



Tarea 01

Modelos de bases de datos.

Antonio Manuel Zamora Ayala

1. Modelo orientado a objetos.

El Modelo de Datos Orientado a Objetos (*OODM*) es el soporte sobre el que descansa el Modelo de Base de Datos Orientada a Objetos (*OODBM*). Y como consecuencia tenemos el Sistema de Gestión de Bases de Datos Orientado a Objetos (*OODBMS*), el cual es quien maneja todo esto. [1]

En este modelo la información de la base de datos se agrupa en paquetes relacionados entre sí, es decir, los datos de cada registro se combinan en un objeto con todos sus métodos y atributos correspondientes. [2] Toda la información se agrupa y almacena directamente en el objeto, en vez de distribuirla en distintas tablas.

Los objetos de una clase también pueden relacionarse con otras clases, permitiendo formar redes. Los objetos simples también pueden combinarse para crear objetos más complejos.

A continuación, se mostrarán algunas ventajas y desventajas de este modelo:

Ventajas	Desventajas
Los conjuntos de datos complejos se pueden guardar y consultar de una manera rápida y sencilla.	La gran complejidad puede acarrear problemas de rendimiento y mantenimiento.
Los identificadores se asignan de forma automática a cada objeto.	Falta de un fundamento teórico, por lo que el significado exacto de modelo de datos orientado a objetos no está bien definido
El modelado de objetos es intuitivamente más cercano al mundo real.	Su implementación es mucho más compleja que el modelo de datos relacional.

Figura 1: Ventajas y desventajas del modelo orientado a objetos.

2. Modelos NoSQL.

Los modelos NoSQL (*No Only SQL*) deben su nombre al hecho de que son modelos **no relacionales**, y aunque también se pueden implementar/manejar con el lenguaje SQL (*Structured Query Language*), la mayoría de modelos NoSQL poseen un lenguaje de consulta de datos único, siendo un limitante para ciertas cuestiones [3]. A continuación se describirán los más comunes.

- **Clave-valor.** Se almacenan datos como un conjunto de pares clave-valor, siendo la clave un identificador único. A groso modo, puede representarse de la siguiente forma:

Clave (<i>ID único</i>)	Valor (<i>cualquier cosa</i>)
----------------------------------	--

Las bases de datos implementadas con un modelo de clave-valor, tienden a ser altamente divisibles, permitiendo el escalado horizontal (aumentando el número de recursos/servidores) a niveles que otros modelos no podrían alcanzar [4].

Algunas ventajas y desventajas que posee este modelo, son:

Ventajas	Desventajas
Alta velocidad y escalabilidad horizontal.	Las lecturas//búsquedas por medio del valor tienden a ser sumamente complejas.
Alto rendimiento para lecturas y escrituras por medio del ID (clave).	Es complicado darle mantenimiento, ya que no se sigue un control en la estructura de los datos.
Gran flexibilidad en los valores, ya que pueden ser cualquier cosa (texto, objetos, listas, etc...).	
Poseen una gran simplicidad al momento de acceder a los datos.	

Figura 2: Ventajas y desventajas del modelo clave-valor.

Y algunos casos de uso de las bases de datos de clave-valor, son:

- Administración de sesiones.
- Carro de compras.
- Almacenamiento en caché.
- Motor de almacenamiento de metadatos.

- **Documental.** Los modelos documentales o de documentos almacenan datos semiestructurados, un documento es un registro en una base de datos que almacena información sobre un objeto y cualquiera de los métodos asociados a él.

Los documentos almacenan datos en pares de *campo-valor*. Los valores pueden ser de varios tipos y estructuras, como cadenas, números, fechas, matrices u objetos. Los documentos se pueden almacenar en formatos como JSON, BSON, y XML [5];, siendo JSON (*notación de objetos de JavaScript*) el tipo de documento más común.

