# **Praktyka**

## **Praca w trybie tekstowym**

Logowanie się przez XAMPP -> **Shell**

|  |
| --- |
| mysql -u nazwa\_użytkownika -p |

Logowanie się przez **CMD**

|  |
| --- |
| **cd** C:\xampp\mysql\bin  mysql -u root -p |

Wylogowanie się

|  |
| --- |
| **exit** |
| Ctrl+c |

Zmiana kodowania znaków w **CMD** i **Shell’u** (przed zalogowaniem się)

|  |  |
| --- | --- |
| **chcp** 65001 | UTF 8 |
| **chcp** 852 | OEM Latin II |

Ustawianie polskich znaków w **Shell’u**

|  |  |
| --- | --- |
| Krok | Opis |
| 1 | Zmiany w pliku my.ini   1. XAMPP -> Config (ten w tej samej linijce co MySQL) -> my.ini 2. Usunięcie jednej linijki, tej zaznaczonej na czerwono |
| 2 | Ustawienie odpowiedniego kodowania znaków za pomocą (Zmiana dotyczy tylko tej jednej sesji)  **chcp** 852 *LUB* (lepiej 852) **chcp** 85001 |
| 3 | Zalogowanie się do mysql |

Wejście do bazy danych

|  |
| --- |
| **USE** nazwa\_bazy\_danych; |

Importowanie bazy danych

|  |
| --- |
| **SOURCE** ścieżka\_dostępu; |

Eksportowanie bazy danych (przed zalogowaniem się, wyeksportowana baza będzie w C:/xampp)

|  |
| --- |
| **mysqldump** -u root -p nazwa\_bazy\_danych > nazwa\_pliku.sql |

## **Klasa 3**

### **PDF 1 – Podstawy**

Sprawdzanie wersji bazy danych

|  |
| --- |
| \s |
| **SHOW** **VARIABLES** **LIKE** '%version%'; |

Wyświetlenie baz danych

|  |
| --- |
| **SHOW** **DATABASES** |

Tworzenie bazy danych

|  |
| --- |
| **CREATE** **DATABASE** nazwa\_bazy\_danych; |

Utworzenie bazy danych z kodowaniem znaków

|  |
| --- |
| **CREATE** **DATABASE** nazwa\_bazy **DEFAULT** **CHARACTER** **SET** utf8 **COLLATE** utf8\_unicode\_ci; |

Usunięcie bazy danych

|  |
| --- |
| **DROP** **DATABASE** nazwa\_bazy\_danych; |

Zmiana kodowania znaków bazy danych

|  |
| --- |
| **ALTER** **DATABASE** nazwa\_bazy\_danych **DEFAULT** **CHARACTER** **SET** utf8 **COLLATE** utf8\_unicode\_ci; |

### **PDF 2 – Uprawnienia**

Tworzenie nowego użytkownika

|  |
| --- |
| **GRANT** przywileje [kolumny]  **ON** nazwa\_bazy\_danych/nazwa\_tabeli/\* (żadna baza)/\*.\* (wszystkie bazy)  **TO** identyfikator\_użytkownika (login) **IDENTIFIED** **BY** 'hasło';  [**WITH** **GRANT** **OPTION**]; |

Usunięcie użytkownika

|  |
| --- |
| **DROP** **USER** 'identyfikator użytkownika'; |
| **DELETE FROM** mysql.user **WHERE USER=**'identyfikator\_użytkownika'; |

Nadanie uprawnień

|  |
| --- |
| **GRANT** przywileje  **ON** obiekt  **TO** identyfikator\_użytkownika; |

Odebranie uprawnień

|  |
| --- |
| **REVOKE** przywileje [kolumny]  **ON** obiekt  **FROM** identyfikator\_użytkownika; |

Zmiana lub utworzenie hasła

|  |
| --- |
| **SET** **PASSWORD** **FOR** 'nazwa\_użytkownika'= **PASSWORD**('hasło'); |

### **PDF 3 – Tabele**

Tworzenie tabeli

|  |
| --- |
| **CREATE** **TABLE** nazwa\_tabeli  (  Nazwa\_kolumny1 typ\_danych atrybuty,  Nazwa\_kolumny2 typ\_danych atrybuty  ); |

