

# Описание процессов обработки данных

---

Реляционная модель

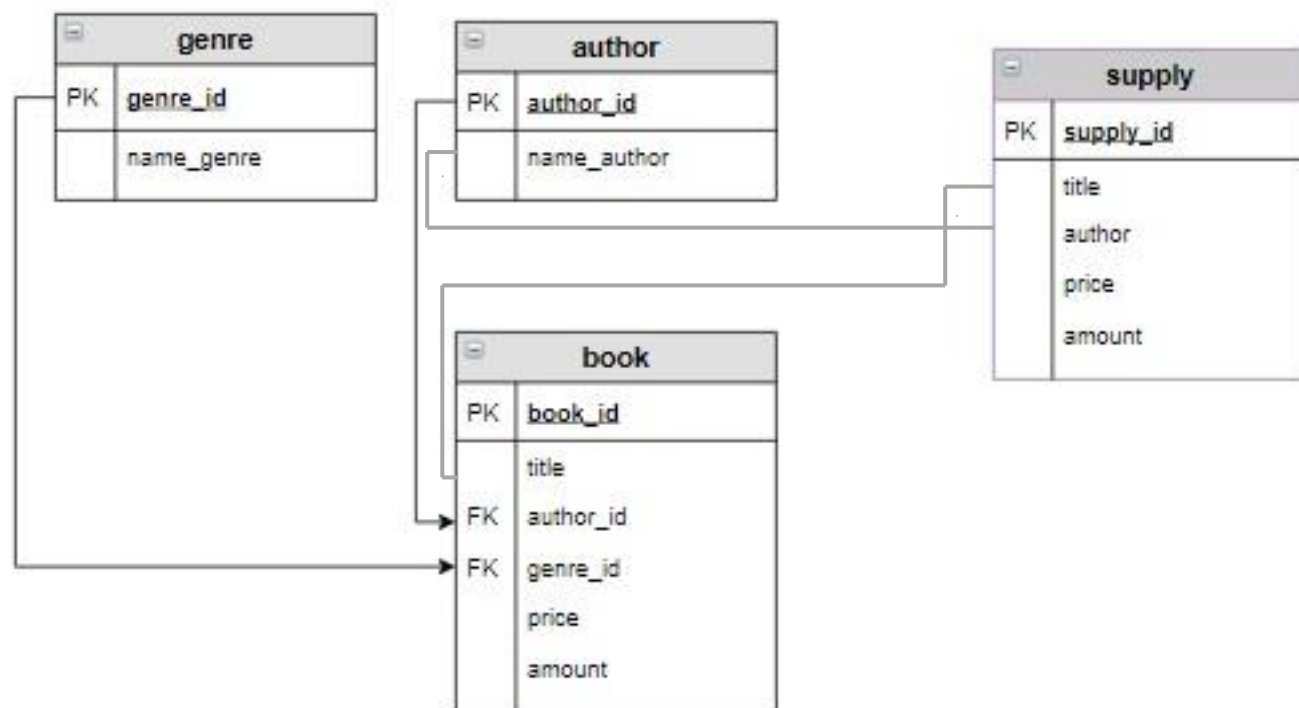
Преподаватель: канд. тех. наук, доц.  
Озерова Г.П.



# Описание предметной области

База данных о книжном складе включает три таблицы **genre**, **author** и **book**. Информация о поставке занесена в таблицу **supply**.

**Логическая схема базы данных:**



# Типовые операции

**Типовые действия**, которые необходимо осуществлять на складе:

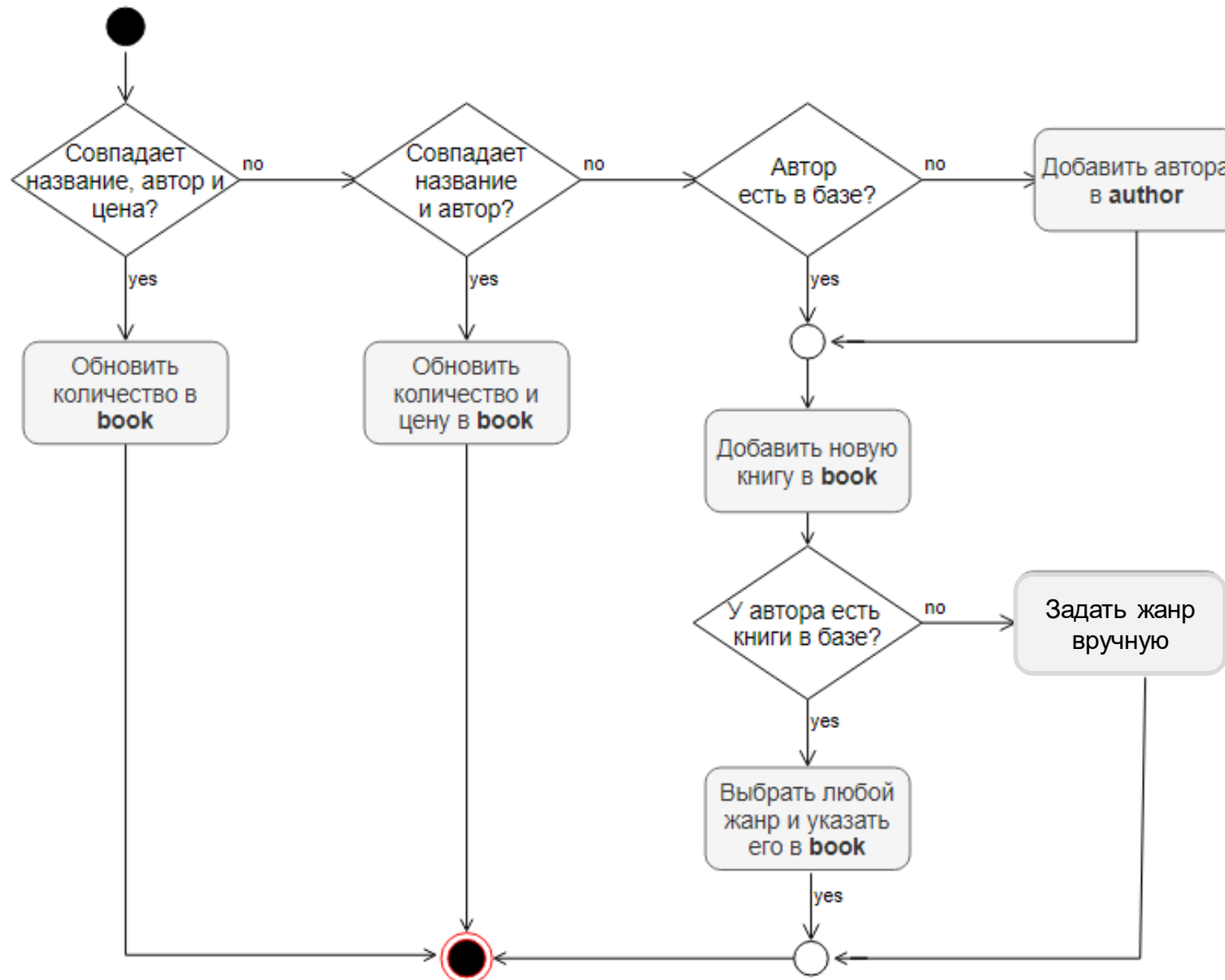
- I. Получение нового товара
- II. Продажа и учет товара
- III. Заказ товара

# Получение нового товара

## I. Получение нового товара :

1. для книг, которые уже есть на складе по той же цене, что и в поставке, - увеличить их количество на значение, указанное в поставке ;
2. 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;
3. для книг, которых на складе нет:
  - ✓ проверить, есть ли автор книги в базе, если нет - занести фамилию автора;
  - ✓ добавить новую запись о книге, оставив поле для описания жанра пустым;
  - ✓ задать жанр для новых книг (если автор новый, задать жанр вручную, в противном случае – выбрать любой).

