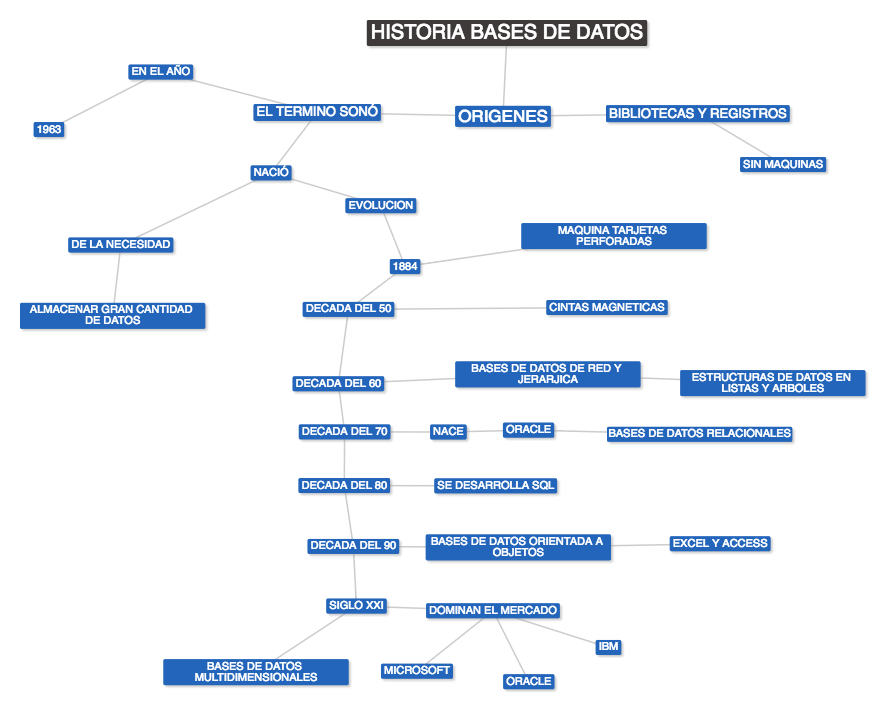
1.

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono.

2. 

3.

**a) Dato:** desde el punto de vista de las bases de datos, podemos decir que un dato es una información que refleja el valor de una característica de un objeto real, sea concreto o abstracto, o imaginario.

Debe cumplir algunas condiciones, por ejemplo, debe permanecer en el tiempo. En ese sentido, extrictamente hablando, una edad no es un dato, ya que varía con el tiempo. El dato sería la fecha de nacimiento, y la edad se calcula a partir de ese dato y de la fecha actual. Además, debe tener un significado, y debe ser manipulable mediante operadores: comparaciones, sumas, restas, etc (por supuesto, no todos los datos admiten todos los operadores).

**b) Tipo de dato:** El tipo de un dato es el conjunto de valores que puede tomar durante el programa. Si se le intenta dar un valor fuera del conjunto se producirá un error. Los tipos de datos que puede haber en un campo, se pueden agrupar en tres grandes grupos: Tipos numéricos: INTEGER - Tipos de Fecha: DATE - Tipos de Cadena: VARCHAR

La asignación de tipos a los datos tiene dos objetivos principales:

Por un lado, detectar errores en las operaciones

Por el otro, determinar cómo ejecutar estas operaciones

**c) Campo:** Un campo es el nombre de la unidad de información. Cada entrada en una base de datos puede tener múltiples campos de diversos tipos. Por ejemplo, un campo de texto llamado 'color favorito', que permite escribirlo, o un menú llamado 'población' que permita escoger de un listado de poblaciones posibles. La combinación de campos diversos nos permitirá recabar toda la información que consideremos relevante sobre los ítems que constituyen la base de datos.

**d) Registro:** En informática, o concretamente en el contexto de una base de datos relacional, un 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. Se le asigna automáticamente un número consecutivo (número de registro) que en ocasiones es usado como índice aunque lo normal y práctico es asignarle a cada registro un campo clave para su búsqueda.

**e) Campo clave:** En cualquier base de datos los registros incluidos en sus diferentes tablas deben estar perfectamente identificados y de esto se encargan las claves o llaves. Trasladando este concepto a la vida real, cada ciudadano tiene un número de DNI, puede haber dos personas con igual nombre e incluso apellidos iguales, pero ambos se diferenciarán por su número de DNI, que es único en "teoría".

Cada tabla debe contener al menos un campo que permita identificar unívocamente cada registro. Este campo puede ser real, como el propio DNI de una persona. O puede ser un valor que nos inventemos para conseguir el mismo fin, como un número que asignemos automáticamente a cada registro, asegurándonos (realmente lo hará Base) de que no se repita. Otra opción es utilizar una combinación de campos para identificar los registros de la tabla. Por ejemplo podríamos definir la combinación de Nombre y Apellidos de nuestra tabla para este fin. Sin embargo, existe la probabilidad de que haya alumnos con igual nombre y apellidos, por lo que no sería una clave del todo válida.

**f) Tabla:** Las tablas de una base de datos conforman la estructura principal de la misma, podríamos decir que se trata de los cimientos de nuestra estructura. Cada tabla contiene la información que utilizarán el resto de objetos de la base de datos: consultas, formularios, informes.

**g) Consulta:** Es una petición al SGBD para que procese un determinado comando SQL. Esto incluye tanto peticiones de datos como creación de bases de datos, tablas, modificaciones, inserciones, etc.

**h) Indice:** Un índice es una estructura de datos definida sobre una columna de tabla (o varias) y que permite localizar de forma rápida las filas de la tabla en base a su contenido en la columna indexada además de permitir recuperar las filas de la tabla ordenadas por esa misma columna.

Funciona de forma parecida al índice de un libro donde tenemos el título del capítulo y la página donde empieza dicho capítulo, en un índice definido sobre una determinada columna tenemos el contenido de la columna y la posición de la fila que contiene dicho valor dentro de la tabla.

La definición de los índices de la base de datos es tarea del administrador de la base de datos, los administradores más experimentados pueden diseñar un buen conjunto de índices, pero esta tarea es muy compleja, consume mucho tiempo y está sujeta a errores, incluso con cargas de trabajo y bases de datos con un grado de complejidad no excesivo.

**Índice simple y compuesto**

Un índice simple está definido sobre una sóla columna de la tabla mientras que un índice compuesto está formado por varias columnas de la misma tabla (tabla sobre la cual está definido el índice.

Cuando se define un índice sobre una columna, los registros que se recuperen utilizando el índice aparecerán ordenados por el campo indexado. Si se define un índice compuesto por las columnas col1 y col2, las filas que se recuperen utilizando dicho índice aparecerán ordenadas por los valores de col1 y todas las filas que tengan el mismo valor de col1 se ordenarán a su vez por los valores contenidos en col2, función igual que la cláusula ORDER BY vista en el tema de consultas simples.

