Dokumentacja projektu aplikacji dostępowej dziennika elektronicznego

Konrad Białek 248993
Wojciech Chojnowski 249477
Prowadzący: mgr inż. Norbert Kozłowski
Zajęcia projektowe: Środa 9:15-11:00

13.06.2021

1 Wstęp

Aplikacja dostępowa została napisana w języku C# w postaci okienkowej z wykorzystaniem technologii WPF.

2 Autoryzacja i autentykacja dostępu do bazy z aplikacji i wykorzystanie frameworków ORM

Listing 1: Uprawnienia użytkownika 'dziennikszkolny'@'localhost'

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, PROCESS, FILE
    , ALTER, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES,
    REPLICATION SLAVE, REPLICATION CLIENT, SHOW VIEW, EXECUTE
ON *.* TO 'dziennikszkolny'@'localhost'
REQUIRE NONE WITH
MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0
MAX_UPDATES_PER_HOUR 0
MAX_USER_CONNECTIONS 0;
```

Listing 2: BazaDanych.cs fragment

```
using MySql.Data.MySqlClient;
public static MySqlConnection Connection { get; set; } = null;

```

```
4  internal static void SetPassword(string password)
5  {
6      byte[] pass = Encoding.UTF8.GetBytes(password);
7      MD5 md5Provider = new MD5CryptoServiceProvider();
8      byte[] md5Hash = md5Provider.ComputeHash(pass);
9      password = BitConverter.ToString(md5Hash).Replace("-", string
           .Empty).ToLower();
10     var ConnString = "server=localhost;uid=dziennikszkolny;pwd="
           + password + ";database=dziennikszkolny;";
11     Connection = new MySqlConnection(ConnString);
12     try
13     {
14         Connection.Open();
15     }
16     catch (MySqlException ex)
17     {
18         MessageBox.Show(ex.Message);
19     }
20 }
```

Listing 3: Logowanie.cs fragment

```
1  public void Zaloguj(object sender, RoutedEventArgs e)
2  {
3      BazaDanych.SetPassword(hasloDB.Password);
4      Login1 = login.Text;
5      byte[] pass = Encoding.UTF8.GetBytes(haslo.Password);
6      MD5 md5Provider = new MD5CryptoServiceProvider();
7      byte[] md5Hash = md5Provider.ComputeHash(pass);
8      string strPassword = BitConverter.ToString(md5Hash).Replace("
           -", string.Empty).ToLower();
9      try
10     {
11         var qwe = ((MainWindow)Application.Current.MainWindow).
               UserInBase(Login1, strPassword);
12         if (qwe != null)
13             ((MainWindow)Application.Current.MainWindow).Zaloguj(
                   qwe);
14     } catch (Exception ex)
15     {
16         MessageBox.Show(ex.Message);
17     }
18 }
```

Użytkownik bazodanowy potrzebuje uprawnień do odczytu, dodawania i aktualizacji danych w bazie. Ponadto musi mieć dostęp do wykonywania archiwum bazy i przywracania danych z niego. Użytkownicy z poziomu aplikacji są podzieleni na 3 grupy: administratora o pełnych uprawnieniach użytkownika bazodanowego, jeżeli te są dostępne z poziomu aplikacji; nauczycieli mających dostęp do aktualizacji części tabel, odczytu z nich i bez dostępu do funkcjonalności archiwum oraz uczniów i rodziców mających dostęp do odczytu tabel w mniejszym zakresie niż nauczyciele.

Wykorzystanie ORM jest realizowane przez utworenie połaczenia poprzez MySQLC-

lient zawierającego klasę MySqlConnection.

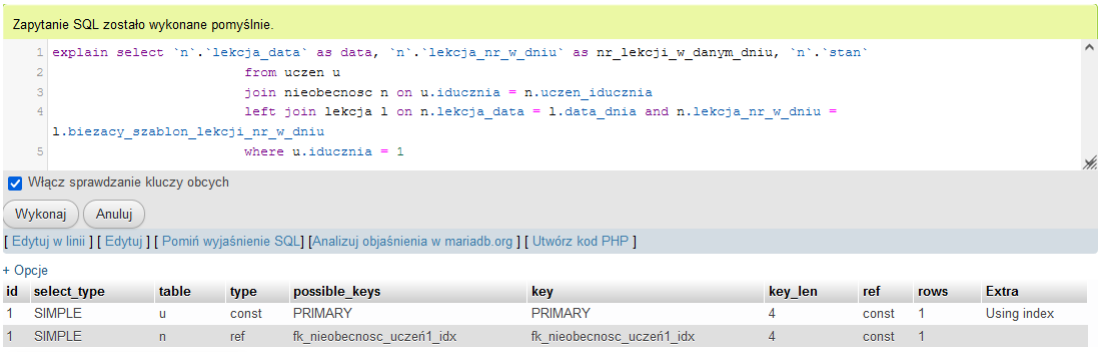
3 Sposoby zabezpieczenia się przed atakami

Listing 4: DodawanieOceny.cs fragment

