



# **Fundamentos de Base de Datos I**

Ing. Francisco José Santana V. MTi

## **Capítulo I**

# Fundamentos de Base de Datos I

Ing. Francisco José Santana V. MTi

---

## I- Conceptos Basicos

### Introducción Base de Datos

Un sistema de gestión de base de datos (DBMS Database Management System) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. La colección de datos, normalmente denominada base de datos, contiene información acerca de una empresa determinada. El objetivo primordial de un DBMS es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

Los sistemas de base de datos están diseñados para gestionar grandes bloques de información. La gestión de datos implica tanto la definición de estructuras para el almacenamiento de información como la provisión de mecanismos para la gestión de la información. Además, los sistemas de bases de datos deben mantener la seguridad de la información almacenada, pese a caídas del sistema o intentos de accesos no autorizados. Si los datos van a ser compartidos por varios usuarios, el sistema debe evitar posibles resultados anómalos.

La importancia de la información en la mayoría de las organizaciones, y por tanto el valor de la base de datos, ha llevado al desarrollo de una gran cantidad de conceptos y técnicas para la gestión eficiente de los datos.

Una **base de datos** es una colección de datos relacionados, como datos, queremos decir hechos conocidos que puedan ser almacenados y tengan significado implícito.

Una base de datos es diseñada, construida y poblada con datos para un propósito específico.

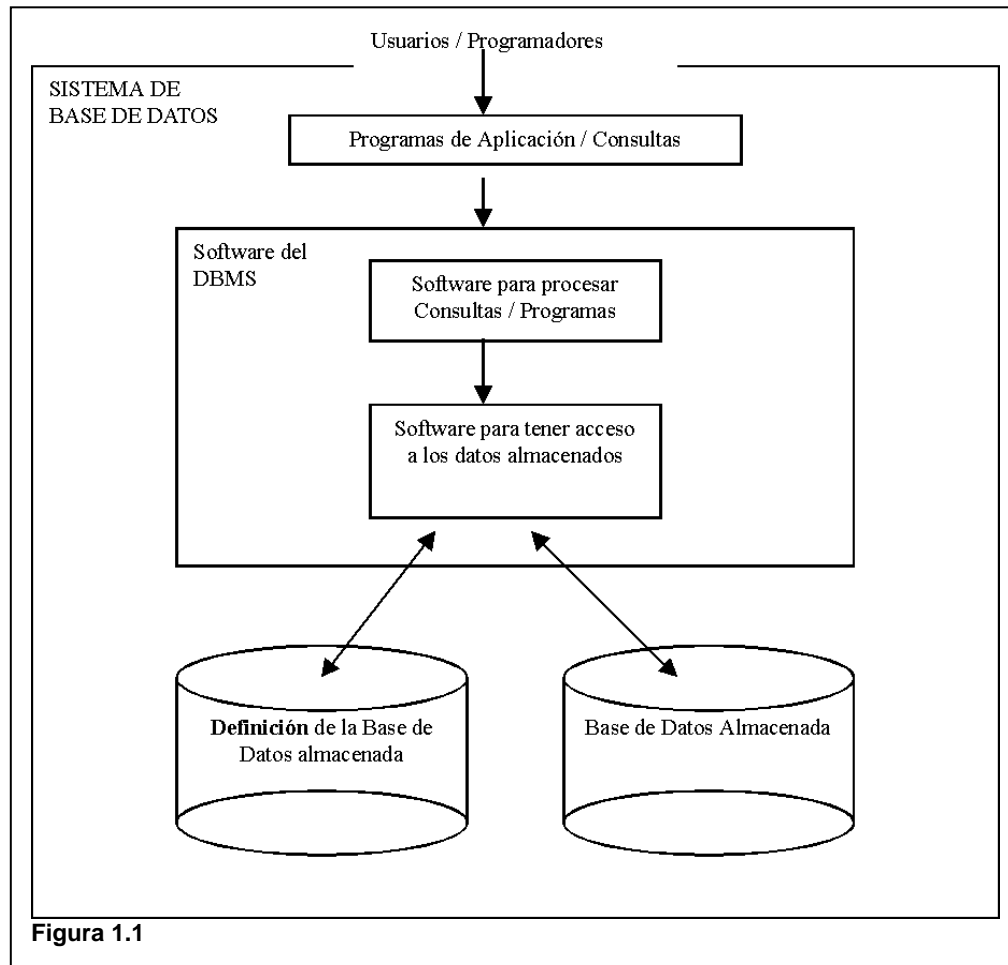
Una base de datos representa algún aspecto del mundo real, algunas veces llamado Minimundo (**miniworld**). Los cambios en el minimundo se reflejan en la base de datos.

**Sistema Manejador de Bases de Datos (DBMS database management system).** Es una colección de programas que permite a los usuarios crear y dar mantenimiento a una base de datos.

En 1964, se conciben los primeros Gestores de Base de Datos (DBMS: Database Management System), por medio de los cuales se pretende dar un viraje a los Sistemas de Archivos, los cuales se limitan a la estructuración del almacenamiento físico de los datos. Con los DBMS se crea el concepto de Administración de los datos, por medio de actividades integradas que permiten verlos físicamente en un solo

almacenamiento pero lógicamente se manipulan a través de esquemas compuesto por estructuras donde se establecen vínculos de integridad, métodos de acceso y organización física sobre los datos, permitiendo así obtener valores agregados de utilización tales como: manejo de usuarios, seguridad, atomicidad e independencia física y lógica de los datos, entre otros.

