Working with Hierarchies in SQL Server

Overview of hierarchies in SQL Server

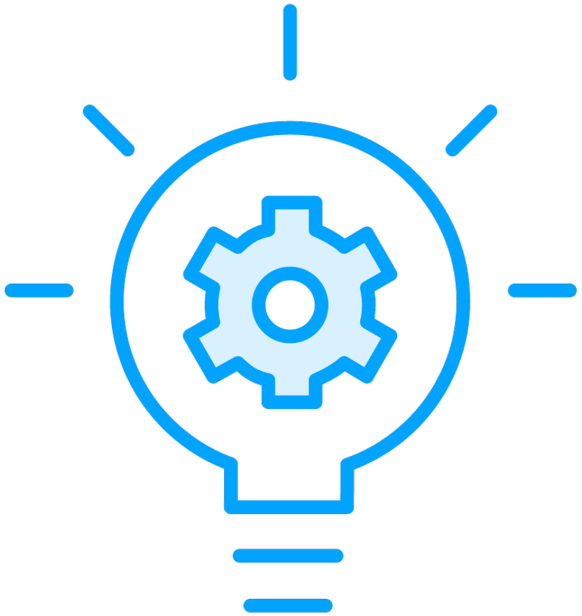
Pinal Dave
SQL Server Expert

@pinaldave | blog.sqlauthority.com

Intended Outcome

Learn to work with hierarchical data in SQL Server, mastering hierarchyid data type, querying, and optimizing performance.

Prerequisites



Skill

- Basics of SQL

Software

- SQL Server 2022
- SQL Server Management Studio

Overview

Overview of hierarchy

hierarchyid data type

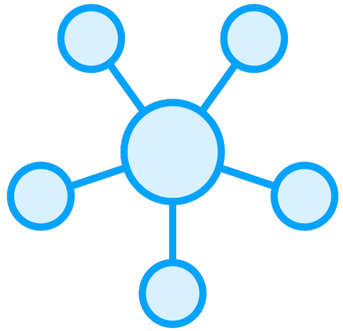
Comparison with other models

Demo

Overview of hierarchies in SQL Server

Hierarchies represent structured relationships between data elements, often organized in a parent-child manner, facilitating organization and analysis

