PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA
W NOWYM SĄCZU

INSTYTUT TECHNICZNY

PRACA DYPLOMOWA

SYSTEM GENEROWANIA RAPORTÓW W PROCESIE REKRUTACJI KANDYDATÓW NA STUDIA PROWADZONE W PWSZ W NOWYM SĄCZU.

Autor: Paweł Mysiński

Kierunek: Informatyka

Nr albumu: 20747

Promotor: dr inż. Antoni Ligęza

NOWY SĄCZ 2015

Spis treści

1. Wprowadzenie

testesttest

1.1. Zagadnienie generowania raportów

1.2. Dotychczasowy proces generowania raportów

1.3. Cel i zakres pracy

2. Testowanie Systemu

Przed wdrożeniem programu do realnego systemu, program należy przetestować. Test powinny być prowadzone na tymczasowej bazie danych, ponieważ idea testów jest taka, że po podmianie bazy danych na realną wszystko ma działać bez zmian. Zmienić się może tylko zapis połączenia z bazą danych w pliku konfiguracyjnym. Dzieki takiemu zabiegowi, będzie można być pewnym tego, że wszystko będzie działać na prawdziwej bazie danych. Do przeprowadzenia testów potrzebne będzie więc odtworzyć przyszłe środowisko, w którym będzie działać program, przygotować szablony dokumentów, które są wytwarzane w procesie rekrutacji oraz wygenerować dokumenty. Ostatnim już krokiem będzie sprawdzenie czy podczas tego procesu nie ma żadnych komplikacji oraz czy wygenerowane dokumenty nie zawierają błędów.

2.1. Generowanie przykładowych danych

W celu przetestowania systemu generowania raportów w procesie rekrutacji kandydatów na studia potrzebne będą fikcyjne dane w dokładnie tej samej strukturze co w systemie rekrutacji, ponieważ w szablonach latexu znajdują się zapytania SQL do danych tabel w bazie danych. W celu otrzymania tych danych potrzebne będzie: 1. Utworzenie nowej bazy danych na silniku Firebird'a 2. Stworzenie wystarczającej struktury tabel odzwierciedlającą strukturę w systemie uczelnianym. 3. Wygenerowanie dużej ilości fikcyjnych osób. 4. Uzupełnienie tabel danymi, które zostały wygenerowane wcześniej oraz dodanie do nich dodatkowych, jednocześnie losowych, informacji na temat procesu rekrutacji. Po wykonaniu tych kroków, powinna powstać baza danych do której bez problem program połączy się i wyciągnie z niej potrzebne dane dokładnie jak z prawdziwej bazy danych.

2.1.1. Tworzenie bazy danych

Do stworzenia pliku bazy danych na silniku firebird'a posłużyć się można narzędziem dostepnym w katalogu bin zainstalowanego serwera firebird. Narzędzie to pozwala z lini komend tworzyć i łączyć się z bazami danych. W tym przypadku użyta zostanie komenda „CREATE DATABASE"

```
C:\Program Files\Firebird\bin>isql
SQL>CREATE DATABASE 'D:\test systemu\rekrutacja.fdb'
CON>user 'SYSDBA' password 'masterkey';
```

Po wykonaniu tego polecenia zostanie utworzona baza danych. Takie same dane należy teraz wpisać do pliku konfiguracyjnego programu czyli DBRaportLatex.bat aby program mogł się połączyć z tą bazą:

```
dbengine=firebirdsql
hostname=//localhost
port=3050
```

```
dbpath=D:\test systemu\rekrutacja.fdb
user=SYSDBA
password= masterkey
```

