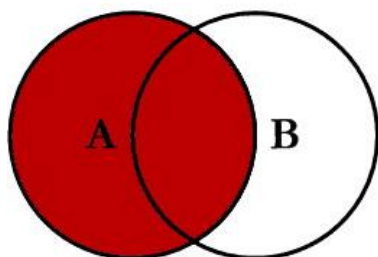


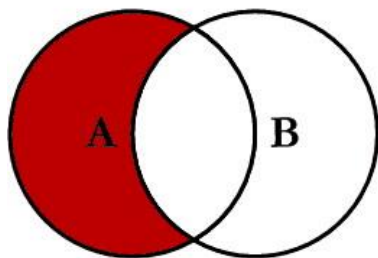
# Объяснение SQL объединений JOIN: LEFT/RIGHT/INNER/OUTER

<https://www.codeproject.com/articles/33052/visual-representation-of-sql-joins>

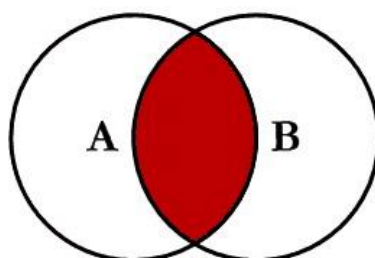
## SQL JOINS



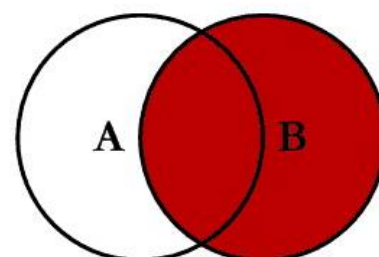
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



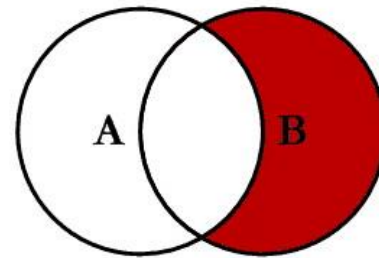
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



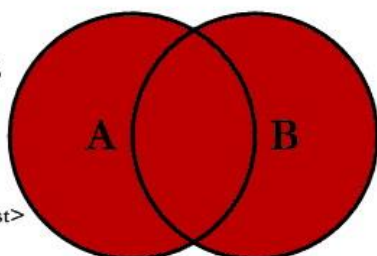
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



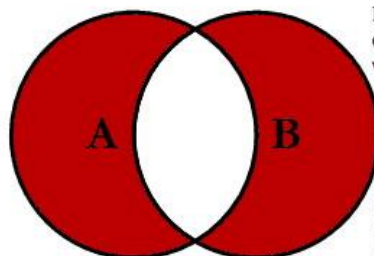
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
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
```

© C.L. Moffatt, 2008

Разберем пример. Имеем две таблицы: пользователи и отделы.

U) users			D) departments		
id	name	d_id	id	name	
--	----	----	--	----	
1	Владимир	1	1	Сейлз	
2	Антон	2	2	Поддержка	
3	Александр	6	3	Финансы	
4	Борис	2	4	Логистика	
5	Юрий	4			

