

TEMA 1: INTRODUCCIÓN A LAS BASES DE DATOS

1.1 INTRODUCCIÓN

Un sistema de base de datos es básicamente un sistema computarizado para llevar registros. Es posible considerar a la base de datos como un armario electrónico para archivar; es decir es un depósito o contenedor de una colección de archivos de datos computarizados. Los usuarios del sistema pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo:

- Agregar nuevos archivos vacíos a la base de datos.
- Insertar datos dentro de los archivos existentes.
- Recuperar datos de los archivos existentes.
- Modificar datos de los archivos existentes.
- Eliminar datos de los archivos existentes.
- Eliminar archivos existentes.

1.2 ¿QUÉ ES UN SISTEMA DE BASE DE DATOS?

Un sistema de Bases de Datos es básicamente un sistema computarizado para guardar registros; es decir, es un sistema computarizado cuya finalidad es almacenar información y permitir a los usuarios recuperar, y actualizar esa información con base en peticiones.

Un sistema de base de datos comprende cuatro componentes: **datos, hardware, software y usuarios.**

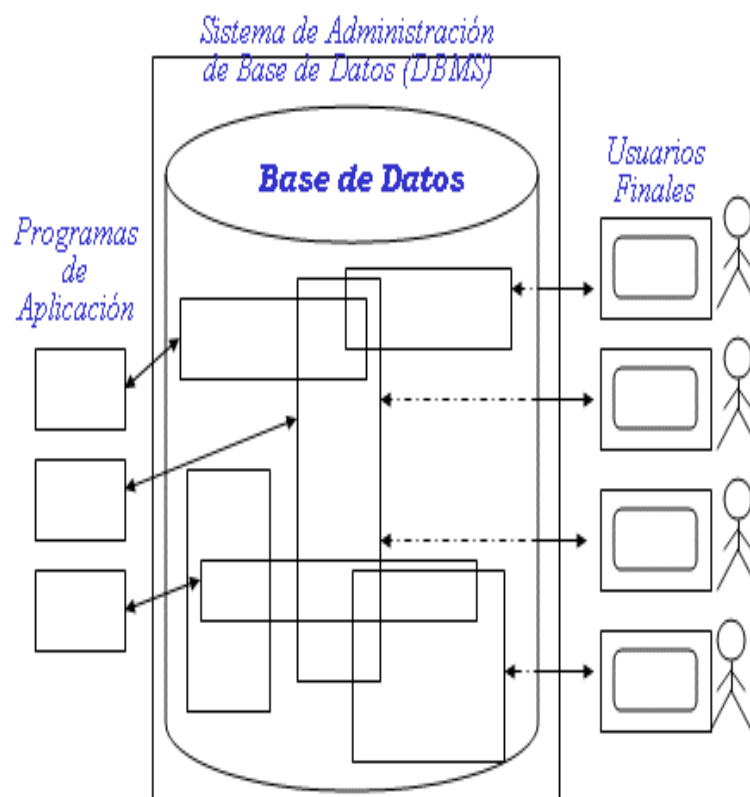


Figura 1: Sistema de Base de Datos

Datos:

Las características más importantes de la información en estos sistemas es que va a estar **integrada y compartida**.

Integrada: La Base de Datos (BD) puede considerarse como una unificación de varios ficheros de datos, que son tratados como uno solo, y en el que se ha eliminado totalmente, o en parte, la redundancia de datos.

Compartida: Los datos pueden compartirse entre varios usuarios distintos. Es posible que varios de estos usuarios accedan al mismo tiempo al mismo elemento de información (acceso concurrente).

Hardware:

Los componentes hardware del sistema constan de:

- Los volúmenes de almacenamiento secundario (principalmente discos magnéticos) que se emplean para contener los datos almacenados, junto con los dispositivos asociados de E/S (unidades de disco, etc.), los controladores de dispositivos, los canales de E/S, entre otros; y
- Los procesadores de hardware y la memoria principal asociada usados para apoyar la ejecución del software del sistema de base de datos.