El primer gestor de bases de datos (DBMS) comercial, IDS: Integrated Data Store , se crea bajo el concepto del Modelo de Datos de Red (Bachgman, 1965); luego se desarrolla el IMS: Information Management System , sobre el concepto del Modelo de Datos Jerárquico. Estos DBMSs eran accedidos normalmente por lenguajes de programación como Cobol usando interfases de bajo nivel haciendo que las tareas de creación de aplicaciones y mantenimiento de los datos fuesen controlables, pero aún complejas.



**Gestor o Sistema Manejador de Base de Datos.** Es un módulo de programa que proporciona el interfaz entre los datos de bajo nivel almacenados en las bases de datos y los programas de aplicación y consultas hechas al sistema. Entre sus funciones están:

El manejador es responsable de las siguientes tareas:

1 **Interacción con el manejador de archivos:** Los datos en la base se guardan en disco mediante el sistema de archivos, proporcionado comúnmente por el sistema operativo. El manejador de la base, traduce las diferentes proposiciones del manejo de datos en comandos del sistema de archivos de bajo nivel. De esta forma el manejador se puede encargar del almacenamiento, recuperación y actualización de los datos en la base.

2 **Implantación de la integridad:** Los valores de los datos que se almacenan en la base, deben satisfacer ciertas limitantes de consistencia, estas limitantes deben ser determinadas por el administrador, pero es el manejador el encargado de verificar que las actualizaciones que se hagan a la base cumplan con dichas normas.

3 **Puesta en práctica de la seguridad:** El manejador de la base es quien verifica que los accesos a la base sean realizados por las personas autorizadas.

4 **Respaldo y recuperación:** Entre las labores que debe ejecutar el manejador está la de verificar de forma constante la integridad de la base, y lograr recuperación de datos y/o mejoras en caso que se requieran.

5 **Control de concurrencia:** Se podría entender, esta, como la principal tarea del manejador de la base, o por lo menos la más difícil. Cuando varios usuarios están accediendo la base al mismo tiempo, es posible que la consistencia de los datos no se conserve. El manejador debe encargarse de coordinar los accesos de los diferentes usuarios, de forma que los datos en la base no se dañen.

En términos ideales, un DBMS debe contar con estas funciones, sin embargo, no todos las poseen, así existen algunos manejadores que no cumplen la función de respaldo o de seguridad, dejándola al usuario o administrador; sin embargo un DBMS que sea completo y que deba manejar una base de datos multiusuario grande, es conveniente que cuente con todas estas operaciones.

### **Archivos Tradicionales vs. Enfoque de bases de Datos**

Considérese parte de una empresa bancaria de ahorros que guarda la información sobre todos los clientes y cuentas de ahorro en archivos de sistemas permanentes en el banco. Además, el sistema tiene diversos programas de aplicación que permiten al usuario manejar los archivos, incluyendo:

- Un programa para hacer cargos o abonos a una cuenta.
- Un programa para añadir una nueva cuenta.
- Un programa para obtener el saldo de una cuenta.
- Un programa para generar estados mensuales.

Estos programas de aplicación los han escrito programadores de sistemas en respuesta a las necesidades de la organización bancaria.

Según surge la necesidad, se añaden nuevos programas de aplicación al sistema. Por ejemplo, supóngase que una nueva ley del Gobierno permite al banco de ahorros ofrecer cuentas de cheques. Como resultado, se crean nuevos archivos permanentes que contienen información acerca de todas las cuentas de cheques que

mantiene el banco, y puede que sea necesario escribir nuevos programas de aplicación. Así, según pasa el tiempo, se añaden más archivos y programas de aplicación al sistema.

El típico sistema de procesamiento de archivos descrito arriba está apoyado por un sistema operativo convencional. Los registros permanentes se almacenan en varios archivos, y se escribe un número de diferentes programas de aplicación para extraer registros de y añadir registros a archivos apropiados. Este sistema tiene un número de desventajas importante.

**- Redundancia e inconsistencia de los datos.**

Puesto que los archivos y los programas de aplicación son creados por distintos programas durante un período largo de tiempo, es probable que los archivos tengan diferentes formatos y los programas pueden estar duplicados en varios sitios (archivos). Por ejemplo, la dirección y el número de teléfono de un cliente determinado pueden aparecer en un archivo que consta de registros de cuentas de cheques.

**- Dificultad para tener acceso a los datos.**

Supóngase que uno de los gerentes del banco necesita averiguar los nombres de todos los clientes que viven dentro del código postal de la ciudad 78733. El gerente pide al departamento de procesamiento de datos que genere la lista correspondiente. Puesto que esta solicitud no fue prevista cuando se diseñó el sistema original, no hay ningún programa de aplicación a mano que la satisfaga. Existe, sin embargo, un programa de aplicación para generar la lista de todos los clientes. El gerente tiene ahora dos elecciones: O bien coger la lista de clientes y extraer la información necesaria manualmente, o pedir al departamento de procesamiento de datos que ponga a un programador de sistemas a escribir el programa de aplicación necesario. Obviamente, ninguna de las dos alternativas es satisfactoria. Supóngase que se escribe un programa semejante y que, varios días después, el mismo gerente necesita arreglar esa lista para incluir sólo aquellos clientes con un saldo de 10,000 dólares o más. Como se esperaba, no existe un programa que genera esa lista. De nuevo el gerente tiene las dos opciones anteriores, ninguna de las cuales es satisfactoria.

