**Содержание урока**

С помощью запросов корректировки данных решим задачу о занесении в базу книг, привезенных на склад поставщиком.

База данных о книгах  включает три таблицы**genre**, **author** и **book**, информация о поставке занесена в таблицу **supply**. С разными типами книг из поставки необходимо выполнить разные действия:

* для книг, которые уже есть на складе по той же цене, что и в поставке, - увеличить их количество на значение, указанное в поставке ([пример](https://stepik.org/lesson/308887/step/2?unit=291013));
* для книг, которые уже есть на складе, но цена книги в поставке отличается, - увеличить количество экземпляров и вычислить новую цену, при расчете учесть количество имеющихся и новых экземпляров книг ([задание](https://stepik.org/lesson/308887/step/2?unit=291013));
* для книг, которых на складе нет, - проверить, есть ли  автор книги в базе, если нет - [занести фамилию автора](https://stepik.org/lesson/308887/step/3?unit=291013), а потом [добавить новую запись о книге](https://stepik.org/lesson/308887/step/4?unit=291013), оставив поле для описания жанра пустым;
* [задать жанр для новых книг](https://stepik.org/lesson/308887/step/5?unit=291013).

Еще одно типовое действие на складе - удаление устаревшей информации. С помощью запросов корректировки удаление данных о жанрах, авторах и книгах выполняется в зависимости от того, какие свойства внешних ключей были указаны в таблицах при их создании:

* [каскадное удаление записей связанных таблиц](https://stepik.org/lesson/308887/step/6?unit=291013);
* [удаление записей в главной таблице с сохранением записей в зависимой](https://stepik.org/lesson/308887/step/7?unit=291013);
* [удаление записей с использованием информации из связанных таблиц](https://stepik.org/lesson/308887/step/8?unit=291013).

**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 |

Таблица **supply**([создание](https://stepik.org/lesson/305012/step/2?unit=287020), [заполнение](https://stepik.org/lesson/305012/step/3?unit=287020)):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **supply\_id** | **title** | **author** | **price** | **amount** |
| 1 | Доктор Живаго | Пастернак Б.Л. | 380.80 | 4 |
| 2 | Черный человек | Есенин С.А. | 570.20 | 6 |
| 3 | Белая гвардия | Булгаков М.А. | 540.50 | 7 |
| 4 | Идиот | Достоевский Ф.М. | 360.80 | 3 |
| 5 | Стихотворения и поэмы | Лермонтов М.Ю. | 255.90 | 4 |
| 6 | Остров сокровищ | Стивенсон Р.Л. | 599.99 | 5 |

**Запросы на обновление, связанные таблицы**

В запросах на обновление можно использовать связанные таблицы:

UPDATE таблица\_1

... JOIN таблица\_2

ON выражение

...

SET ...

WHERE ...;

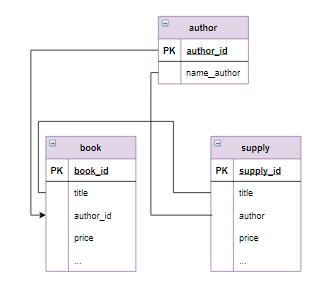
При этом исправлять данные можно во всех используемых в запросе таблицах.

**Пример**

Для книг, которые уже есть на складе (в таблице**book**) по той же цене, что и в поставке (**supply**), увеличить количество на значение, указанное в поставке, а также обнулить количество этих книг в поставке.

Этот запрос должен отобрать строки из таблиц **book**и **supply** такие, что у них совпадают и автор, и название книги. Но в таблице **supply** фамилия автора записана не числом (**id**), а текстом. Следовательно, чтобы выполнить сравнение по фамилии автора нужно "подтянуть" таблицу **author**,  которая связана с **book**по столбцу **author\_id**.  И в логическом выражении, описывающем соединение таблиц, можно будет использовать столбцы из таблиц **book, author**и **supply**.

