Diseño y Administración de Base de Datos

Unidad II-

Introducción al desarrollo de bases de datos

Definición

Una base de datos es una herramienta para recopilar y organizar datos pertenecientes a un mismo contexto y almacenarlos sistemáticamente para su posterior uso. Por ejemplo, en las bases de datos se puede almacenar información sobre personas, productos, pedidos, o cualquier otra cosa.

Existen programas denominados sistemas gestores de bases de datos, abreviado SGBD (o DBMS en inglés), permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Características

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Integridad
- Seguridad
- Concurrencia
- Recuperacion

Integridad: La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

Seguridad: La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

Concurrencia: En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

Recuperación: Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

Sistemas de Administración de Bases de Datos

Un Sistema de Administración de Base de Datos es una herramienta que nos permite ingresar, recuperar y manejar la información contenida en la base de datos. Entendemos por manejar, la posibilidad de ejecutar las siguientes operaciones, entre muchas otras:

- Añadir nueva información a medida que ésta va ingresando.
- Obtener la información ordenada según determinados parámetros (por orden alfabético, según el nombre del autor, según la temática de cada libro, etc.).
- Calcular cómputos referidos a la base (cantidad total de publicaciones,promedios periódicos de ventas, promedios según las diversas categorías,etc.).
- Imprimir la información deseada, ya sea en forma de tablas o de gráficos de diversos tipos.

Arquitectura de los SGBD

Existen tres características importantes, inherentes a los sistemas de bases de datos:

- La separación entre los programas de aplicación y los datos
- El manejo de múltiples vistas por parte de los usuarios.
- El uso de un catálogo para almacenar el esquema de la base de datos.

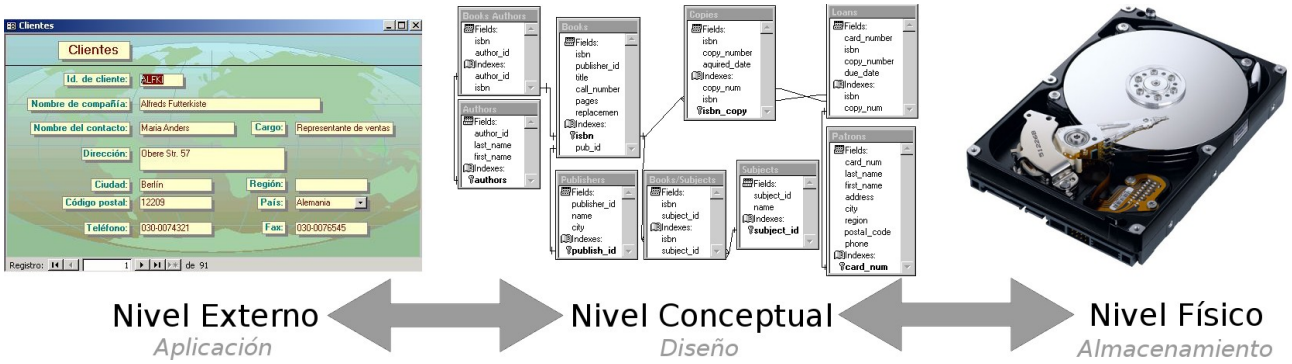
La arquitectura de tres niveles para los Sistemas de Gestión de Base de Datos, cuyo objetivo principal es el de separar los programas de aplicación de la Base de datos física consiste en:

Nivel externo o de visión: es el más cercano al usuario, se describen varios esquemas externos o vistas de estos. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

Nivel conceptual: describe la estructura de toda la base de datos para un grupo determinado de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento.

Nivel interno o físico: describe la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, etc.

La mayoría de los Sistemas de Gestión de Base de Datos no distinguen correctamente entre estos tres niveles. En algunos casos, podemos ver como algunos SGDB incluyen detalles del nivel físico en el esquema conceptual. Prácticamente todos los Sistemas de Gestión de Base de Datos, se manejan vistas de usuario, ya que la mayoría de las bases de datos están pensadas para que otros usuarios puedan añadir, modificar y utilizar los datos.



Hay que destacar que los tres esquemas son sólo descripciones de los mismos datos tratados, pero con distintos niveles de abstracción. Los únicos datos que existen realmente están a nivel físico, en un dispositivo de almacenamiento no volátil. En un Sistema Gestor de Base de Datos basado en la arquitectura que estamos viendo, cada grupo de usuarios hace referencia exclusivamente a su propio esquema externo. El proceso de transformar peticiones y resultados de un nivel a otro se denomina correspondencia o transformación.

Objetivos del Desarrollo de Base de Datos

Disminuir la redundancia e inconsistencia de los datos.- Debido a que los archivos y programas de aplicación son creados por diferentes programadores en un largo período de tiempo, los diversos archivos tienen probablemente diferentes formatos y los programas pueden estar escritos en diferentes lenguajes. La misma información puede estar duplicada en diferentes lugares (archivos). Esta redundancia conduce a un almacenamiento y coste de acceso más altos. Además puede conducir a inconsistencia de datos, es decir, las diversas copias de los mismos datos pueden no coincidir con los datos reales.

