Texto

Descripción generada automáticamente con confianza media

Característica de base de dato jerarquica

Texto

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Acá tenemos un diagrama de la estructura en árbol de una base de datos jerárquica. Desde el punto de vista lógico, tiene tres niveles: departamento, profesor y curso. 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 cómo aparece el problema de la redundancia: un curso puede tener varios profesores de distintas materias. En nuestro ejemplo, el nodo curso uno está repetido ya que tanto el profesor uno como el profesor dos dictan alguna materia en ese curso. Debido a las restricciones de la estructura, un nodo hijo puede tener un solo padre, por lo tanto, el nodo curso uno no puede conectarse con el nodo profesor uno y profesor dos directamente.

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como dijimos, las bases de datos jerárquicas tienen 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 hay nodos con dos 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.

Texto

Descripción generada automáticamente

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 mismo nivel. Cada registro tiene un puntero al anterior y al siguiente. Para eliminar o insertar un registro, se deben modificar los punteros anteriores y siguientes. Si tenemos, por ejemplo, tres 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. 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. 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, deberían modificarse de la misma forma los punteros de A y C, en este caso apuntando al nuevo registro.

Diagrama

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para cerrar el tema, consideremos un resumen de las desventajas por la cual este modelo de base de datos no sea tan utilizado hoy en día. Por un lado, puede dar lugar a inconsistencia de los datos cuando se llevan a cabo las actualizaciones. Puede resultar 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 no sea el mejor. Y al permitir solo relaciones de uno a muchos, se genera automáticamente.