Tworzenie indeksu

- Wiele zapytań odnosi się tylko do niewielkiej części rekordów w tabeli.
- Nieefektywne jest odczytywanie przez system każdego rekordu w celu znalezienia rekordu o określonej wartości
- **Indeksy** to specjalne struktury danych powiązane z tabelami lub widokami, które przyspieszają wykonywanie zapytań
  - dwa typy indeksów: indeks klastrowy i indeks nieklastrowy.
- Polecenie **create index** do tworzenia indeksu

  create index <name> **on** <relation-name> (attribute);

Tworzenie indeksu przykład

- **create table** *student*
  (*ID* **varchar** (5),
  name **varchar** (20) **not null**,
  dept_name **varchar** (20),
  tot_cred **numeric** (3,0) **default** 0,
  primary key (*ID*))
- **create index** *studentID_index* **on** *student*(*ID*)
- Zapytanie:

  select *
  from *student*
  where *ID* = '12345'

  może być wykonane przy użyciu indeksu bez przeglądania wszystkich rekordów relacji *student*

Rodzaje indeksów

- Klastrowy *(clustered,* grupujące)

```
CREATE CLUSTERED INDEX index_name
ON schema_name.table_name (column_list);
```

- struktura B+-drzewa
- narzuca tabelom, jak mają zapisywać fizycznie rekordy na kolejnych stornach
- po utworzeniu indeksu struktura stron ulega reorganizacji wg wskazań indeksu
- 1 indeks klastrujący/1 tabelę
- zazwyczaj PK jako indeks klastrowy

Rodzaje indeksów c.d.

- Nieklastrowy *(nonclustered*, niegrupujące)

```
CREATE NONCLUSTERED INDEX index_name
ON schema_name.table_name (column_list);
```

- struktura B+-drzewa
- zapisywany jako osobny obiekt (jak indeks słów kluczowych na końcu książki)
- wskazuje na dokładnie 1 rekord w tabeli
- może być zakładany na >1 kolumnie
- nie wpływa na organizację fizyczną tabeli => wiele indeksów/tabelę

Indeksy

- Wyłączanie indeksu

  ALTER INDEX index_name ON table_name

  DISABLE;

  ALTER INDEX ALL ON table_name

  DISABLE;

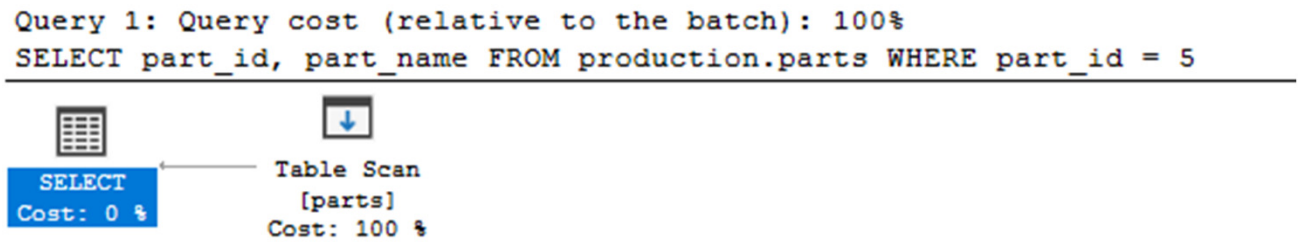
- Usuwanie indeksu

  DROP INDEX [IF EXISTS] index_name

  ON table_name

Demonstracja

- Indeksy
- Optymalizator kosztowy



Transakcje

- **Transakcja** składa się z sekwencji zapytań i/lub instrukcji aktualizacji i jest „jednostką” pracy
- Standard SQL określa, że transakcja rozpoczyna się niejawnie, gdy instrukcja SQL jest wykonywana.
- Transakcja musi kończyć się jedną z następujących instrukcji :
  - **Commit work**. Aktualizacje wykonywane przez transakcję stają się trwałe w bazie danych.
  - **Rollback work**. Wszystkie aktualizacje wykonywane przez instrukcje SQL w transakcji są wycofywane.
- Transakcja atomowa
  - albo w pełni wykonana albo wycofana jakby nigdy nie wystąpiła
- Izolacja od transakcji współbieżnych

Transakcje c.d.

- W wielu SZBD (np. MySQL, PostgreSQL) – domyślnie każde wyrażenie SQL = transakcja
  - automatyczny *commit*
  - zmiana gdy transakcja składa się z wielu wyrażeń
    - np.: **set autocommit off**
  - lepsze rozwiązanie (SQL:1999):
    - **begin atomic...end** albo
    - **begin...commit/rollback work**