

Использование подзапросов

Запросы по типу возвращаемого результата

- **Скалярные** (Scalar Subqueries) – возвращают 1 значение

```
SELECT MAX(orderid) FROM "Sales"."Orders";
```

	123
1	11 077

- **Многозначные** (Multi-Valued Subqueries) – возвращают список значений

```
SELECT custid  
FROM "Sales"."Customers"  
WHERE country = 'Mexico';
```

	custid
1	2
2	3
3	13
4	58
5	80

- **Табличные** (Table-Valued Subqueries) – возвращают таблицу

```
SELECT orderid, productid, (unitprice * qty) AS LineTotal  
FROM "Sales"."OrderDetails" od;
```

	orderid	productid	linetotal
1	10 248	11	\$168.00
2	10 248	42	\$98.00
3	10 248	72	\$174.00
4	10 249	14	\$167.40
5	10 249	51	\$1,696.00
6	10 250	41	\$77.00

Использование подзапросов

Предложение	Scalar Subqueries	Multi-Valued Subqueries	Table-Valued Subqueries
SELECT	Вычисляемый столбец	-	-
FROM	-	Derived Table (именованная таблица из одного столбца)	Derived Table (именованная таблица) CTE (временное представление)
WHERE	Фильтрация записей на основе сравнения со скалярной величиной	Фильтрация записей с использованием предикатов [NOT] IN, SOME, ALL, ANY, [NOT] EXISTS*	Фильтрация записей с использованием предиката [NOT] EXISTS*
GROUP BY	-	-	-
HAVING	Фильтрация групп на основе сравнения со скалярной величиной	-	-
ORDER BY	-	-	-



Вложенные подзапросы и коррелированные подзапросы

Вложенный подзапрос

Внешний запрос: 2

```
SELECT orderid, productid, unitprice
FROM "Sales"."OrderDetails"
WHERE orderid = ( );
```

Внутренний запрос: 1

```
SELECT MAX(orderid) AS lastorder
FROM "Sales"."Orders";
```

Коррелированный подзапрос

Внешний запрос:

```
SELECT orderid, empid, orderdate
FROM "Sales"."Orders" AS 01
WHERE orderdate = ( );
```

Внутренний запрос: 2

```
SELECT MAX(orderdate)
FROM "Sales"."Orders" AS 02
WHERE 02.empid = 01.empid;
```

Использование скалярного вложенного подзапроса

```
SELECT orderid, productid, unitprice, qty
FROM "Sales"."OrderDetails"
WHERE orderid =
    (SELECT MAX(orderid) FROM "Sales"."Orders");
```

```
SELECT orderid,
       orderdate,
       (SELECT min(orderdate)
        FROM "Sales"."Orders") AS FirstDate
FROM "Sales"."Orders";
```

- Если вложенный подзапрос возвращает пустую выборку (empty set), результат конвертируется в NULL



Использование многозначного вложенного запроса с одним столбцом

- Список заказов, сделанных клиентами из 'Mexico'

```
SELECT custid, orderid
FROM "Sales"."Orders"
WHERE custid IN ( SELECT custid
                  FROM "Sales"."Customers"
                  WHERE country = 'Mexico');
```

- Список деталей заказов, включающих продукты, цена которых превышает минимальные цены продуктов в каждой категории

```
SELECT *
FROM "Sales"."OrderDetails" od
WHERE unitprice > ALL( SELECT min(unitprice)
                      FROM "Production"."Products" p
                      GROUP BY categoryid );
```



Использование многозначного вложенного запроса с несколькими столбцами

- Вы можете использовать IN (ANY, ALL, SOME) для сравнения по нескольким столбцам:
 - В предложении WHERE необходимо сгруппировать имена столбцов в скобках
 - Количество сгруппированных столбцов должно соответствовать целевому списку подзапроса, и иметь те же типы данных

Список заказов 76 клиента для которых город доставки **не совпадает** с городом клиента:

```
SELECT custid, orderid, shipcity
FROM "Sales"."Orders"
WHERE (custid, shipcity) != ALL (SELECT custid,city
                                FROM "Sales"."Customers")
AND custid= 76;
```



Использование скалярного коррелированного запроса

- Список заказов с общей суммой каждого заказа

```
SELECT orderid , o.custid,  
       (SELECT SUM(unitprice*qty)  
        FROM "Sales"."OrderDetails" od  
        WHERE od.orderid = o.orderid) AS OrderTotal  
FROM "Sales"."Orders" o;
```

- Список последних заказов, оформленных каждым сотрудником

```
SELECT orderid, empid, orderdate  
FROM "Sales"."Orders" AS 01  
WHERE orderdate =  
       (SELECT MAX(orderdate)  
        FROM "Sales"."Orders" AS 02  
        WHERE 02.empid = 01.empid)  
ORDER BY empid, orderdate;
```




Использование предиката EXISTS

WHERE [NOT] EXISTS (subquery)

- Если вложенный запрос возвращает хотя бы одну запись - EXISTS возвращает TRUE
- Если вложенный запрос не возвращает ни одной записи - EXISTS возвращает FALSE

```
SELECT custid, companyname
FROM "Sales"."Customers" AS c
WHERE EXISTS (
    SELECT 1
    FROM "Sales"."Orders" AS o
    WHERE c.custid=o.custid);
```

```
SELECT custid, companyname
FROM "Sales"."Customers" AS c
WHERE NOT EXISTS (
    SELECT 1
    FROM "Sales"."Orders" AS o
    WHERE c.custid=o.custid);
```