

## Logowanie do MySQL

```
mysql [-h adres_hosta] [-u nazwa_uzytkownika] [-p twoje_haslo]
```

## Zarządzanie bazami:

```
CREATE DATABASE nazwa_bazy;
SHOW DATABASES;
USE nazwa_bazy;
SELECT DATABASE(); - wyświetla nazwę aktualnie wybranej bazy danych
DROP DATABASE nazwa_bazy;
```

## Zarządzanie tabelami:

### Tworzenie tabel

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] nazwa_tabeli [(definicje_kolumn ...)]
definicje_kolumn to lista nazw kolumn, jej tytuł i właściwości.
```

*Składnia pojedynczej linii definicji kolumn wygląda następująco:*

```
nazwa_kolumny typ [NOT NULL | NULL] [DEFAULT wartosc_domyslna] [AUTO_INCREMENT]
```

### Wyświetlanie tabel

```
SHOW TABLES;
DESC nazwa_tabeli;
```

### Modyfikatory typów kolumn

Typ kolumny	Opis pola
AUTO_INCREMENT	Atrybut ten pozwala na automatyczne powiększanie danego pola numerycznego. Pozwala to na tworzenie unikatowych indeksów. Wartość jest inkrementowana, jeśli podana wartość jest pusta lub wynosi 0.
BINARY	Atrybut pól CHAR i VARCHAR. Jego użycie oznacza, że dane pole ma rozróżniać wielkość liter.
DEFAULT wartość	Atrybut ten określa, jaka wartość ma zostać wstawiona, jeśli żadna nie zostanie podana.
NOT NULL	Atrybut ten pozwala tak zdefiniować pole, aby niemożliwe było wstawienie do niego wartości NULL
NULL	Atrybut ten określa, że dane pole może przyjmować wartość NULL
UNSIGNED	Atrybut ten dla pól całkowitych oznacza, że dane pole ma być nioznaczone
ZEROFILL	Atrybut ten pozwala na wypełnienie pola typu całkowitego zerami na początku, aż do osiągnięcia dopuszczalnej długości

## Tworzenie tabel

### Tworzenie prostej tabeli:

```
CREATE TABLE moja (  
-> imie VARCHAR (10),  
-> nazwisko VARCHAR (15),  
->email VARCHAR(25)  
->);
```

### Tworzenie tabeli jeśli taka nie istnieje:

```
CREATE TABLE IF NOT EXISTS moja (  
-> id int(3)  
-> );
```

### Tworzenie tabeli tymczasowej

```
CREATE TEMPORARY TABLE moja_temp (  
-> pole CHAR(4)  
-> );
```

### Tworzenie tabel z polami wyboru

```
CREATE TABLE moja3 (  
-> liczba ENUM (  
-> 'jeden',  
-> 'dwa'  
-> )  
-> );
```

```
CREATE TABLE moja3 (  
-> liczba SET (  
-> 'jeden',  
-> 'dwa'  
-> )  
-> );
```

### Tworzenie tabel z polami o domyślnych wartościach

```
CREATE TABLE moja5 (  
-> tekst VARCHAR(5) DEFAULT 'brak'  
-> );
```

### Tworzenie tabel z polami bez wartości NULL

```
CREATE TABLE moja6 (  
-> pole_1 INT(2),  
-> pole_2 INT(2) NOT NULL  
-> );
```

Drugi pole w przypadku brak wartości  
przyjme wartość 0

### Tworzenie tabel z polami nieoznaczonymi

```
CREATE TABLE moja7 (  
-> liczba_1 INT(5),  
-> liczba_2 INT(5) UNSIGNED  
-> );
```

nieoznaczona czyli przyjmuje wartości od 0

### Tworzenie tabel z polami liczbowymi uzupełnianymi zerami

```
CREATE TABLE moja9 (  
-> liczba_1 INT(5),  
-> liczba_2 INT(5) ZEROFILL
```

### Tworzenie tabel z kluczem inkrementowanym

```
CREATE TABLE moja10 (  
-> id INT(3) AUTO_INCREMENT PRIMARY KEY,  
-> dane VARCHAR(5)  
-> );
```

opcja ta wymaga aby pole było zdefiniowane też jako klucz

### Tworzenie tabel z indeksami

