



SKRYPT DO LABORATORIUM

## WYMIANA I SKŁADOWANIE DANYCH MULTIMEDIALNYCH

### ĆWICZENIE 2: Elektroniczna dokumentacja medyczna – składowanie dokumentów (bazy danych XML) oraz obsługa elektronicznych kart pacjenta

autor:

dr inż. Jacek Rumiński

Gdańsk, 2010



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



## **1. Zagadnienia wstępne**

### **1.1.Wymagania w odniesieniu do studenta**

Osiągnięcie celów ćwiczenia będzie możliwe poprzez właściwe przygotowanie się studenta do zajęć. Będzie to możliwe poprzez:

- powtórzenie wiedzy nabytej w czasie wykładów,
- zapoznanie się z instrukcją do ćwiczenia ilustrującą podstawowe zagadnienia z zakresu tematyki ćwiczenia, a szczególnie formatu XML oraz jego zastosowania w aplikacjach medycznych (HL7).

Ponadto wymagana jest od studenta:

- podstawowa umiejętność programowania,
- umiejętność twórczego myślenia.

### **1.2.Wymagania w odniesieniu do stanowiska laboratoryjnego**

Stanowisko laboratoryjne powinno być wyposażone w komputer klasy PC z działającym oprogramowaniem przygotowanym specjalnie dla potrzeb realizacji ćwiczenia. Na komputerze dostępne powinna być zainstalowana instrukcja do ćwiczenia w wersji elektronicznej oraz pliki testowe.

### **1.3.Cele ćwiczenia**

Celem ćwiczenia jest praktyczna demonstracja wiedzy teoretycznej, przedstawionej w ramach wykładów. Realizując ćwiczenie laboratoryjne studenci zdobędą szereg nowych umiejętności w zakresie zastosowania rekomendacji XML w składowaniu i wymianie danych medycznych, w tym:

- składowania danych XML w bazach danych,
- przetwarzania danych XML z baz danych i ich integracji z danymi relacyjnymi,
- obsługi i interpretacji danych z kart elektronicznych (karty inteligentne, pamięciowe, itd.).

### **1.4.Metody dydaktyczne**

Student w realizuje kolejne zadania opisane w instrukcji. Układ zadań jest tak dobrany, aby zapoznać ćwiczącego z podstawowymi (wybranymi), kolejnymi operacjami przetwarzania i składowania dokumentów XML. Ćwiczący ma do dyspozycji kody źródłowe poleceń SQL oraz przykładowe stanowisko pracy z kartami elektronicznymi w ochronie zdrowia. Wszystkie utworzone dokumenty i zmodyfikowane kody poleceń SQL (oraz wyniki) wpisuje w dokumencie elektronicznym, który jednocześnie staje się sprawozdaniem z ćwiczenia laboratoryjnego.

Student w trakcie ćwiczenia projektuje prosty dokument XML zgodny z CDA lub formatami medycznymi specyficznymi w Polsce (NFZ).

## 1.5. Zasady oceniania

Ocenie podlegać będzie realizacja poszczególnych zadań w ramach danego ćwiczenia laboratoryjnego. Dodatkowo w czasie realizacji ćwiczenia przydzielone zostaną studentowi dodatkowe zadania. Student musi zrealizować ćwiczenie laboratoryjne. Dodatkowym warunkiem zaliczenia jest uzyskanie minimum 16 punktów na podstawie oceny sprawozdań/plików źródłowych.

### Wykaz literatury podstawowej do ćwiczenia:

#### Wykaz literatury podstawowej:

1.	Skrypt z materiałami do przedmiotu „Wymiana i składowanie danych multimedialnych”
2.	Materiały do przedmiotu opracowane w formie edukacji na odległość, dostęp: <a href="http://uno.biomed.gda.pl">http://uno.biomed.gda.pl</a>
3.	HL7, norma i dokumenty HL7, dostęp: <a href="http://www.hl7.org">http://www.hl7.org</a>
4.	Norma SQL, rozdział 14: SQL/XML. Dokumentacja techniczna IBM DB2 w zakresie obsługi XML.

#### Wykaz literatury uzupełniającej:

## 2. Przebieg ćwiczenia

L.p.	Zadanie
1.	Zapoznanie się z instrukcją laboratoryjną
2.	Podstawowa administracja bazą danych z obsługa XML
3.	Tworzenie struktur danych dla XML oraz dodawanie danych XML
4.	Przetwarzanie danych XML z baz danych
5.	Przetwarzanie HL7 CDA w bazach XML
6.	Wykonywanie podstawowych operacji na kartach elektronicznych

### UWAGI!

1. PRZED przystąpieniem do ćwiczenia należy w katalogu d:\dydaktyka\studenci utworzyć własny katalog (imie\_nazwisko), a następnie przekopiować tam zawartość katalogu d:\dydaktyka\wisd\cw2. Proszę pracować wyłącznie na wykonanej kopii plików.

### 3. Wprowadzenie do ćwiczenia

W czasie realizacji ćwiczenia realizowane będą zadania zgodnie z ich wcześniejszym opisem. Każde zadanie (z wyjątkiem zadania realizowanego przed przystąpieniem do ćwiczenia laboratoryjnego) opisano poniżej.

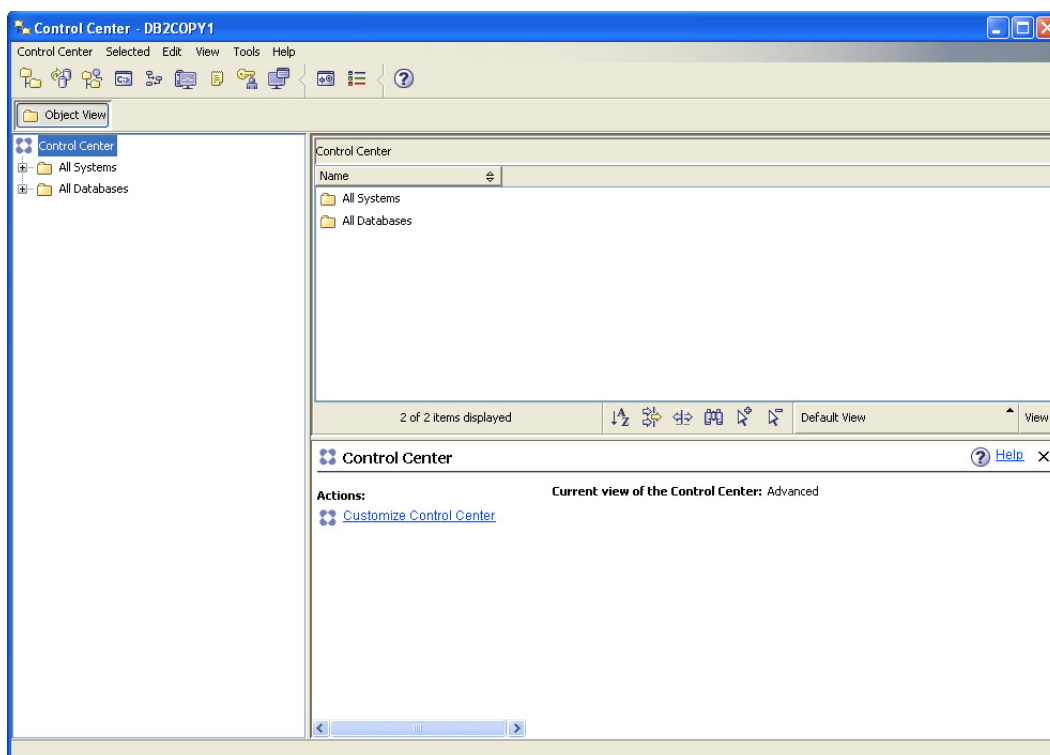
#### Ad 1. Zapoznanie się z instrukcją laboratoryjną (5 min.)

