Динамический код

Динамический SQL - это метод программирования, который позволяет динамически создавать выражения SQL во время выполнения. Это позволяет создавать более универсальные и гибкие запросы SQL, потому что полный текст выражения может быть неизвестен при компиляции.

- Команда EXECUTE в T-SQL
- Хранимая процедура sp_executesql в T-SQL

Пример: оператор Pivot используется для напи перекрестных запросов или кросс табличных вь

Microsoft SQL Server
Динамический код

```
SELECT <non-pivoted column>,
    [first pivoted column] AS <column name>,
    [second pivoted column] AS <column name>,
    ...
    [last pivoted column] AS <column name>
FROM
    (<SELECT query that produces the data>)
    AS <alias for the source query>
PIVOT
(
    <aggregation function>(<column being aggregated>)
FOR
[<column that contains the values that will become column headers>]
    IN ( [first pivoted column], [second pivoted column],
    ... [last pivoted column])
) AS <alias for the pivot table>
<optional ORDER BY clause>;
```

```
CREATE TABLE Sales (EmpId INT, Yr INT, Sales MONEY)
INSERT Sales VALUES(1, 2005, 12000)
INSERT Sales VALUES(1, 2006, 18000)
INSERT Sales VALUES(1, 2007, 25000)
INSERT Sales VALUES(2, 2005, 15000)
INSERT Sales VALUES(2, 2006, 6000)
INSERT Sales VALUES(3, 2006, 20000)
INSERT Sales VALUES(3, 2007, 24000)
```

```
SELECT EmpId, [2005], [2006], [2007]
FROM (SELECT EmpId, Yr, Sales FROM Sales) AS s
PIVOT (SUM(Sales) FOR Yr IN ([2005], [2006], [2007])) AS
```

EmpId	2005	2006	2007
1	12000.00	18000.00	25000.00
2	15000.00	6000.00	NULL
3	NULL	20000.00	24000.00

Команда EXECUTE в T-SQL

Microsoft SQL Server
Динамический код

```
Execute a character string
{ EXEC | EXECUTE }
    ( { @string_variable | [ N ]'tsql_string' } [ + ...n ] )
    [ AS { LOGIN | USER } = ' name ' ]
[;]
```

```
--Объявляем переменные
  DECLARE @SQL_QUERY VARCHAR(200),
  @Var1 INT;

  --Присваиваем значение переменным
  SET @Var1 = 1;

  --Формируем SQL инструкцию
  SET @SQL_QUERY = 'SELECT * FROM Products WHERE ProductID = ' + CAST(@Var1 AS
VARCHAR(10));

  --Смотрим на итоговую строку
  SELECT @SQL_QUERY AS [TEXT QUERY]

  --Выполняем текстовую строку как SQL инструкцию
```

Команда EXECUTE в T-SQL

Microsoft SQL Server
Динамический код

```
Execute a character string
{ EXEC | EXECUTE }
    ( { @string_variable | [ N ]'tsql_string' } [ + ...n ] )
    [ AS { LOGIN | USER } = ' name ' ]
[;]
```

Использование динамического кода с использованием команды EXEC – это не безопасно! Дело в том, что для того чтобы сформировать динамическую SQL инструкцию, необходимо использовать переменные для динамически изменяющихся значений. Так вот, если эти значения будут приходить от клиентского приложения, т.е. от пользователя, злоумышленники могут передать и, соответственно, внедрить в нашу инструкцию вредоносный код в виде текста, а мы его просто исполним в БД, думая, что нам передали обычные параметры. Поэтому все такие значения следует очень хорошо проверять, перед тем как подставлять в инструкцию.

Хранимая процедура sp_executesql в T-SQL

Microsoft SQL Server
Динамический код

Системная хранимая процедура Microsoft SQL Server, которая выполняет SQL инструкции. Эти инструкции могут содержать параметры, тем самым делая их динамическими.

```
sp_executesql [ @stmt = ] statement
[
  { , [ @params = ] N'@parameter_name data_type [ OUT | OUTPUT ][ ,...n ]' }
     { , [ @param1 = ] 'value1' [ ,...n ] }
]
```

Все параметры процедуры sp_executesql необходимо передавать в формате **Unicode** (*тип данных строк должен быть NVARCHAR*).