```
CREATE TABLE moja12 (  
-> imie VARCHAR(10),  
-> nazwisko VARCHAR(15),  
-> rok TINYINT(1),  
-> INDEX(rok)  
-> );
```

Jeśli tabela w zamierzeniu ma zawierać wiele rekordów i wiesz, według jakich kolumn najczęściej będziesz przeszukiwał tę tabelę, warto wtedy taką kolumnę, czy też kolumny, zdefiniować jako indeks. Indeks na dwa pola INDEX(imie, rok).

### Tworzenie tabel z indeksami unikalnymi

```
CREATE TABLE moja15 (  
-> pesel INT(11),  
-> imie VARCHAR(10),  
-> nazwisko VARCHAR(15),  
-> UNIQUE (pesel)  
-> );
```

Czasami możesz potrzebować utworzyć takie pole, które będzie przyjmowało tylko nie powtarzające się wartości. np. numer PESEL.

## Edycja tabel

### Zmiana nazwy tabeli:

```
ALTER TABLE nazwa_tabeli RENAME nowa_nazwa;
```

### Usuwanie tabeli

```
DROP TABLE nazwa_tabeli;
```

### Usuwanie kolumny

```
ALTER TABLE nazwa_tabeli DROP [COLUMN] nazwa_kolumny;
```

### Dodawanie kolumny (na końcu tabeli)

```
ALTER TABLE nazwa_tabeli ADD [COLUMN] definicje_kolumny; np.  
ALTER TABLE moja ADD COLUMN adres VARCHAR(25);
```

### **Dodawanie kolumny jako pierwszej**

ALTER TABLE nazwa\_tabeli ADD [COLUMN] definicje\_kolumn FIRST; np.  
ALTER TABLE moja ADD COLUMN id INT(3) FIRST;

### **Dodawanie kolumny po jakimś polu**

ALTER TABLE nazwa\_tabeli ADD [COLUMN] definicje\_kolumn AFTER nazwa\_kolumn; np.  
ALTER TABLE moja ADD COLUMN telefon VARCHAR(12) AFTER email;

### **Modyfikowanie danych kolumny (w tym również nazwy kolumny)**

ALTER TABLE nazwa\_tabeli CHANGE [COLUMN] nazwa\_kolumny definicje\_kolumn;  
ALTER TABLE moja CHANGE COLUMN telefon telefon\_sluzbowy VARCHAR(25);

### **Modyfikowanie danych kolumny (modyfikacja typu pola)**

ALTER TABLE nazwa\_tabeli MODIFY [COLUMN] nazwa\_kolumny definicje\_kolumn; np.  
ALTER TABLE moja MODIFY COLUMN telefon CHAR(8);

### **Dodawanie indeksu**

ALTER TABLE nazwa\_tabeli ADD INDEX [nazwa] (kolumna, kolumna2..) np.  
ALTER TABLE moja ADD INDEX (id);

### **Dodawanie indeksu unikalnego**

ALTER TABLE nazwa\_tabeli ADD UNIQUE [nazwa] (kolumna, kolumn2); np.  
ALTER TABLE moja ADD UNIQUE (email);

### **Usuwanie indeksów**

ALTER TABLE nazwa\_tabeli DROP INDEX nazwa\_klucza; np.  
ALTER TABLE moja DROP INDEX email;

### **Dodawanie wartości domyślnej**

ALTER TABLE nazwa\_tabeli ALTER [COLUMN] nazwa\_kolumny SET DEFAULT wartosc\_domyslna;  
ALTER TABLE moja ALTER COLUMN email SET DEFAULT 'brak';

### **Usuwanie wartości domyślnej**

ALTER TABLE nazwa\_tabeli ALTER [COLUMN] nazwa\_kolumny DROP DEFAULT;  
ALTER TABLE moja ALTER COLUMN email DROP DEFAULT;

## Zarządzanie zawartością bazy danych

### Dodawanie rekordu

```
INSERT [LOW_PRIORITY | DELAYED] [INTO] nazwa_tabeli [(kolumna, ...)] VALUES (wartosc, ...);
```

```
INSERT INTO moja1 VALUES(1, 'Jan', 'Kowalski', 'jan@kowalski.pl', 'ul. Kowala 5');
```

Lub

```
INSERT [LOW_PRIORITY] [INTO] nazwa_tabeli SET nazwa_kolumny=wartosc, ...;
```

```
INSERT INTO moja1 SET id=2, imie='Józef', nazwisko='Nowak',  
adres_email='jozef@nowak.pl', adres='Dworcowa 15';
```

### Dodawanie rekordu tylko z niektórymi polami

