

MODELO ORIENTADO A OBJETOS

En una base de datos orientada a objetos, los componentes se almacenan como objetos y no como datos, tal y como hace una base relacional, cuya representación son las tablas.

Adoptan como modelo de datos el de los lenguajes orientados a objetos, permitiendo así el uso de estructuras de datos tan complejas como sea necesario y eliminando en gran medida las barreras entre el desarrollo de aplicaciones y la gestión de datos.

La orientación a objetos representa el mundo real y resuelve problemas a través de objetos, ya sean tangibles o digitales. Este paradigma considera un sistema como una entidad dinámica formada de componentes.

Un sistema sólo se define por sus componentes y la manera en que éstos interactúan.

Principales características:

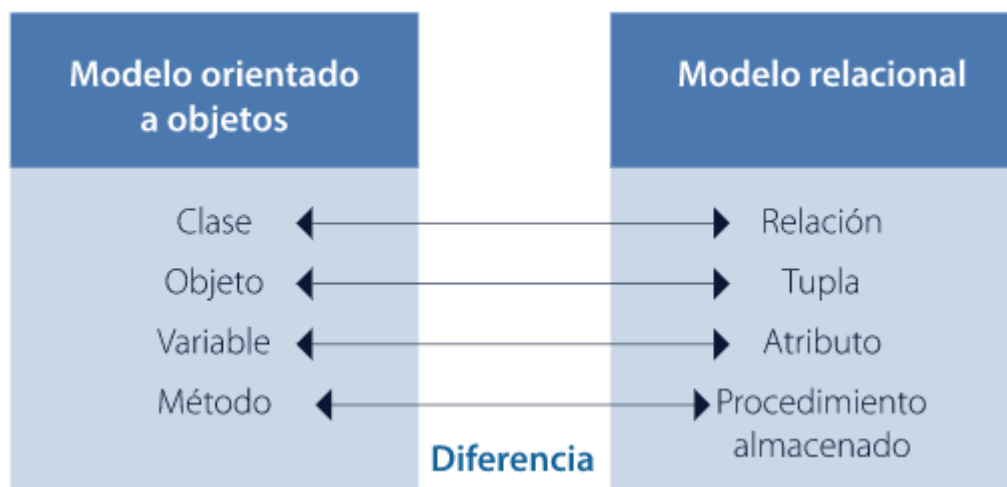
- Cada objeto tiene un nombre, atributos y operaciones.
- Es una tecnología para producir modelos que reflejen un dominio de negocio y utiliza la terminología propia de tal dominio.
- Cuenta con cinco conceptos subyacentes; objeto, mensajes, clases, herencia y polimorfismo, encapsulamiento.
- Un objeto tiene un estado, un comportamiento y una identidad.
- Los mensajes brindan comunicación entre los objetos.
- Las clases son un tipo de plantilla usada para definir objetos, los cuales son instancias del mundo real.

Objetivos propuestos por las BDOO:

- Proporcionar un modelo de datos mucho más rico y extensible, y así complementar (aunque no sustituir) a las bases de datos relacionales.
- Lograr una equivalencia al de los lenguajes de POO como C++ o Java. Tiene la gran ventaja de que, al compartir el modelo de datos, se pueden integrar las BDOO con el software usado para desarrollar aplicaciones, de manera directa y casi transparente.
- Disponer de nuevas características para el modelado de datos complejos (BDOR).

Ventajas

- Mayor capacidad de modelado. El modelado de datos orientado a objetos permite modelar el “mundo real”, de una manera mucho más fiel
- Ampliabilidad. Esto se debe a:
 - Se pueden construir nuevos tipos de datos a partir de los ya existentes.
 - Reduce la redundancia
 - Reusabilidad de clases, lo que repercute en una mayor facilidad de mantenimiento y un menor tiempo de desarrollo.
- Lenguaje de consulta más expresivo. El acceso navegacional desde un objeto al siguiente es la forma más común de acceso a datos en un SGBDOO.
- Adecuación a las aplicaciones avanzadas de base de datos
- Mayores prestaciones. Los SGBDOO proporcionan mejoras significativas de rendimiento con respecto a los SGBD relacionales.



MODELO OBJETO/RELACIONAL

El término de base de datos objeto-relacional se usa para describir una base de datos que evolucionado desde el modelo relaciones hasta una base de datos híbrida, que contiene ambas tecnologías: relacional y de objetos.

Las bases de datos Objeto-relacional son compatibles con estos objetos de datos y las operaciones de mayor complejidad.

Características:

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

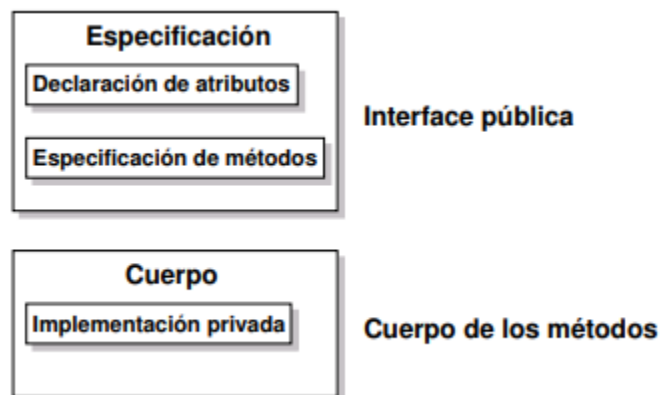
- Encapsulación – Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia – Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo – Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En una base de datos relacional se genera la posibilidad de guardar objetos más complejos en una sola tabla con referencias a otras relaciones, con lo que se acerca más al paradigma de programación orientada a objetos.

