

6b) Kontingenčné - Pivot tabuľky

3a) Klauzula PIVOT

3b) Manualny PIVOT

3c) Dynamický - poloautomatický Pivot

PT je **dvojrozmerná agregáčna** (frekvenčná, sumárna, priemerná, ...) tabuľka, zovšeobecňujúca GROUP BY podľa dvoch [aj viac] stĺpcov, atribútov. PT je dôležitý nástroj na získavanie informácií v DB a aplikáciach na dolovanie dát.

T-SQL podporuje PT pomocou operátorov PIVOT a UNPIVOT (opačná oprácia – vráti štíhlejší výsledok).

Pivot tabuľky môžeme vytvoriť

a) manuálne PIVOT ... IN ...

b) Automatický PIVOT ... IN pomocou reťazcových príkazov QUOTENAME, XML PATH, STUFF

c) bez operátora Pivot pomocou (pozri ZS) SUM(CASE WHEN ...

d) Pivot tabuľky a Excel, R

a) Pivot tabuľky manuálne PIVOT ... IN ...

0) Tabuľka ##T1

1) Pivot:

(Na cvičení:)

2a) Dodajme Vcelku:

2b) Dodajme Vcelku s odstránením NULL Vcelku:

2c) Dodajme Vcelku s úplným odstránením NULL (+ názvy stĺpcov):

3a) Bez pivot:

3b) Bez pivot: SUM(CASE ...

Príklad. Vytvorme tabuľku ##T1 na pivotovanie.

```
USE tempdb;
```

```
if OBJECT_ID('##T1', 'U') IS NOT NULL DROP TABLE ##T1
```

```
CREATE TABLE ##T1(Oddel Char, Rok SMALLINT, Kvartal TINYINT, Obrat
```

```
DECIMAL(2,1))
```

```
GO
```

```
INSERT INTO ##T1 (Oddel, Rok, Kvartal, Obrat)
```

```
SELECT 'A', 2006, 1, 0.6 UNION ALL
```

```
SELECT 'B', 2006, 1, 0.7 UNION ALL
```

```
SELECT 'A', 2006, 3, 0.9 UNION ALL
```

```
SELECT 'B', 2006, 3, 0.7 UNION ALL
```

```
SELECT 'A', 2006, 4, 0.8 UNION ALL
```

```
SELECT 'B', 2006, 4, 0.8 UNION ALL
```

```
SELECT 'A', 2007, 1, 0.7 UNION ALL
```

```
SELECT 'A', 2007, 2, 0.9 UNION ALL
```

```
SELECT 'A', 2007, 2, 0.9 UNION ALL
```

```
SELECT 'A', 2007, 3, 0.8 UNION ALL
```

```
SELECT 'A', 2007, 3, 0.6 UNION ALL
```

```
SELECT 'A', 2007, 4, 0.9 UNION ALL
```

```
SELECT 'B', 2007, 4, 0.7;
```

Syntax-MS: npc ⇔ non-pivoted column, pc ⇔ pivoted column, cn ⇔ column name.

```
SELECT <npc>, [first pc] AS <cn>, [second pc] AS <cn>, ..., [last pc] AS <cn>
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 pc], [second pc], ..., [last pc] )
  ) AS <alias for the pivot table>
<optional ORDER BY clause>;
```

Sumárny obrat v jednotlivých kvartáloch v každom roku:

	Rok	kvart_1	2	3	4
1	2006	1.3	NULL	1.6	1.6
2	2007	0.7	1.8	1.4	1.6

---- a) Pivot manuálne - vo výsledku riadok, stĺpec:

```
SELECT Rok, [1] kvart_1, [2],[3],[4] FROM
  (SELECT Rok, Kvartal, Obrat FROM ##T1) pom
  ---- V hre budu/su tie stlpce, ktore vyssie vo vnorenom dopyte selektujeme.
  PIVOT (SUM(Obrat) FOR Kvartal IN ([1],[2],[3],[4])) piv
  ---- Z hodnot stlpca Kvartal tvorime maximalne 4 nove stlpce.
  -- ORDER BY Rok DESC
```

Sumárny obrat jednotlivých oddelení v každom roku:

```
SELECT Rok, [A],[B]
FROM (SELECT Rok, Oddel, Obrat FROM ##T1) pom
PIVOT (SUM(Obrat) FOR Oddel IN ([A],[B])) piv
ORDER BY Rok
```

	Rok	A	B
1	2006	2.3	2.2
2	2007	4.8	0.7

Dodajme stĺpec TOTAL (plus odstránenie prípadných NULL hodnôt)

```
SELECT Rok, ISNULL([A],0) A, ISNULL([B],0) B, [A]+[B] Total . . .
```

	Rok	A	B	Total
1	2006	2.3	2.2	4.5
2	2007	4.8	0.7	5.5

b) Automatický PIVOT ... IN pomocou reťazcových príkazov QUOTENAME, XML PATH, STUFF

Pozri (nové) <https://www.sqlservertutorial.net/sql-server-basics/sql-server-pivot/>

