

Unidad 1. Introducción a las bases de datos

Desarrollo de Software Semestre 2

Programa de la asignatura:
Bases de datos

Unidad 1. Introducción a las bases de datos

Universidad Abierta y a Distancia de México



Unidad 1. Introducción a las bases de datos

Índice

Presentación de la unidad	3
Propósitos de la unidad	3
Competencia específica	3
1.1. Conceptos generales.....	3
1.1.1. Conceptos informáticos básicos	5
1.1.2. Conceptos específicos de Bases de datos	11
1.2. Tipos de Bases de datos	14
1.2.1. Base de datos de árbol (jerárquica)	17
1.2.2. Base de datos de red.....	21
1.2.3. Base de datos relacional.....	23
1.2.4. Sistema Gestor de Base de Datos (SGBD).....	25
1.2.5. Estructura de un Sistema Gestor de Base de Datos	26
Cierre de la unidad.....	26
Fuentes de consulta.....	27

Unidad 1. Introducción a las bases de datos

Presentación de la unidad

En esta primera unidad se establecen los principios fundamentales de Bases de datos y gestores, con la finalidad de que exista una familiaridad con los diversos conocimientos que la asignatura requiere, como lo son términos informáticos básicos y específicos de bases de datos. Repasar y retroalimentar conceptos informáticos básicos servirá para recordar la importancia que tienen para ser aplicados a esta asignatura, se abordará desde el concepto de informática hasta los Sistemas Gestores de Bases de Datos (SGBD), pasando por lo que son las unidades de almacenamiento, los sistemas numéricos, así como los códigos, entre otros conceptos, lo que permitirá dar cumplimiento al propósito de la unidad.

Propósitos de la unidad

Iniciar el proceso para la elaboración del prototipo de base de datos documental, tomando en cuenta el concepto de bases de datos, los diversos tipos que existen de ellas, así como los diferentes Sistemas Gestores de Bases de Datos.

Competencia específica

Diferenciar los conceptos específicos de Bases de datos para comprender los tipos de éstas, así como sus Sistemas Gestores, por medio de las características que las describen.

1.1. Conceptos generales

En la actualidad la tecnología ha crecido a pasos agigantados, por tal razón hablar de información tiene que ver con el ajuste de esta misma en su totalidad, pero de manera electrónica, dejando a un lado el trabajo rudimentario, por eso el impacto y puntualidad de este tema es parte de la columna vertebral del conocimiento general de informática, lo que permitirá distinguir, identificar y aplicar los conceptos adquiridos en cualquiera que sea el escenario que el estudiante enfrente, tanto en su formación académica y laboral, como en el ámbito personal y social, aplicando las habilidades adquiridas sobre los tipos de bases de datos y gestores.

Teniendo en cuenta estos antecedentes, y antes de continuar con la asignatura, es preciso comentar algunas cuestiones de suma importancia referidas a la información y a la informática, ya que éstas son la base del presente curso.



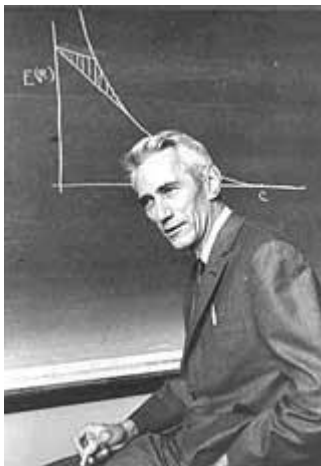
Unidad 1. Introducción a las bases de datos

La palabra INFORMÁTICA se integra por la contracción de los vocablos Información y Automática.

Habitualmente la palabra **información** está ligada a la “adquisición de conocimientos”; se dice haber recibido información cuando se conoce algo que se ignoraba. Relacionado con este término se encuentra el concepto de informática, la cual trata sobre la adquisición, representación, tratamiento y transmisión de la información, es decir, la informática es la ciencia que estudia el tratamiento racional de la información por medio de máquinas automáticas. La Real Academia Española define a la informática como: “conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores”. (RAE, 2017)

Con base en lo anterior, se puede señalar que los primeros ordenadores surgieron ante la necesidad de realizar grandes cálculos en aplicaciones militares, que escapaban a las posibilidades humanas. Actualmente, las ventajas que representan los ordenadores son: rapidez, fiabilidad, gran capacidad de almacenamiento y un precio asequible. Ante esto, la informática, o ciencia de los ordenadores, aparece como un intento de estructurar todos los avances tecnológicos y de programación que se producen vertiginosamente.

Los primeros estudios de la información partieron de la aplicación la teoría de las probabilidades, iniciado por Claude Elwood Shannon, a mediados de los años cuarenta.



Claude Elwood Shannon (1916-2001). Ingeniero electrotécnico y matemático estadounidense. Nació en Michigan y cursó estudios en la Universidad de Michigan. En 1940 obtuvo su doctorado en el Instituto de Tecnología de Massachusetts y en 1941 ingresó en los laboratorios Bell, donde trabajó en el desarrollo de sistemas de encriptación digital. Siete años más tarde estableció las leyes fundamentales de la transmisión y compresión de datos en su trabajo *A mathematical theory of communication* (en este artículo surge por vez primera el término “bit”, contracción de binary unit, para designar la cantidad de información de una fuente que emite mensajes binarios), artículo en el que exponía su concepto inicial de una teoría de unificación de la transmisión y tratamiento de la información, donde explica cómo la cantidad de información de una fuente de mensajes

se puede medir empleando el concepto de entropía y demuestra que esta magnitud es un límite fundamental en la representación de dicha información. Falleció en Massachusetts, a consecuencia del Alzheimer. La biografía de Shannon está colmada de frutos de su ingenio, con numerosas aplicaciones en el campo de las máquinas automáticas, desde un ratón electrónico hasta un WC automático, pasando por diversos juego electrónicos de ajedrez, calculadoras, instrumentos musicales, juguetes mecánicos, relojes.

El estudio de la informática tiene básicamente dos ramas: una teórica, con la que adquiere un carácter científico y otro relacionado con el diseño y la experimentación de aplicación, con lo que adquiere su lugar como ingeniería.

Unidad 1. Introducción a las bases de datos

De acuerdo a Álvarez y Bravo (2000) los siguientes son algunos de los campos de aplicación más destacados de la informática:

- Algoritmos y estructuras de datos.
- Análisis y desarrollo de sistemas informáticos.
- Arquitectura de ordenadores.
- Inteligencia artificial.
- Informática educativa.
- Informática gráfica
- Informática industrial
- Lenguajes de programación
- Metodología de la programación
- Ingeniería del software
- Robótica

1.1.1. Conceptos informáticos básicos

Una vez habiendo abordado lo que es la informática y su importancia en la actualidad, debido a que se ha convertido en el sustento de todas las empresas y de todo lugar donde existe tecnología, desde un aparato doméstico hasta grandes servidores que permiten la comunicación estrecha con el mundo actual, en este apartado se revisarán conceptos relacionados con ella.

De la informática emanan otros términos, como son: computadora, bit, byte, gigabyte, megabyte, terabyte, archivos, directorios, campos, registros, dispositivos, hardware, software, bases de datos, Internet, servidor, red, correo electrónico, wiki, foros, blogs, virus, antivirus, aplicación, lenguaje de programación, programa; entre muchos otros. A continuación trataremos algunos de ellos.

El término **computadora**, desde el punto de vista de la actualidad, empapada de tecnología, se define como un dispositivo electrónico capaz de tratar gran cantidad de información, con tareas generales como: almacenar, organizar, gestionar, actualizar, entre otras. Los nombres que con mayor frecuencia se utilizan para referirse a este dispositivo son: computadora, terminal, servidor, terminal tonta, ordenador, nodo y PC.

Ahora bien, la unidad mínima que soporta un computador es un **bit**, el cual es la representación en sistema binario de un 1 o un 0, que en términos de electrónica 1 simboliza presencia de alto voltaje y 0 baja presencia de voltaje.

Mientras que un **byte** es la unidad mínima *de almacenamiento* del ordenador y representa 8 bits, que en conjunto conforman un carácter cualquiera de los 256 que conforman el código ASCII, que más adelante se mencionará a detalle. Es importante tener clara la diferencia entre bit y byte, ya que son términos esenciales en cuanto a informática se refiere.