Por ejemplo si definimos un índice compuesto basado en las columnas (provincia, localidad), las filas que se recuperen utilizando este índice aparecerán ordenadas por provincia y dentro de la misma provincia por localidad.

**Índice agrupado y no agrupado**

El término índice agrupado no se debe confundir con índice compuesto, el significado es totalmente diferente.

Un índice agrupado (CLUSTERED) es un índice en el que el orden lógico de los valores de clave determina el orden físico de las filas correspondientes de la tabla. El nivel inferior, u hoja, de un índice agrupado contiene las filas de datos en sí de la tabla. Una tabla o vista permite un solo índice agrupado al mismo tiempo.

Los índices no agrupados existentes en las tablas se vuelven a generar al crear un índice agrupado, por lo que es conveniente crear el índice agrupado antes de crear los índices no agrupados.

Un índice no agrupado especifica la ordenación lógica de la tabla. Con un índice no agrupado, el orden físico de las filas de datos es independiente del orden indizado.

**Índice único**

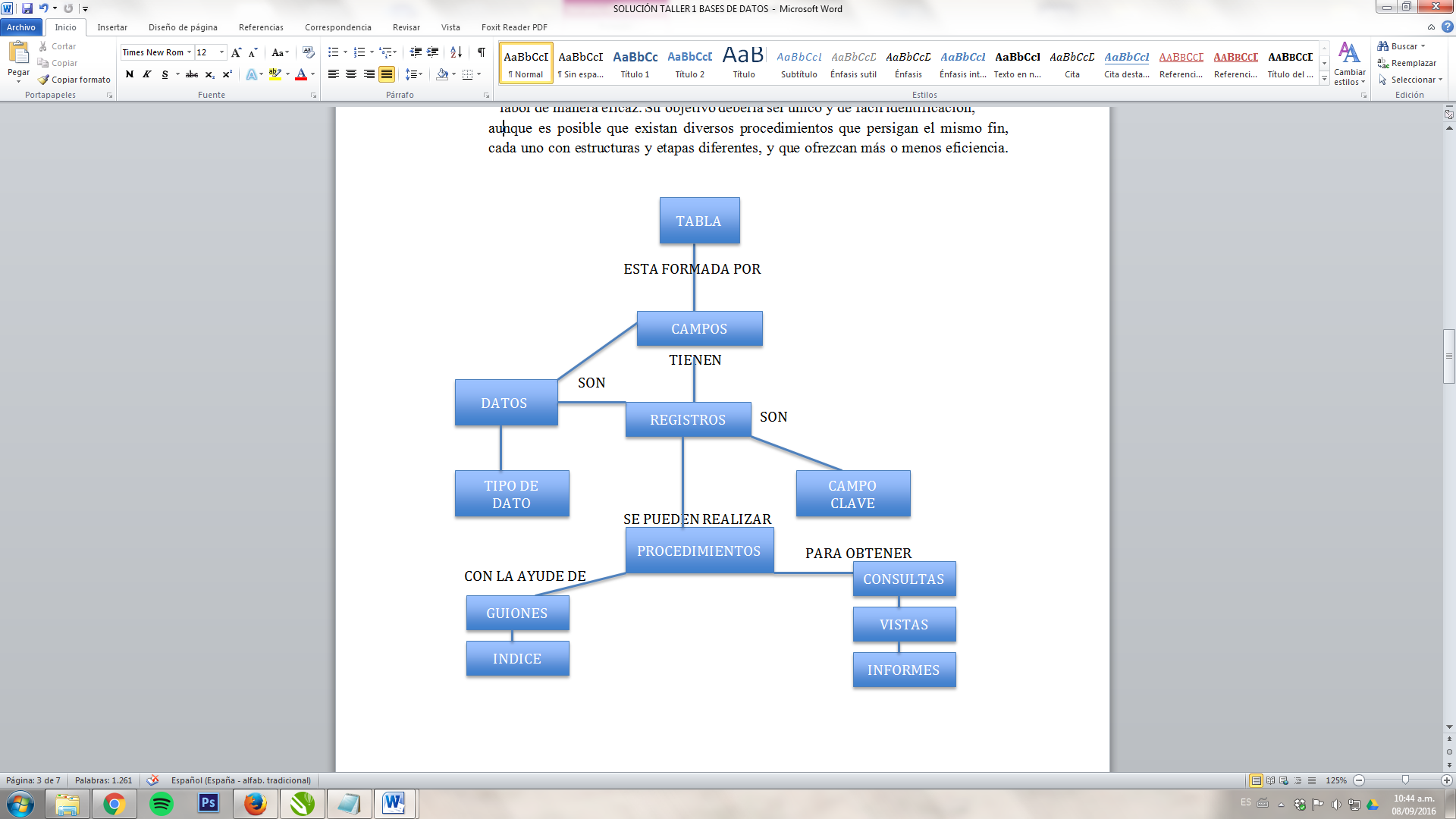
Índice único es aquel en el que no se permite que dos filas tengan el mismo valor en la columna de clave del índice. Es decir que no permite valores duplicados.

**i) Vista:** una vista se puede considerar una tabla virtual o una consulta almacenada que devuelve un conjunto de resultados y a la que se le pone un nombre pero no almacena datos físicamente y se consulta igual que una tabla.

**j) Informe:** Los informes tienen como objetivo proporcionar las herramientas necesarias para obtener una copia impresa de los datos existentes en una base de datos aunque existen otras posibilidades tan interesantes como la generación de archivos en formato PDF. Habitualmente, los informes se suelen construir a partir de los resultados obtenidos de la ejecución de consultas. De esta forma combinamos la posibilidad de seleccionar sólo los datos que deseemos que nos ofrecen las consultas con la ventaja de imprimirlos que aportan los informes.

**k) Guiones:** En informática, un **script**(guion) es un conjunto de instrucciones. Permiten la automatización de tareas, creando pequeñas utilidades.

**l) Procedimientos:** consiste en seguir ciertos pasos predefinidos para desarrollar una labor de manera eficaz. Su objetivo debería ser único y de fácil identificación, aunque es posible que existan diversos procedimientos que persigan el mismo fin, cada uno con estructuras y etapas diferentes, y que ofrezcan más o menos eficiencia.



4.

1. **World Data Centre for Climate** El WDCC (Centro Mundial de datos para el clima) es la base de datos más grande del mundo. Almacena unos 220 terabytes de información y 6 petabytes de información adicional, incluyendo datos sobre el clima, predicciones y simulaciones.

2. **National Energy Research Scientific Computing Center** El NERSC se dedica a investigar sobre distintos tipos de energía. Su base de datos tiene 2.8 Petabytes.

3. **AT&T** Se trata de una compañía de telecomunicaciones que almacena 323 terabytes de información.

