

# ЗАПРОСЫ К НЕСКОЛЬКИМ ТАБЛИЦАМ

## CROSS JOIN – Перекрестное соединение

**FROM** Таблица A **CROSS JOIN** Таблица B

- **CROSS JOIN** - возвращает декартово произведение таблиц
  - Может быть заменено списком таблиц через запятую в разделе FROM

```
SELECT f.FName, l.LName
FROM "LName" AS l, "FName" as f;
```

id	LName
1	Петров
2	Иванов
3	Сергеев
4	Алексеев



id	FName
1	Алексей
2	Илья
3	Олег



FName	LName
1	Алексей Петров
2	Алексей Иванов
3	Алексей Сергеев
4	Алексей Алексеев
5	Илья Петров
6	Илья Иванов
7	Илья Сергеев
8	Илья Алексеев
9	Олег Петров
10	Олег Иванов
11	Олег Сергеев
12	Олег Алексеев

## Предложение FROM

- В предложении **FROM** перечисляются один или несколько источников данных, используемых для формирования результирующей выборки
- При использовании нескольких источников необходимо указать тип их соединения:
  - CROSS JOIN
  - [ INNER ] JOIN
  - LEFT [ OUTER ] JOIN
  - RIGHT [ OUTER ] JOIN
  - FULL [ OUTER ] JOIN

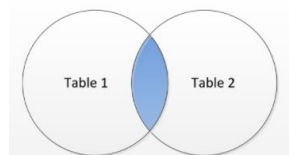
## Соединение посредством предиката

**FROM** Таблица A [Тип соединения] **JOIN** Таблица B **ON** Предикат

Тип соединения	Описание
<b>INNER</b>	Используется для соединения строк из обеих таблиц на основе заданного предиката. Используется по умолчанию, когда тип соединения явно не задан
<b>LEFT [OUTER]</b>	«Левое (внешнее)». Включает в себя все строки из левой таблицы A и те строки из правой таблицы B, для которых выполняется условие предиката. Для строк из таблицы A, для которых не найдено соответствия в таблице B, в столбцы, извлекаемые из таблицы B, заносятся значения NULL
<b>RIGHT [OUTER]</b>	«Правое (внешнее)». Включает в себя все строки из правой таблицы B и те строки из левой таблицы A, для которых выполняется условие предиката. Для строк из таблицы B, для которых не найдено соответствия в таблице A, в столбцы, извлекаемые из таблицы A заносятся значения NULL
<b>FULL [OUTER]</b>	«Полное (внешнее)». Это комбинация левого и правого соединений. В полное соединение включаются все строки из обеих таблиц. Для совпадающих строк поля заполняются реальными значениями, для несовпадающих строк поля заполняются в соответствии с правилами левого и правого соединений

Вместо **ON** можно использовать **USING** при условии совпадения названий столбцов используемых для соединения

# INNER JOIN



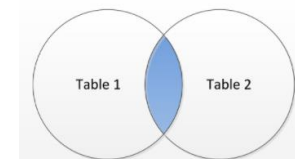
```
SELECT *
FROM "LName" AS l INNER JOIN "FName" as f
ON l.id = f.id;
```

id	LName	id	FName
1	Петров	1	Алексей
2	Иванов	2	Илья
3	Сергеев	3	Олег
4	Алексеев		

```
SELECT *
FROM "LName" AS l INNER JOIN "FName" as f
ON l.id != f.id;
```

id	LName	id	FName
1	Петров	1	Алексей
2	Иванов	2	Илья
3	Сергеев	3	Олег
4	Алексеев	1	Алексей
5	Петров	2	Илья
6	Сергеев	3	Олег
7	Алексеев	2	Илья
8	Петров	3	Олег
9	Иванов	3	Олег
10	Алексеев	3	Олег

