**Содержание урока**

В запросах SQL могут участвовать несколько таблиц базы данных. При этом необходимо указать как эти таблицы соединены между собой.

Операция соединения **JOIN** предназначена для обеспечения выборки данных из двух таблиц и включения этих данных в один результирующий набор. При необходимости соединения не двух, а нескольких таблиц, операция соединения применяется несколько раз (последовательно).

Операторы соединения входят в раздел **FROM** SQL запросов.

В данном уроке будут созданы запросы:

* [для двух таблиц, внутреннее соединение INNER JOIN](https://stepik.org/lesson/308886/step/2?unit=291012);
* [для двух таблиц, внешние соединения LEFT JOIN и RIGHT JOIN](https://stepik.org/lesson/308886/step/3?unit=291012);
* [для двух таблиц, перекрестное соединение CROSS JOIN](https://stepik.org/lesson/308886/step/4?unit=291012);
* [выборки данных из нескольких таблиц](https://stepik.org/lesson/308886/step/5?unit=291012);
* [выборки данных из нескольких таблиц с группировкой](https://stepik.org/lesson/308886/step/6?unit=291012);
* [выборки данных с применением вложенных запросов](https://stepik.org/lesson/308886/step/7?unit=291012);
* [вложенные запросы в операторах соединения](https://stepik.org/lesson/308886/step/8?unit=291012),
* [операторы соединения, использование USING](https://stepik.org/lesson/308886/step/9?unit=291012).

**Структура и наполнение таблиц**

**Концептуальная схема базы данных:**



**Логическая схема базы данных:**



Таблица**author**([создание](https://stepik.org/lesson/308885/step/6?unit=291011), [заполнение](https://stepik.org/lesson/308885/step/7?unit=291011)):

|  |  |
| --- | --- |
| **author\_id** | **name\_author** |
| 1 | Булгаков М.А. |
| 2 | Достоевский Ф.М. |
| 3 | Есенин С.А. |
| 4 | Пастернак Б.Л. |
| 5 | Лермонтов М.Ю. |

Таблица**genre**([создание](https://stepik.org/lesson/297508/step/6?unit=279268), [заполнение](https://stepik.org/lesson/297508/step/7?unit=279268), рассмотрено в качестве примеров):

|  |  |
| --- | --- |
| **genre\_id** | **name\_genre** |
| 1 | Роман |
| 2 | Поэзия |
| 3 | Приключения |

Таблица **book** ([создание](https://stepik.org/lesson/308885/step/9?unit=291011), [заполнение](https://stepik.org/lesson/308885/step/11?unit=291011)):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **book\_id** | **title** | **author\_id** | **genre\_id** | **price** | **amount** |
| 1 | Мастер и Маргарита | 1 | 1 | 670.99 | 3 |
| 2 | Белая гвардия | 1 | 1 | 540.50 | 5 |
| 3 | Идиот | 2 | 1 | 460.00 | 10 |
| 4 | Братья Карамазовы | 2 | 1 | 799.01 | 3 |
| 5 | Игрок | 2 | 1 | 480.50 | 10 |
| 6 | Стихотворения и поэмы | 3 | 2 | 650.00 | 15 |
| 7 | Черный человек | 3 | 2 | 570.20 | 6 |
| 8 | Лирика | 4 | 2 | 518.99 | 2 |

**Соединение INNER JOIN**

Оператор внутреннего соединения INNER JOIN соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является симметричным.

SELECT

...

FROM

таблица\_1 INNER JOIN таблица\_2

ON условие

...

Результат запроса формируется так:

* каждая строка одной таблицы сопоставляется с каждой строкой второй таблицы;
* для полученной «соединённой» строки проверяется условие соединения;
* если условие истинно, в таблицу результата добавляется соответствующая «соединённая» строка;

**Пример**

Вывести название книг и их авторов.

*Запрос:*

SELECT title, name\_author

FROM

author INNER JOIN book

ON author.author\_id = book.author\_id;

Поскольку поля **author\_id** в таблицах **book** и **author** называются одинаково, необходимо в запросах указывать полную ссылку на них (**book.author\_id** и **author.author\_id**).

