

**II Liceum Ogólnokształcące im. Mikołaja Kopernika z Oddziałami
Dwujęzycznymi i Międzynarodowymi w Lesznie
Ul. Bolesława Prusa 33, 64-100 Leszno**

Zadanie nr 102 – Portal społecznościowy

**Praca przygotowana przez
Jakuba Jęśka
kl. 3e2 gr. 2
pod kierunkiem
Dominika Siecińskiego**

Leszno, 10.05.2022

Kod delta 361

Zadanie nr 102 – Portal społecznościowy

Zadanie numer 102 dotyczy pewnego niszowego portalu społecznościowego, w obrębie którego użytkownicy mogą zawierać ze sobą znajomości, a także dzielić się zdjęciami. Jako dane do zadania zostały załączone trzy pliki:

- plik „uzytkownicy.txt”, który opisuje użytkowników portalu. Dla każdego z nich podano jego identyfikator, imię, nazwisko, kraj pochodzenia oraz płeć.
- plik „znajomosci.txt”, który zawiera listę par znajomych. W każdym wierszu, oprócz wiersza nagłówkowego, znajdują się dwa różne identyfikatory użytkowników, którzy są znajomymi, oraz data zawarcia znajomości.
- plik „zdjecia.txt”, który opisuje zdjęcia opublikowane przez użytkowników. Dla każdego z nich podano jego identyfikator, identyfikator użytkownika, który je dodał, oraz podaną w pikselach szerokość oraz wysokość zdjęcia.

Po głębszej analizie czujne oko każdego informatyka powinno zauważyć, że zadanie numer 102 najłatwiej rozwiązać za pomocą języka zapytań SQL. Niniejszy dokument, ze względu na ten fakt, zawiera rozwiązanie zadania w programie LibreOffice Base.

Przed rozwiązaniem zadania należy zaimportować dane z dołączonych plików do bazy danych. W niniejszym rozwiązaniu nazwy tabel odpowiadają nazwom plików, z których importowano dane. Podczas importowania danych warto pamiętać o ustawieniu odpowiednich typów danych.

ROZWIĄZANIE ZADANIA:

Pierwszy podpunkt polega na podaniu liczby zdjęć, które zostały opublikowane w 2014 roku. Kwerenda rozwiązująca to zadanie wygląda następująco:

```
SELECT COUNT( "ID_zdjecia" ) AS "Liczba_zdjec"  
FROM "zdjecia"  
WHERE YEAR( "Data_dodania" ) = '2014'
```

Zdjęcie numer 1

Przedstawia przykładowe rozwiązanie pierwszego podpunktu zadania

Należy zauważyć, że do rozwiązania tego podpunktu potrzebna jest nam tylko tabela zawierająca dane o zdjęciach. Niezbędna okazuje się również funkcja agregująca COUNT(), która zwraca liczbę rekordów, które spełniają określony warunek. Natomiast jedynym warunkiem w tym zadaniu jest

Zadanie nr 102 – Portal społecznościowy

odpowiednia data publikacji zdjęcia – zdjęcie musiało zostać opublikowane w 2014 roku. W takim przypadku przydatna okazuje się funkcja YEAR(), która zwraca rok z przekazanej do niej daty. W celu polepszenia estetyki rozwiązania nazwa zwracanej kolumny została zmieniona.

W kolejnym podpunkcie należy podać wszystkie pary użytkowników, którzy są znajomymi i mają takie samo imię. Dla każdej takiej pary powinno się podać imię, nazwisko i kraj pochodzenia każdego użytkownika. Jednym z możliwych rozwiązań jest kwerenda:

```
SELECT "A"."Imie" AS "Imie1", "A"."Nazwisko" AS "Nazwisko1", "A"."Kraj" AS "Kraj1", "B"."Imie" AS "Imie2", "B"."Nazwisko" AS "Nazwisko2", "B"."Kraj" AS "Kraj2"  
FROM "uzytkownicy" AS "A", "uzytkownicy" AS "B", "znajomosci"  
WHERE "znajomosci"."Znajomy_1" = "A"."ID_uzytkownika" AND "znajomosci"."Znajomy_2" = "B"."ID_uzytkownika" AND "A"."Imie" = "B"."Imie"
```