Если таблицы логически связаны по двум и более столбцам (на рисунке связи обозначены линиями), возможно через другие таблицы, условие соединение будет включать связи по нужным столбцам через логический оператор **AND**. Например, для следующих таблиц логическую связь по названию и автору:



условие соединения можно записать в виде:

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

INNER JOIN supply ON book.title = supply.title

and supply.author = author.name\_author

*Запрос:*

UPDATE book

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

INNER JOIN supply ON book.title = supply.title

and supply.author = author.name\_author

SET book.amount = book.amount + supply.amount,

supply.amount = 0

WHERE book.price = supply.price;

SELECT \* FROM book;

SELECT \* FROM supply;

*Результат:*

Affected rows: 4

Query result:

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

| 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 | 12 |

| 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 | 12 |

| 8 | Лирика | 4 | 2 | 518.99 | 2 |

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

Affected rows: 8

Query result:

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

| supply\_id | title | author | price | amount |

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

| 1 | Доктор Живаго | Пастернак Б.Л. | 380.80 | 4 |

| 2 | Черный человек | Есенин С.А. | 570.20 | 0 |

| 3 | Белая гвардия | Булгаков М.А. | 540.50 | 0 |

| 4 | Идиот | Достоевский Ф.М. | 360.80 | 3 |

| 5 | Стихотворения и поэмы | Лермонтов М.Ю. | 255.90 | 4 |

| 6 | Остров сокровищ | Стивенсон Р.Л. | 599.99 | 5 |

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

Affected rows: 6

Под нужное нам условие подходят две книги «Белая гвардия» Булгакова и «Черный человек» Есенина. В таблице **book** их количество увеличилось, а в таблице **supply** - обнулилось.

**Задание**

Для книг, которые уже есть на складе (в таблице**book**), но по другой цене, чем в поставке (**supply**),  необходимо в таблице **book**увеличить количество на значение, указанное в поставке,  и пересчитать цену. А в таблице  **supply**обнулить количество этих книг. Формула для пересчета цены:

price={(p\_1\*k\_1+p\_2\*k\_2)\over k\_1+k\_2}*price*=*k*1​+*k*2​(*p*1​∗*k*1​+*p*2​∗*k*2​)​где  *p1, p2* - цена книги в таблицах **book**и **supply**;

*k1, k2*- количество книг в таблицах **book**и **supply**.

1

UPDATE book

2

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

3

    INNER JOIN supply ON book.title = supply.title

4

                        and supply.author = author.name\_author

5

SET book.amount = book.amount + supply.amount,

6

  book.price = ((book.price\*book.amount)+(supply.price\*supply.amount))/(book.amount+supply.amount),

7

  supply.amount = 0

8

WHERE book.price /\*!=\*/<> supply.price

9

;

10

​

**Запросы на добавление, связанные таблицы**

Запросом на добавление можно добавить записи, отобранные с помощью запроса на выборку, который включает несколько таблиц:

INSERT INTO таблица (список\_полей)

SELECT список\_полей\_из\_других\_таблиц

FROM

таблица\_1

... JOIN таблица\_2 ON ...

...

**Пример**

В таблице **supply**  есть новые книги, которых на складе еще не было. Прежде чем добавлять их в таблицу **book**,  необходимо из таблицы **supply**отобрать новых авторов, если таковые имеются.

*Запрос:*

SELECT name\_author, supply.author

FROM

author

RIGHT JOIN supply ON author.name\_author = supply.author;

Поскольку таблица **author** и поле в таблице **supply** называются одинаково, желательно указывать полную ссылку на поле (**supply.author**), чтобы запрос был более читабельным.

*Результат:*

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

| name\_author | author |

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

| Булгаков М.А. | Булгаков М.А. |

| Достоевский Ф.М. | Достоевский Ф.М. |

| Есенин С.А. | Есенин С.А. |

| Пастернак Б.Л. | Пастернак Б.Л. |

| Лермонтов М.Ю. | Лермонтов М.Ю. |

| None | Стивенсон Р.Л. |

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

Выполнив правое внутреннее соединение таблиц, получили значение Null (None) в поле **name\_author** в строке того автора, которого нет в таблице**author**, в нашем случае это Стивенсон.

