Desarrollo de aplicaciones multiplataforma
Acceso a datos
Tema 2-Práctica 6: Modificar métodos

Mohamed El Younousi Bentalha

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:

```
CREATE OR REPLACE TYPE empleado AS OBJECT (
  Rut VARCHAR2(10),
  Nombre VARCHAR2(10),
  Cargo VARCHAR2(9),
  fechaIng DATE,
  sueldo NUMBER(9),
  comision NUMBER(9),
  anticipo NUMBER(9),
  MAP MEMBER FUNCTION sueldo_liquido RETURN NUMBER,
  MEMBER PROCEDURE aumento_sueldo(aumento NUMBER)
);
```

2. 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
  MAP MEMBER FUNCTION sueldo_liquido RETURN NUMBER IS
  BEGIN
    -- Calcular sueldo líquido
    RETURN sueldo + comision - anticipo;
  END sueldo_liquido;

  MEMBER PROCEDURE aumento_sueldo(aumento NUMBER) IS
  BEGIN
    -- Aplicar aumento de sueldo
    sueldo := sueldo + aumento;
  END aumento_sueldo;
END;
/
```

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

RESPUESTA:

```
ALTER TYPE empleado ADD MEMBER PROCEDURE setAnticipo(anticipo NUMBER);
```

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

RESPUESTA:

```
CREATE OR REPLACE TYPE BODY empleado AS
  MAP MEMBER FUNCTION sueldo_liquido RETURN NUMBER IS
  BEGIN
    RETURN sueldo + comision - anticipo;
  END sueldo_liquido;

  MEMBER PROCEDURE aumento_sueldo(aumento NUMBER) IS
  BEGIN
    sueldo := sueldo + aumento;
  END aumento_sueldo;

  MEMBER PROCEDURE setAnticipo(anticipo NUMBER) IS
  BEGIN
    self.anticipo := anticipo;
  END setAnticipo;
END;
/
```

5. Crear una tabla empleados de tipo empleado.

RESPUESTA:

```
CREATE TABLE empleados OF empleado;
```

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

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

1 fila creada.

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

1 fila creada.

SQL> insert into empleados values ('3','Elena','vendeddor',sysdate,1000,400,0);

1 fila creada.
```

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);
```

7. 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:

```
Pepe  director  sueldo  2000  sueldo liquido  2500
Pepe  director  sueldo  2400  sueldo liquido  2900
```

RESPUESTA:

```
DECLARE
  v_rut VARCHAR2(10) := '1';
  v_sueldo_anterior NUMBER;
  v_sueldo_liquido_anterior NUMBER;
  v_sueldo_nuevo NUMBER;
  v_sueldo_liquido_nuevo NUMBER;
BEGIN
  -- Obtener el sueldo y sueldo líquido actual del empleado con Rut=1
  SELECT e.Sueldo, e.sueldo_liquido() INTO v_sueldo_anterior, v_sueldo_liquido_anterior
  FROM empleados e
  WHERE e.Rut = v_rut;

  -- Mostrar el sueldo y sueldo líquido actual
  DBMS_OUTPUT.PUT_LINE('Antes del aumento:');
  DBMS_OUTPUT.PUT_LINE('Sueldo de ' || v_rut || ': ' || v_sueldo_anterior);
  DBMS_OUTPUT.PUT_LINE('Sueldo líquido de ' || v_rut || ': ' || v_sueldo_liquido_anterior);

  -- Aumentar el sueldo en 400 euros
  UPDATE empleados
  SET Sueldo = Sueldo + 400
  WHERE Rut = v_rut;

  -- Obtener el nuevo sueldo y sueldo líquido
  SELECT e.Sueldo, e.sueldo_liquido() INTO v_sueldo_nuevo, v_sueldo_liquido_nuevo
  FROM empleados e
  WHERE e.Rut = v_rut;

  -- Mostrar el sueldo y sueldo líquido después del aumento
  DBMS_OUTPUT.PUT_LINE('Después del aumento:');
  DBMS_OUTPUT.PUT_LINE('Sueldo de ' || v_rut || ': ' || v_sueldo_nuevo);
  DBMS_OUTPUT.PUT_LINE('Sueldo líquido de ' || v_rut || ': ' || v_sueldo_liquido_nuevo);
END;
/
```

8. Persistir en la tabla empleados el sueldo aumentado

```
Pepe  director  sueldo  2000  sueldo liquido  2500
Pepe  director  sueldo  2400  sueldo liquido  2900

Procedimiento PL/SQL terminado correctamente.