Estructura de un tipo de objeto

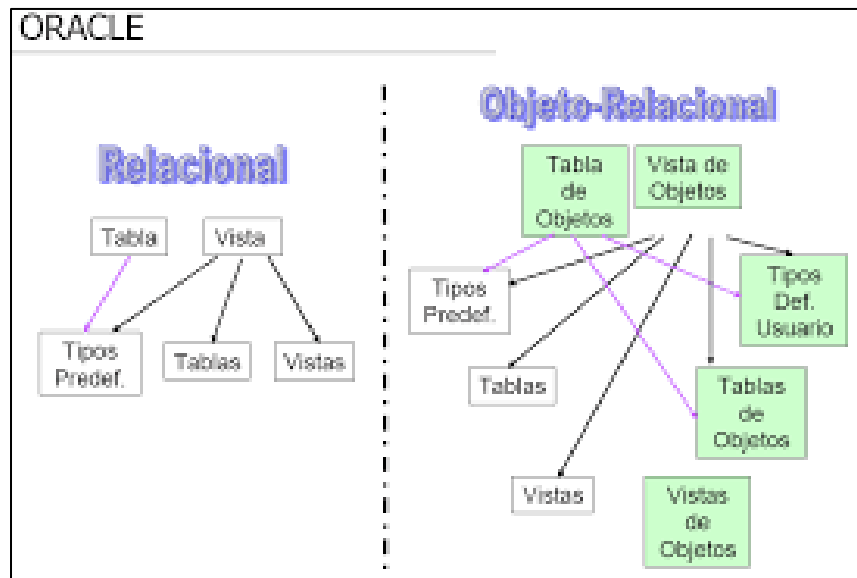
Un tipo de objeto consta de dos partes: especificación y cuerpo:

- Especificación: constituye la interface a las aplicaciones; aquí se declaran las estructuras de datos (atributos) y las operaciones (métodos) necesarios para manipular los datos.
- Cuerpo: define los métodos, es decir, implementa la especificación.



Características:

- En una especificación de tipo de objeto los atributos deben declararse antes que cualquiera de los métodos.
- Todas las declaraciones en la especificación del tipo son públicas. Sin embargo, el cuerpo puede contener declaraciones privadas, que definen métodos internos del tipo de objeto.
- El ámbito de las declaraciones privadas es local al cuerpo del objeto.



MODELO NoSQL

El término NoSQL se refiere a la denominación en inglés Not Only SQL. Plantea modelos de datos específicos de esquemas flexibles que se adaptan a los requisitos de las aplicaciones más modernas. Tienen un conjunto increíble de características y varios modelos que descubriremos más adelante.

Características comunes de bases de datos NoSQL:

- Consistencia eventual.
En la mayoría de sistemas NoSQL, no se implementan mecanismos rígidos de consistencia que garanticen que cualquier cambio llevado a cabo en el sistema distribuido sea visto, al mismo tiempo, por todos los nodos y asegurando, también, la no violación de posibles restricciones de integridad de los datos u otras reglas definidas.
- Flexibilidad en el esquema.
Los esquemas de datos son dinámicos: es decir, la escritura de los datos debe adaptarse a unas estructuras (o tablas) y tipos de datos predefinidos, en los sistemas NoSQL, cada registro puede contener una información con diferente forma cada vez, pudiendo almacenar solo atributos que interesen, facilitando el polimorfismo de datos bajo una misma colección de información.
- Escalabilidad horizontal.
Por escalabilidad se entiende la posibilidad de incrementar el rendimiento del sistema añadiendo, simplemente, más nodos (servidores) e indicando al sistema cuáles son los nodos disponibles.

- Estructura distribuida.

Generalmente los datos se distribuyen, entre los diferentes nodos que componen el sistema.

Ha dos estilos de distribución de datos:

- Particionado (Sharding).
- Réplica.

- Tolerancia a fallos y Redundancia.

Modelos de datos usados en sistemas NoSQL:

1. Base de datos de Documentos: Almacena la información como un documento, usado habitualmente para ello una estructura simple como JSON, BSON o XML y donde se utiliza una clave única para cada registro.
2. Bases de datos Clave-Valor: En este tipo de sistema, cada elemento está identificado por una clave única, lo que permite la recuperación de la información de forma muy rápida, información que suele almacenarse como un objeto binario.
3. Bases de datos orientado a grafos: Los datos se almacenan en estructuras grafo con nodos (entidades), propiedades (información entre entidades) y líneas (conexiones entre las entidades).
4. Base de datos Columnar (o Columna ancha): En vez de “tablas”, en las bases de datos de columna tenemos familias de columnas que, son los contenedores de las filas.
Este tipo de bases de datos se adecuan mejor a operaciones analíticas sobre grandes conjuntos de datos.
5. Bases de datos multivalor: Son sistemas interesantes que incorporan diferentes características multidimensionales y NoSQL para la clasificación y manejo de los datos.
6. Bases de datos orientada a objetos: Este tipo de bases están conformadas por objetos.
7. Bases de datos de Arrays: Sirven para trabajar colecciones de datos conocidas como raster data.

Empresas que utilizan este tipo de bases de datos:

- Amazon
- Google
- Twitter
- Facebook

Marcas de Bases de Datos NoSQL:

- MongoDB
- Apache Cassandra
- CouchDB
- Redis
- Neo4j

