

Fundamentos de Bases de Datos

Tarea 1

Conceptos básicos

18 de febrero del 2019

1. Conceptos generales

- a. ¿Porqué elegirías almacenar datos en un **sistema de bases datos** en lugar de simplemente almacenarlos utilizando el **sistema de archivos** de un sistema operativo? ¿En qué caso no tendría sentido utilizar un sistema de bases de datos?

Elegiría el SBD si vale la pena la inversión en tiempo y recursos en todo el diseño e infraestructura de la relacionados. Es decir, si mis necesidades involucran preservar la integridad de los datos, tener un manejo eficiente y permitir el acceso a varias personas conforme a peticiones. También esperaría sean muchos los datos que se planea almacenar, que estos tengan una estructura bien definida y se relacionen entre sí.

Cuando tengo una situación como la anterior no me conviene usar el sistema de almacenamiento del SO porque puede ser problemático organizar, editar y buscar datos, no tendría las ventajas del SDBD y proporcionar acceso a través de consultas a datos específicos también requeriría mucho trabajo.

Cuando dejan de cumplirse parte de las condiciones planteadas al inicio, pueden existir alternativas viables que me permiten hacer lo que quiero sin tener que pasar por todo el proceso de crear un SBD; más simples y versátiles. Por ejemplo, si sólo quiero hacer unas tablas para mi tarea, general algunas gráficas a partir de ellas y realizar unos calculos (promedios, sumas de secciones, etc.) Excel o alguna herramienta similar sería una buena opción.

- b. ¿Qué **ventajas** y **desventajas** encuentras al trabajar con una **base de datos**?

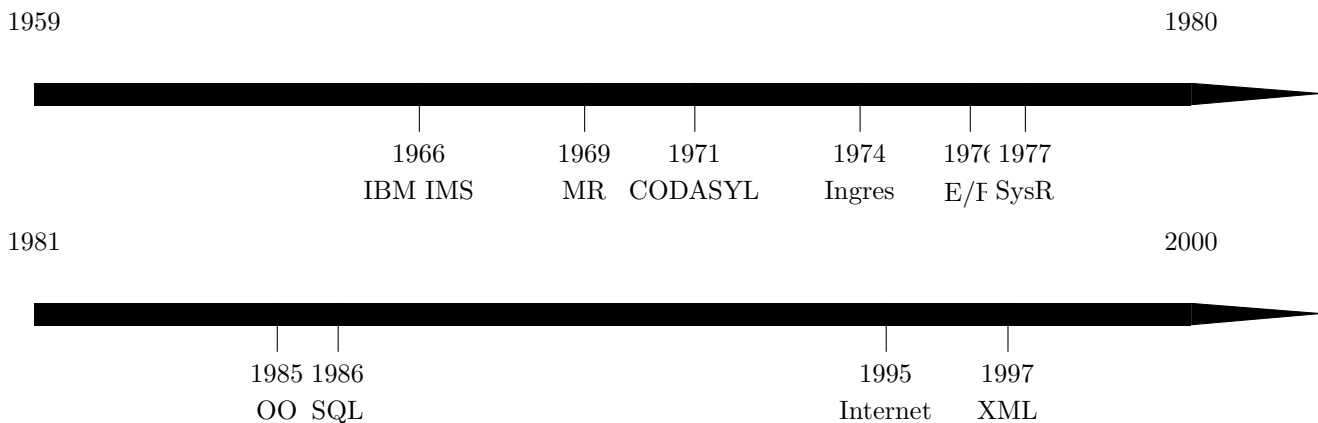
Las ventajas de trabajar con una base de datos estan dadas por las garantías y características que le da el diseño y SDBD; integridad de datos, operacones básicas, chequeo de redundancia, un sistema robusto capaz de crecer...

En cuanto a desventajas, estas dependen de los límites en el diseño, por ejemplo, quiero guardar un historial junto con los datos actuales (Almacenes) o mi modelo no me permite representar lo que necesito) y el costo en tiempo y recursos de realizar una implementacion buena de mi BD. Sin contar la dificultad.

- c. Explica las diferencias entre los esquemas **externos**, **conceptual** y **físico**. ¿Cómo se relacionan estos conceptos con la **independencias física y lógica**?

El esquema externo depende unicamente del planteamiento de mi problema; representa los procesos de los usuarios que me interesa permitir. Una vez que se define se pasa al esquema conceptual, en el cual se busca determinar como va a funcionar la base de datos en abstracto. El esquema físico finalmente se refiere a los detalles de la implementación; metadatos, almacenamiento, etc. La independencia física marca el punto en el que se puede separar el nivel físico de los otros aspectos de la base de datos (Nivel conceptual y externo). Asimismo, la independencia lógica separa el Nivel externo del resto de la infraestructura de la BD, ya que el uso que se le da a la base no debería relacionarse con la infraestructura de la BD; el usuario no necesita saber.

