Unidad 7:
BD objetos relacionales

OBJETIVOS

- Identificar las características de las bases de datos objeto-relacionales.
- Crear tipos de datos objeto, sus atributos y métodos.
- Crear tablas de objetos y tablas de columnas tipo objeto.
- Crear tipos de datos colección.
- Realizar de consultas con objetos.
- Modificar la información almacenada manteniendo la integridad y consistencia de los datos a través de objetos.

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Unidad 6:
Lenguaje PL/SQL

1. Características de las bases de datos objeto-relacionales.
2. Tipos de datos objeto.
3. Definición de tipos de objeto. Arquitectura
4. Utilización de objetos
5. Métodos MAP y ORDER
6. Tipos de datos colección
7. Tablas de objetos

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

BD objetos relacionales

CARACTERÍSTICAS

- Las BD Oracle implementa el modelo orientado a objetos como una extensión del modelo relacional, siguiendo soportando la funcionalidad estándar de las bases de datos relacionales.

- El modelo objeto-relacional ofrece las ventajas de las técnicas orientadas a objetos en cuanto a mejorar la reutilización y el uso intuitivo de los objetos al representar mejor el mundo real, a la vez que se mantiene la alta capacidad de concurrencia y el rendimiento de las bases de datos relacionales.

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

BD objetos relacionales

CARACTERÍSTICAS

- Aplicamos los mismos conceptos que en la programación a objetos, que por ejemplo en Java

- Se definirán:
  - clases con su atributos y sus métodos (procedimientos y funciones)
  - Definiremos objetos a partir de esas clases, p.e. un empleado, un departamento
  - Podremos acceder a sus atributos y a los métodos de un objeto igual que hacemos en java.

- podremos crear tablas donde una columna o mas columnas puede ser elementos objeto, consultándolas y operando como con cualquier tabla

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Definición de Tipos de Datos Objeto

DEFINICIÓN DE TIPOS DE OBJETO (CLASE)

- Es donde definimos la clase genérica
- Es donde se declaran atributos y métodos en la especificación, pero no constantes (CONSTANTS), excepciones (EXCEPTIONS), cursores (CURSORS) o tipos (TYPES).
- Al menos debe tener un atributo declarado, y un máximo de 1000. En cambio los métodos son opcionales, por lo que se puede crear un tipo de objeto sin métodos.

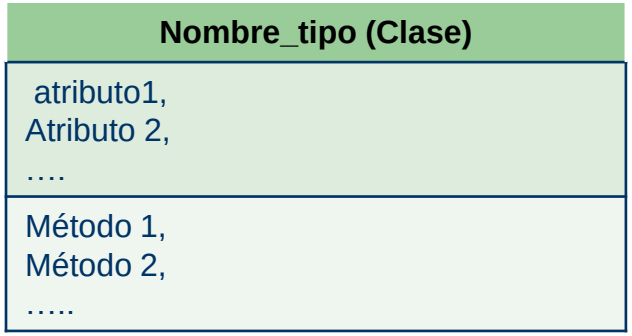
Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Definición de Tipos de Datos Objeto

DEFINICIÓN DE TIPOS DE OBJETO (CLASE)

- Para definir un tipo de objeto

```
CREATE or REPLACE TYPE nombre_tipo AS OBJECT (
   Declaración_atributos ,
   Declaración_métodos
);
```



Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Definición de Tipos de Datos Objeto

DECLARACION DE ATRIBUTOS DEL TIPO (CLASE)

- Para declarar los atributos podemos utilizar cualquier tipo de dato de Oracle excepto:

  Long, log raw, powid , los tipos específicos PL/SQL BINARY_INTEGER, BOOLEAN, PLS_INTEGAR, RECORD, REF CURSOR, %TYPE y %ROWTYPE, así como los tipos definidos dentro de un paquete PL/SQL

- NO pueden inicializar se los atributos usando el operador de asignación, ni la cláusula DEFAULT, ni asignar la restricción NOT NULL.

Ej:

```
CREATE OR REPLACE TYPE Usuario AS OBJECT (
    login VARCHAR2(10),
    nombre VARCHAR2(30),
    f_ingreso DATE,
    credito NUMBER
);
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Definición de Tipos de Datos Objeto

DECLARACION DE ATRIBUTOS DEL TIPO (CLASE)

Para borrar, añadir o modificar un atributo, podemos hacerlo:

- desde el propio SQL Developer, seleccionando el tipo de objeto
- *O utilizando los comandos del propio SQL*

ALTER TYPE Usuario DROP ATTRIBUTE f_ingreso**;** → **borra un atributo**

ALTER TYPE Usuario ADD ATTRIBUTE (apellidos VARCHAR2(40), localidad VARCHAR2(50)); → **añade atributos a la clase**

ALTER TYPE Usuario ADD ATTRIBUTE cp VARCHAR2(5),

MODIFY ATTRIBUTE nombre VARCHAR2(35); → **modifica un atributo**

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Definición de Tipos de Datos Objeto

DEFINICIÓN DE MÉTODOS DEL TIPO (DE LA CLASE)

- Básicamente sigue el mismo concepto que los paquetes. Se define en la propia Clase la cabecera del método (especificación) y se crea a continuación el cuerpo donde se desarrolla cada uno de los métodos.

