Databáze standardu SQL, díl 12.

Kdepak ty dotaze hnízdo máš?

Po delším rozjímání o tématu agregace se plnou parou vrhneme do problematiky zahnízděných dotazů.

Agregace se složeným klíčem

Jistě vás napadlo, že ne všechny skupiny vzniklé agregací jsou hodné naší pozornosti. Nic složitého není naplánovat ještě druhou restrikci po agregaci. Stačí použít klíčové slovo HAVING a za ním logický výraz. Je-li jeho hodnota pro celou skupinu rovna YES, dojde k zobrazení skupiny. Je-li hodnota NO nebo NULL, pak se skupina nepromítne do odpovědi na dotaz. Pokud je dopování pro komisi důvodem k úplné diskvalifikaci jednotlivce, stačí napsat následující dotaz:

```
SELECT JMENO, DRUZSTVO, SUM(BODY) CELKEM
INTO JEDNOTLIVEC
FROM VICEBOJ
GROUP BY JMENO, DRUZSTVO
HAVING MAX(DOPING)<YES OR MAX(DOPING) IS NULL
ORDER BY JMENO, DRUZSTVO;
```

Velmi přísná komise se nepáře ani s družstvem:

```
SELECT DRUZSTVO, SUM(BODY) CELKEM
INTO VYSLEDOVKA
FROM VICEBOJ
GROUP BY DRUZSTVO
HAVING MAX(DOPING)<YES OR MAX(DOPING) IS NULL
ORDER BY JMENO, DRUZSTVO;
```

Úspěšný prodejce uheráku, který je za vodou, myslí takto:

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

Nyní zbývá doplnit pravidlo pro obecný popis agregace o výběr vhodných skupin pomocí druhé restrikce:

```
SELECT popis projekce FROM tabulka
WHERE podmínka restrikce řádků
GROUP BY klíč agregace
HAVING podmínka restrikce skupiny
ORDER BY klíč třídění;
```

Pro pochopení rozdílu mezi výběrem vhodných řádků před agregací do skupin a mezi výběrem skupin po agregaci je dobré si uvědomit pořadí kroků, které provádí SQL server po obdržení dotazu.

```
TABULKA => RESTRIKCE ŘÁDKŮ => AGREGACE =>
RESTRIKCE SKUPIN => TŘÍDĚNÍ SKUPIN => PROJEKCE
```

Tak dochází k poklesu počtu řádků ve třech fázích, z nichž každá má svůj nezávislý hluboký význam pro elegantní vyřešení daného problému. Malá hádanka je ukryta v dotazu:

```
SELECT ONA, COUNT(ON) KOLIK
FROM KDOSKYM
GROUP BY ONA
HAVING COUNT(ON) >= 100
ORDER BY ONA;
```

Zahnízděné dotazy v SQL

Dnes si poprvé užijeme strukturovanost jazyka SQL. Structured Query Language z předchozích dílů známe zatím jenom jako dotazovací jazyk a strukturovanost jako kladná vlastnost konečně přichází na řadu. Nejlepším příkladem strukturovanosti je malá Švejkova epizoda z prvního dílu, kdy byl na pozorování v ústavu a jiný blázen mu vypravoval, že uvnitř zeměkoule je ještě jedna, která je mnohem větší. Později prof. Parkinson publikoval, že uvnitř každého malého problému je velký problém, který usiluje se dostat ven. Proč by tedy strukturovaný SQL dotaz nemohl obsahovat uvnitř jiný SQL dotaz, který je jeho součástí a zároveň ho předběhne. Pokud by v tabulce SAMEC, kterou si vede klub chovatelů angorských králíků, byl též sloupec POCET_POTOMKU, stálo by za to vypsat seznam nejplodnějších samců seřazený podle věku. Takový seznam podle okolností obsahuje jednoho nebo několik samců, které poznáme podle toho, že mají největší možný počet potomků. Dosavadní znalosti SQL nás mohou dovést k improvizovanému řešení, kdy nejprve agregačním dotazem zjistíme maximální počet potomků, a ten si opíšeme do notýsku. Pak sestavíme druhý dotaz pro zobrazení těch samců, jejichž počet potomků je přesně roven číslu z notýsku. Tomu odpovídají dva dotazy, z nichž druhý je poněkud pochybný:

```
SELECT MAX(POCET_POTOMKU) X
FROM SAMEC;
SELECT ID, JMENO, VEK FROM SAMEC WHERE POCET_POTOMKU= 73
ORDER BY VEK, JMENO, ID;
```

