

Etap 1B

Grupa B6 - Loty

1. Przygotowanie i wypełnienie bazy danych danymi testowymi, opisanie wolumetrii.

Wolumetria

Tabela	Liczba wpisów	Zajmowane miejsce [MB]
<u>AIRPORT</u>	2000	0.1875
<u>AIRLINE</u>	1000	0.125
<u>EMPLOYEE</u>	8000	0.5625
<u>PLANE</u>	4000	0.25
<u>FLIGHT</u>	10000	0.9375
<u>FLIGHT_EMPLOYEE</u>	10000	0.1875
<u>RAZEM</u>	35000	2.25

2. Przygotowanie środowiska wykonania pomiarów czasu wykonania próbki obciążenia systemu.

Środowisko

Wykorzystana została maszyna wirtualna dostępna na stronie Oracle. Oprócz bazy danych Oracle 19c, zawiera m.in. Oracle SQL Developer.

<https://www.oracle.com/database/technologies/databaseappdev-vm.html>

Krok 1. Pobranie Oracle VM VirtualBox.

Krok 2. Pobranie obrazu bazy Oracle 19c (wymagane logowanie na konto Oracle).

Próbki obciążenia systemu

▼ Zestaw A

Σ czas: 0.80 s

▼ SELECT

Czas: 0.667 s

964 rows selected.

Stosunek **liczby lotów opóźnionych** (o dowolną dodatnią wartość) do **liczby wszystkich lotów** w historii **danej linii lotniczej**.

```
SELECT SUM(CASE WHEN f.delay > 0 THEN 1 ELSE 0 END) / COUNT(*)
FROM flight f
JOIN plane p ON p.id = f.plane_id
JOIN airline a ON a.id = p.airline_id
INNER JOIN flight_employee fe ON f.id = fe.flight_id
INNER JOIN employee e ON e.id = fe.employee_id
GROUP BY a.id;
```

▼ SELECT

Czas: 0.047 s

```
woman: all = 5094, delayed = 5039
man: all = 4905, delayed = 4856
```

Zestawienie na konkretny dzień:

- Liczby **wszystkich lotów**, **lotów opóźnionych** dla lotów przeprowadzonych przez **pracowników płci żeńskiej** będących **pilotami**.
- Liczby **wszystkich lotów**, **lotów opóźnionych** dla lotów przeprowadzonych przez **pracowników płci męskiej** będących **pilotami**.

```
SELECT SUM(CASE WHEN e.gender = 1 THEN 1 ELSE 0 END) AS "all flights (women)",
       SUM(CASE WHEN e.gender = 1 AND f.delay > 0 THEN 1 ELSE 0 END) AS "delayed flights (women)",
       SUM(CASE WHEN e.gender = 0 THEN 1 ELSE 0 END) AS "all flights (men)",
       SUM(CASE WHEN e.gender = 0 AND f.delay > 0 THEN 1 ELSE 0 END) AS "delayed flights (men)"
FROM flight f
     INNER JOIN flight_employee fe ON f.id = fe.flight_id
     INNER JOIN employee e ON e.id = fe.employee_id;
```

▼ CREATE

Czas: 0.051 s

```
1,000 rows inserted.
```

Stworzenie nowych pracowników grupując przez ID linii lotniczej.

```
DROP SEQUENCE employee_id_seq;
CREATE SEQUENCE employee_id_seq START WITH 4001;

INSERT INTO employee
(id,
 name,
 surname,
 birthdate,
 gender,
 occupation,
 employment_date,
 flown_hours,
 airline_id)
SELECT employee_id_seq.NEXTVAL, a, b, c, d, f, g, h, i
FROM (SELECT MAX(e.name) a,
             MIN(e.surname) b,
             MEDIAN(birthdate) + 10 c,
             ROUND(AVG(CASE WHEN e.gender = 1 THEN 1 ELSE 0 END)) d,
             MIN(e.occupation) f,
             MEDIAN(employment_date) g,
             VARIANCE(flown_hours) / COUNT(*) h,
             e.airline_id i
      FROM employee e
      GROUP BY e.airline_id);
```

▼ UPDATE

Czas: 0.326 s

```
9,000 rows updated.
```

Obliczenie **przelatanych godzin** dla **wszystkich pracowników** na podstawie **długości lotów**, w których brali udział.

```
UPDATE employee eu
SET eu.flown_hours = (SELECT SUM(24 * (f.arrival - f.departure))
                     FROM flight f
                     JOIN flight_employee fe ON f.id = fe.flight_id
                     JOIN employee e ON e.id = fe.employee_id);
```

▼ DELETE

Czas: 0.335 s

803 rows deleted.

