



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA – CIUDAD UNIVERSITARIA

MATERIA. BASES DE DATOS

PROFESOR. FERNANDO ARREOLA FRANCO

TAREA 1. MODELO ORIENTADO A OBJETOS Y MODELOS NOSQL

ALUMNO. BORBOA CASTILLO, CARLOS ALFONSO

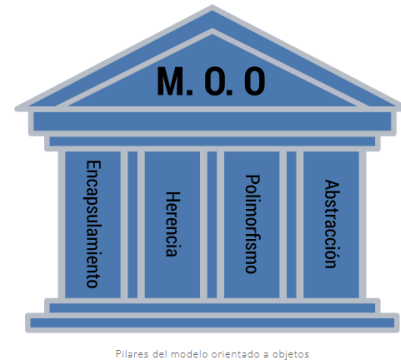
GRUPO 1



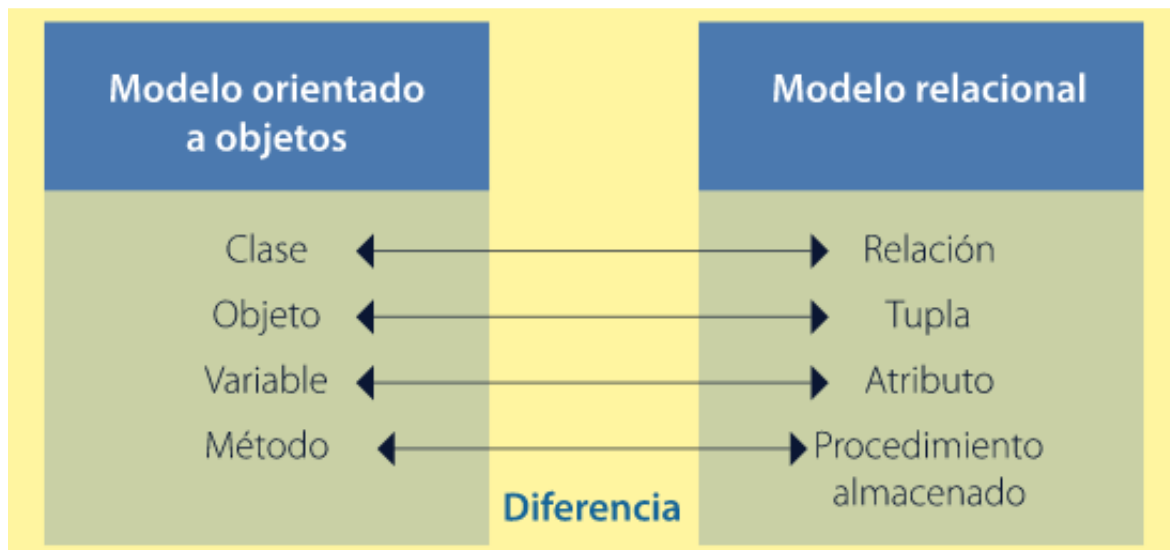
MODELO ORIENTADO A OBJETOS

La tecnología de bases de datos vive un momento de lenta transición del modelo relacional a otros modelos. 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.

Los administradores de base de datos (DBMS por sus siglas en inglés) evolucionan con el afán de satisfacer nuevos requerimientos tecnológicos y de información. Aunque los DBMS relacionales (RDBMS) son actualmente líderes del mercado y brindan las soluciones necesarias a las empresas comerciales, existen aplicaciones que necesitan funciones con las que no cuentan.



Los sistemas multimedia como los geográficos y de medio ambiente, los de gestión de imágenes y documentos y los de apoyo a las decisiones, necesitan de modelos de datos complejos, difíciles de representar como tuplas de una tabla. En general, estas aplicaciones necesitan manipular objetos y los modelos de datos deben permitirles expresar su comportamiento y las relaciones entre ellos.



Como se puede visualizar, para superar los retos antes mencionados, las bases de datos están tomando varias tendencias. Para ser precisos, se están auxiliando de los lenguajes de programación orientados a objetos.

Un sistema de bases de datos orientado a objetos es aquél que integra dicho paradigma de programación, precisamente para cumplir con los objetivos de la manipulación y almacenamiento de los datos.

DEFINICIÓN

La orientación a objetos representa el mundo real y resuelve problemas a través de objetos, ya sean tangibles o digitales. Este paradigma tecnológico 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.



