

Sprawdzian z SQL nr 2

9.05.2023

Dla każdego z poniższych zadań napisz odpowiednie polecenia SQL. Oczekujemy rozwiązania w postaci pliku zawierającego treści poleceń SQL, a nie znalezionej odpowiedzi.

Nie będą sprawdzane zapytania niepoprawne składniowo. Sprawdź swoje rozwiązanie np. używając `\i plik.sql`.

Rozwiązania wysyłaj przez formularz pod adresem <https://dbserv.stud.ii/>. Plik możesz wysyłać wielokrotnie, sprawdzana będzie wyłącznie najnowsza wersja. Rób to jak najczęściej! Wszystkie dane komputerach są kasowane po restarcie. W razie ew. problemów koniecznie skontaktuj się z prowadzącym zajęcia, zanim zrestartujesz komputer.

Format pierwszej liniiki rozwiązania:

```
-- grupa-imie-nazwisko
```

gdzie grupa to inicjały prowadzącego Twoją grupę (jmi/mabi/plg/pwi), np.

```
-- pwi-Jan-Kowalski
```

Wymagany format całego pliku z rozwiązaniem:

```
-- grupa-imie-nazwisko
```

```
-- Zadanie 1
```

```
<zapytanie>
```

```
-- Zadanie 2
```

```
<zapytanie>
```

```
...
```

Wczytaj do swojej bazy danych plik `offers-1.sql`.

Zadanie 1 (3 punkty) Dla każdej umiejętności (`skill.name`) policzmy na ilu różnych stanowiskach (`offer.title`) i w ilu różnych ofertach występuje, a także jak istotne (`skill.value`) są te umiejętności.

Napisz zapytanie, które zwraca nazwę umiejętności, liczbę różnych stanowisk i liczbę różnych ofert, w których występuje, a także minimalną, maksymalną oraz zaokrągloną do najbliższej liczby naturalnej średnią wartość pola `value`.

Wynik posortuj malejąco względem liczby stanowisk, a w drugiej kolejności względem liczby ofert.

Zapytanie wzorcowe zwraca 1456 krotek. Pierwsze cztery wiersze rozwiązania wzorcowego (ocenzurowane):

name	stanowiska	oferty	min	max	avg
(ciekawe co tu jest???)	271	1289	1	5	3
Python	224	1365	1	5	3
Java	221	1601	1	5	4
English	219	1235	3	5	4

Rozwiązanie

```
SELECT
    s.name,
    count(distinct o.title) AS stanowiska,
    count(o.id) AS oferty,
    min(value),
    max(value),
    round(avg(value)) AS avg
FROM offer o JOIN
    skill s ON (o.id=s.offer_id)
GROUP BY s.name
ORDER BY 2 DESC, 3 DESC;
```

Zadanie 2 (2 punkty (+1 bonus)) Napisz zapytanie, które zwraca oferowane stanowiska (`offer.title`) oraz łączną liczbę różnych wymaganych umiejętności (`skill.name`) dla danego stanowiska wśród wszystkich pracodawców. Uwzględnij tylko stanowiska posiadające ponad 20 różnych wymagań. Wynik posortuj malejąco względem liczby wymagań, a w drugiej kolejności alfabetycznie wg stanowiska.

Bonusowy punkt możesz otrzymać za dodanie kolumny zawierającej tablicę (`array`) z 4 dowolnymi, przykładowymi umiejętnościami z ofert dla wymienionego stanowiska.

Zapytanie wzorcowe zwraca 13 krotek. Pierwsze trzy wiersze rozwiązania wzorcowego:

title	count	example_skill
DevOps Engineer	57	{"Amazon Web Services",Ansible,API,ArgoCD}
Java Developer	41	{Agile,"Apache Kafka",AWS,Cloud}
Data Engineer	39	{AI/ML,Airflow,"Apache Spark",AWS}

Rozwiązanie

```
SELECT
    o.title,
    count(distinct s.name),
    (array_agg(distinct s.name))[1:4] AS example_skill
FROM
    offer o JOIN
    skill s ON (s.offer_id=o.id)
GROUP BY
    title
HAVING
    count(distinct s.name)>20 ORDER BY 2 DESC, 1;
```

Zadanie 3 (3 punkty) W mało prawdopodobnym przypadku niezaliczenia tego przedmiotu, może Ci się przydać informacja, do jakich firm możesz się rekrutować bez znajomości baz danych ani SQLa.

