UD 01 - ALMACENAMIENTO DE LA INFORMACIÓN

1. Los ficheros de información. Concepto de fichero.

En los setenta las necesidades empresariales de contabilidad y facturación se solventaban utilizando un número reducido de archivos en papel agrupados y ordenados (ficheros).

Con la primera informatización se posibilita el acceso de forma más rápida a través del ordenador. Por eso se sigue hablando de ficheros, formularios, carpetas, directorios...

Los ficheros permitieron llevar a cabo el almacenamiento de datos de forma permanente en dispositivos de memoria masiva.

Fichero o archivo: Información relacionada tratada como un todo y organizada de forma estructurada. Es una secuencia de dígitos binarios que organiza información relacionada con un mismo aspecto. Están formados por registros lógicos divididos en campos que contienen informaciones elementales que forman un registro.

Los datos pueden ser añadidos, suprimidos, consultados, actualizados en cualquier momento.

Los ficheros son muy voluminosos, por lo que solo pueden ser llevados a la memoria principal partes de ellos.
Registro físico o bloque: Cantidad de información transferida entre el soporte en el que se almacena el fichero y la memoria principal del ordenador, en una operación de lectura/grabación.

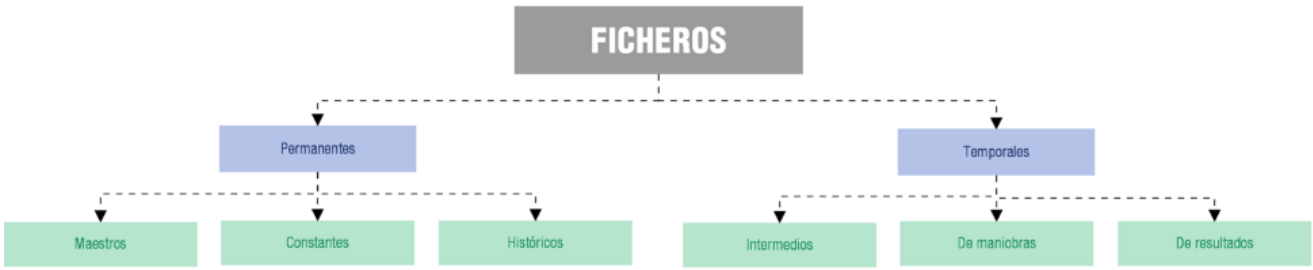
En cada operación de lectura/grabación se transfieren varios registros del fichero.

Factor de blocaje: Número de registros que entran en un bloque.
Bloqueo de registros: Operaciones de agrupar varios registros en un bloque.

Parámetros de uso

El uso de un fichero puede definirse por una serie de parámetros:
a. **Capacidad o volumen**: Espacio, en caracteres, que ocupa el fichero. Se calcula multiplicando el número previsto de registros por su longitud media.
b. **Actividad**: Cantidad de consultas y modificaciones en el fichero.
Debe tenerse en cuenta:
- Tasa de consulta o modificación: Porcentaje de registros consultados o modificados en cada tratamiento del fichero respecto a los registros totales.
- Frecuencia de consulta o modificación: Veces que se accede al fichero para consultar o modificar en un periodo de tiempo fijo.
c. **Volatilidad**: Cantidad de inserciones y borrados en el fichero.
Debe conocerse:
- **Tasa de renovación**: Porcentaje de registros renovados en cada tratamiento respecto al total de registros contenidos.
- **Frecuencia de renovación**: Veces que se accede al fichero para renovarlo en un periodo de tiempo fijo.
d. **Crecimiento**: Variación de la capacidad del fichero que se mide con el porcentaje de registros en que aumenta el fichero en cada tratamiento (tasa de crecimiento)

1.1. Tipos de ficheros



1. **Ficheros permanentes**: Amplio periodo de permanencia en el sistema. Información relevante para el funcionamiento de una aplicación.
    1. **Ficheros maestros**: Parte central de la aplicación, núcleo. Estado actual de los datos que pueden modificarse desde la aplicación. Ej.: Archivo con los datos de los usuarios
    2. **Ficheros constantes**: No suelen ser modificados, se accede a ellos para realizar consultas. Datos fijos de la aplicación. Ej.: Archivo con códigos postales
    3. **Ficheros históricos**: Reconstrucción de situaciones. Fueron considerados ficheros actuales en un periodo o situación anterior. Ej.: Usuarios dados de baja.
2. **Ficheros temporales**:
    1. **Ficheros intermedios**: Resultados de una aplicación que serán usados por otra
    2. **Ficheros de maniobras**: Datos de una aplicación que no pueden mantenerse en memoria por falta de espacio
    3. **Ficheros de resultados**: Datos que van a transferirse a un dispositivo de salida

1.2. Soportes de información

Inicialmente, eran tambores de cinta magnética. Similar a los casetes pero de mayor dimensión y capacidad de almacenamiento (formato digital, ceros y unos, en orden secuencial)
Posteriormente, avances en el hardware: disquete, disco duro. Dispositivos de acceso aleatorio no es necesario pasar por todos los datos desde el inicio hasta la zona donde se encuentra la información.

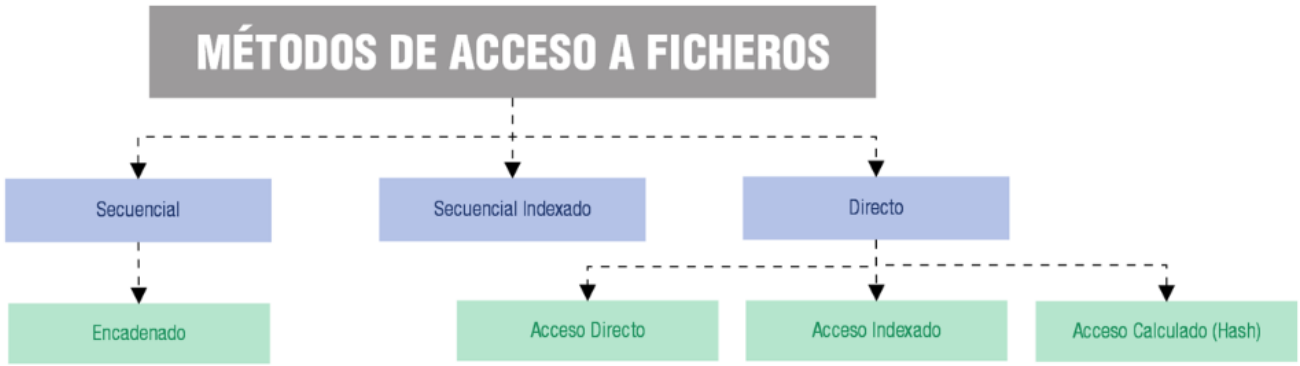
Distinguimos:

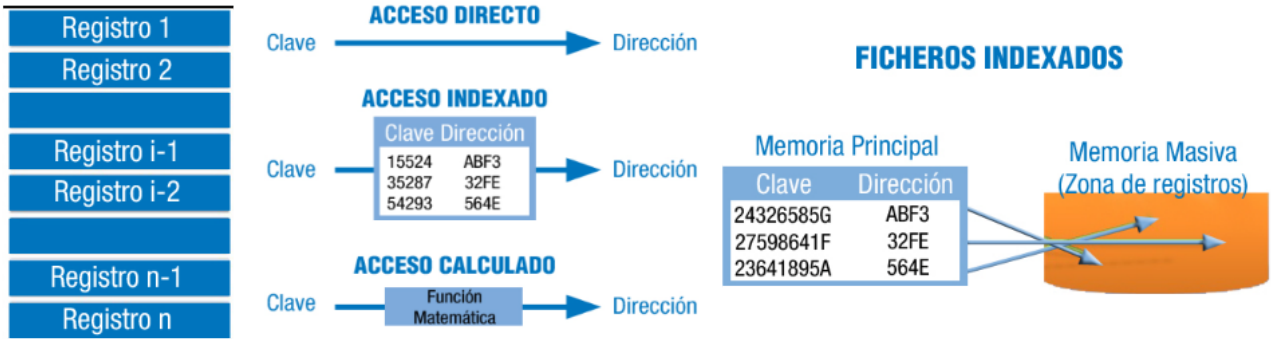
- **Soportes de acceso secuencial o no direccionables**: Para leer dato que está en mitad debe leerse todo hasta llegar. Copias de seguridad.
- **Soportes de acceso directo o direccionables**: Acceso de forma directa ubicándose en la posición deseada.

1.3. Métodos de acceso

Las modificaciones del acceso buscaban:

- Proporcionar acceso rápido a los ficheros
- Economizar el almacenamiento
- Facilitar la actualización de registros
- Permitir que la estructura refleje la organización real de la información





1.4. Ficheros secuenciales

- Los registros están almacenados de forma continua. La única forma de acceder a él es leyendo un registro tras otro de principio a fin. En ellos suele haber una marca indicativa de fin de fichero (EOF).
- Pueden usar dispositivos o soportes no direccionables o de acceso secuencial como las cintas magnéticas de almacenamiento de datos, cd, dvd (espiral continua)
- Los registros se identifican por la información en su campo **clave** o **llave**. Ordenando un archivo secuencial por su clave se puede hacer más rápido la lectura o escritura.
- La lectura siempre es hacia adelante
- No permite multiusuario
- Tiene estructura rígida de campos. Los registros deben aparecer en orden
- La apertura del fichero condiciona la lectura o escritura
- Aprovechan al máximo el soporte (no hay huecos vacíos)
- Se pueden grabar en soporte secuencial y en soporte direccionable
- Los lenguajes de programación pueden trabajar con ellos
- No se pueden insertar registros entre los ya grabados.

1.5. Ficheros de acceso directo o aleatorio

- Se puede acceder indicando la posición relativa dentro del archivo o, usualmente, a través de una clave que forma parte del registro.
- En dispositivos de acceso directo (discos magnéticos)
- No se encuentran en posiciones consecutivas sino en posiciones lógicas
- La clave es transformada y se obtiene la dirección física a la que corresponde el registro. Puede ser **acceso directo**, **indexado** o **calculado**. En el acceso directo la clave es la propia dirección debiendo ser numérica y comprendida en su rango de valores (más rápido)
- La medida básica de posicionamiento del puntero es el byte. Según la codificación sea Unicode o ANSI se utilizarán 1 o 2 bytes por carácter, respectivamente.

Características fundamentales:

- **Posicionamiento inmediato**
- Registros de **longitud fija**
- El registro **puede abrirse en modo mixto**, para lectura y escritura
- Pueden usarlo **múltiples usuarios**
- **Se borran colocando un cero** en la posición que ocupan
- Permiten el uso de **algoritmos de compactación de huecos**
- Los archivos se crean con un **tamaño definido** (hay un máximo de registros establecidos durante la creación)
- **Solo en soportes de acceso directo o direccionales**
- Usados cuando el acceso a los datos de un registro se hace siempre empleando la misma clave y la **velocidad de acceso a un registros es importante**
- Permiten la **actualización de los registros en el mismo fichero, sin necesidad de copiarlo**
- Permiten realizar actualización en tiempo real.

1.6. Ficheros indexados

Utilizan **índices** que permiten el acceso a un registro del fichero de forma directa, sin leer los anteriores. Existe:

- **Zona de registros**: Se encuentran en ella los datos del archivo
- **Zona de índices**: Contiene una tabla con las claves de los registros y posiciones que se encuentran en los mismos. Está ordenada por el campo clave.

La tabla de índices se carga en memoria principal para las búsquedas de la fila con la clave del registro a encontrar. Después se accede a esa fila en la zona de registro y se posiciona en la dirección indicada.

La zona de registros debe incluir todas las direcciones posibles del archivo (**problema determina su tamaño y mantenerla ordenada**)

El registro tiene que tener campo o combinación de campos que permita identificarlo de **forma única** (**campo clave**)
Puede usar tanto **acceso secuencial** (se leen ordenados por el contenido del campo clave, independientemente del orden en el que fueran grabados, accediendo a través del índice) como **acceso directo** (se accede al registro deseado por el índice).

1.7 Ficheros secuenciales indexados o parcialmente indexados

Tienen también una zona de índices y otra de registros pero esta está **dividida en segmentos** ordenados (bloques de registros).
En la tabla de índices, cada fila hace referencia a los segmentos. **La clave corresponde al último registro y el índice apunta al registro inicial**. Al acceder al primer registro del segmento, dentro de él se localiza **secuencialmente** el registro buscado.

- Es **muy usado para cuando hay pocos registros o para aquellos en los que se maneja el fichero -completo** (para todo, vamos).
- Se **permite acceso secuencial**, interesante cuando la tasa de actividad alta. Además leyendo por el campo clave.
- Se **permite el acceso directo**, usando para ello las tablas de índices.
- Se pueden **actualizar los registros** sin necesidad de crear fichero de copia
- **Ocupa más espacio en disco** que los ficheros secuenciales por el uso de índices
- Solo admite **soportes direccionales** (direccionables)
- Es más **caro**: Necesita hardware y software más sofisticado.