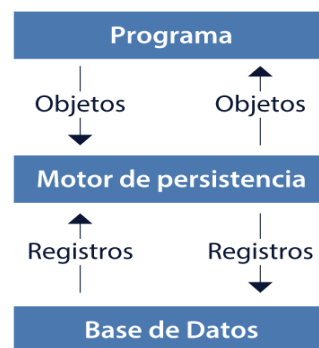
PERSISTENCIA EN EL MODELO ORIENTADO A OBJETOS

La persistencia es una característica necesaria de los datos en un sistema de bases de datos. Recordemos que consiste en la posibilidad de recuperar datos en el futuro. Esto implica que los datos se almacenan a pesar del término del programa de aplicación.

En resumen, todo administrador de base de datos brinda persistencia a sus datos. En el caso de los sistemas de gestión de base de datos orientada a objetos (OODBMS por sus siglas en inglés), la persistencia implica almacenar los valores de atributos de un objeto con la transparencia necesaria para que el desarrollador de aplicaciones no tenga que implementar ningún mecanismo distinto al mismo lenguaje de programación orientado a objetos.

Lo anterior tiene como ventaja que, no es necesario el uso de dos lenguajes de programación para construir una aplicación; es decir, actualmente, el desarrollo de aplicaciones se hace con lenguajes de programación orientada a objetos almacenando datos en bases relacionales, por lo que el desarrollador debe utilizar un lenguaje para la aplicación (Java, PHP, C++) y otro para la base de datos (SQL).

Los sistemas de bases de datos orientados a objetos parecen ser la tecnología más prometedora para los próximos años, aunque carecen de un modelo de datos común y de fundamentos formales, además de que su comportamiento en seguridad y manejo de transacciones no están a la altura de los programas actuales de administradores de bases de datos.



Los administradores de bases de datos orientadas a objetos también deben contar con un lenguaje que puede realizar cualquier procesamiento. En este sentido, lo más común es que los OODBMS integren lenguajes computacionalmente completos dentro de la base de datos. Estos pueden ser los que ya existen en el mercado y que se usan como lenguajes aplicación general (Java, C++, etc.).

| Clase | Objetos | Atributos/datos |
|----------|--------------|--------------------------|
| Empleado | Juan Pérez | Edad: 25 |
| | | Puesto: Psicóloga social |
| | | Salario: 8000 |
| | María Suárez | Edad: 23 |
| | | Puesto: Pedagoga |
| | | Salario: 15 000 |



ALMACENAMIENTO USANDO EL
MODELO ORIENTADO A OBJETOS

MODELOS NOSQL

Una base de datos (BBDD) es un conjunto almacenado de información para ser procesada, por su parte, los sistemas gestores de bases de datos son las aplicaciones de software que permiten almacenar y acceder a la información. Los principales tipos de BBDD son:

Relacionales. La información que almacena la BBDD está relacionada entre sí. Los datos relacionados son almacenados en tablas que constan de varios campos.

No relacionales. Los datos no tienen por qué estar relacionados entre sí y, por lo tanto, no tienen que almacenarse en estructuras fijas como las tablas del modelo de base de datos relacional.

CARACTERÍSTICAS Y VENTAJAS DE NOSQL

Las Bases de Datos NoSQL (Not Only SQL) pertenecen al modelo no relacional. Las principales características y ventajas de este tipo son:

SQL no es el lenguaje de consulta o modificación de datos principal, aunque sí lo soportan, de ahí el nombre no solo SQL. Los datos no tienen que almacenarse en tablas. Generalmente, su arquitectura es distribuida, almacenándose la información en más de una máquina del sistema. Por lo tanto, los sistemas que las soportan tienen una mayor escalabilidad horizontal (a mayor número de nodos mayor rendimiento) y también mayor tolerancia ante fallos en los distintos nodos.

Son más eficientes en el procesamiento de los datos que las BBDD relacionales, por eso, son la elección para aplicaciones que hacen un uso intensivo de éstos. Utilizan lo que se conoce como consistencia eventual, que se refiere a que los cambios realizados en los datos serán replicados a todos los nodos del sistema, lo cual aumenta el rendimiento de estos sistemas en contraposición a las propiedades ACID de las BBDD relacionales.

PUNTOS POSITIVOS Y NEGATIVOS DE NOSQL

Los Sistemas de Gestión de Bases de Datos NoSQL no contemplan por definición la atomicidad de las instrucciones, es decir, cuando una operación sobre los datos consta de varios pasos, no se tienen que ejecutar todos, cosa que sí sucede en los modelos relacionales (transacciones completas). Hay algunas BBDD NoSQL que contemplan la atomicidad.