Structure of Hierarchies

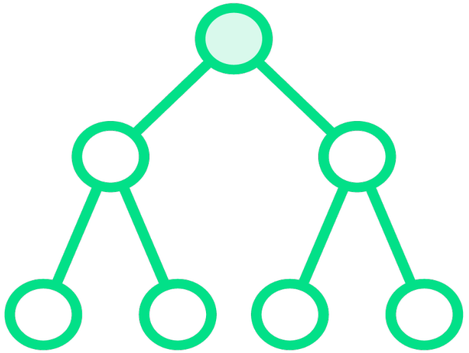


Nodes

Entities – employees, products

Edges

Relationships – reports to, belongs to



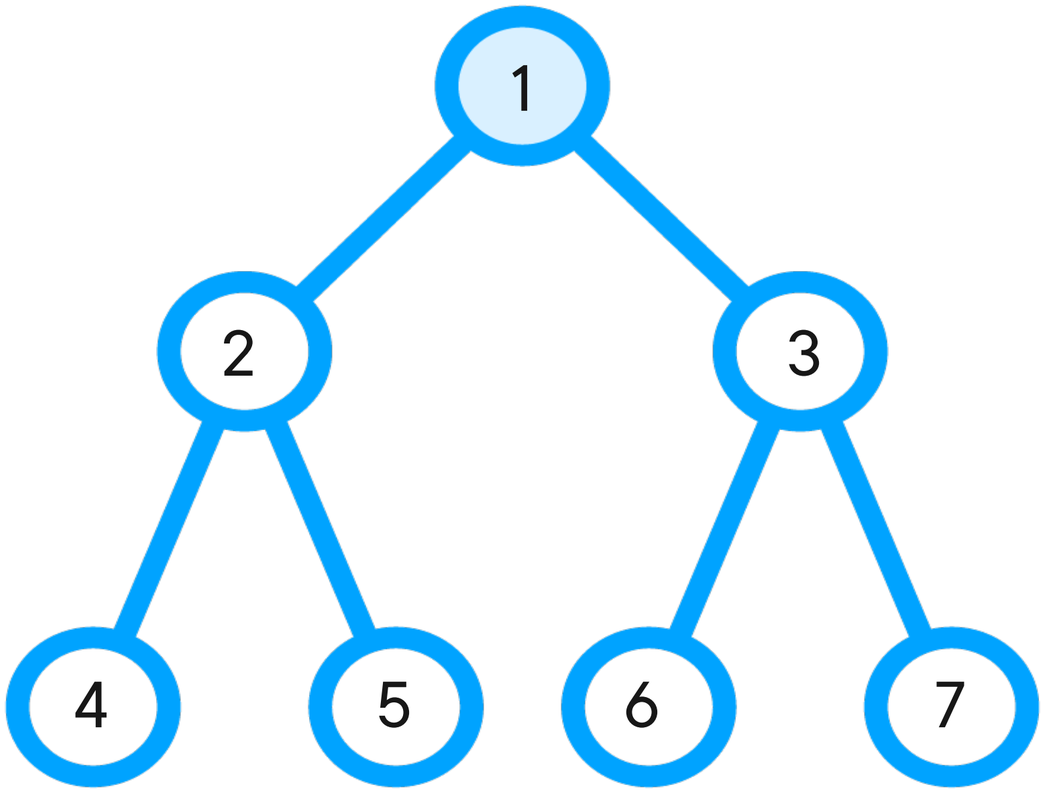
Parent-child

1 parent node
0 or more child node

Structure of Hierarchies

Employee Id	Employee Name	Manager Id	Role
1	John	NULL	CEO
2	Emily	1	CTO
3	Michael	1	CRO
4	Sophia	2	Developer
5	David	2	Developer
6	Jane	3	Sales
7	Rachel	3	Marketing

Table



Hierarchy

Hierarchies in SQL Server with hierarchyid

hierarchyid data type is specialized binary format

Stores nodes in a tree structure for efficient hierarchy

Enables fast querying and manipulation

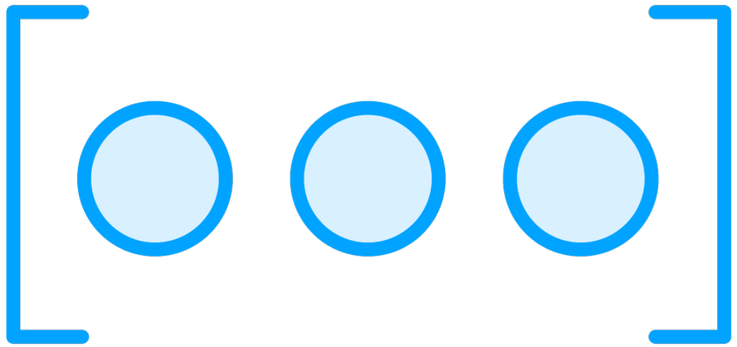
Parent, child, level operations

Demo

Create and add a hierarchy using hierarchyid

Comparing hierarchyid with other hierarchy models

Comparing hierarchyid with Adjacency Lists



Id and parent id define parent-child relationships

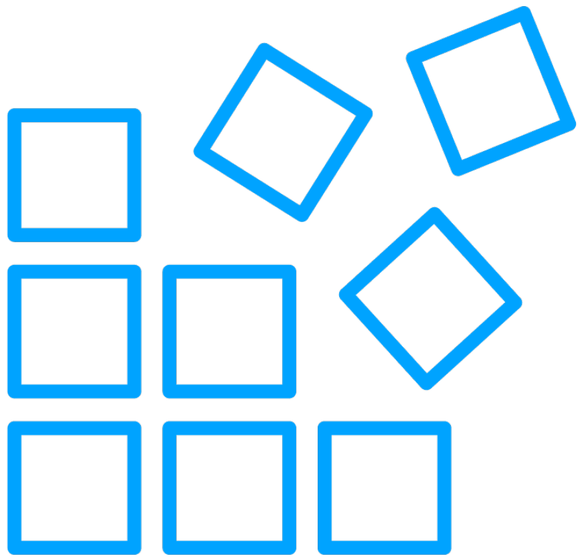
Recursive queries can be slow for finding descendants

hierarchyid data type is more efficient for certain recursive operations

hierarchyid requires more storage space

Trade-off yields substantial performance benefits

Comparing hierarchyid with Nested Sets



Left and right values represent nodes and descendants

Efficient for finding all descendants or ancestors

Updating the hierarchy can be expensive

hierarchyid provides efficient querying and updating

Stores a path from root to each node

Demo

Comparing hierarchyid with adjacency lists

Best Practices

- Design clear data structure with entities, parent-child, relationships
- Prefer hierarchyid for efficiency, fast querying, and manipulation
- Be cautious with recursive CTEs in adjacency lists
- Strike balance between space and performance with hierarchyid
- Document and communicate hierarchy structure for clarity

Up Next:

Working with Nodes in a Hierarchy