TALLER No. 1 – BASES DE DATOS 1

STEVEN FLOREZ HERNANDEZ

CORPORACION DE ESTUDIOS TECNOLOGICOS DEL NORTE DEL VALLE

INGENIERIA DE SISTEMAS

BASES DE DATOS 1

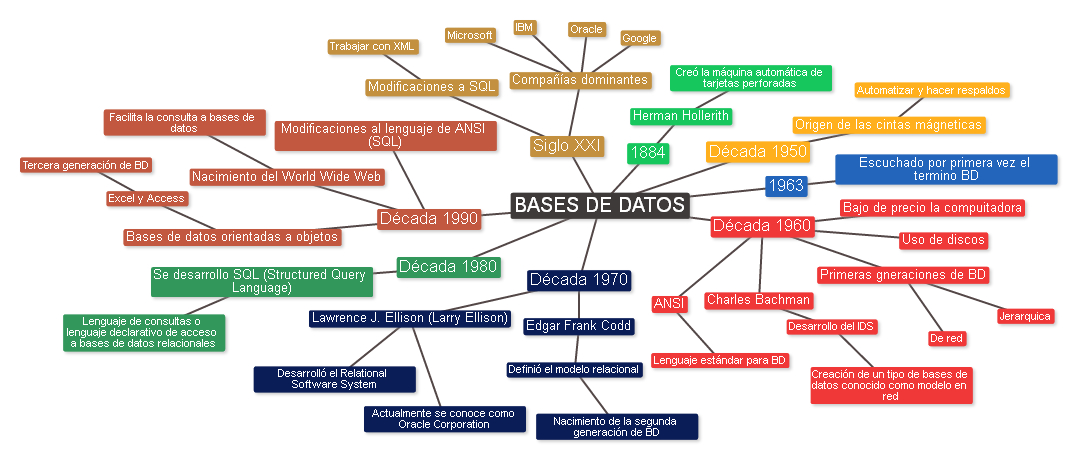
CARTAGO

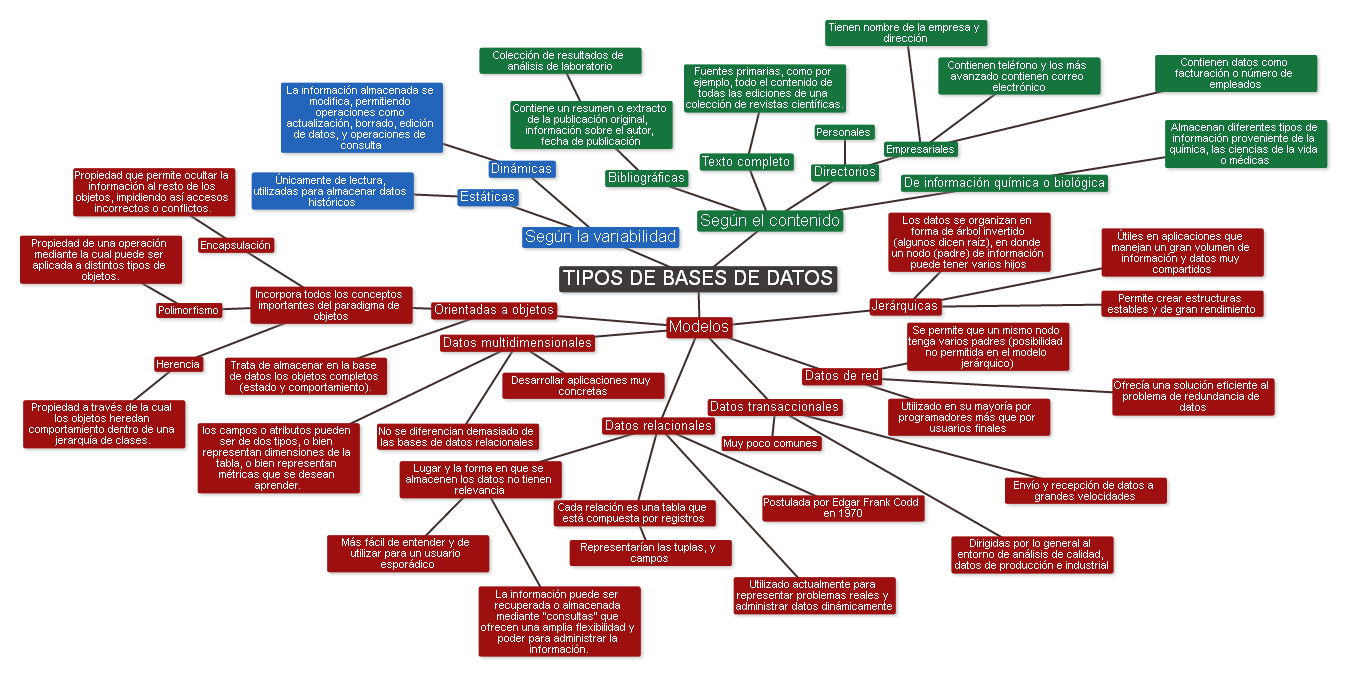
2016

1.  
Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada. La base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

2.



3.   
Dato: Es el elemento primario de la información conformado por símbolos (letras, números, dibujos, señas, gestos) que reunidos pueden cobrar significación. Solo o aislado el dato no posee relevancia, pero utilizado en las premisas de un razonamiento puede llevarnos a una conclusión. Surgen de la observación o la experiencia.

Tipo de dato: es un atributo de los datos que indica al ordenador (y/o al programador) sobre la clase de datos que se va a trabajar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar.

Campo: mínima unidad de información a la que se puede acceder; un campo o un conjunto de ellos forman un registro, donde pueden existir campos en blanco, siendo este un error del sistema operativo. Aquel campo que posee un dato único para una repetición de entidad, puede servir para la búsqueda de una entidad en específico.

Registro: (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla. En términos simples, una tabla de una base de datos puede imaginarse formada de filas y columnas o campos. Cada fila de una tabla representa un conjunto de datos relacionados, y todas las filas de la misma tabla tienen la misma estructura.Un registro es un conjunto de campos que contienen los datos que pertenecen a una misma repetición de entidad.