Po wprowadzeniu kierownika ćwiczenia uczestnicy zapoznają się ze stanowiskiem komputerowym, oprogramowaniem i dokumentami związanymi z ćwiczeniem.

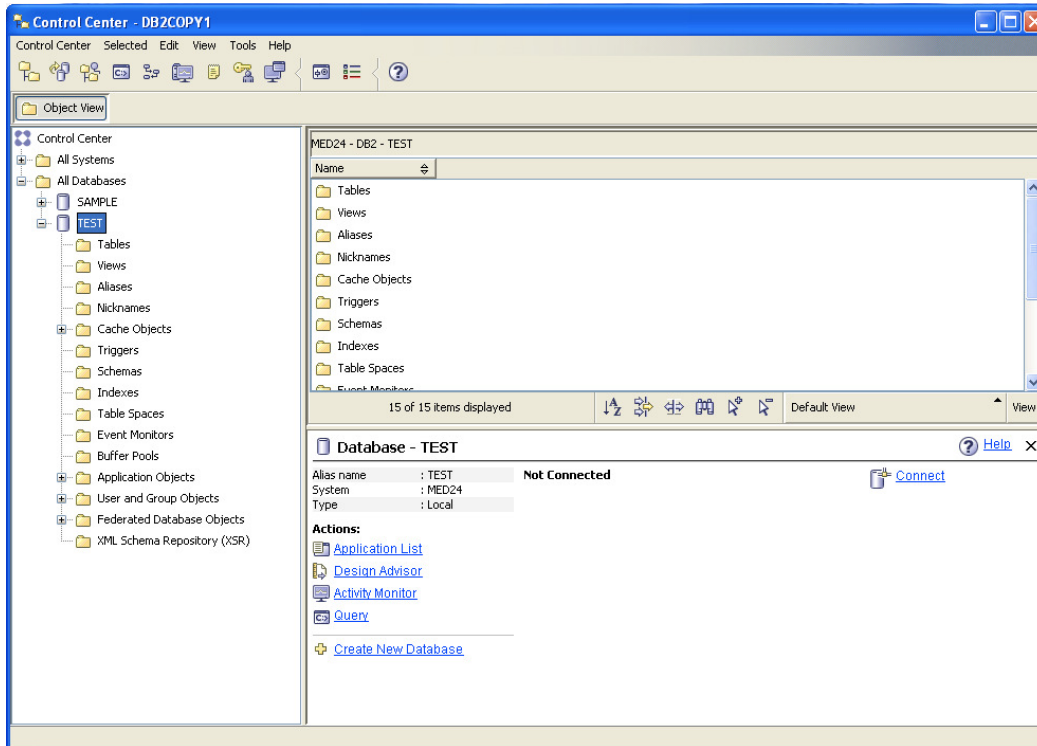
Następnie utworzyć we wskazanym przez kierownika ćwiczenia katalogu własny podkatalog o nazwie zawierającej własne nazwisko. Do katalogu tego przegrać zawartość podkatalogu „DANE” (znajdującego się w folderze zawierającym instrukcję do ćwiczenia). W utworzonym podkatalogu należy przechowywać wszystkie wytworzone w czasie trwania ćwiczenia dokumenty i programy.

#### Ad 2. Podstawowa administracja bazą danych z obsługą XML (15 min.)

Uruchomić oprogramowanie IBM DB2 Control Center. Pojawi się okno dialogowe podobne do ukazanego poniżej.



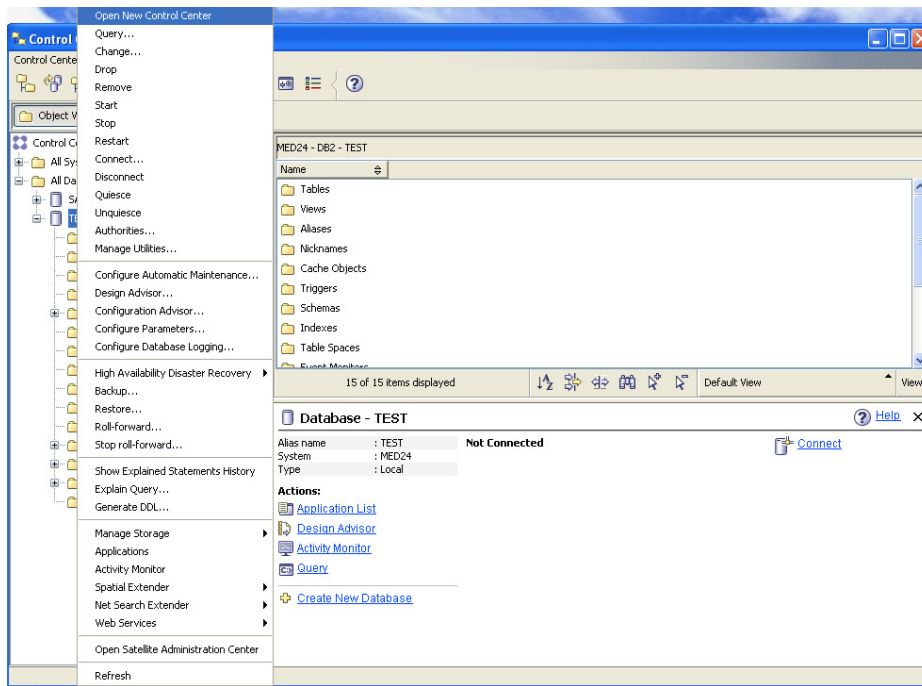
Rozwinąć zakładkę „All Databases” i wybrać bazę danych TEST, co pokazano na poniższym rysunku:



Wykorzystując dokumentację IBM DB2 zapoznać się z podstawowymi elementami zarządzania bazami danych: połączenie z bazą, uruchomienie interpretera poleceń SQL, przeglądanie wyników zapytań, przeglądanie elementów składowych bazy danych (tabele, widoki, itp.).

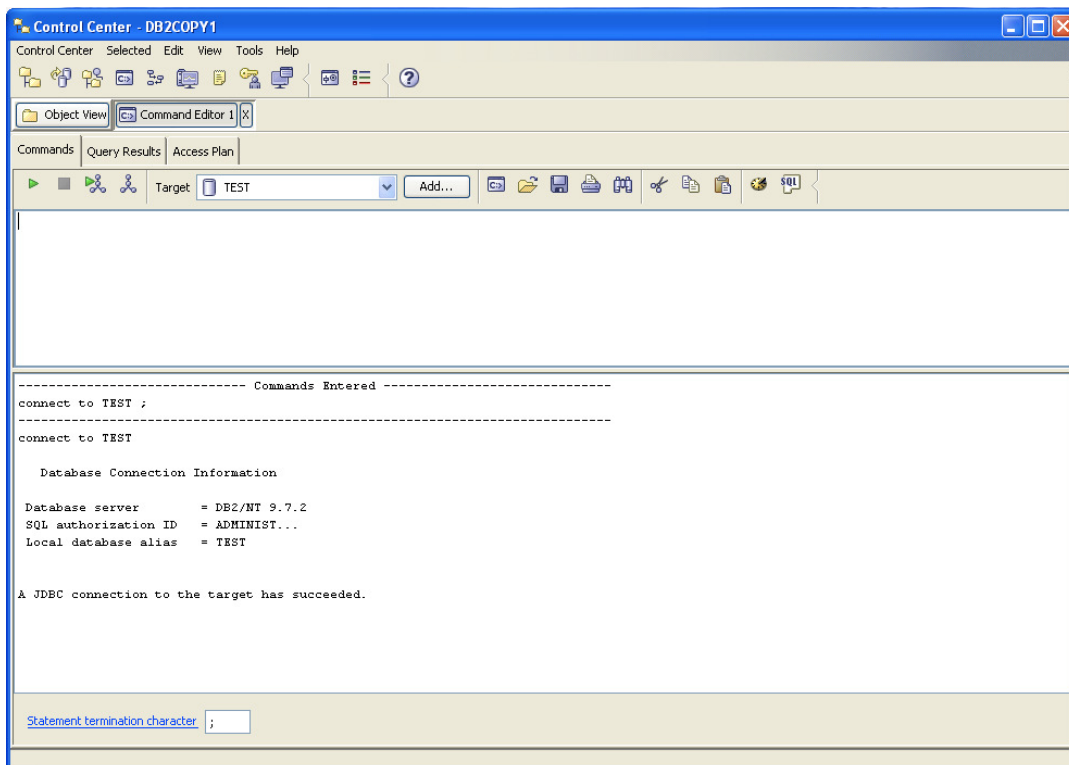
### Ad 3. Tworzenie struktur danych dla XML oraz dodawanie danych XML (20 min.)

