Unidad 3: RECURSOS DE TECNOLOGIA DE INFORMACIÓN - Información

Bases de Datos: Diseño, gestión, nuevas estructuras. Administración de bases de datos: técnicas de generación de información.

Archivos. Concepto

Un archivo es un elemento de información conformado por un conjunto de registros. Estos registros a su vez están compuestos por una serie de caracteres o bytes.

Algunas personas consideran el almacenamiento de los datos como la esencia del sistema de información. El impacto de la estructura de datos sobre la estructura de programa y la complejidad procedimental hace que el diseño de datos tenga una gran influencia en la calidad del software.

Independientemente de las técnicas de diseño usadas, los datos bien diseñados pueden conducir a una mejor estructura de programa, a una modularidad que sea efectiva y a una complejidad procedimental reducida.

Los objetivos generales del diseño de la organización de almacenamientos son:
1) ***Disponibilidad de datos:*** Los datos deben estar disponibles para cuando el usuario desee usarlos. Sólo se debe negar el acceso a datos reservados por razones de incumbencia y legales.
2) ***Accesibilidad:*** deben ser fáciles de acceder por los usuarios. Si no se puede acceder fácilmente, no serán utilizados (y eventualmente pueden ser "inventados"). No deben reflejar las necesidades de un sector particular, o serán difíciles de acceder para los otros.
3) ***Integridad de datos:*** Los datos deben ser *precisos* (cada valor almacenado debe estar dentro de un rango aceptable respecto del valor real) y *consistentes* (no deben reflejar realidades distintas ya en el tiempo (las Cuentas a Pagar no cierran contra las Facturas Pendientes) como en los datos relacionados (un Gerente de Area no puede ser un empleado de categoría mínima). El conjunto de datos debe representar fielmente a la organización.
4) ***Almacenamiento eficiente***.
5) ***Actualización y recuperación eficiente***.
6) ***Recuperación dirigida de la información:*** la información obtenida de los datos almacenados debe contar con un formato útil que facilite la administración, la planeación, y el control o la toma de decisiones.

Los archivos, alojados en dispositivos de almacenamiento conocidos como memoria secundaria, pueden almacenarse de dos formas diferentes: archivos convencionales o bases de datos.

Archivos Convencionales y Bases de Datos

En un sistema de información basado en computadora se cuenta con dos enfoques para el almacenamiento de los datos. El primer método consiste en almacenar los datos en archivos individuales, exclusivos para una aplicación en particular. La figura 1 ilustra una organización con diversos sistemas de información que utilizan archivos convencionales independientes. Existen tres archivos individuales ARCHIVO- VENTAS, el cual contiene la información de los datos históricos, ACTIVIDADES-ACTUALES, las cuales se actualizan con frecuencia y ARCHIVO-PERSONAL, que contiene las direcciones, los títulos y datos

similares. Observe que NUM y NOMBRE existen en ambos archivos. Además del esfuerzo adicional requerido para introducir el NOMBRE hasta tres veces, el cambio de nombre (por ejemplo, por un cambio en el estado civil) requeriría de la actualización de tres archivos individuales.

ARCHIVO VENTAS				
NUM	NOMBRE	86	87	88
1	Jose Perez	1000	400	900
2	Luis Gomez	500	1000	1100
3	Pedro Marmol	700	1000	1200
4	Ana Rodriguez	1500	1400	1500

ARCHIVO ACTIVIDAD ACTUAL			
NUM	NOMBRE	AREA	88
1	JOSE PEREZ	A	1000
2	LUIS GOMEZ	A	1100
3	PEDRO MARMOL	B	1200
4	ANA RODRIGUEZ	C	1500

ARCHIVO PERSONAL						
NUM	DEPTO	NOMBRE	DIRECCION	TELEFONO	SALARIO	TITULO
1	A	Pérez José	Bs.As.58	4222222	600	SEC
2	A	Gómez Luis	La Rioja 45	4211111	600	SEC
3	B	Mármol Pedro	San Luis 96	4233333	600	SEC
4	C	Rodríguez Ana	Jujuy 180	4244444	600	SEC

FIGURA 1 El uso de archivos separados con frecuencia implica que los mismos datos se almacenen en más de un lugar y en muchos casos de manera inconsistente

El segundo enfoque para el almacenamiento de datos en un sistema basado en computadora involucra la elaboración de una base de datos. Una base de datos es un almacenamiento de datos formalmente definido, controlado centralmente para intentar servir a múltiples y diferentes aplicaciones. La figura 2 muestra cómo los diferentes usuarios de distintos departamentos dentro de una organización comparten la misma base de datos. Un usuario puede seleccionar una parte de la base de datos, como se muestra en el conjunto de datos uno, o ciertos renglones en el conjunto de datos dos. El conjunto de datos tres contiene columnas selectas y el conjunto de datos cuatro utiliza ciertos renglones para calcular los totales. Estos conjuntos de datos se superponen indicando que si existieran datos que fueran redundantes en archivos convencionales, solamente se almacenarían una sola vez en la base de datos.

