

Instrukcja laboratoryjna 2	Bazy Danych
	Temat: Obsługa bazy danych za pomocą phpMyAdmin oraz phpPgAdmin
	Przygotował: mgr inż. Maciej Lasota

1) Wprowadzenie do phpMyAdmin oraz phpPgAdmin

phpMyAdmin jest darmowym narzędziem rozpowszechnianym na licencji GNU/GPL służącym do zarządzania bazą danych MySQL. Za pomocą phpMyAdmin możemy zarządzać zarówno lokalną jak i zdalną bazą danych MySQL. Do poprawnego działania wymaga zainstalowanego serwera HTTP wraz z obsługą PHP. Narzędzie to zapewnia realizację wszystkich podstawowych funkcji jak:

- Przeglądanie zawartości,
- Zarządzanie strukturą,
- Wykonywanie zapytań SQL,
- Przeszukiwanie rekordów,
- Dodawanie rekordów,
- Eksport,
- Import,
- Sprawdzanie,
- Analizowanie,
- Naprawianie,
- Optymalizowanie,
- Przeładowanie,
- Czyszczenie zawartości,
- Usuwanie tabeli.

Więcej informacji oraz pełną dokumentację do phpMyAdmin można znaleźć na stronie projektu <http://www.phpmyadmin.net/>.

phpPgAdmin jest darmowym narzędziem rozpowszechnianym na licencji GNU/GPL służącym do zarządzania bazą danych PostgreSQL. Za pomocą phpPgAdmin możemy zarządzać zarówno lokalną jak i zdalną bazą danych PostgreSQL. Do poprawnego działania wymaga zainstalowanego serwera HTTP wraz z obsługą PHP. Narzędzie to zapewnia realizację wszystkich podstawowych funkcji jak:

- Przeglądanie zawartości tabeli i jej rekordów
- Wyszukanie rekordu o zadanych kryteriach
- Dodawanie nowego rekordu
- Usunięcie wszystkich rekordów tabeli
- Usunięcie tabeli wraz z jej zawartością
- Dodanie kolejnych pól tabeli
- Zmianę nazwy i przestrzeni tabeli
- Import danych z zewnętrznego źródła
- Wykonanie kopii bazy danych w określonej formie

Więcej informacji oraz pełną dokumentację do phpPgAdmin można znaleźć na stronie projektu <http://phpPgAdmin.sourceforge.net/>.

2) Typy danych MySQL.

Typ	Opis
INT [(N)] [ZEROFILL SIGNED UNSIGNED AUTO_INCREMENT]	Liczba całkowita standardowa o maksymalnej ilości cyfr <i>N</i> ZEROFILL - wypełnia zerami liczby przed np. <i>INT(3) ZEROFILL</i> -> <i>008</i> , SIGNED - liczba ze znakiem (-2147483648 do +2147483647), UNSIGNED - liczba bez znaku (0 do 4294967295), AUTO_INCREMENT - automatyczne inkrementowanie liczby (N+1)
INTEGER [(N)] [ZEROFILL SIGNED UNSIGNED]	Liczba całkowita standardowa alias na typ <i>INT</i>
TINYINT [(N)] [ZEROFILL SIGNED UNSIGNED]	Liczba całkowita mała ilość cyfr <i>N</i> SIGNED - liczba ze znakiem (-128 do +127), UNSIGNED - liczba bez znaku (0 do 255)
SMALLINT [(N)] [ZEROFILL SIGNED UNSIGNED]	Liczba całkowita mała ilość cyfr <i>N</i> SIGNED - liczba ze znakiem (-32768 do +32767), UNSIGNED - liczba bez znaku (0 do 65535)
MEDIUMINT [(N)] [ZEROFILL SIGNED UNSIGNED]	Liczba całkowita średnia ilość cyfr <i>N</i> SIGNED - liczba ze znakiem (-8388608 do +8388607), UNSIGNED - liczba bez znaku (0 do 16777215)
BIGINT [(N)] [ZEROFILL SIGNED UNSIGNED]	Liczba całkowita duża ilość cyfr <i>N</i> SIGNED - liczba ze znakiem (-9223372036854775808 do +9223372036854775807), UNSIGNED - liczba bez znaku (0 do 18446744073709551615)
DECIMAL [(P, [S])] [ZEROFILL SIGNED UNSIGNED]	Liczba dziesiętna o precyzji P - całkowita ilość znaczących cyfr użyta do wyrażenia liczby (precyzja) jako P podawana jest wartość minimalna, S - całkowita ilość cyfr po kropce (skala)
DEC [(P, [S])] [ZEROFILL SIGNED UNSIGNED]	Liczba dziesiętna o precyzji alias na <i>DECIMAL</i> P - całkowita ilość znaczących cyfr użyta do wyrażenia liczby (precyzja) jako P podawana jest wartość minimalna, S - całkowita ilość cyfr po kropce (skala)
NUMERIC [(M, [S])] [ZEROFILL SIGNED UNSIGNED]	Liczba dziesiętna o precyzji P - całkowita ilość znaczących cyfr użyta do wyrażenia liczby (precyzja) jako P podawana jest wartość rzeczywista, S - całkowita ilość cyfr po kropce (skala)