Lo que se trata de probar aquí es que esos entornos convencionales de procesamiento de archivos no permiten recuperar los datos necesarios de una forma conveniente y eficiente. Deben desarrollarse sistemas de recuperación de datos para uso general.

**- Aislamiento de los datos.**

Puesto que los datos están repartidos en varios archivos, y éstos pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los datos apropiados.

**- Anomalías del acceso concurrente.**

Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los datos simultáneamente. En un entorno así, la interacción de actualizaciones concurrentes puede dar por resultado datos inconsistentes. Puesto que se puede

acceder a los datos por medio de diversos programas de aplicación diferentes que no han sido previamente coordinados, esta supervisión es muy difícil de proporcionar.

**- Problemas de seguridad.**

No todos los usuarios del sistema de base de datos deben poder acceder a todos los datos. Por ejemplo, en un sistema bancario, el personal de las nóminas sólo necesita ver la parte de la base de datos que tiene información acerca de los distintos empleados del banco. No necesitan acceder a información sobre las cuentas de los clientes. Puesto que los programas de aplicación se añaden al sistema de una forma precisa, es difícil implantar tales restricciones de seguridad.

**- Problemas de integridad.**

Los valores de datos almacenados en la base de datos deben satisfacer ciertos tipos de restricciones de consistencia. Por ejemplo, el saldo de una cuenta bancaria nunca debe caer por debajo de una cantidad prescrita (digamos, 25 dólares). Estas restricciones se hacen cumplir en el sistema añadiendo códigos apropiados en los diversos programas de aplicación. Sin embargo, cuando se añaden restricciones nuevas, es difícil cambiar los programas para hacerlos cumplir. El problema se complica aún más cuando las restricciones implican varios elementos de información de distintos archivos.

Estas dificultades, entre otras, han fomentado el desarrollo de sistema de gestión de bases de datos. En lo que sigue, veremos los conceptos y algoritmos que se han desarrollado para sistemas de bases de datos para resolver los problemas anteriormente mencionados.

Después de ver las principales desventajas del manejo de archivos tradicionales veremos el enfoque de Base de Datos.

### **Enfoque de Base de Datos:**

Una serie de características distinguen el enfoque de archivo tradicional, tratado en la clase pasada, del enfoque de base de datos. En el procedimiento tradicional cada usuario define e implementa los archivos necesarios para una aplicación específica.

En base de datos un lugar donde reposan los datos se mantiene, el cual es definido una sola vez y accesado por varios usuarios. Este enfoque evita la redundancia en definir y almacenar información, que resulta en desperdicio de espacio de almacenamiento y esfuerzos redundantes en mantener la información al día.

### **Algunas Características más relevantes:**

#### **- Naturaleza de Auto-Contenido**

Es una característica fundamental en el enfoque de base de datos, es que el sistema contiene no solo esta, sino una descripción o una definición completa de la base de datos. Esta definición es almacenada en el catálogo del sistema, el cual contiene información como la estructura de cada archivo, el tipo y formato de almacenamiento de cada campo y varias limitaciones con relación a los datos. La información almacenada en el catálogo es llamada meta-data y describe la estructura de la base de datos primaria.

En el procesamiento de base de datos, el software puede solo acceder varias bases de datos distintas extrayendo las definiciones de las bases de datos del catálogo y usándolas para acceder correctamente a cualquiera de estas bases de datos.

#### **- Aislamiento de Programas e información**

En contraste con el enfoque tradicional, los programas de acceso de un DBMS están escritos de forma independiente de cualquier archivo específico. La estructura de los archivos están almacenadas en el catálogo separada de cualquier programa, esta propiedad es llamada independencia de programas y datos.

#### **- Abstracción de Datos**

Un objetivo importante de un sistema de bases de datos, es proporcionar a los usuarios una visión de abstracción de datos. Es decir el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo para que el sistema sea manejable, los datos se deben extraer eficientemente. Este requerimiento a llevado al diseño de estructuras de datos complejas para la representación completa

en la base de datos. Puesto que muchos usuarios del sistema de bases de datos no tienen experiencia en computadoras, se le esconde la complejidad a través de diversos niveles de abstracción para simplificar su interacción con el sistema.

### **- Soporte de multiples imagenes de la información**

Una base de datos tiene típicamente muchos usuarios, cada uno de los cuales requiere una perspectiva, visión o imagen diferentes de las bases de datos. Una imagen o visión puede ser un subconjunto de la base de datos o puede contener información virtual que es derivada de las bases de datos, pero que no esta explícitamente almacenada.

Las características antes mencionadas son las más importantes al distinguir un Manejador de Bases de Datos del software de procesamiento de archivo tradicional.

### **Ventajas más importantes de los Manejadores de Bases de Datos:**

#### **- Control de Redundancia**

Se produce al almacenar la misma información varias veces cosa. En base de datos se integran las imágenes o visiones de cada grupo de usuarios durante el diseño de la base de datos. Para mantener la consistencia tendremos un diseño que almacene cada pieza de información en un solo lugar en la base de datos. Esto no permitirá la inconsistencia y además ahorrará espacio de almacenamiento.

