



TC2005B
Módulo de Bases de Datos

TC2005B

Módulo de Bases de Datos

Bases de Datos: Conceptos y Arquitectura

Modelos de Datos, esquemas e Instancias

Un modelo de datos es una colección de conceptos que pueden ser usados para describir la estructura de una base de datos y provee los mecanismos necesarios para lograr un cierto nivel de abstracción ocultando detalles de la implementación que los usuarios no necesitan saber

La estructura de una base de datos se refiere a los tipos de datos, relaciones y restricciones que pueden tener los datos. La mayoría de los modelos de datos incluyen un conjunto de operaciones básicas para especificar accesos y actualizaciones a la base de datos

Modelos de Datos, esquemas e Instancias

Además de las operaciones básicas provistas por el modelo de datos es común incluir conceptos para especificar el aspecto dinámico o las operaciones que definen el comportamiento de una base de datos.

Categorías de los modelos de datos

Se han propuesto muchos modelos de datos que podemos categorizar de acuerdo a los tipos e conceptos que utilizan para describir la estructura de la base de datos.

Los modelos de datos conceptuales o de alto nivel utilizan conceptos que son muy cercanos a la forma en que los usuarios perciben los datos, mientras que los modelos de bajo nivel o modelos de datos físicos proveen conceptos que describen en detalle como los datos son almacenados en la computadora, (generalmente usados por gente de sistemas).

Entre estos dos niveles existen los modelos de datos representacionales o de implementación, que proveen conceptos que pueden ser entendidos por usuarios finales, pero que no están muy alejados de la forma en como los datos están organizados en la computadora.

Los modelos conceptuales utilizan conceptos como:

Entidades. Representa un objeto o concepto del mundo real – ejemplo un empleado o un proyecto

Atributos. Representa alguna propiedad que describe a detalle una entidad – ejemplo nombre, salario, área de un empleado

Relaciones Entre dos o mas entidades – por ejemplo la relación **trabaja en** para relacionar empleados con proyectos.

Los modelos de datos representacionales o de implementación son utilizados en los SGBD comerciales, incluyen los modelos de datos relacionales. Los modelos de datos representacionales representan la información utilizando estructuras grabadas (registros), por eso también son llamados modelos de datos basados en registros.

También podemos mencionar los modelos de datos orientados a objetos como implementaciones de alto nivel que están cercanos a los modelos de datos conceptuales.

Esquemas, Instancias y Estado de la Base de Datos.

La descripción de la base de datos es llamada **el Esquema de la base de datos**, es especificado durante el diseño de la base de datos y se espera que no cambie mucho. Muchos modelos de datos tienen convenciones para desplegar esquemas como diagramas. La siguiente figura muestra un diagrama para el ejemplo visto en el tema anterior.

ESTUDIANTE	Nombre	Numero estudiante	Año	carrera
------------	--------	-------------------	-----	---------

CURSOS	NombreCurso	ClaveCurso	Créditos	Departamento
--------	-------------	------------	----------	--------------

SECCION	Seccion_Id	ClaveCurso	Semestre	Año	Profesor
---------	------------	------------	----------	-----	----------

REPORTE_NOTAS	Numero estudiante	Seccion_Id	Nota
---------------	-------------------	------------	------

PREREQUISITOS	ClaveCurso	Clave Prerrequisito
---------------	------------	---------------------

A cada parte del esquema (por ejemplo estudiante, cursos, sección, etc) los llamamos constructor de esquema (schema construct)

La información en la base de datos puede cambiar en cualquier momento, por ejemplo en fechas de inscripciones, calificaciones, etc. La información en la base de datos en un momento particular en el tiempo es llamado estado de la base de datos o snapshot. También es conocido como el conjunto actual de ocurrencias o instancias de la base de datos.

En un estado dado de la base de datos, cada constructor de esquema tiene su propio conjunto actual de instancias. Cada vez que se hace una modificación en un dato de la base de datos, pasamos de un estado a otro.

Importante:

Diferencia entre

Esquema de la base de datos

vs

Estado de la base de datos.

El SGBD es parcialmente responsable de que cada estado de la base de datos sea un estado válido
(satisfacer restricciones y estructura definidas en el esquema).

