUMBER
IS
BEGIN
    RETURN EDAD;
END;
END;
```

# Introducción a objetos

---

□ Luego las podemos usar como tipos “normales” en PL/SQL

```
set serveroutput on format wrapped line 1000;

declare
  v1 empleado:=empleado('pepe','rodriguez',89);
begin

  dbms_output.put_line(v1.ver_nombre());
end;
/
```

# Introducción a objetos

---

❑ Creamos una variable de ese tipo

La declaramos  
con el tipo

```
set serveroutput on format wrapped line 1000;

declare
  v1 empleado:=empleado('pepe','rodriguez',89);
begin

  dbms_output.put_line(v1.ver_nombre());
end;
/
```

# Introducción a objetos

---

## ❑ Inicializamos los valores con un constructor

```
set serveroutput on format wrapped line 1000;
```

```
declare
```

```
  v1 empleado:=empleado('pepe','rodriguez',89);
```

```
begin
```

```
  dbms_output.put_line(v1.ver_nombre());
```

```
end;
```

```
/
```



El constructor inicializa  
los atributos

# Introducción a objetos

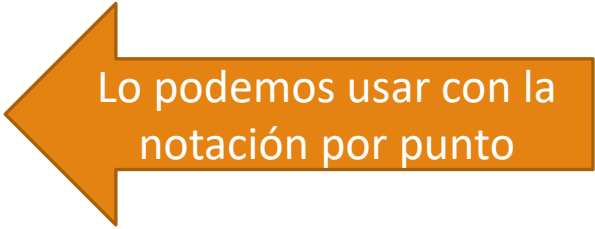
---

## □ Inicializamos los valores con un constructor

```
set serveroutput on format wrapped line 1000;

declare
  v1 empleado:=empleado('pepe','rodriguez',89);
begin

  dbms_output.put_line(v1.ver_nombre());
end;
/
```



Lo podemos usar con la notación por punto