

Conceptos

DATO

- Del Latín: Datum “Lo que se da”
- Es la representación simbólica de un atributo o variable cuantitativa o cualitativa
- Simbólica: Números, letras, figuras, dibujo, etc
- Cuantitativa: Nos referimos a cantidad de cuanto es o cuántos son. 5 Metros, 3 Kilos, 12 años
- Cualitativa: nos referimos a una cualidad. Redondo, Verde, Dulce
- Los datos relacionados entre sí nos brindan información o un significado

BASE DE DATOS

- Sistemas estructurado de forma lógica para la administración de datos
- Colección sistemática de datos con diferentes modos de organización y con un propósito específico
- Conjunto de información almacenada y consultada sistemáticamente
 - **Conjunto de información:** Colección de datos enlazados puede ser poca o mucha (e.g. recetas, notas, recibos, fotos, etc.)
 - **Almacenada:** Se guarda, se persiste la información
 - **Consultada:** Constantemente se realiza búsquedas y se recopila información
 - **Sistemáticamente:** La información administra bajo procesos definidos

Ejemplo

- Me dedico a armar computadoras para una empresa y constantemente voy a comprar partes.
- **Conjunto de información:** Recibos
- **Almacenada:** Recibos en una cajas de partes de computadoras, mochila, bolsillos, etc.
- **Consultada:** Recopilar todos los recibos, búsqueda de recibos, cajas, mochilas, bolsillos
- **Sistemáticamente:** Folder para guardar los recibos y guardamos con un orden e.g. fechas la consulta nos dará resultados más rápidos o más ordenados

DBMS

Sistema Manejador de Base de Datos (Software)

Permite la creación, gestión y administración de base de datos

Su propósito general es el de manejar de manera clara, sencilla y ordenada los datos de una base de datos

Base de Datos no es lo mismo Sistema manejador de Base de Datos

- Base de Datos se le llama al archivo como al manejador
- “¿Qué base de datos utilizas? Oracle”
- “Dame la base de datos de los clientes” hace referencia al archivo

Problemas que nos solucionan:

- **Abstracción de los datos.** Los DBMS ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos **ocupa uno o cientos de archivos**, este hecho se hace **transparente al usuario**. Así, se definen varios niveles de abstracción.
- **Independencia.** Consiste en la capacidad de **modificar el esquema** (físico o lógico) de una base de datos **sin tener que realizar cambios en las aplicaciones** que se sirven de ella.
- **Consistencia.** *LOS DATOS REPETIDOS SE ACTUALICEN DE FORMA SIMULTANEA*
En aquellos casos en los que **no se ha logrado eliminar la redundancia**, será necesario vigilar que aquellos datos que aparecen repetidos se actualicen de forma coherente, es decir, que todos **los datos repetidos se actualicen de forma simultánea**.
- **Seguridad.** Los datos almacenados en una base de datos pueden llegar a tener un gran valor. Los DBMS deben **garantizar que estos datos se encuentren seguros** frente a usuarios malintencionados, que intenten leer datos privilegiados; frente a ataques que deseen manipular o destruir los datos; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los DBMS

disponen de un complejo **sistema de permisos** a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

- **Integridad**. Adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de **proteger los datos ante fallos de hardware**, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper los datos almacenados. Los DBMS proveen mecanismos para garantizar la recuperación de la base de datos hasta un estado consistente conocido en forma automática.
- **Respaldo**. Los DBMS deben proporcionar una forma eficiente de realizar **copias de respaldo de los datos** almacenados y de restaurar a partir de estas copias los datos que se hayan podido perder.
- **Control de la concurrencia**. En la mayoría de entornos lo más habitual es que sean muchas las personas que acceden a una base de datos, y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un DBMS debe controlar este acceso concurrente a la DB, que podría derivar en inconsistencias.
- **Manejo de Transacciones**. Una Transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que el estado, luego de una ejecución, en la que se produce una falla, es el mismo que se obtendría si el programa no se hubiera ejecutado. Los DBMS proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.
- **Tiempo de respuesta**. Lógicamente, es deseable minimizar el tiempo que el DBMS tarda en darnos la información solicitada y en almacenar los cambios realizados.

