

Práca s viacerými tabuľkami (1)

Vnorené dopyty

- 1) Typy výsledkov dopytu
- 2) Základy vnorených dopytov (VD)
- 3) Pravidlá zápisu VD
- 4) Miesta VD v SELECTe
- 5) Typy VD
- 6) Príklady - Poliklinika

1) Typy výsledkov dopytu

- tabuľka (viac riadkov, viac stĺpcov)
- jeden stĺpec/zoznam
- jediná/skalárna hodnota
- nič

SELECT môže byť všade, kde je

- skalárna hodnota
- zoznam
- tabuľka

Kde môže byť zoznam:

```
SELECT select_zoznam
      FROM from_zoznam
      WHERE ... IN (in_zoznam)
      ...
```

2) Základy vnorených dopytov

Vnorený dopyt ([Subquery Types](#)) je dopyt (SELECT), ktorý je vnorený do (vonkajších) príkazov

SELECT, INSERT, UPDATE, DELETE

alebo ďalšieho vnoreného dopytu.

- Vonkajší dopyt
Outer query
- Vnorený dopyt ⇔ Vnútorňý dopyt ⇔ (Poddopyt)
Subquery, Inner query

Vnorený dopyt sa snád' najčastejšie vyskytuje vo WHERE

... IN (http://en.wikipedia.org/wiki/Correlated_subquery, [Correlated Subqueries](#)) klauzule

vonkajšieho SELECTu, ale nie len tam, veď ako SELECT môže byť tam, kde zoznam (pozri vyššie). Ak VD vracia skalárnu hodnotu, potom môže byť všade, kde môže byť výraz.

Väčšina príkazov s vnorenými dopytmi môže byť sformulovaný ako JOIN. Ktoré riešenie je efektívnejšie, závisí od takých faktorov, ako je napr. veľkosť tabuliek, prítomnosť indexov.

Rozdiel medzi VD a JOIN:

- vnorený dopyt je SELECT príkaz, JOIN nie
- vnorený dopyt sa môže nachádzať na rôznych miestach SELECTu, JOIN sa nachádza za FROM.

3) Pravidlá zápisu vnorených dopytov !!!

- VD sa vždy píše do zátvoriek

```
SELECT ... ( SELECT ... ) ...
```

- ak sa nachádza v from_zozname, potom treba alias:

```
SELECT * FROM (SELECT * FROM Lekari L1 ) v1,  
              (SELECT * FROM Lekari L2 ) AS v2
```

- maximalný počet vnorených dopytov 32

Poznámka: síce tu ešte nevidíme význam aliasov L1, L2, ale ich použitie sa odporúča aby zápis bol **prehľadnejší**.

4) Miesta vnorených dopytov v SELECTe

Napr.

- hneď za SELECT v select_zozname

- za FROM v from_zozname

```
SELECT 111 FROM Lekari
```

-- =>

```
SELECT (SELECT krstne FROM Lekari WHERE idL=1)  
      FROM (SELECT * FROM Lekari ) AS s1
```

- v IN

```
SELECT * FROM Lekari L  
      WHERE L.idL*2 IN( SELECT idL FROM Lekari)
```

-- =>

```
SELECT (SELECT krstne FROM Lekari WHERE idL=1)
```

```
FROM (SELECT * FROM Lekari WHERE idL
      IN (SELECT idL FROM Lekari WHERE idL <4)
      ) AS s1
```

- WHERE vyraz [NOT] IN (VD)
- WHERE vyraz < [ANY | ALL] (VD)
- WHERE [NOT] EXISTS (VD)
- GROUP BY, HAVING ...

5) Typy vnorených dopytov

- nezávislý VD / VD bez parametra
- korelovaný VD / VD s parametrom (correlated subquery)
 - postup vykonania!!!
 - počet navštívených riadkov
 - prepojenie tabuliek, kľúče
- self query

6) Príklady

V nasledujúcich príkladoch sa opierame o databázu **Poliklinika**:

Pacienti				Navstevy				
idP	krstne	mesPrijem		idN	idP	idL	den	poplatok
1	Adam	10000		1	1	2	2008-05-05 00:00:00.000	NULL
2	Stefan	9500		2	2	3	2008-05-05 00:00:00.000	NULL
3	Slavo	8500		3	6	3	2008-05-05 00:00:00.000	NULL
4	Klara	9000		4	4	1	2008-06-05 00:00:00.000	200
5	Zuzana	35000		5	5	4	2008-06-05 00:00:00.000	500
6	Tana	20000		6	7	1	2008-06-05 00:00:00.000	200
7	Mato	28000		7	6	1	2008-06-05 00:00:00.000	500
8	Zoli	32000		8	8	3	2008-07-05 00:00:00.000	900
9	Misko	NULL		9	2	1	2008-07-05 00:00:00.000	200
10	Janka	NULL		10	3	3	2008-07-05 00:00:00.000	100
				11	6	2	2008-08-05 00:00:00.000	700
				12	7	2	2008-08-05 00:00:00.000	500
				13	6	4	2008-08-05 00:00:00.000	800
				14	2	1	2008-09-05 00:00:00.000	NULL
				15	3	1	2008-09-05 00:00:00.000	200
				16	8	1	2008-09-05 00:00:00.000	200
				17	9	5	2008-09-05 00:00:00.000	NULL
				18	7	1	2008-10-05 00:00:00.000	300
				19	8	4	2008-10-05 00:00:00.000	800
				20	10	5	2008-10-05 00:00:00.000	300
				21	1	1	2008-11-05 00:00:00.000	350
				22	6	5	2008-11-05 00:00:00.000	400