Campo clave: son aquellos que identifican los registros de una tabla de forma unívoca. Estos registros le añadirán rapidez a la tabla a la hora de realizar consultas tanto de actualización como de selección. El campo clave, no puede tener valores nulos y siempre debe tener un índice único.

Tabla: se refiere al tipo de modelado de datos, donde se guardan los datos recogidos por un programa. Su estructura general se asemeja a la vista general de un programa de hoja de cálculo. Una tabla es utilizada para organizar y presentar información. Las tablas se componen de filas y columnas de celdas que se pueden rellenar con textos y gráficos.

Consulta: una consulta es el método para acceder a los datos en las bases de datos. Forma de buscar y recopilar información de una o más tablas para conseguir información detallada de una base de datos.Con las consultas se puede modificar, borrar, mostrar y agregar datos en una base de datos. Para esto se utiliza un lenguaje de consultas.

Índice: es una estructura de datos que mejora la velocidad de las operaciones, por medio de identificador único de cada fila de una tabla, permitiendo un rápido acceso a los registros de una tabla en una base de datos. El índice tiene un funcionamiento similar al índice de un libro, guardando parejas de elementos: el elemento que se desea indexar y su posición en la base de datos.

Vista: una vista es una consulta que se presenta como una tabla (virtual) a partir de un conjunto de tablas en una base de datos relacional. Las vistas tienen la misma estructura que una tabla: filas y columnas. La única diferencia es que sólo se almacena de ellas la definición, no los datos.

Informes: tienen como objetivo proporcionar las herramientas necesarias para obtener una copia impresa de los datos existentes en una base de datos.

Guiones: Una o varias instrucciones (pasos de guion) que define para automatizar tareas repetitivas o difíciles. Los guiones se gestionan con la función Gestionar guiones. Puede ejecutar un guion haciendo clic en su botón, seleccionando su comando de menú, llamándolo desde otro guion o plug-in o ejecutándolo al arrancar o al cerrar un archivo.

Procedimiento: es un programa (o procedimiento) almacenado físicamente en una base de datos. Su implementación varía de un gestor de bases de datos a otro. La ventaja de un procedimiento almacenado es que al ser ejecutado, en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos, el cual usualmente corre en un servidor separado. Como tal, posee acceso directo a los datos que necesita manipular y sólo necesita enviar sus resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes.

BASES DE DATOS

Facilitan

Tienen

CONSULTAS

Se buscan por

Identificados por

CAMPOS CLAVE

REGISTROS

CAMPOS

Que Tienen

TABLAS

Compuestas por

INDICES

TIPOS DE DATOS

Forman

Formados por

DATOS

Pueden ser

VISTAS

PROCEDIMIENTO

Facilitados por

4. - Almacenar datos de manera masiva  
- Facilitar la búsqueda de los datos que se necesitan  
- Mantener ordenados todos los datos que ingresemos  
- Facilitar el análisis de los datos y obtener resultados de manera estadística  
- Para ayuda en la elaboración de aplicaciones informáticas.

5.

|  |  |
| --- | --- |
| NOMBRE | TAMAÑO EN TB |
| World Data Centre for Climate | 6364 |
| ChoicePoint | 250 |
| National Energy Research Scientific computing Center (NERSC) | 2867 |
| Library of Congress (Congreso de Estados Unidos) | 20 |
| Amazon.com | 42 |
| Sprint | 300 |
| Google | 800 |
| AT&T | 323 |
| YouTube | 45 |
| LexisNexis | 250 |
| Central Intelligence Agency (CIA): | Desconocida |

6.

Edgar Frank Kood

Nace 19 de agosto de 1923 en la Isla de Portland, Inglaterra. Hijo de un curtidor y una profesora, siendo el menor de siete hermanos. Asiste a la escuela secundaria de Poole, y posteriormente Estudió becado matemáticas y química en Oxford. Aunque podría haber evitado participar en la segunda guerra mundial por ser estudiante, se alistó en la Real Fuerza Aérea. A los 25 años viajó a los Estados Unidos y consiguió trabajo en IBM como programador matemático usando un prototipo de computador que ocupaba dos pisos completos de un edificio de oficinas en Manhattan. En 1953 emigró a Ottawa, Canadá, frustrado por la política McCarthy de persecución a los comunistas. Unos años más tarde volvió a Estados Unidos y obtuvó la ciudadanía, aunque nunca perdió su acento británico. En 1965 terminó un doctorado en computación de la Univ. de Michigan en Ann Arbor. Una evaluación negativa de su supervisor en Nueva York significó un traslado a los laboratorios de IBM en San José en 1967. Sería aquí que Codd conocería el mundo de las bases de datos, al que se dedicaría en los años siguientes.  
En 1978 Codd se divorció de su primera esposa, Elizabeth. En 1981 obtuvó el premio Turing de la ACM, el más importante en computación. La vida de Codd cambió en 1983, cuando sufrió una seria caída. Luego de recuperarse, jubiló de IBM y abandonó su diversión favorita: volar. Sin embargo siguió trabajando hasta 1999, en la consultora que formó con Chris Date y Sharon Weinberg, dos ex colaboradores de IBM. Sharon, después de doce años de cortejo, pasaría a ser su segunda esposa en 1990. En 1996 obtuvó el premio de la IEEE a pioneros de la computación. Los últimos años vivió en Williams Island, Florida. Coddtuvó cuatro hijos y tenía seis nietos.

Aporte.