FLOAT [(M,D)] [ZEROFILL SIGNED UNSIGNED]	Liczba zmiennoprzecinkowa (rzeczywista pojedynczej precyzji) M - ilość cyfr przed przecinkiem, D - ilość cyfr po przecinku
DOUBLE [(M,D)] [ZEROFILL SIGNED UNSIGNED]	Liczba zmiennoprzecinkowa (rzeczywista podwójnej precyzji) M - ilość cyfr przed przecinkiem, D - ilość cyfr po przecinku
REAL [(M,D)] [ZEROFILL SIGNED UNSIGNED]	Liczba zmiennoprzecinkowa alias na typ <i>FLOAT</i>
BIT [(N)]	Liczba binarna o maksymalnej ilości bitów <i>N</i> (1 do 64) np. BIT(6) -> 001001
DATE	Typ danych określający datę w formacie: YYYY-MM-DD YYYY - rok, możemy podać również w formacie YY, MM - miesiąc (1 - 12), DD - dzień (1 - 31) np. '2009-05-28'
TIME	Typ danych określający czas w formacie: HH-MM-SS HH - godzina (0 - 23), MM - minuta (1 - 59), SS - sekunda (1 - 59) np. '16:45:01'
TIMESTAMP	Typ danych przechowujący datę i czas zgodny ze standardem SQL-92 w formacie UTC: YYYY-MM-DD HH-MM-SS
DATETIME	Typ danych przechowujący datę i czas w formacie: YYYY-MM-DD HH-MM-SS
YEAR [(2 4)]	Typ określający rok w formacie YYYY lub YY np. YEAR(2) -> '02', YEAR(4) -> '2002'
CHAR (N) [CHARACTER SET nazwa] [COLLATE nazwa]	Typ danych przechowujący ciąg znaków o stałej długości <i>N</i> <i>N</i> - maksymalna długość ciągu znaków (0 do 255) w bajtach
VARCHAR (N) [CHARACTER SET nazwa] [COLLATE nazwa]	Typ danych przechowujący ciąg znaków o zmiennej długości <i>N</i> <i>N</i> - maksymalna długość ciągu znaków (0 do 65535) w bajtach
BINARY (N)	Typ danych przechowujący binarne ciągi danych o stałej długości <i>N</i> , <i>N</i> - maksymalna długość danych (0 do 255) w bajtach
VARBINARY (N)	Typ danych przechowujący binarne ciągi danych o zmiennej długości <i>N</i> , <i>N</i> - maksymalna długość danych (0 do 65535) w bajtach
BLOB [(N)]	Typ danych umożliwiający przechowywanie dużych obiektów binarnych (<i>Binary Large Object</i>), wyróżniamy kilka rodzajów typu danych BLOB: TINYBLOB , BLOB , MEDIUMBLOB , LONGBLOB , wszystkie typy różnią się jedynie maksymalną wielkością przechowywanych danych
TEXT [(N)] [CHARACTER SET nazwa] [COLLATE nazwa]	Typ danych umożliwiający przechowywanie dużych obiektów tekstowych wyróżniamy kilka rodzajów typu danych TEXT: TINYTEXT , TEXT , MEDIUMTEXT , LONGTEXT , wszystkie typy różnią się jedynie maksymalną wielkością przechowywanych danych
ENUM ('wartosc1', 'wartosc2', ...)	Typ danych wyliczeniowy może przechowywać wartości wybrane z danej listy, maksymalna ilość składowych listy wynosi 65535 elementów
SET ('wartosc1', 'wartosc2', ...)	Typ danych podobny do typu wyliczeniowego <i>ENUM</i> , może przechowywać zero lub więcej wartości wybranych z danej listy, maksymalna ilość składowych listy wynosi 64 elementy
SERIAL	Typ danych dla klucza głównego alias na typ <i>BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE</i>
BOOL , BOOLEAN	Typ logiczny alias na typ <i>TINYINT(1)</i>

