

# Ćwiczenia z MySQL w CLI

## Laboratoria 1

### 1 Tworzenie bazy danych

```
MariaDB [(none)]> CREATE DATABASE testowa_db;  
Query OK, 1 row affected (0.001 sec)  
  
MariaDB [(none)]> USE testowa_db;  
Database changed  
MariaDB [testowa_db]> _
```

### 2 Tworzenie dwóch tabel Kliencie i Zamownienia

```
MariaDB [testowa_db]> CREATE TABLE Klienci (  
-> ID INT AUTO_INCREMENT PRIMARY KEY,  
-> Imie VARCHAR(50),  
-> Nazwisko VARCHAR(50),  
-> Email VARCHAR(100)  
-> );  
Query OK, 0 rows affected (0.273 sec)  
  
MariaDB [testowa_db]> CREATE TABLE Zamowienia (  
-> ID INT AUTO_INCREMENT PRIMARY KEY,  
-> Produkt VARCHAR(100),  
-> Cena DECIMAL(10,2),  
-> Data_zamowienia DATE,  
-> Klient_ID INT NOT NULL,  
-> FOREIGN KEY (Klient_ID) REFERENCES Klienci(ID)  
-> );  
Query OK, 0 rows affected (0.210 sec)
```

### 3 Dodawanie danych do tabel

```
MariaDB [testowa_db]> INSERT INTO Klienci (Imie, Nazwisko, Email) VALUES  
-> ('Jan', 'Kowalski', 'jan.kowalski@example.com'),  
-> ('Anna', 'Nowak', 'anna.nowak@example.com');  
Query OK, 2 rows affected (0.116 sec)  
Records: 2 Duplicates: 0 Warnings: 0  
  
MariaDB [testowa_db]> INSERT INTO Zamowienia (Produkt, Cena, Data_zamowienia, Klient_ID) VALUES  
-> ('Laptop', 2500.00, '2024-03-25', 1),  
-> ('Smartphone', 1200.00, '2024-03-26', 2),  
-> ('Monitor', 500.00, '2024-03-27', 1);  
Query OK, 3 rows affected (0.046 sec)  
Records: 3 Duplicates: 0 Warnings: 0
```

## 4 Zmiana klucza obcego

```
MariaDB [testowa_db]> UPDATE Zamowienia SET Klient_ID = 2 WHERE ID = 1;
Query OK, 1 row affected (0.055 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Udało się zaktualizować klucz obcy na nowy, lecz taka operacja jest mało bezpieczna ze względu na możliwość utraty spójności danych oraz naruszenia integralności referencyjnej w bazie danych.

## 5 Usunięcie bazy danych

```
MariaDB [testowa_db]> DROP TABLE Zamowienia;
Query OK, 0 rows affected (0.216 sec)

MariaDB [testowa_db]> DROP TABLE Klienci;
Query OK, 0 rows affected (0.197 sec)

MariaDB [testowa_db]> DROP DATABASE testowa_db;
Query OK, 1 row affected (0.466 sec)
```

# Laboratoria 2

## 1 Wyświetlenie tabeli z użyciem JOIN

```
MariaDB [filmowa_26_03_2024]> SELECT aktorzy.aktor_imie, aktorzy.aktor_nazwisko, rezyserzy.rezyser_imie, rezyser_nazwisko, filmy.film_tytul FROM aktorzy
JOIN filmy ON aktorzy.aktor_id = filmy.film_aktor_id JOIN rezyserzy ON filmy.film_rezyser_id = rezyserzy.rezyser_id;
```

| aktor_imie | aktor_nazwisko | rezyser_imie | rezyser_nazwisko | film_tytul          |
|------------|----------------|--------------|------------------|---------------------|
| Cillian    | Murphy         | Christopher  | Nolan            | Oppenheimer         |
| Bogusław   | Linda          | Władysław    | Pasikowski       | Psy                 |
| Brad       | Pitt           | Quentin      | Tarantino        | Bekarty wojny       |
| Jamie      | Foxx           | Quentin      | Tarantino        | Django              |
| Anthony    | Hopkins        | Jonathan     | Demme            | Milczenie owiec     |
| Al         | Pacino         | Francis Ford | Coppola          | Ojciec Chrzestny II |
| Al         | Pacino         | Martin       | Brest            | Zapach kobiety      |
| Christian  | Bale           | Christopher  | Nolan            | Mroczny rycerz      |
| Brad       | Pitt           | David        | Fincher          | Podziemny krąg      |

9 rows in set (0.000 sec)

## 2 Posortowanie malejąco i wyświetlenie ile razy dany aktor zagrał w filmie

