

ASPECTOS BÁSICOS EN EL MODELADO DE DATOS

AUTOR: MAX JOSÉ BERMÚDEZ LEÓN

DICIEMBRE: 2020



San Marcos

Introducción

Hoy en día las empresas manejan una gran cantidad de datos. Cualquier empresa que se precie debe tener almacenados todos estos datos en una base de datos para poder realizarlos mediante una aplicación profesional; sin esta funcionalidad resultaría imposible tratar y manejar en su totalidad los datos que lleva a cabo la empresa y se perdería un tiempo y un dinero muy valiosos. Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos, en donde lo más importante son LOS MODELOS DE DATOS.

Si los modelos no son definidos apropiadamente, podemos tener muchos dolores de cabeza al momento de ejecutar consultas a la base de datos para tratar de obtener algún tipo de datos. No importa si nuestra base de datos tiene sólo 20 registros, o algunos cuantos miles, es importante asegurarnos que nuestra base de datos está correctamente diseñada para que tenga eficiencia y que se pueda seguir utilizando por largo del tiempo.

En esta lectura se mencionarán algunos principios básicos del diseño de base de datos y se tratarán algunas reglas que se deben seguir cuando se crean bases de datos. Dependiendo de los requerimientos de la base de datos, el diseño puede ser algo complejo, pero con algunas reglas simples que tengamos en la cabeza será mucho más fácil crear una base de datos perfecta para nuestro siguiente proyecto.



Contenido

Introducción.....	1
Modelado de datos.....	3
Datos e información.....	5
Cualidades de la información.....	6
Modelos y tipos de base de datos.....	8
Modelo de datos.....	8
Importancia del modelo de base de datos.....	9
Tipos de modelo de base de datos.....	10
Modelo Jerárquico.....	11
Modelo de Red.....	13
Modelo Relacional.....	14
Modelo Orientado a objetos.....	17
Modelo Transaccional.....	20
Modelo de datos.....	22
Detallando el Modelado de datos.....	22
Conclusiones y recomendaciones.....	28
Referencias bibliográficas.....	29

Un modelo de datos es un conjunto de conceptos utilizados para organizar los datos de interés y describir su estructura en forma comprensible para un sistema informático. Cada modelo de datos provee mecanismos de estructuración, que permiten definir nuevos tipos de datos a partir de tipos elementales predefinidos.

Modelado de datos

conoceremos y comprenderemos las entidades y las relaciones en las diferentes bases de datos, comprensión que conduce al modelado que es la parte más importante del proceso de diseño de la base de datos jerárquica, de red, relacional o bien orientada a objetos.

El diseño de una base de datos no es un proceso sencillo.

La complejidad de la información y la cantidad de

requisitos de los sistemas de información hacen que sea complicado. Por este motivo, cuando se diseñan bases de datos es conveniente aplicar la conocida estrategia de dividir para vencer.

Por lo tanto, conviene descomponer el proceso del diseño en varias etapas; en cada una se obtiene un resultado intermedio que sirve de partida para la etapa siguiente, y en la última etapa se obtiene el resultado requerido. De este modo no hace falta resolver toda la problemática que plantea el diseño, sino que en cada etapa se afronta un sólo tipo de subproblema. Así se divide el problema y, al mismo tiempo, se simplifica el proceso.

La mayoría de los métodos de modelado de datos proporcionan algún método de mostrar gráficamente las entidades y relaciones. Aprenderemos que existen varios estilos y diferentes de diagramas Entidad-Relación (ER).

Los diagramas E-R tienen los siguientes objetivos:

- Modelar las necesidades de información de una organización.
- Identificar entidades y sus relaciones.
- Proporcionar un punto de partida para la definición de datos (diagramas de flujo de datos).
- Proporcionar una excelente fuente de información para los programadores de

aplicaciones, así como para los administradores de bases de datos y de sistemas.

- Crear un diseño lógico de la base de datos que se puede convertir en un esquema físico.

Los diagramas ER se relacionan con los diagramas de estructura de datos (DSD), que se centran en las relaciones de los elementos dentro de las entidades, en lugar de las relaciones entre las entidades mismas. Los diagramas ER a menudo se combinan con los diagramas de flujo de datos (DFD), que trazan el flujo de la información para procesos o sistemas.

El diseño de bases de datos es el proceso que determina su organización y estructura. Este diseño se realiza, generalmente, en tres pasos:

Diseño conceptual: obtenemos una estructura de la información de la futura BD independiente de la tecnología a emplear. Aún no sabemos qué tipo de base de datos relacional, orientada a objetos, jerárquica. Así pues, el diseño conceptual nos permite concentrarnos únicamente en la problemática de la estructuración de la información, sin tener que preocuparnos de resolver, en paralelo, aspectos tecnológicos. El resultado del diseño conceptual se expresa mediante algún modelo de datos de alto nivel. Uno de los más empleados es el modelo entidad relación, que abreviaremos con ER.

