BASE DE DATOS ORIENTADAS A OBJETOS // BASES DE DATOS OBJETO - RELACIONALES

Las Bases de Datos Orientadas a Objetos (BDOO) o Bases de Objetos se integran directamente y sin problemas con las aplicaciones desarrolladas en lenguajes orientados a objetos, ya que soportan un modelo de objetos puro y son ideales para almacenar y recuperar datos complejos permitiendo a los usuarios su navegación directa (sin un mapeo entre distintas representaciones).

Las Bases de Datos Objeto-Relacionales (BDOR) son bases de datos relacionales que han evolucionado hacia una base de datos más extensa y compleja , incorporando conceptos del modelo orientado a objetos. Pero en estas bases de datos aún existe un mapeo de objetos subyacente, que es costoso y poco flexible, cuando los objetos y sus interacciones son complejos.

BASE DE DATOS ORIENTADA A OBJETOS

La principal característica de las BDOO es que soportan un modelo de objetos puro y que el lenguaje de programación y el esquema de la base de datos utilizan las mismas definiciones de tipos.

Otras características importantes de las BDOO son las siguientes: Soportan las características de Orientacion a objetos, OID, Jerarquía, Objetos complejos, Acceso navegacional, Gestión de versiones.

VENTAJAS:

Una de las principales ventajas de los sistemas de bases de datos orientados a objetos es la transparencia, (manipulación directa de datos utilizando un entorno de programación basado en objetos), por lo que el programador, solo se debe preocupar de los objetos de su aplicación, en lugar de cómo los debe almacenar y recuperar de un medio físico.

Otras ventajas: Gran capacidad de modelado, Flexibilidad, Alta velocidad de procesamiento, Extensibilidad...etc.

DESVENTAJAS:

Carencia de modelo de datos universal, Falta de estandares, Complejidad, Dificil optimización de consultas...etc.

Un Sistema Gestor de Bases de Datos Orientada a Objetos (SGBDOO) y en inglés ODBMS, Object Databas Management System) es un software específico, dedicado a servir de interfaz entre la base de objetos, el usuario y las aplicaciones que la utilizan.

A continuación te indicamos algunos ejemplos de SGBOO: Db4o, Matisse, ObjectDB, EyeDB.

Todos los SGBDOO, independientemente de su estrategia de diseño, proporcionan un API (Interfaz de Programación de Aplicaciones), más o menos extenso, disponible para ciertos lenguajes OO. En el caso de Db4o, el API está disponible para Java y .Net.

En general, la conexión de una aplicación Java con una base de objetos se podrá realizar vía:

JDBC.
El API proporcionado por el propio gestor de objetos.

A una BDOO se podrán realizar consultas mediante:

Un lenguaje de consultas como OQL, si el gestor está basado en el estandar ODMG e incluye sentencias del tipo SQL.
El API proporcionado por el propio sistema gestor de bases de datos orientadas a objetos.

*Un objeto de tipo simple u objeto simple es aquel que no contiene a otros objetos y por tanto posee una estructura de un solo nivel de profundidad en este sentido. Recuerda que un objeto simple es un objeto que no contiene a otros objetos.

*Un objeto de tipo estructurado u objeto estructurado incluye entre sus componentes a otros objetos y se define aplicando los constructores de tipos disponibles por el SGBDOO recursivamente a varios niveles de profundidad. Los objetos estructurados son objetos que contienen a su vez a otros objetos (objetos hijo u objetos miembro).

En el caso de objetos estructurados se habla de diferentes niveles de profundidad del objeto. El nivel más alto, nivel 1, será el que corresponde a la definición del objeto estrucutrado (objeto padre), el siguiente nivel, nivel 2, corresponderá a la definición del objeto hijo y así sucesivamente podrá haber un nivel 3, 4... dependiendo de que los objetos hijos a su vez incluyan en su definición a otro u otros objetos miembro.

Entre un objeto y sus componentes de cada nivel, existen dos tipos de referencia: Referencia de propiedad, Referencia de asociación.

LENGUAJE OQL

OQL (Object Query Languaje) es el lenguaje de consulta de objetos propuesto en el estándar ODMG.

