Запросы на выборку и корректировку, связанные таблицы

Реляционная модель

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

Запросы на выборку, связанные таблицы

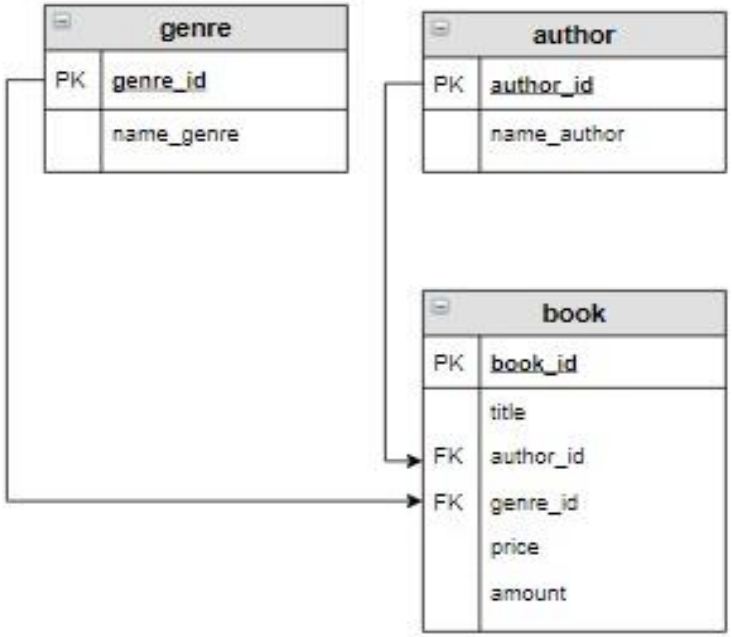
В запросах на выборку можно использовать связанные таблицы.

При реализации таких запросов рекомендуется придерживаться следующего **алгоритма**:

1. Отобразить фрагмент логической схемы базы данных с таблицами, которые участвуют в запросе.
2. Описать связи между этими таблицами.
3. Создать запрос, связи между таблицами разместить в разделе FROM.

Предметная область «Книжный склад»

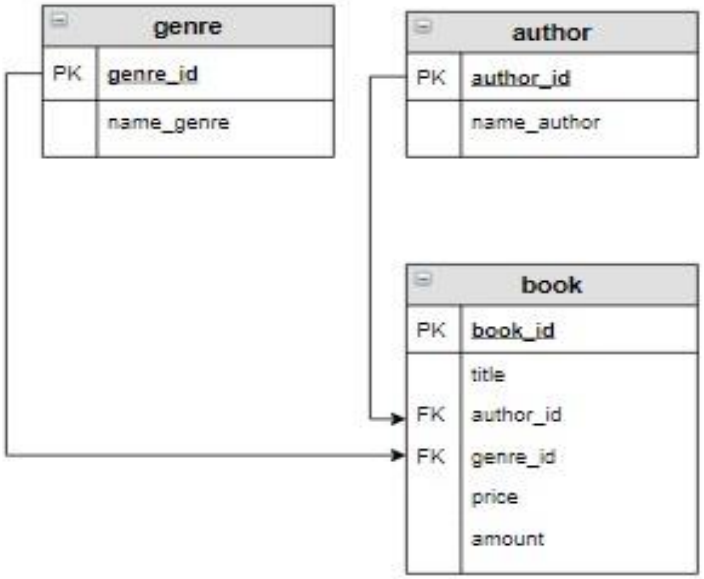
Логическая схема:



Выборка данных, пример

Пример. Вывести информацию о тех книгах, их авторах, жанрах и цене, количество экземпляров которых больше 5.

Шаг 1. Отобразить логическую схему и описать связи




```
author
    INNER JOIN book ON author.author_id = book.author_id
    INNER JOIN genre ON genre.genre_id = book.genre_id
```

Выборка данных, пример

Пример. Вывести информацию о тех книгах, их авторах, жанрах и цене, количество экземпляров которых больше 5.