El SGBD almacena la descripción de los constructores de esquema y restricciones (meta-datos) en el catalogo de la base de datos, para que el software del SGBD pueda hacer referencia al esquema cuando lo necesite.

Esquema – Intención

Estado – Extensión del esquema

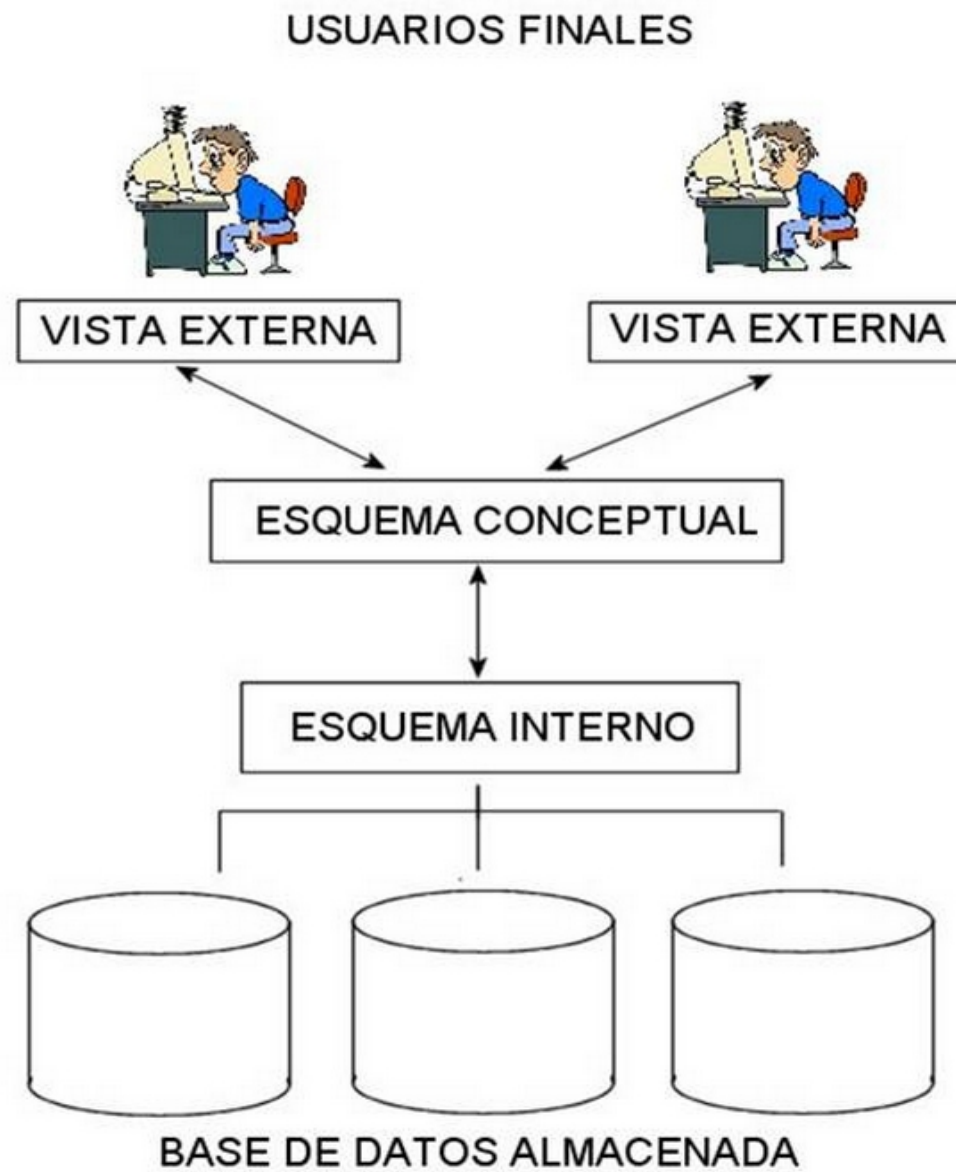
Evolución del esquema. Capacidad de cambiar según los requerimientos de la base de datos.