Diseño Lógico: partimos del diseño conceptual que se adaptará a la tecnología a emplear. Más concretamente, es preciso que se ajuste al modelo del SGBD con el que implementaremos la base de datos. Si se trata de un SGBD relacional, obtendremos un conjunto de relaciones con sus atributos, claves primarias y claves foráneas. Partimos del hecho de que hemos resuelto la problemática de la estructuración de la información en un ámbito conceptual, y nos concentraremos en las cuestiones tecnológicas relacionadas con el modelo de base de datos.

Diseño Físico: transformamos la estructura obtenida en el diseño lógico con el objetivo de conseguir una mayor eficiencia; además, lo completamos con aspectos de implementación física que dependerán del SGBD. Si se trata de una base de datos relacional, la

transformación de la estructura consiste en: tener almacenada alguna relación que será la combinación de varias relaciones obtenidas en el diseño lógico, dividir una relación en varias, añadir algún atributo calculable a una relación, etc. Los aspectos de implementación física a completar consisten normalmente en la elección de estructuras físicas de implementación de las relaciones, la selección del tamaño de las memorias intermedias (buffers) o de las páginas, etc.

La fase inicial del diseño de la base de datos física es la traducción del modelo de datos lógico global a una forma que puede ser implementada.

La siguiente fase, diseña el archivo de organización y métodos de acceso que puedan usarse para almacenar, analizar las transacciones que se ejecutarán en la base de datos, la elección conveniente de archivos de organización basados en estos análisis, añadir índices primarios y secundarios, introducir y controlar la redundancia para mejorar el funcionamiento y, finalmente, estimar el espacio del disco necesario para su implementación.

El objetivo del diseño físico es encontrar un esquema interno que soporte la estructura conceptual y los objetivos del diseño lógico con la máxima eficiencia de los recursos suministrados por la máquina.

Datos e información

Los datos son información codificada, lista para ser introducida y procesada por un computador. Es decir, los datos no son más que una forma de representar información.

Como resultado del procesamiento de datos tenemos información. Una vez que han sido procesados dichos datos, se muestran de modo inteligible.

Los datos son la información (valores o referentes) que recibe el computador a través de distintos medios, y que es manipulada mediante el procesamiento de programas. Los datos pueden ser: estadísticas, números, descriptores, que por separado no tienen relevancia para



los usuarios del sistema, pero que en conjunto pueden ser interpretados para obtener una información completa y específica.

En los lenguajes de programación, crear y organizar los algoritmos que todo sistema informático o computacional requiere, los datos son las características puntuales de las entidades sobre las cuales operan dichos algoritmos. Es decir, son la entrada inicial, para procesar y componer la información.

Los datos son importantes para la rama de la computación denominada estructura de datos que estudia la forma particular de almacenamiento de la información en porciones mínimas para lograr una posterior recuperación eficiente.

En informática se entiende por información al conjunto de datos ordenados, secuenciados o procesados por un algoritmo de programación, que recompone un referente, como un hecho concreto o algún sentido real.

La recuperación de la información a partir de los paquetes o conjuntos de datos es el objetivo final de la computación, dado que los sistemas informáticos codifican y representan la información a través de distintos mecanismos y lenguajes para comunicarse entre sí de manera eficaz y eficiente.

En informática, información es un término muy usado y valorado, debido a que ella se introduce en las computadoras en forma de datos y los mismos son manipulados para que generen distintas soluciones a diferentes problemas.

Un dato es, en general, una expresión que indica las cualidades de los diferentes comandos sobre los que un algoritmo trabaja.

Cualidades de la información

Para que una información sea útil es necesario que facilite, al responsable de la toma de decisiones, una idea clara y completa de la situación, en forma tal que sus decisiones tengan el fundamento objetivo óptimo posible.

Las cualidades de la información son:

Precisión: la información debe ser precisa, se mide en nivel de detalle y discernimiento. Se han vendido 39 artículos es más preciso que decir vendimos varias docenas de artículos. La precisión a exigir depende la aplicación. La falta de precisión es un defecto como un exceso de ella. Decir a un carpintero que la altura de una mesa es de 81.473245 cm., es un exceso de precisión.

Exactitud: La información debe ser exacta. Para ello se mide en términos de porcentaje de error. Es una medida de alejamiento de la realidad. No obtenemos exactitud suficiente partiendo de datos incorrectos o erróneos, podremos corregir algunos pequeños errores o aplicar filtros a datos no válidos.

Oportunidad: La información tiene que ser oportuna. Debe estar disponible para el usuario en el momento, en el lugar indicado y en la calidad justa para resolver un problema. El usuario debe actuar antes de que la realidad haya sufrido un cambio que anule la acción.

Integridad: La información debe ser completa. Aun cuando la integridad al 100 % es inalcanzable en la mayoría de las aplicaciones, conviene en todo caso que la información sea tan completa como sea posible.

Significatividad: La información debe ser clara y relevante. Es importante no forzar la comprensión del destinatario. Cualquier ayuda gráfica, visual, auditiva, o algún tipo que añada facilidad y rapidez a la recepción de la información debe ser considerada.

Recordemos que información es el resultado del procesamiento de datos.



