Open Access Article

CREATING SQL-SUB QUERIES IN RELATIONAL DATABASES

Sh. Pulatov
Doctor of Technical Sciences, Associate Professor of Kokand State Pedagogical Institute,
Uzbekistan, E-mail: psharifjon@mail.ru

S. Khaidarova
Candidate of Technical Sciences, Associate Professor of Kokand State Pedagogical Institute,
Uzbekistan, E-mail: hay-vb1952@umail.uz

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.

Currently, the SQL (Structured Query Language) language is the most popular database language and is intended for formation, manipulating and retrieving data from a relational database. One of the reasons for the popularity of relational databases is that they can handle large volumes of data.

Working with databases requires a good working knowledge of the SQL relational language. In everyday life, we have to work with databases, the SQL language is designed for this.

Every time you select a name in an email address book, you are accessing a database. When you search for something using a search site on the Internet, you send queries to a database. When you log in to your office computer, you enter your username and password, which are then compared with the values stored in the database. And even when you insert your plastic card into an ATM, checking the PIN code and account balance goes through the database [1].

It is known that SQL queries are made using the SELECT statement. In SQL you can create simple queries as well as sub queries.

Sub queries are a powerful tool that can be used in many SQL statements to manipulate data. There are different definitions of the concept of a sub query. Sub queries are queries that are nested within other queries [1]. A sub query is a query contained within another SQL expression [2].

Sometimes there is a need for sub queries. To explain this concept, consider the following example. Let's say product orders are stored in two tables. The Orders table contains the order number, customer ID, and order date.

Table Orders

order_num	order_date	cust_id
20005	2012-05-01	1000000001
20006	2012-01-12	1000000003
20007	2012-01-30	1000000004

Received: October 04, 2023 / Revised: October 30, 2023 / Accepted: November 18, 2023 / Published: December 18, 2023
About the authors：Sh. Pulatov
Corresponding author- Email:

20008	2012-02-03	1000000005
20009	2012-02-08	1000000001

Individual order elements are stored in the table Order Items.

Table 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

The Orders table does not contain customer information. It only stores the client ID. Customer information is located in the Customers table.

Now let's say you want to get a list of all customers who ordered product RGAN01. To do this you need to do the following:

1) extract the numbers of all orders containing product RGAN01;
2) get the identifiers of all customers who made orders, listed in the previous step;
3) extract information about all clients whose identifiers were obtained in the previous step.

Table Customers

cust_id	cust_ name	cust_ address	cust_ city	cust _ state	cust_ zip	cust _ coun try	cust_con tact	cust_email

1000000001	Village Toys	200 Maple Lane	Detroit	MI	44444	USA	John Smith	sales@villagetoys.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	

Each of these points can be performed with a separate request. But you can also use sub queries to read all three queries in one procedure.
The first SELECT statement retrieves the order_num column for all order elements that have the value RGAN01 in the prod_id column:

```
SELECT order_num

FROM Order Items

WHERE prod_id='RGAN01';
```

The result is the numbers of two orders containing this product:

```
order_num

----------

 20007

 20008
```

The next step is to obtain the customer IDs associated with orders 20007 and 20008. Using the IN clause, you can create the SELECT statement shown below.

```
SELECT cust_id

FROM Orders

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

The result looks like this:

cust_id

1000000004

1000000005

Now let's combine these two queries, turning the first one (the one that returns order numbers) into a sub query.

```
SELECT cust_id

FROM Orders

WHERE order_num IN   (SELECT order_num

                      FROM Order Items

                      WHERE prod_id='RGAN01');
```

The result will again be the same as above:

cust_id

1000000004

1000000005

Sub queries are always processed starting with the innermost SELECT statement and working from the inside out. First it executes the following sub query:

SELECT order_num FROM Order ltems WHERE prod_id=1RGAN01'

As a result, two order numbers are returned: 20007 and 20008. These two values are then passed to the WHERE clause of the outer query in the comma-separated format required by the IN operator.

Now the outer request becomes like this:

SELECT oust id FROM orders WHERE order num IN (20007,20008)

Now we have the IDs of all customers who ordered product RGAN01.

The next step is to obtain client information for each of these IDs. The SQL statement that fetches two columns looks like this:

```
SELECT cust_name, cust_contact

FROM Customers

WHERE cust_id IN ('1000000004', '1000000005');
```

But instead of specifying client IDs, you can turn this WHERE clause into a sub query:

```
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'));
```

The result looks like this:

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

To execute such a query, the database management system must essentially process three SELECT statements. Subqueries enable you to create very powerful and flexible SQL statements.

REFERENCES

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.