# Получение нового товара

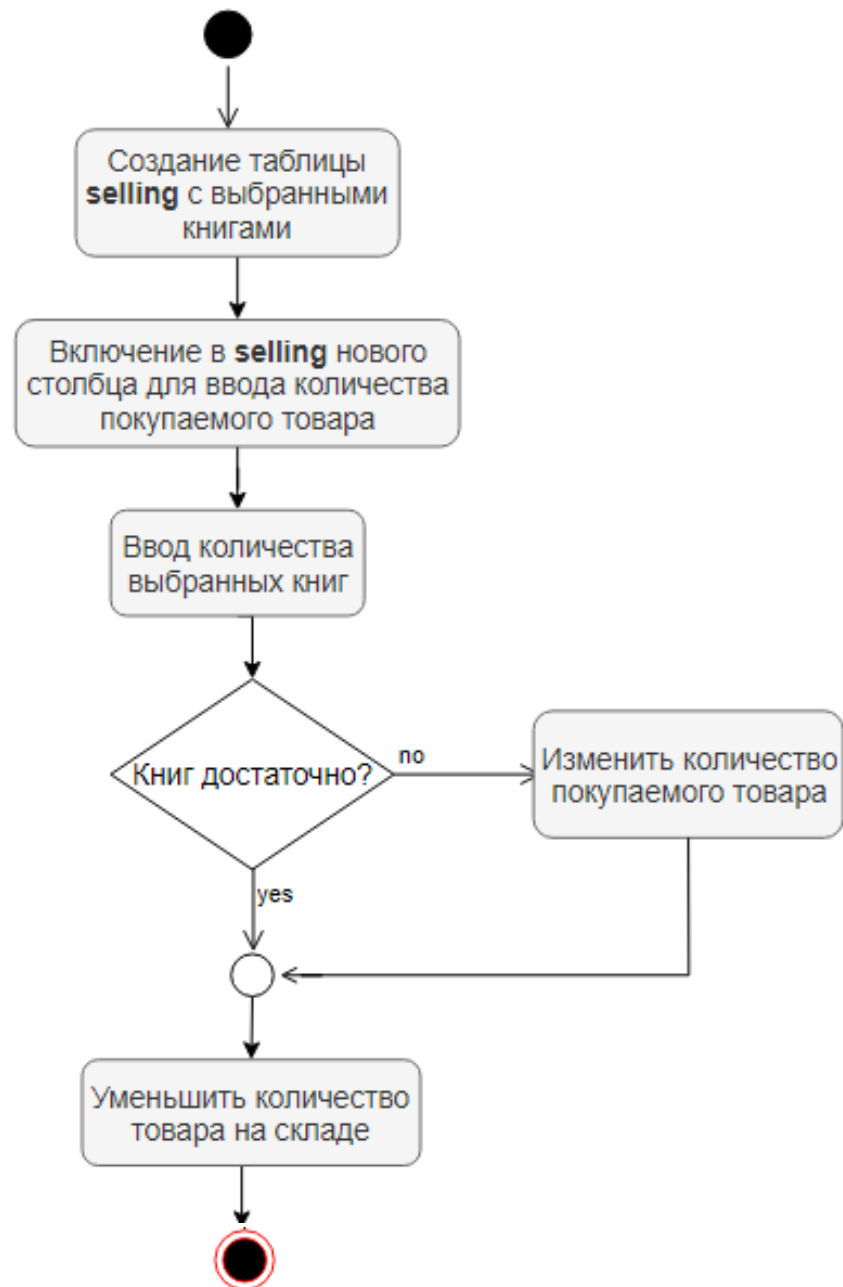


# Получение нового товара

## II. Продажа и учет товара:

1. выбор товара покупателем;
2. ввод количества покупаемого товара;
3. корректировка покупки - если покупатель указал количества товара больше, чем есть на складе, изменить количество покупаемого товара (отдать все имеющееся на складе количество);
4. уменьшение количества товара на складе.

# Получение нового товара



# Получение нового товара

## **III. Заказ товара:**

1. Формирование заказа на те товары, количество которых меньше допустимого остатка.
2. Заказ нового товара.

