

## Лабораторная работа 5

### Подзапросы

Операторы **DML** могут содержать в себе вложенные операторы **SELECT**. Такие запросы называются подзапросами. В зависимости от того, как подзапрос взаимодействует с основным запросом, различают **простые** и **коррелированные** подзапросы.

### Простые подзапросы

Простые подзапросы выполняются один раз, и результат их выполнения используется основным запросом. Для анализа результатов, которые вернул простой подзапрос можно использовать операции сравнения  $=$ ,  $>$ ,  $<$ , а также операторы **IN**, **ANY**, **ALL**.

**Запрос 1.** Найти сотрудников, зарплата которых больше средней зарплаты по всей фирме.

```
SELECT employee_id, department_id, first_name, last_name, salary
FROM Employees
WHERE salary > (SELECT AVG( salary) FROM Employees);
```

В этом примере подзапрос возвращает среднее значение зарплаты, а основной запрос выводит данные о сотрудниках, зарплата которых больше среднего значения.

**Запрос 2.** Найти сотрудника, который был принят на работу раньше всех

```
SELECT employee_id, department_id, job_id, first_name, last_name,
hire_date
FROM Employees
WHERE hire_date = (SELECT MIN(hire_date) FROM Employees )
```

employee_id	department_id	job_id	first_name	last_name	hire_date
100	90	AD_PRES	Steven	King	17-06-1987

В этом примере подзапрос возвращает минимальное значение даты приема на работу (hire\_date), а основной запрос выводит данные о сотрудниках, которые были приняты на работу в эту дату.

### Подзапросы и значение Null

Если подзапрос возвращает значение NULL или не возвращает ни одной строки, то любая операция сравнения, в основном запросе, будет иметь значение неопределенно и результат выполнения основного запроса будет пуст.

**Запрос 3.** Вывести данные о сотрудниках, работающих в отделе 50 и имеющих такой же рейтинг как у сотрудника last\_name = 'Bissot'.

```
SELECT employee_id, department_id, job_id, first_name, last_name,
rating_e
FROM Employees
WHERE department_id = 50
AND rating_e = (SELECT rating_e FROM Employees
WHERE last_name = 'Bissot')
```

employee_id	department_id	job_id	first_name	last_name	rating_e
128	50	ST_CLERK	Steven	Markle	5
129	50	ST_CLERK	Laura	Bissot	5
140	50	ST_CLERK	Joshua	Patel	5
141	50	ST_CLERK	Trenna	Rajs	5
185	50	SH_CLERK	Alexis	Bull	5

Если вместо условия last\_name = 'Bissot' поставить last\_name = 'Nayer' (rating\_e = NULL). или last\_name = 'Noyer' (такой фамилии нет), то результат запроса будет пуст, даже в том случае если в отделе 50 будут сотрудники, у которых столбец rating\_e имеет значение NULL.

## Использование выражения IN

**Запрос 4.** Вывести значения столбцов department\_id, department\_name отделов, расположенных не в Соединенных Штатах Америки (country\_id='US')

```
SELECT department_id, department_name
FROM Departments WHERE location_id NOT IN
(SELECT location_id FROM Locations
WHERE country_id = 'US');
```

```
department_id|department_name |
-----+-----+
          20|Marketing        |
          40|Human Resources |
          70|Public Relations|
          80|Sales           |
```

## Использование выражений ALL и ANY

Выражения ALL и ANY используются в комбинации с операторами сравнения. Выражение ALL возвращает значение TRUE, если оператор сравнения выполняется для всех элементов возвращаемых подзапросом, а выражение ANY возвращает значение TRUE, если оператор сравнения выполняется хотя бы для одного элемента.

**Запрос 5.** Найти сотрудников, чья зарплата выше всех зарплат из отдела 50

```
SELECT employee_id, department_id, job_id, first_name,
last_name, salary
FROM Employees WHERE salary > ALL
(SELECT salary FROM Employees
WHERE department_id=50);
```

## Выражения ALL, ANY и значение NULL

**Запрос 6.** Вывести значения столбцов employee\_id, rating\_e сотрудников, которые работают в отделе 60, и рейтинг которых больше рейтинга любого сотрудника из отдела 100.

```
SELECT employee_id, rating_e
FROM Employees
WHERE department_id = 60
AND rating_e > ALL
(SELECT rating_e
FROM Employees
WHERE department_id = 100);
```

**Запрос 7.** Вывести значения столбцов employee\_id, rating\_e сотрудников, которые работают в отделе 60, и рейтинг которых больше рейтинга хотя бы одного сотрудника из отдела 100.

```
SELECT employee_id, rating_e
FROM Employees
WHERE department_id = 60
AND rating_e > ANY
(SELECT rating_e
FROM Employees
WHERE department_id = 100);
```

Присвоим столбцу rating\_e значение NULL у сотрудника 110, который работает в отделе 100.

Если теперь выполнить запрос 6, то он не вернет ни одной строки, а результат выполнения запроса 7, не изменится, и это остается справедливым для любой операции сравнения.

Причиной этого является то, что если в списке значений, анализируемых оператором ALL, будет присутствовать значение NULL, то этот оператор,

совместно с любой операцией сравнения, вернет значение FALSE, а на результат оператора ANY значения NULL не влияют.

### **Выражения ALL, ANY и пустые подзапросы**

Рассмотрим работу выражений ALL и ANY, в том случае если подзапрос, который они обрабатывают, будет пуст. Для этого в запросе 6 заменим номер отдела 100 на 120.

