



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA



**Katedra
Informatyki i Automatyki**
Politechnika Rzeszowska

Bazy Danych - Projekt

Sprawozdanie

pt.: „Rozwój systemu zarządzania danymi dla
polskich festiwali muzycznych”

Data wykonania: 11.06.2025

Grupa: L2
Dawid Lech

Spis treści

Spis treści	2
1. Cel pracy	3
2. Przebieg Pracy	4
2.1. Etap 1	4
Określenie funkcji bazy danych i ich priorytety:	4
Technologie i wykorzystywane narzędzia	5
2.2. Etap 2	6
Encje bazy danych	6
Dokładny opis tabel wraz z typami	8
Skrypt do stworzenia bazy danych	12
3. Etap 3	15
Proste zapytania SQL (select, insert into)	15
Zaawansowane Zapytania typu SELECT (wykorzystujące JOIN, GROUP BY)	24
Zapytania UPDATE	28
4. Etap 4	30
Prezentacja użytkowników BD i ich ról	30
Procedury obsługujące bazę danych	31
Zarządzanie bazą danych (backup, restore),	34
Warianty testowe i produkcyjne	35
5. Wnioski	36
6. Spis rysunków i tabel	36

1. Cel pracy

Festiwale muzyczne w Polsce to jedno z największych wydarzeń kulturalnych. W tego typu imprezach masowych w Polsce udział biorą setki tysięcy osób. Wydarzenia tego typu mają integrować ludzi dawać im rozrywkę a także promować miasta czy artystów. I nie można zapomnieć o istotnym elemencie finansowym

Celem tej pracy jest stworzenie bazy danych która umożliwi usprawnienie organizacji i analizy danych wydarzeń. Pomoże ona także w podejmowaniu decyzji finansowych poprzez licznie skumulowane dane o wydarzeniach. Dla samych uczestników festiwali System będzie idealnym narzędzie w którym będzie można kupić bilet i dostęp do skumulowanych danych o harmonogramach i cenach biletów na poszczególny festiwal oraz na bieżąco aktualizowane dane. Baza danych umożliwi obsługę różnych festiwali w ramach jednej platformy oraz integrację z aplikacją mobilną czy systemami płatności.

Od strony organizatora da mu możliwość widoczności dostępu artystów, raportów sprzedaży biletów aktualnych jaki i tych w poprzednich latach. Pomoże to także określić np. średni wiek uczestników, płeć czy też historie uczestnictwa. Określanie najpopularniejszych zespołów, przepływu ludzi oraz znajdowanie trendów.

2. Przebieg Pracy

2.1. Etap 1

Określenie funkcji bazy danych i ich priorytety:

Główną funkcjonalnością bazy danych powinny być Rejestracja wydarzeń i wykonawców, Zakupy i kontrola biletów, analiza liczby uczestników oraz preferencji. Poniżej przedstawiona tabelę określającą szczegóły dotyczące wyżej wymienionych funkcji. Priorytety poszczególnych funkcji określone są w trzy stopniowej skali: Wysoki, Średni, Niski.

Tabela 1 Funkcje bazy danych

Funkcja	Opis	Priorytet
Rejestracja i logowanie użytkowników	Przechowywanie danych użytkowników, zwracając uwagę na bezpieczeństwo	Wysoki
Zarządzanie festiwalami	Informacje o nazwie, dacie trwanie, typu festiwalu i szczegółowych informacji	Wysoki
Harmonogram koncertów	Tworzenie harmonogramów koncertów na poszczególnych scenach, przypisywanie artystów	Wysoki
Zarządzanie artystami i zespołami	Informacje biograficzne, rodzaj muzyki, powiązania konkretnymi wydarzeniami	Średni
Zarządzanie scenami	Dane o scenach, lokalizacjach, przypisywanie wydarzeń	Średni
Sprzedaż biletów	Tworzenie kategorii biletów, obsługa zakupów	Średni
Analiza danych uczestników	Statystyki sprzedaży, demografia uczestników, raporty finansowe	Średni
Zarządzanie płatnościami	Rejestrowanie płatności, powiązanie z biletami, status zakupów	Wysoki
Powiadomienia	Notyfikacje wysyłane do zainteresowanych o dostępności czy zmianach w wydarzeniu	Niski
Raporty dla organizatorów	Analiza sprzedaży, popularność wydarzenia	Średni

Technologie i wykorzystywane narzędzia

Projekt będzie opierał się na relacyjnej bazie danych z wykorzystaniem MySQL. Tworzenie diagramów oraz w graficznym przedstawieniu zostanie wykorzystane narzędzie MySQL Workbench 8.0. XAMPP zostanie wykorzystany do postawienia serwera na którym będzie znajdować się baza danych. Sporadycznie do zarządzania również zostanie wykorzystane phpMyadmin.

Link do repozytorium na GITHUB

<https://github.com/sanchez-189/Festiwale-muzyczne.git>

2.2. Etap 2

Encje bazy danych

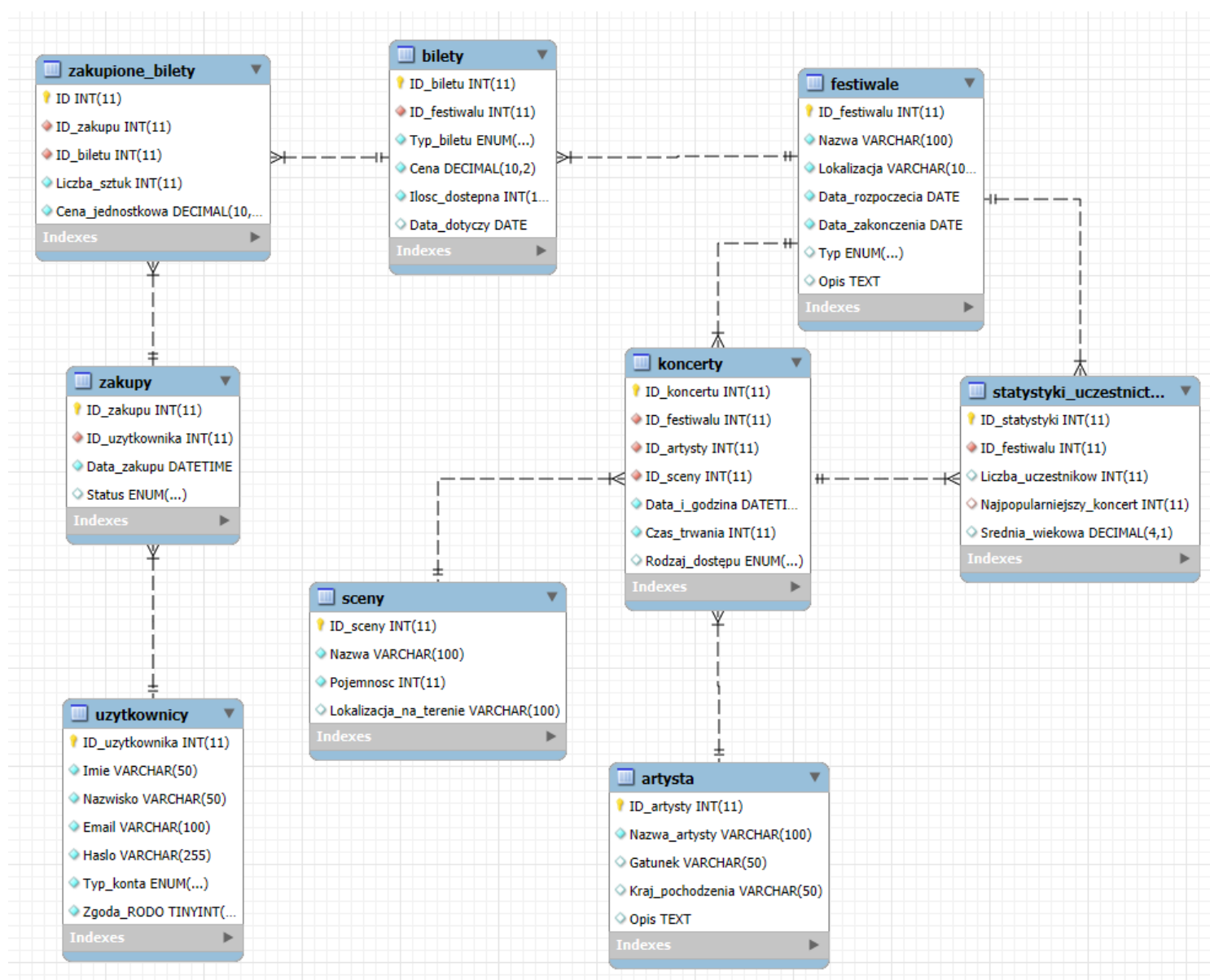
Poniżej znajduje się tabela przedstawiająca wszystkie encje bazy oraz ich cel w jakim zostały stworzone wraz z krótkim opisem.

Tabela 2 Encje Bazy Danych

Nazwa	Opis/Cel
Użytkownicy	Encja zawierająca wszystkie osoby korzystające z systemu – uczestnicy, organizatorzy, artyści
Festiwale	Encja obejmująca wszystkie dane o festiwalu muzycznym
Artysta	Encja zawierająca informacje o artystach lub zespołach muzycznych
Koncerty	Encja określająca pojedynczy występ artysty w ramach festiwalu
Sceny	Miejsce w którym odbywają się koncerty w ramach festiwalu
Bilety	Encja definiuje typy i ceny biletów dostępnych na dany festiwal
Zakupy	Rejestr transakcji zakupu biletów
Statystyki	Encja zawierająca dane analityczne dotyczące festiwalu
Zakupione bilety	Tabele pomocnicza do zakupy są to poszczególne pozycje na rachunki. Umożliwia kupowanie wielu biletów na jednym rachunku

Projekt oparty jest na relacyjnej bazie danych dlatego powyższe encje zostały ze sobą relacjami co przedstawia diagram. Plik z diagramem możliwym do odtworzenia przy pomocy MySQL Workbench również został dołączony do dokumentacji

Rysunek 1 Diagram Bazy Danych



Dokładny opis tabel wraz z typami

Tabela użytkowników zawiera dane o wszystkich użytkownikach systemu z których można wyróżnić: Gościa, Organizatora festiwalu, Administratora

Tabela 3 Opis Encji Użytkownicy

Użytkownicy		
ID_użytkownika	INT(11)	Numer identyfikacyjny użytkownika, automatycznie nadawany przy rejestracji
Imię	VARCHAR(50)	Imię użytkownika
Nazwisko	VARCHAR(50)	Nazwisko użytkownika
Email	VARCHAR(100)	Email użytkownika potrzeby do powiadomień oraz dostarczania biletów
Hasło	VARCHAR(255)	Hasło do konta użytkownika
Typ_konta	ENUM('uczestnik', 'organizator', 'admin')	Określenie rodzaju konta z trzech możliwych: uczestnik, organizator, admin
Zgoda_RODO	BOOLEAN	Udzielenie zgody na przetwarzanie danych osobowych

Tabela Zakupy łączy się ona relacją N:1 z tabelą użytkowników zawiera informacje o wykonanych przez użytkowników zakupach w systemie jest to rodzaj tabeli nagłówkowej zawierającej tylko najważniejsze dane o konkretnym zakupie

Tabela 4 Opis Encji Zakupy

Zakupy		
ID_zakupu	INT(11)	Unikalny identyfikator zakupu (numer rachunku)
ID_użytkownika	INT(11)	Numer identyfikacyjny użytkownika który dokonał zakupu
Data_zakupu	DATETIME	Data oraz godzina wykonania zakupu. Ustawia się automatycznie po zatwierdzeniu zakupu
Status	ENUM('oczekujący', 'opłacony', 'anulowany'),	Status zakupu który może być oczekujący na płatność opłacony lub anulowany

Tabela zakupione bilety jest połączona relacją N:1 z tabelą zakupy zawiera ona szczegółowe informacje o zakupionych biletach na jednym „zamówieniu”

Tabela 5 Opis Encji Zakupione bilety

Zakupione bilety		
ID	INT(11)	Pozycja konkretnego wykupionego bilety na rachunku
ID_zakupu	INT(11)	Numer rachunku którego dotyczy zakupiony bilet
ID_biletu	INT(11)	Numer bilety który został zakupiony przez użytkownika
Liczba_sztuk	INT(11)	Ilość zakupionych konkretnych biletów
Cena_jednostkowa	DECIMAL(10,2)	Cena z jedną sztukę biletu

Tabela Zakupione bilety jest połączona relacją N:1 z tabelą bilety która zawiera szczegółowe informacje na temat zakupionego biletu i do czego on upoważnia.

