



Podstawy języka SQL

Ćwiczenia - część 1

Opracował: Andrzej Nowak

1. Zaloguj się na swój serwer **MySQL**. Utwórz bazę danych (nazwa dowolna – np.: **nasza**)

```
create database nasza;
```

2. Stwórz trzy tabele powiązane relacją o następującej budowie:

kraje						
id	kod	nazwa	waluta	populacja	stolica	kontynent
INT	CHAR(3)	VARCHAR(45)	CHAR(3)	BIGINT	VARCHAR(30)	CHAR(2)

jezyki	
kraj	jezyk
CHAR(3)	VARCHAR(10)

kontynenty	
kod	kontynent
CHAR(2)	VARCHAR(18)

Tabele są powiązane ze sobą relacjami:

Tabela **jezyki** w kolumnie **kraj** przechowuje kody krajów z kolumny **kod** w tabeli **kraje**:

```
jezyki.kraj → kraje.kod
```

Tabela **kraje** w kolumnie **kontynent** przechowuje kody kontynentów z kolumny **kod** w tabeli **kontynenty**:

```
kraje.kontynent → kontynenty.kod
```

3. Utwórz tabele:

```
CREATE TABLE kraje
(
  id INT AUTO_INCREMENT PRIMARY KEY,
  kod VARCHAR(3) NOT NULL,
  nazwa VARCHAR(45) NOT NULL,
  waluta CHAR(3) DEFAULT NULL,
  populacja BIGINT DEFAULT 0,
  stolica VARCHAR(30) DEFAULT NULL,
  kontynent CHAR(2) DEFAULT NULL
);
```

```
create table jezyki
(
kraj char(3) not null,
jezyk varchar(10) not null,
primary key(kraj,jezyk)
);
```

```
Create table kontynenty
(
Kod char(2) primary key,
Kontynent varchar(13) not null
);
```

Sprawdź czy tabele są utworzone w twojej bazie danych:

```
show tables;
```

```
+-----+
| Tables_in_base_nasza |
+-----+
| jezyki                |
| kontynenty            |
| kraje                 |
+-----+
```

Ćwiczenie 1

a. Wprowadź dane do tabeli kontynenty:

```
insert into kontynenty(kod,kontynent) values
('AF','Africa'),
('EU','Europe'),
('AS','Asia'),
('SA','South America'),
('NA','North America'),
('AN','Antarctica'),
('OC','Oceania');
```

b. Wyświetl zawartość tabeli kontynenty:

```
select * from kontynenty;
+-----+-----+
| kod | kontynent |
+-----+-----+
| AF  | Africa    |
| AN  | Antarctica|
| AS  | Asia      |
| EU  | Europe    |
| NA  | North America|
| OC  | Oceania   |
| SA  | South America|
+-----+-----+
```

Ćwiczenie 2

Dane dla pozostałych dwóch tabel wczytaj z pliku. Pliki – jezyki.txt i kraje.txt **prze użyciem** zapisz w swoim katalogu:

- a. Wpisz w **MySQL** polecenie importujące dane do tabeli:

```
load data infile 'ścieżka_do_twojego_katalogu/jezyki.txt' into table jezyki fields terminated by ',';
```

- b. Wyświetl zawartość tabeli:

```
select * from jezyki;
```

- c. Wczytaj tabelę kraje. Wpisz w mysql polecenie:

```
load data infile 'ścieżka_do_twojego_katalogu/kraje.txt' into table kraje fields terminated by ',';
```

- d. Wyświetl zawartość tabeli:

```
select * from kraje;
```

Ćwiczenie 3

Wyszukiwanie informacji w pojedynczej tabeli uzyskamy za pomocą polecenia:

```
SELECT kolumna_1,kolumna_2,...,kolumna_n FROM tabela WHERE warunek;
```