Sprawdź to i napisz zapytanie, które zwraca nazwy firm, które posiadają ofertę, dla której żadne `skill.name` nie zawiera słowa `'SQL'` ani `'database'`. Ignoruj wielkość liter w napisach. Wyniki posortuj alfabetycznie.

Zapytanie wzorcowe zwraca 1019 krotek.

Rozwiązanie

```
SELECT DISTINCT c.name
FROM company c JOIN
      offer o ON c.id=o.company_id
WHERE
      o.id NOT IN (
            SELECT offer_id
            FROM skill
            WHERE
                  name ILIKE '%sql%' OR
                  name ILIKE '%database%'
            )
ORDER BY 1;
```

Zadanie 4 (3 punkty) Napisz zapytanie, które zwraca nazwy firm, które nie wymagają znajomości baz danych ani SQLa w żadnej swojej ofercie (użyj kryterium z poprzedniego zadania, tzn. `skill.name` nie zawiera słowa `'SQL'` ani `'database'`, przy czym ignoruj wielkość liter w napisach). Wyniki posortuj alfabetycznie.

Zapytanie wzorcowe zwraca 799 krotek.

Rozwiązanie

```
SELECT DISTINCT c.name
FROM company c
WHERE c.id NOT IN
      (SELECT o.company_id
      FROM offer o JOIN
            skill s ON o.id=s.offer_id
            WHERE
                  s.name ILIKE '%sql%' OR
                  s.name ILIKE '%database%'
            )
ORDER BY 1;
```

Zadanie 5 (3 punkty) Dla 10 miast z największą liczbą ofert pracy (wg `company_branch`) wypisz liczbę ofert, w których co najmniej jedno z wymagań (`skill.name`) to **'Snowflake'**. Wyniki posortuj wg liczby takich ofert malejąco.

Zapytanie wzorcowe zwraca **oczywiście** dokładnie 10 krotek.

Zauważ, że baza danych zawiera redundancję: *offer* można połączyć z *company_branch* za pomocą atrybutu *company_id* i tabeli *company* lub bezpośrednio, używając atrybutu *company_branch_id*. W obecnym stanie bazy nie ma znaczenia co wybierzesz ale należy zauważyć, że taka konstrukcja schematu bazy może łatwo doprowadzić do niespójności danych.

Rozwiązanie 1

```
WITH cities AS (  
    SELECT city  
    FROM company_branch cb JOIN  
        offer o ON cb.id=o.company_branch_id  
    GROUP BY city  
    ORDER BY count(o.id) DESC  
    LIMIT 10),  
snowflakeSkill AS (  
    SELECT name,  
        offer_id  
    FROM skill  
    WHERE name = 'Snowflake')  
SELECT cb.city,  
    count(s.name)  
FROM cities c JOIN  
    company_branch cb on c.city=cb.city JOIN  
    offer o ON o.company_branch_id = cb.id LEFT JOIN  
    snowflakeSkill s ON (o.id=s.offer_id)  
GROUP BY cb.city  
ORDER BY 2 DESC, 1;
```

Zauważmy, że niewiele brakuje do poprawności następującemu (bardzo) prostemu rozwiązaniu. Jedyne problem to jak posortować wg `COUNT(DISTINCT offer_id)`?

Rozwiązanie 2

```
SELECT city, COUNT(DISTINCT offer_id)  
FROM company_branch  
    LEFT JOIN offer ON company_branch_id = company_branch.id  
    LEFT JOIN skill ON (offer.id = offer_id AND skill.name ILIKE '%Snowflake%')  
GROUP BY city
```

```
ORDER BY COUNT(offer.id) DESC  
LIMIT 10;
```