IT CAREER HUB

Урок 7.1. Подзапросы

Сортировка результатов с помощью ORDER BY

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

 **Подзапрос (или вложенный запрос) — это SQL-запрос, который выполняется внутри другого SQL-запроса.**

- Подзапросы позволяют выполнять запросы внутри других запросов, что может быть полезно для получения промежуточных результатов или для выполнения сложных операций, которые не могут быть выполнены одним запросом.

- Подзапрос — это оператор SELECT, размещенный внутри другого оператора.

Пример:

```
Unset
SELECT * FROM t1 WHERE column1 = (SELECT column1 FROM t2);
```

- SELECT * FROM t1 ... — это внешний запрос (или внешний оператор)
- (SELECT column1 FROM t2) — подзапрос.

Мы говорим, что подзапрос вложен во внешний запрос, и на самом деле можно вложить подзапросы внутри других подзапросов на значительную глубину.

- Подзапрос всегда должен быть заключен в круглые скобки.

IT CAREER HUB

Преимущества и недостатки подзапросов

Основные преимущества подзапросов заключаются в следующем:

- Они позволяют создавать запросы, структурированные таким образом, что можно изолировать каждую часть оператора.
- Они предоставляют альтернативные способы выполнения операций, которые в противном случае потребовали бы сложных соединений (JOIN) и объединений (UNION).Разбивает сложные запросы на более простые части.
- Многие считают подзапросы более читаемыми, чем сложные соединения или объединения. Инкапсуляция: Инкапсулирует сложную логику, что делает запросы более читаемыми.

Недостатки:

- Производительность: Некоторые подзапросы могут быть менее эффективными, чем альтернативные подходы.
- Читаемость: Сложные подзапросы могут затруднить понимание запроса.

IT CAREER HUB

Типы подзапросов

Подзапросы можно классифицировать по разным критериям:

1. **По месту использования:**

- Подзапрос в WHERE

 **Подзапрос в WHERE — это подзапрос, который используется для того, чтобы получить результат, который затем используется для фильтрации данных в основном запросе.**

Это как если бы вы сначала спросили что-то одно, а затем использовали этот ответ для того, чтобы задать другой вопрос.

IT CAREER HUB

Задание для закрепления

Найти все заказы, сделанные клиентами из Лос-Анджелеса.

1. Для начала нужно найти клиентов из этого города. Используем таблицу customers.

```
Unset
USE northwind;
SELECT id from customers
WHERE city = 'Los Angelas';
```

2. В выводимых результатах мы получаем один столбец содержащий id таких клиентов. Нужно отфильтровать данные из таблицы orders, оставив только эти customer_id.

```
Unset
SELECT * FROM orders
WHERE customer_id IN
 (SELECT id from customers
WHERE city = 'Los Angelas');
```

Объяснение:

Представьте, что у вас есть список всех заказов и список всех клиентов. Вы хотите узнать, какие заказы были сделаны клиентами из Лос-Анджелеса. Сначала вы находите всех клиентов из Лос-Анджелеса, а затем используете их идентификаторы, чтобы найти заказы, которые они сделали. Подзапрос в WHERE делает эту работу за вас: сначала находит нужных клиентов, а затем помогает вам найти соответствующие заказы.

Таким образом, подзапрос в WHERE позволяет вам использовать результат одного запроса (например, список клиентов из Лос Анджелеса) для фильтрации данных в другом запросе (например, заказы этих клиентов).

IT CAREER HUB

Подзапрос в FROM

Когда подзапрос используется в FROM, он создает временную таблицу или набор данных, который затем используется основным запросом.

Это полезно, когда вам нужно сначала выполнить некоторую агрегацию или вычисления, а затем использовать результаты этих вычислений в дальнейшем запросе.

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 ссуммарную стоимость).

```
Unset
SELECT product_name, total_orders, total_revenue
FROM (
    SELECT product_id, COUNT(*) AS total_orders, SUM(unit_price * quantity) AS
total_revenue
    FROM order_details
    GROUP BY product_id
) AS product_summary
JOIN products p ON product_summary.product_id = p.id
ORDER BY  total_orders DESC
LIMIT 10;
```

1. Из таблицы order_details подзапрос выбирает product_id, количество заказов на каждый продукт (COUNT(*)), и общую выручку от каждого продукта (SUM(unit_price * quantity)).
2. GROUP BY product_id группирует данные по product_id, так что для каждого продукта подсчитывается общее количество заказов и суммарная выручка. Этот подзапрос создает временную таблицу с колонками product_id, total_orders (общее количество заказов) и total_revenue (суммарная выручка).
3. Основной запрос использует результаты подзапроса, давая ему алиас product_summary.
4. JOIN выполняет соединение между этой временной таблицей product_summary и таблицей products, используя product_id из product_summary и id из таблицы products.
5. В результате соединения получаем доступ к имени продукта (product_name) из таблицы products.
6. Основной запрос выбирает product_name, количество заказов на этот продукт (total_orders), и общую выручку от продукта (total_revenue).

По возвращаемому значению

- Однострочный подзапрос (Scalar Subquery)

Возвращает одно значение.

IT CAREER HUB

Задание для закрепления

Выбрать все строки из таблицы order_details где unit_price больше среднего.

1. Найти среднюю цену - это одна конкретная цифра.

```
Unset
SELECT AVG(unit_price) from order_details
```

2. Сравнить столбец unit_price таблицы order_details с найденным значением.

```
Unset
SELECT *
FROM order_details
WHERE unit_price > (SELECT AVG(unit_price) from order_details)
```

IT CAREER HUB

Многострочный подзапрос (Multi-row Subquery)

- Если подзапрос выводит столбец с несколькими значениями.
- При использовании такого подзапроса в WHERE необходим оператор IN.

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).

```
Unset
 SELECT *
 FROM orders
WHERE employee_id IN
(SELECT id FROM employees WHERE job_title = 'Sales Representative')
```