*Результат:*

+-----------------------+------------------+

| title | name\_author |

+-----------------------+------------------+

| Мастер и Маргарита | Булгаков М.А. |

| Белая гвардия | Булгаков М.А. |

| Идиот | Достоевский Ф.М. |

| Братья Карамазовы | Достоевский Ф.М. |

| Игрок | Достоевский Ф.М. |

| Стихотворения и поэмы | Есенин С.А. |

| Черный человек | Есенин С.А. |

| Лирика | Пастернак Б.Л. |

+-----------------------+------------------+

В данном запросе осуществляется соединение главной таблицы **author** и зависимой таблицы **book** по ключевому столбцу**author.author\_id** и внешнему ключу **book.author\_id**. При этом в результирующую таблицу запроса включаются все строки, в которых значения этих столбцов совпадают. Другими словами строки зависимой таблицы **book** дополняются фамилией и инициалами авторов из таблицы **author**.

**Задание**

Вывести название, жанр и цену тех книг, количество которых больше 8, в отсортированном по убыванию цены виде.

**Логическая схема базы данных:**



***Текст задания****(чтобы не прокручивать страницу):*

*Вывести название, жанр и цену тех книг, количество которых больше 8, в отсортированном по убыванию цены виде.*

**Результат**

**Структура и наполнение таблиц**

Таблица genre:

+----------+-------------+

| genre\_id | name\_genre |

+----------+-------------+

| 1 | Роман |

| 2 | Поэзия |

| 3 | Приключения |

+----------+-------------+

Таблица author:

+-----------+------------------+

| author\_id | name\_author |

+-----------+------------------+

| 1 | Булгаков М.А. |

| 2 | Достоевский Ф.М. |

| 3 | Есенин С.А. |

| 4 | Пастернак Б.Л. |

| 5 | Лермонтов М.Ю. |

+-----------+------------------+

Таблица book:

+---------+-----------------------+-----------+----------+--------+--------+

| book\_id | title | author\_id | genre\_id | price | amount |

+---------+-----------------------+-----------+----------+--------+--------+

| 1 | Мастер и Маргарита | 1 | 1 | 670.99 | 3 |

| 2 | Белая гвардия | 1 | 1 | 540.50 | 5 |

| 3 | Идиот | 2 | 1 | 460.00 | 10 |

| 4 | Братья Карамазовы | 2 | 1 | 799.01 | 3 |

| 5 | Игрок | 2 | 1 | 480.50 | 10 |

| 6 | Стихотворения и поэмы | 3 | 2 | 650.00 | 15 |

| 7 | Черный человек | 3 | 2 | 570.20 | 6 |

| 8 | Лирика | 4 | 2 | 518.99 | 2 |

+---------+-----------------------+-----------+----------+--------+--------+

Enter an SQL query

Correct answer from **30,244** learners

Total **67%** of tries are correct

 Totally right.

Query result: +-----------------------+------------+--------+ | title | name\_genre | price | +-----------------------+------------+--------+ | Стихотворения и поэмы | Поэзия | 650.00 | | Игрок | Роман | 480.50 | | Идиот | Роман | 460.00 | +-----------------------+------------+--------+ Affected rows: 3

Hide

1

SELECT title, name\_genre, price

2

FROM genre INNER JOIN book

3

   ON genre.genre\_id = book.genre\_id

4

WHERE amount > 8

5

ORDER BY price DESC

6

;

[==================================================================================

**Внешнее соединение LEFT и RIGHT OUTER JOIN**

Оператор внешнего соединения LEFT OUTER JOIN  (можно использовать LEFT JOIN) соединяет две таблицы. Порядок таблиц для оператора важен, поскольку оператор не является симметричным.

SELECT

...

FROM

таблица\_1 LEFT JOIN таблица\_2

ON условие

...

Результат запроса формируется так:

1. в результат включается внутреннее соединение (INNER JOIN) первой и второй таблицы в соответствии с условием;
2. затем в результат добавляются те записи первой таблицы, которые не вошли во внутреннее соединение на шаге 1, для таких записей соответствующие поля второй таблицы заполняются значениями NULL.

Соединение RIGHT JOIN действует аналогично, только в пункте 2 первая таблица меняется на вторую и наоборот.

**Пример**

Вывести название всех книг каждого автора, если книг некоторых авторов в данный момент нет на складе – вместо названия книги указать Null.

