

Условия отбора

Операторы сравнения

В рассмотренных примерах мы до сих пор использовали в качестве условий отбора записей только выражения, содержащие знак равенства, например:

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

Это выражение можно словами описать таким образом: «значение поля Tasklist таблицы Tasklists совпадает со строкой 'Ремонт в ванной'».

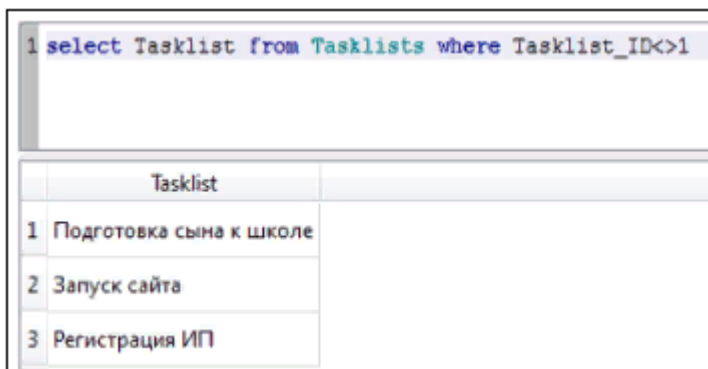
Проверка на совпадение значений — это, конечно, не единственное допустимое выражение в условиях отбора. SQL позволяет использовать в условиях разнообразные *операторы* сравнения. К числу самых распространённых относятся следующие:

= - уже знакомый нам оператор сравнения на совпадение или равенство;

<> или **!=** - оператор сравнения на несовпадение или неравенство. Результат сравнения противоположен результату условия с оператором **=**

Например, следующий запрос выберет из таблицы Tasklists названия всех списков задач, *кроме* списка с номером 1:

```
select Tasklist from Tasklists where Tasklist_ID<>1
```



The image shows a screenshot of a SQL query and its result set. The query is: `1 select Tasklist from Tasklists where Tasklist_ID<>1`. The result set is a table with two columns: an implicit ID column and a 'Tasklist' column. The rows are:

	Tasklist
1	Подготовка сына к школе
2	Запуск сайта
3	Регистрация ИП

<, **>**, **<=**, **>=** - меньше, больше, меньше или равно, больше или равно.

Результат сравнения с этими операторами зависит от типа данных. Для численных данных операторы сравнения работают обычным, знакомым нам со школьных лет способом. Например, следующий запрос выберет из таблицы Tasks все записи с номером задачи больше 5.

```
select * from Tasks where Task_ID>5
```

Task_ID	Task	Description	TaskList_ID	Done
1 6	Купить учебники	Позвонить учительнице, узнать, какие учебники ну...	2	0
2 7	Подобрать хостинг	Подобрать провайдера для хостинга сайта, выбрат...	3	0
3 8	Позвонить плиточнику	Позвонить плиточнику и договориться о дне работы	1	0
4 9	Подготовить документы	Подготовить документы для регистрации	4	0
5 10	Оплатить пошлину	Оплатить пошлину за регистрацию	4	1
6 11	Подать документы	Подать документы на регистрацию в налоговую	4	0

Для данных типов «дата» и «время» эти операторы работают интуитивно понятным образом, используя хронологический порядок: < означает «раньше», > - «позже». <= и >=, соответственно, означают «не позже» и «не раньше».

Для данных типа «текст» производится побуквенное сравнение, начиная с первой буквы. При этом используется алфавитный порядок: выполнение условия 'a' < 'б' означает, что левая буква находится ближе к началу алфавита, чем правая. Но надо сказать, что результат сравнения зависит от типа и языковых настроек используемой базы данных. В частности, заглавные и строчные буквы могут считаться двумя разными наборами букв, так что их сравнение оказывается бессмысленным. Например, может выполняться такое условие: 'a' > 'Б'. У каждого типа баз данных есть собственные настройки таких сравнений для текстовых данных. Если вы не знаете особенностей этих настроек, то лучше избегать сравнения текстовых строк с использованием этих операторов, потому что результат может оказаться совсем не таким, как вы ожидали.

Операторы сравнения удобно использовать, когда мы имеем дело с количественными значениями. В нашей учебной базе «Список дел» числа используются только как идентификаторы или признаки, поэтому условие вроде Task_ID > 5 выглядит оторванным от жизни. Более осмысленные задачи с использованием операторов сравнения вы будете решать со второй учебной базой, «Учёт расходов», в которой хранятся данные о суммах покупок.

