

Lekcja: Serwer SQL

1. W celu możliwości korzystania z baz danych (w tym z języku SQL) konieczne jest zaopatrzenie się w odpowiedni SZBD – System Zarządzania Bazą Danych. Na rynku istnieje wiele różnych SZBD, w różnych na dodatek wersjach. Najbardziej powszechny jest Serwer MS SQL, tworzony przez Microsoft. Zgodny jest ze standardem ANSI/ISO. Jest produktem komercyjnym, istnieją jednak jego darmowe wersje, niestety w znacznym stopniu ograniczone (brak graficznych narzędzi, określony limit połączeń z bazą, brak zautomatyzowanych zadań, np. archiwizacja, itd.)
2. Alternatywę dla powyższego produktu stanowią wolnodostępne, następujące systemy (każdy z nich ma swoje wady i zalety – MySQL jest dobry do nauki, PostgreSQL jest dobry jak na system darmowy, MSSQL jest komercyjny jednak dobrze radzi sobie w ogromnych bazami do miliona rekordów, Oracle jest podobna jak MSSQL, tzn. też dobrze radzi sobie z ogromnymi bazami, przeznaczona głównie dla profesjonalistów wspierana przez CIA):
 - a) Firebird – dostępny dla różnych platform (Linux, Windows, Mac), pojawił się w roku 2000, (ostatnia wersja stabilna 27 grudnia 2019) ciągle rozwijany
 - b) PostgreSQL – dostępny także dla różnych platform (Linux, Windows, Mac), pojawił się w roku 1995, (ostatnia stabilna wersja 13 lutego 2020), ciągle rozwijany
 - c) MySQL – dostępny dla różnych platform (Linux, Windows, Mac), pojawił się w 1995 roku, (ostatnia stabilna wersja 25 kwietnia 2019), ciągle rozwijany
3. Do celów dydaktycznych zostanie wykorzystane oprogramowanie XAMPP. Posiada on pakiet aplikacji umożliwiających nie tylko zarządzanie bazą danych, ale także zainstalowanie na komputerze klienta serwera za pomocą którego można tworzyć i testować strony WWW (obecność między innymi Apache, PHP, itp.)
4. Po poprawnym zainstalowaniu i uruchomieniu aplikacji XAMPP, należy uruchomić serwer Apache oraz serwer MySQL z poziomu panelu kontrolnego.
5. Celem przejścia do narzędzia phpMyAdmin pozwalającego na zarządzanie bazami danych należy wybrać opcję „Admin” w wierszu MySQL panelu kontrolnego. Użytkownik zostaje wówczas przeniesiony do narzędzia obsługiwanego z poziomu przeglądarki internetowej.
6. Z poziomu panelu graficznego phpMyAdmin użytkownik może wykonać szereg różnych operacji:
 - a) Tworzenie bazy danych – pierwsza czynność, którą należy wykonać. W okienku „Utworzenie bazy danych” należy wpisać nazwę bazy, która ma zostać stworzona, np. `zst`. Operację tworzenia bazy danych można też wykonać wykorzystując polecenie SQL:

```
CREATE DATABASE zst;
```
 - b) Tworzenie tabel – po utworzeniu bazy danych (rodzaj bazy danych wybiera się z okienka „phpMyAdmin” po lewej stronie, w opcji „Baza danych”), można przejść do tworzenia tabel (relacji) baz danych. W tym celu należy przejść do zakładki SQL. Z tego poziomu można wykonywać różne polecenia tworzenia, modyfikowania i operowania bazami danych (tworząc relacje w bazach danych można posłużyć się interfejsem graficznym jednak posiada on ograniczone możliwości, dlatego nie zaleca się korzystanie z tego narzędzia).

Tworząc relację bazy danych należy pamiętać o stworzeniu odpowiednich ograniczeń referencyjnych dla atrybutów oraz kluczy relacji.