Tabela 6 Opis Encji Bilety

Bilety		
ID_biletu	INT(11)	Numer identyfikacyjny biletu
ID_festiwalu	INT(11)	Numer identyfikacyjny festiwalu którego bilet dotyczy
Typ_biletu	ENUM('jednodniowy', 'karnet', 'vip')	Rodzaj bilety który może być: jednodniowy, karnet na cały festiwal lub dla VIP-ów
Cena	DECIMAL(10,2)	Cena biletu
Ilość_dostępna	INT(11)	Dostępna ilość biletów do zakupu
Data_dotyczy	DATE	Data na którą został kupiony bilet, Jeżeli wartość NULL oznacza to dostęp do całego festiwalu

Tabela Festiwale jest powiązana z tabelą bilety relacją 1:N zawiera informacje o odbywających się lub zaplanowanych festiwalach

Tabela 7 Opis Encji Festiwale

Festiwale		
ID_festiwalu	INT(11)	Numer identyfikacyjny festiwalu
Nazwa	VARCHAR(100)	Nazwa festiwalu
Lokalizacja	VARCHAR(100)	Miejsce odbywania się festiwalu
Data_rozpoczęcia	DATE	Data rozpoczęcia się festiwalu
Data_zakończenia	DATE	Data zakończenia się festiwalu
Typ	ENUM('rock', 'pop', 'jazz', 'electronic', 'inne')	Przeważający typ muzyki która będzie wykonywana na festiwalu

Tabela Koncerty łączy się relacją N: 1 z tabelą festiwale. Treści w niej zawarte odnoszą się do konkretnych koncertów odbywających się na danym festiwalu

Tabela 8 Opis Encji Koncerty

Koncerty		
ID_koncertu	INT(11)	Numer identyfikacyjny koncertu
ID_Festiwalu	INT(11)	Numer identyfikacyjny Festiwalu na którym odbywa się koncert
ID_artysty	INT(11)	Numer identyfikacyjny artysty grającego na tym koncercie
ID_sceny	INT(11)	Numer identyfikacyjny sceny gdzie będzie odbywał się koncert
Data_i_godzina	DATETIME	Data i czas rozpoczęcia koncertu
Czas_trwania	INT(11)	Liczb godzin ile będzie trwał koncert

Tabela Sceny jest powiązania relacją 1:N z tabelą Koncerty i zawiera ona szczegółowe informacje o danej scenie

Tabela 9 Opis Encji Sceny

Sceny		
ID_sceny	INT(11)	Numer identyfikacyjny sceny
Nazwa	VARCHAR(100)	Potoczna nazwa sceny
Pojemnosc	INT(11)	Liczba osób których pomieści scena
Lokalizacja_na_terenie	VARCHAR(100)	Położenie sceny odnoszące się do charakterystycznych obiektów

Tabela Artysta również połączona jest relacją 1:N z tabelą koncerty i zawiera ona informacje o konkretnych artystach, wykonawcach, zespołach grających na festiwalach

Tabela 10 Opis Encji Artysta

Artysta		
ID_artysty	INT(11)	Numer identyfikacyjny artysty
Nazwa_artysty	VARCHAR(100)	Nazwa artysty lub zespołu
Gatunek	VARCHAR(100)	Gatunek wykonywany przez danego artystę
Kraj_pochodzenia	VARCHR(100)	Kraj pochodzenia artysty
Opis	TEXT	Krótki opis, znaki szczególne, charakterystyczne informacje dla organizatorów

Tabela Statystyki uczestnictwa jest powiązana relacją N:1 z tabelami festiwale, koncerty zawiera ona informacje głównie przydatne dla organizatorów oraz analityków

Tabela 11 Opis Encji Statystyki uczestnictwa

Statystyki uczestnictwa		
ID_Statystyki	INT(11)	Numer identyfikacyjny danej statystyki odnoszącej się do konkretnych festiwali i koncertów
ID_festiwalu	INT(11)	Numer festiwalu którego dotyczy statystyka
Liczba_uczestników	INT(11)	Liczba uczestników festiwalu lub koncertu
Najpopularniejszy_koncert	INT(11)	Koncert na którym było najwięcej widzów
Srednia_wiekowa	DECIMAL(4,1)	Średnia wiekowa uczestników

Skrypt do stworzenia bazy danych

```
CREATE TABLE Uzytkownicy (
    ID_uzytkownika INT AUTO_INCREMENT PRIMARY KEY,
    Imie VARCHAR(50) NOT NULL,
    Nazwisko VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    Haslo VARCHAR(255) NOT NULL,
    Typ_konta ENUM('uczestnik', 'organizator', 'admin') NOT NULL,
    Zgoda_RODO BOOLEAN NOT NULL
);

CREATE TABLE Festiwale (
    ID_festiwalu INT AUTO_INCREMENT PRIMARY KEY,
    Nazwa VARCHAR(100) NOT NULL,
    Lokalizacja VARCHAR(100) NOT NULL,
    Data_rozpoczecia DATE NOT NULL,
    Data_zakonczenia DATE NOT NULL,
```

```
Typ ENUM('rock', 'pop', 'jazz', 'electronic', 'inne'),
Opis TEXT
);
CREATE TABLE Artysta (
    ID_artysty INT AUTO_INCREMENT PRIMARY KEY,
    Nazwa_artysty VARCHAR(100) NOT NULL,
    Gatunek VARCHAR(50),
    Kraj_pochodzenia VARCHAR(50),
    Opis TEXT
);
CREATE TABLE Sceny (
    ID_sceny INT AUTO_INCREMENT PRIMARY KEY,
    Nazwa VARCHAR(100) NOT NULL,
    Pojemnosc INT NOT NULL,
    Lokalizacja_na_terenie VARCHAR(100)
);
CREATE TABLE Koncerty (
    ID_koncertu INT AUTO_INCREMENT PRIMARY KEY,
    ID_festiwalu INT NOT NULL,
    ID_artysty INT NOT NULL,
    ID_sceny INT NOT NULL,
    Data_i_godzina DATETIME NOT NULL,
    Czas_trwania INT NOT NULL,
    Rodzaj_dostepu ENUM('ogólny', 'VIP'),
    FOREIGN KEY (ID_festiwalu) REFERENCES Festiwale(ID_festiwalu),
    FOREIGN KEY (ID_artysty) REFERENCES Artysta(ID_artysty),
    FOREIGN KEY (ID_sceny) REFERENCES Sceny(ID_sceny)
);
CREATE TABLE Bilety (
    ID_biletu INT AUTO_INCREMENT PRIMARY KEY,
    ID_festiwalu INT NOT NULL,
    Typ_biletu ENUM('jednodniowy', 'karnet', 'vip') NOT NULL,
    Cena DECIMAL(10,2) NOT NULL,
    Ilosc_dostepna INT NOT NULL,
```

```
Data_dotyczy DATE,  
FOREIGN KEY (ID_festiwalu) REFERENCES Festiwale(ID_festiwalu)  
);  
CREATE TABLE Zakupy (  
    ID_zakupu INT AUTO_INCREMENT PRIMARY KEY,  
    ID_uzytkownika INT NOT NULL,  
    Data_zakupu DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    Status ENUM('oczekujący', 'opłacony', 'anulowany'),  
    FOREIGN KEY (ID_uzytkownika) REFERENCES Uzytkownicy(ID_uzytkownika)  
);  
CREATE TABLE Zakupione_Bilety (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    ID_zakupu INT NOT NULL,  
    ID_biletu INT NOT NULL,  
    Liczba_sztuk INT NOT NULL,  
    Cena_jednostkowa DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (ID_zakupu) REFERENCES Zakupy(ID_zakupu),  
    FOREIGN KEY (ID_biletu) REFERENCES Bilety(ID_biletu)  
);  
CREATE TABLE Statystyki_uczestnictwa (  
    ID_statystyki INT AUTO_INCREMENT PRIMARY KEY,  
    ID_festiwalu INT NOT NULL UNIQUE,  
    Liczba_uczestnikow INT DEFAULT 0,  
    Najpopularniejszy_koncert INT,  
    Srednia_wiekowa DECIMAL(4,1),  
    FOREIGN KEY (ID_festiwalu) REFERENCES Festiwale(ID_festiwalu),  
    FOREIGN KEY (Najpopularniejszy_koncert) REFERENCES Koncerty(ID_koncertu)  
);
```

3. Etap 3

Proste zapytania SQL (select, insert into)

