

СОЗДАНИЕ SQL-ПОДЗАПРОСОВ В РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ

Сапияхон Хайдарова

Кандидат технических наук, доцент

Кокандский государственный педагогический институт им. Муками,

Коканд, Узбекистан

Аннотация. В статье изложены методы создания SQL-подзапросов в реляционных базах данных. Приведены примеры на составление подзапросов на языке SQL с помощью инструкции SELECT.

Ключевые слова: SQL-оператор SELECT, подзапросы, внутренний подзапрос, внешний запрос.

Annotation. The article outlines the methods for creating SQL sub queries in relational databases. Examples are given for compiling sub queries in SQL using the SELECT statement.

Keywords: SQL SELECT statement, sub queries, internal sub query, external query.

В настоящее время язык SQL (Structured Query Language) является самым популярным языком баз данных (БД) и предназначен для формирования, манипулирования и извлечения данных из реляционной БД. Одна из причин популярности реляционных БД в том, что они могут оперировать большими объемами данных.

Известно, что запросы SQL осуществляются с помощью инструкции SELECT. В SQL можно создавать простые запросы, а также подзапросы.

Подзапросы – мощный инструмент, который можно использовать во многих SQL-выражениях для работы с данными. Существуют разные определения понятия подзапроса. Подзапросы – это запросы, которые вложены в другие запросы [1]. Подзапрос (sub query) – это запрос, содержащийся в другом SQL-выражении [2].

Иногда возникает необходимость в подзапросах. Чтобы объяснить эту концепцию рассмотрим следующий пример: предположим, что заказы на

товар хранятся в двух таблицах. В таблице Orders указываются номер заказа, идентификатор клиента и дата заказа.

Таблица Orders

order_num	order_date	cust_id
20005	2012-05-01	1000000001
20006	2012-01-12	1000000003
20007	2012-01-30	1000000004
20008	2012-02-03	1000000005
20009	2012-02-08	1000000001

Отдельные элементы заказов хранятся в таблице Order Items.

Таблица Order Items

order_num	order_item	prod_id	quantity	item_price
20005	1	BR01	100	5.49
20005	2	BR03	100	10.99
20006	1	BR01	20	5.99
20006	2	BR02	10	8.99
20006	3	BR03	10	11.99
20007	1	BR03	50	11.49
20007	2	BNBG01	100	2.99
20007	3	BNBG02	100	2.99
20007	4	BNBG03	100	2.99
20007	5	RGAN01	50	4.49
20008	1	RGAN01	5	4.99
20008	2	BR03	5	11.99
20008	3	BNBG01	10	3.49
20008	4	BNBG02	10	3.49
20008	5	BNBG03	10	3.49
20009	1	BNBG01	250	2.49
20009	2	BNBG02	250	2.49
20009	3	BNBG03	250	2.49

Таблица Orders не содержит информацию о клиентах. Она хранит только идентификатор клиента. Информация о клиентах находится в таблице Customers.

Теперь предположим, что вы хотите получить список всех клиентов, которые заказали товар RGAN01. Для этого необходимо выполнить следующее:

- 1) извлечь номера всех заказов, содержащих товар RGAN01;
- 2) получить идентификаторы всех клиентов, которые сделали заказы,

перечисленные на предыдущем шаге;

3) извлечь информацию обо всех клиентах, идентификаторы которых были получены на предыдущем шаге.

Таблица Customers

cust_id	cust_name	cust_address	cust_city	cust_state	cust_zip	cust_country	cust_contact	cust_email
1000000001	Village Toys	200 Maple Lane	Detroit	MI	44444	USA	John Smith	sales@villagetoy.com
1000000002	Kids Place	333 South Lake Drive	Columbus	OH	43333	USA	Michelle Green	
1000000003	Fun4All	1 Sunny Place	Muncie	IN	42222	USA	Jim Jones	jjones@fun4all.com
1000000004	Fun4All	829 Riverside Drive	Phoenix	AZ	88888	USA	Denise L. Stephens	dstephens@fun4all.com
1000000005	The Toy Store	4545 53rd Street	Chicago	IL	54545	USA	Kim Howard	

Каждый из этих пунктов можно выполнить в виде отдельного запроса. Но можно также воспользоваться подзапросами для того, чтобы объединить все три запроса в одну-единственную инструкцию.

Первая инструкция SELECT извлекает столбец order_num для всех элементов заказов, у которых в столбце prod_id значится RGAN01:

```
SELECT order_num
FROM Order Items
WHERE prod_id='RGAN01';
```

Результат представляет собой номера двух заказов, содержащих данный товар:

```
order_num
-----
20007
20008
```

Следующий шаг состоит в получении идентификаторов клиентов, связанных с заказами 20007 и 20008. Используя предложение IN, можно создать показанную ниже инструкцию SELECT.

```
SELECT cust_id
```

```
FROM Orders
WHERE order_num IN (20007,20008);
```

Результат выглядит следующим образом:

```
cust_id
-----
1000000004
1000000005
```

Теперь объединим эти два запроса путем превращения первого из них (того, который возвращает номера заказов) в подзапрос.

```
SELECT cust_id
FROM Orders
WHERE order_num IN (SELECT order_num
                    FROM Order Items
                    WHERE prod_id='RGAN01');
```

Результат снова будет таким же, как указано выше:

```
cust_id
-----
1000000004
1000000005
```

Подзапросы всегда обрабатываются, начиная с самой внутренней инструкции SELECT в направлении “изнутри наружу”. Вначале она выполняет следующий подзапрос:

```
SELECT order_num FROM OrderItems WHERE prod_id='RGAN01'
```

В результате возвращаются два номера заказа: 20007 и 20008. Эти два значения затем передаются в предложение WHERE внешнего запроса в формате с разделителем в виде запятой, необходимом для оператора IN. Теперь внешний запрос становится таким:

```
SELECT cust_id FROM orders WHERE order_num IN (20007,20008)
```

Теперь у нас есть идентификаторы всех клиентов, заказавших товар RGAN01.

Следующий шаг состоит в получении клиентской информации для каждого из этих идентификаторов. Инструкция SQL, осуществляющая выборку двух столбцов, выглядит так:

```
SELECT cust_name, cust_contact
FROM Customers
WHERE cust_id IN ('1000000004', '1000000005');
```

Но вместо указания идентификаторов клиентов можно превратить данное предложение WHERE в подзапрос:

```
SELECT cust_name, cust_contact
FROM Customers
WHERE cust_id IN (SELECT cust_id
                  FROM Orders
                  WHERE order_num IN (SELECT order_num
                                      FROM OrderItems
                                      WHERE prod_id=
                                      'RGAN01'));
```

Результат выглядит так:

cust_name	cust_contact
Fun4All	Denise L. Stephans
The Toy Store	Kim Howard

Чтобы выполнить такой запрос, СУБД должна по сути обработать три инструкции SELECT. Благодаря подзапросам можно создавать очень мощные и гибкие инструкции SQL.

Литература

1. Бен Форта. Освой самостоятельно SQL за 10 минут, 4-е изд.: Пер. с англ.— М.: ООО “И.Д. Вильямс”, 2014. 288 с.
2. Алан Бьюли. Изучаем SQL. Вводный курс для разработчиков и администраторов БД. Санкт-Петербург–Москва, Символ ®, 2007

3. Хайдарова, Сапияхон. "Создание SQL-запросов в реляционных базах данных". *Вестник РГГУ. Серия: Информатика. Информационная безопасность. Математика* 3 (2020): 8-19.