Tworzenie tabeli w której klucz główny składa się z dwóch kolumn

|  |
| --- |
| **CREATE** **TABLE** nazwa\_tabeli  (  Nazwa\_kolumny1 typ\_danych atrybuty,  Nazwa\_kolumny2 typ\_danych atrybuty,  **PRIMARY** **KEY**(nazwa\_kolumny1, nazwa\_kolumny2)  ); |

Tworzenie tabeli z kolumną bez wartości pustych

|  |
| --- |
| **CREATE** **TABLE** nazwa\_tabeli  (  Nazwa\_kolumny1 typ\_danych **NOT** **NULL**  ); |

Tworzenie tabeli z kolumną z wartością domyślną

|  |
| --- |
| **CREATE** **TABLE** nazwa\_tabeli  (  Nazwa\_kolumny1 typ\_danych **DEFAULT** ' '  ); |

Tworzenie tabeli bez duplikatów danych

|  |
| --- |
| **CREATE** **TABLE** nazwa\_tabeli  (  Nazwa\_kolumny1 typ\_danych **UNIQUE**  ); |

Tworzenie tabeli z kolumną która automatycznie zwiększa wartość

|  |
| --- |
| **CREATE** **TABLE** nazwa\_tabeli  (  Nazwa\_kolumny1 typ\_danych **AUTO\_INCREMENT** **PRIMARY** **KEY**  ); |

### **PDF 4 – Wyświetlanie informacji w bazie danych**

Wyświetlanie baz tabel w obecnej bazie danych

|  |
| --- |
| **SHOW** **TABLES;** |

Wyświetlenie nazw kolumn i informacji o strukturze tabeli

|  |
| --- |
| **SHOW COLUMNS FROM** nazwa\_tabeli; |
| **DESCRIBE** nazwa\_tabeli; |
| **DESC** nazwa\_tabeli; |

Wyświetlanie nazw kolumn w tabeli za pomocą SELECT

|  |
| --- |
| **SELECT** COLUMN\_NAME  **FROM** INFORMATION\_SCHEMA.COLUMNS  **WHERE** TABLE\_NAME = 'nazwa\_tabeli' |

### **PDF 5 – Modyfikacje tabeli**

Dodanie do tabeli klucza głównego

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **ADD** **CONSTRAINT** nazwa **PRIMARY** **KEY** (nazwa\_kolumny); |

Usunięcie z tabeli klucza głównego

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **DROP** **PRIMARY** **KEY;** |

Dodanie do tabeli klucza głównego który składa się kilku kolumn

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **ADD** **CONSTRAINT** nazwa **PRIMARY** **KEY** (nazwa\_kolumny1, nazwa\_kolumny2); |

Dodanie do tabeli nowej kolumny

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **ADD** **COLUMN** nazwa\_kolumny typ danych; |

Dodanie do tabeli nowej kolumny na pierwszym miejscu w tabeli

|  |
| --- |
| **ALTER TABLE** nazwa\_tabeli  **ADD** **COLUMN** nazwa\_kolumny typ\_danych **FIRST** |

Dodanie do tabeli nowej kolumny za określoną kolumną

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **ADD** **COLUMN** nazwa\_kolumny typ\_danych **AFTER** nazwa\_kolumny |

Usunięcie kolumny z tabeli

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **DROP** **COLUMN** nazwa\_kolumny; |

Zmiana nazwy kolumny w tabeli

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **CHANGE** **COLUMN** obecna\_nazwa\_kolumny nowa\_nazwa\_kolumny typ danych; |

Ustawienie wartości domyślnej w kolumnie

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **ALTER** **COLUMN** nazwa\_kolumny **SET** **DEFAULT** 'wartość\_domyślna'; |

Usunięcie wartości domyślnej z kolumny

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **ALTER** **COLUMN** nazwa\_kolumny **DROP** **DEFAULT;** |

Zmodyfikowanie tabeli tak aby dane w kolumnie nie mogły się powtarzać

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **ADD** **CONSTRAINT** nazwa **UNIQUE** (nazwa\_kolumny); |

Usunięcie atrybutu UNIQUE z kolumny

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **DROP** **CONSTRAINT** nazwa; |

Dodanie atrybutu NOT NULL do kolumny

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **MODIFY** nazwa\_kolumny typ\_danych **NOT** **NULL**; |