En sentido general, la información es un conjunto organizado de datos procesados, que constituyen un mensaje sobre un determinado ente o fenómeno. De esta manera, si organizamos datos sobre un país como: número de habitantes, densidad de población, nombre del presidente, etc. y escribimos el capítulo de un libro, decimos que él contiene información sobre ese país.

Cuando tenemos que resolver un determinado problema o tomar una decisión, empleamos diversas fuentes de información (como el capítulo del libro imaginario) y construimos lo que, en general, se denomina conocimiento o información organizada para la resolución de problemas o la toma de decisiones.

Modelos y tipos de base de datos

El modelado de datos es uno de los elementos más importantes al iniciar el desarrollo de cualquier proyecto. Esta es la estructura, sobre la que realmente reside la verdadera esencia de la aplicación. Incluso determina si el proyecto va a cumplir con su verdadero objetivo.

Modelo de datos

El modelado de datos es una técnica independiente de la implementación a la base de datos. Esto es importante, siempre busca obtener el máximo provecho de diversas herramientas. En particular, el esquema final y su implementación sufren cambios sin afectar de manera drástica la lógica de programación.

Uno de los puntos a resaltar es que el modelado de los datos debe ser llevado como una guía general. Para los profesionales expertos, esto implica el desarrollo de los diagramas de entidades y del modelo entidad-relación. Independientemente de la metodología a utilizar, esta herramienta siempre será importante, para entender las relaciones entre las diversas entidades en la base de datos. Ésta, es un conjunto integrado de datos que modelizan un

universo dado, el cual está compuesto por objetos relacionados, los objetos de un mismo tipo que constituyen una entidad y el vínculo entre entidades se denomina asociación.

La parte esencial de una estructura de base de datos es el modelo de datos: una colección de herramientas conceptuales para describir los datos, las relaciones entre ellos, la semántica y condiciones de consistencia.

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos. Generalmente todo modelo tiene una representación gráfica, para el caso de datos el modelo más popular es el modelo entidad relación o diagrama. Los modelos de bases de datos individuales se diseñan en base a las reglas y los conceptos de cualquier modelo de datos más amplio que los diseñadores adopten. La mayoría de los modelos de datos se representan por un diagrama de base de datos.

Importancia del modelo de base de datos

Está radica en que, si tenemos buenos esquemas en los modelos a desarrollar, los datos tendrán una estructura que refleje de buena manera las entidades del mundo real. Y es el punto de partida para diseñar una base de datos haciendo uso de técnicas que ayuden a realizarlo de una buena manera.

Otro aspecto importante en el modelado de datos durante el diseño de una base de datos es que tenemos una buena organización en la información, realizamos una búsqueda de los datos de manera ordenada, evitamos la duplicidad de información, verificamos que el dato tenga un significado, que toda la información sea oportuna y ayuda a que la base de datos no tenga errores o fallas.

Los modelos de datos definen con claridad cómo se modela la estructura lógica de una base

de datos.

Los modelos, son las entidades necesarias para introducir la abstracción en un SGBD, entendiéndola como el proceso de aislar un elemento de su contexto o del resto de elementos que lo acompañan. Un modelo de base de datos incluye las relaciones y limitaciones que determinan cómo se almacenan los datos y accede a ellos.

El modelado de datos es el proceso de documentar un diseño de sistema de software complejo como un diagrama de fácil comprensión, usando texto y símbolos para representar la forma en que los datos necesitan fluir. El diagrama se utiliza como un mapa para la construcción de un nuevo software o para la reingeniería de una aplicación antigua.

Un modelo de datos es una descripción de algo conocido como contenedor de datos (donde se guarda la información), así como los métodos para almacenar y recuperar información de esos contenedores.

Los modelos de datos no son cosas físicas: son abstracciones para la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Tipos de modelo de base de datos

El proceso de descripción de asociaciones y entidades se llama modelización y se hace con un modelo de datos.

A continuación, una lista de algunos modelos de base de datos:

- Jerárquico
- De red
- Relacional
- Orientado a objetos
- Modelo Transaccional
- Modelo Entidad - Relación

Modelo Jerárquico

el modelo jerárquico no diferencia una vista lógica de una vista física de la base de datos. De manera que las relaciones entre datos se establecen siempre a nivel físico, es decir, mediante referencia a direcciones físicas del medio de almacenamiento (sectores y pistas).

Los datos se almacenan en la forma de registros, el equivalente a las tuplas del modelo relacional. Cada registro consta de un conjunto de campos, el equivalente a los atributos del modelo relacional. Un conjunto de registros con los mismos campos se denomina fichero (record type, en inglés), el equivalente a las relaciones del modelo relacional.

El modelo jerárquico facilita relaciones padre-hijo, es decir, relaciones 1:N (de uno a varios) del modelo relacional. Pero a diferencia de este último, las relaciones son unidireccionales. En justicia, dichas relaciones son hijo-padre, pero no padre-hijo. Por ejemplo, el registro de un empleado (nodo hijo) puede relacionarse con el registro de su departamento (nodo padre), pero no al contrario. Esto implica que solamente se puede consultar la base de datos desde los nodos hoja hacia el nodo raíz. La consulta en el sentido contrario requiere una búsqueda secuencial por todos los registros de la base de datos (por ejemplo, para consultar todos los empleados de un departamento). En las bases de datos jerárquicas si existen índices que faciliten esta tarea.