Отбор по спискам значений

В условии отбора можно указать список возможных значений поля. Для этого используется ключевое слово IN, после которого в скобках перечисляются значения. Значения разделяются запятыми.

Следующий запрос выберет из таблицы **Tasks** записи задач, которые относятся к спискам №№ 2, 3 и 4.

```
select * from Tasks where Tasklist_ID in (2,3,4)
```

```
1 select * from Tasks where Tasklist_ID in (2,3,4)
```

Task_ID	Task	Description	TaskList_ID	Done
1 1	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге	3	0
2 5	Купить форму	Съездить на ярмарку за школьной формой для сына	2	0
3 6	Купить учебники	Позвонить учительнице, узнать, какие учебники ну...	2	0
4 7	Подобрать хостинг	Подобрать провайдера для хостинга сайта, выбрат...	3	0
5 9	Подготовить документы	Подготовить документы для регистрации	4	0
6 10	Оплатить пошлину	Оплатить пошлину за регистрацию	4	1
7 11	Подать документы	Подать документы на регистрацию в налоговую	4	0

Выражение IN с перечислением можно использовать не только с числами, а с любыми типами значений. Например, со строками:

```
select * from Categories where Category in ('Семья', 'Дом')
```

```
1 select * from Categories where Category in ('Семья', 'Дом')
```

Category_ID	Category	Category_Description
1 1	Семья	Семейные дела.
2 3	Дом	Дела по дому: покупки, ремонт, оплата счетов.

Отбор по диапазонам значений

В SQL также есть способ указать диапазон значений в условии отбора. Для этого используется ключевое слово BETWEEN, после которого указываются минимальное и максимальное значения, разделённые словом AND. Следующий запрос тоже отберёт из таблицы Tasks записи задач из списков №№2, 3 и 4, но вместо перечисления мы укажем в нём диапазон значений:

```
select * from Tasks where Tasklist_ID between 2 and 4
```

```

1 select * from Tasks where Tasklist_ID between 2 and 4
2

```

Task_ID	Task	Description	TaskList_ID	Done
1 1	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге	3	0
2 5	Купить форму	Съездить на ярмарку за школьной формой для сына	2	0
3 6	Купить учебники	Позвонить учительнице, узнать, какие учебники ну...	2	0
4 7	Подобрать хостинг	Подобрать провайдера для хостинга сайта, выбрат...	3	0
5 9	Подготовить документы	Подготовить документы для регистрации	4	0
6 10	Оплатить пошлину	Оплатить пошлину за регистрацию	4	1
7 11	Подать документы	Подать документы на регистрацию в налоговую	4	0

Комбинирование условий сравнения

Мы уже использовали комбинацию условий с использованием ключевого слова AND в запросе, выбирающим из двух таблиц данные по задачам, относящимся к списку с номером один:

```

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

```

Ключевое слово AND (и) в данном случае означает, что должны выполняться оба условия, которые оно объединяет.

SQL поддерживает также ключевое слово OR (или). Оно означает, что должно выполняться хотя бы одно из указанных условий. Например, следующий запрос выберет информацию о задачах, входящих в список задач номер 3 или номер 4.

```

select * from Tasks
where Tasklist_ID=3 or Tasklist_ID=4

```

```

1 select * from Tasks
2 where Tasklist_ID=3 or Tasklist_ID=4

```

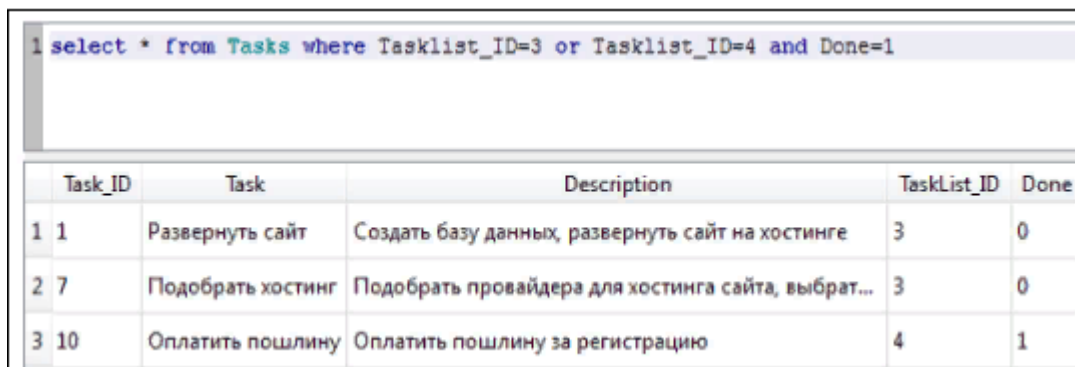
Task_ID	Task	Description	TaskList_ID	Done
1 1	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге	3	0
2 7	Подобрать хостинг	Подобрать провайдера для хостинга сайта, выбрат...	3	0
3 9	Подготовить документы	Подготовить документы для регистрации	4	0
4 10	Оплатить пошлину	Оплатить пошлину за регистрацию	4	1
5 11	Подать документы	Подать документы на регистрацию в налоговую	4	0

