

# Zapytania MySQL Workbench

**SHOW DATABASE;**

*-- sprawdzenie dostępnych baz danych*

**SHOW** tables;

*-- wywołanie listy tabel dostępnych w aktywnej bazie*

**DESC** tableName;

*-- wywoływanie opisu tabeli*

**SHOW COLUMNS FROM** tableName;

*-- wywoływanie opisu tabeli - drugi sposób*

**USE** tableName;

*-- ustawienie tabeli jako aktywnej*

**SELECT** database ();

*-- sprawdzenie aktywnej bazy danych*

**CREATE DATABASE** databaseName;

*-- tworzenie nowej bazy danych*

**DROP DATABASE** databaseName;

*-- usuwanie bazy danych i tabel w niej zawartych*

**CREATE TABLE** tableName;

*-- tworzenie nowej tabeli*

**DROP TABLE** tableName;

*-- usuwanie tabeli*

**DROP TABLE** databaseName .tableName;

*-- usuwanie tabeli z bazy danych innej niż aktywna*

**INSERT INTO** tableName (columnName) **VALUES** ("newValue");

*-- dodanie nowego rekordu*

**HELP;**

*-- uzyskanie pomocy*

**HELP** methodName;

*-- uzyskanie pomocy dla konkretnej metody*

**SELECT \* FROM** tableName;

*-- wywoływanie wszystkich kolumn z tabeli*

**SELECT** columnName 1, columnName 2 **FROM** tableName;

*-- wywoływanie określonych kolumn z tabeli*

-----  
-- SELECT DISTINCT --

SELECT DISTINCT columnName FROM tableName;

-- Zapytanie bez duplikatów

-----  
-- Zapytanie o liczby PARZYSTE i NIEPARZYSTE --

SELECT columnName FROM tableName WHERE id % 2 = 0;

-- parzyste

SELECT columnName FROM tableName WHERE id % 2 <> 0;

-- nieparzyste

-----  
-- WHERE --

SELECT \* FROM tableName WHERE columnName <= 30;

-- Zapytanie o liczbę mniejszą lub równą 30

SELECT \* FROM tableName WHERE columnName > 30 ORDER BY columnName;

-- Zapytanie o liczbę większą od 30 rosnąco

SELECT \* FROM tableName WHERE columnName = "recordValue";

-- Zapytanie o konkretny string

-----  
-- AND, OR, IN, NOT --

SELECT \* FROM tableName WHERE columnName = "recordValue";

SELECT \* FROM tableName WHERE NOT columnName = "recordValue";

-- zapytanie o wszystkie rekordy oprócz określonego rekordu

SELECT \* FROM tableName WHERE columnName1 = "recordValue" AND columnName2 = "recordValue";

-- zapytanie o dwa rekordy

SELECT \* FROM tableName WHERE columnName1 = "recordValue" OR columnName2 = "recordValue";

-- zapytanie o jeden z rekordów

SELECT \* FROM tableName WHERE columnName IN ("recordValue1", "recordValue2", "recordValue3");

-- zapytanie o konkretne rekordy

-----  
**-- IS NULL, != --**

**SELECT \* FROM** tableName **WHERE** columnName **IS NULL**;

*-- wyświetlenie kolumny gdzie istnieją puste rekordy*

**SELECT \* FROM** tableName **WHERE** columnName **IS NOT NULL**;

*-- wyświetlenie kolumny gdzie nie ma pustych rekordów*

**SELECT \* FROM** tableName **WHERE** columnName != value;

*-- wyświetlenie kolumny wykluczając dany rekord*

-----  
**-- LIMIT --**

**SELECT \* FROM** tableName **LIMIT 5**;

*-- zapytanie o konkretną ilość rekordów*

**SELECT** columnName **FROM** tableName **ORDER BY** columnName **LIMIT 1**;

*-- zapytanie od drugiego rekordu*

**SELECT** columnName **FROM** tableName **ORDER BY** columnName **LIMIT 0,4**;

*-- zapytanie od pierwszego rekordu + cztery rekordy*

-----  
**-- MIN, MAX --**

*-- obliczenie wartości minimalnej lub maksymalnej z kolumny*

**SELECT MAX**(columnName) **FROM** tableName ;

**SELECT MIN**(columnName) **FROM** tableName;

-----  
**-- COUNT, AVG, SUM --**

**SELECT COUNT**(columnName) **FROM** tableName **WHERE** columnName = 1;

*-- COUNT - oblicza ile wierszy zostało zwróconych z danego zapytania*

**SELECT AVG**(columnName) **FROM** tableName **WHERE** columnName = 1;

*-- AVG - oblicza średnią z danej kolumny*

**SELECT SUM**(columnName) **FROM** tableName **WHERE** columnName = 1;

*-- SUM - oblicza sumę wszystkich wartości w danej kolumnie*

-----  
-- LIKE, BETWEEN --

SELECT \* FROM tableName WHERE columnName LIKE "%n";