Arquitectura de esquema triple e Independencia de Datos

Tres de las características mas importantes del enfoque de las bases de datos son:

- a) Aislamiento de los programas y los datos
- b) Soporte de multiples vistas de usuarios
- c) Uso de un catalogo para almacenar la descripción de la base de datos (esquema)

La arquitectura de esquema triple fue propuesta para satisfacer estas tres características



NIVEL EXTERNO

mapeo externo / conceptual

NIVEL CONCEPTUAL

mapeo conceptual / interno

NIVEL INTERNO

Arquitectura de esquema triple

La meta de esta arquitectura es separar las aplicaciones del usuario de la base de datos física. En esta arquitectura, los esquemas pueden ser definidos en los siguientes tres niveles:

1. El nivel interno tiene un esquema interno, que describe la estructura del almacenamiento físico de la base de datos. El esquema interno usa un modelo de datos físico y describe los detalles del almacenamiento de la información así como rutas completas de la base de datos.

2. El nivel conceptual tiene un esquema conceptual, que describe la estructura de toda la base de datos para una comunidad de usuarios. El esquema conceptual oculta detalles de las estructuras de almacenamiento físico y se enfoca en describir entidades, tipos de datos, relaciones, operaciones de datos y restricciones.

3. El nivel externo o nivel de vista incluye un número de esquemas externos o vistas de usuarios. Cada esquema externo describe la parte específica de la base de datos en la que un grupo de usuarios puede estar interesado y oculta el resto de la base de datos a ese grupo.

Hay que notar que los tres niveles son solo descripciones de los datos, la información realmente existe exclusivamente en el nivel físico.

En un SGBD basado en la arquitectura de esquema triple, cada grupo de usuarios hace referencia a su propio esquema externo, entonces el SGBD debe hacer una transformación de esa petición al esquema conceptual, y después hacer otra transformación para acceder a los datos en el esquema interno.

Estas transformaciones se conocen como mapeos (mapping), que en ocasiones pueden consumir demasiado tiempo, es por eso que algunos DBMS no soportan los esquemas externos

Características del enfoque de Base de Datos

En donde quiera que tengamos un SGBD de varios niveles, el catálogo debe expandirse para incluir información de como mapear peticiones e información entre los diferentes niveles.

Cuando se hace algún cambio de esquema en algún nivel, el esquema en el nivel siguiente no cambia, solo cambian los mapeos de un nivel a otro, así los programas de aplicación (en el nivel externo) no necesitan cambio alguno.

Independencia de los datos

La independencia de los datos puede verse como la capacidad de cambiar el esquema en un nivel del sistema de base de datos sin tener que cambiar el esquema en el siguiente nivel superior.

Podemos definir dos tipos de independencia de datos:

- Independencia lógica de los datos.

La capacidad de cambiar el esquema conceptual sin tener que cambiar esquemas externos o programas de aplicación

- Independencia física de los datos.

La capacidad de cambiar el esquema interno sin tener que cambiar el esquema conceptual

Lenguajes e Interfaces de Bases de Datos

Lenguajes de los SGBD

Una vez completo el diseño de la base de datos, y cuando ya se tiene el SGBD listo para implementar la base de datos, lo primero que hay que hacer es especificar los esquemas conceptuales e internos para la base de datos y los mapeos entre ambos esquemas.

En algunos SGBD donde no hay una separación estricta de los niveles, se utiliza un lenguaje llamado Lenguaje de definición de datos (DDL – Data Definition Language).

El SGBD debe tener un compilador incorporado para procesar las instrucciones en el DDL para identificar los constructores de esquema y almacenar la descripción del esquema en el catalogo de la base de datos. En SGBD donde hay una clara separación de los niveles conceptual e interno, se utiliza el DDL para especificar solo el esquema conceptual.

Otro lenguaje llamado Lenguaje de definición de almacenamiento (SDL – Storage Definition Language) se usa para especificar el esquema interno.

Para una arquitectura real de arquitectura triple, necesitaremos un tercer lenguaje: Lenguaje de Definición de Vistas (VDL – View Definition Language) para especificar las vistas del usuario y los mapeos a la vista conceptual.

Una vez que los esquemas de la base de datos estan compilados y la base de datos ya tiene datos, se requieren formas de que los usuarios manipulen la base de datos. La manipulación típica incluye acceder, insertar, borrar y modificar los datos. El SGBD debe proveer un lenguaje llamado Lenguaje de Manipulación de Datos (DML – Data Manipulation Language).