Po uruchomieniu interpretera SQL dla danej bazy danych (wybrać nazwę bazy danych TEST i z menu kontekstowego – prawy przycisk myszki – wskazać opcję zapytania – Query).



Następnie w oknie edytora zapytań wprowadzić polecenie języka SQL, za pomocą którego usunięta zostanie tabela „teachers” w schemacie „gut” (o ile taka tabela już istnieje – została wprowadzona przez wcześniejszych wykonawców ćwiczenia).

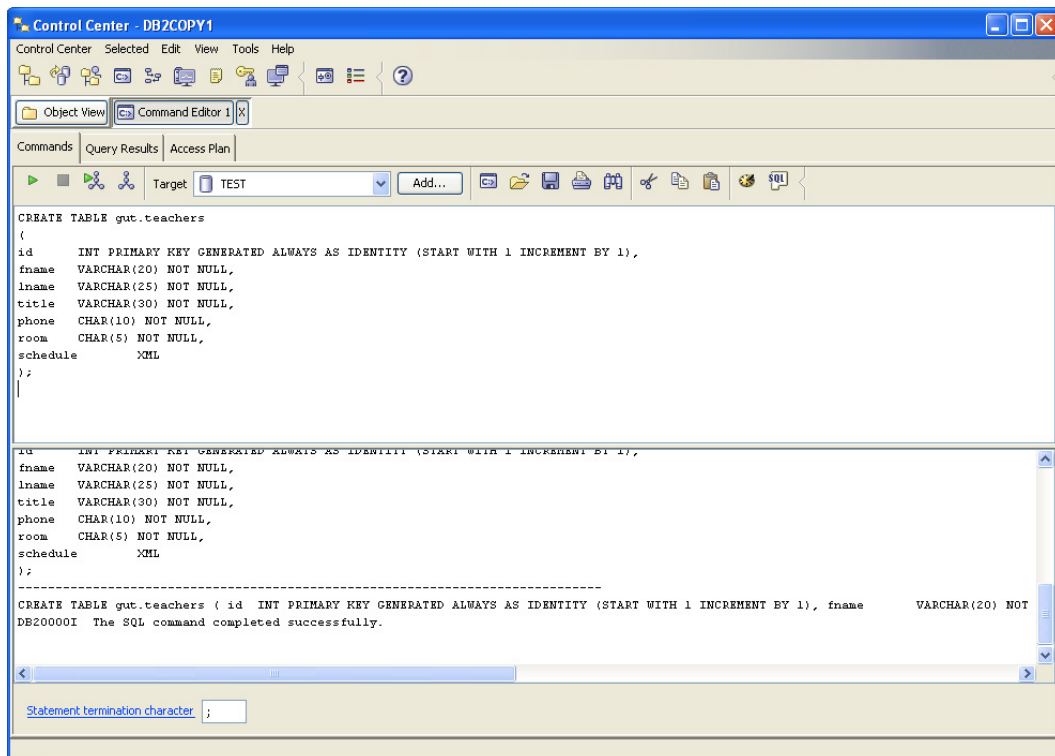
*DROP TABLE gut.teachers;*



Następnie wprowadzić polecenie tworzenie tabelę „teachers” w schemacie „gut”. Tabela zawierać będzie pojedyncze pole typu XML:

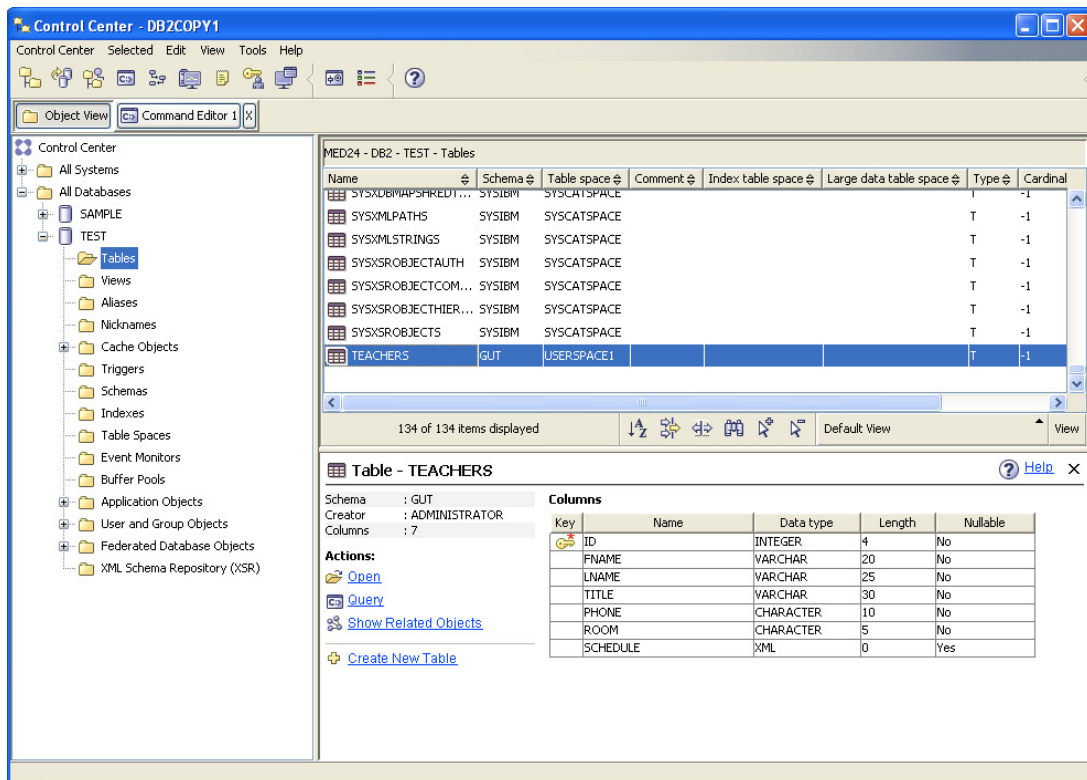
```
CREATE TABLE gut.teachers
(
id      INT PRIMARY KEY GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
fname  VARCHAR(20) NOT NULL,
lname  VARCHAR(25) NOT NULL,
title  VARCHAR(30) NOT NULL,
phone  CHAR(10) NOT NULL,
room   CHAR(5) NOT NULL,
schedule XML
);
```

gdzie gut – to nazwa schematu (jeśli nie ma należy go utworzyć CREATE SCHEMA gut).



