Etap 4B

Grupa B6 – Loty

✈️ Hanna Grodzicka

🇧🇷 Mateusz Najda

Rozszerzenia

- **Tabela** `Airport_geo`

  Dodano:

  1. punkt `Location` – koordynaty takie same jak dla danych, które istniały w miejscu nowej kolumny tj. `Longitude` i `Latitude`,
  2. wielokąt (koło) `Area` – ma swój środek w punkcie `Location` i stałą średnicę o wartości 2.

```
create table Airport_geo (
  ID        number(10) not null,
  NAME      varchar2(255) not null,
  CODE      varchar2(255) not null unique,
  LOCATION  MDSYS.SDO_GEOMETRY, -- 2D POINT
  COUNTRY   varchar2(255) not null,
  CITY      varchar2(255) not null,
  AREA      MDSYS.SDO_GEOMETRY, -- 2D CIRCLE (center=LOCATION, radius=2)
  primary key (ID));
```

  Przykładowy insert:

```
INSERT INTO AIRPORT_GEO (ID,NAME,CODE,LOCATION,COUNTRY,CITY,AREA)
VALUES
(1,'Fusce','M4G 0T5',
 MDSYS.SDO_GEOMETRY(2001,8307,
  MDSYS.SDO_POINT_TYPE(-127.88974,0.48984,NULL),NULL,NULL
 ),
 'Belize','Minatitlán',
 MDSYS.SDO_GEOMETRY(2003,NULL,NULL,
  MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,4),
  MDSYS.SDO_ORDINATE_ARRAY(-128.88974,-2.48984, -126.88974,0.48984, -128.88974,2.48984)
 )
);
```

- **Tabela** `Flight_geo`

  Dodano:

  3. punkt `Destination` – losowe koordynaty,

  4. linia `Route` – łączy punkt `Destination` z losowym punktem.

```
create table Flight_geo (
  ID                  number(10) not null,
  FLIGHT_NUMBER       varchar2(255) not null,
  DESTINATION         MDSYS.SDO_GEOMETRY, -- 2D POINT
  ROUTE               MDSYS.SDO_GEOMETRY, -- COMPOUNT LINE STRING
  DESTINATION_COUNTRY varchar2(255) not null,
  DESTINATION_CITY    varchar2(255) not null,
  DEPARTURE           date not null,
  ARRIVAL             date not null,
  PASSENGERS          number(10) not null,
  STATUS              varchar2(255) not null,
  DELAY               float(10),
  IS_NATIONAL         number(1) not null,
  PLANE_ID            number(10) not null,
```

```
  AIRPORT_ID          number(10) not null,
  primary key (ID));
```

Przykładowy insert:

```
INSERT INTO FLIGHT_GEO (ID,FLIGHT_NUMBER,DESTINATION,ROUTE,DESTINATION_COUNTRY,DESTINATION_CITY,DEPARTURE,ARRIVAL,PASSENGER
VALUES (1,'04856',
    MDSYS.SDO_GEOMETRY(2001,NULL,MDSYS.SDO_POINT_TYPE(42.506317, 1.521835, NULL), NULL, NULL),
    MDSYS.SDO_GEOMETRY(
      2002,
      NULL,
      NULL,
      MDSYS.SDO_ELEM_INFO_ARRAY(1,2,1), -- compound line string
      MDSYS.SDO_ORDINATE_ARRAY(10,10, 10,14)
      ),
    'Andorra','Cap-de-la-Madeleine',to_date('2019-12-11 09:21:41', 'YYYY-MM-DD HH24:MI:SS'),to_date('2019-07-15 06:19:03',
```

Operacje

1. Usunięcie lotów, których cel podróży był w zakresie obszaru obsługiwanego przez lotnisko.

```
DELETE FROM flight_geo f
WHERE f.id IN (SELECT ff.id
               FROM flight_geo ff, airport_geo a
               WHERE SDO_RELATE(a.area,
                                ff.destination,
                                'mask=ANYINTERACT querytype=JOIN') = 'TRUE');
```

