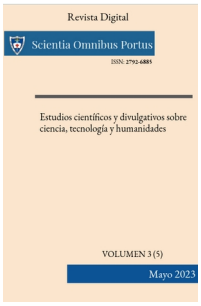
Scientia Omnibus Portus
ISSN: 2792-6885

https://iescelia.org/ojs



Artículo

BASES DE DATOS OBJETO-RELACIONALES : CONCEPTO PRÁCTICO

Eduardo Tapia

etappad369@g.educaand.es

RESUMEN

Vamos a repasar el origen y características del Modelo de Bases de Datos Objeto-Relacional (ODMG : Object Database Management Group). Describiremos los conceptos básicos del Modelo así como los detalles de su implementación y las operaciones básicas sobre activos automatizados. Estudiaremos, mediante un ejemplo, la implementación del mismo por los grandes fabricantes de Sistemas Gestores de Bases de Datos .- Para poder automatizar el Modelo estudiado se implementó el estándar SQL1999 que capacita a los Sistemas Gestores automatizar los objetos del nuevo concepto de diseño de Bases de Datos .-

PALABRAS CLAVE : artículo, científico, divulgación, publicación, redacción, manuscrito, bases de datos objeto-relacionales, postgres, programación orientada a objetos.-

INTRODUCCIÓN

Este Lenguaje de programación insertado dentro de las funcionalidades de un Sistema Gestor de Bases de Datos, que , además de las prestaciones del Lenguaje SQL, soporta la definición de variables, estructuras de carácter modular, sentencias de control y declaración de excepciones.-
Se puede definir una función con PL/SQL al igual que en resto de lenguajes de programación conocidos.- Asimismo, se pueden definir scripts como partes del Lenguaje SQL.-Nació en el año 1988 .- Es un Lenguaje de Procedimiento diseñado para albergar sentencias SQL y ser ejecutado en el servidor de la Base de Datos .-

Este nuevo término de base de datos objeto-relacional es usado para describir a una base de datos que ha evolucionado desde el modelo relacional hasta una base de datos que contiene ambas tecnologías: relacional y de objetos.

Un tipo de objeto consta de dos partes: especificación y cuerpo:

1º.- La especificación está formada por la interface a las aplicaciones; aquí se declaran las estructuras de datos (conjunto de atributos) y las operaciones (métodos) necesarios para manipular los datos.
2º.- El cuerpo define los métodos, es decir, implementa la especificación.

DESCRIPCIÓN DEL MODELO

Tanto el estándar SQL1999 así como el SQL2003 introdujeron las extensiones del lenguaje SQL, utilizado por los Sistemas Gestores de Bases de Datos, para poder implementar un objeto de datos en los mismos .- Los principales componentes del ODMG son :

1º.- Arquitectura del Modelo.-
2º.- Modelo de Objetos.-
3º.- Lenguaje de Definición de Objetos .-
4º.- Lenguaje de Consulta de Objetos .-
5º.- Conexión con los distintos lenguajes de programación (como Java, C++, etc .) .-

Veamos con una simple implementación de un tipo objeto como funciona el Modelo :