Software:

Entre la base de datos física, es decir, los datos como están almacenados físicamente y los usuarios del sistema, hay una capa de software conocida como el administrador de base de datos o el servidor de base de datos; o más comúnmente como el **sistema gestor de base de datos (SGBD)**. Todas las solicitudes de acceso a la base de datos son manejadas por el SGBD. Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Aplicaciones de los SGBD:

- En el **Sistema Bancario de Nicaragua** para entre otros aspectos ofrecer información actualizada a los clientes de estos Bancos sobre el estado de sus cuentas, préstamos y transacciones bancarias en general.
- En las **Compañías de Líneas Aéreas** para efectuar reservaciones, para obtener información sobre los cupos disponibles en una determinada ruta de vuelo, etc.
- En las **Universidades** para informar a los estudiantes sobre las asignaturas disponibles al momento de su matrícula, información sobre las notas obtenidas por el estudiante, matrícula de las asignaturas en un determinado año académico.
- **Transacciones con tarjetas de Crédito:** Compras y Generación del informe mensual a cada cliente.
- **Telecomunicaciones:** Para llevar control de las llamadas efectuadas por los usuarios, la generación mensual de las facturas.
- **Finanzas:** Sistemas contables, Almacenamiento de información sobre grandes empresas, control de ventas y compras de documentos financieros formales como bonos y acciones de la bolsa.
- **Producción:** Para gestionar la cadena de producción de una empresa, seguimiento de la producción en las diferentes sucursales de la empresa, control de inventario.

- **Recursos Humanos:** Información general de los empleados, generación de nominas.
- **Aplicaciones Internet:** Compra de Libros en línea o búsqueda de uno determinado, se está accediendo a una base de datos, cuando el usuario se conecta a una página Web de un determinado banco para conocer el estado de su cuenta, dicha información está contenida en una base de Datos.

Usuarios:

Hay tres grandes clases de usuario:

Primero, hay **programadores de aplicaciones** responsables de escribir los programas de aplicación de base de datos en algún lenguaje de programación como COBOL, C++, Java o algún lenguaje de de alto nivel de la “cuarta generación”.

En consecuencia, la segunda clase de usuarios son los **usuarios finales**, quienes interactúan con el sistema desde estaciones de trabajo o terminales en línea, o bien puede usar una interfaz proporcionada como parte integral del software del sistema de base de datos.

El tercer tipo de usuario, es el **administrador de base de datos o DBA**, los cuales son responsables del control y manejo del sistema de base de datos.

1.3 ¿QUÉ ES UNA BASE DE DATOS (BD)?

Una **base de datos** o **banco de datos** (en inglés: database) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.









Puede definirse, como una: “Colección o depósito de datos integrados con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción, únicas para cada tipo de datos, han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos”.

1.4 ¿POR QUÉ UTILIZAR UNA BASE DE DATOS?

- **Compactación:** No hay necesidad de archivos en papel voluminosos.
- **Velocidad:** La máquina puede recuperar y actualizar datos más rápidamente que un humano.
- **Menos trabajo laborioso:** Se puede eliminar gran parte del trabajo de llevar los archivos a mano. Las tareas mecánicas siempre las realizan mejor las máquinas.
- **Actualidad:** En el momento que la necesitemos, tendremos a nuestra disposición información precisa y actualizada.

Un sistema de base de datos ofrece un control centralizado de sus datos. esta situación contrasta en gran medida con la que se encuentra en una empresa que no cuenta con un sistema de base de datos, en donde por lo general cada aplicación tiene sus propios archivos privados, de modo que los datos están muy dispersos y son difíciles de controlar de una forma sistemática.