-- rekord kończy się literką 'n'

SELECT \* FROM tableName WHERE columnName LIKE "%on";

-- rekord kończy się litery 'on'

SELECT \* FROM tableName WHERE columnName LIKE "P%";

-- rekord zaczyna się literką 'P'

SELECT \* FROM tableName WHERE columnName LIKE "Pa%";

-- rekord zaczyna się litery 'Pa'

SELECT \* FROM tableName WHERE columnName LIKE "p%n";

-- rekord zaczyna się na 'p' i kończy na 'n'

SELECT \* FROM tableName WHERE columnName LIKE "%y%";

-- rekord ma w dowolnym miejscu literę 'y'

SELECT \* FROM tableName WHERE columnName LIKE "%co%";

-- rekord ma w dowolnym miejscu ciąg litery 'co'

SELECT \* FROM tableName WHERE columnName LIKE "\_a%";

-- rekord których druga litera jest 'a';

SELECT \* FROM tableName WHERE columnName LIKE "\_\_r%";

-- rekord których trzecia litera jest 'r';

SELECT \* FROM tableName WHERE columnName LIKE "\_\_ri%";

-- rekord których trzecia i czwarta litery to 'ri';

SELECT \* FROM tableName WHERE columnName LIKE "\_\_ri%s";

-- rekord których trzecia i czwarta litery to 'ri' a ostatnia 's';

SELECT \* FROM tableName WHERE columnName LIKE "\_\_\_\_\_";

-- rekord ma w sobie 7 literek

SELECT \* FROM tableName WHERE columnName LIKE "\_\_\_\_\_";

-- rekord ma w sobie minimum 10 literek i więcej

SELECT \* FROM tableName WHERE columnName >= "2022-08-01" AND columnName <= "2022-09-30";

SELECT \* FROM tableName WHERE columnName BETWEEN "2020-01-01" AND "2021-12-31";

-- krótsze zapytanie używając BETWEEN

SELECT \* FROM tableName WHERE columnName BETWEEN "3" AND "8";

-----  
-- AS - Alias --

-- zmiana nazwy kolumny

**SELECT** columnName1, columnName2 **FROM** tableName;

**SELECT** columnName1 **AS** "name Change1", columnName2 **AS** nameChange2 **FROM** tableName;

**SELECT SUM**(columnName) **AS** nameChange **FROM** tableName;

-----  
-- ORDER BY --

-- sortowanie rosnąco i malejąco

**SELECT** columnName, columnName1 **FROM** tableName **ORDER BY** columnName1;

-- domyślnie rosnąco

**SELECT** columnName, columnName1 **FROM** tableName **ORDER BY** columnName1 **ASC**;

-- rosnąco

**SELECT** columnName, columnName1 **FROM** tableName **ORDER BY** columnName1 **DESC**;

-- malejąco

-- sortowanie w wielu kolumnach

**SELECT** columnName1, columnName2 **FROM** tableName **ORDER BY** columnName2 **ASC**, columnName1 **DESC**;

**SELECT** columnName2, columnName1 **FROM** tableName **ORDER BY** columnName2 **ASC**, columnName1 **ASC**;

-- zmiana kolejności kolumn

-----  
-- GROUP BY, HAVING --

-- grupowanie wierszy, które mają te same wartości - podsumowanie

**SELECT** columnName **SUM**(columnName1) **FROM** tableName **GROUP BY** columnName;

**SELECT** columnName **SUM**(columnName1) **FROM** tableName **GROUP BY** columnName **ORDER BY SUM**(columnName1) ;

**SELECT** columnName, columnName1 **SUM**(columnName2) **FROM** tableName **GROUP BY** columnName, columnName1;

**SELECT** columnName, columnName1 **SUM**(columnName2) **FROM** tableName **GROUP BY** columnName, columnName1 **ORDER BY SUM**(columnName2) ;