2. Zliczenie ile lotnisk obsługuje ten sam fragment obszaru, co wybrane lotnisko.

```
SELECT COUNT(*)
FROM airport_geo a1, airport_geo a2
WHERE a1.id != a2.id
  AND SDO_RELATE(a1.area,
                 a2.area,
                 'mask=OVERLAPBDYINTERSECT querytype=JOIN') = 'TRUE';
```

3. Wyznaczenie odległości między lotniskami i wybór najkrótszej.

```
SELECT a1.id,
       MIN(SDO_GEOM.SDO_DISTANCE(a1.location, a2.location, 0.005, 'unit=KM'))
FROM airport_geo a1, airport_geo a2
WHERE a1.id != a2.id
GROUP BY a1.id;
```

4. Połączenie obszarów obejmowanych przez lotniska, jeśli suma ich powierzchni jest mniejsza niż 30 000.

```
BEGIN
FOR rec IN (SELECT a1.id a1id, a2.id a2id,
                   SDO_GEOM.SDO_UNION(a1.area, a2.area, 0.005) newAirportArea
            FROM airport_geo a1, airport_geo a2
            WHERE ROUND(SDO_GEOM.SDO_AREA(SDO_GEOM.SDO_UNION(a1.area, a2.area, 0.005), 0.005)) < 30000
            AND SDO_RELATE(a1.area, a2.area, 'mask=TOUCH querytype=WINDOW') = 'TRUE' AND a1.id != a2.id ORDER BY a1id ASC)
LOOP
     UPDATE airport_geo
     SET area = rec.newAirportArea
     WHERE id = rec.a1id;

     DELETE FROM airport_geo
     WHERE id = rec.a2id;
END LOOP;
END;
```

5. Wybór wszystkich pracowników, którzy odbyli najdłuższy lot.

```
SELECT e.id
FROM user_sdo_geom_metadata m, employee e
  INNER JOIN flight_employee fe ON e.id = fe.employee_id
```

```
  INNER JOIN flight_geo f ON f.id = fe.flight_id
WHERE SDO_GEOM.SDO_LENGTH(f.route, m.diminfo) =
    (SELECT MAX(SDO_GEOM.SDO_LENGTH(ff.route, mm.diminfo))
     FROM flight_geo ff, user_sdo_geom_metadata mm);
```

Pomiary

Wszystkie wyniki podane są w sekundach.

Operacja nr 3 została wykonana na kolumnie bez założonego indeksu.

Operacja 1

Aa Pomiar	# Wynik (z indeksami)	# Wynik (bez indeksów)
1	0.31	1670.72
2	0.27	1679.07
3	0.27	2037.58
4	0.31	1605.49
5	0.3	2004.24

$$\sigma_{indeks} = 0.018330302779823, \quad \sigma^2_{indeks} = 0.000336$$

$$\sigma = 182.93609561811, \quad \sigma^2 = 33465.61508$$

Operacja 2

Aa Pomiar	# Wynik (z indeksami)	# Wynik (bez indeksów)
1	0.16	357.45
2	0.14	379.19
3	0.15	364.4
4	0.14	367.85
5	0.13	372.35

$$\sigma_{indeks} = 0.010198039027186, \quad \sigma^2_{indeks} = 0.000104$$

$$\sigma = 7.3227874474137, \quad \sigma^2 = 53.623216$$

Operacja 3

Aa Pomiar	# Wynik (bez indeksów)
1	149.7
2	128.07
3	128.97
4	122.2
5	124.51

$$\sigma = 9.8130056557611, \quad \sigma^2 = 96.29508$$

Operacja 4

Aa Pomiar	# Wynik (z indeksami)	# Wynik (bez indeksów)
1	0.42	704.65
2	0.32	657.33
3	0.35	907.07
4	0.25	664.74

