

Бази даних та Інформаційні системи

Тема 13. Вибірка даних з кількох таблиць

СумДУ, каф. КН,
2020

Мета заняття

Після завершення заняття ви повинні вміти і знати наступне:

- Як отримати дані з декількох таблиць, використовуючи різні типи з'єднань.
- Писати запити **SELECT** для доступу до даних декількох таблиць з використанням **еквіз'єднання** і **не-еквіз'єднання**.
- Переглядати дані, які не задовольняють вимогам з'єднань, за допомогою **зовнішнього з'єднання**
- З'єднувати таблицю **саму з собою**.




Отримання даних з кількох таблиць

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.		

Що таке з'єднання?

- ▶ Використовуйте з'єднання, щоб отримати дані більш ніж з однієї таблиці.

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

- ▶ Умови з'єднання записуються у фразі **WHERE**.
- ▶ Якщо однакове ім'я стовпця зустрічається в декількох таблицях, то слід використовувати **префікси** у вигляді імен таблиць.
(Наприклад **EMP.DEPNO**, **DEPT.DEPNO** замість **DEPNO**).



Декартів добуток

- Декартів добуток формується, якщо:
 - Умова з'єднання не вказана;
 - Умова з'єднання некоректна;
 - В результаті з'єднання всі рядки першої таблиці; з'єднуються з усіма рядками другої таблиці.
- Щоб уникнути декартового добутка, завжди включайте коректну умову з'єднання в реченні **WHERE**.



Формування декартового добутку

EMP (14 рядків)

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

DEPT (4 рядки)

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

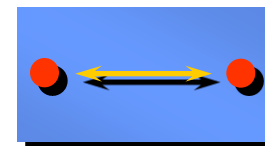
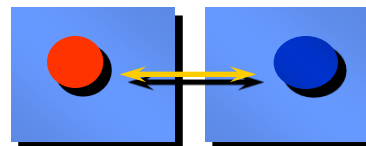
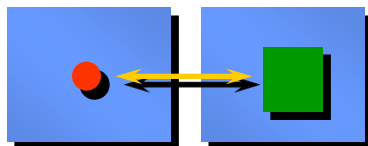
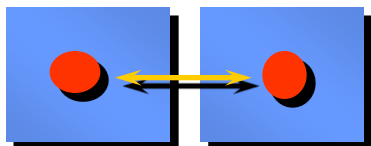
“Декартів
добуток:
 $14 \times 4 = 56$ рядків”

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

Види з'єднань

Не-екві

Само



Екві(внутрішнє)

Зовнішнє



Що таке еквіз'єднання?

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.		

Первинний

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.		

Зовнішній

КЛЮЧ

Отримання записів з використанням еквіз'єднання

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

EMPNO	ENAME	EDEPTNO	DDEPTNO	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.

Вирішення проблеми неоднозначності імен стовпців

- Використовуйте префікс у вигляді імені таблиці в тому випадку, якщо кілька таблиць мають однойменні стовпці.
- Розрізняйте однойменні стовпці різних таблиць за допомогою введення псевдонімів.



Додаткові умови пошуку з використанням оператора AND

EMP	EMPNO	ENAME	DEPTNO		DEPTNO	DNAME	LOC	DEPT
	7839	KING	10		10	ACCOUNTING	NEW YORK	
	7698	BLAKE	30		30	SALES	CHICAGO	
	7782	CLARK	10		10	ACCOUNTING	NEW YORK	
	7566	JONES	20		20	RESEARCH	DALLAS	
	7654	MARTIN	30		30	SALES	CHICAGO	
	7499	ALLEN	30		30	SALES	CHICAGO	
	7844	TURNER	30		30	SALES	CHICAGO	
	7900	JAMES	30		30	SALES	CHICAGO	
	7521	WARD	30		30	SALES	CHICAGO	
	7902	FORD	20		20	RESEARCH	DALLAS	
	7369	SMITH	20		20	RESEARCH	DALLAS	
...					...			
14 rows selected.					14 rows selected.			

```
SQL> SELECT empno, ename, dname, loc
2 FROM emp, dept
3 WHERE emp.deptno = dept.deptno AND ename = 'KING';
```

Використання псевдонімів таблиць

Спростуйте запити за допомогою псевдонімів таблиць.

```
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;
```



З'єднання більше двох таблиць

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.	

ORDER

CUSTID	ORDID
-----	-----
101	610
102	611
104	612
106	601
102	602
106	
106	
...	...
21 rows selected.	

