

SISTEMAS DE FICHEROS

Conjunto de programas que prestan servicio a los usuarios finales. Surgen al tratar de informatizar los archivadores manuales que permiten un acceso más eficiente a los datos.

Inconvenientes de los Sistemas de Ficheros:

- Separación y aislamiento de datos: los datos se separan en distintos ficheros y se hace más complicado el acceso. El programador de aplicaciones es el que debe sincronizar el procesamiento de los ficheros.
- Duplicación de datos: pérdida de consistencia en los datos. Para una misma pregunta se obtienen distintas respuestas. A mayor redundancia, mayor inconsistencia.
- Dependencia de los datos: es difícil de realizar cualquier cambio en la estructura física de los datos. Al cambiar la ubicación de un dato en esta estructura, se la debe cambiar en todos los programas como este dato trabaje.
- Formato de fichero incompatible: utilización de distintos lenguajes para la creación de aplicaciones que manejan datos.
- Consulta fija y proliferación de programas de aplicación: consultas pautadas y para cada consulta se genera una nueva aplicación.

BASE DE DATOS (B.D.)

Conjunto de datos entre los cuales existen relaciones lógicas, diseñada para satisfacer los requerimientos de una organización. Aquí se encuentran los datos y su descripción (“metadatos”) los cuales se almacenan en un diccionario de datos. Por lo tanto, podemos decir que una B.D. es un gran almacén de datos donde estos se definen una sola vez, donde se tiende a minimizar la redundancia de los mismos y que es utilizada al mismo tiempo por muchos usuarios.

SISTEMAS DE BASE DE DATOS

En los Sistemas de Ficheros existían dos factores que producían inconvenientes: la definición de los datos se encuentra codificada dentro de los programas en lugar de estar en forma independiente y no hay control sobre el acceso y la manipulación de los datos más allá de lo impuesto por los programas de aplicación. En respuesta a esto surgen los Sistemas de Base de Datos.

El propósito de estos es mantener información (independiente y segura) y hacer que esté disponible cuando se la necesite. Es la combinación de programas y archivos que se utilizan conjuntamente. La información debe ser cualquier cosa que se considere importante para el individuo o la organización.

Funciones principales:

- Crear y mantener la B.D.
- Establecer y mantener el acceso a la B.D. de manera rápida y segura.
- Manejar los datos de acuerdo con las peticiones de los usuarios.
- Mantener la integridad y seguridad de los datos.
- Registrar el uso de la B.D.

Componentes de un Sistema de B.D.:

- Información o datos: son el objeto principal y, por lo tanto, ningún Sistema de B.D. puede existir sin los datos. Se trata de información integrada y compartida ya que los diferentes usuarios acceden a los mismos datos.
- Equipamiento o hardware: es el conjunto de dispositivos físicos sobre los que reside la B.D. Consiste en una o más computadoras, unidades de disco, monitores, impresoras, unidades de almacenamiento, cableado, etc.
- Programas: un Sistema de B.D. incluye varios tipos de software:
 - El Sistema de Gestión de B.D. (S.G.B.D. o D.B.M.S.).
 - El software de aplicación, que usa las facilidades del D.B.M.S. para manipular la B.D.

- Herramientas para el desarrollo de aplicaciones, como los generadores de pantallas, de menús y de informes; los generadores de aplicaciones, los compiladores y las facilidades para la definición de la vista y los datos.
- Usuarios:
 - *Administrador (D.B.A.):* se encarga del diseño físico de la B.D. y su implementación, realiza el control de seguridad y de la concurrencia, se encarga de que el sistema se encuentre operativo así como de la performance de la B.D. Debe conocer muy bien el S.G.B.D. que se está utilizando así como el hard.
 - *Diseñador:* realiza el diseño lógico de la B.D. identificando los datos, sus relaciones y restricciones. Debe conocer los datos de la empresa y sus reglas de negocios.
 - *Programadores de aplicación:* programan e implementan las aplicaciones que servirán a los usuarios finales.
 - *Usuarios:* clientes de la B.D.

