

Запросы к нескольким таблицам

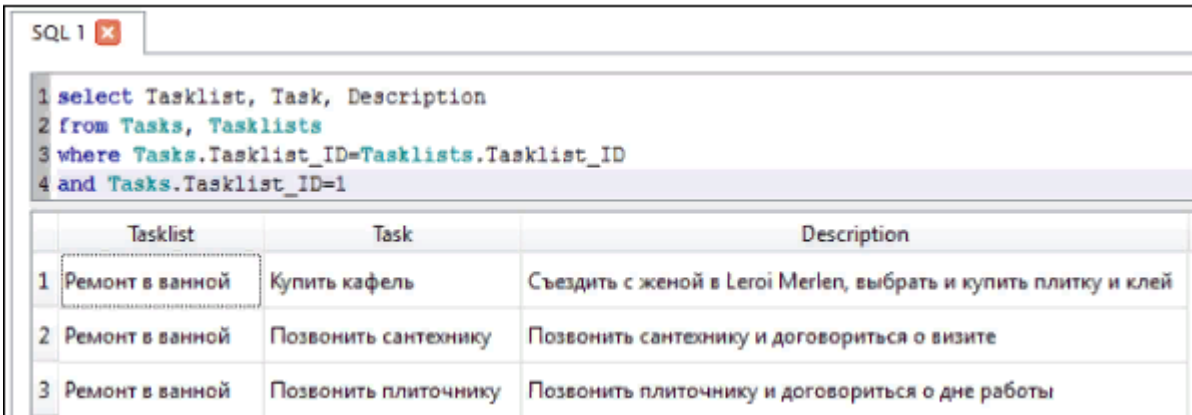
В предыдущей главе нам пришлось выполнить два запроса, чтобы

- а) увидеть все задачи, входящие в первый по номеру список и
- б) узнать название этого списка.

Так получилось, потому что к этому моменту мы знаем, как использовать запросы SELECT для выбора записей только из одной таблицы. Информация о задачах хранится в одной таблице (**Tasks**), а названия списков задач — в другой (**Tasklists**).

Можно ли объединить эти два запроса в один, обратившись сразу к двум таблицам? Конечно, можно! Введите и выполните такой запрос и посмотрите на результат:

```
select Tasklist, Task, Description
from Tasks, Tasklists
where Tasks.Tasklist_ID=Tasklists.Tasklist_ID
and Tasks.Tasklist_ID=1
```



The screenshot shows a SQL query editor window titled 'SQL 1'. The query is as follows:

```
1 select Tasklist, Task, Description
2 from Tasks, Tasklists
3 where Tasks.Tasklist_ID=Tasklists.Tasklist_ID
4 and Tasks.Tasklist_ID=1
```

Below the query, the results are displayed in a table with three columns: Tasklist, Task, and Description. The table contains three rows of data.

	Tasklist	Task	Description
1	Ремонт в ванной	Купить кафель	Съездить с женой в Leroi Merlen, выбрать и купить плитку и клей
2	Ремонт в ванной	Позвонить сантехнику	Позвонить сантехнику и договориться о визите
3	Ремонт в ванной	Позвонить плиточнику	Позвонить плиточнику и договориться о дне работы

Теперь в результатах запроса мы видим и название списка задач, и сами задачи.

В этом запросе появилось три новых элемента. Давайте рассмотрим их подробнее.

Во-первых обобщённая структура запроса теперь выглядит так:

```
SELECT {список полей} FROM {список таблиц} WHERE {условие отбора}
```

Там, где мы раньше использовали имя одной таблицы, можно использовать целый список таблиц. Имена таблиц в этом списке разделяются запятой:

```
... from Tasks, Tasklists ...
```

Во-вторых, поскольку в двух наших таблицах присутствуют поля с одинаковыми именами Tasklist_ID, каждый раз при использовании этого имени нам пришлось дополнительно сообщать движку базы данных, какую из таблиц мы имеем в виду. Для этого использовано расширенное имя поля, которое состоит из имени таблицы и собственно имени поля, разделённых точкой:

```
... where Tasks.Tasklist_ID = Tasklists.Tasklist_ID ...
```

