

6b) Kontingenčné - Pivot tabuľky

3a) Klauzula PIVOT

3b) Manualny PIVOT

3c) Pivot tabuľky bez Pivot

3d) Pivot tabuľky - Power BI, Excel a R

3a) Klauzula PIVOT

PT je dvojrozmerná **agregačná** (frekvenčná, sumárna, priemerná, ...) tabuľka, zovšeobecňujúca GROUP BY podľa dvoch stĺpcov, atribútov. PT je dôležitý nástroj pre 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 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. Vytvoríme tabuľku piv1 na pivotovanie.

```
USE tempdb;
```

```
if OBJECT_ID('piv1', 'U') IS NOT NULL DROP TABLE piv1
```

```
CREATE TABLE piv1(Oddel Char, Rok SMALLINT, Kvartal TINYINT, Obrat DECIMAL(2,1))
```

```
GO
```

```
INSERT INTO piv1 (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 piv1) pom
  PIVOT (SUM(Obrat) FOR Kvartal IN ([1],[2],[3],[4])) piv
  ORDER BY Rok DESC
```

---- V hre budú/su tie stĺpce, ktoré vyššie vo vnorenom dopyte selektujeme.
 ---- Z hodnôt stĺpca Kvartal tvoríme maximálne 4 nové stĺpce.

Sumárny obrat jednotlivých oddelení v každom roku:

```
SELECT Rok, [A],[B]
FROM (SELECT Rok, Oddel, Obrat FROM piv1) 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 FROM ...
```

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

Aj horizontálne sumy pomocou CTE WITH

```
WITH PivotData AS (
  SELECT Rok, ISNULL([A],0) AS A, ISNULL([B],0) AS B, (ISNULL([A],0) + ISNULL([B],0))
AS Total
FROM (SELECT Rok, Oddel, Obrat FROM piv1) pom
PIVOT (SUM(Obrat) FOR Oddel IN ([A],[B])) piv
)
SELECT
  CASE WHEN GROUPING(Rok) = 1 THEN 'TOTAL' ELSE CAST(Rok AS VARCHAR) END AS Rok,
  SUM(A) AS A,
  SUM(B) AS B,
  SUM(Total) AS Total
FROM PivotData
GROUP BY ROLLUP(Rok);
```

	Rok	A	B	Total
1	2006	2.3	2.2	4.5
2	2007	4.8	0.7	5.5
3	TOTAL	7.1	2.9	10.0

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('Bxxxlava', 2, 3, 'ratis'); -- Bratislava
print STUFF('[1],[2],[3],[4]',1, 1, N''); -- POZOR, ak online editor!
-- vs:
CREATE TABLE tuok (x int); INSERT INTO tuok VALUES(111);
SELECT STUFF('[1],[2],[3],[4]',1, 1, N'') FROM tuok;
```

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 piv1)
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]
```

3b) Manualny Pivot

```
DECLARE @sql AS NVARCHAR(MAX)
SET @sql = N'
SELECT Rok, ' + @zoznam + '
FROM (SELECT Rok, Kvartal, Obrat FROM piv1)pom
PIVOT ( SUM(Obrat) FOR Kvartal IN (' + @zoznam + ' ) ) piv
ORDER BY Rok'
```

```
print @sql
EXEC sp_executesql @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

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 piv1)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

```
EXEC sp_executesql @sql;
```

c) Pivot tabuľky bez Pivot

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

```
SELECT Rok, SUM(Obrat) AS Vcelku
FROM piv1
GROUP BY Rok
go
```

	Rok	Vcelku
1	2006	4.5
2	2007	5.5

```
SELECT Rok, Kvartal, SUM(Obrat) AS Vcelku
FROM piv1
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 piv1
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

Pomocou CROSS APPLY

```
SELECT
CASE WHEN GROUPING(t.Rok) = 1 THEN 'TOTAL'
ELSE CAST(t.Rok AS VARCHAR) END AS Rok,
SUM(v.A) AS A,
SUM(v.B) AS B,
-- Horizontálna suma (súčet stĺpcov v riadku)
SUM(v.A + v.B) AS Total
FROM piv1 AS t
CROSS APPLY (
VALUES (
CASE WHEN t.Oddel = 'A' THEN t.Obrat ELSE 0 END,
CASE WHEN t.Oddel = 'B' THEN t.Obrat ELSE 0 END
)
) AS v(A, B)
GROUP BY ROLLUP(t.Rok);
```