Ventajas de los Sistemas de B.D.:

- Por la integración de los datos:
 - *Control de Redundancia de Datos:* la redundancia es perjudicial si no está controlada. Toda redundancia que está puesta a propósito deberá estar controlada y no deberá ser perjudicial. Si está controlada, existen políticas del D.B.A. para manejarla. No existen dos entidades redundantes pero sus datos si lo pueden ser.
Los movimientos en una B.D. se transaccionan para que exista consistencia. Una transacción puede incluir una o más acciones sobre la B.D., y se cumplen todas o fallan todas. Esto garantiza la integridad de la B.D.
 - *Consistencia de Datos:* N entradas a la misma petición deben obtener la misma respuesta. Existe una inconsistencia temporal mientras se está realizando la transacción. Completado el proceso desaparece.
 - *Compartición de Datos:* distintos usuarios (programas de aplicación o usuarios finales) comparten los mismos datos.
 - *Mantenimiento de Estándares:* cuando defino los metadatos, también los estándares. Se pueden establecer normas y cumplirlas, tanto de formato como de nombre o documentación.
Todo sistema de B.D. permite implementar un diccionario de datos que informa los distintos formatos de los atributos. También sirve para documentar en que entidad existe un atributo y qué formato tiene.
- Por la existencia del S.G.B.D.:
 - *Mejora en la Integridad de los Datos:* se refiere a la validez y consistencia de los datos. Se expresa mediante reglas que no se pueden violar. El S.G.B.D. se encarga de mantenerlas.
 - *Mejora en la Seguridad:* proporciona acceso a la B.D. solo a usuarios autorizados para protegerla.
 - *Mejora en la Accesibilidad:* el S.G.B.D. proporciona lenguajes de consulta y generadores de informes (Ej.: S.Q.L.), lo que permite a los usuarios hacer consultas sin necesidad de tener que programar una aplicación para ello.
 - *Mejora en Mantenimiento:* por independencia de datos lo que implica que estos y los programas que de ellos se sirven son mutuamente independientes, de manera que unos u otros pueden ser modificados sin afectar a los restantes. Se necesita dos tipos de independencia:
 - **Lógica:** la modificación de la estructura lógica general no afecta a los programas de aplicación.
 - **Física:** pueden modificarse la distribución y organización física de los datos sin afectar a la estructura lógica general ni a los programas de aplicación.

Es posible distinguir tres tipos de dependencias que se deben solucionar:

 - **Dependencia de los datos:** las instrucciones de los programas dependen de la definición de datos.
 - **Dependencia de la estructura:** la lógica del programa depende de las facilidades para segmentar registros, manejar ciclos anidados, proporcionar y seguir enlaces entre archivos.
 - **Dependencia del programa:** el programa está comprometido con el empleo y estructura de los datos.

- *Aumento de la Concurrencia:* muchos clientes de la B.D. pueden tener acceso a los mismos datos al mismo tiempo.
- *Mejora en la Recuperación de Fallos.*

SISTEMAS DE GESTIÓN DE BASE DE DATOS (S.G.B.D.)

Es el que permite a los usuarios definir, crear y mantener la base de datos y además proporciona acceso controlado a la misma.

Los S.G.B.D. proporcionan los siguientes servicios:

- Permite la definición de la B.D. mediante el lenguaje de definición de datos (D.D.L.), el que permite especificar la estructura, el tipo y las restricciones de los datos. Todo esto se almacena en la B.D.
- Permite inserción, actualización, eliminación y consulta de datos mediante el lenguaje de manejo de datos (D.M.L.)
- Proporciona un acceso controlado a la B.D. mediante:
 - *Sistema de Seguridad:* acceso sólo a usuarios autorizados.
 - *Sistema de Integridad y Consistencia:* se refiere a la validez y consistencia de los datos.
 - *Sistema de Control de Concurrencia:* distintos usuarios acceden simultáneamente a la B.D.
 - *Sistema de Control de Recuperación:* permite reestablecer la B.D. ante una falla en el hard o soft.
 - *Diccionario de Datos:* contiene la descripción de los datos de la B.D.

ARQUITECTURA DE BASE DE DATOS

El objetivo de presentar una arquitectura es establecer un marco de referencia útil para describir los conceptos generales de B.D. y explicar la estructura del sistema.

La arquitectura usada es la de tres niveles:

- Nivel externo: es la que se ocupa de la forma como los usuarios perciben los datos. Todos los usuarios (aplicaciones, terminales en línea, personas, etc.) interactúan con la B.D. a través de este nivel siendo el D.B.A. un caso particular ya que es capaz de ver los tres niveles. Cada usuario dispone de un lenguaje (programadores de aplicación con lenguaje de 4ª generación que es provisto por cada Sistema de B.D. y usuarios a través de lenguaje de consulta o aplicación). Cada lenguaje posee un sublenguaje de datos (alguien que se ocupe de los objetos y operaciones de la B.D.). Este está inmerso en el lenguaje anfitrión (S.Q.L.). Cualquier sublenguaje de datos es una combinación de por lo menos dos subordinados (un Lenguaje de Definición de Datos – DDL y un Lenguaje de Manipulación de Datos – DML).
- Nivel interno: es el más cercano al almacenamiento físico, es el que se ocupa de cómo se almacenan físicamente los datos. Se define con un DDL interno que nos indica como es la correspondencia entre la parte conceptual y la B.D. física.
- Nivel Conceptual: es el nivel de mediación entre los otros dos. Así como el externo se ocupa de las vistas individuales de los usuarios, puede considerarse al conceptual como una vista comunitaria de todos los usuarios. Existen muchas vistas externas que representan cada una alguna parte de la B.D. total, una vista conceptual que representa a la B.D. (y todo lo que está almacenado en ella) y una vista interna que representa como está almacenada físicamente la B.D. La vista conceptual se define mediante un esquema conceptual y se describe con un DDL conceptual debiendo implicar sólo definiciones de contenido (no de almacenamiento ni técnicas de acceso) para lograr la independencia de los datos.

