**Base de datos**: conjunto de datos organizados para que pueda extraerse información y pueda ser compartida.

**Ventajas:**

* **Reducen redundancias:** reduce y centraliza la información para que no haya información de más.
* **Evitan las inconsistencias:** evita que el dato se repita con valores diferentes. Sin redundancia no hay inconsistencia.
* **Propagan actualizaciones:** se replica la modificación en donde esté repetido el dato
* **Comparten datos:** permite que diferentes usuarios utilicen los datos, implica mayor seguridad para toda la organización respecto a la integridad de los datos.
* **Pueden hacerse cumplir las normas:** al plasmarlo en las bases de datos es más fácil hacerlos cumplir. Limita el acceso haciendo que se cumplan los procesos de la organización.
* **Pueden aplicarse restricciones de seguridad:** pueden aplicarse diferentes restricciones de seguridad para el control de la operación de los datos y su uso.
* **Puede conservarse la integridad:** se permite instalar procesos de validación de los datos comunes. También se realizan controles para impedir que el usuario tenga errores de carga u operación de los datos.
* **Pueden equilibrarse requerimientos contradictorios:** se revisan los procesos, solucionan los conflictos entre integrantes o superposiciones de responsabilidades que puedan surgir. Se da una mejor a nivel administrativo.

**Componentes**:

* **Hardware**: evoluciona en velocidad de procesamiento y en la capacidad de almacenamiento de datos.
  + - **Almacenamiento Primario**:
      * **Memoria Caché**: medio de almacenamiento más cotoso y veloz, es gestionado por el HW
      * **Memoria Principal**: rápida y muy importante para las BD ya que para poder operar con grandes bloques de datos es muy conveniente hacerlo en MP.
    - **Almacenamiento Secundario:**
      * **Memoria Flash**: muy rápida para las operaciones de lectura y mucho más lenta para las operaciones de escritura. La información no se borra si hay un corte de energía.
      * **Discos Magnéticos**: tiene los más elevados niveles de confianza para almacenar los datos. Cualquiera sea la operación que se haga debe ser persistida en uno o varios discos magnéticos. Las BD nacen en este medio ya que fueron los primeros en brindar confiabilidad y acceso directo a los datos.
      * **Discos Ópticos**: se usan como back up y algunos permiten operaciones de lectura y escritura.
    - **Almacenamiento Terciario**:
      * **Cintas Magnéticas**: son medios de almacenamiento de resguardo por su confiabilidad, bajo precio y gran capacidad de almacenamiento. Su acceso secuencial a los datos los invalida para un uso operativo. Hay cintas analógicas y cintas digitales.

**Software**: DBMS (sistema de administración de base de datos). Está encargado de crear y operar una BD. Permite la interacción con los sistemas y aplicaciones que usan la BD y posee un lenguaje de programación propio (SQL). Sistema de mantenimiento de registros basado en computadoras, para registrar y mantener información. Objetivo: centralizar los datos en un mismo lugar y compartirlos con todos los usuarios.

Características y funciones relacionadas:

* **Escalabilidad**
  + Escalabilidad Horizontal: capacidad para incrementar cantidad de usuario y/o estaciones “clientes” en la red y así obtener mejoras.
  + Escalabilidad Vertical: capacidad para incrementar tamaño y/o potencia del “servidor” y así obtener mejoras.
* **Portabilidad:** DB2 UDB es la base de datos multiplataforma.
  + Sistemas grandes (mainframes): OS/390, MVS/ESA, VM, VSE
  + Sistemas de rango medio: OS/400, sistemas basados en UNIX, Windows Server
  + Sistemas para redes locales: Linux, Windows.
* **Rendimiento**
* **Disponibilidad:** que la misma esté disponible cada vez que sea necesario para múltiples usuarios a la vez.
* **Universalidad:** que la información esté disponible en distintas formas de representación dependiendo de la necesidad del usuario.
* **Usuarios**: son la finalidad del servicio.
  + - **Programador de aplicaciones**: necesitan acceso a los datos y conocimientos de sus estructuras para desarrollar las aplicaciones para usuarios finales.
    - **DBA**: usuario más importante de un DBMS y tiene una serie de responsabilidades y funciones que se detallan a continuación.
    - **Usuario final:** acceden a los datos de la BD a través de aplicaciones o sistemas o acceden mediante interfaces directas y obtener reportes mediante comandos de SQL.

