

# SQL: выборки из нескольких таблиц, подзапросы

Sumy Educational Center  
Software Quality Assurance (QA)

# Использование вложенного запроса для решения задачи

“У кого зарплата больше, чем у Джона?”

Основной запрос



“Кто из работников имеет зарплату больше, чем Джон?”

Вложенный запрос



“А какая зарплата у Джона?”



# Подзапросы

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


- Подзапросы (вложенные запросы) выполняются единожды перед выполнением основного запроса.
- Результат выполнения подзапроса используется в основном запросе (внешнем запросе).

# Подзапросы

- Подзапросы можно использовать:
  - SELECT:
    - Предложение SELECT
    - Предложение FROM
    - Предложение WHERE
    - Предложение HAVING
  - CREATE VIEW
  - CREATE TABLE
  - ....

# Использование подзапросов

```
SQL> SELECT ename  
2 FROM emp  
3 WHERE sal > 2975  
4 (SELECT sal  
5 FROM emp  
6 WHERE empno=7566) ;
```



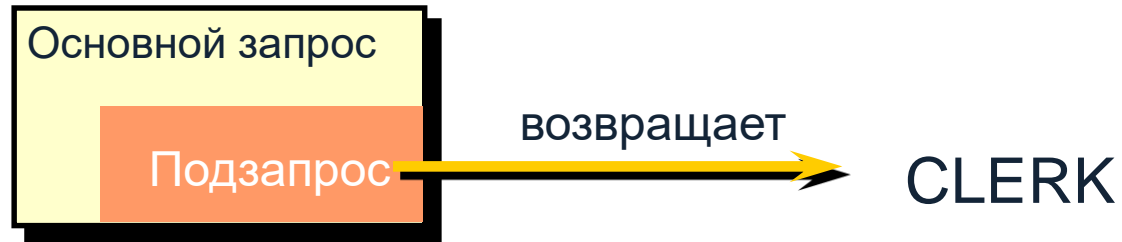
```
ENAME  
-----  
KING  
FORD  
SCOTT
```

# Советы по использованию подзапросов

- Закрывайте подзапросы в скобки.
- Располагайте подзапросы в правой части операторов сравнения.
- Не используйте предложение ORDER BY в подзапросах.
- Используйте «однострочные» операторы для результатов подзапросов, возвращающих одну строку.
- Используйте «многострочные» операторы для результатов подзапросов, возвращающих набор строк.