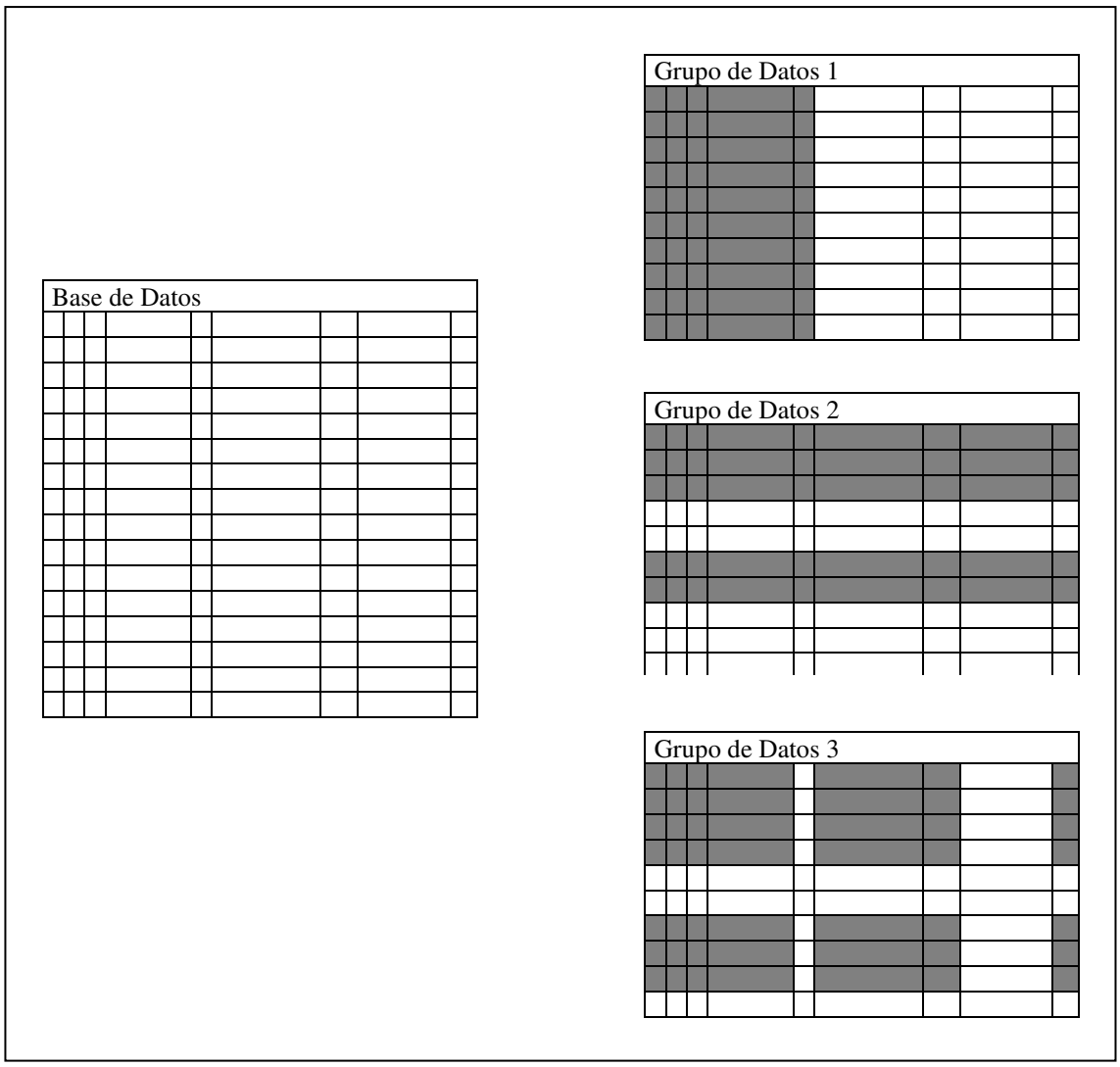


FIGURA 2 El enfoque de base de datos permite que diferentes usuarios compartan la misma base de datos, teniendo acceso a diferentes grupos de datos.

Archivos convencionales

Los archivos convencionales, sin duda alguna, permanecen como una manera práctica de almacenar los datos de ciertas aplicaciones. Un archivo convencional se puede diseñar y elaborar de manera rápida, reduciendo los problemas de disponibilidad de datos y de seguridad. Cuando el diseño de los archivos se realiza de manera cuidadosa, toda la información necesaria queda incluida y se reduce el riesgo de omitir datos de manera accidental.

En muy raras ocasiones, un analista se enfrentará a una situación en la cual no existan aplicaciones.

Existe la probabilidad de que se encuentren actualmente en servicio archivos separados. Si el tiempo de desarrollo es una consideración importante, el analista de sistemas no puede involucrarse en el problema de rehacer el almacenamiento de los datos, sólo con el fin de lograr un enfoque de base de datos. La solución obvia en función de las restricciones de

tiempo consiste en limitar el alcance del proyecto, diseñando otro archivo separado para la nueva aplicación.

La velocidad de procesamiento es otra ventaja para el uso de archivos. Hay posibilidad de elegir una técnica óptima para el procesamiento de los archivos de una sencilla aplicación, pero llega a ser imposible alcanzar un diseño óptimo para tareas muy variadas. En consecuencia, si el analista de sistemas se enfrenta al diseño de un sistema para una aplicación específica, cuando la eficiencia del procesamiento es de gran relevancia, el mejor enfoque puede ser el diseño de un archivo individual para tal propósito.

El uso de archivos individuales tiene diversas consecuencias. Veamos el siguiente ejemplo para ilustrarlo:

Considérese parte de una empresa bancaria de ahorros que guarda la información sobre todos los clientes y sus cuentas de ahorro. 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 enfoque tiene un número de desventajas importante:

- **Falta de potencial para evolucionar:** con frecuencia, los archivos se diseñan con base en las necesidades inmediatas. Cuando llega a ser importante la consulta del sistema mediante una combinación de ciertos atributos, tales atributos pudieran encontrarse en archivos separados, o simplemente no existir. El rediseño de archivos implica a menudo que los programas que los accesan, deban escribirse nuevamente de manera acorde. Esto implica un incremento en el tiempo de programación para el archivo, para el desarrollo y mantenimiento de programas.

- **Redundancia e inconsistencia de los datos.** Puesto que los archivos y los programas de aplicación son creados por distintos programadores durante un periodo largo de tiempo, es probable que los archivos tengan diferentes formatos y los datos 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 ahorros y en un archivo que consta de registros de cuentas de cheques. Esta redundancia aumenta los *costes de almacenamiento* y *acceso*. Además, puede llevar a *inconsistencias* de los datos – esto es, las diversas copias de los mismos datos no concuerdan entre sí. Por ejemplo, una dirección cambiada de un cliente puede

estar reflejada en los registros de cuentas de ahorros, pero en ningún sitio más del sistema. Resultados de inconsistencia de los datos.

- **Dificultad para tener acceso a los datos.** Supóngase que uno de los gerentes del banco necesita averiguar los hombres 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. Puesta que esta solicitud no fue prevista cuando se diseño 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 tomar la lista de clientes y extraer la información necesaria manualmente, o pedir al departamento de procesamiento de datos que ponga a un programador de aplicaciones 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 solo aquellos clientes con un saldo de 10000 dólares o más. Como se esperaba, no existe un programa que genere 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. Considérese una cuenta bancaria A, con 500 dólares. Si dos clientes retiran fondos (digamos 50 y 100 dólares, respectivamente) de la cuenta A casi al mismo tiempo, el resultado de las ejecuciones concurrentes puede dejar la cuenta en un estado incorrecto (o inconsistente). En particular, la cuenta puede contener 450 o 400 dólares, en vez de 350 dólares Para prevenir esta posibilidad, debe mantenerse alguna forma de supervisión en el sistema. Puesto que se puede acceder a tos 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 deben poder acceder a todos los datos. Por ejemplo, en un sistema bancario, el personal de las nóminas solo 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:** la integración de los datos llega a convertirse en una causa de preocupación, ya que los cambios en un archivo requerirán también la modificación de ciertos datos en otros archivos. Aquellos archivos utilizados esporádicamente bien pueden quedar ignorados durante la fase de la actualización. Por otra parte, 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 sistemas de gestión de bases de datos.

Las formas en las cuales pueden organizarse son archivos secuenciales o archivos directos. En los archivos secuenciales los registros están almacenados en una secuencia que depende de algún criterio definido. Por ejemplo, pueden almacenarse los registros de los empleados de la empresa de manera secuencial de acuerdo con el departamento al que pertenecen o de acuerdo a su antigüedad.