**Dato:** representación simbólica de un atributo o variable cuantitativa o cualitativa de un mensaje. Deber ser objetivo. Integrados: reunidos en un mismo lugar y con la menor redundancia posible

* **Compartidos**: deben poder ser accedidos para su lectura o actualización por parte de todos los usuarios de la organización de acuerdo con los permisos que se le otorguen.
* Surge la necesidad de determinar quiénes pueden actualizar cada uno de los datos y quiénes pueden leerlos solamente.
* **Datos persistentes**: se agrupan de entidades, se unen por relaciones. Forman parte de las estructuras de almacenamiento permanentes de la base de datos.
* **Datos transitorios**: no se incluyen (ejemplo, datos de E/S). Se producen a partir de los datos permanentes y sirven para su uso por parte de los usuarios en forma directa o a través de aplicaciones informáticas.
* **Metadatos**: conjunto de datos que describen a los datos y a las estructuras que lo soportan. Descripción de un marco significativo básico para que los datos comiencen a tener sentido y ser funcionales a los sistemas de información donde van a ser usados.

**Información:** son datos con significado (dentro de un marco significativo) que se perciben al recibir el mensaje. Puede ser subjetivo. Es el grado de disminución de la incertidumbre. DATO + PROCESO = INFORMACION

**Jerarquía del Conocimiento/ DIKW**: la información se define en términos de datos. Conocimiento se define en términos de información. Sabiduría en términos de conocimiento. 1. Datos, 2. Información, 3. Conocimiento, 4. Sabiduría.

**Enfoques de Base de Datos**:

* **Sistema de Archivos**: almacenan registros constituidos por campos. Los registros y los campos eran de longitud variable y tanto el almacenamiento como la recuperación eran secuenciales. Permitía a cada dependencia guardar sus datos, pero no integrarlos a una Base de Datos única. Conceptos: archivos, registros y campos. Los datos están repetidos en varios lugares, desactualizados y con mucha gente que controle y arregle eso.
* **Enfoque Jerárquico**: estructura de árbol por medio de enlaces de padre-hijo. Tiene acceso rápido por clave primaria. Poco eficiente para otros accesos. No hay enlaces cruzados. No pueden existir datos o ramas sin raíz. Tiene redundancias no controladas. Están en un solo archivo. Se generaba un archivo único donde se iban agregando nodos y enlaces creciendo continuamente su tamaño y complejidad. ENLACES: SE RELACIONAN TODOS LOS NODOS CON TODOS, LOS PUNTEROS SON SOLO DE UNA IDA (tienen solo una dirección y son posiciones físicas de memoria). Se produce una explosión combinatoria de enlaces.
* **Enfoque de Red**: tiene una estructura de red. Posee enlaces cruzados, su performance decae debido a la explosión combinatoria de enlaces. Son más rápidos para determinados accesos, pero los requerimientos de los usuarios aumentaron dramáticamente. Tiene más posibilidades de búsqueda.
* **Enfoque Relacional**: tiene una estructura de datos (tabla o relación). Los enlaces se reemplazan por campos índices. Hay redundancia admitida y controlada por el modelo. Admite múltiples claves de acceso e índices. Permite varios archivos. Su base matemática es el álgebra relacional, sus operaciones son reversibles. Tiene integridad referencial.   
  **Consulta**: es como una foto, se ven los datos del momento en el que se realizó. Es estática

**Vista**: es como un video, porque se ven los cambios realizados, se van actualizando. Es dinámica

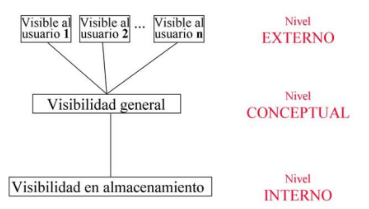
**Características de una tabla relacional:**

* Cada tabla tiene un nombre distinto a las demás
* Los valores de los atributos son atómicos, es decir, tienen un valor.
* No hay dos atributos o columnas iguales en una misma tabla
* Cada tupla/registro o fila es distinta a las demás y no tienen un orden determinado
* Grado: cantidad de atributos
* Cardinalidad: cantidad de tuplas/filas
* Las relaciones son lógicas, los enlaces físicos solamente ocurren para la conexión de claves primarias.
* Restricciones: tuplas diferentes, atributos homogéneos (valores del mismo dominio), celdas mono valuadas.
* **Orientación a Objetos**:
  + Posee mayor encapsulamiento de entidades y procedimientos: se busca encapsular en cada objeto el conjunto de los datos que le son propios y de todos los métodos que hacen a su actividad.
  + Utiliza una jerarquía de objetos: constituye un ahorro en esfuerzo de programación pues pueden existir clases básicas que son adaptadas como subclases para su uso en distintas soluciones de programación.
  + Posee herencia: propiedades y métodos de las mismas que son transmisibles por herencia.
* **Enfoque Relacional con OO**: los métodos referentes a los datos se aplican a la clase que equivale a una tabla en la BD y los métodos que hacen las reglas de negocios se desarrollan en las aplicaciones siguiendo el paradigma de programación en tres capas.

**Arquitectura ANSI-SPARC:**

**CAPAS:** Le oculta al usuario la complejidad de los sistemas ofreciendo una interfaz más amigable y protege el nivel físico de las incursiones o errores de los programadores bien/mal intencionados. Cada capa ofrece un limitado de conjuntos de comando a la capa superior para que ésta le transmita instrucciones.

