TALLER No. 1 – BD I

Presentado a:

Carlos Londoño

Área:

Teoría General De Sistemas

Presentado por:

Leonel García Virgen

CORPORACIÓN DE ESTUDIOS TECNOLÓGICOS

DEL NORTE DEL VALLE

(COTECNOVA)

INGENIERÍA DE SISTEMAS

I SEMETRE

JORNADA NOCTURNA

Sede

Cartago Valle

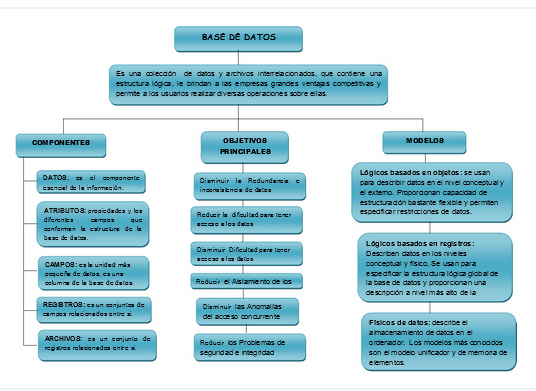
Agosto

2016

**SOLUCION AL TALLER**

1. **Definir el termino base de datos**

El término bases de datos fue escuchado por primera vez en un simposio celebrado en California en 1963. En una primera aproximación, se puede decir que una base de datos es un conjunto de información relacionada que se encuentra agrupada o estructurada.

1. 
2. **Definir los siguientes conceptos y realizar un mapa conceptual donde se evidencie como se relacionan todos los términos.**

* **DATO**: 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, estrictamente 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).
* **TIPO DE DATO:** Un tipo de datos es la propiedad de un valor que determina su dominio (qué valores puede tomar), qué operaciones se le pueden aplicar y cómo es representado internamente por el computador. Se clasifican en
* Dinámicos
* Estáticos
* El tipo cadena
* Estructurados
* Simples
* Ordinales
* No-ordinales
* **CAMPO:** es cada una de las columnas que forman la tabla. Contienen datos de tipo diferente a los de otros campos. En el ejemplo anterior, un campo contendrá un tipo de datos único, como una dirección, o un número de teléfono, un nombre, etc.
* **REGISTRO:** es cada una de las filas en que se divide la tabla. Cada registro contiene datos de los mismos tipos que los demás registros. Ejemplo: en una tabla de nombres y direcciones, cada fila contendrá un nombre y una dirección.
* **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.

* **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:** datos, una consulta es el método para acceder a los datos en las bases 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. El lenguaje de consultas a base de datos más utilizado es el SQL.

Técnicamente hablando, las consultas a la base de datos se realizan a través de un lenguaje de manipulación de datos (DML – Data Manipulation Language). SQL es un lenguaje DML, pero además posee otras características de otros lenguajes. Por ejemplo, permite también crear bases de datos. La consulta básica en SQL es llamada select-from-where.

* **INDICE:** El índice de una base de datos 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.
* **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.
* **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. Hay cuatro tipos de informes básicos: tabulares, en columnas, para gráficos y para etiquetas.
* **GUIONES:** En informática, un script, archivo de órdenes, archivo de procesamiento por lotes o, cada vez más aceptado en círculos profesionales, dicha extensión para ejecutar el guion, por lo tanto, esta suele ser añadida por tradición, o más bien, es útil para que el usuario pueda identificar estos archivos a través de una interfaz de línea de comandos sin necesidad de abrirlo. Difieren de los programas de aplicación, debido a que los últimos son más complejos; además, los guiones son más bien un programa que le da instrucciones a otros más avanzados.
* **PROCEDIMIENTOS:** Un procedimiento almacenado de SQL Server es un grupo de una o varias instrucciones Transact-SQL o una referencia a un método de Common Runtime Language (CLR) de Microsoft .NET Framework. Los procedimientos se asemejan a las construcciones de otros lenguajes de programación, porque pueden:
* Aceptar parámetros de entrada y devolver varios valores en forma de parámetros de salida al programa que realiza la llamada.
* Contener instrucciones de programación que realicen operaciones en la base de datos. Entre otras, pueden contener llamadas a otros procedimientos.
* Devolver un valor de estado a un programa que realiza una llamada para indicar si la operación se ha realizado correctamente o se han producido errores, y el motivo de estos.