```
INSERT INTO moja1 (id, imie) VALUES (3, 'Radosław');
```

## Wybieranie rekordów

```
SELECT [DISTINCT | ALL] kolumna1, kolumna2, .. FROM nazwa_tabeli ;
```

### Proste wybieranie rekordu

```
SELECT id, imie, nazwisko, adres_email, adres FROM moja1;      lub  
SELECT * FROM moja1;
```

### Wybieranie rekordu tylko z niektórymi polami

```
SELECT imie, adres_email FROM moja1;
```

## Używanie warunków

### Wyrażenie AND i OR

```
SELECT * FROM moja1 WHERE id=2;
```

```
SELECT * FROM moja1 WHERE id=2 OR id=3;
```

```
SELECT * FROM moja1 WHERE id>1;
```

```
SELECT * FROM moja1 WHERE id>=2;
```

### Wyrażenie IN (gdzie id jest równe 2 lub 3)

```
SELECT * FROM moja1 WHERE id IN(2, 3);
```

**Wyrażenie NOT IN (gdzie id jest różne od 2 lub 3)**

```
SELECT * FROM moja1 WHERE id NOT IN(2, 3);
```

**Wyrażenie BETWEEN (gdzie określona pola mieszczą się w podanym zakresie)**

```
SELECT * FROM moja1 WHERE id BETWEEN 2 AND 3;
```

**Porównanie LIKE**

   - znak podkreślenia oznacza dokładnie jeden dowolny znak

% - znak procentu oznacza dowolny ciąg znaków

*Pobierz gdzie występuje @*

```
SELECT * FROM moja1 WHERE adres_email LIKE '@%';
```

**Porównanie NOT LIKE**

*Pobierz gdzie imie nie zaczyna się od litery 'J'*

```
SELECT * FROM moja1 WHERE imie NOT LIKE 'J%';
```

**Wybieranie tylko unikalnych rekordów**

```
SELECT DISTINCT imie FROM moja1;
```

*(pobierze tylko raz imie np. imie Jan gdyby występowało więcej razy)*

**Wybieranie rekordów z wielu tabel równocześnie**

```
SELECT moja1.imie, moja_praca.miejsce_pracy, moja_praca.zawod FROM moja1,  
moja_praca WHERE moja1.id = moja_praca.id;
```

**Sortowanie wyników**

```
SELECT imie, nazwisko FROM moja1 ORDER BY imie;
```

```
SELECT imie, nazwisko FROM moja1 ORDER BY imie, nazwisko;
```

```
SELECT id FROM moja1 ORDER BY id DESC;
```

**Porcjowanie wyników zapytania**

