

Próbnny sprawdzian z SQL

13.04.2021

Dla każdego z poniższych zadań napisz odpowiednie polecenia SQL w zadaniu na SKOSIE. Oczekujemy rozwiązania w postaci pliku zawierającego TREŚCI poleceń SQL, a nie znalezionej odpowiedzi. Nie będą sprawdzane jakiegokolwiek zapytania niepoprawne składniowo, sprawdź swoje rozwiązanie używając `\i plik.sql` ! Plik możesz wysyłać wielokrotnie, sprawdzana będzie wyłącznie najnowsza wersja.

Wczytaj do swojej bazy danych plik `mondial-inputs.sql` znajdujący w archiwum <https://skos.ii.uni.wroc.pl/mod/resource/view.php?id=23025> na SKOSIE.

Zachęcam do korzystania przede wszystkim z dokumentacji PostgreSQL:
<https://www.postgresql.org/docs/11/index.html>.

Format nazwy pliku z rozwiązaniem: `grupa-imie-nazwisko.sql`, gdzie grupa to inicjały prowadzącego Twoją grupę: (`pwi/plg/mpy/rfe/pga`), np. `pwi-Jan-Kowalski.sql`. Wymagany format pliku z rozwiązaniem (tu też podaj swoje imię, nazwisko i grupę):

```
-- Imię Nazwisko, grupa np. Jan Kowalski, pwi
-- Zadanie 1
<zapytanie>

-- Zadanie 2
<zapytanie>
```

Do zdobycia są 2 punkty, po 1 punkcie za zadanie.

Zadanie 1 Znajdź wszystkie polskie miasta położone na wysokości poniżej 100 m n.p.m., które posiadają lotnisko (niezależnie od wysokości lotniska). Dla każdego z tych miast podaj jego pełne dane (wszystkie kolumny tabeli `city`). Możesz założyć, że nazwy takich miast są unikalne, a wysokości w bazie są podane w metrach nad poziomem morza. Wyniki posortuj alfabetycznie wg nazw miast.

Rozwiązanie W zadaniu można było założyć, że nazwy miast są unikalne, czyli nie łączyć po kolumnach `Province` oraz `Country`, ale w praktyce powinno tu się znajdować łączenie po kluczu, czyli po trójce `Country`, `Province`, `City`. Słowo kluczowe `DISTINCT` znajduje się tu po to, aby nie wypisywać wielokrotnie miast z wieloma lotniskami (np. w Niemczech są takie miasta).

```
SELECT DISTINCT City.*
FROM Country
JOIN City ON Country.Code = Country
JOIN Airport
ON Airport.Country = City.Country
AND Airport.Province = City.Province
AND Airport.City = City.Name
WHERE Country.Name = 'Poland'
```

```
    AND City.Elevation < 100
ORDER BY City.Name;
```

Zadanie 2 Znajdź nazwy wszystkich zbiorników wodnych ujętych w tabeli `sea`, do których wpływają rzeki o długości ponad 800 km płynące przez Francję (użyj tabel `geo_river`, `river`). Jako wyniki wypisz nazwę zbiornika oraz jego powierzchnię (jeśli jest podana). Usuń duplikaty. Wyniki posortuj wg powierzchni malejąco (nie przejmuj się tym jak PostgreSQL traktuje wartości puste).

Rozwiązanie

```
SELECT DISTINCT Sea.Name, Sea.Area
FROM Sea
  JOIN River ON Sea.Name = River.Sea
  JOIN geo_River ON River.Name = geo_River.River
  JOIN Country ON Country = Country.Code
WHERE Country.Name = 'France'
  AND Length > 800
ORDER BY 2 DESC;
```