1. **NOMBRE Y EXPLIQUE 5 USOS DE LAS BASES DE DATOS**

* Recopilación de URL/Libros/Revistas sobre algún tema educativo.
* Proyectos de tipo Portafolio electrónico.
* Recopilación de conceptos acompañados de imágenes relativas. Se puede orientar como una galería de imágenes comentadas o como un glosario ilustrado.
* Espacio para compartir archivos.
* Presentar contenidos creados por los estudiantes fotos/posters/sitios Web/poemas para ser comentados o revisados por los otros participantes
* Espacio para votar y comentar una lista de potenciales logos/mascotas o nombres de proyectos.

1. **BUSQUE EN INTERNET LAS 10 BASES DE DATOS MAS GRANDES DEL MUNDO. ANOTA SU NOMBRE Y SU TAMAÑO Y EN UNA HOJA DE CALCULO, GENERA UN GRAFICO QUE MUESTRA LA COMPARATIVA DE ESTAS BASES DE DATOS.**

**Este es el listado de las 10 bases de datos más grandes del mundo:**

\* World Data Centre for Climate (6.000 Tb)

\* National Energy Research Scientific Computing Center (2.800 Tb)

\* Google (957.5Tb)

850 TeraBytes – Google Search

220 TeraBytes – Google Analytics

70.5 TeraBytes – Google Earth

9 TeraBytes – Orkut

4 TeraBytes – Personalized Search

2 TeraBytes – Google Base

\* AT&T (323 Tb)

\* Sprint (operador móvil de EEUU) (300 Tb)

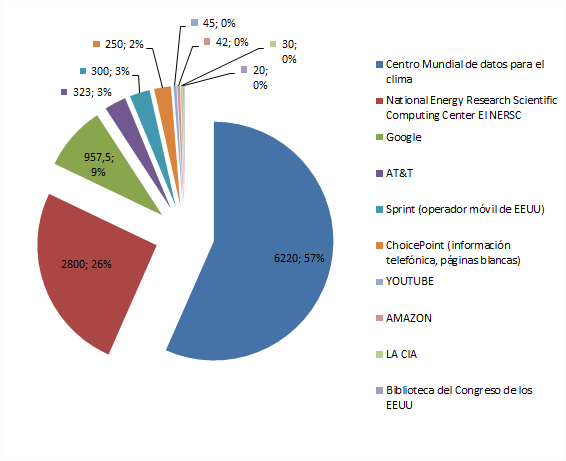
\* ChoicePoint (información telefónica, páginas blancas) (250 Tb)

\* YouTube (45 Tb)

\* Amazon (42 Tb)

\* La CIA (- Tb)

\* Biblioteca del Congreso de los EEUU (20 Tb)



1. **BUSCAR EN INTERNET LA BIOGRAFIA DE LOS SIGUENTES PERSONAJES Y COMENTE SU PRINCIPAL CONTRIBUCION A LA EVOLUCION DE LAS BASES DE DATOS**
2. **EDGAR FRANK CODD**

Edgar Frank Codd, fue un científico informático inglés, conocido por crear el modelo relacional de bases de datos.

Fecha de nacimiento: 19 de agosto de 1923, Isla de Pórtland, Reino Unido

Fecha de la muerte: 18 de abril de 2003, Williams Island

Libros: The Relational Model for Database Management, Cellular automata

Educación: Universidad de Míchigan, Universidad de Oxford, Exeter College

Premios: Premio Turing

1. **LARRY ELLISON**

Lawrence Joseph Ellison, conocido como Larry Ellison, es el fundador de Oracle donde fue Director Ejecutivo de 1977 hasta 2015.