Hay dos niveles de correspondencia, uno entre los niveles externo y conceptual y otro entre el conceptual y el interno. Las alteraciones (modificación en la definición de la estructura de almacenamiento) deberán aislarse por debajo del nivel conceptual a fin de mantener la independencia de los datos.

GRÁFICO DE LA ARQUITECTURA DE 3 NIVELES

SISTEMA DE ADMINISTRACIÓN DE BASE DE DATOS (D.B.M.S.)

Es el conjunto de programas que maneja todo el acceso a la B.D., el D.B.M.S. interpreta y procesa las peticiones del usuario para recobrar la información de la B.D. Constituye la interfaz entre el usuario y el Sistema de Base de Datos. Proporciona la flexibilidad en el almacenamiento y recuperación de los datos y producción de la información.

EL D.B.M.S. es el que permite a los usuarios definir, crear y mantener la B.D. y además proporciona acceso controlado a la misma.

Sucedo lo siguiente:

- Un usuario solicita acceso empleando un sublenguaje de datos determinado.
- El D.B.M.S. analiza e interpreta la solicitud.
- Inspecciona en este orden lo siguiente: desde esquema externo hasta definición de la estructura de almacenamiento pasando por las dos correspondencias y es esquema conceptual.
- Ejecuta las operaciones necesarias sobre la B.D. almacenada.

Funciones del D.B.M.S.:

- Definición de datos: debe ser capaz de aceptar definiciones de datos tales como el esquema externo, conceptual, interno y sus correspondientes correspondencias.
- Manipulación de datos: debe ser capaz de atender todas las solicitudes del usuario para incorporar, ver o modificar datos.
- Seguridad e integridad de los datos: debe ser capaz de supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad.
- Recuperación y concurrencia de los datos: debe cuidar de los controles de recuperación y concurrencia.
- Debe proveer un diccionario de datos: catalogo accesible para el usuario que contiene la descripción de los datos de la B.D.

- Desempeño: deberá ejecutar todas las funciones de la manera más eficiente.

Funciones del Administrador de Base de Datos (D.B.A.):

- Definir el esquema conceptual: es tarea del D.B.A. decidir con exactitud cual es la información que debe mantenerse en la misma, es decir, identificar las entidades que interesan a la empresa y la información que debe registrarse acerca de esas entidades.
- Definir el esquema interno: el debe decidir también como se representará la información en la B.D. almacenada. A este proceso suele llamarsele diseño físico de la B.D. Una vez hecho esto el D.B.A. deberá crear la definición de estructura de almacenamiento correspondiente (es decir el esquema interno); además deberá definir la correspondencia pertinente entre los esquemas interno y conceptual.
- Definir el esquema externo: el D.B.A. debe encargarse de la comunicación con los usuarios, garantizar la disponibilidad de los datos que se requieren y escribir los esquemas externos necesarios. Además, será preciso definir la correspondencia entre cualquier esquema externo y el esquema conceptual.
- Definir las verificaciones de seguridad e integridad: estas pueden considerarse parte del esquema conceptual.
- Definir procedimientos de respaldo y recuperación: cuando una empresa se decide a utilizar un Sistema de B.D., se vuelve dependiente del funcionamiento correcto de ese sistema. En caso de que sufra daño cualquier porción de la B.D. resulta esencial poder reparar los datos implicados con un mínimo de retraso y afectando lo menos posible al resto del sistema. El D.B.A. debe definir y poner en práctica un plan de recuperación adecuado.
- Supervisar el desempeño y responder a cambios en los requerimientos: es responsabilidad del D.B.A. organizar el sistema de modo que se obtenga el desempeño que sea “mejor para la empresa”, y realizar ajustes apropiados cuando cambien los requerimientos.

MODELO RELACIONAL

El modelo relacional está basado en dos ramas de las matemáticas: la teoría de conjunto y la lógica de predicados.