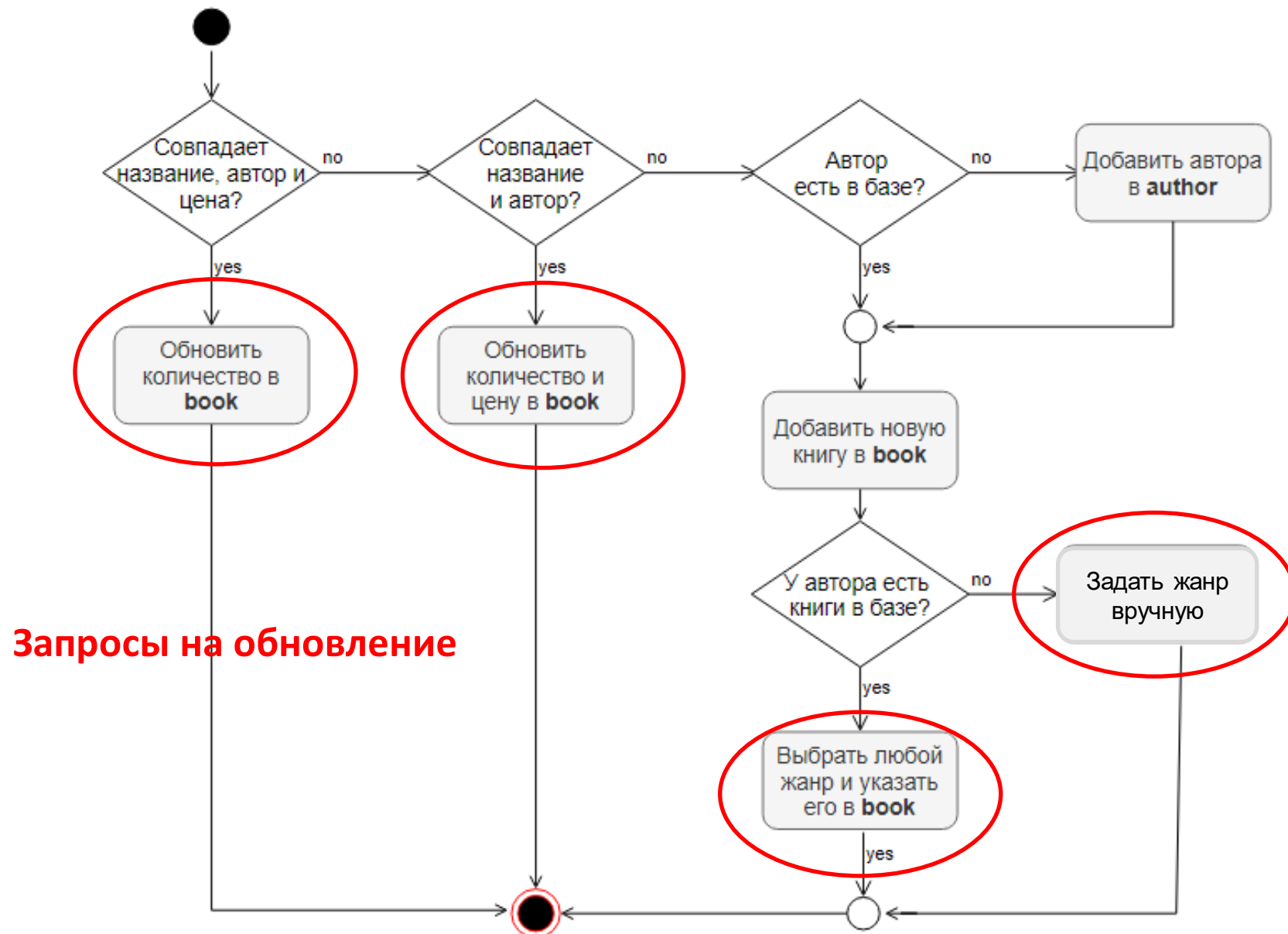


# Запросы корректировки

Реализация каждого отдельного действия, связанного с изменением информации в базе данных, осуществляется с помощью запросов корректировки данных:

- ✓ Запросов на обновление (**UPDATE**);
- ✓ Запросов на добавление (**INSERT**);
- ✓ Запросов на создание таблицы (**CREATE**);
- ✓ Запросов на изменение структуры таблицы (**ALTER**);
- ✓ Запросов на удаление (**DELETE**).

# Получение нового товара



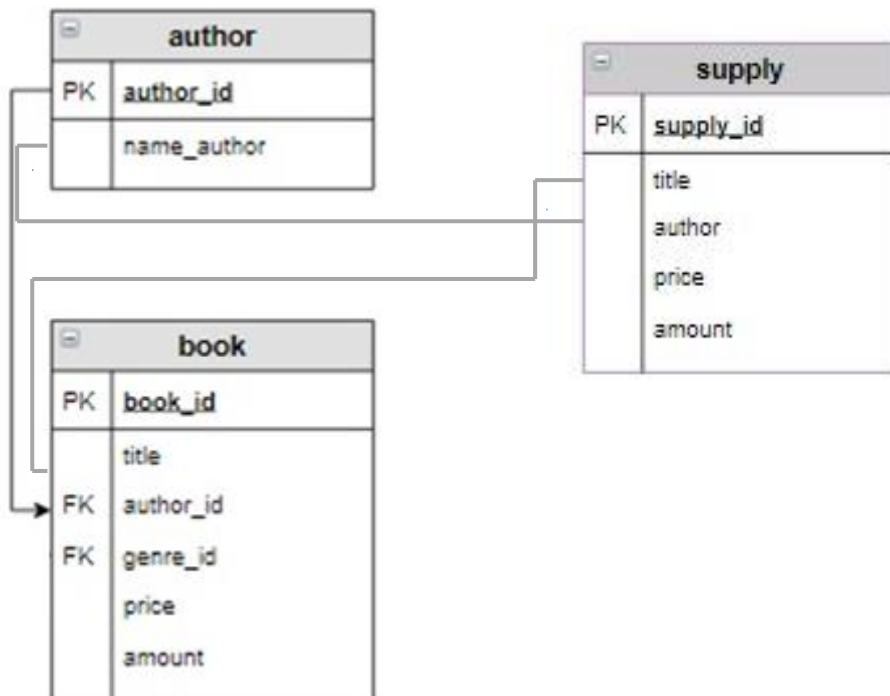
# Получение нового товара



# Поставка товара, операция 1

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

## Фрагмент логической схемы базы данных



## Изменяемые таблицы:

	book
PK	book_id
	title
FK	author_id
FK	genre_id
	price
	amount

	supply
PK	supply_id
	title
	author
	price
	amount





# Поставка товара, операция 1

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

**Запрос на обновление:**

**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;

# Поставка товара, операция 2

**Пример.** Для книг, которые уже есть на складе, но по другой цене, чем в поставке, 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:

$$price = \frac{(p_1 * k_1 + p_2 * k_2)}{k_1 + k_2}$$

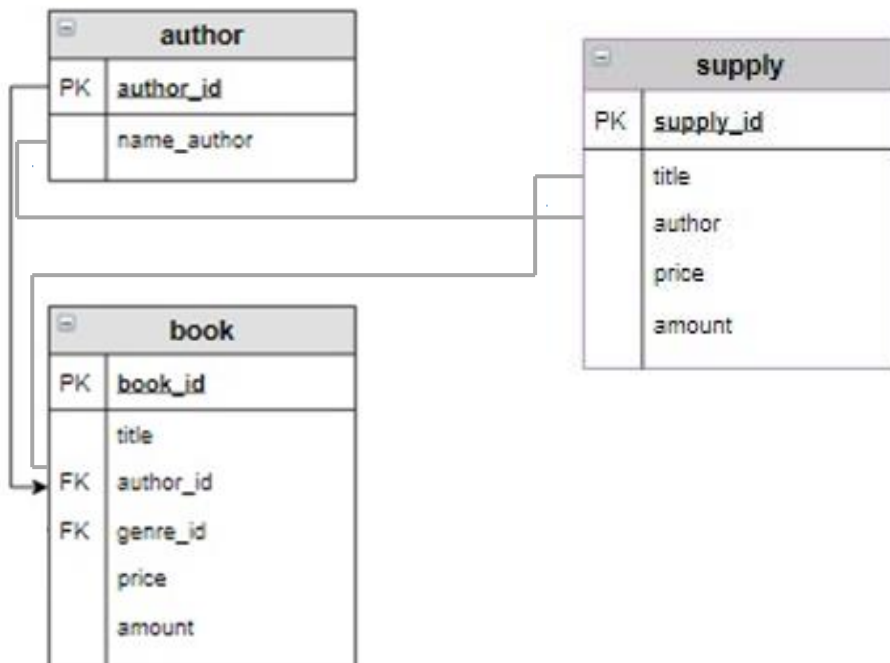
где  $p_1, p_2$  - цена книги в таблицах **book** и **supply**;

$k_1, k_2$  - количество книг в таблицах **book** и **supply**.

# Поставка товара, операция 2

**Пример.** Для книг, которые уже есть на складе, но по другой цене, чем в поставке, необходимо в таблице **book** увеличить количество на значение, указанное в поставке, и пересчитать цену. А в таблице **supply** обнулить количество этих книг.