# INNER JOIN

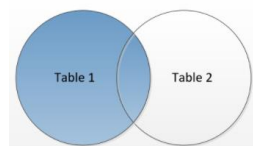


```
SELECT *
FROM "HR"."Employees" AS E
JOIN "Sales"."Orders" AS O
ON E.empid = O.empid;
```

empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	Davis	Sara	Seattle	10 248	85	5	2006-07-04 00:00:00
2	Funk	Don	Tacoma	10 249	79	6	2006-07-05 00:00:00
3	Lew	Judy	Kirkland	10 254	14	5	2006-07-11 00:00:00
4	Peled	Yael	Redmond	10 258	20	1	2006-07-17 00:00:00
				10 264	24	6	2006-07-24 00:00:00
				10 265	7	2	2006-07-25 00:00:00
				10 269	89	5	2006-07-31 00:00:00
				10 270	87	1	2006-08-01 00:00:00
				10 271	75	6	2006-08-01 00:00:00

empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	Davis	Sara	Seattle	10 258	20	1	2006-07-17 00:00:00
2	Davis	Sara	Seattle	10 270	87	1	2006-08-01 00:00:00
3	Funk	Don	Tacoma	10 265	7	2	2006-07-25 00:00:00

# LEFT [OUTER] JOIN

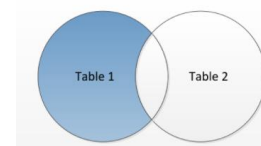


```
SELECT *
FROM "HR"."Employees" AS E
LEFT JOIN "Sales"."Orders" AS O
ON E.empid = O.empid;
```

empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	Davis	Sara	Seattle	10 248	85	5	2006-07-04 00:00:00
2	Funk	Don	Tacoma	10 249	79	6	2006-07-05 00:00:00
3	Lew	Judy	Kirkland	10 254	14	5	2006-07-11 00:00:00
4	Peled	Yael	Redmond	10 258	20	1	2006-07-17 00:00:00
				10 264	24	6	2006-07-24 00:00:00
				10 265	7	2	2006-07-25 00:00:00
				10 269	89	5	2006-07-31 00:00:00
				10 270	87	1	2006-08-01 00:00:00
				10 271	75	6	2006-08-01 00:00:00

empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	Davis	Sara	Seattle	10 258	20	1	2006-07-17 00:00:00
2	Davis	Sara	Seattle	10 270	87	1	2006-08-01 00:00:00
3	Funk	Don	Tacoma	10 265	7	2	2006-07-25 00:00:00
4	Lew	Judy	Kirkland	[NULL]	[NULL]	[NULL]	[NULL]
5	Peled	Yael	Redmond	[NULL]	[NULL]	[NULL]	[NULL]

# LEFT [OUTER] JOIN с исключением



```
SELECT *
FROM "HR"."Employees" AS E
LEFT JOIN "Sales"."Orders" AS O
ON E.empid = O.empid
WHERE O.empid IS NULL;
```

empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	Davis	Sara	Seattle	10 248	85	5	2006-07-04 00:00:00
2	Funk	Don	Tacoma	10 249	79	6	2006-07-05 00:00:00
3	Lew	Judy	Kirkland	10 254	14	5	2006-07-11 00:00:00
4	Peled	Yael	Redmond	10 258	20	1	2006-07-17 00:00:00
				10 264	24	6	2006-07-24 00:00:00
				10 265	7	2	2006-07-25 00:00:00
				10 269	89	5	2006-07-31 00:00:00
				10 270	87	1	2006-08-01 00:00:00
				10 271	75	6	2006-08-01 00:00:00

empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	3	Lew	Judy	Kirkland	[NULL]	[NULL]	[NULL]
2	4	Peled	Yael	Redmond	[NULL]	[NULL]	[NULL]

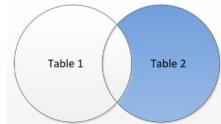
## RIGHT [OUTER] JOIN



```
SELECT *
FROM "HR"."Employees" AS E
RIGHT JOIN "Sales"."Orders" AS O
ON E.empid = O.empid;
```

	empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	1	Davis	Sara	Seattle	10 248	85	5	2006-07-04 00:00:00
2	2	Funk	Don	Tacoma	10 249	79	6	2006-07-05 00:00:00
3	3	Lew	Judy	Kirkland	10 254	14	5	2006-07-11 00:00:00
4	4	Peled	Yael	Redmond	10 258	20	1	2006-07-17 00:00:00
5	[NULL]	[NULL]	[NULL]	[NULL]	10 264	24	6	2006-07-24 00:00:00
6	[NULL]	[NULL]	[NULL]	[NULL]	10 265	7	2	2006-07-25 00:00:00
7	1	Davis	Sara	Seattle	10 269	89	5	2006-07-31 00:00:00
8	1	Davis	Sara	Seattle	10 270	87	1	2006-08-01 00:00:00
9	[NULL]	[NULL]	[NULL]	[NULL]	10 271	75	6	2006-08-01 00:00:00