Si se desea consultar o modificar información, también es necesario buscar uno por uno en los registros hasta encontrarla.

Los archivos directos permiten acceder directamente un registro de información sin tener que buscar uno a uno por todos los registros del archivo, utilizando una llave de acceso dentro del archivo.

Bases de Datos

Concepto

Es un conjunto exhaustivo no redundante de datos estructurados, organizados independientemente de su utilización y de los programas utilizados para su administración, accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente.

En las bases de datos la información esta integrada y compartida.

La información la comparten en cualquier momento y al mismo tiempo muchos usuarios que pueden tener diferentes intereses y objetivos y, muchas aplicaciones.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas. Por este motivo es importante conocer la forma en que están estructuradas las bases de datos y su manejo.

Las bases de datos no son meramente una colección de archivos. Más bien, una base de datos es una fuente central de datos significativos, los cuales son compartidos por numerosos usuarios para diversas aplicaciones.

Los objetivos de eficiencia de la base de datos son:

1) Hay que asegurar que los datos puedan ser compartidos por los usuarios, para una variedad de aplicaciones.
2) Que el mantenimiento de los datos sea preciso y consistente.
3) Hay que asegurar que todos los datos requeridos para las aplicaciones presentes y futuras se encuentren siempre disponibles.

4) Permitir que la base de datos evolucione y se adapte a las necesidades crecientes de los usuarios.
5) Permitir que los usuarios desarrollen su propia visión de los datos, sin preocuparse por la manera en que los datos se encuentren almacenados físicamente.

La anterior lista de objetivos nos advierte las ventajas y desventajas del enfoque de la base de datos.

Primero, al compartir los datos, significa que éstos deben almacenarse por lo menos una sola vez. Esto a su vez apoya que se mantenga la integridad de los datos, ya que el cambio de los datos se realizará de manera más sencilla y confiable si éstos aparecen una vez y no en varios archivos.

Cuando un usuario necesite un dato particular, una base de datos con un buen diseño se debería anticipar tal necesidad (y quizás ya hayan sido utilizados en otra aplicación). En consecuencia, los datos tendrán mayor probabilidad de encontrarse disponibles en una base de datos más que en un sistema de archivos convencionales.

Una base de datos con un buen diseño también llega a ser más flexible que dos archivos separados, esto significa que una base de datos llega a evolucionar conforme se modifican las necesidades de los usuarios y de sus aplicaciones. Finalmente, el enfoque de base de datos tiene la ventaja de permitir que los usuarios expongan sus puntos de vista sobre los datos, sin necesidad de preocuparse de la estructura presente de la base de datos o de su ubicación física.

La primera desventaja del enfoque de base de datos es que todos los datos se almacenan en un solo lugar y en consecuencia, los datos son más vulnerables a accidentes y requerirán de un respaldo completo. Esto en la actualidad ha sido mejorado con técnicas avanzadas de bases de datos distribuidas, las cuales no están almacenadas totalmente en un solo lugar físico, están segmentadas y se comunican por medio de enlaces de comunicaciones a través de una red de computadoras distribuidas geográficamente.

Existe el riesgo de que quien administra la base de datos se convierta en el único privilegiado o habilitado para estar cerca de los datos. Y los procedimientos burocráticos requeridos para modificar o para actualizar la base de datos pueden llegar a ser insuperables. Otras desventajas para la administración de los datos como recurso se presentan al intentar satisfacer dos objetivos de eficiencia:

1) Reducción del tiempo requerido para insertar, actualizar, eliminar y recuperar los datos en tiempos tolerables.
2) Mantenimiento del costo del almacenamiento de datos en una cantidad razonable.

Es necesario recordar que una base de datos no puede optimizar la recuperación de los datos para una aplicación en especial, ya que deberá compartirse con numerosos usuarios y con varias aplicaciones. Además, se puede llegar a requerir de cierto software adicional para el DBMS (Sistema Gestor de Bases de Datos) y, ocasionalmente, también se necesitará una computadora de mayor capacidad.

El enfoque de la base de datos es un concepto que se vuelve cada vez más relevante. El uso de base de datos de relacionales mediante microcomputadoras indica el grado de difusión que este concepto ha alcanzado entre los usuarios. Con este enfoque, los usuarios toman una parte de la base de datos central y cargan sus computadoras personales. Luego

estas pequeñas bases de datos se utilizan para emitir reportes o contestar consultas específicas del usuario final.

Ventajas de las bases de datos

1. Independencia de datos y tratamiento: Si se realizan cambios en los datos no implica que se tenga que realizar cambios en los programas y viceversa. Menor costo en el mantenimiento.

   La independencia de datos implica un divorcio entre programas y datos; es decir, se pueden hacer cambios a la información que contiene la base de datos o tener acceso a la base de datos de diferente manera, sin hace cambios en las aplicaciones o en los programas.

2. Coherencia de resultados. Se logra con la reducción de redundancias al tener un buen diseño de la base de datos, unificando las acciones y evitando inconsistencias.

3. Mejora en la disponibilidad de datos. No hay dueños de los datos lo que no quiere decir que sean públicos, tampoco hay aplicaciones exclusivas para gestionar esos datos. Varios sistemas o usuarios pueden utilizar los mismos datos. . Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.

4. Cumplimiento de ciertas normas. Se establecen restricciones de seguridad para el acceso a la base de datos. También se fijan perfiles para los usuarios referidos en cuanto al acceso a los datos y las operaciones permitidas.

Desventajas

- Para pasar de un Sistema de archivo convencional a un Sistema de Base de Datos Significa: una fuerte inversión en, adquisición del SGBD, personal entrenado y equipos.

Resumen

Sistemas de Almacenamiento de Datos

Archivos Convencionales

Inconvenientes

- Falta de potencial para evolucionar
- Redundancia e inconsistencia de datos
- Dificultades de Acceso
- Problemas de Concurrencia
- Aislamiento de los datos
- Seguridad
- Integridad

Casos en que conviene

- Aplicaciones ya existentes
- Cuestiones de Performance

Bases de Datos

Objetivos

- Compartir información
  - Múltiples usuarios
  - Múltiples aplicaciones
- Mantenimiento de datos preciso y consistente
- Disponibilidad de los datos
- Flexibilidad para evolucionar según las necesidades
- Independencia del almacenamiento físico

Inconvenientes

- Almacenamiento centralizado (tradicionalmente)
- Dependencia burocrática del DBA (Administrador de Bases de Datos)
- Tiempo de actualización
- No se optimiza el acceso para una aplicación especial

Organización de Sistemas de Base de Datos

Una base de datos, a diferencia de un archivo, la comparten muchos usuarios. Y naturalmente cada usuario verá los datos de manera diferente. Nos referimos a la forma en que un usuario concibe y describe los datos desde una presentación o vista de usuario. Sin embargo, el problema es que usuarios diferentes tienen enfoques diferentes. Estas presentaciones se examinan en el modelo lógico global de la base de datos, que eventualmente deberá desarrollarse. Finalmente, el modelo lógico de la base de datos debe transformarse en el correspondiente diseño físico de la base de datos. El diseño físico considera la forma del almacenamiento de los datos y de sus interrelaciones, así como la mecánica del acceso.