# Виды подзапросов

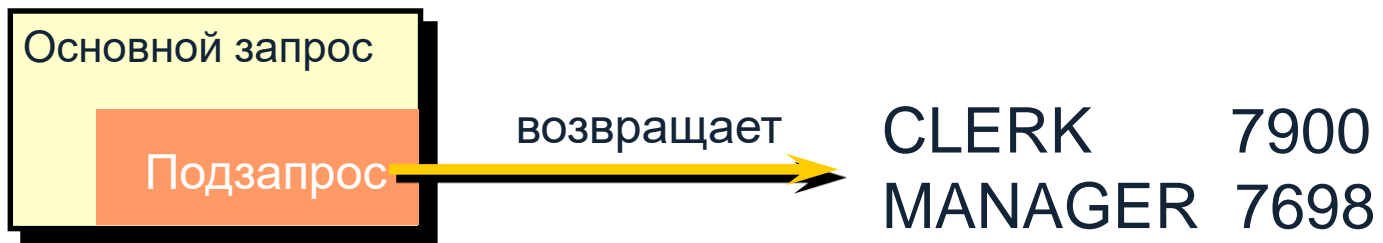
- Однострочный подзапрос



- Многострочный подзапрос



- Многостолбцовый подзапрос



# Однострочные подзапросы

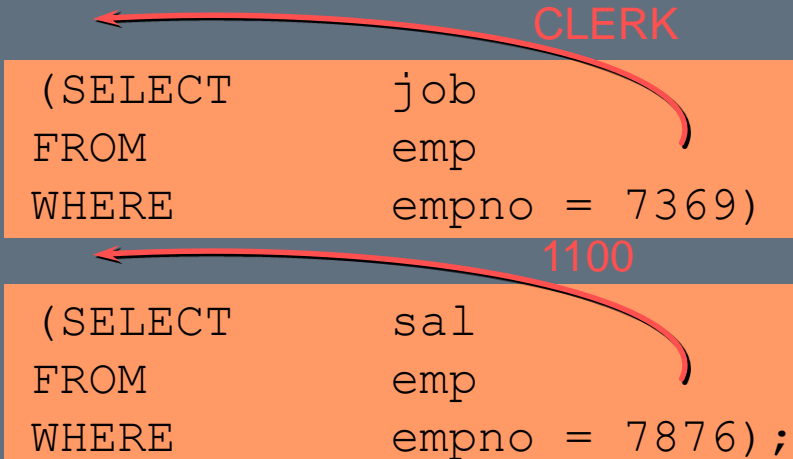
- Возвращает только одну строку
- Используйте «однострочные» операторы

Оператор	Значение
=	Равны
>	Больше
>=	Больше либо равен
<	Меньше
<=	Меньше либо равен
<>	Не равны



# Выполнение однострочных подзапросов


```
SQL> SELECT      ename, job
2  FROM          emp
3  WHERE         job =
4                (SELECT      job
5                 FROM          emp
6                 WHERE         empno = 7369)
7  AND          sal >
8                (SELECT      sal
9                 FROM          emp
10                WHERE         empno = 7876);
```



ENAME	JOB
-----	-----
MILLER	CLERK

# Использование группировок в подзапросах

```
SELECT ename, job
FROM emp
WHERE sal = (SELECT MIN(sal)
             FROM emp);
```

A red box highlights the subquery `(SELECT MIN(sal) FROM emp);`. A red arrow points from the number `2500` to the `sal =` part of the main query's WHERE clause.

ENAME	JOB
-----	-----
MILLER	CLERK

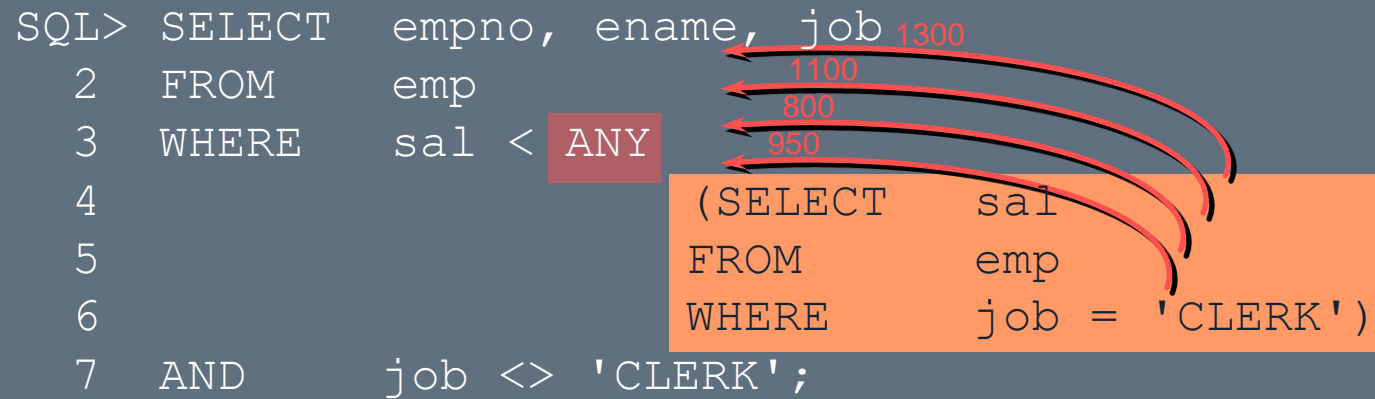
# Многострочные подзапросы

- Возвращают более одной строки
- Используйте «многострочные» операторы сравнения

Оператор	Значение
IN	Совпадает с любым из списка
ANY SOME	Сравнивает значение со всеми значениями в смысле СУЩЕСТВУЕТ
ALL	Сравнивает значение со всеми значениями в смысле ДЛЯ ВСЕХ

# Использование оператора ANY в многострочных подзапросах

```
SQL> SELECT empno, ename, job 1300
2 FROM emp 1100
3 WHERE sal < ANY 800
4 (SELECT sal 950
5 FROM emp
6 WHERE job = 'CLERK')
7 AND job <> 'CLERK';
```



EMPNO	ENAME	JOB
7654	MARTIN	SALESMAN
7521	WARD	SALESMAN

# Использование оператора ALL в многострочных подзапросах

```
SELECT empno, ename, job, sal
FROM emp
WHERE sal < ALL
      (SELECT sal
       FROM empl
       WHERE job = 'CLERK')
AND job <> 'CLERK';
```