## Фрагмент логической схемы базы данных



## Изменяемые таблицы:

	book
PK	<b>book_id</b>
	title
FK	<b>author_id</b>
FK	<b>genre_id</b>
	price
	amount

	supply
PK	<b>supply_id</b>
	title
	author
	price
	amount

# Поставка товара, операция 2

**Пример.** Для книг, которые уже есть на складе, но по другой цене, чем в поставке, необходимо в таблице **book** увеличить количество на значение, указанное в поставке, и пересчитать цену. А в таблице **supply** обнулить количество этих книг.

**Условие соединения:**

У книг из таблиц **book** и **supply** должны совпадать и названия, и автор.

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

# Поставка товара, операция 2

**Пример.** Для книг, которые уже есть на складе, но по другой цене, чем в поставке, необходимо в таблице **book** увеличить количество на значение, указанное в поставке, и пересчитать цену. А в таблице **supply** обнулить количество ЭТИХ КНИГ.

**Запрос на обновление:**

**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.price = (book.amount*book.price + supply.price*supply.amount) /
              (book.amount + supply.amount)
book.amount = book.amount + supply.amount,
supply.amount = 0
```

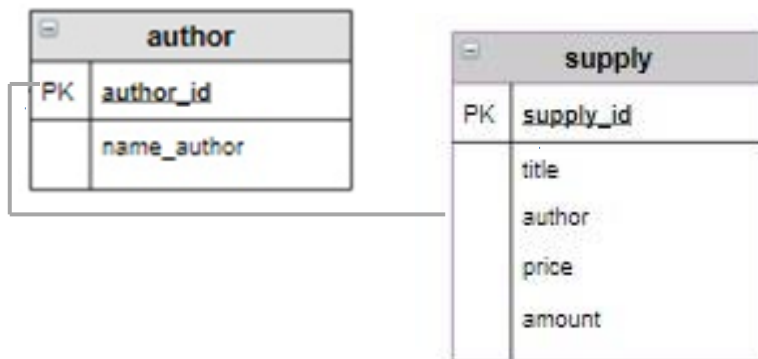
**WHERE** book.price != supply.price;



# Поставка товара, операция 3, часть 1

**Пример.** Из таблицы **supply** отобрать новых авторов, если таковые имеются, и добавить их в таблицу **author**.

**Фрагмент логической схемы базы данных:**



**Изменяемая таблица:**

author	
PK	author_id
	name_author

# Поставка товара, операция 3, часть 1

**Пример.** Из таблицы **supply** отобрать новых авторов, если таковые имеются, и добавить их в таблицу **author**.

**Шаг 1.** Отобрать из таблицы **supply** тех авторов, которых нет в таблице **author**.

**Условие связывания:**

```
author  
RIGHT JOIN supply ON author.name_author = supply.author
```

# Поставка товара, операция 3, часть 1

**Пример.** Из таблицы **supply** отобрать новых авторов, если таковые имеются, и добавить их в таблицу **author**.

**Шаг 1.** Отобрать из таблицы **supply** тех авторов, которых нет в таблице **author**.

**Запрос на выборку:**

```
SELECT
    supply.author
FROM
    author
    RIGHT JOIN supply ON author.name_author = supply.author
WHERE name_author IS NULL;
```

# Поставка товара, операция 3, часть 1

**Пример.** Из таблицы **supply** отобрать новых авторов, если таковые имеются, и добавить их в таблицу **author**.

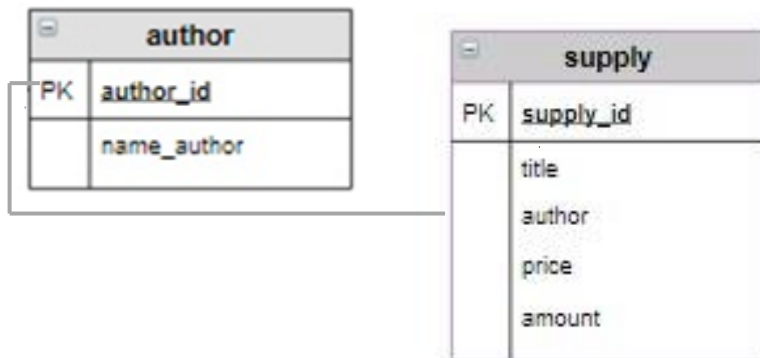
**Шаг 2.** Добавить отобранных авторов в таблицу **author**.

```
INSERT INTO
    author(name_author)
SELECT
    supply.author
FROM
    author
    RIGHT JOIN supply ON author.name_author = supply.author
WHERE name_author IS NULL;
```

# Поставка товара, операция 3, часть 2

**Пример.** Добавить новые книги в таблицу **book**. У этих книг в таблице **supply** значения столбца **amount** не равно 0.

**Фрагмент логической схемы базы данных:**



**Изменяемая таблица:**

book	
PK	book_id
	title
FK	author_id
FK	genre_id
	price
	amount



# Поставка товара, операция 3, часть 2

**Пример.** Добавить новые книги в таблицу **book**. У этих книг в таблице **supply** значения столбца **amount** не равно 0.

**Шаг 1.** Отобрать новые книги из таблицы **supply**, указав код их автора.

**Условие соединения:**

```
author  
INNER JOIN supply ON author.name_author = supply.author
```

# Поставка товара, операция 3, часть 2

**Пример.** Добавить новые книги в таблицу **book**. У этих книг в таблице **supply** значения столбца **amount** не равно 0.

**Шаг 1.** Отобрать новые книги из таблицы **supply**, указав код их автора.

**Запрос на выборку:**

```
SELECT
    title, author_id, price, amount
FROM
    author
    INNER JOIN supply ON author.name_author = supply.author
WHERE amount != 0
```

# Поставка товара, операция 3, часть 2

**Пример.** Добавить новые книги в таблицу **book**. У этих книг в таблице **supply** значения столбца **amount** не равно 0.

**Шаг 2.** Добавить отобранные записи в таблицу.

**INSERT INTO**

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

**SELECT**

title, author\_id, price, amount

**FROM**

author

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

**WHERE** amount != 0

# Поставка товара, операция 3, часть 3

**Пример.** Для тех новых книг, авторы которых уже были в таблице **book**, выбрать любой жанр (например, с максимальным **id**), в котором он пишет.

**Шаг 1.** Для каждого автора найти жанр, в котором он пишет.

```
WITH get_author_genre(author_id, genre_id)
AS (
    SELECT author_id, MAX(genre_id)
    FROM book
    GROUP BY author_id
)
SELECT * FROM get_author_genre;
```

# Поставка товара, операция 3, часть 3

**Пример.** Для тех новых книг, авторы которых уже были в таблице **book**, выбрать любой жанр (например, с максимальным **id**), в котором он пишет.

**Шаг 2.** Обновить данные в таблице.