1.8 Ficheros de acceso calculado o hash

Con el acceso calculado o hash los accesos en ficheros indexados pueden ser más rápidos porque en lugar de consultar una tabla **se usa una transformación o función matemática conocida (hash) que a partir de la clave genera la dirección de cada registro del archivo**. Si la clave es alfanumérica debe previamente ser transformada en un número.

Este tipo de ficheros presenta como problema que a partir de diferentes claves se obtenga la misma dirección al aplicar la función matemática o transformación (**colisión**). Las claves que generan la misma dirección (**sinónimos**). Para evitarlo se aplican métodos como tener un bloque de excedentes o zona de sinónimos, crear archivo de sinónimos...

Algunos de los métodos de transformación son:

- **Módulo**. Dirección igual al resto de división entera entre clave y número de registros
- **Extracción**. Dirección igual a una parte de las cifras que extraen la clave.

Los buenos hash producen el menor número de colisiones. A ser posible, biunívoca (correspondencia uno a uno). Obtiene un número entre 1 y n, siendo n el número de direcciones del fichero.

2. Bases de datos

Los ficheros de almacenamiento provocan que las aplicaciones pierdan independencia y surjan inconvenientes como información duplicada, incoherencia de datos, fallos de seguridad...
Se dio paso a de los sistemas basados en ficheros a los sistemas gestores de bases de datos.

Base de datos: **Colección de datos relacionados lógicamente entre sí**, con una definición y descripción comunes y estructurados de una determinada manera. Representa entidades y sus interrelaciones, almacenados con la mínima redundancia y **posibilitando acceso eficiente** por aplicaciones y usuarios.

Es el conjunto de datos de distinto tipo relacionado entre sí, junto con un programa de gestión de dichos datos

La base de datos consta de:

- **Entidades**: Objeto real o abstracto con características que lo diferencian de otros datos, del cual se almacena información. (Doctor, consulta..)
- **Atributos**: Datos que se almacenan en la entidad. Cualquier propiedad o característica de esta. (Nombre, apellido, hora...)
- **Registros**: Es donde se almacena la información de cada entidad. Conjunto de datos que contienen atributos de una repetición de entidad. (Jesús López Navas 23/03/2010 ...)
- **Campos**: Dónde se almacenan los atributos de cada registro. "Jesús"

2.1. Ventajas

- **Acceso múltiple**: Acceso simultáneo de usuarios y aplicaciones
- **Utilización múltiple**: Cada usuario o aplicación dispone de una visión de la estructura, solo accede a lo que le corresponde.
- **Flexibilidad**: El acceso puede establecerse de diferentes formas. Tiempo de respuesta recudido.
- **Confidencialidad y seguridad**: Control de acceso. Evita acceso de usuarios no autorizados.
- **Protección contra fallos**: Hay mecanismos definidos para recuperar datos de forma fiable.
- **Independencia física**: Cambio de soporte físico no afecta a bases de datos o aplicaciones.
- **Independencia lógica**: Cambios de la base de datos no afecta a las aplicaciones
- **Redundancia**: Datos almacenados una sola vez (salvo casos de necesidad)
- **Interfaz de alto nivel**: Manejo con lenguajes de alto nivel de forma cómoda
- **Consulta directa**: Herramientas interactivas para acceder a los datos

2.2. Usos

Veamos los cuatro roles posibles en los usuarios de bases de datos:

Rol	Funciones
Administrador	Persona encargada de **crear (implementar físicamente) la base de datos**. Escoge tipos de ficheros, índices, ubicación... Toma las **decisiones relacionadas con el funcionamiento físico**, considerando el sistema que lo va a usar Establece **política de seguridad y acceso**.
Diseñador	**Identifican los datos, relaciones entre ellos, restricciones...** Deben conocer a fondo los datos y procesos a representar en la base de datos, **conociendo las reglas de negocio**. El diseñador debe implicar en el proceso a todos los usuarios de la base de datos.
Programador de aplicaciones	**Implementan los programas de aplicación que servirán a los usuarios finales**. Posibilitan consultas, inserción, actualización o eliminación (**CRUD**). Usan lenguajes de tercera o cuarta generación (C, Fortran, Smalltalk, Ada, C++, C#, Cobol, Delphi, Java...)
Usuarios finales	**Clientes finales**. Al implementar, diseñar, mantener se busca cumplir los requisitos establecidos por el usuario final para gestionar su información.

Las bases de datos son usadas en Banca (clientes, cuentas, transacciones); Líneas aéreas (clientes, horarios, vuelos); Universidades (estudiantes, carreras); Telecomunicaciones (llamadas, saldo);

Medicina; Justicia; Legislación; Organismos; Posicionamiento geográfico; Hostelería y tusrismo; Ocio; Cultura...

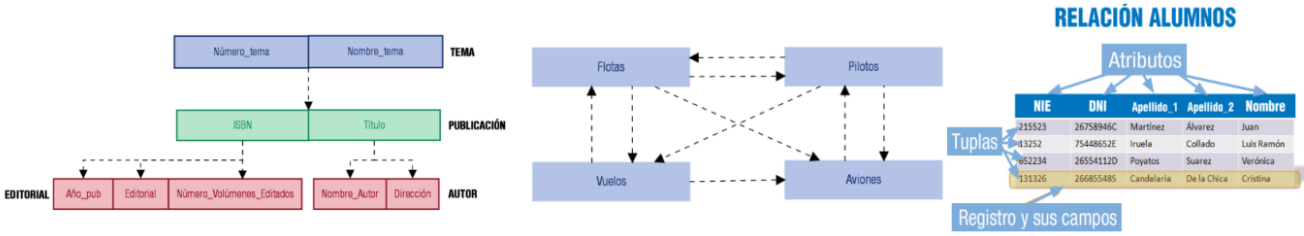
2.3. Ubicación de la información

El tamaño de las bases de datos puede ser muy diferente. Pero todas se suelen almacenar en discos duros y otros dispositivos de almacenamiento.
A veces una gran base de datos podría necesitar servidores en lugares diferentes.