#### **- Distribución de la Información**

Un manejador de base de datos multiusuario, como su nombre lo indica, debe permitir a multiples usuarios acceder a las bases de datos al mismo tiempo. Esto es esencial si la información de varias aplicaciones será integrada y mantenida en la misma base de datos. El DBMS deberá incluir software de control de concurrencia, para asegurar que varios usuarios que traten de actualizar la misma información lo hagan de manera controlada para que el resultado de las actualizaciones sea correcto.

Otro mecanismo que se enmarca en este concepto es la facilidad para definir una visión o imagen del usuario, la cual es utilizada para especificar que porción de base de datos es del interés particular de un grupo de usuarios.

#### **- Restricción de Acceso no autorizado**

Un DBMS debe proveer un subsistema de seguridad y autorización de acceso, el cual es utilizado por el DBA para crear cuentas y especificar restricciones por cuentas del usuario.

#### **- Representar Relaciones complejas entre los datos**

Un DBMS debe tener la capacidad para representar una variedad de relaciones complejas entre los datos así como tomar y actualizar datos relacionados en una forma fácil y eficiente.

#### **- Establecer y Mantener Limitaciones o controles de integridad**



Todas las bases de datos tienen ciertas limitaciones o controles de integridad que permanecen con los datos, el tipo más simple de estos es definir el tipo de dato para cada pieza de información.

#### **- Proveer Seguridad y Recuperación de la información**

Un DBMS debe proveer las facilidades para recuperarse de fallas de software o hardware. El subsistema de backup y Recovery es el responsable.

#### **- Otras ventajas**

- Capacidad para establecer estándares
- Flexibilidad
- Reducción del tiempo empleado en el desarrollo de aplicaciones
- Disponibilidad de información Actualizada.

### **Protagonistas en las escenas de Bases de Datos**

**- Administradores de Bases de Datos ( DBA )** : Son los responsables de tener un control central o descentralizado de los datos, autorizar el acceso a las bases de datos, coordinar y monitorear su uso. Entre las funciones se encuentran:

- Definición de Esquemas
- Definición de la estructura de almacenamiento y del método de acceso.
- Modificación del esquema de la organización física.
- Concesión de autorización para el acceso a los datos.
- Especificación de las restricciones de integridad

#### **Deberes de los DBA.**

Los deberes de un administrador de bases de datos dependen de la descripción del puesto, corporación y políticas de Tecnologías de Información (TI). Por lo general se incluye recuperación de desastres (respaldos y pruebas de respaldos), análisis de desempeño y optimización, y algo de asistencia en el diseño de la base de datos.

**La recuperabilidad** significa que, si se da algún error en los datos, hay un bug de programa ó de hardware, el DBA (Administrador de base de datos) puede traer de vuelta la base de datos al tiempo y estado en que se encontraba en estado consistente antes de que el daño se causara. Las actividades de recuperación incluyen el hacer respaldos de la base de datos y almacenar esos respaldos de manera que se minimice el riesgo de daño ó pérdida de los mismos, tales como hacer diversas copias en medios de almacenamiento removibles y almacenarlos fuera del área en antelación a un desastre anticipado. La recuperación es una de las tareas más importantes de los DBA's.

La recuperabilidad, frecuentemente denominada "recuperación de desastres", tiene dos formas primarias. La primera son los respaldos y después las pruebas de recuperación.

La recuperación de las bases de datos consiste en información y estampas de tiempo junto con bitácoras los cuales se cambian de manera tal que sean consistentes en un momento y fecha en particular. Es posible hacer respaldos de la base de datos que no

incluyan las estampas de tiempo y las bitácoras, la diferencia reside en que el DBA debe sacar de línea la base de datos en caso de llevar a cabo una recuperación.

Las pruebas de recuperación consisten en la restauración de los datos, después se aplican las bitácoras a esos datos para restaurar la base de datos y llevarla a un estado consistente en un tiempo y momento determinados. Alternativamente se puede restaurar una base de datos que se encuentra fuera de línea sustituyendo con una copia de la base de datos.

Si el DBA (o el administrador) intentan implementar un plan de recuperación de bases de datos sin pruebas de recuperación, no existe la certeza de que los respaldos sean del todo válidos. En la práctica, los respaldos de la mayoría de los RDBMSs son raramente válidos si no se hacen pruebas exhaustivas que aseguren que no ha habido errores humanos ó bugs que pudieran haber corrompido los respaldos.

**La integridad de una base de datos** significa que, la base de datos ó los programas que generaron su contenido, incorporen métodos que aseguren que el contenido de los datos del sistema no se rompa así como las reglas del negocio. Por ejemplo, un distribuidor puede tener una regla la cual permita que solo los clientes individuales puedan solicitar órdenes; a su vez cada orden identifique a uno y solo un proveedor. El servidor Oracle y otros DBMSs relacionales hacen cumplir este tipo de reglas del negocio con limitantes, las cuales pueden ser configuradas implícitamente a través de consultas. Para continuar con este ejemplo, en el proceso de inserción de una nueva orden a la base de datos, esta a su vez tendría que cerciorarse de que el cliente identificado existe en su tabla para que la orden pueda darse.