MODELO

Es una representación de la realidad que conserva solo los detalles relevantes

MODELO DE BASE DE DATOS

Conjunto de datos utilizados para organizar los datos de interés y describir su estructura en forma comprensible para un sistema informático

Los Modelos de datos más sobresalientes y usados son:

- Modelo Relacional
- Modelo Jerárquico
- Modelo de Red
- Modelo Orientado a Objetos

MODELO RELACIONAL

- Ordena los datos en Tablas cada una compuesta de filas y columnas
- Cada Columna enumera un atributo
- Dominio es el conjunto de atributos
- Clave primaria, foranea
- Cada fila es una tupla (Una instancia específica)
- También representa el tipo de relaciones
- Trabaja principalmente con el lenguaje SQL

MODELO JERARQUICO

- Organiza los datos en una estructura de árbol
- Cada elemento tiene un único elemento raíz
- Utilizado en los 60 70's por IBM
- Poco utilizado en la actualidad por alguna ineficiencias
- Actualmente utilizado por la notación XML
- Tiene una notación muy clara pero inflexible
- Los registros con hijos se llaman records y los que no tienen se llaman hojas y son los que suelen contener los documentos

MODELO DE RED

- Se basa en el Modelo Jerárquico
- Permite relaciones mucho a muchos
- Basado en teorías matemáticas de conjuntos
- Se construye con conjuntos de registros relacionados
- Cada conjunto consiste en:
 - Un registro propietario o principal
 - Uno o más registros miembro o registros secundarios

MODELO ORIENTADO A OBJETOS

- Suelen utilizarse en plataformas JAVA y .NET
- Destaca el poco uso de memoria

- Acostumbra a trabajar con el lenguaje OQL (Object Query Language)
- En su mayor parte no recurren a la normalización
- Define una base de datos como una colección de objetos

MODELO ORIENTADO A DOCUMENTOS

- Documentos semi-estructurados homogéneos e identificados por un valor único (clave)

DISEÑO DE BASE DE DATOS

Es el proceso por el que se determina la organización de una Base de Datos, incluidas su estructura, contenido y aplicaciones que se van a desarrollar

Etapas del diseño de base de datos:

1) DISEÑO CONCEPTUAL

- Se obtiene una estructura de la información de la futura BD
- No se tiene en cuenta que tipo de modelo de BD se utilizara
- No se tiene en cuenta con qué DBMS ni con qué lenguaje concreto se implementara la BD
- Nos concentramos únicamente en la estructuración de la información
- El resultado se expresa mediante algún modelo de datos de alto nivel
- Uno de los más empleados es el modelo ER

2) DISEÑO LÓGICO

- Se parte del resultado del diseño conceptual
- Es preciso ajustar al modelo con el que se desea implementar e.j. Si es un MBD relacional obtendremos un conjunto de relaciones, atributos, claves primarias y foráneas

3) DISEÑO FÍSICO

- Se transforma la estructura obtenida en el diseño lógico con el objetivo de conseguir mayor eficiencia
Ej. en una BD Relacional: Partir una relación en varias o añadir un atributo a una relación, etc

DISEÑO CONCEPTUAL - MODELO ER

- Modelo conceptual orientado a entidades. Se basa en una técnica de representación gráfica que incorpora información relativa a los datos y las relaciones existentes entre ellos
- Introducida por Peter Chen en 1976
- El modelo E/R permite visualizar la base de datos desde un alto nivel de abstracción. Los elementos interesantes de la realidad que queremos modelar son las entidades; además, modelamos sus atributos y las interacciones entre ellas.
- No tiene en cuenta con qué DBMS se implementara la BD ni qué aplicaciones consumen la BD
- No toma en cuenta restricciones no funcionales como ser espacio, almacenamiento y tiempo de ejecución
- El modelo E/R utiliza una representación gráfica conocida como diagrama entidad-relación (DER). Es importante mencionar que existen distintas representaciones de un DER en las que cambian los aspectos gráficos, pero se modelan los mismos elementos
- El modelo entidad-relación (E/R) ayuda a realizar un diseño de bases de datos sin atender a un modelo en especial —jerárquico, relacional, orientado a objetos—.