Unidad 1. Introducción a las bases de datos

Las computadoras trabajan por medio de sistemas numéricos: binario, decimal, octal y hexadecimal.

Binario	Decimal	Octal	Hexadecima
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	10	8
1001	9	11	9
1010	10	12	A
1011	11	13	B
1100	12	14	C
1101	13	15	D
1110	14	16	E
1111	15	17	F

Sistemas numéricos

Por su sencillez el sistema numérico se utiliza para manejo de datos es el sistema Binario, en él se emplea la notación posicional a partir de sólo dos dígitos (0,1). Su lógica parte de que el cero representa un espacio apagado, vacío o con cero volts, mientras que el uno representa un espacio encendido, energizado o con +5 o +12 volts. Si se invierte esa relación se genera lo que se conoce como lógica negativa.

Los espacio ocupados por los dígitos se llaman bit, de modo que 1010 contiene cuatro bits, 100 contiene

3, 10 contiene dos, y así en lo subsecuente con cualquier unidad que se quiera representar.

Como se dijo anteriormente el sistema Binario parte de la notación posicional, de modo que el valor de cada dígito depende directamente de la posición que tiene en la serie completa. En este caso la base numérica es dos por lo que tenemos que existen dos dígitos (0 ó 1) en cada posición del número. Para obtener el número

decimal de uno en binario, comenzando por el lado derecho del número en binario, se debe multiplicar cada cifra por dos elevado a la potencia consecutiva, iniciando por la potencia 0 (2^0). Una vez realizadas cada una de las multiplicaciones, se suman los resultados de ellas, lo que reflejará el número equivalente al sistema decimal. Por ejemplo, el número 1101_2 (en base 2) es:

$$1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 8 + 4 + 0 + 1 = 13_{10}$$

Unidad 1. Introducción a las bases de datos

El sistema numérico Decimal posee diez dígitos diferentes y es el que actualmente se utiliza más comúnmente. El sistema numérico decimal es un sistema de numeración posicional, por lo que el valor del dígito depende de su posición dentro del número, es decir:

$$4823.29 = 4 \cdot 1000 + 8 \cdot 100 + 2 \cdot 10 + 3 \cdot 1 + 2 \cdot 1 + 9 \cdot 0.1 = 4 \cdot 10^3 + 8 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 + 2 \cdot 10^{-1} + 9 \cdot 10^{-2}$$

El número decimal A se debe multiplicar por 10^n , en donde n es positivo o negativo de acuerdo a la posición de los dígitos respecto del punto. Los dígitos a la izquierda del punto toman el valor correspondiente a las potencias positivas de la base (10, en este sistema), en función de la posición que ocupan en el número, y representan al dígito de las unidades ($10^0=1$), decenas ($10^1=10$), centenas ($10^2=100$), etcétera. Los dígitos a la derecha del punto representan al dígito de las décimas ($10^{-1}=0.1$), centésimas ($10^{-2}=0.01$), milésimas ($10^{-3}=0.001$), etcétera. Los resultados de estas multiplicaciones serán mayores o iguales que uno si el dígito se localiza a la izquierda del punto decimal, en cambio, el valor será menor que uno si el dígito se localiza a la derecha del punto decimal.

Por ejemplo, el número 5432.789 expresado en la notación posicional es:

Primero $2 \cdot 10^0 = 2$	Primero $7 \cdot 10^{-1} = 0.7$
Segundo $3 \cdot 10^1 = 30$	Segundo $8 \cdot 10^{-2} = 0.08$
Tercero $4 \cdot 10^2 = 400$	Tercero $9 \cdot 10^{-3} = 0.009$
Cuarto $5 \cdot 10^3 = 5000$	

Con el fin de aclarar el procedimiento utilizado en el ejemplo anterior, es necesario revisar lo siguiente:

$10^{-6} = 0.000001$
$10^{-5} = 0.00001$
$10^{-4} = 0.0001$
$10^{-3} = 0.001$
$10^{-2} = 0.01$
$10^{-1} = 0.1$
$10^0 = 1$
$10^1 = 10$
$10^2 = 100$
$10^3 = 1000$
$10^4 = 10000$
$10^5 = 100000$
$10^6 = 1000000$

Notación posicional del sistema decimal

A continuación se verán dos maneras para convertir números decimales en binarios.

Unidad 1. Introducción a las bases de datos

1. Al dividir los decimales entre dos, cuyo resultado entero se vuelve a dividir entre dos, y así continuamente, ordenando los restos del último al primero. Por ejemplo, 74 en decimal es 1001010 en binario, porque:

74/2 =	37	Sobrando:	0
37/2 =	18	Sobrando:	1
18/2 =	9	Sobrando:	0
9/2 =	4	Sobrando:	1
4/2 =	2	Sobrando:	0
2/2 =	1	Sobrando:	0
1/2 =	0	Sobrando:	1

2. Al realizar divisiones sucesivas entre dos y colocando un cero o un uno de lado derecho dependiendo si el número es par o impar. Si es impar, se le resta uno y se continúa con las divisiones hasta llegar a uno. Posteriormente, se coloca del lado derecho un 1 si es impar y un cero si es par. Finalmente se ordenan los dígitos de abajo hacia arriba, por ejemplo, 131 en decimal es 10000011 en binario, porque:

131	1
65	1
32	0
16	0
8	0
4	0
2	0
1	1

Otro sistema numérico utilizado en informática es el sistema octal, que tiene la particularidad de utilizar una base (8) que es la potencia exacta de la numeración binaria, por lo que su conversión para utilizar ambos sistemas es más o menos simple.

Se parte de 8 dígitos básicos (0,1,2,3,4,5,6,7) que poseen un valor homólogo al que tienen en el sistema decimal, además del uso de la notación posicional. Por ejemplo el número 3452.32_q (q representa los decimales de las operaciones octales) tenemos:

$$\begin{aligned} 3 \cdot 8^3 + 4 \cdot 8^2 + 5 \cdot 8^1 + 2 \cdot 8^0 + 3 \cdot 8^{-1} + 2 \cdot 8^{-2} &= 3 \cdot 512 + 4 \cdot 64 + 5 \cdot 8 + 2 \cdot 1 + 3 \cdot 0.125 + \\ 2 \cdot 0.015625 &= 1536 + 256 + \\ 40 + 2 + \\ 0.375 + 0.03125 &= 1834 + .40625_d, \text{ entonces } 3452.32_q = 1834.40625_d \end{aligned}$$

Recuerda que un exponente negativo indica cuántas veces dividir por ese número el uno; por ejemplo:

Unidad 1. Introducción a las bases de datos

$8^{-3} = 1 \div 8 \div 8 \div 8 = 0.001953125$, o bien: $8^{-3} = 1 \div (8 \cdot 8 \cdot 8) = 1 \div 512 = 0.001953125$.

Los números octales pueden construirse a partir de números binarios, agrupando cada tres dígitos consecutivos de estos últimos (de derecha a izquierda) y obteniendo su valor decimal. Por ejemplo, el número 74 (en decimal) es 1001010 (en binario), si éste lo agrupamos como 1 001 010, obtenemos que 74 es 112 en octal porque:

Binario	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Es conveniente mencionar que la cantidad correspondiente en octal se concentra de izquierda a derecha.

Un sistema muy utilizado en la informática actual es el Sistema Hexadecimal, de igual manera es gracias a su facilidad de conversión al formato binario. En este caso se utiliza como base el 16, y al igual que los sistemas vistos con anterioridad cada dígito altera su valor dependiendo de su lugar en la cadena de dígitos por ejemplo, el número 1234 es igual a:

$$1 \cdot 16^3 + 2 \cdot 16^2 + 3 \cdot 16^1 + 4 \cdot 16^0 = 4096 + 512 + 48 + 4 = 4660$$

Debido a que el sistema usual de numeración es de base decimal, contando con sólo diez dígitos, se utilizan las primeras seis letras del alfabeto latino para suplir los dígitos faltantes, por lo que el sistema numérico hexadecimal está conformado por los dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F, donde A=10, B=11, C=12, D=13, E=14 y F=15.

