Wydział WIMiIP	Imię i nazwisko Dominik Budzowski	Rok 2	Nr indeksu 400735	Grupa GL01
Temat: **MySQL 5 - CRUD**				
Data wykonania 27.04.2021				OCENA

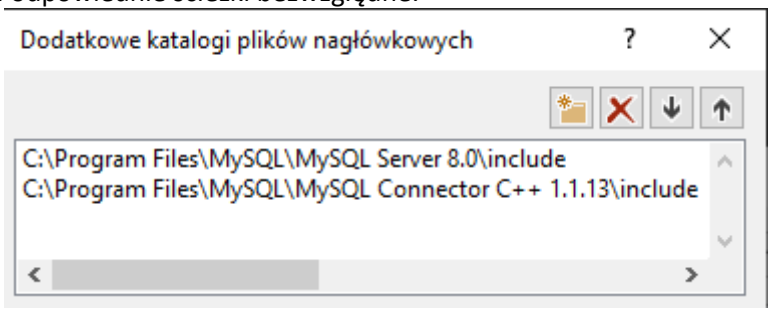
W ćwiczeniu korzystałem z:

- MySQL Server 8.0;
- MySQL Connector C++ 1.1.13;
- Microsoft Visual Studio Community 2019 Wersja 16.8.3.

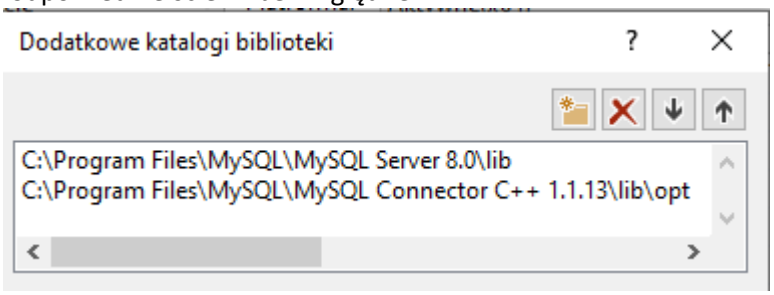
Instalacja

Po zainstalowaniu *MySQL Server 8.0* oraz *MySQL Connector C++ 1.1.13* należało odpowiednio ustawić Visual Studio aby można było korzystać z bibliotek jakie na oferuje MySQL. Omówię w krokach ustawienia projektu.

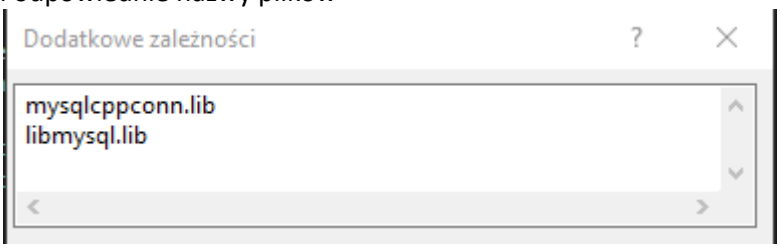
1. Ustawiłem dodatkowe katalogi plików nagłówkowych.
   Ścieżka:
   Projekt -> Właściwości -> C/C++ -> Ogólne -> Dodatkowe katalogi plików nagłówkowych
   I tam dodałem odpowiednie ścieżki bezwzględne:



2. Następnie Ustawiłem dodatkowe katalogi biblioteki.
   Ścieżka:
   Projekt -> Właściwości -> Konsolidator -> Ogólne -> Dodatkowe katalogi biblioteki
   I tam dodałem odpowiednie ścieżki bezwzględne



3. Następnie ustawiłem dodatkowe zależności.
   Ścieżka:
   Projekt -> Właściwości -> Konsolidator -> Dane wejściowe -> Dodatkowe zależności
   I tam dodałem odpowiednie nazwy plikow