Хранимая процедура sp_executesql в T-SQL

Microsoft SQL Server
Динамический код

```
--Объявляем переменные
   DECLARE @SQL_QUERY NVARCHAR(200);

   --Формируем SQL инструкцию
   SELECT @SQL_QUERY = N'SELECT * FROM Products WHERE ProductID
= @Var1;';

   --Смотрим на итоговую строку
   SELECT @SQL_QUERY AS [TEXT QUERY]

   --Выполняем текстовую строку как SQL инструкцию
   EXEC sp_executesql @SQL_QUERY,--Текст SQL инструкции
                  N'@Var1 AS INT', --Объявление переменной @Var1
                  @Var1 = 1 --Передаем значение для переменной @Var1
```

Динамический PIVOT в T-SQL — универсальная процедура формирования запроса

Microsoft SQL Server
Динамический код

Проблема - синтаксис PIVOT требует перечисления вручную имен столбцов, которые мы хотим «*развернуть*».
Данная процедура универсальная, необходимо при запуске только передать соответствующие параметры, остальное процедура сделает сама.
Параметры:

- @TableSRC — имя таблицы или представления;
- @ColumnName — столбец, содержащий значения, которые станут именами столбцов;
- @Field — столбец, над которым проводить агрегацию данных, т.е. к которому применять статистическую функцию;
- @FieldRows — столбец, по которому необходимо сгруппировать данные по строкам;
- @FunctionType — агрегатная функция (SUM, COUNT, MAX, MIN, AVG), по умолчанию SUM, т.е. параметр указывать необязательно;
- @Condition – условие, тоже необязательный параметр.

https://info-comp.ru/obucheniest/631-dynamic-pivot-in-t-sql.html

Динамический PIVOT в T-SQL — универсальная процедура формирования запроса

Microsoft SQL Server
Динамический код

```
-- Создаем универсальную процедуру для динамического PIVOT
   CREATE PROCEDURE SP_Dynamic_Pivot
   (
        @TableSRC NVARCHAR(100),   --Таблица источник (Представление)
        @ColumnName NVARCHAR(100), --Столбец, содержащий значения, которые станут имена
столбцов
        @Field NVARCHAR(100),      --Столбец, над которым проводить агрегацию
        @FieldRows NVARCHAR(100),  --Столбец (столбцы) для группировки по строкам (Column1,
Column2)
        @FunctionType NVARCHAR(20) = 'SUM',--Агрегатная функция (SUM, COUNT, MAX, MIN, AVG)
по умолчанию SUM
        @Condition NVARCHAR(200) = '' --Условие (WHERE и т.д.). По умолчанию без условия
   )
   AS
   BEGIN
      --Отключаем вывод количества строк
        SET NOCOUNT ON;
       --Переменная для хранения строки запроса
        DECLARE @Query NVARCHAR(MAX);
       --Переменная для хранения имен столбцов
        DECLARE @ColumnNames NVARCHAR(MAX);
```

https://info-comp.ru/obucheniest/631-dynamic-pivot-in-t-sql.html

Динамический PIVOT в T-SQL — универсальная процедура формирования запроса


```
--Переменная для хранения заголовков результирующего набора данных
	DECLARE @ColumnNamesHeader NVARCHAR(MAX);

	--Обработчик ошибок
	BEGIN TRY
		--Таблица для хранения уникальных значений,
		--которые будут использоваться в качестве столбцов
		CREATE TABLE #ColumnNames(ColumnName NVARCHAR(100) NOT NULL PRIMARY KEY);

		--Формируем строку запроса для получения уникальных значений для имен столбцо
		SET @Query = N'INSERT INTO #ColumnNames (ColumnName)
					SELECT DISTINCT COALESCE(' + @ColumnName + ', ''Пусто'')
					FROM ' + @TableSRC + ' ' + @Condition + ';'

		--Выполняем строку запроса
		EXEC (@Query);

		--Формируем строку с именами столбцов
		SELECT @ColumnNames = ISNULL(@ColumnNames + ', ','') + QUOTENAME(ColumnName
		FROM #ColumnNames;
```

