**SQL синтаксис**

**Создание таблицы**

CREATE TABLE таблица

( id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(30) );

**Вставка значений**

INSERT INTO таблица(поле1, поле2)

VALUES (значение1, значение2);

**Выборка данных**

SELECT столбцы FROM таблица;

**Выборка вычисляемых значений**

SELECT title, amount, price,

IF(amount<4, price\*0.5, price\*0.7) AS sale

FROM book;

**Выборка по условию**

SELECT title, amount

FROM book

WHERE amount BETWEEN 5 AND 14;

SELECT title, price

FROM book

WHERE author IN ('Булгаков М.А.', 'Достоевский Ф.М.');

SELECT title

FROM book

WHERE title LIKE 'Б%';

**Выборка уникальных значений**

SELECT DISTINCT author

FROM book;

**Выборка данных, групповые функции MIN, MAX и AVG**

SELECT author,

MIN(price) AS Минимальная\_цена,

MAX(price) AS Максимальная\_цена,

ROUND(AVG(price),2) AS Средняя\_цена

FROM book

GROUP BY author

HAVING SUM(price \* amount) > 5000

ORDER BY Минимальная\_цена DESC;

WHERE и HAVING могут использоваться в одном запросе. При этом необходимо учитывать **порядок выполнения SQL запроса на выборку на СЕРВЕРЕ:**

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY

*Пример Вывести максимальную и минимальную цену книг каждого автора, кроме Есенина, количество экземпляров книг которого больше 10.*

SELECT author,

MIN(price) AS Минимальная\_цена,

MAX(price) AS Максимальная\_цена

FROM book

WHERE author <> 'Есенин С.А.'

GROUP BY author

HAVING SUM(amount) > 10;

*Пример Посчитать стоимость всех экземпляров каждого автора без учета книг «Идиот» и «Белая гвардия». В результат включить только тех авторов, у которых суммарная стоимость книг (без учета книг «Идиот» и «Белая гвардия») более 5000 руб. Вычисляемый столбец назвать Стоимость. Результат отсортировать по убыванию стоимости.*

SELECT author,

SUM(price \* amount) AS Стоимость

FROM book

WHERE title <> 'Идиот' AND title <> 'Белая гвардия'

GROUP BY author

HAVING Стоимость > 5000

ORDER BY author DESC;

**Вложенный запрос**

#### **Пример** Вывести информацию о тех книгах, количество которых меньше самого маленького среднего количества книг каждого автора.

**SELECT title, author, amount, price**

**FROM book**

**WHERE amount < ALL (**

**SELECT AVG(amount)**

**FROM book**

**GROUP BY author);**

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

| title | author | amount | price |

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

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

| Братья Карамазовы | Достоевский Ф.М. | 3 | 799.01 |

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

В этом примере amount < ALL ( 4.0000, 7.6667, 15.0000) означает, что подходят все amount меньше 4.0000 из списка.  
Таким образом, наш запрос отобрал все книги книги **Мастер и Маргарита и Братья Карамазовы**, количество которых равно 3.

#### **Пример** Вывести информацию о тех книгах, количество которых меньше самого большого среднего количества книг каждого автора.

**SELECT title, author, amount, price**

**FROM book**

**WHERE amount < ANY (**

**SELECT AVG(amount)**

**FROM book**

**GROUP BY author);**

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

| title | author | amount | price |

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

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

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

| Идиот | Достоевский Ф.М. | 10 | 460.00 |

| Братья Карамазовы | Достоевский Ф.М. | 3 | 799.01 |

| Игрок | Достоевский Ф.М. | 10 | 480.50 |

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

В этом примере amount < ANY ( 4.0000, 7.6667, 15.0000) означает, что подходят amount меньше самого большого значения из списка.

#### **Пример** Вывести информацию о книгах(автор, название, цена), цена которых меньше самой большой из минимальных цен, вычисленных для каждого автора.