Aa Pomiar	# Wynik (z indeksami)	# Wynik (bez indeksów)
5	0.22	818.06

$$\sigma_{indeks} = 0.071386273190299, \quad \sigma^2_{indeks} = 0.005096$$

$$\sigma = 97.176214167871, \quad \sigma^2 = 9443.2166$$

Operacja 5

Aa Pomiar	# Wynik (z indeksami)	# Wynik (bez indeksów)
1	1.55	0.81
2	1.2	0.84
3	1.07	0.82
4	1.06	0.8
5	1.09	0.79

$$\sigma_{indeks} = 0.18488915598271, \quad \sigma^2_{indeks} = 0.034184$$

$$\sigma = 0.017204650534085, \quad \sigma^2 = 0.000296$$

Eksperyment

Pomiary czasu wykonano dla operacji 1, 2, 4 i 5 dla:

- danych niezindeksowanych,
- danych zindeksowanych

Założono indeksy na tabelach `flight_geo` oraz `airport_geo` i zmierzono czasy wykonania.

```
INSERT INTO user_sdo_geom_metadata
   (TABLE_NAME,
    COLUMN_NAME,
    DIMINFO,
    SRID)
 VALUES (
 'flight_geo',
 'route',
 SDO_DIM_ARRAY(   -- 100X100 grid
   SDO_DIM_ELEMENT('X', 0, 100, 0.005),
   SDO_DIM_ELEMENT('Y', 0, 100, 0.005)
    ),
 NULL   -- SRID
);

INSERT INTO user_sdo_geom_metadata
    (TABLE_NAME,
     COLUMN_NAME,
     DIMINFO,
     SRID)
  VALUES (
  'flight_geo',
  'destination',
  SDO_DIM_ARRAY(   -- 100X100 grid
    SDO_DIM_ELEMENT('X', 0, 100, 0.005),
    SDO_DIM_ELEMENT('Y', 0, 100, 0.005)
     ),
  NULL   -- SRID
);

INSERT INTO user_sdo_geom_metadata
    (TABLE_NAME,
     COLUMN_NAME,
     DIMINFO,
     SRID)
  VALUES (
  'airport_geo',
  'area',
  SDO_DIM_ARRAY(   -- 100X100 grid
    SDO_DIM_ELEMENT('X', 0, 100, 0.005),
```

```
    SDO_DIM_ELEMENT('Y', 0, 100, 0.005)
     ),
  NULL   -- SRID
);

CREATE INDEX Route_Index ON flight_geo(route) INDEXTYPE IS MDSYS.SPATIAL_INDEX_V2;
CREATE INDEX Destination_Index ON flight_geo(destination) INDEXTYPE IS MDSYS.SPATIAL_INDEX_V2;
CREATE INDEX Area_Index ON airport_geo(area) INDEXTYPE IS MDSYS.SPATIAL_INDEX_V2;
```

Operacje na danych zindeksowanych dla prawie każdego przypadku są znacznie szybsze niż operacje na danych niezindeksowanych.

Przykładowo, operacja pierwsza bez wykorzystania indeksów zajęła średnio 1799.42 s (prawie 30 min), a po założeniu indeksu czas wykonania wyniósł zaledwie 0.292 s.

Jedyna operacja dla której czasy wykonania pogorszyły się po założeniu indeksów to operacja 5. Jednak różnica jest minimalna, bo około 0.4 s.

Podsumowując, wykorzystanie danych przestrzennych bez założenia na nich indeksów mija się z celem. Zazwyczaj jesteśmy w stanie osiągnąć bardzo duże przyspieszenie.

Wnioski

Operacje na danych przestrzennych nie są skomplikowane, a po wykorzystaniu **indeksów** stają się bardzo szybkie.

Wykonanie tych samych operacji przy pomocy danych i zapytań relacyjnych byłoby bardzo trudne. Wymagałoby ręcznego pisania wzorów matematycznych w zapytaniach, co byłoby nieczytelne, trudne w utrzymaniu albo wręcz niemożliwe.

Dzięki wykorzystaniu operacji przestrzennych działania takie jak sprawdzenie tego czy obiekty się przecinają lub policzenie najkrótszej ścieżki stają się bardzo proste.