Todos estos lenguajes no se consideran diferentes, existen lenguajes integrados comprensivos

SQL (Structured Query Language)

Interfaces de SGBD

Algunas interfaces amigables al usuario pueden ser:

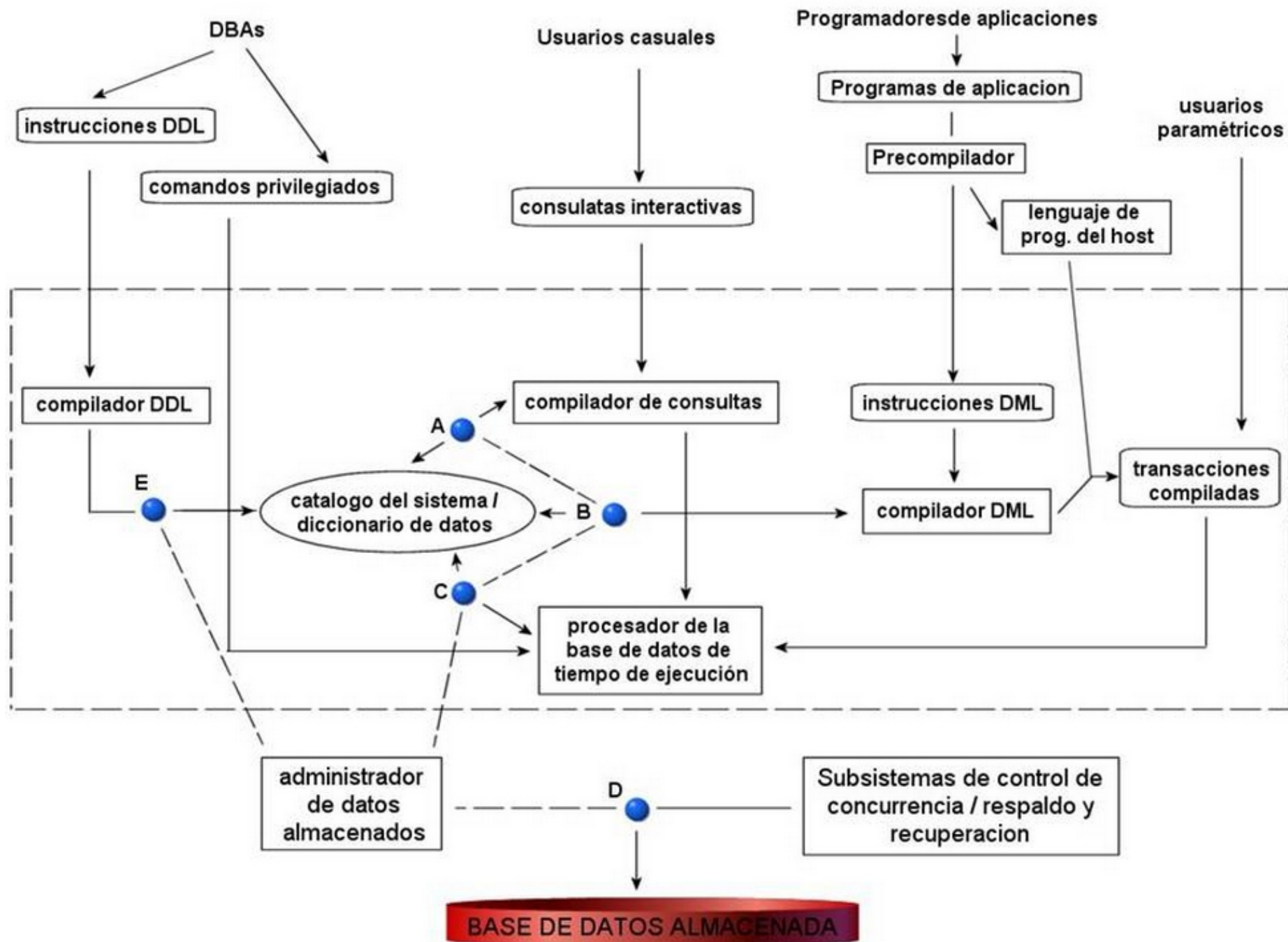
- Interfaces basadas en menús para clientes web o navegación
- Interfaces basadas en formularios
- GUIs
- Interfaces de Lenguaje Natural
- Interfaces para usuarios paramétricos
- Interfaces para los DBAs

El ambiente del Sistema de Base de Datos

Módulos componentes del DBMS

La siguiente figura muestra los componentes de un DBMS de forma simplificada. Por lo general, la base de datos y el DBMS están almacenados en disco, el acceso al disco esta controlado por el sistema operativo, quien programa las operaciones de entrada y salida al disco.