Efekt wykonania polecenia SQL komentowany jest przez oprogramowanie Control Center w dolnej części okna. Jeśli pojawia się informacja typu „The SQL command completed successfully” oznacza to, że operacja zakończyła się sukcesem, w przeciwnym przypadku należy sprawdzić składnię polecenia SQL (zgodnie z pokazanymi informacjami o błędzie).

Weryfikację poprawności utworzonej tabeli „teachers” można sprawdzić przeglądając tabele bazy danych test, co pokazano poniżej.



#### Ad 4. Przetwarzanie danych XML z baz danych (30 min.)

Po utworzeniu struktury danych, umożliwiającej przechowywanie dokumentów XML dodajmy dwa wiersze do tabeli za pomocą polecenia SQL INSERT:

```
INSERT INTO gut.teachers (fname, lname, title, phone, room, schedule) VALUES
('Jacek', 'Ruminski', 'PhD Eng', '583472678', '106',
XMLPARSE( DOCUMENT
'<workplan year="2017" semester="summer">
```

```
  <monday>
    <lecture time="11-13" room="238">Structural Programming</lecture>
    <lecture time="13-14" room="221">Diploma Seminar</lecture>
    <lab time="16-18" room="215">Distributed Systems</lab>
  </monday>
  <tuesday>
    <lecture time="13-14" room="222">Distributed Systems</lecture>
  </tuesday>
  <wednesday>
    <lecture time="12-14" room="222">Operating Systems </lecture>
    <lecture time="15-17" room="222">Electronics </lecture>
  </wednesday>
```



```

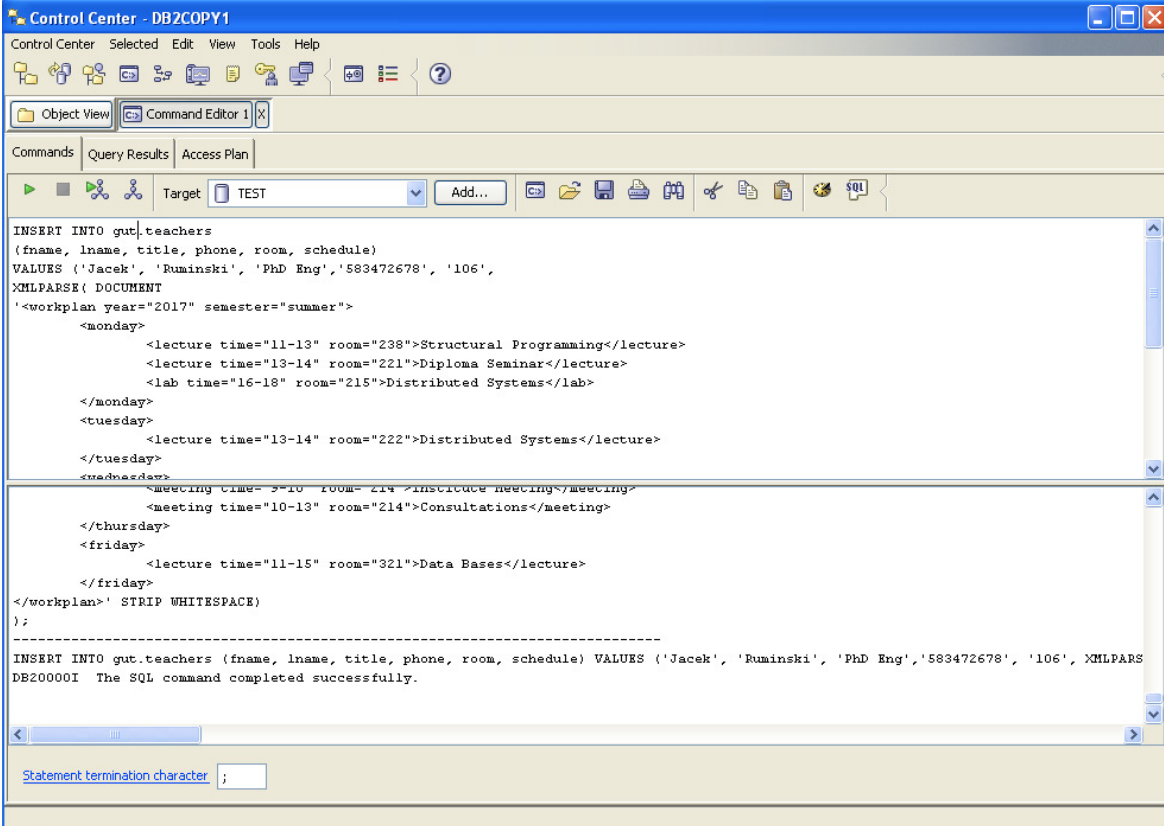
<thursday>
    <meeting time="9-10" room="214">Institute Meeting</meeting>
    <meeting time="10-13" room="214">Consultations</meeting>
</thursday>
<friday>
    <lecture time="11-15" room="321">Data Bases</lecture>
</friday>
</workplan>'
STRIP WHITESPACE) );

INSERT INTO gut.teachers (fname, lname, title, phone, room, schedule) VALUES
('Adam', 'Bujnowski', 'PhD Eng', '583471684', '105',
XMLPARSE( DOCUMENT
'<workplan year="2017" semester="summer">
    <monday>
        <lecture time="11-13" room="238">Data Bases</lecture>
        <lecture time="13-14" room="221">Diploma Seminar</lecture>
        <lab time="14-16" room="215">Data Exchange in Medicine</lab>
    </monday>
    <tuesday>
        <lecture time="13-14" room="222">Data Exchange in Medicine</lecture>
    </tuesday>
    <wednesday>
        <lecture time="12-14" room="221">Application servers</lecture>
        <lecture time="14-16" room="221">Electronics </lecture>
    </wednesday>
    <thursday>
        <meeting time="9-10" room="214">Department Meeting</meeting>
        <meeting time="10-13" room="215">Consultations</meeting>
    </thursday>
    <friday>
        <lecture time="11-15" room="321">Data Bases</lecture>
    </friday>
</workplan>'
STRIP WHITESPACE) );

```

Dodanie wartości typu XML wymagało zastosowania funkcji XMLPARSE. Funkcja ta jest również zdefiniowana w rozdziale 14 normy SQL i reprezentuje wykonanie operacji przeniesienia pliku tekstowego XML (forma serializacji) do struktury drzewa w wewnętrznym modelu systemu bazy danych (w tym przypadku IBM DB2). Operacja odwrotna, XMLSERIALIZE, odwzorowuje dane XML z wewnętrznej reprezentacji w bazie danych do postaci tekstowej (pliku tekstowego XML). Rozdział 14 normy SQL wprowadza jeszcze szereg innych funkcji (np. odwzorowania danych/schematu XML na dane/schemat relacyjny).