-- grupowanie i dodanie warunku przedziału wskazanej kolumny od 2 do 6

```
SELECT columnName, columnName1 SUM(columnName2) FROM tableName WHERE columnName BETWEEN 2 AND 6 GROUP BY columnName, columnName1;
```

-- obliczenie różnicy między całkowitą liczbą wpisów a liczbą duplikatów

```
SELECT columnName COUNT(columnName) – COUNT(DISTINCT (columnName)) FROM tableName;
```

```
SELECT columnName1 COUNT(columnName2) AS nameChange FROM tableName GROUP BY columnName1 ORDER BY COUNT(columnName2) DESC;
```

```
SELECT columnName1, columnName3 COUNT(columnName2) AS nameChange FROM tableName GROUP BY columnName1, columnName3;
```

-- klauzula HAVING jest dodana do SQL, ponieważ WHERE nie może być używane z funkcjami

```
SELECT columnName1 COUNT(columnName2) AS nameChange FROM tableName GROUP BY columnName1 HAVING COUNT(columnName2) > 5;
```

-----

-- INSERT INTO --

-- dodanie nowego rekordu

```
INSERT INTO tableName (columnName1, columnName2, columnName3) VALUES ("newValue1", "newValue2", "newValue3");
```

```
INSERT INTO tableName (columnName1, columnName2, columnName3) VALUES
```

-- dodanie kilku nowych rekordów

```
("newValue1a", "newValue2a", "newValue3a"),
```

```
("newValue1b", "newValue2b", "newValue3b"),
```

```
("newValue1c", "newValue2c", "newValue3c");
```

```
INSERT INTO tableName VALUES ("newValue1", "newValue2", "newValue3");
```

-- szybsze dodanie nowego rekordu

```
INSERT INTO tableName VALUES (NULL, "newValue2", "newValue3");
```

-- automatyczna inkrementacja ID

-----

-- UPDATE --

-- aktualizacja rekordu

UPDATE tableName SET columnName = "changeValue" WHERE columnNameID = 4;

-- aktualizacja wszystkich rekordów, gdzie występują te same nazwy

UPDATE tableName SET columnName1 = "changeValue1", columnName2 = "changeValue2" WHERE columnName2 = Value2;

-----

-- DELETE --

-- usuwanie rekordu

DELETE FROM tableName WHERE columnName LIKE "recordValue";

DELETE FROM tableName WHERE columnName LIKE "%5";

DELETE FROM tableName WHERE columnNameID = 4";

-----

-- INNER JOIN --

-- łączenie danych z co najmniej dwóch tabel

SELECT tableName2.columnNameID, tableName1.columnName1, tableName2.columnName2

FROM tableName2

INNER JOIN tableName1

ON tableName2.columnNameID = tableName1.columnNameID

ORDER BY tableName2.columnNameID;

```
SELECT tableName2.columnNameID, tableName1.columnName, tableName2.columnName, tableName3.columnName  
  
FROM tableName2  
  
INNER JOIN tableName1 ON tableName2.columnNameID = tableName1.columnNameID  
  
INNER JOIN tableName3 ON tableName2.columnNameID2 = tableName3.columnNameID  
  
ORDER BY tableName2.columnNameID;
```

-----

-- LEFT JOIN --

-- zwraca wszystkie rekordy z lewej tabeli oraz pasujące z prawej tabeli

```
SELECT tableName2.columnNameID, tableName1.columnName  
  
FROM tableName2  
  
LEFT JOIN tableName1  
  
ON tableName2.columnNameID = tableName1.columnNameID;
```

-----

-- RIGHT JOIN --

-- zwraca wszystkie rekordy z prawej tabeli oraz pasujące z lewej tabeli

```
SELECT tableName2.columnNameID, tableName1.columnName1, tableName1.columnName2  
  
FROM tableName2  
  
RIGHT JOIN tableName1  
  
ON tableName2.columnNameID = tableName1.columnNameID;
```



-----

-- **UNION** --

-- *łączenie wszystkich zestawień bez duplikatów z obu kolumn*

**SELECT** columnName **FROM** tableName1 **UNION** **SELECT** columnName **FROM** tableName2;

-- **UNION ALL** --

-- *łączenie wszystkich zestawień z duplikatami z obu kolumn*

**SELECT** columnName **FROM** tableName1 **UNION ALL** **SELECT** columnName **FROM** tableName2;

-----