4. **Google** Aunque se desconoce el verdadero tamaño de su base de datos, sí se puede estimar. La compañía recibe unos 91 millones de consultas al día, consultas que son almacenadas por la compañía. Se supone que almacena cientos de terabytes de información.

5. **Sprint** Con 53 millones de clientes, Sprint es una de las mayores compañías de telecomunicaciones del mundo. Guarda los detalles de 365 millones de llamadas al día.

6. **ChoicePoint** Con 250 terabytes de datos personales, el negocio de esta compañía es el de almacenar datos personales sobre la población de EEUU. Se calcula que almacena información de unas 250 millones de personas.

7. **YouTube** Aunque es de esperar que ascienda puesto rápidamente en este ranking, por el momento se encuentra en el séptimo lugar. Su base de datos (desconocida por ahora) crece unos 18.6 terabytes.

8. **Amazon** Se la considera la tienda online más grande del mundo, tanto en número de clientes como en los datos que almacena sobre ellos. Cuenta con 59 millones de clientes y más de 42 terabytes de datos.

9. **Central Intelligence Agency (CIA)** Aunque como en el caso de Google no se conoce el tamaño de su base de datos, la parte 'abierta' al público es inmensa. Se cree que posee datos de más de 250 países y entidades.

10. **Biblioteca del Congreso de EEUU** Aunque los datos que guarda no han sido digitalizados completamente, sí cuenta con gran cantidad de información. Posee más de 5 millones de documentos digitales.

5.

1.-Bases de datos estáticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones. Por ejemplo la empresa IBM se dedico al desarrollo de la computadora y a hecho grandes impactos en el área informática

2.-Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.

3.-Bases de datos bibliográficas

Solo contienen un subrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no, estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias —ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

4.-Bases de datos deductivas

Un sistema de base de datos deductiva, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. Las bases de datos deductivas son también llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática.

5.-Base de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

6.-Bases de datos transaccionales

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

7.-Bases de datos relacionales

Éste es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante “consultas” que ofrecen una amplia flexibilidad y poder para administrar la información.

8.-Bases de datos multidimensionales

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

9.-Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulación – Propiedad que permite ocultar la información al resto de los objetos, º impidiendo así accesos incorrectos o conflictos.

Herencia – Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

Polimorfismo – Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

10.-Bases de datos documentales

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesaurus es un sistema de índices optimizado para este tipo de bases de datos.

6.

a)

Edgar F. Codd (1923-2003)

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 Database Systems, 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.

b)

Larry Ellison

En 1970 Larry leyó un artículo que cambiaría su vida, era de un investigador de IBM sobre el almacenamiento de datos en grandes tablas. Eso permitiría procesar la información para ser utilizable. Ese fue el inicio de Oracle, en el 1983, se creó para satisfacer una necesidad de software para Permision Instrument. Larry y sus socios desarrollaron un lenguaje que lo llamarían SQL (lenguaje de consulta estructurada). Este lenguaje es el que hizo millonario a Larry.

Fue entonces cuando crearon el 1er producto que se llamaría Oracle y que permitiría a las empresas recuperar datos basándose en cualquier parámetro. Como 1er cliente ya tuvieron a la CIA que pagó 50.000$ por aquel software. La gran clave de todo esto es que comercializaron este lenguaje y software para empresas 3 años antes que IBM.

c)

Roger Kent Summit

La historia de Dialog está íntimamente ligada a la figura de Roger Kent Summit, considerado como el “padre‿ de la industria online, y que trabajó para la Lockheed Missiles and Space Company. En 1966, esta empresa obtuvo un contrato con la NASA (National Aeronautics and Space Administration) para desarrollar un software de recuperación que permitiera buscar de manera rápida y eficaz en sus bases de datos al que se le llamó llamado RECON (Remote Console). En 1969-1970, por encargo de la Atomic Energy Commission, comenzó a gestionar la colección de resúmenes de Nuclear Science.

En 1972, la Lockheed decide explotar su sistema de recuperación de información instalando dos grandes ordenadores que unidos a las principales redes de transmisión existentes (Telenet y Tymnet), dan servicio público tanto en USA como en Europa, convirtiéndose en el primer servicio comercial online del mundo. El sistema se denominó desde entonces Dialog, teniendo su sede central en Palo Alto (California). En agosto de 1988, Dialog es adquirido por Knight-Ridder Inc.; esta misma empresa compra en 1993 el servicio Data-Star, que era el distribuidor europeo más utilizado en 1985 y le estaba haciendo una gran competencia sobre todo en el mercado europeo. En 1995, Dialog Information Services pasa a denominarse Knight- Ridder Information Inc., que en 1997 es comprada por M.A.I.D. plc, pasando a denominarse la fusión de las dos compañías The Dialog Corporation; y ya por último, en Abril del 2000, The Thomson Corporation compra la División de Servicios de Información de The Dialog Corporation, entre los que se encuentran Dialog y Data-Sta r.

d)

Bill Gates

A principios de la década de 1970, la invención del microprocesador permitió abaratar y reducir el tamaño de las gigantescas computadoras existentes hasta entonces. Era un paso decisivo hacia un sueño largamente acariciado por muchas empresas punteras en el sector tecnológico: construir ordenadores de tamaño y precio razonable que permitiesen llevar la informática a todas las empresas y hogares. El primero en llegar podría iniciar un negocio sumamente lucrativo y de enorme potencial. Era impensable que una empresa como Microsoft, dedicada solamente al software (sistemas operativos y programas) pudiese jugar algún papel en esta carrera entre fabricantes de hardware, es decir, de máquinas.

Y así fue al principio: una competición entre fabricantes de ordenadores no demasiado honesta, pues hubo más de un plagio. A mediados de los años setenta, en un garaje atestado de latas de aceite y enseres domésticos, Steve Jobs y Stephen Wozniak diseñaron y construyeron una placa de circuitos de computadora, toda una muestra de innovación y de imaginación. Al principio tenían la intención de vender sólo la placa, pero pronto se convencieron de la conveniencia de montar una empresa, Apple, y vender ordenadores. En 1977 empezaron a comercializar la segunda versión de su computadora personal, el Apple II, que se vendía con un sistema operativo también creado por Apple: un hito histórico que marca el nacimiento de la informática personal.

e)

Michael Monty Widenius

en 1995, fundó la compañía MySQL junto a David Axmark, y comenzó a desarrollar el sistema libre de gestión de bases de datos más utilizado y conocido. Fue MySQL, precisamente, la que le catapultó a la fama. Y es que ya en 2002 el sistema era tan utilizado que el libro publicado por Michael Widenius, MySQL Reference Manual, se convirtió en un best-seller y un año después, en 2003, le era reconocida su labor con el premio al emprendedor finés del año.