Rozpoczęcie pracy z bazą zacznę od umieszczenia w niej danych potrzebny do późniejszych zapytań. Plik SQL z danymi również znajduje się w dołączonych do projektu plików.

```
INSERT INTO Uzytkownicy (Imie, Nazwisko, Email, Haslo, Typ_konta, Zgoda_RODO) VALUES
('Anna', 'Krawczyk', 'anna.krawczyk@example.com', 'hashedpassword', 'uczestnik', True),
('Paweł', 'Krawczyk', 'pawel.krawczyk@example.com', 'hashedpassword', 'uczestnik', True),
('Krzysztof', 'Wójcik', 'krzysztof.wojcik@test.org', 'hashedpassword', 'uczestnik', True),
('Maria', 'Nowak', 'maria.nowak@mail.com', 'hashedpassword', 'organizator', True),
('Jan', 'Lewandowski', 'jan.lewandowski@mail.com', 'hashedpassword', 'uczestnik', True),
('Paweł', 'Dąbrowski', 'pawel.dabrowski@test.org', 'hashedpassword', 'admin', True),
('Anna', 'Wójcik', 'anna.wojcik@test.org', 'hashedpassword', 'uczestnik', True),
('Maria', 'Wiśniewski', 'maria.wisniewski@example.com', 'hashedpassword', 'organizator', True),
('Tomasz', 'Krawczyk', 'tomasz.krawczyk@mail.com', 'hashedpassword', 'uczestnik', True),
('Agnieszka', 'Wójcik', 'agnieszka.wojcik@example.com', 'hashedpassword', 'organizator', True),
('Ewa', 'Kowalska', 'ewa.kowalska@test.org', 'hashedpassword', 'organizator', True),
('Paweł', 'Kowalska', 'pawel.kowalska@example.com', 'hashedpassword', 'uczestnik', True),
('Agnieszka', 'Nowak', 'agnieszka.nowak@mail.com', 'hashedpassword', 'admin', True),
('Piotr', 'Wójcik', 'piotr.wojcik@example.com', 'hashedpassword', 'uczestnik', True),
('Piotr', 'Nowak', 'piotr.nowak@mail.com', 'hashedpassword', 'organizator', True),
('Tomasz', 'Nowak', 'tomasz.nowak@example.com', 'hashedpassword', 'admin', True),
('Ewa', 'Nowak', 'ewa.nowak@example.com', 'hashedpassword', 'admin', True),
('Magda', 'Nowak', 'magda.nowak@mail.com', 'hashedpassword', 'admin', True),
('Ewa', 'Mazur', 'ewa.mazur@mail.com', 'hashedpassword', 'uczestnik', True),
('Piotr', 'Dąbrowski', 'piotr.dabrowski@mail.com', 'hashedpassword', 'organizator', True),
('Piotr', 'Lewandowski', 'piotr.lewandowski@example.com', 'hashedpassword', 'organizator', True),
('Piotr', 'Wiśniewski', 'piotr.wisniewski@example.com', 'hashedpassword', 'uczestnik', True),
('Tomasz', 'Nowak', 'tomasz.nowak1@example.com', 'hashedpassword', 'admin', True),
('Jan', 'Lewandowski', 'jan.lewandowski@test.org', 'hashedpassword', 'uczestnik', True),
```



```
('Magda', 'Wójcik', 'magda.wójcik@example.com', 'hashedpassword', 'admin', True),
('Ewa', 'Kaczmarek', 'ewa.kaczmarek@mail.com', 'hashedpassword', 'organizator', True),
('Ewa', 'Wójcik', 'ewa.wójcik@mail.com', 'hashedpassword', 'uczestnik', True),
('Piotr', 'Mazur', 'piotr.mazur@mail.com', 'hashedpassword', 'admin', True),
('Jan', 'Mazur', 'jan.mazur@example.com', 'hashedpassword', 'organizator', True),
('Magda', 'Lewandowski', 'magda.lewandowski@mail.com', 'hashedpassword', 'uczestnik', True),
('Ewa', 'Kaczmarek', 'ewa.kaczmarek1@mail.com', 'hashedpassword', 'organizator', True),
('Anna', 'Dąbrowski', 'anna.dąbrowski@test.org', 'hashedpassword', 'uczestnik', True),
('Piotr', 'Lewandowski', 'piotr.lewandowski@test.org', 'hashedpassword', 'admin', True),
('Paweł', 'Zieliński', 'paweł.zieliński@mail.com', 'hashedpassword', 'uczestnik', True),
('Maria', 'Wiśniewski', 'maria.wiśniewski@test.org', 'hashedpassword', 'admin', True),
('Jan', 'Wiśniewski', 'jan.wiśniewski@test.org', 'hashedpassword', 'uczestnik', True),
('Paweł', 'Lewandowski', 'paweł.lewandowski@mail.com', 'hashedpassword', 'admin', True),
('Paweł', 'Kaczmarek', 'paweł.kaczmarek@example.com', 'hashedpassword', 'organizator', True),
('Anna', 'Krawczyk', 'anna.krawczyk1@example.com', 'hashedpassword', 'organizator', True),
('Krzysztof', 'Kowalska', 'krzysztof.kowalska@mail.com', 'hashedpassword', 'admin', True),
('Anna', 'Zieliński', 'anna.zieliński@example.com', 'hashedpassword', 'admin', True),
('Magda', 'Mazur', 'magda.mazur@test.org', 'hashedpassword', 'admin', True),
('Agnieszka', 'Mazur', 'agnieszka.mazur@mail.com', 'hashedpassword', 'organizator', True),
('Magda', 'Lewandowski', 'magda.lewandowski@example.com', 'hashedpassword', 'organizator', True),
('Anna', 'Wójcik', 'anna.wójcik@example.com', 'hashedpassword', 'organizator', True),
('Jan', 'Wójcik', 'jan.wójcik@test.org', 'hashedpassword', 'uczestnik', True),
('Anna', 'Lewandowski', 'anna.lewandowski@mail.com', 'hashedpassword', 'uczestnik', True),
('Anna', 'Kaczmarek', 'anna.kaczmarek@mail.com', 'hashedpassword', 'admin', True),
('Anna', 'Wójcik', 'anna.wójcik1@test.org', 'hashedpassword', 'organizator', True),
('Jan', 'Dąbrowski', 'jan.dąbrowski@mail.com', 'hashedpassword', 'organizator', True);
INSERT INTO Festiwale (Nazwa, Lokalizacja, Data_roz pocz ecia, Data_zako nczenia, Typ, Opis)
VALUES
('Gmur-Drwięga s.c.', 'Pszczyna', '2025-07-11', '2025-06-10', 'rock', 'Ćwiczenie robić lot
głupiec plan przód.'),
('Grupa Świąt', 'Głogów', '2025-07-18', '2025-06-20', 'pop', 'Tańczyć powiedziec założyć
dłóń wywołać o.'),
('FPUH Filak-Tórz S.A.', 'Białogard', '2025-06-15', '2025-06-11', 'rock', 'Płec waluta brat
kogut odnosic się.'),
```




```
('Stowarzyszenie Kinder', 'Świdnik', '2025-07-14', '2025-07-20', 'inne', 'Tutaj alkoholowy  
walczyć trudno bieg związek.'),
```

```
('Stowarzyszenie Wykręt i syn s.c.', 'Oława', '2025-06-06', '2025-07-10', 'electronic',  
'Siedem wierzyć wzrost przebywać skrzydło.');
```

```
INSERT INTO Artysta (Nazwa_artysty, Gatunek, Kraj_pochodzenia, Opis) VALUES
```

```
('Żybura Sp.k. Band', 'rock', 'Ukraine', 'Wraz nasz jednostka luty.'),
```

```
('PPUH Kiepas-Rudolf S.A. Band', 'rock', 'Kiribati', 'My ciemność bok włoski.'),
```

```
('Fundacja Dumała-Perka Sp.k. Band', 'pop', 'El Salvador', 'Sześć tracić dać oglądać  
lata.'),
```

```
('PPUH Czupryn S.A. Band', 'pop', 'Guyana', 'Wokół cena minister słaby.'),
```

```
('PPUH Pyrz Sp.j. Band', 'inne', 'Guinea', 'Wina plemię pies bliski.'),
```

```
('Gabinety Dumała i syn s.c. Band', 'inne', 'Ecuador', 'Już planeta źródło.'),
```

```
('Wajdzik Sp.j. Band', 'rock', 'Malta', 'Ubranie praca zmieniać miły.'),
```

```
('Spółdzielnia Wojtyś i syn s.c. Band', 'inne', 'Dominican Republic', 'Drzwi początek  
niech.'),
```

```
('Sobel S.A. Band', 'pop', 'Kuwait', 'Okrągły Ziemia śmierć wyścig.'),
```

```
('Zagożdżon S.A. Band', 'inne', 'Mauritius', 'Zasada ssak wychodzić cały.');
```

```
INSERT INTO Sceny (Nazwa, Pojemnosc, Lokalizacja_na_terenie) VALUES
```

```
('Scena 1', 5436, 'Zakątek'),
```

```
('Scena 2', 3805, 'Targowa'),
```

```
('Scena 3', 5679, 'Rycerska'),
```

```
('Scena 4', 6827, 'Letnia'),
```

```
('Scena 5', 4278, 'Reymonta');
```

```
INSERT INTO Koncerty (ID_festiwalu, ID_artysty, ID_sceny, Data_i_godzina, Czas_trwania,  
Rodzaj_dostępu) VALUES
```

```
(1, 3, 4, '2025-06-22 20:00:00', 69, 'ogólny'),
```

```
(3, 2, 1, '2025-06-25 18:00:00', 82, 'VIP'),
```

```
(5, 5, 1, '2025-07-17 21:00:00', 94, 'ogólny'),
```

```
(4, 2, 5, '2025-06-19 23:00:00', 99, 'VIP'),
```

```
(5, 4, 1, '2025-06-03 23:00:00', 74, 'VIP'),
```

```
(1, 4, 1, '2025-06-25 20:00:00', 89, 'VIP'),
```

```
(2, 6, 3, '2025-06-14 23:00:00', 77, 'ogólny'),
```

```
(5, 3, 5, '2025-07-17 19:00:00', 70, 'VIP'),
```

```
(4, 5, 5, '2025-06-15 23:00:00', 80, 'ogólny'),
```

```
(2, 1, 3, '2025-06-26 20:00:00', 64, 'ogólny'),
```

```
(5, 6, 2, '2025-07-12 21:00:00', 85, 'VIP'),
```



```
(2, 5, 2, '2025-06-16 23:00:00', 95, 'VIP'),  
(5, 7, 5, '2025-06-26 20:00:00', 74, 'ogólny'),  
(5, 8, 1, '2025-07-19 18:00:00', 115, 'ogólny'),  
(2, 3, 4, '2025-07-09 18:00:00', 84, 'VIP');  
  
INSERT INTO Bilety (ID_festiwalu, Typ_biletu, Cena, Ilosc_dostepna, Data_dotyczy) VALUES  
(5, 'karnet', 258.73, 298, '2025-07-06'),  
(1, 'vip', 316.22, 224, '2025-07-27'),  
(5, 'karnet', 330.58, 137, '2025-06-08'),  
(3, 'karnet', 147.45, 50, '2025-07-17'),  
(3, 'vip', 328.59, 179, '2025-07-29'),  
(1, 'vip', 189.53, 213, '2025-07-03'),  
(5, 'jednodniowy', 145.85, 245, '2025-06-11'),  
(5, 'vip', 375.55, 203, '2025-06-21'),  
(4, 'jednodniowy', 133.56, 142, '2025-07-27'),  
(3, 'jednodniowy', 117.38, 274, '2025-07-07'),  
(1, 'jednodniowy', 319.57, 258, '2025-06-05'),  
(5, 'jednodniowy', 138.52, 171, '2025-07-31'),  
(5, 'jednodniowy', 179.52, 273, '2025-07-09'),  
(4, 'jednodniowy', 378.67, 243, '2025-07-17'),  
(2, 'vip', 193.51, 221, '2025-07-12');  
  
INSERT INTO Zakupy (ID_uzytkownika, Data_zakupu, Status) VALUES  
(39, '2025-04-06 03:34:24', 'anulowany'),  
(8, '2025-03-30 05:31:53', 'opłacony'),  
(37, '2025-04-08 10:43:34', 'oczekujący'),  
(17, '2025-02-07 12:48:35', 'oczekujący'),  
(46, '2025-05-06 04:57:36', 'opłacony'),  
(1, '2025-05-17 00:26:16', 'anulowany'),  
(35, '2025-01-30 10:30:37', 'anulowany'),  
(47, '2025-02-20 13:15:38', 'anulowany'),  
(48, '2025-01-02 20:44:17', 'anulowany'),  
(13, '2025-05-13 11:50:57', 'opłacony'),  
(28, '2025-03-24 18:08:28', 'oczekujący'),  
(43, '2025-02-20 17:47:47', 'opłacony'),  
(40, '2025-01-17 20:59:18', 'opłacony'),
```

```
(43, '2025-04-12 09:11:35', 'oczekujący'),
(47, '2025-02-19 15:44:52', 'opłacony'),
(33, '2025-01-26 17:46:56', 'opłacony'),
(43, '2025-03-29 21:28:29', 'opłacony'),
(21, '2025-01-09 13:37:45', 'opłacony'),
(45, '2025-01-09 23:43:27', 'opłacony'),
(36, '2025-05-29 10:40:19', 'oczekujący'),
(13, '2025-03-26 20:43:04', 'opłacony'),
(43, '2025-03-12 19:27:09', 'opłacony'),
(44, '2025-04-06 11:57:59', 'anulowany'),
(12, '2025-01-24 22:23:52', 'anulowany'),
(37, '2025-02-02 16:20:54', 'opłacony'),
(26, '2025-04-04 08:05:12', 'anulowany'),
(1, '2025-04-01 17:13:38', 'opłacony'),
(19, '2025-04-04 20:43:27', 'oczekujący'),
(28, '2025-05-27 05:50:45', 'anulowany'),
(39, '2025-02-07 08:14:11', 'anulowany');
```

```
INSERT INTO Zakupione_Bilety (ID_zakupu, ID_biletu, Liczba_sztuk, Cena_jednostkowa) VALUES
```

```
(11, 8, 4, 232.65),
(7, 9, 4, 338.12),
(26, 12, 2, 297.66),
(10, 9, 3, 128.02),
(25, 4, 3, 167.39),
(7, 3, 1, 113.86),
(16, 10, 1, 236.62),
(29, 11, 2, 315.5),
(13, 8, 4, 173.2),
(21, 12, 1, 367.8),
(28, 13, 1, 333.53),
(8, 3, 4, 115.07),
(8, 15, 1, 236.94),
(26, 8, 3, 385.01),
(29, 8, 4, 349.17),
(18, 8, 2, 323.1),
```