Los sistemas de bases 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. En este capitulo presentamos una breve introducción a los principios de los sistemas de bases de datos.

Modelos de Bases Datos

Existen fundamentalmente tres alternativas disponibles para diseñar las bases de datos:

- ✓ Modelo Jerárquico
- ✓ Modelo en Red
- ✓ Modelo Relacional

a) El modelo jerárquico

La forma de esquematizar la información se realiza a través de representaciones jerárquicas o relaciones de padre/hijo, de manera similar a la estructura de un árbol. Así, el modelo jerárquico puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.

En el primer tipo se dice que existe una relación de uno a uno si el padre de la estructura de información tiene un solo hijo y viceversa, si el hijo tiene solamente un padre. En el segundo tipo se dice que la relación es de uno a muchos si el padre tiene más de un hijo, aunque cada hijo tenga un solo padre.

Inconveniente del modelo jerárquico

Supongamos la relación maestro-alumno, donde un maestro tiene varios alumnos, pero un alumno también tiene varios maestros, uno para cada clase. En este caso, si la información estuviera representada en forma jerárquica donde el padre es el maestro y el alumno es el hijo, la información del alumno tendrá que duplicarse para cada uno de los maestros.

Otra dificultad que presenta el modelo jerárquico de representación de datos es respecto a las bajas. En este caso, si se desea dar de baja a un padre, esto necesariamente implicará dar de baja a todos y cada uno de los hijos que dependen de este padre.

b) El modelo de red

El modelo de red evita esta redundancia en la información, a través de la incorporación de un tipo de registro denominado el conector, que en este caso pueden ser las calificaciones que obtuvieron los alumnos de cada profesor.

La dificultad surge al manejar las conexiones o ligas entre los registros y sus correspondientes registros conectores.

c) El modelo relacional

Se está empleando con más frecuencia en la práctica, debido el rápido entendimiento por parte de los usuarios que no tienen conocimientos profundos sobre Sistemas de Bases de Datos y a las ventajas que ofrece sobre los dos modelos anteriores.

Los sistemas relacionales son importantes porque ofrecen tipos de procesos de datos, como: simplicidad y generalidad, facilidad de uso para el usuario final, períodos cortos de aprendizaje y las consultas de información se especifican de forma sencilla.

En este modelo toda la información se representa a través de arreglos bidimensionales o tablas.

Las tablas son un medio de representar la información de una forma más compacta y es posible acceder a la información contenida en dos o más tablas.

Las bases de datos relacionales están constituidas por una o más tablas que contienen la información ordenada de una forma organizada. Cumplen las siguientes leyes básicas:

a. Generalmente, contendrá muchas tablas.
b. Una tabla sólo contiene un número fijo de campos.
c. El nombre de los campos de una tabla es distinto.
d. Cada registro es único.
e. El orden de los registros y de los campos no está determinado
f. Para cada campo existe un conjunto de valores posible.

Terminología Para Describir La Estructura Relacional

Para comprender la estructura relacional es importante definir los siguientes conceptos:

Tabla o Entidad:

Es un conjunto de datos homogéneo que contiene información sobre un tema específico (alumnos, profesores, clientes, proveedores, etc.)
Una tabla está compuesta por columnas y filas.

Fila o Registro:

Es la unidad lógica de información.
Cada registro contiene datos sobre un ítem específico. Por ejemplo, si tenemos la tabla de libros cada uno de los registros contendrá datos sobre un libro diferente.
Una tabla contiene tantos registros como elementos que componen la entidad y sólo un registro para cada elemento.
A su vez cada registro está compuesto por una determinada cantidad de campos.

Columna, Campo o Atributo:
Es la unidad elemental de información
Los campos son los distintos tipos de datos que componen un registro. Por ejemplo: nombre, apellido, domicilio.
Cada campo almacena sólo una clase de información determinada.
Al crear una tabla se deben definir las propiedades de los campos. Estas sirven para describir las características individuales y definir el comportamiento de los diferentes objetos en una base de datos.
Las propiedades básicas son: la longitud y el tipo de información que el campo puede almacenar.

La Longitud: consiste en determinar la cantidad de caracteres que almacenará cada campo. De esta manera cuando se trabaja con bases de datos de longitud fija, la sumatoria de la longitud de cada campo determina la longitud del registro.

El tipo de campo: sirve para definir qué clase de información se puede almacenar dentro de un campo. La existencia de diferentes tipos de datos permite una mayor eficiencia al almacenar la información y una mayor flexibilidad para trabajar con información proveniente de distintas fuentes.

Dentro de cada registro, cada campo guarda sólo una clase de información determinada: números, letras, fechas, etc.

Los Sistemas Gestores de Bases de Datos (SGBD) actuales permiten definir los siguientes tipos de datos:

Tipo	Sirve para
Texto	Texto o combinaciones de texto y números, como nombres o direcciones. También números que no requieran cálculos, como ser números de teléfono, códigos postales, códigos de artículos, etc. Por lo general hasta 255 caracteres.
Memo	Texto y números largos, como notas o descripciones adicionales, que no tengan un límite estricto. Hasta 65.535 caracteres.
Numérico	Datos numéricos para cálculos matemáticos.
Fecha/Hora	Fechas y horas. Permite computar tiempos transcurridos.
Moneda	Importes y valores de moneda.
Autonumérico	Números secuenciales exclusivos (en incremento de 1) o números aleatorios insertados automáticamente cuando se agrega un registro.
Lógicos	Campos que solo contendrán uno de dos valores como Verdadero/Falso, Si/No, Activado/Desactivado
Objeto OLE	Objetos (como documentos de procesadores de textos, planillas de hojas de cálculo, imágenes, sonidos u otros datos en formato binario) creados en otros programas mediante el protocolo OLE
Hipervínculos	Por ejemplo, una dirección de una página Web o una ruta de acceso a un archivo en el disco

Dominio:
Es una colección de valores, de los cuales uno o más campos obtienen sus valores reales. Por ejemplo, para sexo el dominio es Masculino / Femenino, cualquier otro dato fuera de este dominio debería ser rechazado como valor posible para el campo sexo.

Clave:
Atributo (campo), o conjunto de atributos, que identifica unívocamente a una unidad lógica de información. Existen distintos tipos de claves:

- Clave candidata: Conjunto no vacío de atributos que identifican univoca y mínimamente cada registro.
- Clave primaria: la que el usuario escoge de las claves candidatas. Es única para las tablas, es decir, una columna o combinación de columnas con la siguiente propiedad. Nunca existen dos filas de la tabla con el mismo valor en esa columna o combinación de columnas.
- Claves alternativas: claves candidatas que no han sido escogidas.
- Clave ajena (foránea): Conjunto de atributos de la tabla cuyos valores han de coincidir con los de la clave primaria de otra tabla. (Clave ajena y primaria debe estar definida sobre los mismos dominios).