Fecha de nacimiento: 17 de agosto de 1944 (edad 72), Manhattan, Nueva York, Estados Unidos

Patrimonio neto: 51,6 miles de millones USD (2016) Forbes

Organización Fundada: Oracle Corporation, Ellison Medical Foundation

Cónyuge: Melanie Craft (m. 2003–2010),

Hijos: Megan Ellison, David Ellison

Libros: Software: An Intimate Portrait of Larry Ellison and Oracle

1. **ROGER KENT SUMMIT:**

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

Fecha de nacimiento: 1930, Detroit, Míchigan, Estados Unidos

Hijos: Scott Summit, Jennifer Summit

En 1969, Summit amplió considerablemente las funciones del proyecto de tele documentación: 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.

1. **BILL GATES**

William Henry Gates III, conocido como Bill Gates, es un empresario, informático y filántropo estadounidense, cofundador de la empresa de software Microsoft junto con Paul Allen. Wikipedia

Fecha de nacimiento: 28 de octubre de 1955 (edad 60), Seattle, Washington, Estados Unidos

Patrimonio neto: 78,5 miles de millones USD (2016) Forbes

Cónyuge: Melinda Gates (m. 1994)

Organización Fundada: Microsoft, Fundación Bill y Melinda Gates, Más

Hijos: Jennifer Katharine Gates, Rory John Gates, Phoebe Adele Gates

El 4 de abril de 1975, siendo aún estudiante en la Universidad de Harvard crea la empresa de software Microsoft. En 1976 abandonó la universidad y se trasladó a Albuquerque, sede de MITS, para pactar con esa empresa la sesión del 50 % del lenguaje para computadoras Basic. Al año siguiente, se enteró del éxito de la empresa Apple y de que necesitaban un intérprete de Basic.

1. **MICHAEL MONTY WIDENIUS**

Ulf Michael Widenius, 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. Wikipedia

Fecha de nacimiento: 3 de marzo de 1962 (edad 54), Helsinki, Finlandia

Educación: Universidad Politécnica de Helsinki

Organización Fundada: MySQL AB

Libros: MySQL Reference Manual: Documentation from the Source, Más

Hijos: Maria Widenius, Max Widenius

1. **CON EL TERMINO “SISTEMAS GESTORES DE BASES DE DATOS”, RWEALIZAR LAS SIGUIENTES OPCIONES**
2. **DEFINICION:** 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.
3. **ENUMERAR LAS PRINCIPALES FUNCIONES:**

**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.

1. **NOMBRE LOS TIPOS DE SGBD**

**Tipos de gestores de bases de datos**

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:**

Basados en el modelo relacional, los datos se describen como relaciones que se suelen representar como tablas bidimensionales consistentes en filas y columnas. Cada fila (tupla, en terminología Relacional) representa una ocurrencia. Las columnas (atributos) representan propiedades de las filas.

Cada tupla se identifica por una clave primaria o identificadora.

Esta organización de la información, permite recuperar de forma flexible los datos de una o varias tablas, así como combinar registros de diferentes tablas para formar otras nuevas. No todas las definiciones posibles de tablas son válidas según el modelo relacional. En él, deben emplearse diseños normalizados que garantizan que no se producirán anomalías en la actualización de la BD.

De todas formas otras consideraciones, principalmente de rendimiento, llevan en ocasiones a que los diseños que se implantan no estén totalmente normalizados. Hallar el punto de equilibrio entre normalización y rendimiento es, con frecuencia, un punto clave para obtener un buen diseño de la BD cuando se utilizan SGBD relacionales

* **EN RED:**

Los SGBD relacionales se basan en el modelo de datos de red. Los datos en el modelo de red se representan mediante colecciones de registros y las relaciones entre los datos se representan mediante enlaces, que se pueden ver como punteros. Los registros en la base de datos se organizan como colecciones de grafos dirigidos.

* **JERÁRQUICOS:**

Los SGBD relacionales se basan en el modelo de datos jerárquico. El modelo jerárquico es similar al modelo de redes, en el sentido en que los datos y las relaciones entre los datos se representan mediante registros y enlaces, respectivamente. Éste se diferencia del modelo de redes en que los registros se organizan como colecciones de árboles en lugar de grafos dirigidos.

