1. Хранение иерархических данных может осуществляться различными способами. Один из распространенных способов - это использование специального типа данных, такого как **hierarchyid** в Microsoft SQL Server или **CONNECT BY** в Oracle SQL. В других случаях иерархическая структура может храниться в виде древовидной структуры, где каждая запись содержит ссылку на своего родителя или потомков.
2. Типичные задачи, которые решаются при хранении иерархических данных, включают организацию структуры организации или управления персоналом (дерево сотрудников), категоризацию товаров или услуг (древовидная структура категорий), создание форумов или комментариев (дерево ответов на сообщения), а также множество других задач, связанных с организацией данных в виде иерархии.
3. Иерархический тип данных предназначен для хранения и управления данными, организованными в виде иерархической структуры. Он обеспечивает возможность представления связей между элементами, а также облегчает выполнение запросов для работы с данными в иерархической форме.
4. В Microsoft SQL Server тип данных **hierarchyid** обладает следующими свойствами и методами:
   * **GetLevel()**: Возвращает уровень иерархии текущего узла.
   * **GetAncestor(n)**: Возвращает n-го предка текущего узла.
   * **GetDescendant(min, max)**: Создает новый узел-потомок текущего узла в иерархии.
   * **IsDescendantOf(parent)**: Проверяет, является ли текущий узел потомком указанного узла.
   * **GetRoot()**: Возвращает корневой узел иерархии.
   * и другие.
5. В Oracle SQL секции иерархических запросов используются для определения связей между родительскими и дочерними строками в таблице. Они обычно включают **START WITH**, чтобы определить начальную точку в иерархии, и **CONNECT BY**, чтобы определить отношения родитель-потомок. Например:

SELECT \* FROM employees START WITH manager\_id IS NULL CONNECT BY PRIOR employee\_id = manager\_id;

Этот запрос выбирает всех сотрудников, начиная с тех, у которых **manager\_id** равен **NULL**, и затем следует за ними по иерархии.

1. В Oracle SQL псевдофункции иерархических запросов включают **CONNECT\_BY\_ROOT**, **SYS\_CONNECT\_BY\_PATH**, **LEVEL**, **PRIOR**, **CONNECT\_BY\_ISLEAF** и другие. Например, **CONNECT\_BY\_ROOT** используется для получения значения корневого узла внутри иерархического запроса, а **LEVEL** возвращает текущий уровень иерархии.