Relación:
Asociación entre dos entidades (tablas) a través de una clave.

Diseño de una Base de Datos

El primer paso para crear una base de datos es planificar el tipo de información que se quiere almacenar en la misma, teniendo en cuenta dos aspectos: la información disponible y la información que necesitamos.

Es muy importante la fase de planificación y diseño de la base de datos, y por ende de las tablas que la componen, ya que un diseño pobre o incompleto siempre conduce a un sistema de información ineficiente.

La planificación de la estructura de la base de datos, en particular de las tablas, es vital para la gestión efectiva de la misma.

El diseño de la estructura de una tabla consiste en una descripción de cada uno de los campos que componen el registro y los valores o datos que contendrá cada uno de esos campos.

La definición de la estructura de las tablas se realiza de la siguiente manera según el siguiente esquema:

Nombre de la Tabla: Alumnos

#	Nombre de Campo	Tipo de dato	Tamaño	Decimales	Clave Principal
1	num_libreta	Numérico	6		Si
2	nombre	Alfanumérico	35		
3	domicilio	Alfanumérico	35		
4	teléfono	Alfanumérico	12		
5	fecha_nacim	Fecha	9		

Una vez definidos los atributos de los campos que se incluirán en la base de datos, debemos determinar en qué tablas se almacenarán los mismos, de manera de contar con una base de datos eficiente, con el principal objetivo de que no existan redundancias.

Se puede intentar realizar este proceso en forma directa o intuitiva pero se corre el alto riesgo de que las tablas o relaciones que se obtengan presenten algunos de los siguientes problemas:

- Incapacidad para representar ciertos hechos.
- Redundancia en la información, e incoherencias en la misma.
- Ambigüedades.
- Aparición en la base de datos de estados no validos en el mundo real. (anomalías en modificación, inserción, borrado).

Para evitar los mismos se cuenta con una herramienta de diseño llamada Normalización.

Normalización de Bases de Datos

La teoría de normalización consiste en obtener esquemas relacionales que cumplan unas determinadas condiciones y se centra en las determinadas Formas normales.

De hecho, normalizar es ajustar algo a una norma, y en su aspecto matemático, también reducir o simplificar.

La normalización de entidades o tablas da lugar a la descomposición de una información (en principio una sola entidad) en varias entidades, con el objeto principal de evitar redundancias.

Se dice que un esquema de relación está normalizado si satisface un conjunto determinado de restricciones o formas normales.

La relación que está en la forma normal más elevada posible es que mejor se adapta a nuestras necesidades debido a que optimiza las condiciones que son de importancia para nosotros:

- Se eliminan las redundancias;
- Se reduce la cantidad de espacio requerido para almacenar los datos;
- Se facilita la actualización de los datos de la relación o base de datos;
- Se eliminan inconsistencias de datos;
- Se simplifica la explicación de la base de datos

La normalización supone conseguir que cada atributo de una entidad dependa:

- de la clave 1FN
- de toda la clave 2FN
- de nada más que la clave 3FN

Normalizar = Reducir, Simplificar	
Primera Forma Normal	Evitar redundancias: cada atributo en una sola entidad. Se obtiene separando en una nueva tabla o entidad los atributos que tienen múltiples valores para una determinada clave.

Segunda Forma Normal	Evitar dependencia incompleta de atributos con la clave: crear relación entre entidades Se obtiene separando en una nueva tabla los atributos con dependencia incompleta, es decir, que no dependen de toda la clave.
Tercera Forma Normal	Evitar dependencia de atributos con atributos no claves: eliminar dependencias transitivas. Se obtiene eliminando dependencias transitivas, es decir, separando en una nueva tabla los atributos que dependen de otro atributo no clave.

Para que una base de datos se encuentre en tercera forma normal, debe primero estar en segunda y primera forma normal.

Ejemplo de Normalización: Compra de títulos en Bolsa de Valores

Un cliente ordena una compra en Bolsa, compuesta por cuatro diferentes valores mobiliarios.

Suponemos que toda la información necesaria es la siguiente

Tabla sin normalizar

Comprobante	Fecha	Código Cliente	Nombre del Cliente	Código del Valor	Nombre del Valor	Cantidad	Precio
124	31/12/99	19	Juan Pérez	257	Telecom	2500	5
124	31/12/99	19	Juan Pérez	258	Aluar	6000	3
124	31/12/99	19	Juan Pérez	259	Tenaris	1000	15
124	31/12/99	19	Juan Pérez	260	Banco Galicia	500	1
125	31/12/99	21	Luis Terroba	258	Aluar	200	3
125	31/12/99	21	Luis Terroba	259	Tenaris	150	15
125	31/12/99	21	Luis Terroba	260	Banco Galicia	1000	3

Primera forma normal

En una tabla no puede haber grupos repetitivos. Aislamos los datos (campos) que se repiten en cada registro

A la nueva tabla aislada la llamaremos COMPRA, y a la que queda VALOR/COMPRA.

COMPRA

Comprobante	Fecha	Código Cliente	Nombre del Cliente
124	31/12/99	19	Juan Pérez
125	31/12/99	21	Luis Terroba

VALOR/COMPRA

Comprobante	Código del Valor	Nombre del Valor	Cantidad	Precio
124	257	Telecom	2500	5
124	258	Aluar	6000	3
124	259	Tenaris	1000	15
124	260	Banco Galicia	500	1
125	258	Aluar	200	3
125	259	Tenaris	150	15
125	260	Banco Galicia	1000	3

En esta tabla restante, sólo se repite el Comprobante, imprescindible para relacionar las tablas.

La clave de la tabla COMPRA será el Comprobante, puesto que identifica plenamente la compra (cada Comprobante tendrá una sola ocurrencia).
La clave de la tabla VALOR/COMPRA será el Comprobante y el Código del Valor.

Segunda forma normal

Una tabla está en 2FN si además de estar en 1FN todos los atributos que no forman parte de ninguna clave candidata suministran información acerca de la clave completa.

El nombre del valor se repetirá en todas las compras en que se desee comprar el mismo Código del Valor, por lo que dependerá de éste.
Pero sólo depende del Código del Valor, que es parte de la clave y no de toda la clave, por lo que la pasamos a una nueva tabla, a la que llamaremos VALOR.