Algunas ventajas y desventajas de este modelo son las siguientes:

Ventajas	Desventajas
Facilitan las operaciones CRUD (<i>Create, Read, Update, Delete</i>).	Tienen un gran problema al momento de soportar las transacciones ACID (<i>Atomicidad, Consistencia, Aislamiento, Durabilidad</i>).
Se trabaja con datos semi-estructurados.	Son buenos para lecturas complejas, pero son ineficientes para filtrar información.
Poseen un esquema flexible, ya que no todos los documentos necesitan tener los mismos campos.	EL mantenimiento suele ser más complicado.
Permiten el escalamiento horizontal y la distribución de datos.	
Son una gran opción para bases de datos de propósito general.	

Figura 3: Ventajas y desventajas del modelo documental.

Algunos casos de uso, son:

- Administración de contenido web.
- Catálogos.

- **Orientado a grafos.** En estas bases de datos, se proporcionan modelos de grafos para representar las relaciones que existen entre los datos. Gracias a estas, los usuarios pueden hacer consultas transversales basadas en conexiones y aplicar algoritmos de grafos a fin de encontrar patrones, rutas, comunidades, factores influyentes, puntos únicos de fallo y otras relaciones. Esto permite un análisis más eficiente a gran escala con cantidades masivas de datos [6].

A continuación se muestran algunas ventajas y desventajas de este modelo orientado a grafos.

Ventajas	Desventajas
Poseen una representación basada en grafos, haciendo que sea más sencillo el modelado de datos complejos.	A medida que crece la complejidad de los datos y las relaciones, la gestión y optimización de la base de datos pueden volverse más complejas.
Permite añadir información de manera dinámica al crear nodos y relaciones sin tener que modificar el esquema existente.	Escalabilidad horizontal reducida.
Son de esquema flexible.	Poca optimización para grandes volúmenes de datos sin relación.
Las consultas/búsquedas son muy rápidas y escalables gracias a las relaciones entre nodos.	

Figura 4: Ventajas y desventajas del modelo orientado a grafos.

Algunos casos de uso de este modelo orientado a grafos, son:

- Detección de lavado de dinero.
- Redes sociales.
- Detección de fraudes con tarjetas de crédito.

- **Columnar.** Este modelo almacena los datos en columnas en lugar de en filas. También se denominan a veces almacenes «*orientados a columnas*» o «*de columna ancha*» y pueden alcanzar terabytes (o más) de almacenamiento de datos. Las bases de datos columnares aceleran el procesamiento de las consultas y a menudo se utilizan para macrodatos o consultas para analíticas de aprendizaje automático [7].

Entre las ventajas y desventajas que posee, se encuentran las siguientes:

Ventajas	Desventajas
Evaluación de grandes volúmenes de datos (Big Data).	No son buenas para el manejo de transacciones.
Gran escalabilidad horizontal.	Esquema no flexible.
Flexibilidad en las columnas al momento de añadir unas nuevas.	Están limitadas para relaciones complejas.
Gran funcionamiento para realizar cálculos sobre grandes volúmenes de datos (sumas, promedios, conteos).	

Figura 5: Ventajas y desventajas del modelo columnar.

Y algunos casos de uso son:

- Almacenes de datos.
- Recomendaciones y personalización.
- Analítica de datos.
- Monitoreo de sistemas.

Referencias

- [1] PowerData, “Conceptos básicos sobre modelo de datos orientado a objetos.” [Online]. Available: <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/conceptos-basicos-sobre-modelo-de-datos-orientado-a-objetos>
- [2] IONOS, “Base de datos orientada a objetos: el secreto mejor guardado de los modelos de bases de datos.” [Online]. Available: <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/>
- [3] Talend, “Sql vs nosql: Differences, databases, and decisions.” [Online]. Available: <https://www.talend.com/resources/sql-vs-nosql/>
- [4] AWS, “What is a key-value database?.” [Online]. Available: <https://aws.amazon.com/es/nosql/key-value/>
- [5] MongoDB, “What is a document database?.” [Online]. Available: <https://www.mongodb.com/es/resources/basics/databases/document-databases>
- [6] Oracle, “What is a graph database?.” [Online]. Available: <https://www.oracle.com/mx/autonomous-database/what-is-graph-database/>
- [7] P. Storage, “¿qué es una base de datos columnar? .” [Online]. Available: <https://www.purestorage.com/es/knowledge/what-is-a-columnar-database.html>