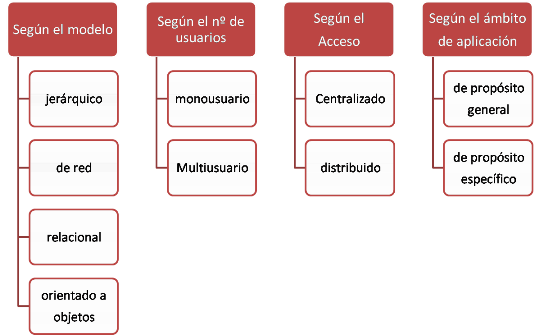
* **ORIENTADOS A OBJETOS**

Una de las novedades más prometedoras y más desarrolladas comercialmente de los nuevos SGBD, son los basados en un nuevo modelo de datos conocido como modelo orientado a objetos. La orientación a objetos es un paradigma que no se aplica sólo al desarrollo de SGBD sino, en general, al desarrollo de sistemas de información.

El modelo orientado a objetos está basado en una colección de objetos. Un objeto contiene valores almacenados en variables de ejemplares dentro de ese objeto. Un objeto también contiene fragmentos de código que operan en el objeto. Estos fragmentos de código se llaman métodos.

Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan juntos en clases. Una clase se puede ver como una definición de tipo para los objetos. Esta combinación de datos y métodos constituyendo una definición de tipo es similar a un tipo abstracto de datos en un lenguaje de programación.

La única manera de que un objeto pueda acceder a los datos de otro objeto es mediante la invocación de un método de ese otro objeto. Esta acción se llama paso de mensaje al otro objeto. Así, la interfaz de llamada de los métodos de un objeto define la parte visible externamente del objeto. La parte interna del objeto no es visible externamente. El resultado es obtener dos niveles de abstracción de datos.



1. **NOMBRAR LOS SGBD MAS POPULARES**

**MYSQL:**

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario seguramente el más usado en aplicaciones creadas como software libre.

Por un lado se ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso.

**Ventajas:**

* Velocidad al realizar las operaciones
* Bajo costo en requerimientos para la elaboración de bases de datos
* Facilidad de configuración e instalación.

**MICROSOFT SQL SERVER:**

Es un sistema de gestión de bases de datos relacionales basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Es un sistema propietario de Microsoft. Sus principales características son:

* Soporte de transacciones.
* Escalabilidad, estabilidad y seguridad.
* Soporta procedimientos almacenados.
* Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
* Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
* Además permite administrar información de otros servidores de datos

Su principal desventaja es el precio, aunque cuenta con una versión EXPRESS que permite usarlo en entornos pequeños. (Aprox. unos 4GB de información y varios millones de registros por tabla)

**ORACLE:**

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

Tradicionalmente Oracle ha sido el SGBS por excelencia, considerado siempre como el más completo y robusto, destacando por:

* Soporte de transacciones.
* Estabilidad.
* Escalabilidad.
* Es multiplataforma.

También siempre ha sido considerado de los más caros, por lo que no se ha estandarizado su uso como otras aplicaciones. Al igual que SQL Server, Oracle cuenta con una versión EXPRESS gratis para pequeñas instalaciones o usuarios personales.

**MICROSOFT ACCESS:**

Es un sistema de gestión de bases de datos Relacional creado por Microsoft (DBMS) para uso personal de pequeñas organizaciones.

Se ha ofrecido siempre como un componente de la suite Microsoft Office aunque no se incluye en el paquete “básico”.

Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas.

Entre las principales funcionalidades reseñables podemos indicar que:

* Permite crear tablas de datos indexadas.
* Modificar tablas de datos.
* Relaciones entre tablas (creación de bases de datos relacionales).
* Creación de consultas y vistas.
* Consultas referencias cruzadas.
* Consultas de acción (INSERT, DELETE, UPDATE).
* Formularios.
* Informes.
* Entorno de programación a través de VBA
* Llamadas a la API de windows.

**POSTGRESQL:**

Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. La comunidad PostgreSQL se denominada el PGDG (PostgreSQL Global Development Group).