En 1969 Edgar Codd inventó el modelo relacional, el modelo de bases de datos más usado hoy en día y para muchas personas, el único que conocen. Desde el sistema R de IBM a Oracle han pasado 30 años y aún es el modelo dominante. Inicialmente el apoyo de IBM a los sistemas de bases de datos tradicionales (de redes) era mayoritario, poderoso y agresivo. Sólo años más tarde, en 1978, durante una reunión técnica de alto nivel el modelo relacional llamó la atención del presidente de IBM, Frank Cary. Más tarde IBM anunció SQL/DS, su primer producto relacional comercial en 1981, seguido de DB2 en 1983. Sin embargo esta tardanza en adoptar el modelo relacional significó perder un mercado que tomaron otros. El trabajo inicial de Codd fue publicado en Communications of the ACM en 1970. Su trabajo sobre normalización de bases de datos fue publicado como un informe técnico de IBM en 1971. Ocho años más tarde, en ACM Transactions of DatabaseSystems, publicó varias extensiones al modelo relacional. En 1985 postuló una lista de 13 reglas que debía cumplir un producto de bases de datos para ser llamado relacional.

Larry Ellison

Lawrence Joseph Ellison nació el 17 de agosto de 1944 en Nueva York, su madre Florence Spellman, soltera, dio a luz con 19 años; posteriormente, entregó su hijo a su hermana para que fuera criado en Chicago. LillianSpellmanEllison y Louis Ellison –de origen ruso– adoptaron a Lawrence cuando este tenía nueve meses. Tras 48 años, Lawrence conoció finalmente a su madre. La identidad del padre es, hasta ahora, desconocida.  
En su época estudiantil, Larry fue un alumno brillante pero poco atento. Dejó la Universidad de Illinois al final de su segundo año sin presentarse a los exámenes finales debido a la muerte de su madre adoptiva. Tras pasar un verano en el norte de California donde vivió con su amigo ChuckWeiss, se matriculó en la Universidad de Chicago en la especialidad de ciencia de la computación. A los 20 años se trasladó definitivamente a California.  
Durante la década de 1970, Ellison trabajó para AmpexCorporation y luego en dos empresas más.  
Para la empresa en la que trabajaba, luego de que un contratista falló, realizó un desarrollo a la medida. Para esto, contrató a sus dos ex-jefes. Formaron una compañía. Al finalizar exitosamente el proyecto les dijo que no volverían a hacer desarrollo a la medida sino algo que pudieran vender muchas veces. Uno de los socios le presentó un artículo escrito por Edgar F. Codd sobre los sistemas de bases de datos relacionales llamado «un modelo relacional de datos para grandes bancos de datos compartidos».

Aporte

Él y sus dos socios fundaron Oracle en 1977, colocando sólo $1400 dólares de su propio dinero, bajo el nombre de Laboratorios de Desarrollo de Software (SDL). En 1979, la empresa pasó a denominarse Relational Software Inc., posteriormente rebautizada como Oracle después del éxito del producto insignia de base de datos Oracle. El nombre lo tomó de un proyecto de Inteligencia, al que le habían vendido la base de datos.

IBM había autorizado publicar el trabajo de Codd, luego de intentar fallidamente hacer un RDBMS IBM System R, que también se basaba en las teorías de Codd. IBM se negó a compartir código del sistema R. La versión inicial de Oracle fue la 2; no hubo Oracle 1. El número de versión pretendía dar a entender que ya se habían corregido todos los errores que pudieran existir en una versión anterior.

Roger Kent Summit

Roger Kent Summit (1930) es un informatólogo, documentalista y empresario estadounidense. Experto en sistemas de información, fue el fundador de la compañía de recuperación de información en línea Dialog y uno de los padres de la teledocumentación.

Roger Summit nace en Detroit en el estado de Míchigan (Estados Unidos). Se licenció en psicología en 1952 por la Universidad Stanford, doctorándose en 1965 con una tesis sobre gestión científica. Empezó a trabajar como ingeniero de la información en la compañía LookheedMissiles and SpaceCompany en 1962 y en 1967, con el fin de unir en línea a todos los archivos de los centros de la NASA, y con el propósito de recuperar la información por computadora, crearía RECON (la futura DIALOG).

En 1969, Summit amplió considerablemente las funciones del proyecto de teledocumentación: creó la primera red de información transeuropea con las bases de datos de la Agencia Espacial Europea, y creó ERIC, una base de datos con recursos educativos para investigadores y profesores de todo el país, el primer servicio de recuperación de información en línea que no versara sobre asuntos militares y de defensa.

En 1972, la compañía adopta el nombre de DIALOG, se establece en Palo Alto en el estado de California y se convierte en la empresa de información y documentación en línea más importante del mundo. Summit fue su presidente hasta 1991 cuando se jubiló, pasándo a tener un puesto emérito en la compañía.

Roger Summit trabajó en la armada de EE.UU.) entre 1952 y 1955 como oficial de comunicación. También ha desempeñado una labor académica dentro del campo de la Información y Documentación Científica, participando con distintas asociaciones bibliotecarias y documentales en asuntos de políticas de información como la American Library Associaton; ha publicado más de 100 artículos científicos. Roger Summit es miembro de numerosas sociedades y academias científicas, entre ellas la American Society of InformationScience and Technology o la Asociación Estadounidense para el Avance de la Ciencia.

En 1991, le fue concedido el Premio ASIST al Mérito Académico.

Bill Gates

(William Henry Gates III; Seattle, Washington, 1955) Informático y empresario estadounidense, fundador de Microsoft. La fortuna de este precoz programador, que no llegó a terminar sus estudios y que a los 31 años era ya multimillonario, procede del éxito arrollador de su sistema operativo, el MS-DOS (1981), que evolucionaría hasta convertirse en el popular Windows 3.1 (1992) y daría lugar a las sucesivas versiones de este sistema operativo, omnipresente hasta nuestros días en la inmensa mayoría de los ordenadores portátiles y de sobremesa.

Bill Gates nació en una familia acomodada que le proporcionó una educación en centros de élite como la Escuela de Lakeside (1967-73) y la Universidad de Harvard (1973-77). Siempre en colaboración con su amigo Paul Allen, se introdujo en el mundo de la informática formando un pequeño equipo dedicado a la realización de programas que vendían a empresas o administraciones públicas. En 1975 se trasladaron a Alburquerque (Nuevo México) para trabajar suministrando a la compañía MITS una serie de programas susceptibles de ser utilizados con el primer microordenador, el Altair, para el cual habían desarrollado una versión del lenguaje de programación BASIC.

