**Содержание урока**

С помощью запросов корректировки данных можно выполнить следующие действия по обработке заказов в интернет-магазине:

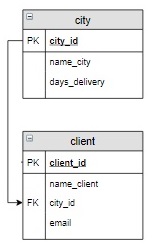
* [включение нового клиента в базу данных](https://stepik.org/lesson/310417/step/2?unit=292723);
* [формирование нового заказа некоторым пользователем](https://stepik.org/lesson/310417/step/3?unit=292723);
* [включение в заказ одной или нескольких книг с указанием их количества](https://stepik.org/lesson/310417/step/4?unit=292723);
* [уменьшение количества книг на складе](https://stepik.org/lesson/310417/step/5?unit=292723);
* создание счета на оплату ([полный счет](https://stepik.org/lesson/310417/step/6?unit=292723), [итоговый счет](https://stepik.org/lesson/310417/step/7?unit=292723));
* [добавление этапов продвижения заказа](https://stepik.org/lesson/310417/step/8?unit=292723);
* фиксация дат прохождения каждого этапа заказа ([начало этапа](https://stepik.org/lesson/310417/step/9?unit=292723), [завершение этапа](https://stepik.org/lesson/310417/step/10?unit=292723)).

Предметная область, логическая структура базы данных, содержание таблиц – как на [предыдущем уроке](https://stepik.org/lesson/308891/step/4?unit=291017).

## Задание

Включить нового человека в таблицу с клиентами. Его имя **Попов Илья**, его email **popov@test**, проживает он в **Москве**.

**Фрагмент логической схемы базы данных:**



**Пояснение**

1. В запросах на добавление можно одновременно заносить и константы, и данные из других таблиц. Для этого в той части запроса INSERT , где задается запрос на выборку, в качестве полей для вставки указываются не только поля других таблиц, но и  константы:

INSERT INTO ...

SELECT 'Попов Илья', city\_id, 'popov@test'

FROM city

WHERE ...;

2. Для просмотра той таблицы, в которую внесены изменения, используйте запрос вида:

SELECT \* FROM таблица;

**Связанные шаги**

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

*Включить нового человека в таблицу с клиентами. Его имя****Попов Илья****, его email****popov@test****, проживает он в****Москве****.*

**Результат**

Affected rows: 1

Query result (выборка из таблицы client):

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

| client\_id | name\_client | city\_id | email |

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

| 1 | Баранов Павел | 3 | baranov@test |

| 2 | Абрамова Катя | 1 | abramova@test |

| 3 | Семенонов Иван | 2 | semenov@test |

| 4 | Яковлева Галина | 1 | yakovleva@test |

| 5 | Попов Илья | 1 | popov@test |

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

**Наполнение таблиц (перед выполнением шага)**

1

INSERT INTO client(name\_client, city\_id, email)

2

SELECT 'Попов Илья', city\_id, 'popov@test'

3

FROM city

4

WHERE name\_city = 'Москва';

5

​

6

SELECT\*FROM client;

7

SELECT \* FROM city;

8

​

## Задание

Создать новый заказ для Попова Ильи. Его комментарий для заказа: «Связаться со мной по вопросу доставки».