**Niveles de la arquitectura ANSI-SPARC**:

* **Nivel externo** (usuarios): Capa de usuarios, visible al usuario. Comprende las aplicaciones y formularios que permiten a los mismos acceder y usar los datos de la BD.
* **Nivel lógico** (conceptual): Capa de negocios, visibilidad general a la BD. Comparte con otros sistemas de BD una visión de datos que es común, a través de diagramas lógicos que constan de tablas y relaciones entre ellas.
* **Nivel físico** (interno): Capa de datos, visibilidad de almacenamiento. Es el nivel de codificación donde se maneja el HW de almacenamiento de datos. Administra la ubicación física.
* 

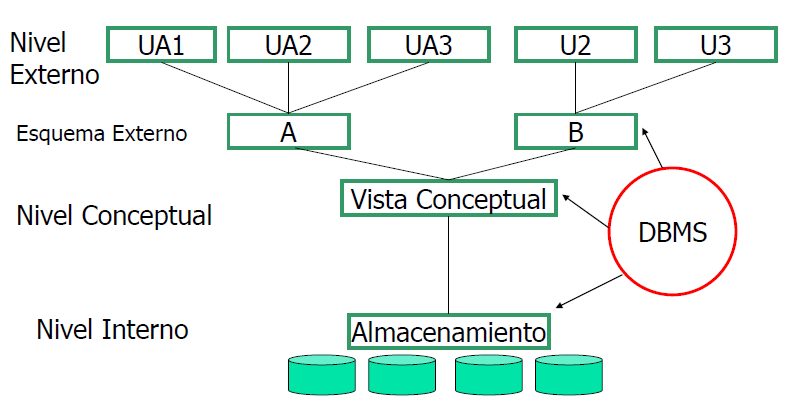
**Arquitectura ANSI-4-SPARC**: es un estándar de diseño abstracto para sistemas de gestión bases de datos que consta de 4 capas. La **ventaja** de utilizar capas es que se puede ocultar al usuario la complejidad del sistema y permite proteger al hardware de las incursiones o errores de los programadores mal o bien intencionados. Los niveles que posee el ANSI-4-SPAC son:

* Nivel Físico: donde se maneja el hardware de almacenamiento de los datos.
* Nivel Lógico: es el que le da visibilidad general a la base de datos.
* Nivel Externo: comprende las aplicaciones y formularios que le permiten al usuario acceder y usar los datos de la base de datos.
* Nivel Conceptual: el objetivo es ofrecer los esquemas de los datos y las vistas con los datos que deben acceder desde cada aplicación que opera

**Relación con el paradigma de programación en tres capas**: la construcción de la capa de datos abarca el nivel de almacenamiento y el nivel conceptual. Se debe mantener una capa de datos compartida e integrada para toda la organización. La capa de negocios debe programarse sobre el nivel de esquema (vistas y esquemas que el DBA pone a su disposición). La capa de programación de formularios y vistas de usuario se desarrollarán generalmente en el mismo lenguaje de programación o en formularios de Internet.

**Relación del Sistema de Administración de Base de Datos con cada capa**:

* Capa de datos: la responsabilidad de esta capa debe ser exclusiva de un DBMS centralizado para toda la organización.
* Capa de Negocios: el DBMS traduce las operaciones equivalentes.



**Independencia de los datos**: es el principal objetivo de un Sistema de Administración de Base de Datos. Consiste en mantener la inmunidad de las aplicaciones ante los cambios en la estrategia de acceso o en la estructura de almacenamiento de los datos.

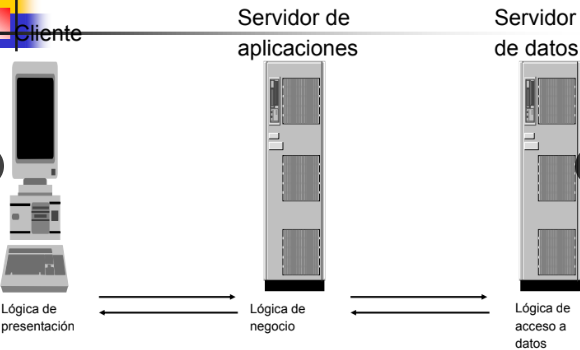
* **Independencia Física de los datos**: capacidad de modificar la definición física y/o método de almacenamiento, sin afectar el esquema/ representación lógica, las aplicaciones de usuario ni vistas.
* **Independencia Lógica de datos:** capacidad de modificar el esquema sin afectar las aplicaciones de usuario ni vistas/ programas de aplicación.

