Operátory ROLLUP a CUBE

Dotazovací jazyky, 2009

Marek Polák
Martin Chytil

Osnova přednášky

- Analýza dat
- Agregační funkce
- GROUP BY a jeho problémy
- Speciální hodnotový typ ALL
- Operátor CUBE
- Operátor ROLLUP
- Hodnota NULL a funkce GROUPING
- Výpočet dotazů s ROLLUP a CUBE

Motivace (1)

- V současnosti jsou nejběžnější relační databáze
- Subjekty uložené v databázi mají velké množství parametrů (vlastností).

- Příklad

  - Zboží – název, cena, kategorie
  - Zákazník – jméno, adresa, email, telefonní číslo

Motivace (2)

- Potřeba dotazovat se z mnoha pohledů – na mnoho parametrů a jejich kombinace.

- Různé typy dotazů
  - Na konkrétní záznam
    - objednávka zákazníka X
    - zajímá prodavače

  - Agregovaná data
    - celková cena všech objednávek zákazníků X a Y za poslední rok
    - hlavně pro manažery

Motivace (3)

- Kolik utratil každý zákazník v každém měsíci za zboží z každé skupiny?
- Kolik celkem v každé skupině?
- Kolik celkem v každém měsíci?
- Kolik celkem utratil každý zákazník?
- Kolik utratili všichni dohromady?

- … trochu jiný dotaz …
- Kolik kdy utratil zákazník „Franta Vopršálek“ za elektroniku?

Datová analýza

- Formulace dotazu
- Získání agregovaných dat
- Vizualizace výsledků
  - Histogramy, křížové tabulky, částečné součty
- Analýza výsledků

Histogramy (1)

- Jeden ze způsobů zobrazení dat
- Agregace přes počítané kategorie
- Počasí(Čas, Zem. Šířka, Zem. Výška, Teplota)
- Chceme: Pro každé území maximální denní teplotu.

Histogramy (2)

- SELECT den, území, MAX(teplota)
  FROM Počasí
  GROUP BY Den(Čas) AS den,
  Území(šířka, výška) AS území

- SQL neumožňuje přímo - nutné používat hnízděné dotazy

- SELECT den, území, MAX(teplota)
  FROM (SELECT Den(čas) AS den, Území(šířka, výška) AS území, teplota FROM Počasí) AS T
  GROUP BY den, území

Operátor GROUP BY - opakování

- Standardní SQL příkaz pro možnost dotazování se po agregacích
- Umožňuje seskupit řádky podle zvolených sloupců do agregací – superřádků
- Na takto seskupená data (superřádky) je možno aplikovat různé agregační funkce

Operátor GROUP BY - syntaxe

- SELECT
  {<sloupec> | <výraz>, …}
  FROM
  <zdroj dat>
  WHERE
  <podmínka>
  GROUP BY {<jméno sloupce> [zvláštní podmínka], …}

Agregační funkce

- Standardní agregační funkce:
  - COUNT()
  - MIN()
  - MAX()
  - SUM()
  - AVG()
- Doménově specifické
- Vlastní

Použitá data v příkladech

- Tabulka produkce automobilů

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	stribrna	146
2	Octavia	2005	stribrna	300
3	Octavia	2007	bila	209
4	Superb	2008	vinova	502
5	Superb	2009	stribrna	450
6	Octavia	2000	bila	100
7	Octavia	2005	bila	135
8	Octavia	2007	stribrna	650

Dotaz pomocí GROUP BY

```
SELECT
    typ,
    rok_vyroby,
    barva,
    sum(pocet) AS pocet
FROM
    auta
WHERE
    typ='Octavia'
GROUP BY
    typ,
    rok_vyroby,
    barva
```

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	bila	100
2	Octavia	2005	bila	135
3	Octavia	2007	bila	209
4	Octavia	2000	stribrna	146
5	Octavia	2005	stribrna	300
6	Octavia	2007	stribrna	650