Sus principales características son:

* Alta concurrencia: mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés)
* Amplia variedad de tipos nativos: provee nativamente varios soportes
* Ahorros considerables de costos de operación
* Estabilidad y confiabilidad

**DB2:**

Este SGBD es propiedad de IBM, bajo la cual se comercializa el sistema de gestión de base de datos. Utiliza XML como motor, además el modelo que utiliza es el jerárquico en lugar del modelo relacional que utilizan otros gestores de bases de datos. Es el único de los gestores que hemos comentado que nos relacional.

Sus caracteristicas más importantes son:

* Permite el manejo de objetos grandes (hasta 2 GB)
* La definición de datos y funciones por parte del usuario, el chequeo de integridad referencial,
* SQL recursivo, soporte multimedia: texto, imágenes, video, audio; queries paralelos, commit de dos fases, backup/recuperación on−line y offline.
* Permite agilizar el tiempo de respuestas de esta consulta
* Recuperación utilizando accesos de sólo índices.
* Predicados correlacionados.
* Tablas de resumen
* Tablas replicadas
* Uniones hash

Su principal desventaja es el precio, está dirigido solo a grandes empresas con necesidades de almacenamiento y procesamiento muy altas.

Al igual que SQL Server y Oracle dispone de una versión EXPRESS gratis pero no de libre distribución.

Existen muchos más gestores de bases de datos en el mercado, pero estos como he comentado son los más usados.

Todos son relacionales (a excepción del BD2) y comparten por tanto lenguaje de consulta (con algunas variantes propias) que es SQL. Es importante por tanto para cualquiera que desee trabajar con bases de datos comenzar por el estudio de este lenguaje común y luego estudiar las peculiaridades de la base de datos en cuestión.

1. **BUSCA EN INTERNET LAS LEYES DE CODD PARA EL FUNCIONAMIENTO DE SGBD RELACIONALES Y ESTABLECE UNA RELACION ENTRE CADA UNA DE LAS LEYES DE CODD**

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:

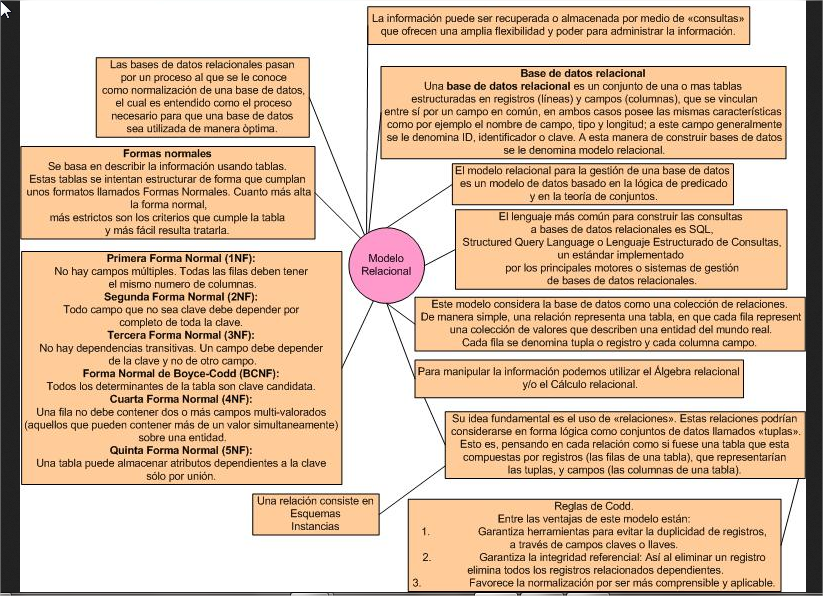
Cuando una versión distribuida del SGBD se introdujo por primera vez

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.

**RELACION ENTRE CADA UNA DE LAS LEYES DE COOD**

**https://prezi.com/vyed3l\_gjnwu/12-reglas-de-codd/**



1. **QUE ES SQL?**

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) 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.

