

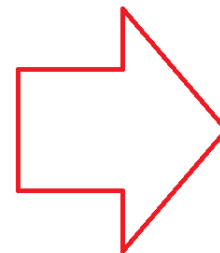
ОБЪЕДИНЕНИЯ ТАБЛИЦ

Сичная Мария, начальник бюро платежных приложений

- ❑ Простое объединение таблиц (JOIN)
- ❑ Самообъединение (SELF-JOIN)
- ❑ Внешнее соединение (OUTER JOIN):
 - LEFT OUTER
 - RIGHT OUTER
 - FULL OUTER
- ❑ Декартово пересечение таблиц (CROSS JOIN)

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



EMPNO	DEPTNO	DNAME
7369	20	RESEARCH
7499	30	SALES
7521	30	SALES
7566	20	RESEARCH
7654	30	SALES
7698	30	SALES
7782	10	ACCOUNTING
7788	20	RESEARCH
7839	10	ACCOUNTING
7844	30	SALES
7876	20	RESEARCH
7900	30	SALES
7902	20	RESEARCH
7934	10	ACCOUNTING

SELECT EMP.EMPNO, EMP.DEPTNO, DEPT.DNAME
FROM EMP

JOIN DEPT ON DEPT.DEPTNO = EMP.DEPTNO;

Условие ON применяется:

- ☐ Для указания столбцов по которым следует объединять таблицы.
- ☐ Синтаксически отделяет раздел с условиями объединения таблиц от раздела поиска результата по условию WHERE.
- ☐ Упрощает код для понимания, т.е. делает его удобочитаемым.

SELECT

E.EMPNO, E.ENAME, E.DEPTNO,
D.DEPTNO, D.DNAME, D.LOC

FROM EMP E

JOIN DEPT D ON D.DEPTNO = E.DEPTNO;

SELECT

E.EMPNO, E.ENAME, E.DEPTNO,
D.DEPTNO, D.DNAME, D.LOC

FROM EMP E, DEPT D

WHERE D.DEPTNO = E.DEPTNO;

EMPNO	ENAME	DEPTNO	DEPTNO	DNAME	LOC
7369	SMITH	20	20	RESEARCH	DALLAS
7499	ALLEN	30	30	SALES	CHICAGO
7521	WARD	30	30	SALES	CHICAGO
7566	JONES	20	20	RESEARCH	DALLAS
7654	MARTIN	30	30	SALES	CHICAGO
7698	BLAKE	30	30	SALES	CHICAGO
7782	CLARK	10	10	ACCOUNTING	NEW YORK
7788	SCOTT	20	20	RESEARCH	DALLAS
7839	KING	10	10	ACCOUNTING	NEW YORK
7844	TURNER	30	30	SALES	CHICAGO
7876	ADAMS	20	20	RESEARCH	DALLAS
7900	JAMES	30	30	SALES	CHICAGO
7902	FORD	20	20	RESEARCH	DALLAS
7934	MILLER	10	10	ACCOUNTING	NEW YORK

```
SELECT
    E.EMPNO, E.ENAME, E.DEPTNO,
    D.DEPTNO, D.DNAME, D.LOC, S.GRADE
FROM EMP E
    JOIN DEPT D ON D.DEPTNO = E.DEPTNO
    JOIN SALGRADE S ON E.SAL BETWEEN S.LOSAL AND
S.HISAL;
```

```
SELECT
    E.EMPNO, E.ENAME, E.DEPTNO,
    D.DEPTNO, D.DNAME, D.LOC, S.GRADE
FROM EMP E, DEPT D, SALGRADE S
WHERE
    D.DEPTNO = E.DEPTNO
    AND E.SAL BETWEEN S.LOSAL AND S.HISAL;
```

Для добавления дополнительных условий используется AND:

```
SELECT E.EMPNO, E.ENAME, E.DEPTNO, D.DEPTNO, D.DNAME  
FROM EMP E  
      JOIN DEPT D ON D.DEPTNO = E.DEPTNO  
                AND E.EMPNO = 7839;
```