Este modelo representa la 2^{da} generación, en él todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico puedan tener una estructura completamente distinta. Utiliza tablas bidimensionales, llamadas relaciones para mostrar como se vinculan lógicamente los datos de un registro. Los renglones de las tablas representan los registros y las columnas muestran los atributos de la entidad.

El orden de los datos en la tabla no es significativo y tampoco implica un orden cuando los registros están incluidos en una relación. El enlace de los datos relacionales padre e hijo se establecen vía un campo común presentes en ambas relaciones.

La ventaja principal del enfoque relacional está en la simplicidad de la representación de la estructura lógica y en la flexibilidad para establecer relaciones de datos por medio de campos de conexión. En una B.D. relacional todas sus tablas deben estar normalizadas.

Este modelo tiene que ver con tres aspectos de los datos: la estructura, la integridad y el manejo de los mismos.

Estructura de datos relacionales:

- Relaciones: se representan gráficamente como una tabla donde las filas corresponden a registros individuales y las columnas a atributos de estos registros. Se utilizan para almacenar información sobre los objetos que se representan en una B.D.
- Atributo: es el nombre de una columna de una relación.
- Tuplas: son las filas de una relación o bien las instancias de una entidad.
- Dominio: es el conjunto de valores posibles para uno o varios atributos, los dominios permiten que se definan en un lugar común el significado y los valores que los atributos pueden tomar; esto hace que haya más información disponible para el sistema, pudiendo evitar relaciones incorrectas.
- Grado: es el número de atributos de una relación.
- Cardinalidad: es el número de tuplas de una relación.

Formalmente, una relación R definida sobre un conjunto de dominios D_1, D_2, \dots, D_n consta de:

- **Cabecera:** conjunto fijo de pares ordenados (atributo, dominio) donde cada atributo A_j se corresponde a un único dominio D_j y todos los A_j son distintos, es decir, no hay atributos repetidos o atributos que se llamen igual. El grado de la relación R es "n".
 $\{(A_1:D_1), (A_2:D_2), \dots, (A_n:D_n)\}$ Grado "n"
- **Cuerpo:** conjunto variable de tuplas donde cada tupla es un conjunto de pares (atributo, valor) con $i=1,2,3,\dots,m$ donde "m" es la Cardinalidad de la relación.
 $\{(A_1:V_{11}), (A_2:V_{12}), \dots, (A_n:V_{in})\}$ donde $i = 1, 2, \dots, m$ Cardinalidad "m".
 Para cada par $(A_j:V_{ij}) \Rightarrow V_{ij} \in D_j$.

Objetivos:

- **Independencia física:** es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos); es decir, el modo en que se almacenan los datos no influye en la manipulación lógica de los mismos. Por lo tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico.
- **Independencia lógica:** es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la B.D. o para reducirla, pero no repercute en los programas y/o usuarios que estén accediendo a conjuntos parciales de los mismos (vistas).
- **Flexibilidad:** se le puede presentar al usuario los datos en la forma en que éste los prefiera o los requiera.
- **Uniformidad:** las estructuras lógicas de los datos presentan aspecto uniforme, lo que facilita la concepción y manipulación de la B.D. por parte de los usuarios.
- **Sencillez:** las características anteriores y los lenguajes sencillos de utilizar producen como resultado que el modelo de B.D. Relacional sea fácil de comprender y utilizar por parte del usuario final.

Propiedades:

- Cada relación tiene un nombre y este es distinto de todas las demás.
- En cada tupla, cada atributo toma un solo valor (Atómicos).
- No hay atributos que se llamen igual.
- El orden de los atributos no importa, los atributos no están ordenados.
- Cada tupla es distinta a las demás.
- El orden de las tuplas no importa.

Tipos de relaciones:

- **Base:** son relaciones reales que tienen nombre y forman parte directa de la B.D. almacenada.
- **Vistas:** son relaciones virtuales con nombres y derivadas. Se representan en términos de otras relaciones. No poseen datos almacenados propios.
- **Instantáneas:** a diferencia de las vistas son reales, no virtuales. Están representadas por su definición y datos almacenados.
- **Resultado de consulta:** son relaciones de alguna consulta específica, pueden o no tener nombre y no persisten en la B.D.
- **Resultado intermedios y temporales:** son relaciones que utiliza el sistema de B.D. para resolver operaciones.