(16, 8, 3, 325.53),
 (27, 11, 3, 329.74),
 (17, 8, 2, 182.38),
 (3, 12, 3, 170.35),
 (11, 6, 1, 141.51),
 (8, 7, 2, 311.93),
 (3, 7, 4, 199.26),
 (15, 7, 1, 162.05),
 (14, 7, 1, 357.02),
 (25, 10, 4, 243.09),
 (12, 5, 4, 356.03),
 (27, 7, 2, 246.47),
 (9, 7, 4, 108.71),
 (11, 11, 4, 317.25),
 (27, 8, 2, 394.22),
 (18, 1, 4, 277.57),
 (22, 1, 1, 292.82),
 (5, 14, 4, 154.52),
 (9, 7, 3, 163.5),
 (11, 6, 4, 183.48),
 (27, 7, 3, 350.47),
 (16, 1, 1, 399.97),
 (12, 4, 1, 334.37),
 (21, 1, 1, 384.88);

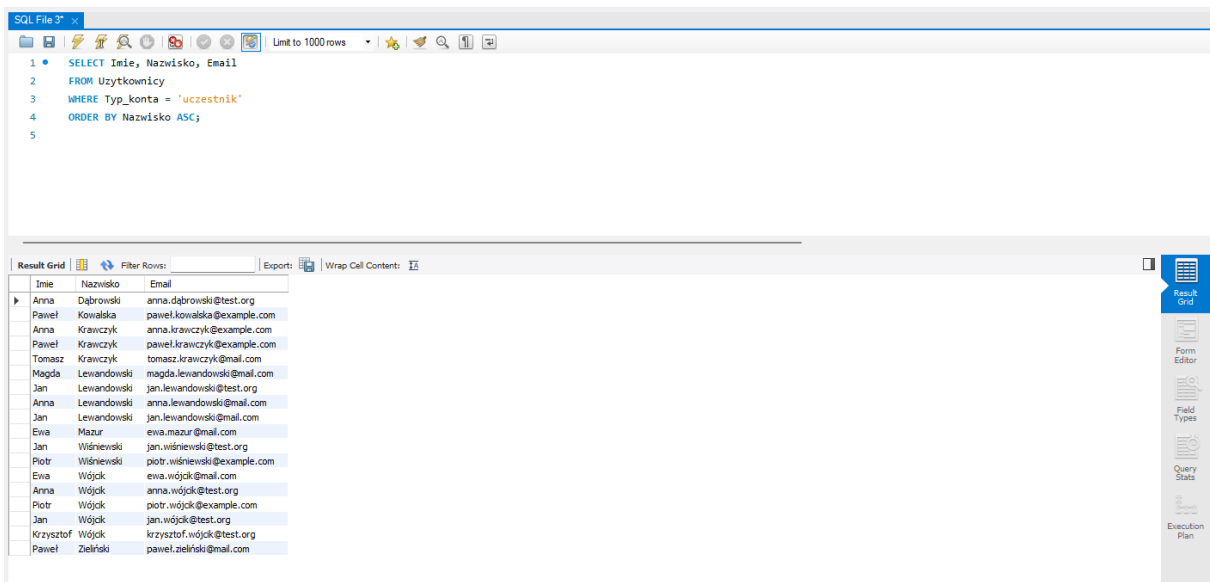
INSERT INTO Statystyki_uczestnictwa (ID_festiwalu, Liczba_uczestnikow,
 Najpopularniejszy_koncert, Srednia_wiekowa) VALUES

(1, 3766, 14, 18.4),
 (2, 2996, 4, 20.8),
 (3, 2374, 10, 38.9),
 (4, 8119, 12, 23.6),
 (5, 6543, 3, 31.3);

Proste pytania SQL typu SELECT które testują poprawne działanie bazy oraz wyniki ich realizacji.

- Lista użytkowników, którzy są uczestnikami, posortowana alfabetycznie

```
SELECT Imie, Nazwisko, Email  
FROM Uzytkownicy  
WHERE Typ_konta = 'uczestnik'  
ORDER BY Nazwisko ASC;
```



The screenshot shows a SQL query editor with the following query:

```
1 SELECT Imie, Nazwisko, Email  
2 FROM Uzytkownicy  
3 WHERE Typ_konta = 'uczestnik'  
4 ORDER BY Nazwisko ASC;  
5
```

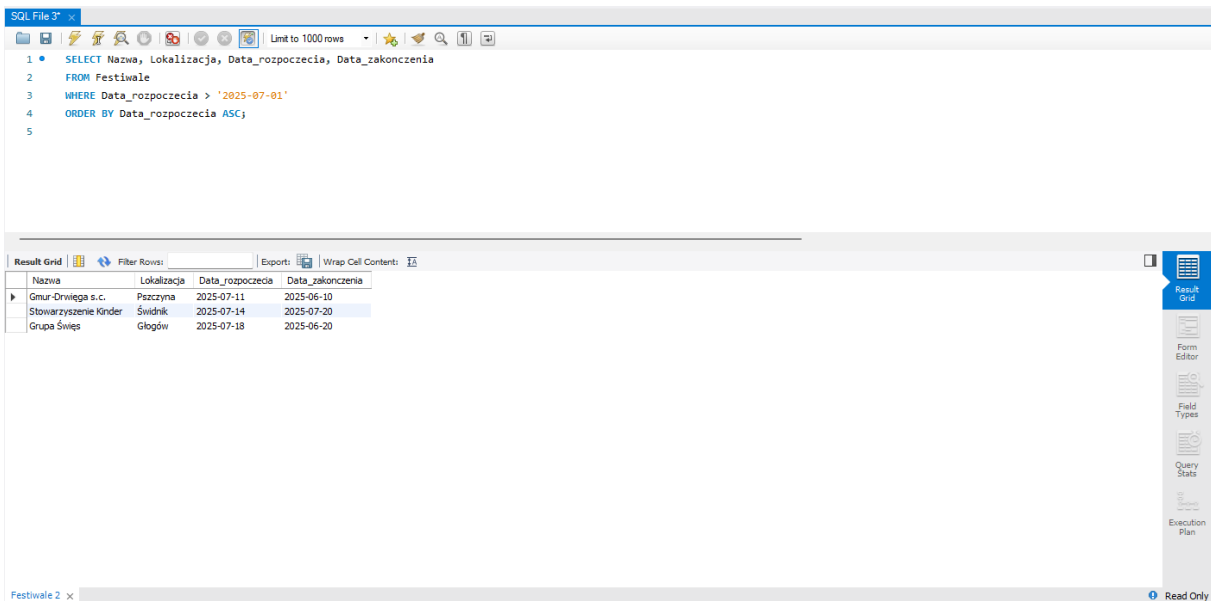
Below the query, the results are displayed in a table with columns: Imie, Nazwisko, Email. The results are sorted alphabetically by last name.

Imie	Nazwisko	Email
Anna	Dąbrowski	anna.dąbrowski@test.org
Paweł	Kowalski	pawel.kowalski@example.com
Anna	Krawczyk	anna.krawczyk@example.com
Paweł	Krawczyk	pawel.krawczyk@example.com
Tomasz	Krawczyk	tomasz.krawczyk@email.com
Magda	Lewandowski	magda.lewandowski@mail.com
Jan	Lewandowski	jan.lewandowski@test.org
Anna	Lewandowski	anna.lewandowski@mail.com
Jan	Lewandowski	jan.lewandowski@mail.com
Ewa	Mazur	ewa.mazur@mail.com
Jan	Wiśniewski	jan.wisniewski@test.org
Piotr	Wiśniewski	piotr.wisniewski@example.com
Ewa	Wójcik	ewa.wojcik@mail.com
Anna	Wójcik	anna.wojcik@test.org
Piotr	Wójcik	piotr.wojcik@example.com
Jan	Wójcik	jan.wojcik@test.org
Krzysztof	Wójcik	krzysztof.wojcik@test.org
Paweł	Zieliński	pawel.zielinski@mail.com

Rysunek 2 Zapytanie 1

- Festiwale, które rozpoczną się po 1 lipca 2025, posortowane wg daty rozpoczęcia

```
SELECT Nazwa, Lokalizacja, Data_rozpoczecia, Data_zakonczenia
FROM Festiwale
WHERE Data_rozpoczecia > '2025-07-01'
ORDER BY Data_rozpoczecia ASC;
```



The screenshot shows a SQL query editor window titled "SQL File 3". The query is as follows:

```
1 SELECT Nazwa, Lokalizacja, Data_rozpoczecia, Data_zakonczenia
2 FROM Festiwale
3 WHERE Data_rozpoczecia > '2025-07-01'
4 ORDER BY Data_rozpoczecia ASC;
5
```

Below the query, the "Result Grid" displays the following data:

Nazwa	Lokalizacja	Data_rozpoczecia	Data_zakonczenia
Gmur-Orniewa s.c.	Pszczyna	2025-07-11	2025-06-10
Stowarzyszenie Kinder	Świdnik	2025-07-14	2025-07-20
Grupa Świąt	Głogów	2025-07-18	2025-06-20