Dotaz pomocí GROUP BY

- Problém
  - Agregaci lze použít pouze pro jednu dimenzi
  - Z předešlého dotazu už například nezjistíme, kolik se vyrobilo Octavií v roce 2000.
  - Řešením může být použití příkazu UNION pro každou požadovanou dimenzi

Hodnotový typ ALL

- Speciální hodnotový typ
- Použití společně s agregačními funkcemi
- Z příslušného sloupce vybere všechny řádky
- Sloupec není zahrnut v příkazu GROUP BY

Dotaz pomocí spojení více dotazů

```
SELECT 'ALL', 'ALL', 'ALL', SUM(pocet)
FROM auta WHERE typ='Octavia'
        UNION
SELECT typ, 'ALL', 'ALL', SUM(pocet)
FROM auta WHERE typ='Octavia'
GROUP BY typ
        UNION
SELECT typ, rok_vyroby, 'ALL',
SUM(pocet)
FROM auta WHERE typ='Octavia'
GROUP BY typ, rok_vyroby
        UNION
SELECT typ, rok_vyroby, barva,
SUM(pocet)
FROM auta WHERE typ='Octavia'
GROUP BY typ, rok_vyroby,barva
        UNION
SELECT typ, 'ALL', barva, SUM(pocet)
FROM auta WHERE typ='Octavia'
GROUP BY typ, barva
        UNION
SELECT 'ALL', rok_vyroby, 'ALL',
SUM(pocet)
FROM auta WHERE typ='Octavia'
GROUP BY rok_vyroby
        UNION
SELECT 'ALL', 'ALL', barva, SUM(pocet)
FROM auta WHERE typ='Octavia'
GROUP BY barva
        UNION
SELECT 'ALL', rok_vyroby, barva,
SUM(pocet)
FROM auta
WHERE typ='Octavia'
GROUP BY rok_vyroby, barva
```

Výsledek „krátkého“ dotazu

	typ	rok_vyroby	barva	pocet
1	ALL	2000	ALL	246
2	ALL	2000	bila	100
3	ALL	2000	stribrna	146
4	ALL	2005	ALL	435
5	ALL	2005	bila	135
6	ALL	2005	stribrna	300
7	ALL	2007	ALL	859
8	ALL	2007	bila	209
9	ALL	2007	stribrna	650
10	ALL	ALL	ALL	1540
11	ALL	ALL	bila	444
12	ALL	ALL	stribrna	1096
13	Octavia	2000	ALL	246
14	Octavia	2000	bila	100
15	Octavia	2000	stribrna	146
16	Octavia	2005	ALL	435
17	Octavia	2005	bila	135
18	Octavia	2005	stribrna	300
19	Octavia	2007	ALL	859
20	Octavia	2007	bila	209
21	Octavia	2007	stribrna	650
22	Octavia	ALL	ALL	1540
23	Octavia	ALL	bila	444
24	Octavia	ALL	stribrna	1096

Nevýhody GROUP BY spolu s UNION

- Pokud má tabulka například 6 sloupců a přes agregaci chceme provést přes všechny (6D křížová tabulka), je potřeba již 64 GROUP BY a 64 sjednocení !!!
- Výpočetně náročné
- Prochází data pro každý poddotaz znovu, výsledek nutné setřídit
- Použitelné pouze pro „malé tabulky“
- Řešení – použití operátoru CUBE