SQL> select * from empleados;

RUT        NOMBRE     CARGO     FECHAING     SUELDO   COMISION   ANTICIPO
---------- ---------- --------- -------- ---------- ---------- ----------
1          Pepe       director  23/10/15       2900        500          0
2          Juan       vendedor  23/10/15       1000        300          0
3          Elena      vendeddor 23/10/15       1000        400          0
```

RESPUESTA:

```
DECLARE
  v_rut VARCHAR2(10) := '1';
  v_sueldo_anterior NUMBER;
  v_sueldo_liquido_anterior NUMBER;
  v_sueldo_nuevo NUMBER;
  v_sueldo_liquido_nuevo NUMBER;
BEGIN
  -- Obtener el sueldo y sueldo líquido actual del empleado con Rut=1
  SELECT e.Sueldo, e.sueldo_liquido() INTO v_sueldo_anterior,
v_sueldo_liquido_anterior
  FROM empleados e
  WHERE e.Rut = v_rut;
  -- Mostrar el sueldo y sueldo líquido actual
  DBMS_OUTPUT.PUT_LINE('Antes del aumento:');
  DBMS_OUTPUT.PUT_LINE('Sueldo de ' || v_rut || ': ' || v_sueldo_anterior);
  DBMS_OUTPUT.PUT_LINE('Sueldo líquido de ' || v_rut || ': ' ||
v_sueldo_liquido_anterior);
  -- Aumentar el sueldo en 400 euros
  UPDATE empleados
  SET Sueldo = Sueldo + 400
  WHERE Rut = v_rut;
  -- Obtener el nuevo sueldo y sueldo líquido
  SELECT e.Sueldo, e.sueldo_liquido() INTO v_sueldo_nuevo,
v_sueldo_liquido_nuevo
  FROM empleados e
  WHERE e.Rut = v_rut;
  -- Mostrar el sueldo y sueldo líquido después del aumento
  DBMS_OUTPUT.PUT_LINE('Después del aumento:');
  DBMS_OUTPUT.PUT_LINE('Sueldo de ' || v_rut || ': ' || v_sueldo_nuevo);
  DBMS_OUTPUT.PUT_LINE('Sueldo líquido de ' || v_rut || ': ' ||
v_sueldo_liquido_nuevo);
   -- Persistir el sueldo aumentado en la tabla empleados
```

```
  COMMIT;
END;
/
```

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

RESPUESTA:

```
DECLARE
  CURSOR c_empleados IS
    SELECT
      e.Rut,
      e.Nombre,
      e.Cargo,
      e.Sueldo,
      e.sueldo_liquido() AS Sueldo_Liquido
    FROM
      empleados e;

  v_rut empleados.Rut%TYPE;
  v_nombre empleados.Nombre%TYPE;
  v_cargo empleados.Cargo%TYPE;
  v_sueldo empleados.Sueldo%TYPE;
  v_sueldo_liquido NUMBER;

BEGIN
  OPEN c_empleados;
  LOOP
    FETCH c_empleados INTO v_rut, v_nombre, v_cargo, v_sueldo,
v_sueldo_liquido;
    EXIT WHEN c_empleados%NOTFOUND;

    -- Mostrar información
    DBMS_OUTPUT.PUT_LINE('Rut: ' || v_rut);
    DBMS_OUTPUT.PUT_LINE('Nombre: ' || v_nombre);
    DBMS_OUTPUT.PUT_LINE('Cargo: ' || v_cargo);
    DBMS_OUTPUT.PUT_LINE('Sueldo: ' || v_sueldo);
    DBMS_OUTPUT.PUT_LINE('Sueldo Líquido: ' || v_sueldo_liquido);
    DBMS_OUTPUT.PUT_LINE('----------------------');
  END LOOP;
  CLOSE c_empleados;
END;
/
```