Вложенный запрос – это запрос внутри другого запроса SQL.

Вложенный запрос используется для выборки данных, которые будут использоваться в условии отбора записей основного запроса. Его применяют для:

- сравнения выражения с результатом вложенного запроса;

- определения того, включено ли выражение в результаты вложенного запроса;

- проверки того, выбирает ли запрос определенные строки.

Вложенный запрос имеет следующие компоненты:

- ключевое слово SELECT после которого указываются имена столбцов или выражения

- таблицы, из которой выбираются данные;

- необязательное предложение WHERE;

- необязательное предложение GROUP BY:

- необязательное предложение HAVING.

Вложенные запросы могут включаться в WHERE или HAVING так (в квадратных скобках указаны необязательные элементы, через | – один из элементов):

WHERE | HAVING выражение оператор\_сравнения (вложенный запрос);

WHERE | HAVING выражение, включающее вложенный запрос;

WHERE | HAVING выражение [NOT] IN (вложенный запрос);

WHERE | HAVING выражение оператор\_сравнения ANY | ALL (вложенный запрос).

**Также вложенные запросы могут вставляться в основной запрос после ключевого слова SELECT.**

Пример:

**SELECT title, author, amount,**

**(SELECT MAX(amount)**

**FROM book**

**)-amount AS 'Заказ'**

**FROM book**

**WHERE amount<all(**

**SELECT MAX(amount)**

**FROM book**

**)**

Пример: Вывести информацию о самых дешевых книгах, хранящихся на складе.

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

**FROM book**

**WHERE price = (**

**SELECT MIN(price)**

**FROM book**

**);**

| title | author | price | amount |

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

Вложенный запрос определяет минимальную цену книг во всей таблице (это 460.00), а затем в основном запросе для каждой записи (имеется в виду ряд) проверяется, равна ли цена минимальному значению, если равна, информация о книге включается в результирующую таблицу запроса.

Пример: Вывести информацию (автора, название и цену) о книгах, цены которых меньше или равны средней цене книг на складе. Информацию вывести в отсортированном по убыванию цены виде. Среднее вычислить как среднее по цене книги.

**SELECT author, title, price**

**FROM book**

**WHERE price <= (**

**SELECT AVG(price)**

**FROM book**

**)**

**ORDER BY price DESC**

author | title | price |

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

| Достоевский Ф.М. | Игрок | 480.50 |

| Достоевский Ф.М. | Идиот | 460.00

Пример: Вывести информацию (автора, название и цену) о тех книгах, цены которых превышают минимальную цену книги на складе не более чем на 150 рублей в отсортированном по возрастанию цены виде.

**SELECT author, title, price**

**FROM book**

**WHERE price BETWEEN (SELECT MIN(price)**

**FROM book) AND**

**(SELECT MIN(price)+150**

**FROM book)**

**ORDER BY price;**

author | title | price |

| Достоевский Ф.М. | Идиот | 460.00 |

| Достоевский Ф.М. | Игрок | 480.50 |

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

| Пушкин А.С. | Евгений Онегин | 610.00

Вложенный запрос может возвращать несколько значений одного столбца. Тогда его можно использовать в разделе WHERE совместно с оператором IN.

WHERE имя\_столбца IN (вложенный запрос, возвращающий один столбец)

Оператор IN определяет, совпадает ли значение столбца с одним из значений, содержащихся во вложенном запросе. При этом логическое выражение после WHERE получает значение истина. Оператор NOT IN выполняет обратное действие – выражение истинно, если значение столбца не содержится во вложенном запросе.

Пример: Вывести информацию о книгах тех авторов, общее количество экземпляров книг которых не менее 12.

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

**FROM book**

**WHERE author IN(SELECT author**

**FROM book**

**GROUP BY author**

**HAVING SUM(amount)>12)**

title | author | amount | price

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

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

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

| Стихотворения и поэмы | Есенин С.А. | 15 | 650.00

Разъяснения: Вложенный запрос отбирает двух авторов (Достоевского и Есенина). А в основном запросе для каждой записи таблицы book проверяется, входит ли автор книги в отобранный список, если входит - информация о книге включается в запрос. т.е. проверяется каждая строка. Такая хуйня ебаная просто пизда.

**Пример**:Вывести (автора, книгу и количество) о тех книгах, количество экземпляров которых в таблице book не дублируется.

