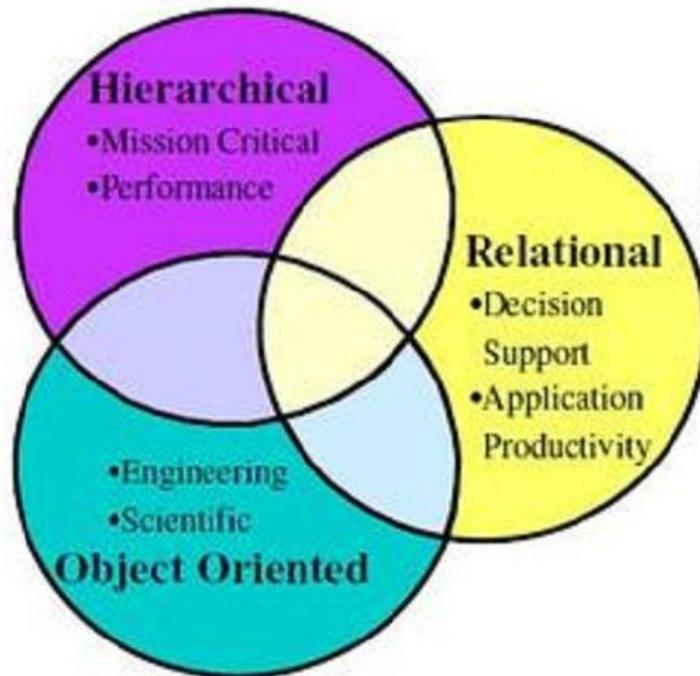


---

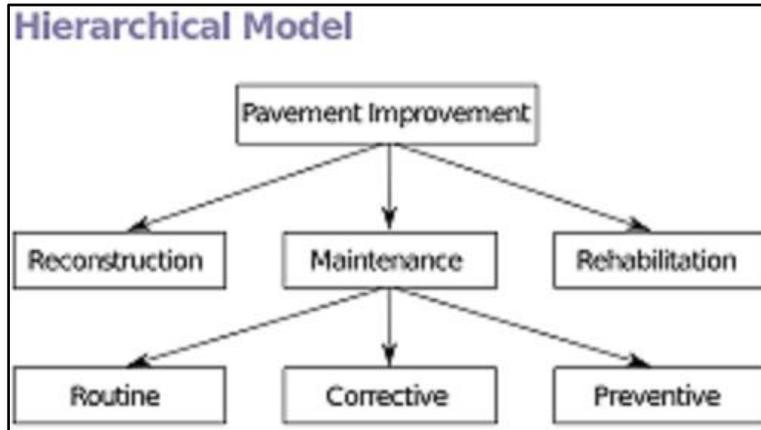
### Difference between Hierarchical Database and Relational Database



We all know that databases are framed to deal with data and its storage. Also, we are even confused about which database to use as we have lots of options to pick! Generally, we choose the database producer or the owner. Besides that, we can also select the right database for our need by analyzing its types such as Hierarchical, a Relational, a Network database, or an Object-oriented database.

#### What is a Hierarchical database?

In a hierarchical database, the data is organized in a tree-like structure. Each individual data is stored in a field and the fields, in turn, form records. These data are accessed with the help of links between them. In this structure, all the data records are linked finally to a single parent record. It is also called as the owner record. The links between the records are often described as parent-child relations. The best usage of hierarchical database is its deployment in a library system as it stores names or book numbers using the Dewey Decimal System. This system resembles a tree-structure by sharing the same parent number and then branches like trees. Similarly, we can use it to store names in a phone directory.



### What is a Relational database?

It stores data in the form of tables with unique keys to access the data. These tables supply the data in the required form with the help of using query languages. The interesting part is that it does not require any data re-grouping to fetch the data of our choice. It is often referred as Relational Database Management Systems (RDBMS).

Name	FName	City	Age	Salary
Smith	John	3	35	\$280
Doe	Jane	1	28	\$325
Brown	Scott	3	41	\$265
Howard	Shemp	4	48	\$359
Taylor	Tom	2	22	\$250

### Differences

- **Simpler to use:** The Hierarchical databases use the logical parent-child relationship and it looks simpler as well. But Relational databases involve tables to store records in the form of table fields. Also in most cases, it requires a unique key for each record.
- **Which is older?** The Hierarchical databases came into existence even before the Relational databases and it is the processor to all the other databases.
- **The Fundamental difference in the data notion:** In hierarchical databases, a category of data is termed as 'Segments' whereas in Relational databases it is termed as 'Fields'.



# Symas OpenLDAP

## How-To Guides

- **Inheritance:** Every child segment/node in a hierarchical database, inherits the properties of its parent. But in relational databases, there is no concept of inheritance as there are no levels of data.
- **Data Linking:** In hierarchical databases, the Segments are implicitly linked as a child is linked to its parent. But in relational databases, we should explicitly link the tables with the help of 'Primary keys' and 'Foreign keys'.
- **Use of keys:** The relational databases are generally framed with unique keys called the Primary key and also keys from other tables called the Foreign keys. These foreign keys are primary keys in some other table and it is referred while accessing the other table from this table. So, the main use of keys is to give a unique identification to the data records and to refer other tables during the data fetching process. But a hierarchical database never uses keys. It has its links to denote the path to be traversed during the data fetch. Therefore, we can consider the keys in relational databases as the equivalent of the paths in hierarchical databases during data fetches. But the paths never represent the uniqueness of data that has been stored in hierarchical databases.
- **Unique & Duplicate data:** As the keys represent the uniqueness of data in relational databases, we can easily list such data on demand. But when the same is required in a hierarchical database, it needs a lots processing. We can have more than one copy of the same book in a library but assigned with different book numbers. In this case, we should compare the book names to identify the duplicates. Therefore, relational databases are suited to store unique data whereas hierarchical databases are good ones for data with duplicates.
- **Data Fetching:** Just imagine that you have a library management system and it stores book details with an assigned book number for each book.  
Consider a book assigned with the book number as 1034. The data fetching process here is just given below.

