



Базы данных и Информационные системы

9/15 Вывод данных из нескольких таблиц

Кузиков Б.О.
Сумы, СумГУ
2013

Задачи занятия

- ▶ После завершения занятия вы должны уметь и знать следующее:
 - ▶ Писать запросы `SELECT` для доступа к данным нескольких таблиц с использованием эквисоединения и неэквисоединения
 - ▶ Просматривать данные, в целом не удовлетворяющие требованиям соединений, с помощью внешнего соединения
 - ▶ Соединять таблицу саму с собой



Получение данных из нескольких таблиц

EMP

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

DEPT

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



EMPNO	DEPTNO	LOC
7839	10	NEW YORK
7698	30	CHICAGO
7782	10	NEW YORK
7566	20	DALLAS
7654	30	CHICAGO
7499	30	CHICAGO
...		
14 rows selected.		



Что есть соединение?

- ▶ Используйте соединения, чтобы получить данные более чем из одной таблицы.

```
SELECT    table1.column, table2.column
FROM      table1, table2
WHERE     table1.column1 = table2.column2;
```

- ▶ Условия соединения записываются в предложении **WHERE**.
- ▶ Если одинаковое имя столбца встречается в нескольких таблицах, то следует использовать префиксы в виде имен таблиц. (Например EMP.DEPTNO, DEPT.DEPTNO вместо DEPTNO)



Декартово произведение

- ▶ Декартово произведение формируется, если:
 - ▶ Условие соединения опущено
 - ▶ Условие соединения некорректно
 - ▶ В результате соединения все строки первой таблицы соединяются со всеми строками второй таблицы
- ▶ Чтобы избежать декартова произведения, всегда включайте корректное условие соединения в предложение **WHERE**.



Формирование декартова произведения

EMP (14 строк)

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

DEPT (4 строки)

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



“Декартово
произведение:
14*4=56 строк”



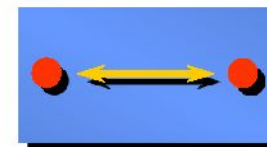
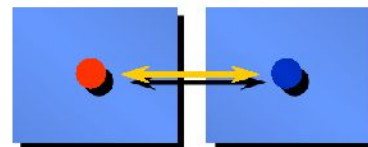
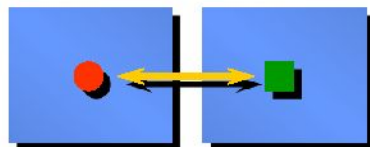
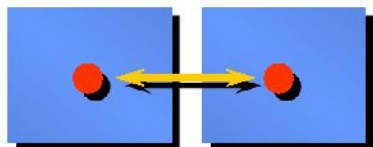
ENAME	DNAME
KING	ACCOUNTING
BLAKE	ACCOUNTING
...	
KING	RESEARCH
BLAKE	RESEARCH
...	
56 rows selected.	



Виды соединений

Не-экви

Само



Экви(внутренне)

Внешнее



Что есть эквисоединение?

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
...		
14 rows selected.		

Первичный
ключ

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS
...		
14 rows selected.		

Внешний
ключ

Получение записей с использованием эквисоединений

```
SQL> SELECT emp.empno, emp.ename, emp.deptno,  
2         dept.deptno, dept.loc  
3 FROM emp, dept  
4 WHERE emp.deptno=dept.deptno;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7566	JONES	20	20	DALLAS
...				

14 rows selected.

Разрешение неоднозначности имен столбцов

- ▶ Используйте префикс в виде имени таблицы в том случае, если несколько таблиц имеют одноименные столбцы.
- ▶ Использование префиксов увеличивает производительность.
- ▶ Различайте одноименные столбцы различных таблиц путем введения псевдонимов.



Дополнительные условия поиска с использованием оператора AND

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
...		
14 rows selected.		

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS
...		
14 rows selected.		

Использование псевдонимов таблиц

- ▶ Упрощайте запросы с помощью псевдонимов таблиц.

```
SQL> SELECT emp.empno, emp.ename, emp.deptno,  
2         dept.deptno, dept.loc  
3 FROM    emp, dept  
4 WHERE   emp.deptno=dept.deptno;
```