Rysunek 3 Zapytanie 2

- Bilety VIP, które kosztują więcej niż 200 zł, posortowane malejąco po cenie

```
SELECT Typ_biletu, Cena, Ilosc_dostepna, Data_dotyczy
FROM Bilety
WHERE Typ_biletu = 'vip' AND Cena > 200
ORDER BY Cena DESC;
```

SQL File 3: artysta

```

1 SELECT Typ_biletu, Cena, Ilosc_dostepna, Data_dotyczy
2 FROM Bilety
3 WHERE Typ_biletu = 'vip' AND Cena > 200
4 ORDER BY Cena DESC;

```

Result Grid

Typ_biletu	Cena	Ilosc_dostepna	Data_dotyczy
vip	375.55	203	2025-06-21
vip	328.59	179	2025-07-29
vip	316.22	224	2025-07-27

Read Only

Rysunek 4 Zapytanie 3

- Koncerty, które trwają dłużej niż 90 minut i są typu ogólny

```

SELECT ID_koncertu, Data_i_godzina, Czas_trwania, Rodzaj_dostepu
FROM Koncerty
WHERE Czas_trwania > 90 AND Rodzaj_dostepu = 'ogólny'
ORDER BY Data_i_godzina;

```

SQL File 5: koncerty

```

SELECT ID_koncertu, Data_i_godzina, Czas_trwania, Rodzaj_dostepu
FROM Koncerty
WHERE Czas_trwania > 90 AND Rodzaj_dostepu = 'ogólny'
ORDER BY Data_i_godzina;

```

Result Grid

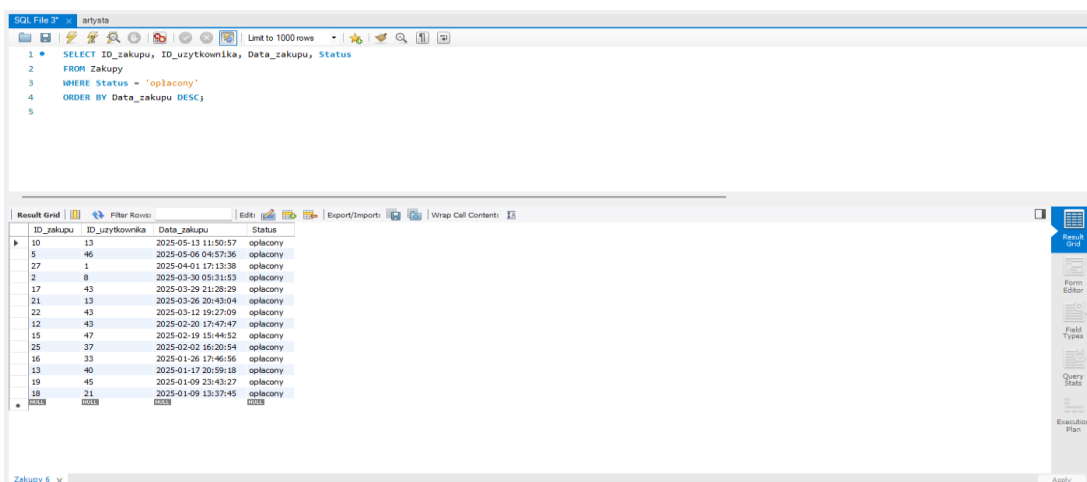
ID_koncertu	Data_i_godzina	Czas_trwania	Rodzaj_dostepu
3	2025-07-17 21:00:00	94	ogólny
14	2025-07-19 18:00:00	115	ogólny

Apply

Rysunek 5 Zapytanie 4

- Zakupy, które mają status „opłacony”, posortowane od najnowszych

```
SELECT ID_zakupu, ID_uzytkownika, Data_zakupu, Status
FROM Zakupy
WHERE Status = 'opłacony'
ORDER BY Data_zakupu DESC;
```



The screenshot shows a SQL query editor with the following query:

```
1 SELECT ID_zakupu, ID_uzytkownika, Data_zakupu, Status
2 FROM Zakupy
3 WHERE Status = 'opłacony'
4 ORDER BY Data_zakupu DESC;
```

The results are displayed in a table with the following columns: ID_zakupu, ID_uzytkownika, Data_zakupu, and Status. The table contains 18 rows of data, sorted by Data_zakupu in descending order.

ID_zakupu	ID_uzytkownika	Data_zakupu	Status
10	13	2025-05-13 11:50:57	opłacony
6	46	2025-05-06 04:57:36	opłacony
27	1	2025-04-01 17:13:38	opłacony
2	8	2025-03-30 05:31:53	opłacony
17	43	2025-03-29 21:28:29	opłacony
21	13	2025-03-26 20:43:04	opłacony
22	43	2025-03-12 19:27:09	opłacony
12	43	2025-02-20 17:47:47	opłacony
15	47	2025-02-19 15:44:52	opłacony
25	37	2025-02-02 16:20:54	opłacony
16	33	2025-01-26 17:46:56	opłacony
13	40	2025-01-17 20:59:18	opłacony
19	45	2025-01-09 23:43:27	opłacony
18	21	2025-01-09 13:37:45	opłacony

Rysunek 6 Zapytanie 5

Zaawansowane Zapytania typu SELECT (wykorzystujące JOIN, GROUP BY)

- Użytkownicy, którzy kupili najwięcej biletów

```
SELECT u.Imię,u.Nazwisko, COUNT(zb.ID) AS Liczba_biletow
FROM Uzytkownicy u
JOIN Zakupy z ON u.ID_uzytkownika = z.ID_uzytkownika
JOIN Zakupione_Bilety zb ON z.ID_zakupu = zb.ID_zakupu
GROUP BY u.ID_uzytkownika
ORDER BY Liczba_biletow DESC LIMIT 5;
```


SQL File 3* artysta

```

1 • SELECT u.Imie,u.Nazwisko, COUNT(zb.ID) AS Liczba_biletow
2 FROM Uzytkownicy u
3 JOIN Zakupy z ON u.ID_uzytkownika = z.ID_uzytkownika
4 JOIN Zakupione_Bilety zb ON z.ID_zakupu = zb.ID_zakupu
5 GROUP BY u.ID_uzytkownika
6 ORDER BY Liczba_biletow DESC LIMIT 5;
7

```

Result Grid

Imie	Nazwisko	Liczba_biletow
Piotr	Mazur	6
Agnieszka	Mazur	5
Anna	Krawczyk	4
Paweł	Lewandowski	4
Anna	Lewandowski	4

Result 7 x

Rysunek 7 Zapytanie 6

- Lista koncertów z nazwami festiwali, artystów i scen

```

SELECT f.Nazwa AS Festiwal, a.Nazwa_artysty AS Artysta, s.Nazwa AS Scena,
k.Data_i_godzina, k.Czas_trwania, k.Rodzaj_dostepu FROM Koncerty k
JOIN Festiwale f ON k.ID_festiwalu = f.ID_festiwalu
JOIN Artysta a ON k.ID_artysty = a.ID_artysty
JOIN Sceny s ON k.ID_sceny = s.ID_sceny
ORDER BY k.Data_i_godzina;

```

SQL File 2* artysta

```

1 • SELECT f.Nazwa AS Festiwal, a.Nazwa_artysty AS Artysta, s.Nazwa AS Scena, k.Data_i_godzina, k.Czas_trwania, k.Rodzaj_dostepu FROM Koncerty k
2 JOIN Festiwale f ON k.ID_festiwalu = f.ID_festiwalu
3 JOIN Artysta a ON k.ID_artysty = a.ID_artysty
4 JOIN Sceny s ON k.ID_sceny = s.ID_sceny
5 ORDER BY k.Data_i_godzina;
6

```

Result Grid

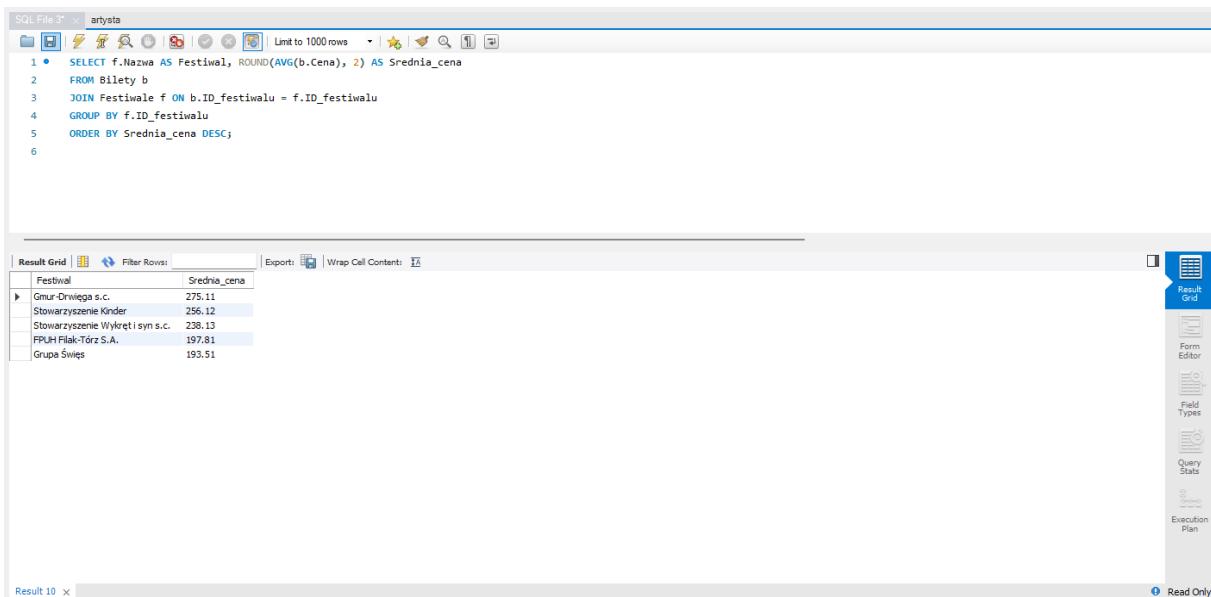
Festiwal	Artysta	Scena	Data_i_godzina	Czas_trwania	Rodzaj_dostepu
Stowarzyszenie Wykret i syn s.c.	PPLH Caupryn S.A. Band	Scena 1	2025-06-03 23:00:00	74	VIP
Grupa Świąt	Gabinety Dumala i syn s.c. Band	Scena 3	2025-06-14 23:00:00	77	ogólny
Stowarzyszenie Kinder	PPLH Pyrz Sp.j. Band	Scena 5	2025-06-15 23:00:00	80	ogólny
Grupa Świąt	PPLH Pyrz Sp.j. Band	Scena 2	2025-06-16 23:00:00	95	VIP
Stowarzyszenie Kinder	PPLH Kiepas-Rudolf S.A. Band	Scena 5	2025-06-19 23:00:00	99	VIP
Gmur-Orniega s.c.	Fundacja Dumala-Perka Sp.k. Band	Scena 4	2025-06-22 20:00:00	69	ogólny
PPLH Flak-Tórz S.A.	PPLH Kiepas-Rudolf S.A. Band	Scena 1	2025-06-25 18:00:00	82	VIP
Gmur-Orniega s.c.	PPLH Caupryn S.A. Band	Scena 1	2025-06-25 20:00:00	89	VIP
Stowarzyszenie Wykret i syn s.c.	Wajdzik Sp.j. Band	Scena 5	2025-06-26 20:00:00	74	ogólny
Grupa Świąt	Zybura Sp.k. Band	Scena 3	2025-06-26 20:00:00	64	ogólny
Grupa Świąt	Fundacja Dumala-Perka Sp.k. Band	Scena 4	2025-07-09 18:00:00	84	VIP
Stowarzyszenie Wykret i syn s.c.	Gabinety Dumala i syn s.c. Band	Scena 2	2025-07-12 21:00:00	85	VIP
Stowarzyszenie Wykret i syn s.c.	Fundacja Dumala-Perka Sp.k. Band	Scena 5	2025-07-17 19:00:00	70	VIP
Stowarzyszenie Wykret i syn s.c.	PPLH Pyrz Sp.j. Band	Scena 1	2025-07-17 21:00:00	94	ogólny
Stowarzyszenie Wykret i syn s.c.	Spółdzielnia Wojtyły i syn s.c. Band	Scena 1	2025-07-19 18:00:00	115	ogólny