DEFINICIÓN EN PL/SQL :

```
CREATE TYPE direccion AS OBJECT (
//creamos el tipo dirección como objeto para ser usado
calle VARCHAR2(250),
// declaramos la variable calle como alfanumérica de 250 caractéres
ciudad VARCHAR2(300),
// declaramos la variable ciudad como alfanumérica de 300 caractéres
provincia CHAR(250),
// declaramos la variable provincia como alfanumérica de 250 caractéres
cod_postal VARCHAR2(5) ) ;
//declaramos la variable cod_postal como alfanumérica de 5 caractéres

/* hemos definido el tipo dirección */

CREATE TYPE alumnoa AS OBJECT (
//creamos el tipo alumnoa como objeto para ser usado
NIE NUMBER,
// declaramos la variable NIE como numérica
nomb VARCHAR2(300),
// declaramos la variable nomb como alfanumérica de 300 caractéres
direc direccion,
```

```
// declaramos la variable direc como objeto dirección
telef VARCHAR2(12),
// declaramos la variable telef como alfanumérica de 12 caractéres
fecnac DATE,
// declaramos la variable fecnac como fecha
MEMBER FUNCTION edad RETURN NUMBER,
PRAGMA RESTRICT_REFERENCES(edad,WNDS)
) ;

/* hemos definido el tipo alumno/a con su función edad */

CREATE OR REPLACE TYPE BODY alumnoa AS
MEMBER FUNCTION edad RETURN NUMBER IS
v1 NUMBER;
v2 DATE;
//*Declaración de variables*/
/* Implementamos la función edad del tipo alumno/a */

BEGIN
v2:= today();
/* captura la fecha del sistema*/
v1:= v2.año – fecnac.año;
*resta al año calculado por el sistema el año dado en fecnac*
IF (v2.mes < fecnac.mes) OR
((v2.mes = fecnac.mes) AND (v2.dia < fecnac.dia))
*comparativas de mes y día para ajustar la edad*
THEN v1:= v1+1;
/*ajuste de la edad */
END IF;
*fin de comparativa condicional *
RETURN v1;
*devuelve la edad calculada en v1 *
END;
END;
/*fin del programa*/
```

VENTAJAS

1. Mejora de la capacidad de representación y tratamiento de datos. Datos mas complejos y lógica de programa asociados a los mismos .-
2. Mejor integración con lenguajes de aplicación basados en la orientación a objeto .-
3. Se pueden abordar problemas difícilmente abordables con las estructuras de datos tradicionales .-
4. Se dota a los objetos generados con los leguajes OO de todas las capacidades de persistencia, gestión y administración proporcionadas por los SGBB .-
5. Se oferta para cada problema el mecanismo de representación y tratamiento más adecuado a su complejidad estructural .-
6. Los datos estructurados se benefician de mecanismos de consulta más simples y uniformes que los proporcionados por BDOO puras .-
7. El usuario se beneficia de la contrastada fiabilidad de los sistemas de BDR sobre los que se evoluciona .-

MÉTODOS

La especificación de un método se hace junto a la creación de su tipo, y debe llevar siempre asociada una directiva de compilación (PRAGMA RESTRICT_REFERENCES), para evitar que los métodos manipulen la base de datos o las variables del paquete PL/SQL.

Tienen el siguiente significado:

• WNDS: no se permite al método modificar las tablas de la base de datos.-
• WNPS: no se permite al método modificar las variables del paquete PL/SQL.-
• RNDS: no se permite al método leer las tablas de la base de datos.-
• RNPS: no se permite al método leer las variables del paquete PL/SQL.-

INICIALIZACIÓN DE OBJETOS

Para crear o instanciar un objeto de un determinado tipo de objeto, debes hacer una llamada a su método constructor. Esto lo puedes realizar empleando la instrucción NEW seguido del nombre del tipo de objeto como una llamada a una función en la que se indican como parámetros los valores que se desean asignar a los atributos inicialmente. En una asignación también puedes optar por hacer eso mismo omitiendo la palabra NEW.-

El orden de los parámetros debe coincidir con el orden en el que están declarados los atributos, así como los tipos de datos.-

CONCLUSIONES

La capacidad de programación orientada a objetos implementada para los Sistemas Gestores de Bases de Datos son suficientemente capaces para cumplir con los requerimientos de cualquier aplicación, si bien se recomienda su uso solo cuando sea imprescindible, al objeto de no sobrecargar el CPD en el que funciona.- Con esta tecnología implementada en PL/SQL y PL/pgSQL el Modelo Objeto-Relacional encuentra el sustrato perfecto para su desarrollo dentro de los Sistemas Gestores de Bases de Datos actuales.-

REFERENCIAS BIBLIOGRÁFICAS :

- Operational Database Management Systems (WWW.ODBMS.ORG)

-Oracle PL/SQL Tapa blanda – 7 febrero 2008 de César Pérez López-ISBN-13 978-8478978465
Editorial RA-MA S.A. Editorial y Publicaciones--

Biografía:

Eduardo Tapia .- Profesor de Enseñanza Secundaria en la especialidad de Informática desde hace más de 20 años .-