ITEM

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

```
SELECT c.name, i.itemid
FROM customer c, order o, item i
WHERE c.custid = o.custid
AND o.ordid = i.ordid;
```

Не-еквів'єднання

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 таблиці EMP знаходиться в діапазоні між значеннями в таблиці SALGRADE”

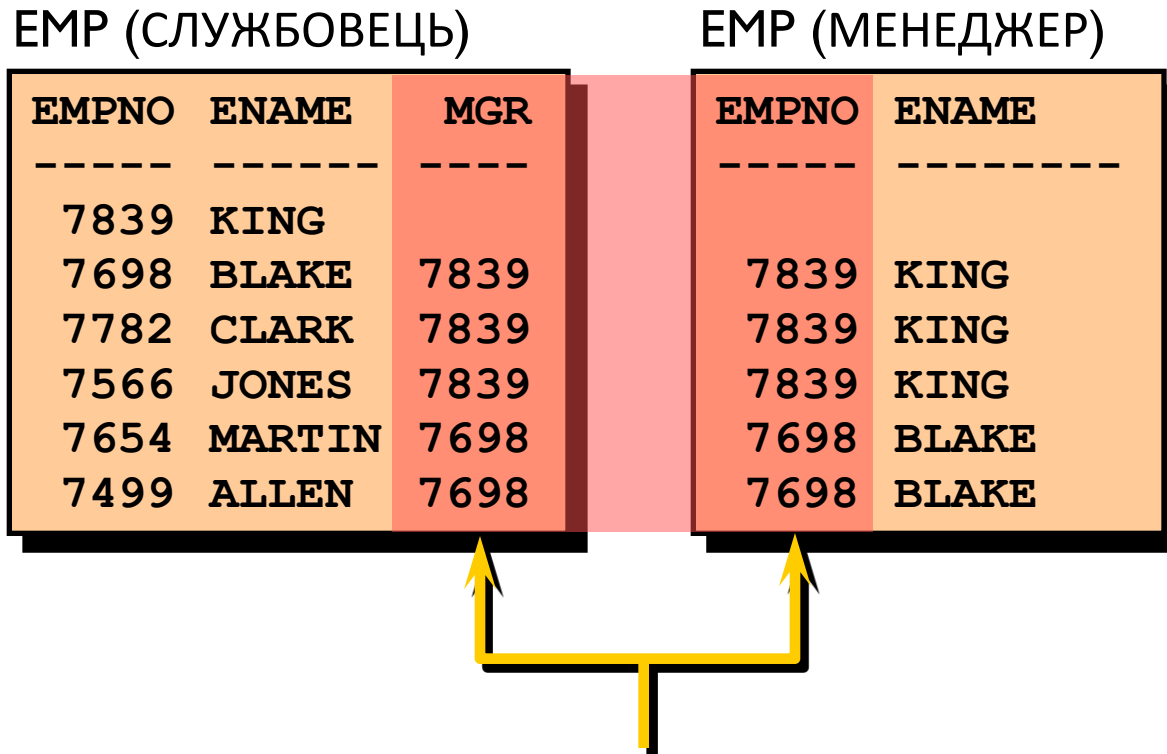
Отримання записів з використанням порівняльних з'єднань (не-еквівіалентів)

```
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.

Само-з'єднання



MGR в таблиці СЛУЖБОВЕЦЬ співпадає
з **EMPNO** в таблиці МЕНЕДЖЕР

Само-з'єднання

```
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.
```



Типи з'єднань

- ▶ Стандарт **SQL: 1999** включає такі типи з'єднань :
 - ▶ **Natural join:**
 - ▶ Оператор **NATURAL JOIN**
 - ▶ Оператор **USING**
 - ▶ Оператор **ON**
 - ▶ **OUTER join:**
 - ▶ **LEFT OUTER JOIN**
 - ▶ **RIGHT OUTER JOIN**
 - ▶ **FULL OUTER JOIN**
 - ▶ **CROSS join**



З'єднання таблиць (Синтаксис SQL:1999)

- ▶ Використовуйте оператор з'єднання **JOIN** щоб з'єднати дві або більше таблиці:

```
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** – з'єднання таблиць на основі стовпців, які мають однакові імена.
- В з'єднання потрапляють всі рядки, які мають однакове значення в стовпцях з однаковими іменами .
- Якщо стовпці мають однакову назву, але різні типи то запит поверне помилку.



Natural Join

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

ORA-25155: column used in NATURAL JOIN cannot have qualifier

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.		

Join з використанням USING

- ▶ **USING** дозволяє вказати за якими стовпцями буде виконано еквіз'єднання, якщо типи стовпців різні.
- ▶ **USING** дозволяє вказати по якому стовпцю виконати з'єднання, якщо підходять кілька стовпців, що мають однакові імена.
- ▶ У фразі **USING** не потрібно вказувати до якої з таблиць належить стовпець.
- ▶ **NATURAL JOIN** і **USING** взаємно виключають один одного.



Приклад з USING

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

ORA-25154: column part of USING
clause cannot have qualifier

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

Join з використанням фрази ON

- ▶ Фраза **ON** використовується для еквіз'єднань і не-еквіз'єднань.
- ▶ Фраза **ON** відокремлює умову з'єднання від інших умов.
- ▶ Фраза **ON** робить запит простіше для розуміння.