```
SELECT * FROM moja1 WHERE id>0 LIMIT 2;
```

LIMIT od\_wiersza, ile\_wierszy

```
SELECT * FROM moja1 WHERE id>0 LIMIT 0, 2;
```

```
SELECT * FROM moja1 WHERE id>0 LIMIT 1, 2; (poczynając od rekordu 2)
```

### Modyfikacja rekordów

UPDATE nazwa\_tabeli SET kolumna=wartość, kolumna2=wartosc2 ... warunek;

UPDATE moja1 SET adres = 'brak\_danych'; *(zamieni wszystkie na brak danych)*

UPDATE moja1 SET adres = 'nowy adres' WHERE id = 2;

UPDATE moja1 SET adres = 'adres' WHERE imie = 'Jan';

### Usuwanie rekordów

DELETE FROM nazwa\_tabeli warunek;

#### Usuwanie wszystkich rekordów

DELETE FROM moja1;

#### Usuwanie wybranych rekordów

DELETE FROM moja\_praca WHERE id=3;

DELETE FROM moja\_praca WHERE id IN (1, 4); *usuń o id równych 1 i 4*

### Funkcje operujące na zbiorach danych

#### AVG – średnia wartość z podanego zakresu rekordów

SELECT AVG(id) FROM moja1;

#### COUNT(nazwisko) – zwraca liczbę rekordów o wartości różnej od NULL

#### COUNT(\*) – zwraca liczbę wszystkich rekordów w tabeli

#### MAX(id) – zwraca największą wartość spośród podanych w wyrażeniu

#### MIN(id) – zwraca najmniejszą wartość spośród podanych w wyrażeniu

#### CURDATE() – zwraca aktualną datę

#### CURTIME() – zwraca aktualny czas

#### DAYNAME('2002-08-15'); - zwraca nazwę dnia tygodnia dla daty np. Tuesday

#### DAYOFMONTH('2002-08-15') – zwraca dzień miesiąca dla daty np. tu 15

#### MONTH(' ') – zwraca miesiąc z podanej daty

#### MONTHNAME('') – zwraca nazwę miesiąca z podanej daty

#### DAYOFWEEK('2002-08-15') – zwraca numer dnia tygodnia

WEEK('2002-08-15') – zwraca numer tygodnia dla daty 2002-08-15

DAYOFYEAR('2002-08-15') – zwraca dzień roku dla danej daty

YEAR('2002-08-15') – zwraca rok z daty ...

HOUR('09:45:36') – zwraca godzinę z podanego czasu

MINUTE – zwraca minuty z podanego czasu

NOW() – zwraca bieżącą datę i czas

QUARTER(' 2002-08-15') – zwraca numer kwartału dla podanej daty

FLOOR(4.78) – zwraca największą liczbę całkowitą mniejszą lub równą podanej (tu 4)

FORMAT(5.47894, 2) – formatuje podaną liczbę do określ. liczby miejsc po przecinku

LAST\_INSERT\_ID – zwraca wartość ostatnio wprowadz. wart. AUTO INCREMENT

LCASE/LOWER – zwraca ciąg, w którym wszystkie wielkie litery zamienia na małe

UCASE/UPPER – zwraca ciąg, w którym wszystkie małe litery zamienia na wielkie

LEFT('to jest tekst',7) – zwraca określoną ilość znaków, zaczynając od lewej strony

RIGHT('to jest tekst',7) – zwraca określoną ilość znaków, zaczynając od prawej strony

LENGTH('to jest tekst') – zwraca długość podanego ciągu

LOCATE ('SQL', Baza MySQL i SQL') – z. pozycję podanego tekstu w danym ciągu

SUBSTRING('To jest prosty tekst', 9, 6) – zwraca ciąg, który znajduje się w źródłowym ciągu na określonej pozycji i ma określoną długość

PASSWORD('tekst') – zwraca podany ciąg w formie zaszyfrowanej

RAND() – zwraca losową liczbę z przedziału od 0 do 1

REPEAT('MySQL', 5) – zwraca podany ciąg powielony określoną ilość razy

REPLACE('to jest baza', 'baza', 'MySQL') – zwraca nowy ciąg, podane słowo zostaje zamienione

ROUND(3.567, 2) – zwraca zaokrągloną liczbę z określoną ilością miejsc po przecinku

RTRIM ('TEKST ') – zwraca ciąg znaków, usuwa końcowe spacje

SIGN(-8) – zwraca -1, 0, 1 gdy ujemne, zero, czy dodatnie

SPACE(10) – zwraca podaną liczbę spacji

TRIM(' tekst '), TRIM(BOTH FROM ' tekst ') – usuwa z ciągu początkowe i końcowe spacje. (możliwe modyfikatory aby usuwać tylko spacje lewe lub prawe)

USER() – zwraca nazwę obecnie zalogowanego użytkownika

VERSION() – zwraca numer wersji naszej bazy MySQL



1. Zaloguj się do bazy danych
2. Utwórz tabelę moja

Field	Type	Null	Key	Default	Extra
id	int(3)	YES	MUL	NULL	