Теперь достаточно в запросе задать условие отбора, и список новых авторов готов для включения в таблицу **author**.

*Запрос:*

SELECT supply.author

FROM

author

RIGHT JOIN supply on author.name\_author = supply.author

WHERE name\_author IS Null;

*Результат:*

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

| author |

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

| Стивенсон Р.Л. |

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

**Задание**

Включить новых авторов в таблицу **author** с помощью запроса на добавление, а затем вывести все данные из таблицы **author**.  Новыми считаются авторы, которые есть в таблице **supply**, но нет в таблице **author**.

1

INSERT INTO author(name\_author)

2

SELECT supply.author

3

FROM

4

  author

5

   RIGHT JOIN supply on author.name\_author = supply.author

6

WHERE name\_author IS Null;

7

​

8

SELECT\*FROM author;

**Запрос на добавление, связанные таблицы**

Следующий шаг - добавить новые записи о книгах, которые есть в таблице **supply** и нет в таблице **book**. (В таблицах **supply** и **book** сохранены изменения предыдущих шагов). Поскольку в таблице **supply** не указан жанр книги, оставить его пока пустым (занести значение Null).

**Пример**

Прежде всего необходимо сформировать запрос с полями, которые соответствуют полям таблицы **book**, так как использовать только таблицу **supply** нельзя - в ней вместо кода автора стоит его фамилия.

*Запрос:*

SELECT title, author\_id, price, amount

FROM

author

INNER JOIN supply ON author.name\_author = supply.author;

*Результат:*

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

| title | author\_id | price | amount |

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

| Доктор Живаго | 4 | 380.80 | 4 |

| Черный человек | 3 | 570.20 | 0 |

| Белая гвардия | 1 | 540.50 | 0 |

| Идиот | 2 | 360.80 | 0 |

| Стихотворения и поэмы | 5 | 255.90 | 4 |

| Остров сокровищ | 6 | 599.99 | 5 |

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

Далее необходимо отобрать только новые книги из таблицы **supply**. Как видно из таблицы с результатами запроса, в тех записях, которые нужно добавить, значения столбца **amount** не равны 0 (количество уже учтенных книг обнулены предыдущим запросом). Добавим это условие в запрос.

*Запрос:*

SELECT title, author\_id, price, amount

FROM

author

INNER JOIN supply ON author.name\_author = supply.author

WHERE amount <> 0;

*Результат:*

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

| title | author\_id | price | amount |

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

| Доктор Живаго | 4 | 380.80 | 4 |

| Стихотворения и поэмы | 5 | 255.90 | 4 |

| Остров сокровищ | 6 | 599.99 | 5 |

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

**Задание**

Добавить новые книги из таблицы **supply** в таблицу **book** на основе сформированного выше запроса. Затем вывести для просмотра таблицу **book**.

1

INSERT INTO book(title, author\_id, price, amount)

2

SELECT title, author\_id, price, amount

3

FROM

4

  author

5

   INNER JOIN supply ON author.name\_author = supply.author

6

WHERE amount <> 0;

7

​

8

SELECT \* FROM book;

9

​

10

​

11

​

**Запрос на обновление, вложенные запросы**

После того, как новые книги добавлены в таблицу **book**, нужно указать к какому жанру они относятся. Для этого используется запрос на обновление, в котором можно указать значения столбцов из других таблиц, либо использовать вложенные запросы для получения этих значений.

**Пример**

Задать для книги Пастернака «Доктор Живаго»  жанр «Роман».

Если мы знаем код этой книги в таблице **book**(в нашем случае это 9) и код жанра «Роман» в таблице **genre** (это 1), запрос будет очень простым.