Usunięcie tych linii lotniczych, które zatrudniają jakichkolwiek nieletnich.

```
DELETE (SELECT *
        FROM airline a
        JOIN employee e ON e.airline_id = a.id
        WHERE TRUNC(MONTHS_BETWEEN(sysdate, e.birthdate)/12) < 18);
```

▼ Zestaw B

Σ czas: 1.13 s

▼ SELECT

Czas: 0.029 s

COUNT(*) = 192

Liczba lotów na wszystkich lotniskach, których lokalizacja zawiera się pomiędzy długościami geograficznymi skrajnych punktów Brazylii (73°58'58.19" W i 34°47'35.33" W) oraz których wylot nastąpił w pierwszej połowie roku.

```
SELECT COUNT(*)
FROM flight f
JOIN airport a ON a.id = f.airport_id
WHERE EXTRACT(MONTH FROM TRUNC(f.departure)) IN (1, 6)
      AND a.longitude BETWEEN -73.9828305555555 AND -34.7931472222222;
```

▼ SELECT

Czas: 0.041 s

11 rows selected.

Wszystkie unikalne typy samolotów, którymi lecieli pracownicy pracujący dla linii lotniczej (zatrudniającej ich), których pierwsza litera imienia należy do pierwszej połowy alfabetu (A-M).

```
SELECT DISTINCT model
FROM plane p
JOIN airline a ON a.id = p.airline_id
JOIN employee e ON a.id = e.airline_id
WHERE SUBSTR(e.name, 1, 1) BETWEEN 'A' AND 'M';
```

▼ CREATE

Czas: 0.062 s

30 rows inserted.

Wybranie samolotu o największej pojemności pasażerów i dodanie dla niego lotów.

```
DROP SEQUENCE flight_id_seq;
DROP SEQUENCE flight_passengers_seq;

CREATE SEQUENCE flight_id_seq START WITH 5001;
CREATE SEQUENCE flight_passengers_seq START WITH 1 INCREMENT BY 30;

INSERT INTO flight
(id,
 flight_number,
 destination_country,
 destination_city,
 departure,
 arrival,
 passengers,
```

```

status,
delay,
is_national,
plane_id,
airport_id)
SELECT flight_id_seq.NEXTVAL,
'123',
airline.country,
airline.country,
(SELECT MAX(arrival)+1 FROM flight),
(SELECT MAX(arrival)+2 FROM flight),
flight_passengers_seq.NEXTVAL,
'CONFIRMED',
0,
(CASE WHEN airline.country = airport.country THEN 1 ELSE 0 END),
p.id,
airport.id
FROM plane p
JOIN airline ON airline.id = p.airline_id
JOIN flight f ON p.id = f.plane_id
JOIN airport ON airport.id = f.airport_id
WHERE p.seat_count = (SELECT MAX(seat_count) FROM plane);

```

▼ UPDATE

Czas: 0.048 s

18 rows updated.

Zaktualizowanie **czasu przylotu** dodając 20% czasu długości samego lotu dla lotów, które mają wpisane to samo **państwo dla lotniska**, z którego startują oraz jako **państwo destynacji**, ale pomimo tego **nie są lotami narodowymi**.

```

UPDATE (SELECT f.arrival as OLD,
              f.arrival + 0.2 * (f.arrival - f.departure) as NEW
        FROM flight f
        JOIN airport a ON a.id = f.airport_id
        WHERE f.destination_country = a.country
              AND f.is_national = 0) t
SET t.OLD = t.NEW;

```

▼ DELETE

Czas: 1.016 s

6,236 rows deleted.

Usunięcie **wszystkich lotów**, dla których **liczba pasażerów** przekracza **liczbę siedzeń w samolocie**.

```

DELETE (SELECT *
        FROM flight f
        INNER JOIN plane p ON p.id = f.plane_id
        WHERE f.passengers > p.seat_count);

```

▼ Zestaw C

Σ czas: 3.32 s

▼ SELECT

Czas: 0.052 s

average employee age = 49.7688022

Średni wiek **wszystkich pracowników** przypisanych do **wszystkich lotów** z **opóźnieniem** pomiędzy 5.0 a 10.0.

```

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;
```

▼ SELECT

Czas: 0.077 s

company = Lectus Nullam Associates

Zestawienie wszystkich **nazw firm linii lotniczych** z **sumą długości lotów** (w dniach) odbytych **w ciągu dnia najbliższego do teraźniejszej daty**.

```
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;
```

▼ CREATE

Czas: 0.085 s

4005 rows inserted.

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.

```
DROP SEQUENCE plane_id_seq;
CREATE SEQUENCE plane_id_seq START WITH 4001;
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);
```

Czas: 0.138 s

▼ UPDATE

Czas: 0.046 s

1,000 rows updated.

Uaktualnienie wszystkich **dat stworzenia linii lotniczych** wpisując **datę wylotu pierwszego** wykonanego dla nich **lotu**.

```
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);
```

▼ DELETE

Czas: 1.867 s

5,320 rows deleted.

Usunięcie tych pracowników, którzy zostali zatrudnieni przez linię lotniczą przed jej utworzeniem.

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