En los años siguientes, se demostró que la apuesta de MySQL de hacer un software de gestión de bases de datos sencillo y ligero y, sobre todo, apostar por un modelo basado en software libre en el que la base del negocio no está en el cobro por licencias, era una apuesta ganadora. En 2008, Sun Microsystems, adquirió MySQL valorándola en 1.000 millones de dólares, refrendando el hecho de que el software libre es un modelo para la empresa capaz de generar riqueza tanto para las personas directamente involucradas en un proyecto como para la comunidad.

A raiz de la compra de MySQL por parte de Sun Microsystems, Michael Widenius se integra en Sun Microsystems, sin embargo, acostumbrado a dirigir su propio proyecto, abandona al gigante en 2009 para formar su nueva empresa, MariaDB, heredera de MySQL y en la que desde entonces Widenius hace uso de todo su conocimiento y la experiencia cuando se trata de aplicar modelos de software libre al mundo de la empresa.

7. sistemas gestores de bases de datos

a) Definición: Sistema Gestor de Base de Datos. Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: DataBase Management System) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos.

Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos.

b) La ejecución de las operaciones sobre la base de datos para luego proporcionarlos al usuario en función de su requerimiento se realiza de un modo eficiente y seguro. Sus características de un SGDB posibilitan el cumplimiento de una serie de **funciones**, que pueden agruparse de la siguiente manera:

**1. Definición de los datos:** El SGBD ha de poder definir todos los objetos de la base de datos partiendo de definiciones en versión fuente para convertirlas en la versión objeto.

**2. Manipulación de los datos:** El SGBD responde a las solicitudes del usuario para realizar operaciones de supresión, actualización, extracción, entre otras gestiones. El manejo de los datos ha de realizarse de forma rápida, según las peticiones realizadas por los usuarios, y permitir la modificación del esquema de la base de datos gracias a su independencia.

**3. Seguridad e integridad de los datos**: Además de registrar el uso de las bases de datos, ante cualquier petición, también aplicará las medidas de seguridad e integridad de los datos (adopta medidas garantizar su validez) previamente definidas. Un SGBD debe garantizar su seguridad frente a ataques o simplemente impedir su acceso a usuarios no autorizados por cualquier razón.

**4. Recuperación y restauración de los datos:** La recuperación y restauración de los datos ante un posible fallo es otra de las principales funciones de un SGBD. Su aplicación se realizará a través de un Plan de recuperación y restauración de los datos que sirva de respaldo.

c) La tipología de los SGBD es muy variada, en función del criterio que utilicemos para su clasificación. Agruparlos atendiendo al modelo de datos, número de usuarios o de sitios suele ser lo más habitual, si bien la tipología puede obedecer a otras muchas pautas, según convenga desde un determinado enfoque práctico:

Si atendemos al modelo de datos, los testores de bases de datos pueden ser:

Relacionales

EnRed

Jerárquicos

Orientados a objetos

Por su parte, es posible diferenciarlos según sean o no propietarios, en función de la licencia, de acuerdo con el número de usuarios monousuario o multiusuario y, por ejemplo, también gruparlos en centralizados y distribuidos, esta vez según el número de sitios.

d) MySQL

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes.

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras.

Microsoft SQL Server

SQL Server es un sistema gestor de base de datos relacionales producido por Microsoft. Es un sistema cliente/servidor que funciona como una extensión natural del sistema operativo Windows. Entre otras características proporciona integridad de datos, optimización de consultas, control de concurrencia y backup y recuperación.

Es relativamente fácil de administrar a través de la utilización de un entorno gráfico para casi todas las tareas de sistema y administración de bases de datos. Utiliza servicios del sistema operativo Windows para ofrecer nuevas capacidades o ampliar la base de datos, tales como enviar y recibir mensajes y gestionar la seguridad de la conexión. Es fácil de usar y proporciona funciones de almacenamiento de datos que sólo estaban disponibles en Oracle y otros sistemas gestores de bases de datos más caros.

PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, derivado de Postgres, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley. Es un gestor de bases de datos de código abierto, brinda un control de concurrencia multi-versión (MVCC por sus siglas en inglés) que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación.

Posee características significativas del motor de datos, entre las que se pueden incluir las subconsultas, los valores por defecto, las restricciones a valores en los campos (constraints) y los disparadores (triggers). Ofrece funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales.

El código fuente se encuentra disponible para todos sin costo alguno. Está disponible para 34 plataformas con la última versión estable.Es totalmente compatible con ACID (acrónimo de Atomicity, Consistency, Isolation and Durability; en español: Atomicidad, Consistencia, Aislamiento y Durabilidad).

Posee una integridad referencial e interfaces nativas para lenguajes como ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG; PYTHON y RUBY. Funciona en todos los sistemas operativos Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.

Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico.

8. 12 reglas de codd

Codd se percató de que existían bases de datos en el mercado las cuales decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar estas tablas literalmente normalizadas; entonces éste publicó 12 reglas que un verdadero sistema relacional debería tener aunque en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse “más relacional” cuanto más siga estas reglas.

–          **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.

–          **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.

–          **Regla 2:** la regla del acceso garantizado, todos los datos deben ser accesibles sin ambigüedad. Esta regla es esencialmente una nueva exposición del requisito fundamental para las llaves primarias. Dice que cada valor escalar individual en la base de datos debe ser lógicamente direccionable especificando el nombre de la tabla, la columna que lo contiene y la llave primaria.

–          **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.

–          **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).

–          **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.

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

–          **Regla 7:** alto nivel de inserción, actualización, y cancelación, el sistema debe soportar suministrar datos en el mismo tiempo que se inserte, 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.

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

–          **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.

–          **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.

–          **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 redistribuyen en todo el sistema.

–          **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.

**Implementación de las Reglas de Codd en los Gestores de Bases de Datos**

**ORACLE**

ORACLEdice cumplir las 12 reglas de Codd y se considera la base de datos relacional por excelencia, aquí muestro un extracto de cómo maneja ORACLE las 12 reglas.