*Запрос:*

UPDATE book

SET genre\_id = 1

WHERE book\_id = 9;

SELECT \* FROM book;

*Результат:*

Affected rows: 1

Query result:

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

| 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 | 12 |

| 3 | Идиот | 2 | 1 | 437.11 | 13 |

| 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 | 12 |

| 8 | Лирика | 4 | 2 | 518.99 | 2 |

| 9 | Доктор Живаго | 4 | 1 | 380.80 | 4 |

| 10 | Стихотворения и поэмы | 5 | NULL | 255.90 | 4 |

| 11 | Остров сокровищ | 6 | NULL | 599.99 | 5 |

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

Более сложным будет запрос, если известно только название жанра (результат будет точно таким же):

*Запрос:*

UPDATE book

SET genre\_id =

(

SELECT genre\_id

FROM genre

WHERE name\_genre = 'Роман'

)

WHERE book\_id = 9;

SELECT \* FROM book;

**Задание**

 Занести для книги «Стихотворения и поэмы» Лермонтова жанр «Поэзия», а для книги «Остров сокровищ» Стивенсона - «Приключения». (Использовать два запроса).

1

/\*UPDATE book

2

SET genre\_id = 2

3

WHERE author\_id = 5

4

;

5

​

6

UPDATE book

7

SET genre\_id = 3

8

WHERE author\_id = 6

9

;\*/

10

​

11

UPDATE book

12

SET genre\_id =

13

   (

14

       SELECT genre\_id

15

       FROM genre

16

       WHERE name\_genre = 'Поэзия'

17

   )

18

WHERE book\_id = 10

19

;

20

​

21

UPDATE book

22

SET genre\_id =

23

   (

24

       SELECT genre\_id

25

       FROM genre

26

       WHERE name\_genre = 'Приключения'

27

   )

28

WHERE book\_id = 11

29

;

30

​

31

SELECT\*FROM book;

32

​

33

​

**Каскадное удаление записей связанных таблиц**

При создании таблицы для внешних ключей с помощью ON DELETE[устанавливаются опции](https://stepik.org/lesson/308885/step/9?unit=291011), которые определяют действия , выполняемые при удалении связанной строки из главной таблицы.

В частности, ON DELETE CASCADE автоматически удаляет строки из зависимой таблицы при удалении  связанных строк в главной таблице.

В таблице **book** эта опция установлена для поля **author\_id**.

**Пример**

Удалим из таблицы **author** всех авторов, фамилия которых начинается на «Д», а из таблицы **book**  - все книги этих авторов.

*Запрос:*

DELETE FROM author

WHERE name\_author LIKE "Д%";

SELECT \* FROM author;

SELECT \* FROM book;

*Результат:*

Affected rows: 1

Query result:

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

| author\_id | name\_author |

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

| 1 | Булгаков М.А. |

| 3 | Есенин С.А. |

| 4 | Пастернак Б.Л. |

| 5 | Лермонтов М.Ю. |

| 6 | Стивенсон Р.Л. |

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

Affected rows: 5

Query result:

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

| 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 | 12 |

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

| 7 | Черный человек | 3 | 2 | 570.20 | 12 |

| 8 | Лирика | 4 | 2 | 518.99 | 2 |

| 9 | Доктор Живаго | 4 | 1 | 380.80 | 4 |

| 10 | Стихотворения и поэмы | 5 | 2 | 255.90 | 4 |

| 11 | Остров сокровищ | 6 | 3 | 599.99 | 5 |

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

Одним запросом удаляются связанные записи из главной и зависимой таблицы. В нашем случае удалился автор Достоевский и все его книги.

**Задание**

Удалить всех авторов и все их книги, общее количество книг которых меньше 20.

1

DELETE FROM author

2

WHERE author\_id IN(

3

SELECT author\_id

4

FROM book

5

GROUP BY author\_id

6

HAVING SUM(amount) < 20

7

);

8

​

9

SELECT\*FROM author;

10

SELECT\*FROM book;

**Удаление записей главной таблицы с сохранением записей в зависимой**

При создании таблицы для внешних ключей с помощью ON DELETE[устанавливаются опции](https://stepik.org/lesson/308885/step/9?unit=291011), которые определяют действия, выполняемые при удалении связанной строки из главной таблицы.