Las siguientes son algunas de las características más relevantes de OQL: Lenguaje declarativo como SQL, Sintaxis similar a SQL, No posee primitivas para modificar el estado de los objetos, Permite accerso tanto asociativo como navegacional.

Matisse: Es un gestor orientado a objetos que incorpora características del estándar ODMG, como los lenguajes ODL y OQL Todas las interacciones entre una aplicación Java y la base de objetos Matisse se realizan en el contexto de una transacción

JDBC: También es posible realizar consultas a una base de objetos Matisse (o de otro gestor) mediante sentencias OQL embebidas en Java. Para ello, la conexión a la base de objetos se realizará vía JDBC.

BASE DE DATOS OBJETO - RELACIONAL

Las BDOR las podemos ver como un híbrido de las BDR y las BDOO que intenta aunar los beneficios de ambos modelos, aunque por descontado, ello suponga renunciar a algunas características de ambos. Los objetivos que persiguen estas bases de datos son:

Mejorar la representación de los datos mediante la orientación a objetos.
Simplificar el acceso a datos, manteniendo el sistema relacional.

Algunas características: Tipos definidos por el usuario, Tipos objeto, Reusabilidad, Creación de funciones, Tablas anidadas, Herencia.

Podemos decir que un sistema gestor de bases de datos objeto-relacional (SGBDOR) contiene dos tecnologías; la tecnología relacional y la tecnología de objetos, pero con ciertas restricciones. Algunos ejemplos de estos son: Oracle, PosgreSQL, Mysql.

La creacion de tipos de objeto se realiza de la siguiente forma: Un tipo de objeto representa una entidad del mundo real. Encapsula datos y operaciones, por lo que en la especificación sólo se pueden declarar atributos y métodos, pero no constantes, excepciones, cursores o tipos. Los tipos de objetos actúan como plantillas para los objetos de cada tipo [Atributos, Métodos, Constructor, SELF]. Una vez que se ha definido un tipo de objeto y se ha instalado en el esquema de la base de datos, es posible usarlo en cualquier bloque PL/SQL. Las instancias de los objetos se crean en tiempo de ejecución.

Una vez definidos los tipos, éstos pueden utilizarse para definir nuevos tipos, tablas que almacenen objetos de esos tipos, o para definir el tipo de los atributos de una tabla (existen dos tipos, las tablas de objetos y las tablas que contienen alguna columna con un tipo de dato objeto). Los identificadores únicos asignados por Oracle a los objetos que se almacenan en una tabla, permiten que éstos puedan ser referenciados desde los atributos de otros objetos o desde las columnas de tablas. El tipo de datos proporcionado por Oracle para soportar esta facilidad se denomina REF. Un atributo de tipo REF almacena una referencia a un objeto del tipo definido, e implementa una relación de asociación entre los dos tipos de objetos.

Para poder implementar relaciones 1:N, Oracle permite definir tipos colección. Un dato de tipo colección está formado por un número indefinido de elementos, todos del mismo tipo. De esta manera, es posible almacenar en un atributo un conjunto de tuplas en forma de array (VARRAY), o en forma de tabla anidada.

El tipo VARRAY

Un array es un conjunto ordenado de elementos del mismo tipo. Cada elemento tiene asociado un índice que indica su posición dentro del array. Oracle permite que los VARRAY sean de longitud variable, aunque es necesario especificar un tamaño máximo cuando se declara el tipo VARRAY.

Tablas anidadas

Una tabla anidada es un conjunto de elementos del mismo tipo sin ningún orden predefinido. Estas tablas solamente pueden tener una columna que puede ser de un tipo de datos básico de Oracle, o de un tipo de objeto definido por el usuario. En este último caso, la tabla anidada también puede ser considerada como una tabla con tantas columnas como atributos tenga el tipo de objeto.

Mediante la gestión de transacciones, los sistemas gestores proporcionan un acceso concurrente a los datos almacenados, mantienen la integridad y seguridad de los datos, y proporcionan un mecanismo de recuperación de la base de datos ante fallos. Las transacciones deben cumplir el criterio ACID (Atomicidad, Consistencia, Isolation, Durabilidad).