Для добавления дополнительных условий используется WHERE:

```
SELECT E.EMPNO, E.ENAME, E.DEPTNO, D.DEPTNO, D.DNAME  
FROM EMP E  
      JOIN DEPT D ON D.DEPTNO = E.DEPTNO  
WHERE E.EMPNO = 7839;
```

Альтернативный способ:

```
SELECT E.EMPNO, E.ENAME, E.DEPTNO, D.DEPTNO, D.DNAME  
FROM EMP E, DEPT D  
WHERE E.EMPNO = 7839  
      AND D.DEPTNO = E.DEPTNO;
```

ОДНОИМЕННЫЕ СТОЛБЦЫ В ЗАПРОСЕ

- ❑ Для одноименных столбцов таблиц необходимо использовать префиксы (имена таблиц или псевдонимы таблиц).
- ❑ Для повышения производительности используются префиксы для столбцов с указанием таблицы или псевдонима.
- ❑ Чем меньше SQL код, тем меньше места он занимает в памяти и тем быстрее происходит разбор SQL инструкции.

Случай, когда в запросе используется одна и та же таблица несколько раз, и данная таблица соединяется сама с собой, называется самообъединением.

Сотрудники

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

Менеджеры

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30

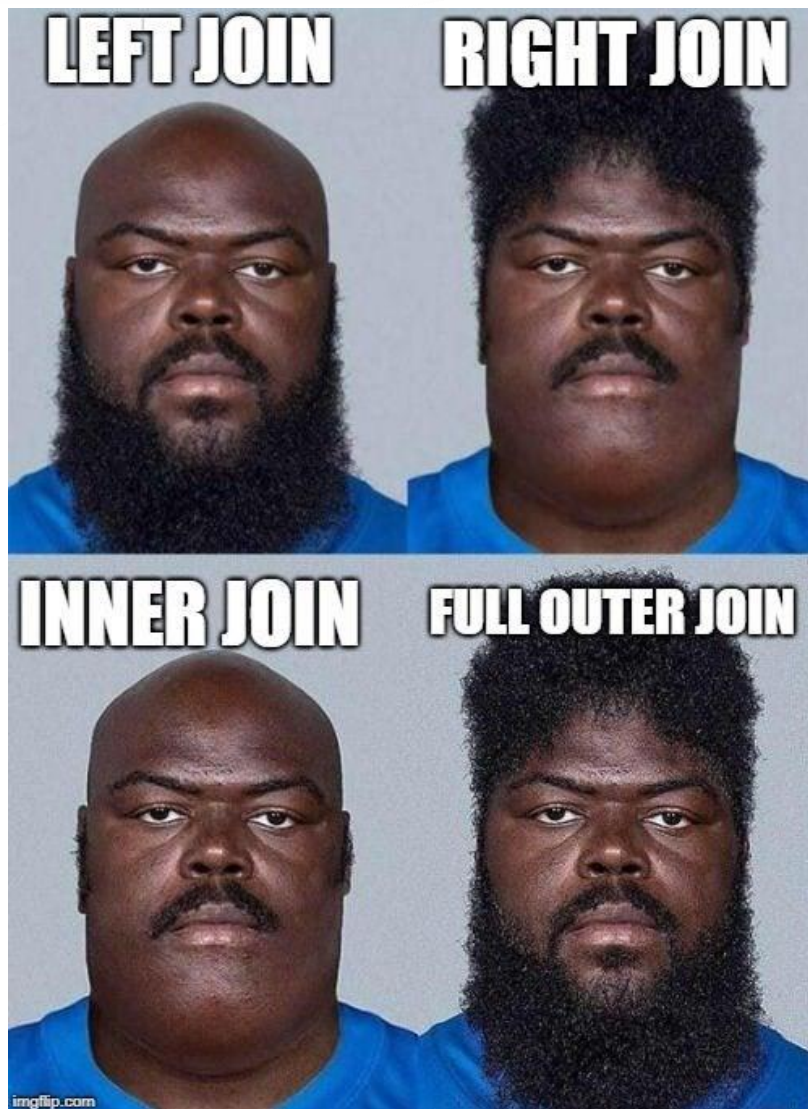
```
SELECT E.ENAME EMPLOYEE, M.ENAME MANAGER  
FROM EMP E  
      JOIN EMP M ON E.MGR = M.EMPNO;
```

```
SELECT E.ENAME EMPLOYEE, M.ENAME MANAGER  
FROM EMP E, EMP M  
WHERE E.MGR = M.EMPNO;
```