Chtělo by to mít možnost oba dotazy spojit do jednoho, abychom mohli zahodit notes s magickým číslem 73 na první stránce.

Dotaz a poddotaz

Může-li mít šéf jednoho nebo několik podřízených, pak i hlavnímu dotazu může být podřízeno několik poddotazů. Pro jednoduchost uvažujme dotaz typu SELECT, uvnitř kterého se vyskytuje podřízený poddotaz také typu SELECT, který je pověřen vykonáním podřadných prací předem. Aby si poddotaz moc nevyskakoval na hlavní dotaz, musíme ho

umístit do kulatých závorek. Vypadá to navenek prazvláštně, ale při dodržení pravidel to funguje:

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

Nejprve je jednou proveden vnitřní poddotaz v závorce a výsledek poddotazu je použit ku prospěchu hlavního dotazu. Použijeme-li místo teček konkrétní vyjádření podmínek projekce, agregace a restrikce, dostaneme lepší řešení králičího problému jedním dotazem:

```
SELECT ID, JMENO, VEK FROM SAMEC WHERE POCET_POTOMKU=(SELECT MAX(POCET_POTOMKU) FROM SAMEC) ORDER BY VEK, JMENO, ID;
```

Pojem zahnízděný dotaz neboli NESTED QUERY, který se běžně používá, odpovídá představě, že poddotaz si udělal uvnitř dotazu hnízdo, ve kterém bydlí mezi závorkami. Pokud má šéf dva podřízené, jsou si většinou rovni. Tomu odpovídá konstrukce dotazu se dvěma poddotazy:

```
SELECT ... (SELECT ...1...)...(SELECT ...2...)...;
```

Nejprve je jednou proveden první vnitřní poddotaz v závorce, potom druhý vnitřní poddotaz a výsledky obou poddotazů jsou použity v hlavním dotazu. Sháníme-li na smetanu mladého vypaseného králíka, stačí hledat mezi králíky, jejichž věk je menší než průměrný a hmotnost větší než průměrná:

```
SELECT ID, JMENO, VEK, HMOTNOST FROM SAMEC WHERE VEK< (SELECT AVG(VEK) FROM SAMEC) AND HMOTNOST>(SELECT AVG(HMOTNOST) FROM SAMEC)ORDER BY HMOTNOST DESC, VEK, JMENO, ID;
```

Pokud bychom měli sudý počet králíků – polovina z nich by byli hubení mladíci a druhá polovina by byli starší cvalíci – pak nebude zobrazen ani jeden adept a my budeme mít vegetariánský den. Někteří šéfové vlastnící dva podřízené jsou šťastnější, když jeden podřízený otravuje druhého. Podobně může vnitřní poddotaz používat jako svého pomocníka poddotaz o patro níž podle schématu:

```
SELECT ...(SELECT ...1...(SELECT ...2...)...)...;
```

Nejprve je jednou proveden druhý nejvnitřnější poddotaz v závorce, potom první vnitřní poddotaz nad ním a výsledky prvního poddotazu jsou použity v hlavním dotazu, aniž by ten tušil, že na tom pracoval i druhý poddotaz. Autor seriálu není vegetarián a není tak mladý, aby vůbec ukousl starého králíka. Proto budu hledat mezi mladými ty určené pomocí dotazu:

```
SELECT ID, JMENO, VEK, HMOTNOST FROM SAMEC WHERE VEK< (SELECT AVG(VEK) FROM SAMEC) AND HMOTNOST>(SELECT AVG(HMOTNOST) FROM SAMEC WHERE VEK< (SELECT AVG(VEK) FROM SAMEC)) ORDER BY HMOTNOST DESC, VEK, JMENO, ID;
```