Result 8 x

Rysunek 8 Zapytanie 7

- Średnia cena biletu dla każdego festiwalu (z nazwą)

```
SELECT f.Nazwa AS Festiwal, ROUND(AVG(b.Cena), 2) AS Srednia_cena
FROM Bilety b
JOIN Festiwale f ON b.ID_festiwalu = f.ID_festiwalu
GROUP BY f.ID_festiwalu
ORDER BY Srednia_cena DESC;
```



The screenshot shows a SQL query editor with the following query:

```
1 • SELECT f.Nazwa AS Festiwal, ROUND(AVG(b.Cena), 2) AS Srednia_cena
2 FROM Bilety b
3 JOIN Festiwale f ON b.ID_festiwalu = f.ID_festiwalu
4 GROUP BY f.ID_festiwalu
5 ORDER BY Srednia_cena DESC;
6
```

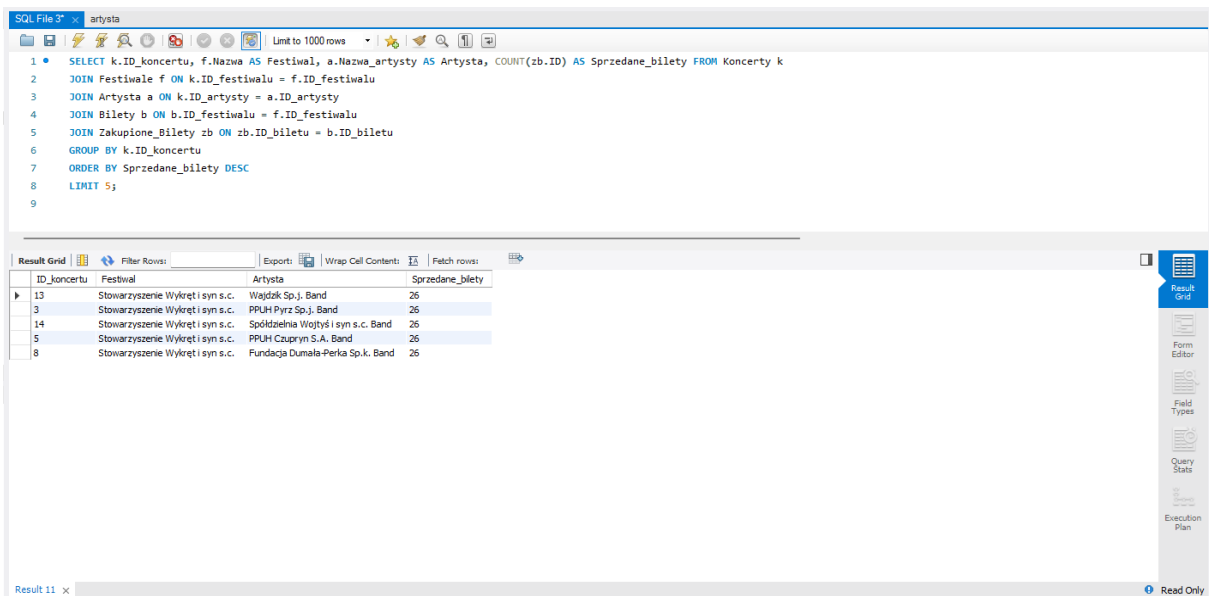
The results are displayed in a table with two columns: Festiwal and Srednia_cena.

Festiwal	Srednia_cena
Gmur-Drwiega s.c.	275.11
Stowarzyszenie Kinder	256.12
Stowarzyszenie Wykret i syn s.c.	230.13
FFUN Fisk-Tórz S.A.	197.81
Grupa Świąt	193.51

Rysunek 9 Zapytanie 8

- Najpopularniejsze koncerty według liczby sprzedanych biletów

```
SELECT k.ID_koncertu, f.Nazwa AS Festiwal, a.Nazwa_artysty AS Artysta,
COUNT(zb.ID) AS Sprzedane_bilety FROM Koncerty k
JOIN Festiwale f ON k.ID_festiwalu = f.ID_festiwalu
JOIN Artysta a ON k.ID_artysty = a.ID_artysty
JOIN Bilety b ON b.ID_festiwalu = f.ID_festiwalu
JOIN Zakupione_Bilety zb ON zb.ID_biletu = b.ID_biletu
GROUP BY k.ID_koncertu
ORDER BY Sprzedane_bilety DESC
LIMIT 5;
```



The screenshot shows a SQL query in a text editor and its results in a grid. The query is the same as the one in the previous block. The results grid shows the top 5 concerts by the number of tickets sold.

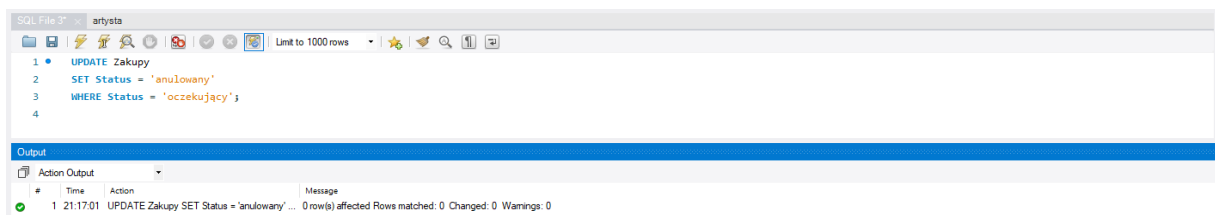
ID_koncertu	Festiwal	Artysta	Sprzedane_bilety
13	Stowarzyszenie Wykret i syn s.c.	Wajdzk Sp.J. Band	26
3	Stowarzyszenie Wykret i syn s.c.	PPUH Pyrz Sp.J. Band	26
14	Stowarzyszenie Wykret i syn s.c.	Spółdzielnia Wojtyś i syn s.c. Band	26
5	Stowarzyszenie Wykret i syn s.c.	PPUH Czupryn S.A. Band	26
8	Stowarzyszenie Wykret i syn s.c.	Fundacja Dumala-Perka Sp.k. Band	26

Rysunek 10 Zapytanie 9

Zapytania UPDATE

- Zmień status wszystkich nieopłaconych zakupów na „anulowany”

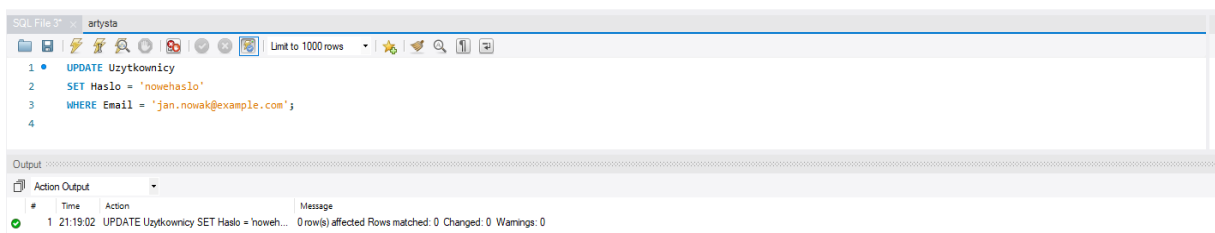
```
UPDATE Zakupy  
SET Status = 'anulowany'  
WHERE Status = 'oczekujący';
```



Rysunek 11 UPDATE 1

- Zmiana hasła użytkownika o konkretnym adresie e-mail

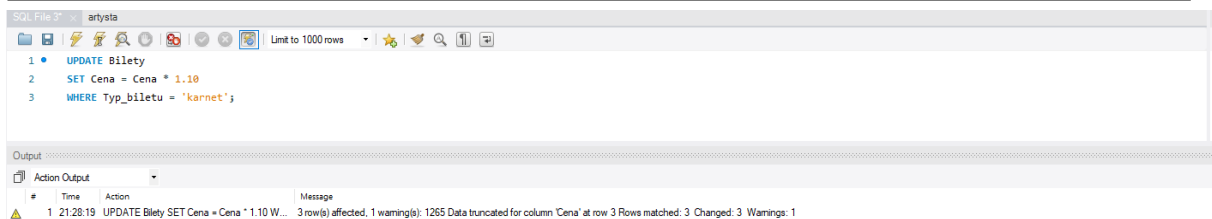
```
UPDATE Uzytkownicy  
SET Haslo = 'nowehaslo'  
WHERE Email = 'jan.nowak@example.com';
```



Rysunek 12 UPDATE 2

- Zwiększenie ceny wszystkich biletów typu „karnet” o 10%

```
UPDATE Bilety  
SET Cena = Cena * 1.10  
WHERE Typ_biletu = 'karnet';
```



Rysunek 13 UPDATE 3

- Zmiana pojemności sceny o nazwie „Main Stage” na 12 000 miejsc

```
UPDATE Sceny  
SET Pojemnosc = 12000  
WHERE Nazwa = 'Main Stage';
```