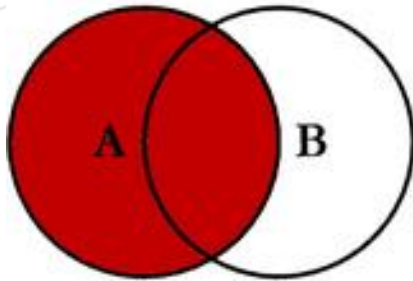
EMPLOYEE	MANAGER
SCOTT	JONES
FORD	JONES
ALLEN	BLAKE
WARD	BLAKE
MARTIN	BLAKE
TURNER	BLAKE
JAMES	BLAKE
MILLER	CLARK
ADAMS	SCOTT
JONES	KING
BLAKE	KING
CLARK	KING
SMITH	FORD

- ❑ Объединение двух таблиц, возвращающее только совпадающие строки, называется внутренним соединением.
- ❑ Объединение двух таблиц, возвращающее результат внутреннего соединения, а также строки из левой таблицы, отсутствующие в правой (или левой), называется левым (или правым) внешним соединением.
- ❑ Объединение двух таблиц, возвращающее результат внутреннего соединения, а также результаты слева и справа, отсутствующие в другой таблице, называется полным внешним соединением.

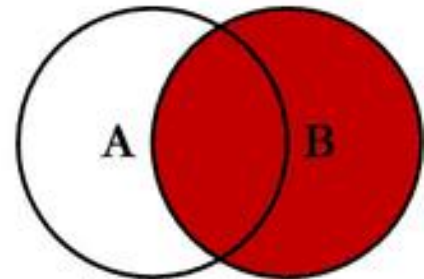
ВНУТРЕННЕЕ И ВНЕШНЕЕ СОЕДИНЕНИЯ



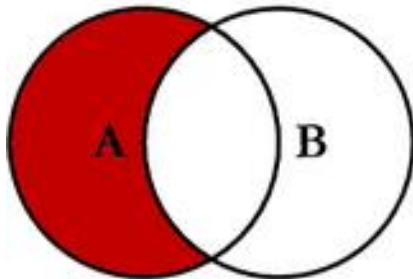
SQL JOINS



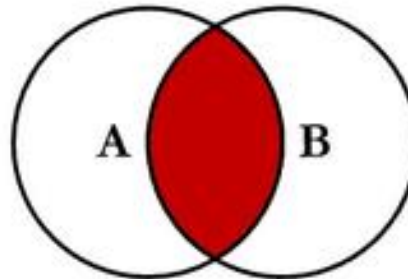
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



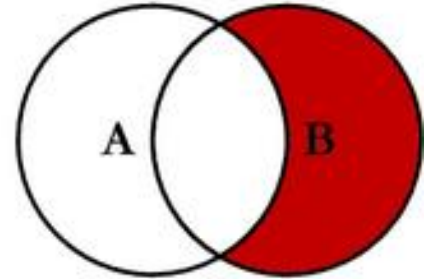
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



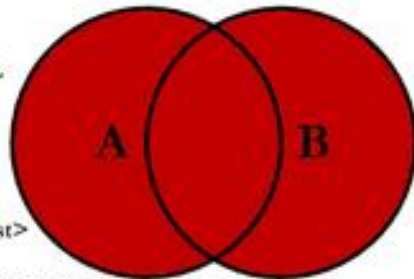
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



