

# Optimalizace 2

přednáška č. 6

# Co vzít v potaz

- Špatně definované indexy a také nedostatek vhodných indexů jsou hlavním zdrojem výkonostních problémů v DB.
- Navrhování indexů je takovým balancováním mezi rychlostí dotazů a cenou úprav dat v db.
- „Úzké“ indexy (index definovaný pro jeden či malý počet atributů) mají menší nároky na místo a menší náklady na udržování.
- „Široké“ indexy budou zase na druhou stranu vhodné pro více dotazů.
- Vzhledem k možnosti definovat indexy bez nutnosti modifikovat db schéma či aplikaci netřeba se bát experimentování.
- Jednou z možností je důvěřovat optimalizátoru dotazů.

## Jak postupovat při návrhu indexů II.

- Posoudit možné výhody využití dostupných přepínačů pro indexy.
- Využít raději nonclustered index nebo clustered index?
  - Nonclustered index je možné uložit ve stejné soboru jako vlastní tabulka nebo i jinde.
  - Jaký to má vliv na I/O operace?
  - Index uložený v jiném souboru může zlepšit výkon, protože lze číst současně jak tabulku tak i index

# Kdy použít indexy?

- Zkušení správci databází můžou navrhnout řadu indexů, tak aby se zlepšila průchodnost DB (počet dotazů zpracovaných DB za jednotku času).
- Nicméně je to vždy velice komplexní, časově náročný a na chyby bohatý proces a to i pro středně složité databáze!!!
- Pochopení principů, charakteristik a specifik databáze je nutná podmínka pro navržení optimálních indexů

# Co je třeba vzít v potaz

- Na úrovni databáze (tedy při návrhu tabulek)?
- Na úrovni dotazů?
- Na úrovni jednotlivých atributů?

## Na úrovni databáze

- Indexování malých tabulek nemusí být optimální, protože optimalizátor může strávit více času průchodem indexů, než pokud by jednoduše sekvenčně prošel tabulky.

# Na úrovni databáze (pokr.)

- Velký počet indexů definovaných pro tabulku ovlivňuje průchodnost operací INSERT, UPDATE a DELETE, protože všechny indexy musejí být upravovány vzhledem k provedeným změnám
  - Vyhněte se přeindexování u tabulek s velkým počtem úprav. Dále se snažte, aby indexy nebyly definovány na mnoha attributech
  - Větší počet indexů zlepšuje průchodnost u tabulek s nízkým počtem úprav obsahující mnoho dat.

# Na úrovni databáze (pokr.)

- Indexy definované pro pohledy obsahující agregace, spojení tabulek nebo jejich kombinace mají za následek zlepšení výkonu.



# Na úrovni dotazů

- Vytvořit nonclustered indexy pro atributy, které se často vyskytují v predikátech a v podmínkách při spojování tabulek
  - **Pozor:** Moc indexů vede k problémům s efektivitou uložení a také správou indexů

# Na úrovni dotazů (pokr.)

- Indexy definované pro více atributů mohou zlepšit výkon, protože všechna data nutná pro vyhodnocení dotazu jsou uloženy v indexu.
- Např. máme dotaz na atributy a,b a také máme index na attributech a,b,c.

# Na úrovni dotazů (pokr.)

- Psát dotazy tak, aby vkládání a modifikace atributů byla řešena v rámci jednoho dotazu a ne pomocí více dotazů pro změnu stejného atributu.
- Zjistit typy dotazů a atributů vyskytujících se v dotazu. Atributy využívané v dotazech s podmínkou na rovnost jsou dobrým kandidátem na nonclustered nebo clustered index.

# Na úrovni atributů

- Clustered indexy jsou vhodné pro unique či not null atributy.
- Prostudovat unikátnost dat v atributu. Unikátní index dá optimalizátoru další informaci pro zlepšení průchodnosti.

# Na úrovni atributů (pokr.)

- Distribuci dat v rámci atributu.
- Časté a dlouho trvající dotazy mohou být způsobeny indexováním atributů s několika různými hodnotami, případně spojováním pomocí takových atributů.
- Například, máme-li telefonní seznam pro osoby a v rámci města se všichni jmenují Newman nebo Nowak.

# Na úrovni atributů (pokr.)

- Také si rozmyslete pořadí atributů v indexu.
- Atributy, které jsou používány v podmínkách WHERE (=,>,<), nebo BETWEEN, případně jsou využívány v JOIN, by měly být definovány jako první.
- Další pořadí určuje různorodost atributů (nejvíce k nejméně)