1.5 CARACTERÍSTICAS DESEABLES DE UNA BASE DE DATOS

-  **Versatilidad para representar la información:** Ofrecer diferentes visiones de la información que se almacena en la BD.
-  **Desempeño:** Debe dar respuesta en un tiempo adecuado, permitiendo el acceso simultáneo al mismo o diferentes datos.
-  **Redundancia de datos mínima.**
-  **Capacidad de acceso:** Debe responder en tiempo adecuado a consultas previstas e imprevistas.
-  **Simplicidad:** Cambios en los requerimientos no deben suponer grandes cambios en el modelo de datos.
-  **Seguridad:** Capacidad para proteger los datos contra pérdidas totales y/o parciales,
 - Contra destrucción causada por el entorno (fuego, inundación, ...)
 - Contra destrucción causada por fallos del sistema.
 - Contra accesos no autorizados a la Base de Datos.
 - Contra accesos indebidos a los datos.
-  **Afinación:** Organización de datos afines para obtener buenos tiempos de respuesta.
-  **Integridad:** Que los datos sean correctos y se correspondan a los requerimientos del dominio.
 - Integridad frente a fallos HW o SW o de acceso concurrente.
 - Integridad asegurando que los datos se ajustan a los requerimientos del problema.

1.6 SISTEMAS DE BASES DE DATOS FRENTE A SISTEMAS DE ARCHIVOS

Como un aspecto histórico del desarrollo de los sistemas de Bases de datos nos encontramos con los **Sistemas de Archivos** (filesystem en inglés), los cuales estructuran la información guardada en una unidad de almacenamiento (normalmente un disco duro) de una computadora, que luego será representada ya sea textual o gráficamente utilizando un gestor de archivos. Esta metodología para el procesamiento de los datos dio paso a los sistemas de base de datos actuales, por las problemáticas generadas para almacenar y obtener de forma eficiente los datos requeridos algunas de las cuales se mencionan a continuación:

Redundancia e Inconsistencia de los Datos: Esta problemática se origina del hecho de que tanto los archivos de datos como los programas de aplicación son creados por diferentes programadores en un período de tiempo más o menos largo, programas generalmente escritos en diferentes lenguajes con archivos de datos que utilizan formatos diferentes y por tanto incompatibles entre sí. La independencia de estos archivos genera **redundancia** (Exceso innecesario) de la información así por ejemplo la dirección y el teléfono de los clientes de un banco puede estar contenida en un archivo de cuentas de ahorro y en otro de cuentas corrientes. Por otra parte el problema de la **inconsistencia de los datos** es factible en el sistema de archivos dado que la redundancia tiene como consecuencia, en la medida que los datos crecen la posible representación de la copia de un dato de forma diferente en diferentes archivos así supongamos que un determinado cliente tiene su dirección en 2 archivos diferentes e independientes y este la actualiza solamente en uno de ellos, esto produciría la inconsistencia de los datos.

Dificultad en el acceso de los Datos: El tener la información aislada produce dificultad en obtener la información requerida de los archivos. Así supongamos que se desea averiguar en un determinado banco el nombre de todos los clientes que viven en el distrito 2 de Managua, dado que no se ha requerido de esta información, el programa no existe, solamente podemos acceder a los nombres de todos los clientes del banco, claro está que el o los programadores del mencionado banco pueden diseñar dicho programa pero no es una solución eficiente ya que si posteriormente se requiere los nombres de los clientes del

distrito 2 pero que tengan cuentas de ahorro superiores a 5,000 córdobas, de nuevo se tendrá que diseñar el respectivo programa, en otras palabras para cada consulta particular se tendrá que esperar un determinado tiempo para el diseño de la misma y el control adicional que se debe invertir para averiguar si ya fue diseñada o no una determinada consulta dado que el número de ellas crecería de forma considerable en el tiempo.

Aislamiento de los datos: Los programas de Aplicación se tornan cada vez más difíciles de diseñar debido a la independencia que existe entre sí entre los diferentes archivos, además que la información en general está dispersa.

Problemas de Integridad: En general los datos almacenados deben de cumplir ciertos tipos de restricciones de consistencia así por ejemplo que una cuenta bancaria sea siempre mayor o igual a 1000 córdobas, como se ha indicado antes se pueden diseñar los programas respectivos siempre y cuando la información se encuentre en un mismo archivo, el problema resulta cuando dichas restricciones de integridad se refieren a datos que se encuentran en archivos diferentes con formatos diferentes.