W środowisku Control Center należy wykonać polecenia SQL INSERT.



```

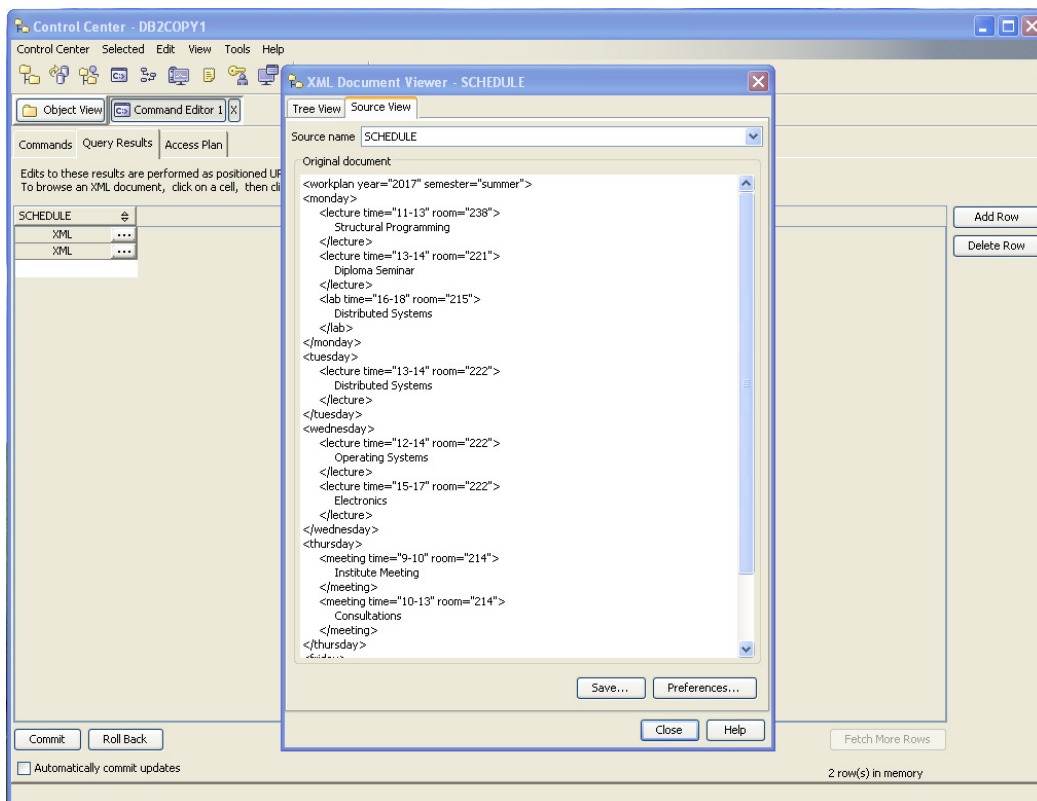
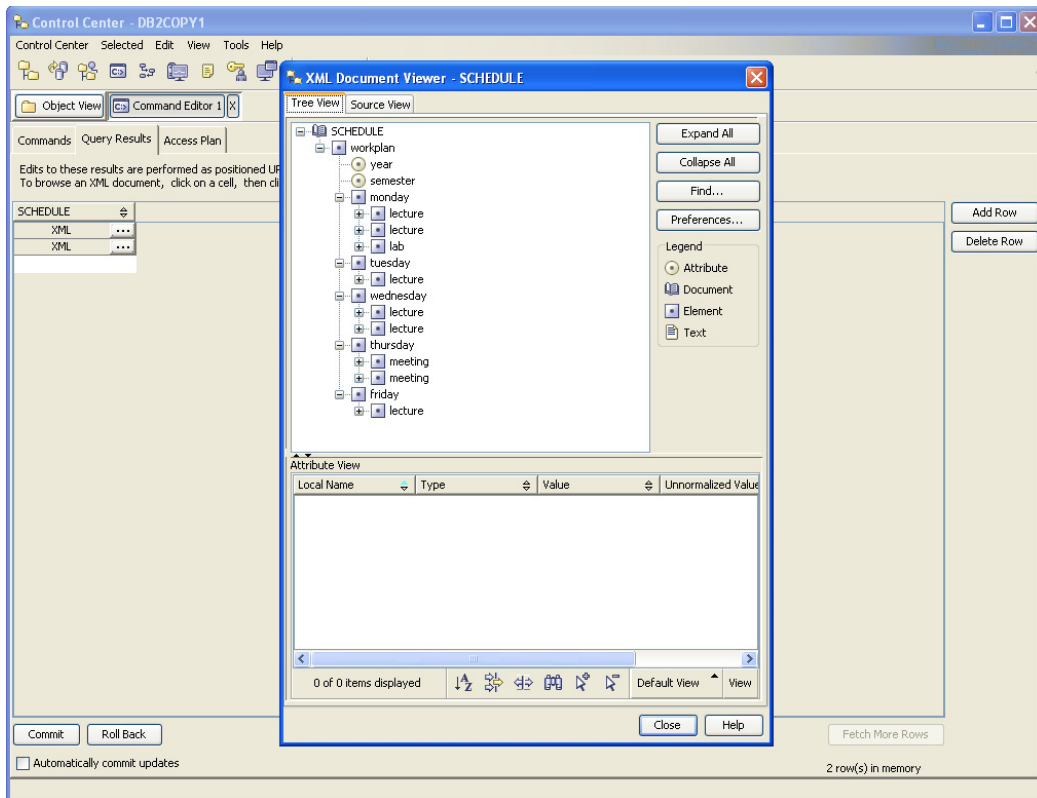
Control Center - DB2COPY1
Control Center Selected Edit View Tools Help
Object View Command Editor 1 X
Commands Query Results Access Plan
Target TEST Add...
INSERT INTO gut.teachers
(fname, lname, title, phone, room, schedule)
VALUES ('Jacek', 'Ruminski', 'PhD Eng', '583472678', '106',
XMLPARSE( DOCUMENT
'<workplan year="2017" semester="summer">
  <monday>
    <lecture time="11-13" room="238">Structural Programming</lecture>
    <lecture time="13-14" room="221">Diploma Seminar</lecture>
    <lab time="16-18" room="215">Distributed Systems</lab>
  </monday>
  <tuesday>
    <lecture time="13-14" room="222">Distributed Systems</lecture>
  </tuesday>
  <wednesday>
    <meeting time="9-10" room="214">Instructor Meeting</meeting>
    <meeting time="10-13" room="214">Consultations</meeting>
  </wednesday>
  <thursday>
    <meeting time="10-13" room="214">Consultations</meeting>
  </thursday>
  <friday>
    <lecture time="11-15" room="321">Data Bases</lecture>
  </friday>
</workplan>' STRIP WHITESPACE)
);
-----
INSERT INTO gut.teachers (fname, lname, title, phone, room, schedule) VALUES ('Jacek', 'Ruminski', 'PhD Eng', '583472678', '106', XMLPARS
DB20000I The SQL command completed successfully.
Statement termination character ;

```

Po wprowadzeniu dwóch wierszy sprawdzimy, czy dane rzeczywiście zapisane są w bazie. W tym celu wybierzmy pojedynczą kolumnę (XML) z naszej tabeli:

```
select schedule from gut.teachers;
```

Wynik możemy zobaczyć w formie drzewa jak i pliku tekstowego XML.