To odpovídá ještě složitějšímu zahnízdění:

```
SELECT...(SELECT...1...)...(SELECT...2...(SELECT...3...)...)...;
```

Rekurzivní definice strukturovaného dotazu potom zní:
Strukturovaný neboli zahnízděný dotaz obsahuje uvnitř konečný počet poddotazů uzavřených v nepřekrývajících se závorkách. Poddotaz je také jenom dotazem, a proto pro něj platí předchozí tvrzení.
Nabízí se otázka, zda při používání poddotazů platí ještě další omezení. Odpověď je bohužel kladná. Z požadavku aktivní spolupráce mezi dotazem a poddotazem nutně plyne, že nestačí pouze dříve formulovat poddotaz, ale že výsledek poddotazu musí být pro další použití zdárně předán do hlavního dotazu. Nezbývá než se v jednotlivých případech zorientovat.

Tabulka 1 x 1

Nejméně starostí nám přidělá poddotaz, jehož výsledkem je tabulka obsahující právě jeden sloupec a nejvýše jeden řádek. Taková tabulka může vzniknout několika způsoby. Všechny mají společné použití právě jednoho výrazu při projekci. U prvního způsobu použijeme při popisu projekce jednu z agregačních funkcí a nebudeme definovat grupy. Takto se dají řešit nejen králičí úlohy. Následují příklady samostatných poddotazů prvního typu:

```
(SELECT COUNT( HRUBA) FROM CLOVEK WHERE MESIC=3)

(SELECT MIN(MNOZSTVI) FROM SKLAD WHERE JEDNOTKA="kg")
(SELECT MAX(ZNAMKA) FROM VYSVEDCENI WHERE PREDMET="CJ" AND ROCNIK=6)
(SELECT AVG(IQ) FROM CLOVEK WHERE BOSS AND PSC LIKE "4%")
```

Poddotazy druhého typu vycházejí z požadavku 5NF, a tím ze zaručené entitní integrity tabulek. Pak může jeden řádek jednosloupcové tabulky vzniknout porovnáním unikátního klíče s konstantou:

```
(SELECT VYSKA FROM CLOVEK WHERE RC="5403273249")
(SELECT HRUBA-DAN FROM VYPLATA WHERE CPRAC=251 AND MESIC=12)
(SELECT ZNAMKA FROM VYSVEDCENI WHERE PREDMET="CJ" AND ROCNIK=6 AND RC="5403273249")
```

Hazardéři mají rádi poddotazy třetího typu, kde za WHERE je libovolný logický výraz. Pak se jen modlí, aby byl splněn jen v jednom případě. Odstrašujícím případem může být poddotaz založený na víře, že MICHEO je málo časté jméno:

```
(SELECT VAHA+100-VYSKA FROM CLOVEK WHERE JMENO="MICHEO")
```

Když už poddotazem vznikne tabulka 1 x 1, je třeba si uvědomit, že je možné ji použít místo jednotlivé hodnoty v libovolném výrazu. Nemá-li tabulka ani jeden řádek, pak je hodnota tabulky ve výrazu rovna NULL. Má-li tabulka více sloupců nebo řádků, nesmí být používána ve výrazech jako hodnota. Málokdy najdeme pro zahnízděnou tabulku 1 x 1 uplatnění ve výrazech pro projekci. Většinou je nalezneme při popisu restrikce řádků nebo restrikce skupin po agregaci. Neuškodí několik příkladů. Máme-li nalézt nejchudší zlatníky v okrese Benešov, stačí napsat:

```
SELECT RC, JMENO FROM CLOVEK WHERE PROFESE="ZLATNIK" AND OKRES="BN" AND PRACHY=(SELECT PRACHY FROM CLOVEK WHERE PROFESE="ZLATNIK" AND OKRES="BN") ORDER BY JMENO, RC;
```

JAROMÍR KUKAL