¿QUÉ ES UN DATO?

Hechos conocidos que pueden **registrarse** y tienen un **valor implícito**.

¿PARA QUÉ SIRVE?

Evita redundancia e inconsistencia de datos.

Permite acceso a los datos en todo momento. Ejemplo: hace décadas era normal que sistemas funcionaran de 8hs a 20hs (horario laboral).

Evita anomalías en acceso concurrente. Ejemplo: Múltiples alumnos se anotan a las materias y el SIU colapsa.

Restringir accesos no autorizados.

Suministro de almacenamiento persistente de datos, incluso ante fallos.

Integridad de datos (Integridad Referencial).

Backups. Acceso posible incluso durante backups. backups incrementales en vez de backups completos cada vez que se realizan.

Base de Datos

DBMS(Data Base Managament System)

Programa que facilita los procesos de definición, manipulación, construcción y consulta de la BD.

DEFINICIONES Y CARACTERÍSTICAS

- *Coleccion de **datos relacionados** que sirven un propósito.
- *Representa aspectos del mundo real.
- *Sustentada **fisicamente** en archivos en dispositivos de almacenamiento.
- *Es una colección **coherente** de datos. Una BD no puede ser un conjunto **aleatorio** de datos.

COMPONENTES

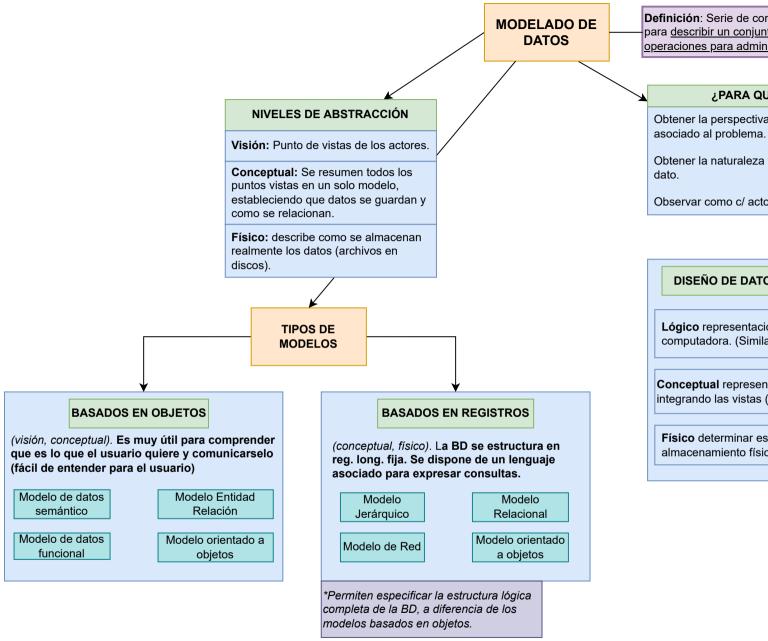
Data Definition Language:

Para especificar el esquema de la DB.

Data Manipulation Language: Para Consulta y ABM de la DB.

Procedimentales (SQL): Requiere que el usuario especifique que datos se muestran y como obtenerlos.

No precedimentales (QBE): Solo requiere esicificar los datos que se quieren



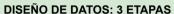
Definición: Serie de conceptos que sirven para describir un conjunto de datos y las operaciones para administrarlos.

¿PARA QUÉ SIRVE?

Obtener la perspectiva de cada actor

Obtener la naturaleza y necesidad de cada

Observar como c/ actor utiliza cada dato.



Lógico representación en una computadora. (Similar a un plano).

Conceptual representación abstracta integrando las vistas (como maqueta).

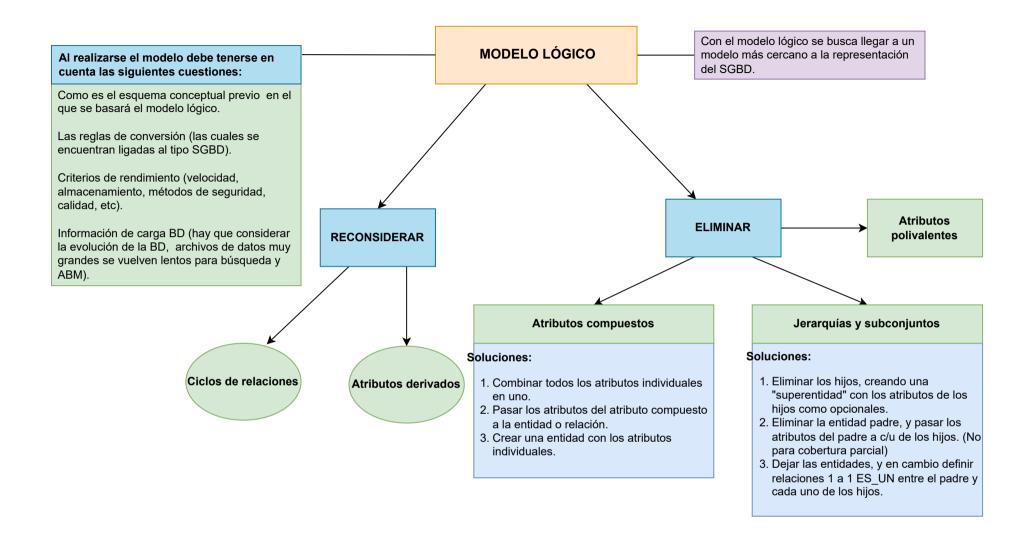
Físico determinar estructuras de almacenamiento físico.

