



Etap 5A

Grupa B6 – Loty



Hanna Grodzicka



Mateusz Najda



Rozszerzenia

Tabele do rozszerzenia zostały wybrane na podstawie największej częstości wykonywania na nich operacji oraz potencjalnej poprawy efektywności

składowania danych w stosunku do tabeli wierszowej, np. niektóre kolumny odnoszące się do pracowników mogą zawierać takie same wartości.

Każda z tabel pojawia się co najmniej w dwóch lub trzech operacjach.

1. `Employee`

Opis przeznaczenia: Tabela z informacjami o linii lotniczej, która posiada samoloty oraz pracowników.

Rozszerzenie schematu: Przepisania istniejącej tabeli wierszowej na tabelę kolumnową.

Potencjalnie powtarzające się wartości w kolumnach: `name`, `surname`, `gender`, `occupation`, `airline_id`.

Używana jest przez operacje: 1, 5.

Generowanie danych: Wykorzystanie dla tabeli opcji `INMEMORY` (*In-Memory Column Store*), która służy do przechowywania tabel i partycji w pamięci przy użyciu formatu kolumnowego zoptymalizowanego do szybkiego skanowania.

2. `Flight`

Opis przeznaczenia: Tabela zawiera szczegóły dot. odbytego lotu. Każdy lot odbywa się w danym samolocie, jest obsługiwany przez daną załogę lotniczą oraz startuje z lotniska.

Rozszerzenie schematu: Przepisania istniejącej tabeli wierszowej na tabelę kolumnową.

Potencjalnie powtarzające się wartości w kolumnach:

`destination_country`, `destination_city`, `passengers`, `status`, `delay`, `is_national`, `plane_id`, `airport_id`.

Używana jest przez operacje: 1, 2, 4.

Generowanie danych: Wykorzystanie dla tabeli opcji `INMEMORY` (*In-Memory Column Store*), która służy do przechowywania tabel i partycji w pamięci przy użyciu formatu kolumnowego zoptymalizowanego do szybkiego skanowania.

3. `Plane`

Opis przeznaczenia: Z tabeli można odczytać specyfikację samolotu. Łączy się z linią lotniczą, do której należy, oraz z lotami, które są w nim przeprowadzane.

Rozszerzenie schematu: Przepisania istniejącej tabeli wierszowej na tabelę kolumnową.

Potencjalnie powtarzające się wartości w kolumnach: `manufacturer`, `model`, `seat_count`, `fuel_capacity`, `cruising_range`, `airline_id`.

Używana jest przez operacje: 2, 3, 4.

Generowanie danych: Wykorzystanie dla tabeli opcji `INMEMORY` (*In-Memory Column Store*), która służy do przechowywania tabel i partycji w pamięci przy użyciu formatu kolumnowego zoptymalizowanego do szybkiego skanowania.

Operacje

Zdecydowano się na wykorzystanie operacji z zestawu C ze względu na ich najdłuższy czas wykonania.

1. **SELECT** Średni wiek pracowników

Opis: Średni wiek wszystkich pracowników przypisanych do wszystkich lotów z opóźnieniem pomiędzy 5.0 a 10.0. Oczekiwany wynik: `average employee age = 49.7688022`

Operacja SQL:

```
SELECT AVG(TRUNC((sysdate - e.birthdate) / 365)) AS 'average employee age'
FROM employee e
  JOIN flight_employee fe ON e.id = fe.employee_id
  JOIN flight f ON fe.flight_id = f.id
WHERE f.delay > 5.0 AND f.delay < 10.0;
```

2. **SELECT** Firmy i długości odbytych w nich lotów

Opis: Zestawienie wszystkich nazw firm linii lotniczych z sumą długości lotów (w dniach) odbytych w ciągu dnia najbliższego do teraźniejszej daty. Oczekiwany wynik: `company = Lectus Nullam Associates`

Operacja SQL:

```
SELECT a.company, SUM(TO_NUMBER(f.arrival - f.departure))
FROM airline a
  JOIN plane p ON a.id = p.airline_id
  JOIN flight f ON p.id = f.plane_id
WHERE f.departure = (SELECT MAX(departure) FROM flight)
GROUP BY a.company;
```

3. **CREATE** Dodanie wydajnych samolotów

Opis: Znalezienie dla danej linii lotniczej samolotów, które mają najbardziej korzystny stosunek pojemności baku do zasięgu lotu i dodanie po jednym samolocie o tych samych atrybutach dla każdego z nich. Oczekiwany wynik: 4005 rows inserted.

Operacja SQL:

```
INSERT INTO plane (id,
                  manufacturer,
                  model,
                  production_date,
                  seat_count,
                  fuel_capacity,
                  cruising_range,
                  airline_id)
SELECT plane_id_seq.NEXTVAL, a, b, c, d, e, f, g
FROM (SELECT p.manufacturer a,
            p.model b,
            p.production_date c,
            p.seat_count d,
            p.fuel_capacity e,
            p.cruising_range f,
            p.airline_id g
      FROM plane p
     ORDER BY TRUNC(p.fuel_capacity) / p.cruising_range
    FETCH NEXT 5 ROWS ONLY);
```

4. **UPDATE** Nowe daty stworzenia linii lotniczych

Opis: Uaktualnienie wszystkich dat stworzenia linii lotniczych wpisując datę wylotu pierwszego wykonanego dla nich lotu. Oczekiwany wynik: 1,000 rows updated.

Operacja SQL:

```
UPDATE airline
SET establishment_date = (SELECT MIN(f.departure)
                        FROM flight f
                       JOIN plane p ON p.id = f.plane_id
                       JOIN airline a ON a.id = p.airline_id);
```

5. **DELETE** Usunięcie niektórych pracowników

Opis: Usunięcie tych pracowników, którzy zostali zatrudnieni przez linię lotniczą przed jej utworzeniem. Oczekiwany wynik: 5,320 rows deleted.

Operacja SQL:

```
DELETE (SELECT *  
        FROM employee e  
        INNER JOIN airline a ON a.id = e.airline_id  
        WHERE e.employment_date < a.establishment_date);
```

Eksperymenty

1. Sprawdzenie różnicy w czasie wykonania trzeciej operacji dla opcji składowania kolumn w pamięci:
 - wyłączonej (składowanie wierszowe)
 - włączonej dla jednej kolumny `model` z tabeli `Plane`
 - włączonej dla całej tabeli `Plane`
2. Sprawdzenie czasu wykonania (i/lub wielkości skompresowanej tabeli) dla trzeciej operacji przy wykorzystaniu kilku różnych algorytmów kompresji danych:
 - Query Low
 - Query High
 - Archive Low
 - Archive High

Zgodność wyników można porównać z tymi otrzymanymi dla Oracle 11g:

Type of Compression	Compressed Table Size
Uncompressed	100.00%
Query Low	14.70%
Query High	8.82%
Archive Low	6.62%
Archive High	4.41%

Źródło: <https://www.oracle.com/technical-resources/articles/enterprise-manager/11g-compression.html>