

Использование подзапросов для решения запросов

Цели

Изучив материал этого занятия, вы сможете:

- Определять подзапросы
- Описывать типы проблем, для решения которых можно использовать подзапросы
- Перечислять типы подзапросов
- Создавать однострочные и многострочные подзапросы

План занятия

- Подзапрос: типы, синтаксис и указания
- Однострочные подзапросы:
 - групповые функции в подзапросе
 - предложение `HAVING` с подзапросами
- Многострочные подзапросы
 - использование оператора `ALL` или `ANY`
- Значения `NULL` в подзапросе

Использование подзапроса для решения проблемы

У кого оклад больше, чем у Abel?

**Основной
запрос:**



У кого из сотрудников оклады больше, чем оклад у Abel?

**Подзапро
с:**



Какой оклад у Abel?



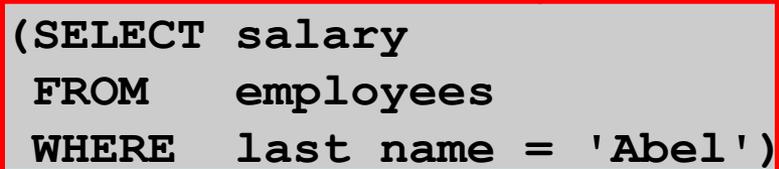
Синтаксис подзапроса

```
SELECT select_list
FROM   table
WHERE  expr operator
        (SELECT select_list
             FROM table);
```

- Подзапрос (внутренний запрос) выполняется *перед* основным (внешним) запросом.
- Результат подзапроса используется основным запросом.

Использование подзапросов

```
SELECT last_name, salary
FROM employees
WHERE salary >
      (SELECT salary
       FROM employees
       WHERE last name = 'Abel');
```



	LAST_NAME	SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000
4	Hartstein	13000
5	Higgins	12000

Указания по использованию подзапросов

- Заклучайте подзапросы в скобки.
- Для повышения наглядности кода размещайте подзапросы справа от условия сравнения (хотя в принципе запрос допустимо помещать с любой стороны от условия сравнения).
- Используйте с однострочными подзапросами однострочные операторы, а с многострочными подзапросами – многострочные операторы.

Типы подзапросов

- Однострочный подзапрос



- Многострочный подзапрос



План занятия

- Подзапрос: типы, синтаксис и указания
- Однострочные подзапросы:
 - групповые функции в подзапросе
 - предложение `HAVING` с подзапросами
- Многострочные подзапросы
 - использование оператора `ALL` или `ANY`
- Значения `NULL` в подзапросе

Однострочные подзапросы

- Возвращают только одну строку
- Используют однострочные операторы сравнения

Оператор	Значение
=	Равно
>	Больше
>=	Больше или равно
<	Меньше
<=	Меньше или равно
<>	Не равно

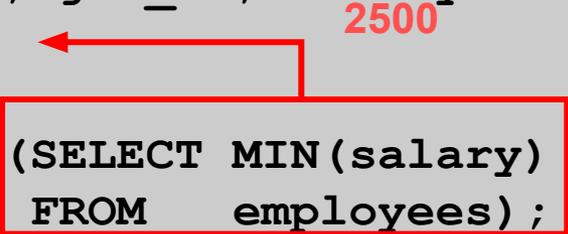
Выполнение однострочных подзапросов

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id =  SA_REP
AND salary > 
  (SELECT job_id
   FROM employees
   WHERE last_name = 'Taylor')
  (SELECT salary
   FROM employees
   WHERE last_name = 'Taylor');
```

	LAST_NAME	JOB_ID	SALARY
1	Abel	SA_REP	11000

Использование в подзапросах групповых функций

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary =
    (SELECT MIN(salary)
     FROM employees);
```



	LAST_NAME	JOB_ID	SALARY
1	Vargas	ST_CLERK	2500

Предложение HAVING в подзапросах

- Сервер Oracle выполняет подзапросы первыми.
- Сервер Oracle возвращает результаты в предложение HAVING основного запроса.

```
SELECT  department_id, MIN(salary)
FROM    employees
GROUP BY department_id
HAVING  MIN(salary) > (SELECT MIN(salary)
                       FROM    employees
                       WHERE   department_id = 50);
```

2500

	DEPARTMENT_ID	MIN(SALARY)
1	(null)	7000
2	90	17000
3	20	6000
...		
7	10	4400

Что неправильно в этой инструкции?

```
SELECT employee_id, last_name
FROM employees
WHERE salary =
      (SELECT MIN(salary)
       FROM employees
       GROUP BY department_id);
```

ORA-01427: single-row subquery returns more than one ...



