**Практическое занятие № 9**

**Язык SQL. Подзапросы и обобщенные табличные выражения**

**Цель занятия**. Научиться использовать подзапросы в составе оператора запроса. Получить практические навыки использования обобщенных табличных выражений

**9.2 Обобщенные табличные выражения**

Обобщенные табличные выражения (Common Table Expression - CTE) позволяют в более удобной форме составлять сложные запросы. CTE можно рассматривать как определение временных таблиц, которые существуют только для одного запроса. CTE записываются в предложении WITH, и их можно рассматривать как представление или временную таблицу внутри запроса. Преимуществом WITH является возможность многократного обращения к колонкам подзапроса в других частях запроса.

Запрос с использованием предложения WITH может формировать несколько временных таблиц и записывается по следующим правилам:

WITH имя\_таблицы\_1 as (подзапрос\_1),

имя\_таблицы\_2 as (подзапрос\_2),

… -- после последнего подзапроса запятая не ставится

SELECT … -- основной запрос

При этом каждый следующий подзапрос может обращаться к колонкам любого из предшествующих. Основной запрос может использовать как основные таблицы БД, так и вспомогательные таблицы, подготовленные предложением WITH.

Рассмотрим примеры использования WITH на примере схемы о сотрудниках организации.

Организация включает несколько отделов. В штате организации состоят сотрудники. Каждый *Сотрудник* организации работает в некотором отделе (задается колонкой внешнего ключа - Код отдела). В каждом *Отделе* имеется начальник (задается внешним ключом Код начальника - ссылка на сотрудника, который является начальником). Логическая и физическая модели данных представлены соответственно на рис.1 и рис.2.

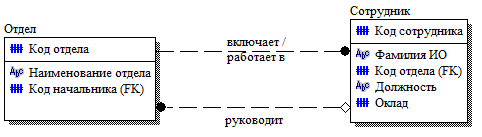


Рис.1 Логическая модель

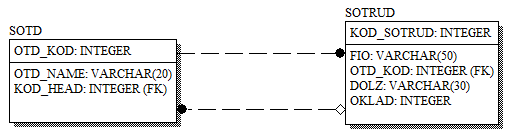
****

Рис.2 Физическая модель

Выполним запросы на следующих данных.

|  |  |  |
| --- | --- | --- |
|  | **Отдел (SOTD)** |  |
| **Otd\_Kod** | **Otd\_Name** | **Kod\_Head** |
| 1 | Техотдел | 1 |
| 2 | Отдел продаж | 6 |
| 3 | Отдел закупок | 10 |

**Сотрудник (SOTRUD)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sotrud\_Kod** | **FIO** | **Otd\_Kod** | **Dolz** | **Oklad** |
| 1 | Марков В.В. | 1 | Нач. отдела | 150000 |
| 2 | Семенов В.А. | 1 | Инженер | 50000 |
| 3 | Исаев А.М. | 2 | Менеджер | 40000 |
| 4 | Акимова А.С. | 3 | Ст.инженер | 70000 |
| 5 | Павленко И.И. | 1 | Вед.инженер | 95000 |
| 6 | Морозова А.И. | 2 | Нач. отдела | 140000 |
| 7 | Сомова Е.Н. | 2 | Вед.менеджер | 65000 |
| 8 | Пименов С.А. | 1 | Инженер | 45000 |
| 9 | Селезнева Н.М. | 2 | Менеджер | 55000 |
| 10 | Конева А.М. | 3 | Нач.отдела | 130000 |

Для тестирования примеров создадим 2 виртуальные таблицы и заполним их данными с использованием предложения WITH. Это удобно для тестирования, т.к. не требует создавать таблицы в БД.

WITH Sotd (otd\_kod,otd\_name,head\_kod) as

(SELECT 1,'Техотдел',1

union SELECT 2,'Отдел продаж',6

union SELECT 3,'Отдел закупок',10),

Sotrud (sotrud\_kod,fio,otd\_kod,dolz,oklad) as

(SELECT 1,'Марков В.В.',1,'Нач.отдела',150000

union SELECT 2,'Семенов В.А.',1,'Инженер',50000

union SELECT 3,'Исаев А.М.',2,'Менеджер',40000

union SELECT 4,'Акимова А.С.',3,'Ст.инженер',70000

union SELECT 5,'Павленко И.И.',1, 'Вед.инженер',95000

union SELECT 6,'Морозова А.И.',2,'Нач.отдела',140000

union SELECT 7,'Сомова Е.Н.',2,'Вед.менеджер',65000

union SELECT 8,'Пименов С.А.',1,'Инженер',45000

union SELECT 9,'Селезнева Н.М.',2,'Менеджер',55000

union SELECT 10,'Конева А.М.',3,'Нач.отдела',130000)

*Пример 5.* Вывести информацию о сотрудниках и численности отдела, в которых они работают.

Вначале создадим оператор запроса с использованием обычного подзапроса. Одну из таблиц в разделе FROM сформируем подзапросом, который возвращает численность всех отделов:

SELECT fio,so.otd\_sotr

FROM Sotrud s,

-- таблица-подзапрос

(SELECT otd\_kod as otd\_sotr, COUNT(\*) AS kol\_sotr

FROM sotrud

GROUP BY otd\_kod) so

WHERE s.otd\_kod = so.otd\_sotr

ORDER BY fio;

*Замечание.*

Т.к. таблиц Отдел и Сотрудник в БД мы не создавали, то для получения результата запроса создадим виртуальные таблицы и заполним их данными, как было описано выше с помощью предложения WITH перед основным запросом:

WITH

-- формирование таблицы Отдел с данными

sotd (otd\_kod,otd\_name,head\_kod) as (SELECT 1…),

-- формирование таблицы Сотрудник с данными

sotrud (kod\_sotrud,fio,otd\_kod,dolz,oklad) as (SELECT 1,…),

-- основной запрос (см.выше)

SELECT ...