Zdjęcie numer 2

Przedstawia przykładowe rozwiązanie drugiego podpunktu zadania

Kluczowym etapem rozwiązania tego zadania jest określenie potrzebnych tabel. Dane o zawartych znajomościach znajdują się w tabeli „znajomości”. W takim wypadku na pewno będzie trzeba z niej skorzystać. Niestety, w wymienionej wcześniej tabeli znajdują się tylko unikatowe identyfikatory użytkowników – my potrzebujemy natomiast imion użytkowników. Potrzebne dane znajdują się w tabeli „uzytkownicy”. Jednak jedna taka tabela nie wystarczy – w każdym rekordzie tabeli „znajomosci” znajdują się identyfikatory dwóch różnych użytkowników. W takim wypadku do tabeli „znajomosci” należy dołączyć aż 2 tabele „uzytkownicy”. Umożliwia nam to zastosowanie aliasów do tabel. Oczywiście dla każdej z tabel „uzytkownicy” należy zdefiniować odpowiednie relacje z tabelą „znajomosci” w klauzuli WHERE. Oprócz tego, w klauzuli WHERE należy również zapisać warunek dotyczący imion.

Trzeci podpunkt polega na podaniu zestawienia 10 krajów, z których pochodzi najwięcej zdjęć. Dla każdego kraju należy podać jego nazwę oraz liczbę zdjęć z niego przesłanych. Zestawienie należy posortować malejąco według liczby opublikowanych zdjęć. Przykładowa kwerenda realizująca to zadanie:

Zadanie nr 102 – Portal społecznościowy

```
SELECT "Kraj", COUNT( "ID_zdjecia" ) AS "Ilosc_zdjec"  
FROM "zdjecia", "uzytkownicy"  
WHERE "zdjecia"."ID_uzytkownika" = "uzytkownicy"."ID_uzytkownika"  
GROUP BY "Kraj"  
ORDER BY "Ilosc_zdjec" DESC LIMIT 10
```

Zdjęcie numer 3

Przedstawia przykładowe rozwiązanie trzeciego podpunktu zadania

Na początku należy zastanowić się, jakie tabele będą potrzebne do rozwiązania tego podpunktu. Dane dotyczące zdjęć znajdują się w tabeli „zdjecia”. Niestety, w wymienionej tabeli nie znajdują się informacje dotyczące kraju pochodzenia użytkownika, który opublikował zdjęcie. Wymagane informacje znajdują się w tabeli „uzytkownicy”. Potrzebne będą nam więc te dwie tabele. Na szczęście, owe tabele można połączyć bez użycia kolejnej tabeli – klucz główny tabeli „uzytkownicy” (ID_uzytkownika) jest kluczem obcym tabeli „zdjecia”. Naturalnie należy pamiętać o uwzględnieniu odpowiednich relacji w kwerendzie. W zwracanej tabeli powinny znaleźć się 2 atrybuty – nazwa kraju oraz ilość zdjęć dodanych przez użytkowników pochodzących z tego kraju. Ponownie niezbędna okazuje się funkcja agregująca COUNT(), która posłuży nam do zliczania zdjęć. Jednak warto zwrócić uwagę, że nasza kwerenda powinna zliczać zdjęcia osobno dla każdego z krajów. W takim przypadku kwerenda musi, przed zliczeniem zdjęć, pogrupować je ze względu na kraj pochodzenia użytkownika. Umożliwia nam to klauzula GROUP BY, w której należy określić, że kryterium grupowania w tym przypadku jest nazwa kraju, z którego pochodzi użytkownik. Ostatnie zdanie polecenia wyraźnie określa, że zwracane zestawienie powinno również zostać posortowane ze względu na ilość dodanych zdjęć. W celu spełnienia tego zadania należy użyć klauzuli ORDER BY. Ze względu na fakt, że zestawienie powinno być posortowane malejąco według ilości zdjęć, należy za nazwą atrybutu dopisać skrót DESC, który zapewni odpowiednią kolejność sortowania. Ostatnim aspektem, o spełnieniu którego należy pamiętać, jest ograniczenie liczby zwracanych rekordów do 10. Dokonuje się tego za pomocą klauzuli LIMIT, która musi znaleźć się na końcu kwerendy.

