

ЛАБОРАТОРНАЯ РАБОТА №12

Тема: Обобщенные табличные выражения (CTE). Разработка SQL запросов на основе внутренних представлений.

Цель: Пробрести навыки составления SQL запросов использующих внутренние представления.

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>.

Обобщенные табличные выражения (CTE) можно представить себе как временные результирующие наборы, определенные в области выполнения единичных инструкций SELECT, INSERT, UPDATE, DELETE или CREATE VIEW. CTE, как и производные таблицы, не сохраняются в базе данных в виде объектов, время их жизни ограничено продолжительностью запроса. Но, в отличие от производных таблиц, CTE могут ссылаться сами на себя, а на них один и тот же запрос может ссылаться несколько раз.

CTE предназначены для:

- Создания рекурсивных запросов. Эта разновидность CTE рассматривается в следующей лабораторной работе.
- Замены представлений в тех случаях, когда использование представления не оправдано, то есть тогда, когда нет необходимости сохранять в метаданных базы его определение.
- Группирования по столбцу, производного от скалярного подзапроса выборки или функции, которая недетерминирована или имеет внешний доступ.

- Многократных ссылок на результирующую таблицу из одной и той же инструкции.

Применение CTE позволяет значительно повысить читаемость и упростить работу со сложными запросами, разбив его на отдельные логические строительные блоки. Из них можно составлять более сложные промежуточные CTE для формирования конечного результирующего набора.

CTE могут быть определены в пользовательских подпрограммах (функциях, хранимых процедурах, триггерах, представлениях).

Структура CTE

CTE состоит из имени выражения, необязательного списка столбцов и определяющего CTE запроса. После определения CTE на него можно ссылаться из инструкций SELECT, INSERT, UPDATE и DELETE как на таблицу или представление. CTE также можно указать в предложении CREATE VIEW в определяющей инструкции SELECT.

Базовая структура синтаксиса ОТВ:

```
WITH expression_name [ ( column_name [, ...n] ) ]  
AS  
( CTE_query_definition )
```

Список имен столбцов необязателен только в том случае, если всем результирующим столбцам в определении запроса присвоены уникальные имена.

Инструкция для обращения к CTE:

```
SELECT <column_list>  
FROM expression_name
```

Пример

В следующем примере показаны компоненты структуры CTE: имя выражения, список столбцов и запрос. CTE vstreet содержит один столбец (typstr) и определяет список типов улиц, имеющих в таблице tb_elektrostal_2018. Выполняемая инструкция ссылается на CTE для вывода списка типов улиц.

```
with vstreet(typstr) as (  
    select distinct typstr from tb_elektrostal_2018
```

```
)  
select * from vstreet
```

Безусловно, составить запрос для вывода списка типов улиц легко и без использования механизма СТЕ. Рассмотрим более сложный пример. В лабораторной работе 8 приведен запрос, который выводит список из двух колонок. В первой колонке – имя работника. Во второй колонке – список занимаемых им в разное время должностей в течение периода работа на данном предприятии. Для формирования списка используется агрегатная функция *listagg*. Перепишем этот запрос на основе использования механизма СТЕ.

```
with v0(emp_id,job_id) as (  
    SELECT a.employee_id  
        , a.job_id  
    from hr.job_history a  
    union all  
    select b.employee_id  
        , b.job_id  
    from hr.employees b  
) ,  
v1(emp_id,job_list) as (  
    select v0.emp_id  
        , listagg(c.job_title,'; ') as job_list  
    from hr.jobs c  
        inner join v0 on v0.job_id=c.job_id  
    group by v0.emp_id  
) ,  
v2(emp_name,job_list) as (  
    select e.first_name||' '||e.last_name  
        , v1.job_list  
    from hr.employees e  
        inner join v1 on e.employee_id = v1.emp_id  
)  
select * from v2
```

В данном примере используется три внутренних представления v0, v1 и v2. Представление v0 формирует список должностей для каждого работника с учетом текущей должности. Представление v1 использует представление v0 для формирования символического списка должностей каждого сотрудника. Представление v2 добавлением к данным представления v1 фамилии

сотрудника решает поставленную задачу. Опорный запрос имеет простейшую форму. Он просто обращается к представлению v2. Рассмотренный пример иллюстрирует пошаговое усложнение запроса для решения поставленной задачи.

Порядок выполнения работы

Составить следующие запросы:

- Составить запрос к БД HR. Результат запроса должен содержать две колонки. Первая колонка - имя и фамилия работника. Вторая колонка – первая должность, которую он занял при поступлении на предприятие. Указание: необходимо использовать механизм CTE предложения SQL.
- Составить запрос к БД HR. Результат запроса должен содержать две колонки. Первая колонка – название должности. Вторая колонка перечень имен и фамилий сотрудников, работающих в этой должности в настоящее время. В качестве разделителя между реквизитами сотрудников использовать символ «точка с запятой» Указание: необходимо использовать механизм CTE при построении предложения SQL.
- Составить запрос к БД stackoverflow. Результат запроса должен содержать список имен пользователей, которые писали посты как в 2010 году так и в 2015 году Для составления запроса обязательно требуется использовать механизм CTE.
- Составить запрос к БД stackoverflow. Результат запроса должен содержать список имен пользователей, которые писали посты в 2010 году, но не написали ни одного поста в 2015 году. Для составления запроса обязательно требуется использовать механизм CTE.
- Составить запрос к БД stackoverflow. Результат запроса должен содержать список имен пользователей, которые не написали ни одного поста в 2010 году, но писали посты в 2015 году. Для составления запроса обязательно требуется использовать механизм CTE.

- С использованием механизма CTE составить запрос к БД HR, который выводит четыре колонки: полное имя сотрудника (выражение, объединяющее поля `first_name` и `last_name` через пробел), оклад сотрудника (поле `salary`), средний оклад по отделу (департаменту), разница между окладом сотрудника и средним окладом по отделу.

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. Опишите назначение механизма CTE.
2. Какие изменения нужно внести в представленный пример запроса, чтобы занимаемые должности были упорядочены в хронологическом порядке?
3. Пример запрос, представленный в данных методических указаниях, работает в среде `livesql.oracle.com`. Но не работает в среде `apex.oracle.com`. Внесите необходимые изменения в запрос, чтобы он работал в среде `apex.oracle.com`. Объясните эти изменения.
4. Представьте формат оператора выбора данных SQL с использованием механизма CTE.
5. С какого ключевого слова начинается запрос, составленный с использованием механизма CTE?
6. Какие части запроса, составленного с использованием механизма CTE, разделяет ключевое слово **as**?
7. Является ли обязательным список параметров в описании внутреннего представления (механизм CTE)? Ответ проиллюстрируйте примерами.

Файл: Лабораторная работа 12
Каталог: C:\Users\sss\Documents
Шаблон: C:\Users\sss\AppData\Roaming\Microsoft\Шаблоны\Normal.dotm
Заголовок:
Содержание:
Автор: sss
Ключевые слова:
Заметки:
Дата создания: 17.11.2018 21:59:00
Число сохранений: 32
Дата сохранения: 18.11.2018 16:31:00
Сохранил: sss
Полное время правки: 84 мин.
Дата печати: 18.11.2018 16:31:00
При последней печати
 страниц: 6
 слов: 1 132 (прибл.)
 знаков: 6 459 (прибл.)