

Лекция 5

Формирование списка полей оператора SELECT.

Оператор SELECT используется для отбора записей, удовлетворяющих сложным критериям поиска, и имеет следующий формат:

```
SELECT [модификаторы] { * \ < Список полей > }  
FROM < Список таблиц >  
[WHERE < Условия отбора >]  
[GROUP BY <Список полей для группирования >]  
[HAVING < Условия группирования>]  
[UNION <Вложенный оператор SELECT>]  
[ORDER BY < Список полей для сортировки >]
```

Результат выполнения SQL-запроса, заданного оператором SELECT, представляет собой выборку записей, отвечающих заданным в нем условиям. При рассмотрении оператора SELECT будем предполагать, что SQL-запрос набран и выполнен с помощью в инструменте SQL Commands компонента SQL Workshop облачной среды APEX.ORACLE.COM. Регистрация в этой среде – это задача лабораторной работы №1. Напоминаем, что средой выполнения запросов является СУБД ORACLE. Помимо этого в Вашей рабочей области должны находиться база данных HR и таблица TB_ELEKTROSTAL_2018.

Модификаторы запроса.

В качестве модификаторов запроса можно использовать ключевое слово DISTINCT и средства ограничения количества выводимого количества строк. Последние средства часто используют для разбиения выводимых строк на страницы.

Ключевое слово DISTINCT.

Обычно в результирующий набор данных выводятся все строки отвечающие требованиям, указанным в соответствующих разделах SQL запроса. В результирующем наборе данных могут быть разрешены или

запрещены повторяющиеся записи. Этим режимом управляет модификатор DISTINCT. Если он отсутствует, то в наборе данных разрешаются повторяющиеся записи. В случае, если в запросе указано ключевое слово DISTINCT повторяющиеся записи в результирующем наборе запрещены.

Запрос к базе данных сайта stackoverflow о странах, которые указаны в таблице users, выглядит следующим образом:

```
select location from Users
```

Этот запрос можно модифицировать, чтобы выводить только уникальные названия стран. Этим самым исключается дублирование строк.

```
select distinct location from Users
```

Этот запрос выполняется довольно длительное время. Причиной этого являются вычислительные трудности, возникающие при поиске уникальных записей.

Если требуется подсчитать количество уникальных записей, можно использовать следующий формат оператора:

```
Select count(distinct location) from users
```

Ограничение вывода количества выводимых строк.

В большинстве SQL серверов поддерживаются инструменты ограничения количества выводимых строк. К сожалению, эти средства различаются в различных SQL серверах. В СУБД ORACLE можно использовать ключевое слово ROWNUM.

Например, вывести 10 строк таблицы TB_ELEKTROSTAL_2018:

```
select * from tb_elektrostal_2018  
where rownum<=10
```

Отметим, что организовать постраничный вывод строк в ORACLE довольно сложно. Это объясняется отсутствием удобных базовых средств.

Такое же положение относится и к СУБД MS SQL. Здесь предусмотрены средства для ограничения количества выводимых записей. Например, вывести 10 первых строк из таблицы users:

```
select top 10 * from users
```

Организовать же постраничный вывод также как и в СУБД ORACLE довольно трудно.

В СУБД изначально разработанных для использования в сетевых интернет приложениях разбиение на страницы поддерживается специальными базовыми средствами.

Например, в СУБД MySQL и Postresql можно использовать следующий оператор для вывода второй страницы списка пользователей (при размере страницы 10 строк)

```
select * from users limit 10 skip 10
```

Здесь ключевое слово limit ограничивает количество выводимых записей (размер страницы). А ключевое слово skip пропускает заданное количество записей (организуя переход к требуемой странице).

Тот же запрос в СУБД Firebird будет выглядеть следующим образом:

```
select first 10 skip 10 * from users
```

Здесь вместо ключевого слова limit используется ключевое слово first. Также изменяется место расположения модификатора. Он располагается непосредственно после ключевого слова select. В то время как в СУБД MySQL и Postgresql модификатор располагается в конце оператора select.

Отметим, что используя ограничитель количества выводимых записей, можно составить запрос на получение самой «лучшей» или «худшей» записи по заданному критерию. Для этого записи сортируются по заданному критерию и выводится первая запись.

Составление списка полей запроса

В оператор SELECT обязательно включается список полей. Элементы списка полей, разделяются запятой. К качестве элемента списка полей может быть использовано любое SQL выражение, возвращающее единственное значение. Например, таким выражением может быть внутренний запрос. Рассмотрим более детально варианты составления списка полей.

В качестве элемента списка полей может использоваться константа любого допустимого типа. Например:

```
select '->',10 from employees
```

```
select '->',first_name from employees
```

Допускает присвоение псевдонима элементу списка полей

```
select '->' h,first_name from employees
```

Использование арифметических и строковых выражений:

```
select 10*20+30 from dual
```

Отметим, что dual – специальная таблица в СУБД ORACLE, которая состоит из одной строки. Эта таблица предназначена для использования в запросах, в которых источник данных не требуется. Т.е. она применяется в качестве мнимой или фиктивной таблицы.

Для конкатенации строк используется оператор || - две вертикальных черты.

```
select first_name||' '||last_name from employees
```

В современных СУБД поддерживается стандартный набор математических функций таких, как sin(), cos(), log() и т.д. Эти функции можно использовать в математических выражениях в элементах списка полей.

```
select sin(employee_id),first_name||' '||last_name from employees
```

Однако, использование тригонометрических функций не характерно для приложений баз данных. Довольно широко используются функции округления и взятия абсолютной величины.

Обычно в качестве элемента списка полей используется имя поля из источника данных, сформированного в разделе FROM оператора SELECT.

```
select first_name,last_name,salary from employees
```

Рассмотрим использование широко распространенных функций, которые специфичны для приложений баз данных.

Анализ неопределенных значений. Часто логика запроса требует специальной обработки неопределенных null значений. Для этой цели

используется функция `coalesce`. `COALESCE` возвращает первое не-null выражение из указанного списка. Пользователь должен указать по крайней мере два выражения в списке параметров функции `coalesce()`. Если все параметры функции `coalesce` возвращают null, то функция также возвращает null.

```
SELECT product_id
       , list_price
       , min_price
       , COALESCE(0.9*list_price, min_price, 5) "Sale"
FROM oe.product_information
WHERE supplier_id = 102050
ORDER BY product_id, list_price, min_price, "Sale"
```

В данном примере используется база данных `oe`. Эту базу данных можно загрузить в среду `SQL Workshop`. Также можно зарегистрироваться в облачном сервисе `livesql.oracle.com`. Здесь база данных `oe` наряду с `hr` установлена по умолчанию.

Файл: 5 Список полей
Каталог: E:\Projects\documents\md\predmety\BD_2018\Materials\Lectons
Шаблон: C:\Users\sss\AppData\Roaming\Microsoft\Шаблоны\Normal.dotm
Заголовок:
Содержание:
Автор: sss
Ключевые слова:
Заметки:
Дата создания: 11.10.2018 21:30:00
Число сохранений: 2
Дата сохранения: 11.10.2018 21:32:00
Сохранил: sss
Полное время правки: 2 мин.
Дата печати: 11.10.2018 21:33:00
При последней печати
 страниц: 5
 слов: 1 031 (прибл.)
 знаков: 5 877 (прибл.)