**Seguridad** significa la capacidad de los usuarios para acceder y cambiar los datos de acuerdo a las políticas del negocio, así como, las decisiones de los encargados. Al igual que otros metadatos, una DBMS relacional maneja la seguridad en forma de tablas. Estas tablas son las "llaves del reino" por lo cual se deben proteger de posibles intrusos

Disponibilidad

La disponibilidad significa que los usuarios autorizados tengan acceso a los datos cuando lo necesiten para atender a las necesidades del negocio. De manera incremental los negocios han ido requiriendo que su información esté disponible todo el tiempo (7x24", o siete días a la semana, 24 horas del día). La industria de TI ha respondido a estas necesidades con redundancia de red y hardware para incrementar las capacidades administrativas en línea.

**El desempeño** significa que la base de datos no cause tiempos de respuesta poco razonables. En sistemas muy complejos cliente/servidor y de tres capas, la base de datos es solo uno de los elementos que determinan la experiencia de los usuarios en línea y los programas desatendidos. El desempeño es una de las mayores motivaciones de los DBA para coordinarse con los especialistas de otras áreas del sistema fuera de las líneas burocráticas tradicionales.

### **Desarrollo/Soporte a pruebas**

Uno de los deberes menos respetados por el administrador de base de datos es el desarrollo y soporte a pruebas, mientras que algunos otros encargados lo consideran como la responsabilidad más importante de un DBA. Las actividades de soporte incluyen la colecta de datos de producción para llevar a cabo pruebas con ellos; consultar a los programadores respecto al desempeño; y hacer cambios a los diseños

de tablas de manera que se puedan proporcionar nuevos tipos de almacenamientos para las funciones de los programas.

Otros actores:

- **Diseñadores de Bases de Datos: ( DBD ):** Tienen la responsabilidad de identificar los datos a ser almacenados en las bases de datos y de seleccionar las estructuras apropiadas para representar y almacenar esos datos.

- **Analistas de Sistemas:** Determinan los requerimientos de los usuarios finales especialmente los usuarios paramétricos y casuales, desarrollando especificaciones para las transacciones que cumplen con esos requerimientos. Los analistas generalmente no son usuarios de las bases de datos.

- **Programadores de Aplicaciones:** Traducen las especificaciones de los analistas a un programa o aplicación en el computador. Los programadores por la naturaleza de su trabajo pudieran interactuar con las bases de datos como usuarios sofisticados.

- **Usuarios Finales:** Son las personas cuyo trabajo requiere acceder la base de datos para actualizar, consultar o generar reportes. Existen varios tipos de usuarios, entre ellos ingenuos, especializados y Sofisticados.

- **Sofisticados:** Los usuarios sofisticados interaccionan con el sistema sin escribir programas, en cambio escriben sus preguntas en lenguaje de consultas de base de datos.

- **Casuales:** Algunos usuarios sofisticados escriben aplicaciones de base de datos especializadas que no encajan en el marco tradicional de procesamiento de datos. Ejem. Graficas y datos de audio.

- **Ingenuos:** Los usuarios no sofisticados interactúan con el sistema invocando uno de los programas de aplicación permanente que se han escrito anteriormente. Eje. Sistema de comunicaciones, nomina, bancario etc.

## **Niveles de Abstracción de datos y la Arquitectura de Tres Esquemas**

### **Niveles de Abstracción de datos**

Un DBMS provee a los usuarios una representación conceptual de los datos que no incluyen muchos detalles de como estos son almacenados. Un **Modelo de Datos** es un tipo de abstracción de datos que es utilizada para proveer esta representación conceptual. El modelo de datos utiliza conceptos lógicos, como los objetos, sus propiedades y sus interrelaciones, los cuales serán mucho más fáciles de comprender para la mayoría de los usuarios que conceptos de almacenamiento en computadoras.

