

5.1 Основы работы с выражением WHERE

Выражение WHERE в запросах Oracle SQL для фильтрации возвращаемых значений, условия сравнения (conditions) в Oracle

Если мы обратимся к базовому синтаксису команды SELECT (см. раздел 2.1), то увидим, что после списка столбцов должно идти ключевое слово FROM с указанием источника для извлечения информации (таким источником может быть таблица, представление, подзапрос, функция и т.п.). Сразу после выражения FROM должно следовать ключевое слово WHERE с выражением, которое определяет фильтр для извлечения записей из источника. Возвращаются только те записи, для которых выражение WHERE возвращает true.

Пример запроса с выражением WHERE может выглядеть так:

```
SELECT * FROM hr.employees WHERE hire_date < '01.01.2000'
```

Конечно же, для того, чтобы возвращаемые значения можно было сравнить с условием, используются специальные операторы сравнения. В OracleSQL почему-то термин "операторы сравнения" (comparison operators) не употребляется. Вместо этого для них предусмотрен специальный термин "условия" (conditions).

Операторов (условий) сравнения в Oracle SQL предусмотрено множество, и далеко не все из них имеют аналоги в других системах управления базами данных, например, в Microsoft SQL Server.

5.2 Простые условия в фильтре WHERE (простые операторы сравнения)

Простые условия (операторы сравнения) в запросах Oracle SQL

Простые условия сравнения в Oracle вполне привычны:

Условие	Значение
=	Равно
<	Меньше
>	Больше
<=	Меньше или равно
>=	Больше или равно
<> != ^= ¬=	Не равно

Отметим только два момента:

- иногда у пользователей бывают проблемы с тем, как именно пишется больше или равно (или меньше или равно) — что идет вначале: символ равенства или знак больше/меньше. Здесь можно просто запомнить принцип: как говорим, так и пишем;
- "не равно" в Oracle может обозначаться сразу четырьмя разными способами, в том числе достаточно экзотическими. Однако рекомендуется по возможности использовать стандартное обозначение вида (<>).

Пример применения простого условия сравнения уже приводился в предыдущем разделе:

```
SELECT * FROM hr.employees WHERE hire_date < '01.01.2000'
```

Задания 5.1 Применение простых условий

Простые условия в запросах Oracle SQL

Задание:

Напишите запрос, возвращающий фамилию и имя тех сотрудников из таблицы hr.employees, для которых размер оклада (столбец salary) больше или равен 10000. Результат запроса должен быть таким, как представлено на рис. Лаб. 5.1.

	Имя	Фамилия	Оклад
1	Michael	Hartstein	13000
2	Hermann	Baer	10000
3	Shelley	Higgins	12000
4	Steven	King	24000
5	Neena	Kochhar	17000
6	Lex	De Haan	17000
7	Nancy	Greenberg	12000
8	Den	Raphaely	11000
9	John	Russell	14000

Рис. Лаб. 5.1

5.3 Групповые условия в Oracle SQL

Групповые условия (операторы сравнения) в запросах Oracle SQL, ключевые слова ALL, ANY, SOME

Для групповых условий сравнения в Oracle SQL используются три ключевых слова: ALL, SOME и ANY. Все эти ключевые слова применяются для сравнения указанного вами значения с набором значений, который возвращает подзапрос (или явно указанный набор значений):

- ALL — сравнение будет производиться со всеми записями, которые возвращает подзапрос (или просто со всеми значениями в набор). True вернется только в том случае, если все записи, которые возвращает подзапрос, будут удовлетворять указанному вами условию. Кроме того, в Oracle значение True вернется в ситуации, когда подзапрос не вернет ни одной записи.

В качестве примера приведем такой запрос:

```
select * from hr.employees where salary <= ALL(SELECT salary FROM hr.employees WHERE job_id = 'SH_CLERK')
```

Он вернет записи для всех сотрудников, для которых зарплата меньше или равна самой маленькой зарплате у сотрудников с должностью SH_CLERK.

