**Poddotazy (Subqueries)**

Analýzy často vykonávame vo vrstvách, pričom každá vrstva analýzy využíva predchádzajúcu vrstvu alebo na nej nadväzuje. Táto prax vkladania vrstiev do analytických procesov je v skutočnosti veľmi bežná. Napríklad, keď vytvoríme dotaz pomocou iného dotazu ako zdroja údajov, rozvrstvíme analýzu. Keď vytvoríme dotaz na základe dočasnej tabuľky vytvorenej vytváracím dotazom, rozvrstvíme analýzu. Všetky tieto konvenčné metódy vrstvenia majú dve veci spoločné:

* Všetky pridávajú krok k našim analytickým procesom. Každý dotaz, ktorý musí byť spustený, aby sa splnil ďalší dotaz, alebo každá dočasná tabuľka, ktorá sa musí vytvoriť, aby pokročila naša analýza, pridáva ďalšiu úlohu, ktorú je potrebné dokončiť predtým, ako získame konečné výsledky.
* Všetky vyžadujú vytvorenie dočasných tabuliek alebo dočasných dotazov, zaplavia našu databázu tabuľkami a ďalšími dotazmi, ktoré vedú k mätúcemu analytickému procesu, ako aj databázu, ktorá sa ľahko nafúkne. Tu môžu pomôcť poddotazy.

Tvorba poddotazov si vyžaduje aspoň základné znalosti SQL.

***Vylepšenie analýz pomocou poddotazov***

Poddotazy sú výberové dotazy, ktoré sú vnorené do iných dotazov. Primárnym účelom poddotazu je umožniť nám použiť výsledky jedného dotazu v rámci vykonávania iného dotazu. Pomocou poddotazov môžeme odpovedať na otázku s viacerými časťami, špecifikovať kritériá pre ďalší výber alebo definovať nové polia, ktoré sa majú použiť v analýze. Na nasledovnom obrázku je jednoduchý príklad použitia poddotazu:



V SQL zobrazení to vyzerá nasledovne:



Myšlienka poddotazu spočíva v tom, že poddotaz sa vykoná ako prvý a výsledky sa použijú vo vonkajšom dotaze (dotaz, v ktorom je poddotaz vložený) ako kritérium, výraz, parameter atď. V príklade poddotaz najprv vráti zoznam pobočiek, ktoré patria na kalifornský trh. Potom vonkajší dotaz použije tento zoznam ako kritérium na odfiltrovanie všetkých zamestnancov, ktorí nepatria na kalifornský trh.

***Prečo používať poddotazy***

Poddotazy často bežia pomalšie ako štandardné dotazy pomocou spojenia. Je to preto, že poddotazy sa buď vykonávajú proti celej množine údajov, alebo sa vyhodnocujú viackrát, raz na každý riadok spracovaný vonkajším dotazom. Vďaka tomu sa spomalia, najmä ak máte veľký súbor údajov. Prečo ich teda používať?

Mnohé analýzy vyžadujú viackrokové procesy, ktoré používajú dočasné tabuľky alebo prechodné dotazy. Hoci na dočasných tabuľkách a dotazoch nie je vo svojej podstate nič zlé, ich nadmerné množstvo v analytických procesoch môže viesť k mätúcemu analytickému procesu, ako aj k databáze, ktorá sa ľahko nafúkne.

***Základné pravidlá poddotazov***

* Poddotaz musí mať vo svojom reťazci SQL minimálne príkaz SELECT a klauzulu FROM.
* Poddotaz musí byť uzatvorený do zátvoriek.
* Teoreticky môžeme do dotazu vnoriť až 31 poddotazov. Toto číslo je však založené na dostupnej pamäti systému a zložitosti poddotazov.
* Poddotaz môžeme použiť ako výraz, pokiaľ vráti jednu hodnotu.
* Klauzulu ORDER BY môžeme použiť v poddotaze iba vtedy, ak je poddotazom príkaz SELECT TOP alebo SELECT TOP PERCENT.
* Kľúčové slovo DISTINCT nemôžeme použiť v poddotaze, ktorý obsahuje klauzulu GROUP BY.
* Aliasy tabuliek musíte implementovať do dotazov, v ktorých sa tabuľka používa vo vonkajšom dotaze aj v poddotaze.

***Vytváranie poddotazov bez zadávania príkazov SQL***

Predstavme si napríklad, že sme boli požiadaní, aby sme uviedli počet account manažérov, ktorí majú čas v službe dlhší ako priemerný čas v službe pre všetkých account manažérov. Znie to ako relatívne jednoduchá analýza a je jednoduchá, keď použijeme poddotaz. Ale kde začať? Môžeme jednoducho napísať príkaz SQL do zobrazenia SQL dotazu a spustiť ho. Pravdou však je, že len málo používateľov Accessu vytvára príkazy SQL úplne od začiatku. Inteligentní využívajú vstavané funkcie Accessu na šetrenie času. Trik je rozdeliť analýzu na zvládnuteľné časti:

1. Nájdeme priemerný čas v službe pre account manažérov vytvorením dotazu podľa obrázka.



1. Zmeňte zobrazenie na SQL zobrazenie a skopírujte SQL výraz.

SELECT Avg(DateDiff("m",[HireDate],Date())) AS Avg\_TIS\_in\_Months

FROM Dim\_AccountManagers;

1. Vytvorte dotaz, ktorý bude zisťovať počet account manažérov podľa času v službe.



1. Kliknite pravým tlačidlom myši do riadku Kritériá v poli TIS\_in\_Months a vyberte nástroj Lupa. Lupa pomáha pri komfortnejšom písaní dlhých výrazov.
2. Prilepte do tohto dialógového okna SQL výraz z predošlého dotazu.