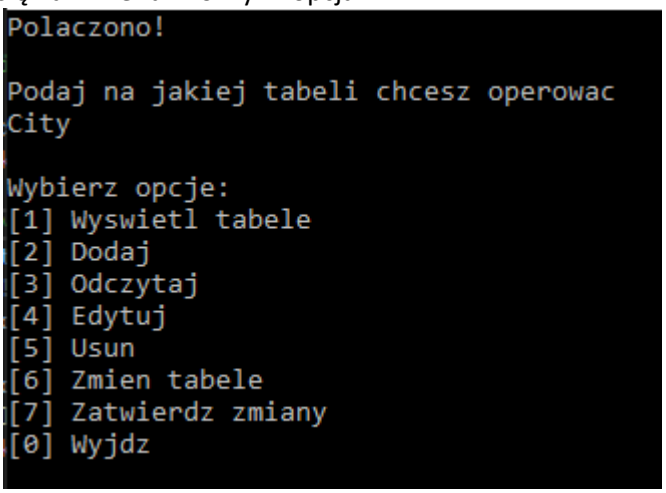
   Następnie należało dać [Zastosuj] zapisując ustawienia.

4. Kolejną rzeczą było przekopiowanie pliku libmysql.dll do katalogu w którym znajduje się projekt.
   Ścieżka bezwzględna w której znajduje się w plik w moim przypadku:
   C:\Program Files\MySQL\MySQL Server 8.0\lib

To wystarczyło w moim przypadku, w celu odpowiedniej konfiguracji korzystałem ze źródła:
https://www.youtube.com/watch?v=yNniOHn9Xe0

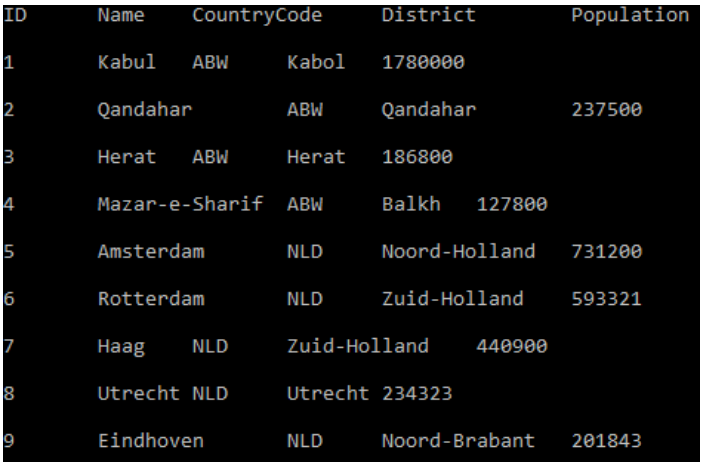
Działanie programu

Na początku dostajemy informacje czy połączono i podajemy dla jakiej tabeli będą wykonywane operacje.
Po podaniu wyświetla się nam menu z różnymi opcjami



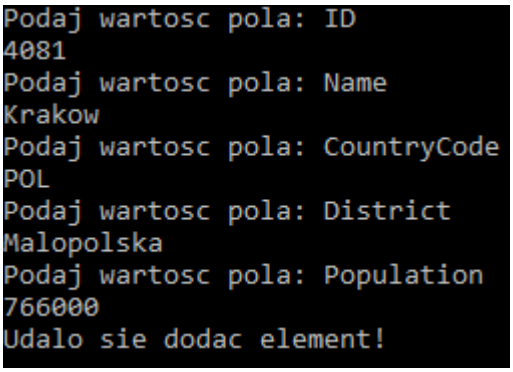
Po wpisaniu odpowiedniej cyfry program wykonuje pewne operacje, a po wykonaniu ich wyświetla ponownie menu

1. Wyświetl tabele – wyświetla tabele, a dokładnie nazwy kolumn i wszystkie jej wartości. Wyświetlanie w standardowej konsoli c++ nie uwzględnia prawidłowo wszystkich znaków np. ś, ć, ó itd.
   Fragment wyniku z konsoli:



ID	Name	CountryCode	District	Population
1	Kabul	ABW	Kabol	1780000
2	Qandahar	ABW	Qandahar	237500
3	Herat	ABW	Herat	186800
4	Mazar-e-Sharif	ABW	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843