Los más usados:

- **Discos SATA**: (Serial Advanced Technology Attachment) Interfaz de transferencia entre la blaca base y los dispositivos de almacenamiento (disco duro, cd, dvd, unidades de estado sólido). Proporciona mayor velocidad y aprovechamiento con varias unidades, mayor longitud del cable de transmisión, capacidad de conectar las unidades al instante sin tener que apagar el ordenador. Transferencias de 150 Megabytes por segundo (SATA 150 MB/s, Serial ATA-150). Hay dispositivos SATA II a 300 MB/s (Serial ATA-300) y Sata III hasta 600 MB/s
- **Discos SCSI**: (Small Computers System Interface): Para discos duros con gran capacidad de almacenamiento y velocidad de rotación. SCSI Estándar, SCSI Rápido, SCSI Ancho-Rápido (FastWide). Tiempo medio de acceso 7 milisegundos, velocidad de transmisión secuencial 5MB/s en estándar; 10 MB/ps en Rápidos, 20 MBps en FastWide. El controlador SCSI puede manejar hasta 7 discos SCSI.
- **RAID**: (Redundant Array of Independent Disk) Matríz de discos independientes. Basada en almacenamiento redundante. Se montan dos o más discos duros formando un bloque de trabajo para obtener ampliación de capacidad, mejor velocidad, seguridad de almacenamiento. Hay varios sistemas de RAID:
- **Sistemas NAS**: (Network Attached Storage): Almacenamiento masivo en red. Permiten compartir la capacidad del servidor con ordenadores personales o clientes a través de la red. Hay sistema operativo optimizado para dar acceso a los datos por protocolos específicos. Dispositivos con capacidades muy altas y conectan por red.
- **Sistemas SAN**: (Storage Area Network): Red de área de almacenamiento. Red para conectar servidores, arrays de discos y librerías de soporte. Los recursos están disponibles para variso servidores en red de área local amplia. La información no reside en ninguno de los servidoresd. Se optimiza el poder de procesamiento para aplicaciones comerciales y se le puede proporcionar capacidad al servidor que más lo necesite.

3. Modelos de bases de datos



Principales tipos de Bases de Datos

Los principales tipos de Bases de Datos son:

- Relacionales: la información que almacena la Base de Datos está relacionada entre sí. Los datos relacionados (registros o filas) son almacenados en tablas que constan de varios campos (columnas).
- No relacionales: los datos no tienen porqué estar relacionados entre sí y por lo tanto no tienen que almacenarse en estructuras fijas como las tablas del modelo de base de datos relacional.

Las Bases de Datos NoSQL pertenecen al modelo no relacional. Las principales características y ventajas de este tipo son:

- SQL no es el lenguaje de consulta/modificación de datos principal, aunque sí lo soportan, de ahí el nombre No Sólo SQL.
- Los datos no tienen que almacenarse en tablas.

- Generalmente, su arquitectura es distribuida almacenándose la información en más de una máquina del sistema. Por lo tanto, los sistemas que las soportan tienen una mayor escalabilidad horizontal (a mayor número de nodos mayor rendimiento) y también mayor tolerancia ante fallos en los distintos nodos.
- Son más eficientes en el procesamiento de los datos que las Bases de Datos relacionales
- Son más eficientes en el procesamiento de los datos que las Bases de Datos relacionales, por eso son la elección para aplicaciones que hacen un uso intensivo de estos ("streaming", etc.).
- Utilizan lo que se conoce como consistencia eventual que consiste en que los cambios realizados en los datos serán replicados a todos los nodos del sistema, lo cual aumenta el rendimiento de estos sistemas en contraposición a las propiedades ACID de las Bases de Datos relacionales.

¿Por qué JSON en bases de datos relacionales?

Las bases de datos Not only SQL, se basan en un esquema flexible de datos, en los cuales no necesitas declarar o crear primero dicho esquema para comenzar a almacenar información como tampoco es estrictamente necesario el proceso de normalización.Dentro de la industria desde hace años, llegaron a irrumpir las bases de datos de tipo documento tales como MongoDB, las cuales mostraron que al no estar amarradas al esquema tradicional de SQL, podían ofrecer una velocidad de escritura y lectura aún muy superior a lo manejado por las bases de datos relacionales; sin embargo esa realidad se ha vuelto a modificar gracias a los últimos esfuerzos de MySQL gracias a su implementación nativa para guardar, modificar y eliminar datos en formato JSON.

3.1. Modelo jerárquico o en árbol

La información se organiza en una jerarquía en la que la relación de entidades siempre es de padre/hijo. Sigue estructura de modelo de árbol invertido. Hay nodos con atributos o campos. Un nodo puede tener más de un hijo pero solo un padre.

Los datos se almacenan en **estructuras lógicas (segmentos)** que se relacionan entre sí usando **arcos**.

IBM creó un sistema administrador de información (IMS) con las bases del modelo que la mayoría de SGI de los setenta adoptaron. Este modelo están en desuso actualmente.

3.2. Modelo en red (Bases de datos de primera generación)

La información se organiza en **registros (nodos) y enlaces**. En los registros se almacenan los datos. Los enlaces permiten relacionar estos datos.

Parecido con las jerárquicas pero **pueden tener más de un padre.**

Aparece en los sesenta como respuesta a los límites del modelo jerárquico. IDS de Bachman es el primer modelo de base de datos en red. Se trató de crear un estándar por parte de CODASYL, con gran aceptación.

3.3. Modelo relacional (Bases de datos de segunda generación)

- Desarrollado por Codd en 1970. Las más utilizadas actualmente.
- El usuario lo percibe como un conjunto de tablas (nivel lógico). A nivel físico, en cambio puede tener diferentes estructuras de almacenamiento.
- Usa **tablas bidimensionales** (relaciones) para representación lógica de datos y relaciones. Cada tabla (entidad) posee nombre único y contiene un conjunto de columnas
- Se llama registro, entidad, ocurrencia de entidad o tupla a cada registro de la tabla y campo o atributo a cada columna
- Conjunto de valores que puede tomar un atributo es el dominio
- La clave es un conjunto de atributos que identidica de forma única a una tupla.

Las tablas deben cumplir:

- Todos los registros son del mismo tipo
- Solo puede tener un tipo de registro

- No hay campos o atributos repetidos
- No hay registros duplicados
- No hay orden en el almacenamiento de registros
- Cada registro o tupla se identifica por clave compuesta por uno o varios atributos

Las consultas se construyen en SQL.
En su diseño tiene gran relevancia la **normalización**. Consiste en definir las reglas que determinan las dependencias entre los datos de una base de datos relacional. Si definimos esta relación o dependencia entre los elementos de una determinada base de datos de la manera más sencilla posible, conseguiremos que la cantidad de espacio necesario para guardar los datos sea el menor posible y la facilidad para actualizar la relación sea la mayor posible. Es decir, optimizaremos su funcionamiento.

