

Agregace v SQL

(dokončení)

Jde přece jen o jednoduchou problematiku, takže jedeme dál...

Naše předchozí dotazy do tabulky NAKUP budou optimální, pokud vytvoříme index podle sloupce CO, který hraje roli jak při restrikci, tak při agregaci a třídění:

```
CREATE INDEX NAKUP_CO ON NAKUP ( CO );
```

Tabulka NAKUP umožní demonstrovat i další pohledy na data pomocí agregace. Nabízí se například denní pohled na výdaje na potraviny umožněný příkazem:

```
SELECT DEN, SUM(PLACENO) UTRATIL
FROM NAKUP
GROUP BY DEN
ORDER BY DEN;
```

Tak vznikne z tabulky 8 názorná tabulka 12, sledující rozmařilost po dnech v týdnu. Škoda jen, že data v ní obsažená jsou setříděna podle abecedy, a ne podle pořadí dne v týdnu.

DEN	UTRATIL
PATEK	138
PONDELI	87
UTERY	60

Tabulka 12. Které dny se utrácí.

Tabulka 13 vznikla dotazem agregujícím podle sloupce JEDNOTKA.

```
SELECT SUM(KOLIK) MNOZSTVI, JEDNOTKA
FROM NAKUP
WHERE DEN="PATEK"
GROUP BY JEDNOTKA
ORDER BY JEDNOTKA;
```

MNOZSTVI	JEDNOTKA
2	kg
10	ks
3	l

Tabulka 13. Páteční nákupní taška.

V tomto konkrétním případě jen zdánlivě nedošlo k agregaci tří pátečních potravin podle jednotky množství, protože byl nákup málo rozmanitý. Pokud by nás zajímal obsah tašky při pondělním předzásobení na celý týden, stačí vnechat restrikci na pátek a následujícím dotazem vznikne tabulka 14:

```
SELECT SUM(KOLIK) MNOZSTVI, JEDNOTKA
FROM NAKUP
GROUP BY JEDNOTKA
ORDER BY JEDNOTKA;
```

MNOZSTVI	JEDNOTKA
4,6	kg
10	ks
5	l

Tabulka 14. Předzásobení na týden.

Z tabulky NAKUP snadno odvodíme jednotkové ceny zboží příkazem SQL bez agregace:

```
SELECT CO, DEN, PLACENO/KOLIK CENA,
JEDNOTKA FROM NAKUP
ORDER BY CO, DEN;
```

V tabulce 15 je pak dobrý přehled o týdenním kolísání cen.

CO	DEN	CENA	JEDNOTKA
ANANAS	PATEK	30	kg
ANANAS	PONDELI	30	kg
KURE	PONDELI	70	kg
KURE	UTERY	60	kg
MLEKO	PATEK	16	l
MLEKO	UTERY	15	l
VEJCE	PATEK	3	ks

Tabulka 15. Kolísání cen potravin.

Z ekonomického a statistického pohledu je rozumné mít přehled o minimální, průměrné a maximální jednotkové ceně pro každou potravinu. Tak vznikne velmi užitečná tabulka 16 pomocí dotazu:

```
SELECT CO, COUNT(*) PO CET_HODNOT,
MIN(PLACENO/KOLIK) MINIMALNI,
AVG(PLACENO/KOLIK) PRUMERNA,
MAX(PLACENO/KOLIK) MAXIMALNI
FROM NAKUP
GROUP BY CO
ORDER BY CO;
```

Co dobrého přinášejí následující příkazy „ze života“?

```
SELECT RODNE_CISLO, SUM(PRIJEM-VYDEJ)
FROM DENIK
GROUP BY RODNE_CISLO
ORDER BY RODNE_CISLO;
```

```
SELECT MESTO, AVG( HRUBA - DAN ), MAX
( HRUBA ) FROM VYPLATA
WHERE VYSKA < 130 AND MUZ
GROUP BY MESTO
ORDER BY MESTO;
```

```
SELECT PORAD, COUNT(*) FROM TELEVI
ZOR
GROUP BY PORAD
ORDER BY PORAD;
```

CO	POCET_HODNOT	MINIMALNI	PRUMERNA	MAXIMALNI
ANANAS	2	30	30	30
KURE	2	60	65	70
MLEKO	2	15	15.5	16
VEJCE	1	3	3	3

Tabulka 16. Statistické údaje o ceně potravin.

```
SELECT STANICE, COUNT(*) FROM TELEVI-
ZOR
GROUP BY STANICE
ORDER BY STANICE;
```