**Фрагмент логической схемы базы данных:**



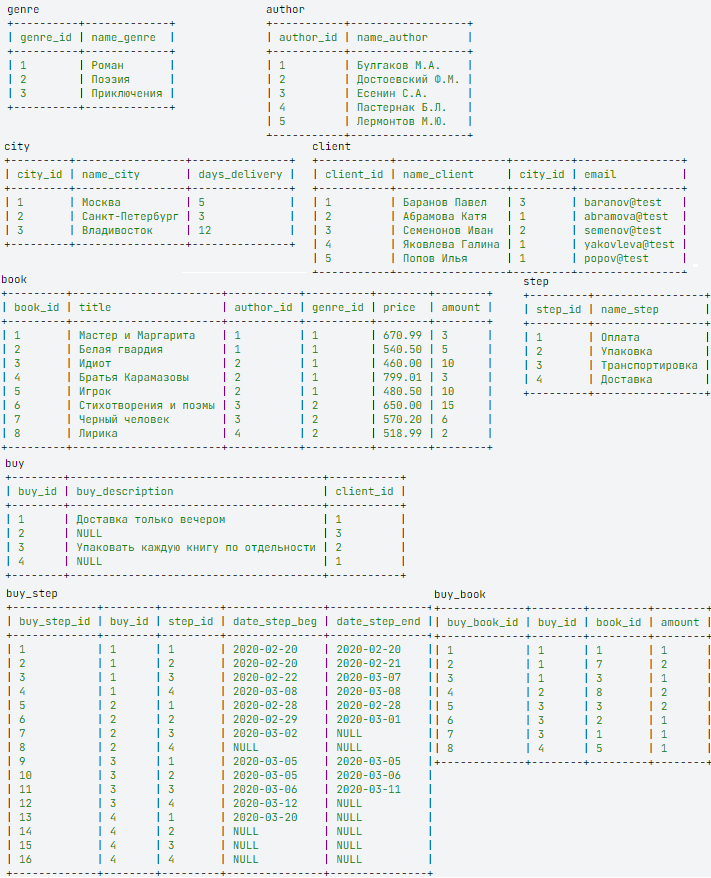
**Связанные шаги**

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

*Создать новый заказ для Попова Ильи. Его комментарий для заказа: «Связаться со мной по вопросу доставки».*

***Важно****! В решении нельзя использоваться****VALUES****и делать отбор по****client\_id****.*

**Результат**

**Наполнение таблиц(перед выполнением шага)**

1

INSERT INTO buy(buy\_description, client\_id)

2

SELECT 'Связаться со мной по вопросу доставки', client\_id

3

FROM client

4

WHERE name\_client = 'Попов Илья';

5

​

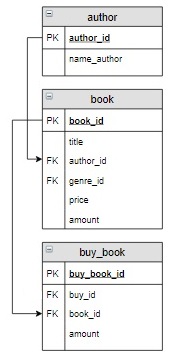
6

SELECT\*FROM buy;

**Задание**

В таблицу **buy\_book** добавить заказ с номером 5. Этот заказ должен содержать книгу Пастернака «Лирика» в количестве двух экземпляров и книгу Булгакова «Белая гвардия» в одном экземпляре.

**Фрагмент логической схемы базы данных:**



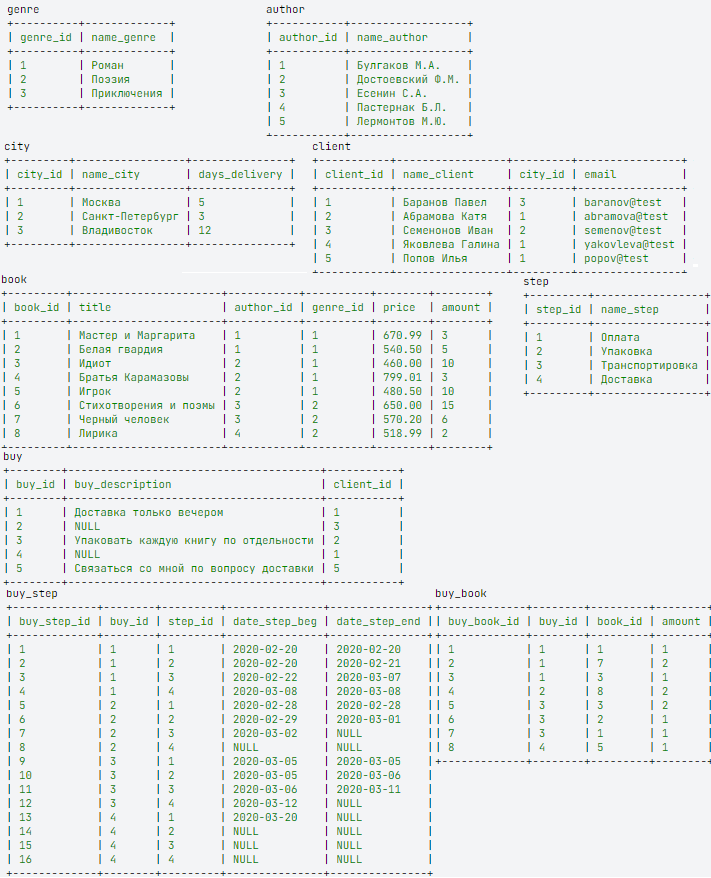
**Пояснение**

**Связанные шаги**

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

*В таблицу****buy\_book****добавить заказ с номером 5. Этот заказ должен содержать книгу Пастернака «Лирика» в количестве двух экземпляров и книгу Булгакова «Белая гвардия» в одном экземпляре.*