2. Dodaj – dodaje element to tabeli, musimy podać wartości każdej kolumny.



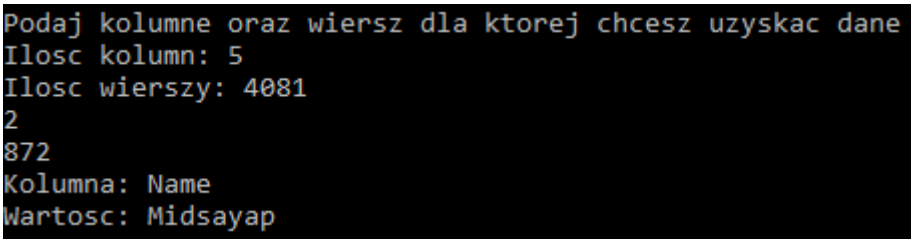
```
Podaj wartosc pola: ID
4081
Podaj wartosc pola: Name
Krakow
Podaj wartosc pola: CountryCode
POL
Podaj wartosc pola: District
Malopolska
Podaj wartosc pola: Population
766000
Udalo sie dodac element!
```

   Po wypisaniu tabeli doda nam się wiersz.

```
4081    Krakow  POL     Malopolska      766000
```

   W tym przypadku jeśli podamy istniejące już ID lub nie istniejący ContryCode nie uda się dodać elementu. Wynika to z własności ustawionych kluczy w tabeli.

3. Odczytaj – odczytuje wartość pola we wskazanym miejscu w tabeli.



```
Podaj kolumne oraz wiersz dla ktorej chcesz uzyskac dane
Ilosc kolumn: 5
Ilosc wierszy: 4081
2
872
Kolumna: Name
Wartosc: Midsayap
```

4. Edytuj – edytuje wartość pola we wskazanym miejscu tabeli na podaną wartość.

```
Podaj kolumne oraz wiersz dla ktorej chcesz uzyskac dane i je zmienic:
Ilosc kolumn: 5
Ilosc wierszy: 4080
3
302
Kolumna: CountryCode
Wartosc: BRA
Podaj wartosc na jaka chcesz podmienic:
POL
Udalo sie edytowac element!
```

Znowu mogą wystąpić problemy gdy podamy niewłaściwą wartość w przypadku kluczy.

5. Usun – Usuwa podany wiersz tabeli.

```
Podaj wiersz ktory chcesz usunac:
Ilosc wierszy: 4080
1
ID      Name    CountryCode     District        Population

1       Kabul   ABW     Kabol   1780000

Udalo sie usunac element!
```

Teraz początek tabeli będzie wyglądać tak:

```
ID      Name    CountryCode     District        Population

2       Qandahar        ABW     Qandahar        237500

3       Herat   ABW     Herat   186800

4       Mazar-e-Sharif  ABW     Balkh   127800

5       Amsterdam       NLD     Noord-Holland   731200
```

6. Zmien tabele – prosi o podanie nazwy tabeli na której będziemy teraz pracować.

```
Podaj nowa nazwe tabeli na jakiej chcesz operowac
Country
```

I teraz po wybraniu opcji 1 wyświetli się:

```
Code    Name    Continent       Region  SurfaceArea     IndepYear       Population      LifeExpectancy  GNP     GNPOld LocalName
                                                                                        GovernmentForm  HeadOfState     Capital Code2

ABW     Aruba   North America   Caribbean       193.00  2021    103000  78.4    828.00  828.00  Aruba   Nonmetropolitan Territory of The Netherlands
                                                                                        Beatrix 129     AW

AFG     Afghanistan     Asia    Southern and Central Asia       652090.00       1919    22720000        45.9    5976.00
                                                                                        NULL    Afganistan/Afqanestan   Islamic Emirate Mohammad Omar   1       A
F

AGO     Angola  Africa  Central Africa  1246700.00      1975    12878000        38.3    6648.00 7984.00 Angola  Republic
                                                                                        Jos├ę Eduardo dos Santos 56     AO

AIA     Anguilla        North America   Caribbean       96.00   NULL    8000    76.1    63.20   NULL    Anguilla        Dependent Territory of the UK
                                                                                        Elisabeth II    62      AI

ALB     Albania Europe  Southern Europe 28748.00        1912    3401200 71.6    3205.00 2500.00 Shqip├źria       Republic
                                                                                        Rexhep Mejdani  34      AL
```