- d. ¿Qué es el **diccionario de datos** y por qué es importante?
Es la parte de la base de datos encargada de preservar el modo en el que se accede/describe a los datos. Desde el cómo se ven almacenados en la comp. hasta que nombre tiene la columna a la que pertenece este dato. Es decir, contiene a los metadatos que describen en general de la BD.
- e. Indica las principales características de alguno de los siguientes modelos de bases de datos: **jerárquico, de red, orientado a objetos**.
- **Jerárquico**
Los datos se almacenan en una estructura de árbol, donde cada dato tiene un padre, excepto un dato raíz, y posiblemente varios o ningún hijo.
Funciona de forma similar al sistema de archivos de un sistema operativo.
 - **Estructura**
En general, es un árbol. La estructura de los subárboles no está limitada.
 - **Restricciones de integridad**
Cada nodo tiene un solo dato. Se pueden definir los tipos de nodos posibles, cada nodo con atributos que a su vez son nodos.
Puede o no haber restricciones de unicidad, pero como para acceder a un dato hay que recorrer el árbol, esto puede ser difícil de mantener.
Tampoco hay restricciones de integridad, pues no es posible definir llaves foráneas al sólo poder tener un sólo padre.
 - **Operaciones**
Las mismas operaciones que en un árbol. Notemos que como hay padre únicos, no se pueden modelar de forma sencilla relaciones de muchos a muchos. Además, para mantener el árbol, insertar un dato puede cambiar radicalmente toda la estructura, y para realizar consultas hay que realizar recorridos por el árbol.
 - **De red**
Dados los problemas del modelo jerárquico, se creó el modelo de redes. En lugar de tener una estructura de árbol, se tiene una estructura más general de gráfica. Esto da más flexibilidad en cuanto a las relaciones posibles.
 - **Estructura**
Es una estructura de gráfica. Cada nodo puede tener varios padres e hijos
 - **Restricciones de integridad**
Igualmente que en el modelo jerárquico, se pueden definir los tipos de nodo.
 - **Operaciones**
Las mismas operaciones que en una gráfica. Aunque se puede definir relaciones de muchos a muchos, y por lo tanto se pueden dar restricciones de integridad referencial, acceder a un dato sigue requiriendo un recorrido de la gráfica, lo que dificulta mantener la integridad de unicidad.
 - **Orientado a objetos**
Los datos se modelan como objetos. Cada dato es una instancia de un objeto, donde el estado de cada objeto (los atributos) son otros objetos.
 - **Estructura**
Son objetos
 - **Restricciones de integridad**
Las mismas que en la orientación a objetos. Hay restricciones de tipo, y si se manejan los atributos como apuntadores, se tiene integridad referencial. También se tiene el encapsulamiento de datos
 - **Operaciones**
Son los métodos de los objetos.
- f. Elabora una **línea de tiempo** en dónde indiques los principales hitos en el desarrollo de las bases de datos.



Donde

- IBM IMS: IBM Information Management System
- MR: Modelo relacional, propuesto por E. Codd
- CODASYL: Estandar del modelo de redes, definido en la conferencia CODASYL de 1971
- Ingres: origen de Postgres
- Modelo Entidad/Relación
- System R: origen de IBM DB2
- Modelo Orientado a Objetos
- SQL: Standard Query Language se define como estándar en 1971 por la INS y la ANSI
- Primeras aplicaciones de internet que acceden a bases de datos
- XML: se utiliza para procesar las bases de datos para resolver problemas varios

g. Indica las responsabilidades que tiene un **Sistema Manejador de Bases de Datos** y para cada responsabilidad, explica los problemas que surgirían si dicha responsabilidad no se cumpliera.