*Запрос:*

SELECT name\_author, title

FROM author LEFT JOIN book

ON author.author\_id = book.author\_id

ORDER BY name\_author;

Результат:

+------------------+-----------------------+

| name\_author | title |

+------------------+-----------------------+

| Булгаков М.А. | Мастер и Маргарита |

| Булгаков М.А. | Белая гвардия |

| Достоевский Ф.М. | Игрок |

| Достоевский Ф.М. | Идиот |

| Достоевский Ф.М. | Братья Карамазовы |

| Есенин С.А. | Стихотворения и поэмы |

| Есенин С.А. | Черный человек |

| Лермонтов М.Ю. | NULL |

| Пастернак Б.Л. | Лирика |

+------------------+-----------------------+

Так как в таблице**book** нет книг Лермонтова, напротив этой фамилии стоит Null.

**Задание**

Вывести все жанры, которые не представлены в книгах на складе.

**Логическая схема базы данных:**



**Пояснение**

***Текст задания (чтобы не прокручивать страницу):***

*Вывести все жанры, которые не представлены в книгах на складе.*

**Результат**

**Структура и наполнение таблиц**

Enter an SQL query

Correct answer from **29,370** learners

Total **68%** of tries are correct

 You're right!

Query result: +-------------+ | name\_genre | +-------------+ | Приключения | +-------------+ Affected rows: 1

Hide

1

SELECT name\_genre

2

FROM genre LEFT JOIN book

3

   ON genre.genre\_id = book.genre\_id

4

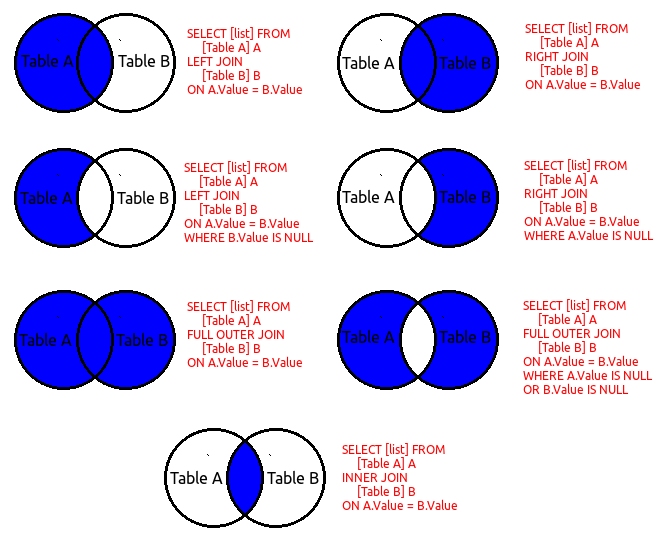
WHERE title IS NULL

5

;

6

​



**Перекрестное соединение CROSS JOIN**

Оператор перекрёстного соединения, или декартова произведения CROSS JOIN (в запросе вместо ключевых слов можно поставить запятую между таблицами) соединяет две таблицы. Порядок таблиц для оператора неважен, поскольку оператор является симметричным. Его структура:

SELECT

...

FROM

таблица\_1 CROSS JOIN таблица\_2

...

или

SELECT

...

FROM

таблица\_1, таблица\_2

...

Результат запроса формируется так: каждая строка одной таблицы соединяется с каждой строкой другой таблицы, формируя  в результате все возможные сочетания строк двух таблиц.

Например, запрос:

SELECT name\_author, name\_genre

FROM

author, genre;

каждому автору из таблицы **author** поставит в соответствие все возможные жанры из таблицы **genre**:

+------------------+-------------+

| name\_author | name\_genre |

+------------------+-------------+

| Булгаков М.А. | Роман |

| Булгаков М.А. | Поэзия |

| Булгаков М.А. | Приключения |

| Достоевский Ф.М. | Роман |

| Достоевский Ф.М. | Поэзия |

| Достоевский Ф.М. | Приключения |

| Есенин С.А. | Роман |

| Есенин С.А. | Поэзия |

| Есенин С.А. | Приключения |

| Пастернак Б.Л. | Роман |

| Пастернак Б.Л. | Поэзия |

| Пастернак Б.Л. | Приключения |

| Лермонтов М.Ю. | Роман |

| Лермонтов М.Ю. | Поэзия |

| Лермонтов М.Ю. | Приключения |

+------------------+-------------+

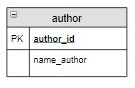
**Задание**

Есть список городов, хранящийся в таблице **city**:

|  |  |
| --- | --- |
| **city\_id** | **name\_city** |
| 1 | Москва |
| 2 | Санкт-Петербург |
| 3 | Владивосток |

Необходимо в каждом городе провести выставку книг каждого автора в течение 2020 года. Дату проведения выставки выбрать случайным образом. Создать запрос, который выведет город, автора и дату проведения выставки. Последний столбец назвать **Дата**. Информацию вывести, отсортировав сначала в алфавитном порядке по названиям городов, а потом по убыванию дат проведения выставок.