Ese mismo año fundaron en Alburquerque su propia empresa de producción de software informático, Microsoft Corporation, con Bill Gates como presidente y director general. Su negocio consistía en elaborar programas adaptados a las necesidades de los nuevos microordenadores y ofrecérselos a las empresas fabricantes más baratos que si los hubieran desarrollado ellas mismas. Cuando, en 1979, Microsoft comenzó a crecer (contaba entonces con dieciséis empleados), Bill Gates decidió trasladar su sede a Seattle.

Microsoft ha creado algunos gestores de bases de datos muy importantes como Microsoft Access y Microsoft SQL Server, siendo de los más usados en el mundo.

Michael MontyWidenius

Ulf Michael Widenius (a menudo llamado Monty), nació el 3 de marzo de 1962, en Helsinki, Finlandia. Es el autor principal de la versión original de la base de datos de código abierto MySQL y miembro fundador de la empresa MySQL AB.

Después de estudiar en la Universidad Tecnológica de Helsinki (aunque sin llegar a graduarse), Widenius comenzó a trabajar para Tapio LaaksoOy en 1981. En 1985 fundó TCX DataKonsult AB (una empresa sueca de almacenamiento) con Allan Larsson En 1995 comenzó a escribir la primera versión de la base de datos MySQL con David Axmark, que fue lanzada en 1996. Es co-autor del Manual de Referencia de MySQL, publicado por O'Reilly en junio de 2002, y en 2003 fue galardonado con el premio al “empresario de software finlandés del año”. Hasta la venta de MySQL AB a Sun Microsystems en enero de 2008,1 era el director técnico de MySQL AB y siguió siendo una de las principales fuerzas detrás del desarrollo de MySQL. 2 Gracias a la venta de MySQL a Sun, Widenius tuvo beneficios cercanos a los 16,6 millones de euros en 2008 lo que le convirtió en una de las 10 personas con mayores ingresos en Finlandia ese año.3 4

El 5 de febrero de 2009, anunció que se iba de Sun para crear su propia empresa.5

El 12 de diciembre de 2009, Monty pidió a los clientes de MySQL que presionaran a la UE como consecuencia de la adquisición de Oracle de Sun, dado que le preocupaba que Oracle controlara el desarrollo de MySQL,6 esto dio lugar a una campaña de petición en línea llamado "Ayuda de MySQL".

Actualmente vive en Helsinki con su segunda esposa Anna y su hija Maria. Tiene también otra hija My (que parece ser que es la que dio nombre a MySQL) y un hijo Max (que dio nombre a MaxDB) de su primer matrimonio.

Aporte.

En 1995 comenzó a escribir la primera versión de la base de datos MySQL con David Axmark, lanzado en 1996. Él es el co-autor del Manual de referencia de MySQL, publicado por O'Reilly en junio de 2002, y en 2003 fue galardonado con el premio del año de software finlandés.

Autor principal de la versión original de la base de datos de código abierto MySQL.  
Actualmente se encuentra trabajando en una rama de del código base de MySQL, en un proyecto llamado MariaDB, en honor a su hija menor. Este proyecto incluye varios parches y plugins desarrollados la comunidad de software libre.

7.   
SGBD

Definición: Sistema de gestión de base de datos ó en inglés Databasemanagementsystem (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una bases de datos.

El objetivo de estos programas, consiste en proporcionan las interfaces entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas al sistema.

Principales Funciones.

Un S.G.B.D. debe incluir al menos las siguientes funciones.

Definición de los Datos. El SGBD debe ser capaz de aceptar las definiciones de datos en versión fuente y convertirlas en la versión objeto. El SGBD debe incluir componentes procesadores para cada uno de los lenguajes de definición de datos (LDD). (DATE, 1993).

Manipulación de los Datos. El SGBD debe atender las solicitudes de los usuarios para extraer, actualizar, adicionar o suprimir datos. El SGBD debe incluir un componente procesador del Lenguaje de manipulación de datos (LMD).

Seguridad e Integridad de los Datos. El SGBD debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el Administrador de la Base de Datos DBA. (DATE, 1993).

Recuperación y concurrencia de los datos. El principal objetivo de la implantación de una base de datos es poner a disposición de un gran número de usuarios en conjunto integrado de datos, estos datos podrán ser manipulados por los diferentes usuarios y es ahora cuando se debe garantizar la coherencia de los datos después de las diversas manipulaciones.

Estos temas ya se han analizado con anterioridad por consiguiente no se describirán.

Tipos de SGBD

Bases de datos centralizadas.

Los sistemas de bases de datos centralizados son aquellos que se ejecutan en un único sistema informático sin interaccionar con ninguna otra computadora. Tales sistemas comprenden el rango desde los sistemas de bases de datos monousuario ejecutándose en computadoras personales hasta los sistemas de bases de datos de alto rendimiento ejecutándose en grandes sistemas.

Una computadora moderna de propósito general consiste en una o unas pocas CPU's y un número determinado de controladores para los dispositivos que se encuentren conectados a través de un bus común, el cual proporciona acceso a la memoria compartida. Las CPU's poseen memorias caché locales donde se almacenan copias de ciertas partes de la memoria para acelerar el acceso a los datos. Cada controlador de dispositivo se encarga de un tipo específico de dispositivos (por ejemplo, una unidad de disco, una tarjeta de sonido o un monitor). La CPU y los controladores de dispositivo pueden ejecutarse concurrentemente, compitiendo así por el acceso a la memoria. La memoria caché reduce el acceso a la memoria, ya que la CPU necesita acceder a la memoria compartida un número de veces menor.

