Надо результат выборки получить в виде текста. Например, если результат выборки:

1. Миша  
2. Коля  
3. Дима  
4. Антон  
то результирующая строка должна быть такого вида: 'Миша Коля Дима Антон'.

SELECT @str=@str+name+’ ‘ FROM names

**EXECUTE (Transact-SQL)**

Выполняет командную строку — строку символов, в которой содержится пакет Transact-SQL.

Пример кода с EXEC/EXECUTE:

DECLARE @sql varchar(1000)

DECLARE @columnList varchar(75)

DECLARE @city varchar(75)

SET @columnList = 'CustomerID, ContactName, City'

SET @city = 'London'

SELECT @sql = ' SELECT CustomerID, ContactName, City ' +

FROM dbo.customers WHERE 1 = 1 '

SELECT @sql = @sql + ' AND City LIKE ''' + @city + ''''

EXEC (@sql)

**Использование операторов PIVOT и UNPIVOT**

Реляционные операторы PIVOT и UNPIVOT можно использовать для изменения возвращающего табличное значение выражения в другой таблице. Оператор PIVOT разворачивает возвращающее табличное значение выражение, преобразуя уникальные значения одного столбца выражения в несколько выходных столбцов, а также, в случае необходимости, объединяет оставшиеся повторяющиеся значения столбца и отображает их в выходных данных. Оператор UNPIVOT производит действия, обратные PIVOT, преобразуя столбцы возвращающего табличное значение выражения в значения столбца.

## Базовый пример PIVOT

В следующем примере кода создается таблица, включающая два столбца и четыре строки.

USE AdventureWorks2008R2 ;

GO

SELECT DaysToManufacture, AVG(StandardCost) AS AverageCost

FROM Production.Product

GROUP BY DaysToManufacture;

Ниже приводится результирующий набор.

|  |  |
| --- | --- |
| DaysToManufacture | AverageCost |
| 0 | 5.0885 |
| 1 | 223.88 |
| 2 | 359.1082 |
| 4 | 949.4105 |

Для значения DaysToManufacture, равного трем, продукты не определены.

Следующий код отображает тот же самый результат, сведенный так, что значения DaysToManufacture становятся заголовками. Для значения трех [3] дней приводится столбец, даже если результат равен NULL.

-- Pivot table with one row and five columns

SELECT 'AverageCost' AS Cost\_Sorted\_By\_Production\_Days,

[0], [1], [2], [3], [4]

FROM

(SELECT DaysToManufacture, StandardCost

FROM Production.Product) AS SourceTable

PIVOT

(

AVG(StandardCost)

FOR DaysToManufacture IN ([0], [1], [2], [3], [4])

) AS PivotTable;

Ниже приводится результирующий набор.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cost\_Sorted\_By\_Production\_Days | 0 | 1 | 2 | 3 | 4 |
| AverageCost | 5.0885 | 223.88 | 359.1082 | NULL | 949.4105 |

## Сложный пример PIVOT

Обычно оператор PIVOT может быть полезен при создании отчетов с перекрестными ссылками для сведения данных. Например, пусть необходимо обратиться к таблице PurchaseOrderHeader образца базы данных AdventureWorks2008R2 для определения количества заказов на покупку, размещенных некоторым сотрудником. Требуемые данные, отсортированные по поставщикам, можно извлечь при выполнении следующего запроса.

USE AdventureWorks2008R2;

GO

SELECT VendorID, [250] AS Emp1, [251] AS Emp2, [256] AS Emp3, [257] AS Emp4, [260] AS Emp5

FROM

(SELECT PurchaseOrderID, EmployeeID, VendorID

FROM Purchasing.PurchaseOrderHeader) p

PIVOT

(

COUNT (PurchaseOrderID)

FOR EmployeeID IN

( [250], [251], [256], [257], [260] )

) AS pvt

ORDER BY pvt.VendorID;

Здесь приводится частичный результирующий набор.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VendorID | Emp1 | Emp2 | Emp3 | Emp4 | Emp5 |
| 1492 | 2 | 5 | 4 | 4 | 4 |
| 1494 | 2 | 5 | 4 | 5 | 4 |
| 1496 | 2 | 4 | 4 | 5 | 5 |
| 1498 | 2 | 5 | 4 | 4 | 4 |
| 1500 | 3 | 4 | 4 | 5 | 4 |