

Практическая работа № 15

Представления и табличные объекты

1. Цель работы

1. Изучить создание и удаление представлений.
2. Изучить табличные переменные.
3. Изучить временные таблицы.
4. Изучить производные таблицы.

2. Теоретическая часть

Представление – это виртуальная таблица, содержимое которой определяется запросом. Как и таблица, представление состоит из ряда именованных столбцов и строк данных.

Представления, как таблицы, могут иметь до 1024 столбцов.

Запрос для создания представления может обращаться не более чем к 256 таблицам.

Можно создавать представления на основе других представлений, при этом уровень вложенности не может быть больше 32-х.

Представление можно использовать как обычную таблицу.

Упрощенный синтаксис создания представления имеет следующий вид:

```
CREATE VIEW <название> <список столбцов>
```

```
AS <запрос SELECT>
```

Запрос SELECT, используемый в определении представления, не может включать предложение ORDER BY, если только в списке выбора инструкции SELECT нет также предложения TOP.

Для удаления представления используется команда DROP VIEW, его синтаксис:

```
DROP VIEW <название>
```

В Transact-SQL есть специальный тип данных для хранения результирующего набора для обработки в будущем. Его используют в основном для временного хранения набора строк, возвращаемых как результирующий набор функций с табличным значением. Функции и переменные могут быть объявлены как табличные переменные. Табличные переменные могут использоваться в функциях, хранимых процедурах и пакетах. Для объявления табличных переменных используется следующий синтаксис:

```
DECLARE <@название переменной> TABLE (<объявление столбцов>)
```

Табличная переменная ведет себя как локальная переменная, она имеет точно определенную область применения.

Табличная переменная может быть применена в любом месте, где используется таблица или табличное выражение в инструкциях SELECT, INSERT, UPDATE и DELETE. Но табличную переменную нельзя использовать в инструкции SELECT ... INTO ...

Табличные переменные автоматически очищаются в конце функции, хранимой процедуры или пакета, в котором они были определены.

Операция присвоения между табличными переменными не поддерживается.

В MS SQL Server для хранения промежуточных данных можно использовать временные таблицы. Они по поведению не отличаются от базовых таблиц. Создание, удаление и обращение к ним аналогично к базовым.

Первый символ в названии временной таблицы должен быть знак решетки #. Для локальных временных таблиц используется один знак #. Локальные временные таблицы доступны в течение текущей сессии и удаляются, когда пользователь отсоединяется от сервера.

Для глобальных временных таблиц используются два знака ##. Глобальные временные таблицы доступны всем открытым сессиям базы данных и удаляются, когда все пользователи, ссылающиеся на таблицы, отсоединяются от сервера.

Временные таблицы хранятся в системной базе данных TEMPDB.

Для принудительного удаления временных таблиц используется команда DROP TABLE.

В MS SQL Server можно создать временно именованный результирующий набор, называемый обобщенным табличным выражением. Он формируется при выполнении простого запроса.

За обобщенным табличным выражением должны следовать одиночные инструкции SELECT, INSERT, UPDATE или DELETE, ссылающиеся на некоторые или на все столбцы.

Обобщенные табличные выражения хранятся в оперативной памяти и существуют только во время первого выполнения запроса, который представляет эту таблицу.

3. Практическая часть

Пример 1: Создайте представление, содержащее список стран, население которых меньше 1 млн. чел., а площадь больше 100 тыс. кв. км, и используйте его:

```
CREATE VIEW Пример1
AS
SELECT
    Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент
FROM
    Страны
```

WHERE

Население < 1000000

AND

Площадь > 100000

SELECT

Название

,Столица

,Площадь

,Население

,Континент

FROM

Пример1

Пример 2: Создайте представление, содержащее список континентов, суммарную площадь и суммарное население стран, которые находятся на каждом континенте и используйте его:

CREATE VIEW Пример2

(

Континент

,Площадь

,Население

)

AS

SELECT

Континент

,SUM(Площадь)

,SUM(Население)

FROM

Страны

GROUP BY

Континент

SELECT

Континент

,Площадь

,Население

FROM

Пример2

Пример 3: Создайте представление, содержащее фамилии преподавателей, должность, каждого преподавателя, звание, степень, место работы, зарплату и используйте его:

```
CREATE VIEW Пример3
```

```
(
```

```
    Фамилия
```

```
    ,Должность
```

```
    ,Звание
```

```
    ,Степень
```

```
    ,Кафедра
```

```
    ,Зарплата
```

```
)
```

```
AS
```

```
SELECT
```

```
    Фамилия
```

```
    ,Должность
```

```
    ,Звание
```

```
    ,Степень
```

```
    ,Название
```

```
    ,Зарплата
```

```
FROM
```

```
    Сотрудник С
```

```
        INNER JOIN Преподаватель П ON С.Таб_номер = П.Таб_номер
```

```
        INNER JOIN Кафедра К ON С.Шифр = К.Шифр
```

```
SELECT
```

```
    Фамилия
```

```
    ,Должность
```

```
    ,Звание
```

```
    ,Степень
```

```
    ,Кафедра
```

```
    ,Зарплата
```

```
FROM
```

```
    Пример3
```