**SELECT author, title, price**

**FROM book**

**WHERE price < ANY(**

**SELECT MIN(price)**

**FROM book**

**GROUP BY author);**

# Вложенный запрос после SELECT

Вложенный запрос может располагаться после ключевого слова SELECT. В этом случае результат выполнения запроса выводится в отдельном столбце результирующей таблицы. При этом результатом запроса может быть только одно значение, тогда оно будет повторяться во всех строках. Также вложенный запрос может использоваться в выражениях.

#### **Пример** Вывести информацию о книгах, количество экземпляров которых отличается от среднего количества экземпляров книг на складе более чем на 3, а также указать среднее значение количества экземпляров книг.

SELECT title, author, amount,

( SELECT AVG(amount)

FROM book

) AS Среднее\_количество

FROM book

WHERE abs(amount - (SELECT AVG(amount) FROM book)) >3;

--результат

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

| title | author | amount | Среднее\_количество |

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

| Мастер и Маргарита | Булгаков М.А. | 3 | 7.6667 |

| Братья Карамазовы | Достоевский Ф.М. | 3 | 7.6667 |

| Стихотворения и поэмы | Есенин С.А. | 15 | 7.6667 |

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

Во вложенном запросе вычисляется среднее количество экземпляров книг на складе. Значения столбца одинаковы во всех строках, поскольку вложенный запрос возвращает одно значение.

Среднее количество в виде дробного числа выглядит не очень правильно. Полученное значение можно округлить "вниз" - до ближайшего меньшего целого.

SELECT title, author, amount,

FLOOR(

(SELECT AVG(amount)

FROM book)

) AS Среднее\_количество

FROM book

WHERE ABS(amount - (SELECT AVG(amount) FROM book)) >3;

--результат

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

| title | author | amount | Среднее\_количество |

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

| Мастер и Маргарита | Булгаков М.А. | 3 | 7 |

| Братья Карамазовы | Достоевский Ф.М. | 3 | 7 |

| Стихотворения и поэмы | Есенин С.А. | 15 | 7 |

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

# Создание пустой таблицы

CREATE TABLE supply(

supply\_id INT PRIMARY KEY AUTO\_INCREMENT,

title VARCHAR(50),

author VARCHAR(30),

price DECIMAL(8,2),

amount INT

)

# Добавление записей в таблицу

INSERT INTO supply(title, author, price, amount)

VALUES ('Лирика', 'Пастернак Б.Л.', 518.99, 2),

('Черный человек', 'Есенин С.А.', 570.20, 6),

('Белая гвардия', 'Булгаков М.А.', 540.50, 7),

('Идиот', 'Достоевский Ф.М.', 360.80, 3);

SELECT \* FROM supply;

# Добавление записей из другой таблицы, вложенные запросы

#### **Пример** Занести все книги из таблицы supply в таблицу book

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

SELECT title, author, price, amount

FROM supply;

С помощью этого запроса в таблицу book включены все книги из supply, даже те, которые в book уже есть («Белая гвардия» и «Идиот»). Для реляционной модели это нежелательная ситуация. Устранить эту проблему можно с помощью вложенных запросов.

#### **Пример** Занести из таблицы supply в таблицу book только те книги, названия которых отсутствуют в таблице book.

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

SELECT title, author, price, amount

FROM supply

WHERE title NOT IN (

SELECT title

FROM book

);

SELECT \* FROM book;

Вложенным запросом отбираются все названия книг, которые есть в таблице book. Основным запросом SELECT из таблицы supply выбираются книги, названия которых нет в результате вложенного запроса. Отобранные записи добавляются в конец таблицы book.

# Запросы на обновление

**UPDATE таблица SET поле = выражение**

#### **Пример** Уменьшить на 30% цену книг в таблице book.

**UPDATE book**

**SET price = 0.7 \* price;**

С помощью запросов на обновление можно также изменять только часть записей. С помощью ключевого слово WHERE, после которого указывается условие отбора строк для изменения.