2.1.2. Stworzenie struktury

Dotychczasowy system wykorzystywał tabelę (widok) która była generowana dynamicznie i która zawiera wszystkich studentów w rekrutacji. Zawiera ona wszystkie dane potrzebne do wytworzenia dokumentów. Jeden rekord to jeden student ze wszystkimi informacjami na jego temat. Dodatkowo jeszcze potrzebna jest tabela z informacjami na temat rekrutacji, takimi jak na przykład nazwisko i imie przewodniczącego komisji, czy data wydania dezycji przyjęcia studenta. Z tych tabel będą pobierane informacje, natomiast do wygenerowania danych potrzebne będą dwie dodatkowe tymaczasowe tabele. Tabela główna z kandydatami(zapis SQL):

```
CREATE TABLE KANDYDAT_ALIGEZA (
    STUD_ID                          INTEGER NOT NULL,
    OSOBA_ID                         INTEGER NOT NULL,
    STUD_NRTECZKI                    INTEGER,
    NAZWISKO                         VARCHAR(100),
    IMIE                             VARCHAR(100),
    NAZWISKOIMIONA                   VARCHAR(200),
    ADR_ULICA_MIEJSCOWOSC_NR_DOMU    VARCHAR(200),
    ADR_KOD_POCZTOWY_POCZTA          VARCHAR(100),
    STUDIA_NAZWA                     VARCHAR(100),
    STUD_ILPUNKTOW                   INTEGER,
    STUD_ILPUNKTOWKREM               INTEGER,
    TOKNAUKI_NAZWATOKU               VARCHAR(200),
    KIERUNEK                         VARCHAR(100),
    SPEC_ID                          INTEGER,
    DATAPRZYJECIAPODANIA             DATE,
    TOKNAUKI_ID                      INTEGER,
    OSOBA_PESEL                      VARCHAR(50),
    KIERUNEK_MY                      VARCHAR(200),
    FORMA_STUDIOW_MY                 VARCHAR(200),
    STOPIEN_STUDIOW_MY               VARCHAR(100),
    KIERUNEK_FORMA_SKROT_MY          VARCHAR(100),
    NR_DECYZJI                       VARCHAR(100),
    CZY_PRZYJETY                     INTEGER,
    DATA_DECYZJI                     DATE,
    ILE_PUNKTOW                      INTEGER,
    PANPANI                          CHAR(1)
);
```

Tabela z dodatkowymi informacjami(wartości przypisane są do kluczy tekstowych, jest to tablica asocjacyjna):

```
CREATE TABLE SETUP_ALIGEZA (
    KLUCZ     VARCHAR(50) NOT NULL,
    WARTOSC   VARCHAR(100)
```

```
);
```

Tymczasowa tabela do zaimportowania listy imion i nazwisk oraz losowych peseli.

```
CREATE TABLE DANE (
    IMIE_NAZ    VARCHAR(200),
    ADRES       VARCHAR(200),
    PESEL       VARCHAR(50),
    NAZWISKO    VARCHAR(100),
    IMIE        VARCHAR(100)
);
```

Tymczasowa tabela do procedury losowego uzupełniania informacji o rekrucie o kierunku jaki wybrał.

```
CREATE TABLE TOKNAUKI_ALIGEZA (
    TOKNAUKI_ID                 INTEGER NOT NULL,
    KIERUNEK_MY                 VARCHAR(50),
    FORMA_STUDIOW_MY            VARCHAR(50),
    STOPIEN_STUDIOW_MY          VARCHAR(50),
    KIERUNEK_FORMA_SKROT_MY     VARCHAR(10),
    LICZBA_MIEJSC               SMALLINT,
    DATA_DECYZJI_OD             TIMESTAMP,
    DATA_DECYZJI_DO             TIMESTAMP,
    KOD_IKR                     VARCHAR(3)
);
```

2.1.3. Generowanie fikcyjnych danych osobowych

Do wygenerowania kandydatów potrzeba imienia nazwiska oraz adresu. Takie dane dostępne są w książkach telefonicznych. Posługując się jedną z takich książek stworzony został plik csv o seperatorze „;” zawierajacy po kolei imie z nazwiskiem, adres, pesel, nazwisko oraz imie. Pesel został dodany do każdej osoby jako losowy ciąg cyfr spełniający walidację pesela. Ze względu na fakt iż pesel został wygenerowany losowo, może zdarzyć się iż mężczyzna posiadać będzie kobiecy pesel, w następstwie czego, we wygenerowanych dokumentach wypisane zostanie „Pani” i na odwrót. Struktura pliku:

```
Abram Andrzej; Lwowska 116;88071640299;Abram;Andrzej
Abram Halina; Ludwika Zamenhofa 2;86111210691;Abram;Halina
...
```

Taki plik bardzo łatwo zaimportować do bazy danych do tabeli „dane”. Do importu wykorzystana została funkcja programu IBExpert „import data”. Jedna linijka w pliku zostaje zaimportowana jako jeden rekord, w którym każde pole po kolei odpowiada wartościom między średnikami. Zaimportowanych w ten sposób zostało 10001 rekordów (osób) do tabeli „dane” do dalszych manipulacji.