Obsérvese que, a priori, no existen relaciones N:M (de muchos a muchos) en el modelo jerárquico. Salvo que se simulen mediante varias relaciones 1:N. No obstante, esto puede provocar problemas de inconsistencia, ya que el gestor de base de datos no controla estas relaciones.

Como ya se ha mencionado, las relaciones se establecen mediante punteros entre registros. Es decir, un registro hijo contiene la dirección física en el medio de almacenamiento de su registro padre. Esto tiene una ventaja fundamental sobre las bases de datos relacionales: el rendimiento. El acceso de un registro a otro es prácticamente inmediato sin necesidad de



consultar tablas de correspondencia.

Las relaciones jerárquicas entre diferentes tipos de datos pueden hacer que sea muy sencillo responder a determinadas preguntas, pero muy difícil el contestar a otras.

Limitaciones del Modelo Jerárquico

A continuación, se mencionan los problemas típicos de las bases de datos jerárquicas y que no existen en las bases de datos relacionales. Todos estos problemas derivan del hecho de que el sistema gestor de base de datos no implementa ningún control sobre los propios datos, sino que queda en manos de las aplicaciones garantizar que se cumplen las condiciones invariantes que se requieran (por ejemplo, evitar la duplicidad de registros). Dado que todas las aplicaciones están sujetas a errores y fallos, esto es imposible en la práctica. Además, dichas condiciones suelen romperse ex profeso por motivos operativos (generalmente, ajustes debidos a cambios en el negocio) sin evaluarse sus consecuencias.

Duplicidad de registros

No se garantiza la inexistencia de registros duplicados. Esto también es cierto para los campos "clave". Es decir, no se garantiza que dos registros cualesquiera tengan diferentes valores en un subconjunto concreto de campos.

Integridad referencial

No existe garantía de que un registro hijo esté relacionado con un registro padre válido. Por ejemplo, es posible borrar un nodo padre sin eliminar antes los nodos hijo, de manera que estos últimos están relacionados con un registro inválido o inexistente.

Desnormalización

Este no es tanto un problema del modelo jerárquico como del uso que se hace de él. Sin embargo, a diferencia del modelo relacional, las bases de datos jerárquicas no tienen controles que impidan la desnormalización de una base de datos. Por ejemplo, no existe el

concepto de campos clave o campos únicos.

La desnormalización permite ingresar redundancia de una forma controlada, seguir a una serie de pasos conlleva a:

- Combinar las relaciones
- Duplicar los atributos no claves
- Introducción de grupos repetitivos
- Crear tablas de extracción
- Cuando se debe desnormalizar:
- Se debe desnormalizar para optimizar el esquema relacional
- Para hacer referencia a la combinación de 2 relaciones que forman una sola relación

Modelo de Red

Una base de datos de red es una base de datos conformada por una colección o set de registros, los cuales están conectados entre sí por medio de enlaces en una red. El registro es similar al de una entidad como las empleadas en el modelo relacional.

Un registro es una colección o conjunto de campos (atributos), donde cada uno de ellos contiene solamente un único valor almacenado.

El enlace es exclusivamente la asociación entre dos registros, así que podemos verla como una relación estrictamente binaria.

Una estructura de base de datos de red, llamada algunas veces estructura de plex, abarca más que la estructura de árbol: un nodo hijo en la estructura red puede tener más de un nodo padre. En otras palabras, la restricción de que en un árbol jerárquico cada hijo puede tener sólo un padre, se hace menos severa.

Así, la estructura de árbol se puede considerar como un caso especial de la estructura de red.

Modelo Relacional

En los primeros años de las bases de datos, cada aplicación almacenaba datos en su propia estructura única. Cuando los desarrolladores querían crear aplicaciones para usar esos datos, tenían que conocer muy bien esa estructura de datos concreta a fin de encontrar los datos que necesitaban. Esas estructuras de datos eran poco eficaces, el mantenimiento era complicado y era difícil optimizarlas para ofrecer un buen rendimiento en las aplicaciones. El modelo de base de datos relacional se diseñó para resolver el problema causado por estructuras de datos múltiples y arbitrarias.

El modelo relacional proporcionó una forma estándar de representar y consultar datos que podía utilizarse en cualquier aplicación. Desde el principio, los desarrolladores se dieron cuenta de que la virtud principal del modelo de base de datos relacional era el uso de tablas, ya que era una forma intuitiva, eficiente y flexible de almacenar y acceder a información estructurada.

Con el tiempo, los desarrolladores comenzaron a usar el lenguaje de consulta estructurado (SQL) para escribir y hacer consultas en una base de datos: esto sería otra de las grandes virtudes de este modelo. Durante muchos años, el SQL se ha utilizado como el lenguaje para realizar consultas en bases de datos. Se basa en el álgebra relacional y proporciona un lenguaje matemático de uniformidad interna que facilita la mejora del rendimiento de todas las consultas en bases de datos. Otros métodos empleados necesitan definir consultas individuales.