9000, 6000, 4200

# Значения null в подзапросах

```
SQL> SELECT  employee.ename
2  FROM      emp employee
3  WHERE     employee.empno NOT IN
4           (SELECT manager.mgr
5            FROM   emp manager);
no rows selected.
```

<b>NOT IN</b>	<b>!=ALL</b>
<b>IN</b>	<b>=ANY</b>

# Использование подзапросов в предложении FROM

```
SQL> SELECT ename, sal  
2 FROM (SELECT * FROM EMP ORDER BY sal)  
3 WHERE rownum > 10;
```

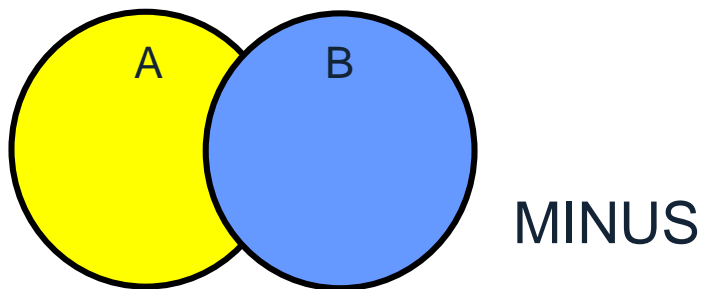
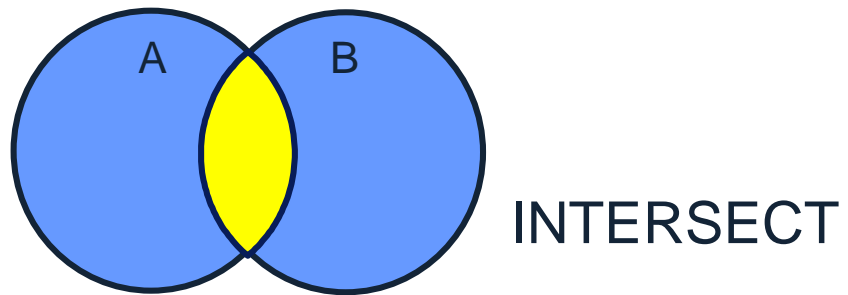
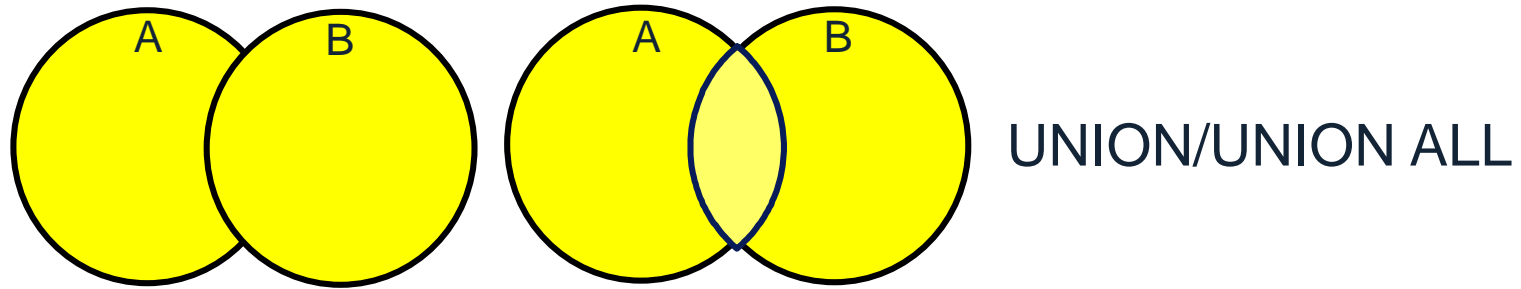
```
SQL> SELECT ename, sal  
2 FROM (SELECT * FROM EMP WHERE depno = 20)  
3 WHERE sal > 1000
```

# Использование подзапросов в предложении SELECT

```
SQL> SELECT ename, (SELECT SYSDATE FROM DUAL)  
2          FROM emp;
```



# Операции над множествами



# Руководство по операциям на множествах

- В запросах `SELECT` должно возвращаться одинаковое количество столбцов
- Типы столбцов первого и второго запроса должны попарно совпадать.
- Для удобства можно использовать скобки.
- `ORDER BY` может быть использован только в самом конце выражения