Z powodu dużej ilości kolumn i długich wierszy nie układa się pięknie w konsoli.

7. Zatwierdz zmiany – zatwierdza i zapisuje zmiany poleceniem COMMIT co spowoduje, że gdy wyjdziemy z programu i wrócimy na nowo zmiany tam będą. Po zapisie zmian rozpoczyna nową instrukcje START TRANSACTION.

```
Zatwierdzono zmiany
```

0. Wyjdz – Kończy działanie programu.

Kilka zdań na temat działania programu:

- Program nie umożliwia wpisywania do pól wyrazów oddzielonych znakiem białym,
- Program kontroluje błędy i wyświetla informacje czy coś się udało czy nie,
- Program nie rozwiązuje problemu podawania nieprawidłowego znaku (np. w menu podamy wartość „kjsladhja”).

Podsumowanie

Po odpowiedniej konfiguracji środowiska języka programowania możemy korzystać z baz danych. W celu użycia poleceń edycji (np. INSERT, UPDATE, DELETE) przesyła się odpowiedni ciąg znaków do serwera. Dzięki połączeniu bazy danych z językiem programowania możemy w łatwy sposób korzystać z danych, wykonywać na nich operacja, oraz jeśli jest taka konieczność to usuwać, aktualizować bądź zapisywać nowe dane. Połączenie z bazą danych, można traktować jako nowy etap gdzie już nie jesteśmy zmuszeni operować z danymi pobranymi z notatnika.