Ventajas de las bases de datos relacionales

El modelo relacional es sencillo pero muy potente, y lo utilizan organizaciones de todos los tipos y tamaños para una gran variedad de aplicaciones con datos. Las bases de datos relacionales se usan para rastrear inventarios, procesar transacciones de comercio electrónico, administrar cantidades enormes y esenciales de información de clientes y mucho



más. Las bases de datos relacionales se pueden emplear para cualquier aplicación de datos en la que los puntos de datos se relacionen entre sí y deban gestionarse de forma segura, conforme a normas y de un modo uniforme.

Las bases de datos relacionales han existido desde la década de los setenta. En la actualidad, el modelo relacional sigue siendo el más aceptado para las bases de datos, gracias a todas sus virtudes.

Uniformidad de los datos

El modelo relacional es el ideal para mantener la uniformidad de los datos en todas las aplicaciones y copias de la base de datos (llamadas instancias). Por ejemplo, cuando un cliente deposita dinero en un cajero automático y, a continuación, mira el saldo en un teléfono móvil, el cliente espera ver ese depósito reflejado inmediatamente. Las bases de datos relacionales son perfectas para este tipo de uniformidad, y garantizan que todas las instancias de una base de datos tengan los mismos datos en todo momento.

Es muy difícil que otros tipos de bases de datos mantengan este nivel de concordancia tan inmediato con grandes cantidades de datos. Algunas bases de datos recientes, como NoSQL, solo pueden suministrar una “uniformidad final”. Según este principio, cuando la base de datos se adapta o cuando varios usuarios acceden a los mismos datos al mismo tiempo, los datos necesitan tiempo para “actualizarse”. La uniformidad final es aceptable en algunos casos (por ejemplo, para mantener listas en un catálogo de productos), pero para operaciones comerciales críticas, como transacciones de carritos de compra, la base de datos relacional sigue siendo la referencia.

Compromiso y atomicidad

Las bases de datos relacionales gestionan las reglas y políticas comerciales a un nivel muy detallado, y tienen políticas estrictas sobre el compromiso (es decir, el establecimiento de un cambio en la base de datos como algo permanente). Por ejemplo, imaginemos una base de



datos de inventario para rastrear tres piezas que siempre se usan juntas. Cuando se saca una pieza del inventario, las otras dos también deben salir. Si una de las tres piezas no está disponible, ninguna de ellas debe salir del inventario; las tres deben estar disponibles antes de que la base de datos establezca el compromiso. Una base de datos relacional no se comprometerá sobre una pieza hasta que pueda comprometerse sobre las tres. Esta capacidad de compromiso multifacético se llama atomicidad. La atomicidad es la clave para mantener los datos precisos y garantizar el cumplimiento de las reglas, normas y políticas de la empresa.

Procedimientos almacenados y bases de datos relacionales

El acceso a los datos implica muchas acciones repetitivas. Por ejemplo, una consulta simple para obtener información de una tabla de datos puede necesitar repetirse cientos o miles de veces para producir el resultado deseado. Estas funciones de acceso a datos requieren algún tipo de código para acceder a la base de datos. Para los desarrolladores de aplicaciones, no es práctico tener que escribir código nuevo para estas funciones en cada nueva aplicación. Sin embargo, las bases de datos relacionales permiten procedimientos almacenados: bloques de código a los que se puede acceder con una simple llamada a la aplicación. Por ejemplo, un solo procedimiento almacenado puede proporcionar un etiquetado de registro uniforme para usuarios de múltiples aplicaciones. Los procedimientos almacenados también pueden ayudar a los desarrolladores a garantizar que ciertas funciones de datos en aplicación se implementen de una manera específica.

Bloqueo y concurrencia de base de datos

En una base de datos pueden surgir conflictos cuando varios usuarios o aplicaciones intentan cambiar los mismos datos al mismo tiempo. Las técnicas de bloqueo y concurrencia reducen la posibilidad de conflictos y mantienen la integridad de los datos.

El bloqueo evita que otros usuarios y aplicaciones accedan a los datos mientras se están

actualizando. En algunas bases de datos el bloqueo se aplica a toda la tabla, lo que influye de forma negativa en el rendimiento de la aplicación. Otras bases de datos, como las bases de datos relacionales de Oracle, aplican bloqueos en el nivel de registro y mantienen la disponibilidad de los otros registros dentro de la tabla; esto ayuda a garantizar un mejor rendimiento de la aplicación.

La concurrencia gestiona la actividad cuando varios usuarios o aplicaciones hacen llamadas a consultas al mismo tiempo y en la misma base de datos. Esta función proporciona el acceso correcto a los usuarios y aplicaciones de acuerdo con las políticas definidas para el control de datos.

ACID y las bases de datos relacionales

Las transacciones en bases de datos relacionales están definidas por cuatro propiedades fundamentales: atomicidad, uniformidad, aislamiento y durabilidad; se suele hacer referencia a estas cuatro funciones por el nombre ACID (su acrónimo en inglés).

La atomicidad define todos los elementos que conforman una transacción completa de base de datos.