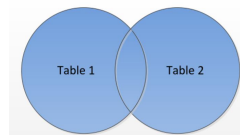
## RIGHT [OUTER] JOIN с исключением



```
SELECT *
FROM "HR"."Employees" AS E
RIGHT JOIN "Sales"."Orders" AS O
ON E.empid = O.empid
WHERE E.empid IS NULL;
```

	empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	[NULL]	[NULL]	[NULL]	[NULL]	10 248	85	5	2006-07-04 00:00:00
2	[NULL]	[NULL]	[NULL]	[NULL]	10 249	79	6	2006-07-05 00:00:00
3	[NULL]	[NULL]	[NULL]	[NULL]	10 254	14	5	2006-07-11 00:00:00
4	[NULL]	[NULL]	[NULL]	[NULL]	10 264	24	6	2006-07-24 00:00:00
5	[NULL]	[NULL]	[NULL]	[NULL]	10 269	89	5	2006-07-31 00:00:00
6	[NULL]	[NULL]	[NULL]	[NULL]	10 270	87	1	2006-08-01 00:00:00
7	[NULL]	[NULL]	[NULL]	[NULL]	10 271	75	6	2006-08-01 00:00:00

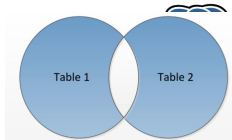
## FULL [OUTER] JOIN



```
SELECT *
FROM "HR"."Employees" AS E FULL
JOIN "Sales"."Orders" AS O
ON E.empid = O.empid;
```

	empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	1	Davis	Sara	Seattle	10 258	20	1	2006-07-17 00:00:00
2	1	Davis	Sara	Seattle	10 270	87	1	2006-08-01 00:00:00
3	2	Funk	Don	Tacoma	10 265	7	2	2006-07-25 00:00:00
4	3	Lew	Judy	Kirkland	[NULL]	[NULL]	[NULL]	[NULL]
5	4	Peled	Yael	Redmond	[NULL]	[NULL]	[NULL]	[NULL]
6	[NULL]	[NULL]	[NULL]	[NULL]	10 248	85	5	2006-07-04 00:00:00
7	[NULL]	[NULL]	[NULL]	[NULL]	10 249	79	6	2006-07-05 00:00:00
8	[NULL]	[NULL]	[NULL]	[NULL]	10 254	14	5	2006-07-11 00:00:00
9	[NULL]	[NULL]	[NULL]	[NULL]	10 264	24	6	2006-07-24 00:00:00
10	[NULL]	[NULL]	[NULL]	[NULL]	10 269	89	5	2006-07-31 00:00:00
11	[NULL]	[NULL]	[NULL]	[NULL]	10 271	75	6	2006-08-01 00:00:00

## FULL [OUTER] JOIN с исключением



```
SELECT *
FROM "HR"."Employees" AS E
FULL JOIN "Sales"."Orders" AS O
ON E.empid = O.empid
WHERE E.empid is null OR O.empid IS NULL;
```

	empid	lastname	firstname	city	orderid	custid	empid	orderdate
1	3	Lew	Judy	Kirkland	[NULL]	[NULL]	[NULL]	[NULL]
2	4	Peled	Yael	Redmond	[NULL]	[NULL]	[NULL]	[NULL]
3	[NULL]	[NULL]	[NULL]	[NULL]	10 248	85	5	2006-07-04 00:00:00
4	[NULL]	[NULL]	[NULL]	[NULL]	10 249	79	6	2006-07-05 00:00:00
5	[NULL]	[NULL]	[NULL]	[NULL]	10 254	14	5	2006-07-11 00:00:00
6	[NULL]	[NULL]	[NULL]	[NULL]	10 264	24	6	2006-07-24 00:00:00
7	[NULL]	[NULL]	[NULL]	[NULL]	10 269	89	5	2006-07-31 00:00:00
8	[NULL]	[NULL]	[NULL]	[NULL]	10 271	75	6	2006-08-01 00:00:00