**Запрос 8.** Вывести значения столбцов employee\_id, rating\_e сотрудников, которые работают в отделе 60, и рейтинг которых больше рейтинга любого сотрудника из отдела 120.

```
SELECT employee_id, rating_e
FROM Employees
WHERE department_id = 60
AND rating_e > ALL
(SELECT rating_e
FROM Employees
WHERE department_id = 120);
```

В таблице Employees нет сотрудников работающих в отделе 120, следовательно, подзапрос не вернет ни одной строки, поэтому результат выполнения запроса 8 будет пуст, и это справедливо для любой операции сравнения.

Если в запросе 8 вместо выражения ANY использовать выражения ALL, то запрос вернет данные обо всех сотрудниках, которые работают в отделе 60. Выражение ALL для пустого подзапроса всегда возвращает TRUE, так указано в стандарте.

## Многостолбцовые подзапросы

Многостолбцовые подзапросы возвращают значения нескольких столбцов. Они могут быть однострочными и многострочными. Этот вид подзапросов можно использовать в предложении WHERE. Для сравнения значений нескольких столбцов со списком значений возвращаемых многостолбцовым подзапросом, используется оператор IN, который в этом случае должен иметь следующий синтаксис:

```
WHERE ({список столбцов}) IN  
(SELECT {список значений} FROM ... .)
```

Список столбцов и список значений должны содержать одинаковое количество элементов, а также тип столбца должен совпадать с типом соответствующего ему значения.

**Запрос 9.** Вывести данные о сотрудниках, у которых должность и зарплата совпадает с должностью и зарплатой сотрудника 106.

```
SELECT employee_id, job_id, salary  
FROM Employees  
WHERE (job_id, salary) IN  
(SELECT DISTINCT job_id, salary  
FROM Employees  
WHERE employee_id=106)  
AND employee_id <>106;
```

Подзапросы можно использовать в качестве источника данных. В одних случаях это позволяет существенно сократить время выполнения запроса, в других обойти ограничения в использовании элементов языка SQL.

**Запрос 9.** Требуется вывести данные об отделах, в которых работает более 10 сотрудников.

```

SELECT d.department_id, d.department_name, d.manager_id,
e.count_emp
FROM Departments d JOIN
    (SELECT department_id, COUNT(*) AS count_emp
     FROM Employees
     GROUP BY department_id
     HAVING COUNT(*) > 10) e
ON (d.department_id=e.department_id);

```

### Коррелированные подзапросы

Коррелированный подзапрос использует один или несколько столбцов основного запроса и выполняется для каждой строки основного запроса.

Запросы, содержащие коррелированный подзапрос, выполняются следующим образом. Выбирается первая строка таблицы, сформированная основным запросом. Значения определенных столбцов этой таблицы передаются в подзапрос. Если эти значения удовлетворяют условиям подзапроса, то эта строка помещается в результат выполнения основного запроса. После этого выбирается вторая строка и т.д..

**Запрос 10.** Найти сотрудников, у которых зарплата выше средней зарплаты отдела, в котором они работают.

```

SELECT first_name, last_name, department_id, salary
FROM Employees o
WHERE o.salary > (SELECT AVG(i.salary)
                  FROM Employees i
                  WHERE i.department_id = o.department_id);

```

Этот запрос выполняется следующим образом:

1. Выбирается первая строка таблицы Employees o;

2. Значение столбца `o.department_id` передается в коррелированный подзапрос, который возвращает среднее значение зарплаты отдела, в котором работает рассматриваемый сотрудник;
3. Если зарплата рассматриваемого сотрудника больше средней зарплаты отдела, в котором он работает, то данные о рассматриваемом сотруднике помещаются в результат выполнения основного запроса.

### Оператор EXISTS

При работе с коррелированными подзапросами часто используется оператор **EXISTS**, который возвращает значение **TRUE** в том случае, если результат выполнения подзапроса не пуст.

**Запрос 11.** Вывести данные о сотрудниках, которые работают в отделе 80 и руководят другими служащими.

```
SELECT department_id, first_name, last_name, salary
FROM Employees o
WHERE department_id =80 AND EXISTS
(SELECT employee_id
FROM Employees i
WHERE i.manager_id = o.employee_id) ;
```

В этом запросе, в подзапрос передается значение столбца **employee\_id**. Если это значение встречается в столбце **manager\_id** то это означает что рассматриваемый сотрудник руководит другими служащими. Результат выполнения подзапроса будет не пуст, **EXISTS** вернет значение **TRUE** и основной запрос выведет данные об этом сотруднике.



## Задание

**Задача 1.** Найти сотрудников, зарплата которых с учетом комиссионных больше зарплаты их начальников.

**Задача 2.** Найти отделы, которые являются единственными в той стране, где они находятся.

**Задача 3.** Вывести значения столбцов `departments_id`, `employee_id`, `salary` сотрудников, у которых оба значения `salary` и `commission_pct` совпадают со значениями `salary` и `commission_pct` хотя бы одного сотрудника из отдела 30.

**Задача 4.** Для каждого сотрудника вывести число месяцев прошедших между датой приема на работу этого сотрудника и датой приема на работу первого сотрудника в отдел, в котором работает сотрудник.

**Задача 5.** Определить год и месяц, когда у сотрудника 152 была максимальная сумма продаж.

**Задача 6.** Для каждого дня продаж, осуществленных в мае 2017, вывести данные о заказе, который имеет максимальную сумму из всех заказов, которые были оформлены в этот день.

**Задача 7.** В таблице `Orders` найти продавцов (`salesman_id`), у которых список клиентов, совпадает со списком клиентов продавца 179. Клиентом является покупатель (`customer_id`), которым продавец оформлял заказы.