В условиях отбора можно комбинировать сколько угодно выражений сравнения, объединяемых ключевыми словами AND и OR. Но при этом нужно очень хорошо представлять, в каком порядке будут выполняться эти сравнения. Проблема в том, что при сравнении ключевые слова AND и OR имеют

разный приоритет: сначала обрабатываются все условия, связанные словом AND, и только после этого — комбинации условий с помощью OR.

Например, если мы хотим отобразить записи по всем выполненным задачам (Done = 1), входящим в списки №3 и №4, то следующий запрос вернёт неправильный результат:

```
select * from Tasks where Tasklist_ID=3 or Tasklist_ID=4 and Done=1
```



Task_ID	Task	Description	TaskList_ID	Done
1 1	Развернуть сайт	Создать базу данных, развернуть сайт на хостинге	3	0
2 7	Подобрать хостинг	Подобрать провайдера для хостинга сайта, выбрат...	3	0
3 10	Оплатить пошлину	Оплатить пошлину за регистрацию	4	1

Почему так получилось? Потому что при разборе запроса движок сначала объединил условия, связанные словом AND:

```
Tasklist_ID=4 and Done=1
```

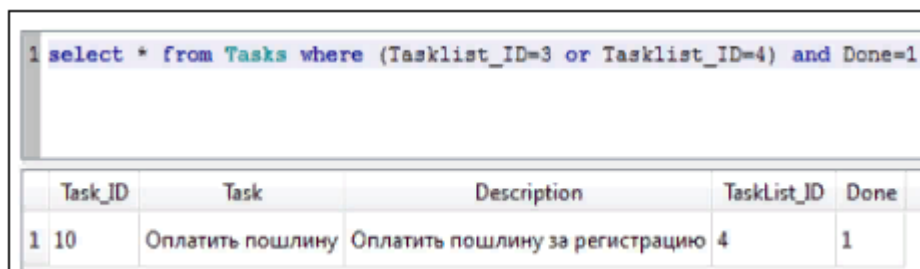
А потом добавил условие, стоящее перед OR:

```
Tasklist_ID=3
```

Полученную комбинацию условий можно словами описать так: «задача входит в список №4 и выполнена, либо задача входит в список №3 (и при этом неважно, выполнена или нет)».

Чтобы избежать подобной путаницы, можно использовать скобки. Комбинации условий в скобках обрабатываются с наивысшим приоритетом:

```
select * from Tasks where (Tasklist_ID=3 or Tasklist_ID=4) and Done=1
```



Task_ID	Task	Description	TaskList_ID	Done
1 10	Оплатить пошлину	Оплатить пошлину за регистрацию	4	1

Для этого примера есть ещё лучший способ избежать путаницы. Если указать диапазон значений в условии отбора по полю Tasklist_ID, то неоднозначность пропадает:

```
select * from Tasks where Tasklist_ID in (3,4) and Done=1
```

Это хорошая практика составления удобочитаемых запросов: использовать условия с ключевыми словами IN (задавая перечень значений) или BETWEEN (указывая диапазон значений) вместо комбинирования условий с помощью OR. Кроме того, в большинстве случаев такие запросы ещё и быстрее выполняются.

Условия отбора — практика

Составьте запрос, возвращающий значения всех полей таблицы Spendings для покупок на сумму более 1000 руб.

Составьте запрос к таблице Spendings, возвращающий название магазина и потраченную сумму для покупок, совершённых не позже 3 июня 2015 года.

Подсказка. Эту дату в запросе нужно указать в виде такой текстовой строки: '2015-06-03'

Составьте запрос, возвращающий названия категорий с номерами 1, 3 и 4 (с использованием ключевого слова IN).

Составьте запрос к таблице Spendings, возвращающий название магазина, дату покупки и сумму для покупок, сумма которых находится в диапазоне от 1000 до 2000 руб (с использованием ключевого слова between).

Составьте запрос, обращающийся к таблицам Goods и Goods_Spendings и возвращающий названия отслеживаемых товаров и потраченные на них суммы, превышающие 500 руб.