En la conversión de un número hexadecimal a binario se logra sustituyendo los correspondientes cuatro bits para cada dígito que componen la serie. Por ejemplo A B 0 C D h = 1010 1011 0000 1100 1101 b:

Unidad 1. Introducción a las bases de datos

Binario	Hexadecima
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Una vez que se han abordado los sistemas numéricos con los que trabajan las computadoras, es necesario hablar de los códigos EBCDIC (Extended Binary Coded Decimal Interchange Code / Código Ampliado de Intercambio Decimal Codificado en Binario) y ASCII (American Standard Code for Information Interchange / Código Estadounidense Estándar para el Intercambio de Información), como los más importantes y conocidos para representar datos.

El primero es un código para texto y comunicaciones basado en el sistema binario que se creó para comunicaciones y control de impresoras por IBM. Es un código de 8 bits y almacena un carácter alfanumérico o dos dígitos en un byte.

Por otra parte, ASCII es un código que se basa en un conjunto de caracteres del alfabeto latino utilizado en el idioma inglés y otras lenguas del mundo occidental. Con el objetivo de controlar dispositivos digitales que manipulan texto o para representar textos en pantalla, la mayoría de los sistemas informáticos del día de hoy utilizan el código ASCII, el cual fue desarrollado como una evolución de los códigos telegráficos en 1963, por el entonces llamado ASA (American Standards Association / Asociación Estadounidense de Estándares), el cual pasó a llamarse ANSI (American National Standards Institute / Instituto Estadounidense de Estándares Nacionales) en 1969 (Master Magazine, 2005).

Por medio de estos códigos y sistemas numéricos se crean lenguajes de programación, que son sistemas de comunicación entre la instrucción del usuario y el sistema de cómputo. Estos se distinguen en tres niveles:

Bajo Nivel: Es totalmente dependiente de la máquina, lo que significa que sólo funcionará en la máquina que fue desarrollado y no se puede migrar ya que está diseñado prácticamente a la

Unidad 1. Introducción a las bases de datos

medida del hardware, pero tiene como ventaja que aprovecha al máximo las características del mismo. Un ejemplo de este es el lenguaje máquina, que es el encargado de instruir a la computadora sus operaciones básicas de funcionamiento. También tiene la ventaja de ser significativamente rápido aunque es muy complejo su manejo, ya que se desarrollan códigos fuente enormes en los que encontrar un fallo es casi imposible.

Alto Nivel: Se caracterizan por ser más cercanos al lenguaje natural que al de la máquina, además de que utilizan Estructuras Dinámicas de Datos, que son estructuras que pueden cambiar de tamaño durante la ejecución y por tanto pueden crear estructuras que se adaptan fácilmente a las necesidades reales de un programa. A diferencia de los de bajo nivel son completamente independientes a la arquitectura del ordenador, por lo que son completamente migrables. Por lo tanto el programador sólo requiere de un traductor capaz de entender el código fuente y las características particulares de la máquina.

Medio Nivel: Estos son lenguajes de punto medio, como por ejemplo C. Estos pueden acceder a los registros del sistema y trabajar con direcciones de memoria, lo que es una capacidad del lenguaje de bajo nivel, y al mismo tiempo pueden realizar operaciones en alto nivel.

1.1.2. Conceptos específicos de Bases de datos

Haber revisado conceptos básicos de informática en el apartado anterior nos da las bases suficientes para entrar de lleno en los conceptos propios de las bases de datos, como son: dato, campo, registro, archivo, entre otros. La importancia de abordar este tipo de conceptos radica en la necesidad de unificar términos que se irán enlazando y utilizando conforme se avance en los temas, por lo que a continuación se definirán algunos de ellos.

Se comenzará definiendo el término **dato**, el cual es cualquier carácter, sin embargo debe conformar información coherente y ordenada. Cuando tenemos un conjunto de datos es un **campo**, y cuando a este campo le es anexada más información u otros campos, se convierten en un **registro**, por lo que éste es definido como un conjunto de campos. De esta manera se van amarrando los conceptos hasta llegar al término de **archivo**, que es un conjunto de registros.

Ahora bien, una base de datos se define como grandes cantidades de información almacenada en un espacio de memoria (de PC, servidor, laptop, etcétera), que es procesada para su posterior uso, según las necesidades de información de la empresa, institución, etc., dicha información ayudará a la toma de decisiones y ésta a su vez dará soluciones. Una de las principales ventajas de las bases de datos es que se puede manejar información compartida en red, que en cualquier parte del mundo se pueda consultar, siempre y cuando se cuente con los permisos de acceso, si en su caso así lo requiere.

Las bases de datos pueden ser diseñadas de manera lógica o documental para diseñar prototipos, pero la realidad de las cosas es que se pueden diseñar en algunas aplicaciones, desde

Unidad 1. Introducción a las bases de datos

las más sencillas como Excel y Access, hasta otras más complejas, las cuales abordaremos más adelante en los contenidos.

Componentes que conforman las bases de datos. Para hablar sobre los componentes que conforman las bases de datos, es importante comenzar con el **software**; éste está constituido por un conjunto de programas que se conocen como Sistema de Gestión de Base de Datos (DBMS, por sus siglas en inglés: *Database Management System*). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.

Las bases de datos son utilizadas por usuarios en diferentes niveles, pero antes de abordar cada uno de ellos es preciso aclarar que un usuario es aquella persona que utiliza la base de datos dependiendo de sus necesidades. Existe el usuario quien la utilizará (usuario final), quien ejecuta las aplicaciones y el que administra la Base de datos, cada uno de ellos tiene un nivel de acceso a la Base de datos. El **programador de aplicaciones** es el encargado de crear programas de aplicación que utiliza la base de datos; el **usuario final** es quien accesa a la base de datos por medio de un lenguaje de consulta o de programas de aplicación, sin ser el especialista; mientras que el **administrador de la Base de datos** (DBA, por sus siglas en inglés: *Data Base Administrator*) es quien se encarga del control general del Sistema de Base de Datos.

Todas las bases de datos tienen como punto de partida una tabla, que es la forma de organizar la información en filas y columnas para que su acceso sea más fácil y veloz. Las filas de datos las conocemos como registros, mientras que las columnas se llaman campos. Así mismo existe un campo clave, que asigna a cada registro un número irrepetible y, de ser necesario secuencial. Esta llave es muy importante puesto que es común que algunos registros repitan su información, y con ella se le otorga a cada registro una diferencia que le otorga unicidad. Por su nombre en inglés es común verla referida con sus siglas PK (primary key).


Un buen ejemplo de una base de datos es una lista de nombres y teléfonos de un grupo de personas:

TABLA O ENTIDAD

Campo
Clave

id	Nombre	Apellido	Teléfono
001	Lucero	Maza	Ontiveros
002	Gabriela	Sánchez	Medina
003	Berenice	Ortega	Maza
004	Celia	Correa	González

Campos



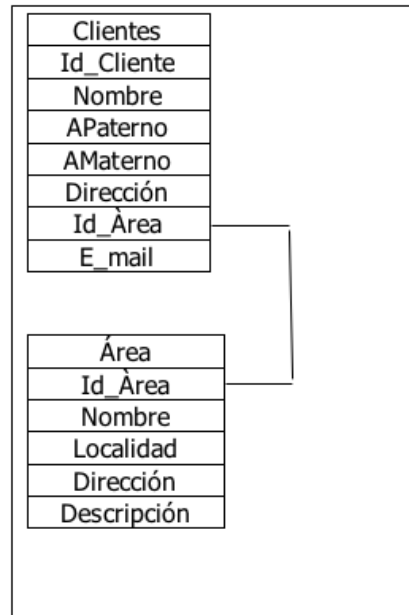
Registros

Ejemplo de una tabla en Access

Aparte del campo clave, existe la clave externa, también conocida como llave foránea, la cual es uno o más campos de una tabla que hacen referencia al campo o campos de la clave principal

