

XIV Seminarium
ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICIE' 2004
Oddział Gdański PTETiS

**SYSTEM EWIDENCJI STUDENTÓW WSPOMAGAJĄCY PRACĘ
DZIEKANATU BAZUJĄCY NA ŚRODOWISKU DELPHI I SQL**

Marcin GŁĄB¹, Mirosław WŁAS²

1. Politechnika Gdańska, Katedra Energoelektroniki i Maszyn Elektrycznych
tel. (058) 347-23-37 e-mail: marglab@ely.pg.gda.pl
2. Politechnika Gdańska, Katedra Energoelektroniki i Maszyn Elektrycznych
tel. (058) 347-23-37 e-mail: mwlas@ely.pg.gda.pl

Niniejszy artykuł przedstawia projekt systemu bazy danych na przykładzie aplikacji *Student 2003'*. Przedstawiono w nim m.in.: charakterystykę tworzenia oprogramowania w Delphi, funkcjonowanie aplikacji na serwerze Interbase, tworzenie bazy danych w języku SQL (tworzenie pliku bazy danych, tabel), tworzenie zapytań, procedur składowanych, obsługę procedur składowanych w Delphi.

1. WSTĘP

Wraz ze wzrostem zaawansowania technik informatycznych, wdrażaniem nowych algorytmów rozwiązań oraz ciągłym rozwojem oprogramowania, obserwujemy coraz to bogatsze systemy baz danych operujące na nowoczesnym oprogramowaniu.

Aplikacja *Student 2003'* jest kompletnym systemem wspomagającym pracę dziekanatu. Jej możliwości obejmują: pracę w sieci, pełen zakres edycji danych, pomoc oraz przewodniki o programie. Bogata gama opcji ustawień programu pozwala użytkownikowi w dowolny sposób go skonfigurować. Aplikację *Student 2003'* stworzono głównie do celów funkcjonowania w dziekanacie Wydziału Elektrycznego Politechniki Częstochowskiej. Zakres zastosowania tego oprogramowania nie ogranicza się tylko wyłącznie do funkcjonowania w szkolnictwie wyższym. Słowem aplikacja *Student 2003'* „jest elastycznym środowiskiem pracy”, które można wykorzystywać na wiele sposobów.

2. SYSTEM EWIDENCJI *STUDENT 2003'*

Program *Student 2003'* został napisany za pomocą pakietu Delphi. W programie zaproponowano wykorzystanie sterownika Interbase, jako łącznika dwóch struktur programowania: Delphi i SQL. Okno główne aplikacji (rys. 1) zawiera funkcjonalne Menu główne oraz szybko dostępne przyciski odpowiadające za wybór formularza. Menu główne zawiera pozycje: *Wybór Okna, Serwer, Drukuj, Pomoc*. Menu *Serwer* oferuje możliwość połączenia się z bazą danych, konfiguracji ustawień serwera oraz ustawień

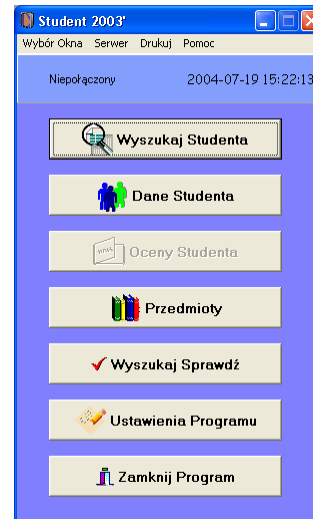
transakcji . Menu *Drukuj* pozwala wydrukować odpowiednio wybrane formularze: *Lista Studentów*, *Karta Egzaminacyjna*, *Zaświadczenie WKU*, *Lista Stypendystów*. Menu *Pomoc* zawiera: plik pomocy Student2003.hlp, przewodnik wyświetlający obsługę programu za pomocą sekwencji AVI, oraz formularz: *O programie*.

Wybierając okno *Wyszukaj Studenta*, odnajdujemy rekord z interesującymi nas danymi. Następnie można dokonać edycji rekordu w oknach: *Dane Studenta* oraz *Oceny Studenta*, odpowiednio wybierając przycisk *Edycja danych*. Formularz: *Przedmioty* posiada zbiór rekordów z wpisanymi przez użytkownika danymi o przedmiotach.

Formularz: *Wyszukaj Sprawdź* pozwala zweryfikować dane umieszczone w rekordach.

Okno *Ustawienia Programu* oferuje dużą gamę opcji związanych z wyglądem i kolorystyką okien oraz formularzy danych. Posiada również możliwość edycji menu szybkiego wyboru, opcja *Ustawienia lokalne*.

Szczegółowy opis programu zamieszczono w [1] i [2].