- ANY — сравнение вернет True, если условию будет удовлетворять любая запись из набора (или подзапроса). Например, такой запрос вернет всех пользователей, зарплата которых совпадает с зарплатой клерка:

```
select * from hr.employees where salary = ANY(SELECT salary FROM hr.employees WHERE job_id = 'SH_CLERK')
```

- SOME — сравнение вернет True, если условию будут удовлетворять некоторые записи из набора (или подзапроса).

Такие же возможности предусмотрены и в других системах управления базами данных, например, Microsoft SQL Server.

Задания 5.2 Применение групповых условий в Oracle

Групповые условия в запросах Oracle SQL

Задание:

Создайте запрос, который вернет из таблицы hr.employees информацию об имени, фамилии, окладе и дате приема на работу для сотрудников, которые были приняты на работу раньше, чем любой из сотрудников департамента номер 70. Результат запроса должен быть таким, как представлено на рис. 2.3-1.

	Имя	Фамилия	Оклад	Дата приема на работу
1	Lex	De Haan	17000	13-JAN-93
2	Bruce	Ernst	6000	21-MAY-91
3	Alexander	Hunold	9000	03-JAN-90
4	Neena	Kochhar	17000	21-SEP-89
5	Jennifer	Whalen	4400	17-SEP-87
6	Steven	King	24000	17-JUN-87

Рис. 2.3-1

5.4 Сравнение значений с плавающей запятой: IS NAN или IS INFINITE

Ключевые слова IS NAN (не является числом) и IS INFINITE (значение равно бесконечности) в запросах Oracle SQL

Для сравнения значений с плавающей запятой в Oracle SQL предусмотрено два специальных условия:

- IS NAN (от is Not a Number) — значение не является числом (а, например, получилось в результате какой-нибудь недопустимой операции, например, деления на 0);
- IS INFINITE — значение равно бесконечности (все равно, плюс или минус).

Каждое из этих условий можно обратить:

IS NOT NAN или IS NOT INFINITE

Конечно же, для значений с плавающей запятой можно использовать и обычные условия сравнения. Однако всегда нужно помнить, что при работе с такими значениями связаны потенциальные проблемы, связанные с округлением, поэтому при возможности значений с плавающей запятой лучше избегать.

5.5 Логические условия в Oracle SQL

Логические условия в фильтрах WHERE в запросах Oracle SQL, ключевые слова AND, OR, NOT

Логические условия в Oracle SQL позволяют определить требуемую комбинацию из нескольких обычных условий. Таких условий в Oracle SQL предусмотрено три:

- AND — логическое И. Чтобы выражение вернуло True, нужно, чтобы истинными были оба условия:

```
SELECT * FROM hr.employees WHERE department_id = 50 AND JOB_ID = 'SH_CLERK'
```

- OR — логическое ИЛИ. Чтобы выражение вернуло True, достаточно, чтобы истинным было только одно условие:

```
SELECT * FROM hr.employees WHERE department_id = 50 OR JOB_ID = 'SH_CLERK'
```

- NOT — "обращает" условие. При добавлении NOT выражение будет возвращать True, когда без него оно возвращало бы False, и наоборот:

```
SELECT * FROM hr.employees WHERE NOT department_id = 50
```

5.6 Условия для работы со значениями типа NULL (пустыми значениями)

Пустые значения в запросах Oracle SQL, специальное значение NULL, выражения IS NULL и IS NOT NULL

Если в какой-то ячейке в таблице базы данных нет значения, то считается, что в этой ячейке находится специальное значение NULL. NULL — это не 0 и не пустая строка. NULL считается специальным значением, для которого существуют специальные приемы работы.

Применить специальные операторы сравнения для значений типа NULL не получится: NULL никогда не равен другому значению NULL. Поэтому для работы с ним предусмотрены два специальных условия:

- IS NULL — это условие вернет True, если проверяемое значение равно NULL;
- IS NOT NULL — это условие вернет True, если проверяемое значение не равно NULL.

Например, вернуть все записи из таблицы Locations, для которых не указан почтовый индекс, можно так:

```
SELECT * FROM hr.locations WHERE postal_code IS NULL;
```

А вернуть все остальные записи — так:

```
SELECT * FROM hr.locations WHERE postal_code IS NOT NULL;
```

Такие же возможности для работы с пустыми значениями предусмотрены в Microsoft SQL Server.