Шаг 2. Реализовать 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 amount > 5;
```

Выборка данных, пример

Пример. Вывести информацию о тех книгах, их авторах, жанрах и цене, количество экземпляров которых больше 5.

title	name_author	name_genre	price	amount
Идиот	Достоевский Ф.М.	Роман	460.00	10
Игрок	Достоевский Ф.М	Роман	480.50	10
Стихотворения и поэмы	Есенин С.А.	Поэзия	650.00	15
Черный человек	Есенин С.А.	Поэзия	570.20	6

Соединение вложенных запросов и таблиц

Вложенные запросы можно использовать в операторах соединения как обычную таблицу.

Обязательное требование – наличие имени у вложенного запроса.

```
таблица
JOIN (
    SELECT ...
    ) имя_вложенного_запроса
ON условие_соединения
```

```
(
 SELECT ...
) имя_вложенного_запроса
JOIN таблица
ON условие_соединения
```

Пример: таблицы и вложенные запросы

Пример. Вывести авторов, пишущих книги в самом популярном жанре. Самым популярным считать жанр, общее количество экземпляров книг которого на складе максимально.

Шаг 1. Вычислим количество экземпляров книг в каждом жанре, отсортируем результат по убыванию общего количества и выделим первую строку.

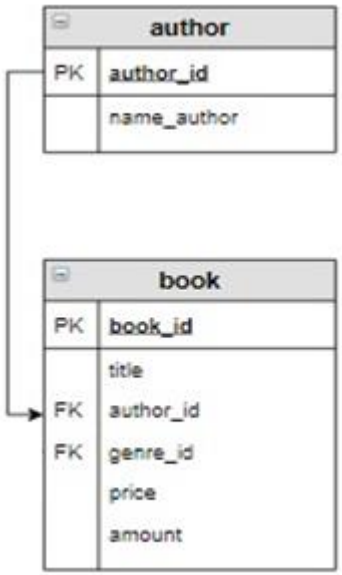
```
SELECT
     genre_id, SUM(amount)
FROM book
GROUP BY genre_id
ORDER BY 2 DESC
LIMIT 1
```

genre_id	SUM(amount)
001	18

Пример: таблицы и вложенные запросы

Пример. Вывести авторов, пишущих книги в самом популярном жанре. Самым популярным считать жанр, общее количество экземпляров книг которого на складе максимально.

Шаг 2. Выведем всех авторов и в каких жанрах они пишут книги



Пример: таблицы и вложенные запросы

Пример. Вывести авторов, пишущих книги в самом популярном жанре. Самым популярным считать жанр, общее количество экземпляров книг которого на складе максимально.

Шаг 2. Выведем всех авторов и в каких жанрах они пишут книги, записи в результате не должны повторяться.

name_author	genre_id
Булгаков М.А.	001
Достоевский Ф.М.	001
Есенин С.А.	002

```
SELECT DISTINCT
    name_author, genre_id
FROM
    author
    INNER JOIN book USING(author_id)
```

Пример: таблицы и вложенные запросы

```
SELECT name_author
FROM
    (SELECT
         genre_id, SUM(amount)
     FROM book
     GROUP BY genre_id
     ORDER BY sum_amount DESC
     LIMIT 1
     ) get_genre
     INNER JOIN
    (SELECT DISTINCT
         name_author, genre_id
     FROM author
     INNER JOIN book USING(author_id)
    ) get_author
    USING(genre_id);
```

Пример: таблицы и вложенные запросы

Пример. Вывести авторов, пишущих книги в самом популярном жанре. Самым популярным считать жанр, общее количество экземпляров книг которого на складе максимально.

name_author
Булгаков М.А.
Достоевский Ф.М.

Табличные выражения в соединениях

Табличные выражения также можно использовать в операторах соединения как обычную таблицу.

В этом случае в условии соединения используются имена столбцов, указанные в заготовке табличного выражения.

Табличные выражения в соединениях

Синтаксис запроса:

```
WITH табличное_выражение(поле_тв, ...)
AS(
     SELECT
         ...
    )
SELECT ...
FROM
    таблица
    JOIN табличное_выражение
    ON таблица.поле_таблицы = табличное_выражение.поле_тв
...
```

Пример: таблицы и табличные выражения

Пример. Вывести авторов, пишущих книги в самом популярном жанре. Самым популярным считать жанр, общее количество экземпляров книг которого на складе максимально.

Шаг 1. Вычислим количество экземпляров книг в каждом жанре, отсортируем результат по убыванию общего количества и выделим первую строку.