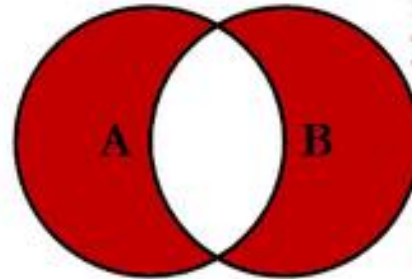
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

ЛЕВОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ LEFT JOIN

```
SELECT D.DNAME, E.ENAME, E.DEPTNO  
FROM DEPT D  
LEFT JOIN EMP E ON E.DEPTNO = D.DEPTNO;
```

```
SELECT D.DNAME, E.ENAME, E.DEPTNO  
FROM DEPT D, EMP E  
WHERE E.DEPTNO (+) = D.DEPTNO;
```

Знак (+) устанавливается после имени столбца таблицы, в которой есть недостаток строк.

DNAME	ENAME	DEPTNO
RESEARCH	SMITH	20
SALES	ALLEN	30
SALES	WARD	30
RESEARCH	JONES	20
SALES	MARTIN	30
SALES	BLAKE	30
ACCOUNTING	CLARK	10
RESEARCH	SCOTT	20
ACCOUNTING	KING	10
SALES	TURNER	30
RESEARCH	ADAMS	20
SALES	JAMES	30
RESEARCH	FORD	20
ACCOUNTING	MILLER	10
OPERATIONS	-	-

```
SELECT E.ENAME, E.DEPTNO, D.DNAME  
FROM EMP E  
      RIGHT JOIN DEPT D ON D.DEPTNO = E.DEPTNO;
```

```
SELECT E.ENAME, E.DEPTNO, D.DNAME  
FROM EMP E, DEPT D  
WHERE D.DEPTNO = E.DEPTNO (+);
```

Знак (+) устанавливается после имени столбца таблицы, в которой есть недостаток строк.

ENAME	DEPTNO	DNAME
SMITH	20	RESEARCH
ALLEN	30	SALES
WARD	30	SALES
JONES	20	RESEARCH
MARTIN	30	SALES
BLAKE	30	SALES
CLARK	10	ACCOUNTING
SCOTT	20	RESEARCH
KING	10	ACCOUNTING
TURNER	30	SALES
ADAMS	20	RESEARCH
JAMES	30	SALES
FORD	20	RESEARCH
MILLER	10	ACCOUNTING
-	-	OPERATIONS

ПОЛНОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ FULL JOIN

table1

num	name
1	a
2	b
3	c

table2

num	value
1	xxx
3	yyy
5	zzz

```
SELECT *  
FROM TABLE1 A  
FULL JOIN TABLE2 B ON B.NUM=A.NUM;
```

result

a.num	name	b.num	value
1	a	1	xxx
2	b		
3	c	3	yyy
		5	zzz

Декартово пересечение образуется когда:

- ☐ Условие объединения отсутствует;
- ☐ Условие объединения ошибочно;
- ☐ Необходимо все строки из первой таблицы объединить со всеми строками из второй таблицы.

Чтобы избежать неправильных декартовых пересечений таблиц, необходимо использовать условия объединения.

```
SELECT *  
FROM TABLE1 A  
      CROSS JOIN TABLE2 B;
```

```
SELECT *  
FROM TABLE1 A, TABLE2 B;
```

result

table1.num	name	table2.num	value
1	a	1	xxx
1	a	3	yyy
1	a	5	zzz
2	b	1	xxx
2	b	3	yyy
2	b	5	zzz
3	c	1	xxx
3	c	3	yyy
3	c	5	zzz

Операторы множеств объединяют результаты двух или более запросов в один результат.

- ☐ UNION;
- ☐ UNION ALL;
- ☐ INTERSECT;
- ☐ MINUS.

Правила применения операторов множеств:

- ☐ Количество столбцов объединяемых запросов, заданных в предложении SELECT, должно совпадать;
- ☐ Тип данных (групповой тип данных: NUMBER, CHAR, DATE) каждого столбца во втором запросе должен соответствовать типу данных соответствующего столбца в первом запросе;
- ☐ Допускается использование скобок для изменения последовательности действий при выполнении запроса;
- ☐ Предложение ORDER BY можно размещать только последней инструкцией оператора множеств.