#### **Пример** Уменьшить на 30% цену тех книг в таблице book, количество которых меньше 5.

**UPDATE book**

**SET price = 0.7 \* price**

**WHERE amount < 5;**

**SELECT \* FROM book;**

Запросом UPDATE можно обновлять значения нескольких столбцов одновременно.

**UPDATE таблица SET поле1 = выражение1, поле2 = выражение2**

#### **Пример** В столбце buy покупатель указывает количество книг, которые он хочет приобрести. Для каждой книги, выбранной покупателем, необходимо уменьшить ее количество на складе на указанное в столбце buy количество, а в столбец buy занести 0.

**UPDATE book**

**SET amount = amount - buy,**

**buy = 0;**

**SELECT \* FROM book;**

#### **Пример** В таблице book необходимо скорректировать значение для покупателя в столбце buy таким образом, чтобы оно не превышало количество экземпляров книг, указанных в столбце amount. А цену тех книг, которые покупатель не заказывал, снизить на 10%.

**UPDATE book**

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

**price = IF(buy = 0, price \* 0.9, price);**

**SELECT \* FROM book;**

# Запросы на обновление нескольких таблиц

#### **Пример** Если в таблице supply есть те же книги, что и в таблице book, добавлять эти книги в таблицу book не имеет смысла. Необходимо увеличить их количество на значение столбца amount таблицы supply.

**UPDATE book, supply**

**SET book.amount = book.amount + supply.amount**

**WHERE book.title = supply.title AND book.author = supply.author;**

В этом запросе увеличилось количество двух книг: «Белая гвардия», которая в supply имеет ту же цену, и «Идиот», но цена этой книги в таблицах book и supply отличается.

#### **Пример** Для книг в таблице book , которые есть в таблице supply, увеличить количество в таблице book ( увеличить их количество на значение столбца amount таблицы supply), и пересчитать их цену (для каждой книги найти сумму цен из таблиц book и supply и разделить на 2).

**UPDATE book, supply**

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

**book.price = (book.price + supply.price)/2**

**WHERE book.title = supply.title;**

**SELECT \* FROM book;**

# Запросы на удаление

DELETE FROM таблица;

Этот запрос удаляет все записи из указанной после FROM таблицы.

Запрос на удаления позволяет удалить не все записи таблицы, а только те, которые удовлетворяют условию, указанному после ключевого слова WHERE:

DELETE FROM таблица

WHERE условие;

# Запросы на создание таблицы

Новая таблица может быть создана на основе данных из другой таблицы. Для этого используется запрос SELECT, результирующая таблица которого и будет новой таблицей базы данных. Запрос имеет вид:

**CREATE TABLE имя\_таблицы AS**

**SELECT ...**

#### **Пример** Создать таблицу заказ (ordering), куда включить авторов и названия тех книг, количество экземпляров которых в таблице book меньше 4. Для всех книг указать одинаковое количество экземпляров 5.

**CREATE TABLE ordering AS**

**SELECT author, title, 5 AS amount**

**FROM book**

**WHERE amount < 4;**

**SELECT \* FROM ordering;**

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

| author | title | amount |

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

| Булгаков М.А. | Мастер и Маргарита | 5 |

| Достоевский Ф.М. | Братья Карамазовы | 5 |

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

При создании таблицы можно использовать вложенные запросы как после SELECT, так и после WHERE.

#### **Пример** Создать таблицу заказ (ordering), куда включить авторов и названия тех книг, количество экземпляров которых в таблице book меньше 4. Для всех книг указать одинаковое значение - среднее количество экземпляров книг в таблице book.

**CREATE TABLE ordering AS**

**SELECT author, title,**

**(SELECT ROUND(AVG(amount))**

**FROM book**

**) AS amount**

**FROM book**

**WHERE amount < 4;**