Unidad 1. Introducción a las bases de datos

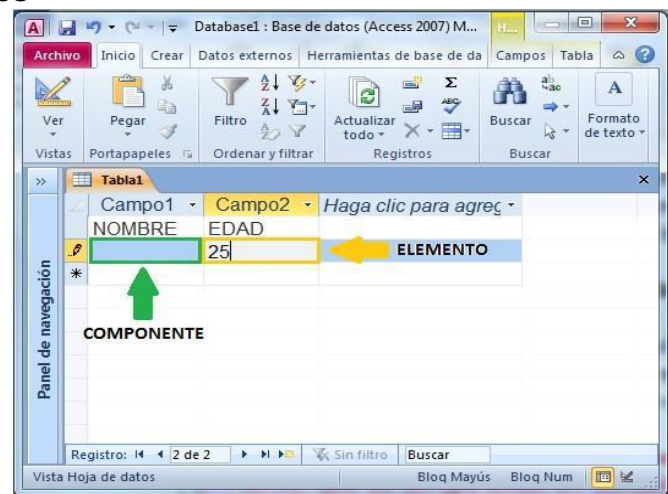
de otra tabla. Una clave externa se encarga de indicar cómo están relacionadas las tablas. Es importante mencionar que los datos en los campos de clave externa y clave principal deben coincidir, aunque los nombres de los campos no sean los mismos. De la misma forma como se selecciona la llave primaria, se selecciona la llave o clave externa o foránea. Esta llave tiene la tarea específica de determinar en el momento de las relaciones entre las tablas dónde la llave primaria realiza la unión con el otro campo de otra tabla y automáticamente se vuelve una llave foránea. Las siguientes tablas son un ejemplo de la relación de llaves primarias con llaves foráneas.



Como se puede observar, la llave primaria de cada una de las tablas se encuentra en otra tabla donde se genera la relación, misma que, en la otra tabla se convierte en llave ó clave secundaria.

Elementos que conforman las bases de datos

En esta parte es importante tomar en cuenta la diferencia que existe entre un componente y un elemento de la base de datos; cuando se habla de un elemento es cuando se utiliza ya un gestor, sin embargo, cuando se habla de componente, es meramente teórico.



Unidad 1. Introducción a las bases de datos

Ventajas de las bases de datos

Dentro de las ventajas primordiales que tienen las bases de datos se encuentran:

- Ahorro de tiempo, ya que los mismos datos se introducen una sola vez.
- Ahorro de espacio, porque la base de datos tiene dimensiones más reducidas.
- Disminución de errores determinados por la introducción de datos, ya que no se presenta redundancia de datos.
- Fiabilidad y disponibilidad; comúnmente la fiabilidad se define a grandes rasgos como la probabilidad de que un sistema esté en funciones en un momento determinado, y la disponibilidad es la probabilidad de que el sistema esté disponible continuamente durante un intervalo de tiempo, la posibilidad de compartir los datos al tiempo que se mantiene un cierto grado de control local.
- Mejor rendimiento cuando una base de datos grande es distribuida en múltiples sitios, ya que hay bases de datos más pequeñas en cada uno de éstos, y, en consecuencia, las consultas locales y las transacciones que tienen acceso a datos a un sólo sitio tienen un mejor rendimiento porque las bases de datos son más pequeñas, además, cada sitio tiene un mejor número de transacciones en ejecución, que si todas las transacciones se enviaran a una sola base de datos centralizada.

Desventajas

Como en cualquier sistema informático, el margen de error siempre se presenta, y las bases de datos no son la excepción, la principal desventaja que tienen se refiere al control y al manejo de los datos, porque éstos residen en muchos nodos diferentes y se pueden consultar por nodos diversos de la red; la probabilidad de violaciones de seguridad es creciente si no se tiene en cuenta la seguridad, que es un punto primordial para resguardar la información de manera fiable. Las fallas no predecibles, tanto de componentes de hardware como de software, son complejas, ya que tratando de evitar estas fallas la integridad de la información no tendrá problemas. La integridad se refiere a la consistencia, validez y exactitud de la información.

Algunas otras desventajas son que los datos pueden estar replicados, así como también el control de concurrencia y los mecanismos de recuperación que son mucho más complejos que en un sistema centralizado, y la distribución produce un aumento en la complejidad del diseño y en la implementación del sistema.

1.2. Tipos de Bases de datos

El procesamiento de datos ha impulsado al crecimiento de las computadoras, como ocurrió cuando éstas tuvieron muy fuerte auge en las computadoras comerciales, de hecho la automatización de las tareas de procesamiento de datos precede a las computadoras. Las tarjetas perforadas, inventadas por Hollerith, se usaron en los principios del siglo XX para registrar los datos del censo de los EE.UU, se utilizaron sistemas mecánicos para procesar las tarjetas y

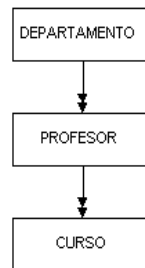
Unidad 1. Introducción a las bases de datos

para tabular los resultados. Las tarjetas perforadas se usaron ampliamente como medio para introducir datos de las computadoras posteriormente. La evolución de los sistemas de bases de datos inició de la mano con la evolución de los soportes de almacenamiento de información. En un inicio alrededor de los años 50 se desarrollaron las cintas magnéticas, lo que fue dejando detrás a las tarjetas perforadas. Con el surgimiento de los discos fijos el manejo de datos cambió de manera radical, ya que éstos permitían el acceso directo a los datos, sin importar su ubicación u orden. Este cambio permitió el desarrollo de bases de datos en red y jerárquicas, con lo que las estructuras de datos, además de poder construir y manipular directamente estas estructuras.

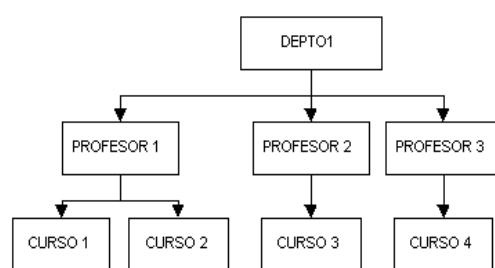


Modelo de Red

Estructura lógica

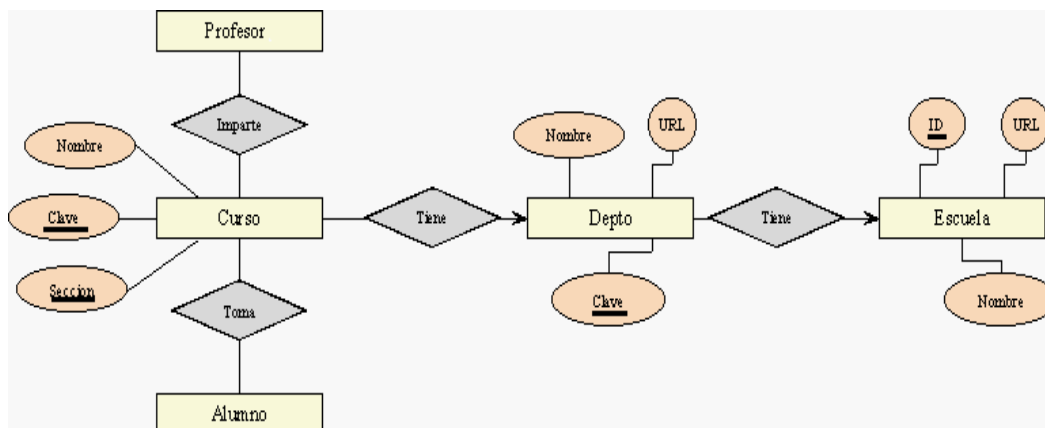


Ejemplo de base de datos



Modelo Jerárquico

Después surgió el modelo no relacional y el de formas procedimentales de consulta, que tenía como ventaja la simplicidad de desarrollo y la posibilidad de ocultar los detalles de implementación al programador. A pesar de que este modelo llamó la atención de muchos programadores no se utilizaba en la práctica, porque su rendimiento no fue tan alto como las bases de datos de red y jerárquicas existentes. Esto cambió cuando **IBM** desarrolló el prototipo *System R*, con lo que se desarrolló el primer producto comercial de construcción de bases de datos relacionales: SQL/DS. Estos sistemas de bases de datos rápidamente sustituyeron las bases de datos jerárquicas y de red, que eran de bajo nivel, mientras que las relacionales las operaciones que eran de ese nivel se realizaban automáticamente, de modo que el programador no se tenía que concentrar tan arduamente en el nivel lógico.