# Особенности Oracle Server

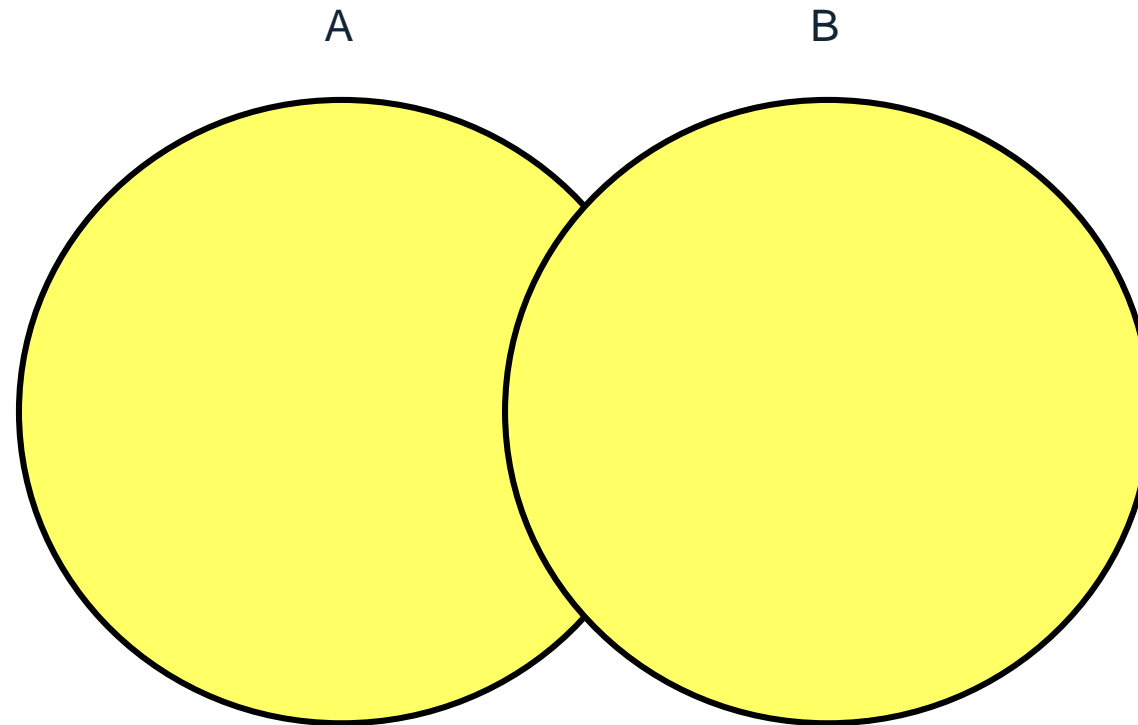
- Повторяющиеся строки автоматически исключаются, за исключением операции `UNION ALL`.
- Имена столбцов первого запроса становятся именами столбцов результирующего запроса.
- По умолчанию результаты отсортированы по возрастанию, за исключением `UNION ALL`.

## Таблицы в примерах

- EMP: Таблица хранит текущую должность рабочего
- JOB\_HISTORY: Таблица хранит пару значений (рабочий-должность). В таблице сохранены все когда-либо работавшие служащие и все должности которые занимал служащий **за исключением текущей**.

# UNION

- Оператор `UNION` вернет строки из обеих таблиц, за исключением повторяющихся.