COMPRA

Comprobante	Fecha	Código Cliente	Nombre del Cliente
124	31/12/99	19	Juan Pérez
125	31/12/99	21	Luis Terroba

VALOR/COMPRA

Comprobante	Código del Valor	Cantidad	Precio
124	257	2500	5
124	258	6000	3
124	259	1000	15
124	260	500	1
125	258	200	3
125	259	150	15
125	260	1000	3

VALOR

Código del Valor	Nombre del Valor
257	Telecom
258	Aluar
259	Tenaris
260	Banco Galicia
258	Aluar
259	Tenaris
260	Banco Galicia

Tercera forma normal

Si además de estar en 2FN, los atributos que no forman parte de ninguna clave candidata facilitan información sólo acerca de las claves y no acerca de otros atributos.

En la tabla COMPRA, el nombre del Cliente se repetirá siempre que haga compras al mismo cliente. Depende del Código Cliente, atributo que no es clave en esta tabla.

Eliminamos esta dependencia transitiva creando la tabla CLIENTE.

COMPRA

Comprobante	Fecha	Código Cliente
124	31/12/99	19
125	31/12/99	21

CLIENTE

Código Cliente	Nombre del Cliente
19	Juan Pérez
21	Luis Terroba

VALOR/COMPRA

Comprobante	Código del Valor	Cantidad	Precio
124	257	2500	5
124	258	6000	3
124	259	1000	15
124	260	500	1
125	258	200	3
125	259	150	15
125	260	1000	3

VALOR

Código del Valor	Nombre del Valor
257	Telecom
258	Aluar
259	Tenaris
260	Banco Galicia
258	Aluar
259	Tenaris
260	Banco Galicia

De esta manera, partiendo de una tabla sin normalizar obtenemos una base de datos normalizada, compuesta en este caso por cuatro tablas.

Ordenamiento de Registros en una Tabla

Al cargar los datos en una base de datos no es relevante el orden con que los mismos son ingresados. De hecho, en la vida real los hechos nos se presentan en un orden determinado. Solamente ocurren.

No obstante, ello para analizar información es necesario que la misma se encuentre ordenada con algún criterio (por código, alfabéticamente, por zonas, etc.)

Cuando se trabaja con bases de datos existen dos formas de ordenar los mismos: física o lógicamente.

El ordenamiento físico (SORT) consiste en crear partiendo de una tabla desordenada una nueva tabla ordenada en función de algún campo. De esta manera se cuenta con dos tablas: la original y la nueva ordenada.

Esta forma de ordenamiento tiene como principal ventaja el hecho que los registros se encuentran grabados en la secuencia del orden establecido, por lo que toda consulta que se requiera parte de la base ordenada. La principal desventaja es que ante una nueva alta, o modificación de datos, el orden se pierde, siendo necesario reordenar la tabla en cuestión.

El ordenamiento lógico o indexado (INDEX) consiste en definir uno o más campos como índices de la tabla, generando el Sistema Gestor de Bases de Datos un archivo índice sin generar una tabla nueva. Los datos continúan "desordenados" en la tabla en el orden en que fueron ingresados, pero en las consultas a través del índice se los visualiza ordenados.

La ventaja de este tipo de ordenamiento reside en que al estar activado el índice toda alta o modificación que se realiza genera una actualización en el índice establecido, siendo el ordenamiento permanente. La desventaja es que las consultas son más lentas que en el esquema de ordenamiento físico ya se debe ejecutar en cada consulta un ordenamiento previo en función de los índices utilizados.

Sistemas Gestores de Bases de Datos

Definición de SGBD o DBMS (Data Base Management Sistem): Conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministran, tanto a los usuarios como a los analistas, programadores o administrador los medios necesarios para describir, recuperar y

manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad.

Clasificación de usuarios:

- Usuarios, los que utilizan la información, solamente pueden leer o modificar, pero desde aplicaciones desarrolladas para tal fin.

- Administradores de Bases de Datos, personas que gestionan la BD. Se encargan de la actualización, gestión de permisos, volcado histórico, copias de seguridad (Backup), etc. Un DBA es una persona con conocimientos profundos acerca del software y de gestión bases de datos, es el responsable de los datos que están almacenados en la base de datos. Entre sus funciones está la de mantener la seguridad en integridad, para ello debe definir los usuarios y sus permisos, encargarse de las copias de seguridad, etc. En general debe mantener la base de datos funcionando en todo momento.

- Programadores, se encargan del desarrollo de aplicaciones para los usuarios.

Un sistema de gestión de base de datos (DBMS) 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.

La esencia de una base de datos es el Sistema Administrador de la Base de Datos (DBMS: Database Management System), el cual permite la creación, modificación y actualización de la base de datos, la recuperación de los datos y la emisión de reportes. A la persona responsable de asegurar que la base de datos satisfaga los objetivos programados se le denomina administrador de la base de datos.

El Sistema Gestor de Bases de Datos administra el contenido de las Bases de Datos a través de tres operaciones básicas:

- Seleccionar filas de alguna tabla (SELECT)
- Seleccionar columnas de alguna tabla (PROJECT)
- Unir o juntar información de varias tablas (JOIN)

En la actualidad existen SGBD para utilizar en una PC personal para uso diario por parte de cualquier usuario sin conocimientos avanzados de Gestión de Bases de Datos hasta sistemas de Gestión de Grandes Bases de Datos a instalarse en Mainframes o Minicomputadoras, llamados también Motores de Bases de Datos.

Un ejemplo de un SGBD para PC es el producto de Microsoft llamado ACCESS que se presenta más adelante, otro con un poco más de complejidad es Visual FoxPro 6 también de Microsoft y para PC.

Para servidores de bases de datos, es decir un servidor dedicado a soportar una base de datos puede ser ORACLE, INFORMIX, SYBASE, etc. Estos sistemas permiten gestionar bases de datos de gran volumen, hablamos de Gb o Tb de información y están instalados en servidores, llamados servidores de bases de datos.

Estos sistemas son muy complejos que necesitan de uno o más DBA (Data Base Administrator o Administrador de Bases de Datos) para su administración.

MICROSOFT ACCESS

Como ya se dijo, es un producto de Microsoft, permite gestionar una base de datos de unos cuantos Mb de información. Está pensado para usuarios finales, es decir usuarios sin muchos conocimientos de programación. No tiene niveles de seguridad.