Если бы мы не указали принадлежность каждого из полей определённой таблице, движок вернул бы нам сообщение об ошибке примерно такого содержания: «не могу понять, к какой таблице относится поле Tasklist_ID».

```
ambiguous column name: Tasklist id
```

В-третьих, в этом запросе мы использовали условие, составленное из двух проверок:

Tasks.Tasklist_ID=Tasklists.Tasklist_ID and Tasks.Tasklist_ID=1

Эти две проверки соединены ещё одним ключевым словом — **AND** (и). Оно означает, что для отбора записей должны быть выполнены оба условия:

- в таблицах Tasklists и Tasks должны совпадать значения полей Tasklist_ID
- значение этого поля в таблице Tasks должно быть равно 1.

Этот запрос уже далеко не такой простой, как те, что мы использовали раньше. Чтобы лучше понять, как работают условия отбора, давайте попробуем выполнить его без условия **WHERE**:

```
select Tasklist, Task, Description
from Tasks, Tasklists
```

The screenshot shows a SQL query execution window with the following SQL code:

```
1 select Tasklist, Task, Description
2 from Tasks, Tasklists
3
```

The result is a table with 40 rows, showing a Cartesian product of the Tasks and Tasklists tables. The columns are Tasklist, Task, and Description. The first four rows show the same Tasklist ('Ремонт в ванной', 'Подготовка сына к школе', 'Запуск сайта', 'Регистрация ИП') paired with the same Task ('Развернуть сайт'). The next four rows show the same Tasklist paired with a different Task ('Купить кафель'). The last four rows show the same Tasklist paired with a third Task ('Съездить с женой в Leroi Merlen, выбрать и купить ...').

	Tasklist	Task	Description
1	Ремонт в ванной	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге
2	Подготовка сына к школе	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге
3	Запуск сайта	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге
4	Регистрация ИП	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге
5	Ремонт в ванной	Купить кафель	Съездить с женой в Leroi Merlen, выбрать и купить ...
6	Подготовка сына к школе	Купить кафель	Съездить с женой в Leroi Merlen, выбрать и купить ...
7	Запуск сайта	Купить кафель	Съездить с женой в Leroi Merlen, выбрать и купить ...
8	Регистрация ИП	Купить кафель	Съездить с женой в Leroi Merlen, выбрать и купить ...

40 Rows returned from: select Tasklist, Task, Description from Tasks, Tasklists (took 2ms)

Результат запроса состоит аж из 40 строк! И посмотрите: задача «Развернуть сайт» появилась в результатах запроса четыре раза, потому что при обработке запроса движок «включил» её в каждый из четырёх списков. То же самое произошло со всеми задачами. А поскольку списков у нас четыре, а задач десять, в результате и получилось $4 \times 10 = 40$ строк.

Почему так получилось? Дело в том, что наша база данных на самом деле не знает, к какому списку относится какая задача. Хотя мы и дали одинаковые названия полям Tasklist_ID в двух таблицах, это сделано только для нашего удобства. Имена полей могли быть и разными. Чтобы связать список задач в таблице Tasklists с задачами из таблицы Tasks, мы должны явно сообщить в запросе, что таблицы связаны именно по этим полям. Для этого и служит первая часть условия:

```
where Tasks.Tasklist_ID = Tasklists.Tasklist_ID
```

Если мы выполним запрос, добавив к нему только это условие, то движок базы данных отберёт из этих сорока строк только те комбинации записей таблиц Tasks и Tasklists, в которых значения полей Tasklist_ID совпадают:

```
select Tasklist, Task, Description
from Tasks, Tasklists
where Tasks.Tasklist_ID=Tasklists.Tasklist_ID
```

SQL 1 ✕

```

1 select Tasklist, Task, Description
2 from Tasks, Tasklists
3 where Tasks.Tasklist_ID=Tasklists.Tasklist_ID
4