Bases de datos Distribuidas

Los sistemas de bases de datos distribuidas son un caso particular de los sistemas de cómputo distribuido en los cuales un conjunto de elementos de procesamiento autónomos (no necesariamente homogéneos) se interconectan por una red de comunicaciones y cooperan entre ellos para realizar sus tareas asignadas. Entre los términos más comunes que se utilizan para referirse al cómputo distribuido podemos encontrar: funciones distribuidas, procesamiento distribuido de datos, multiprocesadores, multicomputadores, procesamiento satelital, procesamiento tipo "backend", computadoras dedicadas y de propósito específico, sistemas de tiempo compartido, sistemas funcionalmente modulares.

En el cómputo distribuido existen muchos componentes para realizar una tarea. Los elementos que se pueden distribuir son:

Control. Las actividades relacionadas con el manejo o administración del sistema.

Datos. La información que maneja el sistema.

Funciones. Las actividades que cada elemento del sistema realiza.

Procesamiento lógico. Las tareas específicas involucradas en una actividad de procesamiento de información.

SGBD libres

– PostgreSQL (www.postgresql.org) Licencia Libre  
– MySQL (www.mysql.com) Licencia Dual  
– Firebird (www.firebirdsql.org) Initial Developer Public License  
– SQLite (ww.sqlite.org) Licencia Dominio Público  
– DB2 Express-C (www.ibm.com/software/data/db2)  
– Apache Derby (http://db.apache.org/derby/)

SGBD gratuitos

– Microsoft SQL Server Compact Edition  
– Sybase ASE Express Edition para Linux

SGBD comerciales

– dBase  
– Fox Pro  
– IBM DB2 Universal Database (DB2 UDB)  
– IBM Informix  
– Microsoft Access  
– Microsoft SQL Server  
– Oracle  
– Paradox  
– Sybase ASE  
– Sybase ASA  
– Sybase IQ

8. Las leyes de Codd para el Funcionamiento de SGBD relacionales y establece una relación entre cada una de las leyes de codd

Ley 0 y Ley 2

Regla 0: El sistema debe ser relacional, base de datos y administrador de sistema. Ese sistema debe utilizar sus facilidades relacionales (exclusivamente) para manejar la base de datos.

Relación. Facilidad de acceso a los datos mediante claves primarias

Regla 2: Regla de acceso garantizado: Todos los valores son accesibles mediante Nombre de tabla + Clave primaria + Nombre de campo: Esta regla es esencialmente una nueva exposición de lo que se precisa fundamentalmente para las claves primarias.

Ley 1 y Ley 3

Regla 1: La regla de la información, toda la información en la base de datos es representada unidireccionalmente, por valores en posiciones de las columnas dentro de filas de tablas. Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera: con valores en tablas.

Relación. Un valor posible en una tabla es un valor NULO

Regla 3: Tratamiento sistemático de valores nulos, el sistema de gestión de base de datos debe permitir que haya campos nulos. Debe tener una representación de la “información que falta y de la información inaplicable” que es sistemática, distinto de todos los valores regulares

Ley 4 - 1

Regla 4: Catálogo dinámico en línea basado en el modelo relacional, el sistema debe soportar un catálogo en línea, el catálogo relacional debe ser accesible a los usuarios autorizados. Es decir, los usuarios deben poder tener acceso a la estructura de la base de datos (catálogo).

Relación. La regla 4 es una consecuencia de la ley 1. Los metadatos se almacenan usando el modelo relacional

Regla 1: La regla de la información, toda la información en la base de datos es representada unidireccionalmente, por valores en posiciones de las columnas dentro de filas de tablas. Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera: con valores en tablas.

Ley 5 y 7

Regla 5: La regla comprensiva del sublenguaje de los datos, el sistema debe soportar por lo menos un lenguaje relacional que:

Tenga una sintaxis lineal.

Puede ser utilizado de manera interactiva.

Soporte operaciones de definición de datos, operaciones de manipulación de datos (actualización así como la recuperación), seguridad e integridad y operaciones de administración de transacciones.

Relación. Soporte del sistema - Alto nivel de actualización “Sublenguaje”

Regla 7: Alto nivel de inserción, actualización, y cancelación, el sistema debe soportar suministrar datos en el mismo tiempo que se inserta, actualiza o esté borrando. Esto significa que los datos se pueden recuperar de una base de datos relacional en los sistemas construidos de datos de filas múltiples y/o de tablas múltiples.

Ley 6 y 11

Regla 6: Regla de actualización, todas las vistas que son teóricamente actualizables deben ser actualizables por el sistema.

Relación. Transparencia al usuario

Regla 11: Independencia de la distribución, la distribución de las porciones de la base de datos a las varias localizaciones debe ser invisible a los usuarios de la base de datos. Los usos existentes deben continuar funcionando con éxito:

Cuando una versión distribuida del SGBD se introdujo por primera vez

Cuando se distribuyen los datos existentes se distribuyen en todo el sistema.

Ley 8 y 12

Regla 8: Independencia física de los datos, los programas de aplicación y actividades del terminal permanecen inalterados a nivel lógico cuando quiera que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

Relación. Así se viole la regla de la no subversión y se realicen cambios estos no afectan el nivel lógico de la base de datos gracias a la dependencia física de los datos

– Regla 12: La regla de la no subversión, si el sistema proporciona una interfaz de bajo nivel de registro, a parte de una interfaz relacional, que esa interfaz de bajo nivel no se pueda utilizar para subvertir el sistema, por ejemplo: sin pasar por seguridad relacional o limitación de integridad. Esto es debido a que existen sistemas anteriormente no relacionales que añadieron una interfaz relacional, pero con la interfaz nativa existe la posibilidad de trabajar no relacionalmente.

Ley 9 y 10

– Regla 9: independencia lógica de los datos, los cambios al nivel lógico (tablas, columnas, filas, etc.) no deben requerir un cambio a una solicitud basada en la estructura. La independencia de datos lógica es más difícil de lograr que la independencia física de datos.

Relación. Por más cambios que tenga la estructura los datos no van a cambiar, al igual que si se cambia las limitaciones de integridad estas no deben afectar las aplicaciones

Regla 10: independencia de la integridad, las limitaciones de la integridad se deben especificar por separado de los programas de la aplicación y se almacenan en la base de datos. Debe ser posible cambiar esas limitaciones sin afectar innecesariamente las aplicaciones existentes.

