

Bazy Danych

Ćwiczenie 11: Wybrane funkcje wbudowane

opracował: dr hab. inż. Artur Gramacki (a.gramacki@issi.uz.zgora.pl)

1. Poruszane zagadnienia: poznanie wybranych funkcji oraz operatorów dostępnych w serwerze MySQL. Ich znajomość w wielu przypadkach pozwala na większą elastyczność przy tworzeniu zapytań SQL. Omówione zostaną tylko wybrane, najważniejsze według nas, funkcje. W rzeczywistości jest ich dużo więcej. Zachęcamy więc do zapoznania się z nimi odnajdując ich opis w dostępnej literaturze. Niektórych funkcji używałeś już we wcześniejszych ćwiczeniach, być może nieco nieświadomie.
2. Warunkiem uzyskania w ćwiczeniu takich jak poniżej wyników jest praca na demonstracyjnym schemacie relacyjnym, o którym mowa jest w jednym z poprzednich ćwiczeń. Dlatego też należy upewnić się, że odpowiedni skrypt wykonał się bezbłędnie (poprawnie utworzyły się wszystkie tabele oraz załadowały się wszystkie dane). Dla pewności warto przed rozpoczęciem ćwiczenia po prostu utworzyć schemat od nowa, nawet jeżeli wydaje ci się, że ten, który masz aktualnie zainstalowany jest poprawny.
3. Funkcje operujące na łańcuchach

Zadanie 1

W trakcie porównywania łańcuchów MySQL nie uwzględnia wielkości liter. Przykładowo poniższe polecenie zwróci jeden rekord, mimo że nazwisko pracownika jest napisane bez dbałości o zachowanie wielkości liter:

```
mysql> SELECT last_name, salary FROM emp WHERE last_name LIKE 'noZaki';
+-----+-----+
| last_name | salary |
+-----+-----+
| Nozaki    | 1200.00 |
+-----+-----+
1 row in set (0.00 sec)
```

Należy zmodyfikować powyższe zapytanie SQL tak, aby wielkość liter była honorowana. Ponadto należy odnaleźć w dokumentacji informację jak można zdefiniować kolumnę łańcuchową (CHAR, VARCHAR, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT), aby na trwałe wymusić uwzględnianie wielkości liter. Utworzyć następnie przykładową tabelę oraz sprawdzić „nieczułość” na wielkość liter.

Zadanie 2

Wyświetlić dane o pracowniku. Użyć funkcji CONCAT. Jako wynik powinniśmy otrzymać:

```
+-----+
| Pracownik Ngao      |
+-----+
| -->LaDoris-Ngao<-- |
+-----+
1 row in set (0.00 sec)
```

Zadanie 3

Wyświetlić dane o pracowniku, których nazwiska zaczynają się na literę D. Zwróć uwagę, że wyświetlany tekst został uzupełniony do 20 znaków (policz to) sekwencją znaków #*. Należy użyć funkcji RPAD. Czym różni się w działaniu funkcja LPAD? Jako wynik powinniśmy otrzymać:

```
+-----+
| Pracownicy          |
+-----+
| Dumas#####       |
| Dancs#####       |
+-----+
2 rows in set (0.01 sec)
```

Zadanie 4

Wyświetlić nazwy regionów (tabela *region*). W drugiej kolumnie wyświetlono tylko 3 znaki z nazwy poczynając od 2. znaku. Użyć funkcji SUBSTR. Jako wynik powinniśmy otrzymać:

```
+-----+-----+
| name                | Nazwa regionu - 3 znaki od 2. |
+-----+-----+
| Africa / Middle East | fri                             |
| Asia                 | sia                             |
| Europe              | uro                             |
| North America       | ort                             |
| South America       | out                             |
+-----+-----+
5 rows in set (0.00 sec)
```

Zadanie 5

Wyświetlić dane o pracownikach, których nazwiska są dłuższe niż 6 znaków. Wyświetlić długość nazwisk i imion tych pracowników i wynik posortować wg. długości nazwiska w porządku od największej do najmniejszej. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+-----+
| first_name | Długość imienia | last_name  | Długość nazwiska |
+-----+-----+-----+-----+
| Mark       | 4               | Quick-To-See | 12               |
| Carmen     | 6               | Velasquez   | 9                |
| Antoinette | 10              | Catchpole   | 9                |
| Alexander  | 9               | Markarian   | 9                |
| Midori     | 6               | Nagayama    | 8                |
| Audry      | 5               | Ropeburn    | 8                |
```