Problemas de atomicidad: La atomicidad en este contexto se refiere a un conjunto de operaciones (Una Transacción) que se deben de efectuar en la base de datos de manera completa, lo cual no siempre se produce debido a la ocurrencia de fallos, así por ejemplo supongamos que una transacción consta de 10 operaciones y el fallo se produce al comenzar la sexta operación, en este caso decimos que se ha violado la atomicidad de la transacción. Por lo antes expuesto el sistema debe de contar con un mecanismo para mantener la base de datos en un estado consistente, este existe en los SGBD pero no en un Sistema de Archivos.

Anomalías en el acceso concurrente: Los sistemas operativos actuales hacen posible diseñar sistemas de bases de datos con módulos encargados de resolver la problemática del acceso concurrente a los datos, este aspecto sería muy difícil de implementar en los sistemas de archivos debido a la independencia total de los datos y en general a la diferencia de formato de estos. Una de estas anomalías puede verse claramente en el siguiente ejemplo: Considérese una cuenta bancaria A con 500 córdobas, supongamos que dos clientes retiran fondos de dicha cuenta, uno retira 50 y el otro 100 en aproximadamente el mismo tiempo (Desde dos terminales distintas), ambos usuarios leen de disco 500, y luego el disco será actualizado a 450 o 400 dependiendo quien guarde por ultimo, esto por supuesto dejaría la base de datos inconsistente.

Problemas de Seguridad: En un sistema de archivos este aspecto es también difícil de implementarlo ya que los archivos no conforman un sistema y sería extremadamente difícil el acceso a determinada porción de los datos, este aspecto de la seguridad es muy importante en las bases de datos debido a que en una empresa no interesa y no es conveniente que todos tengan acceso a todos los datos así en un sistema bancario el personal de nóminas no tiene porque conocer la información almacenada sobre las cuentas de los clientes.

1.7 VISIÓN DE LOS DATOS

ANSI/SPARC es un grupo de normalización creado en 1969 para estudiar el impacto de los SGBD. en los sistemas de información y cuyos resultados, publicados en 1975 propusieron el uso de tres niveles de descripción de datos:

Nivel interno o físico. Se refiere al almacenamiento físico en el se describe cómo se almacenan realmente los datos en memorias secundarias, en qué archivos, su nombre y dirección. También estarán los registros, longitud, campos, índices y las rutas de acceso a esos archivos.

Nivel Conceptual. En el se describen cuáles son los datos reales almacenados en la Base de Datos y que relaciones existen entre ellas. Este nivel lo definen los administradores de la Base de Datos que son los que deciden que información se guarda en la BD. Este nivel corresponde a la estructura organizacional de los datos obtenida al reunir los requerimientos de todos los usuarios, sin preocuparse de su organización física ni de las vías de acceso. Podría contener:

- Entidades del mundo real (clientes, artículos, pedidos, ...)
- Atributos de las entidades (nombre_cliente, NIF, ...)
- Asociaciones entre entidades (compra de artículos)
- Restricciones de integridad (son las normas que deben cumplir los datos).

Nivel externo o vistas. Es el nivel más cercano al usuario y representa la percepción individual de cada usuario. Si los niveles interno y conceptual describen toda la BD, este nivel describe únicamente la parte de datos para un usuario o grupo de usuarios. Habrá usuarios que podrán acceder a más de un esquema externo y uno de éstos puede ser compartido por varios usuarios, se protege así el acceso a los datos por parte de personas no autorizadas. A la hora de construir un esquema externo:

- Se pueden omitir una o más entidades del sistema.
- Se pueden omitir uno o más atributos de una entidad.
- Se pueden omitir una o más relaciones entre los datos.
- Se pueden cambiar el orden de los atributos.

Para una BD habrá un único esquema interno, un único esquema conceptual, pero puede haber varios esquemas externos.