```

	Tasklist	Task	Description
1	Запуск сайта	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге
2	Ремонт в ванной	Купитть кафель	Съездить с женой в Leroi Merlen, выбрать и купить ...
3	Ремонт в ванной	Позвонить сантехнику	Позвонить сантехнику и договориться о визите
4	Подготовка сына к школе	Купитть форму	Съездить на ярмарку за школьной формой для сына
5	Подготовка сына к школе	Купитть учебники	Позвонить учительнице, узнать, какие учебники ну...
6	Запуск сайта	Подобрать хостинг	Подобрать провайдера для хостинга сайта, выбрат...
7	Ремонт в ванной	Позвонить плиточнику	Позвонить плиточнику и договориться о дне работы
8	Регистрация ИП	Подготовить документы	Подготовить документы для регистрации
9	Регистрация ИП	Оплатить пошлину	Оплатить пошлину за регистрацию
10	Регистрация ИП	Подать документы	Подать документы на регистрацию в налоговую

10 Rows returned from: select Tasklist, Task, Description
from Tasks, Tasklists
where Tasks.Tasklist_ID=Tasklists.Tasklist_ID (took 4ms)

Теперь количество строк в результате запроса совпадает с количеством записей таблицы Tasks, и каждой задаче поставлен в соответствие только один список.

Обратите внимание: полю Tasklist_ID, которое мы использовали в условии отбора, совсем не обязательно присутствовать в списке возвращаемых полей.

Что происходит, когда мы добавляем к запросу второе условие?

and Tasks.Tasklist_ID=1

Как вы, наверное, уже поняли, оно дополнительно отфильтровывает из результата предыдущего запроса только те строки, при формировании которых использовались записи таблицы Tasks, в которых значение поля Tasklist_ID равно 1.

SQL 1 ✕

```

1 select Tasklist, Task, Description
2 from Tasks, Tasklists
3 where Tasks.Tasklist_ID=Tasklists.Tasklist_ID
4 and Tasks.Tasklist_ID=1

```

	Tasklist	Task	Description
1	Ремонт в ванной	Купитть кафель	Съездить с женой в Leroi Merlen, выбрать и купить плитку и клей
2	Ремонт в ванной	Позвонить сантехнику	Позвонить сантехнику и договориться о визите
3	Ремонт в ванной	Позвонить плиточнику	Позвонить плиточнику и договориться о дне работы

В предыдущей главе говорилось, что можно избежать запоминания номеров записей в запросе. Наверное, вы и сами уже догадались, что в качестве условия отбора записей можно использовать поле **Tasklists.Tasklist** (название списка) вместо поля **Tasks.Tasklist_ID** (номер списка):

```

select Tasklist, Task, Description
from Tasks, Tasklists
where Tasks.Tasklist_ID=Tasklists.Tasklist_ID
and Tasklists.Tasklist='Ремонт в ванной'

```

	Tasklist	Task	Description
1	Ремонт в ванной	Купить кафель	Съездить с женой в Leroi Merlen, выбрать и купить плитку и клей
2	Ремонт в ванной	Позвонить сантехнику	Позвонить сантехнику и договориться о визите
3	Ремонт в ванной	Позвонить плиточнику	Позвонить плиточнику и договориться о дне работы

Если в запросе SQL используется строка, то она должна быть обрамлена одинарными кавычками:

```
and Tasklists.Tasklist='Ремонт в ванной'
```

Таблиц в списке может быть и больше двух, и в большинстве случаев каждая из перечисленных таблиц должна упоминаться хотя бы в одном условии, связывающем её с другими таблицами. Следующий запрос выбирает данные из трёх таблиц, добавляя к каждой задаче информацию о том, в какой список она входит, а также к какой категории дел относится этот список:

```

select Category, Tasklist, Task, Description
from Categories, Tasks, Tasklists
where Tasks.Tasklist_ID=Tasklists.Tasklist_ID
and Tasklists.Category_ID=Categories.Category_ID