La uniformidad define las reglas para mantener los puntos de datos en un estado correcto después de una transacción.

El aislamiento impide que el efecto de una transacción sea visible a otros hasta que se establezca el compromiso, a fin de evitar confusiones.

La durabilidad garantiza que los cambios en los datos se vuelvan permanentes cuando la transacción se haya fijado y hayamos llegado a un compromiso.

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.

Una base de datos orientada a objetos es un sistema de gestión de base de datos mediante el cual representamos la información en forma de objetos que son utilizados en programación orientada a objetos.

La investigación y el desarrollo de la base de datos orientada a objetos fue impulsada en gran medida por su alto rendimiento en almacenamiento de datos y los requerimientos de acceso que tenían entornos de soporte de diseño tales como el CAD (Computer Aided Design) y el CASE (Computer Aided Software Engineering).

A medida que ha ido aumentando el uso de la tecnología web, muchas organizaciones se muestran más interesadas en el concepto de base de datos orientada a objetos, dado que es útil para mostrar los datos complejos de estas tecnologías.

Algunas bases de datos orientadas a objetos han sido diseñadas para trabajar bien con lenguajes de programación orientados a objetos tales como Delphi, Ruby, Python, Perl, Java, Visual Basic.NET, etc.

A partir de 2004, la base de datos orientada a objetos ha vuelto a experimentar un crecimiento debido al surgimiento de bases de datos orientadas a objetos de código abierto. Son fáciles de usar y asequibles. Están totalmente escritas en lenguaje de programación orientado a objetos, como Smalltalk, Java o C.

Uso actual de la base de datos orientada a objetos

La base de datos orientada a objetos ha encontrado su nicho de aplicación en áreas tales como la ingeniería, las bases de datos espaciales, las telecomunicaciones y en áreas científicas como la física de alta energía y la biología molecular.

Otro conjunto de bases de datos orientadas a objetos se centra en el uso integrado en

dispositivos, paquetes de software y sistemas en tiempo real.

Utilidad del modelo de bases de datos orientada a objetos

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. Las CAD/CAM, 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.

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.

Modelo Transaccional

Las bases de datos transaccionales son bases de datos optimizadas para ejecutar sistemas de producción, pueden utilizarse desde sitios web hasta bancos y pequeños negocios.

Sobresalen en la lectura y escritura de datos individuales y mantienen la integridad de los datos almacenados. Las bases de datos transaccionales no se construyen específicamente para el análisis de datos.

Sin embargo, a menudo se convierten en entornos analíticos porque ya están implementadas como bases de datos de producción. Esto, gracias a que han existido por décadas, son comunes, accesibles y eficaces.

Las bases de datos transaccionales se almacenan en el disco como filas, en lugar de columnas. Esto es estupendo ya que si por ejemplo, necesitas saber todo acerca de un cliente en la tabla de usuarios, tendrás la opción de tomar solo los datos que necesitas.

Existen algunas características que sin duda son bastante beneficiosas como, por ejemplo:

Aseguran la integridad de los datos

Las bases de datos transaccionales están diseñadas para ser compatibles con ACID, lo que permite que la base de datos pueda mantener un alto nivel de integridad de los datos que se incluyen en ella. Esto es de vital importancia sobre todo en las transacciones bancarias y comerciales.

Recordemos que ACID, es un conjunto de propiedades que describe cómo se diseñan las bases de datos transaccionales para preservar la integridad de la escritura en la base de datos.

Baja latencia

Debido a que las bases de datos transaccionales están diseñadas para ejecutar sistemas de producción, son muy buenas en operaciones que deben completarse en milisegundos. En pocas palabras, son increíblemente rápidas.

Componentes básicos

Entidades

En bases de datos, una entidad es la representación de un objeto con existencia independiente o concepto del mundo real que se describe en una base de datos.

Tipo de entidad Persona, organización, tipo de objeto o concepto sobre los que se almacena información. Describe el tipo de la información que se está controlando. Normalmente un tipo de entidad corresponde a una o varias tablas relacionadas en la base de datos.

Atributos

Los atributos son las características que definen o identifican a una entidad. Estas pueden ser muchas, y el diseñador sólo utiliza o implementa las que considere más relevantes.

Relaciones

Es un vínculo que define una dependencia entre varias entidades, es decir, exige que compartan ciertos atributos de forma indispensable.

Cardinalidad

La cardinalidad se representa en un diagrama ER como una etiqueta que se ubica en ambos extremos de la línea de relación de las entidades y que contiene diversos valores entre los que destacan el 1 (uno) y el * (asterisco).

Claves

Es el campo o atributo de una entidad o tabla que distingue cada registro del conjunto, sirviendo sus valores como datos vinculantes de una relación entre registros de varias tablas.

Diferencias entre el modelo de datos y el modelado de datos

Modelo de datos

El resultado de un modelado de datos es una representación que tiene dos componentes: las propiedades estáticas que se definen en un esquema y las propiedades dinámicas que se definen como especificaciones de transacciones, consultas e informes.

Un esquema consiste en una definición de todos los tipos de objetos de la aplicación, incluyendo sus atributos, relaciones y restricciones estáticas.