Veamos la diferencia entre estos tres niveles mediante un ejemplo:

Comencemos con el tercer nivel: **Nivel de Vista de Cliente**

Nombre_Cliente	Id_Cliente	Calle_Cliente	Ciudad_Cliente
Juan López	281-060483-0006D	Las Flores	Granada
Pedro Tercero	081-160483-0006D	Los Tulipanes	León
María Prado	281-161278-0701S	Las Rosas	Managua

Segundo Nivel (Nivel Lógico)

Type Cliente = Record

Nombre_Cliente : Array[1..50] of Char ;
Id_Cliente : Array[1..20] of Char ;
Calle_Cliente : Array[1..30] of Char ;
Ciudad_Cliente : Array[1..50] of Char ;

End ;

Primer Nivel (Nivel Físico)

En este nivel se describe como la información es almacenada físicamente en disco, en este nivel se especifica, el formato de archivo a utilizar, el número de bloques que utilizan los

datos, el número de bytes que utiliza cada registro para su almacenamiento, el lugar que ocupa en cada bloque así en el ejemplo anterior cada registro ocupará 150 bytes (Un Byte por carácter), el formato es de longitud fija, siendo el primero:

Juan López, 2134, Las Flores, Granada. Este nivel y el nivel 2 es de suma importancia para los diseñadores del SGBD pues de él en gran medida va a depender la velocidad de acceso a los datos, por otra parte el nivel de vista está orientado a los usuarios finales del sistema.

Una vez diseñado el sistema de gestión de base de datos, el usuario diseña aplicaciones para una determinada empresa para lo cual es necesario modelar los datos para que estos y sus relaciones puedan ser representados mediante una colección de objetos, de forma tal que el diseño se pueda presentar de forma sencilla, para que se puedan efectuar modificaciones al mismo y pueda ser compartido por el grupo de usuarios diseñadores.

Dependiendo de la metodología utilizada en el diseño del SGBD, se cuenta con los siguientes Modelos de Datos:

1.8 MODELO DE LOS DATOS

1.8.1 El modelo Relacional

En este modelo los datos y las relaciones entre ellos son representados por un solo objeto denominado Tabla donde cada tabla está compuesta por Columnas y filas. A continuación se presenta un ejemplo de este modelo:

Id_cliente	Nombre_Cliente	Calle_Cliente	Ciudad_Cliente
1	Juan	Sutiava	León
2	Pedro	Jalteva	Granada
3	Carlos	Monzón	Matagalpa
4	María	Central	León

Tabla: Clientes

Número_Cuenta	Saldo
C1	1000
C2	1200
C3	1500
C4	600
C5	1700

Tablas: Cuenta

Id_cliente	Número_Cuenta
1	C2
2	C3
3	C4
4	C1
2	C5

Tabla: Cuenta_Cliente

En este caso la tabla **Clientes** almacena información sobre cada uno de los clientes, del Banco Popular por otra parte la tabla **Cuenta** almacena información sobre el estado de cada una de las cuentas del mismo banco y finalmente la tabla **Cuenta_Cliente** lo que almacena es una relación entre los clientes y las cuentas, es decir cuales son los clientes de este banco que poseen cuentas y cuales son estas.

El modelo relacional es un ejemplo de modelo basado en registros pues en general la implementación se realiza utilizando registros de formato fijo de varios tipos.

1.8.2 Modelo Entidad Relación

El modelo de base datos Entidad Relación está basado en una percepción del mundo real que consta de: Una colección de objetos básicos denominados Entidades y de Relaciones entre estos objetos.

En el ejemplo anterior: **1, Juan, Sutiava, León** es una entidad que define de forma única a un Cliente del Banco Popular,

Por otra parte: **C5, 1700** identifica de forma única el estado de esta cuenta es decir define una Cuenta.

Finalmente **2,C5** es una relación la cual indica que el cliente cuyo **Id_cliente** es 2 tiene asignado la cuenta cuyo código es C5, decimos entonces que **2,C5** vincula a dos entidades: 2, Pedro, Jalteva, Granada y las entidad : C5, 1700.

