Ćwiczenia 11

Zadanie 1.

Analizowane zapytanie:

```
SELECT SalesOrderID, SalesOrderDetailID
FROM Sales.SalesOrderDetail
WHERE SalesOrderID = 43683 AND SalesOrderDetailID = 240;
```

Explain Analyse (Raw):

```
"Index Only Scan using ""PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID"" on
salesorderdetail  (cost=0.42..8.44 rows=1 width=8) (actual time=0.018..0.019 rows=1
loops=1)"
  Index Cond: ((salesorderid = 43683) AND (salesorderdetailid = 240))
  Heap Fetches: 0
Planning Time: 0.071 ms
Execution Time: 0.033 ms
```

Wnioski:
Analiza wykonana za pomocą Explain Analyze dla tego zapytania wykazała, że zapytanie to jest bardzo efektywne. Wykorzystano indeks i klucz główny, co przyczyniło się do minimalizacji czasu wykonania i operacji dostępu do danych. Czas planowania jest niski, a operacje odbyły się praktycznie natychmiastowe.

Explain Analyse (Raw) dla zapytania bez klucza głównego:

```
Seq Scan on salesorderdetail  (cost=0.00..3299.76 rows=1 width=8) (actual
time=0.055..29.485 rows=1 loops=1)
  Filter: ((salesorderid = 43683) AND (salesorderdetailid = 240))
  Rows Removed by Filter: 121316
Planning Time: 0.060 ms
Execution Time: 29.515 ms
```

Wnioski:
Po usunięciu klucza głównego, zauważalna jest znaczna różnica w wydajności wykonania zapytania. Zamiast korzystać z indeksu, baza danych musiała przeszukać całą tabelę sekwencyjnie, co wpłynęło negatywnie na czas wykonania. Pokazuje to, że usuwanie klucza głównego może prowadzić do spadku wydajności zapytań, szczególnie przy dużych zbiorach danych.

Analizowane zapytanie:

```
SELECT SalesOrderID, SalesOrderDetailID
FROM Sales.SalesOrderDetail
WHERE SalesOrderID = 43683 OR SalesOrderDetailID = 240;
```

Explain Analyse (Raw):

```
Bitmap Heap Scan on salesorderdetail  (cost=2258.76..2285.42 rows=7 width=8) (actual time=3.562..3.565 rows=13 loops=1)
  Recheck Cond: ((salesorderid = 43683) OR (salesorderdetailid = 240))
  Heap Blocks: exact=1
  ->  BitmapOr  (cost=2258.76..2258.76 rows=7 width=0) (actual time=3.555..3.556 rows=0 loops=1)
        ->  Bitmap Index Scan on pk_salesorderdetail_salesorderid_salesorderdetailid (cost=0.00..4.46 rows=6 width=0) (actual time=0.029..0.029 rows=13 loops=1)
              Index Cond: (salesorderid = 43683)
        ->  Bitmap Index Scan on pk_salesorderdetail_salesorderid_salesorderdetailid (cost=0.00..2254.30 rows=1 width=0) (actual time=3.524..3.524 rows=1 loops=1)
              Index Cond: (salesorderdetailid = 240)
Planning Time: 0.072 ms
Execution Time: 3.603 ms
```

Explain Analyse (Raw) dla zapytania bez klucza głównego:

```
Seq Scan on salesorderdetail  (cost=0.00..3299.76 rows=7 width=8) (actual time=0.072..14.409 rows=13 loops=1)
  Filter: ((salesorderid = 43683) OR (salesorderdetailid = 240))
  Rows Removed by Filter: 121304
Planning Time: 0.083 ms
Execution Time: 14.438 ms
```

Wnioski:
Klucz główny jest kluczowy dla optymalnej wydajności zapytań. Usunięcie go prowadzi do spadku wydajności, zwłaszcza przy skanowaniu całej tabeli. Warto zawsze brać pod uwagę strukturę bazy danych i rodzaj zapytań przy decyzjach dotyczących indeksów i kluczy głównych.

Zadanie 2.

Analizowane zapytanie:

```
SELECT SalesOrderID, ProductID
FROM Sales.SalesOrderDetail
WHERE ProductID < 800;
```

Explain Analyse (Raw) bez używania indeksów:

```
Seq Scan on salesorderdetail  (cost=0.00..2996.46 rows=47205 width=8) (actual
time=0.035..20.104 rows=47065 loops=1)
  Filter: (productid < 800)
  Rows Removed by Filter: 74252
Planning Time: 0.077 ms
Execution Time: 21.858 ms
```

Explain Analyse (Raw) z użyciem indeksu:

```
Bitmap Heap Scan on salesorderdetail  (cost=538.13..2608.19 rows=47205 width=8)
(actual time=2.140..13.162 rows=47065 loops=1)
  Recheck Cond: (productid < 800)
  Heap Blocks: exact=1439
  ->  Bitmap Index Scan on idx_productid  (cost=0.00..526.33 rows=47205 width=0)
(actual time=1.977..1.977 rows=47065 loops=1)
        Index Cond: (productid < 800)
Planning Time: 0.089 ms
Execution Time: 14.903 ms
```

Wnioski:
Analiza zapytania bez indeksów wykazała długotrwałe przeszukiwanie sekwencyjne tabeli, prowadząc do wolniejszego wykonania. Użycie indeksu na kolumnie ProductID skróciło czas, poprawiając efektywność operacji odczytu. Wnioski potwierdzają, że stosowanie indeksów zwiększa wydajność, szczególnie w przypadku warunków WHERE, minimalizując ilość przetwarzanych danych.