Правила применения операторов множеств:

- ☐ Дублирующие строки автоматически исключаются из результата, кроме оператора UNION ALL.
- ☐ Имена столбцов результата формируются из имен столбцов, заданных в первом операторе SELECT.
- ☐ Полученный результат по умолчанию сортируется по возрастанию значений первого столбца, кроме оператора UNION ALL.

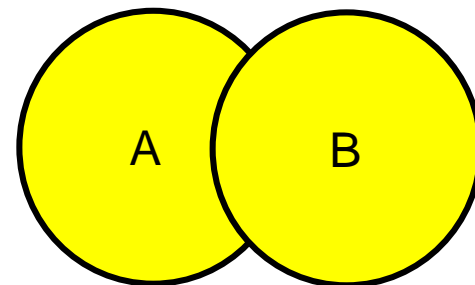
Результаты запросов, используемые в операторах множеств, по количеству столбцов и их типов данных (по группам) должны совпадать:

- ☐ Если оба запроса выбирают значения типа CHAR, равные по длине, то результат имеет тип CHAR данной длины;
- ☐ Если запросы выбирают значения типа CHAR с различными длинами, то результат является типом VARCHAR2 с длиной, соответствующей большему из значений CHAR;
- ☐ Если запрос/запросы выбирают значения типа VARCHAR2, то результат имеет тип VARCHAR2;
- ☐ Если запросы выбирают числовые данные, то результирующий тип данных является числовым;
- ☐ В запросах, использующих операторы множеств, сервер Oracle не выполняет неявное преобразование между разными группами типов данных, поэтому **если результаты объединяемых столбцов имеют различные типы, то сервер Oracle возвращает ошибку.**

Оператор UNION возвращает все строки, которые выбираются из запросов, предварительно устранив любые повторяющиеся строки.

Необходимо соблюдать следующие правила использования оператора UNION:

- ☐ Число столбцов, выбираемых запросами, и их типы данных должны совпадать;
- ☐ Наименования столбцов НЕ обязательно должны быть идентичными;
- ☐ Объединение применяется ко всем выбираемым столбцам;
- ☐ NULL-значения не игнорируются во время проверки дубликатов;
- ☐ Оператор IN имеет больший приоритет, чем оператор UNION;
- ☐ По умолчанию результат сортируется по значению первого столбца из оператора SELECT.



```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
UNION  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20;
```



JOB
MANAGER
PRESIDENT
CLERK



JOB
CLERK
MANAGER
ANALYST
CLERK
ANALYST



?

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
UNION  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20;
```



JOB
MANAGER
PRESIDENT
CLERK



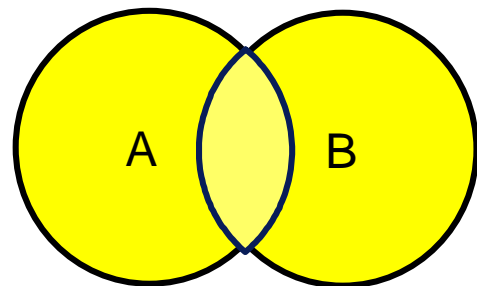
JOB
CLERK
MANAGER
ANALYST
CLERK
ANALYST



JOB
ANALYST
CLERK
MANAGER
PRESIDENT

Оператор UNION ALL возвращает строки из всех запросов, объединенных этим оператором, включая дублирующие строки.

- ❑ Для оператора UNION ALL справедливы такие же правила как для UNION, однако результат содержит дублирующие строки и не является отсортированным.
- ❑ Применение оператора DISTINCT не позволяет избавиться от дублирующих строк, т.к. они формируются в результате объединения результатов, а не в результате выборки оператором SELECT.
- ❑ Оператор UNION ALL производительнее UNION, так как не выполняются дополнительные проверки на уникальность.



ОПЕРАТОР UNION ALL

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
UNION ALL  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20;
```



JOB
MANAGER
PRESIDENT
CLERK



JOB
CLERK
MANAGER
ANALYST
CLERK
ANALYST



?