Задания 5.3 Логические условия в Oracle

Логические условия в запросах Oracle SQL

Задание:

Напишите запрос, который бы возвращал информацию об имени, фамилии, окладе и ставке комиссии сотрудников из таблицы hr.employees. При этом должна возвращаться информация только для сотрудников, для которых сумма оклада больше или равна

10000 и для которых установлена ставка комиссии. Результат выполнения запроса должен выглядеть так, как представлено на рис. 2.4-1.

	Имя	Фамилия	Оклад	Ставка комиссии
1	John	Russell	14000	0.4
2	Karen	Partners	13500	0.3
3	Alberto	Errazuriz	12000	0.3
4	Gerald	Cambraut	11000	0.3
5	Eleni	Zlotkey	10500	0.2
6	Peter	Tucker	10000	0.3
7	Janette	King	10000	0.35
8	Clara	Vishney	10500	0.25
9	Lisa	Ozer	11500	0.25

Рис. 2.4-1

5.7 Поиск по шаблонам с использованием метасимволов.

Операторы LIKE и REGEXP_LIKE

Операторы LIKE и REGEXP_LIKE в запросах Oracle SQL, символы % и _ , регулярные выражения в запросах

Очень часто в практической работе возникает необходимость провести поиск по набору символов в любом месте столбца — например, для поиска всех записей с названиями товаров, содержащими какое-либо слово, или для поиска какого-либо слова как в единственном, так и во множественном числе. Для этой цели в Oracle SQL можно использовать оператор LIKE (его можно использовать и в SQL Server) и оператор REGEXP_LIKE (этого оператора в SQL Server не предусмотрено).

Сразу скажем, что LIKE и REGEXP_LIKE используются только в относительно простых ситуациях. Если вам нужно производить поиск с учетом грамматики, фильтрации шумовых слов и т.п., то следует подумать о применении полнотекстового поиска.

Оператор LIKE используется для поиска символьных значений по шаблону со специальными подстановочными символами (метасимволами). Отметим, что в Oracle предусмотрено четыре разновидности этого оператора:

- "обычный" LIKE — применяется для запроса к строковым столбцам с традиционными кодировками;
- LIKEC — применяется для столбцов с кодировкой Unicode (в терминологии Oracle — Unicode complete);
- LIKE2 — для кодировки UCS2;
- LIKE4 — для кодировки UCS4.

Пример применения этого оператора может выглядеть так:

```
select last_name from hr.employees where last_name like 'R%';
```

В этом примере last_name после ключевого слова where — это столбец, из которого берутся строковые значения для проверки, а 'R%' — это условие для проверки. Отличительной особенностью оператора LIKE является то, что он может включать в условие специальные подстановочные символы (метасимволы). Для этого оператора предусмотрено только два подстановочных символа:

- % — представляет любую последовательность из нуля или более символов. При этом значение '%' никогда не будет равно NULL (для проверки таких значений используется IS NULL);
- _ — представляет любой одиночный символ.

В нашем примере мы использовали подстановочный символ %, поэтому запрос вернет всех сотрудников, чья фамилия начинается на R.

Если мы хотим, чтобы символ процента (%) и подчеркивания (_) воспринимались Oracle как обычные символы, по которым можно производить поиск (а не как служебные подстановочные символы), в нашем распоряжении — параметр ESCAPE. При помощи этого параметра можно определить специальный символ, после которого подстановочный символ будет считаться обычным:

```
select last_name from hr.employees where last_name like 'R\%' ESCAPE
'\';
```

В нашем примере мы определили символ \ в качестве отмены подстановочного значения, и если у нас найдется сотрудник с фамилией R%, то запрос его обязательно вернет. Чаще всего для символа отмены подстановочного значения используется обратный слэш (\) — наиболее привычное значение для программистов на C, однако вы вполне можете использовать любой символ, определив его при помощи ESCAPE.

