



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

BASES DE DATOS

1644

GRUPO: 01

ING. FERNANDO ARREOLA FRANCO

TAREA 03

**MODELO OBJETO-RELACIONAL, MODELO NO
SQL, MODELO ORIENTADO A OBJETOS**

SÁNCHEZ MANJARREZ ANDREW

MODELO ORIENTADO A OBJETOS: Las bases de datos relacionales, iniciaron año 2000 eran las más utilizadas en la mayoría de los ámbitos de aplicación. La estructura de datos básica que ofrece, la tabla relacional, es apropiada para muchas aplicaciones habituales. Sin embargo, existen casos de uso en los que presenta serios inconvenientes prácticos, generalmente si se requiere gestionar datos muy complejos o no convencionales (imágenes, documentos...), para los que las estructuras relacionales resultan muy complejas e ineficientes. Algunos ejemplos de gran importancia práctica son las bases de datos multimedia, las bases de datos científicos y los sistemas de apoyo al diseño industrial (cad/cam).

Las bases de datos orientadas a objetos intentan dar respuesta a estos problemas, incorporando las siguientes características: 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. Permiten la extensibilidad con nuevos tipos de datos complejos, permitiendo incorporar operaciones arbitrarias sobre ellos. Estas características han motivado el desarrollo de numerosos sistemas orientados a objetos. Juntamente al avance vertiginoso de desarrollo de WebApps que requieren estructuras de datos muy flexibles.

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 programación orientados a objetos, como C++ o Java. Esto 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. (en las BDR es necesario usar dos lenguajes: a) uno de programación para la aplicación, y b) sql para el acceso a la base de datos, lo que implica realizar costosas conversiones. Disponer de nuevas características para el modelado de datos complejos (BDOR. Las últimas versiones del estándar sql, a partir de sql99, también incluyen muchas de estas funcionalidades. Entre ellas se incluyen la posibilidad de almacenar en tablas relacionales grandes objetos binarios (denominados «blobs», por Binary Large OBjects), como imágenes, sonido o video; las herramientas de gestión de documentos (índices y operadores específicos para buscar texto, soporte de xml); y otras extensiones para soportar datos geográficos)

VENTAJAS DEL MODELO ORIENTADO A OBJETOS:

Lenguajes de Consulta: Entre los aspectos más positivos que poseen dichas bases de datos se encuentra su lenguaje de consulta. Estas implementan un sistema declarativo de permiten encontrar y recuperar los objetos de la base de datos. Como ya mencionamos, estas bases de datos son capaces de funcionar con varios tipos de lenguajes de programación conocidos.

Persistencia transparente: Esta es una característica muy importante. Se refiere a la posibilidad de la base de datos de utilizar uno o varios objetos a través del lenguaje de programación para obtener análisis o manipulación de la data.

Transacciones ACID: Estas bases de datos tienen un alto nivel de confiabilidad en la ejecución de transacciones ACID (atomicidad, consistencia, aislamiento y durabilidad). Esto significa que las transacciones se cumplen o terminan únicamente cuando no existen conflictos con los datos. También los usuarios pueden abortar las acciones en caso de que se presente algún error en el sistema de la base de datos.

Almacenamiento en cache: Las bases de datos orientadas a objetos tienen un mecanismo de almacenamiento en cache que crea replicas parciales de la base de datos. El almacenamiento en caché es un método de almacenamiento y resguardo de datos que permite minimizar el acceso al disco o el flujo de tráfico de la red. Esto permite que las aplicaciones accedan a objetos desde la memoria del programa en lugar del utilizar el almacenamiento en disco.

También es importante mencionar que estas bases de datos tienen amplias capacidades para recuperación de datos, manejo complejo de altos volúmenes de datos. Sus características le proporcionan gran rendimiento y un manejo mínimo de código.

DESVENTAJAS DE LAS BASES DE DATOS ORIENTADAS A OBJETOS

Al igual que diversas bases de datos NoSQL, carecen de madurez en el desarrollo. Esto lleva a un mayor riesgo de sostenibilidad de la base de datos en el largo plazo. No hay criterios claros de estandarización, ya que estas apenas están reemergiendo en el mercado digital.