W następnym podpunkcie należy podać wysokość i szerokość zdjęcia, które ma najwięcej pikseli, oraz imię i nazwisko użytkownika, który je opublikował. Przykładowa kwerenda rozwiązująca ten problem:

Zadanie nr 102 – Portal społecznościowy

```
SELECT "Szerokosc", "Wysokosc", "Imie", "Nazwisko"  
FROM "zdjecia" JOIN "uzytkownicy" ON "uzytkownicy"."ID_uzytkownika" = "zdjecia"."ID_uzytkownika"  
ORDER BY ( "Wysokosc" * "Szerokosc" ) DESC LIMIT 1
```

Zdjęcie numer 4

Przedstawia przykładowe rozwiązanie czwartego podpunktu zadania

Klasycznie, na początku należy ustalić, jakie tabele będą potrzebne do wykonania zadania. Dane o zdjęciach znajdują się w tabeli „zdjecia”. Jednak brakuje w niej danych dotyczących użytkowników – znajdują się one w tabeli „uzytkownicy”. W takim wypadku będziemy potrzebowali zarówno tabeli „zdjecia”, jak i tabeli „uzytkownicy”. Oczywiście, obie tabele należy połączyć odpowiednią relacją. W powyższej kwerendzie, do łączenia tabel została wykorzystana instrukcja JOIN. W tym przypadku jest to jej odmiana domyślna, której efekty działania nie różnią się dużo od efektów połączenia tabel za pomocą odpowiedniego warunku w klauzuli WHERE. Zwracaną tabelę należy posortować malejąco według liczby pikseli. W tym celu należy użyć klauzuli ORDER BY. Warto zwrócić uwagę, że sortować można również według wyrażenia, a nie tylko według atrybutów. Ponownie, malejący kierunek sortowania należy określić za pomocą skrótu DESC, a liczbę zwracanych rekordów powinno ograniczyć się za pomocą klauzuli LIMIT.

W ostatnim podpunkcie należy znaleźć wśród użytkowników mężczyzn, którzy opublikowali w portalu zdjęcie, nie mając w momencie jego publikacji żadnych znajomych. Dla każdego takiego użytkownika powinno się podać jego imię, nazwisko i kraj pochodzenia. Listę należy posortować rosnąco według alfabetycznej kolejności nazwisk użytkowników. Jedną z kwerend rozwiązyujących ten problem:

```
SELECT "Imie","Nazwisko","Kraj" FROM "uzytkownicy" WHERE "Plec" = 'M' AND "ID_uzytkownika" IN  
(SELECT "znajomosci"."Id" FROM  
  (SELECT "ID_uzytkownika" AS "Id", MIN("Data_dodania") AS "Data_dodania_1_zdjecia" FROM "zdjecia" GROUP BY "ID_uzytkownika") AS "zdjecia" LEFT OUTER JOIN  
  (SELECT "Id", MIN("Data") AS "Data_zawarcia_1_znajomosci" FROM  
    (SELECT "Znajomy_1" AS "Id", "Data" FROM "znajomosci"  
     UNION  
     SELECT "Znajomy_2", "Data" FROM "znajomosci")  
   GROUP BY "Id") as "znajomosci" ON  
  "znajomosci"."Id" = "zdjecia"."Id" WHERE "zdjecia"."Data_dodania_1_zdjecia" < "znajomosci"."Data_zawarcia_1_znajomosci" OR "zdjecia"."Data_dodania_1_zdjecia" IS NULL)  
ORDER BY "Nazwisko"
```