```
WITH get_genre(genre_id, sum_amount)
AS(
    SELECT genre_id, SUM(amount)
    FROM book
    GROUP BY genre_id
    ORDER BY sum_amount DESC
    LIMIT 1 )
```

Пример: таблицы и табличные выражения

Пример. Вывести авторов, пишущих книги в самом популярном жанре. Самым популярным считать жанр, общее количество экземпляров книг которого на складе максимально.

Шаг 2. Выведем всех авторов и в каких жанрах они пишут книги, записи в результате не должны повторяться.

```
WITH get_author(name_author, genre_id)
AS(
    SELECT DISTINCT
        name_author, genre_id
    FROM
       author
       INNER JOIN book USING(author_id)
)
```

Пример: таблицы и табличные выражения

```
WITH get_genre(genre_id, sum_amount)
AS(
    SELECT genre_id, SUM(amount)
    FROM book
    GROUP BY genre_id
    ORDER BY sum_amount DESC
    LIMIT 1
),
get_author(name_author, genre_id)
AS(
    SELECT DISTINCT name_author, genre_id
    FROM
        author
        INNER JOIN book USING(author_id)
)
SELECT name_author
FROM
    get_genre
    INNER JOIN get_author USING(genre_id);
```

Пример: таблицы и табличные выражения

Пример. Вывести авторов, пишущих книги в самом популярном жанре. Самым популярным считать жанр, общее количество экземпляров книг которого на складе максимально.

name_author
Булгаков М.А.
Достоевский Ф.М.

Запросы корректировки

К запросам корректировки данных относятся:

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

Обновление, связанные таблицы

В запросах **на обновление** можно использовать связанные таблицы.

Причем исправления можно вносить в любую из них.

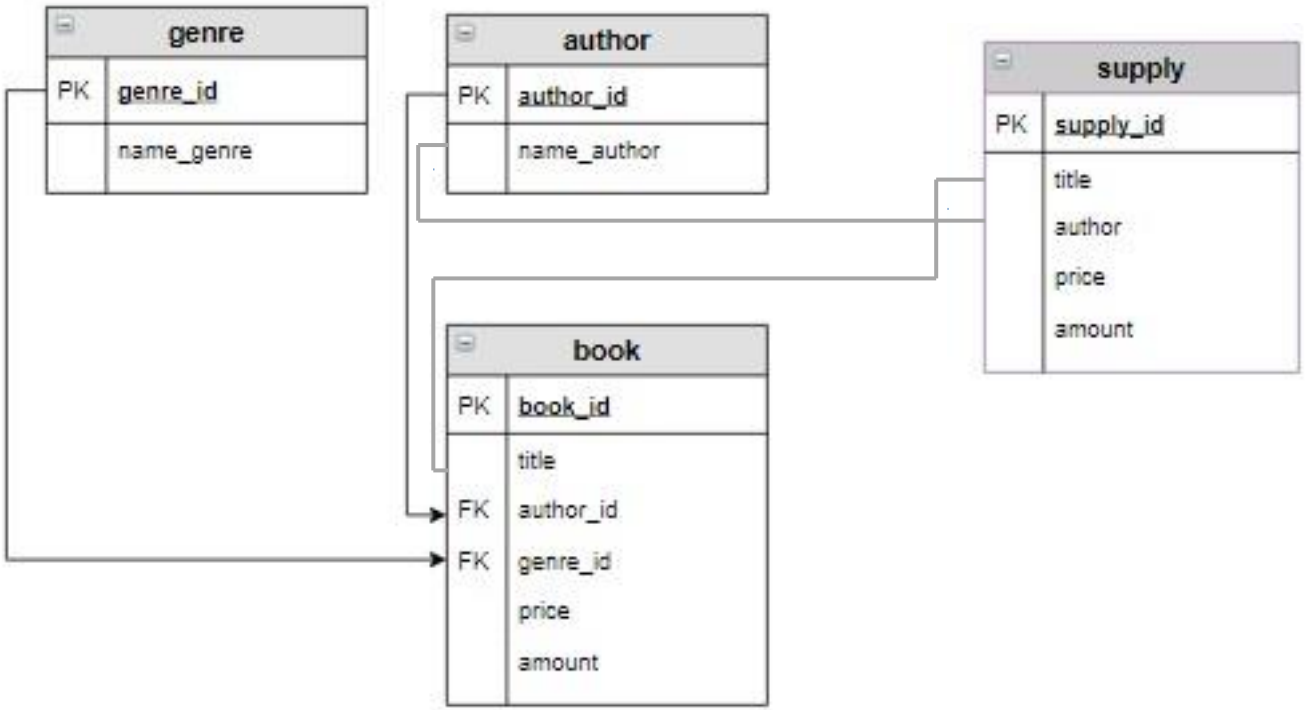
Синтаксис запроса:

```
UPDATE
     таблица_1 ...
     JOIN таблица_2 ON условие_связывания
     ...
SET
    ...
WHERE
    ...;
```

Описание предметной области

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

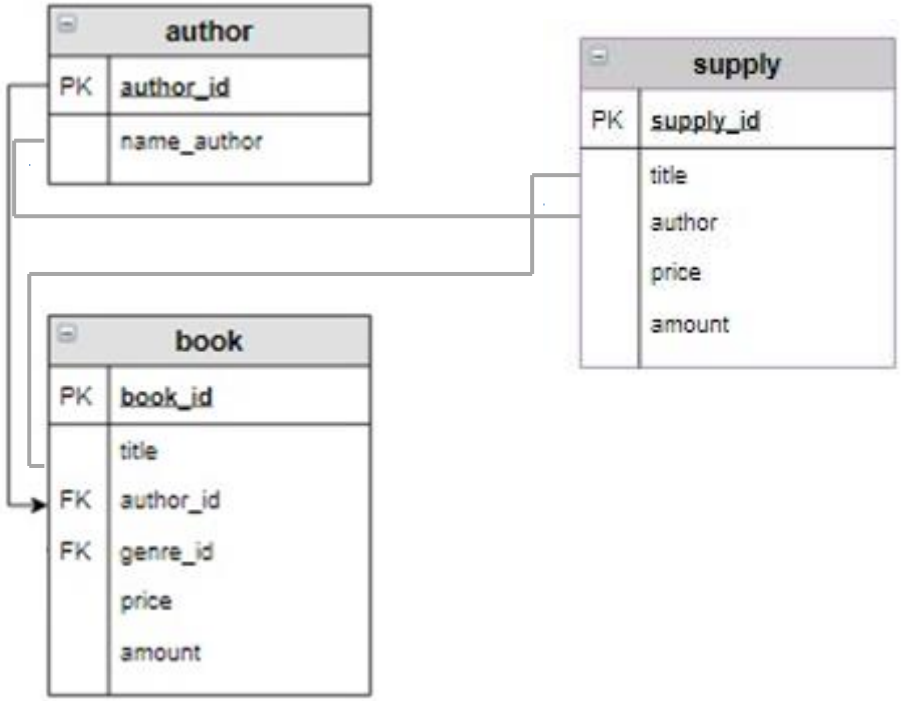
Логическая схема базы данных:



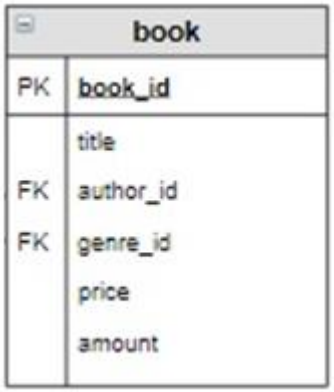
Пример

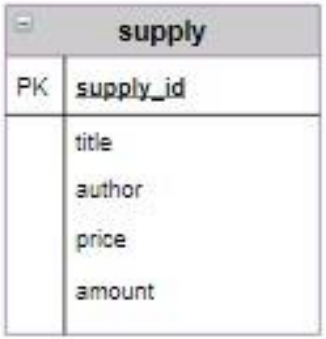
Пример. Для книг, которые уже есть на складе по той же цене, что и в поставке, увеличить количество на значение, указанное в поставке, а также обнулить количество этих книг в поставке.

Фрагмент логической схемы базы данных



Изменяемые таблицы:





Пример

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.

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
```

Пример

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;
```

Табличные выражения и UPDATE

В запросах на обновление можно использовать табличные выражения.

Синтаксис такого запроса:

```
WITH табличное_выражение(…)
AS (
    SELECT …
    ...
)
UPDATE
    таблица
    JOIN табличное_выражение ON …
SET
    поле = выражение,
    …
WHERE …
```

Пример, табличные выражения и UPDATE

Пример. Для всех книг изменить цену на максимальную цену книги, написанную автором книги.

Шаг 1. Найти максимальную цену книги каждого автора. Просмотреть результат.