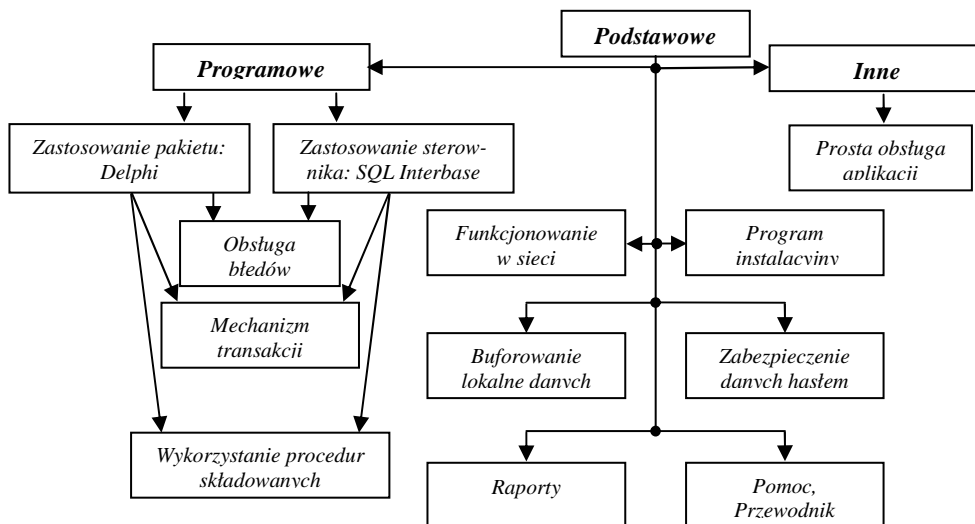


Rys. 1. Główne okno programu

3. PROJEKT APLIKACJI STUDENT 2003'

3.1 Założenia projektowe

Bardzo ważnym elementem są właściwie dobrane założenia projektowe. Na rysunku 2 przedstawiono założenia projektowanej aplikacji. W niniejszym projekcie podstawowe założenia podzielono na: programowe oraz inne. Elementy grup można dowolnie konfigurować, pamiętając o końcowej funkcjonalności programu.



Rys. 2. Założenia projektowe aplikacji Student 2003'

3.2 Tworzenie pliku bazy danych

W aplikacji opartej na sterowniku Interbase, plik bazy danych posiada rozszerzenie *.GDB. Plik ten zawiera wszystkie obiekty bazy, takie jak: tabele, indeksy, procedury składowane, wyzwalacze, domeny.

Aby stworzyć plik *.GDB w języku SQL, należy skorzystać z polecenia:

```
CREATE DATABASE 'C:\Student 2003'\Baza\BAZA.GDB' DEFAULT
CHARACTER SET DOS852
USER 'SYSDBA' PASSWORD 'masterkey' ;
```

Powyższy kod programu tworzy plik *BAZA.GDB*, oraz nadaje uprawnienia użytkownikowi *SYSDBA*, hasło *masterkey*. Domyślnie ustawiono język polski.

Bazę danych utworzono w programie *IBISQL(Interbase Interactive SQL)*, który jest dołączony wraz ze sterownikiem Interbase.

3.3 Konstrukcja tabel w bazie danych

Tabelę w bazie danych można utworzyć korzystając z polecenia:

```
CREATE TABLE tablename ( colname1 characteristics[, colname2
characteristics, ...][, tableconstraint ...])
```

gdzie:

colname1 – nazwa pierwszej kolumny w tabeli,

characteristics – odpowiada m.in. za typ pola, język, lub też ustalenie ograniczeń dla tego pola.

W pliku bazy danych utworzono cztery tabele, dwie z nich są powiązane ze sobą referencyjnie. Przykładowy kod tabeli stworzonej w języku SQL przedstawiono poniżej:

```
/* Extract Table STUDENTDANE */
/* Table: STUDENTDANE, Owner: SYSDBA */
CREATE TABLE STUDENTDANE (PESEL VARCHAR(11) CHARACTER SET DOS852 NOT NULL,
NR_ALBUMU INTEGER NOT NULL,
NAZWISKO VARCHAR(20) CHARACTER SET DOS852 NOT NULL,
IMIE VARCHAR(20) CHARACTER SET DOS852 NOT NULL,
DATA_UR DATE,
DATA_REKRUTACJI DATE,
IMIE_OJCA VARCHAR(20) CHARACTER SET DOS852,
KIER_STUDIOW VARCHAR(20) CHARACTER SET DOS852 NOT NULL,
SECJALNOSC VARCHAR(20) CHARACTER SET DOS852 NOT NULL,
GRUPA INTEGER,
SREDNIA FLOAT DEFAULT 0,
PRIMARY KEY (PESEL));

/* Extract Table OCENY */
/* Table: OCENY, Owner: SYSDBA */
CREATE TABLE OCENY (KIER_STUDIOW VARCHAR(20) CHARACTER SET DOS852,
SPECJALNOSC VARCHAR(20) CHARACTER SET DOS852,
SEMESTR INTEGER,
PRZEDMIOT VARCHAR(35) CHARACTER SET DOS852,
TER_EGZAMINU INTEGER,
RODZAJ_STUDIOW CHAR(7) CHARACTER SET DOS852 NOT NULL,
ZALICZENIE FLOAT,
PESEL VARCHAR(11) CHARACTER SET DOS852,
ROK INTEGER NOT NULL,
FOREIGN KEY (PESEL) REFERENCES STUDENTDANE(PESEL) ON UPDATE CASCADE);
```