Результат запроса



Тот же запрос можно записать с использованием обобщенного табличного выражения. Подзапрос, формирующий вспомогательную таблицу (в примере он определяет численность каждого отдела), перенесем в предложение WITH, а затем используем в основном операторе запроса. Основной запрос обращается к колонкам как базовой таблицы sotrud, так и временной otd\_count

WITH

-- формирование таблицы Отдел с данными

sotd (otd\_kod,otd\_name,head\_kod) as (SELECT 1…),

-- формирование таблицы Сотрудник с данными

sotrud (kod\_sotrud,fio,otd\_kod,dolz,oklad) as (SELECT 1,…),

-- временная таблица, определяющая численность отделов

otd\_count AS (SELECT otd\_kod, COUNT(\*) as kol\_sotr

FROM sotrud

GROUP BY otd\_kod)

-- Основной запрос

SELECT fio,oc.kol\_sotr

FROM sotrud as s JOIN otd\_count as oc

on s.otd\_kod = oc.otd\_kod

ORDER BY fio;

Результат будет таким же, как и при использовании обычного подзапроса.

*Пример 6.* Вывести фамилии сотрудников всех отделов и дополнительно для каждого сотрудника фамилию руководителя отдела и количество сотрудников в отделе.

Также сначала составим оператора запроса с подзапросом.

SELECT s1.fio,

otd\_count.kol\_sotr,

s2.fio AS head\_fio

FROM sotrud s1 JOIN

-- подзапрос формирует виртуальную таблицу

(SELECT otd\_kod, COUNT(\*) AS kol\_sotr

FROM sotrud

GROUP BY otd\_kod) otd\_count

on s1.otd\_kod = otd\_count.otd\_kod

JOIN sotd on s1.otd\_kod = sotd.otd\_kod

JOIN sotrud s2 -- копия таблицы Сотрудник

on sotd.head\_kod = s2.sotrud\_kod

ORDER BY fio;

В данном запросе соединяем таблицы для получения информации:

- таблица сотрудников sotrud s1 используется для получения информации о сотрудниках;

- подзапрос, формирующий виртуальную таблицу otd\_count, используется для получения численности каждого отдела организации;

- таблица отделов sotd используется для получения кода сотрудника, который является начальником отдела;

- копия таблицы сотрудников sotrud s2 используется, чтобы получить фамилию сотрудника-начальника отдела.

Результат запроса:



Перепишем запрос с использованием предложения WITH.

WITH

-- Создание базовых таблиц

sotd (otd\_kod,otd\_name,head\_kod) as (

SELECT 1…),

sotrud (kod\_sotrud,fio,otd\_kod,dolz,oklad) as (

SELECT 1,…),

-- Создание временной таблицы

otd\_count AS (

SELECT otd\_kod, COUNT(\*) AS kol\_sotr

FROM sotrud

GROUP BY otd\_kod)

SELECT s1.fio,

so1.kol\_sotr,

s2.fio AS head\_fio

FROM sotrud s1 JOIN otd\_count so1

on s1.otd\_kod = so1.otd\_kod

JOIN sotd on s1.otd\_kod =sotd.otd\_kod

JOIN sotrud s2 on sotd.head\_kod = s2.sotrud\_kod;

Результат будет таким же.

***Пример 3*** Вывести список отделов в которых средний оклад превышает средний оклад в организации

WITH -- Создание базовых таблиц ……

-- Создание временных таблиц...,

**otd\_oklads** AS (

SELECT otd\_name,

AVG(OKLAD) as avg\_oklad

FROM sotrud s, sotd so

WHERE s.otd\_kod = so.otd\_kod

GROUP BY otd\_name),

**avg\_oklad** AS (

SELECT AVG(avg\_oklad) as avg\_org FROM otd\_oklads)

**SELECT** OTD\_NAME, cast(AVG\_OKLAD as numeric(10,2))

FROM otd\_oklads, avg\_oklad

WHERE avg\_oklad > avg\_org

ORDER BY otd\_name;

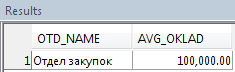
В предложении WITH создаются временные таблицы:

- временная таблица с именем **otd\_oklads** определяет средние оклады по отделам;

- временная таблица с именем **avg\_oklad** на основе данных первой временной таблицы **otd\_oklads** определяет средний оклад по организации.

Результирующий запрос выводит отделы и средние оклады в них, если этот оклад превышает средний оклад по организации. При этом условие соединения двух виртуальных таблиц (**otd\_oklads, avg\_oklad**) не задается, т.к. вторая таблица содержит одну строку. Функция CAST округляет результат до двух дробных разрядов.

Результат.



***Упражнение 5.*** Вывести список сотрудников (отдел, ФИО, должность, оклад), у которых оклад ниже среднего оклада по организации. Данные упорядочить по отделам, ФИО

***Упражнение 6*.** Вывести список должностей (должность, средний оклад), у которых средний оклад меньше среднего оклад по организации. Данные упорядочить в порядке убывания окладов

**Задание**

1. Выполните упражнения (обязательно включить в отчет)
2. Составьте похожие запросы для вашей БД (подзапросы включать в предложение WITH)