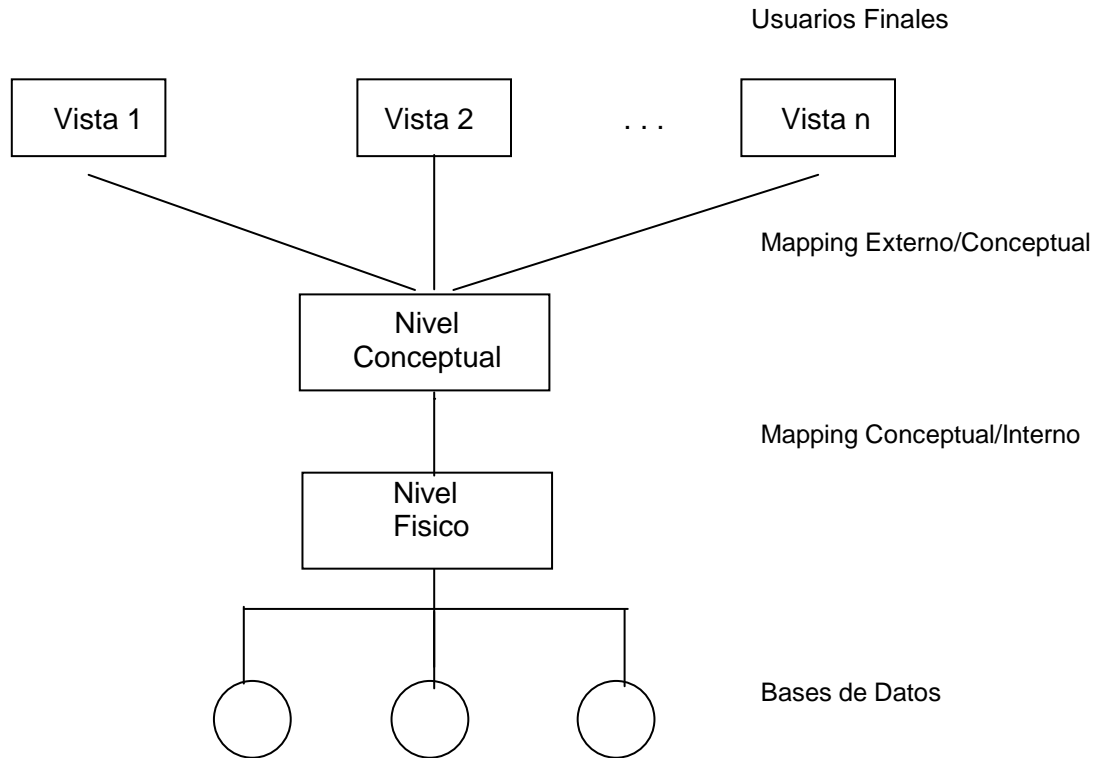
Existen tres niveles de Abstracción de datos:

**Nivel Físico:** Es el nivel más bajo de abstracción describe como se almacenan realmente los datos. En el nivel físico se describen en detalles la estructuras de los datos complejas del nivel bajo.

**Nivel Conceptual:** El nivel medio de abstracción qué datos son realmente almacenados en las bases de datos y las relaciones que existen entre los datos, a qui se describe la base de datos en terminos de un numero pequeño de estructuras relativamente sencillas. Aunque la implementación de las estructuras sencillas en el nivel conceptual puede implicar estructuras complejas en el nivel físico, el usuario del nivel conceptual no necesita darse cuenta de esto.

El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir que información se va a guardar en la base de datos.

**Nivel de Visión:** El nivel más alto de abstracción describe sólo parte de la base de datos. A pesar del uso de estructuras más sencillas en el nivel conceptual, permanece algo de complejidad debido al tamaño de la bases de datos. Muchos usuarios del sistema de bases de datos no se interesaran por toda esta información. En cambio, dichos usuarios sólo necesitan una parte de la bases de datos. Para simplificar su interacción con el sistema, se define el nivel de abstracción de visión. El sistema puede proporcionar muchas visiones para la misma base de datos.

**Figura 1.2**

### Arquitectura de Tres Esquemas

El Objetivo de la arquitectura de tres esquemas es separar las aplicaciones de los usuarios de la base de datos física. En esta arquitectura los esquemas pueden ser definidos a los tres niveles :

**Nivel Interno o Físico tiene un Esquema Interno:** este describe la estructura de almacenamiento físico de la base de datos. El esquema interno utiliza un modulo de datos físicos y describe completamente los detalles de almacenamiento y las rutas (paths) de acceso para las bases de datos.

**Nivel Conceptual tiene un Esquema Conceptual:** este describe la estructura completa de la base de datos para una comunidad de usuarios. El esquema conceptual es una descripción global de la base de datos que ignora los detalles de las estructuras de almacenamiento físico, y se concentra en describir entidades, tipos de datos, relaciones y limitaciones o controles. Un modelo de datos o de implementación puede ser utilizado a este nivel.

**Nivel Externo o de Visiones:** este incluye un número de esquemas externos o de imagenes del usuario. Cada esquema externo describe la visión o imagen de la base de datos para un grupo de usuarios de esta. Cada imagen describe la parte de la

base de datos en la que un grupo de usuarios particular esta interesado, y “esconde” el resto de la base de datos.

Notese que los tres esquemas son solo descripciones de datos, la unica información que en verdad existe esta en el nivel físico. Un DBMS basado en tres esquemas cada grupo de usuario se refiere unicamente a sus esquemas externo particular. El DBMS entonces transforma el pedimento de usuario especificado en el esquema externo en un procedimiento al esquema conceptual, y luego en uno al esquema interno para ser procesado en la base de datos almacenada.

Los procesos de transformar estos pedimentos en resultados entre los niveles son llamados mappings (direccionamiento).

## **Independencia de Datos**

La Arquitectura de tres esquemas puede ser utilizada para explicar el concepto de independencia de datos, el cual puede ser definido como la capacidad de cambiar el esquema a un nivel en un sistema de bases de datos, sin necesidad de cambiar el esquema a nivel inmediatamente superior. Pueden definirse dos tipos de independencias de datos:

### **1- Independencia de Datos Lógica:**

Es la capacidad de cambiar el esquema conceptual sin cambiar los esquemas externos o los programas de aplicaciones. Nosotros podemos tener la necesidad de cambiar el esquema conceptual para expandir la base de datos, ya sea añadiendo un nuevo tipo de registro o un nuevo campo.

### **2- Independencia de Datos física:**

Es la capacidad de cambiar el esquema interno sin cambiar los esquemas conceptuales o externos. Los cambios al esquema interno pueden ser necesarios porque algunos archivos físicos debar ser reorganizados. Ejemplo: crear estructuras de acceso adicionales para mejorar el desempeño en acceso y actualización.