ОПЕРАТОР UNION ALL

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
UNION ALL  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20;
```



JOB
MANAGER
PRESIDENT
CLERK



JOB
CLERK
MANAGER
ANALYST
CLERK
ANALYST

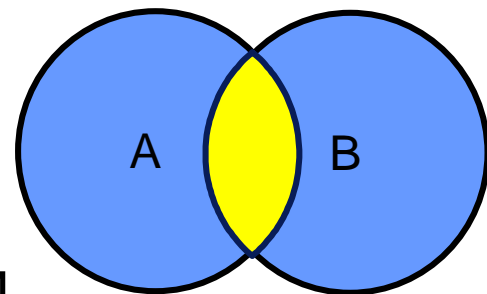


JOB
MANAGER
PRESIDENT
CLERK
CLERK
MANAGER
ANALYST
CLERK
ANALYST

Оператор INTERSECT возвращает только общие строки для всех запросов, к которым он применяется.

Необходимо соблюдать следующие правила использования оператора INTERSECT:

- ☐ Число столбцов, выбираемых в предложении SELECT, и их типы данных должны совпадать;
- ☐ Наименования столбцов НЕ обязательно должны быть идентичными;
- ☐ Пересечение (INTERSECT) не игнорирует NULL-значения ;
- ☐ Порядок применения оператора INTERSECT к трем и более запросам не изменяет результат.



ОПЕРАТОР INTERSECT

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
INTERSECT  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20;
```



JOB
MANAGER
PRESIDENT
CLERK



JOB
CLERK
MANAGER
ANALYST
CLERK
ANALYST



?

ОПЕРАТОР INTERSECT

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
INTERSECT  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20;
```



JOB
CLERK
MANAGER

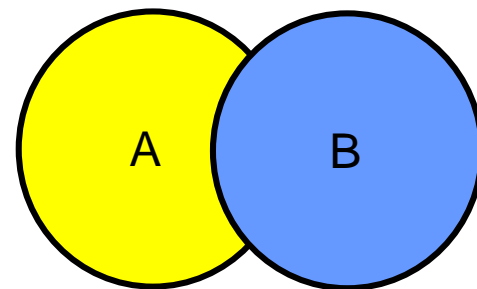


JOB
MANAGER
PRESIDENT
CLERK



JOB
CLERK
MANAGER
ANALYST
CLERK
ANALYST

Оператор MINUS возвращает уникальные значения из результата первого оператора SELECT, отсутствующие в результате второго оператора SELECT.



- ❑ Оператор MINUS также как и UNION обладает ограничением на результат запросов, которые должны быть совместимы по объединению, т.е. содержать одинаковое количество столбцов, и каждый столбец первого запроса должен быть того же типа данных (или автоматически приводиться к нему), что и находящийся в том же месте столбец второго запроса.

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
MINUS  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20;
```



JOB
MANAGER
PRESIDENT
CLERK



JOB
CLERK
MANAGER
ANALYST
CLERK
ANALYST



?

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
MINUS  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20;
```



JOB
MANAGER
PRESIDENT
CLERK



JOB
CLERK
MANAGER
ANALYST
CLERK
ANALYST



JOB
PRESIDENT

ИСПОЛЬЗОВАНИЕ ORDER BY С ОПЕРАТОРАМИ МНОЖЕСТВ

- ❑ Предложение ORDER BY можно использовать только один раз, последней инструкцией составного запроса;
- ❑ Предложение ORDER BY может содержать наименование столбца, псевдоним или позиционный номер столбца, объявленного в первом операторе SELECT;
- ❑ Если предложение ORDER BY не указано, то результат сортируется в порядке возрастания значений из первого столбца первого запроса.

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
UNION  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20  
ORDER BY JOB;
```

```
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 10  
UNION  
SELECT JOB  
FROM EMP  
WHERE DEPTNO = 20  
ORDER BY 1;
```

СПАСИБО ЗА ВНИМАНИЕ !
ВОПРОСЫ ?

www.compassplus.ru