Claves:

- **Superclave:** es un conjunto de atributos que identifican de modo único a las tuplas de una relación. La mayor superclave es la suma de todos los atributos.
- **Claves candidatas:** es un conjunto no vacío de atributos que identifican unívoca y mínimamente a cada tupla. Toda relación tendrá como mínimo una clave candidata. Esta debe cumplir con:
 - **Unicidad:** no existen dos tuplas que posean el mismo valor en la clave candidata.
 - **Minimalidad:** no se puede eliminar ningún atributo de la clave candidata (si es una clave compuesta) sin que se pierda la propiedad de unicidad.
- **Clave primaria:** aquella clave candidata que el usuario elegirá, por consideraciones ajenas al modelo relacional, para identificar las tuplas de una relación. Si la clave primaria es compuesta, ésta debe ser no nula en su totalidad, es decir, ninguno de sus atributos puede ser nulo.

- Claves alternativas: aquellas claves candidatas que no han sido elegidas como clave primaria.
- Claves Ajenas: una clave ajena es un conjunto de atributos de una relación R2 cuyos valores, o son completamente nulos o coinciden con los de la clave primaria de una relación R1. La relación donde se encuentra la clave ajena se denomina relación referencial y la relación donde se encuentra la clave primaria es la relación referida u objetivo. A su vez, una relación referida puede ser relación referencial en otra relación y viceversa.
Las claves ajenas pueden admitir nulos (ausencia de información) a diferencia de las claves primarias.

Reglas de integridad:

- Reglas de integridad para las entidades: ningún componente de la clave primaria de una relación puede aceptar nulos; es decir, en una B.D. Relacional no se puede almacenar información sobre algo que no se puede identificar.
- Reglas de integridad referencial: toda clave ajena debe existir en el dominio de la clave primaria a la que está referenciando, es decir, no puede haber valores de claves ajenas sin concordancia con valores de la claves primarias. Para cualquier valor no nulo de la clave ajena existe un valor asociado de la relación objetivo.
- Barrado de claves primarias:
 - *Restringido*: no se puede borrar una tupla si esta referenciada.
 - *Propagación o cascada*: se borra la tupla junto con las tuplas que la están referenciando.
 - *Valor nulo*: se borra la tupla y se pone valor nulo a todas las claves ajenas que la están referenciando.
- Modificación de claves primarias:
 - *Restringido*: no se puede modificar una tupla si esta referenciada.
 - *Propagación o cascada*: se modifica la tupla junto con las tuplas que la están referenciando.
 - *Valor nulo*: se modifica la tupla y se pone valor nulo a todas las claves ajenas que la están referenciando.

DISEÑO DE BASE DE DATOS – MODELO ENTIDAD - RELACIÓN (M.E.R.)

El diseño como actividad se puede entender en distintos niveles de abstracción, separándolo en diseño conceptual, diseño lógico y diseño físico: <http://doblev.wordpress.com>

El diseño conceptual es el que no se encuentra asociado con ninguna plataforma de implementación, sino que es el que modela el problema a solucionar.

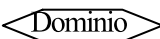


El diseño lógico se acerca más a la implementación del producto en una plataforma computacional tomando consideraciones de la plataforma en la cual va a ser implementado.

El diseño físico es la traducción del diseño conceptual en físico (realidad).

El modelo de datos debe considerar:

- Las entidades.
- Las relaciones entre las entidades.
- Las restricciones del modelo.
- Las suposiciones que se hicieron para modelar esta realidad.
- Se deben incorporar atributos.
- Se definen: dominios, identificadores, operaciones permitidas, entidades débiles y entidades fuertes.

Elementos del M.E.R.:

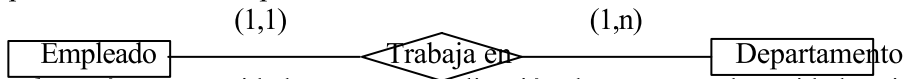
- Dominio: conjunto de valores de un mismo tipo. 
- Atributo: algo que caracteriza a una entidad y está comprendido en su dominio. Cuando un atributo es identificador debe cumplir con las características de unicidad y minimalidad. 
- Atributo Compuesto: corresponde al grupo de atributos que tiene afinidad en cuanto a su significado o al uso. Ej.: Domicilio = Calle Número Localidad.
- Tipos de entidad: representan clases de objetos de la realidad, además se componen de atributos los cuales representan las características de un tipo de entidad. Ej.: persona, facturas. 

- **Identificador:** es algo que identifica de manera única a la entidad (cumple con la condiciones de unicidad y minimalidad). Puede ser un atributo o una combinación de ellos. Si el atributo pertenece a la entidad, es fuerte, si es externo, es débil. ● →
- **Tipos de interrelación:** representan agregaciones de dos o más entidades no necesariamente diferentes. Un identificador de un tipo de interrelación se forma a partir de los identificadores de los tipos de entidad que relaciona.