9. Qué es SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés StructuredQueryLanguage) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones.

Revisión del lenguaje sql a través del tiempo

En 1986, el ANSI adoptó SQL (sustancialmente adoptó el dialecto SQL de IBM) como estándar para los lenguajes relacionales y en 1987 se transformó en estándar ISO. Esta versión del estándar va con el nombre de SQL/86. En los años siguientes, éste ha sufrido diversas revisiones que han conducido primero a la versión SQL/89 y, posteriormente, a la actual SQL/92.

El hecho de tener un estándar definido por un lenguaje para bases de datos relacionales abre potencialmente el camino a la intercomunicabilidad entre todos los productos que se basan en él. Desde el punto de vista práctico, por desgracia las cosas fueron de otro modo. Efectivamente, en general cada productor adopta e implementa en la propia base de datos sólo el corazón del lenguaje SQL (el así llamado Entrylevel o al máximo el Intermediatelevel), extendiéndose de manera individual según la propia visión que cada cual tenga del mundo de las bases de datos.

Actualmente, está en marcha un proceso de revisión del lenguaje por parte de los comités ANSI e ISO, que debería terminar en la definición de lo que en este momento se conoce como SQL3. Las características principales de esta nueva encarnación de SQL deberían ser su transformación en un lenguaje stand-alone (mientras ahora se usa como lenguaje hospedado en otros lenguajes) y la introducción de nuevos tipos de datos más complejos que permitan, por ejemplo, el tratamiento de datos multimediales.

SQL Injection

Inyección directa de comandos SQL o SQL injection) Técnica utilizada por personas maliciosas con el fin de alterar o atacar un sitio o servidor a través de comandos SQL.

Las inyecciones utilizan información de entrada del usuario combinado con comandos SQL para construir una consulta SQL maliciosa. En otras palabras, se "inyecta" un código SQL malicioso para alterar el funcionamiento normal de las consultas SQL programadas por los diseñadores/webmasters. Al no haber seguridad, el código se ejecuta con consecuencias alarmantes.

Con estas inyecciones se pueden obtener datos escondidos, eliminar o sobrescribir datos en la base de datos y hasta lograr ejecutar comandos peligrosos en la máquina donde está la base de datos.

El hecho de que un servidor pueda verse afectado por las inyecciones SQL se debe a la falta de medidas de seguridad por parte de sus diseñadores/programadores, especialmente por una mala filtración de las entradas (por formularios, cookies o parámetros).

Porque un administrador debe protegerse frente a él.

Los ataques cibernéticos son el pan de cada dia y sus daños cuando se realizan son irreparables cuando hablamos de bases de datos es información importante la cual no se puede perder ni secuestrar..esto no se puede permitir ni por prestigio propio ni de la empresa

10. Defina los siguientes terminos

Lenguaje DML

Lenguaje de Manipulación de Datos

Lenguaje de Manipulación de Datos (Data ManipulationLanguage, DML) es un idioma proporcionado por los sistemas gestores de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos. El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.

Lenguaje DDL

Lenguaje de definición de datos (DDL: Data DefinitionLanguage):Sencillo lenguaje artificial para definir y describir los objetos de la base de datos, su estructura, relaciones y restricciones. En la práctica puede consistir en un subconjunto de instrucciones de otro lenguaje informático. Aparte suele poseer dos subconjuntos de instrucciones:

Lenguaje de definición del almacenamiento de los datos (DSDL: Data Storage DefinitionLanguage): permite especificar características físicas de la base de datos (volúmenes y archivos donde van a ser almacenados los datos, etc).

Lenguaje de control de datos (DCL: Data Control Language): encargado del control y seguridad de los datos (privilegios y modos de acceso, etc).

Lenguaje DCL

DCL (Data Control Language)

Es la parte menos conocida de deSql , siendo la finalidad controlar el acceso a datos denegando y otorgando privilegios sobre los objetos existentes.

COMMIT : Guarda los trabajos realizados en las transacciones

ROLLBACK : Restaura la base de datos al estado original desde el comando COMMIT pasado en las transacciones

SAVEPOINT : establecer un punto en que es posible un ROLLBACK.

SAVE TRANSACTION : Establece un punto de almacenamiento dentro de una transacción

Lenguaje TCL

Tcl/Tk es un lenguaje de script con soporte para desarrollar interfaces gráficas de usuario. Sencillo de aprender permite desarrollar sofisticadas interfaces de aspecto profesional en pocas horas. Es el lenguaje adecuado para escribir scripts que necesiten interfaz gráfica. En este artículo se resumen los conceptos básicos necesarios para comenzar a desarrollar aplicaciones en Tcl/Tk.

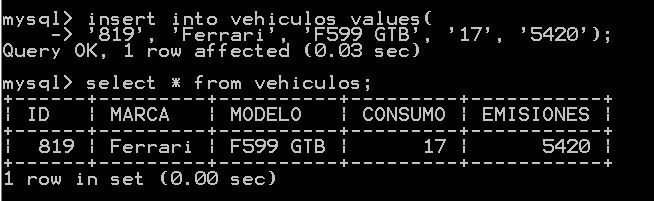
11.  
a.



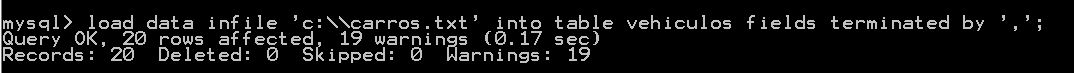
b.