3.4. Modelo orientado a objetos (Bases de datos de tercera generación)

Define la base de datos en términos de objetos: propiedades y operaciones. Objetos con misma estructura y comportamiento pertenecen a una clase y las clases se organizan en jerarquías. Las operaciones de cada clase son métodos.

Hay sistemas basados en el modelo relacional que han evolucionado para incorporar objetos (sistemas objeto-relacionales).

Se basa como toda la POO en:

- **Encapsulación**. Se oculta información al resto de objetos
- **Herencia**. Los objetos heredan comportamientos en la jerarquía de clases
- **Polimorfismo**. Una operación puede ser aplicada a distintos tipos de objetos.

3.5. Modelo NoSQL

Como ventajas tienen:

- Pueden ejecutarse **en máquinas con pocos recursos**
- Tienen **escalabilidad horizontal** (simplemente se añaden más nodos)
- Pueden manejar **gran cantidad de datos** (estructura distribuida, tablas hash)
- **No generan cuellos de botella.** No necesitan transcribir sentencias SQL para ejecutarlas en el punto de entrada.

Diferencias:

- **No usan SQL** o solo lo usan como pequeñísimo apoyo
- **No usan estructuras fijas de tablas**. Usan otros tipos de almacenamiento como clave-valor, objetos, grafos.
- **No permiten operaciones JOIN.** Deben desnormalizarse los datos o realizar JOIN por software
- **Arquitectura distribuida**. Puede estar compartida en varias máquinas con mecanismos hash

Tipos:

- **Bases de datos clave-valor**: El modelo más popular y más sencillo. Cada elemento está identificado por llave única que permite la recuperación de la información de forma rápida (almacenada en objeto binario BLOB). Son eficientes en lectura y escritura. Ej.: Cassandra, Big Table, HBase.
- **Bases de datos documentales**: Información almacenada como documento JSON o XML. Se usa clave única para cada registro. Se pueden hacer búsquedas por clave-valor y también por contenido. Son más versátiles. Pueden emplearse en proyectos que tradicionalmente funcionaban con BBDD relacionales. Ej.: MongoDB, CouchDB.
- **Bases de datos en grafo**: Información representada como nodos de un grafo y relaciones son aristas del mismo. Se puede usar la teoría de grafos para recorrerla. Ofrecen negación eficiente. Ej.: Neo4j, InfoGrid, Virtuoso.

Las más usadas actualmente son:
Cassandra: Apache. Tipo clave-valor. Lenguaje propio (CQL). En Java.

Redis: Tipo clave-valor. Como array gigante para almacenar datos (cadenas, hash, conjuntos, listas)
MongoDB: Orientada a documentos. De esquema libre. Rápido ya que está en C++.
CouchDB: Apache. En GNU/Linux. Javascript como lenguaje de interacción. Permite creación de vistas para retornar valores de varios documentos. Permite uso de JOIN.

3.6. Otros modelos

Bases de datos objeto-relacionales: Híbrido entre relacionales y orientadas a objetos. Así se evita el coste de conversión de relacionales a orientadas a objetos. Se usa lo bueno del modelo relacional incorporando los objetos. Se almacenan tuplas aunque la estructura no está restringida sino que pueden definirse a partir de otras (herencia directa). Se basa en SQL99 (añadir procedimientos almacenados de usuario, triggers, tipos definidos, consultas recursivas, bases de datos OLAP , tipos LOB...). Permite incorporar funciones de lenguajes como SQL, Java, C. Las bases relacionadas clásicas de gran tamaño Oracle, SQL Server son objeto-relacionales.

Bases de datos deductivas o lógicas: Almacenan información y realizan deducciones a través de inferencias (derivan nuevas informaciones a partir de las existentes en base de datos introducidas por el usuario). Contrarrestan limitaciones del modelo relacional en respuesta a consultas recursivas y deducción de relaciones indirectas entre datos.

Bases de datos multidimensionales: Tienen varias dimensiones. En vez de un valor se encuentran varios dependiendo de los ejes definidos o con estructura orientada a consultas complejas y alto rendimiento. Matrices multidimensionales, cuadros de múltiples entradas, funciones de varias variables sobre conjuntos finitos (cubo). Facilita manejo de grandes cantidades de datos en una empresa.

Bases de datos transaccionales: Velocidad para gestionar el intercambio de información. Muy fiables. Sistemas bancarios, análisis de calidad y producción industrial.

4. Tipos de bases de datos

1. Según su contenido
    1. Con información actual: Información muy concreta y actualizada, de tipo numérico (estadísticas, series históricas...)
    2. Directorios: Datos sobre personas o instituciones (profesionales, investigadores, empresas, editoriales)
    3. Bases de datos documentales: Cada registro es un documento (puublicación impresa, documento audiovisual)
        1. Bases de datos de texto completo. Documentos en formato electrónico, volcado de su texto
        2. Archivos electrónicos de imágenes. Enlace directo con la imagen del documento original
        3. Bases de datos referenciales: No contienen el texto general sino solo información para describir y permitir la localización de los documentos impresos, sonoros, audivisuales, electrónicos. Luego habrá que localizarlos por otro servicio (archvivo, bibilioteca, fonoteca..)
2. Según su uso
    1. Individual: Mismo usuario.
    2. Compartida
    3. Acceso público
    4. Propietarias o bancos de datos
3. Según la variabilidad de la información
    1. Estáticas: Solo lectura. Datos históricos que pueden ser analizados
    2. Dinámicas: Se modifica con el tiempo. Actualización, adición..
4. Según la localización de la información
    1. Centralizadas: Datos se almacenan en un único punto
        1. Basada en anfitrión: Cliente y máquina servidor son la misma. Se conectan directamente a la máquina donde están los datos
        2. Basada en Cliente/Servidor. BBDD está en máquina servidor y acceden desde el cliente a través de red
    2. Distribuidas Se almacenan en lugares diferentes

5. Según el organismo producto
    1. Organismos públicos y administración
        1. De acceso público
        2. De acceso interno
    2. Instituciones sin ánimo de lucro
    3. Entidades privadas o comerciales
        1. Uso interno para circulación de información dentro de la empresa
        2. Uso interno ocasionalmente ofreciendo servicios al exterior
        3. Comerciales (uso externo)
    4. Cooperación en red: Elaboración compartida por instituciones...
6. Modo de acceso
    1. Acceso local
    2. CD-ROM
    3. En línea
        1. Vía telnet o internet
        2. Vía web
7. Según cobertura temática
    1. Científico-tecnológicas
        1. Multidisciplinares
        2. Especializadas
    2. Económico-empresariales
    3. Medios de comunicación
    4. Político-administrativo y jurídico
    5. Sanitario
    6. Gran público

5. Sistemas gestores de bases de datos

Sistema Gestor de Bases de Datos: Conjunto coordinado de programas, procedimientos, lenguajes.. que suministra a cualquiera de los roles que usan la base de datos, los medos para describir y manipular los datos contenidos en ella manteniendo su integridad, confidencialidad y seguridad.