**Структура таблицы:**



**Пояснение**

***Текст задания (чтобы не прокручивать страницу):***

*Есть список городов, хранящийся в таблице****city****:*

|  |  |
| --- | --- |
| **city\_id** | **name\_city** |
| 1 | Москва |
| 2 | Санкт-Петербург |
| 3 | Владивосток |

*Необходимо в каждом городе провести выставку книг каждого автора в течение 2020 года. Дату проведения выставки выбрать случайным образом. Создать запрос, который выведет город, автора и дату проведения выставки. Последний столбец назвать****Дата****. Информацию вывести, отсортировав сначала в алфавитном порядке по названиям городов, а потом по убыванию дат проведения выставок.*

**Результат**

**Структура и наполнение таблиц**

Enter an SQL query

1

SELECT name\_city, name\_author,

2

    DATE\_ADD('2020-01-01', INTERVAL FLOOR(RAND() \* 365)DAY) AS Дата

3

FROM city, author

4

ORDER BY name\_city ASC, Дата DESC

5

;

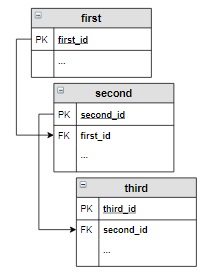
6

​

**Запросы на выборку из нескольких таблиц**

Запрос на выборку может выбирать данные из двух и более таблиц базы данных. При этом таблицы должны быть логически связаны между собой. Для каждой пары таблиц, включаемых в запрос, необходимо указать свой оператор соединения. Наиболее распространенным является внутреннее соединение INNER JOIN, поэтому в примерах будем использовать его.

Пусть таблицы связаны между собой следующим образом:



тогда запрос на выборку для этих таблиц будет иметь вид:

SELECT

...

FROM

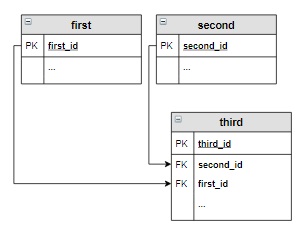
first

INNER JOIN second ON first.first\_id = second.first\_id

INNER JOIN third ON second.second\_id = third.second\_id

...

Если же таблицы связаны так:



то запрос на выборку выглядит следующим образом:

SELECT

...

FROM

first

INNER JOIN third ON first.first\_id = third.first\_id

INNER JOIN second ON third.second\_id = second.second\_id

...

В этом случае рекомендуется соединение таблиц записывать последовательно, «по кругу»: **first → third → second**.

**Пример**

Вывести информацию о тех книгах, их авторах и жанрах, цена которых принадлежит интервалу от 500  до 700 рублей  включительно.

*Запрос:*

SELECT title, name\_author, name\_genre, price, amount

FROM

author

INNER JOIN book ON author.author\_id = book.author\_id

INNER JOIN genre ON genre.genre\_id = book.genre\_id

WHERE price BETWEEN 500 AND 700;

*Результат:*

+-----------------------+----------------+------------+--------+--------+

| title | name\_author | name\_genre | price | amount |

+-----------------------+----------------+------------+--------+--------+

| Мастер и Маргарита | Булгаков М.А. | Роман | 670.99 | 3 |

| Белая гвардия | Булгаков М.А. | Роман | 540.50 | 5 |

| Стихотворения и поэмы | Есенин С.А. | Поэзия | 650.00 | 15 |

| Черный человек | Есенин С.А. | Поэзия | 570.20 | 6 |

| Лирика | Пастернак Б.Л. | Поэзия | 518.99 | 2 |

+-----------------------+----------------+------------+--------+--------+

**Задание**

 Вывести информацию о книгах (жанр, книга, автор), относящихся к жанру, включающему слово «роман» в отсортированном по названиям книг виде.

**Логическая схема базы данных:**



***Текст задания (чтобы не прокручивать страницу):***

*Вывести информацию о книгах (жанр, книга, автор), относящихся к жанру, включающему слово «роман» в отсортированном по названиям книг виде.*

1

SELECT name\_genre, title, name\_author

2

FROM genre

3

       INNER JOIN book ON genre.genre\_id = book.genre\_id

4

       INNER JOIN author ON book.author\_id = author.author\_id

5

WHERE name\_genre = 'роман'

6

ORDER BY title