**INDICA LAS REVISIONES Q HA SUFRIDO EL LENGUAJE A TRAVES DEL TIEMPO?**

Este lenguaje se llamaba SEQUEL (Structured English Query Language) y se implementó en un prototipo llamado SEQUEL-XRM entre 1974 y 1975. Las experimentaciones con ese prototipo condujeron, entre 1976 y 1977, a una revisión del lenguaje (SEQUEL/2), que a partir de ese momento cambió de nombre por motivos legales, convirtiéndose en SQL. El prototipo (System R), basado en este lenguaje, se adoptó y utilizó internamente en IBM y lo adoptaron algunos de sus clientes elegidos. Gracias al éxito de este sistema, que no estaba todavía comercializado, también otras compañías empezaron a desarrollar sus productos relacionales basados en SQL. A partir de 1981, IBM comenzó a entregar sus productos relacionales y en 1983 empezó a vender DB2. En el curso de los años ochenta, numerosas compañías (por ejemplo Oracle y Sybase, sólo por citar algunos) comercializaron productos basados en SQL, que se convierte en el estándar industrial de hecho por lo que respecta a las bases de datos relacionales.

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.

**BUSCA EL SIGNIFICADO DE 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).

**INDICA POR QUE UN ADMINISTRADOR DEBE PROTEGERSE FRENTE A EL?**

No sólo pone en riesgo la integridad de la aplicación, sino de todos los datos almacenados de los usuarios que la utilicen, y que se produce cuando no se filtra de forma correcta la información enviada por los usuarios.

Como consecuencias de estos ataques y dependiendo de los privilegios que tenga el usuario de la base de datos bajo el que se ejecutan las consultas, se podría acceder no sólo a las tablas relacionadas con la aplicación, sino también a otras tablas pertenecientes a otras bases de datos alojadas en ese mismo servidor.

Lo comentado anteriormente es posible gracias a que el uso de ciertos caracteres en los campos de entrada de información por parte del usuario, ya sea mediante el uso de los campos de los formularios que son enviados al servidor mediante POST o bien por medio de los datos enviados mediante GET en las urls de las páginas web, posibilitan coordinar varias consultas SQL o ignorar el resto de la consulta, permitiendo al hacker ejecutar la consulta que elija, de ahí que sea necesario realizar un filtrado de esos datos enviados para evitar problemas.

1. **DEFINA LOS SIGUIENTES TERMINOS**
2. **Lenguaje DML**

**Lenguaje de manipulación de datos (DML)**

Un lenguaje de manipulación de datos (Data Manipulation Language, o DML en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

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.

1- INSERT

Una sentencia INSERT de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.

Ejemplo 1 (inserto valores alumno pepe en la materia spd2 a la tabla cursada):

INSERT INTO ''cursada'' (''alumno'', ''materia'') VALUES (''pepe'', ''spd2'')

2- UPDATE

Una sentencia UPDATE de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

Ejemplo 1 (modifico la materia donde el alumno sea pepe):

UPDATE ''cursada'' SET ''materia''= ''spd3'' WHERE ''alumno''= ''pepe''

3- DELETE

Una sentencia DELETE de SQL borra uno o más registros existentes en una tabla.

Ejemplo 1 (borro todos los valores de las columnas alumno y materia donde la materia sea spd2):

DELETE FROM ''cursada'' WHERE ''materia''= ''spd2''

1. **Lenguaje DDL:**

**Lenguaje de definición de datos:**

Un lenguaje de definición de datos (Data Definition Language, DDL por sus siglas en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

La definición de la estructura de la base de datos incluye tanto la creación inicial de los diferentes objetos que formarán la base de datos, como el mantenimiento de esa estructura. Las sentencias del DDL utilizan unos verbos que se repiten para los distintos objetos. Por ejemplo para crear un objeto nuevo el verbo será CREATE y a continuación el tipo de objeto a crear. CREATE DATABASE es la sentencia para crear una base de datos, CREATE TABLE nos permite crear una nueva tabla, CREATE INDEX crear un nuevo índice… Para eliminar un objeto utilizaremos el verbo DROP (DROP TABLE, DROP INDEX…) y para modificar algo de la definición de un objeto ya creado utilizamos el verbo ALTER (ALTER TABLE, ALTER INDEX…).