3) Typy danych PostgreSQL.

Typ	Opis
SMALLINT	Liczba całkowita mała (2 bajty) ze znakiem z przedziału (-32768 do +32767)
INTEGER	Liczba całkowita standardowa (4 bajty) ze znakiem z przedziału (-2147483648 do +2147483647)
INT, INT4	Liczba całkowita standardowa (4 bajty) alias na typ <i>INTEGER</i>
BIGINT	Liczba całkowita duża (8 bajtów) ze znakiem z przedziału (-9223372036854775808 do +9223372036854775807)
INT8	Liczba całkowita duża (8 bajtów) alias na typ <i>BIGINT</i>
DECIMAL[(P, [S])]	Liczba dziesiętna o precyzji P - całkowita ilość znaczących cyfr użyta do wyrażenia liczby (precyzja) jako P podawana jest wartość minimalna, S - całkowita ilość cyfr po kropce (skala)
NUMERIC[(P, [S])]	Liczba dziesiętna o precyzji P - całkowita ilość znaczących cyfr użyta do wyrażenia liczby (precyzja) jako P podawana jest wartość rzeczywista, S - całkowita ilość cyfr po kropce (skala)
REAL	Liczba zmiennoprzecinkowa pojedynczej precyzji (4 bajty) (6 miejsc po przecinku)
FLOAT4	Liczba rzeczywista (4 bajty) alias na typ <i>REAL</i>
DOUBLE PRECISION	Liczba zmiennoprzecinkowa podwójnej precyzji (8 bajtów) (15 miejsc po przecinku)
FLOAT8	Liczba rzeczywista (8 bajtów) alias na typ <i>DOUBLE PRECISION</i>
SERIAL	Typ całkowity automatycznie inkrementowany (4 bajty) bez znaku z przedziału (1 do 2147483647)
SERIAL4	Typ całkowity automatycznie inkrementowany (4 bajty) alias na typ <i>SERIAL</i>
BIGSERIAL	Typ całkowity automatycznie inkrementowany (8 bajtów) bez znaku z przedziału (1 do 9223372036854775807)
SERIAL8	Typ całkowity automatycznie inkrementowany (8 bajtów) alias na typ <i>BIGSERIAL</i>
MONEY	Typ danych określający walutę (8 bajtów) w formacie: 00,000.000 np. '\$1,00.000'
CHARACTER VARYING[(N)]	Typ danych przechowujący ciąg znaków o zmiennej długości N N - maksymalna długość ciągu znaków w bajtach, nie podanie N oznacza <i>CHARACTER VARYING(1)</i> .
VARCHAR[(N)]	Typ danych przechowujący ciąg znaków o zmiennej długości N alias na typ <i>CHARACTER VARYING</i>
CHARACTER[(N)]	Typ danych przechowujący ciąg znaków o stałej długości N N - maksymalna długość ciągu znaków (0 do 126) w bajtach, nie podanie N oznacza <i>CHARACTER(1)</i>
CHAR[(N)]	Typ danych przechowujący ciąg znaków o stałej długości N alias na typ <i>CHARACTER</i>
TEXT	Typ danych umożliwiający przechowywanie bardzo dużych obiektów tekstowych
BYTEA	Typ danych przechowujący binarne ciągi danych o zmiennej długości
TIMESTAMP[(P)] [WITH TIME ZONE WITHOUT TIME ZONE]	Typ danych przechowujący datę i czas zgodny ze standardem SQL-92 w formacie UTC: YYYY-MM-DD HH-MM-SS , P - precyzja określa ilość liczb po przecinku dla sekund (0 do 6) WITH TIME ZONE - dodaje strefę czasową, WITHOUT TIME ZONE - nie dodaje strefy czasowej np. '2008-12-31 16:45:01', '2008-12-31 16:45:01+02' '2008-12-31 16:45:01 PST'