Zdjęcie numer 5

Przedstawia przykładowe rozwiązanie piątego podpunktu zadania

W pierwszej kolejności, w celu rozwiązania tego podpunktu należy znaleźć datę zawarcia pierwszej znajomości dla każdego użytkownika, który ma co najmniej jednego znajomego. Niestety, dużym utrudnieniem w tym przypadku jest budowa tabeli „znajomosci.txt” – unikatowy identyfikator

Zadanie nr 102 – Portal społecznościowy

użytkownika, dla którego szukamy datę zawarcia pierwszej znajomości, może znajdować się zarówno w pierwszej, jak i w drugiej kolumnie. Należy zatem uwzględnić obie możliwości. Zastosowanie dyrektywy grupującej GROUP BY nie jest możliwe w takim przypadku. W pierwszej kolejności należy przerobić tabelę „znajomosci” w tabelę, która będzie zawierała tylko dwie kolumny – kolumnę z unikatowym identyfikatorem użytkownika, który zawarł znajomość oraz datę zawarcia tej znajomości. Z pomocą przychodzi operator UNION, który służy do pionowego łączenia tabel wynikowych. Wystarczy, że za pomocą pierwszej kwerendy wyciągniemy z każdego rekordu tabeli „znajomosci” identyfikator pierwszego znajomego oraz datę zawarcia znajomości, a za pomocą drugiej kwerendy identyfikator drugiego znajomego oraz datę zawarcia znajomości, a następnie połączymy pionowo tabele wynikowe. Kwerenda, która realizuje to zadanie wygląda następująco:

```
SELECT "Znajomy_1" AS "Id", "Data" FROM "znajomosci"  
UNION  
SELECT "Znajomy_2","Data" FROM "znajomosci"
```

Zdjęcie numer 6

Przedstawia jedną z kwerend pomocniczych, która została wykorzystana do wykonania piątego podpunktu zadania

Po uzyskaniu dwukolumnowej tabeli, dla każdego użytkownika, który się w niej znajduje należy znaleźć datę zawarcia pierwszej znajomości. Najłatwiej dokonać tego za pomocą klauzuli grupującej GROUP BY oraz funkcji agregującej MIN(). Kod realizujący to zadanie wygląda następująco:

```
SELECT "Id", MIN("Data") AS "Data_zawarcia_1_znajomosci" FROM  
    (SELECT "Znajomy_1" AS "Id", "Data" FROM "znajomosci"  
    UNION  
    SELECT "Znajomy_2","Data" FROM "znajomosci")  
GROUP BY "Id"
```

Zdjęcie numer 7

Przedstawia jedną z kwerend pomocniczych, która została wykorzystana do wykonania piątego podpunktu zadania

Następnie należy znaleźć datę dodania pierwszego zdjęcia dla każdego użytkownika, który udostępnił na portalu co najmniej jedno zdjęcie. W celu uzyskania takiego rezultatu ponownie

Zadanie nr 102 – Portal społecznościowy

posłużymy się klauzulą grupującą GROUP BY oraz funkcją agregującą MIN(). Kod realizujący to zadanie wygląda następująco:

```
SELECT "ID_uzytkownika" AS "Id", MIN("Data_dodania") AS "Data_dodania_1_zdjecia"  
FROM "zdjecia"  
GROUP BY "ID_uzytkownika"
```

Zdjęcie numer 8

Przedstawia jedną z kwerend pomocniczych, która została wykorzystana do wykonania piątego podpunktu zadania

