**Desarrollo de aplicaciones multiplataforma**

**Acceso a datos**

**Tema 2-Práctica 6: Modificar métodos**

1. Crear el tipo “empleado” con los atributos “Rut varchar(10)”, “Nombre varchar(10)”, “Cargo varchar(9)”, “fechaIng date”, “sueldo number(9)”, “somision number (9)”, y “anticipo number (9)”, la función “sueldo\_liquido”, que devuelve un number, y el procedimiento “aumento\_sueldo” que recibe como parámetro el aumento (number).

**RESPUESTA:**

DROP TYPE empleado;

CREATE OR REPLACE TYPE empleado AS OBJECT (

rut VARCHAR(10),

nombre VARCHAR(10),

cargo VARCHAR(9),

fechaing DATE,

sueldo NUMBER(9),

comision NUMBER(9),

anticipo NUMBER(9),

MEMBER FUNCTION sueldo\_liquido RETURN NUMBER,

MEMBER PROCEDURE aumento\_sueldo (

aumento NUMBER

)

);

1. Crear el body para dicho tipo desarrollando la función y el procedimiento. El sueldo líquido se calculará como “(sueldo +comisión )- anticipo”, y el aumento de sueldo se calculará como “sueldo+aumento”.

**RESPUESTA:**

CREATE OR REPLACE TYPE BODY empleado AS

MEMBER FUNCTION sueldo\_liquido RETURN NUMBER AS

BEGIN

return(sueldo + comision - anticipo);

END;

MEMBER PROCEDURE aumento\_sueldo (

aumento NUMBER

) AS

BEGIN

sueldo :=sueldo + aumento;

END;

END;

1. Altera el tipo Empleado y añade el procedimiento “setAnticipo” que recibe como parámetro el anticipo de tipo number.

**RESPUESTA:**

DROP TYPE empleado;

CREATE OR REPLACE TYPE empleado AS OBJECT (

rut VARCHAR(10),

nombre VARCHAR(10),

cargo VARCHAR(9),

fechaing DATE,

sueldo NUMBER(9),

comision NUMBER(9),

anticipo NUMBER(9),

MEMBER FUNCTION sueldo\_liquido RETURN NUMBER,

MEMBER PROCEDURE aumento\_sueldo (

aumento NUMBER

),

MEMBER PROCEDURE setanticipo (

anticipo NUMBER

)

);

/

1. Crea el body para el nuevo método “setAnticipo”.

**RESPUESTA:**

CREATE OR REPLACE TYPE BODY empleado AS

MEMBER FUNCTION sueldo\_liquido RETURN NUMBER AS

BEGIN

return(sueldo + comision - anticipo);

END;

MEMBER PROCEDURE aumento\_sueldo (

aumento NUMBER

) AS

BEGIN

self.sueldo := self.sueldo + aumento;

END;

MEMBER PROCEDURE setanticipo (

anticipo NUMBER

) AS

BEGIN

self.anticipo := anticipo;

END;

END;

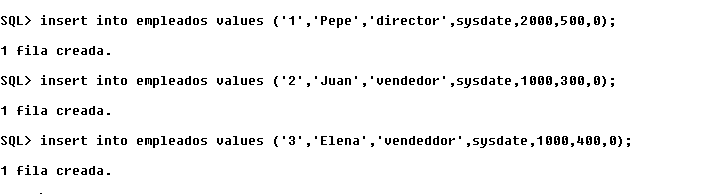
1. Crear una tabla empleados de tipo empleado.

**RESPUESTA:**

create table empleados of empleado;

/

1. Insertar dos o tres empleados, con estos datos.



**RESPUESTA:**

insert into empleados values('1','Pepe','director',sysdate,2000,500,0);

insert into empleados values('2','Juan','vendedor',sysdate,1000,300,0);

insert into empleados values('3','Elena','vendedor',sysdate,1000,400,0);

1. Crear un bloque PL/SQL para listar el sueldo líquido del empleado rut=1. Aumentarle el sueldo con 400 euros. Listar el sueldo aumentado. La salida será como sigue:



**RESPUESTA:**

DROP PROCEDURE ver\_sueldo\_liquido;

CREATE PROCEDURE ver\_sueldo\_liquido (

ruta IN VARCHAR2

) AS

emp empleado;

BEGIN

SELECT

value(e)

INTO emp

FROM

empleados e

WHERE

e.rut = ruta;

dbms\_output.put\_line(emp.nombre

|| ' '

|| emp.cargo

|| ' Sueldo: '

|| emp.sueldo

|| ' Sueldo Líquido: '

|| emp.sueldo\_liquido);

emp.aumento\_sueldo(400);

END ver\_sueldo\_liquido;

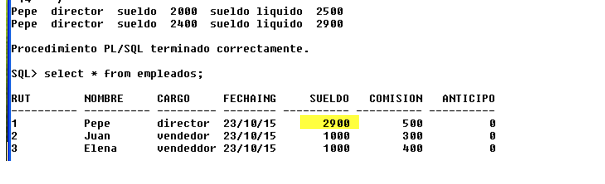
/

SET SERVEROUTPUT ON;

EXECUTE ver\_sueldo\_liquido('1');

EXECUTE ver\_sueldo\_liquido('1');

1. Persistir en la tabla empleados el sueldo aumentado



**RESPUESTA:**

DROP PROCEDURE ver\_sueldo\_liquido;

CREATE PROCEDURE ver\_sueldo\_liquido (

ruta IN VARCHAR2

) AS

emp empleado;

BEGIN

SELECT

value(e)

INTO emp

FROM

empleados e

WHERE

e.rut = ruta;

dbms\_output.put\_line(emp.nombre

|| ' '

|| emp.cargo

|| ' Sueldo: '

|| emp.sueldo

|| ' Sueldo Líquido: '

|| emp.sueldo\_liquido);

emp.aumento\_sueldo(400);

-- Actualizar la tabla empleados con los cambios en la instancia emp

UPDATE empleados

SET sueldo = emp.sueldo,

comision = emp.comision,

anticipo = emp.anticipo

WHERE rut = emp.rut;

END ver\_sueldo\_liquido;

/

SET SERVEROUTPUT ON;

EXECUTE ver\_sueldo\_liquido('1');

EXECUTE ver\_sueldo\_liquido('1');

/

select \* from empleados;

**Tabla

Descripción generada automáticamente**

1. Sacar los sueldos y sus sueldos líquidos de todos los empleados. OJO: poner alias en las funciones dentro de cursores.

**RESPUESTA:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**CURSOR c\_empleados IS**

**SELECT**

**\***

**FROM**

**empleados e;**

**BEGIN**

**FOR emp IN c\_empleados LOOP**

**dbms\_output.put\_line(emp.nombre**

**|| ' '**

**|| emp.cargo**

**|| ' Sueldo: '**

**|| emp.sueldo**

**|| ' Sueldo Líquido: '**

**|| (emp.sueldo + emp.comision - nvl(emp.anticipo,0)));**

**END LOOP;**

**END;**

**/**