An error was encountered performing the requested operation:

ORA-01427: single-row subquery returns more than one row
01427. 00000 - "single-row subquery returns more than one row"

*Cause:

*Action:

Error at Line:1

**Однострочный
оператор
с многострочным
подзапросом**

Внутренний запрос не вернул ни одной строки

```
SELECT last_name, job_id
FROM employees
WHERE job_id =
      (SELECT job_id
       FROM employees
       WHERE last_name = 'Haas');
```

0 rows selected

Подзапрос не вернул ни одной строки, поскольку нет сотрудника с фамилией «Haas».

План занятия

- Подзапрос: типы, синтаксис и указания
- Однострочные подзапросы:
 - групповые функции в подзапросе
 - предложение `HAVING` с подзапросами
- **Многострочные подзапросы**
 - использование оператора `ALL` или `ANY`
- Значения `NULL` в подзапросе

Многострочные подзапросы

- Возвращают больше одной строки
- Используют многострочные операторы сравнения

Оператор	Значение
IN	Равен любому элементу из списка
ANY	Перед ним должны использоваться операторы =, !=, >, <, <=, >=. Сравнивает текущее значение с каждым значением, перечисленным в списке или возвращенным запросом. Получает значение FALSE, если запрос не вернул ни одной строки.
ALL	Перед ним должны использоваться операторы =, !=, >, <, <=, >=. Сравнивает текущее значение со всеми значениями, перечисленными в списке или возвращенными запросом. Получает значение TRUE, если запрос не вернул ни одной строки.

Использование оператора ANY в многострочных подзапросах

```
SELECT employee_id, last_name, job_id, salary
FROM employees 9000, 6000, 4200
WHERE salary < ANY
  (SELECT salary
   FROM employees
   WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	144	Vargas	ST_CLERK	2500
2	143	Matos	ST_CLERK	2600
3	142	Davies	ST_CLERK	3100
4	141	Rajs	ST_CLERK	3500
5	200	Whalen	AD_ASST	4400

...

9	206	Gietz	AC_ACCOUNT	8300
10	176	Taylor	SA_REP	8600

Использование оператора ALL в многострочных подзапросах

```
SELECT employee_id, last_name, job_id, salary
FROM employees 9000, 6000, 4200
WHERE salary < ALL
  (SELECT salary
   FROM employees
   WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	141	Rajs	ST_CLERK	3500
2	142	Davies	ST_CLERK	3100
3	143	Matos	ST_CLERK	2600
4	144	Vargas	ST_CLERK	2500

План занятия

- Подзапрос: типы, синтаксис и указания
- Однострочные подзапросы:
 - групповые функции в подзапросе
 - предложение `HAVING` с подзапросами
- Многострочные подзапросы
 - использование оператора `ALL` или `ANY`
- Значения `NULL` в подзапросе

Значения NULL в подзапросе

```
SELECT emp.last_name
FROM   employees emp
WHERE  emp.employee_id NOT IN
      (SELECT mgr.manager_id
       FROM   employees mgr);
```

```
0 rows selected
```


Заключение

На этом занятии были изучены следующие темы:

- Ситуации, когда для решения проблемы целесообразно использовать подзапросы
- Создание подзапросов для запросов, учитывающих неизвестные значения

```
SELECT select_list
FROM   table
WHERE  expr operator
       (SELECT select_list
        FROM   table);
```

Упражнение 7: обзор

Это упражнение охватывает следующие темы:

- Создание подзапросов для запроса данных с учетом неизвестных условий
- Использование подзапросов для поиска значений, существующих в одном наборе данных и отсутствующих в другом