**Tipos de Modelos de datos**:

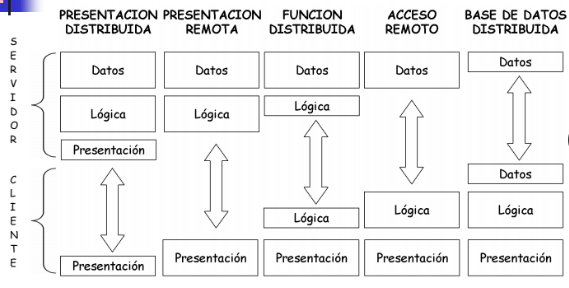
* Para diseñar y planificar BD:
  + Modelo Entidad-Relación
  + Modelo Orientado a Objetos (UML)
* Para implementar DBMS:
  + Modelo Relacional (Object-relational)
  + Modelo Orientado a Objetos.

**Tipos de Aplicaciones**

* Cliente ligero- Servidor Pesado
* Cliente pesado- Servidor ligero.
* Aplicaciones distribuidas en 3 niveles:



**Tipos de Distribución de lógica:**

  
**Responsabilidades del Departamento de Sistemas**:

* **Administración de Datos (DA):** está orientado a los usuarios finales. Hay modelos conceptuales, redundancia y estándares, diccionario de datos, seguridad de datos, educación de usuario y herramientas de usuario.
* **Administración de Base de Datos (DBA):** está orientado a los especialistas. Hay conocimientos del DBMS como el diseño, la parte física, control de rendimiento, recuperación y recomendaciones.
  + **Responsabilidades:**
    - Decidir el contenido de la BD: aprueba la incorporación de nuevos datos persistentes y adapta la estructura para alojarlos.
    - Decidir estructura de almacenamiento y estrategia de acceso: responsable de optimizar las estructuras de datos y estrategias de acceso para mantener la eficacia y performance de la BD.
    - Definir autorizaciones y validaciones: tiene la responsabilidad de dar las autorizaciones pertinentes a cada usuario. Debe definir los procesos de validación generales de cada dato.
    - Definir estrategias de respaldo y recuperación: es responsable de mantener actualizados los planes de back up y de recovery de las BD. Este conjunto de planes se denominan planes de contingencia y deben realizar simulaciones periódicas para su comprobación.
    - Controlar desarrollo, prueba y puesta en operación de aplicaciones: participa en todo el ciclo de desarrollo dando paquetes de datos necesarios para los desarrolladores, para los encargados de las pruebas y finalmente probando su puesta en funciones sobre la BD en producción. Debe coordinar con los jefes de proyectos.
    - Controlar performance y nuevos requerimientos: es necesario un constante monitoreo, vigilando que las estructuras de datos sean eficientes y que el HW sea suficiente.
    - Vincularse con los usuarios: es capaz de interactuar con los usuarios para mantener una adecuada retroalimentación de las necesidades de los sistemas.
    - Rutinas de mantenimiento: debe contar con herramientas de monitoreo y de mantenimiento para mantener la BD ordenada y estable.

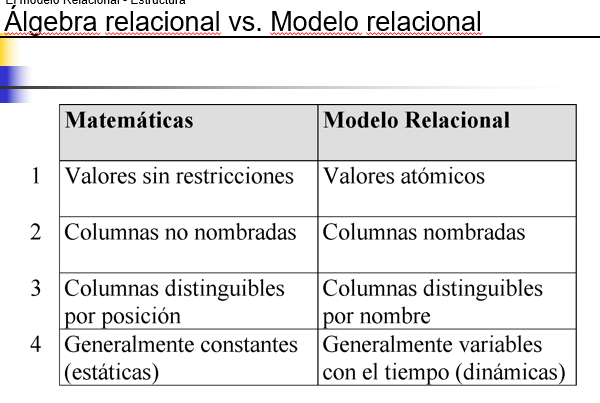
**Vistas:** una vista es una tabla virtual que en realidad no existe como tal. Es el resultado de una o varias operaciones relacionales realizadas sobre las tablas base. Se pueden manipular como si se tratara de una relación base, pero no está almacenada físicamente. Su contenido está definido como una consulta sobre una o varias tablas base. Cualquier operación que se realice sobre la vista se traduce a operaciones sobre las relaciones de las que se deriva. Son dinámicas. Ventajas:

* Proporcionan un poderoso mecanismo de seguridad, ocultando partes de la BD a ciertos usuarios.
* Permiten que los usuarios accedan a los datos en el formato que ellos necesitan, es decir, con distintos formatos.
* Se pueden simplificar operaciones sobre las relaciones base que son complejas.
* Ayudan a proporcionar independencia de datos a nivel lógico.
* Cuando se actualiza una tabla base, el cambio se refleja automáticamente en todas las vistas que la referencias y viceversa.

**Abstracción**: se extraen del mundo real aquellos elementos y objetos con las propiedades que se consideran pertinentes para el análisis que se efectúa.

