

SQL - optimalizace dotazů

Existují tři hlediska:

- 1. optimalizace místa na pevném disku**
- 2. optimalizace rychlosti získání informací**
(zkrácení času nutného pro čtení z DB)
- 3. optimalizace rychlosti aktualizace**
(zkrácení času nutného pro aktualizaci dat v DB)

1. optimalizace místa na pevném disku

- odstranit indexy
(tím se ovšem zpomalí provádění dotazů)
- zamezit opakovaným hodnotám ve sloupcích pomocí velkého množství samostatných číselníků

2. optimalizace rychlosti získání informací (zkrácení času nutného pro čtení z DB)

- vytvořit indexy pro všechny atributy, přes které je potřeba spojovat tabulky, třídit nebo omezovat pomocí podmínek
- omezit potřebu spojovat tabulky tím, že jich bude v databázi co nejmenší počet
- ponechat v tabulkách i duplicitní informace, aby nebylo potřeba tabulku spojovat s jinými
(POZOR – problémová aktualizace!)

3. optimalizace rychlosti aktualizace (zkrácení času nutného pro aktualizaci dat v DB)

- omezit používání indexů, resp. před aktualizací je zrušit a po ní znovu vytvořit
- omezit duplicity v datech na minimum, aby se data aktualizovala jen na jednom místě
- aktualizace tabulek s menším počtem řádků a sloupců probíhá rychleji => snaha vytvářet větší množství jednodušších tabulek

Optimalizace zpracování SQL příkazů

Obecná pravidla zpracování SQL příkazů (zejména dotazu **SELECT**):

Přednost restrikce před spojováním tabulek

restrikce se provádí dříve než spojování tabulek (tím se zabrání zpracování řádků, které budou později stejně vynechány). Mezi podmínkami se navíc nejdříve vyhodnocují ty nejsilnější (např. = má přednost před <)

Optimalizace spojování tabulek

je výhodnější k řádkům větší tabulky dohledávat řádky z menší tabulky. Pokud navíc existují v obou tabulkách indexy, použije se tzv. *slévání*, kdy se zároveň procházejí oba indexy a vyhledávají se v nich shodné hodnoty.

Využití existujících indexů

indexy se používají všude, kde je to možné (restrikce, třídění)

Zjištění postupu zpracování příkazu

ve většině databázových systému existuje příkaz **EXPLAIN PLAN FOR** *<příkaz>*, který zobrazí postup, kterým bude příkaz vykonán

Ovlivnění zpracování SQL příkazu

u velmi rozsáhlých databází má smysl zakazovat nebo vynucovat používání indexů v jednotlivých příkazech – některá databázová prostředí proto rozšiřují syntaxi SQL příkazů o značky, kterými lze určit požadovaný postup při zpracování SQL příkazu