Ventajas:

- **Independencia física**: La visión del usuario y la manipulación es independiente de cómo está físicamente
- **Visión abstracta** de los datos, oculta esa complejidad.
- **Integridad**
- **Facilidad de acceso**, rapidez, evita pérdidas
- **Seguridad** y privacidad
- **Eficiencia**
- Accesos **concurrentes** y posibilidad de compartir datos
- Facilidad de **intercambio** entre sistemas
- **Copias de seguridad** y **recuperación** en caso de fallo
- El **SGBD interacciona con el sistema operativo.** Los datos almacenados serán usados por otras aplicaciones, el SGBD ofrecerá facilidad a estas para el acceso y manipulación de la información basándose en los métodos del sistema operativo.

5.1. Funciones

Tres funciones fundamentales:

Función de descripción o definición: El diseñador de la BBDD puede crear estructuras apropiadas para integrar los datos (estructura, relaciones, restricciones). Permite definir la estructura interna, conceptual y externa. Se hace mediante el lenguaje de definición de datos o DDL.

A nivel interno: Espacio de disco reservado para la BBDD, longitud de campos, modo de representación (lenguaje para definición de estructura externa)
A nivel conceptual: Definición de entidades, atributos, relaciones, restricciones...
A nivel externo: Vistas de usuarios

Función de manipulación: Permite buscar, añadir, suprimir, modificar datos de la misma. Mediante el lenguaje de manipulación de datos o DML. También se define la vista externa de los usuarios de la base de datos o vistas parciales que cada usuario tiene de los datos.
Se entiende por manipulación el CRUD.

Función de control: El administrador establece mecanismos de protección. Creación y modificación se usuarios, sistemas para crear copias de seguridad, cargas de ficheros, auditoría, protección de ataques, configuración. Se hace mediante el lenguaje de control de datos o DCL.

DDL, DML, DCL mediante el SQL.

5.2. Componentes

El SGBD es un paquete software complejo que proporciona servicios para almacenamiento y explotación de datos de forma eficiente. Se compone de:

1. **Lenguajes de la base de datos**: Uso de lenguajes e interfaces para los diferentes tipos de usuarios. Especificar datos, estructura, relaciones, reglas, control de acceso, características. DDL, DML, DCL.
2. **Diccionario de datos**: Descripción de los datos almacenados. Lugar donde se deposita la información sobre los datos que forman. Interesante para programadores de aplicaciones. Características lógicas de las estructuras que almacenan los datos, nombre, descripción, contenido, organización. En BBDD relacionales: Definición de tablas, vistas, índices, disparadores, procedimientos, funciones. Espacio asignado y usado. Restricciones. Permisos. Auditoría.

3. **Gestor de la base de datos**: Garantizar acceso correcto, seguro, íntegro y eficiente. Proporciona interfaz entre los datos y los programas que los manejan. Interactúa con el sistema operativo. Garantiza los accesos concurrentes...
4. **Usuarios de la base de datos**: Con diferentes permisos. Administrador de base de datos (DBA) tiene el control centralizado y es responsable del buen funcionamiento, de autorizar, de vigilar su uso, de adquirir software y hardware necesario. Usuarios con diferentes necesidades, accesos y privilegios (diseñadores, operadores y mantenimiento, analistas y programadores, usuarios finales ocasionales simples avanzados autónomos)
5. **Herramientas de la base de datos**: Conjunto de aplicaciones que permiten a los administradores la gestión de bbdd, usuarios, permisos, generar formularios, informes, interfaces gráficas, aplicaciones.

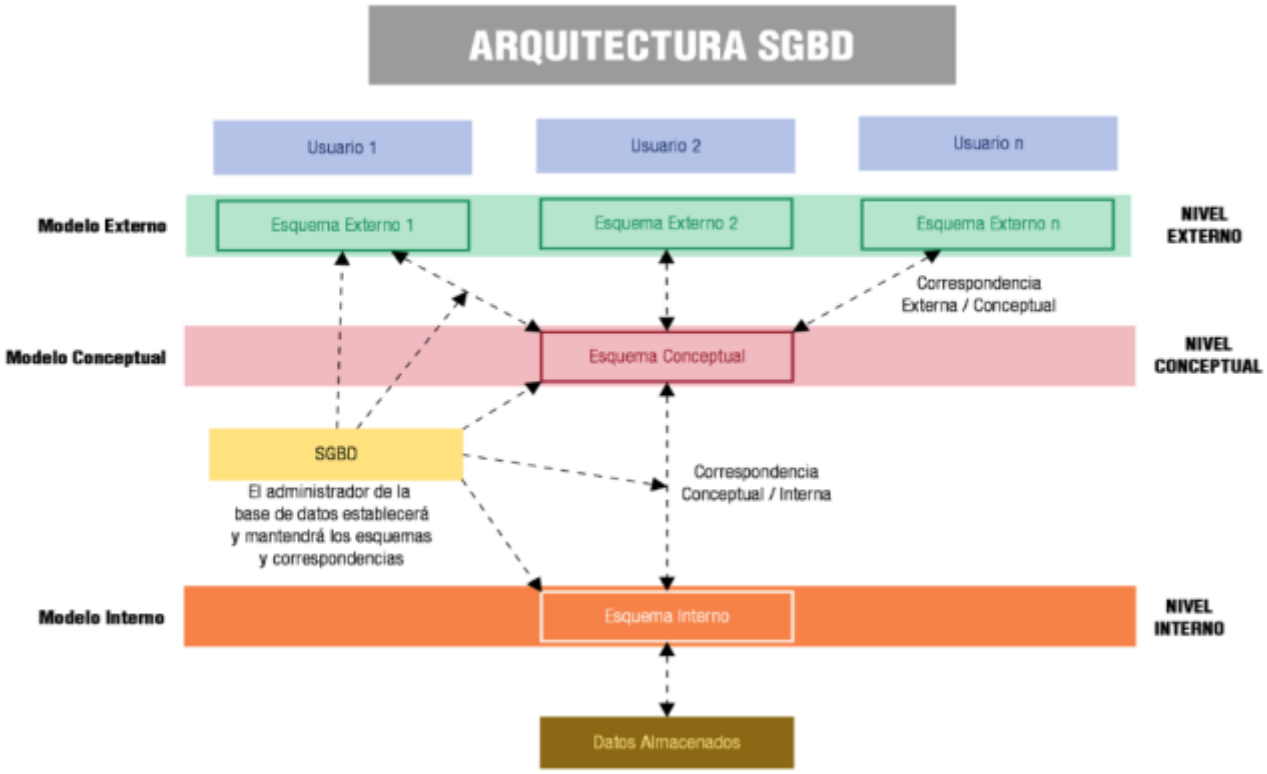
5.3. Arquitectura

Los estándares que han cobrado más importancia son ANSI/SPARC/X3, CODASYL, ODMG (este para las de objetos). ANSI e ISO son el referente de estandarización conformando un único modelo.