Un modulo del DBMS administrador de datos almacenados de un nivel mas alto controla los accesos a la información del DBMS almacenada en disco, ya sea información de la base de datos o del catalogo.



Las líneas punteadas y círculos etiquetados A, B, C, D y E muestran los accesos que están bajo el control de este modulo administrador de datos almacenados.

El administrador de datos almacenados puede utilizar servicios básicos del Sistema Operativo para llevar a cabo transferencias de datos de bajo nivel entre el disco y la memoria principal de la computadora. Pero controla otros aspectos de la transferencia de datos como el manejo de buffers en la memoria principal, para que los datos sean manipulados por otros módulos del BDMS o incluso por programas de aplicación.

Algunos DBMSs tienen su propio modulo administrador del buffer, mientras que otros utilizan el sistema operativo para manejar el “buffereo” de paginas de disco.

El compilador DDL procesa definiciones de esquema especificados en el DDL, y almacena las descripciones (meta datos) en el catalogo del DBMS.

El catalogo incluye información como nombres y tamaños de los archivos, nombres y tipos de datos de los datos, detalles de almacenamiento de cada archivo, información de mapeo entre esquemas y restricciones, y mucha información requerida por otros módulos del DBMS.

El procesador de la base de datos de tiempo de ejecución administra los accesos a la base de datos en tiempo de ejecución, recibe operaciones de recuperación o actualización y las lleva a cabo en la base de datos. Los accesos a disco pasan por el administrador de datos almacenados, y el administrador del buffer mantiene un registro en memoria de las paginas de la base de datos.

El compilador de consultas trata las consultas de alto nivel, analiza y compila o interpreta una consulta creando el código de acceso a la base de datos, generando llamadas al procesador de la base de datos de tiempo de ejecución para ejecutar el código.

El precompilador extrae los comandos DML de un programa de aplicación escrito en algún lenguaje de programación del host. Estos comandos son enviados al compilador DML para ser compilados en código objeto para acceso a la base de datos.

El resto del programa se envía al compilador de lenguaje del host. Los códigos objeto de los comandos DML y el resto del programa se ligan , formando una transacción, cuyo código ejecutable incluye llamadas al procesador de la base de datos de tiempo de ejecución.

Es común en nuestros días tener programas clientes que accedan al DBMS ejecutándose en una computadora diferente de la computadora en la que reside la base de datos (CLIENTE - SEVIDOR). En algunos casos, el cliente accesa una computadora intermedia llamada servidor de aplicación.

Arquitecturas centralizadas y cliente-servidor para SGBDs

Arquitectura centralizada

La arquitectura de los SGBD ha seguido caminos similares a las arquitecturas de sistemas computacionales generales.

En las primeras arquitecturas se contaba con un servidor con gran poder de cómputo al cual se conectaban equipos con poder de computo limitado que se conectaban como terminales.

Todo el procesamiento de las aplicaciones e interfaces se llevaba a cabo en los servidores, los sistemas gestores de bases de datos estaban centralizados.

Utilerías de los sistemas de bases de datos

Además de los módulos descritos, la mayoría de los SGBD cuentan con utilerías de apoyo para el DBA. Las utilerías comunes tienen los siguientes tipos de funciones:

- Cargado
- Respaldo
- Reorganización de archivos
- Monitoreo de Desempeño

Cargado

cargar archivos de datos existentes (de texto o secuénaciales) Por lo general los formatos de los datos actuales (fuente) y de los datos deseados (destino) son especificados. Herramientas de Conversión

Respaldo

Una utilería de respaldo se utiliza para hacer copias de seguridad de la base de datos completa en algún medio de almacenamiento.

Reorganización de archivos

Reorganización utilizada para mejorar el desempeño de la base de datos.