Usunięcie atrybutu NOT NULL z kolumny

|  |
| --- |
| **ALTER TABLE** nazwa\_tabeli  **MODIFY** nazwa\_kolumny typ\_danych **NULL;** |

Zmiana nazwy tabeli

|  |
| --- |
| **ALTER** **TABLE** obecna\_nazwa\_tabeli **RENAME** **TO** nowa\_nazwa\_tabeli; |

### **PDF 6 – Indeksy i wprowadzanie danych**

Utworzenie indeksu na jednej kolumnie w tabeli

|  |
| --- |
| **CREATE** **INDEX** nazwa\_indexu **ON** nazwa\_tabeli (nazwa\_kolumny); |

Utworzenie indeksu na dwóch kolumnach w tabeli

|  |
| --- |
| **CREATE** **INDEX** nazwa\_indexu **ON** nazwa\_tabeli (nazwa\_kolumny1, nazwa\_kolumny2); |

Utworzenie tabeli z indeksem

|  |
| --- |
| **CREATE** **TABLE** nazwa\_tabeli  (  Nazwa\_kolumny1 typ\_danych,  Nazwa\_kolumny2 typ\_danych,  **INDEX** nazwa\_indeksu (Nazwa\_kolumny)  ); |

Usunięcie indeksu

|  |
| --- |
| **DROP** **INDEX** nazwa\_indeksu **ON** nazwa\_tabeli; |
| **DROP INDEX** nazwa\_indeksu; |

Wprowadzenie danych do tabeli, w określonej kolejności kolumn

|  |
| --- |
| **INSERT** **INTO** nazwa\_tabeli (nazwa\_kolumny1, nazwa\_kolumny2, nazwa, kolumny3) **VALUES**  (wartość\_dla\_kolumny1, wartość\_dla\_kolumny2, wartość\_dla\_kolumny3); |
| **INSERT INTO** nazwa\_tabeli **SET**  nazwa\_kolumny1='wartość\_dla\_tej\_kolumny',  nazwa\_kolumny2='wartość\_dla\_tej\_kolumny',  nazwa\_kolumny3='wartość\_dla\_tej\_kolumny'; |

Wprowadzenie danych do tabeli, w domyślnej kolejności kolumn

|  |
| --- |
| **INSERT** **INTO** nazwa\_tabeli **VALUES**  (wartość\_dla\_kolumny1, wartość\_dla\_kolumny2, wartość\_dla\_kolumny3); |
| **INSERT INTO** nazwa\_tabeli **VALUES**  (pierwsza\_wartość\_dla\_kolumny1, pierwsza\_wartość\_dla\_kolumny2, pierwsza\_wartość\_dla\_kolumny3),  (druga\_wartość\_dla\_kolumny1, druga\_wartość\_dla\_kolumny2, druga\_wartość\_dla\_kolumny3),  (trzecia\_wartość\_dla\_kolumny1, trzecia\_wartość\_dla\_kolumny2, trzecia\_wartość\_dla\_kolumny3); |

### **PDF 7 – Modyfikowanie danych**

Zmiana rekordów

|  |
| --- |
| **UPDATE** nazwa\_tabeli  **SET** nazwa\_kolumny\_1 = wyrażenie, nazwa\_kolumny\_2 = wyrażenie…  [**WHERE** warunek]  [**ORDER** **BY** kryteria\_porządkowania]  [**LIMIT** ilość]; |

Zmienienie wartości w określonej kolumnie, we wszystkich rekordach

|  |
| --- |
| **UPDATE** nazwa\_tabeli  **SET** nazwa\_kolumny = nowa\_wartość; |

Zmiana wartości tylko w jednym rekordzie

|  |
| --- |
| **UPDATE** nazwa\_tabeli  **SET** nazwa\_kolumny = wartość  **WHERE** jakiś\_id = wartość; |

Zmiana typu danych w kolumnie

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **MODIFY** nazwa\_kolumny nowy\_typ\_danych; |

Usunięcie rekordów z określonego przedziału

|  |
| --- |
| **DELETE** **FROM** nazwa\_tabeli  **WHERE** jakiś\_id **BETWEEN** wartość\_1 **AND** wartość\_2; |