W kolejnym kroku należy połączyć obie uzyskane tabele w jedną całość oraz zwrócić identyfikatory użytkowników, którzy dodali pierwsze zdjęcie, gdy nie mieli żadnego znajomego. W celu połączenia tabel należy skorzystać z polecenia OUTER JOIN. Dzięki użyciu tego polecenia nie stracimy użytkowników, którzy nadal nie posiadają znajomych – jeżeli użytkownik posiada znajomego to zostanie dołączona data zawarcia pierwszej znajomości, a w przeciwnym razie zostanie dodana wartość NULL. Zastosowanie WHERE lub INNER JOIN spowodowałoby zwrócenie tylko tych użytkowników, którzy posiadają znajomych. W klauzuli filtrującej należy również uwzględnić dwie możliwości – warunki zadania zostaną spełnione, jeżeli data dodania pierwszego zdjęcia jest mniejsza od daty zawarcia pierwszej znajomości, lub gdy data dodania pierwszej znajomości jest równa NULL (oznacza to, że użytkownik nadal nie posiada żadnego znajomego, a więc musiał dodać zdjęcie nie mając znajomych). Kwerenda wykonująca to zadanie wygląda następująco:

```
SELECT "znajomosci"."Id" FROM  
(SELECT "ID_uzytkownika" AS "Id", MIN("Data_dodania") AS "Data_dodania_1_zdjecia" FROM "zdjecia" GROUP BY "ID_uzytkownika") AS "zdjecia" LEFT OUTER JOIN  
(SELECT "Id", MIN("Data") AS "Data_zawarcia_1_znajomosci" FROM  
(SELECT "Znajomy_1" AS "Id", "Data" FROM "znajomosci"  
UNION  
SELECT "Znajomy_2", "Data" FROM "znajomosci")  
GROUP BY "Id") as "znajomosci" ON  
"znajomosci"."Id" = "zdjecia"."Id" WHERE "zdjecia"."Data_dodania_1_zdjecia" < "znajomosci"."Data_zawarcia_1_znajomosci" OR "zdjecia"."Data_dodania_1_zdjecia" IS NULL
```

Zdjęcie numer 9

Przedstawia jedną z kwerend pomocniczych, która została wykorzystana do wykonania piątego podpunktu zadania

Wreszcie należy zwrócić rekordy w porządku alfabetycznym według nazwisk wszystkich użytkowników, których identyfikator został zwrócony przez powyższą kwerendę oraz są płci męskiej. W tym przypadku przydatny okazuje się operator IN, który sprawdza, czy użytkownika znajduje się w zwracanym zbiorze. Nie można również zapominać o dodaniu dyrektywy ORDER BY, która sortuje zwracane wyniki.

Zadanie nr 102 – Portal społecznościowy

Część praktyczna polegała na przeprowadzeniu testu z języka SQL. Każde z 10 przygotowanych pytań zostało zadane jednej z osób obecnych na lekcji informatyki, która odbyła się w dniu 20.04.2022 na czwartej godzinie lekcyjnej. Wszystkie udzielone odpowiedzi były poprawne. Potwierdzeniem tego wydarzenia są liczne wspomnienia uczniów oraz nauczyciela informatyki.

TEST:

1. Czym skutkuje dodanie dyrektywy DISTINCT w kwerendzie wybierającej?
 - a.) niczym
 - b.) zwróceniem tylko unikatowych rekordów, pod względem wybranych atrybutów
 - c.) zwróceniem rekordów, które posiadają wartości (ich wartość jest inna niż NULL)
 - d.) zwróceniem rekordów w porządku alfabetycznym
2. Do czego służy funkcja agregująca AVG?
 - a.) sumuje wartości podanego atrybutu
 - b.) szuka wartości maksymalnej
 - c.) szuka wartości minimalnej
 - d.) oblicza średnią wartość podanego atrybutu
3. Do czego służy funkcja ROUND?
 - a.) do zaokrąglania wartości
 - b.) do zmiany typu wartości
 - c.) zwraca aktualną godzinę
 - d.) zmienia date na liczbę całkowitą, która określa liczbę dni, które upłynęły od 1 stycznia 1900 roku
4. Do czego służy klauzula LIMIT?
 - a.) sortuje rekordy w porządku alfabetycznym
 - b.) ogranicza liczbę zwracanych rekordów
 - c.) określa się w niej tabele, z których będą pobierane wartości
 - d.) sortuje rekordy w odwrotnym porządku alfabetycznym
5. Jak nazywają się bazy danych, w których tabele są ze sobą połączone za pomocą relacji?
 - a.) bazy obiektowe
 - b.) bazy relacyjne
 - c.) bazy proste (kartotekowe)
 - d.) bazy strumieniowe
6. Do czego służy funkcja CAST?
 - a.) do zaokrąglania wartości
 - b.) do zmiany typu wartości
 - c.) zwraca aktualną godzinę
 - d.) zmienia date na liczbę całkowitą, która określa liczbę dni, które upłynęły od 1 stycznia 1900 roku
7. Do czego służy klauzula JOIN?
 - a.) pozwala na łączenie ze sobą danych znajdujących się w różnych tabelach
 - b.) ogranicza liczbę zwracanych rekordów