```

| Molly          |          5 | Urguhart      |          8 |
| Sylvie        |          6 | Schwartz      |          8 |
| Yasmin        |          6 | Sedeghi       |          7 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

```

Zadanie 6

Wyświetlić nazwy regionów (tabela *region*). W drugiej kolumnie wszystkie litery *a* zostały zamienione na *X*. Użyć funkcji `REPLACE`. Jako wynik powinniśmy otrzymać:

```

+-----+-----+
| name                | name2                |
+-----+-----+
| Africa / Middle East | AfricX / Middle EXst |
| Asia                 | AsiX                 |
| Europe               | Europe               |
| North America        | North AmericX       |
| South America        | South AmericX       |
+-----+-----+
5 rows in set (0.01 sec)

```

Zadanie 7

Wyświetlić nazwy regionów (tabela *region*). W drugiej kolumnie nazwy pisane są wspak. Użyć funkcji `REVERSE`. Jako wynik powinniśmy otrzymać:

```

+-----+-----+
| name                | eman                 |
+-----+-----+
| Africa / Middle East | tsaE elddiM / acirfA |
| Asia                 | aisA                 |
| Europe               | eporuE               |
| North America        | aciremA htroN       |
| South America        | aciremA htuoS       |
+-----+-----+
5 rows in set (0.00 sec)

```

Zadanie 8

Wyświetlić dane o pracownikach, których nazwiska zaczynają się na literę *N*. W nazwisku wszystkie litery mają być małe a w identyfikatorze (kolumna *userid*) duże. Użyć funkcji `LOWER` oraz `UPPER`. Jako wynik powinniśmy otrzymać:

```

+-----+-----+
| last_name          | USERID              |
+-----+-----+
| ngao               | LNGAO                |
| nagayama           | MNAGAYAM            |
| nguyen             | MNGUYEN              |
| nozaki             | ANOZAKI              |
| newman             | CNEWMAN              |
+-----+-----+
5 rows in set (0.00 sec)

```

Zadanie 9

Wyświetlić dane o wielkości stanów produktów znajdujących się w hurtowni zlokalizowanej w mieście *Hong Kong*. Ograniczyć się tylko do tych produktów, których stan (kolumna *amount_in_stock*) jest mniejszy niż 100. Dane należy pobierać z trzech tabel (*warehouse*, *inventory*, *product*) i odpowiednio wykonać złączenie w klauzuli *WHERE*.

Jako wynik powinniśmy otrzymać tabelkę DOKŁADNIE taką, jak pokazano poniżej. Zwrócić uwagę na wielkości liter, na etykiety, na długości poszczególnych kolumn, sposób wyrównania tekstu (do lewej lub prawej strony) itd. Jako wynik powinniśmy otrzymać:

```
+-----+
| Stany magazynowe |
+-----+
| W hurtowni ---HONG KONG--- produktu o nazwie:  BLACK HAWK KNEE PADS          jest 88 sztuk |
| W hurtowni ---HONG KONG--- produktu o nazwie:  BLACK HAWK ELBOW PADS          jest 75 sztuk |
| W hurtowni ---HONG KONG--- produktu o nazwie:  GRAND PRIX BICYCLE TIRES       jest 85 sztuk |
| W hurtowni ---HONG KONG--- produktu o nazwie:  NEW AIR PUMP                    jest 0 sztuk  |
| W hurtowni ---HONG KONG--- produktu o nazwie:  ALEXEYER PRO LIFTING BAR        jest 47 sztuk |
| W hurtowni ---HONG KONG--- produktu o nazwie:  PRO CURLING BAR                jest 50 sztuk |
| W hurtowni ---HONG KONG--- produktu o nazwie:  PROSTAR 10 POUND WEIGHT         jest 80 sztuk |
| W hurtowni ---HONG KONG--- produktu o nazwie:  PROSTAR 20 POUND WEIGHT         jest 91 sztuk |
| W hurtowni ---HONG KONG--- produktu o nazwie:  PROSTAR 100 POUND WEIGHT        jest 75 sztuk |
+-----+
9 rows in set (0.20 sec)
```

Zadanie 10

Wyświetlić nazwy regionów (tabela *region*). W drugiej kolumnie wypisać, w którym miejscu po raz pierwszy pojawiła się litera *a*. Użyć funkcji *INSTR*. Jako wynik powinniśmy otrzymać:

```
+-----+-----+
| name          | Pierwsze wystapienie litery 'a' |
+-----+-----+
| Africa / Middle East | 1 |
| Asia          | 1 |
| Europe        | 0 |
| North America | 7 |
| South America | 7 |
+-----+-----+
5 rows in set (0.00 sec)
```

4. Funkcje operujące na liczbach

Zadanie 11

Użyć SQL-a do obliczenia pola i obwodu koła o promieniu 10. Użyć funkcji *POW*. Jako wynik powinniśmy otrzymać:

```
+-----+-----+
| Pole | Obwód |
+-----+-----+
| 314 | 62.80 |
+-----+-----+
1 row in set (0.01 sec)
```

Zadanie 12

Użyć SQL-a do wygenerowanie liczby losowej. Użyć funkcji RAND. Jak „zmusić” funkcję RAND, aby za każdym wywołaniem generowała tą samą liczbę losową (przydatne przy testowaniu oprogramowania). Jako wynik powinniśmy otrzymać (u Ciebie liczba może być inna):

```
+-----+
| Liczba losowa |
+-----+
| 0.21308653076467 |
+-----+
1 row in set (0.00 sec)
```

Zadanie 13

Zaokrąglić ceny towarów (tabela *product*) do liczb całkowitych. Użyć funkcji ROUND. Jako wynik powinniśmy otrzymać:

```
+-----+-----+
| Cena      | Cena zaokr. |
+-----+-----+
| 150.00    | 150         |
| 200.00    | 200         |
| 410.00    | 410         |
| 16.25     | 16          |
| 21.95     | 22          |
| 40.95     | 41          |
| 11.00     | 11          |
| 28.00     | 28          |
| 123.00    | 123         |
| 9.00      | 9           |
| 8.00      | 8           |
| 1669.00   | 1669        |
| 582.00    | 582         |
| 16.00     | 16          |
| 18.25     | 18          |
| 20.00     | 20          |
| 7.00      | 7           |
| 60.00     | 60          |
| 65.00     | 65          |
| 50.00     | 50          |
| 8.00      | 8           |
| 12.00     | 12          |
| 25.00     | 25          |
| 35.00     | 35          |
| 45.00     | 45          |
| 4.29      | 4           |
| 22.89     | 23          |
| 80.00     | 80          |
| 75.00     | 75          |
| 80.00     | 80          |
| 45.00     | 45          |
| 47.00     | 47          |
| 50.00     | 50          |
+-----+-----+
33 rows in set (0.00 sec)
```

Zadanie 14

Wyświetlić bieżącą datę oraz czas. W pierwszych dwóch kolumnach format wyświetlania jest domyślny, w trzech ostatnich ustalony przez nas. Jako wynik powinniśmy otrzymać:

```
+-----+-----+-----+-----+
| A          | B          | C          | D          | E          |
+-----+-----+-----+-----+
| 2006-02-01 | 14:34:36  | Wednesday-February-2006 | February, 1st 2006 | 14::34::36-PM |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Zadanie 15

Wyświetl ile dni minęło od dnia Twoich urodzin. Użyć funkcji DATEDIFF. Pamiętaj, że domyślnym formatem dla daty jest YYYY-MM-DD HH:MM:SS. Jako wynik powinniśmy otrzymać:

```
+-----+-----+
| current_date() | Zyję już tyle dni! |
+-----+-----+
| 2006-02-01    | 14251              |
+-----+-----+
1 row in set (0.00 sec)
```

Zadanie 16

Do bieżącej daty i godziny dodać 2 dni, 5 godziny oraz 8 minut. Użyć funkcji DATE_ADD. Jako wynik powinniśmy otrzymać:

```
+-----+-----+
| CURRENT_TIMESTAMP() | Po dadaniu          |
+-----+-----+
| 2006-02-01 14:54:27 | 2006-02-03 20:02:27 |
+-----+-----+
1 row in set (0.00 sec)
```

Zadanie 17

Wyliczyć, używając odpowiednich funkcji MySQL, „parametry” bieżącego dnia. Jako wynik powinniśmy otrzymać:

```
+-----+-----+
| 2006-02-01 | Dziś mamy 32 dzień roku, 5 tydzień roku oraz 4 dzień tygodnia. |
+-----+-----+
1 row in set (0.00 sec)
```