Al tener relativamente poco tiempo en el mercado, existe poca experiencia y documentación sobre proyectos que las implementen. Esto pudiese extender los tiempos de desarrollo de un proyecto con este modelo de bases de datos.

Sin duda también representaría gastos de capacitación adicional para los equipos de desarrolladores ya que no existe una familiaridad en los entornos del mercado digital con el paradigma de objetos.

MODELO RELACIONAL: modelo de organización y gestión de bases de datos consistente en el almacenamiento de datos en tablas compuestas por filas, o tuplas, y columnas o campos. Se distingue de otros modelos, como el jerárquico, por ser más comprensible para el usuario inexperto, y por basarse en la lógica de predicados para establecer relaciones entre distintos datos. Surge como solución a la creciente variedad de los datos que integran el concepto como una colección de tablas (relaciones).

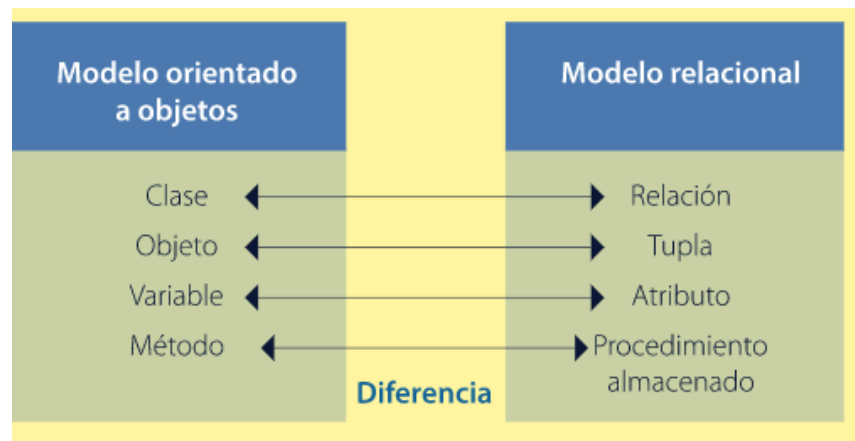
Existen 4 tipos básicos de restricciones de integridad:

los datos requeridos (los campos o columnas siempre deben poseer un atributo o un valor)

la comprobación de validez (las tablas deben contener solo los datos correspondientes a la correspondiente relación definida por cada tabla)

las integridades de entidad y referencial (las primeras aseguran que las claves primarias posean un valor único para cada tupla, y las segundas que las claves principales y las externas mantengan su integridad)

Reglas de integridad: reglas que garantizan la integridad de los datos, es decir, la correspondencia plausible de los datos con la realidad.



MODELO NoSQL: Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala.

Las bases de datos NoSQL se adaptan perfectamente a muchas aplicaciones modernas, como dispositivos móviles, web y juegos, que requieren bases de datos flexibles, escalables, de alto rendimiento y altamente funcionales para proporcionar excelentes experiencias de usuario.

Flexibilidad: las bases de datos NoSQL generalmente ofrecen esquemas flexibles que permiten un desarrollo más rápido y más iterativo. El modelo de datos flexible hace que las bases de datos NoSQL sean ideales para datos semiestructurados y no estructurados.

Escalabilidad: las bases de datos NoSQL generalmente están diseñadas para escalar usando clústeres distribuidos de hardware en lugar de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones en segundo plano, como un servicio completamente administrado.

Alto rendimiento: la base de datos NoSQL está optimizada para modelos de datos específicos y patrones de acceso que permiten un mayor rendimiento que el intento de lograr una funcionalidad similar con bases de datos relacionales.

Altamente funcional: las bases de datos NoSQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.

REFERENCIAS:

Bases de datos NoSQL | Bases de datos orientadas a objetos. (2020). Retrieved 15 October 2020, from <https://www.grapheverywhere.com/bases-de-datos-orientadas-objetos/>

Modelo relacional en la gestión de bases de datos. (2020). Retrieved 15 October 2020, from <https://blog.es.logicalis.com/analytics/conceptos-basicos-del-modelo-relacional-en-la-gestion-de-bases-de-datos>

Bases de datos no relacionales | Bases de datos de gráficos | AWS. (2020). Retrieved 15 October 2020, from <https://aws.amazon.com/es/nosql/>