

Definiciones Básicas

Base de Datos

Conjunto de datos relacionados entre sí con las siguientes propiedades:

representa algún aspecto del mundo real

Lógicamente coherente

Dirigida a un grupo de usuarios específicos

“Una base de datos tiene una fuente de la cual derivan los datos (mundo real), cierto grado de interacción con los acontecimientos del mundo real, y un público activamente interesado en su contenido”

Sistema de Gestión de Bases de Datos - SGBD (en Inglés DBMS)

Software que permite a los usuarios crear y mantener una Base de Datos

Es de propósito general

Definir => especificar los tipos de datos, interrelaciones y restricciones entre ellos

Construir => almacenar los datos (usando el SGBD)

Manipular => actualizar, consultar

Es posible tener una Base de Datos sin usar un SGBD (así se hacía antiguamente).

A veces se llama Sistema de Bases de Datos al conjunto Base de Datos + SGBD

Bases de Datos vs. Archivos

Separación entre Programas y Datos

En sistemas basados en archivos, la estructura de ellos aparece integrada al interior de los programas

Cambio en definición => modificar y recompilar muchos programas

Al usar un SGBD la estructura de los archivos (o tablas) se almacena en el catálogo, separada de los programas de acceso

EL almacenar la Base de Datos + Descripción de la Base de Datos (metadatos) es una de las características distintivas de este enfoque

Esta característica se denomina *independencia de los programas con respecto a los datos*

SGBD ofrece al usuario una *representación conceptual* de los datos sin los detalles de cómo se almacenan => *Modelo de Datos*

Seguridad e Integridad

La información es un recurso muy importante de la organización =>

Control de acceso (quien puede hacer qué con qué dato)

Integridad (asegurar información consistente, eliminar redundancia, etc)

Recuperación ante fallas

Bases de Datos vs. Archivos (cont)

Acceso Concurrente a la Información

SGBD provee de mecanismos para asegurar la operación correcta del sistema en un ambiente de muchos usuarios simultáneos

Permite disponer de información actualizada de inmediato en todos lados (“on-line”)

Menor tiempo para construir aplicaciones

Muchas de las aplicaciones en ambientes de Sistemas de Información consisten en generar un nuevo informe o extraer información hacia pantallas

Una vez que la BD está creada obtener ese tipo de aplicación es trivial

En general, se estima que en un ambiente con SGBD, el tiempo de desarrollo de nuevas aplicaciones es de 1/6 a 1/4 de lo requerido con programación tradicional (Cobol)

Diferentes interfaces

El SGBD puede ofrecer diversas interfaces para los distintos actores:

- lenguajes de consulta

- Formularios

- Interfaz con lenguaje de programación

- Interfaz con menus

Breve Historia

1961

Bachman diseña el primer SGBD, Integrated Data Store (IDS) de GE

1965 - 1970

IBM desarrolla su sistema IMS (Information Management System) que constituyó el paradigma de SGBD jerárquico
IBM y AA crean SABRE

1970

Ted Codd (asociado a IBM) desarrolla el modelo de datos relacional

1971

Informe del grupo de trabajo sobre BD (DBTG) de CODASYL

1975

Primera conferencia internacional del SIGMOD
Primera conferencia internacional del VLDB

1976

Chen introduce el modelo Entidad-Vínculo (ER)

Años 70's

Varios proyectos de investigación:

System R (IBM)

INGRES (UC Berkeley)

System 2000 (U. of Texas)

ADABAS (U.T de Darmstadt)

Se desarrollan lenguajes de consulta:

SQUARE, SEQUEL, SQL, QBE, QUEL

Breve Historia (cont)

Años 80's

Desarrollo de SGBD para computadores personales
DBASE, Paradox, etc.

1983

Estudio de ANSI/SPARC revela mas de 100 sistemas relacionales implementados (DB2, ORACLE, SYBASE, INFORMIX, etc.)

