

PIVOT and UNPIVOT in Sql Server

PIVOT and UNPIVOT relational operators - change a table-valued expression into another table.

PIVOT rotates a table-valued expression by turning the unique values from one column in the expression into multiple columns in the output, and performs aggregations where they are required on any remaining column values that are wanted in the final output. When aggregate functions are used, the presence of any null values in the value column are not considered when computing an aggregation. PIVOT- an easy mechanism in Sql Server to transform rows into columns.

UNPIVOT performs the opposite operation to PIVOT by rotating columns of a table-valued expression into column values. UNPIVOT performs almost the reverse operation of PIVOT, by rotating columns into rows.

Notice that UNPIVOT is not the exact reverse of PIVOT. PIVOT performs an aggregation and, therefore, merges possible multiple rows into a single row in the output. UNPIVOT does not reproduce the original table-valued expression result because rows have been merged. Besides, null values in the input of UNPIVOT disappear in the output, whereas there may have been original null values in the input before the PIVOT operation.

PIVOT

Consider a Temporary Table #CourseSales with sample records

```
use DBMS_Lab3
```

```
go
```

```
--Create Temporary Table #CourseSales
```

```
CREATE TABLE #CourseSales (Course  
VARCHAR(50),Year INT,Earning MONEY)
```

```
GO
```

```
--Populate Sample records
```

```
INSERT INTO #CourseSales
```

```
VALUES( '.NET',2012,10000)
```

```
INSERT INTO #CourseSales
```

```
VALUES( 'Java',2012,20000)
```

```
INSERT INTO #CourseSales
```

```
VALUES( '.NET',2012,5000)
```

```
INSERT INTO #CourseSales
```

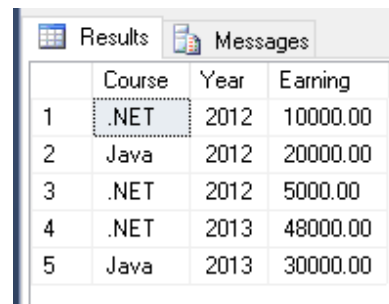
```
VALUES( '.NET',2013,48000)
```

```
INSERT INTO #CourseSales
```

```
VALUES( 'Java',2013,30000)
```

```
GO
```

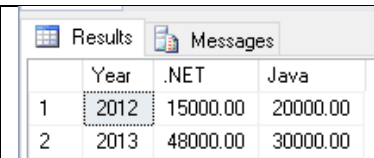
```
select * from #CourseSales
```



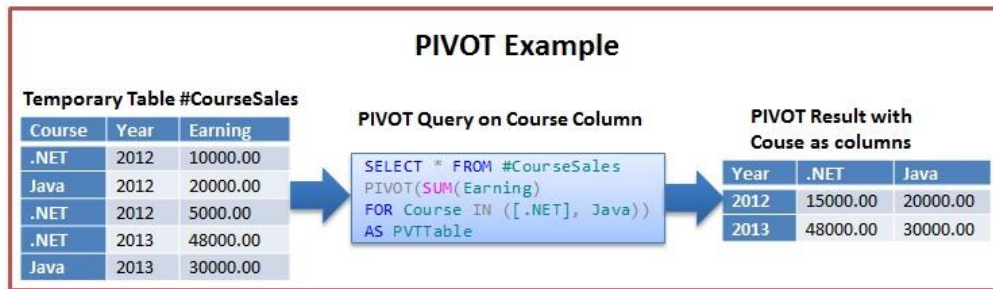
	Course	Year	Earning
1	.NET	2012	10000.00
2	Java	2012	20000.00
3	.NET	2012	5000.00
4	.NET	2013	48000.00
5	Java	2013	30000.00

Example 1: the #CourseSales Table data is Pivoted so that the Course becomes the column headings.

```
-- example 1 - the Course columns distinct  
values are transformed as Columns in the result  
set  
SELECT *  
FROM #CourseSales  
PIVOT(SUM(Earning)  
FOR Course IN ([.NET], Java)) AS PVTTable
```



	Year	.NET	Java
1	2012	15000.00	20000.00
2	2013	48000.00	30000.00

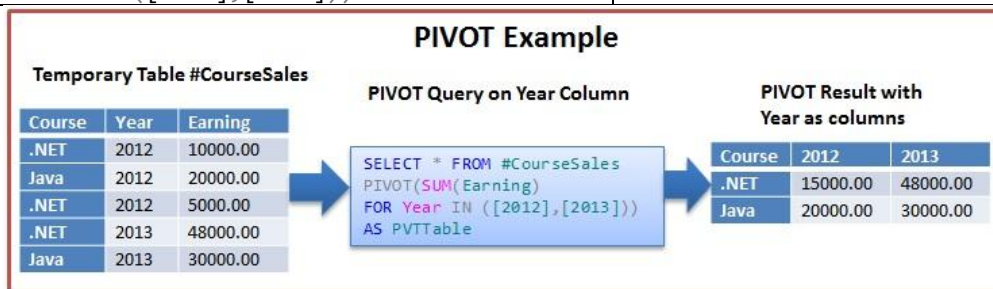


Example 2: the #CourseSales Table data is Pivoted so that the Year becomes the column headings.

```
-- example 2 - the Year columns distinct values are
transformed as Columns in the result set

SELECT *
FROM #CourseSales
PIVOT(SUM(Earning)
FOR Year IN ([2012],[2013])) AS PVTTable
```

	Course	2012	2013
1	.NET	15000.00	48000.00
2	Java	20000.00	30000.00



Example 3: simulate the PIVOT and UNPIVOT operation on the #CourseSales Temporary

```
--PIVOT the #CourseSales table data on the
Course column
SELECT *
INTO #CourseSalesPivotResult
FROM #CourseSales
PIVOT(SUM(Earning)
FOR Course IN ([.NET], Java)) AS
PVTTable
GO

-----
(2 row(s) affected)
```

```
--UNPIVOT the #CourseSalesPivotResult table
data on the Course column
SELECT Course, Year, Earning
FROM #CourseSalesPivotResult1
UNPIVOT(Earning
FOR Course IN ([.NET], Java)) AS
UNPVTTable

-----
```

	Course	Year	Earning
1	.NET	2012	15000.00
2	Java	2012	20000.00
3	.NET	2013	48000.00
4	Java	2013	30000.00

References:

<https://www.codeproject.com/Tips/500811/Simple-Way-To-Use-Pivot-In-SQL-Query>
<http://sqlhints.com/2014/03/10/pivot-and-unpivot-in-sql-server/>