```

	Category	Tasklist	Task	Description
1	Своё дело	Запуск сайта	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге
2	Дом	Ремонт в ванной	Купить кафель	Съездить с женой в Leroi Merlen, выбрать и купить ...
3	Дом	Ремонт в ванной	Позвонить сантехнику	Позвонить сантехнику и договориться о визите
4	Семья	Подготовка сына к школе	Купить форму	Съездить на ярмарку за школьной формой для сына
5	Семья	Подготовка сына к школе	Купить учебники	Позвонить учительнице, узнать, какие учебники ну...
6	Своё дело	Запуск сайта	Подобрать хостинг	Подобрать провайдера для хостинга сайта, выбрат...
7	Дом	Ремонт в ванной	Позвонить плиточнику	Позвонить плиточнику и договориться о дне работы
8	Своё дело	Регистрация ИП	Подготовить документы	Подготовить документы для регистрации
9	Своё дело	Регистрация ИП	Оплатить пошлину	Оплатить пошлину за регистрацию
10	Своё дело	Регистрация ИП	Подать документы	Подать документы на регистрацию в налоговую

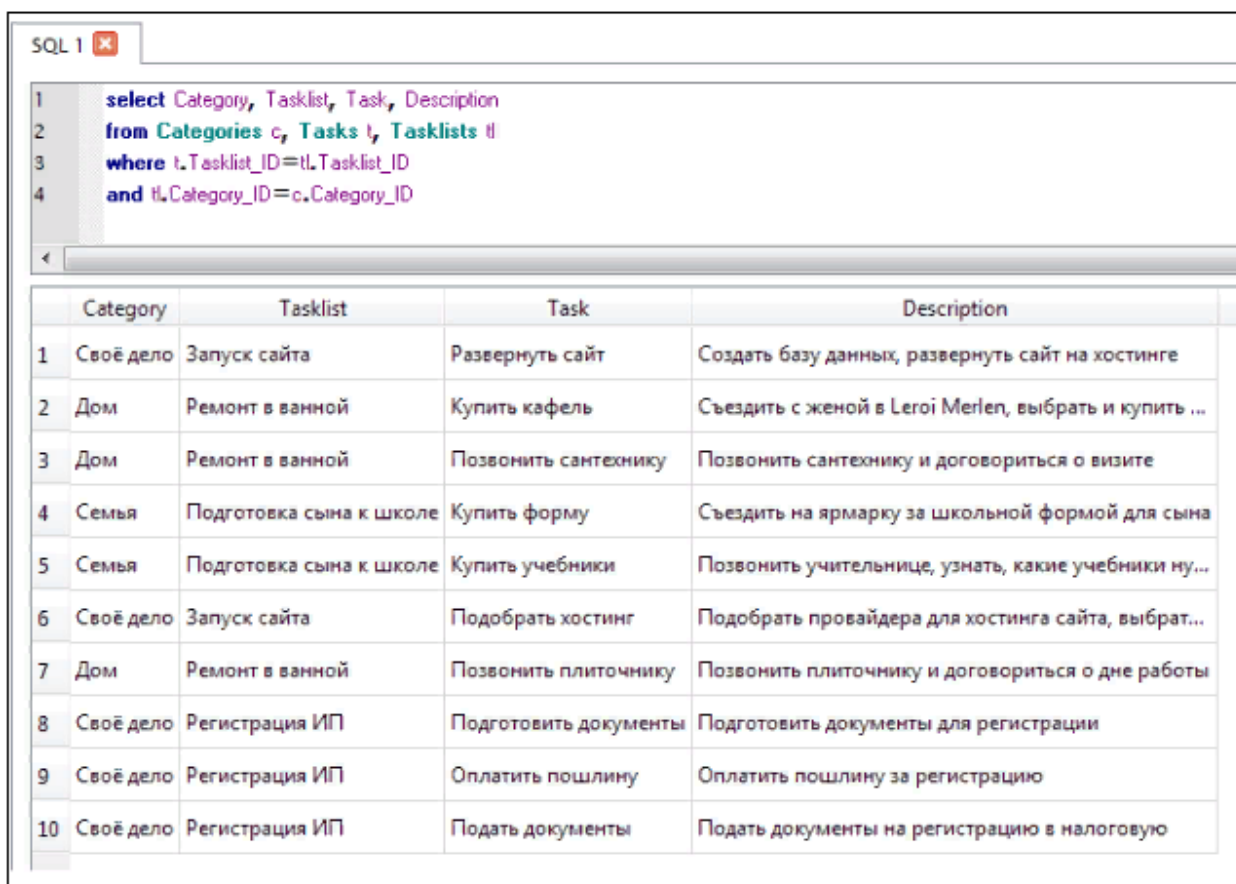
Сокращение запросов с помощью псевдонимов

Изучаемые нами запросы становятся всё более громоздкими. Нам приходится по несколько раз включать в запрос названия таблиц, которые могут быть достаточно длинными. Это не очень удобно и увеличивает вероятность опечатки при вводе запроса.