- In a hierarchical database:

```
If book-no > 500 {... }  
If book-no > 1000 {  
  
If book-no >1500 {... }  
  
Else {if book-no >1100  
  
If book-no > 1050 {...}  
  
Else {if book-no >1025 {if book-no >1030 {if book-no > 1035 {...}  
  
Else {if book-no = 1031} ...  
  
Else {if book-no = 1032} ...  
  
Else {if book-no = 1033} ...  
  
Else {if book-no = 1034} ... Match found here
```



# Symas OpenLDAP

## How-To Guides

The above process takes place step-by-step as we reach a branch of the tree climbing from its trunk.

Consider that we need to fetch the 'date of birth' field whose employee-ID is 12345. Here the employee-ID is the primary key and we frame queries as below.

- In a Relational Database: Here, the data is fetched with the help of the Primary keys and the Foreign keys. We can directly access the required fields with its matching key.

*Fetch Employee-name, Employee-DOB*

*From Employee-table*

*Where employee-ID = '12345'.*

Here we can fetch the required fields directly and we need not beat about the bush.

- **Many-to-Many or One-to-Many data linking:** These kinds of data links are not possible with the hierarchical databases as a parent can have more than 1 child whereas a child cannot have more than 1 parent. In the latter case, we would encounter the many-to-one or the many-to-many data linking or relationship. But these kinds of data relationships are possible with relational databases.
- **The Fields in relational database Vs the Nodes in hierarchical database:** In relational databases, the data classification is based on the 'field' whereas in hierarchical databases it is based on the 'nodes or segments'. Every field is present in every record in relational databases. Similarly, we can see every segment in the final data i.e. book number, book name, etc. in the case of a library management system. This is often referred as the fundamental difference between the two databases, which we have mentioned at the initial stages of our article.
- **Where it finds its usage?** Each database finds its usage in an application or system and is purely based on the requirement. For example, the library management systems use a decimal system that numbers the books similar to a tree. In these systems, the RDBMS does not work well as its concept is different. But when we consider an organization, the details of employees or goods cannot fit a tree-like structure. Therefore, tables can be a better solution to store such details. So, here relational database is a better choice.

### Comparison Chart

Differences in	Hierarchical database	Relational database
Storage fashion	It uses a hierarchical storage of data.	It stores data in a tabular fashion.



# Symas OpenLDAP

## How-To Guides

<b>Simplicity in usage and representation</b>	It is complex than the other.	It looks very simpler to represent and to understand.
<b>Which is older?</b>	It is older to the other.	It came only after the hierarchical databases.
<b>The Fundamental difference in the data notion</b>	The category of data is termed as 'Segments'.	The category of data is termed as 'Fields'.
<b>Inheritance</b>	Every child segment/node inherits the properties of its parent.	There is no concept of inheritance.
<b>Data Linking</b>	The Segments are implicitly linked as a child is linked to its parent.	Not linked by default. We should explicitly link the tables with the help of 'Primary keys' and 'Foreign keys'.
<b>Use of key</b>	These are framed with unique keys called the Primary key and also keys from other tables called the Foreign keys. These foreign keys are primary keys in some other table and it is referred while accessing the other table from this table. The keys give a unique identification to the data records and to refer other tables during the data fetching process.	It never uses keys. It has its links to denote the path to be traversed during the data fetch. Therefore, we can consider the keys in relational databases as the equivalent of the paths in hierarchical databases during data fetches. But the paths never represent the uniqueness of data that has been stored in hierarchical databases.
<b>Unique &amp; Duplicate data</b>	Unique data can be easily fetched as it is stored with no duplicates with respect to the primary key.	It needs a little more processing to fetch the unique data.
<b>Data Fetching</b>	Data is fetched from the top most nodes and then traversed along the paths till the required node or segment is reached.	Data is fetched from the tables with the help of the keys.



# Symas OpenLDAP

## How-To Guides

<p><b>Many-to-Many or One-to-Many data linking</b></p>	<p>Such linking is not possible here as a parent can have many children and not the reverse i.e. a child cannot have many parents. Therefore, Many-to-Many or One-to-Many data linking is not at all possible.</p>	<p>These kinds of data relationships are possible here.</p>
<p><b>Fields Vs Nodes</b></p>	<p>The data classification is based on the 'segment or node'</p>	<p>The data classification is based on the 'field'</p>
<p><b>Where it finds its usage?</b></p>	<p>In hierarchical structures such as library management system, to store employee designations starting from CEO to employees, etc.</p>	<p>In structures that can be represented easily as tables such as to store employee details, etc.</p>