Приклад з ON

```
SELECT empno, ename,  
       loc, e.deptno  
FROM   emp e
```

Обов'язково!!! Інакше
**ORA-00918:column ambiguosly
defined**

```
JOIN dept d ON (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
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS

Порівняйте запити:

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

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

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



З'єднання кількох таблиць з ON

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



Само-з'єднання з 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	7698	BLAKE
7654	MARTIN	7698	7698	BLAKE
7499	ALLEN	7698		

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

Додаткові умови в Join

- Для накладання додаткових обмежень можна використовувати речення **WHERE** або **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 = 8000 );
```

Або

```
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 = 8000;
```


Не-екві з'єднання з 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 таблиці EMP
знаходиться в діапазоні
значень в таблиці SALGRADE”

OUTER Join (зовнішнє з'єднання)

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



У департаменті OPERATIONS
співробітників немає

INNER Vs OUTER Joins

- У SQL: 1999 з'єднання, яке повертає рядки, якщо значення присутні в обох таблицях, називається **INNER join**.
- З'єднання, яке повертає результати **INNER join** і рядки з лівої (правої) таблиці, для яких не знайшлося збігів, називається **left (right) OUTER join**.
- З'єднання, яке повертає результати **INNER join** і рядки з лівої і правої таблиці, для яких немає збігів, називається **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.ename, e.deptno, d.deptno  
FROM   emp e  
LEFT OUTER JOIN dept d  
ON     (e.deptno = d.deptno) ;
```

▶ FULL OUTER JOIN

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



Декартів добуток

EMP (14 рядків)

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

DEPT (4 рядка)

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

“Декартів
добуток:
 $14 \times 4 = 56$ рядків”

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

Cross Joins

- **CROSS JOIN** повертає декартів добуток двох таблиць.

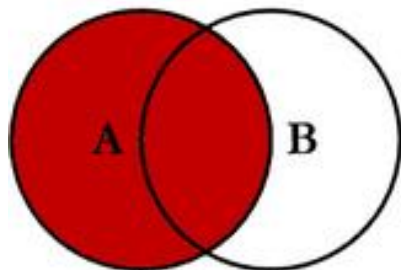
```
SELECT ename, dname  
FROM emp  
CROSS JOIN dept;
```

ENAME	DNAME
-----	-----
KING	ACCOUNTING
BLAKE	ACCOUNTING
...	
KING	RESEARCH
BLAKE	RESEARCH
...	

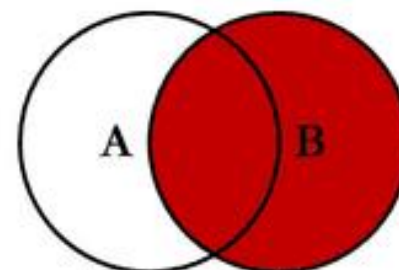
56 rows selected.

Підсумки

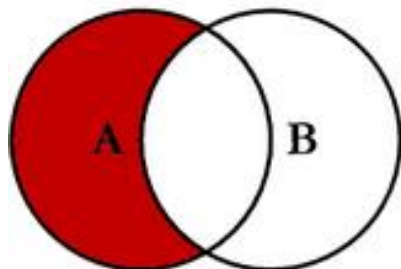
SQL JOINS



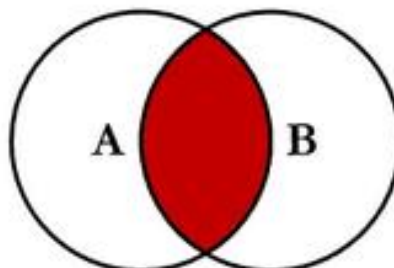
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



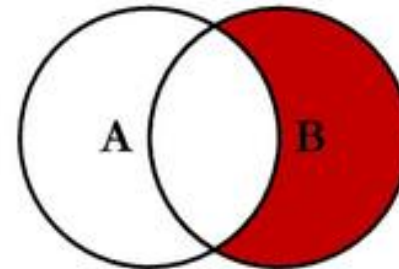
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



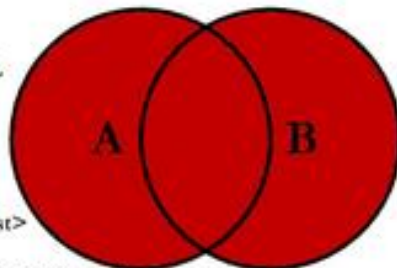
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



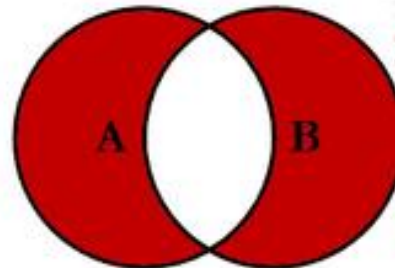
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



```
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
```

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;
```