## Self Join

- Для объединения записей внутри одной таблицы необходимо использовать объединение таблицы со своей «копией»
- Для создания «копии» – используются псевдонимы таблиц



empid
lastname
firstname
title
titleofcourtesy
birthdate
hiredate
address
city
region
postalcode
country
phone
mgrid

```
SELECT Managers.lastname AS Boss,
       Managers.title AS BossTitle,
       Empl.lastname AS Employee,
       Empl.title AS EmployeeTitle
FROM "HR"."Employees" AS Managers
JOIN "HR"."Employees" AS Empl
ON Managers.empid = Empl.mgrid;
```

	Boss	BossTitle	Employee	EmployeeTitle
1	Davis	CEO	Funk	Vice President, Sales
2	Funk	Vice President, Sales	Lew	Sales Manager
3	Lew	Sales Manager	Peled	Sales Representative
4	Funk	Vice President, Sales	Buck	Sales Manager
5	Buck	Sales Manager	Suurs	Sales Representative
6	Buck	Sales Manager	King	Sales Representative
7	Lew	Sales Manager	Cameron	Sales Representative
8	Buck	Sales Manager	Dolgopyatova	Sales Representative
9	Funk	Vice President, Sales	Johnson	Sales Manager%

## Требования к наборам записей

- Оба набора должны иметь одинаковое количество столбцов, совместимых по типу данных
- ORDER BY не допускается во входных запросах, но может использоваться для сортировки результирующего набора
- NULL значения считаются равными при сравнении наборов

```
<SELECT query_1>
<SET_OPERATOR> {UNION | UNION ALL | INTERSECT | EXCEPT}
<SELECT query_2>
[ORDER BY <sort_list>]
```

## Использование операторов наборов записей

### UNION и UNION ALL

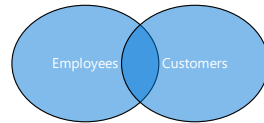


- UNION возвращает результирующий набор уникальных строк, объединенных из двух входных наборов
  - Дубликаты удаляются при обработке запроса (влияет на производительность)

```
SELECT country, region, city FROM "HR"."Employees"
UNION --72 записи
SELECT country, region, city FROM "Sales"."Customers";
```
- UNION ALL возвращает результирующий набор со всеми строками из двух входных наборов.
  - Чтобы избежать потери производительности, вызванной фильтрацией дубликатов, используйте UNION ALL вместо UNION всякий раз, когда это позволяют требования

```
SELECT country, region, city FROM "HR"."Employees"
UNION ALL --101 запись
SELECT country, region, city FROM "Sales"."Customers";
```

# INTERSECT

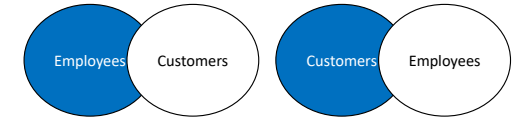


- INTERSECT возвращает уникальный результирующий набор строк, которые присутствуют в обоих входных наборах

```
SELECT country, region, city FROM "HR"."Employees"  
INTERSECT --3 записи  
SELECT country, region, city FROM "Sales"."Customers";
```

	country	region	city
1	UK	[NULL]	London
2	USA	WA	Kirkland
3	USA	WA	Seattle

# EXCEPT (MINUS)



- EXCEPT возвращает только уникальные строки, которые присутствуют в левом наборе, но нет в правом.
  - Порядок, в котором указаны наборы, имеет значение

```
SELECT country, region, city FROM "HR"."Employees"  
EXCEPT --3 записи  
SELECT country, region, city FROM "Sales"."Customers";
```

	country	region	city
1	Slovenia	[NULL]	Ljubljana
2	USA	WA	Redmond
3	USA	WA	Tacoma

```
SELECT country, region, city FROM "Sales"."Customers"  
EXCEPT --66 записей  
SELECT country, region, city FROM "HR"."Employees";
```

	country	region	city
1	Argentina	[NULL]	Buenos Aires
2	Austria	[NULL]	Graz
3	Austria	[NULL]	Salzburg
4	Belgium	[NULL]	Bruxelles
...			
63	USA	WY	Lander
64	Venezuela	DF	Caracas
65	Venezuela	Lara	Barquisimeto