Correspondientemente, existe un repositorio de información, la base de datos, que es una instancia del esquema. Un determinado tipo de procesos sólo necesita acceder a un subconjunto predeterminado de entidades definidas en un esquema, por lo que este tipo de procesos requiere sólo un subconjunto de las propiedades estáticas del esquema general.

Detallando el Modelado de datos

Entendemos al modelado de datos como el proceso de documentar un diseño de sistema de software complejo como un diagrama de fácil comprensión, usando texto y símbolos para representar la forma en que los datos necesitan fluir.

El diagrama se utiliza como un mapa o plano para la construcción de un nuevo software o para la reingeniería de una aplicación antigua.

Los modeladores de datos utilizan varios modelos para ver los mismos datos y garantizar que todos los procesos, entidades, relaciones y flujos de datos han sido identificados. Hay varios enfoques diferentes para el modelado de datos, incluyendo:

Modelado conceptual de datos

Identifica las relaciones de más alto nivel entre diferentes entidades.

Modelado lógico de datos

Ilustra las entidades, atributos y relaciones específicas que participan en una función de



negocios. Sirve como soporte a la creación del modelo de datos físico.

Modelado de datos físicos

Representa una aplicación e implementación específica de base de datos de un modelo de datos lógicos.

Los modelos de datos definen cómo se modela la estructura lógica de una base de datos. Los modelos de datos son entidades fundamentales para introducir la abstracción en una base de datos.

Los modelos de datos definen cómo los datos se conectan entre sí y cómo se procesan y almacenan dentro del sistema.

El primer modelo de datos fue el modelo de datos planos, donde todos los datos utilizados se mantendrían en el mismo plano.

Un modelo de datos puede ser concreto o abstracto, y están representados por la notación de modelado de datos, que a menudo se presenta en formato gráfico.

Su enfoque principal es apoyar y ayudar a los sistemas de información mostrando el formato y la definición de los diferentes datos involucrados.

También ayudan a evitar la redundancia de datos. La información almacenada en los modelos de datos es de gran importancia para las empresas porque dicta las relaciones entre las tablas de la base de datos, las claves externas y los eventos involucrados.



Modelos de datos conceptuales

Un modelo conceptual de datos identifica las relaciones de más alto nivel entre las diferentes entidades.

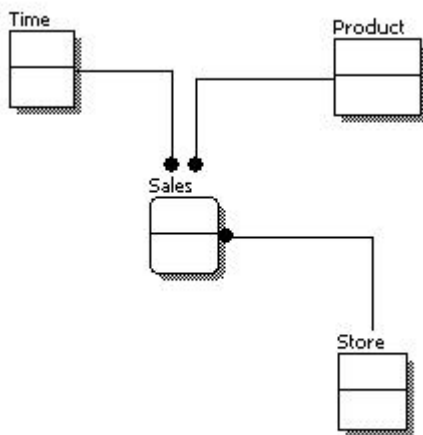
Las características del modelo conceptual de datos incluyen:

Incluye las entidades importantes y las relaciones entre ellas.

No se especifica ningún atributo.

No se especifica ninguna clave principal.

La siguiente figura es un ejemplo de un modelo conceptual de datos.



Modelos de datos físicos

El modelo de datos físicos representa cómo se construirá el modelo en la base de datos.

Un modelo de base de datos física muestra todas las estructuras de tabla, incluidos el nombre de columna, el tipo de datos de columna, las restricciones de columna, la clave principal, la clave externa y las relaciones entre las tablas.

Las características de un modelo de datos físicos incluyen:

Especificación de todas las tablas y columnas.

Las claves externas se usan para identificar relaciones entre tablas.

La desnormalización puede ocurrir según los requisitos del usuario.

Las consideraciones físicas pueden hacer que el modelo de datos físicos sea bastante diferente del modelo de datos lógicos.

El modelo de datos físicos será diferente para diferentes Sistemas de Gestión de Base de datos. Por ejemplo, el tipo de datos para una columna puede ser diferente entre MySQL y SQL Server.

Los pasos básico para el diseño del modelo de datos físicos son los siguientes:

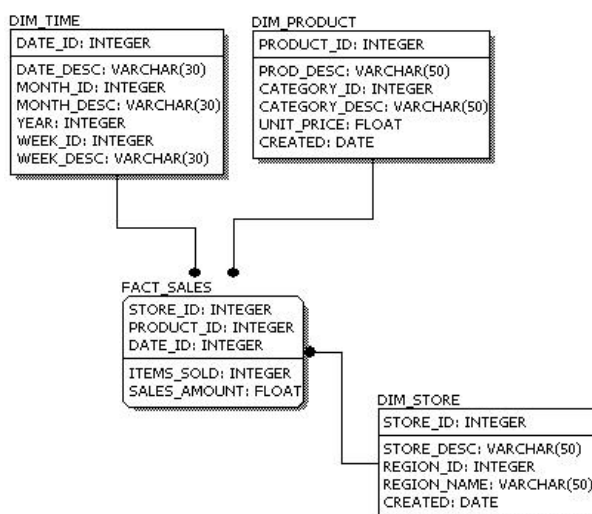
Convertir entidades en tablas.