```
WITH get_author_genre(author_id, genre_id)
AS (
    SELECT author_id, MAX(genre_id)
    FROM book
    GROUP BY author_id
)
UPDATE
    book
    INNER JOIN get_author_genre USING(author_id)
SET
    book.genre_id = get_author_genre.genre_id
WHERE book.genre_id IS NULL
```



# Поставка товара, операция 3, часть 3

**Пример.** Для тех новых книг, авторов которых в таблице **book** не было, установить жанр «вручную».

Например, с помощью запроса:

**UPDATE**

book

**INNER JOIN** author **USING** (author\_id)

**SET**

genre\_id = (**SELECT** genre\_id  
                  **FROM** genre  
                  **WHERE** name\_genre = 'указать\_жанр')

**WHERE**

title = 'указать название' **AND** name\_author = 'указать автора';

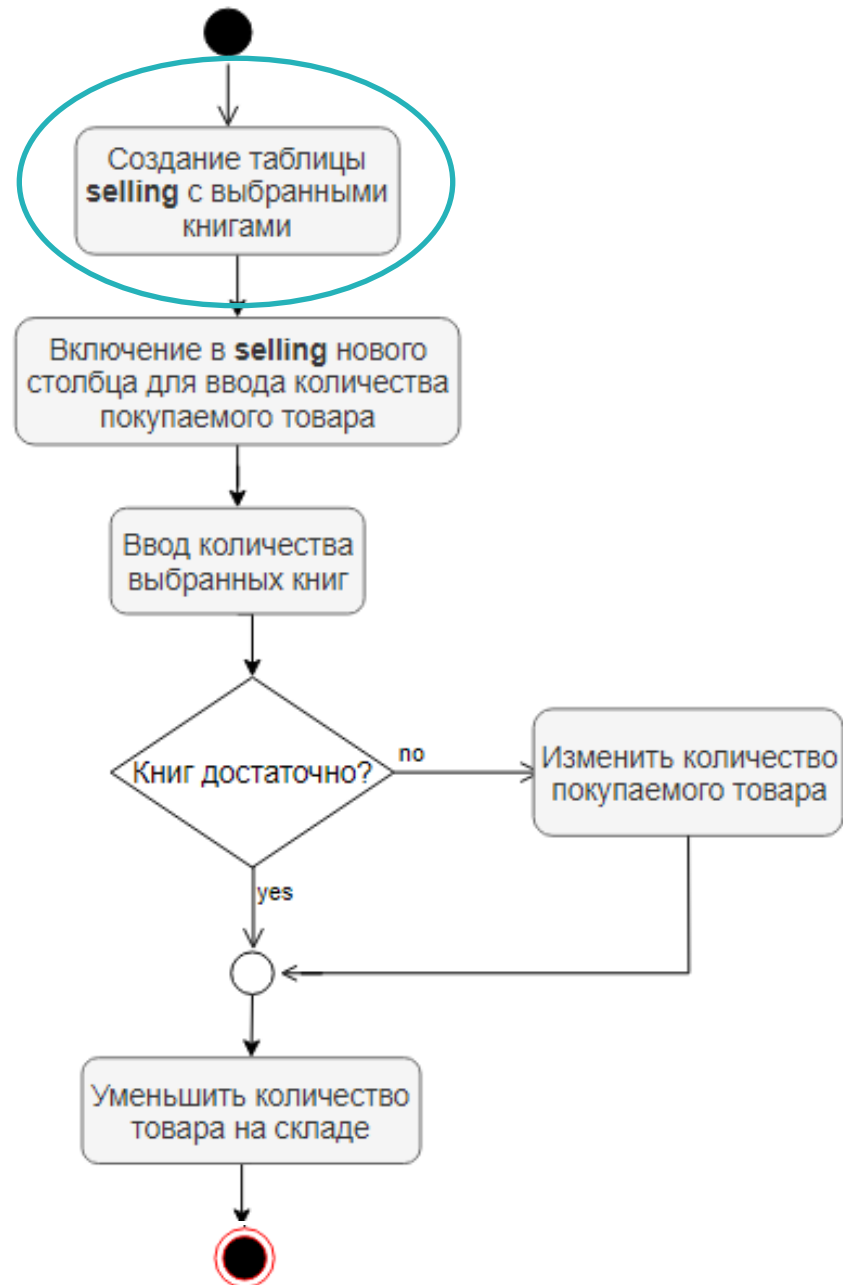
# Продажа и учет товара

На этапе **продажи и учета товара** необходимо

1. Отобрать книги по критериям, указанных пользователем, результат сохранить в виде таблицы.
2. Добавить в таблицу столбец, в котором пользователь будет указывать количество нужных ему книг;