Modelo Relacional

Unidad 1. Introducción a las bases de datos

Estos modelos de bases de datos se han convertido en el modelo dominante, pero más importante que eso, han sido el punto de partida para apuntalar la investigación que generó el desarrollo de bases de datos paralelas y distribuidas así como los primeros proyectos de bases de datos orientadas a objetos. Mientras que en la década de los ochenta fueron las aplicaciones de transacciones, en las que se hacen actualizaciones constantemente, pero gracias al desarrollo del SQL se comenzaron a utilizar para la toma de decisiones, que se basan más en la consulta, lo que a su vez llevó a un importante crecimiento en las herramientas para analizar grandes cantidades de datos.

El que quizá se convirtió en el acontecimiento clave para el desarrollo del uso de bases de datos fue el crecimiento del World Wide Web, lo que llevó su implementación a un público más extenso con el desarrollo de interfaces basadas en red. Lo anterior da cuenta de cómo han aparecido los diferentes tipos de bases de datos. Ahora bien, antes de continuar con el siguiente tema, es conveniente centrar nuestra atención en algunas de las características de los tipos de bases de datos más conocidos:

Modelo Jerárquico

El modelo jerárquico es un tipo de Sistema Gestor que organiza la información en forma de árbol genealógico, en el que un nodo puede tener a su vez varios nodos que deriven de él.

Modelo de Red

El modelo de red parte de una base de datos integrada por una colección de registros, conectados por medio de enlaces en una red.

Modelo Relacional

El funcionamiento del modelo relacional, parte de lo que su nombre indica: relaciones. Estas pueden ser consideradas en forma lógica, y genera conjuntos de datos llamados registros. De forma clara se pueden explicar como una tabla compuesta por registros y campos.

Al surgir nuevos lenguajes de Programación, también surgen nuevos modelos de Bases de Datos para cubrir las necesidades de ciertas aplicaciones, como son por ejemplo los Modelos de Bases de Datos Orientados a Objetos. Este modelo es reciente, este tipo de BD se diseña para trabajo en conjunción con lenguajes de programación Orientados a Objetos. Los cuáles proporcionan y manipulan información que puede ser presentada como objetos y de ágil acceso.

Entre los principales conceptos que utilizan las BDOO son:

- Identidad de objetos
- Clases
- Herencia
- Polimorfismo
- Encapsulamiento
- Compatibilidad con los lenguajes de Programación Orientada a Objetos
- Entre otros.

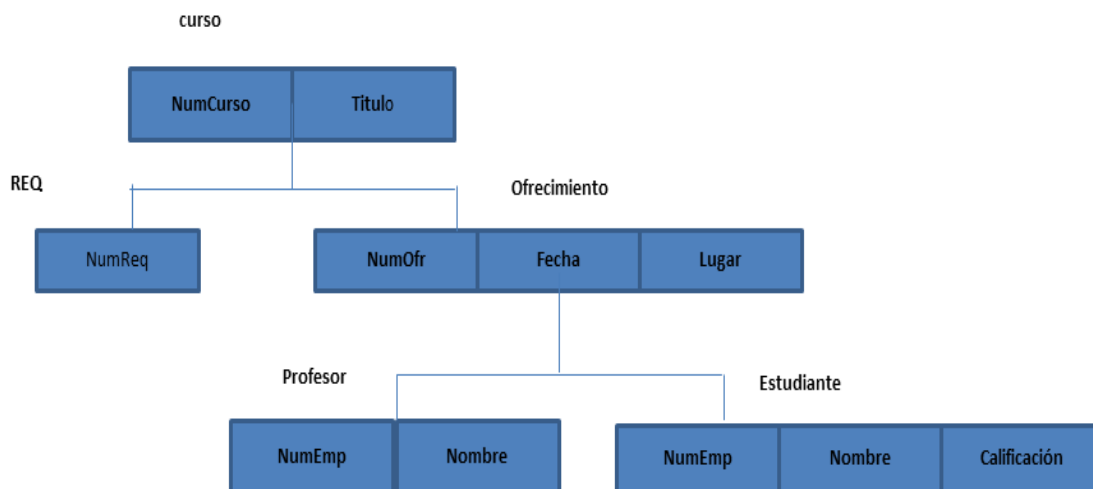
Unidad 1. Introducción a las bases de datos

1.2.1. Base de datos de árbol (jerárquica)

Una Base de datos jerárquica se compone de un conjunto ordenado de árboles, dicho de manera más precisa, un conjunto ordenado formado por múltiples ocurrencias de un sólo tipo de árbol. Para comprender mejor esto, se analizan primero los tipos y después se tratarán las ocurrencias.

Un tipo de árbol consiste en un sólo tipo de registro “raíz”, junto con un conjunto ordenado de cero o más tipos de subárbol dependientes (de nivel más bajo). Un tipo de subárbol a su vez consiste en un sólo tipo de registro de la raíz -tipo del subárbol- junto con un conjunto ordenado de cero a más tipos de subárboles dependientes de nivel más bajo, y así sucesivamente. Por tanto, el tipo de árbol completo es un arreglo jerárquico de tipo de registro, además los tipos de registro están formados por tipos de campo.

Por ejemplo, consideremos la base de datos académica de la siguiente figura, la cual contiene información acerca del sistema educacional interno de una compañía industrial grande. La compañía en cuestión mantiene un departamento educacional, cuya función es impartir varios cursos de adiestramiento para los empleados de la compañía; cada curso se ofrece en diferentes lugares dentro de la organización, y la base de datos contiene detalles tanto de los ofrecimientos ya impartidos, como de los programados para impartirse en el futuro.



Estructura de la base de datos académica.

La base de datos contiene la siguiente información:

- Para cada curso: número de curso, título de curso, detalles de todos los cursos que sean requisitos inmediatos y detalles de todos los ofrecimientos.
- Para cada curso que sea requisito de un curso dado: número de curso de ese requisito.
- Para cada ofrecimiento de un curso dado: número de ofrecimiento, fecha, lugar, detalles de todos los profesores y detalles de todos los estudiantes.
- Para cada profesor de un ofrecimiento dado: número de empleado y nombre.

Unidad 1. Introducción a las bases de datos

- Para cada estudiante de un ofrecimiento dado: número de empleado, nombre y calificación.

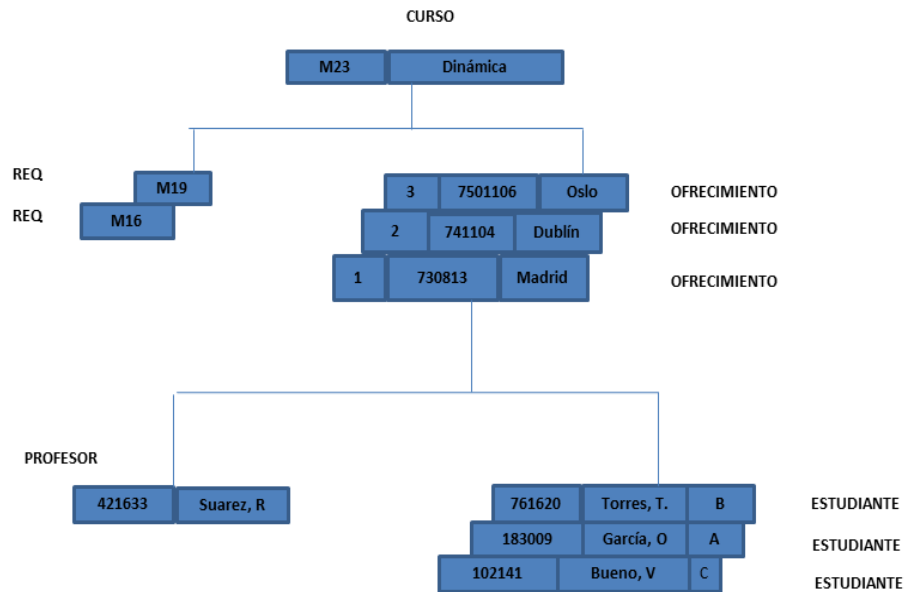
El tipo de árbol para la base de datos académica tiene CURSO como su tipo de registro de raíz y posee dos tipos de subárbol, cuyas raíces son los tipos de REQ y OFRECIMIENTO, respectivamente (y adviértase que este conjunto de dos tipos de subárbol está ordenado; es decir, el tipo de subárbol REQ precede al tipo de subárbol OFRECIMIENTO, como se muestra en la figura). El tipo de subárbol con raíz en REQ es sólo raíz; en cambio, el tipo de subárbol con raíz en OFRECIMIENTO tiene a su vez dos tipos de subárbol de nivel más bajo, los dos de sólo raíz, cuyas raíces son los tipos de registro PROFESOR Y ESTUDIANTE, respectivamente. Una vez más, los tipos de subárbol están ordenados. Así pues, la base de datos contiene cinco tipos de registro: CURSO, REQ, OFRECIMIENTO, PROFESOR y ESTUDIANTE. CURSO como tipo de registro raíz, y los demás como tipos de registros dependientes. CURSO es el tipo de registro padre de los tipos de registro REQ y OFRECIMIENTO, y éstos son tipos de registro hijos de tipo de registro CURSO. De manera similar, OFRECIMIENTO es el tipo de registro padre de PROFESOR y ESTUDIANTE, y éstos son tipos de registro hijos de OFRECIMIENTO. La conexión entre un tipo hijo dado y su tipo padre correspondiente, se llama tipo liga.