Genérico Alejado del tipo de DBMS Aleiado del producto particular

Específico Orientado a un producto

Más específico Orientado a un tipo de DBMS Alejado del producto particular

OBJETIVOS MODELO CONCEPTUAL ER • Abstraer la información de un problema. Captar las necesidad del cliente respecto al problema. Meiora la interacción cliente-desarrollador disminuvendo la **CARACTERÍSTICAS** El modelo ER no llega a tener brecha entre la realidad del problema v el sistema a Que se buscan tener en un buen modelo implementacion fisica. desarrollar Expresividad: disponer de todos los medios necesarios para describir un problema. Formalidad: cada elemento representado sea preciso y **MEJORAR MODELO CONCEPTUAL** bien definido, con una sola interpretación posible. Autoexplicativo: el modelo se expresa a sí mismo si puede presentarse con los elementos definidos, sin necesidad de Minimalidad: cada elemento tiene una única utilizar aclaraciones. representación posible. Completitud: Un modelo está completo cuando estas todas las **RELACIONES** (Agregación) Simplicidad: el modelo debe ser fácil de entender por el caracteristicas del dominio de aplicación(requerimientos) cliente v por el desarrollador. contempladas en dicho modelo. COMPONENTES Representan agregaciones entre 2 o más entidades. Correción: Un modelo es correcto si cada elemento en su construcción fue utilizado con propiedad (todas las cardinalidades, coberturas, identificadores deben estar expresados). **ATRIBUTO ENTIDAD Expresividad:** Un modelo es expresivo si a partir de su Representa una propiedad básica obversación puede notarse todos los detalles del modelo (Por de una entidad o relación. ej: usando jerarquías que sean coherentes). Representa un elemento u Equivale a un campo de un obieto del mundo real con **ATRIBUTO** Extensible: es extensible si es fácilmente modificable para registro. identidad. COMPUESTO incoporar nuevos conceptos en él. Representa un atributo Legibilidad: Es legible si la representación es adecuada. generado a partir de una **IDENTIFICADOR** combinación de atributos Mínimo: un esquema es mínimo cuando cada concepto se simples. representa una sola vez en el modelo (Atributos derivados y Es un atributo o conjunto de ciclos de relaciones). atributos que permite reconocer una entidad dentro del conjunto de ATRIBUTOS DERIVADOS **CONJUNTO DE** entidades. **ENTIDADES** Existen de los siguientes tipos: Representan información que se puede obtener por otros medios. Simples o Internos o Externos Compuestos **JERARQUÍAS SUBCONJUNTO** (Generalización) Ventaja: más rápido tenerlo Desventaja: información en una variable y consultarlo repetida que debo actualizar Permite extraer propiedades Una jerarquía que presenta un solo que calcularlo cada vez cuando se modifica el valor comúnes de varias entidades (o conjunto de entidades como hijo. relaciones) y generar una Tiene obligatoriamente Cobertura superentidad que las contenga. parcial exclusiva. Cobertura total/parcial y superpuesta/exclusiva.



TABLAS/RELACIONES

El modelo relacional representa la BD como una colección de archivos denominados tablas, las cuales se conforman por registros.

Las filas o tuplas representan un registro o elemento del archivo. mientras que las columnas representan un atributo.

INTEGRIDAD REFERENCIAL

Propiedad deseable en BD relacionales. Asegura que un valor que aparece para un atributo en una tabla, aparezca en otra para el mismo atributo.

Plantea restricciones entre tablas y sirve para mantener la **consistencia** entre las tuplas de dichas tablas.

Clave Foránea: atributo de una tabla que es CP en otra tabla, sirviendo así como enlace entre ambas.

MODELO FÍSICO RELACIONAL

MODELO FÍSICO

PROCESO DE

CONVERSIÓN A

ELIMINACIÓN IDENTIFICADORES EXTERNOS.

Solución: se incorporan dentro de la identidad aquellos atributos que sean necesarios para la definición del identificador de forma interna.