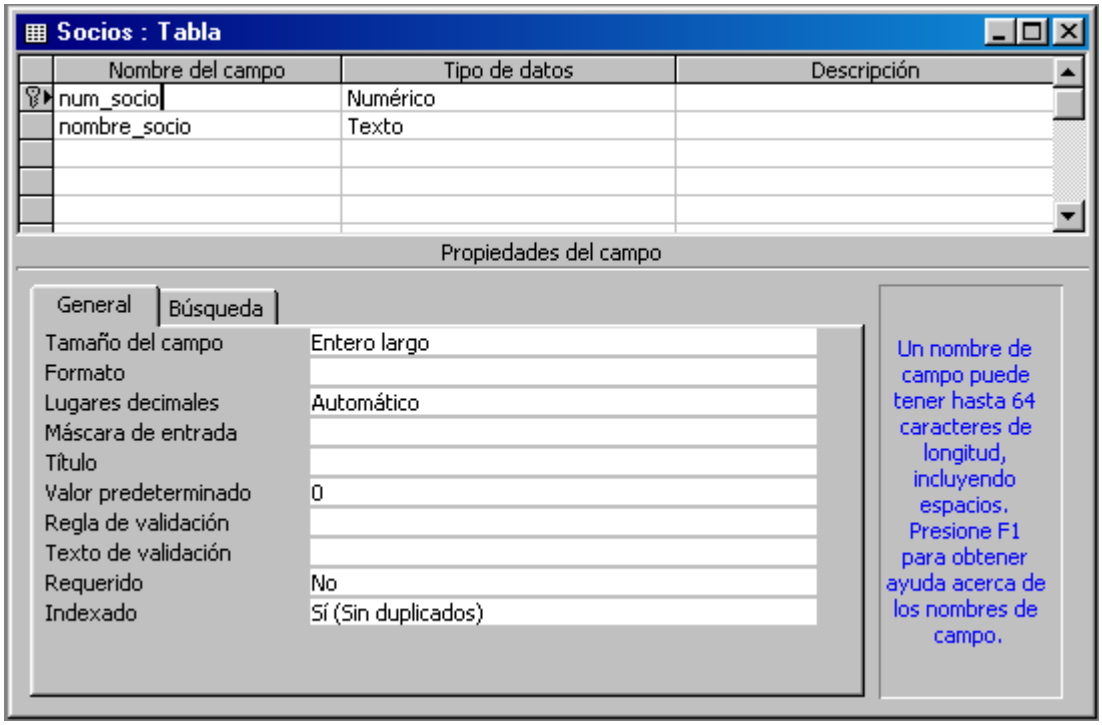
Los elementos u objetos que contiene son Tablas, Consultas, Formularios, Informes, Páginas, Macros y Módulos, de los cuales son que nos interesan son los cuatro primeros.

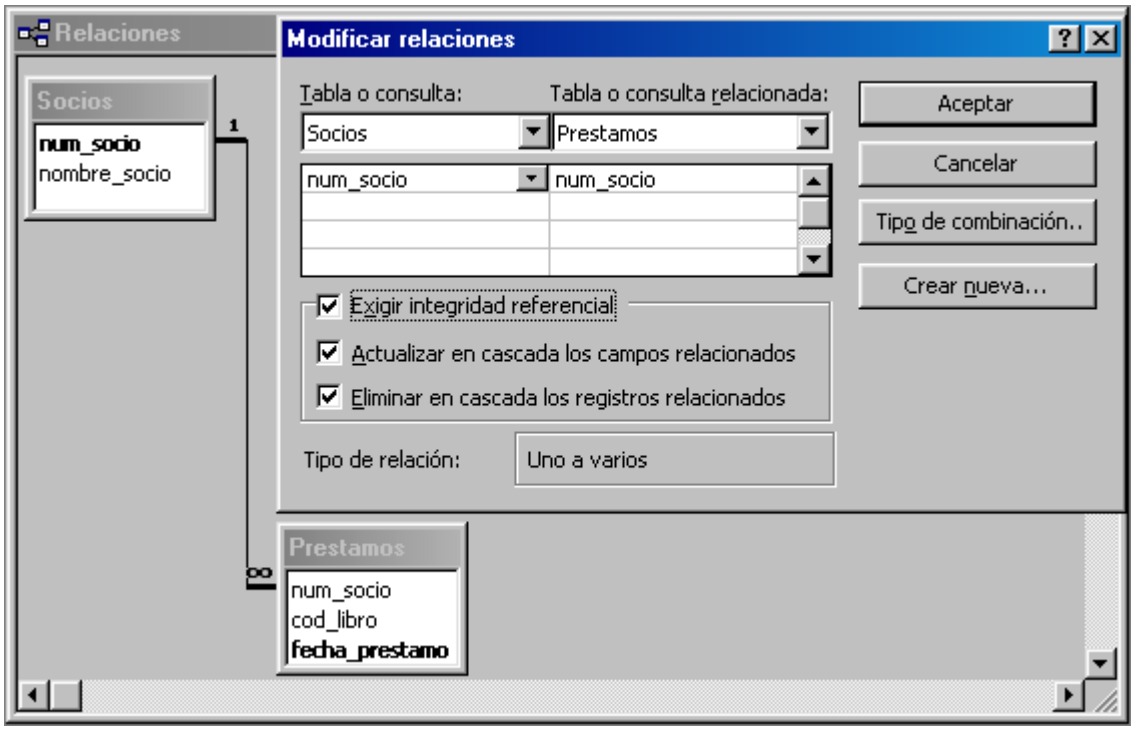
Tabla: Una tabla es una colección de datos sobre un tema específico, como libros o socios. La utilización de una tabla diferente para cada tema significa que se almacenan los datos sólo una vez, lo cual hace aumentar la eficacia de la base de datos, y reduce errores de entrada de datos.

Las tablas organizan datos en columnas (denominadas campos) y filas (denominadas registros).

Un campo común relaciona dos tablas de manera que Microsoft Access puede combinar los datos de las dos tablas para ver, modificar o imprimir.

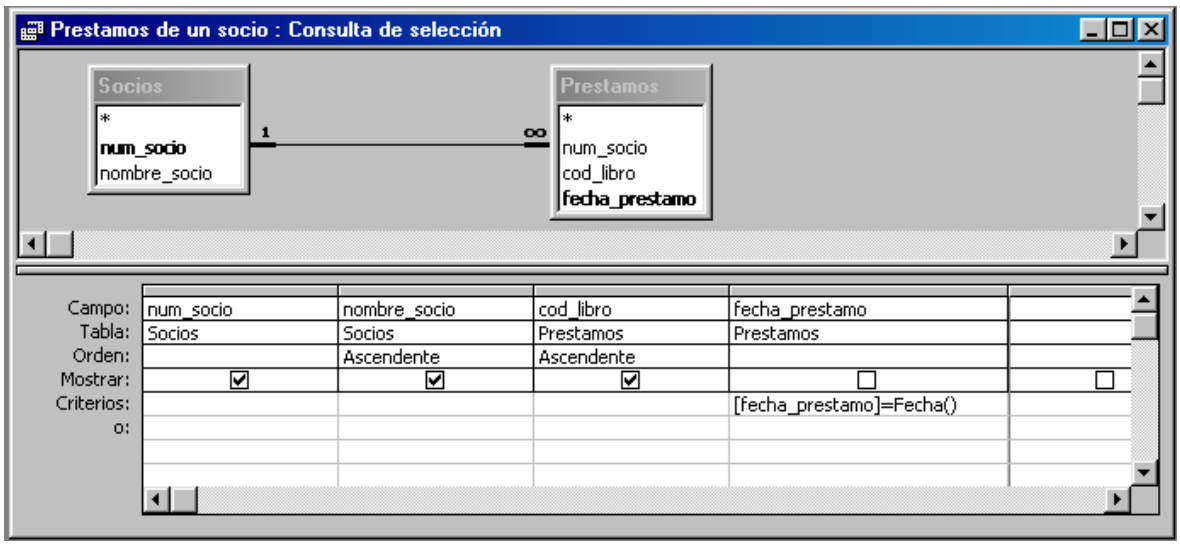
En la vista Diseño de la tabla, puede crear una tabla entera desde el principio, o agregar, eliminar o personalizar los campos en una tabla existente.