Dla przykładu zostanie stworzona relacja „pracownik” z lekcji numer 21. W tym celu należy w zakładce SQL wpisać następujący skrypt (**UWAGA: aby stworzyć relację pracownik należy najpierw utworzyć relację firma, jako pierwszą relację bazy danych nie związaną z inną**):

Utworzenie relacji firma:

```
CREATE TABLE firma (  
  id_firma INT(3) NOT NULL,  
  nazwa VARCHAR(10) NOT NULL,  
  adres_m VARCHAR(10) NOT NULL,  
  adres_ul VARCHAR(10),  
  adres_nr VARCHAR(10) NOT NULL,  
  PRIMARY KEY (id_firma)  
) ENGINE=INNODB;
```

Utworzenie relacji pracownik:

```
CREATE TABLE pracownik (  
  id_pracownika INT(3) NOT NULL AUTO_INCREMENT,  
  imie VARCHAR(10) NOT NULL,  
  nazwisko VARCHAR(10) NOT NULL,  
  adres_miasto VARCHAR(10) NOT NULL,  
  adres_ulica VARCHAR(10),  
  adres_nr VARCHAR(10),  
  pesel INT(11),  
  telefon INT(9),  
  id_firmy INT(3) NOT NULL,  
  
  PRIMARY KEY (id_pracownika),  
  INDEX (id_firmy),  
  FOREIGN KEY (id_firmy) REFERENCES firma (id_firma)  
) ENGINE=INNODB;
```

Po nazwie atrybutu należy wpisać typ danej wraz z jej ograniczeniami. Do najważniejszych typów należą: VARCHAR (n) – określa ciąg znaków o zmiennej długości nie większej niż liczba n podana w nawiasie; INT (n) – określa liczbę o długości n cyfr; DATE – określa datę, która może zawierać się z przedziału 1.01.1000 do 31.12.9999. Warto także zauważyć, że w zależności od wersji SZBD niektóre typy mogą nie być jeszcze obsługiwane lub kryć się pod inną nazwą.

W końcu ograniczenia integralnościowe dotyczą głównie atrybutów, które muszą być wpisane do bazy danych (NOT NULL), lub nie muszą być wpisane. *Jeżeli dana nie musi być wpisana nie ma konieczności zapisywania tego w strukturze budulcowej relacji poleceniem „NULL”, a wówczas po wprowadzeniu rekordu z pustą daną do bazy, uzyskuje ona ukryty status NULL. Deklaracja „NOT NULL” nie musi gwarantować jednak niemożliwości wprowadzenia danych pustych, a wprowadzone rekordy nie uzyskują ukrytego statusu NULL.* Poza tym należy określić klucz podstawowy lub obce tabeli, jak powyżej.

Jedną z ważnych czynności, którą warto wykonać jest określenie typu danych w relacji (polski typ to „utf8_polish_ci” – warto ustawić go z poziomu zakładki graficznej). Na spójność bazy danych będą wpływać nie tylko ograniczenia integralnościowe, ale także obsługa transakcji (czyli operacja wykona się w całości lub wcale się nie wykona). W tym przypadku może to być typ BerkeleyDB (nie w pełni zintegrowany z MySQL) lub InnoDB.

- c) Wstawianie danych do tabeli – w celu wstawienia danej do tabeli należy wykonać przykładowe, następujące polecenia:

Wstawianie danych do tabeli firma (w pierwszej kolejności):

```
INSERT INTO firma (id_firma, nazwa, adres_m, adres_ul, adres_nr)
VALUES ('1','ABC', 'Leszno', 'Zielona','3/4');
```

W drugiej kolejności dopiero dane do pracownika (bo id_firmy w pracowniku odnosi się do id_firma w relacji firma):

```
INSERT INTO pracownik (id_pracownika, imie, nazwisko, adres_miasto,
adres_ulica, adres_nr, pesel, telefon, id_firmy)
VALUES ('001', 'Jan', 'Nowak', 'Leszno', 'Słowiańska', '23/3',
'1234567897', '655260001', '1');
```

- d) Modyfikacja danych – w przypadku wstawienia błędnych danych do tabeli można je zmodyfikować następującym poleceniem (założmy poprawę nazwiska Nowak na Nowicki):

```
UPDATE `pracownik` SET `nazwisko` = 'Nowicki'
WHERE `id_pracownika` = '001' ;
```

7. Pliki z bazą danych znajdują się w katalogu „C:\xampp\mysql\data” w folderze o nazwie jaką nadano bazie danych (np. zst). *Można je z tego miejsca dowolnie kopiować w celu szybkiego archiwizowania danych z bazy na bieżąco. Celem jednak wykonywania w pełni prawidłowej i funkcjonalnej kopii bazy danych należy do tego wykorzystać funkcje eksportu.*