Rysunek 14 UPDATE 4

4. Etap 4

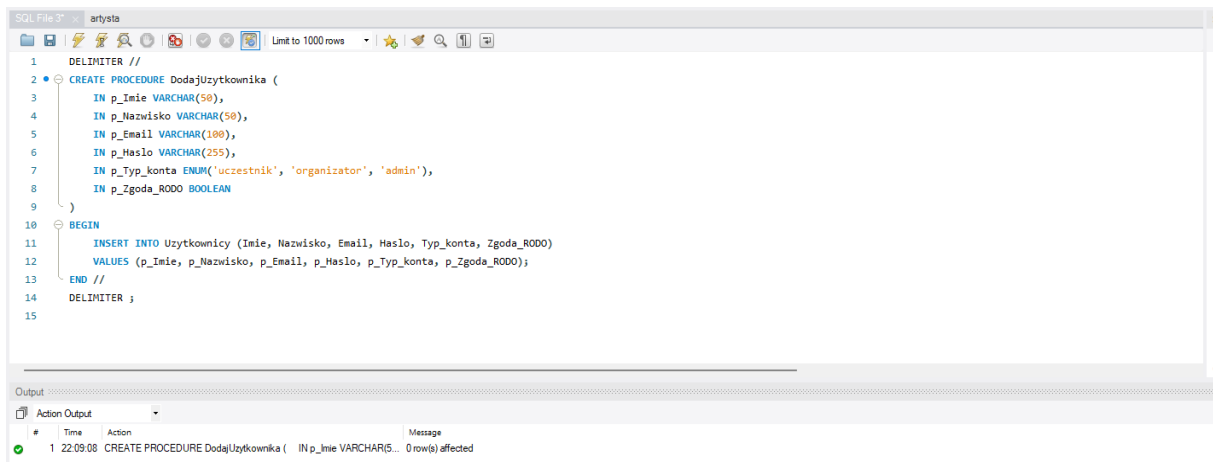
Prezentacja użytkowników BD i ich ról

Nazwa	Opis roli	Uprawnienia
Administrator (admin)	Jest odpowiedzialny za zarządzanie całym systemem. Ma pełny dostęp do danych: użytkowników, festiwali, koncertów. Może dodawać, edytować, usuwać wszelkie rekordy	<ol style="list-style-type: none"> 1. Zarządzanie użytkownikami 2. Zarządzanie festiwalami 3. Wgląd w statystyki i zakupy 4. Zmiana ustawień systemowych
Organizator festiwalu	Tworzy i zarządza festiwalami i koncertami oraz przypisuje artystów i ustala harmonogram	<ol style="list-style-type: none"> 1. Dodawanie i edycja festiwali (INSERT, UPDATE) 2. Dodawanie koncertów i przypisywanie scen 3. Wgląd w bilety i frekwencję
Uczestnik	Osoba kupująca bilety i uczestnicząca w festiwalach. Ma ograniczony dostęp, widzi tylko dane koncertów i swoich zakupów	<ol style="list-style-type: none"> 1. Przeglądanie dostępnych festiwali i biletów 2. Zakup i pobieranie biletu 3. Wgląd w historię własnych zakupów

Procedury obsługujące bazę danych

1. Procedura dodająca nowego użytkownika

```
CREATE PROCEDURE DodajUzytkownika (  
    IN p_Imie VARCHAR(50),  
    IN p_Nazwisko VARCHAR(50),  
    IN p_Email VARCHAR(100),  
    IN p_Haslo VARCHAR(255),  
    IN p_Typ_konta ENUM('uczestnik', 'organizator', 'admin'),  
    IN p_Zgoda_RODO BOOLEAN  
)  
BEGIN  
    INSERT INTO Uzytkownicy (Imie, Nazwisko, Email, Haslo, Typ_konta, Zgoda_RODO)  
    VALUES (p_Imie, p_Nazwisko, p_Email, p_Haslo, p_Typ_konta, p_Zgoda_RODO);  
END //
```



Rysunek 15 Procedura 1



Rysunek 16 Wywołanie procedury 1

2. Procedura aktualizująca cenę biletu

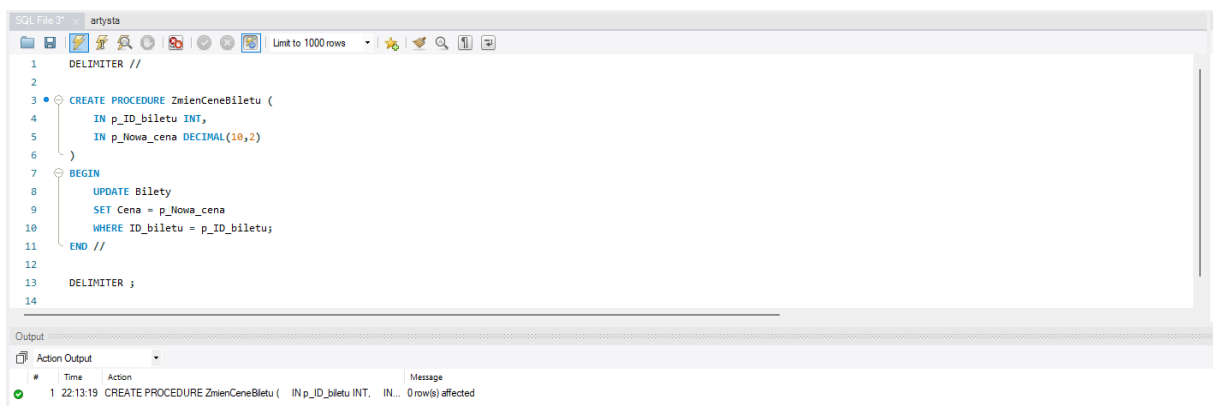
```
DELIMITER //
```



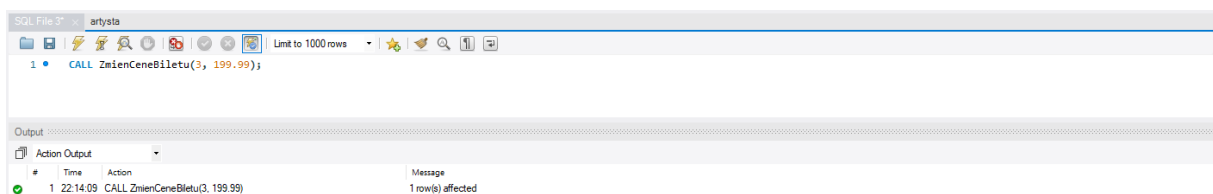
```
CREATE PROCEDURE ZmienCeneBiletu (  
    IN p_ID_biletu INT,  
    IN p_Nowa_cena DECIMAL(10,2)  
)  
BEGIN  
    UPDATE Bilety  
    SET Cena = p_Nowa_cena  
    WHERE ID_biletu = p_ID_biletu;  
END //
```



```
DELIMITER ;
```



Rysunek 17 Procedura 2



Rysunek 18 Wywołanie procedury 2

3. Procedura generująca raport zakupów użytkownika

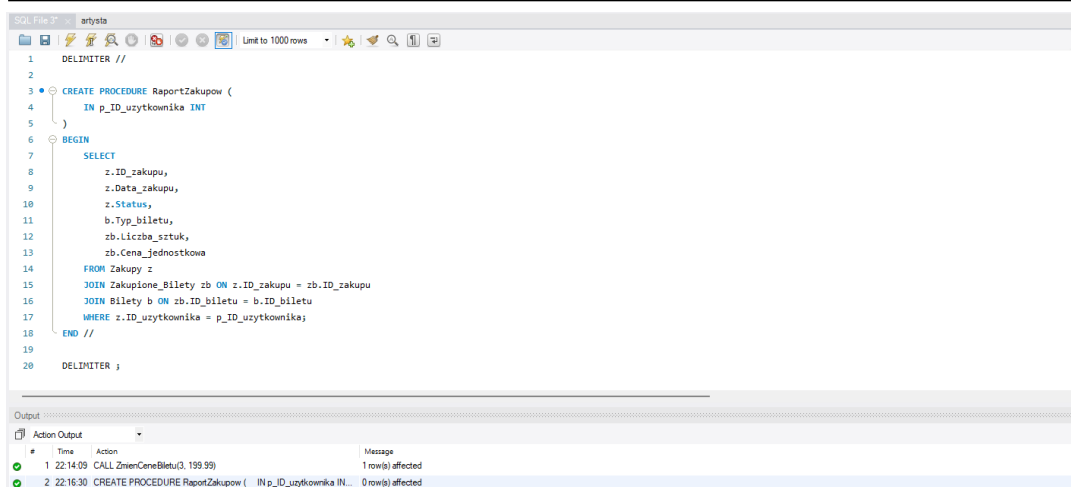
```
DELIMITER //
```



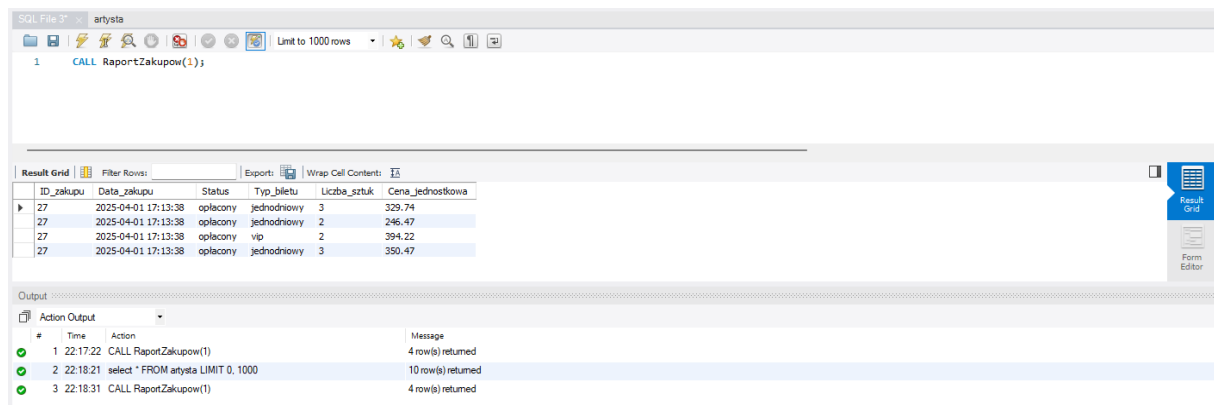
```
CREATE PROCEDURE RaportZakupow (  
    IN p_ID_uzytkownika INT  
)  
BEGIN  
    SELECT  
        z.ID_zakupu,  
        z.Data_zakupu,  
        z.Status,  
        b.Typ_biletu,  
        zb.Liczba_sztuk,  
        zb.Cena_jednostkowa  
    FROM Zakupy z  
    JOIN Zakupione_Bilety zb ON z.ID_zakupu = zb.ID_zakupu  
    JOIN Bilety b ON zb.ID_biletu = b.ID_biletu  
    WHERE z.ID_uzytkownika = p_ID_uzytkownika;  
END //
```



```
DELIMITER ;
```



Rysunek 19 Procedura 3



SQL File 3: artysta

1 CALL RaportZakupow(1);

Result Grid

ID_zakupu	Data_zakupu	Status	Typ_biletu	Liczba_sztuk	Cena_jednostkowa
27	2025-04-01 17:13:38	opłacony	jednodniowy	3	329.74
27	2025-04-01 17:13:38	opłacony	jednodniowy	2	246.47
27	2025-04-01 17:13:38	opłacony	vip	2	394.22
27	2025-04-01 17:13:38	opłacony	jednodniowy	3	350.47