```
CREATE OR REPLACE TYPE Usuario AS OBJECT (
    login VARCHAR2(10),
    nombre VARCHAR2(30),
    f_ingreso DATE,
    credito NUMBER,
    MEMBER PROCEDURE incrementoCredito(inc NUMBER),
    MEMBER FUNCTION decrementaCredito( valor NUMBER) return NUMBER
);
```

ESPECIFICACIÓN: solo las cabeceras del método

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Definición de Tipos de Datos Objeto

DEFINICIÓN DE MÉTODOS DEL TIPO (DE LA CLASE)

- Una vez creada el tipo con sus atributos y las especificaciones de los métodos (las cabeceras de los métodos) procedemos a crear el cuerpo de los métodos, donde desarrollamos lo que hace cada uno de ellos.

```
CREATE OR REPLACE TYPE BODY Usuario AS
    MEMBER PROCEDURE incrementoCredito(inc NUMBER) IS
    BEGIN
       credito := credito + inc;
    END incrementoCredito;

    MEMBER FUNCTION decrementaCredito(cant NUMBER) RETURN NUMBER IS
    BEGIN
       credito := credito - cant;
       return cantidad;
    END decrementaCredito;
END;
```

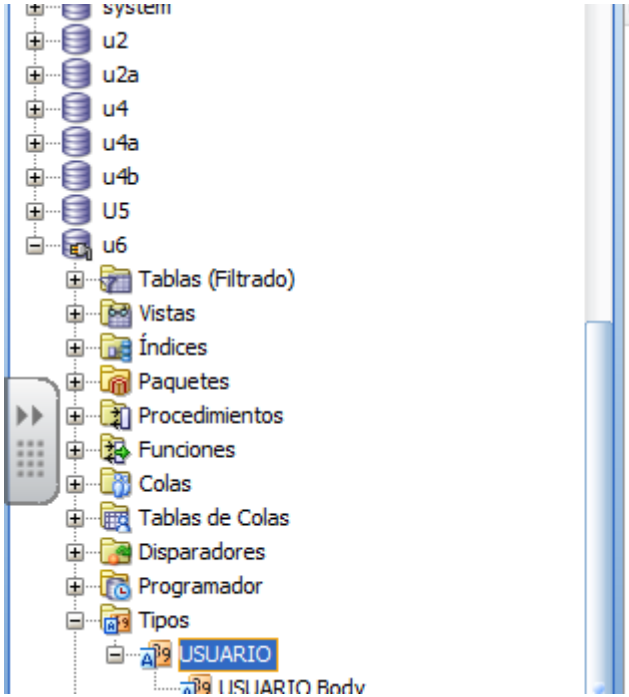
CUERPO: donde desarrollamos cada uno de los métodos

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Definición de Tipos de Datos Objeto

DEFINICIÓN DE MÉTODOS DEL TIPO (DE LA CLASE)

SQL Developer




```
create or replace
TYPE Usuario AS OBJECT (
login VARCHAR2(10),
nombre VARCHAR2(30),
f_ingreso DATE,
credito NUMBER,
CONSTRUCTOR FUNCTION Usuario(login VA
   RETURN SELF AS RESULT ,
CONSTRUCTOR FUNCTION Usuario(login VA
   RETURN SELF AS RESULT ,
MEMBER PROCEDURE incrementoCredito(inc
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

SELF

Es el equivalente al THIS de Java.

```
MEMBER PROCEDURE setNombre(Nombre VARCHAR2) IS
BEGIN
/* El primer elemento (SELF.Nombre) hace referencia al atributo del tipo (de la
    clase), el segundo (Nombre) hace referencia al parámetro del método */
    SELF.Nombre := Nombre;
END setNombre;
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

SOBRECARGA

Podemos utilizar varios métodos con el mismo nombre de método **siempre que sus parámetros formales sean** diferentes (en cantidad o tipo de dato).