Cierta información que en una base de datos relacional se representaría con claves ajenas, en una base de datos jerárquica se presenta mediante ligas padre-hijo. En la base de datos académica, por ejemplo, la conexión entre los ofrecimientos y los cursos está representada no por un campo NumCurso en el registro OFRECIMIENTO, sino por la liga CURSO-OFRECIMIENTO.

Una vez tratada la cuestión de los tipos, a continuación se revisan las ocurrencias.

La terminología raíz/padre/hijo, que se señaló anteriormente en los tipos, se refleja también en las ocurrencias. De esta manera, cada ocurrencia de árbol consiste en una sola ocurrencia de registro raíz, junto con un conjunto ordenado de cero o más ocurrencias de cada uno de los tipos de subárbol que dependen en forma inmediata del tipo de registro raíz. Cada una de esas ocurrencias de subárbol a su vez consiste también de una sola ocurrencia de registro -la raíz de la ocurrencia de subárbol- junto con un conjunto ordenado de cero o más ocurrencias de cada uno de los tipos de subárbol que dependen en forma inmediata de este tipo de registro raíz, y así sucesivamente. Dicho de otro modo, para cualquier ocurrencia de cualquier tipo de registro padre, existen n ocurrencias de cada uno de sus tipos de registro hijos (donde n es mayor o igual a cero).

Unidad 1. Introducción a las bases de datos



Muestra de un árbol de la Base de datos académica.

La explicación del árbol de la figura es la siguiente; por definición, ese árbol contiene una sola ocurrencia de CURSO (la raíz del árbol). Ese CURSO tiene dos requisitos subordinados y tres ofrecimientos subordinados (en términos más precisos, esa ocurrencia de CURSO tiene un conjunto ordenado de dos ocurrencias de REQ subordinadas y un conjunto ordenado de tres ocurrencias de OFRECIMIENTO subordinadas). Se dice que CURSO es una ocurrencia padre, con dos hijos REQ y tres hijos OFRECIMIENTO, es decir, ocurrencias hijas. De manera similar, el primer ofrecimiento también es un padre con un hijo PROFESOR y varios hijos ESTUDIANTE (sólo se muestran tres). Los otros dos ofrecimientos no tienen hijos PROFESOR ni ESTUDIANTE en este momento, además, cada (ocurrencia) padre se considera como el padre en tantas ligas (es decir, ocurrencia de liga) como hijos tenga ese padre, es decir, el registro CURSO. Por ejemplo, es el padre en dos ocurrencias de la liga CURSO – REQ, y el padre en tres ocurrencias de la liga CURSO-OFRECIMIENTO. Por otro lado, cada (ocurrencia) hija es el hijo en una y sólo una (ocurrencia de) liga.

Además de lo expuesto anteriormente, hay un vocablo que se aplica a las ocurrencias y no a los tipos: el término “gemelo”. Se dice que todas las ocurrencias de un tipo hijo dado que comparten la misma ocurrencia padre son gemelas. Así, por ejemplo, los tres ofrecimientos de la anterior figura son gemelos (aunque haya tres). Es preciso mencionar que los requisitos en esa figura no son gemelos de esos ofrecimientos, porque aunque tienen el mismo padre, son de un tipo diferente.

A lo largo del subtema se ha hecho mención frecuentemente del concepto de ordenamiento, de hecho la noción de ordenamiento es fundamental en la estructura de datos jerárquica. Considerando algún tipo de subárbol T, cuyo registro raíz es R y cuyos tipos de subárbol son S1, S2..., Sn (en ese orden). Sea t una ocurrencia de T, cuya raíz es r (una ocurrencia de R) y cuyos

Unidad 1. Introducción a las bases de datos

subárboles son s_1, s_2, \dots, s_n (ocurrencias de S_1, S_2, \dots, S_n , respectivamente). Definimos entonces de manera recursiva la secuencia jerárquica de t como la secuencia que se obtiene tomando primero el registro r , seguido de todos los registros de s_1 en secuencia jerárquica, después de todos los registros de s_2 en secuencia jerárquica, seguidos de todos los registros de s_n en secuencia jerárquica. La secuencia jerárquica del árbol anterior es la siguiente:

CURSO	M23
REQ	M16
REQ	M19
OFRECIMIENTO	1
PROFESOR	421633
ESTUDIANTE	102141
ESTUDIANTE	183009
ESTUDIANTE	761620
OFRECIMIENTO	2
OFRECIMIENTO	3

Esto se da en orden de arriba hacia abajo y de izquierda a derecha, llamado también “ordenamiento de profundidad”.

Cada uno de los árboles en la base de datos se puede considerar como un subárbol de algún registro raíz “del sistema” hipotético, en consecuencia toda la base de datos se puede considerar como un sólo árbol. Por tanto, la noción de secuencia jerárquica antes definida se aplica a toda la base de datos así como a cada subárbol individual, es decir, el concepto de secuencia jerárquica define un ordenamiento total para el registro de la base de datos, y debe considerarse a las bases de datos almacenadas (lógicamente) de acuerdo con ese ordenamiento.

Manipulación de los datos en el modelo jerárquico

Un lenguaje para manipulación de datos con estructura jerárquica se compone de un conjunto de operadores para procesar datos representados en forma de árboles. Como ejemplos de tales operadores podemos mencionar los siguientes:

- Un operador para localizar un árbol específico en la base de datos; por ejemplo, un operador para localizar el árbol del curso M23.
- Un operador para pasar uno de estos árboles al siguiente; por ejemplo, un operador para pasar del árbol correspondiente al curso M23 al árbol siguiente en la secuencia jerárquica de base de datos.
- Operadores para pasar de un registro a otro dentro de estos árboles, desplazándose hacia arriba o hacia abajo por los diversos trayectos jerárquicos; por ejemplo, un operador para pasar de registro CURSO correspondiente al curso M23 al primer registro de OFRECIMIENTO de ese curso.

Unidad 1. Introducción a las bases de datos

- Operadores para pasar de un registro a otro de acuerdo con la secuencia jerárquica de la base de datos; por ejemplo, un operador para pasar de un registro PROFESOR de un OFRECIMIENTO determinado a un registro de ESTUDIANTE correspondiente a ese OFRECIMIENTO o a otro OFRECIMIENTO SUBSECUENTE.
- Un operador para insertar un registro nuevo en una posición especificada dentro de uno de esos árboles; por ejemplo, un operador para inserta un nuevo OFRECIMIENTO en el árbol del curso M23.
- Un operador para eliminar un registro especificado; por ejemplo, un operador para eliminar un cierto OFRECIMIENTO del árbol del curso M23.

Como se ha visto en los ejemplos pasados, tales operadores trabajan por lo regular a nivel de registros. Es verdad que ciertos sistemas manejan también operadores a nivel de conjuntos, pero tales operadores quedan fuera del alcance “modelo jerárquico” en la acepción usual del término.