SELECCIÓN DE CLAVES PRIMARIAS Y CANDIDATAS

Un **identificador** sirve para distinguir una entidad del conjunto de entidades.

Se elige como CP aquel identificador que sea en lo posible simple y ocupe menos espacio, ya que es más eficiente para el acceso físico(hashing).

El resto de los identificadores serán CC, que se utilizarán para generar índices secundarios (más rápido para acceso secuencial)

El autoincremental es una opción para elegir una CP, se agrega a lo último ,no se puede modificar modificar v lo administra el DBMS. evitando el error humano del usuario.

CONVERSIÓN DE ENTIDADES!!!!

1 a 1: Se genera una única tabla que contenga a ambas entidades.

N a N: Se genera una tabla que relaciona a las 2 entidades.

1 a N - Participación total de N: Se pone la CP de la entidad de muchos como CF en la entidad de

1 a N - Participación parcial de N: Se resuelve iqual que el caso anterior.

1 a N - Participación parcial de 1: Se genera una tabla que relaciona a las 2 entidades. Si se resolviera como la opción anterior, hay riesgo de atributos nulos.

1 a N - Participación parcial de ambos: Resuelve igual que caso anterior.

Jerarquía: Las entidades hijas tienen como CP la CP de la entidad padre, pero solo será un autoincremental en la tabla del padre.

Recursivas: Se tiene la CP de la entidad 2 veces como atributos distintos (Ej: correlativas).

ESCENARIOS I.R.

Restringir operación: si se intenta borrar/modificar una tupla que tiene IR con otra, no se permite llevar a cabo. Ej: Teniendo las tablas CLIENTES y FACTURAS, para borrar un cliente es necesario que no tenga facturas, sino no se permite borrar al cliente.

Realizar operación en Cascada: Si se intenta borrar/modificar una tupla donde esta definida la CP de la IR, la operación se realiza en cadena sobre todas las tuplas de la tabla que tiene definida la CF.

Establecer CF en nulo: si se borra/modifica el valor del atributo que es CP. sobra la CF se establece en nulo.

No hacer nada: No se controla la IR. Equivale a no tener restricciones de IR.

DEPENDENCIA FUNCIONAL DEPENDENCIAS MULTIVALUADAS Un atributo Y depende funcionalmente de un atributo X (X=>Y), cuando para un valor de X Una Dependencia Multivaluada, denotada siempre se encuentra el mismo valor para Y. como X->-> Y. siendo X e Y coniuntos de atributos en una tabla, indica que para un EL ATRIBUTO X DETERMINA EL ATRIBUTO Y. valor determinado de X es posible determinar múltiples valores para el atributo Y. Si se define DF X=> Y. siendo que no es En un DF X=>Y. al atribuvo X Una DM no es necesariamente una posible encontrar otra dependencia Z=> Y se lo denomina determinante situación anómala para una BD. donde Z sea un subconjunto de X, se tiene v al atributo Y. consecuente. una **DF completa**. El problema de la multideterminación de datos comienza cuando en una tabla cualquier se tiene 3 atributos, X, Y, Z y se analizan las siguientes DM: **TIPOS DE DF** (X,Y)->> Z $(X,Z) \longrightarrow Y$ Pero en realidad, tanto Y como Z **DF PARCIALES** DF de BOYCE-CODD solamente dependen de X en cada una de **DF TRANSITIVAS** las DF. Una DF X-> Y se denomina parcial Una DF X-> Y es DFBC cuando cuando existe otra dependencia Z->Y, Una DF->Y es transitiva X no es una CP o CC, e Y es siendo Z subconjunto de X. cuando existe un atributo una CP o CC, o parte de ella. Z, tal que X-> Y y Y-> Z. DICTA= (nombre alumno, nombre materia, nombre docente) PEDIDOS=(idpedido,idproducto, ALUMNO =(idalumno, nombre, direccion, descripcionproducto, fechapedido, cantidad) idcarrera, nombre carrera) Nombre alumno, nombre materia -> nombre docente Nombre alumno, nombre docente -> nombre materia **Dependencias Funcionales:** idalumno->nombre idpedido,idproducto-> descripcionproducto idalumno->direccion Nombre docente-> nombre materia idpedido,idproducto-> fechapedido idalumno->idcarrera idpedido,idproducto-> cantidad idcarrera->nombre carrera idalumno->nombre carrera idproducto->descripcionproducto idpedido-> fechapedido

SOLUCIÓN:
ALUMNO=(idalumno, nombre, direccion, idcarrera)

CARRERA=(idcarrera,nombre carrera)

SOLUCIÓN:

DETALLESPEDIDOS = (<u>idpedido,idproducto</u>,cantidad)
PEDIDOS = (<u>idpedido</u>,fechapedido)
PRODUCTOS=(<u>idproducto</u>,descripciónproducto)

