

## 4.1 Базовый синтаксис запросов SELECT

### ***Основы запросов SELECT в Oracle SQL, базовый синтаксис запросов, список столбцов, псевдонимы и литералы, оператор q', ключевое слово DISTINCT***

Основное средство для получения информации из базы данных — это запросы, то есть команда SELECT. Базовые возможности этой команды будут рассмотрены в этом разделе, другие, более сложные возможности — в следующих разделах. В каждом разделе будут указываться отличия по сравнению с аналогичными возможностями в Microsoft SQL Server.

Поскольку во всех главных базах данных поддерживается стандарт ANSI SQL, то базовые возможности и в Oracle, и в других базах данных (MicrosoftSQL Server, IBM DB2, Informix, Sybase, Microsoft Access) будут схожи. Отличия — в частности, которые стандарт SQL не определяет, но которые довольно существенны, и которые могут привести к сложностям для пользователей, которые привыкли создавать запросы к другим базам данных.

Базовый синтаксис команды SELECT в Oracle выглядит следующим образом:

```
SELECT [DISTINCT]
список_столбцов FROM источник WHERE фильтр ORDER BY выражение_с
ортировки
```

Приведем пример такого запроса:

```
SELECT employee_id, first_name, last_name, hire_date FROM
hr.employees WHERE hire_date < '01.01.2000' ORDER BY last_name
```

Как мы видим, пока в запросе не используются специальные возможности, он выглядит абсолютно стандартным и полностью идентичен аналогичным запросам, например, в SQL Server. Если нам нужно выбрать все столбцы из базы данных или представления, мы можем указать вместо списка столбцов звездочку:

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

Выводимое имя столбца можно поменять при помощи псевдонима (alias). Для этого достаточно просто написать имя псевдонима сразу после названия столбца или использовать ключевое слово AS (оба синтаксиса полностью равнозначны):

```
SELECT last_name фамилия FROM hr.employees
```

```
SELECT last_name AS фамилия FROM hr.employees
```

Особенно удобно использовать псевдонимы для вычисляемых столбцов, для которых изначально имя не предусмотрено:

```
SELECT last_name AS Фамилия, salary*12 AS "Зарплата за год" FROM hr.employees
```

Если в псевдониме используются пробелы или какие-то служебные символы, то такой псевдоним нужно поместить в двойные кавычки.

Отметим и некоторые особенности, связанные со списком столбцов в Oracle. Если в SQL Server "проблемные" имена столбцов (с пробелами и зарезервированными словами) можно было помещать в квадратные скобки (а обычно и в двойные кавычки — в зависимости от настроек сеанса), то в Oracle и тот, и другой вариант вызовет ошибку.

В список столбцов в Oracle можно включать литералы (literals). Литералы — это любое строковое значение, дата или число, которое помещается в список столбцов и при этом не является ни названием столбца, ни псевдонимом. Литералы будут повторяться для каждого возвращаемого столбца. Обычно они используются для создания пояснений в возвращаемом наборе результатов:

```
SELECT last_name AS Фамилия, 'Зарплата за год: ', salary*12 FROM hr.employees
```

В этом примере 'Зарплата за год: ' — это литерал, который будет выводиться для каждой записи.

Строковые литералы и литералы с датой должны заключаться в одинарные кавычки. Литералы в виде числовых значений в кавычки не помещаются.

Отметим, что литералы часто сливаются в выводимых результатах с возвращаемыми из базы данных значениями при помощи оператора конкатенации (||). При работе с литералами иногда возникает проблема, связанная с тем, что литерал сам по себе может содержать зарезервированный символ, например, одинарные кавычки. Oracle, в отличие, например, от SQL Server, позволяет пользователю самостоятельно определять символ, который будет использоваться в качестве кавычек. Выглядеть такое переопределение может, например, так:

```
select department_name, q'[It's assigned manager ID: ]', manager_id from departments
```

Попытка применить обычные одинарные кавычки для литерала приведет к ошибке, поскольку кавычка встречается внутри самого символического значения литерала. Поэтому в этом примере в качестве кавычек выступают квадратные скобки.

Отметим некоторые особенности применения оператора `q`:

- можно использовать как строчную букву `q`, так и заглавную (`Q`);
- после `q` должна идти обычная одинарная кавычка. Следующая одинарная кавчка определяет завершение области действия оператора `q`;
- сразу после одинарной кавычкой должен идти символ, выбранный пользователем для применения в качестве кавычек. Этот символ может быть любым, за исключением пробела, перевода строки или табуляции (в том числе и одинарная кавычка). Если пользователь выбрал символ `(`, `{`, `[` или `<`, то закрываться кавычки должны соответствующим символом `)`, `}`, `]` и `>`. В других случаях закрытие кавычек производится при помощи того же символа, который использовался для их открытия.

И последний момент, связанный с базовым синтаксисом запросов. По умолчанию запрос возвращает из базы данных все значения, соответствующие параметрам запроса, даже если некоторые из возвращаемых значений будут полностью одинаковыми. Например, запрос

```
SELECT salary FROM hr.employees
```

вернет 107 строк, несмотря на то, что значительная часть значений будет повторяться.

Чтобы вернуть только уникальные значения (или наборы значений, если возвращается несколько столбцов), в запросе можно использовать ключевое слово `DISTINCT`:

```
SELECT DISTINCT salary FROM hr.employees
```

Такой запрос вернет уже 57 значений, при этом все значения будут уникальными.

