**Соединение JOIN**

***CROSS JOIN***

Оператор перекрёстного соединения, или декартова произведения CROSS JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является [симметричным](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BE%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5).

Заголовок таблицы-результата является объединением ([конкатенацией](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%BA%D0%B0%D1%82%D0%B5%D0%BD%D0%B0%D1%86%D0%B8%D1%8F)) заголовков соединяемых таблиц.

Тело результата логически формируется следующим образом. Каждая строка одной таблицы соединяется с каждой строкой второй таблицы, давая тем самым в результате все возможные сочетания строк двух таблиц.

SELECT

e1.empid, e1.lastname,

e2.empid, e2.lastname

FROM HR.Employees e1, HR.Employees e2

Или

SELECT

e1.empid, e1.lastname,

e2.empid, e2.lastname

FROM HR.Employees e1 CROSS JOIN HR.Employees e2

Вывести информацию о сотруднике Lew (lastname)

**INNER JOIN**

**Оператор внутреннего соединения INNER JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является** [**симметричным**](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BE%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5)**.**

*Заголовок* таблицы-результата является объединением ([конкатенацией](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%BA%D0%B0%D1%82%D0%B5%D0%BD%D0%B0%D1%86%D0%B8%D1%8F)) заголовков соединяемых таблиц.

*Тело* результата логически формируется следующим образом. Каждая строка одной таблицы сопоставляется с каждой строкой второй таблицы, после чего для полученной «соединённой» строки проверяется условие соединения (вычисляется предикат соединения). Если условие истинно, в таблицу-результат добавляется соответствующая «соединённая» строка.

**TableA TableB**

id name id name

-- ---- -- ----

1 Pirate 1 Rutabaga

2 Monkey 2 Pirate

3 Ninja 3 Darth Vader

4 Spaghetti 5 Ninja

Рассмотрим классический выбор данных которые присутствуют в обеих таблицах

|  |  |
| --- | --- |
| SELECT \* FROM TableA  **INNER** JOIN TableB  ON TableA.name = TableB.name  id name id name  -- ---- -- ----  1 Pirate 2 Pirate  3 Ninja 4 Ninja |  |

***Внутреннее соединение INNER JOIN производит выборку только записей, которые соответствуют как в таблице А, так и в таблице В.***

Написать запросы к базе данных TSQLFundamentals2008:

* Выбрать информацию о сотруднике и его заказах (номер заказа, стоимость доставки)
* Выбрать информацию о сотруднике Sara и ее заказах(номер заказа, стоимость доставки)

***LEFT OUTER JOIN***

Оператор левого внешнего соединения LEFT OUTER JOIN соединяет две таблицы. Порядок таблиц для оператора важен, поскольку оператор не является [симметричным](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BE%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5).

Тело результата логически формируется следующим образом. Пусть выполняется соединение левой и правой таблиц по полю(атрибуту).

* В результат включается внутреннее соединение (INNER JOIN) левой и правой таблиц по полю *p*.
* Затем в результат добавляются те записи левой таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие правой таблице, заполняются значениями *NULL*.

|  |  |
| --- | --- |
| SELECT \* FROM TableA  LEFT OUTER JOIN TableB  ON TableA.name = TableB.name  id name id name  -- ---- -- ----  1 Pirate 2 Pirate  2 Monkey null null  3 Ninja 4 Ninja  4 Spaghetti null null |  |

Левое внешнее объединение LEFT OUTER JOIN производит полный набор записей из таблицы А, с соответствующей записи (если таковые имеются) к Таблице B. Если совпадения нет, то правая часть будет содержать NULL.

Написать запросы к базе данных TSQLFundamentals2008:

* Выбрать имена **всех** заказчиков и их заказы.

**LEFT OUTER JOIN с исключением записей**

|  |  |
| --- | --- |
| SELECT \* FROM TableA  LEFT OUTER JOIN TableB  ON TableA.name = TableB.name  WHERE TableB.id IS null  id name id name  -- ---- -- ----  2 Monkey null null  4 Spaghetti null null |  |

Написать запросы к базе данных TSQLFundamentals2008:

* Выбрать имена **всех** заказчиков не сделавших НИ ОДНОГО ЗАКАЗА
* Посчитать количество заказов у каждого заказчика

***RIGHT OUTER JOIN***

Оператор *правого внешнего соединения* RIGHT OUTER JOIN соединяет две таблицы. Порядок таблиц для оператора важен, поскольку оператор не является [***симметричным***](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BE%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5).

*Заголовок* таблицы-результата является объединением ([конкатенацией](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%BA%D0%B0%D1%82%D0%B5%D0%BD%D0%B0%D1%86%D0%B8%D1%8F)) заголовков соединяемых таблиц.

*Тело* результата логически формируется следующим образом. Пусть выполняется соединение левой и правой таблиц по предикату (условию) *p*.

* В результат включается внутреннее соединение (INNER JOIN) левой и правой таблиц по предикату *p*.
* Затем в результат добавляются те записи правой таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие левой таблице, заполняются значениями *NULL*.

***FULL OUTER JOIN***

Оператор *полного внешнего соединения* FULL OUTER JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является [***симметричным***](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BE%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5)***.***

*Заголовок* таблицы-результата является объединением ([конкатенацией](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%BA%D0%B0%D1%82%D0%B5%D0%BD%D0%B0%D1%86%D0%B8%D1%8F)) заголовков соединяемых таблиц.

*Тело* результата логически формируется следующим образом. Пусть выполняется соединение первой и второй таблиц по предикату (условию) *p*. Слова «первой» и «второй» здесь не обозначают порядок в записи (который неважен), а используются лишь для различения таблиц.

* В результат включается внутреннее соединение (INNER JOIN) первой и второй таблиц по полю *p*.
* В результат добавляются те записи первой таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие второй таблице, заполняются значениями *NULL*.
* В результат добавляются те записи второй таблицы, которые не вошли во внутреннее соединение на шаге 1. Для таких записей поля, соответствующие первой таблице, заполняются значениями *NULL*.

**FULL OUTER JOIN**

|  |  |
| --- | --- |
| SELECT \* FROM TableA  FULL OUTER JOIN TableB  ON TableA.name = TableB.name  id name id name  -- ---- -- ----  1 Pirate 2 Pirate  2 Monkey null null  3 Ninja 4 Ninja  4 Spaghetti null null  null null 1 Rutabaga  null null 3 Darth Vader |  |

Написать запросы к базе данных SellerJoin:

* Выбрать имена **всех** покупателей и имена всех продавцов.

**UNION**

Объединяет результаты двух или более запросов в один результирующий набор, в который входят все строки, принадлежащие всем запросам в объединении. Операция UNION отличается от соединений столбцов из двух таблиц.

Ниже приведены основные правила объединения результирующих наборов двух запросов с помощью операции UNION:

* Количество и порядок столбцов должны быть одинаковыми во всех запросах.
* Типы данных должны быть совместимыми.

***Домащнее задание***

Прочитать Ицик Бен Ган глава 3.

Создать таблицу Person - поля имя, фамилия, дата рождения, пол (мужской или женский), номер паспорта, Создать таблицу Children - поля имя, фамилия, дата рождения, пол.

Таблицы связаны между собой отношением один ко многим.

Несколько персон не имеют детей

* Выбрать имя, фамилия, дата рождения, пол из таблицы Person и всех их детей.
* Написать запрос, который выводит по пять записей каждой персоны.
* Выбрать имя, фамилия, дата рождения, пол персоны и количество их детей.