1985

Se publica norma preliminar de SQL
Aparecen SGBD orientados a objetos
Arquitecturas Cliente-Servidor
Bases de Datos distribuidas

Años 90's

Demanda por extender capacidades del SGBD a nuevas aplicaciones (datos espaciales y temporales, multimedia, capacidades activas y deductivas)
Aparición de SGBDOO comerciales
Mejora del rendimiento de los SGBD comerciales
Demanda por aplicaciones que utilicen datos de diversas fuentes (sistemas heterogéneos)

Arquitectura General de 3 Niveles

Nivel Interno (Esquema Interno)

Describe estructura física de la base de datos

Emplea modelo físico

Describe detalles de almacenamiento y caminos de acceso

Nivel Conceptual (Esquema Conceptual)

Describe estructura de toda la BD para la comunidad de usuarios

Oculto los detalles de estructuras de almacenamiento

Describe entidades, tipos, vínculos, operaciones, restricciones

Nivel Externo (Vistas de Usuario)

Describe una parte de la BD que interesa a un grupo de usuarios determinado

Arquitectura de 3 niveles permite obtener independencia física y lógica c/r a los datos

Independencia Lógica

Modificación de esquema conceptual no requiere alterar esquemas externos ni programas de aplicación

Independencia Física

Modificación de esquema interno no requiere modificar esquema conceptual (ni externos)

Lenguajes

En sus inicios ...

DDL - Lenguaje de definición de datos => descripción de esquema conceptual

SDL - Lenguaje de definición de almacenamiento => descripción de esquema interno

DML - Lenguaje de manipulación de datos => inserción, eliminación, actualización

Los SGBD actuales unifican todo esto en un sólo lenguaje integrado que combina todos ellos => SQL

Dos tipos de DML

Alto nivel (no procedural, operaciones complejas) también llamado lenguaje de consulta

Bajo nivel (procedural, registro a registro sobre lenguaje huesped)

Interfaces

Interfaz basada en menús

Se presenta al usuario opciones en forma de menús (no se requiere memorizar comandos ni sintaxis)

Interfaces gráficas

Se presenta esquema en forma de diagrama

Consulta se hace manipulando el diagrama

Puede combinarse con la anterior (menú)

Interfaces de formularios

Usuario llena ciertos espacios del formulario, y el sistema busca los registros que coincidan

Interfaz de lenguaje natural

Solicitud en Inglés

Hay un diccionario de palabras

Interfaz para usuarios paramétricos

Usuario paramétrico típico -> cajero de un banco

Interfaz especial con conjunto muy reducido de opciones

Pueden usarse teclas de funciones, etc.

Interfaz para el DBA

Hay operaciones que solo el administrador de la BD puede realizar

- Crear cuentas

- Modificar esquemas

- Reorganizar estructura de almacenamiento de la BD

- establecer parámetros del sistema

Otros Actores

Administrador de la BD (DBA)

Responsable de administrar los dos recursos fundamentales de este enfoque:

La Base de Datos misma

El SGBD

Diseñadores de la BD

Identificar los datos necesarios

Crear esquemas en los distintos niveles

Usuarios Finales

Esporádicos (diferente info cada vez, lenguaje de consulta)

Simple o Paramétricos (transacciones programadas)

Avanzados (ingenieros, científicos, analistas)

Analistas de Sistemas y Programadores

Analistas =>

determinan requerimientos de los usuarios (simples o paramétricos)

Especificación de transacciones programadas

Programadores =>

implementación de especificaciones

Prueba, documentación, depuración

Diseño Conceptual (Modelo E-R)

El diseño conceptual de la BD es una etapa importantísima

Descripción concisa de los requerimientos de info de los usuarios

Contiene descripciones detalladas de tipos de datos, vínculos y restricciones expresadas en lenguaje de alto nivel (diagramas)

Es fácil de entender => herramienta de comunicación con usuarios no técnicos

El modelo Entidad-Vínculo (E-R) es uno de los lenguajes de alto nivel mas utilizado para este fin

Después de tener este diseño conceptual debe llevarse a un diseño lógico sobre el modelo soportado por el SGBD (típicamente relacional)

Conceptos Básicos del Modelo E-R

Entidades y Atributos

Entidad es un objeto del mundo real (físico o conceptual)

Persona, auto, empleado, curso, empresa, etc.

