IT CAREER HUB

Урок 7.2. Табличные выражения CTE

Common Table Expression

 **Важно!**
База данных с доступом на чтение:
hostname: ich-db.edu.itcareerhub.de
username: ich1
password: password

 **Common Table Expression (CTE) — это конструкция в SQL, которая позволяет создать временный набор данных, который можно использовать в основном запросе.**

CTE объявляется с помощью ключевого слова WITH и действует как виртуальная таблица, доступная только в рамках одного SQL-запроса.

С помощью CTE можно полностью отказаться от подзапросов, вынеся их из основного запроса в виде табличных выражений.

Зачем нужны CTE?

CTE упрощают работу с SQL-запросами, делая их более читабельными и поддерживаемыми.

Они особенно полезны в следующих ситуациях:

1. Улучшение читаемости: Вместо того, чтобы вкладывать сложные подзапросы, можно вынести их в CTE и сделать код более понятным.
2. Повторное использование логики: Если одна и та же логика используется в нескольких частях запроса, CTE позволяет избежать дублирования кода.
3. Упрощение сложных операций: CTE помогают разбить сложные запросы на более управляемые части.

IT CAREER HUB

Синтаксис CTE

```
None
WITH CTE_Name AS (
    -- Подзапрос
    SELECT column1, column2, ...
    FROM table_name
    WHERE conditions
)


SELECT column1, column2, ...
FROM CTE_Name
WHERE conditions;
```

IT CAREER HUB

Задание для закрепления

Найти все заказы, сделанные клиентами из Лос-Анджелеса.

Неправильно

```
None
with
LA_clients AS
(SELECT id from customers
WHERE city = 'Los Angelas')
SELECT * FROM orders
WHERE customer_id IN LA_clients;
```

Мы должны явно указать из какого столбца CTE нужно брать значения, даже если там один столбец.

Правильно

```
None
with
LA_clients AS
(SELECT id from customers
WHERE city = 'Los Angelas')
SELECT * FROM orders
WHERE customer_id IN (SELECT id FROM LA_clients);
```

IT CAREER HUB

Плюсы и Минусы CTE

Плюсы:

- **Читаемость**: Делают сложные запросы проще и понятнее.
- **Модульность**: Логику можно разделить на части и использовать повторно.
- **Легкость в использовании**: Не требует явного создания и удаления, как временные таблицы.

Минусы:

- **Ограниченная область действия**: CTE доступны только в рамках одного запроса, они не сохраняются между выполнениями запросов.

IT CAREER HUB

Задание для закрепления

Найти 10 продуктов, которые были заказаны больше всего, и узнать общую сумму заказов по этим продуктам (то есть продукты с наибольшим количеством заказов и их суммарную стоимость).

```
None
with  product_summary AS (
     SELECT product_id, COUNT(*) AS total_orders, SUM(unit_price * quantity) AS
total_revenue
    FROM order_details
    GROUP BY product_id
)
SELECT product_name, total_orders, total_revenue
FROM product_summary
JOIN products p ON product_summary.product_id = p.id
ORDER BY  total_orders DESC
LIMIT 10;
```

IT CAREER HUB

Задание для закрепления

Выбрать все строки из таблицы order_details где unit_price больше среднего.

```
None
WITH avg_price AS (
    SELECT AVG(unit_price) AS ap
    FROM order_details
)
SELECT *
FROM order_details
WHERE unit_price > (SELECT ap FROM avg_price);
```

IT CAREER HUB

Задание для закрепления

Выбрать все строки из таблицы order_details где unit_price больше среднего и меньше среднего, умноженного на 1,5.

```
None
WITH avg_price AS (
    SELECT AVG(unit_price) AS ap
    FROM order_details
)
SELECT *
FROM order_details
WHERE unit_price > (SELECT ap FROM avg_price)
AND unit_price<  (SELECT ap*1.5 FROM avg_price);
```

IT CAREER HUB

Задание для закрепления

Найти все заказы таблица orders оформленных сотрудниками employee_id, в контактах которых таблица employees указан Sales Representative (столбец job_title).

```
None
with sales_repr AS
(SELECT id FROM employees WHERE job_title = 'Sales Representative')
 SELECT *
 FROM orders
WHERE employee_id IN
(SELECT id FROM sales_repr)
```

IT CAREER HUB

Когда использовать CTE

- Когда запрос сложный и требует разбиения на логические части.
- Когда важно сделать запросы более читабельными и поддерживаемыми.
- Когда нужно многократно использовать один и тот же набор данных в запросе.

IT CAREER HUB

Когда использовать подзапросы

- Когда запрос простой и подзапрос логически вписывается в его структуру.
- Когда важна производительность и известно, что SQL-движок лучше оптимизирует подзапросы.

Выбор между CTE и подзапросами зависит от конкретной задачи.

- Если приоритет — читаемость и структурированность кода, особенно в сложных запросах или при работе с иерархиями, предпочтительны CTE.
- Если важна производительность и простота, подзапросы могут быть более подходящим вариантом.