**1. Как осуществляется хранение иерархических данных?**

Иерархические данные обычно хранятся в реляционных базах данных с использованием одной из следующих моделей:

* **Ссылочная модель**: В таблице имеется внешний ключ, указывающий на родительский узел. Например, в таблице сотрудников может быть столбец boss\_id, который ссылается на employee\_id того, кто является начальником.
* **Модель с использованием вложенных наборов данных**: Данные хранятся в виде вложенных структур, например, в формате JSON или XML.
* **Adjacency List**: Каждый элемент хранит ссылку на своего родителя, что позволяет легко реализовывать иерархические связи.

**2. Какие типичные задачи решаются при хранении иерархических данных?**

* **Навигация по иерархии**: Получение всех дочерних узлов для заданного родителя или всех предков для заданного узла.
* **Анализ структуры**: Оценка иерархии, например, для определения количества уровней, распределения ресурсов и т. д.
* **Управление данными**: Перемещение узлов, добавление новых узлов и удаление узлов с учётом их подчинённой структуры.

**3. Для чего предназначен иерархический тип данных?**

Иерархический тип данных предназначен для представления структур, где элементы имеют родительско-дочерние связи. Это удобно для:

* Моделирования организационных структур (например, сотрудники и их начальники).
* Хранения данных о категориях и подкатегориях (например, товарные категории).
* Обработки и представления данных в виде дерева (например, файловая система).

**4. Перечислите известные вам свойства и методы иерархического типа данных.**

**Свойства:**

* **Уровень (Level)**: Позволяет определить глубину узла в иерархии.
* **Родительский узел (Parent)**: Ссылка на родительский узел, что позволяет навигацию по структуре.

**Методы:**

* **Добавление узла**: Метод для вставки нового узла в иерархию.
* **Удаление узла**: Метод для удаления узла, который может также удалять подчинённые узлы.
* **Перемещение узлов**: Метод для изменения родительского узла для существующего узла.

**5. Поясните секции иерархических запросов в Oracle.**

Иерархические запросы в Oracle обычно состоят из следующих секций:

* **START WITH**: Определяет начальную точку для иерархии, указывая узел, с которого начинается запрос.
* **CONNECT BY**: Определяет, как связываются родительские и дочерние узлы. Используется для рекурсивного определения иерархии.
* **ORDER BY**: Опциональная секция, позволяющая сортировать результаты запроса.

**6. Перечислите известные вам псевдофункции иерархических запросов в Oracle.**

* **LEVEL**: Возвращает уровень узла в иерархии, начиная с 1 для корневого узла.
* **CONNECT\_BY\_ISLEAF**: Возвращает 1, если узел является листом (не имеет дочерних узлов), и 0 в противном случае.
* **PRIOR**: Используется для обращения к родительскому значению узла в иерархии.
* **SYS\_GUID()**: Генерирует уникальный идентификатор, часто используемый для создания уникальных значений в иерархических структурах.