ENTIDAD

- Objeto real o abstracto que podemos distinguir del resto de objetos y del que nos interesan algunas propiedades
- Ejemplos: Empleado, Producto, Asignatura, Pedido
- Se representa con un rectángulo, Mayúsculas

ATRIBUTOS

- Cada una de las propiedades o características que tiene una entidad
- Ejemplos: Carnet, Código, Nombre, etc
- Minúsculas, descriptivos
- Tipos
 - Identificadores
 - Monovaluados
 - Multivaluados
 - Derivados o calculados

RELACIÓN

- Es una correspondencia entre dos o más entidades
- Se representa con un rombo el cual enlaza a 1, 2 o más entidades
- Normalmente se asocia a una acción (o verbo)

CARDINALIDAD

- La cardinalidad es obtenida en base a las posibilidades de relación entre las entidades
 - Uno a uno (1 : 1)
 - Uno a muchos (1 : N)
 - Muchos a muchos (N : N)
- Cardinalidad 1:1, que es cuando una entidad A se relaciona solo con otra entidad B y viceversa. Por ejemplo una alcaldía solo puede pertenecer a una ciudad
- Cardinalidad 1:N, que es cuando una entidad A se puede relacionar con N entidades B pero no al revés. Por ejemplo un despacho puede tener N EMPLEADOS pero un empleado es solo de un despacho o Un libro puede tener N ejemplares, pero un ejemplar es solo de un libro.
- Cardinalidad N:M, que es cuando una entidad A se relaciona con N entidades B y viceversa. Por ejemplo, un estudiante puede inscribirse a varias materias y una materia puede ser tomada por varios estudiantes

GRADO DE RELACIÓN

- Es el número de entidades que participan en la relación
 - Reflexivas
 - Binarias
 - Ternarias
- Decimos que una relación es Reflexivas (Grado 1) cuando se relaciona con ocurrencias de la propia entidad ejemplo
- Decimos que una relación es Binarias (Grado 2) cuando participan dos entidades ejemplo
- Relación es Ternaria o de Grado 3 es aquella relación donde participan tres entidades ejemplo

PASOS PARA CONSTRUIR UN DIAGRAMA ER

1. Identificar las entidades
2. Describir los atributos (campos que representan valores que queremos almacenar para cada entidad)
3. Determinar las claves primarias (es decir, que identifican de manera única a cada entidad)
4. Establecer las relaciones entre las entidades
5. Definir la cardinalidad de las relaciones

ACTIVIDAD 1 - Se crea

DISEÑO LÓGICO

- Basados en el diseño conceptual:
 - Todo tipo de entidad se convierte en Relación (Tabla)
 - Las interrelaciones 1:1 y 1:N dan lugar a claves foráneas (Para interrelaciones 1:N se realiza una propagación de clave a partir del lado 1 hacia el lado N)
 - Las interrelaciones N:N se traducen en nuevas relaciones (Entidades o tablas)

NORMALIZACIÓN

Técnica que busca dar eficiencia y fiabilidad a una BD relacional. Busca el aprovechamiento de espacio y manejar información de manera rápida.

PROBLEMAS

- **Redundancia:** Duplicidad de datos continua e innecesaria
- **Ambigüedades:** Datos que no clarifican suficientemente el registro al que representan
- **Pérdida de restricciones de integridad:** Normalmente debido a Dependencias Funcionales
- **Anomalías en operaciones de modificación de datos**

La normalización nos permite eliminar estos problemas forzando a la división de una tabla en dos o más

PRIMERA FORMA NORMAL

Una tabla se encuentra en primera forma normal si impide que un atributo de una tupla pueda tomar más de un valor

SEGUNDA FORMA NORMAL

Toda clave principal debe hacer dependientes al resto de atributos, si hay atributos que dependen solo de parte de la clave entonces esa parte de la clave y esos atributos formarán otra tabla

TERCERA FORMA NORMAL

Ningún atributo que no sea clave depende funcionalmente de ningún otro atributo no clave

Dependencia Funcional: Existe DF entre dos atributos de una tabla si para cada valor del primer atributo existe un solo valor del segundo

Cumplir las reglas de normalización hacen que la BD sea flexible, adaptable y escalable, al estar normalizada, cada porción de los datos es atómica, es decir está dividida en parte útiles lo más pequeñas posibles.