

ЛАБОРАТОРНАЯ РАБОТА №4

Тема: Структура предложения SELECT. Методы конструирования списка полей запроса.

Цель: Пробрести навыки составления списка полей запроса и использования модификаторов запроса.

В данной лабораторной работе используются база данных HR, созданная в рамках первой лабораторной работы. Также используется таблица TB_ELEKTROSTAL_2018, которая создана и наполнена данными в процессе выполнения лаб. работы №.1.

Дополнительно используется удаленная база данных проекта stackoverflow.com. Структура этой базы данных описана в методических указаниях к выполнению лабораторной работы № 3. Напомним, адрес ресурса <http://data.stackexchange.com/stackoverflow/query/new>.

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. В 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
```

Допускается присвоение псевдонима элементу списка полей. Псевдоним используется для изменения заголовка колонки, а также может использоваться для ссылок на поля внутренних запросов.

```
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 содержит одну строку. 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 базой hr установлена по умолчанию.

Функция разделения даты на отдельные элементы.

Для разделения даты на элементы (год, месяц, день и т.д.) стандартом SQL предусматривается использование функции EXTRACT.

С помощью этой функции можно выделить год, номер месяца, день даты, а также часы, минуты и секунды из поля времени. Функция EXTRACT имеет следующий формат:

```
EXTRACT(<keyword> from <datefield>)
```

Здесь <keyword>- одно из ключевых слов YEAR, MONTH, DAY и др

<datefield> - выражение типа DATE или DATETIME.

Например

```
Select extract(year from hire_date) from employees
```

Функция анализа вариантов.

Во многих языках программирования присутствует оператор ветвления. В языке SQL для анализа выражений и принятия возврата того

или иного выражения используется функция CASE. Данная функция имеет две формы.

Первая форма позволяет проанализировать одно выражение и в зависимости от его значения вернуть то или иное возвращаемое значение.

Формат этой функции:

```
case <expr>
  when <val1> then <ret1>
  when <val2> then <ret2>
  ....
  else <retn>
end
```

Здесь: <expr> - анализируемое выражение

<val1>, <val2>, <valn> - предполагаемые варианты значений выражения

<expr>

<ret1>, <ret2>, <retn> - варианты возвращаемых значений.

В данном варианте функция анализирует выражение <expr> на совпадение со значениями <val1>, <val2>, <valn>. Если, например, совпадает значение val2, то возвращается значение выражения <ret2>. В случае, если нет совпадений ни с одним из вариантов, то возвращается значение выражения из ветви else.

Пример.

```
select case typstr
  when 'улица' then 'ул.'
  when 'проспект' then 'пр.'
  when 'переулок' then 'пер.'
  else 'кв.'
end
from tb_elektrostat_2018
```

Второй вариант функции case обладает более широкими возможностями. Он позволяет анализировать последовательно несколько

независимых логических выражений. Функция возвращает значение, указанное для первого совпадения условия. Формат второго варианта функции case:

```
case
  when <cond1> then <ret1>
  when <cond2> then <ret2>
  ....
  else <retn>
end
```

Здесь <cond1>,...,<condn> логические выражения.

Например

```
select case
  when typstr='улица' then 'ул.'
  when typstr= 'проспект' then 'пр.'
  when typstr= 'переулок' then 'пер.'
  else 'кв.'
end
from tb_elektrostal_2018
```

Порядок выполнения работы

Порядок выполнения работы.

Составить следующие запросы

- Вывести список сотрудников, при этом первым полем должна быть строка '-->', а последним полем – строка '<- -'.
- Вывести список улиц из таблицы tb_elektrostal_2018, требуется исключить дублирование названий улиц. В результирующем запросе должна быть одна строка – название улицы.
- Подсчитать количество типов улиц в таблице tb_elektrostal_2018.

- Составить запрос, в результате выполнения которого получается таблица из двух колонок. В первой колонке – имя (first_name) и фамилия (last_name) сотрудника в виде одной строки. Во втором столбце – заработная плата сотрудника.
- Вывести таблицу из двух колонок. В первой колонке имя и фамилия работника (first_name и last_name), разделенных одним пробелом. Во второй колонке процент вознаграждения. Для тех сотрудников, у которых нет вознаграждения вывести 0. (Использовать таблицу employees из БД HR).
- Вывести таблицу из двух колонок имя работника и номер квартала его дата приема на работу (hire_date). (Использовать функции extract и case).
- Для каждого работника вывести дату приема на работу в формате дд-мес-год. Например, 04-окт-2018.

Содержание отчета:

1. Тема, цель лабораторной работы.
2. Примеры выполнения запросов к базе данных.
3. Составленные согласно заданию запросы и скриншоты полученных результатов.
5. Выводы.

Контрольные вопросы:

1. Какие модификаторы запросов Вы знаете?
2. Дайте характеристику целей использования модификатора DISTINCT.
3. Охарактеризуйте способы страничного вывода результатов запроса в различных моделях СУБД.
4. Опишите назначение и особенности использования функции coalesce.
6. Дайте характеристику функции EXTRACT.
7. Охарактеризуйте функцию CASE. Приведите примеры использования.
8. Операция конкатенация строк.
9. Функция преобразования типов.

Файл: Лабораторная работа 4
Каталог: E:\Projects\documents\md\predmety\BD_2018
Шаблон: C:\Users\sss\AppData\Roaming\Microsoft\Шаблоны\Normal.dotm
Заголовок:
Содержание:
Автор: sss
Ключевые слова:
Заметки:
Дата создания: 08.10.2018 11:35:00
Число сохранений: 95
Дата сохранения: 06.11.2018 20:57:00
Сохранил: sss
Полное время правки: 393 мин.
Дата печати: 06.11.2018 20:57:00
При последней печати
 страниц: 10
 слов: 1 784 (прибл.)
 знаков: 10 173 (прибл.)