3. Проверить, есть ли нужное количество книг на складе, если покупатель указал больше доступного количества, изменить покупку – отдать покупателю весь остаток.
4. Уменьшить количество экземпляров книг на значения, указанные в покупке.

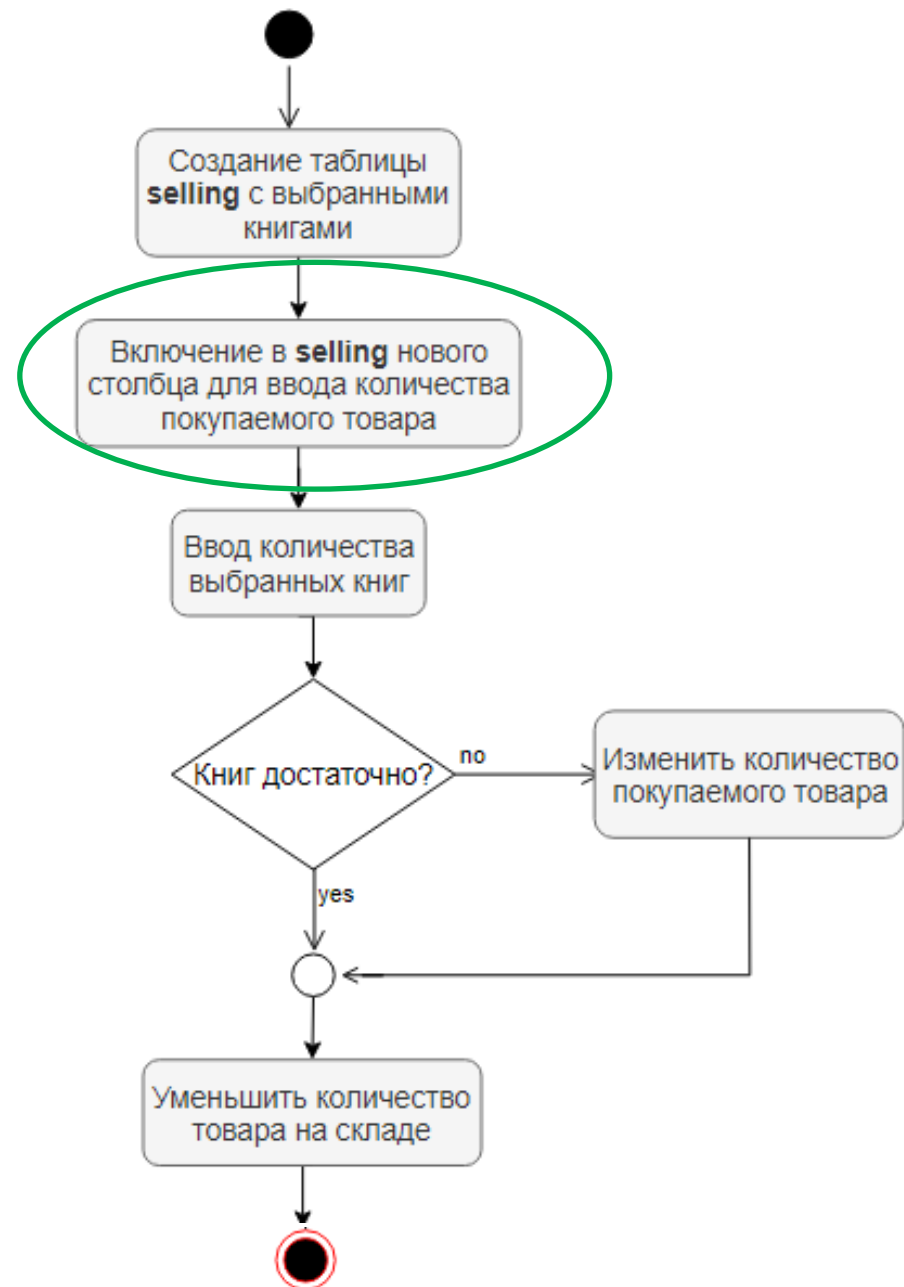
# Получение нового товара

Запрос на создание таблицы

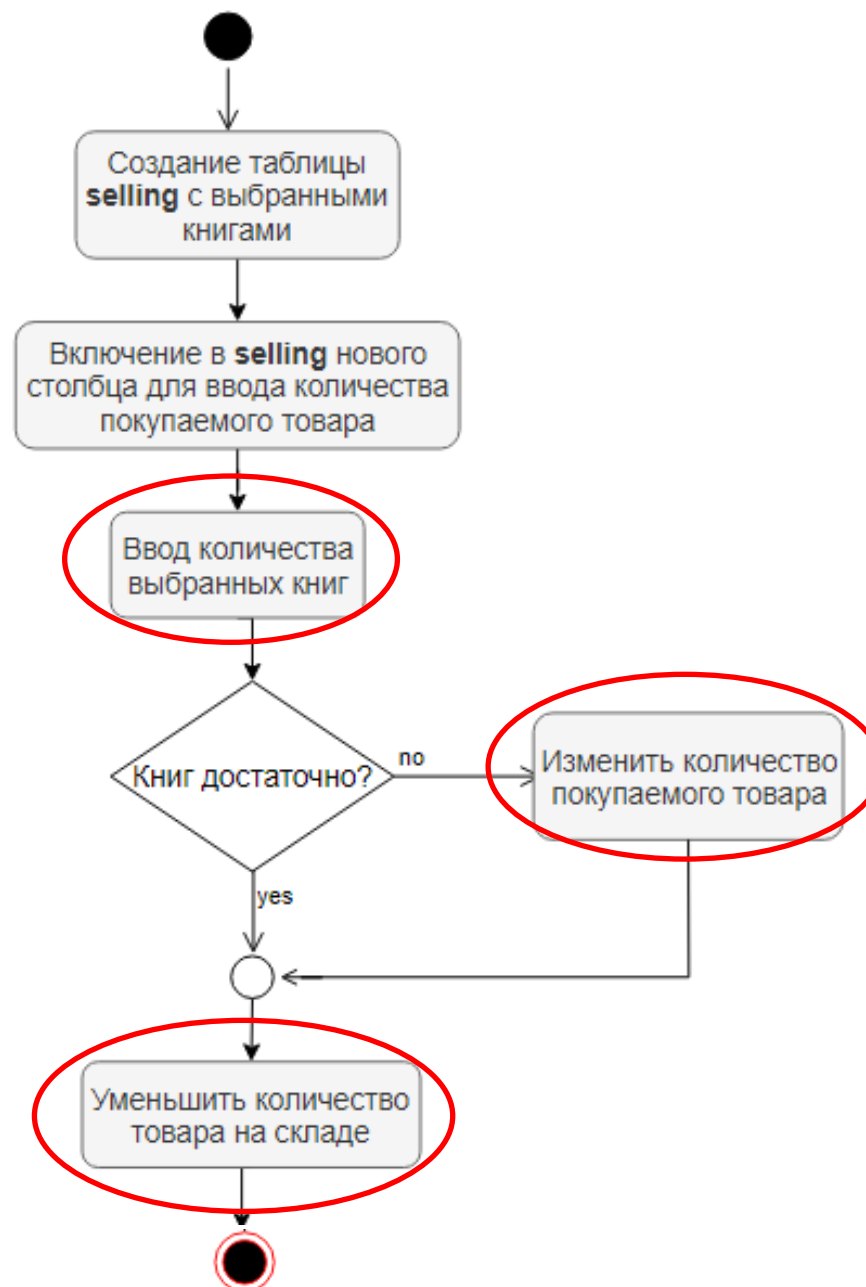


# Получение нового товара

Запрос на изменение  
структуры таблицы



# Получение нового товара

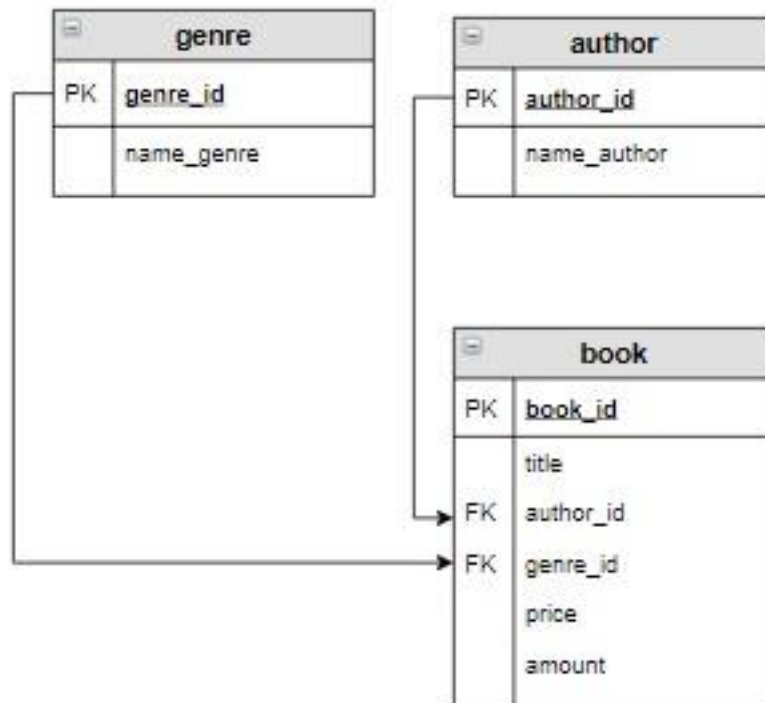


Запрос на обновление

# Продажа и учет товара, операция 1

**Пример.** Создать новую таблицу, в которую занести книги (код книги, название книги, ее автора, жанр, цену книги), которые предполагает купить покупатель.

**Схема данных:**



# Продажа и учет товара, операция 1

**Пример.** Создать новую таблицу, в которую занести книги (код книги, название книги, ее автора, жанр, цену книги), которые предполагает купить покупатель.

Шаг 1. Отобрать нужные книги с помощью запроса на выборку.