Динамический PIVOT в T-SQL — универсальная процедура формирования запроса


```
 --Формируем строку для заголовка динамического перекрестного запроса (PIVOT)
            SELECT @ColumnNamesHeader = ISNULL(@ColumnNamesHeader + ', ','')
                                                + 'COALESCE('
                                                + QUOTENAME(ColumnName)
                                                + ', 0) AS '
                                                + QUOTENAME(ColumnName)
            FROM #ColumnNames;

            --Формируем строку с запросом PIVOT
            SET @Query = N'SELECT ' + @FieldRows + ' , ' + @ColumnNamesHeader + '
                                FROM (SELECT ' + @FieldRows + ', ' + @ColumnName + ', ' +
@Field
                                                + ' FROM ' + @TableSRC  + ' ' + @Condition + ') AS SRC
                                PIVOT ( ' + @FunctionType + '(' + @Field + ')' +' FOR ' +
                                                @ColumnName + ' IN (' + @ColumnNames + ')) AS
PVT
                                ORDER BY ' + @FieldRows + ';'
```

https://info-comp.ru/obucheniest/631-dynamic-pivot-in-t-sql.html

Динамический PIVOT в T-SQL — универсальная процедура формирования запроса

Microsoft SQL Server
Динамический код

```
--Удаляем временную таблицу
        DROP TABLE #ColumnNames;

        --Выполняем строку запроса с PIVOT
        EXEC (@Query);

        --Включаем обратно вывод количества строк
        SET NOCOUNT OFF;

    END TRY
    BEGIN CATCH
        --В случае ошибки, возвращаем номер и описание этой ошибки
        SELECT ERROR_NUMBER() AS [Номер ошибки],
               ERROR_MESSAGE() AS [Описание ошибки]
    END CATCH
END
```

https://info-comp.ru/obucheniest/631-dynamic-pivot-in-t-sql.html

Динамический PIVOT в T-SQL — универсальная процедура формирования запроса

Microsoft SQL Server
Динамический код

```
--Пример 1. Получаем суммы по годам с группировкой по категории

   EXEC SP_Dynamic_Pivot @TableSRC = 'TestTable',  --Таблица источник
(Представление)
                         @ColumnName = 'YearSales',--Столбец, содержащий
значения для столбцов в PIVOT
                         @Field = 'Summa',        --Столбец, над которым проводить
агрегацию
                         @FieldRows = 'CategoryName',--Столбец для группировки по
строкам
ер 3. Получаем суммы по годам с группировкой по товару в конкретной категории
SP_Dynamic_Pivot @TableSRC = 'TestTable',  --Таблица источник (Представление)
    умолчанию SUM
                @ColumnName = 'YearSales',--Столбец, содержащий значения для столбцов в PIV
                @Field = 'Summa',        --Столбец, над которым проводить агрегацию
                @FieldRows = 'ProductName',--Столбец для группировки по строкам
                @FunctionType = 'SUM',    --Агрегатная функция, по умолчанию SUM
                @Condition = 'WHERE CategoryName = ''Комплектующие компьютера'''


https://info-comp.ru/obucheniest/631-dynamic-pivot-in-t-
sql.html
```

QUOTENAME()

Добавляет разделители к входной строке, чтобы сделать эту строку допустимым идентификатором с разделителями SQL Server

```
QUOTENAME ( input_string [ , quote_character ] )
```

Функция QUOTENAME()принимает два аргумента:

- input_string— это a SYSNAME, максимальная длина которого равна 128. Если длина input_stringбольше 128 символов, функция вернет NULL.
- quote_character— это символ, который используется в качестве разделителя.

Ниже приведены допустимые символы кавычек:

- Одинарная кавычка (')
- Левая или правая скобка ([])
- Двойная кавычка (”)
- Левая или правая скобка (())
- Знак больше или меньше (><)
- Левая или правая скобка ({})
- Обратный апостроф (`).

Если вы используете недопустимый символ, функция вернет NULL. По умолчанию quote_character используются скобки, если вы их пропустите.

QUOTENAME()

Добавляет разделители к входной строке, чтобы сделать эту строку допустимым идентификатором с разделителями SQL Server


```
DECLARE @tablename VARCHAR(128) = 'customer details';
DECLARE @sql NVARCHAR(100) = 'SELECT * FROM ' + @tablename;
EXECUTE (@sql);
```

Он возвращает следующую ошибку:

```
Invalid object name 'customer'.
```

```
DECLARE @tablename VARCHAR(128) = 'customer details';
DECLARE @sql NVARCHAR(100) = 'SELECT * FROM ' + QUOTENAME(@tablename);
EXECUTE (@sql);
```