Atributos son propiedades específicas que sirven para describir las entidades

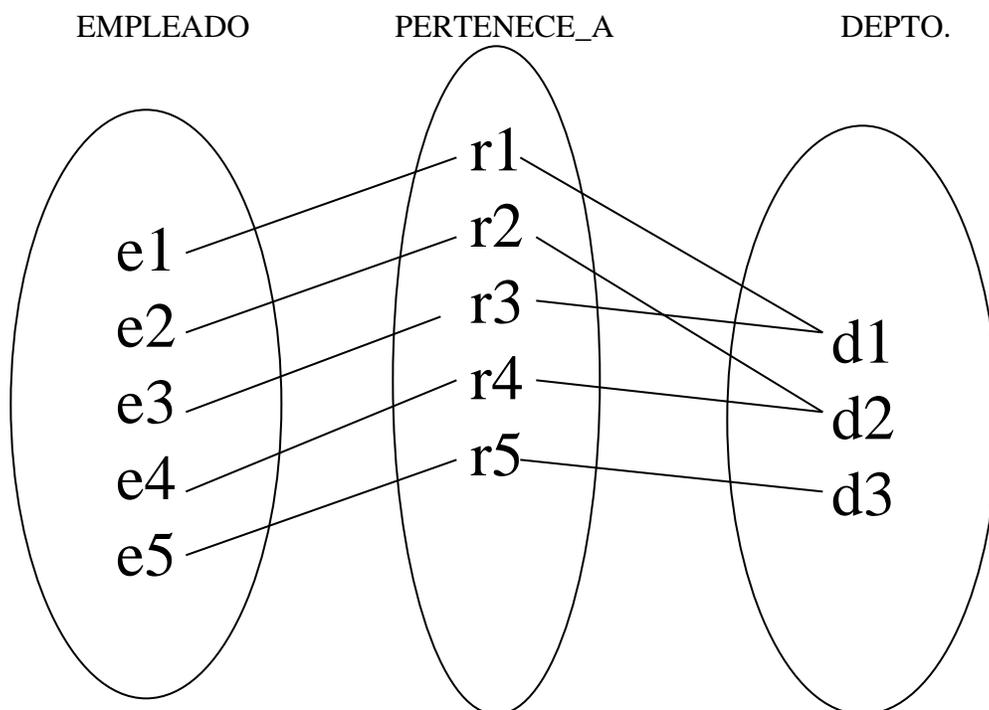
Ej: Empleado puede tener como atributos: nombre, edad, dirección, sueldo

Cada uno de estos atributos tendrá asociado un valor para una entidad determinada

Vínculos

Vínculo (relationship) permite asociar dos o más entidades

Cada instancia de un vínculo R es una asociación de una de cada una de los tipos de entidades que participan en ella (ver ejemplo)



Mas Detalles ...

Atributos

Simple (nombre, edad) o Compuestos (dirección: calle, ciudad, país, etc.)

Monovaluados (edad) o Multivaluados (colores, grados académicos, etc.)

Almacenados (fecha nacimiento) o derivados (edad)

Atributo Clave => identifica de manera única la entidad

Vínculos

binarios - participan dos entidades (la mayoría)

ternario - participan tres entidades

grado n - participan n entidades

A veces una misma entidad participa mas de una vez en un vínculo.

vínculos recursivos

entidades se distinguen asignando un *rol* a cada uno de ellos

Ejemplo: Supervisa_A corresponde a un vínculo binario entre Empleado y Empleado pero existen dos roles: supervisor y supervisado

Razón de cardinalidad

numero de instancias de vínculos en que puede participar una entidad.

en el caso de vínculos binarios podemos tener:

1:1 - DIRIGE entre EMPLEADO y DEPTO

1:N - TRABAJA_EN entre DEPTO y EMPLEADO

M:N - la misma anterior si los empleados pueden trabajar en mas de un Depto.