```
SET @tit ="a";      SET @auth ="";   SET @gen ="o";

SELECT book_id, title, name_author, name_genre, price
FROM
  author
  JOIN book USING (author_id)
  JOIN genre USING (genre_id)
WHERE
  title LIKE CONCAT("%",@tit,"%")
  AND name_author LIKE CONCAT("%", @auth, "%")
  AND name_genre LIKE CONCAT("%", @gen, "%");
```

# Продажа и учет товара, операция 1

**Пример.** Создать новую таблицу, в которую занести книги (код книги, название книги, ее автора, жанр, цену книги), которые предполагает купить покупатель.

Шаг 2. Создать на основе запроса таблицу **selling** .

```
SET @tit ="a";      SET @auth ="";   SET @gen ="o";  
CREATE TABLE selling AS  
SELECT book_id, title, name_author, name_genre, price  
FROM  
    author  
    JOIN book USING (author_id)  
    JOIN genre USING (genre_id)  
WHERE  
    title LIKE CONCAT("%",@tit,"%")  
    AND name_author LIKE CONCAT("%", @auth, "%")  
    AND name_genre LIKE CONCAT("%", @gen, "%");
```



# Продажа и учет товара, операция 1

**Пример.** Создать новую таблицу, в которую занести книги (код книги, название книги, ее автора, жанр, цену книги), которые предполагает купить покупатель.

**selling**

book_id	title	name_author	name_genre	price
1	Мастер и Маргарита	Булгаков М.А.	Роман	670.99
2	Белая гвардия	Булгаков М.А.	Роман	540.50
4	Братья Карамазовы	Достоевский Ф.М.	Роман	799.01
8	Лирика	Пастернак Б.Л.	Поэзия	518.99
9	Доктор Живаго	Пастернак Б.Л.	Роман	380.80

Affected rows: 5

# Продажа и учет товара, операция 2

**Пример.** В таблицу **selling** включить новый столбец **buy**, в который покупатель будет вносить нужное ему количество книг.

```
ALTER TABLE selling ADD buy INT;
```

# Продажа и учет товара, операция 2

**Пример.** В таблицу **selling** включить новый столбец **buy**, в который покупатель будет вносить нужное ему количество книг.

**selling**

book_id	title	name_author	name_genre	price	buy
1	Мастер и Маргарита	Булгаков М.А.	Роман	670.99	None
2	Белая гвардия	Булгаков М.А.	Роман	540.50	None
4	Братья Карамазовы	Достоевский Ф.М.	Роман	799.01	None
8	Лирика	Пастернак Б.Л.	Поэзия	518.99	None
9	Доктор Живаго	Пастернак Б.Л.	Роман	380.80	None

Affected rows: 5

# Продажа и учет товара, операция 2

**Пример.** Указать, что покупатель хочет купить

```
UPDATE selling  
SET buy = 7  
WHERE book_id = 1;
```

```
UPDATE selling  
SET buy = 1  
WHERE book_id = 4;
```

```
UPDATE selling  
SET buy = 3  
WHERE book_id = 8;
```

# Продажа и учет товара, операция 2

**Пример.** Указать, что покупатель хочет купить

**selling** (до выполнения запроса)

book_id	title	name_author	name_genre	price	buy
1	Мастер и Маргарита	Булгаков М.А.	Роман	670.99	None
2	Белая гвардия	Булгаков М.А.	Роман	540.50	None
4	Братья Карамазовы	Достоевский Ф.М.	Роман	799.01	None
8	Лирика	Пастернак Б.Л.	Поэзия	518.99	None
9	Доктор Живаго	Пастернак Б.Л.	Роман	380.80	None

Affected rows: 5

**selling** (после выполнения запроса)

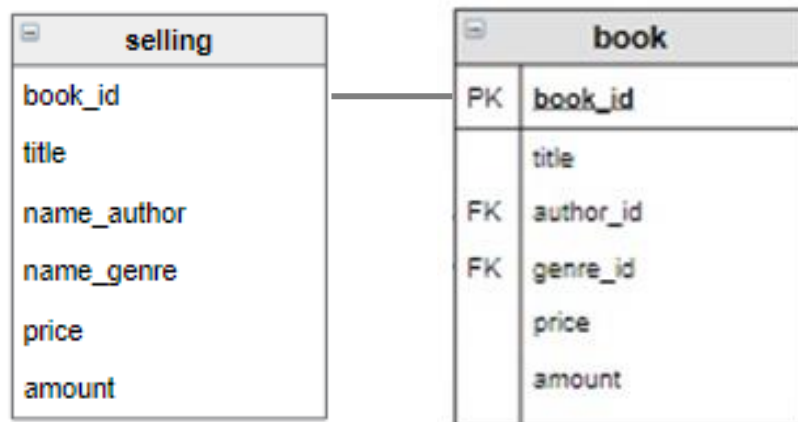
book_id	title	name_author	name_genre	price	buy
1	Мастер и Маргарита	Булгаков М.А.	Роман	670.99	7
2	Белая гвардия	Булгаков М.А.	Роман	540.50	None
4	Братья Карамазовы	Достоевский Ф.М.	Роман	799.01	1
8	Лирика	Пастернак Б.Л.	Поэзия	518.99	3
9	Доктор Живаго	Пастернак Б.Л.	Роман	380.80	None

Affected rows: 5

# Продажа и учет товара, операция 3

**Пример.** Проверить, есть ли нужное количество книг на складе, если покупатель указал больше доступного количества, изменить покупку – отдать покупателю весь остаток.

**Логическая схема базы данных:**



# Продажа и учет товара, операция 3

**Пример.** Проверить, есть ли нужное количество книг на складе, если покупатель указал больше доступного количества, изменить покупку – отдать покупателю весь остаток.

**UPDATE**

selling

**JOIN** book **USING** (book\_id)

**SET**

buy = **IF**(buy > amount, amount, buy)

**WHERE**

buy **IS NOT NULL**;

# Продажа и учет товара, операция 3

**Пример.** Проверить, есть ли нужное количество книг на складе, если покупатель указал больше доступного количества, изменить покупку – отдать покупателю весь остаток.

**selling** (до выполнения запроса)

book_id	title	name_author	name_genre	price	buy
1	Мастер и Маргарита	Булгаков М.А.	Роман	670.99	7
2	Белая гвардия	Булгаков М.А.	Роман	540.50	None
4	Братья Карамазовы	Достоевский Ф.М.	Роман	799.01	1
8	Лирика	Пастернак Б.Л.	Поэзия	518.99	2
9	Доктор Живаго	Пастернак Б.Л.	Роман	380.80	None

book

**selling** (после выполнения запроса)

book_id	title	name_author	name_genre	price	buy
1	Мастер и Маргарита	Булгаков М.А.	Роман	670.99	3
2	Белая гвардия	Булгаков М.А.	Роман	540.50	None
4	Братья Карамазовы	Достоевский Ф.М.	Роман	799.01	1
8	Лирика	Пастернак Б.Л.	Поэзия	518.99	2
9	Доктор Живаго	Пастернак Б.Л.	Роман	380.80	None

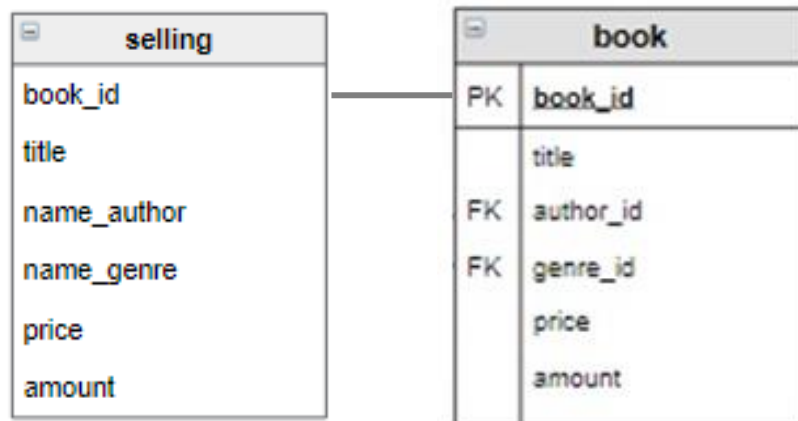
book_id	title	author_id	genre_id	price	amount
1	Мастер и Маргарита	1	1	670.99	3
2	Белая гвардия	1	1	540.50	12
3	Идиот	2	1	437.11	13
4	Братья Карамазовы	2	1	799.01	3
5	Игрок	2	1	480.50	10
6	Стихотворения и поэмы	3	2	650.00	15
7	Черный человек	3	2	570.20	12
8	Лирика	4	2	518.99	2
9	Доктор Живаго	4	1	380.80	4
		5	2	255.90	4
		6	1	599.99	5



# Продажа и учет товара, операция 4

**Пример.** Уменьшить количество экземпляров книг на значения, указанные в покупке.

**Логическая схема базы данных:**



# Продажа и учет товара, операция 4

**Пример.** Уменьшить количество экземпляров книг на значения, указанные в покупке.

**UPDATE**

selling

**JOIN** book **USING** (book\_id)

**SET**

amount = amount - buy

**WHERE**

buy **IS NOT NULL;**

# Продажа и учет товара, операция 4

**Пример.** Уменьшить количество экземпляров книг на значения, указанные в покупке.

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	2	255.90	4
11	Остров сокровищ	6	1	599.99	5

book_id	title	name_author	name_genre	price	buy
1	Мастер и Маргарита	Булгаков М.А.	Роман	670.99	3
2	Белая гвардия	Булгаков М.А.	Роман	540.50	None
4	Братья Карамазовы	Достоевский Ф.М.	Роман	799.01	1
8	Лирика	Пастернак Б.Л.	Поэзия	518.99	2
9	Доктор Живаго	Пастернак Б.Л.	Роман	380.80	None

book_id	title	author_id	genre_id	price	amount
1	Мастер и Маргарита	1	1	670.99	0
2	Белая гвардия	1	1	540.50	12
3	Идиот	2	1	437.11	13
4	Братья Карамазовы	2	1	799.01	2
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	0
9	Доктор Живаго	4	1	380.80	4
10	Стихотворения и поэмы	5	2	255.90	4
11	Остров сокровищ	6	1	599.99	5

# Заказ товара

Периодически на складе формируют заказ на товар двух категорий:

1. имеющиеся товары, количество которых меньше допустимого остатка;
2. новые товары, которых на складе еще не было.

# Заказ товара, операция 1

**Пример.** Создать новую таблицу, в которую занести книги (название книги, ее автора), количество которых меньше 3. Также указать количество книг в заказе. Это количество вычислить как среднее значение книг на складе.

**Шаг 2.** Отобрать нужные книги, указать необходимое количество.