	Rok	A	B	Total
1	2006	2.3	2.2	4.5
2	2007	4.8	0.7	5.5
3	TOTAL	7.1	2.9	10.0

Na ozreženie úlohy v(A,B), uvažujme kód

```
SELECT
    t.Oddel, t.Rok, t.Kvartal, t.Obrat,
    v.Q1, v.Q2, v.Q3, v.Q4
FROM piv1 AS t
CROSS APPLY (
    VALUES (
        CASE WHEN t.Kvartal = 1 THEN t.Obrat END,
        CASE WHEN t.Kvartal = 2 THEN t.Obrat END,
        CASE WHEN t.Kvartal = 3 THEN t.Obrat END,
        CASE WHEN t.Kvartal = 4 THEN t.Obrat END
    )
) AS v(Q1, Q2, Q3, Q4);
```

	Oddel	Rok	Kvartal	Obrat	Q1	Q2	Q3	Q4
6	B	2006	4	0.8	NULL	NULL	NULL	0.8
7	A	2007	1	0.7	0.7	NULL	NULL	NULL
8	A	2007	2	0.9	NULL	0.9	NULL	NULL
9	A	2007	2	0.9	NULL	0.9	NULL	NULL
10	A	2007	3	0.8	NULL	NULL	0.8	NULL
11	A	2007	3	0.6	NULL	NULL	0.6	NULL
12	A	2007	4	0.9	NULL	NULL	NULL	0.9
13	B	2007	4	0.7	NULL	NULL	NULL	0.7

3d) Pivot tabuľky - Power BI, Excel a R

d1) Power BI

	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](#)

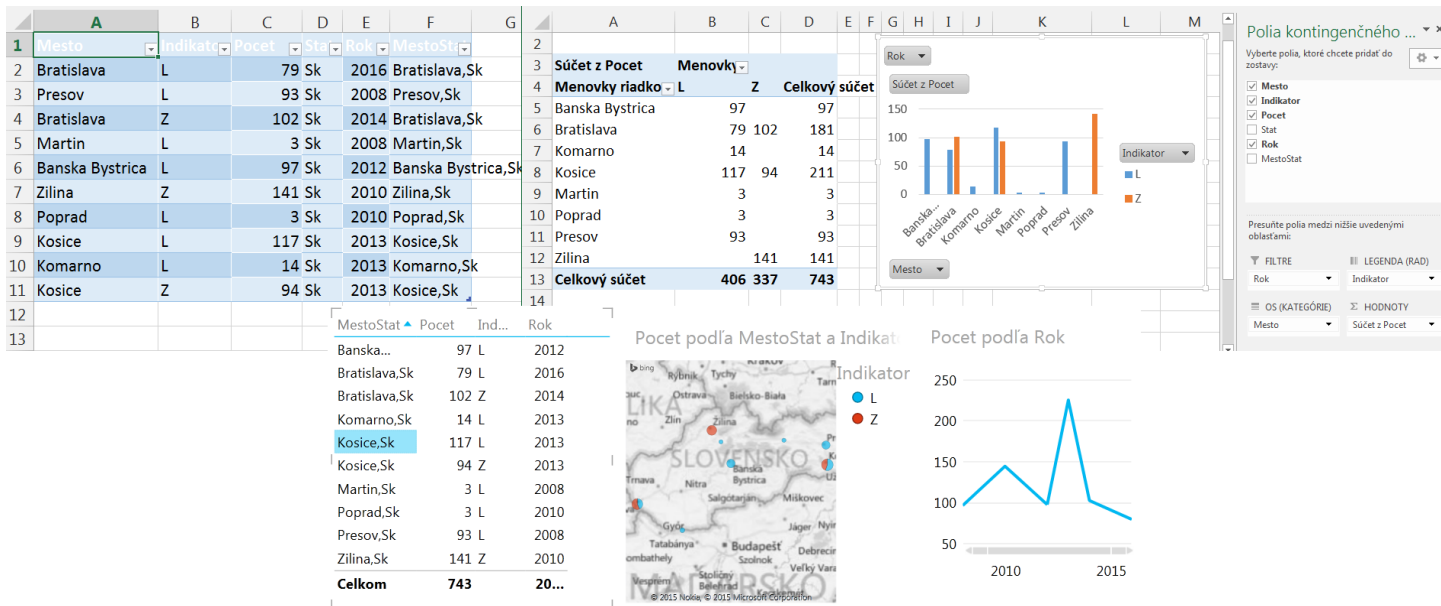
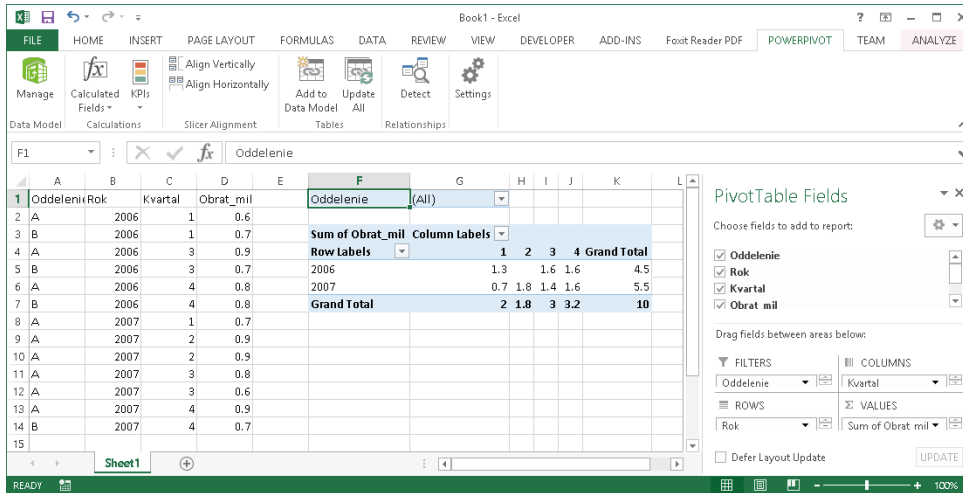
[Formula expression query language - Data Analysis Expressions \(DAX\)](#)