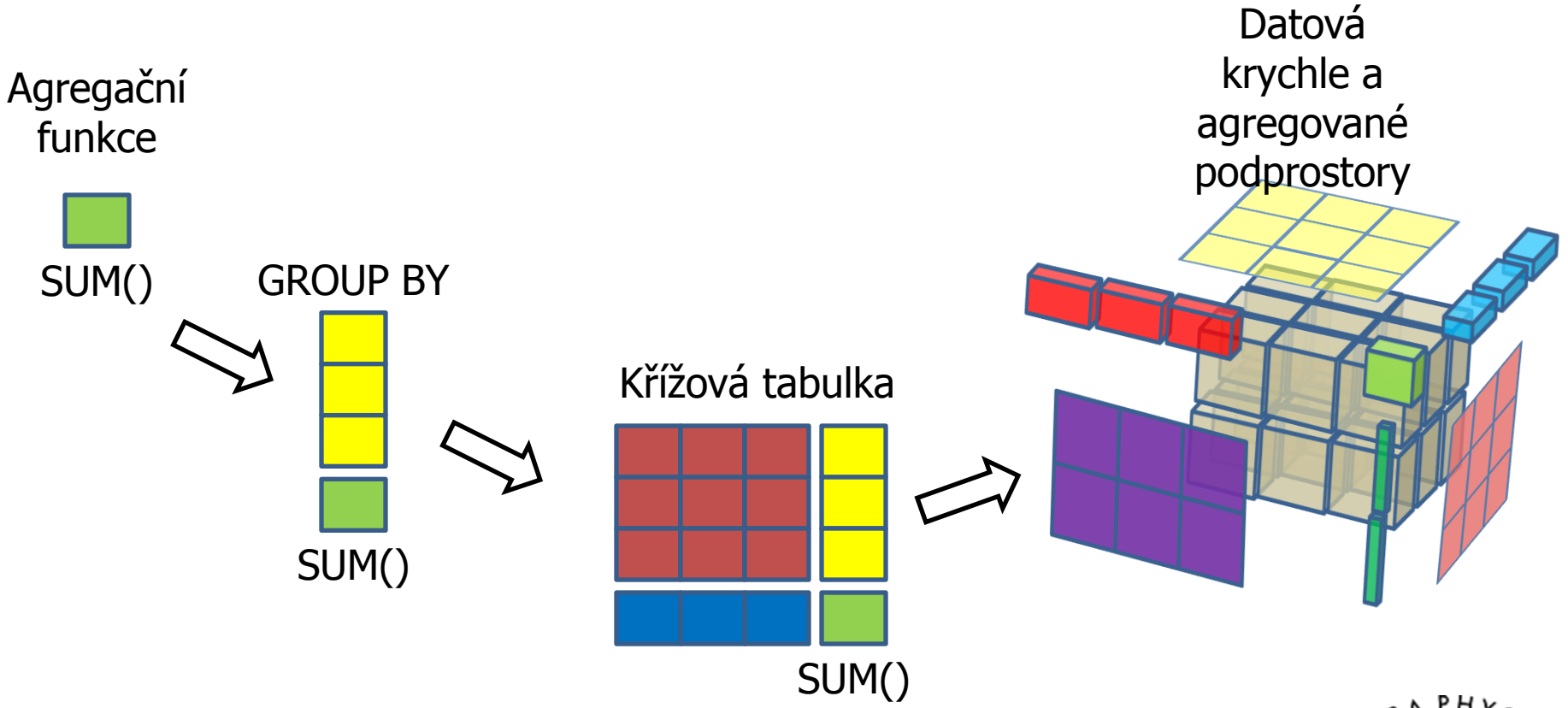
ROLLUP, DRILLDOWN

- Základní techniky pohybu pro více atributů (dimenzí) v agregacích
- ROLLUP
  - směr nahoru – zobecňování dotazu
- DRILLDOWN
  - směr dolu - upřesňovaní dotazu

Operátor CUBE

- Umožňuje použít agregační funkce pro více dimenzí současně
- Tzv. Data Cube – datová krychle
- Agregace je použita pro veškeré sloupce zahrnuté v GROUP BY

Datová krychle - ilustrace



Velikost datové kostky (1)

- Operátor CUBE agreguje přes všechny uvedené atributy.

- Pokud je atributů N, pak přibude 2^N - 1 super-agregovaných hodnot.

Velikost datové kostky (2)

- Bez znalostí dat nelze přesná velikost kostky určit – pouze horní odhad
- Pokud je kardinalita na N atributech C_1, C_2, … C_N, pak kardinalita výsledné kostky je $\Pi(C_i + 1)$.
- Tedy pokud máme například tabulku 1 x 3 x 2, potom operátor CUBE zobrazí 2 x 4 x 3 řádků – vždy se přičte jedna za hodnotu ALL.