```
@avg_amount := (SELECT AVG(amount)
                 FROM book);
```

**SELECT**

# Заказ товара, операция 1

**Пример.** Создать новую таблицу, в которую занести книги (название книги, ее автора), количество которых меньше 3. Также указать количество книг в заказе. Это количество вычислить как среднее значение книг на складе.

**Шаг 2.** Отобрать нужные книги, указать необходимое количество.

```
@avg_amount := (SELECT AVG(amount)
                 FROM book);
```

```
SELECT
    title, name_author,
    ROUND(@avg_amount) AS amount
FROM
    book
    JOIN author USING(author_id)
WHERE amount < 3;
```

# Заказ товара, операция 1

**Пример.** Создать новую таблицу, в которую занести книги (название книги, ее автора), количество которых меньше 3. Также указать количество книг в заказе. Это количество вычислить как среднее значение книг на складе.

**Шаг 3.** Создать таблицу на основе запроса

```
@avg_amount := (SELECT AVG(amount)
                  FROM book);
```

```
CREATE ordering AS
SELECT
    title, name_author,
    ROUND(@avg_amount) AS amount
FROM
    book
    JOIN author USING(author_id)
WHERE amount < 3;
```

# Заказ товара, операция 1

**Пример.** Создать новую таблицу, в которую занести книги (название книги, ее автора), количество которых меньше 3. Также указать количество книг в заказе. Это количество вычислить как среднее значение книг на складе.

## ordering

title	name_author	amount
Мастер и Маргарита	Булгаков М.А.	7
Братья Карамазовы	Достоевский Ф.М.	7
Лирика	Пастернак Б.Л.	7



# Заказ товара, операция 2

**Пример.** В заказ добавить новые книги, которых нет на складе, для них тоже в качестве заказываемого количества указать среднее значение количества экземпляров книг на складе.

```
@avg_amount := (SELECT AVG(amount)
                 FROM book);
```

```
INSERT INTO ordering
VALUES
```

```
("Война и Мир", "Толстой Л.Н.", ROUND(@avg_amount)),
("Анна Каренина", "Толстой Л.Н.", ROUND(@avg_amount));
```

# Заказ товара, операция 2

**Пример.** В заказ добавить новые книги, которых нет на складе, для них тоже в качестве заказываемого количества указать среднее значение количества экземпляров книг на складе.

## ordering

title	name_author	amount
Мастер и Маргарита	Булгаков М.А.	7
Братья Карамазовы	Достоевский Ф.М.	7
Лирика	Пастернак Б.Л.	7
Война и Мир	Толстой Л.Н.	7
Анна Каренина	Толстой Л.Н.	7

Последовательное применение запросов корректировки позволяет реализовать некоторый процесс, который должны «проходить» данные в выбранной предметной области.

Например, для склада это

1. Поставка товара.
2. Продажа товара.
3. Заказ товара.

# Резюме

Каждый этап включает последовательность шагов, которые выполняются при некотором условии.

Каждый этап включает последовательность шагов, которые выполняются при некотором условии.

Каждый шаг реализуется одним или несколькими запросами.

Для описания каждого этапа можно использовать либо словесное описание, либо диаграмму.



Спасибо за внимание!