### **PDF 8 – Select**

Wyświetlenie rekordów z tabeli

|  |
| --- |
| **SELECT** nazwa\_kolumny  **FROM** nazwa\_tabeli; |

Wyświetlenie wszystkich rekordów z tabeli

|  |
| --- |
| **SELECT** \*  **FROM** nazwa\_tabeli; |

Wyświetlenie rekordów z określonym ID

|  |
| --- |
| **SELECT** nazwa\_kolumny  **FROM** nazwa\_tabeli  **WHERE** jakiś\_id = warunek |

### **PDF 9 – JOIN**

Wyświetlenie danych z kilku tabel (iloczyn kartezjański)

|  |
| --- |
| **SELECT** \*  **FROM** nazwa\_tabeli\_1, nazwa\_tabeli\_2, …; |

Powiązanie ze sobą kilku tabel

|  |
| --- |
| **SELECT** tabela\_1.nazwa\_kolumny1,  tabela\_1.nazwa\_kolumny2,  tabela\_2.nazwa\_kolumny1,  tabela\_2.nazwa\_kolumny2  **FROM** Tabela\_1, Tabela\_2  **WHERE** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id |

Złączenie typu CROSS JOIN (iloczyn kartezjański)

|  |
| --- |
| **SELECT** tabela\_1.nazwa\_kolumny1,  tabela\_1.nazwa\_kolumny2,  tabela\_2.nazwa\_kolumny1,  tabela\_2.nazwa\_kolumny2  **FROM** Tabela\_1 CROSS JOIN Tabela\_2  **WHERE** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id |

Złączenie typu INNER JOIN

|  |
| --- |
| **SELECT** tabela\_1.nazwa\_kolumny1,  tabela\_1.nazwa\_kolumny2,  tabela\_2.nazwa\_kolumny1,  tabela\_2.nazwa\_kolumny2  **FROM** Tabela\_1 **INNER** **JOIN** Tabela\_2  **ON** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id |

Złączenie typu LEFT OUTER JOIN

|  |
| --- |
| **SELECT** tabela\_1.nazwa\_kolumny1,  tabela\_1.nazwa\_kolumny2,  tabela\_2.nazwa\_kolumny1,  tabela\_2.nazwa\_kolumny2  **FROM** Tabela\_1 **LEFT** **OUTER** **JOIN** Tabela\_2  **ON** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id |

Złączenie typu RIGHT OUTER JOIN

|  |
| --- |
| **SELECT** tabela\_1.nazwa\_kolumny1,  tabela\_1.nazwa\_kolumny2,  tabela\_2.nazwa\_kolumny1,  tabela\_2.nazwa\_kolumny2  **FROM** Tabela\_1 **RIGHT** **OUTER** **JOIN** Tabela\_2  **ON** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id |

Złączenie typu FULL OUTER JOIN

|  |
| --- |
| **SELECT** tabela\_1.nazwa\_kolumny1,  tabela\_1.nazwa\_kolumny2,  tabela\_2.nazwa\_kolumny1,  tabela\_2.nazwa\_kolumny2  **FROM** Tabela\_1 **LEFT** **JOIN** Tabela\_2  **ON** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id  **UNION**  **SELECT** tabela\_1.nazwa\_kolumny1,  tabela\_1.nazwa\_kolumny2,  tabela\_2.nazwa\_kolumny1,  tabela\_2.nazwa\_kolumny2  **FROM** Tabela\_1 **RIGHT** **JOIN** Tabela\_2  **ON** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id |

### **PDF 10 – JOIN**

Złączenie z warunkiem

|  |
| --- |
| **SELECT** tabela\_1.nazwa\_kolumny1,  tabela\_1.nazwa\_kolumny2  tabela\_2.nazwa\_kolumny1,  tabela\_2.nazwa\_kolumny2  **FROM** Tabela\_1 **LEFT** **JOIN** Tabela\_2  **ON** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id  **WHERE** warunek |

Złączenie z użyciem aliasów

|  |
| --- |
| **SELECT** Alias\_1.nazwa\_kolumny1,  Alias\_1.nazwa\_kolumny2,  Alias\_2.nazwa\_kolumny1,  Alias\_2.nazwa\_kolumny2  **FROM** Tabela\_1 **AS** Alias\_1 **LEFT** **JOIN** Tabela\_2 **AS** Alias\_2  **ON** Tabela\_1.jakiś\_id = Tabela\_2.jakiś\_id |