|  |  |
| --- | --- |
| Regla 1 – La Información | Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico y exactamente de una manera – por los valores de las tablas. |
| Regla 2 – Acceso Garantizado | Todos y cada dato (valor atómico) en una base de datos relacional se garantiza que sea lógicamente accesibles recurriendo a una combinación de nombre de tabla, valor de clave principal, y el nombre de la columna. |
| Regla 3 – Tratamiento sistemático de valores nulos | Los valores nulos (distintos de la cadena de caracteres vacía de caracteres en blanco y distintos de los números cero u otro) son compatibles con completamente relacional DBMS para representar la información que falta y la información aplicable de manera sistemática. |
| Regla 4 – Catálogo dinámico en línea basado en el Modelo Relacional | La descripción de la base de datos está representado en el nivel lógico de la misma manera que los datos ordinarios, por lo que los usuarios autorizados pueden aplicar el mismo lenguaje relacional para su interrogatorio que se aplican a los datos normales. |
| Regla 5 – Sublenguaje de datos completa | Un sistema relacional puede soportar varios lenguajes y varios modos de uso de terminal (por ejemplo, el modo de relleno en los espacios en blanco). Sin embargo, debe haber al menos una lengua cuyas declaraciones son expresables, por una sintaxis bien definida, como cadenas de caracteres, que es integral en el apoyo de todos los siguientes elementos:-          Definición de datos  –          Ver definición  –          Manipulación de datos (interactivo y por el programa)  –          Restricciones de integridad  –          Autorización  –          Límites de transacción (begin, commit y rollback) |
| Regla 6 – Actualización de Vistas | Todas las vistas que son teóricamente actualizables también son actualizables por el sistema. |
| Regla 7 – Insert de Alto Nivel, Update y Delete | La capacidad de manejar una relación base o una relación derivada como un solo operando se aplica no sólo a la recuperación de los datos, sino también para la inserción, actualización y supresión de los datos. |
| Regla 8 – Independencia de Datos Físico | Los programas de aplicación y actividades del terminal permanecen lógicamente irreprochable cuando se realiza algún cambio en cualquiera de las representaciones de almacenamiento o métodos de acceso. |
| Regla 9 – Independencia lógica de datos | Los programas de aplicación y actividades del terminal permanecen lógicamente irreprochable cuando la información de preservación de cambios de toda naturaleza que discapacidad de permiso teóricamente se hacen a las tablas base. |
| Regla 10 – Independencia de Integridad | Las restricciones de integridad específicos para una base de datos relacional en particular deben ser definibles en los datos relacionales sub-lenguaje y almacenables en el catálogo, no en los programas de aplicación. |
| Regla 11 – Independencia Distribución | Un DBMS relacional tiene dependencia de distribución. |
| Regla 12 – No Subversión | Si un sistema relacional tiene un (solo registro a la vez) de bajo nivel de lenguaje, ese bajo nivel no puede ser utilizado para subvertir o pasar por alto las reglas de integridad y las limitaciones expresadas en el lenguaje relacional de alto nivel (varios registros a la vez). |

9. SQL

La sigla que se conoce como SQL corresponde a la expresión inglesa Structured Query Language (entendida en español como Lenguaje de Consulta Estructurado), la cual identifica a un tipo de lenguaje vinculado con la gestión de bases de datos de carácter relacional que permite la especificación de distintas clases de operaciones entre éstas. Gracias a la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla.

El científico Edgar Frank Codd (1923–2003) fue quien propuso un modelo relacional para las bases de datos y creó un sublenguaje para acceder a los datos a partir del cálculo de predicados. En base al trabajo de Codd, IBM (International Business Machines) definió el lenguaje conocido como Structured English Query Language (SEQUEL). El SEQUEL se considera el antecesor de SQL, un lenguaje de cuarta generación que se estandarizó en 1986. La versión más primitiva de SQL, por lo tanto, fue la que se bautizó como SQL-86 (también conocida como SQL1).

En esencia, el SQL es un lenguaje declarativo de alto nivel ya que, al manejar conjuntos de registros y no registros individuales, ofrece una elevada productividad en la codificación y en la orientación a objetos. Una sentencia de SQL puede resultar equivalente a más de un programa que emplee un lenguaje de bajo nivel.

Una base de datos, dicen los expertos, implica la coexistencia de múltiples tipos de lenguajes. El denominado Data Definition Language (también conocido como DDL) es aquél que permite modificar la estructura de los objetos contemplados por la base de datos por medio de cuatro operaciones básicas. SQL, por su parte, es un lenguaje que permite manipular datos (Data Manipulation Language o DML) que contribuye a la gestión de las bases de datos a través de consultas.

REVISIONES

AÑO NOMBRE ALIAS

1986 SQL-86 SQL-87 Primera publicación hecha por ANSI. Confirmada por ISO en 1987.

1989 SQL-89 Revisión menor.

1992 SQL-92 SQL2 Revisión mayor.

1999 SQL:1999 SQL2000 Se agregaron expresiones regulares, consultas recursivas (para relaciones jerárquicas), triggers y algunas características orientadas a objetos.

2003 SQL:2003 Introduce algunas características de XML, cambios en las funciones, estandarización del objeto sequence y de las columnas autonuméricas.

2005 SQL:2005 ISO/IEC 9075-14:2005 Define las maneras en las cuales SQL se puede utilizar conjuntamente con XML. Define maneras de importar y guardar datos XML en una base de datos SQL, manipulándolos dentro de la base de datos y publicando el XML y los datos SQL convencionales en forma XML. Además, proporciona facilidades que permiten a las aplicaciones integrar dentro de su código SQL el uso de XQuery, lenguaje de consulta XML publicado por el W3C (World Wide Web Consortium) para acceso concurrente a datos ordinarios SQL y documentos XML.

2008 SQL:2008 Permite el uso de la cláusula ORDER BY fuera de las definiciones de los cursores. Incluye los disparadores del tipo INSTEAD OF. Añade la sentencia TRUNCATE.

SQL INJECTION

SQL Injection es una vulnerabilidad que permite a un atacante realizar consultas a una base de datos, se vale de un incorrecto filtrado de la información que se pasa a través de los campos y/o variables que usa un sitio web, es por lo general usada para extraer credenciales y realizar accesos ilegítimos, práctica un tanto neófita, ya que un fallo de este tipo puede llegar a permitir ejecución de comandos en el servidor, subida y lectura de archivos, o peor aún, la alteración total de los datos almacenados.Una vulnerabilidad de tipo SQL Injection puede ser explotada tanto a través del método GET como del método POST. La práctica más común es hacerlo a través del primero, sin embargo hacerlo a través del método POST puede llegar a devolver los mismos resultados. En el mundo de la seguridad informática quienes descubren o investigan vulnerabilidades suelen hacer scripts que automatizan la explotación de las mismas, para el caso de SQL Injection las más comunes son:

Sqlmap: Tal vez la más famosa, desarrollada en python por Bernardo Damele y Miroslav Stampar.

Havij: Desarrollada por la empresa ITSecTeam.

SqlNinja: Desarrollada en Perl, usada para explotar aplicaciones web que usan como back-end a Microsoft Sql Server.

10. lenguaje SQL se divide en cuatro tipos de instrucciones de lenguaje primarios: DML, DDL, DCL y TCL. El uso de estas declaraciones, podemos definir la estructura de una base de datos mediante la creación y la modificación de los objetos de base de datos, y podemos manipular los datos de una tabla a través de actualizaciones o supresiones. También podemos controlar qué usuarios pueden leer / escribir datos o es propietario de transacciones para crear una única unidad de trabajo.