Las principales funcionalidades de SQL como lenguaje de definición (DDL) son la creación, modificación y borrado de las tablas que componen la base de datos, así como de los índices, vistas, sinónimos, permisos, etc. que pudieran definirse sobre las mismas. Este documento introduce los comandos para el trabajo básico con tablas.

CREATE TABLE: Crear una tabla

SHOW TABLES: mostrar tablas

DROP TABLE<nombre de tabla>: Borrar tabla

DESCRIBE <nombre de tabla> Mostrar estructura de una tabla

1. **Lenguaje DCL:**

Un Lenguaje de Control de Datos (DCL por sus siglas en inglés: Data Control Language) es un lenguaje proporcionado por el Sistema de Gestión de Base de Datos que incluye una serie de comandos SQL que permiten al administrador controlar el acceso a los datos contenidos en la Base de Datos.

Algunos ejemplos de comandos incluidos en el DCL son los siguientes:

GRANT: Permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas.

REVOKE: Permite eliminar permisos que previamente se han concedido con GRANT.

Las tareas sobre las que se pueden conceder o denegar permisos son las siguientes:

CONNECT

SELECT

INSERT

UPDATE

DELETE

USAGE

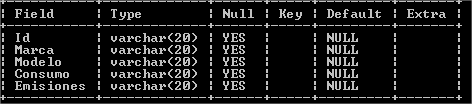
En Oracle, la ejecución de un comando DCL implica un COMMIT de forma implícita. Sin embargo, en PostgreSQL, la ejecución de un comando DCL forma parte de una transacción, por lo que puede ser deshecha mediante el comando ROLLBACK.

1. **Lenguaje TCL:**

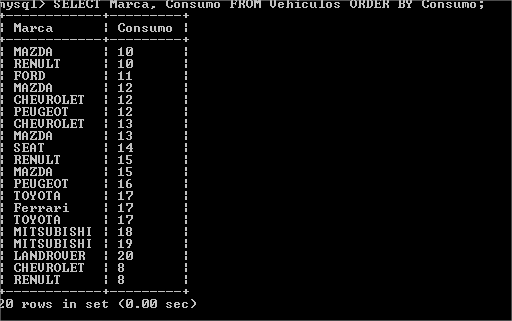
Originado del acrónimo en inglés "Tool Command Language" o "lenguaje de herramientas de comando", actualmente se escribe como "Tcl" en lugar de "TCL"), es un lenguaje de script creado por John Ousterhout, que ha sido concebido con una sintaxis sencilla para facilitarse su aprendizaje, sin detrimento de la funcionalidad y expresividad.

Se utiliza principalmente para el desarrollo rápido de prototipos, aplicaciones "script", interfaces gráficas y pruebas. La combinación de Tcl con Tk (del inglés Tool Kit) es conocida como Tcl/Tk, y se utiliza para la creación de interfaces gráficas.

1. A) 

B) 

C), D), E) 

F) 

1. 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.

El proceso de trasladar un problema del mundo real a un ordenador, usando bases de datos, se denomina modelado.

1. **Modelos de Datos Conceptuales:** Son los orientados a la descripción de estructuras de datos y restricciones de integridad. Se usan fundamentalmente durante la etapa de Análisis de un problema dado y están orientados a representar los elementos que intervienen en ese problema y sus relaciones. El ejemplo más típico es el Modelo Entidad-Relación.

**Modelos de Datos Lógicos:** Son orientados a las operaciones más que a la descripción de una realidad. Usualmente están implementados en algún Manejador de Base de Datos. El ejemplo más típico es el Modelo Relacional, que cuenta con la particularidad de contar también con buenas características conceptuales (Normalización de bases de datos).

**Modelos de Datos Físicos:** Son estructuras de datos a bajo nivel implementadas dentro del propio manejador. Ejemplos típicos de estas estructuras son los Árboles B+, las estructuras de Hash, etc.