## **Tipos de Bases de Datos por su Localización**

---

Las bases de datos por su localización física se pueden dividir en dos grupos principales: Centralizadas y Distribuidas:

### **Centralizadas:**

Un sistema de base de datos centralizado es aquel en el que los datos se encuentran en una sola localidad y controlado por un solo computador, es decir que se deba acceder a un lugar centrico para obtener los datos, este es el típico

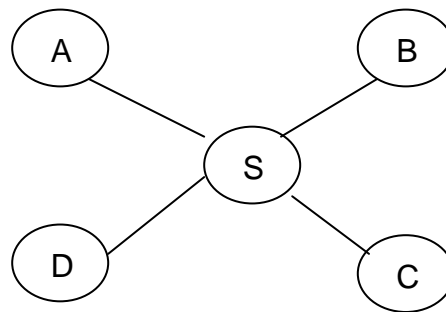
panorama normal de base de datos, Ejem: El sistema de base de datos de una clinica, supermercado etc.

Ventajas:

- Menor costo para el desarrollo de software y hardware.
- Menor posibilidad de errores.
- Menor tiempo extra de procedimientos.
- Un solo DBA central
- Detectar los fallos de forma mas rápida

Desventaja:

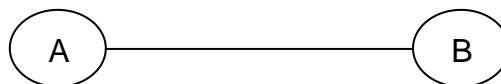
- Si falla el sistema principal se detendran todas las transacciones y operaciones.



### Base de Datos Distribuidas:

Un sistema distribuido de base de datos consiste en un conjunto de localidades, cada una de las cuales mantiene un sistema de base de datos local. Cada localidad puede procesar transacciones locales, aquella que solo acceden a datos que residen en esa localidad. Además una localidad puede participar en la ejecución de transacciones globales, aquellos que acceden a datos de varias localidades; la ejecución de transacciones globales requiere de comunicación entre las localidades.

Ejm: **localidad** = diferentes lugares donde se encuentran los sistemas de base de datos:



Las diferencias entre el sistema de base de datos centralizadas y distribuidas:

**Coste de instalación:** El coste de conectar físicamente las localidades del sistema.

**Coste de comunicación:** El coste en tiempo y dinero que implica enviar un mensaje desde la localidad A a la B.

**Fiabilidad:** La frecuencia con que falla una línea de comunicación o una localidad.

**Disponibilidad:** La posibilidad de acceder a información a pesar de fallos en algunas localidades o línea de comunicación.

Las localidades de un sistema de base de datos distribuido pueden estar dispersas de forma física, ya sea por un área geográfica extensa (como República Dominicana) o en un área reducida por ejemplo: un edificio o varios edificios adyacentes. La primera es un LAN (Local Area Network) y WAN (Wide Area Network).

Ventajas:

- Utilización compartida de los datos y distribución del control.
- Fiabilidad y disponibilidad.
- Agilización del proceso de consultas.

Desventajas:

- Coste de desarrollo de software.
- Mayor posibilidad de errores.
- Mayor tiempo extra de procedimientos.

El siguiente ejemplo servirá para ilustrar el presente tema:

Un sistema bancario que cuenta con 4 sucursales situadas en provincias diferentes de nuestro país, cada sucursal tiene su propio computador con un sistema de base de datos que incluye todas las cuentas que tiene esa sucursal. Así cada una de estas instalaciones es una localidad. También existe una localidad única que mantiene información de todas las sucursales del banco.

## **Arquitectura Cliente- Servidor**

Los sistemas cliente-servidor están contruidos de tal modo que la base de datos puede residir en un equipo central, llamado servidor y ser compartida entre varios usuarios. Los usuarios tienen acceso al servidor a través de una aplicación de cliente o de servidor:

- En un sistema cliente-servidor de dos estratos, los usuarios ejecutan una aplicación en su equipo local, llamado cliente, que se conecta a través de la red con el servidor que ejecuta SQL Server.
- La aplicación de cliente ejecuta las reglas de la compañía y el código necesario para presentar el resultado al usuario; también se conoce como cliente amplio.
- En un sistema cliente-servidor de varios componentes, la lógica de la aplicación de cliente se ejecuta en dos ubicaciones:



- El cliente reducido se ejecuta en el equipo local del usuario y se encarga de presentar resultados al usuario.
- La lógica de la compañía se encuentra en aplicaciones de servidor que se ejecutan en un servidor. Los clientes reducidos solicitan funciones a la aplicación de servidor, que, a su vez, es una aplicación multiproceso capaz de operar con varios usuarios simultáneos. La aplicación de servidor es la que abre las conexiones con el servidor de la base de datos y se puede ejecutar en el mismo servidor que la base de datos, o se puede conectar a través de la red con otro servidor que opere como servidor de base de datos. Éste es el escenario típico de las aplicaciones de Internet. Por ejemplo, una aplicación de servidor se puede ejecutar en un equipo con Microsoft Internet Information Services (IIS) y dar servicio a miles de clientes reducidos que se ejecuten en Internet o en una Intranet. La aplicación de servidor utiliza un grupo de conexiones para comunicarse con una copia de SQL Server. SQL Server puede estar instalado en el mismo equipo que IIS o en otro servidor de la red.