### **PDF 11 – Funkcje Agregujące**

Policzenie ilości wierszy w tabeli

|  |
| --- |
| **SELECT** **COUNT**(nazwa\_kolumny)  **FROM** nazwa\_tabeli;  **WHERE** warunek; |

Sprawdzenie najwyższej wartości w tabeli

|  |
| --- |
| **SELECT** **MAX**(nazwa\_kolumny)  **FROM** nazwa\_tabeli; |

Sprawdzenie najmniejszej wartości w tabeli

|  |
| --- |
| **SELECT** **MIN**(nazwa\_kolumny)  **FROM** nazwa\_tabeli; |

Policzenie sumy wartości w kolumnie

|  |
| --- |
| **SELECT** **SUM**(nazwa\_kolumny)  **FROM** nazwa\_tabeli; |

Wyświetlenie wszystkich wartości z określonej kolumny

|  |
| --- |
| **SELECT** **GROUP\_CONCAT**(nazwa\_kolumny)  **FROM** nazwa\_tabeli; |

Wyświetlenie wszystkich wartości (unikatowe, posortowane rosnąco, „ ; ” jako separator) z określonej kolumny

|  |
| --- |
| **SELECT** **GROUP\_CONCAT**(DISTINCT nazwa\_kolumny ORDER BY ASC|DESC SEPARATOR „; ”)  **FROM** nazwa\_tabeli; |

## **Klasa 4**

### **PDF 9 - DELETE**

Tworzenie tabeli z kluczem obcym

|  |
| --- |
| **CREATE** **TABLE** nazwa\_tabeli  (  Nazwa\_kolumny\_1 Typ\_danych Atrybuty,  Nazwa\_kolumny\_2 Typ\_danych Atrybuty,  Nazwa\_kolumny\_3 Typ\_danych Atrybuty,  **FOREIGN** **KEY** (nazwa\_kolumny\_z\_tej\_tabeli) **REFERENCES** nazwa\_innej\_tabeli(nazwa\_kolumny) ) |

Dodanie klucza obcego do istniejącej tabeli

|  |
| --- |
| **ALTER** **TABLE** nazwa\_tabeli  **ADD** **FOREIGN** **KEY** (nazwa\_kolumny) **REFERENCES** nazwa\_innej\_tabeli(nazwa\_kolumny) |

### **PDF 10 – Tworzenie Widoków**

Ogólna składnia tworzenia widoków

|  |
| --- |
| **CREATE** **VIEW** nazwa\_widoku **AS**  **SELECT** … **FROM** nazwa\_tabeli  **WHERE** …; |

Wyświetlenie danych z kilku kolumn w jednej kolumnie

|  |
| --- |
| **SELECT** **CONCAT**(nazwa\_kolumny\_1, „”, nazwa\_kolumny\_2)  **FROM** nazwa\_tabeli; |

Wyświetlenie widoków

|  |
| --- |
| **SHOW** **TABLES**; |

Usuwanie widoków

|  |
| --- |
| **DROP** **VIEW** nazwa\_widoku; |

Modyfikacja widoków

|  |
| --- |
| **CREATE** **OR** **REPLACE** **VIEW** nazwa\_widoku **AS**  **SELECT** …  **FROM** nazwa\_tabeli; |

# **Teoria**

## **Uprawnienia**

**Uprawnienia użytkowników**

|  |  |  |
| --- | --- | --- |
| Uprawnienie | Poziom | Opis |
| SELECT | Tabele, kolumny | Pozwala wyszukać rekordy w tabelach |
| INSERT | Tabele, kolumny | Pozwala wstawiać nowe wiersze w tabelach |
| UPDATE | Tabele, kolumny | Pozwala zmieniać wartości zapisane w tabeli |
| DELETE | Tabele | Pozwala usuwać wiersze z tabeli |
| INDEX | Tabele | Pozwala tworzyć i usuwać indeksy w tabelach |
| ALTER | Tabele | Pozwala zmienić strukturę istniejących tabel, np. dodawanie kolumn, zmiany nazw kolumn i tabel, zmiany typów danych w kolumnach |
| CREATE | Bazy danych, tabele | Pozwala tworzyć nowe tabele i bazy danych |
| DROP | Bazy danych, tabele | Pozwala usuwać bazy lub tabele |