Nivel interno o físico: Se describe estructura física por un esquema que describe el sistema de almacenamiento y sus métodos de acceso. Es el más cercano al almacenamiento físico. Archivos que contienen la información, organización, métodos de acceso, tipos de registro, lognitud, campos.
Nivel lógico o conceptual: Estructura completa de la BBDD. Esquema entidades, atributos, relaciones, operaciones de usuarios, restricciones.
Nivel externo o de visión de usuario: Vistas que los usuarios perciben. Solo la parte que les interesa, ocultando el resto.

Existirá un único esquema interno, un único esquema conceptual y varios esquemas externos para uno o varios usuarios.

Esto garantiza:
Independencia lógica: Modificar esquema conceptual sin alterar esquemas externos, ni programas.
Independencia física: Modificar esquema interno sin modificar el conceptual o el externo. (Reorganizar sistema, ficheros...)



5.4. Tipos

Según el modelo lógico en que se basan: Relacional, orientado a objetos, jerarquico, en red
Según el número de usuarios: monousuario, multiusuario
Según el número de sitios en los que se distribuye la base de datos: centralizado, distribuidos

(homogeneos que usan el mismo SGBD o heterogéneos con acceso a varias BBDD autónomas preexistentes dando lugar a SGBD federados o sistemas multibase de datos en los que los SGBD participantes tienen cierta autonomía local)
Coste: Entre 10000 y 100000 euros. Los más económicos mono usuarios entre 0 y 3000. Los más completos más de 100000 euros.
Propósito: General (cualquier tipo de BBDD y aplicación) o específico cuando el rendimiento es fundamental y se necesita para una aplicación específica (ejemplo sistemas de procesamiento de transacciones en líneas en reservas o cosas así)

5.5. SGBD comerciales

SGBD	Descripción
ORACLE	Reconocido como uno de los mejores a nivel mundial. Es multiplataforma, confiable y seguro. Es Cliente/Servidor. Basado en el modelo de datos Relacional. De gran potencia, aunque con un precio elevado hace que sólo se vea en empresas muy grandes y multinacionales. Ofrece una versión gratuita Oracle Database Express Edition 11g Release 2.
MYSQL	Sistema muy extendido que se ofrece bajo dos tipos de licencia, comercial o libre. Para aquellas empresas que deseen incorporarlo en productos privativos, deben comprar una licencia específica. Es Relacional, Multihilo, Multiusuario y Multiplataforma. Su gran velocidad lo hace ideal para consulta de bases de datos y plataformas web.
DB2	Multiplataforma, el motor de base de datos relacional integra XML de manera nativa, lo que IBM ha llamado pureXML, que permite almacenar documentos completos para realizar operaciones y búsquedas de manera jerárquica dentro de éste, e integrarlo con búsquedas relacionales.
Microsoft SQL SERVER	Sistema Gestor de Base de Datos producido por Microsoft. Es relacional, sólo funciona bajo Microsoft Windows, utiliza arquitectura Cliente/Servidor. Constituye la alternativa a otros potentes SGBD como son Oracle, PostgreSQL o MySQL.
SYBASE	Un DBMS con bastantes años en el mercado, tiene 3 versiones para ajustarse a las necesidades reales de cada empresa. Es un sistema relacional, altamente escalable, de alto rendimiento, con soporte a grandes volúmenes de datos, transacciones y usuarios, y de bajo costo.
Otros SGBD comerciales importantes	DBASE, ACCESS, INTERBASE y FOXPRO

5.6. SGBD libres

SGBD	Descripción
MySQL	Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Distribuido bajo dos tipos de licencias, comercial y libre. Multiplataforma, posee varios motores de almacenamiento, accesible a través de múltiples lenguajes de programación y muy ligado a aplicaciones web.
PostgreSQL	Sistema Relacional Orientado a Objetos. Considerado como la base de datos de código abierto más avanzada del mundo. Desarrollado por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Es multiplataforma y accesible desde múltiples lenguajes de programación.
Firebird	Sistema Gestor de Base de Datos relacional, multiplataforma, con bajo consumo de recursos, excelente gestión de la concurrencia, alto rendimiento y potente soporte para diferentes lenguajes.
Apache Derby	Sistema Gestor escrito en Java, de reducido tamaño, con soporte multilenguaje, multiplataforma, altamente portable, puede funcionar embebido o en modo cliente/servidor.

SGBD	Descripción
SQLite	Sistema relacional, basado en una biblioteca escrita en C que interactua directamente con los programas, reduce los tiempos de acceso siendo más rápido que MySQL o PostGreSQL, es multiplataforma y con soporte para varios lenguajes de programación.

Otro SGBD libre importante es MariaDB,

7. Bases de datos centralizadas

SGBD centralizado: SGBD implantando en una sola plataforma (mainframe) desde la que se gestiona directamente la totalidad de los recursos. Los centros de procesos de datos tradicionales funcionan con esta arquitectura.

- Son tecnologías sencillas, experimentadas, robustas.
- No hay múltiples elementos de procesamiento, ni comunicaciones entre BBDD
- Se componen de: Los datos, el software de gestión de bases de datos y los dispositivos de almacenamiento asociados
- Su seguridad podría verse comprometida más fácilmente.