Pri poloautomatickom riešení prvého príkladu budeme využívať nasledujúce funkcie (a tabuľku #T1): QUOTENAME, STUFF - maže a potom vkladá, FOR XML PATH a COALESCE.

```
SELECT QUOTENAME('Podme-domov') -- [Podme-domov]
-- ⇔
SELECT QUOTENAME('Podme-domov',']') -- [Podme-domov]
```

FOR XML PATH - získanie výsledku SQL dotazu vo formáte XML (reťazec). ____
XML bude venovaná jedna prednáška.

```
print STUFF('Bxxlava', 2, 3, 'ratis'); -- Bratislava
```

Postup:

- 1a) Zoznam všetkých hodnôt Kvartálu: `-- 1 // 2 // 3 // 4`
- 1b) Zoznam všetkých hodnôt Kvartálu v [] QUOTENAME : `-- ,[1] // ,[2] // ,[3] // ,[4]`
- 1c) Všetky hodnoty do jedného riadku FOR XML PATH('') `-- ,[1],[2],[3],[4]`
- 1d) Všetky hodnoty s odstránením prvej čiarky STUFF (...) `-- [1],[2],[3],[4]`
- 2c) Odstránenie NULL: COALESCE.

2c) ISNULL nahradí NULL zadanou náhradnou hodnotou.

---- Pivot kódovo, poloautomaticky:

```
DECLARE @zoznam VARCHAR(100)
-- Vrati zoznam: [1],[2],[3],[4]
-- 1a) SELECT DIST., 1b) SELECT QUOTEN., 1c) FOR XML PATH, id) STUFF
SET @zoznam =
STUFF(
(
    SELECT ',' + QUOTENAME(x)
    FROM (
        SELECT DISTINCT(Kvartal) x
        FROM( SELECT * FROM ##T1)
        AS pom
    ) pom
    ORDER BY x
    FOR XML PATH('') -- ,[1],[2],[3],[4]
),1, 1, N''); -- maze z predu ciarku - nahradi s prazdnym znakom

print @zoznam          -- ⇔ [1],[2],[3],[4]
```

```
DECLARE @sql AS NVARCHAR(MAX)
SET @sql = N'
SELECT Rok, ' + @zoznam + '
FROM (SELECT Rok, Kvartal, Obrat FROM ##T1) pom
PIVOT ( SUM(Obrat) FOR Kvartal IN (' + @zoznam + ') ) piv
ORDER BY Rok'
```

```
print @sql
```

	Rok	1	2	3	4
1	2006	1.3	NULL	1.6	1.6
2	2007	0.7	1.8	1.4	1.6

```
EXEC sp_executesql @sql;
```

Nahradíme NULL s 0 pomocou COALESCE, ktorý vráti prvú nie null hodnotu.

Napr.

```
SELECT N FROM ( VALUES(-1),(NULL),(-5) ) xxxTab(N)
SELECT COALESCE(N,0) FROM ( VALUES(-1),(NULL),(-5) ) xxxTab(N)
SELECT COALESCE(N,null,null,55) FROM ( VALUES(-1),(NULL),(-5) ) xxxTab(N)
```

```
SET @sql = N'
```

```
SELECT Rok,
COALESCE([1],0)[1], COALESCE([2],0)[2],COALESCE([3],0)[3],COALESCE([4],0)[4],
COALESCE([1],0)+COALESCE([2],0)+ COALESCE([3],0)+ COALESCE([4],0)Vcelku
FROM (SELECT Rok, Kvartal, Obrat FROM ##T1) pom
PIVOT ( SUM(Obrat) FOR Kvartal IN (' + @zoznam + ') ) piv
ORDER BY Rok'
```

	Rok	1	2	3	4	Vcelku
1	2006	1.3	0.0	1.6	1.6	4.5
2	2007	0.7	1.8	1.4	1.6	5.5

c) Pivot tabul'ky bez Pivot

Pomocou `SUM(CASE WHEN Kvartal = 1 THEN ...`

```
SELECT Rok, SUM(Obrat) AS Vcelku
FROM ##T1
GROUP BY Rok
go
```

	Rok	Vcelku
1	2006	4.5
2	2007	5.5

```
SELECT Rok, Kvartal, SUM(Obrat) AS Vcelku
FROM ##T1
GROUP BY Rok, Kvartal
Order by rok, Kvartal
Go
```

	Rok	Kvartal	Vcelku
1	2006	1	1.3
2	2006	3	1.6
3	2006	4	1.6
4	2007	1	0.7
5	2007	2	1.8
6	2007	3	1.4
7	2007	4	1.6

```
SELECT Rok,
SUM(CASE WHEN Kvartal = 1 THEN Obrat ELSE 0 END) [Kv 1],
SUM(CASE WHEN Kvartal = 2 THEN Obrat ELSE 0 END) [Kv 2],
SUM(CASE WHEN Kvartal = 3 THEN Obrat ELSE 0 END) [Kv 3],
SUM(CASE WHEN Kvartal = 4 THEN Obrat ELSE 0 END) [Kv 4],
SUM(Obrat) AS Vcelku
FROM ##T1
GROUP BY Rok
```