imie	varchar(10)	YES		NULL	
nazwisko	varchar(15)	YES		NULL	
adres_email	varchar(25)	YES	MUL	NULL	
adres	varchar(25)	YES		NULL	

3. Dodaj do tabeli moja następujące rekordy

id	imie	nazwisko	adres_email	adres
1	Jan	Kowalski	jan@kowalski.pl	ul. Kowalskiego 5
2	Józef	Nowak	jozef@nowak.pl	ul. Dworcowa 15
3	Radosław	NULL	NULL	NULL

4. Pobierz z tabeli moja wszystkie kolumny z rekordu o wartości w polu id równym 2

id	imie	nazwisko	adres_email	adres
2	Józef	Nowak	jozef@nowak.pl	ul. Dworcowa 15

5. Pobierz z tabeli moja wszystkie kolumny z rekordu o wartości w polu równym 2 lub 3

id	imie	nazwisko	adres_email	adres
2	Józef	Nowak	jozef@nowak.pl	ul. Dworcowa 15
3	Radosław	NULL	NULL	NULL

6. Pobierz za pomocą wyrażenia IN te rekordy z tabeli moja, których id jest równe 2 lub 3.

7. Pobierz za pomocą wyrażenia NOT IN te rekordy z tabeli moja, których id jest różne od 2 lub 3

8. Pobierz za pomocą wyrażenia BETWEEN te rekordy z tabeli moja, których id jest większe lub równe od 2 i mniejsze lub równe 3.

9. Pobierz z tabeli moja te rekordy, które w polu adres\_email zawierają znak @. Wykorzystaj do tego celu porównanie LIKE

10. Pobierz z tabeli moja te rekordy, w których pole imie nie zaczyna się od litery J.

11. Dodaj do tabeli moja osobę: imie: Jan, nazwisko: Kot, adres: ul. Zielona 12

12. Wyświetl rekordy, a następnie wyświetl rekordy używając słowa kluczowego DISTINCT.

13. Utwórz tabelę o nazwie moja\_praca i umieść w niej kolumny

Field	Type	Null	Key	Default	Extra
id	int(3)		PRI	NULL	auto_increment
miejsce_pracy	Enum('dom', 'biuro', 'autobus')	YES		NULL	
zawod	varchar(12)	YES		NULL	

## 14. Wypełnij tabelę danymi

id	miejsce_pracy	zawod
1	dom	informatyk
2	autobus	kierowca
3	biuro	manager

15. Pobierz z tabeli moja i moja\_praca informacje z kolumn imie, miejsce\_pracy, zawod

16. Pobierz z tabeli moja liste imion i nazwisk i posortuj rekordy według imienia

17. Pobierz z tabeli moja1 listę imion i nazwisk i posortuj rekordy według imienia i nazwiska.

18. Pobierz z tabeli moja dwa rekordy, których numer id jest większy od zera.

19. Pobierz z tabeli moja dwa rekordy, których numer id jest większy od zera poczynając od rekordu drugiego.

1. Zaloguj się do mysql
2. Utwórz tabelę pracownicy

Field	Type	Null	Key	Default	Extra
imie	varchar(30)	YES		NULL	
nazwisko	varchar(30)	YES		NULL	
data_urodzenia	date	YES		1950-01-01	
adres	varchar(25)	YES		NULL	
placa	decimal(10,2)	YES		1000.00	

## 3. Wypełnij tabelę danymi

imie	nazwisko	Data_urodzenia	adres	Placa
Aleksander	Kowalski	2002-07-20	Kwiatowa 8, Poznań	1200.00
Aniela	Kwiatkowska	NULL	NULL	NULL
Krzysztof	Borowiecki	1952-08-06	ul. Wycieczkowa 8/12, Wrocław	1500.34
Izabela	Michałkowska	1970-05-23	Al. Kwiatowa 15, Krakow	854.29
Zofia	Kowalska	2002-07-02	Kwiatowa 8, Poznan	1200.00

4. Zmień imie Kowalskiej na Maria (UPDATE)

5. Wyświetl osoby zarabiające powyżej 1000

6. Wyświetl osoby których nazwisko rozpoczyna się na literę K

7. Wyświetl osoby których nazwisko rozpoczyna się na kowalsk ....

8. Wyświetl osoby których placa jest większa od 500 mniejsza od 1000 lub nazywają się kowalski

9. Wyświetl posortowane wyniki (sortowanie według nazwisk – rosnąco i malejąco)

10. Wyświetl tylko 2 wiersze poczynając od rekordu 3

1. Załoguj się do zainstalowanej bazy danych.
2. Wyświetl pomoc dotyczącą monitora MySQL.
3. Wyjdź z monitora MySQL.
4. Utwórz dwie nowe bazy danych. Nazwij je *baza* i *moja\_baza*.
5. Wyświetl listę baz danych dostępnych w systemie.
6. Usuń bazę *moja\_baza*, posługując się poleceniem DROP DATABASE. Następnie wyświetl listę dostępnych baz.