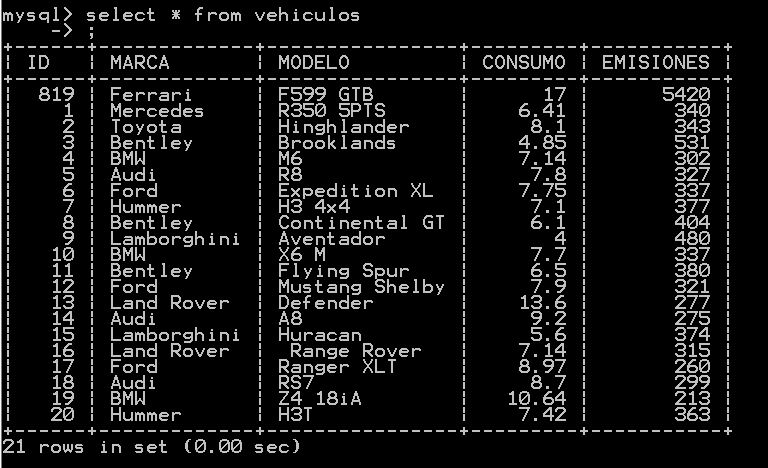
c.



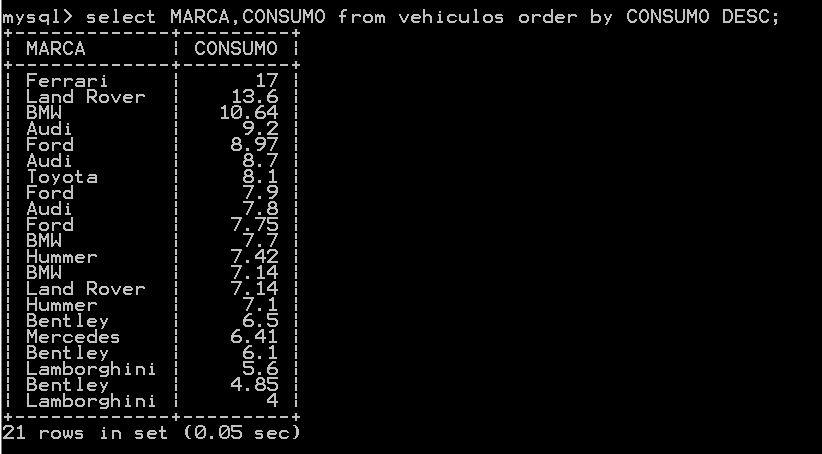
d.



e.



f.



12.  
Es una representación abstracta de los datos de una organización y las relaciones entre ellos. Más aun, podemos decir que, en cierta medida, un modelado de datos describe una organización

El propósito de un modelado de datos es, por una parte, representar los datos y, por otra, ser comprensible

Un modelo de datos es por tanto una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. Conceptualmente, una aplicación puede ser caracterizada por:

Propiedades estáticas: entidades (u objetos), propiedades (o atributos)12 de esas entidades, y relaciones entre esas entidades.

Propiedades dinámicas: operaciones sobre entidades, sobre propiedades o relaciones entre operaciones.

Reglas de integridad sobre las entidades y las operaciones (por ejemplo, transacciones).

13.  
Modelo conceptual:

En el diseño conceptual se hace una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD (Sistema Gestor de Bases de Datos) que se vaya a utilizar para manipularla. Su objetivo es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar dicha información.

Modelo Lógico:

Recibe como entrada el esquema conceptual y da como resultado un esquema lógico para describir datos en el nivel conceptual y de visión. Se caracterizan porque proporcionan capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Los más conocidos son: el orientado a objetos, el binario, el semántico de datos, el modelo lógico y el modelo funcional de datos.

Modelo Físico

Recibe como entrada el esquema lógico y da como resultado un esquema físico, que es una descripción de la implementación de una base de datos en la memoria secundaria, describe las estructuras de almacenamiento y los métodos usados para tener un acceso efectivo a los datos.

Modelos Físicos: Modelo Unificador, Memoria de Elementos.

14.  
a. Una entidad caracteriza a un tipo de objeto, real o abstracto, del problema a modelizar. Toda entidad tiene existencia propia, es distinguible del resto de las entidades, tiene nombre y posee atributos definidos en un dominio determinado. Una entidad es todo aquello de lo que se desea almacenar información. En el diagrama E-R las entidades se representan mediante rectángulos.

Una entidad fuerte existe por sí misma sin depender la existencia de alguna otra entidad. Por el contrario la existencia de una instancia de una entidad débil depende de la existencia previa de otra entidad. Si la entidad débil puede ser identificada sin necesidad de identificar previamente la entidad de cuya existencia depende, diremos que la entidad débil lo es por existencia únicamente. Si la entidad débil no puede ser identificada independientemente, sino que previamente es necesario identificar a la entidad de cuya existencia depende, diremos que la entidad débil lo es por identificación.

b. Una relación es una asociación lógica entre diferentes entidades que describe alguna interacción entre las mismas

Relaciones Unarias

Las relaciones en las que sólo participan una entidad se les denomina unarias, de anillo o grado 1; Relaciona una entidad consigo misma por lo que también se les llaman relaciones reflexivas

Ejemplo: un empleados puede ser jefes de otros empleados

Relaciones Binarias

Las relaciones que en las que participan dos entidades son binarias o de grado dos.

Ejemplo: En un aula se encuentra un grupo, y en un grupo se encuentran alumnos

Relaciones Ternarias

Cuando en la relación participan tres entidades serán ternarias o de grado tres.

Ejemplo: En una aula se encuentra un grupo y un maestro y a la inversa

c. Participación es la dependencia o asociación entre los conjuntos de entidades

(0,1) Mínimo Cero, máximo uno  
(1,1) Mínimo uno, máximo uno  
(0,n) mínimo cero, máximo n muchos  
(1,n) mínimo uno, máximo n muchos

d. La cardinalidad de la interrelación es el número de entidades con las que se puede relacionar otra entidad bajo la interrelación

Uno a uno: Una entidad se relaciona únicamente con otra y viceversa. Por ejemplo, si tuviésemos una entidad con distintos chasis y otra con matrículas deberíamos de determinar que cada chasis solo puede tener una matrícula (y cada matrícula un chasis, ni más en ningún caso).

Uno a varios o varios a uno: determina que un registro de una entidad puede estar relacionado con varios de otra entidad, pero en esta entidad existir solo una vez. Como ha sido en el caso anterior del trabajador del taller.