<https://powerbi.com/>

d2) Excel

- Power pivot – milióny riadkov

- Power view



d3) R

```
{ # PT rok vs Oddel
library(dplyr)
library(tidyr)
#install.packages("janitor")
library(janitor)
```

```

t1 <- data.frame(
  Oddel = c('A', 'B', 'A', 'B', 'A', 'B', 'A', 'A', 'A', 'A',
'A', 'A', 'B'),
  Rok = c(2006, 2006, 2006, 2006, 2006, 2006, 2007, 2007,
2007, 2007, 2007, 2007, 2007),
  Kvartal = c(1, 1, 3, 3, 4, 4, 1, 2, 2, 3, 3, 4, 4),
  Obrat = c(0.6, 0.7, 0.9, 0.7, 0.8, 0.8, 0.7, 0.9, 0.9, 0.8,
0.6, 0.9, 0.7)
)
t1 %>%
  group_by(Rok, Oddel) %>%
  summarise(Total_Obrat = sum(Obrat), .groups = 'drop') %>%
  pivot_wider(
    names_from = Oddel,
    values_from = Total_Obrat
  ) %>%
  arrange(Rok) %>%
  # 1. Replace NA with 0 so totals calculate correctly
  mutate(across(everything(), ~replace_na(., 0))) %>%
  # 2. Add row totals and column totals
  adorn_totals(where = c("row", "col"))
}

{ # Hierarchicka PT - rok, kvartal vs Oddel
t1 %>%
  # Agregácia hodnôt podľa Roku, Kvartálu a Oddelenia
  group_by(Rok, Kvartal, Oddel) %>%
  summarise(Suma_Obrat = sum(Obrat), .groups = 'drop') %>%

  # Rozdelenie Oddelení (A, B) do stĺpcov
  pivot_wider(names_from = Oddel, values_from = Suma_Obrat)
%>%

  # Pridanie súčtového stĺpca (Grand Total na úrovni riadku)
  rowwise() %>%
  mutate(Grand_Total = sum(c_across(A:B), na_rm = TRUE)) %>%
  ungroup() %>%

  # Pridanie súčtového riadku (Grand Total na úrovni stĺpcov)
  pomocou balíka janitor
  adorn_totals("row")
}

```