Convertir relaciones en claves externas.

Convertir atributos en columnas.

Modificar el modelo de datos físicos en función de las restricciones / requisitos físicos.

La siguiente figura es un ejemplo de un modelo de datos físicos.



Modelos de datos lógicos

Un modelo de datos lógicos describe los datos con el mayor detalle posible, independientemente de cómo se implementarán físicamente en la base de datos.

Las características de un modelo de datos lógicos incluyen:

Incluye todas las entidades y relaciones entre ellos.

Todos los atributos para cada entidad están especificados.

La clave principal para cada entidad está especificada.

Se especifican las claves externas (claves que identifican la relación entre diferentes entidades).

La normalización ocurre en este nivel.

Los pasos para diseñar el modelo de datos lógicos son los siguientes:

Especifique claves primarias para todas las entidades.

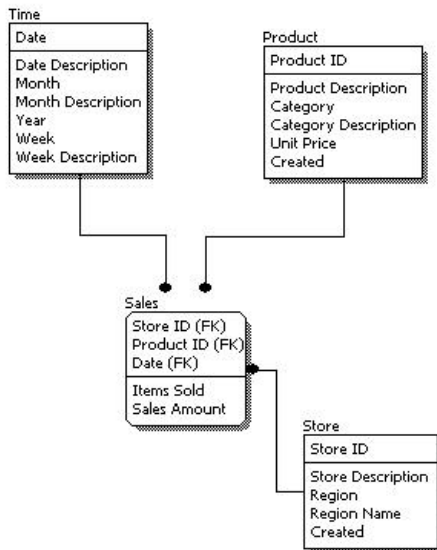
Encuentra las relaciones entre diferentes entidades.

Encuentra todos los atributos para cada entidad.

Resuelva las relaciones de muchos a muchos.

Normalización.

La siguiente figura es un ejemplo de un modelo de datos lógicos.



La tabla a continuación compara las diferentes características:

Característica	Conceptual	Lógico	Física
Nombres de entidades	X	X	
Relaciones de entidades	X	X	
Atributos		X	
Teclas principales		X	X
Foreign Keys		X	X
Nombres de tabla			X
Nombres de columnas			X
Tipos de datos de columna			X

Ventajas y desventajas de los modelos de datos

Ventajas

El objetivo principal de un modelo de datos es asegurarse de que los objetos de datos ofrecidos por el equipo funcional se representen con precisión.

El modelo de datos debe ser lo suficientemente detallado para ser utilizado para construir la base de datos física.

La información en el modelo de datos se puede utilizar para definir la relación entre tablas, claves primarias y externas y procedimientos almacenados.

El modelo de datos ayuda a las empresas a comunicarse dentro y entre las organizaciones.

El modelo de datos ayuda a documentar las asignaciones de datos en el proceso ETL

Ayuda a reconocer las fuentes de datos correctas para poblar el modelo.

Desventajas

Para desarrollar el modelo de datos se deben conocer las características físicas de los datos almacenados.

Incluso los cambios más pequeños realizados en la estructura requieren modificaciones en toda la aplicación.

No hay un lenguaje de manipulación de modelos establecido en DBMS.

Conclusiones y recomendaciones

Algunos de los aspectos aprendidos y que de gran peso es la base de datos su definición, requerimiento, ventajas y características donde podemos decir que la base de datos: Es una colección de datos o información usados para dar servicios a muchas aplicaciones al mismo tiempo.

En cuanto al requerimiento podemos decir que cumple las mismas tareas de análisis que del software y tiene como característica relacionar la información como vía organización y asociación donde la base de datos tiene una ventaja que es utilizar la plataforma para el desarrollo del sistema de aplicación en las organizaciones.

Otro aspecto importante sería el diseño y creación de la base de datos, donde existen distintos modos de organizar la información y representar las relaciones entre por datos los tres modelos lógicos principales dentro de una base de datos son el jerárquico, de redes y el relacional, los cuales tiene ciertas ventajas de procesamiento y de negocios.

Referencias bibliográficas

- Bertone, N. (2017). *Introducción a las bases de datos: fundamentos y diseño*. Pearson Educación
- Silberschatz, A. & Sudarshan, S. (2018). *Fundamentos de bases de datos* (6a. ed.). McGraw-Hill Interamericana.
- Bernal, Nieto. (2017). *Diseño de base de datos*. Universidad del Norte
- Pulido Romero, E. Escobar Domínguez, Ó. y Núñez Pérez, J. Á. (2019). *Base de datos*. Grupo Editorial Patria.
- tecnologias-informacion.com. (2018). Modelos de datos: Modelo Conceptual, Físico y Lógico. Sitio de obtención de la información: <https://www.tecnologias-informacion.com/modelos-datos.html>
- Oracle.com. (2020). Que es una base de dato relacional. Sitio de obtención de la información: <https://www.oracle.com/es/database/what-is-a-relational-database/>
- Unam. (2020). Modelo Orientado a Objetos. Sitio de obtención de la información: https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html



www.usanmarcos.ac.cr

San José, Costa Rica