**Результат**

**Наполнение таблиц(перед выполнением шага)**

1

INSERT INTO buy\_book(buy\_id, book\_id, amount)

2

SELECT 5, book\_id, 2

3

FROM book

4

WHERE title = 'Лирика' AND author\_id = 4;

5

​

6

INSERT INTO buy\_book(buy\_id, book\_id, amount)

7

SELECT 5, book\_id, 1

8

FROM book

9

WHERE title = 'Белая гвардия' AND author\_id = 1;

10

​

11

SELECT\*FROM author;

12

SELECT\*FROM book;

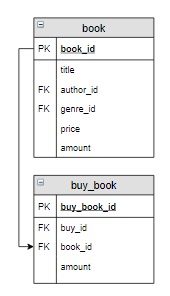
13

SELECT\*FROM buy\_book;

**Задание**

Количество тех книг на складе, которые были включены в заказ с номером 5, уменьшить на то количество, которое в заказе с номером 5  указано.

**Фрагмент логической схемы базы данных:**



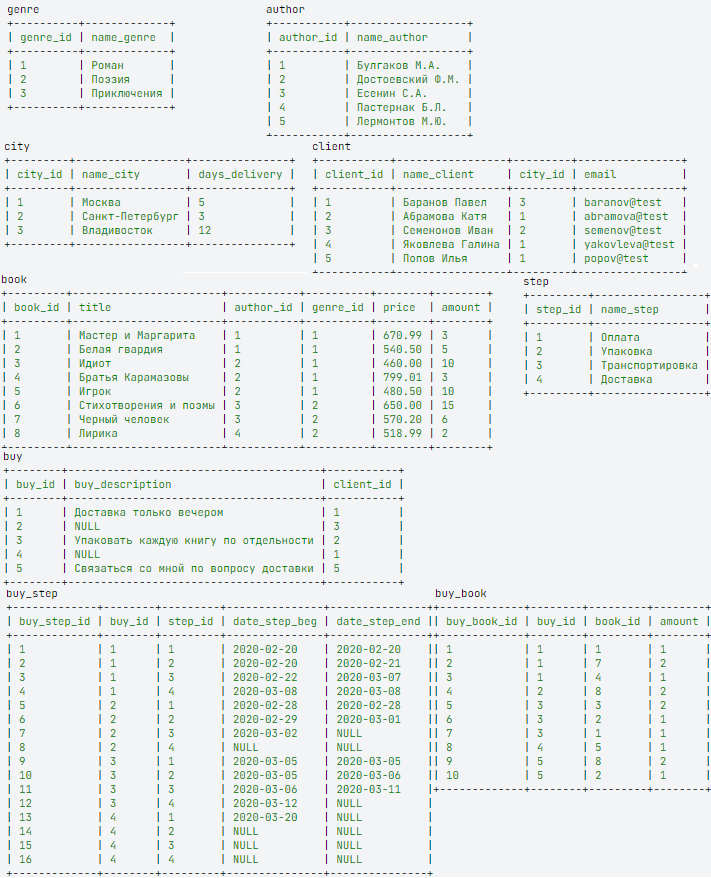
**Пояснение**

**Связанные шаги**

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

*Уменьшить количество тех книг на складе, которые были включены в заказ с номером 5.*

**Результат**

**Наполнение таблиц(перед выполнением шага)**

1

UPDATE book

2

SET amount = amount - 2

3

WHERE title = 'Лирика' AND author\_id = 4;

4

​

5

UPDATE book

6

SET amount = amount - 1

7

WHERE title = 'Белая гвардия' AND author\_id = 1;

8

​

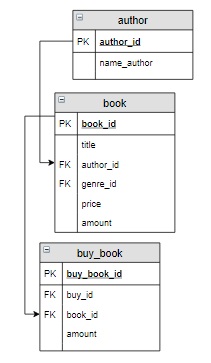
9

SELECT\*FROM book;

**Задание**

Создать счет (таблицу **buy\_pay**) на оплату заказа с номером 5, в который включить название книг, их автора, цену, количество заказанных книг и  стоимость. Последний столбец назвать **Стоимость**. Информацию в таблицу занести в отсортированном по названиям книг виде.