**Conceptos Adicionales**:

* **Especialización:** una entidad puede tener atributos compartidos y atributos propios. Permite manejar el concepto de herencia ya que determinados atributos (los que pertenecen a una superclase), se definen una única vez y en un único lugar y se pueden reutilizar en las clases inferiores, mientras que los propios son definidos en el nivel inferior.
* **Generalización:** busca imponer ciertas restricciones al modelo, como la pertenencia a entidades de niveles inferiores. Las subentidades heredan atributos, interrelaciones y generalizaciones de la superentidad.
* **Agregación**: es una abstracción a través de la cual las relaciones se tratan como entidades de nivel superior.

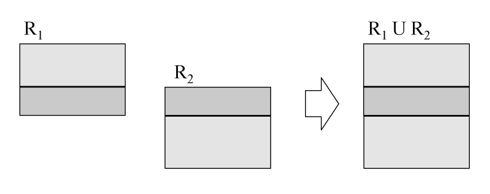


**Modelo Relacional**: se basa en el concepto matemático de relación representado mediante una tabla. Una tabla relacional está compuesta por columnas y filas. Un atributo es el nombre de una columna de una relación. Una tupla es una fila. Es un modelo de datos, para modelar problemas reales y administrar datos dinámicamente. Su idea principal es el uso de relaciones las cuales se representan como “tuplas”.

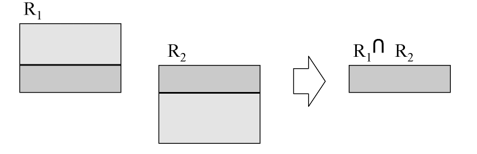
**Dominio**: conjunto de valores legales de uno o varios atributos.

**Operaciones del Algebra Relacional**:

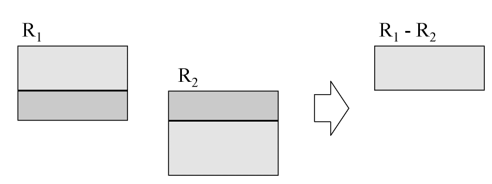
* **Tradicionales:**
  + **Unión**: operación binaria que produce una tabla relacional que contiene todas las tuplas de la primera y de la segunda tabla, incluyendo sólo una vez aquellas que tienen en común. Las tablas deben ser del mismo grado y los dominios de las columnas deben ser iguales. Se usa para cuando hay tablas muy grandes que tuve que partir y las uno en memoria para poder operarlas.



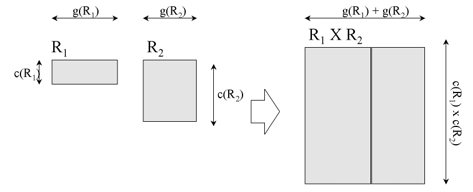
* + **Intersección**: operación binaria que produce una tabla que contiene las tuplas de la primera y segunda tabla, pero son aquellas que se encuentran repetidas en ambas. Hay tuplas/filas que están en una tabla y también se encuentran en la segunda tabla.



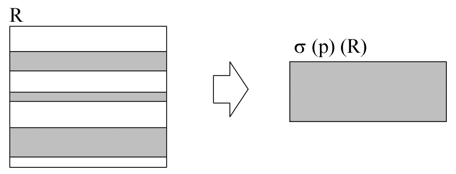
* + **Diferencia**: operación binaria que produce una tabla que contiene todas las tuplas de la primera tabla y aquellas que no se encuentren en la segunda. Deben ser compatibles, del mismo grado y dominio. Todas las tuplas que no se encuentran en la segunda tabla. Le saco los campos que tienen en común



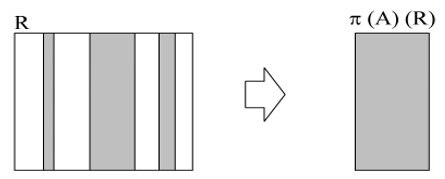
* + **Producto Cartesiano**: operación binaria que produce una tabla relacional que contiene como tuplas a todas las combinaciones de la primera tabla con las de la segunda, teniendo como columnas la concatenación de las columnas de ambas. Cada tupla de una tabla se combina con todas las tuplas de la tabla siguiente



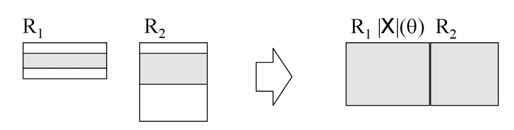
* **Especiales:**
  + **Selección:** operación unitaria que produce una tabla relacional con tuplas seleccionadas que cumplan una condición. Se agarran los valores que cumplan una condición. Tengo proyección, reunión y selección



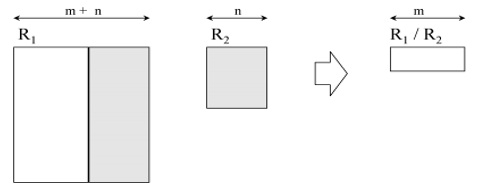
* + **Proyección:** operación unitaria que produce una tabla relacional con las columnas de la tabla relacional operando que se den como argumento A y eliminando de este resultado las tuplas repetidas. Selecciono por columnas.



