<https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-1-converting-a-table-to-a-hierarchical-structure>

# Lesson 1: Converting a Table to a Hierarchical Structure

2017-3-1 1 min to read Contributors

* [BYHAM](https://github.com/BYHAM)

* [Craig Guyer](https://github.com/craigg-msft)

Customers who have tables using self joins to express hierarchical relationships can convert their tables to a hierarchical structure using this lesson as a guide. It is relatively easy to migrate from this representation to one using **hierarchyid**. After migration, users will have a compact and easy to understand hierarchical representation, which can be indexed in several ways for efficient queries.

This lesson, examines an existing table, creates a new table containing a **hierarchyid** column, populates the table with the data from the source table, and then demonstrates three indexing strategies. This lesson contains the following topics:+

* [Examining the Current Structure of the Employee Table](https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-1-1-examining-the-current-structure-of-the-employee-table)
* [Populating a Table with Existing Hierarchical Data](https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-1-2-populating-a-table-with-existing-hierarchical-data)
* [Optimizing the NewOrg Table](https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-1-3-optimizing-the-neworg-table)
* [Summary: Converting a Table to a Hierarchical Structure](https://docs.microsoft.com/en-us/sql/relational-databases/tables/lesson-1-4-summary-converting-a-table-to-a-hierarchical-structure)