```
1  private void Button_Click(object sender, RoutedEventArgs e)
2  {
3      var user = ((MainWindow)Application.Current.MainWindow).
           LoggedUser;
4      BazaDanych.Execute("INSERT INTO ocena (data, wartosc,
           nauczyciel_id, przedmiot_nazwa, waga, opis,
           uczen_idu cznia) VALUES (@data, @wartosc, @nauczyciel_id,
           @przedmiot_nazwa, @waga, @opis, @uczen_idu cznia)",
5           new MySqlParameter("data", DateTime.Today),
6           new MySqlParameter("wartosc", float.Parse(Wartosc.Text.
                Replace(',', '.'), CultureInfo.InvariantCulture)),
7           new MySqlParameter("nauczyciel_id", user.ID),
8           new MySqlParameter("przedmiot_nazwa", Przedmiot.
                SelectedValue),
9           new MySqlParameter("waga", int.Parse(Waga.Text)),
10          new MySqlParameter("opis", Opis.Text),
11          new MySqlParameter("uczen_idu cznia", Uczen.
                SelectionBoxItem)
12          );
13     MessageBox.Show("Dodano ocene.");
14     ((MainWindow)Application.Current.MainWindow).DataContext =
           new Oceny();
15 }
```

Do zabezpieczenia bazy przed atakami wykorzystano zapytania sparametryzowane.

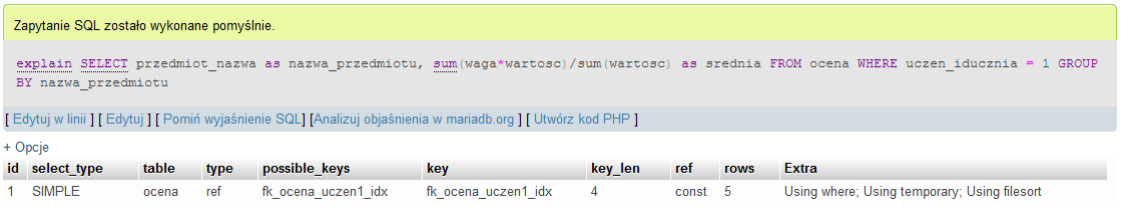
4 Analiza złożoności wykonywanych zapytań



Rysunek 1: Wykorzystanie polecenia EXPLAIN na nieobecnościach ucznia.

Zarówno zapytanie dotyczące tabeli uczen (u) jak i tabeli nauczyciel (n) nie zawierają podzapytań. Typ ref oznacza możliwość krótszego czasu przeszukiwania tablicy

w związku z tym, że dane są posortowane. MySQL wykorzystał klucze wskazane przed procesem optymalizacji. Długość klucza określa ilość bajtów potrzebnych do wykonania zapytania.Liczba rzędów określa średnią ilość rekordów, które będzie należało przejrzeć aby otrzymać potrzebne dane. Using index w ostatniej kolumnie oznacza, że podczas wykonywania zapytania zostanie użyty indeks.


Zapytanie SQL zostało wykonane pomyślnie.