	Rok	Kv 1	Kv 2	Kv 3	Kv 4	Vcelku
1	2006	1.3	0.0	1.6	1.6	4.5
2	2007	0.7	1.8	1.4	1.6	5.5

d) Pivot tabul'ky - Power BI a Excel

	Excel Power BI		Power BI
	Power Query	Power Pivot	Desktop & PowerBI.com
Role	Import and shape data	Data modeling and calculations	Complete business intelligence tool
Language	M	DAX	M and DAX
Key strengths	<ul style="list-style-type: none"> Nice easy to use interface Powerful tools to import and clean data All Excel users can benefit from this tool 	<ul style="list-style-type: none"> Easily handle millions of rows of data Modeling tools for efficient data storage and analysis Powerful DAX calculations going beyond standard Excel 	<ul style="list-style-type: none"> Incredible visualization options Simple built-in interactive options Powerful DAX calculations Simple publishing to PowerBI.com and mobile devices

<https://www.goskills.com/Excel/Resources/Power-query-vs-power-pivot-power-bi>

Zdroj

[Query formula language M to combine and filter data](#)

[Formula expression query language - Data Analysis Expressions \(DAX\)](#)

<https://powerbi.com/>

Excel

- Power pivot – milióny riadkov

- Power view

Oddel	Rok	Kvartal	Obrat
A	2006	1	0.6
B	2006	1	0.7
A	2006	3	0.9
B	2006	3	0.7
A	2006	4	0.8
B	2006	4	0.8
A	2007	1	0.7
A	2007	2	0.9
A	2007	2	0.9
A	2007	3	0.8
A	2007	3	0.6
A	2007	4	0.9
B	2007	4	0.7

Book1 - Excel

POWERPIVOT

PivotTable Fields

Choose fields to add to report:

- Oddelenie
- Rok
- Kvartal
- Obrat_mil

Drag fields between areas below:

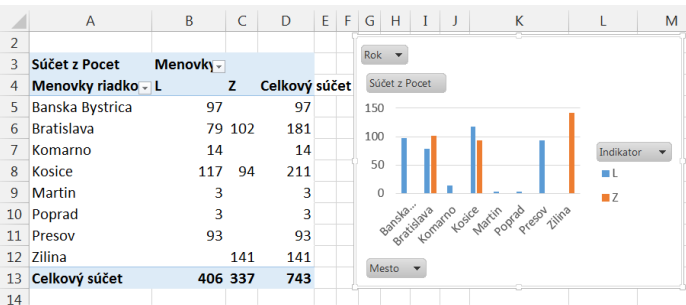
FILTERS: Oddelenie, Rok

COLUMNS: Kvartal

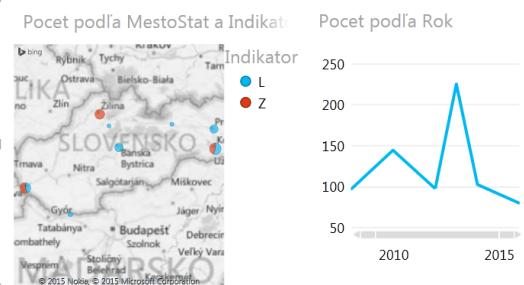
ROWS: Rok

VALUES: Sum of Obrat_mil

Mesto	Indikator	Pocet	Stat	Rok	MestoStat
Bratislava	L	79	Sk	2016	Bratislava,Sk
Presov	L	93	Sk	2008	Presov,Sk
Bratislava	Z	102	Sk	2014	Bratislava,Sk
Martin	L	3	Sk	2008	Martin,Sk
Banska Bystrica	L	97	Sk	2012	Banska Bystrica,Sk
Zilina	Z	141	Sk	2010	Zilina,Sk
Poprad	L	3	Sk	2010	Poprad,Sk
Kosice	L	117	Sk	2013	Kosice,Sk
Komarno	L	14	Sk	2013	Komarno,Sk
Kosice	Z	94	Sk	2013	Kosice,Sk



MestoStat	Pocet	Ind...	Rok
Banska...	97	L	2012
Bratislava,Sk	79	L	2016
Bratislava,Sk	102	Z	2014
Komarno,Sk	14	L	2013
Kosice,Sk	117	L	2013
Kosice,Sk	94	Z	2013
Martin,Sk	3	L	2008
Poprad,Sk	3	L	2010
Presov,Sk	93	L	2008
Zilina,Sk	141	Z	2010
Kelcom	743	20...	



Polia kontingenčného ...

Vyberte polia, ktoré chcete pridať do zostavy:

- Mesto
- Indikator
- Pocet
- Stat
- Rok
- MestoStat

Presuňte polia medzi nižšie uvedenými oblasťami:

FILTRE: Rok, Mesto

LEGENDA (RAD): Indikator

OS (KATEGÓRIE): Mesto

HODNOTY: Súčet z Pocet