**Uprawnienia administratorów**

|  |  |
| --- | --- |
| Uprawnienie | Opis |
| CREATE TEMORARY TABLES | Pozwala administratorowi używać słowo kluczowe TEMPORARY w instrukcji CREATE TABLE |
| File | Pozwala wczytywać dane z plików do tabel i odwrotnie |
| LOCK TABLES | Pozwala jawnie używać instrukcji LOCK TABLES |
| PROCESS | Pozwala śledzić procesy wykonywane przez serwer i je przerywać |
| RELOAD | Pozwala powtórnie załadować tabele zwierające informacje na temat praw dostępu oraz na odświeżenie przywilejów, listy nazw łączących się komputerów, dziennika zdarzeń i tabel |
| REPLICATION CLIENT | Pozwala używać instrukcję SHOW STATUS na nadawcach i odbiorcach replikacji |
| REPLICATION SLAVE | Pozwala serwerom będącym odbiorcami replikacji łączyć się z serwerem nadawcą |
| SHOW DATABASES | Pozwala odczytywać listę wszystkich baz danych przy użyciu instrukcji SHOW DATABASES. Użytkownicy, którzy nie mają tego uprawnienia, mogą zobaczyć tylko bazy, do których przydzielono im dostęp |
| SHUTDOWN | Pozwala zakończyć pracę serwera MySQL |
| SUPER | Pozwala zabijać wątki, należące do dowolnego użytkownika |

**Przywileje specjalne**

|  |  |
| --- | --- |
| Uprawnienie | Opis |
| ALL | Nadaje wszystkie uprawnienia |
| USAGE | Nie nadaje żadnych uprawnień. Powoduje zarejestrowanie użytkownika i pozwala mu na zalogowanie się, lecz jakiekolwiek czynności są dla niego niedostępne. |

## **Atrybuty kolumn:**

* **NOT NULL** – w każdym rekordzie kolumny z tym atrybutem, musi znajdować się jakaś wartość
* **NULL** – w każdym rekordzie kolumny z tym atrybutem, nie musi znajdować się jakaś wartość, pojawi się wtedy wartość NULL
* **AUTO\_INCREMENT** – automatycznie doda to w tej kolumnie unikalny identyfikator, stosowana w kolumnach które są kluczem głównym
* **PRIMARY** **KEY** – kolumna jest kluczem głównym tabeli, wartości muszą być unikalne (MySQL automatycznie indeksuje takie kolumny)

## **Typy danych**

**Możemy wyróżnić typy liczbowe:**

* **Liczby** **całkowite** - na przykład 2, 5, -2, 233, itp. Ten typ można zawęzić do liczb nieujemnych - **UNSIGNED**. W zestawieniu liczb całkowitych pokazanych w tabeli, zakres po słowie lub odpowiada zastosowaniu tego właśnie atrybutu
* **Liczby** **zmiennoprzecinkowe** - są to liczby rzeczywiste, przy czym jest możliwość zadeklarowania liczby cyfr po przecinku (a dokładnie po kropce, którą będziemy stosować zamiast przecinka). W tabelach poniżej, liczba ta jest oznaczona symbolem D. Maksymalnie parametr D może przyjąć wartość 30, przy czym nie może być większy od M - 2. Skąd wzięła się liczba 2? Stąd, że liczbę wszystkich znaków musimy pomniejszyć o znak kropki i przynajmniej jeden znak liczby całkowitej

**Liczby całkowite**

|  |  |  |  |
| --- | --- | --- | --- |
| Typ | Zakres liczb | Ilość zajmowanych bajtów w pamięci | Opis |
| TINYINT()  BIT()  BOOL() | -127 – 128  0 – 255 | 1 | Bardzo małe liczby całkowite |
| SMALLINT() | -32768 32767  0 65535 | 2 | Małe liczby całkowite |
| MEDIUMINT() | -8388608 8388607  0..16777215 | 3 | Średnie liczby całkowite |
| INT()  INTEGER() | -231 – 231-1  0 – 232-1 | 4 | Liczby całkowite |
| BIGINT() | -263 – 263-1  0 – 264-1 | 8 | Duże liczby całkowite |