Przykład 1. Kod SQL tabel STUDENTDANE i OCENY

Kluczem głównym tabeli *STUDENTDANE* jest pole *PESEL*. Tabele *STUDENTDANE* oraz *OCENY* są ze sobą powiązane referencyjnie poprzez pole *PESEL*, ponadto powiązanie zapisane poniżej:

```
FOREIGN KEY (PESEL) REFERENCES STUDENTDANE(PESEL) ON UPDATE CASCADE
```

powoduje, że dane w tabeli *OCENY* w polu *PESEL* będą kaskadowo modyfikowane. Usunięcie studenta z tabeli *STUDENTDANE* jest niemożliwe, jeśli ma on jakiegokolwiek rekordy ocen.

3.4 Obsługa zapytań w Delphi

Zapytania w Delphi można wywołać używając odpowiedniego komponentu jakim jest *Query*. Przykładowy kod programu obsługujący przycisk w Delphi, oraz wywołujący przy tym proste zapytanie typu:

```
SELECT * FROM STUDENTDANE WHERE STUDENTDANE.PESEL = :PESEL
```

przedstawia przykład 2.

```
procedure TStudent.BitBtn1Click(Sender: TObject);
begin
try
begin
Query1.Close;
Query1.SQL.Clear;
Query1.UnPrepare;
Query1.SQL.Add('SELECT * FROM STUDENTDANE');
Query1.SQL.Add('WHERE STUDENTDANE.PESEL=:PESEL');
Query1.ParamByName('PESEL').AsString:=MaskEdit1.Text;
Query1.Prepare;
Query1.Open;
end;
except
On EDBEngineError Do
begin
MessageDlg('Zły parametr wyszukiwania!', mtError, [mbOk], 0);
Abort;
end;
end;
end;
```

Przykład 2. Obsługa zapytań w Delphi

Jest to zapytanie parametryczne. Parametrem przekazywanym do serwera SQL jest *PESEL*. Parametr musi być odpowiednio przypisany, tak jak w przykładzie 2. Przed przystąpieniem do wykonania zapytania, należy wyłączyć komponent *Query1*, następnie wyczyścić pole SQL w komponencie *Query1*, poleceniem *Query1.Clear* oraz przygotować zapytanie do wykonania (klauzula *Query1.UnPrepare* - *Query1.Prepare*). Przygotowanie zapytania jest konieczne ponieważ przyspiesza jego wykonanie.

W przykładzie 2 pokazano obsługę błędów w czasie wykonywania zapytania (klauzula *try* - *except*). Program wyświetla komunikat na temat błędu, oraz opuszcza wykonywanie zapytania poleceniem *Abort*.

Ważnym elementem każdej aplikacji jest obsługa błędów. Komunikaty podpowiedzi, występujące podczas niewłaściwego użytkownika programu, stają się bardzo cenną informacją dla użytkownika. Niewłaściwe zabezpieczenie programu przed użytkownikiem może spowodować nawet zawieszenie systemu operacyjnego i utratę nie zapisanych wcześniej danych.