Пример 4: Создайте табличную переменную, содержащую три столбца («Номер недели», «Дата начала», «Дата конца»). Заполните ее для текущего года и используйте:

```
DECLARE @Пример4 TABLE
```

```
(
```

```
    [Номер недели] INT,
```

```
    [Дата начала] DATE,
```

```
    [Дата конца] DATE
```

```
)
```

```

DECLARE @T AS DATE, @N INT = 1
SET @T = CAST(YEAR(GETDATE()) AS CHAR(4)) + '0101'
WHILE DATEPART(WEEKDAY, @T) > 1
    SET @T = DATEADD(DAY, -1, @T)
PRINT DATEPART(WEEK, @T)
WHILE YEAR(@T) < YEAR(DATEADD(YEAR, 1, GETDATE()))
BEGIN
    INSERT
        @Пример4
    VALUES
        (@N, @T, DATEADD(DAY, 6, @T))

    SET @T = DATEADD(DAY, 7, @T)
    SET @N = @N + 1
END

SELECT
    [Номер недели]
    ,[Дата начала]
    ,[Дата конца]
FROM
    @Пример4

```

Пример 5: Создайте табличную переменную, содержащую список стран, площадь которых в 1000 раз меньше, чем средняя площадь стран в мире и используйте:

```

DECLARE @Пример5 TABLE
(
    Название VARCHAR(50),
    Столица VARCHAR(50),
    Площадь FLOAT,
    Население BIGINT,
    Континент VARCHAR(50)
)
INSERT INTO
    @Пример5
SELECT
    Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент

```

```

FROM
    Страны
WHERE
    Площадь * 1000 < (
        SELECT
            AVG(Площадь)
        FROM
            Страны
    )
SELECT
    Название
    ,Столица
    ,Площадь
    ,Население
    ,Континент

FROM
    @Пример5

```

Пример 6: Создайте локальную временную таблицу, имеющую три столбца («Название месяца», «Количество экзаменов», «Количество студентов»), заполните и используйте ее:

```

SELECT
    DATENAME(MONTH, Дата) AS [Название месяца]
    , COUNT(DISTINCT Код) AS [Количество экзаменов]
    , COUNT(DISTINCT Рег_номер) AS [Количество студентов]
INTO
    #Пример6
FROM
    Экзамен
GROUP BY
    DATENAME(MONTH, Дата)

SELECT * FROM #Пример6

```

Пример 7: Создайте глобальную временную таблицу, содержащую название стран и плотность их населения, заполните и используйте ее:

```

CREATE TABLE ##Пример7
(
    Название VARCHAR(50),
    Плотность FLOAT
)
INSERT INTO
    ##Пример7
    (Название, Плотность)

```

```

SELECT
    Название, ROUND(Население / Площадь, 0) AS Плотность
FROM
    Страны
SELECT * FROM ##Пример7
DROP TABLE #Пример6

```

Пример 8: С помощью обобщенных табличных выражений, напишите запрос для вывода списка сотрудников, чьи зарплаты меньше, чем средняя зарплата по кафедре, их зарплаты и название кафедры:

```

WITH СЗК AS
(
    SELECT
        К.Название AS Кафедра
        ,К.Шифр
        ,AVG(Зарплата) AS [Средняя зарплата по кафедре]
    FROM
        Сотрудник С
        INNER JOIN Кафедра К ON С.Шифр = К.Шифр
    GROUP BY
        К.Название, К.Шифр
)
SELECT
    С.Фамилия
    , С.Зарплата
    , З.Кафедра
    , З.[Средняя зарплата по кафедре]
FROM
    Сотрудник С
    INNER JOIN СЗК З ON С.Шифр = З.Шифр
WHERE
    С.Зарплата < З.[Средняя зарплата по кафедре]

```

4. Задание

1. Создайте представление, содержащее список африканских стран, население которых больше 10 млн. чел., а площадь больше 500 тыс. кв. км, и используйте его.
2. Создайте представление, содержащее список континентов, среднюю площадь стран, которые находятся на нем, среднюю плотность населения, и используйте его.
3. Создайте представление, содержащее фамилии преподавателей, их должность, звание, степень, место работы, количество их экзаменов, и используйте его.
4. Создайте табличную переменную, содержащую три столбца («Номер месяца», «Название месяца», «Количество дней»), заполните ее для текущего года, и используйте ее.
5. Создайте табличную переменную, содержащую список стран, площадь которых в 100 раз меньше, чем средняя площадь стран на континенте, где они находятся, и используйте ее.
6. Создайте локальную временную таблицу, имеющую три столбца («Номер недели», «Количество экзаменов», «Количество студентов»), заполните и используйте ее.
7. Создайте глобальную временную таблицу, содержащую название континентов, наибольшую и наименьшую площадь стран на них, заполните и используйте ее.
8. С помощью обобщенных табличных выражений напишите запрос для вывода списка сотрудников, чьи зарплаты меньше, чем средняя зарплата по факультету, их зарплаты и название факультета.
9. Напишите команды для удаления всех созданных вами представлений.