Extensiones al M.E.R.:

- **Cardinalidad de asignación:** caracteriza a los atributos de un tipo de entidad y a los tipos de interrelación.
- **Cardinalidad de atributo con respecto a un tipo de entidad:** para los atributos de Cardinalidad mínima indica el número mínimo de valores de un atributo asociado con cada ocurrencia de una entidad o interrelación. Ídem para Cardinalidad máxima.
- **Cardinalidad de tipo de entidades con respecto a un tipo de interrelación:** para los tipos de interrelación, la cardinalidad mínima, máxima establece el menor, mayor número de correspondencias de cada tipo de entidad involucrada en la interrelación.

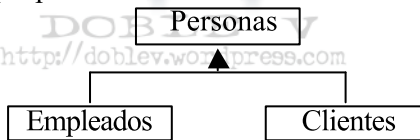


- **Generalización:** una entidad es una generalización de un grupo de entidades si cada una de éstas (subconjunto) es a su vez del tipo de entidad que generaliza.

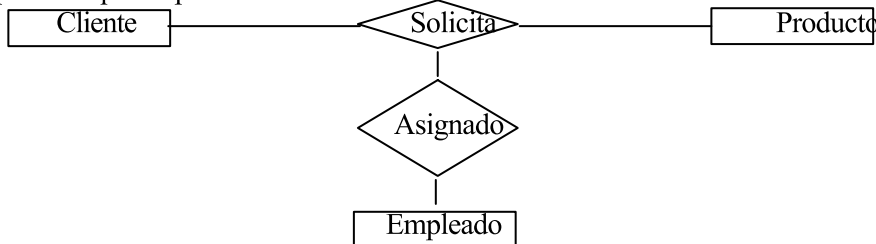
Se caracteriza por su cobertura, la cual puede ser total o parcial y exclusiva o superpuesta.

- **Total y exclusiva:** todas las personas son empleados o clientes del banco, pero no ambas simultáneamente.
- **Total y superpuesta:** todas las personas son empleados o clientes, permitiéndose que un empleado sea cliente.
- **Parcial y exclusiva:** hay personas, algunas de las cuales son empleados o clientes.
- **Parcial y superpuesta:** no todas las personas están clasificadas como empleados o clientes, y hay empleados que pueden ser clientes.

DOB
http://doblev.wordpress.com



- **Agregación:** una relación y las entidades que relaciona puede ser manejada como una entidad, lo cual permite que se pueda relacionar con otras entidades.



Estrategia para modelar con M.E.R.:

- Identificar los tipos de entidad y las relaciones entre ellas.
- Descomponer cada entidad para descubrir generalizaciones o agregaciones.
- Verificar si no se subdividieron las relaciones.
- Identificar los atributos, ya sea de entidades o relaciones.
- Definir los identificadores para las entidades.
- Definir las restricciones de cardinalidad y cobertura.
- Verificar el esquema resultante. Si no representa la realidad que quiero describir, se deben realizar modificaciones.

LENGUAJE DE CONSULTA ESTRUCTURADA (S.Q.L.)

Es un lenguaje de B.D. normalizado utilizado por las B.D. para operaciones sobre los datos o sobre la estructura de los datos. El lenguaje está compuesto por comandos, cláusulas, operadores y funciones de agregado; estos elementos se combinan en las instrucciones para crear, actualizar y manipular la B.D.

Comandos:

- DLL: son los que permiten crear y definir nuevas B.D., campos e índices. CREATE-DROP-ALTER
- DML: permite generar consultas, ordenar, filtrar y extraer datos de la B.D. SELECT-INSERT-UPDATE-DELETE.

Cláusulas:

- Son condiciones utilizadas para definir que datos se desean seleccionar o manipular. FROM-WHERE-GROUP BY-HAVING-ORDER BY.

Operadores:

- Lógicos: AND-OR-NOT.
- Comparación: <, >, <>, <=, >=, =.
- Otros: BETWEEN-LIKE-IN.

Funciones de agregado:

- Se usan dentro de la cláusula SELECT para devolver un único valor para el grupo seleccionado. AVG-COUNT-SUM-MAX-MIN.

Consultas de selección:

- Se utilizan para indicar al motor de datos que devuelva información sobre la B.D. Esta información es devuelta en forma de un conjunto de registros, este conjunto de registros es modificado. La mayoría de los motores lo llama "recordset".

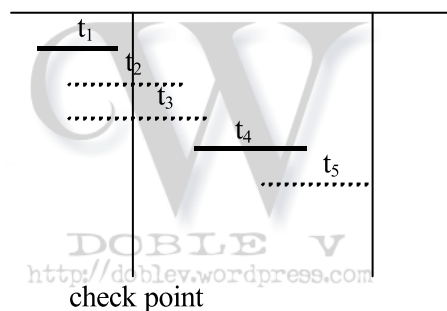
SEGURIDAD Y PROTECCIÓN DE LOS DATOS

La protección de los datos debe realizarse contra fallos físicos, lógicos y humanos. En este contexto estudiamos los temas de recuperación, concurrencia, seguridad e integridad. El D.B.M.S. proporciona mecanismos para prevenir fallos, estos son los subsistemas de corrección.