3.5 Procedury składowane w SQL-u i ich obsługa w Delphi

Procedury składowane są to procedury wykonywane bezpośrednio na serwerze bazy danych. Mogą one być bardziej skomplikowane aniżeli zwykłe zapytania dostępne z języka SQL. Interbase pozwala tworzyć dwa rodzaje procedur składowanych: wykonawcze i wybierające. Procedury składowane tworzy się za pomocą wyrażenia SQL – *CREATE PROCEDURE*. Wyrażenie to zbudowane jest z nagłówka oraz bloku kodu. Nagłówek zawiera nazwę procedury, która musi być unikatową nazwą. W nagłówku procedury definiujemy parametry wejściowe oraz wyjściowe. Blok kodu procedury zawiera listę zmiennych lokalnych, a zaraz po nich znajduje się kod właściwy otoczony słowami kluczowymi *BEGIN* i *END*.

```
CREATE PROCEDURE SREDNIA (PESEL VARCHAR(11) CHARACTER SET DOS852,
ROK INTEGER)
AS
DECLARE VARIABLE ILOSC INTEGER;
BEGIN
SELECT COUNT(*) FROM OCENY WHERE PESEL=:PESEL AND ROK=:ROK AND (ZALICZENIE IS
NULL OR EGZAMIN IS NULL)
INTO:ILOSC;
IF (ILOSC=0) THEN BEGIN
UPDATE OCENY SET SREDNIA=((SELECT SUM (ZALICZENIE) FROM OCENY WHERE
PESEL=:PESEL AND ROK=:ROK) +(SELECT SUM (EGZAMIN) FROM OCENY WHERE
PESEL=:PESEL AND ROK=:ROK)) / ((SELECT COUNT(ZALICZENIE) FROM OCENY WHERE
PESEL=:PESEL AND ROK=:ROK AND ZALICZENIE>0)+(SELECT COUNT(EGZAMIN) FROM OCENY
WHERE PESEL=:PESEL AND ROK=:ROK AND EGZAMIN>0))
WHERE PESEL=:PESEL AND ROK=:ROK;
END
```

Przykład 3. Procedura składowana zliczająca średnią

Przykład 3 pokazuje zastosowanie procedury składowanej w aplikacji. Procedura ta sumuje wszystkie wiersze tabeli *OCENY* z pola *ZALICZENIE*, *EGZAMIN* pod warunkiem parametrów *PESEL* i *ROK*. Następnie wynik ten jest dzielony przez wiersze tabeli *OCENY* z pola *ZALICZENIE*, *EGZAMIN* pod warunkiem parametrów *PESEL* i *ROK* oraz pola *ZALICZENIE>0* i *EGZAMIN>0* i wstawiany do kolumny *SREDNIA*.

Obsługa procedur składowanych w Delphi przedstawiona jest w przykładzie 4. Wcześniej stworzoną procedurę składowaną, która funkcjonuje na serwerze bazy danych, w Delphi wywołuje się przy użyciu komponentu *StoredProc*, poleceniem *ExecProc*. Podobnie jak w przypadku komponentu *Query*, można wykorzystać klauzule *UnPrepare* - *Prepare*.

```
procedure TOceny.BitBtn1Click(Sender: TObject);
begin
with StoredProc1 do
begin
Close;
UnPrepare;
ParamByName('PESEL').AsString:=DBText1.Field.AsString;
ParamByName('ROK').AsInteger:=DBText2.Field.AsInteger;
Prepare;
ExecProc;
end;
end;
```

Przykład 4. Obsługa procedur składowanych w Delphi

4. WNIOSKI

W obecnych czasach systemy baz danych są szeroko rozpowszechnionym narzędziem w wielu dziedzinach pracy. Mają one zastosowanie wszędzie tam, gdzie potrzeba szybko i sprawnie magazynować dane. Ciągły rozwój technik informatycznych sprawia iż projekty systemów baz danych, powstają małym nakładem finansowym. W niniejszym artykule przedstawiono prostą aplikację, która spełnia podstawowe zadania funkcjonowania systemu bazy danych.

Aplikacje napisano za pomocą Delphi 5.0 Enterprise oraz wykorzystano sterownik Interbase 5.5, jako łącznik języka SQL z platformą Delphi. W aplikacji zastosowano architekturę typu klient-serwer.

Pakiet Delphi jest doskonałym narzędziem do tworzenia baz danych, natomiast sterownik Interbase oferuje podstawowe elementy języka SQL. Sterownik ten jest dołączany wraz z pakietem Delphi.

Wszystkie przykłady zamieszczone w niniejszym artykule są rzeczywistymi blokami aplikacji *Student 2003'*. Zastosowano standardowe komponenty dostępne w pakiecie Delphi do budowy programu.

5. BIBLIOGRAFIA

1. Głąb M., Praca dyplomowa, Wydział Elektryczny Politechniki Częstochowskiej, Częstochowa 2003.
2. Głąb M., Krzemiński J., System ewidencji wspomagający pracę dziekanatu, V MSKAE 2003, Poraj.
3. Jakubowski A., Tworzenie systemów baz danych, Helion, Gliwice 1999.
4. Coburn R., SQL dla każdego, Helion, Gliwice 2000.
5. Interbase 5 documentation.
6. Delphi 5.0 Enterprise – pomoc programu.

THE DELPHI AND SQL BASED THE DATABASE SYSTEM FOR THE DEAN'S OFFICE

This paper presents the project of the database application *Student 2003'*. It was introduced: the characteristic of software in Delphi, working the application on server Interbase, creating in SQL language the database (creating the database file, tables), creating a questions, stored procedures, service of stored procedures in Delphi.