```
WITH get_max_price(author_id, max_price)
AS(
     SELECT author_id, MAX(price)
     FROM book
     GROUP BY author_id
)
SELECT * FROM get_max_price;
```

```
Query result:
+-----------+-----------+
| author_id | max_price |
+-----------+-----------+
| 1         | 670.99    |
| 2         | 799.01    |
| 3         | 650.00    |
| 4         | 518.99    |
+-----------+-----------+
Affected rows: 4
```

Пример, табличные выражения и UPDATE

Пример. Для всех книг изменить цену на максимальную цену книги, написанную автором книги.

Шаг 2. Обновить данные в таблице.

```
WITH get_max_price(author_id, max_price)
AS(
     SELECT author_id, MAX(price)
     FROM book
     GROUP BY author_id
)
UPDATE
     book
     JOIN get_max_price USING(author_id)
SET price = max_price;
```

Пример, табличные выражения и UPDATE

Пример. Для всех книг изменить цену на максимальную цену книги, написанную автором книги.

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	670.99	5
3	Идиот	2	1	799.01	10
4	Братья Карамазовы	2	1	799.01	3
5	Игрок	2	1	799.01	10
6	Стихотворения и поэмы	3	2	650.00	15
7	Черный человек	3	2	650.00	6
8	Лирика	4	2	518.99	2

Affected rows: 8

Запрос на добавление

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

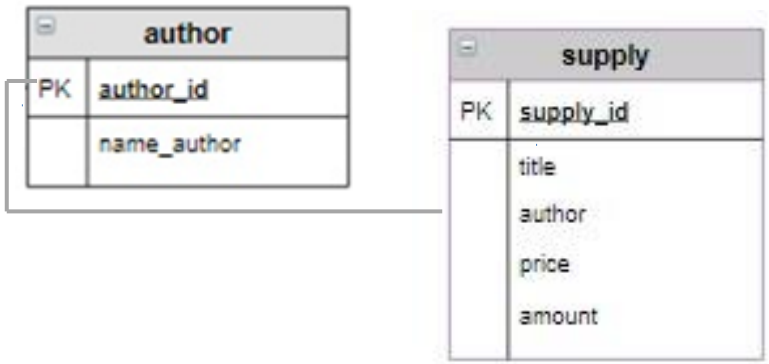
Синтаксис запроса:

```
INSERT INTO
     таблица (список_полей)
SELECT список_полей_из_таблиц_1_и_2
FROM
    таблица_1
    JOIN таблица_2 ON условие_соединения
...
```

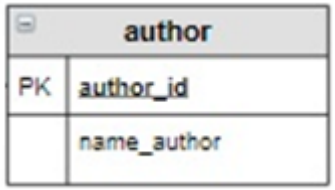
Пример, запрос на добавление

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

Фрагмент логической схемы базы данных:



Изменяемая таблица:



Пример, запрос на добавление

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
```

Пример, запрос на добавление

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 = su
WHERE name_author IS NULL;
```

```
Query result:
+----------------+
| author         |
+----------------+
| Стивенсон Р.Л. |
+----------------+
Affected rows: 1
```

Пример, запрос на добавление

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;
```

Пример, запрос на добавление

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

```
Query result:
+-----------+-------------------+
| author_id | name_author       |
+-----------+-------------------+
| 1         | Булгаков М.А.     |
| 2         | Достоевский Ф.М.  |
| 3         | Есенин С.А.       |
| 4         | Пастернак Б.Л.    |
| 5         | Лермонтов М.Ю.    |
| 6         | Стивенсон Р.Л.    |
+-----------+-------------------+
Affected rows: 6
```

Добавление и табличные выражения

В запросах на добавление можно использовать табличные выражения.