```
SQL> SELECT e.empno, e.ename, e.deptno,  
2         d.deptno, d.loc  
3 FROM    emp e, dept d  
4 WHERE   e.deptno=d.deptno;
```


Соединение более двух таблиц

CUSTOMER

NAME	CUSTID
JOCKSPORTS	100
TKB SPORT SHOP	101
VOLLYRITE	102
JUST TENNIS	103
K+T SPORTS	105
SHAPE UP	106
WOMENS SPORTS	107
...	...
9 rows selected.	

ORD

CUSTID
101
102
104
106
102
106
106
...
21 rows

ITEM

ORDID	ITEMID
610	3
611	1
612	1
601	1
602	1
...	...
64 rows selected.	

Неэквисоединения

EMP

EMPNO	ENAME	SAL
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
...		
14 rows selected.		

SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999



“зарплата в таблице EMP находится в диапазоне между значениями в таблице SALGRADE”



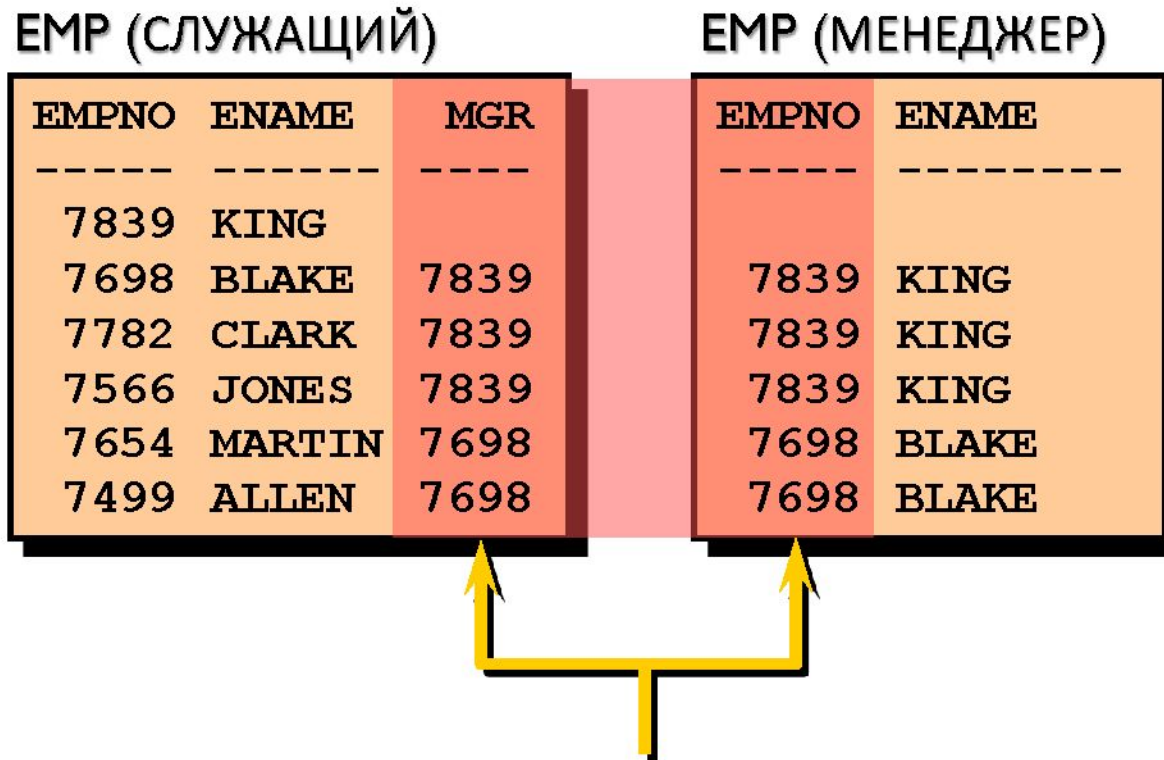
Получение записей с использованием неэквисоединений

```
SQL> SELECT e.ename, e.sal, s.grade
2 FROM emp e, salgrade s
3 WHERE e.sal
4 BETWEEN s.losal AND s.hisal;
```

ENAME	SAL	GRADE
JAMES	950	1
SMITH	800	1
ADAMS	1100	1
...		

14 rows selected.

Самосоединение



"MGR в таблице СЛУЖАЩИЙ совпадает с EMPNO в таблице МЕНЕДЖЕР"



Соединение таблицы с ней же

```
SQL> SELECT worker.ename || ' works for ' || manager.ename  
2 FROM emp worker, emp manager  
3 WHERE worker.mgr = manager.empno;
```

```
WORKER. ENAME | | 'WORKSFOR' | | MANAG  
-----
```

```
BLAKE works for KING  
CLARK works for KING  
JONES works for KING  
MARTIN works for BLAKE  
...  
13 rows selected.
```

Типы соединений

- ▶ Стандарт SQL:1999 включает такие типы соединений:
 - ▶ Natural joins:
 - ▶ Предложение `NATURAL JOIN`
 - ▶ Предложение `USING`
 - ▶ Предложение `ON`
 - ▶ OUTER joins:
 - ▶ `LEFT OUTER JOIN`
 - ▶ `RIGHT OUTER JOIN`
 - ▶ `FULL OUTER JOIN`
 - ▶ Cross joins



Объединение таблиц (Синтаксис SQL:1999)

- ▶ Используйте предложение Join чтобы объединить две или более таблицы:

```
SELECT      table1.column, table2.column
FROM        table1
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
  ON (table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
  ON (table1.column_name = table2.column_name)] |
[CROSS JOIN table2];
```



Natural Join

- ▶ NATURAL JOIN - объединение таблиц на основе колонок, которые имеют одинаковые имена
- ▶ В объединение попадают все строки, которые имеют одинаковое значение в совпадающих колонках
- ▶ Если колонки имеют одинаковое имя, но разные то запрос вернет ошибку.



Natural Join

```
SELECT empno, emp.deptno, loc
FROM emp
NATURAL JOIN dept ;
```

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

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

EMPNO	DEPTNO	LOC
7839	10	NEW YORK
7698	30	CHICAGO
7782	10	NEW YORK
7566	20	DALLAS
...		

14 rows selected.

Joins с предложением USING

- ▶ USING позволяет указать по каким колонкам будет произведено эквисоединение, если типы колонок разные
- ▶ USING позволяет указать по какому столбцу производить объединение, если подходят несколько столбцов имеют одинаковые имена
- ▶ В предложении USING не нужно указывать какой из таблиц принадлежит столбец
- ▶ NATURAL JOIN и USING взаимно исключают друг друга



Пример с USING

```
SELECT empno, ename,  
       loc, e.deptno  
FROM   emp e JOIN dept  
       USING (deptno) ;
```

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
...		

14 rows selected.

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS
...		

14 rows selected.

Joins с использованием предложения ON

- ▶ Предложение ON используется для экви и не-экви соединений
- ▶ Предложение ON отделяет условие соединения от других условий
- ▶ Предложение ON **делает** запрос проще для понимания.



Пример с ON

```
SELECT empno, ename,  
       loc, e.deptno  
FROM   emp e JOIN dept d  
       USING (d.deptno=e.deptno) ;
```

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
...		

14 rows selected.

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS
...		

14 rows selected.

Объединение нескольких таблиц с ON

```
SELECT emplno, deptno, city
FROM emp e
JOIN dept d
ON d.deptno = e.deptno
JOIN locations l
ON d.locno = l.locno;
```

...



Само-объединение с ON

```
SELECT worker.ename emp, manager.ename mgr
FROM   emp worker JOIN emp manager
ON     (worker.mgr = manager.empno) ;
```

EMP (СЛУЖАЩИЙ)			EMP (МЕНЕДЖЕР)	
EMPNO	ENAME	MGR	EMPNO	ENAME
7839	KING		7839	KING
7698	BLAKE	7839	7839	KING
7782	CLARK	7839	7839	KING
7566	JONES	7839	7839	KING
7654	MARTIN	7698	7698	BLAKE
7499	ALLEN	7698	7698	BLAKE

"MGR в таблице СЛУЖАЩИЙ совпадает с EMPNO в
таблице МЕНЕДЖЕР"

Дополнительные условия в Join

- ▶ Для наложения дополнительных ограничений можно использовать как предложение **WHERE** так и **ON**

```
SELECT e.empno, e.ename, e.deptno,  
       d.deptno, d.loc  
FROM   emp e JOIN dept d  
ON     (e.deptno = d.deptno  
AND    e.mgr = 149 );
```

Или

```
SELECT e.empno, e.ename, e.deptno,  
       d.deptno, d.loc  
FROM   emp e JOIN dept d  
ON     (e.deptno = d.deptno)  
WHERE  e.mgr = 149 ;
```

Не-екви соединения с Join On

```
SELECT e.salary, j.losal, j.hisal,  
FROM emp e JOIN salgrade j  
ON e.sal  
BETWEEN j.losal AND j.hisal;
```

EMP

EMPNO	ENAME	SAL
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
...		
14 rows selected.		

SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

“зарплата в таблице EMP находится в диапазоне между значениями в таблице SALGRADE”

OUTER Joins (Внешнее соединение)

EMP		DEPT	
ENAME	DEPTNO	DEPTNO	DNAME
-----	-----	-----	-----
KING	10	10	ACCOUNTING
BLAKE	30	30	SALES
CLARK	10	10	ACCOUNTING
JONES	20	20	RESEARCH
...		...	
		40	OPERATIONS



В департаменте OPERATIONS
сотрудников нет



INNER Vs OUTER Joins

- ▶ В SQL:1999 объединение возвращающее строки, если значения присутствуют в обеих таблицах называется **INNER join**.
- ▶ Объединение, которое возвращает результаты **INNER join** и несовпавшие строки из левой (правой) таблицы называется **left (right) OUTER join**.
- ▶ Объединение, которое возвращает результаты **INNER join** и несовпавшие строки из левой и правой таблицы называется **full OUTER join**.

А	Б

А	Б

А	Б



RIGHT OUTER JOIN

