

Base de datos3

Modelos de BD

Modelos Relacionales y otros: El sistema relacional es el mas utilizado. Para estudiarlos los dividiremos en sistemas pre-relacionales y sistemas relacionales.

•**Sistemas Pre-Relacionales:** Ejemplos de sistemas gestores pre-relacionales son:*Sistema Jerarquicos.*Sistemas de red.*Sistema de lista invertida.

Caracteristicas:-En general,son anteriores a los sistemas relacionales y mas primitivos.-No se desarrollan sobre una base teorica solida. Solo los SGBDR se construyen sobre una base teorica formal.-Tienen un nivel de abstraccion mas bajo y poca independencia fisica y logica.-Estan muy vinculados a los sistemas de almacenamiento subyacente.-En general son sistemas de programacion.el unico usuario posible es el desarrollador de aplicaciones.-En general trabajan registro a registro.

-**Modelo Jerarquico:** Una BD jerarquica es un conjunto de ejemplares de un tipo de arbol. Forma de trabajo: se trabaja con 2 objetos: Registros: Un registro padre puede tener un numero indeterminado de hijos, pero un hijo no puede tener mas de un padre. Ligaduras.Forma de guardar informacion: Se informacion se guarda arbol a arbol, de izquierda a derecha y de arriba a abajo. Manipulacion de los datos: seleccionar un arbol determinado.Pasar de un arbol a otro. Pasar de un padre a su primer hijo. Pasar de un registro a otro dentro del mismo nivel. Insertar/borrar registros de un nivel.Mantenimiento de la integridad: Referencial. No se puede tener un hijo sin su padre.Permite borrados y actualizaciones en cascada.Ventajas: Muy eficientes en accesos asociados con la jerarquia del arbol.Inconvenientes: Un pequeño cambio en el programa puede suponer grandes modificaciones en el sistema de base de datos.

-**Modelo de red:** Este modelo se ejemplificara mediante el producto comercial IDMS de la casa Computer Associates. Puede considerarse como una forma extendida de la jerarquia, en el que un registro puede tener un numero ilimitado de padres.Forma de trabajo: se trabajan con 2 elementos: Registro.Ligadura:una ligadura tiene un padre para varios hijos. Manipulacion de los datos: Localizar un registro.Pasar de un registro padre a su primer hijo de una ligadura.Moverse de un hijo a otro. Pasar de un hijo a su padre. Ventajas:Se mantiene la integridad referencial. Actualizaciones y borrados en cascada.Inconvenientes: Un pequeño cambio en el programa puede suponer grandes modificaciones en el sistema de BD.

-**El modelo de lista invertida:** Estan basados en sistemas de ficheros o sistemas de navegacion, como por ejemplo el caso de clipper o dBase. Forma de trabajo: Se trabaja con 3 tipos de elementos: *Archivos de datos: Se perciben como tablas en que las filas son registros y las columnas son los campos. Los registros tienen una ordenacion fisica.*Archivos de indices: los utiliza el programador mediante codigo para acelerar los accesos de datos.*Un lenguaje de programacion que maneje estas estructuras.Manipulacion de los datos: la localizacion de un registro en un archivo.Localizacion de un registro utilizando indice.Acceso directo a un registro.Desplazamientos relativos. Leer-modificar-borrar el registro actual. Ventajas: Son muy rapidas para aplicaciones cerradas y con pocos recursos de maquina. Inconvenientes: Mala ejecucion de consultas AD-HOC o las consultas dia a dia. Las modificaciones son complicadas.No se incluyen reglas de integridad. El programador tiene que implementarlas en sus aplicaciones.

•**Sistemas Relacionales:** Ejemplo de este tipo de sistemas gestores son: DB2 de IBM. Ingress. Informix. Oracle. SQL Server de Microsoft. MS Access de Microsoft. Caracteristicas generales: *A diferencia de los modelos vistos hasta ahora, el modelo relacional si que se basa en un modelo formal.*El modelo relacional fue creado por un matematico que trabajaba para IBM llamado E.F. Codd.*Resuelven los fallos que poseian los gestores antecesores a estos, como lo son: la inconsistencia, redundancia de datos, falta de independencia fisica/logica y poca flexibilidad.Bases

de modelo relacional: *Estructura de datos: Los datos siempre se perciben en forma de tablas. El modelo relacional hace la abstracción del carácter lógico o físico de estas estructuras. Estas estructuras tienen una primera fila con valores fijos y un conjunto de filas variables en el tiempo.

Propiedades de las relaciones: No existen filas repetidas. No existe orden. Los valores de los campos son autónomos. Estas propiedades se derivan de considerar las relaciones como conjunto en el sentido matemático.

Tipos de relaciones: *Tablas o relaciones base: Son los verdaderos contenedores de datos. Son persistentes.*Vistas o tablas virtuales: Solo se guardan las definiciones que son persistentes. no guardan datos.*Resultados de consultas: No tienen datos, ni se guardan sus definiciones.*Tablas temporales: Si que contienen datos pero no son persistentes.*Operadores entre estructuras.*Definiciones de integridad.

Vistas: son tablas virtuales derivadas de tablas base y referenciadas por un nombre. No contienen datos, solo se almacenan su definición en la BD.

Unidades de las vistas: *Obtener una vista de los datos optimizados según las necesidades de los usuarios que no necesitan conocer los detalles del diseño de la BD. Ofrecen diferentes formas de ver los datos.*Ocultar filas o columnas por motivos de confidencialidad.*Aumentar la independencia lógica de los datos.

Actualizaciones: Existen actualizaciones, es decir, que permiten modificar desde las vistas los datos de las tablas base a que pertenecen. No todas las vistas son actualizables.

*Incluyan todos los campos que constituyen la clave. Si montásemos una vista sin uno de los campos que constituye la clave, al modificar los datos de la vista, ese campo que es parte de la clave se rellenaría con un nulo y estaría violando una de las reglas de la integridad de clave primaria.*Incluir todos los campos que tengan restricciones a nivel de esquema. *Nunca son actualizables consultas de totalización.

Vistas en que se incluyen más de una tabla: *Se permitirán actualizaciones en dos tablas con una relación 1:1.* Se permitirán actualizaciones en el lado n de una relación 1:n. El lado n siempre estará en la tabla que contiene la clave externa.

Definiciones de integridad: definiciones almacenadas a nivel de esquema que controlan el conjunto de valores posibles para los campos. Estas definiciones las mantiene y las observa el DBMS. Existen 2 reglas muy importantes que se deben tener en cuenta cuanto a la integridad de los datos se refiere. A saber: **1. Regla de integridad de entidades o integridad de clave primaria**: Clave primaria, clave, identificador o primary key. Campos de una tabla que identifican de manera unívoca un registro de una tabla. Los hay simples y compuestos. *Claves candidatas*. Campos que pueden ser claves primarias. *Claves alternas o alternativa*: aquellas claves candidatas que no se ha elegido como clave principal.

Criterios para escoger claves: Universalidad. Minimalidad. Sencillez.

Reglas de integridad de claves primarias: *ningún componente de la clave primaria puede aceptar nulos.*La clave primaria siempre es única.

2. Regla de integridad de claves externas. Integridad referencial: Clave externa, ajena, foránea o foreign key: campos de una tabla en que los valores deben concordar con los datos de la clave primaria de la otra tabla. *Todos los valores de la clave externa deben existir en la tabla en que son clave principal. *Se aceptan nulos?