

Modelo Orientado a Objetos (OODBMS)

El modelo Object-Oriented Database Management System combina las capacidades de las bases de datos con lo aprendido en la asignatura que llevamos en tercer semestre, que es Programación Orientada a Objetos (POO). Básicamente, los datos se convierten en objetos, tal como se definen en lenguajes como Java o C++, por lo que la información se representa mediante objetos que encapsulan tanto datos (atributos) como comportamientos (métodos).

En este modelo se incluyen conceptos como objeto, identidad, encapsulación, herencia y polimorfismo, entre otros. Estos principios permiten organizar mejor la información, reutilizar código y modelar situaciones del mundo real de una forma más natural. Además, los objetos pueden ser persistentes, es decir, almacenarse en la base de datos manteniendo su estructura.

Ventajas

- Permite manejar datos complejos como imágenes, audio, video, documentos, además de texto y números.
- Facilita la reutilización de código y el modelado del mundo real.
- Mayor flexibilidad en el diseño del sistema.
- Reduce la necesidad de convertir datos entre objetos y tablas (no requiere tanto mapeo objeto-relacional).
- Puede disminuir costos de mantenimiento gracias a la encapsulación.

Desventajas

- No existe un modelo de datos estandarizado universalmente para todos los OODBMS.
- Menor adopción y soporte comercial comparado con los DBMS relacionales.
- Mayor complejidad de implementación y administración.
- En algunos casos puede ser menos eficiente para consultas masivas que los sistemas relacionales tradicionales.

Finalmente, algunos casos de uso de este modelo son el diseño e ingeniería asistida por computadora, sistemas de información gráfica, aplicaciones científicas, simulaciones complejas, entre otras más.

Modelos NoSQL

Los métodos o bases de datos NoSQL son sistemas de almacenamiento de datos no relacionales que utilizan esquemas flexibles y modelos especializados (como clave-valor, documentos, columnas o grafos) para manejar grandes volúmenes de información de forma rápida y escalable.

1.Clave-Valor (Key-Value)

Es un tipo de base de datos NoSQL donde cada dato se almacena como un par clave-valor, en el que la clave es un identificador único definido por el programador y el valor puede ser cualquier tipo de información. Los datos se recuperan directamente mediante su clave.

Ventajas

- Alto rendimiento, con operaciones de lectura y escritura muy rápidas (generalmente $O(1)$ en promedio).
- Escalabilidad sencilla, ya que es fácil particionar los datos horizontalmente (sharding).
- Diseño simple y fácil de implementar.

Desventajas

- Consultas limitadas: solo se puede buscar por la clave, no por el contenido del valor.
- No maneja relaciones entre datos ni permite operaciones complejas como joins.
- Menor flexibilidad para consultas analíticas.

Se utiliza comúnmente en carritos de compra en tiempo real, almacenamiento de sesiones de usuario, caché, conteo de visitas y preferencias de usuario. Ejemplos de sistemas son Redis, Amazon DynamoDB y Riak.

2.Modelo Documental

Almacena datos en documentos semi-estructurados (JSON, BSON, XML) agrupados en colecciones. No requiere un esquema fijo (cada documento puede tener campos distintos).

Ventajas

- Flexibilidad extrema para cambios de estructura.

- Mapeo directo a objetos de programación
- Alto rendimiento en lecturas.

Desventajas

- Redundancia de datos (se duplica información para evitar joins)
- Mayor consumo de almacenamiento.

Se usa en Sistemas de gestión de contenidos (CMS), catálogos de e-commerce, perfiles de usuario.

3.Modelo de Grafos

Representa la información mediante nodos (entidades) y aristas (relaciones). Está diseñado específicamente para almacenar y navegar interconexiones complejas.

Ventajas

- Velocidad superior en consultas de relaciones profundas
- Visualización intuitiva de redes.
- No requiere índices complejos para conectar datos.

Desventajas

- Difícil de escalar horizontalmente (sharding complejo).
- Lenguaje de consulta específico (ej. Cypher) distinto a SQL.

Se utiliza comúnmente en redes sociales, motores de recomendación, detección de fraude, rutas logísticas.

4. Modelo Columnar (Wide-Column)

Organiza los datos por columnas en lugar de filas. Las columnas relacionadas se agrupan en "familias". Optimizado para leer y escribir grandes volúmenes de datos.

Ventajas

- Alta compresión de datos.
- Excelente rendimiento en consultas de agregación (sumas, promedios) sobre millones de registros.
- Escalabilidad masiva.

Desventajas

- Bajo rendimiento en actualizaciones de filas individuales.
- Diseño de esquema rígido orientado a la consulta (query-first).

Algunos casos de uso es Big Data, análisis en tiempo real, internet de las cosas (IoT), registros de eventos.

Bibliografía:

[1] K. H. E. David, "Modelo orientado a objetos," Apr. 21, 2020. <https://recursoseducativos.unam.mx/handle/123456789/21856>

[2] Amazon Web Services, "¿Qué es NoSQL?," AWS, 2026. <https://aws.amazon.com/es/nosql/>.

[3] S. Migani, C. Vera y MI Lund, "NoSQL: modelos de datos y sistemas de gestión de bases de datos", 2018. <https://sedici.unlp.edu.ar/handle/10915/67258>

[4] Utel Universidad, "Bases de datos NoSQL," *Material de clase: Administración de Base de Datos*, semana 4, 2012. https://gc.scalahed.com/recursos/files/r161r/w25213w/M1CDN112_S4_Bases_de_datos_NoSQL.pdf.

[5] M.Q. Esteban. Universidad Nacional de Educación Enrique Guzmán y Valle, "Diseño de sistemas de datos: Introducción a los sistemas," *Repositorio Institucional UNE*, 2018. <https://repositorio.une.edu.pe/server/api/core/bitstreams/c96596f0-909d-4e39-85b8-d99fa767a027/content>.