Operátor CUBE - syntaxe

- GROUP BY <seznam sloupců> WITH CUBE
  - MS SQL Server do verze 2005

- GROUP BY CUBE(<seznam sloupců>)
  - Dle standardu SQL 1999
  - Oracle
  - IBM DB2
  - MS SQL Server 2008

Příklad použití CUBE

```
SELECT typ,
rok_vyroby,
barva ,
SUM(pocet) AS pocet
FROM auta
WHERE typ='OCTAVIA'
GROUP BY typ, rok_vyroby, barva
WITH CUBE
```

	typ	rok_vyroby	barva	pocet
1	ALL	2000	ALL	246
2	ALL	2000	bila	100
3	ALL	2000	stribrna	146
4	ALL	2005	ALL	435
5	ALL	2005	bila	135
6	ALL	2005	stribrna	300
7	ALL	2007	ALL	859
8	ALL	2007	bila	209
9	ALL	2007	stribrna	650
10	ALL	ALL	ALL	1540
11	ALL	ALL	bila	444
12	ALL	ALL	stribrna	1096
13	Octavia	2000	ALL	246
14	Octavia	2000	bila	100
15	Octavia	2000	stribrna	146
16	Octavia	2005	ALL	435
17	Octavia	2005	bila	135
18	Octavia	2005	stribrna	300
19	Octavia	2007	ALL	859
20	Octavia	2007	bila	209
21	Octavia	2007	stribrna	650
22	Octavia	ALL	ALL	1540
23	Octavia	ALL	bila	444
24	Octavia	ALL	stribrna	1096

GROUP BY vs. CUBE

```
SELECT typ, rok_vyroby,  barva,
SUM(pocet) as pocet
FROM auta
WHERE typ='OCTAVIA'
GROUP BY typ, barva, rok_vyroby
```

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	bila	100
2	Octavia	2005	bila	135
3	Octavia	2007	bila	209
4	Octavia	2000	stribrna	146
5	Octavia	2005	stribrna	300
6	Octavia	2007	stribrna	650

```
SELECT typ,
rok_vyroby,
barva,
SUM(pocet) as pocet
FROM auta
WHERE typ='OCTAVIA'
GROUP BY typ, barva, rok_vyroby
WITH CUBE
```

	typ	rok_vyroby	barva	pocet
1	ALL	2000	ALL	246
2	ALL	2000	bila	100
3	ALL	2000	stribrna	146
4	ALL	2005	ALL	435
5	ALL	2005	bila	135
6	ALL	2005	stribrna	300
7	ALL	2007	ALL	859
8	ALL	2007	bila	209
9	ALL	2007	stribrna	650
10	ALL	ALL	ALL	1540
11	ALL	ALL	bila	444
12	ALL	ALL	stribrna	1096
13	Octavia	2000	ALL	246
14	Octavia	2000	bila	100
15	Octavia	2000	stribrna	146
16	Octavia	2005	ALL	435
17	Octavia	2005	bila	135
18	Octavia	2005	stribrna	300
19	Octavia	2007	ALL	859
20	Octavia	2007	bila	209
21	Octavia	2007	stribrna	650
22	Octavia	ALL	ALL	1540
23	Octavia	ALL	bila	444
24	Octavia	ALL	stribrna	1096

Efektivita operátoru CUBE

- Příklad doby výpočtu při použití spojení dotazů pomocí UNION a při použití příkazu CUBE.
- Použit byl předchozí příklad. Testováno na MS SQL Serveru 2005.

Při použití UNION	**Při použití CUBE**
Doba výpočtu: 0,100272 s	Doba výpočtu: 0,0508809 s

⇒ dvakrát rychlejší !

Operátor ROLLUP

- Někdy není potřeba vytvářet celou datovou krychli
  - Taková data nepotřebujeme získat.
  - Zbytečně výpočetně náročné.
- Operátor přidává řádek „ALL“ ke všem sloupcům v seznamu za GROUP BY.
- Přidává hierarchicky – záleží na pořadí parametrů.

