

# 1 Kontingenčné/Pivot tabuľky

## 2 Ošetrovanie chýb

### 1) Kontingenčné/Pivot tabuľky

PT je dvojrozmerná agregáčna (frekvenčná, sumárna, priemerná, ...) tabuľka, zovšeobecňujúca GROUP BY podľa dvoch stĺpcov, atribútov.

Pivot tabuľky môžeme vytvoriť

- a) manuálne PIVOT ... IN ...
- b) dynamicky IN pomocou reťazcových príkazov STUFF, XML PATH a COALESCE
- c) bez Pivot SUM(CASE WHEN ...
- d) Pivot tabuľky a Excel

#### 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 ...

```
USE tempdb;  
GO
```

```
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;  
GO
```

**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>;
```

---- a) Pivot manuálne - vo výsledku riadok, stĺpec:  
---- Sumárny obrat v jednotlivých kvartáloch v každom roku:

```
SELECT Rok, [1],[2],[3],[4]
FROM (SELECT Rok, Kvartal, Obrat FROM #T1) pom
PIVOT (SUM(Obrat) FOR Kvartal IN ([1],[2],[3],[4])) piv
ORDER BY Rok
```

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

---- 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
```

## b) Dynamická PT

0a) QUOTENAME, STUFF

```
SELECT QUOTENAME('Podme-domov','') -- [Podme-domov]
print STUFF('Bxxlava', 2, 3, 'ratis'); -- Bratislava
```

0b) Tabuľka #T1

1a) Zoznam všetkých hodnôt Kvartálu:

1b) Zoznam všetkých hodnôt Kvartálu v [ ]:

1c) Všetky hodnoty s odstránením prvej čiarky

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

---- b) Pivot kódovo, dynamicky:

```
DECLARE @zoznam VARCHAR(100)
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'

---- COALESCE - prva ne-null hodnota
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'
print @sql
EXEC sp_executesql @sql;

```

### c) Pivot tabuľky bez Pivot

```

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

```

```

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

	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

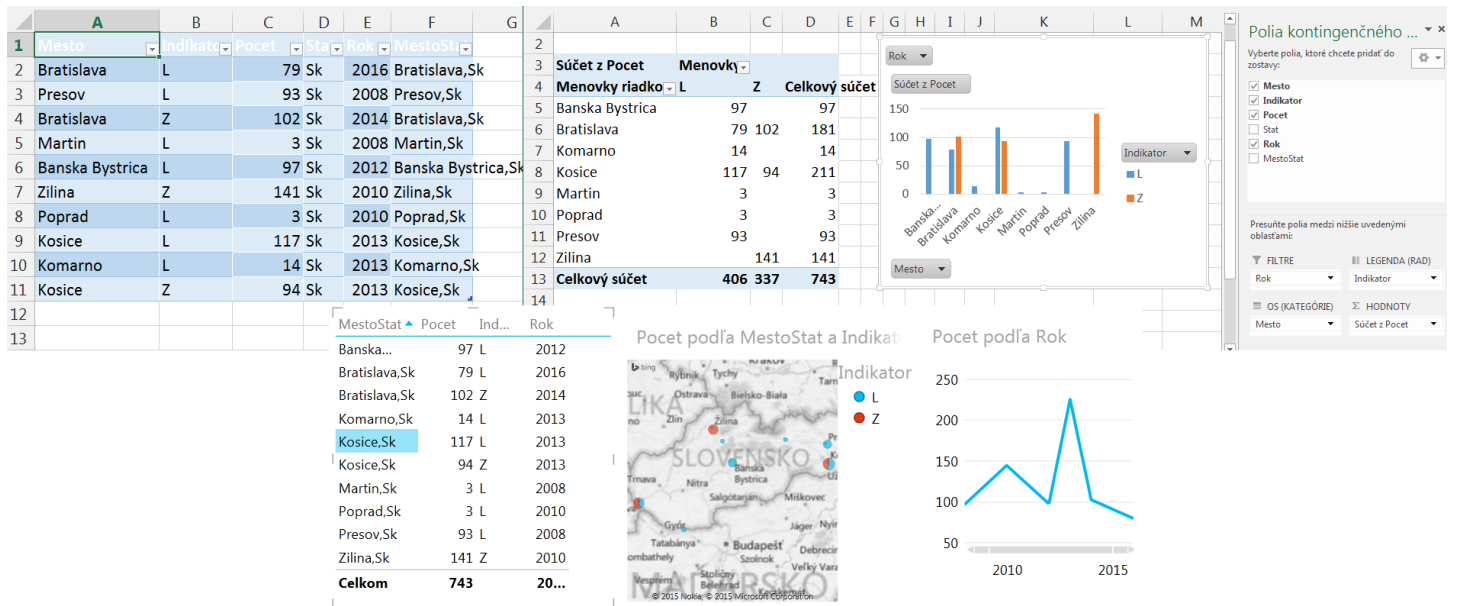
```