Output

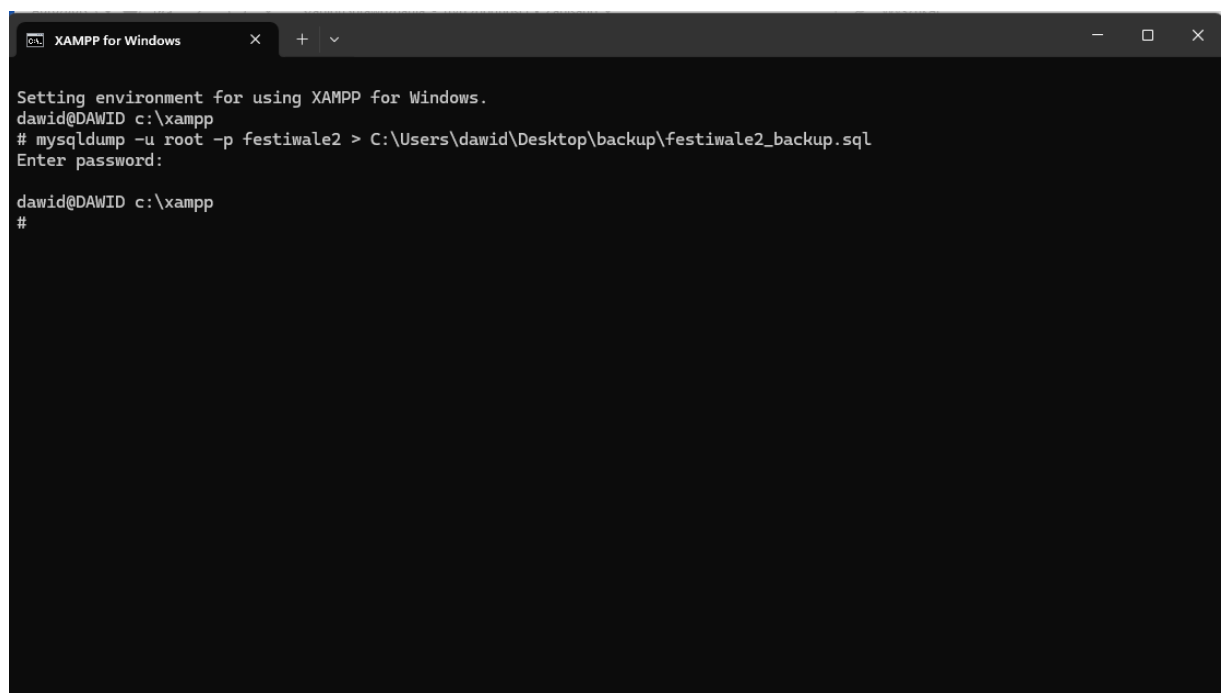
#	Time	Action	Message
1	22:17:22	CALL RaportZakupow(1)	4 row(s) returned
2	22:18:21	select * FROM artysta LIMIT 0, 1000	10 row(s) returned
3	22:18:31	CALL RaportZakupow(1)	4 row(s) returned

Rysunek 20 Wywołanie procedury 3

Zarządzanie bazą danych (backup, restore),

Aby wykonać kopię zapasową bazy danych należy w XAMPP wejść w terminal i użyć komendy która wykonuje kopie zapasową. Oczywiście ścieżka dostępu będzie się różnić ale w moim przypadku tak ona wygląda

```
mysqldump -u root -p festiwale > C:\Users\dawid\Desktop\backup
\festiwale2_backup.sql
```

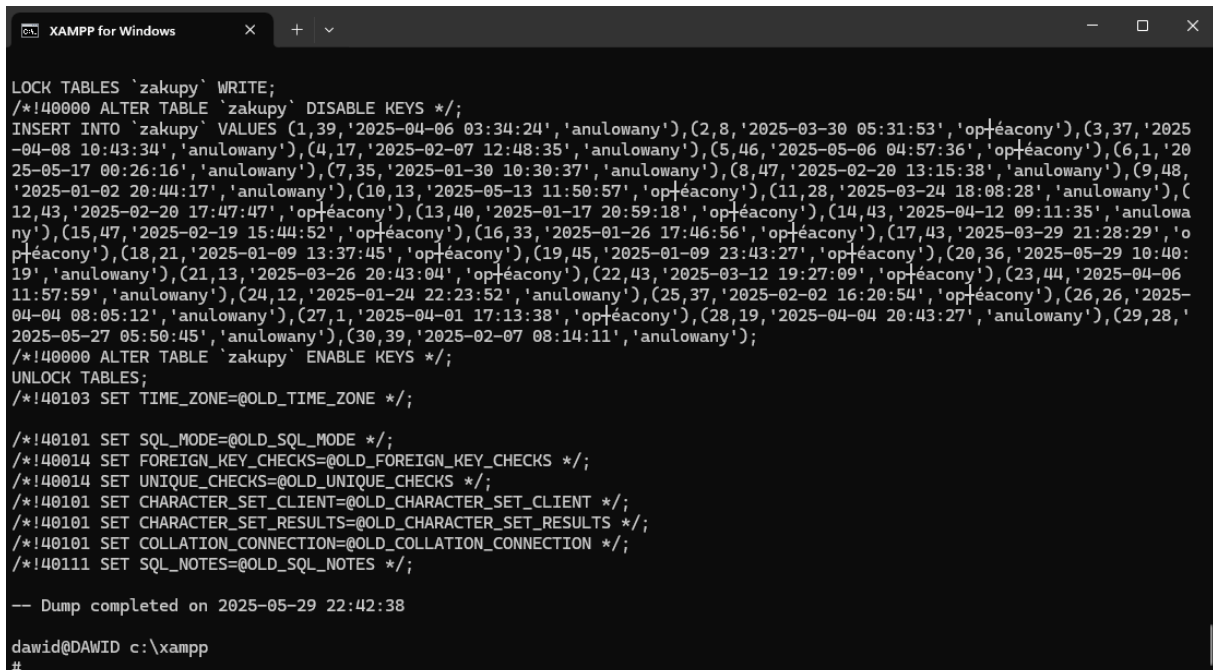


```
XAMPP for Windows
Setting environment for using XAMPP for Windows.
dawid@DAWID c:\xampp
# mysqldump -u root -p festiwale2 > C:\Users\dawid\Desktop\backup\festiwale2_backup.sql
Enter password:
dawid@DAWID c:\xampp
#
```

Rysunek 21 Tworzenie kopii zapasowej

Natomiast aby przywrócić zapisaną wcześniej bazę danych także w terminalu należy użyć polecenia:

```
mysqldump -u root -p festiwale < C:\Users\dawid\Desktop\backup  
\festiwale2_backup.sql
```



```
XAMPP for Windows
LOCK TABLES `zakupy` WRITE;
/*!40000 ALTER TABLE `zakupy` DISABLE KEYS */;
INSERT INTO `zakupy` VALUES (1,39,'2025-04-06 03:34:24','anulowany'),(2,8,'2025-03-30 05:31:53','opłacaony'),(3,37,'2025-04-08 10:43:34','anulowany'),(4,17,'2025-02-07 12:48:35','anulowany'),(5,46,'2025-05-06 04:57:36','opłacaony'),(6,1,'2025-05-17 00:26:16','anulowany'),(7,35,'2025-01-30 10:30:37','anulowany'),(8,47,'2025-02-20 13:15:38','anulowany'),(9,48,'2025-01-02 20:44:17','anulowany'),(10,13,'2025-05-13 11:50:57','opłacaony'),(11,28,'2025-03-24 18:08:28','anulowany'),(12,43,'2025-02-20 17:47:47','opłacaony'),(13,40,'2025-01-17 20:59:18','opłacaony'),(14,43,'2025-04-12 09:11:35','anulowany'),(15,47,'2025-02-19 15:44:52','opłacaony'),(16,33,'2025-01-26 17:46:56','opłacaony'),(17,43,'2025-03-29 21:28:29','opłacaony'),(18,21,'2025-01-09 13:37:45','opłacaony'),(19,45,'2025-01-09 23:43:27','opłacaony'),(20,36,'2025-05-29 10:40:19','anulowany'),(21,13,'2025-03-26 20:43:04','opłacaony'),(22,43,'2025-03-12 19:27:09','opłacaony'),(23,44,'2025-04-06 11:57:59','anulowany'),(24,12,'2025-01-24 22:23:52','anulowany'),(25,37,'2025-02-02 16:20:54','opłacaony'),(26,26,'2025-04-04 08:05:12','anulowany'),(27,1,'2025-04-01 17:13:38','opłacaony'),(28,19,'2025-04-04 20:43:27','anulowany'),(29,28,'2025-05-27 05:50:45','anulowany'),(30,39,'2025-02-07 08:14:11','anulowany');
/*!40000 ALTER TABLE `zakupy` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

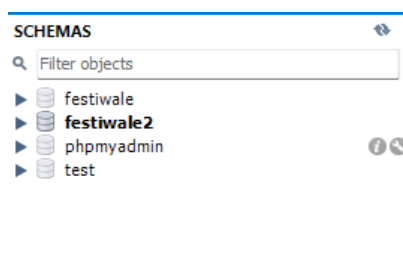
-- Dump completed on 2025-05-29 22:42:38

dawid@DAWID c:\xampp
#
```

Rysunek 22 Przywracanie kopii zapasowej

Warianty testowe i produkcyjne

System ten powinien funkcjonować w dwóch środowiskach gdzie testowym środowiskiem będzie festiwale2 natomiast produkcyjnym festiwale. Środowisko testowe powinno zawierać kopie struktury produkcyjnej w najbardziej aktualnej wersji. Umożliwia ono testowanie zapytań procedur i funkcji bez wpływania na użytkowników końcowych. Każda zmiana powinna być najpierw testowana na środowisku testowym.



Rysunek 23 Środowiska testowe i produkcyjne

5. Wnioski

Baza danych umożliwia przechowywanie wszystkich informacji o festiwalach, koncertach, artystach i użytkownikach. Pozwala na śledzenie sprzedaży biletów z podziałem na typy (VIP, karnet, jednodniowy). W systemie także są wydzielone role (admin, organizator, uczestnik) co pozwala dostosować uprawnienia do funkcji użytkownika. Dzięki temu uczestnik nie ma dostępu do edycji wydarzeń, a organizator nie zarządza użytkownikiem. Zaprojektowana struktura pozwala na przyszłą rozbudowę. Dodawanie nowego typu użytkowników, rozszerzanie informacji o koncertach. Baza zawiera wszystkie najważniejsze aspekty systemu który miałby obsługiwać problem tego typu

6. Spis rysunków i tabel

Tabela 1 Funkcje bazy danych	4
Tabela 2 Encje Bazy Danych	6
Tabela 3 Opis Encji Użytkownicy	8
Tabela 4 Opis Encji Zakupy	8
Tabela 5 Opis Encji Zakupione bilety	9
Tabela 6 Opis Encji Bilety	9
Tabela 7 Opis Encji Festiwale	10
Tabela 8 Opis Encji Koncerty	10
Tabela 9 Opis Encji Sceny	11
Tabela 10 Opis Encji Artysta	11
Tabela 11 Opis Encji Statystyki uczestnictwa	12
 Rysunek 1 Diagram Bazy Danych	 7
Rysunek 2 Zapytanie 1	21
Rysunek 3 Zapytanie 2	22

Rysunek 4 Zapytanie 3	23
Rysunek 5 Zapytanie 4	23
Rysunek 6 Zapytanie 5	24
Rysunek 7 Zapytanie 6	25
Rysunek 8 Zapytanie 7	25
Rysunek 9 Zapytanie 8	26
Rysunek 10 Zapytanie 9	27
Rysunek 11 UPDATE 1	28
Rysunek 12 UPDATE 2	28
Rysunek 13 UPDATE 3	29
Rysunek 14 UPDATE 4	29
Rysunek 15 Procedura 1	31
Rysunek 16 Wywołanie procedury 1	31
Rysunek 17 Procedura 2	32
Rysunek 18 Wywołanie procedury 2	32
Rysunek 19 Procedura 3	33
Rysunek 20 Wywołanie procedury 3	34
Rysunek 21 Tworzenie kopii zapasowej	34
Rysunek 22 Przywracanie kopii zapasowej	35
Rysunek 23 Środowiska testowe i produkcyjne	35