* + **Reunión**: operación binaria que produce una tabla relacional que contiene como tuplas aquellas combinaciones de las tuplas de la primera tabla relacional operando con las de la segunda que satisfagan un predicado. El Join Natural es un caso particular donde el predicado es una comparación por igualdad de las columnas “homónimas” suprimiendo las columnas redundantes del resultado. Me quedo con las tablas que cumplen con una condición lógica. Mezcla de todas las filas, pero solamente donde coinciden atributos entre tablas.



* + **División**: operación binaria que produce una tabla relacional que si la multiplicamos cartesianamente con la segunda, produce tuplas contenidas en la primera tabla relacional. Tomo los valores repetidos que hay en una parte de la tabla uno y en la tabla dos. Y me quedan el resto de los datos donde se compartan los datos. Ejemplo: si en R1 tengo películas y actores, y en R2 hay actores. Me quedan las películas de esos actores.



**Campos:**

* **Normales:** se define por defecto, no participa de ninguna clave.
* **Superclaves**: es un atributo o conjunto de atributos que identifican de modo único las tuplas de la relación.
* **Claves candidatas**: superclave en la que ninguno de sus subconjuntos es una superclave de la relación. Debe satisfacer estos requisitos si o si:
  + **Minimalidad**: tiene la mínima cantidad de atributos necesarios para ser clave.
  + **Unicidad**: si pierde uno de sus atributos, deja de ser clave.
* **Clave Primaria**: clave candidata que se escoge para identificar sus tuplas de modo único. Puede estar formada por todos los atributos de la relación, pero normalmente habrá un pequeño subconjunto de los atributos que haga esta función
* **Clave Foránea:** atributo o conjunto de atributos de una tabla relacional cuyos valores coinciden con los valores de la clave primaria de otra tabla, representan relaciones entre datos.
* **Clave compuesta:** formada por más de un atributo.
* **Clave alternativa:** claves candidatas no elegidas como clave primaria.
* SQL tiene tabla de dispersión y tabla de… Se agranda cuando se está trabajando y luego achica su tamaño.
* SIRVE QUE ESTA BASADO EN LA MATEMATICA PARA TENER OPERACIONES REVERSIBLES Y OPTIMIZA LAS TRANSACCIONES🡪 descompone en operaciones algebraicas. Le ofrece al usuario una interfaz más amigable y el usuario a través del motor puede operar descomponer y proponer vías de solución.

RESTRICCIONES: nos referimos a que defino la tabla en el modelo relacional, aplico las restricciones básicas, pero después agrego restricciones más.

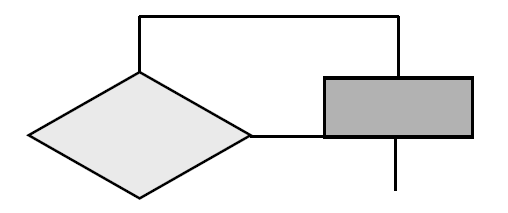
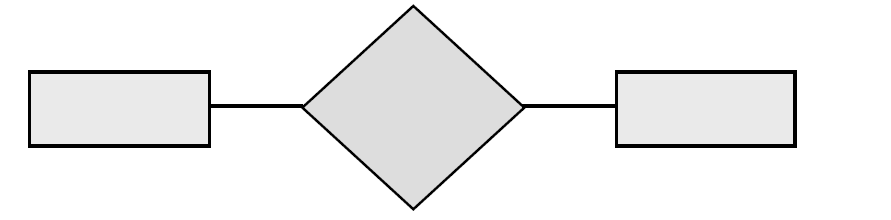
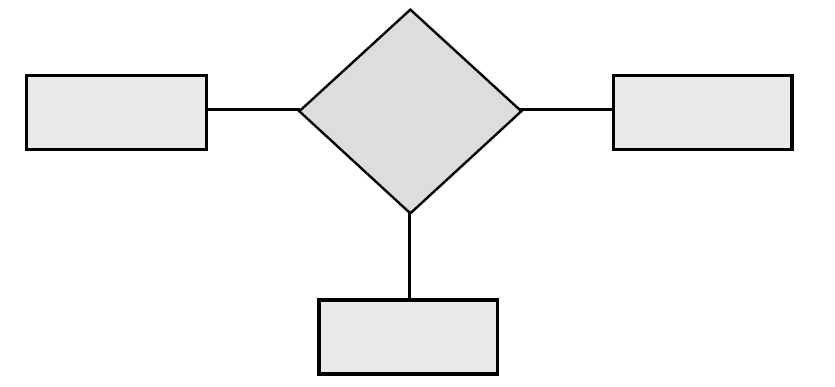
* + - * **De Columna**: tipo de dato. Se permite a un dato estar marcado para no contener valor alguno. Un dato puede estar marcado como nulo si: Existe la posibilidad de desconocer la información o No tiene sentido asignar un valor del dominio.
      * **De Dominio:** tipo de dato. Se implementa a través de tipos de datos definidos por el usuario en base a los tipos provistos por el RDBMS y a través de restricciones sobre los dominios base.
      * **Integridad de entidad**: restricción de la clave primaria. Ningún valor de la clave primara puede ser nulo🡪 valor que no existe, falta de valor. Ningún componente de una llave primaria puede contener valores NULOS. Ningún componente de una llave foránea debe permitir un valor NULO por inaplicable, aunque si puede permitir valor NULO por desconocimiento de información.
      * **Integridad referencial**: restricción de la clave foránea. Ningún valor de la clave foránea puede ser nulo 🡪 **todo valor de la clave foránea debe tener un valor de clave primaria.** Para cada llave foránea debe existir un valor equivalente de una llave primaria y en el mismo dominio.
      * **Integridad definida**: NO ES OBLIGATORIA. El usuario agrega las restricciones que quiere. Puede tener diversos aspectos, principalmente para consistencia de la información.