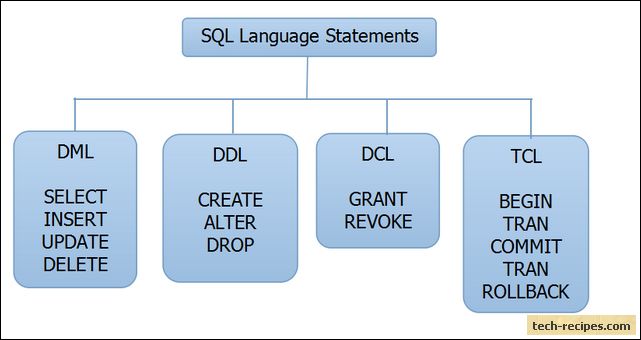
Las cuatro categorías principales de las sentencias SQL son los siguientes:

1. DML (Data Manipulation Language)

2. DDL (Data Definition Language)

3. DCL (Lenguaje de control de datos)

4. TCL (Lenguaje de control de transacciones)



DML (Data Manipulation Language)

DML afectan a los registros de una tabla. Estas son las operaciones básicas que llevamos a cabo en los datos como la selección de un par de registros de una tabla, insertar nuevos registros, eliminar registros innecesarios, y actualizar / modificar los registros existentes.

DML incluyen los siguientes:

SELECCIONAR - seleccione registros de una tabla

INSERTAR - insertar nuevos registros

ACTUALIZAR - actualizar / modificar los registros existentes

BORRAR - eliminar registros existentes

DDL (Data Definition Language)

instrucciones DDL se utilizan para alterar / modificar una estructura y un esquema de base de datos o tabla. Estas declaraciones manejan el diseño y el almacenamiento de objetos de base de datos.

CREAR - crear una nueva tabla, base de datos, esquema

ALTERAR - alterar la tabla existente, descripción de la columna

SOLTAR - eliminar los objetos existentes de la base de datos

DCL (Lenguaje de control de datos)

declaraciones DCL controlar el nivel de acceso que tienen los usuarios sobre los objetos de base de datos.

CONCEDER - permite a los usuarios leer / escribir sobre ciertos objetos de la base

REVOCAR - Mantiene a los usuarios de permiso de lectura / escritura en los objetos de base

TCL (Lenguaje de control de transacciones)

TCL declaraciones le permiten controlar y gestionar transacciones para mantener la integridad de los datos dentro de las sentencias SQL.

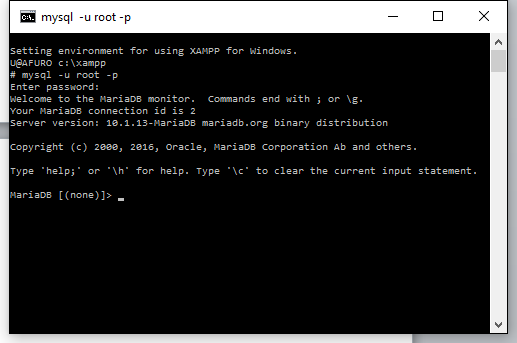
iniciar la transacción - se abre una transacción

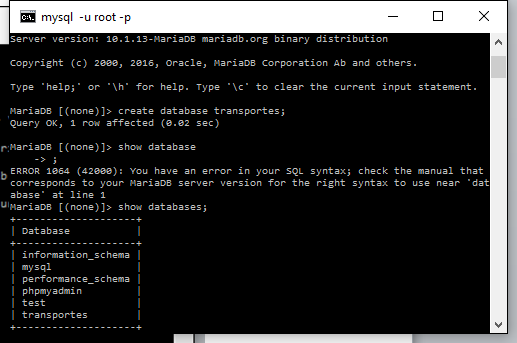
COMMIT Transacción - comete una transacción

Transacción ROLLBACK - ROLLBACK una transacción en caso de cualquier error.

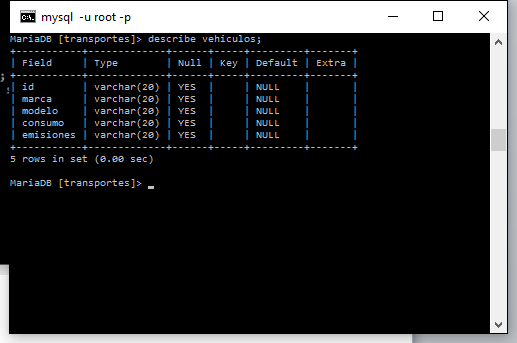
11.

a)

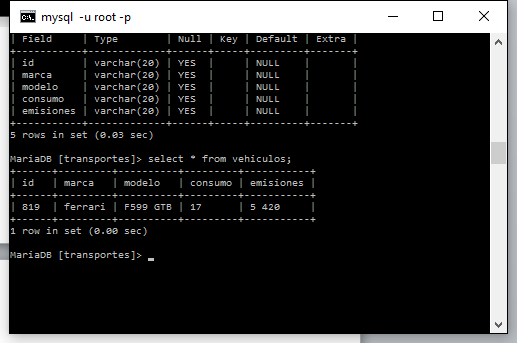
. 



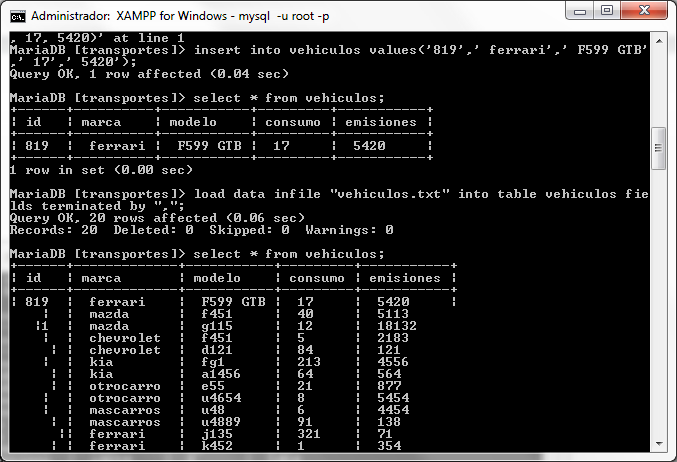
b)



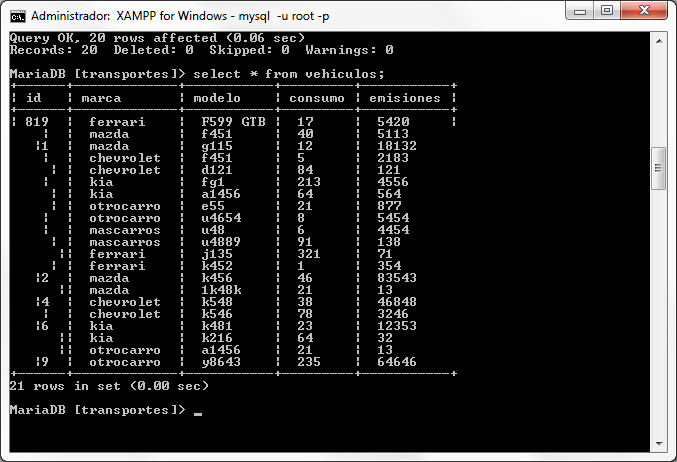
c)