**SELECT author, title, amount**

**FROM book**

**WHERE amount IN (SELECT amount**

**FROM book**

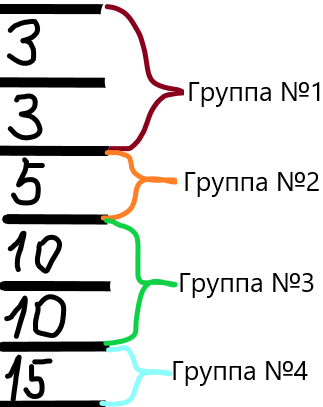
**GROUP BY amount**

**HAVING count(amount)=1**

**)**

Здесь подзапросом с помощью GROUP BY разбиваем все значения **amount** по группам, а затем с помощью count(amount) считаем количество строк в каждой группе.

Получается что HAVING как-бы работает не в book таблице а в таблице созданной GROUP BY , думаю эта таблица выглядит примерно так:

Я предполагаю что GROUP BY разбил book таблицу и сформировал вот такую таблицу состоящую из групп. Как видно здесь 4 группы, в каждой группе определённое количество аmount в зависимости от их количества и повторений в book таблице.

Что дальше? Как считает HAVING? HAVING не считает, считает count(amount)

Я думаю что он считает количество записей в каждой группе (строки). Если ввести **SELECT count(amount)**

**FROM book**

**GROUP BY amount** то результат будет:

**count(amount)**

**| 2**

**| 1**

**| 2**

**| 1**

А вот теперь с помощью HAVING мы как бы определяем где сколько лежит.

**Пример**: Посчитать сколько и каких экземпляров книг нужно заказать поставщикам, чтобы на складе стало одинаковое количество экземпляров каждой книги, равное значению самого большего количества экземпляров одной книги на складе. Вывести название книги, ее автора, текущее количество экземпляров на складе и количество заказываемых экземпляров книг. Последнему столбцу присвоить имя Заказ. В результат не включать книги, которые заказывать не нужно.

По прочтению задания нужно произвести его декомпозицию:

1) Для начала находим максимальное значение на текущий момент по amount (это будет вложенный select)

select max(amount) from book - вложенный запрос на поиск максимального значения "amount" будет равен "15"

2) Формируем столбец заказа

(пункт 1)-amount) as Заказ - формирование столбца "Заказ"

3) Нам нужно исключить позицию с максимальным amount из итоговой скорректированной таблицы

where amount not in (пункт 1) - убирается строка "6 | Стихотворения и поэмы | Есенин С.А. | 650.00 | 15"

4) Формирование итогового запроса

SELECT ... , ... , ... , пункт 2 FROM ... WHERE пункт 3 ;

**SELECT title, author, amount,**

**(SELECT MAX(amount)**

**FROM book**

**)-amount AS 'Заказ'**

**FROM book**

**WHERE amount<all(**

**SELECT MAX(amount)**

**FROM book**

**)**

title | author | amount | Заказ

| Мастер и Маргарита | Булгаков М.А. | 3 | 12

| Белая гвардия | Булгаков М.А. | 5 | 10

| Идиот | Достоевский Ф.М. | 10 | 5

| Братья Карамазовы | Достоевский Ф.М. | 3 | 12

| Игрок | Достоевский Ф.М. | 10 | 5

**Запросы для нескольких таблиц (join) с вложенными запросами**

В запросах, построенных на нескольких таблицах, можно использовать вложенные запросы. Вложенный запрос может быть включен: после ключевого слова SELECT, после FROM и в условие отбора после WHERE (HAVING).

Пример: Вывести авторов, общее количество книг которых на складе максимально.