**Liczby zmiennoprzecinkowe**

|  |  |  |  |
| --- | --- | --- | --- |
| Typ | Zakres liczb | Ilość zajmowanych bajtów w pamięci | Opis |
| FLOAT(precyzja) | Zależne od precyzji | Różne | Liczby zmiennoprzecinkowe o pojedynczej lub podwójnej precyzji |
| FLOAT() | ±1.175494351E-38 ±3.402823466E+38 | 4 | Liczby zmiennoprzecinkowe pojedynczej precyzji. Ten typ jest równoznaczny z FLOAT(4), pozwala dodatkowo określić liczbę wyświetlanych cyfr i liczbę cyfr po przecinku |
| DOUBLE()  DOUBLE PRECISION()  REAL() | ±1.7976931348623157E-308 ±2.2250738585072014E+308 | 8 | Liczby zmiennoprzecinkowe podwójnej precyzji. Ten typ jest równoznaczny z FLOAT(8), pozwala dodatkowo określić liczbę wyświetlanych cyfr i liczbę cyfr po przecinku |
| DECIMAL()  NUMERIC()  DEC()  FIXED() | Różny | M+2 | Liczba zmiennoprzecinkowa, przechowywana jako zmienna typu CHAR. Zakres zależy od wartości M - liczby wyświetlanych znaków. |

**Typy daty i czasu**

|  |  |  |
| --- | --- | --- |
| Typ | Zakres | Opis |
| DATE | 1000-01-01 do 9999-12- 31 | Data wyświetlana w formacie RRRR-MM-DD. Punktem odniesienia jest 1 stycznia 1000 roku |
| TIME | -838:59:59 do 838:59:59 | Czas w formacie GG:MM:SS |
| DATETIME | 1000-01-01 00:00:00 do 9999-12-31 23:59:59 | Połączenie daty i czasu wyświetlane w formacie RRRR-MM-DD GG:MM:SS |
| TIMESTAMP(M) | 1970-01-01 do roku 2037 | Typ szczególnie przydatny do śledzenia transakcji. Format wyświetlania zależy od parametru M, górny zakres zależy od systemu operacyjnego |
| YEAR(2|4) | 70-69 (1970-2069) lub 1901-2155 | Rok pokazywany w formie dwu- lub czterocyfrowej, przy czym formy mają różne zakresy lat. |

**Format wyświetlanie wartości typu TIMESTAMP**

|  |  |
| --- | --- |
| Typ | Format wyświetlania |
| TIMESTAMP | RRRRMMDDGGMMSS |
| TIMESTAMP(14) | RRRRMMDDGGMMSS |
| TIMESTAMP(12) | RRMMDDGGMMSS |
| TIMESTAMP(10) | RRMMDDGGMM |
| TIMESTAMP(8) | RRRRMMDD |
| TIMESTAMP(6) | RRMMDD |
| TIMESTAMP(4) | RRMM |
| TIMESTAMP(2) | RR |

**Typy łańcuchowe je podzielić na 3 grupy:**

* Klasyczne łańcuchy (stringi) znaków
  + **CHAR** - łańcuchy o stałej długości
  + **VARCHAR** - łańcuchy o zmiennej długości
* Typy **TEXT** do przechowywania długich tekstów oraz BLOB dla dużych danych binarnych
* Typy specjalne **SET** do określonego zbioru wartości oraz **ENUM** - typ wyliczeniowy, gdzie można wybrać jedną wartość spośród określonego zbioru wartości

**Zwykłe typy łańcuchowe**

|  |  |  |
| --- | --- | --- |
| Typ | Zakres | Opis |
| [NATIONAL] CHAR() [BINARY|ASCII|UNICODE] | 1 – 255 znaków | Łańcuch znaków stałej długości M, gdzie M może przyjmować wartości od 1 do 255. Słowo kluczowe NATIONAL, wymusza zastosowanie domyślnego zbioru znaków. BINARY wyłącza rozpoznawanie wielkości liter (domyślnie wielkość liter jest rozpoznawana). ASCII lub UNICODE definiują używany w kolumnie zestaw znaków. |
| CHAR | 1 | Synonim typu CHAR(1) |
| [NATIONAL] VARCHAR() [BINARY] | 1 – 255 | Łańcuch znaków różnej długości. Pozostałe słowa kluczowe, jak wyżej |