Un SDBD tiene como responsabilidades

- Definir tipos, estructuras y restricciones. De no funcionar bien, ni siquiera se podría hablar de integridad, pues no se podría definir ningún tipo de regla u estructura correcta.
- Construir la parte física de la base de datos. De no realizarse correctamente, se podrían perder o corromper los datos
- Manipular los datos, como en funciones u operaciones. De no funcionar adecuadamente, podría pasar que datos perfectamente válidos pierdan su integridad (sean alterados) después de ser manipulados
- Compartir la base de datos en diferentes vistas. De no funcionar bien, los datos a los que se acceden pueden ser erróneos, a pesar de que internamente la base de datos funcione perfectamente

h. Supón que una pequeña compañía desea almacenar su información en una base de datos. Desea comprar la que tenga la menor cantidad de características posibles, se desea ejecutar la aplicación en una sólo computadora personal y no se planea compartir la información con nadie. Para cada una de las siguientes características, explica por qué se debería o no incluir en la base de datos que se desea comprar (suponiendo que se pueden comprar por separado): **seguridad, control de concurrencia, recuperación en caso de fallos, lengua de consulta, mecanismo de visitas, manejo de transacciones.**

- Seguridad
Esto es inclusive independiente del tipo de sistema que se tenga. La seguridad es indispensable, y más aún al ser un negocio, pues manejan información delicada acerca de clientes, proveedores, empleados, entre otros.

- Control de concurrencia
Cómo no se planea tener múltiples usuarios, entonces probablemente no hay muchos problemas de concurrencia, por lo que no es necesario tener un control de ella
- Recuperación en caso de fallos
Como toda la información estará en una sola computadora, entonces es de suma importancia preferir la información es esa única computadora. Por lo que la recuperación de fallos es indispensable
- Lengua de consulta
Esto es independiente de la cantidad de usuarios o de computadoras, además de indispensable, pues no se podría manipular la información sin ella
- Mecanismo de visitas
Cómo no se planea tener visitas (pues sólo habrá un usuario), entonces no es necesario contar con un mecanismo de visitas.
- Manejo de transacciones
Esto también es independiente de la cantidad de usuarios o de la cantidad de computadoras. Sin esto, no se podría garantizar la integridad de los datos, por lo que es indispensable

2. Investigación

- a. ¿Qué es la **Calidad de Datos** y cómo se relaciona con las bases de datos?
La calidad de datos se refiere a los procesos y técnicas enfocadas a mejorar la eficacia de los datos existentes en nuestras bases de datos. En este sentido, para que un proceso de calidad de datos sea realmente eficaz, éste debería ser repetible y fácil de entender, de manera que permitiera generar un proceso que se vuelva un ciclo de mejora y que cada vez que fuera ejecutado generara datos con mayor calidad, permitiendo desarrollar reportes para dar seguimiento a los progresos y proporcionar la mejora continua de la calidad de los datos. Por lo tanto, La calidad de datos está relacionada con las bases de datos justo porque se enfoca en mejorar la eficacia de los datos existentes en una base de datos, es decir, que los datos se mantengan en el mejor estado posible, que no haya duplicidades, que sean correctos, que estén actualizados de forma continua, etc. [2, pag. 6]
- b. Especifica las características más importantes de las bases de datos **NoSQL**, indica el modelo de datos que utilizan y principales proveedores.
Es un modelo no relacional, sus principales características son:
 - Proporcionan un rendimiento superior al que ofrecen los sistemas RDBS convencionales.
 - Pueden soportar lenguajes de consulta de tipo SQL.
 - Se ejecutan en máquinas con pocos recursos.
 - Pueden manejar gran cantidad de datos.
 - Ofrecen operaciones bastante simples sobre estructuras de datos flexibles. De hecho, por lo general, aprovechan esta simplicidad para proporcionar una alta escalabilidad y un rendimiento masivo.
 - Manejan elementos individuales, identificados por claves únicas.
 - Se clasifican de acuerdo a su forma de almacenar los datos: almacenes de registros extensibles, almacenes de documentos y almacenes de valor clave.

Principales proveedores: Hypertable, Cassandra, MongoDB, DynamoDB, HBase y Redis.

- c. ¿Qué es un **ORM**?
El mapeo objeto-relacional que se conoce comúnmente como ORM por sus siglas en inglés Object Relational Mapping, es una técnica de programación para convertir datos entre el sistema de tipos utilizando un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional. Es una

tecnica que se utiliza para poder ligar las bases de datos y los conceptos de orientación a objetos creando "base de datos virtuales". Esta transformación permite el uso de las bondades de la programación orientada a objetos como son la herencia y polimorfismo. [5]

Referencias

- [1] Gilfillan, I. (2015, June 8). Exploring Early Database Models. Retrieved February 17, 2019, from <https://mariadb.com/kb/en/library/exploring-early-database-models/>
- [2] PowerData.Calidad de Datos. URL: https://landings.powerdata.es/hubfs/POWERDATA_TOFU_-_calidad_de_datos.pdf
- [3] Paolo Atzenin, Uniform access to NoSQL systems. Information Systems,43, July 2014, Pages 117-133.
- [4] acens, the cloud services company. Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar. URL: <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
- [5] https://en.wikipedia.org/wiki/Object-relational_mapping