Agregace se složeným klíčem

Vraťme se k obrázkům 5 a 6 z minulého čísla. Chceme-li použít víc sloupců ta-

ZBOZI	MESTO	DEN	MNOZSTVI	CENA
RUM	PRAHA	PO	1000	100
MLEKO	PRAHA	UT	10000	20
UHERAK	BRNO	PO	50	200
UHERAK	PRAHA	ST	60	250
UHERAK	PRAHA	CT	500	110
MLEKO	BRNO	ST	2000	18
UHERAK	BRNO	PA	400	103

Tabulka 17. Jak nám to jde v tabulce OBCHOD.

bulky jako klíč ke shlukování řádků do agregátů, stačí uvést za klíčové slovo GROUP BY názvy příslušných sloupců oddělené čárkou. Nejde o nic jiného než o popis složeného klíče pro agregaci. Představme si obchodní firmu, která sleduje prodej zboží v jednotlivých městech. Malá ukáзка dat je v tabulce 17.

Pokud nás zajímá pohled na prodej jednotlivého zboží podle měst, stačí napsat příkaz:

```
SELECT ZBOZI, MESTO, SUM(MNOZSTVI)
TOTAL FROM OBCHOD
GROUP BY ZBOZI, MESTO
ORDER BY ZBOZI, MESTO;
```

Tak vznikne tabulka 18.

V tabulce 19 vidíme výsledek obdobného dotazu, ale s jiným pořadím položek:

```
SELECT MESTO, ZBOZI, SUM(MNOZSTVI)
TOTAL FROM OBCHOD
GROUP BY ZBOZI, MESTO
ORDER BY MESTO, ZBOZI;
```

Je vidět rozdíl mezi agregací a tříděním. V obou případech vznikly stejné agregáty, ale mají pokaždé jiné pořadí v odpovědi na SQL dotaz. Agregace provádí spojování informací a sčítání množství, zatímco třídění pouze usnadňuje pohled na výsledek pod požadovaným úhlem. Dodavatel čehokoli do Brna potřebuje tabulku 19, zatímco výrobce uheráku nepohrdne tabulkou 18. Mnohem pragmatičtější je ekonomický po-

Copak asi znamená a komu by se hodil dotaz:

```
SELECT MESTO, DEN, SUM (MNOZST-
VI*CENA) PRIJEM
FROM OBCHOD
GROUP BY ZBOZI, DEN
ORDER BY ZBOZI, DEN;
```

Nejen obchodníci potřebují agregaci. Určitě i zpracování výsledků sportovního víceboje jednotlivců a družstev je

ZBOZI	MESTO	TOTAL
MLEKO	BRNO	2000
MLEKO	PRAHA	10000
RUM	PRAHA	1000
UHERAK	BRNO	450
UHERAK	PRAHA	560

Tabulka 18. Zboží podle měst.

krásnou zámkou k agregaci. Bude vhodné si představit tabulku VICEBOJ se sloupci DRUZSTVO, CLEN, DISCIPLINA, DOPING a BODY. První tři sloupce jsou jasné a obsahují názvy, zkratky nebo čísla družstev, sportovců a disciplín víceboje. Trojice DRUZSTVO, CLEN, DISCIPLINA je zároveň unikátním klíčem pro přímý přístup, který zároveň zabraňuje podvádění s vícenásobným vkládá-

MESTO	ZBOZI	TOTAL
BRNO	MLEKO	2000
BRNO	UHERAK	450
PRAHA	MLEKO	10000
PRAHA	RUM	1000
PRAHA	UHERAK	560

Tabulka 19. Zboží podle měst.

ním dat. Sloupec DOPING obsahuje logické hodnoty YES a NO jako výsledky dopingových zkoušek po jednotlivých disciplínách. Sloupec BODY je určen pro ohodnocení výsledku jednotlivce v dané disciplíně. Průběžné i celkové bodové hodnocení družstev je snadno zjistitelné dotazem:

```
SELECT DRUZSTVO, SUM(BODY) CELKEM
INTO VYSLEDOVKA
FROM VICEBOJ
GROUP BY DRUZSTVO
ORDER BY DRUZSTVO;
```

JAROMÍR KUKAL