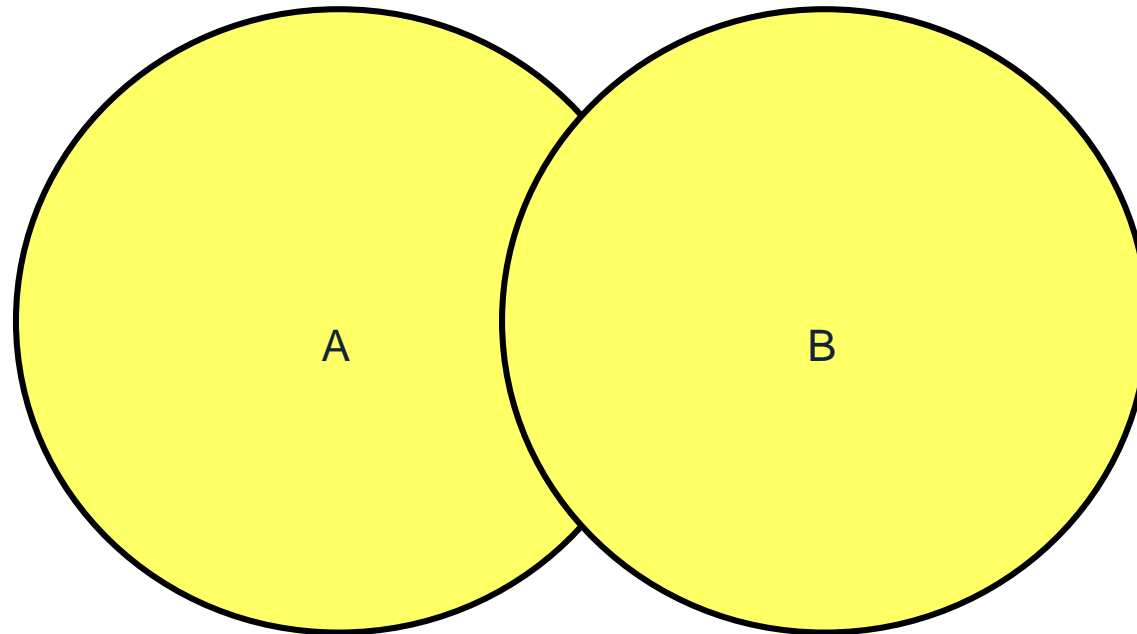
## UNION

- Запрос вернет историю всех должностей всех рабочих (включая уволенных).

```
SELECT empno, job
FROM emp
UNION
SELECT empno, job
FROM job_history;
```

## UNION ALL

- Оператор `UNION ALL` вернет строки из обеих таблиц, включая повторяющиеся.



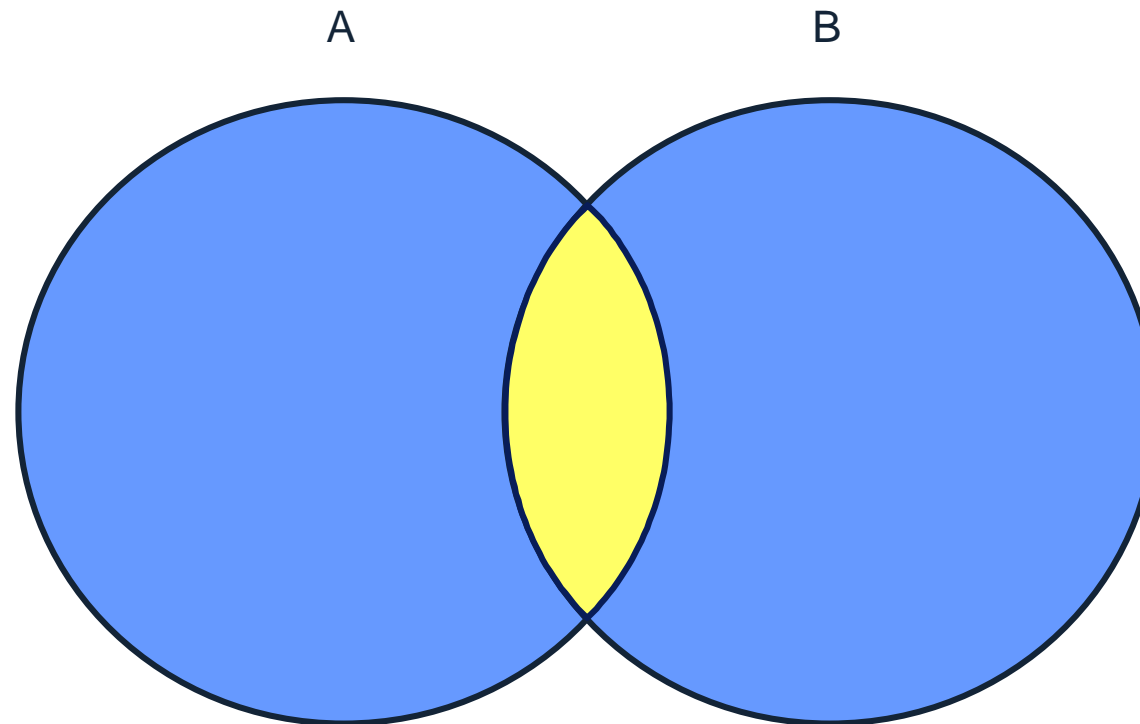
# UNION ALL

```
SELECT empno, job, depno
FROM emp
UNION ALL
SELECT emp, job, depno
FROM job_history
ORDER BY empno;
```