- a. Wyszukaj wszystkie kraje, w których walutą jest USD.

```
select nazwa, waluta from kraje where waluta='USD';
```

```
+-----+-----+
| nazwa                | waluta |
+-----+-----+
| American Samoa       | USD    |
| Bonaire               | USD    |
| Ecuador              | USD    |
| Micronesia           | USD    |
| Guam                 | USD    |
| British Indian Ocean Territory | USD    |
| Marshall Islands     | USD    |
| Northern Mariana Islands | USD    |
| Puerto Rico          | USD    |
| Palau                | USD    |
| El Salvador          | USD    |
| Turks and Caicos Islands | USD    |
| East Timor           | USD    |
| U.S. Minor Outlying Islands | USD    |
| United States        | USD    |
| British Virgin Islands | USD    |
| U.S. Virgin Islands  | USD    |
+-----+-----+
```

- b. W podobny sposób znajdź kraje korzystające z EUR, PLN, GBP.

- c. Jeszcze raz wyświetlmy kraje używające USD, lecz posortujmy je w porządku alfabetycznym - wynik zapytania posortuj wg wybranej kolumny rosnąco **ASC**

```
select nazwa, waluta from kraje where waluta='USD' order by nazwa ASC;
```

nazwa	waluta
American Samoa	USD
Bonaire	USD
British Indian Ocean Territory	USD
British Virgin Islands	USD
East Timor	USD
Ecuador	USD
El Salvador	USD
Guam	USD
Marshall Islands	USD
Micronesia	USD
Northern Mariana Islands	USD
Palau	USD
Puerto Rico	USD
Turks and Caicos Islands	USD
U.S. Minor Outlying Islands	USD
U.S. Virgin Islands	USD
United States	USD

d. Jeszcze raz wyświetlmy kraje używające GBP, lecz posortujmy je w porządku alfabetycznym Wynik zapytania posortuj wg wybranej kolumny malejąco DESC

select nazwa,waluta **from** kraje **where** waluta='GBP' **order by** nazwa **DESC**;

e. W wyniku zapytania zmień nazwę kolumny nazwa na Kraj, a waluta na Waluta

select nazwa **AS** Kraj,waluta **AS** Waluta **from** kraje **where** waluta='USD' **order by** nazwa **ASC**;

Kraj	Waluta
American Samoa	USD
Bonaire	USD
British Indian Ocean Territory	USD
British Virgin Islands	USD
East Timor	USD
Ecuador	USD
El Salvador	USD
Guam	USD
Marshall Islands	USD
Micronesia	USD
Northern Mariana Islands	USD
Palau	USD
Puerto Rico	USD
Turks and Caicos Islands	USD
U.S. Minor Outlying Islands	USD
U.S. Virgin Islands	USD
United States	USD

f. Wyszukaj wszystkie kraje leżące w Europie. Symbol Europy to EU:

```
select nazwa AS 'Kraj w Europie' from kraje where kontynent='EU' order by nazwa ASC;
```

g. W podobny sposób wyszukaj kraje w Azji (AS), Afryce (AF) i Ameryce Północnej (NA).

h. Wyszukajmy kraje, w których liczba ludności jest większa od 100000000. W wyniku podaj informację o nazwie kraju, populacji i nazwie kontynentu:

```
select nazwa AS Kraj, populacja AS Ludność, kontynent AS Kontynent from kraje where populacja>100000000 order by nazwa;
```

Niestety, w wyniku otrzymaliśmy jedynie kody kontynentów, a nie same nazwy kontynentów:

Kraj	Ludność	Kontynent
Bangladesh	156118464	AS
Brazil	201103330	SA
China	1330044000	AS
India	1173108018	AS
Indonesia	242968342	AS
Japan	127288000	AS
Mexico	112468855	NA
Nigeria	154000000	AF
Pakistan	184404791	AS
Russia	140702000	EU
United States	310232863	NA

Nazwy kontynentów są w innej tabeli. Aby otrzymać te nazwy w wyniku zapytania, musimy w instrukcji **SELECT** użyć dwóch tabel połączonych relacją.

Relacja ta jest taka, że kolumna kontynent w tabeli kraje przechowuje klucz podstawowy rekordu w tabeli kontynenty.