Monitoreo de desempeño

Monitorea el uso de la base de datos y proporciona estadísticas al DBA para decidir si se hace una reorganización de archivos o no

Algunas otras utilerías pueden incluir ordenamiento de archivos, manejo de compresión de datos, monitoreo de acceso de usuarios, interacción con la red, y algunas otras funciones..

Arquitecturas centralizadas y cliente-servidor para SGBDs

Con la evolución de la tecnología, las computadoras se volvieron cada vez mas accesibles, de tal forma que el usuario ya contaba con poder de computo en lo que antes era una simple terminal, de esta manera se empezó a explotar el poder de cómputo en las computadoras cliente.

Esto dio origen a las arquitecturas cliente-servidor

Arquitecturas cliente – servidor básicas

La arquitectura cliente servidor fue desarrollada para trabajar con ambientes de computo con un numero considerable de computadoras, estaciones de trabajo, servidores de archivos, impresores, servidores de bases de datos, servidores web y otro equipo conectado a una red.

La idea es definir servidores especializados con funcionalidades especificas, de esta forma los recursos provistos por cualquier servidor pueden ser accedados por varias maquinas cliente, Dichas maquinas cliente proveen al usuario las interfaces apropiadas para utilizar estos servidores, así como el poder de computo para ejecutar aplicaciones.

Clasificación de SGBDs

Existen varios criterios para clasificar a los Sistemas Gestores de Bases de Datos.

Clasificación por modelo de Datos.

El modelo de datos utilizado en la mayoría de los DBMSs comerciales es el modelo relacional. El modelo de datos orientado a objetos es implementado en algunos sistemas comerciales pero no ha tenido mucho éxito.

Algunas otras aplicaciones utilizan los modelos jerárquico y el modelo de red. El modelo relacional evoluciona continuamente y ha incorporado ya muchos conceptos desarrollados en bases de datos orientadas a objetos, esto ha dado lugar a SGBD objeto-relacional. Entonces, según el modelo de datos, podemos categorizar a los SGBD en: relacional, objetos, objeto-relacional, jerárquico, de red y otros.

Modelo de red.-

representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros.

Los registros se organizan en un conjunto de gráficas (grafos) arbitrarias. La dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Modelo jerárquico.-

Es similar al modelo de red en cuanto a las relaciones y datos, ya que estos se representan por medio de registros y sus ligas.

La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Modelo Orientado a Objetos

Este modelo, reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulación

Herencia

Polimorfismo

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.

Modelo relacional.-

En este modelo se representan los datos y las relaciones entre éstos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla

Consideremos una empresa que requiere controlar a los vendedores y las ventas que ellos realizan; de este problema determinamos que los objetos o entidades principales a estudiar son el empleado (vendedor) y el artículo (que es el producto en venta).

Clasificación por número de usuarios soportados por el sistema

Monousuario.- Permiten que un solo usuario esté trabajando en la base de datos al mismo tiempo. Por lo general son utilizados en computadoras personales.

Multiusuario.- Permiten más de un usuario al mismo tiempo. La mayoría de los SGBD son multiusuario.

Clasificación por distribución de la base de datos

Centralizado.- si toda la información está almacenada en una sola computadora. Puede ser monousuario o multiusuario.

Distribuido.- (DDBMS - SGBDD) Es cuando el sistema de base de datos puede estar en dos o mas computadoras, conectadas por una red.

En un SGBDD homogéneo, se utiliza el mismo software en cada una de las computadoras que componen el sistema, pero también existen SGBD distribuidos heterogéneos, es decir, cada una de las diferentes computadoras de la red puede tener un software distinto.

Esto lleva a tener bases de datos federativas o un sistema gestor de base de datos de múltiples bases de datos, en las que cada componente tiene cierto nivel de autonomía.

Clasificación por costo

La mayoría de los paquetes de SGBD cuestan entre 10,000 y 100,000 USD.

Los sistemas simples monousuario que trabajan en PCs deben estar entre 100 y 300 USD.

También existen SGBD libres, por ejemplo: PostgreSQL, *MySQL (MariaDB), Firebird, SQLite

Otras clasificaciones

También podemos clasificar los SGBDs por el tipo de acceso y ordenamiento de los archivos, o por propósito (general / específico), procesamiento de transacciones en línea.