Varios a varios: determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa. Por ejemplo, en el taller un coche puede ser reparado por varios mecánicos distintos y esos mecánicos pueden reparar varios coches distintos.

e. Los atributos de una entidad representan la definición de propiedades, características o cualidades de un tipo de entidad o de una interrelación.

Atributo Obligatorio:  
Corresponden a todos los datos que se requieren conocer dentro del contexto del sistema, como puede ser el nombre, el apellido, la dirección y el teléfono de una persona en el caso de un formulario de contacto

Atributo Opcional:  
Corresponden a aquellos datos que podrían llegar a requerirse a futuro o que pueden no tener siempre un valor de instancia requerido, es decir el almacenamiento de estos atributos no influye en el funcionamiento del sistema

Atributo Compuesto:   
Corresponde a grupos de atributos que  
Tienen afinidad en cuanto a su significado o a su uso.  
Ejemplo: Dirección = calle + número + ciudad

Atributo multivaluado: (o multivalorado)   
Se dice del atributo tal que para una misma entidad puede tomar varios valores diferentes, es decir, varios valores del mismo dominio.

Por ejemplo, cada persona puede ser localizada en varios números de teléfono. Considerando el teléfono de contacto como un atributo de persona, podemos afirmar que tal atributo es multivaluado.

Atributo univaluado  
Es aquel que tiene un único valor para cada ocurrencia de una entidad.

El atributo sueldo de la entidad empleado, por ejemplo, toma valores del dominio de los reales y únicamente toma un valor para cada empleado concreto; por lo tanto, ningún empleado puede tener más de un valor para el sueldo.

Atributo derivado  
Atributo cuyo valor puede derivarse de los valores de otros atributos o entidades relacionados

f. Un dominio describe un conjunto de posibles valores para cierto atributo. Como un dominio restringe los valores del atributo, puede ser considerado como una restricción. Matemáticamente, atribuir un dominio a un atributo significa "todos los valores de este atributo deben de ser elementos del conjunto especificado".

Cadenas de caracteres: es una secuencia ordenada de longitud arbitraria (aunque finita) de elementos que pertenecen a un cierto alfabeto. En general, una cadena de caracteres es una sucesión de caracteres

Fecha: diferentes tipos de fecha y sus formatos como el año, mes, el día, y la hora.

Número Entero: son los valores que no tienen punto decimal, pueden ser positivos o negativos y el cero.

Número Real: son los valores que tienen punto decimal, pueden ser positivos o negativos

15.  
a. (1:1)  
b. (1:N)  
c. (N:M)  
d. (1:N)  
e. (N:M)  
f. (N:1)  
g. (1:N)  
h. (1:N)

16.

a. Fecha de nacimiento: Univaluado, ya que toma un solo valor para cada persona y no cambia con el tiempo

b. Lugar de nacimiento: Opcional, ya que puede llegar a requerirse en un futuro pero no influye en el funcionamiento del sistema

c. Edad: derivado, ya que puede derivarse del valor del atributo fecha de nacimiento y el valor de la fecha actual

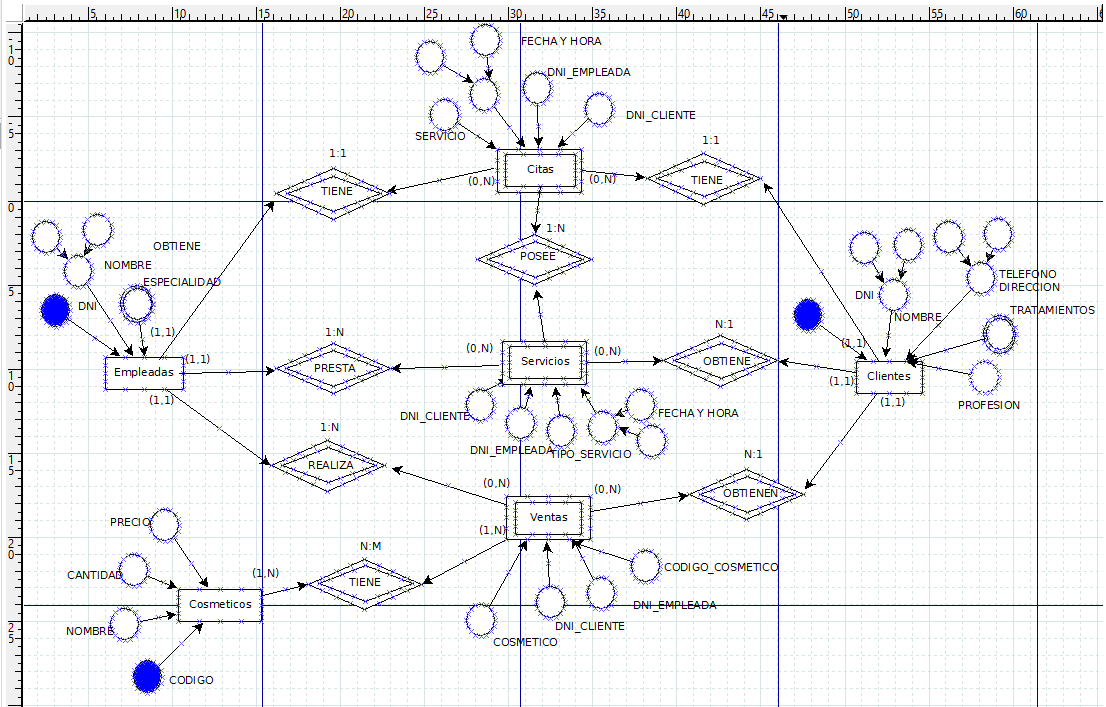
d. EsMayorDeEdad: derivado, ya que puede calcular según el valor del atributo edad

e. DNI: Obligatorio, ya que es un dato que se requiere para identificar a cada persona

f. Teléfonos: Multivaluado, ya que una persona puede tener varios teléfonos

g. Apellidos: Compuestos, ya que está formado por más de un valor

Apellido paterno + Apellido materno

17.