**Фрагмент логической схемы базы данных:**



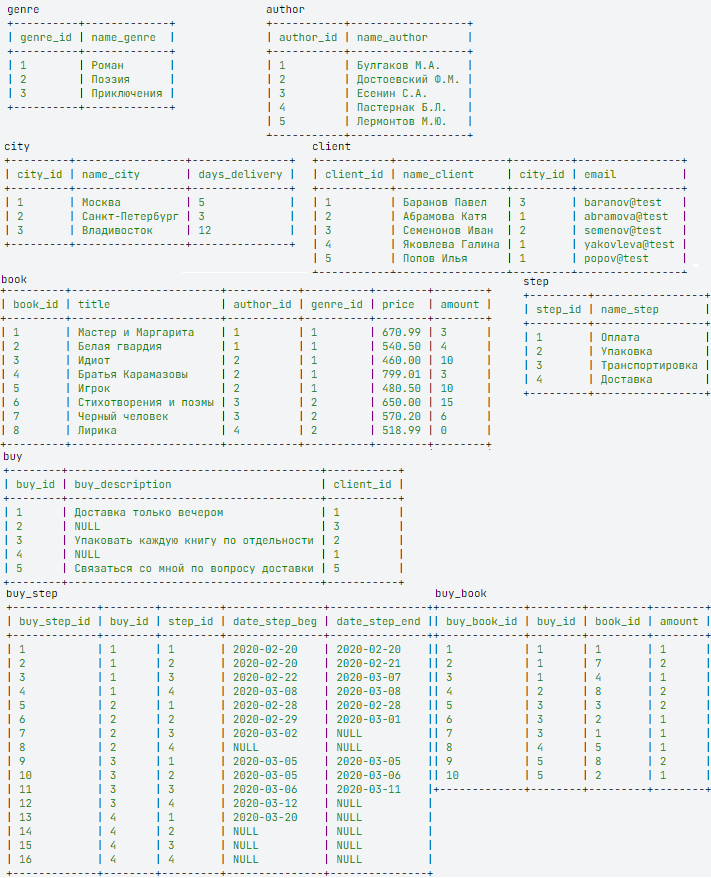
**Пояснение**

**Связанные шаги**

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

*Создать счет (таблицу****buy\_pay****) на оплату заказа с номером 5, в который включить название книг, их автора, цену, количество заказанных книг и  стоимость. Последний столбец назвать****Стоимость****. Информацию в таблицу занести в отсортированном по названиям книг виде.*

**Результат**

**Наполнение таблиц(перед выполнением шага)**

1

CREATE TABLE buy\_pay AS

2

SELECT title, name\_author, price, buy\_book.amount, SUM(price\*buy\_book.amount) AS 'Стоимость'

3

FROM

4

  author

5

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

6

   JOIN buy\_book ON book.book\_id = buy\_book.book\_id

7

WHERE buy\_book.buy\_id = 5

8

GROUP BY title, name\_author, price, buy\_book.amount

9

ORDER BY title;

10

​

11

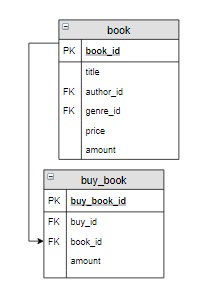
SELECT\*FROM buy\_pay;

12

## ​Задание

Создать общий счет (таблицу **buy\_pay**) на оплату заказа с номером 5. Куда включить номер заказа, количество книг в заказе (название столбца **Количество**) и его общую стоимость (название столбца **Итого**). Для решения используйте ОДИН запрос.