Синтаксис запроса:

```
INSERT INTO таблица(поле_1, поле_2, ....)
WITH табличное_выражение(...)
AS(
   ...
 )
SELECT значение_поле_1, значение_поле_2, ....
 FROM
    табличное выражение ...
 ...;
```

Пример, запрос на добавление

Пример. Включить в таблицу **book** все "новые" книги из таблицы **supply**. То есть либо книги нового автора, либо книги, автор которых в таблице **book** есть, но название книги - встречается впервые.

Шаг 1. Отобрать все новые книги. Просмотреть результат

```
WITH get_new_book(title, author, price, amount)
AS(
    SELECT title, author, price, amount
    FROM supply
    WHERE (author, title) NOT IN ( SELECT author, title FROM book )
 )
SELECT *
FROM get_new_book;
```

```
Query result:
+-----------------+----------------+-------+--------+
| title           | author         | price | amount |
+-----------------+----------------+-------+--------+
| Доктор Живаго   | Пастернак Б.Л. | 380.80 | 4      |
| Остров сокровищ | Стивенсон Р.Л. | 599.99 | 5      |
+-----------------+----------------+-------+--------+
Affected rows: 2
```

Пример, запрос на добавление

Пример. Включить в таблицу **book** все "новые" книги из таблицы **supply**. То есть либо книги нового автора, либо книги, автор которых в таблице **book** есть, но название книги - встречается впервые.

Шаг 2. Добавить отобранные записи в таблицу.

```
INSERT INTO book(title, author_id, price, amount)
WITH get_new_book(title, author, price, amount)
AS(
    SELECT title, author, price, amount
    FROM supply
    WHERE (author, title) NOT IN ( SELECT author, title FROM book )
 )
SELECT title, author_id, price, amount
FROM
     get_new_book
     JOIN author ON get_new_book.author = author.author_name
```

Пример, запрос на добавление

Пример. Включить в таблицу **book** все "новые" книги из таблицы **supply**. То есть либо книги нового автора, либо книги, автор которых в таблице **book** есть, но название книги - встречается впервые.

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	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
9	Доктор Живаго	4	NULL	380.80	4
10	Остров сокровищ	6	NULL	599.99	5

Affected rows: 10

Запрос на создание таблицы

Новая таблица может быть создана на основе данных из нескольких таблиц.

Для этого используется запрос **SELECT**, результирующая таблица которого и будет новой таблицей базы данных.

При этом имена столбцов запроса становятся именами столбцов новой таблицы.

Запрос на создание таблицы

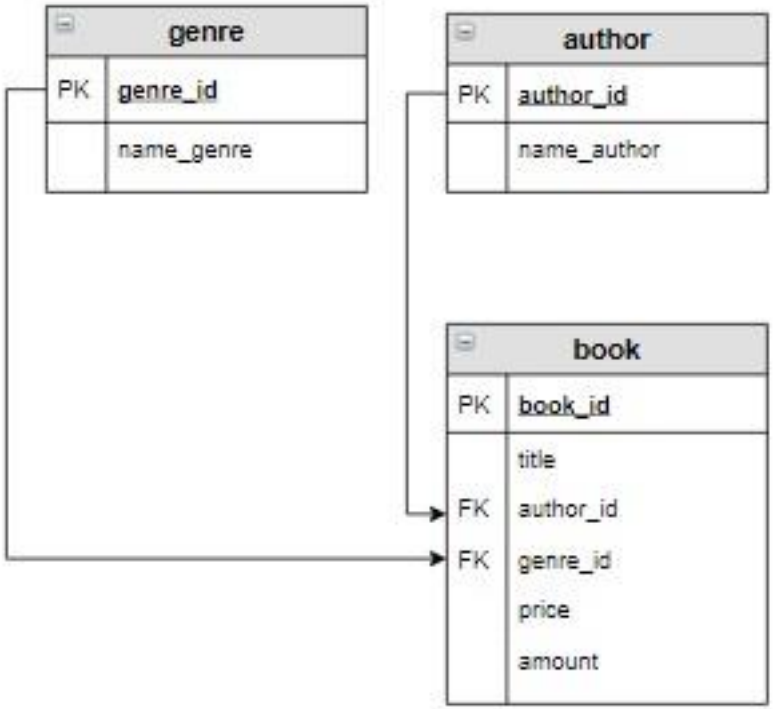
Запрос на создание новой таблицы имеет вид:

```
CREATE TABLE имя_таблицы AS
SELECT столбец_1, столбец_2, …
FROM
    таблица_1
    JOIN таблица_2 …
…
```

Пример, создание таблицы

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

Схема данных:



Пример, создание таблицы

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.

```
SELECT book_id, title, name_author, name_genre, price
FROM
    author
    JOIN book USING(author_id)
    JOIN genre USING(genre_id)
WHERE
     title LIKE "%a%"
     AND name_genre LIKE "%o%";
```

Пример, создание таблицы

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** .

```
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 "%a%"
     AND name_genre LIKE "%o%";
```

Пример, создание таблицы

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

Запрос на создание таблицы

Запрос на создание новой таблицы может включать табличные выражения.