Operátor ROLLUP - syntaxe

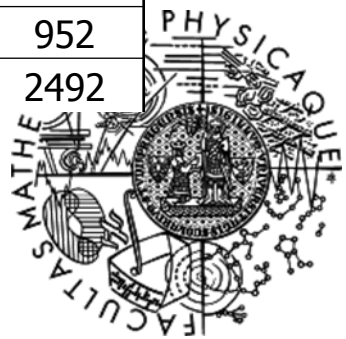
- GROUP BY <seznam sloupců> WITH ROLLUP
  - MS SQL Server do verze 2005

- GROUP BY ROLLUP(<seznam sloupců>)
  - Dle standardu SQL 1999
  - Oracle
  - IBM DB2
  - MS SQL Server 2008

Dotaz s příkazem ROLLUP

```
SELECT
    typ,
    rok_vyroby,
    barva,
    sum(pocet) AS pocet
FROM
    auta
GROUP BY
    typ,
    rok_vyroby,
    barva
WITH ROLLUP
```

	typ	rok_vyroby	barva	počet
1	Octavia	2000	bila	100
2	Octavia	2000	stribrna	146
3	Octavia	2000	ALL	246
4	Octavia	2005	bila	135
5	Octavia	2005	stribrna	300
6	Octavia	2005	ALL	435
7	Octavia	2007	bila	209
8	Octavia	2007	stribrna	650
9	Octavia	2007	ALL	859
10	Octavia	ALL	ALL	1540
11	Superb	2008	vinova	502
12	Superb	2008	ALL	502
13	Superb	2009	stribrna	450
14	Superb	2009	ALL	450
15	Superb	ALL	ALL	952
16	ALL	ALL	ALL	2492



Pořadí parametrů při použití ROLLUP

```
SELECT typ,
rok_vyroby,
barva,
SUM(pocet) AS pocet
FROM auta
WHERE typ='OCTAVIA'
GROUP BY  typ, rok_vyroby, barva
WITH ROLLUP
```

```
SELECT typ,
rok_vyroby,
barva,
SUM(pocet) AS pocet
FROM auta
WHERE typ='OCTAVIA'
GROUP BY typ, barva, rok_vyroby
WITH ROLLUP
```

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	bila	100
2	Octavia	2000	stribrna	146
3	Octavia	2000	ALL	246
4	Octavia	2005	bila	135
5	Octavia	2005	stribrna	300
6	Octavia	2005	ALL	435
7	Octavia	2007	bila	209
8	Octavia	2007	stribrna	650
9	Octavia	2007	ALL	859
10	Octavia	ALL	ALL	1540
11	ALL	ALL	ALL	1540

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	bila	100
2	Octavia	2005	bila	135
3	Octavia	2007	bila	209
4	Octavia	ALL	bila	444
5	Octavia	2000	stribrna	146
6	Octavia	2005	stribrna	300
7	Octavia	2007	stribrna	650
8	Octavia	ALL	stribrna	1096
9	Octavia	ALL	ALL	1540
10	ALL	ALL	ALL	1540

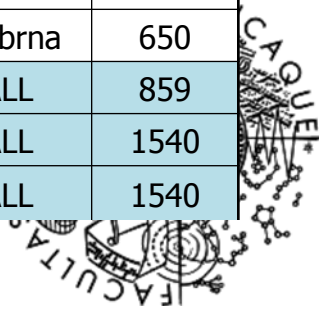
GROUP BY vs. ROLLUP

```
SELECT typ,
rok_vyroby,
barva,
SUM(pocet) AS pocet
FROM auta
WHERE typ='OCTAVIA'
GROUP BY typ, barva, rok_vyroby
```

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	bila	100
2	Octavia	2005	bila	135
3	Octavia	2007	bila	209
4	Octavia	2000	stribrna	146
5	Octavia	2005	stribrna	300
6	Octavia	2007	stribrna	650

```
SELECT typ,
rok_vyroby,
barva,
SUM(pocet) AS pocet
FROM auta
WHERE typ='OCTAVIA'
GROUP BY typ, barva, rok_vyroby
WITH ROLLUP
```