Ventajas	Inconvenientes
Evita redudancia (No inconsistencias, no desperdicio de espacio)	Menor poder de cómputo que una distribuida
No inconsistencia (Una sola entrada representa el hecho)	Si falla, se pierde el procesamiento y la información de todo el sistema
Seguridad centralizada	En caso de un desastre o catástrofe, la recuperación es difícil de sincronizar.
Integridad	Las cargas de trabajo no se pueden difundir entre varias computadoras, ya que los trabajos siempre se ejecutarán en la misma máquina.
Mejor rendimiento en procesamiento	Los departamentos de sistemas retienen el control de toda la organización.
Más barato	Necesario mantenimiento central de datos

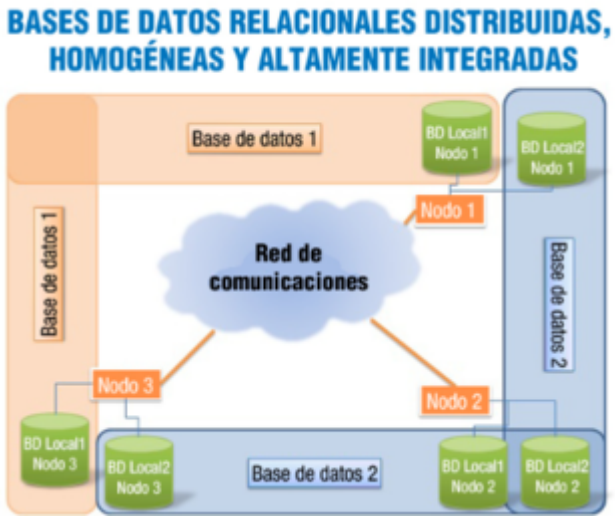
8. Bases de datos distribuidas

Base de datos distribuida (BDD): Conjunto de múltiples bases de datos lógicamente relacionadas que se encuentran distribuidas entre diferentes nodos interconectados por una red.
Sistema de bases de datos distribuida (SBDD): Sistema en el que múltiples sitios de bases de datos están ligados por un sistema de comunicaciones. El usuario en cualquier siito puede acceder a los datos de cualquier parte de la red.
Sistema gestor de bases de datos distribuida (SGBDD): Se encarga del manejo de la BDD dando un mecanismo de acceso que hace que la distribución sea transparente (inapreciable) a los usuarios.

El SGBDD trabaja a través de un conjunto de sitios o nodos con sistema de procesamiento de datos completo con base de datos local, sistema gestor de bases de datos e interconectados entre sí. Si están muy dispersos (Red WAN, Wide Area Network, Red de area amplia), si están cerca (Red LAN, Local Area Network, Red de area local)

Ventajas	Inconvenientes
Acceso y procesamiento de los datos es más rápido (varios nodos comparten carga de trabajo)	La probabilidad de violaciones de seguridad es creciente si no se toman las precauciones debidas.
Desde un lugar puede accederse a información alojada en diferentes lugares.	Complejidad añadida que es necesaria para garantizar la coordinación apropiada entre los nodos.

Ventajas	Inconvenientes
Más barato que centralizadas	La inversión inicial es menor, pero el mantenimiento y control pueden resultar costosos.
Tolerancia a fallos. Mediante la replicación, si un nodo deja de funcionar el sistema completo no deja de funcionar.	Control de concurrencia y los mecanismos de recuperación son mucho más complejos (los datos pueden estar replicados)
Se adapta más naturalmente a la estructura de las organizaciones. Permiten la incorporación de nodos de forma flexible y fácil.	El intercambio de mensajes y el cómputo adicional necesario para conseguir la coordinación entre los distintos nodos constituyen una forma de sobrecarga.
Aunque los nodos están interconectados, tienen independencia local.	Dada la complejidad del procesamiento entre nodos es difícil asegurar la corrección de los algoritmos, el funcionamiento correcto durante un fallo o la recuperación.



8.1. Fragmentación

Extraer los datos en SGBDD se hace mediante fragmentación de distintas tablas que están en diferentes lugares.
Se llama **problema de fragmentación** al particionamiento de la información para distribuir cada parte a los diferentes sitios de la red. Encontrar el nivel de particionamiento adecuado.

Debe considerarse el **grado de fragmentación** a aplicar porque afectará a las consultas. Sin fragmentación las relaciones o tablas se toman como unidad de fragmentación. También puede fragmentarse a nivel de tupla (fila, registro) o a nivel de atributo (columna, campo) de una tabla.
La fragmentación no debe ser ni nula, ni demasiado alta. Debe estar equilibrado y contemplar la casuística de las aplicaciones.

Reglas de la fragmentación
Si la relación R se descompone en R1, R2,...,Rn

- **Completitud**: cada elemento de R debe estar en algún fragmento de Ri.
- **Reconstrucción**: la reconstrucción de la relación a partir de los fragmentos debe asegurar que se preservan las restricciones
- **Disyunción**: si la relación se descompone verticalmente, sus atributos primarios se repiten en todos los fragmentos.

Tipos de fragmentación:

- **Fragmentación horizontal**: Sobre las tuplas (registros). La relación tiene subrelaciones que contienen un subconjunto de las tuplas de la primera. Existen la primaria y la derivada.
- **Fragmentación vertical:** Sobre los atributos. Tiene subconjuntos de atributos de R y la clave primaria de R. Así se particiona en conjunto de relaciones más pequeñas. La fragmentación

óptima tiene un esquema que minimiza el tiempo de consultas del usuario. Es más complicada que la horizontal porque hay gran cantidad de alternativas posibles.
- **Fragmentación híbrida o mixta:** Ambas combinadas. Primero una vertical y luego horizontal (HV). Al contrario (VH). Para representarlo se usan árboles.

9. Primeros pasos en MySQL Server

MySQL es un sistema gestor de base de datos simple y de buen rendimiento. Es software libre (GNU GPL). Tiene estabilidad y alto grado de desarrollo.
Está realizado en C/C++. Hay ejecutables para diecinueve plataformas. Hay API para C,C++,Eiffel, Java, Perl, PHP, Python, Ruby, TCL. Optimizado para múltiples procesadores. Rápido. Puede usarse como cliente servidor. Tiene una administración de usuarios y privilegios. Puede conectarse por TCP/Ip, Sockets UNIX, sockets NT, soporta ODBC.
Se descarga y se instala.

Se puede instalar después MySQL Workbench, herramientas de diseño de bases de datos que integra desarrollo de software, administración de BBDD, diseño de BBDD, creación y mantenimiento del SGBD MySQL.
Se descarga y se instala y listo.

10. Primeros pasos en Oracle Database

Oracle Database Express Edition 11g Release 2 es un sistema de bases de datos libre para desarrollo, implementación y distribución basado en Oracle.
Buen sistema para desarrolladores, para administradores de bases de datos, para proveedores independientes, estudiantes que necesiten practicar, adiestramiento, base de datos inicial...