Listingi kodu:

```
#include <iostream>
#include <mysql.h>    //Biblioteka umożliwiająca nam kożsytanie z baz
danych

using namespace std;

//Funkcja wyświetlająca tabele
void Wyswietl( MYSQL * mysql, string tabela)
{
    MYSQL_ROW row;              //Zmienne przechowywująca wiersz tabeli
    MYSQL_RES* res;             //Zmienna przechowywująca resultat
polecenia
    MYSQL_FIELD* field;         //Zmienna przechowywująca nazwy kolumn

    //Odczytywanie wszystkich danych z tabeli
    string tabel = "SELECT * FROM " + tabela;
    const char* t = tabel.c_str();
    int qstate = mysql_query(mysql, t);


    //Udalo sie wykonac polecenie
    if (!qstate)
    {

        res = mysql_store_result(mysql);    //Zapisanie wyniku do zmiennej
        int count = mysql_num_fields(res);  //Zapisanie ilosci kolumn
```

```
        //Wypisanie nazw kolumn
        while (field = mysql_fetch_field(res))
        {
            cout << field->name << "\t";
        }
        cout << endl << endl;

        //Wypisanie wszystkich wierszy, w przypadku braku wartości pola
wypisanie NULL
        while ((row = mysql_fetch_row(res)) != NULL)
        {
            for (int i = 0; i < count; i++)
            {
                if (row[i] == NULL)
                    cout << "NULL\t";
                else
                    cout << row[i] << "\t";
            }
            cout << endl << endl;
        }
    }
    else
    {
        cout << "Nie udalo sie wyswietlic tabeli!" << endl;
    }
    cout << endl;
}

//Funkcja dodająca nowy wiersz do tabeli
void Dodaj(MYSQL* mysql, string tabela)
{
    MYSQL_ROW row;                  //Zmienne przechowywująca wiersz tabeli
    MYSQL_RES* res;                 //Zmienna przechowywująca resultat
polecenia
    MYSQL_FIELD* field;             //Zmienna przechowywująca nazwy kolumn

    //Odczytywanie wszystkich danych z tabeli
    string tabel = "SELECT * FROM " + tabela;
    const char* t = tabel.c_str();
    int qstate = mysql_query(mysql, t);

    //Udalo sie wykonac polecenie
    if (!qstate)
    {

        res = mysql_store_result(mysql);        //Zapisanie wyniku do
zmiennej
        int count = mysql_num_fields(res);      //Zapisanie ilosci kolumn

        //Allokacja pamieci do zapisania nazw kolumn
        string* columns = new string[count];

        //zmienne przechowywująca zapytanie
        string query = "INSERT INTO `" + tabela + "` VALUES( ";

        //Podanie wartości pól
        for (int i = 0; i < count; i++)
        {
            field = mysql_fetch_field(res);
            cout << "Podaj wartosc pola: " << field->name << endl;
            cin >> columns[i];
```

```
            if (i < (count -  1))
                query = query + "'" + columns[i] + "', ";
            else
                query = query + "'" + columns[i] + "'";
        }
        query = query + " )";
        const char* q = query.c_str();
        qstate = mysql_query(mysql, q);         //Wysłanie zbudowanego
polecenia na serwer

        if (!qstate)
        {
            cout << "Udalo sie dodac element!" << endl;
        }
        else
        {
            cout << "Nie udalo sie dodac elementu!" << endl;
        }

        delete[] columns;
    }
    else
    {
        cout << "Nie udalo sie znalezc tabeli" << endl;
    }

    cout << endl;
}

//Funkcja zapisuje wszystkie dane z tabeli (umiejętność odczytywania
danych) ale wypisuje tylko tą o którą poprosimy
void Odczytaj(MYSQL* mysql, string tabela)
{
    MYSQL_ROW row;
    MYSQL_RES* res;
    MYSQL_FIELD* field;

    string tabel = "SELECT * FROM " + tabela;
    const char* t = tabel.c_str();
    int qstate = mysql_query(mysql, t);

    if (!qstate)
    {
        res = mysql_store_result(mysql);
        int countf = mysql_num_fields(res);     //Ilość kolumn
        int countr = mysql_num_rows(res);       //Ilość wierszy

        //Allokacja pamieći w celu zapisania nazw kolumn i wszystkich
wartości wierszy
        string* columns = new string[countf];
        string** spot = new string * [countf];
        for (int i = 0; i < countf; i++)
            spot[i] = new string[countr];

        //Zapisanie wszystkich wartości do tablicy dwuwymiarowej gdzie
pierwszy argument to kolumna, a drugi to wiersz
        for(int i=0; i<countr; i++)
        {
            row = mysql_fetch_row(res);

            for (int j = 0; j < countf; j++)
```

```
            {
                if (row[j] == NULL)