Evitar dificultad en el acceso a los datos.- El entorno de procesamiento de archivos convencional no permite que los datos necesarios sean obtenidos de una forma práctica y eficiente. Se deben desarrollar sistemas de recuperación de datos más interesantes para un uso general.

Evitar el aislamiento de datos.- Debido a que los datos están dispersos en varios archivos, y los archivos pueden estar en diferentes formatos, es difícil escribir nuevos programas de aplicación para recuperar los datos apropiados.

Evitar los problemas de Integridad.- Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de ligaduras de inconsistencia. Los desarrolladores hacen cumplir esas ligaduras en el sistema añadiendo el código apropiado en los diversos programas de aplicación. Sin embargo, cuando se añaden ligaduras, es difícil cambiar los programas para hacer que se cumplan. El problema es complicado cuando las ligaduras implican diferentes elementos de datos de diferentes archivos.

Evitar el problema de atomicidad.- Un sistema de una computadora, como cualquiera otro dispositivo mecánico o eléctrico, está sujeto a fallo. En muchas aplicaciones es crucial asegurar que una vez que un fallo a ocurrido y se ha detectado, los datos se restauran al estado de consistencia que existía antes del fallo, es decir, las modificaciones deben de ocurrir por completo o no ocurrir en absoluto.

Evitar anomalías en el acceso concurrente.- Conforme se ha ido mejorando el conjunto de ejecución de los sistemas y ha sido posible una respuesta en tiempo más rápida, muchos sistemas han ido permitiendo a múltiples usuarios actualizar los datos simultáneamente. En tales sistemas un entorno de interacción de actualizaciones concurrentes puede dar lugar a datos inconsistentes.

Evitar los problemas de seguridad.- No todos los usuarios de un sistema de base de datos deberían poder acceder a todos los datos.

Proceso de Desarrollo de la Base de Datos

Como muchos sabemos, ponerse a desarrollar una base de datos con cierta complejidad y tamaño “a ojo” es perder el tiempo. Para que la aplicación cumpla eficientemente sus objetivos y los resultados sean buenos, debemos seguir un proceso:

- Análisis.
- Diseño del modelo entidad / relación.
- Diseño del modelo relacional.
- Lenguaje SQL y base de datos final.

Análisis

Debemos comenzar estudiando a fondo el mundo real que deseamos representar en la aplicación y base de datos. Por ejemplo: una universidad, un hospital, una empresa tecnológica.

A partir de este estudio, debemos crear el Modelo de Datos, que es simplemente la visión del mundo real bajo unos determinados objetivos.

Modelo entidad / relación (e/r)

Cuando se inicia el diseño de una base de datos, uno de los mayores problemas que existe, es que el diseñador concibe la base de datos con un modelo de datos orientado a la maquina/plataforma. Y esto hace que en un alto grado se pierda la conceptualidad del problema quedando tal diseño contaminado por cuestiones de implementación.

Al final será un desastre puesto que nuestra aplicación no será capaz de representar fidedignamente la realidad del asunto. Ademas de conllevar otra serie de problemas como: falta de eficiencia, peor optimizado, imposibles actualizaciones, consultas extremadamente largas…

El diseñador debe concebir la base de datos en un nivel superior, abstrayéndose de cualquier consideración técnica o de implementación en sistema, plataforma o aplicación.

Para ello puede contar con la ayuda de un modelo de datos como el E/R, presentado por *Peter P. Chen*. Con el podrá centrarse en la estructura lógica y abstracta de la información, siendo capaz de representar toda la semántica del mundo real por medio de entidades y relaciones.

Modelo relacional

Llegados a este punto, el diseñador se acerca (ahora si es el momento) más a la maquina que al mundo real. El diseñador debe transformar el modelo E/R en el modelo relacional, teniendo muy en cuenta la teoría de la normalización. Esta es una operación de cierta complejidad.

El modelo relacional, presentado por el Dr. *E. F. Codd*, fue revolucionario puesto que consigue la independencia de las aplicaciones respecto a los datos.

Este modelo de datos esta basado en las teorías matemáticas de las relaciones, haciendo que los datos se estructuren lógicamente en forma de relaciones (tablas).

Presenta beneficios como:

- **Sencillez y uniformidad**: Al tener como resultado una colección de tablas, y ser la tabla la estructura básica se da como resultado una gran uniformidad, junto con la sencillez de los lenguajes de usuario que pueden operar con ellas.
- **Flexibilidad**: Ofreciendo a los usuarios los datos de la forma mas adecuada a su aplicación.
- **Independencia del interfaz de usuario**: El modo en el que se almacena los datos no influye en su manipulación lógica.

Lenguaje sql y base de datos final

Quizás esta sea la etapa mas fácil, pues el trabajo duro queda hecho con los dos modelos de datos anteriores. Ahora solamente tendremos que codificar en lenguaje SQL el modelo relacional expuesto anteriormente.

Para ello necesitaremos de:

- LDD: Con el que (por ejemplo) codificar las sentencias para la creación de las distintas tablas de la base de datos.
- LMD: Para codificar las instrucciones (que por ejemplo) se encargarán de realizar: consultas, adiciones, eliminaciones de registros.