```
MEMBER PROCEDURE setNombre(Nombre VARCHAR2)

MEMBER PROCEDURE setNombre(Nombre VARCHAR2, Apellidos VARCHAR2)
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

CONSTRUCTORES

Cada tipo de objeto tiene un método constructor, que se trata de una **función con el mismo nombre que el tipo de objeto y que se encarga de inicializar los atributos y retornar una nueva instancia de ese tipo de objeto.**

Oracle crea un método constructor *por defecto para cada tipo de objeto declarado, cuyos parámetros formales coinciden en orden, nombres y tipos de datos con los atributos del tipo de objeto.*

Pero también ofrece la posibilidad de crear constructores propios y sobrecargarlos para crear objetos e inicializar los atributos

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

CONSTRUCTORES

Para crear un método constructor, en la declaración del tipo de objeto (es decir lo que va a ser la clase) indicamos la cabecera del método constructor

```
CONSTRUCTOR FUNCTION Usuario(login VARCHAR2, credito NUMBER)  RETURN SELF AS RESULT
```

Y en el cuerpo desarrollamos el método:

```
CREATE OR REPLACE TYPE BODY Usuario AS
   CONSTRUCTOR FUNCTION Usuario(login VARCHAR2, credito NUMBER) RETURN SELF AS RESULT
   IS
     BEGIN
                IF (credito >= 0) THEN
                        SELF.credito := credito;
                ELSE
                        SELF.credito := 0;
                END IF;
                RETURN;
   END;
….. -- resto de métodos del tipo de objeto (clase)
END;
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

CREACION DE OBJETOS Y USO

Una vez definido el tipo de objeto (la clase) ya puedo declarar variables de dicho tipo (clase) y crearlos como objeto para usarlos.

```
var_objeto     Tipo_Objeto;
var_objeto := New Tipo_Objeto( parametors del constructor,….);
```

Ej:

u1 Usuario; → creo la variable objeto

u1 := NEW Usuario('luitom64', 'LUIS ', 'TOMAS BRUÑA', '24/10/07', 100);

Suponiendo que tenemos un constructor definido con dichos parámetros o en su caso usando el constructor por defecto que usa todos los parámetros.

Hasta que no se instancie el objeto (NEW) este tiene el valor NULL

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

ACCESO A LOS ATRIBUTOS DEL OBJETO

Para hacer referencia a un atributo de un objeto :

```
nombre_objeto.nombre_atributo
```

Ej.

```
unNombre := usuario1.nombre;
dbms_output.put_line(usuario1.nombre);

usuario1.nombre:= 'Nuevo Nombre';
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

ACCESO A LOS MÉTODOS DEL OBJETO

Para ejecutar el método de un objeto:

```
usuario1.setNombreCompleto('Juan', 'García Fernández');
```

Si el método no tiene parámetros, se indicará la lista de parámetros reales vacía (sólo con los paréntesis), aunque se pueden omitir los paréntesis.

```
credito := usuario1.getCredito();
```

los métodos STATIC se invocan usando el tipo de objeto, en lugar de una de sus instancias:

```
nombre_tipo_objeto.metodo()
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

HERENCIA

Se permite la herencia, en este caso, para indicar que un tipo de objeto es heredado de otro hay que usar la palabra reservada **UNDER**, y además hay que tener en cuenta que el tipo de objeto del que hereda debe tener la propiedad **NOT FINAL.**

```
CREATE TYPE Persona AS OBJECT (
    nombre VARCHAR2(20),
    apellidos VARCHAR2(30)
) NOT FINAL;

CREATE TYPE UsuarioPersona UNDER Persona (
    login VARCHAR(30),
    f_ingreso DATE,
    credito NUMBER
);
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

HERENCIA

```
DECLARE
   u1          UsuarioPersona;
BEGIN
   u1 := NEW UsuarioPersona('nombre1', 'apellidos1', 'user1',
                                                    '01/01/2001', 100);
   dbms_output.put_line(u1.nombre);
END;
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

MÉTODOS MAP

Las instancias de un tipo de objeto no tienen un orden predefinido. Si deseas establecer un orden en ellos, con el fin de **hacer una ordenación o una comparación, es necesario crear un método MAP,** donde indicaría los atributos que sería necesario comparar para saber como ordenar dos objetos.

```
CREATE OR REPLACE TYPE Usuario AS OBJECT (
    login VARCHAR2(30),
    nombre VARCHAR2(30),
    apellidos VARCHAR2(40),
    f_ingreso DATE,
    credito NUMBER,
    MAP MEMBER FUNCTION ordenarUsuario RETURN VARCHAR2,
    …..
);
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