Los componentes básicos tanto de las entidades como la de las relaciones reciben el nombre de **atributos**. En este modelo las entidades del mismo tipo se denominan: **Conjunto de entidades**, similarmente las relaciones del mismo tipo se denominan **Conjunto de Relaciones**, ambas tiene una representación gráfica en este modelo.

Componentes Básicos del Modelo:

- **Rectángulos:** Representan Conjunto de entidades.
- **Elipses:** Representan atributos.
- **Rombos:** Representan Conjunto de Relaciones.
- **Líneas:** Unen los atributos a los conjuntos entidades o Conjuntos Relaciones, unen también Conjunto de entidades con Conjunto de relaciones.

A continuación se presenta el gráfico completo del diagrama Entidad-Relación referido al Ejemplo Bancario anterior:

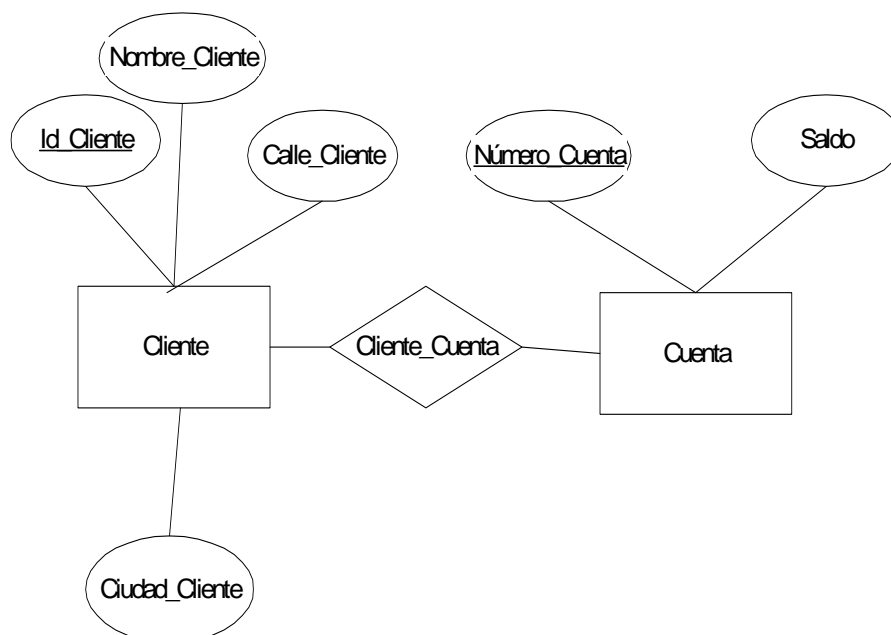


Figura 2: Ejemplo de Diagrama E-R

1.8.3 Otros modelos de Datos:

- **El modelo de Datos Orientado a Objetos:** Este es un modelo que en los últimos años ha recibido una atención creciente de parte de los diseñadores de Base de Datos, este podemos verlo como una extensión del modelo E-R fortalecido por la programación orientada a objetos
- **El modelo de Datos Relacional Orientado a Objetos:** Este modelo combina las características del modelo de datos orientado a objetos y las del modelo relacional.
- **Los Modelos de Datos Semiestructurados :** Estos se distinguen porque permiten asignar conjuntos de atributos diferentes a las filas individuales a diferencia de los modelos mencionados anteriormente en que todos los objetos básicos que componen los modelos deben de tener los mismos atributos.

1.9 LENGUAJES DE BASES DE DATOS

Un sistema de base de Datos proporciona dos tipos de lenguajes:

Un lenguaje de Definición de Datos (LDD): Tiene como función primordial el definir la estructura de la base de datos: Nombre de las tablas, sus atributos, tipos de los atributos, etc.

Lenguaje de Manipulación de Datos (LMD): Tiene como función trabajar sobre los datos de la base de datos para obtener la información pertinente o modificar la base de Datos.

El lenguaje utilizado ampliamente por la mayoría de SGBD es el denominado **SQL (Structured Query Language)**, el cual se puede utilizar tanto como LDD como LMD.