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	bila	100
2	Octavia	2000	stribrna	146
3	Octavia	2000	ALL	246
4	Octavia	2005	bila	135
5	Octavia	2005	stribrna	300
6	Octavia	2005	ALL	435
7	Octavia	2007	bila	209
8	Octavia	2007	stribrna	650
9	Octavia	2007	ALL	859
10	Octavia	ALL	ALL	1540
11	ALL	ALL	ALL	1540



GROUP BY, ROLLUP, CUBE

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	bila	100
2	Octavia	2005	bila	135
3	Octavia	2007	bila	209
4	Octavia	2000	stribrna	146
5	Octavia	2005	stribrna	300
6	Octavia	2007	stribrna	650

	typ	rok_vyroby	barva	pocet
1	Octavia	2000	bila	100
2	Octavia	2000	stribrna	146
3	Octavia	2000	ALL	246
4	Octavia	2005	bila	135
5	Octavia	2005	stribrna	300
6	Octavia	2005	ALL	435
7	Octavia	2007	bila	209
8	Octavia	2007	stribrna	650
9	Octavia	2007	ALL	859
10	Octavia	ALL	ALL	1540
11	ALL	ALL	ALL	1540

	typ	rok_vyroby	barva	pocet
1	ALL	2000	ALL	246
2	ALL	2000	bila	100
3	ALL	2000	stribrna	146
4	ALL	2005	ALL	435
5	ALL	2005	bila	135
6	ALL	2005	stribrna	300
7	ALL	2007	ALL	859
8	ALL	2007	bila	209
9	ALL	2007	stribrna	650
10	ALL	ALL	ALL	1540
11	ALL	ALL	bila	444
12	ALL	ALL	stribrna	1096
13	Octavia	2000	ALL	246
14	Octavia	2000	bila	100
15	Octavia	2000	stribrna	146
16	Octavia	2005	ALL	435
17	Octavia	2005	bila	135
18	Octavia	2005	stribrna	300
19	Octavia	2007	ALL	859
20	Octavia	2007	bila	209
21	Octavia	2007	stribrna	650
22	Octavia	ALL	ALL	1540
23	Octavia	ALL	bila	444
24	Octavia	ALL	stribrna	1096

ROLLUP vs. CUBE - přehled

ROLLUP

- Přidává nový řádek pro každý sloupec zahrnutý v sekci GROUP BY
- Záleží na pořadí jednotlivých parametrů v sekci GROUP BY

CUBE

- Vytvoří všechny možné kombinace pro sloupce zahrnuté v sekci GROUP BY
- Nezáleží na pořadí jednotlivých parametrů v sekci GROUP BY

Algebra GROUP BY, CUBE, ROLLUP

- CUBE(ROLLUP) = CUBE
- CUBE(GROUP BY) = CUBE
- ROLLUP(GROUP BY) = ROLLUP
- GROUP BY <sloupce>
  ROLLUP <sloupce>
  CUBE <sloupce>

Rozšíření syntaxe GROUP BY

- GROUP BY <sloupce>
  [ROLLUP <sloupce>]
  [CUBE <sloupce>]

<sloupce> ::=
{ (<sloupec> | <výraz>)
[AS <alias>]
[<splňující podmínka>]
}

Hodnota ALL

- Jak reprezentovat?
- Návrh: množina
  - Potřeba funkce ALL() vracející celou množinu hodnot
- Problémy:
  - Nové klíčové slovo
  - Možnost přidat ALL [NOT] ALLOWED do definice sloupců a systémového katalogu
  - Přetížení operátorů =, IN, …
  - a mnoho dalších …

Funkce GROUPING

- Operátor CUBE vrací v agregovaných sloupcích speciální hodnotu NULL místo ALL.
- Má jiný význam, než běžný NULL – jak rozlišit?
- Syntaxe funkce: GROUPING(jmeno_sloupce)
- Návratové hodnoty
  - 1pokud je hodnota NULL vytvořena agregací
  - 0 pokud je hodnota datová nebo se jedná
    o implicitní NULL