1. Dokončite dotaz pridaním znamienka > pred poddotaz a zmeňte výraz v riadku Celkom výraz GROUP BY na WHERE. Spustením dotazu zistíme, že je 12 manažérov, ktorý majú čas v službe väčší ako priemer spoločnosti.



SQL výraz by mal byť nasledovný:

SELECT Count(Dim\_AccountManagers.AccountManagerID) AS MyCount

FROM Dim\_AccountManagers

WHERE (((DateDiff("m",[HireDate],Date()))

>(SELECT Avg(DateDiff("m",[HireDate],Date())) AS Avg\_TIS\_in\_Months

FROM Dim\_AccountManagers;)));

**Používanie IN a NOT IN s poddotazmi**

Operátory IN a NOT IN nám umožňujú spustiť dva dotazy v jednom. Myšlienka je taká, že najprv sa vykoná poddotaz a potom výsledný súbor údajov použije vonkajší dotaz na filtrovanie konečného výstupu.

Príklad znázornený na obrázku najprv spustí poddotaz, ktorý vyberie všetkých zákazníkov so sídlom v Kalifornii (CA). Vonkajší dotaz potom použije výslednú množinu údajov ako kritérium na vrátenie súčtu LineTotal len pre tých zákazníkov, ktorí zodpovedajú číslam zákazníkov vráteným v poddotaze.



***Používanie poddotazov s operátormi porovnávania***

Ako už názov napovedá, operátor porovnávania (=, <, >, <=, >=, <>) porovnáva dve položky
a vracia hodnotu True alebo False. Keď použijeme poddotaz s operátorom porovnania, požiadame Access, aby porovnal výslednú množinu údajov nášho vonkajšieho dotazu
s poddotazom. Ak chceme napríklad vrátiť všetkých zákazníkov, ktorí majú nákupy vyššie ako je priemerný nákup pre všetkých zákazníkov, môžeme použiť dotaz zobrazený na obrázku.



Ako prvý sa spustí čiastkový dopyt, ktorý nám poskytne priemerný nákup pre všetkých zákazníkov. Toto je hodnota, ktorú potom Access použije na porovnanie výslednej množiny údajov vonkajšieho dotazu. Inými slovami, maximálny nákup pre každého zákazníka sa porovnáva s priemerom spoločnosti. Ak je maximálny nákup zákazníka vyšší ako priemer spoločnosti, je zahrnutý do konečného výstupu; v opačnom prípade je to vylúčené.

***Použitie poddotazov ako výrazov***

V každom doterajšom príklade sme použili poddotazy v spojení s klauzulou WHERE, čím sme efektívne použili výsledky poddotazu ako kritériá pre vonkajší dotaz. Ako výraz však môžeme použiť aj poddotaz, pokiaľ poddotaz vráti jednu hodnotu. Dotaz zobrazený na obrázku ukazuje, ako môžeme použiť poddotaz ako výraz vo výpočte.



Tento príklad používa poddotaz na získanie priemerných jednotiek predaných za celú spoločnosť; tento poddotaz vráti jednu hodnotu. Túto hodnotu potom použije vo výpočte na určenie rozdielu medzi priemernými predanými jednotkami každého trhu a priemerom pre spoločnosť. Výsledné zobrazenie:



**Použitie korelovaných poddotazov**

Korelovaný dotaz je v podstate poddotaz, ktorý odkazuje späť na stĺpec, ktorý je vo vonkajšom dotaze. Korelované poddotazy sú jedinečné v tom, že zatiaľ čo štandardné poddotazy sa na získanie výsledku vyhodnocujú raz, korelovaný poddotaz sa musí vyhodnotiť viackrát pre každý riadok spracovaný vonkajším dotazom. Na ilustráciu skúste nasledujúce dva príkazy SQL.

*Nekorelované poddotazy*

Nasledujúci príkaz SQL používa nekorelovaný poddotaz. Poddotaz neodkazuje na žiadny stĺpec vo vonkajšom dotaze. Tento čiastkový dopyt bude vyhodnotený raz, aby sme získali priemerné výnosy pre celú množinu údajov.

SELECT MainSummary.Branch\_Number,

(SELECT Avg(Revenue)FROM MainSummary)

FROM MainSummary

*Korelované poddotazy*

Tento nasledujúci príkaz SQL používa korelovaný poddotaz. Poddotaz siaha späť do vonkajšieho dotazu a odkazuje na stĺpec Branch\_Number, čím efektívne núti vyhodnotiť poddotaz pre každý riadok, ktorý je spracovaný vonkajším dotazom. Konečným výsledkom tohto dotazu bude súbor údajov, ktorý zobrazuje priemerné príjmy pre každú pobočku v spoločnosti. Obrázok ukazuje, ako nasledujúci príkaz SQL vyzerá v návrhovom zobrazení.

SELECT MainSummary.Branch\_Number,

(SELECT Avg(Revenue)FROM MainSummary AS M2

WHERE M2.Branch\_Number = MainSummary.Branch\_Number) AS AvgByBranch

FROM MainSummary

GROUP BY MainSummary.Branch\_Number