Условие LIKE использовать совсем несложно, но и набор возможностей у него очень ограничен. Намного большие функциональные возможности предоставляет условие REGEXP_LIKE, которое позволяет определять условие поиска при помощи стандартных POSIX-совместимых регулярных выражений. Регулярные выражения — очень большая тема, которая выходит за рамки этого учебного курса. Полную справку по регулярным выражениям, поддерживаемым Oracle, можно посмотреть в приложении C книги SQL Reference документации Oracle. Здесь же мы приведем только простой пример:

```
SELECT first_name FROM employees
WHERE REGEXP_LIKE (first_name, '^Ste(v|ph)en$');
```

В этом примере мы возвращаем информацию о всех сотрудниках, у которых имя начинается на Ste, заканчивается на en, а между ними находится либо v, либо ph. Результат может быть таким:

Steven

Steven

Stephen

Задания 5.4 Поиск по шаблонам

Поиск по шаблонам и применение регулярных выражений в запросах Oracle SQL

Задание 1:

Напишите запрос, который бы возвращал информацию об имени, фамилии и должности (столбец JOB_ID) для каждого сотрудника на основе информации из таблицы employees в в схеме hr. При этом должна возвращаться информация только для тех сотрудников, которые работают в сфере продаж (наименование должности начинается на SA) или которые выполняют роль менеджера (в должности присутствует сочетание MGR). Результат запроса должен выглядеть так, как представлено на рис. Лаб. 2.5-1.

	Имя	Фамилия	Телефон
1	John	Russell	011.44.1344.429268
2	Karen	Partners	011.44.1344.467268
3	Alberto	Errazuriz	011.44.1344.429278
4	Gerald	Cambraut	011.44.1344.619268
5	Eleni	Zlotkey	011.44.1344.429018
6	Peter	Tucker	011.44.1344.129268
7	David	Bernstein	011.44.1344.345268
8	Peter	Hall	011.44.1344.478968
9	Christopher	Olsen	011.44.1344.498718

Рис. 2.5-1

Задание 2:

Напишите запрос, который возвращал бы информацию о имени, фамилии, телефонном номере для каждого сотрудника в таблице hr.employees. При этом должна выводиться информация только для тех сотрудников, телефонный номер которых не соответствует маске XXX.XXX.XXXX, где X-любое число от 0 до 9. Результат выполнения запроса должен выглядеть так, как представлено на рис. 2.5-2.

	Имя	Фамилия	Должность
1	Shelley	Higgins	AC_MGR
2	Nancy	Greenberg	FI_MGR
3	John	Russell	SA_MAN
4	Karen	Partners	SA_MAN
5	Alberto	Errazuriz	SA_MAN
6	Gerald	Cambraut	SA_MAN
7	Eleni	Zlotkey	SA_MAN
8	Peter	Tucker	SA_REP
9	David	Bernstein	SA_REP
10	Peter	Hall	SA_REP

Рис. 2.5-2

Примечание.

В методических материалах выше полный справочник по регулярным выражениям не приводится. Попробуйте найти решение самостоятельно, используя книгу Application Developer's Guide - Fundamentals из официальной документации Oracle. Если это не удастся, просто просмотрите ответ.

5.8 Условие BETWEEN для сравнения диапазонов

Оператор BETWEEN для сравнения диапазонов в запросах ORACLE SQL

Иногда необходимо вернуть все значения, которые попадают в какой-либо диапазон. Для этой цели очень удобно использовать специальный оператор для сравнения диапазонов BETWEEN. Например, чтобы вернуть информацию о всех сотрудниках, которые получают зарплату в диапазоне от 3000 до 4000, можно использовать запрос такого вида:

```
SELECT * FROM hr.employees WHERE salary BETWEEN 3000 AND 4000;
```

Обратите внимание, что при использовании ключевого слова BETWEEN граничные значения также попадают в диапазон, поэтому запрос вернет в том числе сотрудников с зарплатой в 3000 и 4000.

Конечно же, мы всегда можем заменить запрос с ключевым словом BETWEEN на запрос с обычными операторами сравнения:

```
SELECT * FROM hr.employees WHERE salary >= 3000 AND salary <= 4000;
```

Однако применение BETWEEN тоже вполне допустимо.

Такие же возможности работы с оператором BETWEEN предусмотрены и в Microsoft SQL Server.