**Typy TEXT (nie rozróżnia wielkości liter) i BLOB (rozróżnia wielkość liter, może przechowywać każdy typ danych)**

|  |  |  |
| --- | --- | --- |
| TYP | Maksymalna długość | Opis |
| TINYBLOB | 28 - 1 (co daje 255) | Mały obiekt BLOB |
| TINYTEXT | 28 - 1 (co daje 255) | Krótkie pole tekstowe |
| BLOB | 216 - 1 (co daje 65535) | Zwykły obiekt BLOB |
| TEXT | 216 - 1 (co daje 65535) | Zwykłe pole tekstowe |
| MEDIUMBLOB | 224 - 1 (co daje 16777215) | Średni obiekt BLOB |
| MEDIUMTEXT | 224 - 1 (co daje 16777215) | Pole tekstowe średniej długości |
| LONGBLOB | 232 - 1 (co daje 4294967295) | Duży obiekt BLOB |
| LONGTEXT | 232 - 1 (co daje 4294967295) | Duże pole tekstowe |

**SET** – typ zbiorowy, służy do zawężenia wartości danych, które mogą być zapisywane w danej kolumnie, do zdefiniowanego zbioru wartości. Zbiór wartości może zawierać co najwyżej 64 elementy

**ENUM** – typ wyliczeniowy, tym różni się od typu zbiorowego, że kolumny tego typu mogą zawierać tylko jedną spośród zdefiniowanego zbioru wartości lub wartość NULL. Zbiór dopuszczalnych wartości może zawierać do 65535 elementów

**Typy specjalne SET i ENUM**

|  |  |  |
| --- | --- | --- |
| Typ | Maksymalna ilość wartości w zbiorze | Opis |
| SET('wartość1', 'wartość2'…) | 64 | W kolumnie tego typu, może się znajdować podzbiór  zbioru wartości dopuszczalnych lub NULL |
| ENUM('wartość1', 'wartość2'…) | 65535 | W kolumnie tego typu, może się znajdować podzbiór  zbioru wartości dopuszczalnych lub NULL |

## **Kwerendy**

Operatory porównania, stosowane w klauzuli WHERE

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Nazwa | Przykład | Opis |
| = | Równy | Klient\_id = 5 | Sprawdza czy obie wartości są sobie równe |
| > | Większy od | Wartość > 10.00 | Sprawdza czy pierwsza wartość jest większa od drugiej |
| < | Mniejszy od | Wartość < 10.00 | Sprawdza czy pierwsza wartość jest mniejsza od drugiej |
| >= | Większy lub równy | Wartość >= 10.00 | Sprawdza czy pierwsza wartość jest większa lub równa drugiej |
| <= | Mniejszy lub równy | Wartość <= 10.00 | Sprawdza czy pierwsza wartość jest mniejsza lub równa drugiej |
| != lub <> | Różny | Wartość !=66.30 | Sprawdza czy dwie wartości są różne |
| IS NULL |  | Adres **IS NULL** | Sprawdza czy pole jest puste |
| IS NOT NULL |  | Adres **IS NOT NULL** | Sprawdza czy pole posiada wartość |
| BETWEEN |  | Cena **BETWEEN** 10.00 **AND** 50.00 | Sprawdza czy wartość należy do zbioru wartości |
| IN |  | Miejscowość **IN** ('Rzeszów', 'Warszawa') | Sprawdza czy wartość należy do podanego zbioru wartości |
| NOT IN |  | Miejscowość **NOT** **IN** ('Rzeszów', 'Warszawa') | Sprawdza czy wartość nie należy do podanego zbioru wartości |
| LIKE |  | Nazwisko **LIKE** 'Kowalski %' | Sprawdza czy wartość pasuje do określonego wzoru |
| NOT LIKE |  | Nazwisko **NOT** **LIKE** 'Kowalski %' | Sprawdza czy wartość nie pasuje do określonego wzorca |

## **Funkcje Agregujące**

|  |  |
| --- | --- |
| Nazwa | Opis |
| COUNT | Zwraca liczbę wartości występujących w określonej kolumnie |
| SUM | Zwraca sumę wartości występujących w określonej kolumnie |
| AVG | Zwraca średnią wartości występujących w określonej kolumnie |
| MIN | Zwraca najmniejszą wartość występującą w określonej kolumnie |
| MAX | Zwraca największą wartość występującą w określonej kolumnie |