**Diseño/Modelo Conceptual:** se realiza a un nivel de abstracción más cercano al problema real. El objetivo es obtener un diseño apegado al problema real del sistema antes de implicarse en los problemas de su implementación dentro de una base de datos en particular. Sirve de documentación del modelo y es independiente de su implementación en una determinada base de datos.

**Modelo Lógico:** es dependiente de su implementación, pero respeta reglas universales que facilitan su interpretación.

**Modelo Físico:** nace de la compilación del modelo lógico, es muy dependiente de su implementación y de la plataforma que lo soporta, depende del motor de base de datos y su tecnología no es transparente al usuario.

**Modelo Entidad-Relación:** es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Es una herramienta de diseño y de documentación de las bases de datos. Permite representar la realidad sobre la cual versa la base de datos a partir del lenguaje natural.

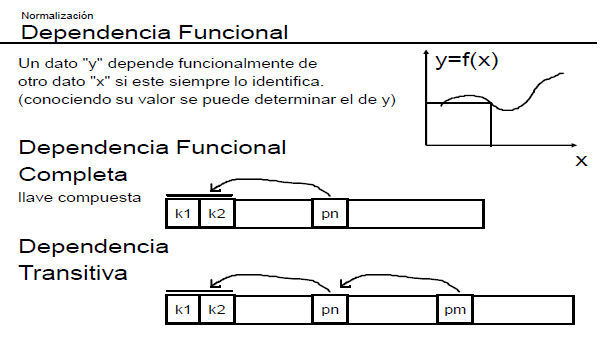
* **Entidades**: son elementos que existen en el sistema objeto de análisis y como tales se necesitan identificar dentro de mi modelo. Es una cosa u objeto en el mundo real que es distinguible de todos los demás objetos.
  + **Entidad concreta**: objeto con existencia física
  + **Entidad abstracta**: objeto con existencia conceptual
  + **Entidad regular**: entidad con existencia propia, es decir, no debe su existencia a otra entidad
  + **Entidad débil:** entidad que no tiene existencia propia, es decir, debe su existencia a otra entidad. No posee atributos para formar su clave primaria, por lo que debe complementarla heredando la clave primaria de la entidad que depende (entidad fuerte).  RECTANGULO CON DOBLE LINEA
    - **Relación débil**: une una entidad fuerte con una débil.  ROMBO CON DOBLE LINEA
* **Atributos:** son características de las entidades que considero necesario individualizar para mi modelo, es decir, definen o identifican una entidad.
  + Atributo multivaluado: atributos que pueden tomar más de un valor para cada ocurrencia de la entidad. 🡪 OVALO CON DOBLE LINEA.
  + Atributo Monovaluado: puede adoptar un solo valor.
  + Atributo Derivable: puede ser obtenido a través de un cálculo realizado a partir de otro atributo que conocemos. 🡪 OVALO CON LINEA PUNTEADA.
  + Atributo Identificador: obligatorio y monovaluado
    - Simple interno: formado por un solo campo propio de la entidad
    - Compuesto interno: formado por dos o más campos propios de la entidad
    - Externo, compuesto y mixto: el identificador de una entidad débil requiere el identificador de la entidad de la que depende y sus campos para componer su identificador.
  + Atributo Compuesto: se puede descomponer en partes
  + Atributo optativo: no es obligatorio, puede no contener un valor
  + Atributo obligatorio
* **Relaciones**: objetos que vinculan dos o más entidades con un caso especial en que una entidad se vincula consigo misma.
* Grado de la relación: cantidad de entidades que relaciona.
  + Interrelación unitaria: 
  + Interrelación binaria o de grado dos: 
  + Interrelación ternaria o de grado tres: 
* Cardinalidad: cantidad de elementos de una entidad que pueden vincularse con los elementos de la otra.
  + Muchos a muchos (n:m)
  + Uno a muchos (1:n)
  + Muchos a uno (n:1)
  + Uno a uno (1:1) 🡪 lo poseen las relaciones unitarias.