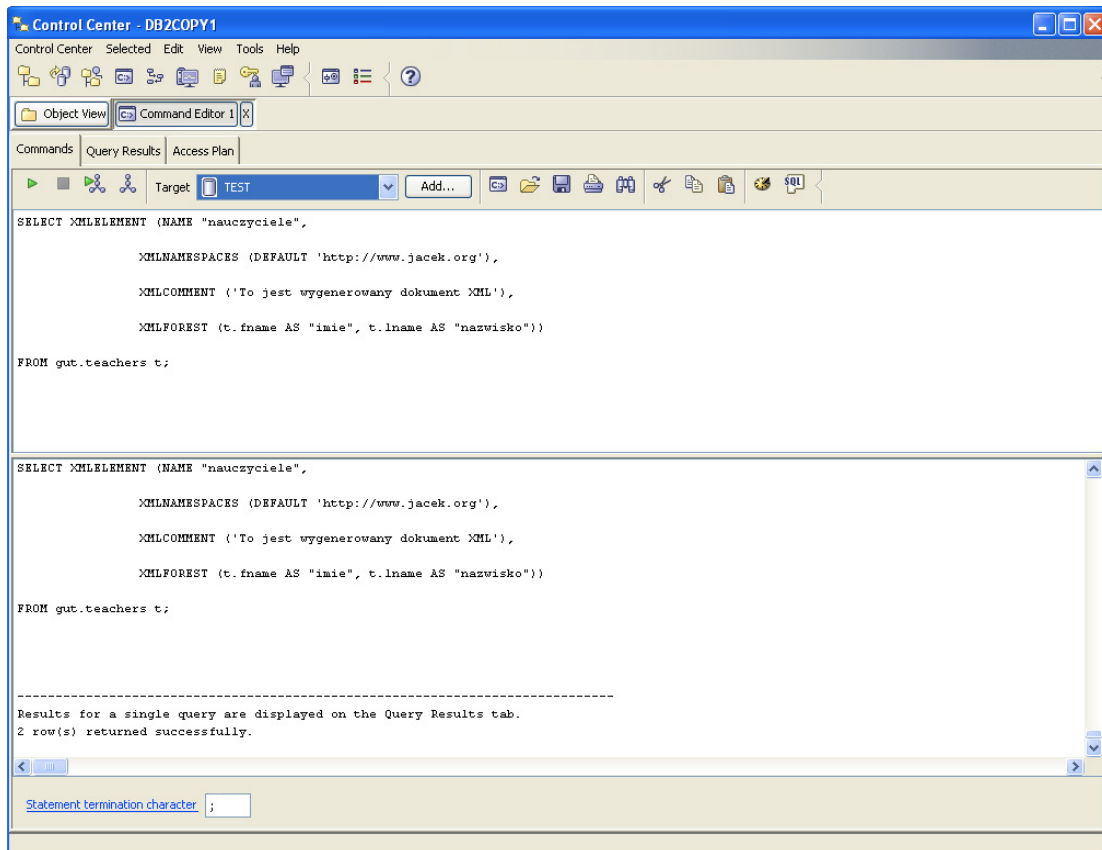


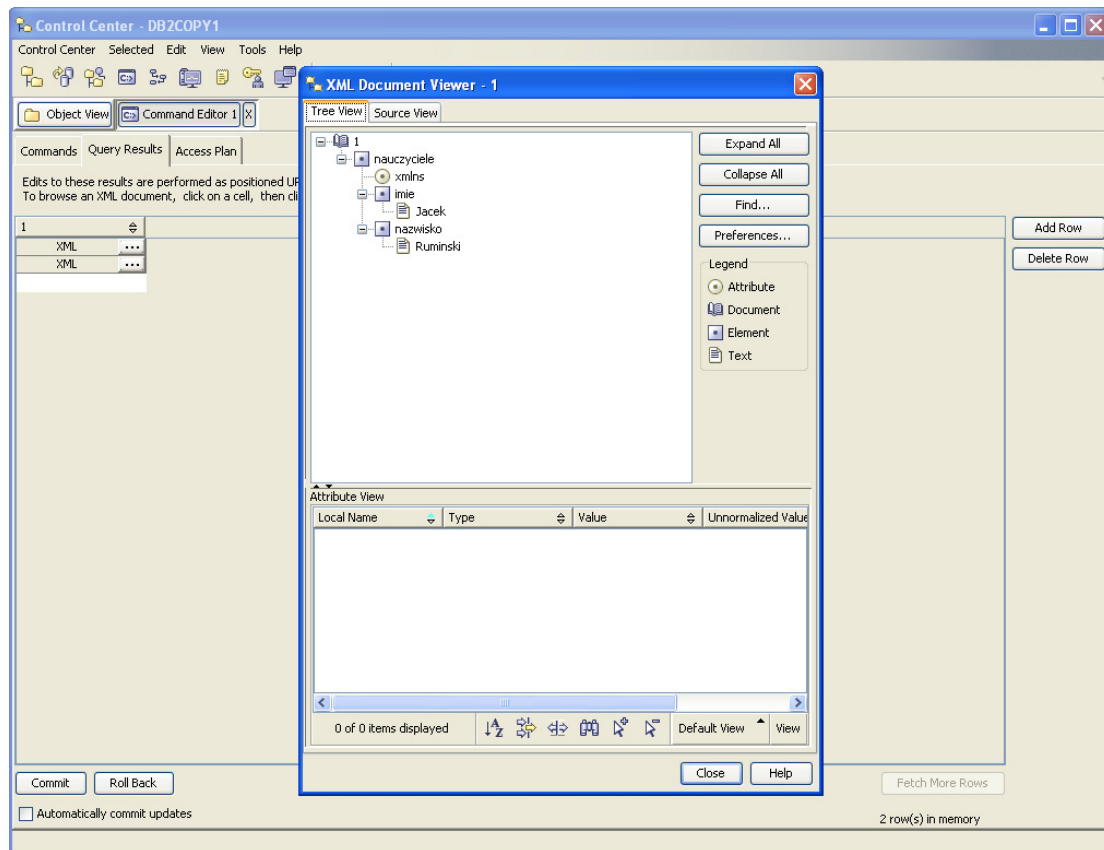
Po wprowadzeniu danych do bazy danych możemy przeprowadzić szereg operacji na danych XML. W tym celu możemy wykorzystać tradycyjne zapytania SQL (relacyjne) w połączeniu ze specjalnymi funkcjami SQL/XML.

Najpierw zapoznajmy się z możliwością wygenerowania dokumentu XML ze zbioru danych, wierszy relacyjnych (czyli atrybutów o typach danych innych niż XML). W tym celu zastosujemy funkcje XMLELEMENT, XMLFOREST, XMLCOMMENT:

```
SELECT XMLELEMENT (  
    NAME "nauczyciele",  
    XMLNAMESPACES (DEFAULT 'http://www.jacek.org'),  
    XMLCOMMENT ('To jest wygenerowany dokument XML'),  
    XMLFOREST (t.fname AS "imie", t.lname AS "nazwisko")  
) FROM gut.teachers t;
```

Wynik pokazano poniżej.