                    spot[j][i] = "NULL";
                else
                    spot[j][i] = row[j];
            }
        }

        //Zapisannie nazw kolumn
        for (int i = 0; i < countf; i++)
        {
            field = mysql_fetch_field(res);
            columns[i] = field->name;
        }

        cout << "Podaj kolumne oraz wiersz dla ktorej chcesz uzyskac dane"
<< endl;
        cout << "Ilosc kolumn: " << countf << endl;
        cout << "Ilosc wierszy: " << countr << endl;
        int w, k;
        cin >> k >> w;

        if (k > countf || k <= 0 || w > countr || w <= 0)
        {
            cout << "Podano bledne dane" << endl;
            return;
        }

        cout << "Kolumna: " << columns[k-1] << endl << "Wartosc: " <<
spot[k-1][w-1] << endl;

        for (int i = 0; i < countf; i++)
            delete[] spot[i];
        delete[] spot;
        delete[] columns;
    }
    else
    {
        cout << "Nie udalo sie znalezc tabeli" << endl;
    }

    cout << endl;
}

//Edycja wartości w tabeli
void Edytuj(MYSQL* mysql, string tabela)
{
    MYSQL_ROW row;
    MYSQL_RES* res;
    MYSQL_FIELD* field;

    string tabel = "SELECT * FROM " + tabela;
    const char* t = tabel.c_str();
    int qstate = mysql_query(mysql, t);

    if (!qstate)
    {
        res = mysql_store_result(mysql);
        int countf = mysql_num_fields(res);
        int countr = mysql_num_rows(res);
```

```
        string* columns = new string[countf];
        string** spot = new string * [countf];
        for (int i = 0; i < countf; i++)
            spot[i] = new string[countr];

        for (int i = 0; i < countr; i++)
        {
            row = mysql_fetch_row(res);

            for (int j = 0; j < countf; j++)
            {
                if (row[j] == NULL)
                    spot[j][i] = "NULL";
                else
                    spot[j][i] = row[j];
            }
        }

        for (int i = 0; i < countf; i++)
        {
            field = mysql_fetch_field(res);
            columns[i] = field->name;
        }

        cout << "Podaj kolumne oraz wiersz dla ktorej chcesz uzyskac dane i
je zmienic: " << endl;
        cout << "Ilosc kolumn: " << countf << endl;
        cout << "Ilosc wierszy: " << countr << endl;
        int w, k;
        cin >> k >> w;

        if (k > countf || k <= 0 || w > countr || w <= 0)
        {
            cout << "Podano bledne dane" << endl;
            return;
        }

        cout << "Kolumna: " << columns[k - 1] << endl << "Wartosc: " <<
spot[k - 1][w - 1] << endl;

        cout << "Podaj wartosc na jaka chcesz podmienic:" << endl;

        //Zmienna przechowywująca zapytanie UPDATE
        string wartosc;
        cin >> wartosc;
        wartosc = "UPDATE `" + tabela + "` SET " + columns[k - 1] + "='" +
wartosc + "' WHERE ";

        //Budowanie zapytania jako stringa aby napewno trafic w odpowiedni
rekord, wartości NULL są będą nieuwzględniane w klauzuli WHERE
        for (int i = 0; i < countf; i++)
        {
            if (i == k - 1)
                continue;
            if (spot[i][w - 1] == "NULL")
                continue;
            wartosc = wartosc + columns[i] + "='" + spot[i][w-1] + "' AND
";
        }
        wartosc = wartosc + columns[k - 1] + "=" + columns[k - 1];
```

```
        //Wysłanie zapytania
        const char* q = wartosc.c_str();
        qstate = mysql_query(mysql, q);

        if (!qstate)
        {
            cout << "Udalo sie edytowac element!" << endl;
        }
        else
        {
            cout << "Nie udalo sie edytowac elementu!" << endl;
        }

        for (int i = 0; i < countf; i++)
            delete[] spot[i];
        delete[] spot;
        delete[] columns;
    }
    else
    {
        cout << "Nie udalo sie znalezc tabeli" << endl;
    }
    cout << endl;
}

//Funkcja usuwająca wiersz tabeli
void Usun(MYSQL* mysql, string tabela)
{
    MYSQL_ROW row;
    MYSQL_RES* res;
    MYSQL_FIELD* field;

    string tabel = "SELECT * FROM " + tabela;
    const char* t = tabel.c_str();
    int qstate = mysql_query(mysql, t);

    if (!qstate)
    {
        res = mysql_store_result(mysql);
        int countf = mysql_num_fields(res);
        int countr = mysql_num_rows(res);

        string* columns = new string[countf];
        string** spot = new string * [countf];
        for (int i = 0; i < countf; i++)
            spot[i] = new string[countr];