Ćwiczenie 4

a. Wyświetl wszystkie kraje z informacją o kontynencie, na którym się znajdują:

```
select nazwa AS Kraj, kontynenty.kontynent AS Kontynent from kraje, kontynenty where kraje.kontynent=kontynenty.kod order by nazwa;
```

Uwaga:

Ponieważ korzystamy w zapytaniu z dwóch tabel, to może się zdarzyć, że w tych tabelach istnieją kolumny o takich samych nazwach. W takim przypadku należy zawsze odwoływać się do kolumny poprzez nazwę tabeli, kropkę i nazwę kolumny w tej tabeli. Inaczej **MySQL** zgłosi błąd polegający na niejednoznaczności nazw kolumn. Oczywiście, najlepiej jest tak zaprojektować bazę danych, aby kolumny nie miały tych samych nazw.

b. Wyszukaj kraje, w których liczba ludności jest większa od 100000000. W wyniku podaj informację o nazwie kraju, populacji i **koniecznie** nazwie kontynentu. Aby uzyskać dodatkowo informację o nazwach kontynentów, należy w zapytaniu wykorzystać operator logiczny **AND**.

```
select nazwa AS Kraj, populacja AS Ludność, kontynenty.kontynent AS Kontynent from kraje, kontynenty where populacja>100000000 AND kraje.kontynent=kontynenty.kod order by nazwa;
```

Kraj	Ludność	Kontynent
Bangladesh	156118464	Asia
Brazil	201103330	South America
China	1330044000	Asia
India	1173108018	Asia
Indonesia	242968342	Asia
Japan	127288000	Asia
Mexico	112468855	North America
Nigeria	154000000	Africa
Pakistan	184404791	Asia
Russia	140702000	Europe
United States	310232863	North America

Ćwiczenia dodatkowe

1. Sprawdź, czy waluta 'EUR' jest używana w Azji.
2. Poszukaj krajów, gdzie nikt nie mieszka.
3. Wyszukaj kraje, w których używa się języka niemieckiego (`de`), francuskiego (`fr`) i odmian angielskiego (`LIKE 'en%'`).
4. Wyszukaj kraje w Azji, których nazwa stolicy ma jako drugą literę 'a'.



Podsumowanie poznanych poleceń SQL

Tabele w bazie danych

```
SHOW TABLES;
```

Wprowadzanie do tabeli wielu rekordów

```
INSERT INTO tabela (kolumna_1,kolumna_2,...,kolumna_n) VALUES  
(wartości kolumn w pierwszym rekordzie),  
(wartości kolumn w drugim rekordzie),  
...  
(wartości kolumn w ostatnim rekordzie);
```

Odczyt zawartości tabeli z pliku

```
LOAD DATA INFILE 'plik' INTO TABLE tabela FIELDS TERMINATED BY 'znak';
```

Wyświetlanie rekordów spełniających warunek

```
SELECT kolumna_1,kolumna_2,...,kolumna_n  
FROM tabela_1,tabela_2,...,tabela_n  
WHERE warunek ORDER BY kolumna ASC|DESC;
```

Odwoływanie się do nazw kolumn o tych samych nazwach w różnych tabelach

```
tabela.kolumna
```

Zmiana nazwy kolumny w wyniku zapytania

```
SELECT kolumna AS nazwa...
```

Przydatne operatory w warunkach WHERE

```
= <> != > < >= <=  
AND  
OR  
LIKE wzorzec  
NOT LIKE wzorzec  
!> – nie większy niż ...  
!< – nie mniejszy niż ...
```

wzorzec jest tekstem, w którym % oznacza dowolny ciąg znaków, a _ zastępuje jeden dowolny znak.

Na przykład, jeśli kolumna `imie` zawiera imiona, to warunek:

```
imie LIKE 'A%'
```

będzie prawdziwy dla imion na literę A. Z kolei warunek:

```
imie NOT LIKE 'A%'
```

będzie prawdziwy dla wszystkich imion, które nie rozpoczynają się literą A.