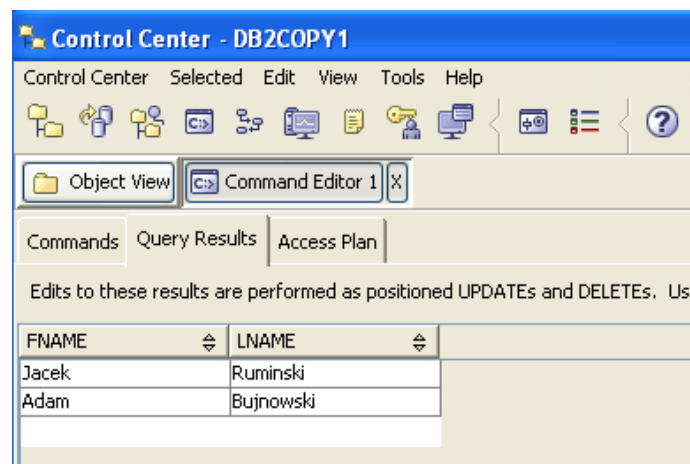
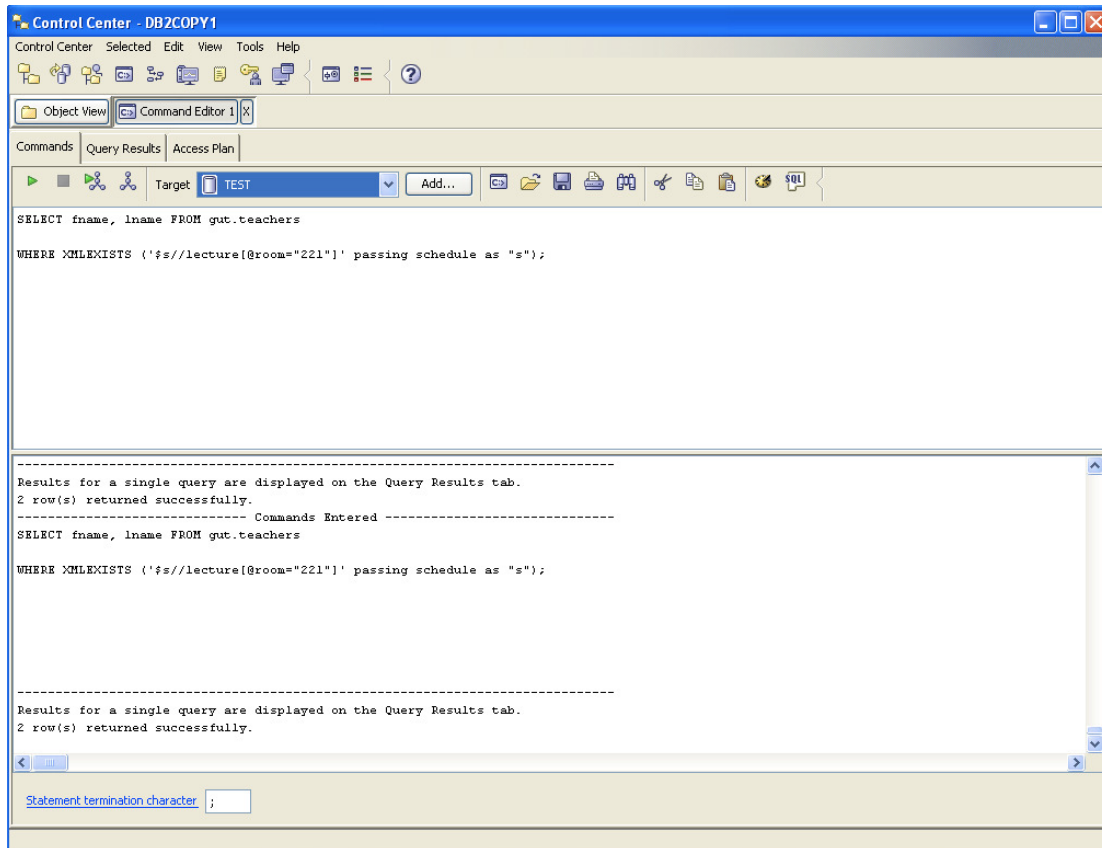


Wyżej pokazana operacja jest niezwykle istotna w systemach wymiany danych wykorzystujących wspólny interfejs (format) bazujący na XML. W dziedzinie informatyki w medycynie jest to niezwykle popularna metoda (HL7 CDA, ASTM CCR, Google Health, NFZ, itp.). Dane mogą być przechowywane w dowolnej strukturze, a dopiero potrzeba wymiany danych wymusza sprowadzenie ich do wspólnego (znormalizowanego) formatu.

Kolejny przykład przetwarzania danych wykorzystuje funkcję/operator XMLEXISTS, za pomocą którego można zdefiniować warunek w odniesieniu do danych XML:

```
SELECT fname, lname FROM gut.teachers WHERE XMLEXISTS ('$s//lecture[@room="221"]' passing schedule as "s");
```

(„Wybierz imię i nazwisko nauczyciela, który ma wykłady w sali 221”)



Kluczową jednak metodą przetwarzania danych XML jest zastosowanie języka XQuery. Przykładowo dla IBM DB2 składnia polecenia wyszukującego wykłady (lecture) w sali 221 ma postać:

*xquery*

```
<room id='221'>
```

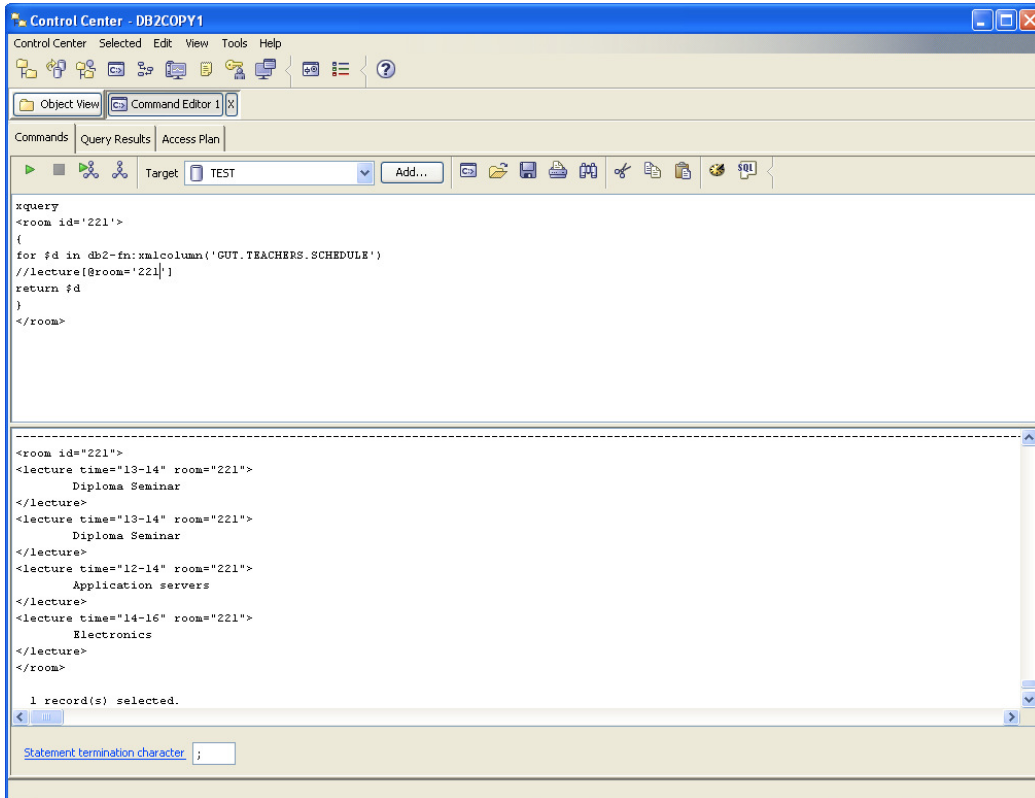
```
{
```

```
for $d in db2-fn:xmlcolumn('GUT.TEACHERS.SCHEDULE') //lecture[@room='221']
```

```
return $d
```