В SQL есть способ сократить длинные запросы с помощью так называемых *псевдонимов*. Псевдоним таблицы представляет собой короткую замену названию таблицы, которая может использоваться в пределах запроса в полных именах полей.

Например, наш предыдущий запрос можно записать таким образом:

```
select Category, Tasklist, Task, Description
from Categories c, Tasks t, Tasklists tl
where t.Tasklist_ID=tl.Tasklist_ID
and tl.Category_ID=c.Category_ID
```



The screenshot shows a SQL query editor window titled "SQL 1" with a close button. The query is as follows:

```
1 select Category, Tasklist, Task, Description
2 from Categories c, Tasks t, Tasklists tl
3 where t.Tasklist_ID=tl.Tasklist_ID
4 and tl.Category_ID=c.Category_ID
```

Below the query editor, the results of the query are displayed in a table with 10 rows and 5 columns: Category, Tasklist, Task, and Description. The results are as follows:

	Category	Tasklist	Task	Description
1	Своё дело	Запуск сайта	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге
2	Дом	Ремонт в ванной	Купить кафель	Съездить с женой в Leroi Merlen, выбрать и купить ...
3	Дом	Ремонт в ванной	Позвонить сантехнику	Позвонить сантехнику и договориться о визите
4	Семья	Подготовка сына к школе	Купить форму	Съездить на ярмарку за школьной формой для сына
5	Семья	Подготовка сына к школе	Купить учебники	Позвонить учительнице, узнать, какие учебники ну...
6	Своё дело	Запуск сайта	Подобрать хостинг	Подобрать провайдера для хостинга сайта, выбрат...
7	Дом	Ремонт в ванной	Позвонить плиточнику	Позвонить плиточнику и договориться о дне работы
8	Своё дело	Регистрация ИП	Подготовить документы	Подготовить документы для регистрации
9	Своё дело	Регистрация ИП	Оплатить пошлину	Оплатить пошлину за регистрацию
10	Своё дело	Регистрация ИП	Подать документы	Подать документы на регистрацию в налоговую

Для каждой таблицы в этом запросе определён псевдоним. Псевдоним указывается сразу после «настоящего» имени таблицы в конструкции FROM:

```
from Categories c, Tasks t, Tasklists tl
```

В нашем примере псевдонимы использовались для сокращения текста запроса в условиях отбора после ключевого слова WHERE. Но их точно так же можно использовать и до ключевого слова FROM в полных именах полей, например:

```
select tl.Category_ID, Tasklist
from Categories c, Tasklists tl
where tl.Category_ID=c.Category_ID
```

Запросы к нескольким таблицам — практика

Составьте запрос, выбирающий из таблиц Goods и Goods_Spendings и возвращающий перечень названий отслеживаемых товаров и всех потраченных на них сумм.

Составьте запрос к таблицам Categories и Spendings, который возвращает название категории покупки, название магазина и потраченную сумму для всех покупок, относящихся к категории с номером 4.

Составьте запрос к таблицам Goods и Goods_Spendings, возвращающий наименование товара и количество купленных единиц по всем покупкам для товара «Пиво».

Доработайте предыдущий запрос, добавив к результатам дату покупки из таблицы Spendings. Таким образом, ваш запрос должен обращаться к трём таблицам (Goods, Spendings и Good_Spendings) и возвращать наименование товара, количество купленных единиц и дату каждой покупки для товара «Пиво».

Составьте запрос, обращающийся ко всем четырём таблицам (Categories, Spendings, Goods, Goods_Spendings), возвращающий следующую информацию обо всех покупках отслеживаемых товаров: название категории, название товара, название магазина и дату покупки.