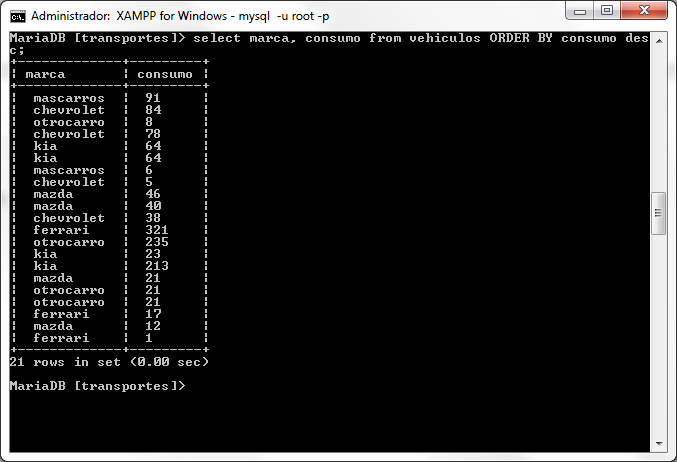
d)



e)



f) de esta parte me surge una incognita, solo está tomando como referencia para ordenar el primer carácter de consumo.



12. MODELADO DE DATOS

El modelado de datos es el proceso de documentar un diseño de sistema de software complejo como un diagrama de fácil comprensión, usando texto y símbolos para representar la forma en que los datos necesitan fluir. El diagrama se puede utilizar como un mapa para la construcción de un nuevo software o para la reingeniería de una aplicación antigua.

Tradicionalmente, los modelos de datos se han construido durante las fases de análisis y diseño de un proyecto, para asegurar que los requisitos para una nueva aplicación se entienden completamente. Un modelo de datos puede ser pensado como un diagrama de flujo que ilustra las relaciones entre los datos. A pesar de que la captura de todas las posibles relaciones en un modelo de datos puede consumir mucho tiempo, es un paso importante que no debería ser apresurado. Los modelos de datos físicos, lógicos y conceptuales bien documentados permiten que las partes interesadas identifiquen errores y hagan cambios antes de que cualquier código de programación se haya escrito.

13. Los **modeladores de datos** suelen utilizar varios modelos para ver los mismos datos y garantizar que todos los procesos, entidades, relaciones y flujos de datos han sido identificados. Hay varios enfoques diferentes para el modelado de datos, incluyendo:

**Modelado conceptual de datos:** Identifica las relaciones de más alto nivel entre diferentes entidades.

**Modelado de datos empresariales:** Similar al modelado de datos conceptuales, pero se dirige a los requisitos únicos de un negocio específico.

**Modelado lógico de datos:** Ilustra las entidades, atributos y relaciones específicas que participan en una función de negocios. Sirve como base para la creación del modelo de datos físico.

**Modelado de datos físicos:** Representa una aplicación e implementación específica de base de datos de un modelo de datos lógicos.

14. **ENTIDAD FUERTE:** Representa un objeto/elemento del Universo de discurso (del problema a resolver) que no necesita de otra para su identificación, es una entidad principal

**ENTIDAD DEBIL:** Representa un objeto/elemento del Universo de discurso (del problema a resolver) pero que necesita de una relación con otra entidad para su identificación.

**RELACIONES:** Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

El grado de la relación determina cuántas entidades participan de la relación

Binarias 2 entidades relacionadas

unarias (o recursivas) 1 entidad relacionada

n-arias (ternarias, etc.) 3 o más entidades relacionadas

**PARTICIPACION:** Participación en la relación indica si la existencia de una entidad requiere la existencia de la entidad asociada (mandatoria o total) o no es requerida en todos los casos (opcional o parcial).

(0,1 – 1,1 – 0,n – 1,n)

**CARDINALIDAD:** o conectividad de la relación especifica el número posibles instancias de la relación para cada entidad participante:

- 1:1, 1:N, N:N (en relaciones binarias)

- 1:1:1, 1:1:N, 1:N:N, N:N:N (en relaciones ternarias) (en relaciones ternarias)

**ATRIBUTOS:** Definen o identifican las características de entidad (es el contenido de esta entidad). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...).

**Obligatorio:** son el atributo único que tiene un registro en una entidad o lista el cual lo podrá seleccionar de entre todos los demás. Ejemplo, cedula, o placa de vehículo

**Opcionales**: Son campos no obligatorios. Ejemplo, correo electrónico.

**Simple:** son aquellos que no están divididos en sub partes. Ejemplo, una persona no puede tener más de una edad. O género, H o M

**Compuesto:** es cuando está formado por más de un atributo. Ejemplo, nombre, esta compuesto por nombre, apellido paterno y apellido materno.

**Univaluado:** Son los que tienen un solo valor. Ejemplo, la entidad persona solo puede tener una “edad”

**Multivaluado:** son aquellos que pueden representar varios valores simultáneamente para una misma entidad. Ejemplo, Una persona puede tener varios teléfonos.

**Almacenados:** son los atributos que cuyo valor guardan una cantidad que se utiliza para realizar cálculos con otros atributos o entidades.

**Derivados:** cuyo valor se puede derivar del valor de otros atributos. Ejemplo, se puede calcular a partir de la fecha de nacimiento y la fecha actual.

**Valor Nulo:** un atributo toma un valor nulo cuando una entidad no tiene un valor para un atributo

**DOMINIO DE ATRIBUTO:** Son los valores que son posible asignar a ese atributo para cada entidad

Ejemplo – dominio de EDAD en BD Compañía: 16-65.

15.

a) Hombre está casado con mujer en una sociedad monogamica. ( 1:1)

b) Hombre está casado con una mujer en una sociedad machista poligamica. (1:N)

c) Hombre está casado con una mujer, en una sociedad polígama liberal. (N:N)

d) Pescador pesca Pez. (1:N)

e) Arquitecto diseña casa. (1:N)

f) Piezas forman producto. ( N:1)

g) Turista viaja hotel. (N:1)

h) Jugador juega en equipo. (N:N)

16.