1.2.2. Base de datos de red

Las bases de datos de red pueden considerarse como una forma extendida del modelo jerárquico definido en el subtema anterior. La distinción principal entre las dos es la siguiente: en una estructura jerárquica, un registro hijo tiene un sólo padre. Mientras que en una estructura de red, un registro hijo puede tener cualquier número de padres.

Una base de datos de red se compone de dos conjuntos, un conjunto de registros y un conjunto de ligas, en términos más precisos, un conjunto de ocurrencias múltiples de cada uno de varios tipos de registros junto con un conjunto de ocurrencias múltiples de uno de varios tipos de ligas. Cada tipo de liga implica dos tipos de registro, un tipo de registro padre y un tipo de registro hijo. Cada ocurrencia de un tipo de liga dado consiste en una sola ocurrencia del tipo de registro padre, junto con un conjunto ordenado de múltiples ocurrencias del tipo de registro hijo. Dado un tipo específico de liga (L) con tipo de registro padre (P) y tipo de registro hijo (H):

- a) Cada ocurrencia de P es el padre en una y sólo una ocurrencia de L;
- b) Cada ocurrencia de H es hijo en cuanto más una currencia de L; Además, por supuesto, los tipos de registro están formados por tipos de campo.

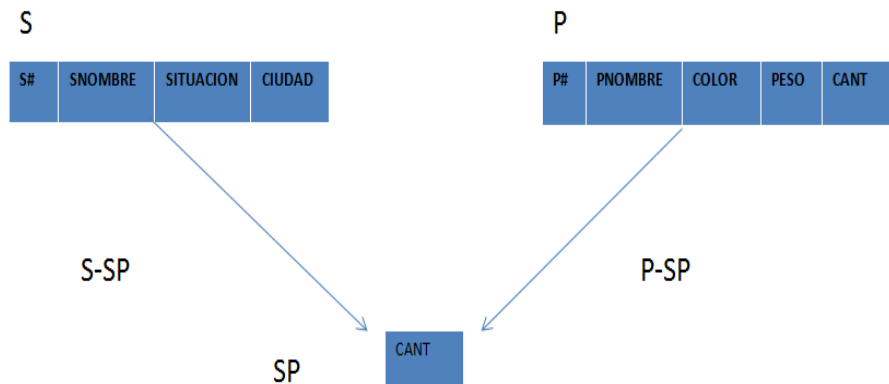
Nota: De ahora en adelante dejaremos a un lado los calificadores “tipo” y “ocurrencia”, cuando no provoque confusión hacerlo.

Como ejemplo de lo anterior se tiene en la siguiente figura, la forma como podría representarse la base de datos de proveedores y partes en forma de red. La Base de datos contiene tres tipos de registro: S, P y SP. Es decir que S y P son idénticos a sus contrapartes relacionales; SP, en cambio, contiene sólo un cambio de cantidad (CANT). En vez de dos claves ajenas SP, se tienen dos tipos de ligas S-SP y P-SP.

- a) Cada ocurrencia de S-SP está formada por una sola ocurrencia de S, junto con una ocurrencia de SP por cada envío del proveedor representado por esa ocurrencia de S.

Unidad 1. Introducción a las bases de datos

- b) Cada ocurrencia de P-SP está formada por una sola ocurrencia de P, junto con una ocurrencia de SP por cada envío de la parte representada por esa ocurrencia de P.



La base de datos de proveedores y partes: versión de red (estructura de los datos)

Manipulación de datos de red

Un lenguaje de manipulación de datos de red consiste en un conjunto de operadores para procesar datos representados en forma de registros y ligas. Como ejemplos de tales operadores podemos mencionar los siguientes:

- Un operador para localizar un registro específico, dado un valor de un campo de ese registro; por ejemplo, un operador para localizar el registro S del proveedor S1.
- Un operador para pasar del padre a su primer hijo en alguna liga; por ejemplo, un operador para pasar del registro S del proveedor S1 al registro SP del proveedor S1 y la parte P1 (a través de la liga S-SP).
- Un operador para pasar de un hijo al siguiente en alguna liga; por ejemplo, un operador para pasar del registro SP del proveedor S1 y la parte P1 al registro SP del proveedor S1 y parte P2 (a través de la liga S-SP).
- Un operador para pasar de un hijo a su padre dentro de alguna liga; por ejemplo, un operador para pasar de registro SP del proveedor S1 y la parte P2 al registro P de la parte P2 (a través de la liga P-SP).
- Un operador para crear un registro nuevo; por ejemplo, un operador para crear un registro S para un nuevo proveedor.
- Un operador para destruir un registro ya existente; por ejemplo, un operador para destruir el registro S de un proveedor ya existente.
- Un operador para conectar un registro hijo ya existente dentro de una liga; por ejemplo, un operador para conectar un registro SP específico dentro de las ligas S-SP y P-SP.
- Un operador para desconectar un registro hijo ya existente de una liga; por ejemplo, un operador para desconectar un registro SP específico de las ligas S-SP y P-SP.

Unidad 1. Introducción a las bases de datos

- Un operador para desconectar un registro hijo ya existente de una ocurrencia de un tipo de liga dado y reconectarlo dentro de otro; por ejemplo, un operador para desconectar un registro SP específico de la liga S- SP con padre S1 y reconectarlo dentro de la liga S-SP con padre S2.

Es importante mencionar que, como lo sugiere el ejemplo, tales operadores trabajan por lo regular a nivel de registros, como en los modelos de lista invertida y jerárquico.

1.2.3. Base de datos relacional

Las bases de datos relacionales, que en la actualidad son las más utilizadas, trabajan a través de tablas, que están conformadas de filas y columnas para la realización de sus búsquedas y otras operaciones. Estas tablas se encuentran relacionadas entre sí, de ahí el nombre de relacional.

Las tablas están conformadas por **campos**, que son las características de las tablas; **dominios**, que son los valores posibles que puede contener un campo; y un conjunto de dominios relacionados entre sí, en términos de bases de datos se les conoce como **tuplas**. Cada tabla está compuesta de llaves primarias (PK, por sus siglas en inglés: *Primary Key*) y llaves foráneas (FK, por sus siglas en inglés: *Foreign Key*) que serán quienes hagan la relación entre diferentes tablas de información. Las llaves primarias son campos únicos que identifican a la información como única e irrepetible; que no permiten duplicidad de información. Mientras que las llaves foráneas, también llamadas llaves externas, son campos que hacen la relación entre las tablas existentes en la base de datos, estas llaves deben de existir primero en otras tablas como llaves primarias.

Para tener una definición más clara, el autor Javier Quiroz comenta que la estructura del modelo relacional está fundamentada por la *relación*, es decir, una tabla bidimensional constituida por filas (tuplas) y columnas (atributos). Sus relaciones se presentan en las tablas que están formando parte de la base de datos. Todas y cada una de las instancias de la entidad encontrarán sitio en una tupla de la relación, mientras que los atributos de la relación representan las propiedades de la entidad en las columnas de la tabla. Por ejemplo, si en la base de datos se tienen que representar personas, podrá definirse una relación llamada *Personas*, cuyos atributos describen las características de las personas y cada tupla de la relación *Personas* representará una persona concreta, esto es, la relación. Otra definición dice que una base de datos relacional es un repositorio compartido de datos.

Para hacer disponible los datos de una base de datos relacional a los usuarios, hay que considerar los siguientes aspectos:

- **La forma en la cual el usuario solicita los datos.** ¿Cuáles son los diferentes lenguajes de consulta que utilizan? SQL es el lenguaje de consulta más usado, así como QBE y Datalog, que ofrecen enfoques alternativos a la consulta de datos relacionales.