```
SELECT e.ename, d.deptno, d.dname
FROM   emp e RIGHT OUTER JOIN dept d
ON     (e.dept = d.dept) ;
```

ENAME	DEPTNO	DNAME
-----	-----	-----
KING	10	ACCOUNTING
CLARK	10	ACCOUNTING
...		
NULL	40	OPERATIONS

15 rows selected.

OUTER JOIN

▶ LEFT OUTER JOIN

```
SELECT e.last_name, e.department_id, d.department_name
FROM   employees e LEFT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

▶ FULL OUTER JOIN

```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e FULL OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```



Декартово произведение

EMP (14 строк)

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

DEPT (4 строки)

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



“Декартово
произведение:
14*4=56 строк”



ENAME	DNAME
KING	ACCOUNTING
BLAKE	ACCOUNTING
...	
KING	RESEARCH
BLAKE	RESEARCH
...	
56 rows selected.	



Cross Joins

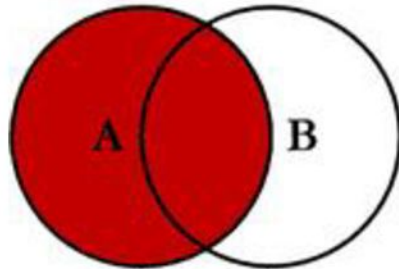
- ▶ CROSS JOIN возвращает декартово произведение двух таблиц

```
SELECT ename, dname  
FROM emp  
CROSS JOIN dept;
```

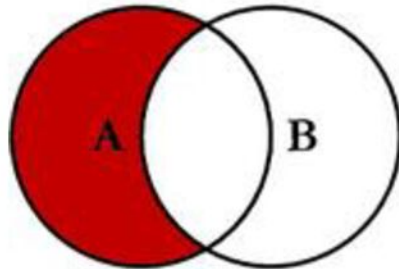
ENAME	DNAME
-----	-----
KING	ACCOUNTING
BLAKE	ACCOUNTING
...	
KING	RESEARCH
BLAKE	RESEARCH
...	
56 rows selected.	

ИТОГИ

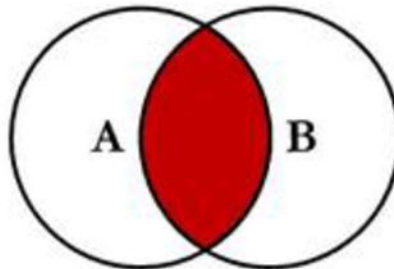
SQL JOINS



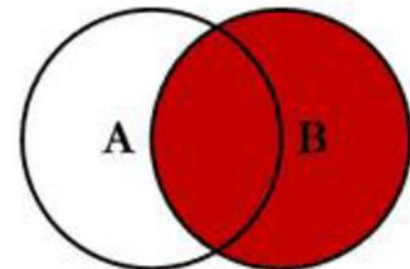
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



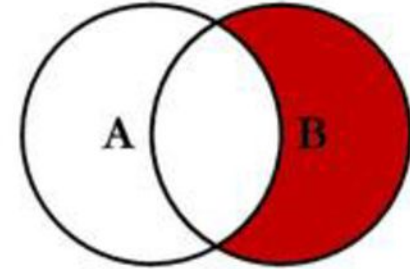
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



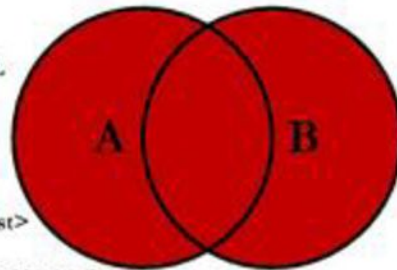
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



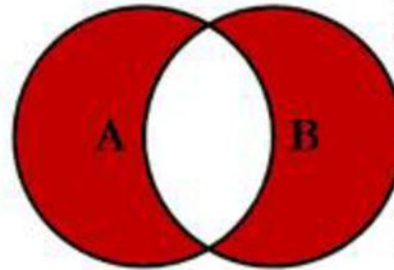
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
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
```

Синтаксис Oracle, для (Left / Right) Join

```
SELECT e.empno, e.ename, e.mgr,  
m.ename  
FROM Emp e  
left join emp m on  
e.mgr=m.empno;
```

```
SELECT e.empno, e.ename, e.mgr,  
m.ename  
FROM Emp e, emp m  
Where e.mgr=m.empno(+);
```

```
SELECT e.empno, e.ename, e.mgr,  
m.ename  
FROM Emp e  
RIGHT join emp m on  
e.mgr=m.empno;
```

```
SELECT e.empno, e.ename, e.mgr,  
m.ename  
FROM Emp e, emp m  
Where e.mgr(+)=m.empno;
```