*Фрагмент логической схемы базы данных:*

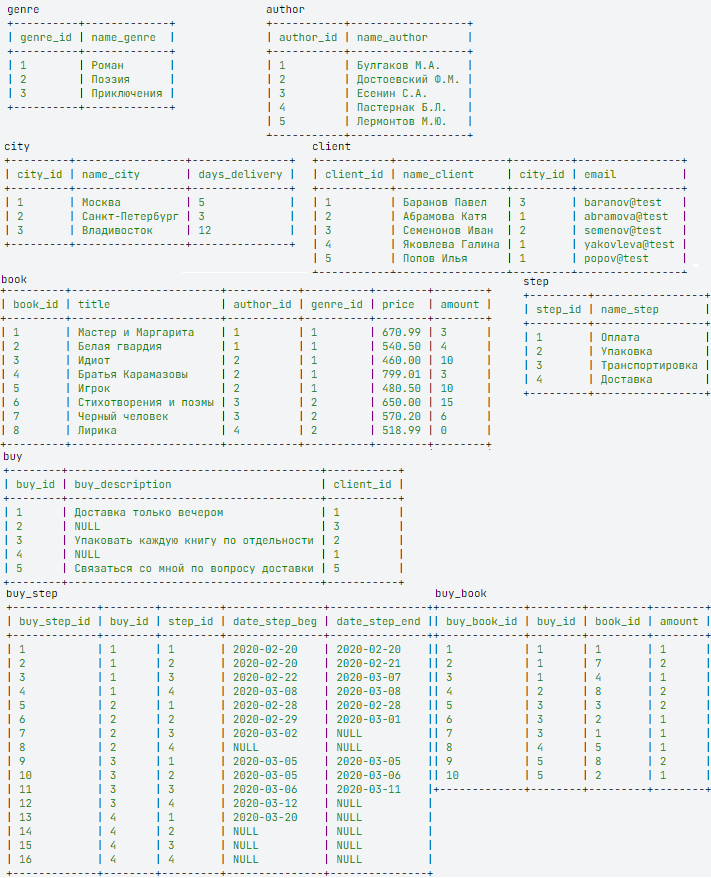


**Связанные шаги**

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

*Создать общий счет (таблицу****buy\_pay****) на оплату заказа с номером 5. Куда включить номер заказа, количество книг в заказе (название столбца****Количество****) и его общую стоимость (название столбца****Итого****).  Для решения используйте ОДИН запрос.*

**Результат**

**Наполнение таблиц(перед выполнением шага)**

1

CREATE TABLE buy\_pay AS

2

SELECT buy\_id,

3

   SUM(buy\_book.amount) AS 'Количество',

4

   SUM(price\*buy\_book.amount) AS 'Итого'

5

FROM

6

  book

7

   JOIN buy\_book ON book.book\_id = buy\_book.book\_id

8

   AND buy\_id = 5;

9

​

10

​

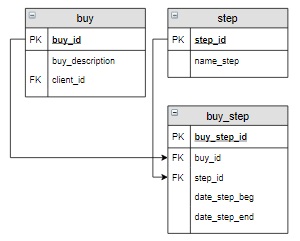
11

SELECT\*FROM buy\_pay;

## Задание

В таблицу **buy\_step** для заказа с номером 5 включить все этапы из таблицы **step**, которые должен пройти этот заказ. В столбцы **date\_step\_beg** и **date\_step\_end**всех записей занести Null.

