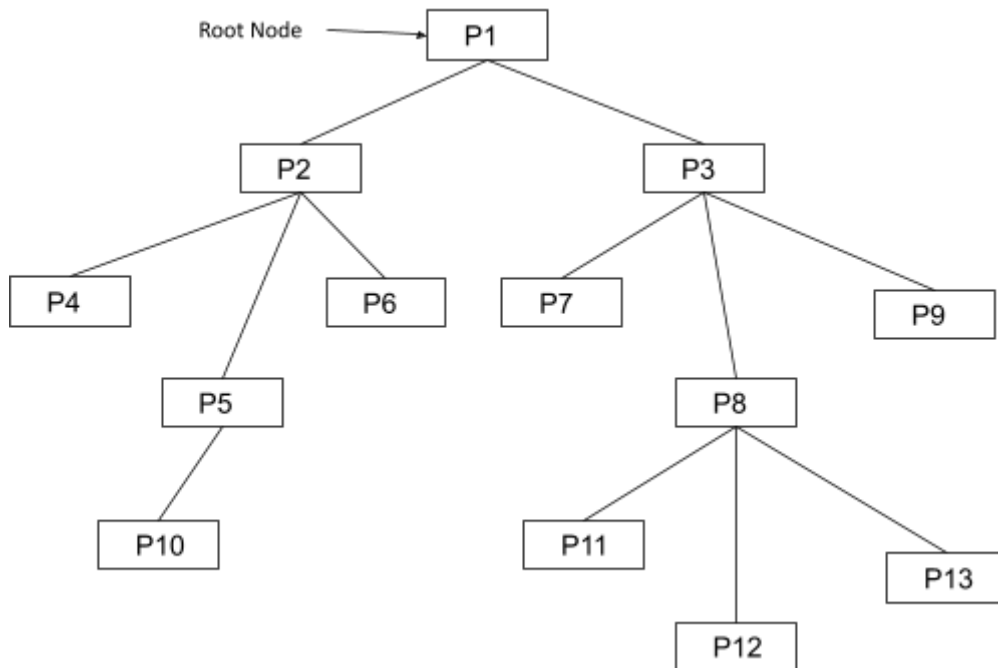


**Hierarchical database:**

In this kind of data model the data is stored in the form of records and organized into a tree-like structure, in which a node can have as many sub-nodes as it wants, connected through edges.



In the above diagram P1 is the root directory, and its children are P2 and P3. P2 is a parent to P4, P5, and P6, which in turn has children of its own.

The topmost node is known as root node and the nodes which don't have any children further are called leaf nodes.

All the records/files in this database are represented in parent-child manner, where each parent node has multiple child nodes (one or more), although every child node has only one parent node.

**Few features of Hierarchical Databases:**

- They have useful real world applications as some real-world occurrences of events are hierarchical in nature like biological structures, political, or social structures.
- Since the disk storage system is also inherently a hierarchical structure, these models can also be used as physical models.
- Deletion of parent node, will lead to deletion of child node.
- It supports one to many relationships but couldn't bear many to many relationships.

- As parent and child nodes are placed closer in the memory, the load of input and output on the hard disk are minimized.

**Advantages:**

- Promotes data sharing.
- Data Security is ensured.
- Data can be accessed quickly due to the links present between the relations.
- Performance is high
- Referential integrity is always maintained.
- It's design is simple to comprehend.

**Disadvantages:**

- No data manipulation or data definition language.
- Can't handle Complex Relations
- Redundancy which results in inaccurate information.
- Offers Poor flexibility i.e. a child entry needs to have parent entry, and also tasks like deletion, updation are quite hideous.

Some examples of Hierarchical databases are IBM Information Management System (IMS) and the RDM Mobile.