Consulta: Se utilizan consultas para ver, modificar y analizar datos de formas diferentes. También pueden utilizarse como el origen de registros para formularios, informes y páginas de acceso a datos.

El tipo de consulta más habitual es una consulta de selección. Una consulta de selección recupera datos de una o más tablas utilizando los criterios que especifique y después los muestra en el orden que desee.



Las consultas realizan operaciones de selección cuando se aplica criterios como en el ejemplo que selecciona la fecha de hoy, de proyección como es el caso que se tilda en Mostrar cuales campos se van a utilizar, en este caso también se dice proyección cuando se muestran campos calculados en función de los existentes.

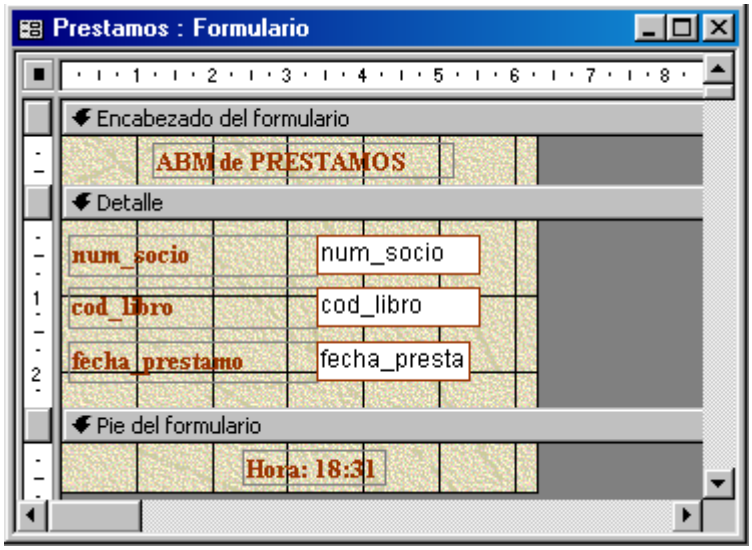
Cuando en una consulta se utilizan más de una tabla se está realizando la operación de relación, puesto que utiliza la relación definida para hacer la correspondencia entre los datos de una tabla y la otra.

Existe una operación parecida llamada UNION que realiza la combinación de todos los datos de una tabla con todos los de la otra, pero es muy poco utilizada en bases de datos bien definidas.

Formularios: Pueden utilizarse formularios con varios propósitos. El principal es la actualización de la información en la base de datos.

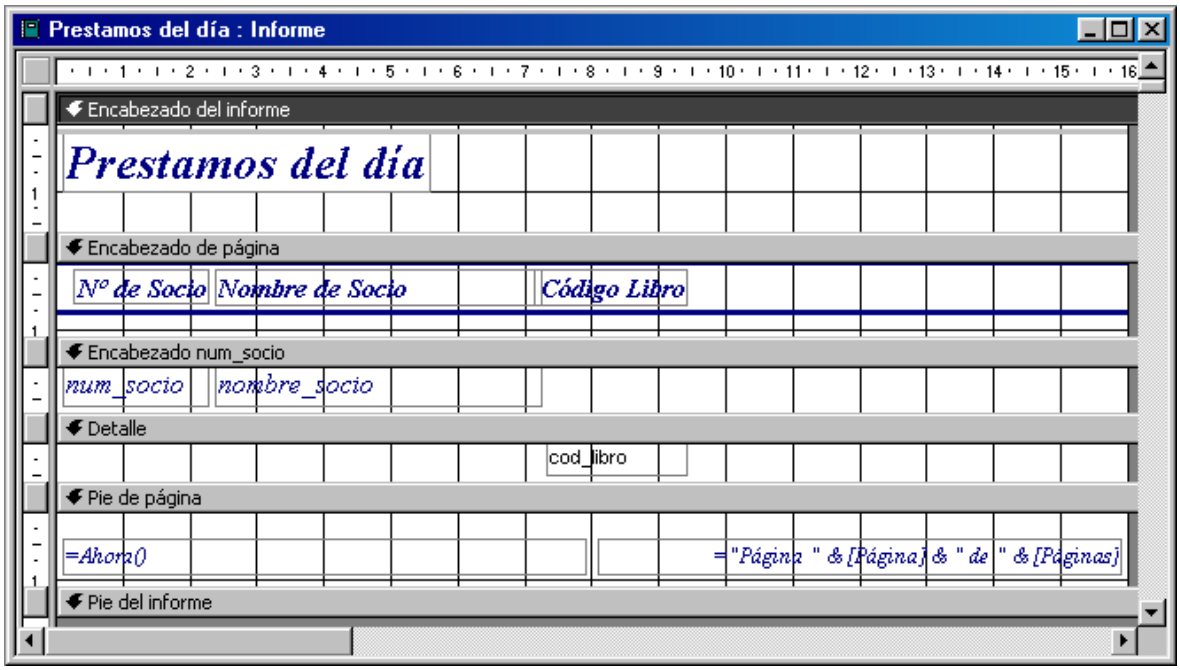
La mayor parte de la información de un formulario procede de un origen de registros base. Parte de la información del formulario se almacena en el diseño del mismo.

El vínculo entre un formulario y su origen de registros se crea por medio de objetos gráficos denominados controles. El tipo de control más comúnmente utilizado para mostrar y escribir datos es un cuadro de texto.



Informes: Un informe es un método eficaz de presentar los datos en formato impreso. Dado que tiene el control sobre el tamaño y el aspecto de todo el informe, puede mostrar la información en la manera que desee verla.

La mayor parte de la información de un informe procede de una tabla, consulta o instrucción SQL base, que es el origen de los datos del informe. Parte de la información del informe se almacena en el diseño del mismo.



Tipos de Datos

Los tipos de datos que se pueden utilizar son: Texto, Memo, Numérico, Fecha/Hora, Moneda, Autonumérico, Si/No, Objeto OLE y Asistente para búsqueda.

Propiedades de los campos

Las propiedades de los campos que se pueden definir son el Tamaño del campo, Formato, Máscara de entrada, Título, Valor predeterminado, Regla de validación, Requerido, etc.