```
SELECT u.id, u.name, d.name AS d_name
FROM users u
INNER JOIN departments d ON u.d_id = d.id
```

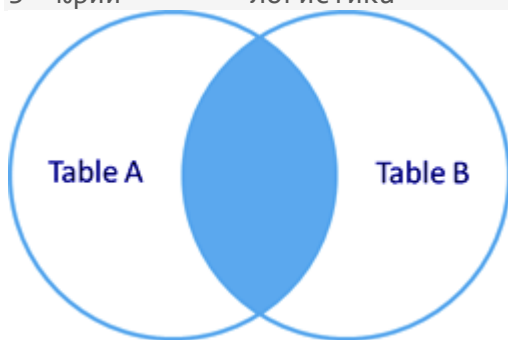
Запрос вернет объединенные данные, которые пересекаются по условию, указанному в INNER JOIN ON <..>.

В нашем случае условие <таблица\_пользователей>.<идентификатор\_отдела> должен совпадать с <таблица\_отделов>.<идентификатор>

В результате *отсутствуют*:

- пользователь Александр (отдел 6 - не существует)
- отдел Финансы (нет пользователей)

id	name	d_name
--	-----	-----
1	Владимир	Сейлз
2	Антон	Поддержка
4	Борис	Поддержка
3	Юрий	Логистика



Внутреннее объединение INNER JOIN (синоним JOIN, ключевое слово INNER можно опустить).

Выбираются только совпадающие данные из объединяемых таблиц.

Чтобы получить данные, которые подходят по условию частично, необходимо использовать внешнее объединение - OUTER JOIN.

Такое объединение вернет данные из обеих таблиц (совпадающие по условию объединения) ПЛЮС дополнит выборку оставшимися данными из внешней таблицы, которые по условию не подходят, заполнив недостающие данные значением NULL.

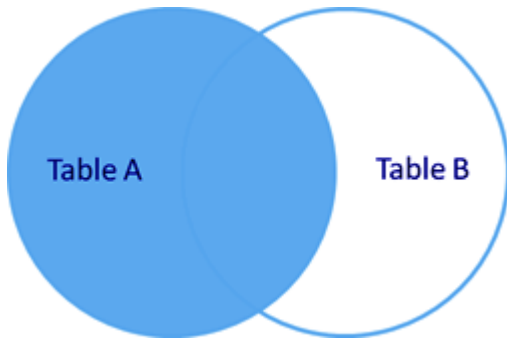


рис. Left join

Существует два типа внешнего объединения OUTER JOIN - LEFT OUTER JOIN и RIGHT OUTER JOIN.

Работают они одинаково, разница заключается в том что LEFT - указывает что "внешней" таблицей будет находящаяся слева (в нашем примере это таблица users). Ключевое слово OUTER можно опустить. Запись LEFT JOIN идентична LEFT OUTER JOIN.

```
SELECT u.id, u.name, d.name AS d_name
FROM users u
LEFT OUTER JOIN departments d ON u.d_id = d.id
```

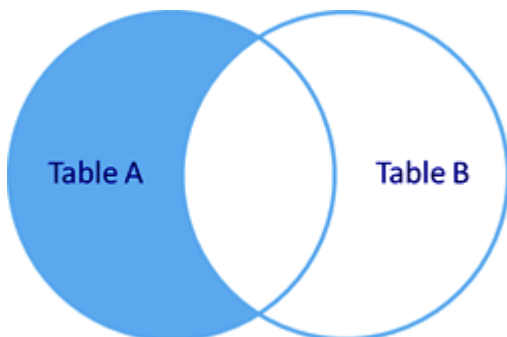
Получаем полный список пользователей и сопоставленные департаменты.

id	name	d_name
--	-----	-----
1	Владимир	Сейлз
2	Антон	Поддержка
3	Александр	NULL
4	Борис	Поддержка
5	Юрий	Логистика

Добавив условие

```
WHERE d.id IS NULL
```

в выборке останется только 3#Александр, так как у него не назначен департамент.



RIGHT OUTER JOIN вернет полный список департаментов (правая таблица) и сопоставленных пользователей.

```
SELECT u.id, u.name, d.name AS d_name
FROM users u
RIGHT OUTER JOIN departments d ON u.d_id = d.id
```

id	name	d_name
--	-----	-----
1	Владимир	Сейлз
2	Антон	Поддержка
4	Борис	Поддержка
NULL	NULL	Финансы
5	Юрий	Логистика

Дополнительно можно отфильтровать данные, проверяя их на NULL.

```
SELECT d.id, d.name
FROM users u
RIGHT OUTER JOIN departments d ON u.d_id = d.id
WHERE u.id IS null
```

В нашем примере указав WHERE u.id IS null, мы выберем департаменты, в которых не числятся пользователи. (3#Финансы)