Unidad 1. Introducción a las bases de datos

- La integridad de los datos y la seguridad. Las bases de datos necesitan proteger los datos del daño provocado por los usuarios, ya sean intencionados o no. El componente de mantenimiento de la integridad de una base de datos asegura que las actualizaciones no violan las restricciones de integridad que hayan especificado sobre los datos. El componente de seguridad de una base de datos incluye la autenticación de usuarios y el control de acceso para restringir las posibles acciones de cada usuario. Los aspectos de integridad y seguridad se presentan independientemente del modelo de datos, pero se estudian en el contexto de modelo de datos relacional para ejemplificarlos. Las restricciones de integridad forman la base del diseño de base de datos relacionales.
- **El diseño de base de datos relacionales.** El diseño del esquema relacional es el primer paso en la construcción de aplicaciones de base de datos. Este diseño de esquemas es de los principios que se pueden usar para distinguir los buenos diseños de bases de datos que se formalizan mediante varias formas normales que ofrecen diferentes compromisos entre la posibilidad de inconsistencias y la eficiencia de ciertas consultas.

Estructura básica de las bases de datos relacionales

Como ya se explicó estas consisten en relaciones, y cada una de ellas tiene un nombre único. Cada relación permite usar valores nulos, que indican que el valor puede ser inexistente o nulo.

En estas bases cada fila representa la relación de un conjunto de valores. Cada tabla es un conjunto de dichas relaciones, por lo que existe correspondencia entre el concepto de tabla y de relación. Su estructura la ejemplificamos en la siguiente tabla.

Numero-	Nombre-sucursal	Saldo
c-	Centro	500
101	Bancomer	400
c102	Galapagar	900
c-	Becerril	700
201	Galapagar	750
c-	Moralzarzal	700
215	Collado	350

Base de datos relacional

Unidad 1. Introducción a las bases de datos

1.2.4. Sistema Gestor de Base de Datos (SGBD)

Al tener los conceptos, características, ventajas y desventajas de los tipos de Bases de datos, es preciso mencionarlo que son los Sistemas Gestores de Bases de Datos (SGBD o DBMS, por sus siglas en inglés: *Data Base Management System*).

Un Sistema Gestor de Base de Datos es un software específico dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Su objetivo principal es manejar de manera ordenada y clara un conjunto de datos que se convertirán en información importante para una organización. En otras palabras, un SGBD es un software que sirve para crear, programar, administrar y controlar la información de una base de datos, asegurando la integridad, seguridad y confiabilidad de la información.

Existen dos tipos de Sistemas Gestores de Bases de Datos: libres y comerciales.

Los más utilizados son:

Libres

- **PostgreSQL.** Es un sistema de gestión de base de datos relacional orientada a objetos, publicadobajo la licencia BSD (*Berkeley Software Distribution* / Distribución de Software Berkeley), que no es más que una licencia de software otorgada para este tipo de sistemas. Es una licencia de software libre, como la GPL (*General Public License* / Licencia Pública General) pero ésta es más cercana al dominio público. En el caso de BSD, permite el uso del código fuente en software no libre, a diferencia de GLP.

- **MySQL.** Es un sistema multiusuario y de código abierto. El uso de MySQL es muy popular en aplicaciones web, y es componente de las plataformas LAMP, MAMP, WAMP, entre otras, y suele combinarse con el popular lenguaje PHP. como sus principales características podemos decir que: está escrito en C y C++; utiliza *multi- threaded* mediante *threads* de kernel; pueden usarse fácilmente múltiples CPUs si están disponibles; emplea el lenguaje SQL para consultas a la base de datos; está disponible como freeware bajo licencia GPL; proporciona sistemas de almacenamiento, transaccionales y no transaccionales; trabaja en las plataforma AIX, BSDi, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OS/2 Warp, QNX, SGI IRIX, SunOS, SCO OpenServer, Solaris, SCO UnixWare, Tru64 y Microsoft Windows (95, 98, ME, NT, 2000, XP y Vista); utiliza tablas en disco B-tree muy rápidas con compresión de índice y es relativamente sencillo añadir otro sistema de almacenamiento, esto es útil si se desea añadir una interfaz SQL para una Base de datos propia.

Comerciales

- **Oracle.** Es una herramienta imporante y con muchas capacidades del tipo cliente/servidor para gestión de base de datos, aunque de precio muy alto.

Unidad 1. Introducción a las bases de datos

1.2.5. Estructura de un Sistema Gestor de Base de Datos

En apartados anteriores se abordó lo que eran los Sistemas Gestores de Bases de Datos, también conocidos como Sistemas de Gestión de Bases de Datos, por lo que en esta sección se abarcará la estructura y el funcionamiento de un SGBD.

Ahora bien, la estructura de un SGBD está conformada por:

- **Procesador de consultas:** Es su elemento principal de un SGBD, transforma las consultas en instrucciones en lenguaje de bajo nivel, que se envían al gestor.
- **Gestor:** Es la interfaz entre los datos de bajo nivel almacenados en la base de datos, acepta consultas y examina los esquemas externo y conceptual, posteriormente determina qué registros se requieren para satisfacer la petición.
- **Gestor de ficheros:** Se encarga de manejar y controlar los ficheros en almacenamiento en la base de datos.
- **Lenguaje de Manipulación de Datos (MDL):** trabaja con el procesador de consultas para el código apropiado.
- **Lenguaje de Definición de Datos (LDD o DD:** por sus siglas en inglés *Data Definition Language*): Convierte las sentencias en un conjunto de tablas que contienen datos almacenados en el diccionario de datos.
- **Diccionario de datos:** Almacena la información de la base de datos.
- **Gestor del diccionario:** Controla los accesos y la información del Diccionario de datos.
- **Control de autorización:** Se encarga de revisar las credenciales de los usuarios para acceder a determinada información.
- **Optimizador de consultas:** Establece la estrategia para el procesamiento de las transacciones.
- **Gestor de transacciones:** Es la parte encargada de realizar las transacciones.
- **Planificador:** Organiza las transacciones que llegan a la base de datos para su realización.
- **Gestor de recuperación:** Cuida la integridad de la base de datos en caso del algún fallo.
- **Gestor de buffers:** Es el enlace entre la memoria principal y la memoria de almacenamiento.

Cierre de la unidad

Ha concluido la primera unidad del curso. A lo largo de ésta se recordaron conceptos básicos informáticos, posteriormente se involucró al estudiante en la terminología específica de las bases de datos, cuyo propósito fue dar un panorama para identificar una base de datos en su entorno social, personal y laboral, y para poder elaborar un propio concepto, reconocer los tipos de bases de datos y distinguir sus particularidades. De la misma manera, se identificaron los Sistemas Gestores de Bases de Datos y sus características.

Unidad 1. Introducción a las bases de datos

Es aconsejable revisar nuevamente la unidad en caso de que los temas que se acaban de mencionar no sean familiares o no se recuerden, de no ser este el caso, ya se está preparado(a) para seguir con la unidad dos, en donde se continuará con la construcción del prototipo de una base de datos documental, al elegir un caso de estudio, llevar a cabo un análisis y un estudio de factibilidad, identificar los requerimientos del usuario, administrar tiempos, así como revisar modelados de abstracción de información. Todo ello con el fin de obtener el prototipo final al terminar la tercera y última unidad del curso Bases de datos.

Fuentes de consulta

- Álvarez, S., & Bravo, S. (2000). *Informática Automática*. Recuperado el 2 de Mayo de 2011, de Universidad de Salamanca: http://ocw.usal.es/enseanzas-tecnicas/aplicaciones-informaticas-para-humanidades/contenidos/Temas/Tema1-Fundamentos_de_Informatica_-_2ppt.pdf
- Celma, M.; Casamayor, J.C.; Mota, L. (2003) *Bases de datos relacionales*. Madrid: Pearson-Prentice Hall.
- MasterMagazine (2005) *Definición de ASCII*. Recuperado el 21 de enero de 2011, de <http://www.mastermagazine.info/termino/3926.php>
- Pérez L, César. (2008) *Oracle 10g: administración y análisis de bases de datos*. Segunda edición. México: Alfaomega.
- Quiroz, Javier. (2003) "El modelo relacional de bases de datos" en *Boletín de Política Informática* Núm. 6 (Versión electrónica). Recuperado el 26 de enero de 2011, de <http://www.inegi.org.mx/inegi/contenidos/espanol/prensa/contenidos/articulos/tecnologia/relacional.pdf>
- RAE Real Academia Española (2017). *Informática*. Recuperado de http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=inform%E1tica
- Silberschatz, Abraham. (2006). *Fundamentos de Bases de Datos*. España: McGraw-Hill.