



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

BASES DE DATOS

ING. FERNANDO ARREOLA FRANCO

GRUPO: 01

ARCE HERNÁNDEZ CHRISTIAN ALEXIS

NO. DE CUENTA: 314159993

SEMESTRE 2021-1

MODELO ORIENTADO A OBJETOS.

El modelo de base de datos orientada a objetos agrupa la información en paquetes relacionados entre sí: los datos de cada registro se combinan en un solo objeto, con todos sus atributos. De esta manera, toda la información está disponible en el objeto, ya que sus datos quedan agrupados en lugar de distribuidos en diferentes tablas. En los objetos no solo pueden guardarse los atributos, sino también los métodos, lo que refleja la afinidad de estas bases de datos con los lenguajes de programación orientados a objetos: al igual que en estos, cada objeto presenta un conjunto de acciones que pueden llevarse a cabo.

Los objetos se dividen a su vez en clases. Más concretamente, un objeto es una unidad concreta de una clase abstracta, lo que crea una jerarquía de clases y subclases. Dentro de esta estructura, las subclases adoptan las propiedades de las clases superordinadas y las complementan con sus propios atributos. Al mismo tiempo, los objetos de una clase también pueden relacionarse con otras clases, lo que rompe la jerarquía estricta y permite formar redes. Los objetos simples también pueden combinarse para crear objetos más complejos.

Para gestionar los diversos objetos, el SGBD orientado a objetos correspondiente asigna automáticamente un código de identificación único a cada registro, que permite recuperar los objetos una vez que se han guardado.

MODELO OBJETO-RELACIONAL.

El modelo de base de datos objeto-relacional integra los conceptos de la tradicional base de datos relacional y los conceptos de paradigma de objetos que se utiliza en la programación orientada a objetos (POO).

El objetivo de este concepto es poder aplicar la tecnología madura de bases de datos relacionales sobre la organización de los datos complejos es decir datos de texto e imagen, mapas, datos en el rango de audio etc. Las bases de datos Objeto-relacional son compatibles con estos objetos de datos y las operaciones de mayor complejidad.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

En una base de datos objeto relacional los dominios de dicha base de datos ya no son sólo atómicos por esta razón no cumplen la 1FN debido a que las tuplas también pueden ser una relación, que llevará a la creación de una relación de relaciones es así que no se puede aplicar el concepto de normalización. Esto porque ni siquiera se puede aplicar la primera forma normal y como consecuencia ni la segunda, ni tercera etc. De este modo, se genera la posibilidad de guardar objetos más complejos en una sola tabla con referencias a otras relaciones, con lo que se acerca más al paradigma de programación orientada a objetos.

MODELO NO SQL.

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/modificación de datos principal, aunque sí lo soportan, de ahí el nombre No Sólo 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 estos (*“streaming”*, etc.).

Utilizan lo que se conoce como consistencia eventual que consiste en 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 (*“Atomicity, Consistency, Isolation and Durability”* – Atomicidad, Consistencia/Integridad, Aislamiento y Durabilidad).

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 (por ejemplo, no violación de ninguna restricción de tipos de datos o reglas).

Referencias:

FCA UNAM. (2020). Modelo Orientado a Objetos. SUAyED. Recuperado de:

https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html

Digital Guide. (2020). Bases de Datos Orientada a Objetos. IONOS. Recuperado de:

<https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/>

Universidad Internacional de la Rioja. (2020). Bases de Datos NO SQL. Fundacion UNIR. Recuperado de:

<https://www.unir.net/ingenieria/revista/noticias/bases-de-datos-nosql/549204809132/>