```
MariaDB [filmowa_26_03_2024]> SELECT aktorzy.aktor_imie, aktorzy.aktor_nazwisko, COUNT(*) AS liczba_filmow
FROM Aktorzy JOIN Filmy ON aktorzy.aktor_id = filmy.film_aktor_id GROUP BY aktorzy.aktor_id ORDER BY liczba_filmow DESC;
```

| aktor_imie | aktor_nazwisko | liczba_filmow |
|------------|----------------|---------------|
| Al         | Pacino         | 2             |
| Brad       | Pitt           | 2             |
| Bogusław   | Linda          | 1             |
| Jamie      | Foxx           | 1             |
| Cillian    | Murphy         | 1             |
| Anthony    | Hopkins        | 1             |
| Christian  | Bale           | 1             |

7 rows in set (0.001 sec)

### 3 Wyświetlenie imiona i nazwiska reżyserów filmów nakręconych w XXI wieku, bez duplikatów

```
MariaDB [filmowa_26_03_2024]> SELECT DISTINCT rezyserzy.rezyser_imie, rezyserzy.rezyser_nazwisko FROM rezyserzy
JOIN filmy ON rezyserzy.rezyser_id = filmy.film_rezyser_id WHERE filmy.film_data_produkcji >= 2000;
```

| rezyser_imie | rezyser_nazwisko |
|--------------|------------------|
| Christopher  | Nolan            |
| Wladyslaw    | Pasikowski       |
| Quentin      | Tarantino        |
| Jonathan     | Demme            |
| Francis Ford | Coppola          |
| Martin       | Brest            |
| David        | Fincher          |

7 rows in set, 1 warning (0.001 sec)

### 4 Wykorzystanie klauzuli PARTITION BY

```
MariaDB [filmowa_26_03_2024]> SELECT film_id, film_tytul, YEAR(film_data_produkcji) AS rok, film_box_office_mln,
SUM(film_box_office_mln) OVER (PARTITION BY YEAR(film_data_produkcji)) AS film_box_P_Y FROM filmy;
```

| film_id | film_tytul          | rok  | film_box_office_mln | film_box_P_Y |
|---------|---------------------|------|---------------------|--------------|
| 1500    | Ojciec Chrzestny II | 1974 | 193                 | 193          |
| 1400    | Milczenie owiec     | 1991 | 272                 | 272          |
| 1100    | Psy                 | 1992 | 5                   | 139          |
| 1600    | Zapach kobiety      | 1992 | 134                 | 139          |
| 1800    | Podziemny krag      | 1999 | 101                 | 101          |
| 1700    | Mroczny rycerz      | 2008 | 1000                | 1000         |
| 1200    | Bekarty wojny       | 2009 | 321                 | 321          |
| 1300    | Django              | 2012 | 426                 | 426          |
| 1000    | Oppenheimer         | 2023 | 954                 | 954          |

9 rows in set (0.001 sec)

### 5 Wykorzystanie klauzuli GROUP BY

```
MariaDB [filmowa_26_03_2024]> SELECT aktorzy.aktor_imie, aktorzy.aktor_nazwisko, COUNT(*) as LiczbaFilmow
-> FROM aktorzy
-> JOIN filmy ON aktorzy.aktor_id = filmy.film_aktor_id
-> GROUP BY aktorzy.aktor_id
-> ORDER BY LiczbaFilmow DESC;
```

| aktor_imie | aktor_nazwisko | LiczbaFilmow |
|------------|----------------|--------------|
| Brad       | Pitt           | 2            |
| Al         | Pacino         | 2            |
| Cillian    | Murphy         | 1            |
| Anthony    | Hopkins        | 1            |
| Christian  | Bale           | 1            |
| Boguslaw   | Linda          | 1            |
| Jamie      | Foxx           | 1            |

7 rows in set (0.004 sec)

**GROUP BY** agreguje dane w grupy, ograniczając wynik do jednego wiersza dla każdej grupy, podczas gdy **PARTITION BY** nie ogranicza liczby wierszy w wyniku, ale pozwala na obliczanie agregatów na podstawie każdej partycji.

### 6 Wnioski

Podczas pierwszych laboratoriów poznaliśmy podstawowe komendy MySQL'a. Zmiana klucza obcego nauczyła nas by być przy tym ostrożnym. Na kolejnych laboratoriach poznaliśmy bardziej zaawansowane komendy, takie jak JOIN pozwalająca łączyć dwie tabele czy PARTITION BY i GROUP BY pozwalające na podział danych i ich agregację.