Lekari			
idL	krstne	spec	datNar
1	Oto	Ocny	1960-05-05 00:00:00.000
2	Zoli	Zubny	1961-11-14 00:00:00.000
3	Klara	Kardiolog	1980-02-15 00:00:00.000
4	Zuzka	Zubny	1970-04-02 00:00:00.000
5	Imro	Intery	1956-11-09 00:00:00.000

Nižšie budeme uvádzať riešenia aj s JOIN, napr. tomu, že je to téma ďalšej prednášky.

---- 1a) Zistite id pacientov, ktorí uz boli u kardiologa
---- - teda u lekara s id 3.

```
SELECT idP FROM Navstevy  
        WHERE idL = 3
```

---- Pokracovanie:
---- 1b) Zistite id a mena pacientov, ktorí uz boli u
---- kardiologa
---- 3 riesenia:

---- 1. Riesenie:

---- JE V ZOZNAME/VD <=> ROVNA SA PRVKU ZOZNAMU/VD:

```
SELECT P.idP, P.krstne  
        FROM Pacienti P  
        WHERE P.idP IN( SELECT N.idP --zoznam pac. u 3  
                        FROM Navstevy N  
                        WHERE N.idL = 3 )
```

---- - nezavisly VD
---- - Su nutne P a N? ().
---- - Moze byt: SELECT N.idP ...

---- <=>:

---- 2. riesenie:

---- ROVNA SA NIEKTOREMU /EXISTUJE TAKY/ PRVKU V ZOZNAME/VD:

```
SELECT P.idP, P.krstne  
        FROM Pacienti P  
        WHERE P.idP = Any( SELECT N.idP  
                           FROM Navstevy N  
                           WHERE N.idL = 3 )
```

---- <=>:

---- 3. riesenie (po preštudovaní nasl. príkladu, ľahko
napíšete ďalšie verzie dopytu):

```
SELECT P.idP, P.krstne FROM Pacienti P  
        JOIN Navstevy N On P.idP = N.idP  
        WHERE idL = 3
```

---- 3) Zistite id a mena pacientov, ktorí su aj lekarmi

```
---- (pacient a lekar s tym istym menom)
---- 6 rieseni:
----   - 1, 2) nezavisle VD - IN, ANY
----   - 3) korelovany VD - Exists }
----   - 4) zoznam tabuliek       }
----   - 5, 6) CROSS a INNER JOIN }
```

```
---- 1):
SELECT idP, krstne
      FROM Pacienti
      WHERE krstne IN (SELECT krstne
                      FROM Lekari)
-- <=>
```

```
SELECT P.idP, P.krstne
      FROM Pacienti P
      WHERE P.krstne IN (SELECT L.krstne
                       FROM Lekari L)
```

```
---- <=> 2):
SELECT P.idP, P.krstne
      FROM Pacienti P
      WHERE P.krstne = ANY (SELECT L.krstne
                          FROM Lekari L)
```

```
---- <=> 3) Korelovany dopyt
----         EXISTS / Existuje / taky riadok vo VD:
SELECT P.idP, P.krstne
      FROM Pacienti P
      WHERE EXISTS (SELECT * FROM Lekari L
                   WHERE L.krstne = P.Krstne)
```

```
---- <=> 4):
SELECT P.idP, P.krstne
      FROM Pacienti P, Lekari L
      WHERE L.krstne = P.Krstne
```

```
---- <=> 5):
SELECT P.idP, P.krstne
      FROM Pacienti P CROSS JOIN Lekari L
      WHERE L.krstne = P.Krstne
```

----- <=> 6):

```
SELECT P.idP, P.krstne
      FROM Pacienti P JOIN Lekari L
      On L.krstne = P.Krstne
```

Subquery Fundamentals [http://msdn.microsoft.com/en-us/library/ms189575\(v=sql.105\).aspx](http://msdn.microsoft.com/en-us/library/ms189575(v=sql.105).aspx)
Join Fundamentals [http://msdn.microsoft.com/en-us/library/ms191517\(v=sql.105\).aspx](http://msdn.microsoft.com/en-us/library/ms191517(v=sql.105).aspx)