Zadanie nr 102 – Portal społecznościowy

- c.) sortuje rekordy w porządku alfabetycznym
- d.) grupuje rekordy według określonych atrybutów

8. Operator logiczny AND...

- a.) zwraca fałsz, jeśli wszystkie warunki logiczne są prawdziwe
- b.) zwraca prawdę, jeśli co najmniej jeden warunek logiczny jest prawdziwy
- c.) zwraca prawdę, jeśli wszystkie warunki logiczne są prawdziwe
- d.) neguje warunek logiczny (zamienia prawdę na fałsz i odwrotnie)

9. Który z poniższych zapisów odwołuje się do atrybutu „cena” znajdującego się w tabeli „Produkty”?

- a.) Produkty(cena)
- b.) Produkty.cena
- c.) cena(Produkty)
- d.) Produkty >> cena

10. Do czego służy instrukcja TRUNCATE?

- a.) do usuwania wszystkich rekordów z tabeli
- b.) do tworzenia nowej tabeli
- c.) do wstawiania nowych rekordów do tabeli
- d.) do aktualizowania wartości rekordów

POPRAWNE ODPOWIEDZI: 1.b 2.d 3.a 4.b 5.b 6.b 7.a 8.c 9.b 10.a

Zadanie nr 102 – Portal społecznościowy

Analizując wyniki testu można stwierdzić, że uczniowie, którzy rozszerzają informatykę w II Liceum Ogólnokształcącego im. Mikołaja Kopernika w Lesznie, wykazują wysoką znajomość języka zapytań SQL.

Zadanie nr 102 – Portal społecznościowy

Źródła:

1. Zadanie ze strony <http://code.kopernik-leszno.pl/news/w-361/zadanie-nr-102>
pobrano z dnia 10.05.2022

Spis zdjęć, rysunków, tabel i wykresów:

1. Zdjęcie przedstawiające przykładowe rozwiązanie pierwszego podpunktu zadania
zdjęcie własne
2. Zdjęcie przedstawiające przykładowe rozwiązanie drugiego podpunktu zadania
zdjęcie własne
3. Zdjęcie przedstawiające przykładowe rozwiązanie trzeciego podpunktu zadania
zdjęcie własne
4. Zdjęcie przedstawiające przykładowe rozwiązanie czwartego podpunktu zadania
zdjęcie własne
5. Zdjęcie przedstawiające przykładowe rozwiązanie piątego podpunktu zadania
zdjęcie własne
6. Zdjęcie przedstawiające jedną z kwerend pomocniczych, która została wykorzystana do rozwiązania podpunktu piątego zadania
zdjęcie własne
7. Zdjęcie przedstawiające jedną z kwerend pomocniczych, która została wykorzystana do rozwiązania podpunktu piątego zadania
zdjęcie własne
8. Zdjęcie przedstawiające jedną z kwerend pomocniczych, która została wykorzystana do rozwiązania podpunktu piątego zadania
zdjęcie własne
9. Zdjęcie przedstawiające jedną z kwerend pomocniczych, która została wykorzystana do rozwiązania podpunktu piątego zadania
zdjęcie własne