**SELECT name\_author, SUM(amount) as Количество**

**FROM**

**author INNER JOIN book**

**on author.author\_id = book.author\_id**

**GROUP BY name\_author**

**HAVING SUM(amount) =**

**(/\* вычисляем максимальное из общего количества книг каждого автора \*/**

**SELECT MAX(sum\_amount) AS max\_sum\_amount**

**FROM**

**(/\* считаем количество книг каждого автора \*/**

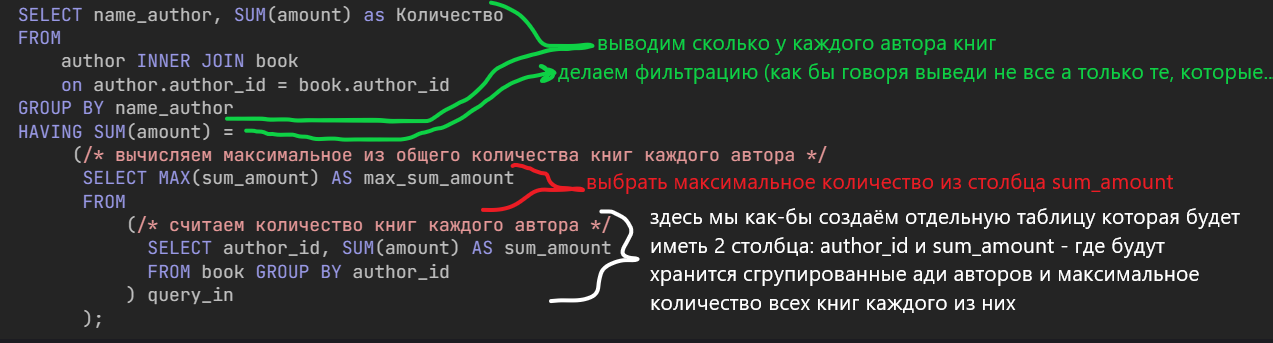
**SELECT author\_id, SUM(amount) AS sum\_amount**

**FROM book GROUP BY author\_id**

**) query\_in**

**);**

ПОЯСНЕНИЯ:



Наверно следует рассматривать данный запрос с самого вложенного, который возврасчает таблицу(грубо говоря). Эта таблица имеет только 2 столбца со значениями. В среднем по вложенности запросе мы просто находим максимальное значение в одном из столбцов, это значение =23-. Ано то нам и надо для условия в самом внешнем запросе. Теперь по сути запрос выглядит вот так:

**SELECT name\_author, SUM(amount) as Количество**

**FROM**

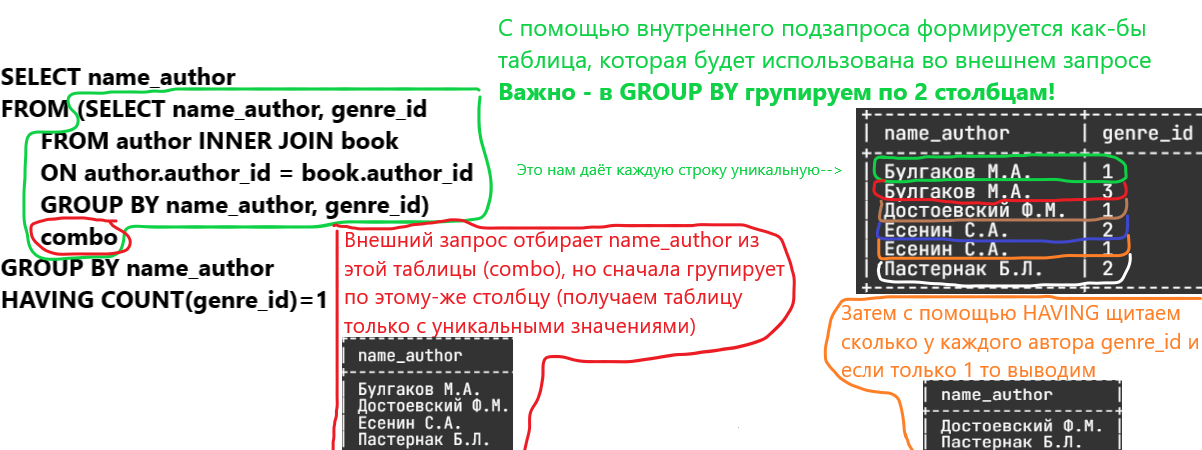
**author INNER JOIN book**

**on author.author\_id = book.author\_id**

**GROUP BY name\_author**

**HAVING SUM(amount) = 23**

Пример: Вывести в алфавитном порядке всех авторов, которые пишут только в одном жанре.



Из интересного:

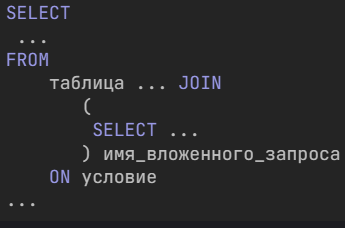
- Есть вложенный подзапрос у которого есть JOIN

- у вложенного подзапроса есть имя ‘combo’ это важно, иначе FROM не сработает.

- в вложенном подзапросе группируем по **2 столбцам**

**Вложенные запросы в операторах соединения**

Вложенные запросы могут использоваться в операторах соединения JOIN. При этом им необходимо присваивать имя, которое записывается сразу после закрывающей скобки вложенного запроса.



Вложенный запрос может стоять как справа, так и слева от оператора JOIN. Допускается использование двух запросов в операторах соединения