Если задано SET NULL, то при удалении связанной строки из главной таблицы в зависимой, в столбце внешнего ключа, устанавливается значение **NULL**. (При этом в столбце внешнего ключа должно быть допустимо значение**NULL**)

В таблице **book** эта опция установлена на поле **genre\_id**.

**Пример**

Удалим из таблицы **genre** все  жанры, название которых заканчиваются на «я» , а в таблице **book**  -  для этих жанров установим значение Null.

*Запрос:*

DELETE FROM genre

WHERE name\_genre LIKE "%я";

SELECT \* FROM genre;

SELECT \* FROM book;

*Результат:*

Affected rows: 2

Query result:

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

| genre\_id | name\_genre |

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

| 1 | Роман |

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

Affected rows: 1

Query result:

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

| 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 | 12 |

| 3 | Идиот | 2 | 1 | 437.11 | 13 |

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

| 5 | Игрок | 2 | 1 | 480.50 | 10 |

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

| 7 | Черный человек | 3 | Null | 570.20 | 12 |

| 8 | Лирика | 4 | Null | 518.99 | 2 |

| 9 | Доктор Живаго | 4 | 1 | 380.80 | 4 |

| 10 | Стихотворения и поэмы | 5 | Null | 255.90 | 4 |

| 11 | Остров сокровищ | 6 | Null | 599.99 | 5 |

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

Affected rows: 11

В нашем случае удалились жанры «Поэзия» и «Приключения».

**Задание**

Удалить все жанры, к которым относится меньше 4-х книг. В таблице **book** для этих жанров установить значение Null.

1

DELETE FROM genre

2

WHERE genre\_id IN

3

(

4

   SELECT genre\_id

5

   FROM book

6

   GROUP BY genre\_id

7

   HAVING COUNT(amount) < 4

8

);

9

​

10

SELECT\*FROM book;

11

SELECT\*FROM genre;

12

​

## Удаление записей, использование связанных таблиц

При удалении записей из таблицы можно использовать информацию из других связанных с ней таблиц. В этом случае синтаксис запроса имеет вид:

DELETE FROM таблица\_1

USING

таблица\_1

INNER JOIN таблица\_2 ON ...

WHERE ...

**Пример**

Удалить всех авторов из таблицы **author**, у которых есть книги, количество экземпляров которых меньше 3. Из таблицы **book** удалить все книги этих авторов.

Запрос:

DELETE FROM author

USING

author

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

WHERE book.amount < 3;

SELECT \* FROM author;

SELECT \* FROM book;

Результат:

Affected rows: 1

Query result:

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

| author\_id | name\_author |

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

| 1 | Булгаков М.А. |

| 2 | Достоевский Ф.М. |

| 3 | Есенин С.А. |

| 5 | Лермонтов М.Ю. |

| 6 | Стивенсон Р.Л. |

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

Affected rows: 5

Query result:

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

| 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 | 12 |

| 3 | Идиот | 2 | 1 | 437.11 | 13 |

| 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 | 12 |

| 10 | Стихотворения и поэмы | 5 | 2 | 255.90 | 4 |

| 11 | Остров сокровищ | 6 | 3 | 599.99 | 5 |

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

Книги из таблицы **book** будут удалены автоматически, так как для столбца **author\_id** из таблицы **book** установлено [каскадное удаление записей](https://stepik.org/lesson/308887/step/6?unit=291013).

## Задание

Удалить всех авторов, которые пишут в жанре "Поэзия". Из таблицы **book** удалить все книги этих авторов. В запросе для отбора авторов использовать полное название жанра, а не его **id**.

1

DELETE FROM author

2

USING

3

  author

4

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

5

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

6

WHERE genre.name\_genre = 'Поэзия'

7

;

8

​

9

SELECT \* FROM book;

10

​

11

SELECT \* FROM author;