MÉTODOS MAP

En el cuerpo del tipo (clase) desarrollo el método MAP, donde indicaría de todos los atributos que hay cuales son los que necesito comparar para saber como ordenar dos objetos

```
CREATE OR REPLACE TYPE BODY Usuario AS
    MAP MEMBER FUNCTION ordenarUsuario RETURN VARCHAR2 IS
    BEGIN
      RETURN (apellidos || ' ' || nombre);
    END ordenarUsuario;
    …. –resto de métodos
END;
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

MÉTODOS MAP

El lenguaje PL/SQL utiliza los métodos MAP para evaluar expresiones lógicas que resultan valores booleanos como objeto1 > objeto2, y para realizar las comparaciones implícitas en las cláusulas DISTINCT, GROUP BY y ORDER BY.

Cada tipo de objeto sólo puede tener un método MAP declarado, y sólo puede retornar alguno de los siguientes tipos: DATE, NUMBER, VARCHAR2, CHARACTER o REAL.

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

MÉTODOS ORDER

Permite **establecer un orden entre los objetos** instanciados de dicho tipo.

Un tipo de objeto solo puede tener un método ORDER, el cual debe retornar un valor numérico que permita establecer el orden entre los objetos:

- Si deseas que un objeto sea menor que otro puedes retornar, por ejemplo, el valor -1.
- Si vas a determinar que sean iguales, devuelve 0,
- y si va a ser mayor, retorna 1.

Sólo podemos declarar un método MAP o un método ORDER, pero no los dos.

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

MÉTODOS ORDER

```
CREATE OR REPLACE TYPE BODY Usuario AS
    ORDER MEMBER FUNCTION ordenUsuario(u Usuario) RETURN INTEGER IS
BEGIN
/* comparo los login para ordenar
  IF SELF.login < u.login THEN
                    RETURN -1;
    ELSIF SELF.login > u.login THEN
                    RETURN 1;
    ELSE
       RETURN 0;
  END IF;
 END;
END;
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

MAP o ORDER

Sólo podemos declarar un método MAP o un método ORDER, pero no los dos.

Cuando se vaya a ordenar o mezclar un alto número de objetos, es preferible usar un método MAP, ya que en esos casos un método ORDER es menos eficiente.

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

TABLAS DE OBJETOS

Podemos crear tablas de objetos, para ellos

```
CREATE TABLE NombreTabla OF TipoObjeto;
```

Ej:

```
CREATE TABLE TablaUsuarioObjeto OF Usuario;
```

Lo que tengo es una tabla donde cada una de las filas es un objeto con sus atributos y sus métodos

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

TABLAS DE OBJETOS

Ej:

```
CREATE TABLE TablaUsuarioObjeto OF Usuario;
```

Lo que tengo es una tabla donde cada una de las filas es un objeto con sus atributos y sus métodos

```
DECLARE
    u1 Usuario;
    u2 Usuario;
BEGIN
    u1 := NEW Usuario('luitom64', 'LUIS TOMAS BRUNA', '24/10/2007', 50);
    u2 := NEW Usuario('caragu72', 'CARLOS AGUDO SEGURA', '06/07/2007', 100);
    INSERT INTO TablaUsuarioObjeto VALUES (u1);
    INSERT INTO TablaUsuarioObjeto VALUES (u2);
END;
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

TABLAS DE OBJETOS

Creada la tabla con los objetos, ya podría usarla como cualquier tabla

```
select * from TablaUsuarioObjeto
```

```
select * from TablaUsuarioObjeto
            where     login='luitom64'
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

TABLAS CON COLUMNAS DE OBJETOS

Podemos crear tablas donde alguna de sus columnas sean objetos, para ello

```
CREATE TABLE Gente (
        dni             VARCHAR2(10),
        colObjeto       Usuario,
        partidasJugadas SMALLINT
);
```

En este caso habrá que tener presente que la segunda columna es una columna compleja, es decir un objeto con sus métodos y atributos.

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

TABLAS CON COLUMNAS DE OBJETOS

Para hacer consultas:

```
select * from gente;
```

Si queremos ver la columna dni y el atributo login de cada fila, deberemos de usar un ALIAS de la tabla

```
select  g.dni, g.colObjeto.login    from     gente  g
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

OPERACIONES SOBRE TABLAS DE OBJETOS

Podemos utilizar las mismas operaciones que sobre cualquier tabla, UPDATE, INSERT, DELETE

```
UPDATE TablaUsuarioObjeto
SET TablaUsuarioObjeto .credito = 0
WHERE TablaUsuarioObjeto .login = 'luitom64';
```

Es muy habitual abreviar el nombre de la tabla con un **alias:**

```
UPDATE TablaUsuarioObjeto   u
SET  u.unUsuario.credito = 0
WHERE u.unUsuario.login = 'luitom64';
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio

Tipos de Datos Objeto

VALUE

Cuando tengas la necesidad de hacer referencia a un objeto en lugar de alguno de sus atributos, puedes utilizar la función VALUE junto con el nombre de la tabla de objetos o su alias, dentro de una sentencia SELECT.

```
INSERT INTO Favoritos SELECT VALUE(u) FROM TablaUsuarioObjeto  u
                    WHERE u.credito >= 100;

SELECT u.login FROM TablaUsuarioObjeto u
                    JOIN Favoritos f ON VALUE(u)=VALUE(f);
```

Roberto Portabales Enríquez
IES JJ Calvo Miguel - Sotrondio