

## WYBRANE FUNKCJE MYSQL

### Funkcje znakowe

#### **CHAR\_LENGTH**(str)

Określa długość tekstu *str*, wynikiem jest liczba znaków.

**Przykład:** `SELECT CHAR_LENGTH('MySQL') -> 5`

*Uwaga:* parametr funkcji może być wartością stałą lub nazwą kolumny w tabeli.

#### **CONCAT**(str1, str2, ...)

Łączy teksty *str1*, *str2*, ... w jeden ciąg znaków, wynikiem jest połączony ciąg znaków

**Przykład:** `SELECT CONCAT('My', 'S', 'Q', 'L') -> 'MySQL'`

*Uwaga 1:* jeden z tekstów jest pusty (NULL) zwracana wartość jest również pusta (NULL).

*Uwaga 2:* parametry funkcji mogą być stałymi wartościami tekstowymi (umieszczonymi w apostrofach) lub nazwami kolumn w tabeli.

#### **CONCAT\_WS**(separator, str1, str2, ...)

Działa jak funkcja **CONCAT**, dodatkowo łączone teksty są rozdzielone *separator*em.

**Przykład:** `SELECT CONCAT_WS('-', 'My', 'S', 'Q', 'L') -> 'My-S-Q-L'`

### Funkcje daty i czasu

#### **CURRENT\_DATE**() , **CURDATE**()

Określa aktualną datę na podstawie zegara systemowego, zwraca datę jako tekst (w formacie ‘YYYY-MM-DD’) lub jako liczbę (w formacie YYYYMMDD) zależnie od kontekstu użycia.

**Przykłady** (polecenie zostało wykonane 20. maja 2020 roku):

`SELECT CURDATE() -> '2020-05-20'`

`SELECT CURDATE()+0 -> 20200520`

#### **CURRENT\_TIME**() , **CURTIME**()

Określa aktualny czas na podstawie zegara systemowego, zwraca czas jako tekst (w formacie ‘HH:MM:SS’) lub jako liczbę (w formacie HHMMSS) zależnie od kontekstu użycia.

**Przykłady** (polecenie zostało wykonane o godzinie 10:33:05):

`SELECT CURTIME() -> '10:33:05'`

`SELECT CURTIME()+0 -> 103305`

#### **DATE\_ADD**(dt, INTERVAL liczba jednostki)

Dodaje do daty *dt* określoną *liczbę jednostek*, wynikiem jest powiększona data.

Dostępne *jednostki*: `MICROSECOND`, `SECOND`, `MINUTE`, `HOURL`, `DAY`, `WEEK`, `MONTH`, `QUARTER`, `YEAR`

**Przykład:** `SELECT DATE_ADD('2020-04-20', INTERVAL 31 DAY) -> '2020-05-21'`

*Uwaga:* parametry funkcji mogą być wartościami stałymi lub nazwami kolumn w tabeli.

**DATE\_DIF**(dt1, dt2)

Odejmuje datę dt2 od dt1, wynikiem jest różnica w dniach.

**Przykład:** `SELECT DATEDIFF('2020-05-20', '2020-04-20')` -> 30

*Uwaga:* parametry funkcji mogą być wartościami stałymi lub nazwami kolumn w tabeli.

**DATE\_SUB**(dt, INTERVAL liczba jednostki)

Odejmuje od daty dt określoną liczbę jednostek, wynikiem jest pomniejszona data.

Dostępne jednostki jak w funkcji DATE\_ADD

**Przykład:** `SELECT DATE_SUB('2020-04-20', INTERVAL 1 YEAR)` -> '2019-04-20'

*Uwaga:* parametry funkcji mogą być wartościami stałymi lub nazwami kolumn w tabeli.

**EXTRACT**(jednostka FROM dt)

Pobiera wskazaną przez jednostkę część daty dt, wynikiem jest pobrany składnik daty.

Dostępne jednostki jak w funkcji DATE\_ADD

**Przykład:** `SELECT EXTRACT(DAY FROM '2020-04-20')` -> 20

*Uwaga:* parametry funkcji mogą być wartościami stałymi lub nazwami kolumn w tabeli.

**TIMESTAMPDIFF**(jednostka, d1, d2)

Określa całkowitą liczbę jednostek zawartych pomiędzy datami dt1 i dt2, wynikiem jest obliczona liczba jednostek.

Dostępne jednostki jak w funkcji DATE\_ADD

**Przykład:** `SELECT TIMESTAMPDIFF(MONTH, '2020-04-20', '2020-05-30')` -> 1

*Uwaga:* parametry funkcji mogą być wartościami stałymi lub nazwami kolumn w tabeli.

**Funkcja warunkowa**

```
CASE WHEN warunek1 THEN wynik1
      WHEN warunek2 THEN wynik2 ...
      [ELSE wynikN] END
```

Zwraca wynik zależny od wartości warunków: jeżeli spełniony jest warunek1 zwraca wynik1, jeżeli spełniony jest warunek2 zwraca wynik2, itd. Jeżeli żaden z warunków nie jest spełniony zwraca wynikN umieszczony za ELSE (element opcjonalny).

**Przykład:** `SELECT CASE
 WHEN 1>0 THEN 'większe'
 ELSE 'mniejsze lub równe' END;` -> 'większe'

**Pełny wykaz funkcji MySQL jest dostępny na stronie:**

[https://www.w3schools.com/sql/sql\\_ref\\_mysql.asp](https://www.w3schools.com/sql/sql_ref_mysql.asp)