2.1.4. Generowanie kandydatów na studentów

Kolejnym krokiem jest uzupełnienie tabeli z tokami studiów. W testach dodanych zostało 8 przykładowych toków nauki.

```
1 Informatyka    niestacjonarne  pierwszego stopnia      INF-n
2 Mechatronika   niestacjonarne  pierwszego stopnia      MT-n
3 Mechatronika   stacjonarne     pierwszego stopnia      MT-s
...
```

Uzupełnienia wymaga także tabela z dodatkowymi informacjami „SETUP_ALIGEZA" przykładowymi danymi.

```
dataWydaniaDecyzji          09.10.2015
miejsceWydaniaDecyzji       Nowy Sącz
przewodniczacyIKR           mgr inż. Sławomir Jurkowski
rokAkademicki    2015/2016
czyUwzglednicDateWydaniaDecyzji N
...
```

Mając już to wszystko potrzebna jest procedura, która utworzy listę kandydatów z tych wszystkich danych.

```
create procedure GENERUJ
returns (
    TESTCHAR varchar(50),
    TEST integer)
as
declare variable IMIE varchar(100);
declare variable NAZ varchar(100);
declare variable IMIENAZ varchar(200);
declare variable ADRES varchar(200);
declare variable PESEL varchar(50);
declare variable LICZNIK integer;
declare variable STOPIEN varchar(50);
declare variable KIERUNEK varchar(50);
declare variable FORMA varchar(50);
declare variable SKROT varchar(10);
declare variable RANDINT integer;
declare variable PUNKTY integer;
declare variable CZY_PRZYJETY integer;
declare variable DATA_DEC varchar(100);
begin
licznik = 1;
for select * from dane into
:imienaz,:adres,:pesel,:naz,:imie
do
begin
randint = CAST(round(rand()*7+1) as INTEGER);
punkty = CAST(round(rand()*500) as INTEGER);
if(punkty > 250) then czy_przyjety = 1;
if(punkty <= 250) then czy_przyjety = 2;

select kierunek_my,forma_studiow_my,stopien_studiow_my,
kierunek_forma_skrot_my
```

```
FROM toknauki_aligeza where toknauki_id = :randint
INTO :kierunek,:forma,:stopien,:skrot;

select wartosc FROM setup_aligeza
WHERE klucz='dataWydaniaDecyzji'
INTO :data_dec;

INSERT INTO kandydat_aligeza
(stud_id,osoba_id,stud_nrteczki,nazwisko,imie,
nazwiskoimiona,adr_ulica_miejscowosc_nr_domu,
adr_kod_pocztowy_poczta,osoba_pesel,panpani,
studia_nazwa,toknauki_nazwatoku,kierunek,
kierunek_my,forma_studiow_my,stopien_studiow_my,
kierunek_forma_skrot_my,     stud_ilpunktow,
stud_ilpunktowkrem,ile_punktow,czy_przyjety,
data_decyzji,nr_decyzji,dataprzyjeciapodania)
VALUES (:licznik,:licznik,cast(round(rand()*200+1) as integer),
:naz,:imie,:imienaz,:adres,
cast( 'Nowy Sacz 33-300' as varchar(100)),
:pesel,cast( 'M' as char(1)),     :forma,:kierunek ||
' N inz. 3.50 2015/2016 zimowy',:kierunek,:kierunek,:forma,
:stopien,:skrot,     :punkty,:punkty,:punkty,:czy_przyjety,
cast(:data_dec as DATE),'328/2015','2015-08-14');

    licznik = :licznik + 1;
end
test = :licznik;
suspend;
end
```

Powyższa procedura z jednej osoby z tabeli dane tworzy jednego kandydata, losując mu tok nauki, ilość punktów oraz czy zostanie przyjęty lub nie. Dorzucane są także pewne stałe wartości, podobne do tych w oryginalnej bazie danych, które nie wymagają uzmiennienia. Po wykonaniu jednorazowym tej procedury powinniśmy otrzymać wszystkie dane potrzebne do przeprowadzenia procesu generowania dokumentów potrzebnych w rekrutacji.

Bibliografia