**Фрагмент логической схемы базы данных:**



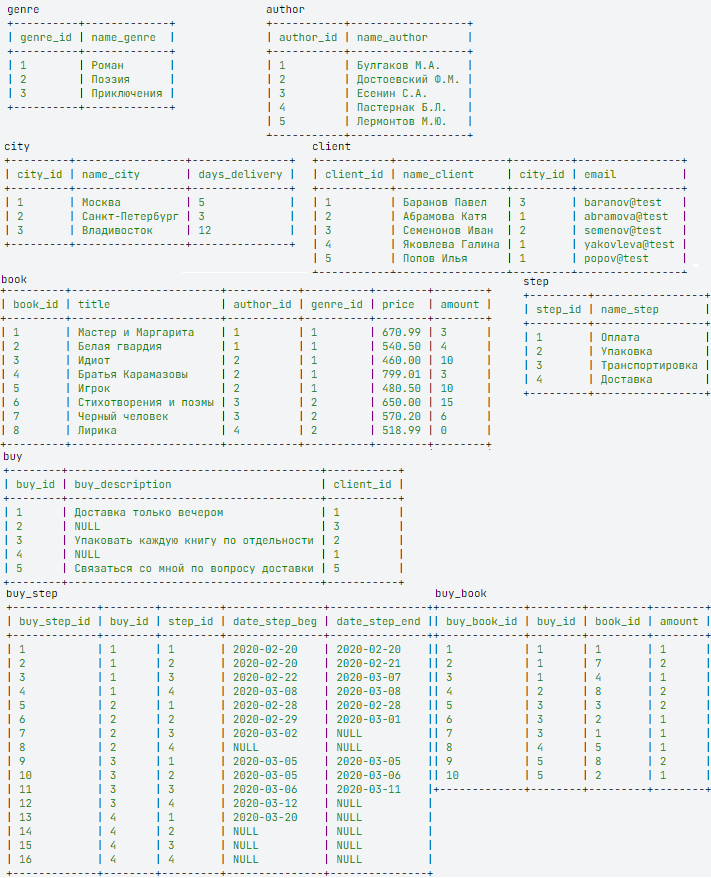
**Пояснение**

**Связанные шаги**

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

*В таблицу****buy\_step****для заказа с номером 5 включить все этапы из таблицы****step****, которые должен пройти этот заказ. В столбцы****date\_step\_beg****и****date\_step\_end****всех записей занести Null.*

**Результат**

**Наполнение таблиц(перед выполнением шага)**

1

/\*INSERT INTO buy\_step (buy\_id, buy\_step.step\_id)

2

SELECT 5, step.step\_id

3

FROM

4

  step

5

  JOIN buy\_step ON step.step\_id = buy\_step.step\_id

6

  JOIN buy ON buy.buy\_id = buy\_step.buy\_id

7

GROUP BY 1, 2;\*/

8

​

9

INSERT INTO buy\_step (buy\_id, step\_id, date\_step\_beg, date\_step\_end)

10

SELECT buy.buy\_id, step\_id, Null, Null

11

FROM buy, step

12

WHERE buy.buy\_id = 5;

13

​

14

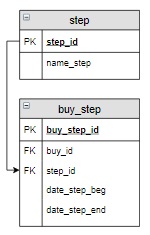
SELECT \* FROM buy\_step;

**Задание**

В таблицу **buy\_step** занести дату 12.04.2020 выставления счета на оплату заказа с номером 5.

Правильнее было бы занести не конкретную, а текущую дату. Это можно сделать с помощью функции **Now()**. Но при этом в разные дни будут вставляться разная дата, и задание нельзя будет проверить, поэтому  вставим дату 12.04.2020.

**Фрагмент логической схемы базы данных:**



**Пояснение**

**Связанные шаги**

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

*В таблицу****buy\_step****занести дату 12.04.2020 выставления счета на оплату заказа с номером 5.*

*Правильнее было бы занести не конкретную, а текущую дату. Это можно сделать с помощью функции****Now()****. Но при этом в разные дни будут вставляться разная дата, и задание нельзя будет проверить, поэтому  вставим дату 12.04.2020.*

**Результат**

Affected rows: 1

Query result (выборка из таблицы buy\_step для заказа с номером 5):