        for (int i = 0; i < countr; i++)
        {
            row = mysql_fetch_row(res);

            for (int j = 0; j < countf; j++)
            {
                if (row[j] == NULL)
                    spot[j][i] = "NULL";
                else
                    spot[j][i] = row[j];
            }
        }

        for (int i = 0; i < countf; i++)
```

```
        {
            field = mysql_fetch_field(res);
            columns[i] = field->name;
        }

        cout << "Podaj wiersz ktory chcesz usunac: " << endl;
        cout << "Ilosc wierszy: " << countr << endl;
        int w;
        cin >> w;

        if ( w > countr || w <= 0)
        {
            cout << "Podano bledne dane" << endl;
            return;
        }

        for(int i=0; i<countf; i++)
        {
            cout << columns[i] << "\t";
        }
        cout << endl << endl;

        for (int i = 0; i < countf; i++)
        {
            cout << spot[i][w-1] << "\t";
        }
        cout << endl << endl;

        string wartosc;
        wartosc = "DELETE FROM `" + tabela + "` WHERE ";
        for (int i = 0; i < countf; i++)
        {
            if (spot[i][w - 1] == "NULL")
                continue;
            wartosc = wartosc + columns[i] + "='" + spot[i][w - 1] + "' AND
";
        }
        wartosc = wartosc + columns[0] + "=" + columns[0];

        const char* q = wartosc.c_str();
        qstate = mysql_query(mysql, q);

        if (!qstate)
        {
            cout << "Udalo sie usunac element!" << endl;
        }
        else
        {
            cout << "Nie udalo sie usunac elementu!" << endl;
        }

        for (int i = 0; i < countf; i++)
            delete[] spot[i];
        delete[] spot;
        delete[] columns;
    }
    else
    {
        cout << "Nie udalo sie znalezc tabeli" << endl;
    }
    cout << endl;
```

```
}

//Funkcja główna programu
int main(int argc, const char* argv[])
{
    //Tworzenie zmiennej odpowiedzialnej za połączenie z bazą danych
    MYSQL* mysql;

    //Łączenie z programem
    mysql = mysql_init(0);
    mysql = mysql_real_connect(mysql, "127.0.0.1", "root", "1234", "World",
0, NULL, 0);

    //W przypadku gdy się udało połączyć
    if (mysql)
    {
        //Zaczecie transakcji
        int qstate = mysql_query(mysql, "START TRANSACTION");
        int x = 1;

        cout << "Polaczono!" << endl << endl;

        //Wybor tabeli z naszej bazy danych (w tym przypadku z bazy danych
world)
        cout << "Podaj na jakiej tabeli chcesz operowac" << endl;
        string tabel;
        cin >> tabel;
        cout << endl;

        //Menu Główne
        while (x)
        {
            cout << "Wybierz opcje:" << endl;
            cout << "[1] Wyswietl tabele" << endl;
            cout << "[2] Dodaj" << endl;
            cout << "[3] Odczytaj" << endl;
            cout << "[4] Edytuj" << endl;
            cout << "[5] Usun" << endl;
            cout << "[6] Zmien tabele" << endl;
            cout << "[7] Zatwierdz zmiany" << endl;
            cout << "[0] Wyjdz" << endl;

            cin >> x;

            switch (x)
            {
            case 1:
                Wyswietl(mysql, tabel);
                break;
            case 2:
                Dodaj(mysql, tabel);
                break;
            case 3:
                Odczytaj(mysql, tabel);
                break;
            case 4:
                Edytuj(mysql, tabel);
                break;
            case 5:
                Usun(mysql, tabel);
                break;
```

```
            case 6:
                //Zmiana nazwy tabeli
                cout << "Podaj nowa nazwe tabeli na jakiej chcesz operowac"
<< endl;
                cin >> tabel;
                break;
            case 7:
                //COMMIT aktualnych zmiann i rozpoczęcie nowej transakcji
                qstate = mysql_query(mysql, "COMMIT");
                if (!qstate)
                    cout << "Zatwierdzono zmiany" << endl << endl;
                else
                    cout << "Nie zatwierdzono zmiany" << endl << endl;
                mysql_query(mysql, "START TRANSACTION");
                break;
            case 0:
                x = 0;
                break;
            }
        }
    }
    else
        cout << "Nie polaczono" << endl << mysql_errno(mysql) <<
mysql_error(mysql) << endl;;



    mysql_close(mysql); // zamknij połączenie

    return 0;
}
```