Funkce GROUPING – příklad použití

```
SELECT
CASE WHEN GROUPING(typ) = 1 THEN 'vše' ELSE typ END
AS typ,
CASE
WHEN GROUPING(rok_vyroby) = 1 THEN 'vše' ELSE rok_vyroby END
AS rok_vyroby,
CASE WHEN GROUPING(barva) = 1 THEN 'vše' ELSE barva END
AS barva,
SUM(pocet) AS pocet

FROM auta
WHERE typ='OCTAVIA'

GROUP BY typ, rok_vyroby, barva
WITH CUBE
```

Funkce GROUPING – výsledek

	typ	rok_vyroby	barva	pocet
1	vše	2000	vše	246
2	vše	2000	bila	100
3	vše	2000	stribrna	146
4	vše	2005	vše	435
5	vše	2005	bila	135
6	vše	2005	stribrna	300
7	vše	2007	vše	859
8	vše	2007	bila	209
9	vše	2007	stribrna	650
10	vše	vše	vše	1540
11	vše	vše	bila	444
12	vše	vše	stribrna	1096
13	Octavia	2000	vše	246
14	Octavia	2000	bila	100
15	Octavia	2000	stribrna	146
16	Octavia	2005	vše	435
17	Octavia	2005	bila	135
18	Octavia	2005	stribrna	300
19	Octavia	2007	vše	859
20	Octavia	2007	bila	209
21	Octavia	2007	stribrna	650
22	Octavia	vše	vše	1540
23	Octavia	vše	bila	444
24	Octavia	vše	stribrna	1096

CUBE a ROLLUP pod pokličkou

- ROLLUP
  - Setřídění podle sdružovaných atributů
  - Výpočet agregačních funkcí
- CUBE
  - Naivně: sjednocení ROLLUP výsledků

Techniky výpočtu agregačních funkcí

- Výpočet na co nejnižší systémové úrovni
- Použití polí nebo hašování k organizaci agregovaných sloupců v paměti
- Mapování dlouhých řetězců do menších typů
- Organizace velkých dat
  - Třídění nebo hybridní hašování
  - Sekvenční průchod
- Paralelní zpracování roztroušených dat

Rozdělení agregačních funkcí

- Distributivní
  - COUNT, MIN, MAX, SUM
- Algebraické
  - Průměr, směrodatná odchylka
- Celostní - holistické
  - Medián, modus

Udržování krychlí

- Potřeba dynamicky měnit hotovou krychli – udržovat
- Udržování kostky je jiné než její vytváření

- Př. Funkce MAX()
  - SELECT
    - Funkce je distributivní
  - INSERT
    - Projdu jen pár řádků
  - DELETE ☹
    - Musím projít všechno
    - MAX() je na operaci DELETE holistická

Udržování krychlí - výsledky

- Algebraické funkce na INSERT, UPDATE, DELETE
  - Lze udržovat jednoduše
- Distributivní funkce na INSERT, UPDATE, DELETE
  - Složitější, ale ne nemožné
- Holistické funkce
  - Zpravidla pouze na operaci DELETE
  - Velmi obtížné

ROLLUP, CUBE a podpora SŘBD

- Podporují
  - MS SQL Server
  - Oracle
  - DB2
- Nepodporují
  - Postgres SQL
  - MySQL

Závěr

- Operátory CUBE a ROLLUP
  - Umožňují pracovat s více agregacemi najednou
  - Zjednodušují zápis dotazů
  - Použití při dotazování nad velkými daty
  - Umožňuje vytvářet dotazy přes více dimenzí – GROUP BY umožňuje dotaz maximálně přes jednu dimenzi.

Použité zdroje

- http://paul.rutgers.edu/~aminabdu/cs541/cube_op.pdf
- http://technet.microsoft.com
- http://www.dba-oracle.com/
- http://chiragrdarji.wordpress.com/2008/09/09/group-by-cube-rollup-and-sql-server-2005/

… prostor pro dotazy …

Jaké jsou 2 hlavní rozdíly mezi operátory ROLLUP a CUBE
?

K čemu slouží funkce GROUPING ?

K čemu se používají techniky ROLLUP a DRILLDOWN ?