**BASES DE DATOS JERARQUICAS**

Características

* Consisten en una colección de registros que se conectan entre sí por medio de enlaces.
* Fue uno de los primeros modelos que surgieron con la creación del concepto de “base de datos”. Hoy en día no está muy extendido su uso, porque fueron desplazados por las bases de datos relacionales, de uso más común en las empresas.
* La organización de estas bases se establece en forma de árbol donde la raíz es un nodo ficticio, es decir no contiene realmente registros que estarán en los niveles inferiores.
* El contenido de un registro específico podría repetirse en varios sitios en el mismo árbol.
* Se utilizan punteros para recorrer los nodos que apuntan al registro anterior y al siguiente. El modelo no diferencia la vista lógica de una vista física, de manera que las relaciones entre datos se establecen siempre a nivel físico mediante referencia a direcciones físicas del medio de almacenamiento.
* Se estructura en niveles múltiples de acuerdo a una relación estricta padre hijo, donde un padre puede tener más de un hijo. pero un hijo únicamente puede tener un padre.
* El árbol se organiza en un conjunto de niveles.
* El nivel cero que se corresponde al nodo raíz y es el nivel mas alto de la jeraquía.
* Los arcos o enlaces representan las asociaciones jerárquicas entre 2 nodos.
* Todo no debe tener un padre, a excepción del nodo raíz que no tiene antecesor.
* Los nodos del nivel más bajo, sin descendientes se llaman hojas.
* Sólo pueden existir relaciones de uno a uno o de uno a muchos.
* La estructura del árbol no se puede modificar cuando ha quedado establecida.

Acá tenemos un diagrama de la estructura en árbol de una Base de datos jerárquica desde

punto de vista lógico tiene 3 niveles: departamento profesor y curo.

El esquema de nodos y enlaces muestra **un único nodo raíz**, que es el padre de los nodos hijos profesor y **cada nodo profesor a su vez es padre de los nodos hijos curso**. Estos **nodos curso son nodos hojas ya que no tienen hijos**. Vemos como aparece el problema de la **redundancia:** un curso puede tener varios profesores de distintas materias. En nuestro ejemplo **el nodo curso uno está repetid, ya que tanto el profesor 1 como el profesor 2 dictan alguna materia en ese curso.** Debido a las restricciones de la estructura, un nodo hijo puede tener un solo padre, por lo anto el nodo curso uno no puede conectarse con el nodo profesor uno y profesor 2 directamente.

Diagrama

Descripción generada automáticamente

Diagrama, Forma

Descripción generada automáticamente

La recuperación de datos en una base jerárquica se hace siguiendo un esquema denominado **Preorden**. Se recorren los nodos yendo de los nodos padres a los hijos y de izquierda a derecha. El recorrido de acceso a los nodos de este árbol ejemplo sería: parte del nodo 1, va a nodo 2, o sea el primero a la izquierda y abajo. Luego al nodo 3, de ahí vuelve al nodo 2 y después va al 4. Del 4 va al 5, después vuelve al cuatro y de ahí va al 6. Por último, vuelve al cuatro ,vuelve al 2 vuelve al uno y va al 7 y así con lo siguiente.

La base de datos jerárquicas tiene **LIMITACIONES** por las que al menos parcialmente fueron desplazadas del mercado:

* Las operaciones de insertar y borrar nodos son complejas.
* Al borrar un nodo padre desaparecen también sus nodos subordinados, rompiéndose el recorrido del grafo. En nuestro ejemplo, si elimináramos el nodo profesor 3, el nodo curso cuatro queda desconectado e imposible de acceder.

Diagrama

Descripción generada automáticamente

?

* Sólo podrá añadirse un nodo hijo si existe el nodo padre
* Lo más significativo es la rigidez de su estructura: solo un padre por hijo o sea un nodo hijo no puede conectarse con 2 padres y ausencia de relaciones entre los nodos hijos entre sí. Siempre se debe volver al nodo padre para cambiar de rama en el grafo.

**MANIPULACIÓN DE DATOS**

Para la manipulación de los datos, el sistema gestor de esta base de datos debe poder:

* Seleccionar un árbol determinado.
* Pasar de un árbol a otro.
* Pasar de un padre a su primer hijo.
* Pasar de un registro a otro, dentro de un mismo nivel pasando por el padre.
* Insertar o borrar registros de un nivel y cada registro tiene un puntero al anterior y al otro siguiente. Para eliminar o insertar un registro se deben modificar los punteros anteriores y siguientes

Sí tenemos, por ejemplo, 3 registros A, B y C, el registro a tiene un puntero a su registro anterior, en este caso su padre y un puntero al registro siguiente B. El registro B tiene un puntero a su anterior, A, y a su siguiente, C. El registro C, a su vez tiene un puntero a su anterior, B y a su siguiente, que es la vuelta al nodo padre.

Diagrama

Descripción generada automáticamente con confianza media

Si quisiéramos eliminar el nodo B, el registro A debería modificar su puntero al siguiente que ahora es C, y el nodo C debe cambiar su anterior, que ahora es A. Para insertar un nodo, por ejemplo, si queremos insertar uno nuevo entre el nodo A y el C, debería modificarse de la misma forma los punteros de A y C, en este caso apuntando al nuevo registro.

**DESVENTAJAS** por la cual este modelo de base de datos no sea tan utilizado hoy en día:

* Puede dar lugar a inconsistencia de los datos cuando se llevan a cabo las actualizaciones.
* Inevitable el desaprovechamiento del espacio.
* La recuperación de la información de una unidad que se encuentra bajo varios niveles requiere navegar por un camino extenso a través de unidades y sus relaciones hasta llegar a ella. haciendo que el tiempo de recuperación el mejor
* Al permitir sólo relaciones de uno a muchos se genera redundancia automáticamente