

Лабораторная работа № 19

Динамический SQL, XML и JSON

1. Цель работы

1. Изучить использование динамического SQL.
2. Изучить вывод данных в формате XML.
3. Изучить вывод данных в формате JSON.

2. Теоретическая часть

В интерактивных приложениях часто состав SQL-запросов заранее не известен и может изменяться в зависимости от условий.

Для решения таких ситуаций используется динамический SQL. Запросы формируются в виде текстовых строк непосредственно во время работы программы. Выполнение динамического SQL медленнее, чем статического.

Для выполнения динамического SQL используется команда EXEC | EXECUTE. Упрощенный синтаксис имеет следующий вид:

EXEC | EXECUTE (<строка динамического SQL>)

Результат выполнения запроса можно получать в формате XML. Для этого в запросе необходимо указать предложение FOR XML.

В предложении FOR XML можно указать один из следующих режимов:

RAW – в этом режиме создается одиночный элемент <row> для каждой строки набора строк, возвращенного инструкцией SELECT. XML-иерархию можно создать с помощью написания вложенных запросов FOR XML.

AUTO – в этом режиме вложенность XML создается эвристически, в зависимости от метода определения инструкции SELECT.

EXPLICIT – этот режим предоставляет больше возможностей по управлению формой XML-структуры. В XML-структуре могут быть использованы смешанные атрибуты и элементы. Однако написание запросов в режиме EXPLICIT может быть очень обременительным.

PATH – этот режим совместно с вложенным запросом FOR XML обеспечивает гибкость режима EXPLICIT более простым образом.

Эти режимы на самом деле работают только при выполнении запросов, для которых они установлены. Они не влияют на результаты любых последующих запросов.

Результат выполнения запроса можно получать в формате JSON. Для этого в запросе необходимо указать предложение FOR JSON. Предложение FOR JSON упрощает клиентские приложения, делегируя форматирование выходных данных JSON из приложения в SQL Server.

В предложении FOR XML можно указать один из следующих режимов:

PATH – этот режим используется для того чтобы сохранить полный контроль над форматом выходных данных JSON. Можно создавать объекты-оболочки и вкладывать сложные свойства друг в друга.

AUTO – этот режим используется для того чтобы отформатировать выходные данные автоматически на основе структуры инструкции SELECT.

3. Практическая часть

Пример 1: Создайте и запустите динамический SQL-запрос, выбирающий все данные из заданной таблицы:

```
DECLARE @DSQL VARCHAR(100)
DECLARE @TN VARCHAR(50)

SET @TN = 'Ученики'
SET @DSQL = 'SELECT * FROM ' + @TN

EXECUTE (@DSQL)
```

Пример 2: Создайте и запустите динамический SQL-запрос, выбирающий всех учеников из таблицы «Ученики», вторая буква фамилии которых расположена в заданном диапазоне:

```
DECLARE @L1 CHAR(1)
DECLARE @L2 CHAR(1)

SET @L1 = 'а'
SET @L2 = 'д'

EXECUTE ('SELECT * FROM Ученики WHERE Фамилия LIKE "_" + @L1 + '-' + @L2
+ ']'%)
```

Пример 3: Создайте временную таблицу #Temp и добавить в нее название пяти таблиц из базы данных «Учебная». Создайте курсор, который, построчно читая таблицу #Temp, выбирает из каждой таблицы все данные:

```
CREATE TABLE #TEMP
(
    TABLENAME VARCHAR(50)
)

INSERT INTO
    #TEMP
```

VALUES

('Дисциплина'),
('Заявка'),
('Инженер'),
('Кафедра'),
('Сотрудник')

DECLARE TC CURSOR

FOR SELECT TABLENAME FROM #TEMP

OPEN TC

DECLARE @TN VARCHAR(50)

FETCH FROM TC INTO @TN

WHILE @@FETCH_STATUS = 0

BEGIN

EXECUTE ('SELECT * FROM ' + @TN)

FETCH FROM TC INTO @TN

END

CLOSE TC

DEALLOCATE TC

DROP TABLE #TEMP

Пример 4: Выберите данные из таблицы «Ученики» и экспортируйте в RAW формате XML:

SELECT

ID
,Фамилия
,Предмет
,Школа
,Баллы

FROM

Ученики

FOR XML RAW

Пример 5: Выберите данные из таблицы «Страны» и экспортируйте в PATH формате XML:

SELECT

Название
,Столица

```

        ,Площадь
        ,Население
        ,Континент
FROM
        Страны
FOR XML PATH

```

Пример 6: Выберите данные из таблицы «Ученики» и экспортируйте в PATH формате JSON:

```

SELECT
        ID
        ,Фамилия
        ,Предмет
        ,Школа
        ,Баллы
FROM
        Ученики
FOR JSON PATH

```

4. Задание

1. Создайте и запустите динамический SQL-запрос, выбирающий первые N строк из заданной таблицы.
2. Создайте и запустите динамический SQL-запрос, выбирающий все страны из таблицы «Страны», последняя буква названия которых расположена в заданном диапазоне.
3. Создайте временную таблицу #Temp и добавьте к ней название столбцов таблицы «Страны». Создайте курсор, который, построчно читая таблицу #Temp, выбирает каждый раз соответствующий столбец из таблицы «Страны».
4. Создайте процедуру, которая принимает как параметр список столбцов, название таблицы и выбирает заданные столбцы из заданной таблицы.
5. Создайте процедуру, принимающую как параметр список столбцов, название таблицы, название проверяемого столбца, знак сравнения, значение проверки и выбирающую заданные столбцы из заданной таблицы в заданных условиях.
6. Выберите список европейских стран из таблицы «Страны» и экспортируйте в RAW формате XML.
7. Выберите список стран с населением больше 100 млн. чел. из таблицы «Страны» и экспортируйте в PATH формате XML.
8. Выберите список учеников из школы «Лицей» из таблицы «Ученики» и экспортируйте в PATH формате JSON