**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.

SELECT

...

FROM

таблица ... JOIN (SELECT ...) имя\_вложенного\_запроса

ON условие

...

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

**Пример**

Вывести авторов, пишущих книги в самом популярном жанре. Самым популярным считать жанр, общее количество экземпляров книг которого на складе максимально. Таких жанров может быть несколько, если они имеют одинаковое максимальное значение общего количества экземпляров. Только для этого шага изменена запись в таблице **book**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **book\_id** | **title** | **author\_id** | **genre\_id** | **price** | **amount** |
| 8 | Лирика | 4 | 2 | 518.9910 | 10 |

А также добавлены новые записи:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **book\_id** | **title** | **author\_id** | **genre\_id** | **price** | **amount** |
| 9 | Герой нашего времени | 5 | 3 | 570.59 | 2 |
| 10 | Доктор Живаго | 4 | 3 | 740.50 | 5 |

Рассмотрим реализацию этого запроса по шагам.

**Шаг 1.** Найдем общее количество книг по каждому жанру, отсортируем его по убыванию и ограничим вывод одной строкой. Рекомендуется, если запрос будет использоваться в качестве вложенного (особенно в операциях соединения), вычисляемым полям запроса давать собственное имя.

*Запрос:*

SELECT genre\_id, SUM(amount) AS sum\_amount

FROM book

GROUP BY genre\_id

ORDER BY sum\_amount DESC

LIMIT 1

*Результат:*

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

| genre\_id | sum\_amount |

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

| 1 | 31 |

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

Кажется, что, уже используя этот запрос, можно получить**id** самого популярного жанра. Но это не так, поскольку несколько жанров могут иметь одинаковую популярность. Поэтому нам необходим запрос, который отберет ВСЕ жанры, суммарное количество книг которых равно **sum\_amount**.

**Шаг 2.** Используя запрос с предыдущего шага, найдем **id** самых популярных жанров.

*Запрос:*

SELECT query\_in\_1.genre\_id

FROM (SELECT genre\_id, SUM(amount) AS sum\_amount

FROM book

GROUP BY genre\_id

)query\_in\_1

INNER JOIN (SELECT genre\_id, SUM(amount) AS sum\_amount

FROM book

GROUP BY genre\_id

ORDER BY sum\_amount DESC

LIMIT 1

) query\_in\_2

on query\_in\_1.sum\_amount= query\_in\_2.sum\_amount

*Результат:*

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

| genre\_id |

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

| 1 |

| 2 |

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

**Шаг 3.** Используя запрос с шага 2, выведем фамилии авторов, которые пишут в самых популярных жанрах. В этом запросе обязательно выполнить группировку по фамилиям авторов и **id** жанров, так как без этого фамилии авторов будут повторяться, поскольку в таблице**book** есть разные книги, написанные автором в одном жанре.

*Запрос:*

SELECT name\_author

FROM author INNER JOIN book

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

GROUP BY name\_author, genre\_id

HAVING genre\_id IN

(SELECT query\_in\_1.genre\_id

FROM (SELECT genre\_id, SUM(amount) AS sum\_amount

FROM book

GROUP BY genre\_id

)query\_in\_1

INNER JOIN (SELECT genre\_id, SUM(amount) AS sum\_amount

FROM book

GROUP BY genre\_id

ORDER BY sum\_amount DESC

LIMIT 1

) query\_in\_2

on query\_in\_1.sum\_amount= query\_in\_2.sum\_amount

);

*Результат:*

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

| name\_author |

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

| Булгаков М.А. |

| Достоевский Ф.М. |

| Есенин С.А. |

| Пастернак Б.Л. |

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