Recuperación:

- Objetivo: es el de proteger a la B.D. contra fallos lógicos o físicos. Independientemente de los fallos estos pueden afectar a dos aspectos de almacenamiento, como son:
 - Fallos que provocan la pérdida de memoria volátil.
 - Fallos que provocan la pérdida de memoria secundaria.
- Transacciones: para asegurar que la B.D. este siempre consistente se crean unidades de ejecución llamadas transacciones que pueden definirse como secuencias de operaciones que han de ejecutarse en forma atómica, es decir, se realizan todas las operaciones que comprende la transacción o no se realiza ninguna.
- Administrador de Transacciones (A.T.): es el componente encargado de lograr la atomicidad, donde las operaciones de commit y roll back son las claves de su funcionamiento.
 - *Commit*: indica el éxito de la transacción o sea le dice al A.T. que ha terminado con éxito una etapa de trabajo. LA B.D. debe estar de nuevo en un estado consistente por lo cual se pueden hacer permanentes todas las modificaciones efectuadas en esa unidad de trabajo.
 - *Roll back*: indica el éxito no dado de la transacción y le señala al A.T. que la B.D. puede estar inconsistente y que todas las modificaciones deben ser retrocedidas y anuladas (hasta que la B.D. recupere el estado anterior).
- Características de una transacción:
 - *Atomicidad*: se ejecutan todas las transacciones o ninguna.
 - *Preservación de la consistencia*.
 - *Aislamiento*.
 - *Persistencia*: realizada la transacción los datos quedan en la B.D.
 - *Seriabilidad*.

- Métodos para anular y recuperar transacciones:
 - *Un archivo diario o Log:* es el método más usado y consiste en guardar toda la información necesaria para deshacer en caso de fracaso o rehacer en caso de recuperar las transacciones. Este archivo consta básicamente de:
 - Un identificador de la transacción.
 - La hora de la modificación.
 - La identificación del registro afectado.
 - El tipo de acción que se llevó a cabo.
 - El valor anterior del registro.
 - El nuevo valor del registro.
 - Otras informaciones.
 - *Dos archivos Log:* uno con la imagen posterior y el otro con la anterior. Usualmente el archivo de Log una vez lleno va eliminando registros por tratarse de un espacio fijo. Según van entrando nuevos registros, de acuerdo al grado de actualización dependerá el espacio reservado.
 - *Check point:* un punto de revisión me posibilita no tener que recorrer todo el archivo de Log ante una falla. Establecerlo implica:
 - Grabar el contenido en la B.D.
 - Grabar físicamente un registro donde figure un punto de revisión donde figure específicamente el archivo de Log.
 Ej.: un movimiento de cuenta bancaria.
- Procedimiento que debe realizar el sistema al reiniciarse: toma las transacciones del archivo de Log y los carga en una lista de anular y crea otra llamada repetir. Examina el archivo de Log a partir del último check point (recuperación en caliente).



La recuperación en frío consiste en disponer de un back up que respalda la B.D. que permitirá junto con los archivos de Log que se han ido produciendo, reconstruir la B.D. para dejarla consistente. Es el D.B.A. el que debe definir procedimientos, situaciones y plazos en los que se debe realizar las copias de seguridad y el archivo de Log especificando a los operadores cuales son los procedimientos de recuperación ante la caída del sistema. Un error fatal se produce si se pierde el archivo de Log para poder recuperar el archivo y la B.D.

Concurrencia:

- Objetivo: administrar la concurrencia de los datos debido a que estos pueden estar accedidos al mismo tiempo por dos o más transacciones y estas pueden ir produciendo modificaciones temporales que dejan a la B.D. inconsistente.
- Técnicas de control concurrente - Técnicas de bloqueo: se produce un bloqueo cuando se impide a otros usuarios la recuperación o actualización de los elementos bloqueados para evitar la inconsistencia en los elementos concurrentes. Los D.B.M.S. proveen bloqueos por tupla o tabla completa.
 - *Bloqueos exclusivos:* cuando una transacción mantiene al objeto y ninguna otra puede acceder al mismo. Recién es libre cuando está la libera. Este tipo de bloqueo se requiere para actualizar datos.
 - *Bloqueos compartidos:* más de una transacción tiene el objeto bloqueado pero no de forma exclusiva. Se utiliza como lectura pero no para actualización.
- Problemas de las técnicas de bloqueo: pueden producirse abrazos mortales cuando dos o más transacciones está esperando que una o la otra libere el objeto a seguir. Se da a nivel de tupla o campo. Las soluciones que hay son:

- Prevenir el abrazo mortal obligando a que las transacciones bloqueen todos los elementos que necesiten por adelantado (al inicio). Si no puede bloquear los elementos, no bloquea ninguno y decide esperar para volver a intentarlo.
- Detectando el abrazo mortal se controla en forma periódica si este se ha construido. Se construye un grafo de espera donde cada nodo es una transacción y un arco de la transacción t_i a t_j en el caso de que t_i este esperando un elemento que espera t_j ; verificando el grafo, si existe un ciclo estamos en presencia de un abrazo mortal. La solución está en escoger transacciones víctimas y eliminarlas hasta que desaparezca dicho abrazo.
- Granularidad: los D.B.M.S. pueden bloquear tuplas, tablas, B.D. total y otros. A esto se le llama granularidad. Este concepto implica gestionar un menor número de bloqueos pero si ésta es muy fina, existe mayor concurrencia y con esto mayores posibilidades de abrazo mortal.

Integridad:

- Objetivo: el Subsistema de Integridad de un D.B.M.S. debe detectar y corregir en lo posible las operaciones incorrectas escapando a estos las posibles fallas de corrección, Ej.: introducir una fecha de nacimiento 05/04/45 cuando en realidad es 05/04/54.
 - Operaciones semánticamente incorrectas: son las que transgreden las restricciones que ha definido el administrador al diseñar la B.D.
 - Restricciones estáticas:
 - Restricciones sobre dominio. Ej.: el dominio del atributo edad está entre 18 y 65 años.
 - Restricciones sobre atributo. Ej.: la edad de los empleados ingresantes no debe ser mayor a 21 años.
 - Restricciones dinámicas: Ej.: el sueldo de un empleado no puede disminuir.
 - Restricciones simples: se aplica a una ocurrencia de un atributo con independencia de los demás. Ej.: el sueldo de un empleado debe ser menor que 600.
 - Restricciones compuestas: son las que implican más de una ocurrencia, como en el caso de restricciones de comparación (Ej.: el sueldo de un empleado debe ser menor que el de su jefe) o en el caso de comparaciones llamadas de globalidad (Ej.: el sueldo medio de los empleados de un departamento debe ser menor que 25000).
- Todos los S.G.B.D. deben ofrecer en su lenguaje de definición, facilidades que permitan describir las restricciones. Ej.: CREATE DOMAIN-CREATE INTESRITY RULE.
- Componentes:
 - Las restricciones propiamente dichas, o sea la condición que deben cumplir los datos.
 - La respuesta a la trasgresión, o sea especificar las acciones a tomar como rechazar la operación, informar al usuario, etc.
 - La condición de disparo, o sea cuando debe desencadenarse la operación especificada. Esto es antes, durante o después de cierto evento.
 - Ventajas: las reglas de integridad deben almacenarse en el diccionario de datos como parte integrante de los datos de modo que no han de incluirse en los programas. Esto trae estas ventajas:
 - Las reglas de integridad son más sencillas de mantener y entender.
 - Se detectan mejor las inconsistencias.
 - Se protege mejor la integridad ya que ningún usuario podrá escribir un programa que las transgreda.
 - Funciones:
 - Comprobar la coherencia de las reglas que se definen.
 - Controlar las distintas transacciones y detectar las trasgresiones de integridad.
 - Cuando se produce una trasgresión, ejecutar las acciones pertinentes.

Seguridad:

- Objetivo: proteger a la B.D. contra accesos no autorizados.
- Se incluyen los siguientes aspectos:
 - Legales, sociales y éticos.
 - Políticas de la empresa.
 - Niveles de información pública y privada.
 - Controles de tipo físico. Ej.: acceso a las instalaciones.
 - Identificación del usuario. Ej.: voz, retina de ojo, sistema biométrico, etc.

- Controles de Sistemas Operativos.
- Tipos de usuarios: el S.G.B.D. debe mantener información de todos los usuarios, su tipo, los accesos y operaciones permitidas a estos.
 - *El D.B.A.:* se le permiten todas las operaciones, conceder privilegios y establecer usuarios.
 - *Usuarios:* unos con derecho a crear, borrar y modificar objetos y que además pueden conceder privilegios a otros usuarios sobre los objetos que ha creado. Otros con derecho a consulta y actualización sin derecho a crear o borrar objetos.
- Mecanismos: un mecanismo de seguridad que ofrecen los S.G.B.D. es el de entregar información a los usuarios a través de las vistas.