Los gestores NoSQL no contemplan obligatoriamente la consistencia o integridad de la BBDD, esto quiere decir que no se comprueba que la operación a ejecutar sobre los datos se pueda completar desde un estado de la Base de Datos válido a otro válido. Al utilizar

el mecanismo de consistencia eventual se puede dar el caso de que la misma consulta a diferentes máquinas del sistema produzca resultados diferentes porque las modificaciones de la BBDD aún no han sido replicadas a todos los nodos.

Estas BBDD utilizan sus propios lenguajes de consulta de datos y APIs, por lo que no tienen una gran interoperabilidad (por ejemplo, dificultad de migraciones de una BBDD a otra, integración con aplicaciones, consultas heredadas en SQL, etc.). No hay estandarización para este tipo de BBDD, algo que sí es un punto fuerte de las relacionales.

Las Bases de Datos NoSQL funcionan ampliamente en máquinas Linux, pero no existe en general soporte a otros Sistemas Operativos. Las interfaces de gestión de estas BBDD no son intuitivas ni sencillas y en algunos casos carecen de ellas, gestionándose directamente desde consola de comandos.

¿CUÁNDO ES RECOMENDABLE UTILIZAR UNA BBDD NOSQL?

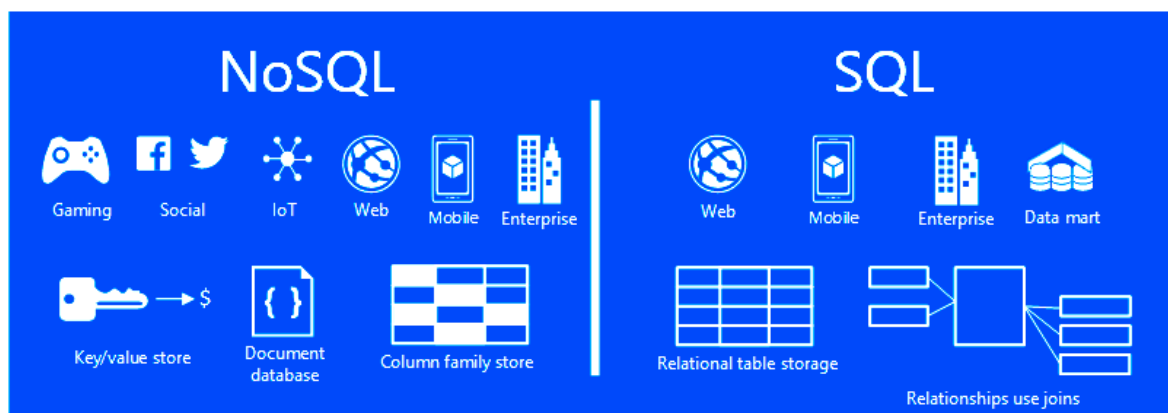
Cuando se necesita una BBDD para una aplicación que hace una consulta o lectura intensiva de grandes cantidades de datos. También es recomendable cuando no hay la necesidad de que los datos sean consistentes o si los datos a almacenar no tienen una estructura fija. Cabe resaltar que, una misma aplicación puede usar una BBDD relacional y una BBDD NoSQL y guardar cosas diferentes en cada una de ellas.

EJEMPLOS DE USO DE ESTE TIPO DE BBDD

- Amazon.
- Facebook.
- Google.

GESTORES DE BBDD NOSQL

- Redis
- MongoDB
- CouchDB



BIBLIOGRAFÍA

Atkinson, M., Bancilhon, F., DeWitt, D., Dittrich, K., Maier, D. y Zdonik, S. "*The Object Oriented Database System Manifesto*". Publicado en Kyoto, First International Conference on Deductive. Editorial Morgan Kaufmann, 1992, pp. 223-40

Date, C. J. "*Introducción a los sistemas de bases de datos, séptima edición*". Publicado en Ciudad de México. Editorial Pearson Educación. 2001.

Hughes, J. G. "*Bases de datos. Modelos, lenguajes, diseño*". Publicado en la Ciudad de México. Editorial Oxford University Press. EL año de publicación es 1999.

Johnson, J. L. "*Object oriented databases*". Publicado en Nueva York. Editorial Prentice Hall. El año de la publicación del documento es 1991.

Bertino, E. y Martino, L. "*Sistemas de bases de datos orientadas a objetos. Conceptos y arquitectura*". Publicado en Wilmington, Delaware. Editorial Addison 1995.

Silberschatz, A., Korth, H. F. y Sudarshan, S. "*Fundamentos de bases de datos*". Publicado en Madrid. Editorial McGraw-Hill. Año de publicación es 2006.