El tener los datos almacenados y administrados en una ubicación central ofrece varias ventajas:

- Todos los elementos de datos están almacenados en una ubicación central en donde todos los usuarios pueden trabajar con ellos.
- No se almacenan copias separadas del elemento en cada cliente, lo que elimina los problemas de hacer que todos los usuarios trabajen con la misma información.
- Las reglas de la organización y las reglas de seguridad se pueden definir una sola vez en el servidor para todos los usuarios. Esto se puede hacer en una base de datos mediante el uso de restricciones, procedimientos almacenados y desencadenadores. También se puede hacer en una aplicación de servidor.
- Los servidores de base de datos relacionales optimizan el tráfico de la red al devolver sólo los datos que la aplicación necesita. Por ejemplo, si una aplicación que trabaja con un servidor de archivos tiene que presentar la lista de los representantes comerciales de Oregón, debe obtener el archivo de empleados completo. Si la aplicación trabajara con un servidor de bases de datos relacionales, enviaría este comando:

```
SELECT first_name, last_name FROM employees WHERE emp_title = 'Sales Representative' AND emp_state = 'OR'
```

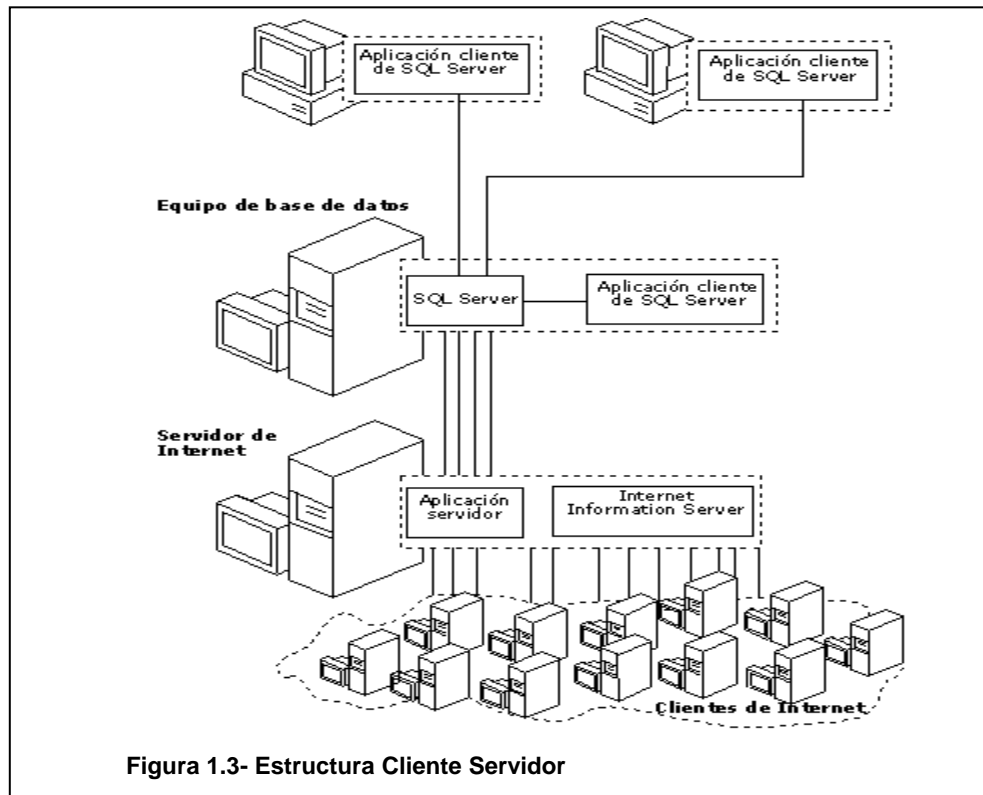
La base de datos relacional sólo devuelve los nombres de los representantes comerciales de Oregón, no toda la información de todos los empleados.

- Los gastos en hardware se pueden minimizar. Como los datos no están almacenados en los clientes, éstos no tienen que dedicar espacio de disco a almacenarlos. Los clientes tampoco necesitan la capacidad de proceso para administrar los datos localmente y el servidor no tiene que dedicar capacidad de proceso para presentar los datos.

El servidor se puede configurar para optimizar la capacidad de E/S de disco necesaria para obtener los datos y los clientes se pueden configurar para optimizar el formato y presentación de los datos obtenidos desde el servidor.

El servidor puede estar situado en una ubicación relativamente segura y estar equipado con dispositivos como Sistemas de alimentación ininterrumpida (SAI), lo que resulta más económico que si se protegieran todos los clientes.

Las tareas de mantenimiento como las copias de seguridad y restauración de los datos son más sencillas porque están concentradas en el servidor central.



## Sistemas de bases de datos de escritorio

Aunque SQL Server funciona muy eficientemente como servidor, también se puede utilizar en aplicaciones que necesiten bases de datos independientes almacenadas de forma local en el cliente.

SQL Server se puede autoconfigurar dinámicamente para que se ejecute más eficientemente con los recursos disponibles en el cliente, sin tener que dedicar un administrador de bases de datos a cada cliente. Los fabricantes de aplicaciones también pueden incrustar SQL Server como componente de almacenamiento de datos en sus aplicaciones.