**Bases de Datos Jerárquicas**

**Definición**:   
Una base de datos jerárquica es un tipo de sistema de gestión de bases de datos que almacena la información en una estructura jerárquica que enlaza los registros en forma de estructura de árbol, en donde un nodo padre de información puede tener varios nodos hijo, y así sucesivamente, esto suele denominarse relación uno a muchos. Los nodos inferiores se subordinan a los que se hallan a su nivel inmediato superior. Un nodo que no tiene padre es llamado raíz, en tanto que los que no tienen hijos son conocidos como hojas.

**Estructura**:  
A diferencia del modelo relacional, el modelo jerárquico no diferencia una vista lógica de una vista física de la base de datos. De manera que las relaciones entre datos se establecen siempre a nivel físico, es decir, mediante referencia a direcciones físicas del medio de almacenamiento (sectores y pistas).

Los **datos** se almacenan en la forma de **registros**, el equivalente a las tuplas del modelo relacional. Cada registro consta de un conjunto de **campos**, el equivalente a los atributos del modelo relacional. Un conjunto de registros con los mismos campos se denomina **fichero** (**record type**, en inglés), el equivalente a las relaciones del modelo relacional.

Cada nodo del árbol representa un tipo de registro conceptual, es decir, una entidad. A su vez, cada registro o segmento está constituido por un número de campos que los describen - las propiedades o atributos de las entidades. Las relaciones entre entidades están representadas por las ramas.

**Funcionamiento:**

El modelo jerárquico facilita relaciones padre-hijo, es decir, relaciones 1:N (de uno a varios) del modelo relacional. Pero a diferencia de éste último, las relaciones son unidireccionales. En justicia, dichas relaciones son hijo-padre, pero no padre-hijo. De esta forma, tampoco existen relaciones N:M (de muchos a muchos) en el modelo jerárquico. Salvo que se simulen mediante varias relaciones 1:N. No obstante, esto puede provocar problemas de inconsistencia, ya que el gestor de base de datos no controla estas relaciones.

Las relaciones se establecen mediante punteros entre registros. Es decir, un registro hijo contiene la dirección física en el medio de almacenamiento de su registro padre. Esto tiene una ventaja fundamental sobre las bases de datos relacionales: el rendimiento. El acceso de un registro a otro es prácticamente inmediato sin necesidad de consultar tablas de correspondencia.

Las relaciones jerárquicas entre diferentes tipos de datos pueden hacer que sea muy sencillo responder a determinadas preguntas, pero muy difícil el contestar a otras.

Una base de datos de tipo jerárquico recorre los distintos nodos de un árbol en un orden que requiere tres pasos:

1. Visitar la raíz.
2. Visitar el hijo más a la izquierda, si lo hubiera, que no haya sido visitado.
3. Si todos los descendientes del segmento considerado se han visitado, volver a su padre e ir al punto 1.

**Historia**:  
Se dice que comienza a mediados de los 60 con el proyecto Apolo (hombre a la luna), para gestionar la inmensa cantidad de datos que requería el mismo. La encargada NAA (North American Aviation), desarrolló el software GUAM (General Update Access Method) basado en el concepto árbol, para poder unir varias piezas pequeñas en una pieza más grande, y así sucesivamente hasta ensamblar el producto final. A mediados de los 60, NAA se une con IBM para perfeccionar a GUAM en el conocido software IMS (Information Management System).

**Desventajas:**

Todos sus problemas derivan del hecho de que el sistema gestor de base de datos no implementa ningún control sobre los propios datos, sino que queda en manos de las aplicaciones garantizar que se cumplen las condiciones invariantes que se requieran (por ejemplo, evitar la duplicidad de registros). Dado que todas las aplicaciones están sujetas a errores y fallos, esto es imposible en la práctica.

Duplicidad de registros: No se garantiza la inexistencia de registros duplicados. Esto también es cierto para los campos "clave". Es decir, no se garantiza que dos registros cualesquiera tengan diferentes valores en un subconjunto concreto de campos.

Integridad referencial: No existe garantía de que un registro *hijo* esté relacionado con un registro *padre* válido. Por ejemplo, es posible borrar un nodo *padre* sin eliminar antes los nodos *hijo*, de manera que éstos últimos están relacionados con un registro inválido o inexistente.

Desnormalización: Este no es tanto un problema del modelo jerárquico como del uso que se hace de él. Sin embargo, a diferencia del modelo relacional, las bases de datos jerárquicas no tienen controles que impidan la desnormalización de una base de datos. Por ejemplo, no existe el concepto de campos clave o campos únicos.

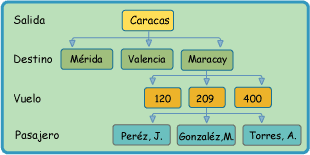
La desnormalización permite ingresar redundancia de una forma controlada, seguir a una serie de pasos conlleva a:

* Combinar las relaciones
* Duplicar los atributos no claves
* Introducción de grupos repetitivos
* Crear tablas de extracción

Cuando se debe desnormalizar:

* Se debe desnormalizar para optimizar el esquema relacional
* Para hacer referencia a la combinación de 2 relaciones que forman una sola relación

**Ejemplos**:  
a) Sistema de reservaciones de una línea aérea:

El Nodo Padre en esta base de datos jerárquica es la Ciudad de Salida en este caso es (Caracas), Nodos Hijos representando las Ciudades Destino que tiene a su vez Nodos Hijos, que son el Número de Vuelo. El Número de Vuelo tendrá también Nodos Hijos, que son los Pasajeros.  
  
  
  
b) Departamento académico:

Cada departamento es una entidad que mantiene una relación de uno a muchos con los profesores, que a su vez mantienen una relación de uno a muchos con los cursos que imparten.