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

```

## d) Pivot tabuľky a Excel

- Power pivot – milióny riadkov
- Power view



## 2) Ošetrenie chýb

- a1) `sys.messages` pre `@@ERROR`
- a2) `RAISERROR <--> PRINT:`
- b1) Starý spôsob: `@@ERROR, @@ROWCOUNT;`
- b21) Nový spôsob: `TRY - CATCH`
- b22) S užívateľskou procedúrou

```
---- a1) sys.messages pre @@ERROR
select * from sys.messages where message_id = 547
```

```
---- a2) RAISERROR <--> PRINT:
--      druhy par. severity: 0-10, 11-25:
--      treti par. state: 0-127
RAISERROR('Haha', 10, 1)
RAISERROR('Haha', 11, 1)
```

- b1) Starý spôsob SQL Server 2000: `@@ERROR, @@ROWCOUNT;`

```
----- The DELETE statement conflicted with the REFERENCE constraint ...
```

```
USE Poliklinika
```

```
GO
```

```
DECLARE @ErrorVar INT;
```

```
DECLARE @RowCountVar INT;
```

```
DELETE FROM Lekari WHERE idL = 1;
```

```
-- ULOZ!!!
```

```
SELECT @ErrorVar = @@ERROR,
       @RowCountVar = @@ROWCOUNT;
```

```
IF (@ErrorVar <> 0)
```

```
    BEGIN
```

```
        PRINT N' - Cislo chyby = ' + CAST(@ErrorVar AS NVARCHAR(8));
```

```
        PRINT N' - Pocet vymazaných riadkov = ' + CAST(@RowCountVar AS
```

```
NVARCHAR(8));
```

```
    END
```

```
GO
```

- b21) Nový spôsob: `BEGIN TRY - BEGIN CATCH:`

```
USE Poliklinika
```

```
GO
```

```
BEGIN TRY
```

```
    DELETE FROM Lekari WHERE idL = 1;
```

```
END TRY
```

```
BEGIN CATCH
```

```
    SELECT
```

```
        ERROR_NUMBER() AS ErrorNumber,
```

```
        ERROR_SEVERITY() AS ErrorSeverity,
```

```
        ERROR_STATE() AS ErrorState,
```

```
        ERROR_PROCEDURE() AS ErrorProcedure,
```

```
        ERROR_LINE() AS ErrorLine,
```

```
        ERROR_MESSAGE() AS ErrorMessage;
```

```
    RAISERROR ('Chyba ...', 11, 1)
```

```
END CATCH
```

### Ďalší příklad:

```
BEGIN TRY
    SELECT GETDATE()
    SELECT 1/0 Bude_Chya
    SELECT GETDATE()
END TRY
BEGIN CATCH
    RAISERROR ('Chyba pri deleni', 11, 1)
END CATCH;
```

### b22) S uživatelskou procedúrou Chyba:

```
USE Poliklinika;
GO
```

```
IF OBJECT_ID ( 'Chyba', 'P' ) IS NOT NULL DROP PROCEDURE Chyba;
GO
```

```
CREATE PROCEDURE Chyba
AS
```

```
    SELECT -- DALO BY SA AJ PRINT
        ERROR_NUMBER() AS ErrorNumber,
        ERROR_SEVERITY() AS ErrorSeverity,
        ERROR_STATE() AS ErrorState,
        ERROR_PROCEDURE() AS ErrorProcedure,
        ERROR_LINE() AS ErrorLine,
        ERROR_MESSAGE() AS ErrorMessage;
    RAISERROR ('Chyba! Pozri vysled dopytu', 11, 1)
GO
```

```
BEGIN TRY
    DELETE FROM Lekari WHERE idL = 1;
END TRY
BEGIN CATCH
    EXEC Chyba
END CATCH;
```