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

| buy\_step\_id | buy\_id | step\_id | date\_step\_beg | date\_step\_end |

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

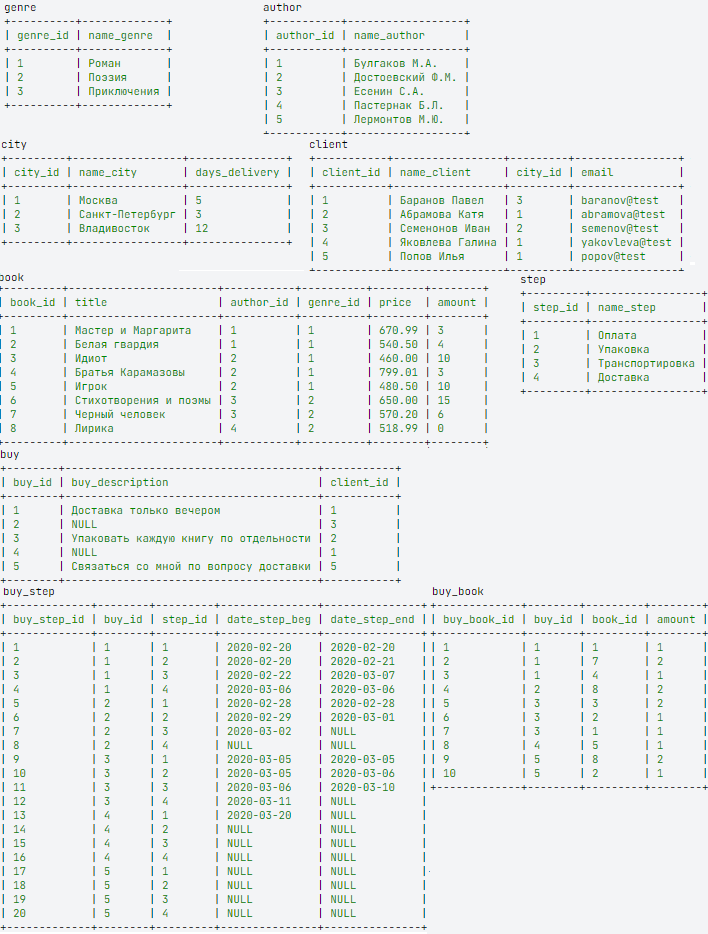
| 17 | 5 | 1 | 2020-04-12 | NULL |

| 18 | 5 | 2 | NULL | NULL |

| 19 | 5 | 3 | NULL | NULL |

| 20 | 5 | 4 | NULL | NULL |

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

**Наполнение таблиц(перед выполнением шага)**

Enter an SQL query

Correct answer from **16,290** learners

Total **67%** of tries are correct



1

​

2

UPDATE buy\_step

3

SET date\_step\_beg = '2020-04-12' /\*Now()\*/

4

WHERE buy\_id = 5 AND step\_id = 1;

5

/\*2

6

UPDATE buy\_step JOIN step

7

              USING(step\_id)

8

SET date\_step\_beg = '2020-04-12'

9

WHERE buy\_id = 5 AND name\_step = 'Оплата';\*/

10

/\*3

11

UPDATE buy\_step

12

SET date\_step\_beg = '2020-04-12'

13

WHERE buy\_id = 5 AND step\_id = (SELECT step\_id

14

                              FROM step

15

                              WHERE name\_step = 'Оплата');\*/

16

/\*4

17

UPDATE buy\_step

18

SET date\_step\_beg = '2020-04-12'

19

WHERE (buy\_id, step\_id) = (5, 1);\*/

20

/\*5

21

UPDATE buy\_step

22

    INNER JOIN step

23

    ON buy\_step.step\_id = step.step\_id

24

SET date\_step\_beg = DATE('2020-04-12')

25

WHERE buy\_id = 5 AND step\_id = 1;\*/

26

/\*6

27

UPDATE buy\_step bs, step s

28

SET bs.date\_step\_beg = '2020-04-12'

29

WHERE bs.buy\_id = 5

30

AND bs.step\_id =

31

  (

32

    SELECT s.step\_id FROM step s

33

    WHERE s.name\_step = "Оплата"

34

  );\*/

35

​

36

SELECT \*

37

FROM buy\_step

38

WHERE buy\_id = 5;

39

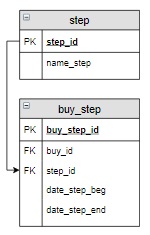
​

**Задание**

Завершить этап «**Оплата**» для заказа с номером 5, вставив в столбец **date\_step\_end** дату 13.04.2020, и начать следующий этап («**Упаковка**»), задав в столбце **date\_step\_beg** для этого этапа ту же дату.

Реализовать два запроса для завершения этапа и начала следующего. Они должны быть записаны в общем виде, чтобы его можно было применять для любых этапов, изменив только текущий этап. Для примера пусть это будет этап «**Оплата**».

