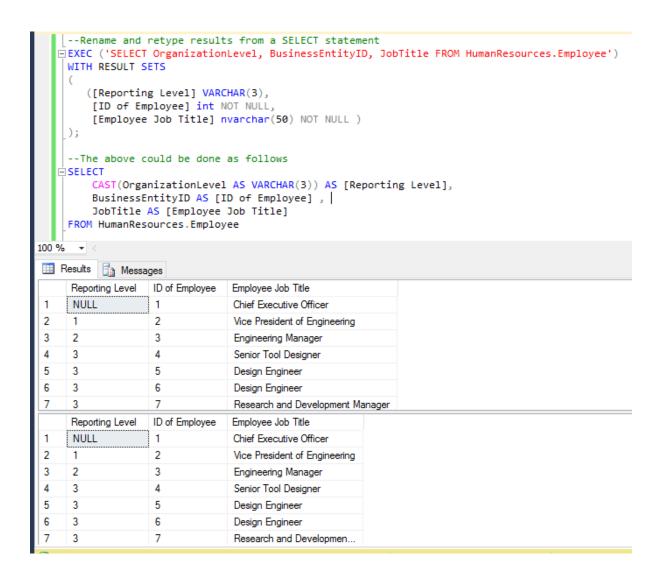
SQL Server – EXECUTE Statement with RESULT SET Clause

SQL Server 2012 introduced a RESULT SET clause to the EXECUTE statement.

It can be used to specify alternate data types and column names for result sets returned by an EXECUTED statement or Stored Procedure.

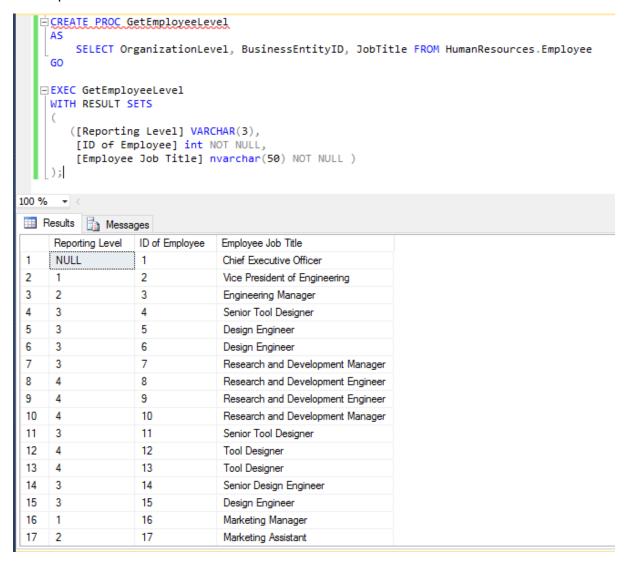
The following example shows its use with an ad-hoc query example.

- The first query uses the RESULT AS clause to define the names datatypes for three returned columns.
- The second query uses a CAST and column aliasing to achieve the same result.

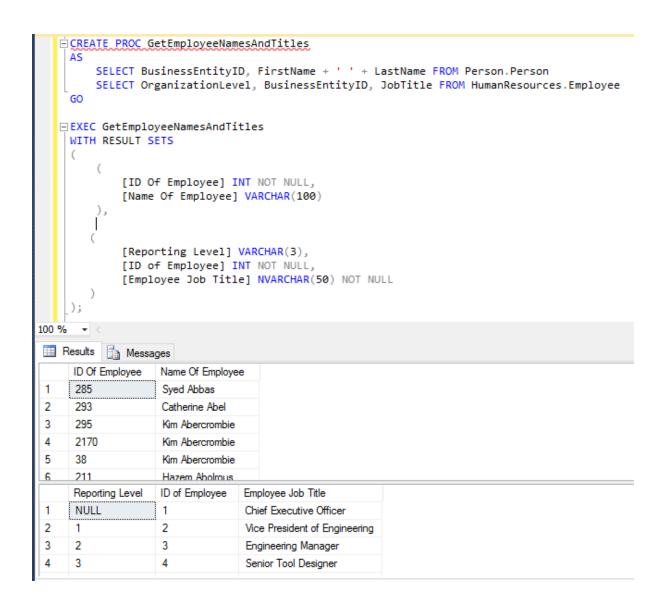


The RESULT SET clause is more useful when working with Stored Procedures that provide no opportunity to change the column names defined within the stored procedure or the data types derived in the underlying Transact SQL code within the procedure.

The following example shows a stored procedure definition and then an EXECUTE statement that changes the column names and data types of the results set returned by the stored procedure.



The following example demonstrates that multiple results sets can be handled where a stored procedure returns more than one result set.



The examples are based on the AdvetureWorks2014 database. Here is the code for the examples in this document:

```
--Using the RESULT SETS clause of the EXECUTE Statement
--Rename and retype results from a SELECT statement
EXEC ('SELECT OrganizationLevel, BusinessEntityID, JobTitle FROM HumanResources.Employee')
WITH RESULT SETS
   ([Reporting Level] VARCHAR(3),
    [ID of Employee] int NOT NULL,
    [Employee Job Title] nvarchar(50) NOT NULL )
);
-- The above could be done as follows
SELECT
       CAST(OrganizationLevel AS VARCHAR(3)) AS [Reporting Level],
       BusinessEntityID AS [ID of Employee] ,
       JobTitle AS [Employee Job Title]
FROM HumanResources. Employee
--With a Stored Procedure
CREATE PROC GetEmployeeLevel
       SELECT OrganizationLevel, BusinessEntityID, JobTitle FROM HumanResources.Employee
GO
EXEC GetEmployeeLevel
WITH RESULT SETS
   ([Reporting Level] VARCHAR(3),
    [ID of Employee] int NOT NULL,
    [Employee Job Title] nvarchar(50) NOT NULL )
--Stored Procedure that returns two results sets
CREATE PROC GetEmployeeNamesAndTitles
       SELECT BusinessEntityID, FirstName + ' ' + LastName FROM Person.Person
       SELECT OrganizationLevel, BusinessEntityID, JobTitle FROM HumanResources.Employee
G0
EXEC GetEmployeeNamesAndTitles
WITH RESULT SETS
(
               [ID Of Employee] INT NOT NULL,
               [Name Of Employee] VARCHAR(100)
       ),
   (
               [Reporting Level] VARCHAR(3),
               [ID of Employee] INT NOT NULL,
               [Employee Job Title] NVARCHAR(50) NOT NULL
);
```

If you are itching to learn more why not book on to our SQL Server Database Querying training courses? This link will take you to the course outlines:

http://ptr.co.uk/databases-business-intelligence-courses.