



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Bases de Datos

Grupo: 01

Tarea 01. Modelos de bases de datos

Profesor: FERNANDO ARREOLA FRANCO

Alumno: López Romero David Baruc

No. de Cuenta: 321018067

Semestre: 2026-2

Ciudad Universitaria, CDMX
Febrero 2026

1. Modelo orientado a objetos

Las Bases de Datos Orientadas a Objetos (OODBMS) surgieron para satisfacer la necesidad de almacenar estructuras de datos complejas que no se ajustaban bien al modelo relacional tradicional. En una base de datos que sigue este modelo, los componentes se almacenan como objetos y no como datos, tal como se hace en la programación orientada a objetos (POO).

Descripción

Los datos se almacenan como objetos, los cuales son instancias de una clase. Soportan características fundamentales de la POO como encapsulamiento, herencia y polimorfismo. A diferencia de las tablas relacionales, los objetos pueden contener tipos de datos complejos (como audio, video o geometrías) directamente.

Los manejadores de bases de datos orientados a objetos deben tomar en cuenta lo siguiente:

- Ser capaces de definir sus propios tipos de datos
- El tamaño de los datos puede ser muy grande
- La duración de las transiciones puede ser muy larga
- Recuperar rápidamente datos complejos
- Lenguajes de consulta de objetos, un ejemplo es OQL
- Mecanismos de seguridad basados en la noción de objeto
- Funciones para definir reglas deductivas

Ventajas y Desventajas

- **Ventajas:**
 - Elimina el "desajuste de impedancia" entre el lenguaje de programación y la base de datos.
 - Alto rendimiento en la gestión de datos complejos y relaciones jerárquicas.
- **Desventajas:**
 - Falta de un estándar universal robusto (como lo es SQL para las relacionales).
 - Curva de aprendizaje más elevada y menor soporte comercial en comparación con RDBMS.

Casos de Uso

Sistemas de Diseño Asistido por Computadora (CAD), Fabricación Asistida por Computadora (CAM), Sistemas de Información Geográfica (GIS) y sistemas multimedia.

2. Modelos NoSQL

El término NoSQL (*Not Only SQL*) engloba una variedad de tecnologías de bases de datos diseñadas para manejar grandes volúmenes de datos, con esquemas flexibles y alta escalabilidad horizontal.

2.1. Modelo Clave-Valor (Key-Value)

Descripción: Es el modelo NoSQL más simple. Almacena datos como un conjunto de pares, donde una clave única sirve como identificador para acceder a un valor asociado (que puede ser un string, un JSON, o un binario).

- **Ventajas:** Escalabilidad masiva y latencia extremadamente baja en operaciones de lectura/escritura simples.
- **Desventajas:** Capacidades de consulta limitadas (no se puede filtrar por el contenido del valor fácilmente).
- **Casos de uso:** Caché de sesiones, carritos de compras, preferencias de usuario.
- **Ejemplos:** Redis, Amazon DynamoDB.

2.2. Modelo Documental

Descripción: Almacena y recupera documentos, generalmente en formatos como JSON, BSON o XML. Cada documento es autodescriptivo y puede tener una estructura diferente a los demás en la misma colección.

- **Ventajas:** Esquema flexible (schema-less) que permite iteraciones rápidas en el desarrollo; mapeo natural a objetos de código.
- **Desventajas:** Las transacciones complejas que involucran múltiples documentos pueden ser menos eficientes que en SQL.
- **Casos de uso:** Gestión de contenidos (CMS), catálogos de e-commerce, plataformas de blogging.
- **Ejemplos:** MongoDB, CouchDB.

2.3. Modelo de Grafos

Descripción: Utiliza estructuras de grafos con nodos (entidades), aristas (relaciones) y propiedades para representar y almacenar los datos. Las relaciones son ciudadanos de primera clase. Ofrecen una experiencia de búsqueda mas eficiente entre relaciones, esto respecto a las bases de datos relacionales.

- **Ventajas:** Rendimiento superior para consultas que implican relaciones profundas o complejas entre datos conectados.

- **Desventajas:** Dificultad para escalar horizontalmente (sharding) debido a la interconexión de los datos.
- **Casos de uso:** Redes sociales, motores de recomendación, detección de fraude, análisis de redes.
- **Ejemplos:** Neo4j, Amazon Neptune, FlockDB, InfiniteGraph.

2.4. Modelo Columnar (Wide-Column)

Descripción: Organiza los datos en filas y columnas, pero a diferencia de las relacionales, los nombres y formatos de las columnas pueden variar de una fila a otra. Están optimizadas para consultas sobre grandes conjuntos de datos.

- **Ventajas:** Alta eficiencia en compresión de datos y consultas de agregación en Big Data. Escrituras muy rápidas.
- **Desventajas:** Diseño de esquema complejo orientado a la consulta (query-driven design).
- **Casos de uso:** Análisis de series temporales, registros de eventos (logs), Internet de las Cosas (IoT).
- **Ejemplos:** Apache Cassandra, HBase.

3. Resumen Comparativo

Cuadro 1: Comparación de Modelos de Bases de Datos

Modelo	Ventaja Principal	Desventaja Principal	Caso de Uso
Orientado a Objetos	Manejo de datos complejos y herencia	Falta de estandarización	CAD/CAM
Clave-Valor	Velocidad y simplicidad	Consultas limitadas	Caché/Sesiones
Documental	Flexibilidad de esquema	Transacciones complejas	CMS/Catálogos
Grafos	Relaciones complejas	Escalabilidad horizontal difícil	Redes Sociales
Columnar	Manejo de Big Data	Diseño de esquema rígido	Logs/IoT

Referencias

- [1] Bases de datos NoSQL: características y tipos”, Stackscale, 02-ene-2023.

- [2] E. D. K. Hernández, “Modelo Orientado a Objetos”, Unam.mx. [En línea]. Disponible en: https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/2731/mod_resource/content/1/UAPA-Modelo-Orientado-Objetos/index.html. [Consultado:09-feb-2026].
- [3] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA, USA: Pearson, 2016.
- [4] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. McGraw-Hill Education, 2019.