**Фрагмент предметной области:**



**Пояснение**

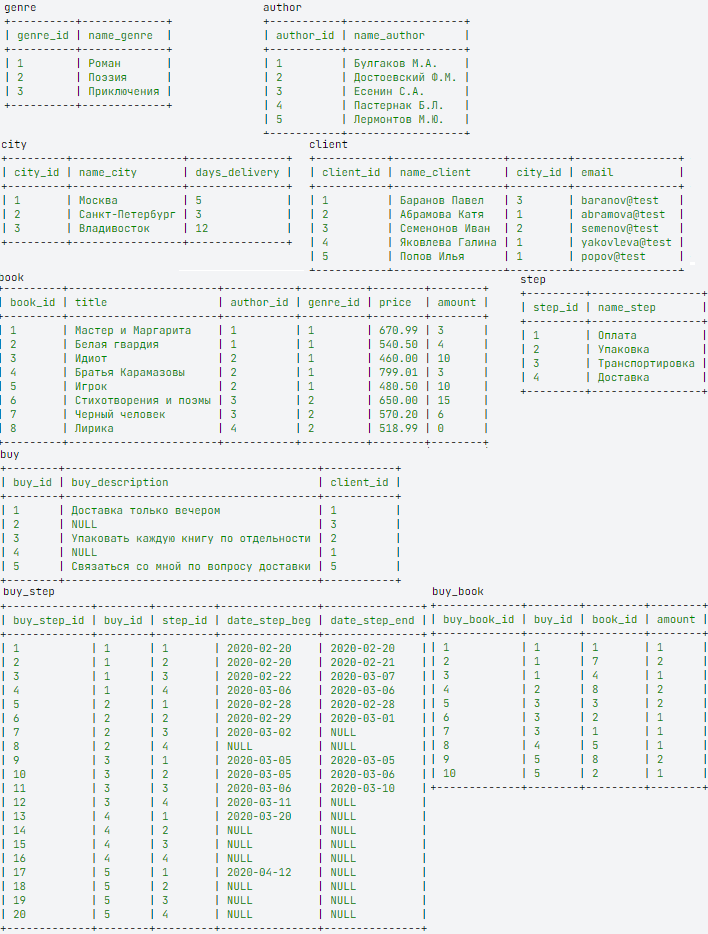
**Связанные шаги**

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

*Завершить этап «Оплата» для заказа с номером 5, вставив в столбец****date\_step\_end****дату 13.04.2020, и начать следующий этап («Упаковка»), задав в столбце****date\_step\_beg****для этого этапа ту же дату.*

*Реализовать два запроса для завершения этапа и начале следующего. Они должны быть записаны в общем виде, чтобы его можно было применять для любых этапов, изменив только текущий этап. Для примера пусть это будет этап «****Оплата****».*

**Результат**

**Наполнение таблиц(перед выполнением шага)**

1

​

2

/\*1

3

UPDATE buy\_step

4

SET date\_step\_end = '2020-04-13'

5

WHERE buy\_id = 5 AND step\_id = 1;

6

UPDATE buy\_step

7

SET date\_step\_beg = '2020-04-13'

8

WHERE buy\_id = 5 AND step\_id = 2;\*/

9

​

10

/\*2

11

UPDATE buy\_step

12

SET date\_step\_end = IF(step\_id = (SELECT step\_id FROM step WHERE name\_step = "Оплата"),

13

                    '2020-04-13', date\_step\_end),

14

  date\_step\_beg = IF( step\_id = (SELECT step\_id FROM step WHERE name\_step = "Упаковка"),

15

                    '2020-04-13', date\_step\_beg)

16

WHERE buy\_id = 5;\*/

17

​

18

/\*3

19

UPDATE buy\_step AS bs, buy\_step AS bs\_bs

20

SET bs.date\_step\_end = '2020-04-13',

21

  bs\_bs.date\_step\_beg = '2020-04-13'

22

WHERE bs.buy\_id = 5

23

    AND

24

    bs.step\_id = 1

25

    AND

26

    bs\_bs.buy\_id = 5

27

    AND

28

    bs\_bs.step\_id = 2;\*/

29

/\*4 (@wert - перемінна :) )

30

SET @wert = DATE('2020-04-13');

31

SET @step = 1;

32

​

33

UPDATE buy\_step

34

SET date\_step\_end = @wert

35

WHERE buy\_id = 5 AND step\_id = @step;

36

​

37

UPDATE buy\_step

38

SET date\_step\_beg = @wert

39

WHERE buy\_id = 5 AND step\_id = (@step + 1);\*/

40

​

41

SET @date = DATE('2020-04-13');

42

UPDATE buy\_step JOIN step

43

      ON step.step\_id = buy\_step.step\_id

44

SET date\_step\_end = @date

45

WHERE buy\_id = 5 AND name\_step = 'Оплата';

46

​

47

UPDATE buy\_step JOIN step

48

      ON step.step\_id = buy\_step.step\_id

49

SET date\_step\_beg = @date

50

WHERE buy\_id = 5 AND buy\_step.step\_id = (

51

                    SELECT step\_id + 1

52

                    FROM step

53

                    WHERE name\_step = 'Оплата');

54

​

55

SELECT\*

56

FROM buy\_step

57

WHERE buy\_id = 5;

58

​