## INTERSECT (Пересечение)

- INTERSECT возвращает строки, которые есть в обоих запросах



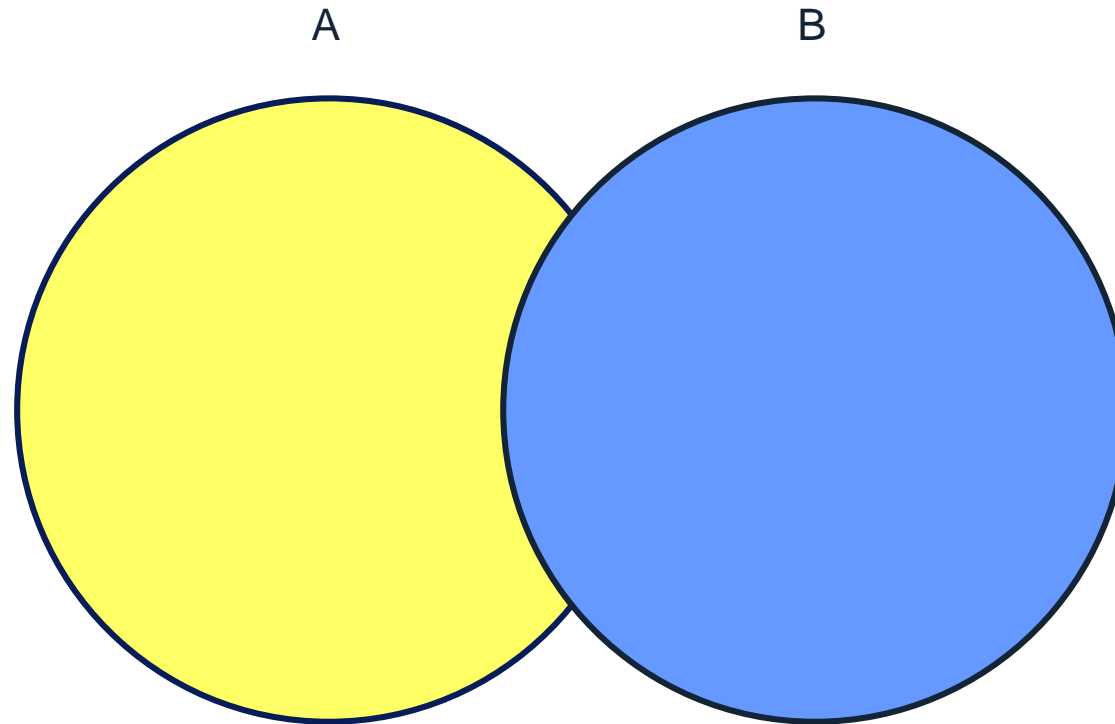
## Using the INTERSECT Operator

- Запрос вернет служащих, которые когда либо раньше работали на должностях, которые занимают сейчас.

```
SELECT empno, job
FROM emp
INTERSECT
SELECT empno, job
FROM job_history;
```

# MINUS

- Оператор `MINUS` все неповторяющиеся строки из первого запроса, которых нет во втором запросе



# MINUS

- Запрос вернет служащих, которые никогда не меняли должность (их нет в таблице job\_history)

```
SELECT empno
FROM emp
MINUS
SELECT empno
FROM job_history;
```

# Использование операторов

- Используя операторы действий над множествами
  - Убедитесь что все запросы возвращают одинаковое количество атрибутов
  - Типы атрибутов попарно совпадают

```
SELECT empno, job, salary
FROM emp
UNION
SELECT empno, job, 0
FROM job_history;
```

## Использование предложения ORDER BY в запросах с операциями на множествах

- Предложение ORDER BY может быть использовано только раз, в конце запроса.
- Отдельные подзапросы внутри запроса с операциями на множества не могут использовать предложение ORDER BY.
- В предложении ORDER BY используют колонки (или их псевдонимы) первого SELECTа
- По умолчанию результаты запроса отсортированы по первому столбцу по возрастанию.

Q&A

Thank You