Отличительной особенностью Oracle является то, что вместо `DISTINCT` можно использовать ключевое слово `UNIQUE`. SQL Server такого не позволяет.

## 4.2 Арифметические операторы Oracle. Специальная таблица `DUAL`

### ***Арифметические операторы в запросах Oracle SQL, унарные и бинарные операторы, специальная таблица `DUAL`***

Операторы — это синтаксические конструкции языка, которые предназначены для выполнения определенных действий.

В Oracle SQL предусмотрено множество операторов, как вполне привычных (например, арифметические или логические), так и экзотических (операторы для работы с иерархическими наборами записями, со множествами записей —

MULTISET). Кроме того, пользователям разрешено определять свои собственные операторы. Каждый из типов операторов будет рассмотрен в своем разделе. В этом разделе будет рассмотрен только самый простой тип операторов — арифметические.

Все операторы Oracle SQL можно поделить на два главных типа: унарные и бинарные. Унарные операторы работают только с одним значением. Например, к унарным операторам относится оператор (-): если для него использовать только одно значение, он поменяет знак числа.

Бинарные операторы работают с двумя значениями. Например, в выражении 10-7 оператор (-) — это бинарный оператор, который производит вычитание.

В Oracle SQL предусмотрен стандартный набор арифметических операторов, которые знакомы нам с детства:

Оператор	Действие
+	сложение
-	вычитание
*	умножение
/	деление

К арифметическим также относятся унарные операторы (+) и (-), которые определяют знак числа. Приоритет арифметических операторов также привычен: наивысший приоритет у унарных операторов, затем выполняется умножение и деление, а после этого — сложение и вычитание. Порядок применения операторов регулируется круглыми скобками.

Применение унарного оператора к значению типа NULL даст в результате NULL.

Конечно, нельзя использовать два оператора - последовательно (как бинарный, а затем унарный оператор) — такая последовательность символов зарезервирована за комментариями SQL.

Арифметические операторы можно использовать даже в таких запросах, в которых вообще не используются столбцы:

```
SELECT 10*15 FROM hr.employees;
```

Однако в этом случае результат будет повторен 107 раз, что, скорее всего, нам не нужно. Для выполнения запросов, в которых вообще не используются столбцы из базы данных, в Oracle зарезервирована специальная таблица DUAL (в SQL Server она не предусмотрена). В этой таблице — всего один столбец DUMMY с единственным значением х. Наш запрос с использованием этой служебной таблицы может выглядеть так:

```
SELECT 10*15 FROM DUAL;
```

### 2.3 Оператор конкатенации

#### ***Операторы конкатенации (слияния строковых значений) в запросах Oracle SQL, слияние строковых данных разных типов***

В реальной работе очень часто приходится производить слияние строковых значений. Такая операция называется конкатенацией.

В Oracle SQL предусмотрен несколько необычный (по крайней мере, с точки зрения специалистов, работавших с SQL Server) оператор конкатенации — две вертикальные черты (||). Отметим, что в некоторых версиях сервера Oracle, например, предназначенных для работы на мэйнфреймах IBM, используется другой оператор конкатенации. Поэтому на всякий случай в Oracle SQL предусмотрена еще и встроенная функция CONCAT, которая также производит конкатенацию и работает на любых платформах.

Пример применения оператора конкатенации может выглядеть так:

```
SELECT first_name || ' ' || last_name FROM hr.employees;
```

В этом примере у нас сливается три значения: имя, пробел (в виде литерала) и фамилия.

Оператор конкатенации можно применять как для строковых значений и значений CLOB (больших строковых значений), так и для чисел и дат (эти значения будут автоматически конвертированы в строковые). При слиянии строкового значения со значением типа NULL Oracle вернет строковое значение (в отличие от поведения по умолчанию SQL Server), однако Oracle предупреждает, что такое поведение в будущих версиях Oracle не гарантируется. Поэтому в таких ситуациях рекомендуется использовать функцию NVL, которая преобразует значения типа NULL в строки нулевой длины.

При выполнении конкатенации рекомендуется подумать над ограничениями типов данных, которые используются для конкатенации. Oracle использует для результата конкатенации больший тип данных из тех значений, которые используются для конкатенации. Например, при слиянии двух значений с типом данных CHAR (максимальный размер — 2000 символов) вернется CHAR, при слиянии CHAR и VARCHAR2 (максимальный

размер — 4000 символов) вернется VARCHAR2, а при слиянии любого типа данных с CLOB вернется CLOB.

Пробелы при конкатенации всегда сохраняются.

### **Завдання:**

Напишите запрос с использованием столбцов таблицы employees в схеме hr, который бы возвращал результаты в следующем виде (см. рис. Лаб. 2.1-1)

	ФИО	ОКЛАД	Оклад минус подоходный
1	Donald OConnell	2600	2262
2	Douglas Grant	2600	2262
3	Jennifer Whalen	4400	3828
4	Michael Hartstein	13000	11310
5	Pat Fay	6000	5220
6	Susan Mavris	6500	5655
7	Hermann Baer	10000	8700
8	Shelley Higgins	12000	10440
9	William Gietz	8300	7221
10	Steven King	24000	20880

Рис. Лаб. 4.1.

При этом сумму подоходного налога нужно считать равной 13 процентам.