Синтаксис такого запроса имеет вид:

```
CREATE TABLE новая_таблица AS
WITH табличное_выражение(результат, …)
AS(
 ...
 )
SELECT ..., результат, ...
FROM
    табличное выражение ...
...;
```

Пример, создание таблицы

Пример. Создать таблицу заказ **book_order**, в которую включить все книги из таблицы **book**, у которых количество экземпляров меньше среднего количества экземпляров книг своего автора.

Шаг 1. Найти среднее количество книг каждого автора.

```
WITH get_avg (author_id, avg_amount)
AS(
    SELECT author_id, CEIL(AVG(amount))
    FROM book
    GROUP BY author_id
)
SELECT * FROM get_avg;
```

```
Query result:
+-----------+------------+
| author_id | avg_amount |
+-----------+------------+
| 1         | 4          |
| 2         | 8          |
| 3         | 11         |
| 4         | 2          |
+-----------+------------+
Affected rows: 4
```

Пример, создание таблицы

Пример. Создать таблицу заказ **book_order**, в которую включить все книги из таблицы **book**, у которых количество экземпляров меньше среднего количества экземпляров книг своего автора.

Шаг 2. Выбрать необходимые данные из таблиц.

```
WITH get_avg (author_id, avg_amount)
AS(
    SELECT author_id, CEIL(AVG(amount))
    FROM book
    GROUP BY author_id
)
SELECT title, name_author, price, avg_amount
FROM
    book
    JOIN get_avg USING (author_id)
    JOIN author USING (author_id)
WHERE amount < avg_amount;
```

Пример, создание таблицы

Пример. Создать таблицу заказ **book_order**, в которую включить все книги из таблицы **book**, у которых количество экземпляров меньше среднего количества экземпляров книг своего автора.

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

```
CREATE TABLE book_order
WITH get_avg (author_id, avg_amount)
AS(
    SELECT author_id, CEIL(AVG(amount))
    FROM book
    GROUP BY author_id
)
SELECT title, name_author, price, avg_amount
FROM
    book
    JOIN get_avg USING (author_id)
    JOIN author USING (author_id)
WHERE amount < avg_amount;
```

Пример, создание таблицы

Пример. Создать таблицу заказ **book_order**, в которую включить все книги из таблицы **book**, у которых количество экземпляров меньше среднего количества экземпляров книг своего автора.

Query result:

title	name_author	price	avg_amount
Мастер и Маргарита	Булгаков М.А.	670.99	4
Братья Карамазовы	Достоевский Ф.М.	799.01	8
Черный человек	Есенин С.А.	570.20	11

Affected rows: 3

Изменение структуры таблицы

Для изменения структуры таблицы используется оператор **ALTER TABLE**.

С его помощью можно:

- вставить новый столбец,
- удалить существующий,
- переименовать столбец
- и пр.

Изменение структуры таблицы

Вставка нового столбца:

- после последнего:

```
ALTER TABLE таблица ADD имя_столбца тип;
```

- перед первым:

```
ALTER TABLE таблица ADD имя_столбца тип FIRST;
```

- после указанного:

```
ALTER TABLE таблица ADD имя_столбца тип
                    AFTER имя_столбца_1;
```

Изменение структуры таблицы

Удаление столбца:

- один столбец с заданным именем :

```
ALTER TABLE таблица DROP COLUMN имя_столбца;

ALTER TABLE таблица DROP имя_столбца;
```

- несколько столбцов:

```
ALTER TABLE таблица DROP имя_столбца,
                    DROP имя_столбца_1;
```

Изменение структуры таблицы

Для **переименования** столбца используется запрос:

- с сохранением типа столбца:

```
ALTER TABLE таблица
CHANGE имя_столбца новое_имя_столбца;
```

- с изменением типа:

```
ALTER TABLE таблица
CHANGE имя_столбца новое_имя_столбца НОВЫЙ_ТИП;
```

Пример, изменение структуры таблицы

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

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

Пример, изменение структуры таблицы

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

Запрос на удаление

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 ...
```

Запрос на удаление

При удалении записей из таблицы можно использовать информацию из других связанных с ней таблиц.

Другой вариант синтаксиса:

```
DELETE таблица_1
FROM
    таблица_1
    INNER JOIN таблица_2 ON ...
...;
```

Пример, удаление данных из таблицы

Пример. Удалить всех авторов из таблицы **author**, у которых есть книги, количество экземпляров которых меньше 3.

```
DELETE FROM author
USING
    author
    INNER JOIN book ON author.author_id = book.author_id
WHERE amount < 3;
```

Важно! Из базы данных будут удалены не только авторы, отвечающие заданному условию, но и все их книги (так как для столбца **author_id** из таблицы book установлено каскадное удаление записей)

Удаление и табличные выражения

В запросах на удаление можно использовать табличные выражения.

Синтаксис запроса:

```
WITH табличное_выражение(…)
AS (
    ...
)
DELETE таблица
FROM
    таблица
    JOIN табличное выражение …
…;
```

Пример, удаление записей

Пример. Удалить из таблицы **book** все книги, у которых количество экземпляров меньше среднего количества экземпляров книг своего автора.

Шаг 1. Найти среднее количество книг каждого автора.

```
WITH get_avg (author_id, avg_amount)
AS(
    SELECT author_id, AVG(amount)
    FROM book
    GROUP BY author_id
)
SELECT * FROM get_avg;
```

```
Query result:
+-----------+------------+
| author_id | avg_amount |
+-----------+------------+
| 1         | 4          |
| 2         | 8          |
| 3         | 11         |
| 4         | 2          |
+-----------+------------+
Affected rows: 4
```

Пример, удаление записей

Пример. Удалить из таблицы **book** все книги, у которых количество экземпляров меньше среднего количества экземпляров книг своего автора.

Шаг 2. Удалить отобранные записи из таблицы **book** .

```
WITH get_avg (author_id, avg_amount)
AS(
    SELECT author_id, AVG(amount)
    FROM book
    GROUP BY author_id
)
DELETE book
FROM
    book
    JOIN get_avg USING (author_id)
WHERE amount < avg_amount;
```

Пример, удаление записей

Пример. Удалить из таблицы **book** все книги, у которых количество экземпляров меньше среднего количества экземпляров книг своего автора.

```
Query result:
+---------+-----------------------+-----------+----------+--------+--------+
| book_id | title                 | author_id | genre_id | price  | amount |
+---------+-----------------------+-----------+----------+--------+--------+
| 2       | Белая гвардия         | 1         | 1        | 540.50 | 5      |
| 3       | Идиот                 | 2         | 1        | 460.00 | 10     |
| 5       | Игрок                 | 2         | 1        | 480.50 | 10     |
| 6       | Стихотворения и поэмы | 3         | 2        | 650.00 | 15     |
| 8       | Лирика                | 4         | 2        | 518.99 | 2      |
+---------+-----------------------+-----------+----------+--------+--------+
Affected rows: 5
```

Запрос на удаление таблицы

Из базы данных может быть удалена таблица целиком.

В этом случае синтаксис запроса имеет вид:

DROP таблица;

Резюме

В запросах на выборку можно использовать связанные таблицы.

При реализации таких запросов рекомендуется придерживаться следующего **алгоритма**:

1. Отобразить фрагмент логической схемы базы данных с таблицами, которые участвуют в запросе.
2. Описать связи между этими таблицами.
3. Создать запрос, связи между таблицами разместить в разделе **FROM**.

Резюме

В запросах на выборку в качестве связанных таблиц можно использовать вложенные запросы и табличные выражения.

Резюме

Запросы корректировки данных для связанных таблиц используются для

- добавления информации в таблицы;
- удаления записей из таблицы;
- корректировки данных в таблицах;
- создания и удаления таблиц;
- изменения структуры таблицы.

Спасибо за внимание!