**ENTIDAD PERSONA:**

a) Fecha de nacimiento. univaluado

b) Lugar de nacimiento. univaluado

c) Edad. Derivado (de fecha nacimiento – fecha actual)

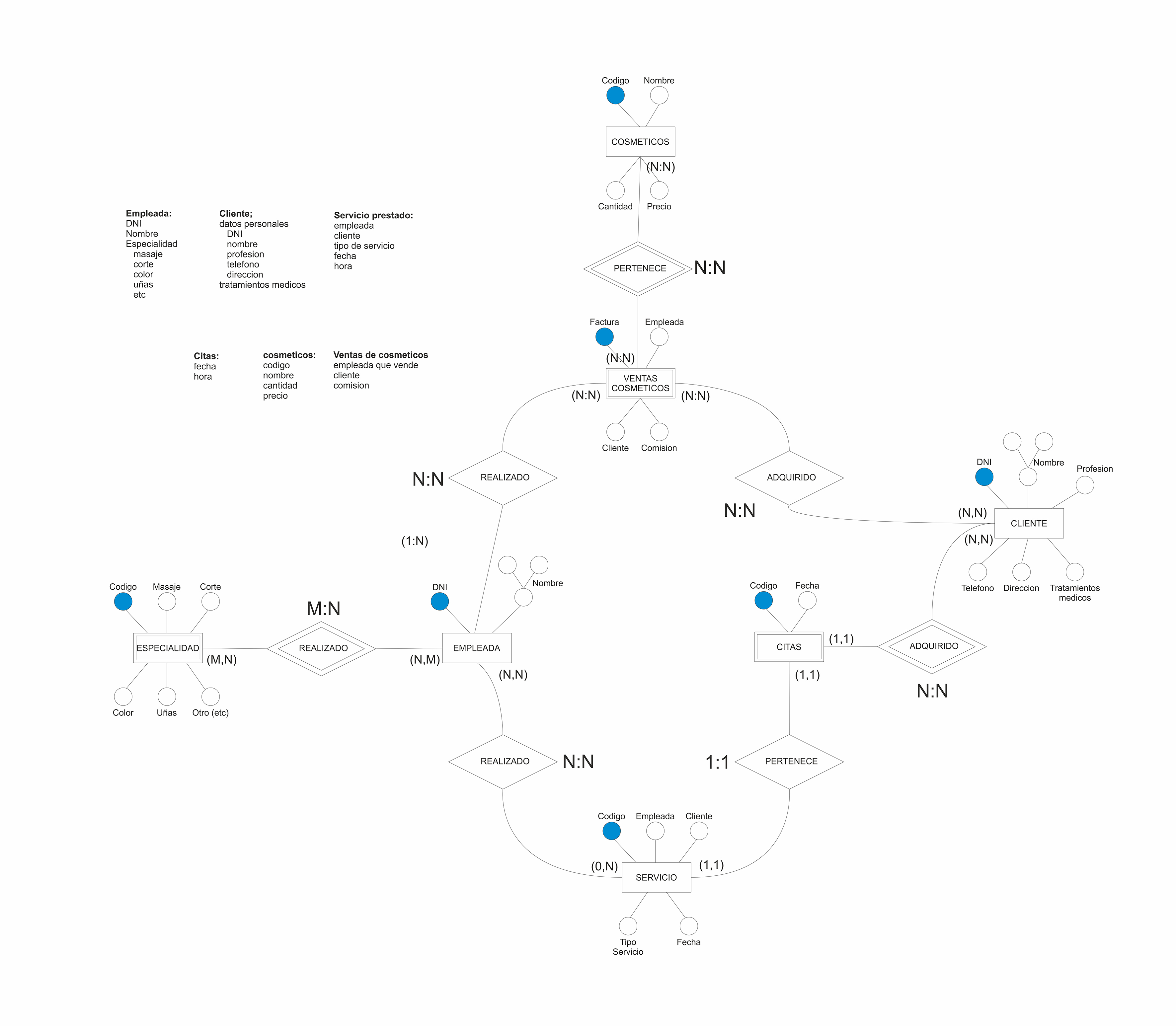
d) EsMayorDeEdad. Simple – derivado

e) DNI. Obligatorio o identificador principal

f) Telefonos. Opcional

g) Apellidos. Compuesto junto a nombre para formar el atributo de nombre nombreCompleto.

17.



http://mysql.conclase.net/curso/?cap=001

http://teleformacion.edu.aytolacoruna.es/PASCAL/document/tipos.htm

http://www.desarrolloweb.com/articulos/1054.php

https://docs.moodle.org/all/es/Campos\_de\_la\_base\_de\_datos

https://es.wikipedia.org/wiki/Registro\_(base\_de\_datos)

http://www.ite.educacion.es/formacion/materiales/93/cd/m2\_1/index.html

http://www.aulaclic.es/sqlserver/t\_8\_15.htm

http://es.slideshare.net/LilianaValenzuela1/vistas-en-bases-de-datos

http://www.ite.educacion.es/formacion/materiales/93/cd/m1\_1/informes.html

http://www.unizar.es/3w/cursoFMice/05\_guiones\_web.htm#que

https://susanagonzalezavalos.wordpress.com/070211-usos-y-aplicaciones-de-base-de-datos/

http://www.20minutos.es/noticia/203609/0/bases/datos/grandes/

https://users.dcc.uchile.cl/~rbaeza/inf/codd.html

http://tiempodenegocios.com/conociendo-y-aprendiendo-de-larry-ellison-el-genio-que-se-adelanto-a-ibm/

https://sites.google.com/site/basededatosgustavoa07/about-us

http://www.biografiasyvidas.com/biografia/g/gates.htm

http://garumfundatio.org/dia-garum-michael-widenius-el-software-libre-y-empresa/

https://www.ecured.cu/Sistema\_Gestor\_de\_Base\_de\_Datos

http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/406547/Tipos-y-funci-n-de-los-gestores-de-bases-de-datos

https://medievalstrucos.wordpress.com/2013/07/18/12-reglas-de-codd-para-bases-de-datos-relacionadas/

http://definicion.de/sql/

http://www.monografias.com/trabajos13/trsqlinf/trsqlinf.shtml

http://www.cristalab.com/tutoriales/como-funciona-sql-injection-seguro-eres-vulnerable-c113268l/

http://searchdatacenter.techtarget.com/es/definicion/Modelado-de-datos

https://docs.google.com/presentation/d/1s1DuKDcXn173ZG6rYyt6zUCJLZ3mFwNYzSfDDbbg4\_I/edit#slide=id.g16d49c3bcf\_0\_19

http://www.exa.unicen.edu.ar/catedras/dbases1/apuntes/1\_MERE-derivacion-DDL.pdf

http://www.genbetadev.com/bases-de-datos/fundamento-de-las-bases-de-datos-modelo-entidad-relacion

http://arantxa.ii.uam.es/~epulido/bdatos/tema2.pdf

https://www.uazuay.edu.ec/analisis/Modelo%20Entidad%20Relacion.pdf

http://tavoberry.com/MER/tipos\_de\_atributos.html

http://www.masadelante.com/faqs/base-de-datos