7. Utwórz tabelę o nazwie **moja**. Niech tabela ta zawiera trzy kolumny: *imię*, *nazwisko* i *email*. Tabela ta będzie reprezentowała taką strukturę, jak bazy z rozdziału 1. Kolumny niech będą typu znakowego o zmiennej długości (VARCHAR).
8. Utwórz tabelę o nazwie **moja2**. Niech ta tabela zawiera trzy kolumny: *id*, *imie* i *opis*. Kolumna *id* niech będzie polem liczb całkowitych o długości 3 znaków, *imie* – polem znakowym o zmiennej długości do 10 znaków, a *opis* – polem binarnym BLOB.
9. Spróbuj utworzyć tabelę o dowolnej jednej kolumnie o nazwie już istniejącej, na przykład **moja**. Spójrz, jaki komunikat błędu system zwróci. Następnie dodaj w poleceniu CREATE TABLE opcję IF NOT EXISTS i zwróć uwagę na wynik. Aby potwierdzić, że na liście tabel nic się nie zmieniło, wylistuj tabelę dostępne w bazie *baza*.
10. Utwórz tabelę tymczasową o nazwie *moja\_temp* o dowolnej jednej kolumnie. Na liście tabel tabela tymczasowa nie jest wyświetlana. Aby sprawdzić, czy dana tabela rzeczywiście została utworzona, możesz posłużyć się poleceniem DESCRIBE, które wymaga podania nazwy tabeli. Polecenie to pokazuje strukturę danej tabeli. Następnie wyloguj się z MySQL-a i załoguj ponownie, próbując wyświetlić opis dla tabeli *moja\_temp*.
11. Utwórz tabelę o nazwie *moja3*. Niech tabela ta zawiera jedną kolumnę typu ENUM z dopuszczalnymi wartościami jeden, dwa i trzy. Następnie wyświetl strukturę tej tabeli.
12. Utwórz tabelę o nazwie *moja4*. Niech tabela ta zawiera jedną kolumnę typu SET z dopuszczalnymi wartościami jeden, dwa i trzy. Następnie wyświetl strukturę tej tabeli.
13. Utwórz tabelę o nazwie *moja5*. Niech tabela ta zawiera jedną kolumnę typu VARCHAR z dopuszczalną długością 5 znaków i domyślną wartością ustawioną na brak.
14. Utwórz tabelę *moja6* o dwóch kolumnach. Niech obie będą typu całkowitego, z tym że druga niech nie przyjmuje wartości NULL.
15. Utwórz tabelę *moja8* o dwóch kolumnach. Niech obie będą typu całkowitego z tym, że druga niech będzie nieoznaczona.
16. Utwórz tabelę *moja9* o dwóch kolumnach. Niech obie będą typu całkowitego z tym, że druga niech będzie wypełniona zerami.
17. Utwórz tabelę *moja10* o dwóch kolumnach. Niech pierwsza będzie polem liczb całkowitych o długości 3 znaków z opcją autoinkrementacji. Druga kolumna natomiast niech będzie innym dowolnym polem.
18. Utwórz tabelę *moja12* o trzech kolumnach. Niech pierwsza i druga będą polami znakowymi o zmiennej długości. Trzecia kolumna natomiast niech będzie polem liczb całkowitych z założonym indeksem. Tabela ta będzie minitabelą studentów. W pierwszych dwóch kolumnach będzie imię i nazwisko, natomiast w trzeciej rok, na którym uczy się student.
19. Utwórz tabelę *moja14* z polami imie, nazwisko i rok z założonymi indeksami na pola imie i rok.
20. Utwórz tabelę *moja15* z polami pesel, imie, nazwisko. Niech pole pesel będzie polem unikalnym.

21. Dodaj do tabeli moja nową kolumnę typu znakowego o nazwie adres.
22. Dodaj do tabeli moja dwie nowe kolumny. Pierwszą kolumnę typu całkowitego nazwij id. Kolumnę tę dodaj na samym początku tabeli. Niech drugą kolumną umieszczoną po kolumnie email będzie telefon.
23. Dodaj do tabeli moja indeks na kolumnę id.
24. Dodaj do tabeli moja indeks unikalny na kolumnę z adresem email.
25. Dodaj do tabeli moja wartość domyślną dla pola email.
26. Usuń z tabeli moja wartość domyślną dla pola email.
27. Zmień nazwę tabeli email na adres\_email. Typy pola pozostaw bez zmian.
28. Zmień typ kolumny telefon na typ znakowy CHAR o długości 8 znaków. Nazwę pola pozostaw bez zmian.
29. Usuń kolumnę telefon z tabeli moja.
30. Usuń indeks założony w tabeli moja na kolumnie adres\_email.
31. Zmień nazwę tabeli moja na moja1.
32. Usuń tabelę moja7 i wyświetl aktualną listę tabel w bazie baza.