**Cobertura:** grado en que las ocurrencias de las subentidades coinciden con las ocurrencias de la superentidad. Cuantos elementos de la superentidad se encuentra en la subentidad

* **Cobertura Total**: cuando todos los elementos de la superentidad aparecen en las subentidades
* **Cobertura Parcial**: cuando no todos los elementos de la superentidad aparecen en las subentidades.
* **Cobertura Exclusiva**: cuando todos los elementos de la superentidad aparecen en las subentidades y no hay elementos que pertenezcan a dos entidades simultáneamente.
* **Cobertura Superpuesta**: cuando los elementos de la superentidad pertenecen simultáneamente a dos o más subentidades.

**Notación alternativa🡪 Pata de gallo** (para representar la razón de cardinalidad.)

**Dependencia Funcional**: noción semántica. Se define por los modelos mentales del usuario y las reglas del negocio. Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos. Existe dependencia funcional cuando entre dos atributos, los valores de uno de ellos determinan en forma unívoca los valores de otro. Primero se deben distinguir: Atributos primos: forman parte de la clave. Atributos NO primos: no forman parte de ninguna clave. Tipos de dependencias:

* **Completa:** un campo no primo debe depender de toda la clave y no de un campo de una clave compuesta
* **Transitiva**: un campo no depende directamente de la clave, sino que depende de otro atributo no primo.
* 

**Normalización:** serie de pasos que corresponde a una forma normal con propiedades.

* **Primera Forma Normal**: todos los dominios de la tabla contienen valores atómicos, es decir, no hay grupos repetitivos. Los grupos repetitivos es el atributo o grupo de atributos que tienen múltiples valores por cada tupla de la tabla relacional. Para eliminarlos podemos, asignar un solo valor para cada valor de los grupos repetitivos o realizar una tabla relacional aparte con los valores necesarias, haciendo una clave foránea a donde estaba. No hay atributos multivaluados, no compuestos, solo monovaluados.
* **Segunda Forma Normal**: la tabla relacional se encuentra en segunda forma normal si se encuentra en 1FN y cuando todos los atributos que no se encuentran en la clave primaria son dependientes de esta. Se debe eliminar las dependencias parciales de la clave primaria por lo que se eliminan los atributos funcionalmente dependientes y se ponen en una tabla relacional con una copia determinante.
* **Tercera Forma Normal**: la tabla relacional se encuentra en tercera forma normal si se encuentra en 2FN y cuando cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. Para eliminar las dependencias transitivas se eliminan los atributos que dependen transitivamente y se ponen en una tabla relacional con una copia en su determinante.
* **Forma Normal Boyce-Cood (FNBC):** la tabla relacional tiene que estar en 3FN y los únicos determinantes son claves candidatas.
* **Cuarta Forma Normal:** la tabla debe estar en 3FN o FNBC y se asegura que las dependencias multivaluadas independientes estén correctas y físicamente representadas en un diseño de base de datos.
* **Quinta Forma Normal:** la tabla debe estar en 4FN y cada dependencia de unión en ella es implicada por claves candidatas.

Desnormalización: proceso de procurar optimizar el desempeño de una base de datos por medio de agregar datos redundantes.

**Mecanismos de abstracción:**

* **Abstracción de Clasificación**: construcción de una clase a partir de un conjunto de objetos tomando sus propiedades en común (MIEMBRO DE...)
* **Abstracción de Agregación**: construcción de una clase a partir de un conjunto de clases que son componentes (ES PARTE DE…). Relación varios a varios que tiene características de otras entidades.
* **Abstracción de Generalización**: Relación de subconjunto de una clase respecto a otra clase raíz (es subconjunto de… HERENCIA) se utiliza el concepto de herencia.

Trigger: se ejecuta lo cundo pasa algo en la base de datos

Procedimiento almacenado: hay que invocarlo para que se pueda ejecutar

Campos calculados: no se almacenan porque hay una redundancia encubierta, puedo calcularlo si tengo los datos del cálculo almacenado.

Dominio: anterior al tipo de dato, más restrictivo que el tipo de dato. Conjunto posible de datos.

Tipo de dato: acota al dominio. Menos restrictivo.

Hay que asegurar que no haya redundancia, mantener la integridad.

Algebra relacional: Motor de optimización, crear un plan de ejecución y permite que las operaciones sean reversibles. Hace más fácil imponer un estándar.

Niveles de aislamiento: el archivo no tiene aislamiento, la base de datos sí. Aislamiento de capas.

Unión es diferente a reunión.