```
explain SELECT przedmiot_nazwa as nazwa_przedmiotu, sum(waga*wartosc)/sum(wartosc) as srednia FROM ocena WHERE uczen_iducznia = 1 GROUP BY nazwa_przedmiotu
```

[Edytuj w linii] [Edytuj] [Pomiń wyjaśnienie SQL] [Analizuj objaśnienia w mariadb.org] [Utwórz kod PHP]

+ Opcje

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	ocena	ref	fk_ocena_uczen1_idx	fk_ocena_uczen1_idx	4	const	5	Using where; Using temporary; Using filesort

Rysunek 2: Wykorzystanie polecenia EXPLAIN na średniej ocen ucznia.

Poza elementami omówionymi powyżej Using where oznacza konieczność przefiltrowania otrzymanych wyników, Using temporary oznacza wykorzystanie tymczasowej tablicy do sortowania„a Using filesort oznacza wykorzystanie sortowania dopiero po uzuskaniu danych z bazy.

5 Skalowanie i replikacja bazy danych

Skalowanie bazy będzie się różnić w zależności od powodu, dla którego rozszerzenie będzie potrzebne. Z racji tego, że baza zawiera dane pracowników, uczniów i rodziców w szkole powinna znajdować się na jej terenie. Jeżeli baza miałaby zostać rozszerzona na inną szkołę konieczny będzie pdział bazy według tenanta - placówki i bazy będą znajdować się na terenie odpowiedniej placówki. W przypadku, gdy baza będzie zbyt mała dla danej placówki konieczne będzie podzielenie jej według domeny i osobno trzymać największe tabele. Oczywiście szczegóły tego podziały muszą zostać opracowane indywidualnie dla danej placówki w oparciu o aktualne wymogi placówki i przewidywane powiększanie bazy. Replikacja bazy danych będzie przeprowadzana w formacie mixed-format logging.

6 Skrypty wspomagające “disaster recovery”

Listing 5: Archiwum.cs fragment

```
1  private void Archiwizuj(object sender, RoutedEventArgs e)
2  {
3      var dialog = new SaveFileDialog();
4      dialog.Filter = "SQL|*.sql";
5      dialog.ShowDialog();
6      if (dialog.FileName != null)
7      {
8          using (MySqlCommand Cmd = new MySqlCommand())
9          {
10             using (MySqlBackup Mb = new MySqlBackup(Cmd))
```

```
11                {
12                    try
13                    {
14                        Cmd.Connection = BazaDanych.Connection;
15                        Mb.ExportToFile(dialog.FileName);
16                    }
17                    catch (MySqlException ex)
18                    {
19                        MessageBox.Show(ex.Message);
20                    }
21                    catch (Exception ex)
22                    {
23                        MessageBox.Show(ex.Message);
24                    }
25                }
26            }
27        }
28    }
```

Zabezpieczenie przed utratą danych jest realizowane przez okresowe (umownie co rok w okresie wakacji) wykonywanie backupu bazy danych i przywrócenie jej w przypadku awarii. Dlatego dane wprowadzane do bazy powinny też być zapisywane na innych nośnikach np. dziennikach papierowych.

7 Strategia aktualizacji bazy danych podczas aktualizacji aplikacji

W związku z dużą złożonością bazy danych nie przewiduje się dodawania do niej nowych tabel. Aplikacja dostępowa może zostać zaktualizowana w związku z tym, że nie wszystkie założone funkcjonalności zostały zaimplementowane, ale w związku z tym, że aplikacja pełni funkcję klienta bazy dane prezentowane przez aplikację będą aktualne z dokładnością do odświeżenia okna aplikacji.