5.9 Условие IN

Работа со списками значения в запросах Oracle SQL, условие IN

Если условие BETWEEN предназначено для работы с диапазонами, то условие IN используется для проверки значения на соответствие списку значений. Например, вернуть все записи для сотрудников с должностями Sales Manager, Sales Representative и Shipping Clerk можно так:

```
SELECT * FROM hr.employees WHERE job_id IN ('SA_MAN', 'SA_REP', 'SH_CLERK');
```

Чтобы вернуть все значения, которые не попадают в указанный список, можно использовать ключевое слово NOT. Например, чтобы вернуть информацию о всех сотрудниках с другими должностями, можно использовать запрос

```
SELECT * FROM hr.employees WHERE job_id NOT IN ('SA_MAN', 'SA_REP', 'SH_CLERK');
```

Конечно же, как и в случае с BETWEEN, синтаксические конструкции с IN можно заменить на обычные операторы сравнения с OR. Однако при сравнении с большим списком значений синтаксис с IN явно удобнее.

Такие же возможности для работы с условием IN предусмотрены и в Microsoft SQL Server.

5.10 Условие Exists и проверка существования набора значений

Условие EXISTS и проверка существования набора значений в ORACLE SQL

Условие EXISTS используется только в одной ситуации — когда вы используете в запросе и подзапрос и хотите проверить, возвращает ли подзапрос записи. Если подзапрос возвращает хотя бы одну запись, то условие EXISTS вернет True, если нет — то false. Пример применения этого условия может выглядеть так:

```
SELECT * FROM departments d WHERE EXISTS  
  
(SELECT * FROM employees e WHERE d.department_id = e.department_id);
```

В этом примере мы возвращаем информацию о всех департаментах, для которых в таблице employees есть хотя бы один сотрудник.

Как обычно, это условие можно обращать при помощи NOT:

```
SELECT * FROM departments d WHERE NOT EXISTS
```



```
(SELECT * FROM employees e WHERE d.department_id = e.department_id);
```

Те же возможности предусмотрены и для Microsoft SQL Server.

5.11 Ограничение количества выводимых данных: псевдостолбец ROWNUM

Псевдостолбец ROWNUM для ограничения количества выводимых данных в запросах ORACLE SQL

Oracle умеет нумеровать строки в возвращаемых результатах. Для этого используется специальный псевдостолбец ROWNUM. Однако использовать этот столбец напрямую в запросе невозможно: например, такой запрос:

```
SELECT ROWNUM, * FROM hr.employees
```

вернет ошибку.

Вместо этого ROWNUM используется для ограничения количества выводимых записей, например:

```
SELECT * FROM employees WHERE ROWNUM < 10;
```

В этом случае будет выведено всего 9 записей.

Такую возможность очень удобно использовать, когда запрос потенциально может возвращать очень большое количество значений и сильно загрузить сервер, а вам нужно просто убедиться в том, что он выполняется нормально.

Задания 5.5 Работа с диапазонами и ограничение количества выводимых записей

Работа с диапазонами и ограничение количества выводимых записей в Oracle SQL

Задание:

Напишите запрос, который возвращал бы информацию об имени, фамилии, должности (столбец JOB_ID) и времени приема на работу (столбец HIRE_DATE) для сотрудников из таблицы hr.employees. При этом должна возвращаться только информация для тех сотрудников, которые были приняты на работу с 1995 по 1999 годы включительно, плюс тех, чьи должности выглядят как AD_PRES, AD_VP и AD_ASST. Ограничьте число выводимых записей пятью. Используйте для этого запроса выражения BETWEEN и IN. Результат такого запроса должен выглядеть так, как представлено на рис. 2.6-1.

	Имя	Фамилия	Должность	Дата приема на работу
1	Donald	OConnell	SH_CLERK	21-JUN-99
2	Jennifer	Whalen	AD_ASST	17-SEP-87
3	Michael	Hartstein	MK_MAN	17-FEB-96
4	Pat	Fay	MK_REP	17-AUG-97
5	Steven	King	AD_PRES	17-JUN-87

Рис. 2.6-1