DATE	Typ danych określający datę w formatach: YYYY-MM-DD (standardowy), DD/MM/YYYY , MM/DD/YYYY YYYY – rok, możemy podać również w formacie YY, MM – miesiąc (1 – 12) lub w postaci January/Jan, DD – dzień (1 – 31) np. '2009-05-28', '2009-Jan-26'
TIME[(P)][WITH TIME ZONE WITHOUT TIME ZONE]	Typ danych określający czas w formacie: HH-MM-SS HH – godzina (0 – 23), MM – minuta (1 – 59), SS – sekunda (1 – 59) P – precyzja określa ilość liczb po przecinku dla sekund (0 do 6) lub (0 do 10) WITH TIME ZONE – dodaje strefę czasową, WITHOUT TIME ZONE – nie dodaje strefy czasowej np. '16:45:01' '04:05 AM' '16:45:06.4566' '12:32:12 PST'
INTERVAL[(P)]	Typ danych określający przedział czasowy między dwoma typami danych <i>TIMESTAMP</i> P – precyzja określa ilość liczb po przecinku dla sekund(0 do 6)
BIT[(N)]	Liczba binarna o maksymalnej ilości bitów N np. BIT(6) -> '001001'
BOOLEAN	Typ logiczny
BOOL	Typ logiczny alias na typ <i>BOOLEAN</i>
Typ wyliczeniowy	Typ wyliczeniowy w PostgreSQL należy wcześniej odpowiednio utworzyć za pomocą polecenia SQL: CREATE TYPE nazwaTypu AS ENUM ('wartosc1','wartosc2'); Typ wyliczeniowy może przechowywać wartości wybrane jedynie z uprzednio utworzonej listy
Typ tablicowy	Typ tablicowy można utworzyć w oparciu o typy proste np. INTEGER[12], REAL[10]
XML	Typ danych umożliwiający przechowywanie obiektów XML
BLOB	Typ danych umożliwiający przechowywanie dużych obiektów binarnych (<i>Binary Large Object</i>), istniał we wcześniejszych wersjach PostgreSQL poniżej wersji 6.0. Obecnie duże obiekty binarne można przechowywać z wykorzystaniem typu <i>BYTEA</i> oraz <i>OID</i> lub z użyciem specjalnego modułu <i>LO</i> oraz typu <i>LO (Large Object)</i>
POINT	Typ danych opisujący punkt na płaszczyźnie (X,Y)
LINE	Typ danych opisujący linię na płaszczyźnie ((X,Y),(X,Y))
LSEG	Typ danych opisujący odcinek linii na płaszczyźnie
BOX	Typ danych opisujący prostokąt/kwadrat na płaszczyźnie
PATH	Typ danych opisujący otwarty kształt geometryczny na płaszczyźnie
POLYGON	Typ danych opisujący zamknięty kształt geometryczny na płaszczyźnie
CIRCLE	Typ danych opisujący okrąg na płaszczyźnie
CIDR	Typ danych umożliwiający przechowywanie adresów sieci w formacie ADRES/MASKA zarówno IPv4 jak i IPv6 np. '192.168.0.0/32' '2001:4f8:3:ba::/64' '81.26.7.210/25'
INET	Typ danych umożliwiający przechowywanie adresów hostów i sieci w formacie ADRES/MASKA zarówno IPv4 jak i IPv6 np. '192.168.0.1' '2001:4f8:3:ba::' '81.26.7.208'

MACADDR	Typ danych umożliwiający przechowywanie adresów MAC np. '08002b:010203' '08002b-010203' '0800.2b01.0203' '08-00-2b-01-02-03' '08:00:2b:01:02:03'
----------------	---