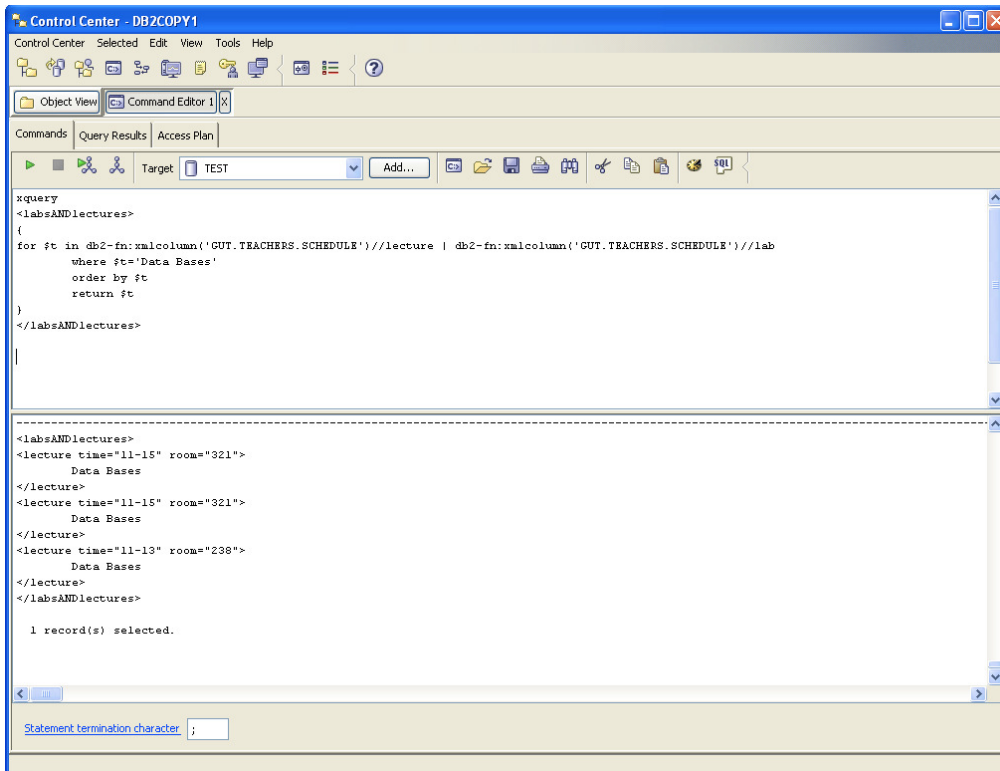
```
}
```

```
</room>
```



Składnia polecenia wyszukiującego wykłady (lecture) lub laboratoria (lab) z przedmiotu „Data Bases” ma postać:

```
xquery
<labsANDlectures>
{
for $t in db2-fn:xmlcolumn('GUT.TEACHERS.SCHEDULE')//lecture | db2-
fn:xmlcolumn('GUT.TEACHERS.SCHEDULE')//lab
where $t='Data Bases'
order by $t
return $t
}
</labsANDlectures>
```



Należy wykonać wszystkie omówione powyżej polecenia SQL/XQuery.

#### Ad. 5. Przetwarzanie HL7 CDA w bazach XML (35 min.)

Na podstawie wiedzy i umiejętności zdobytych w poprzednim punkcie należy:

1. utworzyć tabelę danych zawierających podstawowe dane osobowe pacjenta (standardowe atrybuty/kolumny relacyjne) oraz informacje o alergiach pacjenta (dokument XML o formacie bazującym na HL7 CDA)
2. dodać co najmniej dwa różne wiersze danych
3. przeprowadzić operacje przetwarzania danych, analogiczne do tych, ukazanych w punkcie 4. Polecenia i wyniki zapisać do dokumentu elektronicznego, stanowiącym sprawozdanie z ćwiczenia.

W celu opisu alergii posłużymy się przykładowym dokumentem XML (LOINC – system klasyfikacji/kodowania obserwacji laboratoryjnych - Logical Observation Identifiers Names and Codes; SNOMED – system hierarchicznej klasyfikacji/kodowania procedur leczniczych - Systematized Nomenclature of Medicine):

```
<?xml version="1.0"?>
```



```

<ClinicalDocument>
  <section>
    <caption>
      <caption_cd V="48765-2" S="LOINC"/>
      Allergies and Adverse Reactions
    </caption>
    <list>
      <item>
        <content ID="A1">Penicillin - Hives</content>
      </item>
      <item>
        <content>Aspirin - Wheezing</content>
      </item>
      <item>
        <content>Codeine - Itching and nausea</content>
      </item>
    </list>
    <coded_entry>
      <coded_entry.value ORIGTXT="A1" V="DF-10074" S="SNOMED" DN="Allergy to Penicillin"/>
    </coded_entry>
  </section>
</ClinicalDocument>

```

Warto zauważyć, że wartość atrybutu *ID* (A1) wykorzystana jest w odwzorowaniu na wartość kodową w systemie SNOMED (DF-10074).

#### UWAGA!

Tworząc własny przykład dokumentu XML należy wyszukać w Internecie (np. <http://bioportal.bioontology.org/visualize/44777/?conceptid=294377000>) kodu SNOMED innej formy alergii i stosowną alergię (wraz z kodem) zapisać w dokumencie XML.

### Ad. 6. Wykonywanie podstawowych operacji na kartach elektronicznych (30 min.)

W tej części ćwiczenia wykorzystany zostanie zestaw eHealth Demo Kit, stanowiący element systemu pilotażowego wprowadzania systemy kart elektronicznych (lekarza i pacjenta!) w Niemczech.

Zestaw składa się z dwóch czytników kart, oprogramowania oraz dwóch rodzajów kart:

- pacjenta (PDC - Patient Data Card)



- lekarza (HPC - Healthcare Professional Card)



W celu zapoznania się z ideą wykorzystania dwóch kart pamięci wykonamy kilka ćwiczeń demonstracyjnych. Najpierw należy:

- podłączyć dwa czytniki kart do portów USB komputera,
- uruchomić serwera aplikacji webowych – Apache Tomcat (Start...).

W pierwszej kolejności zapoznamy się z możliwością podstawowych operacji na karcie (symulacja). W tym celu należy do czytnika włożyć kartę pacjenta (PDC), a następnie uruchomić program: eHealthDemo->eHealth card initialization.

Następnie należy wcisnąć przycisk „Initialize Card”, który wywoła akcję wpisania do formularza przykładowych danych pacjenta. Następnie należy wybrać zapisu danych na karcie „Write card” (w razie potrzeby potwierdzić obsługę modułów programowych w przeglądarce).

**Patient Data Card Initialization**

Card reader:

PIN:

Card type:

First name:

Last name:

Date of birth:

1/4 Flatrate:

Insurance:

Insurance no.:

Personal no.:

Expiration date:

Street:

Postal code:

City:

Country:

Sex:  male  female

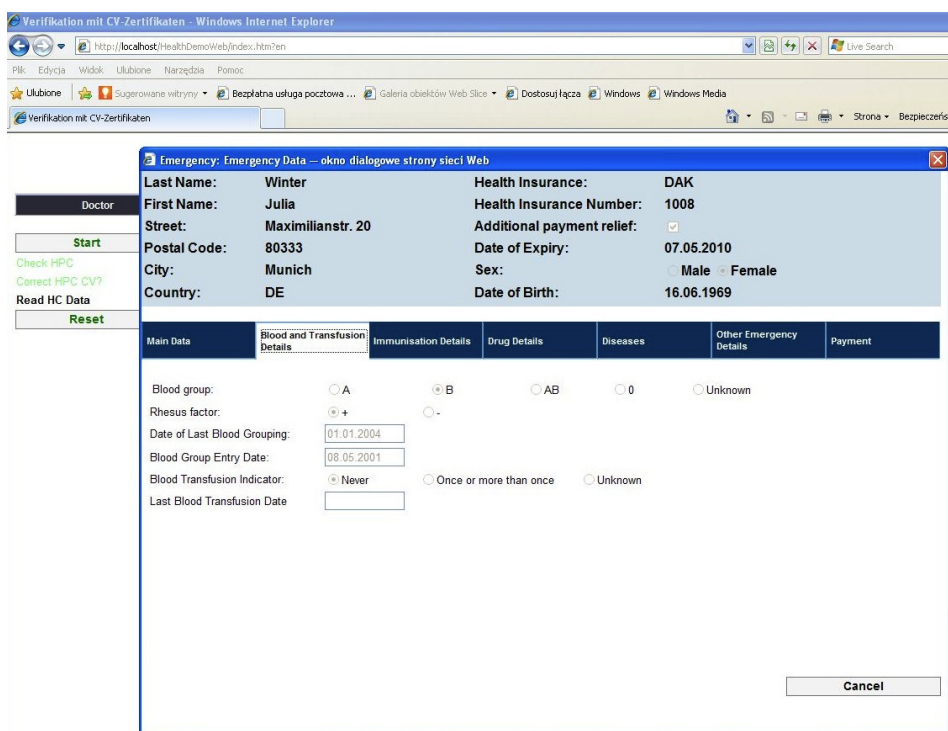
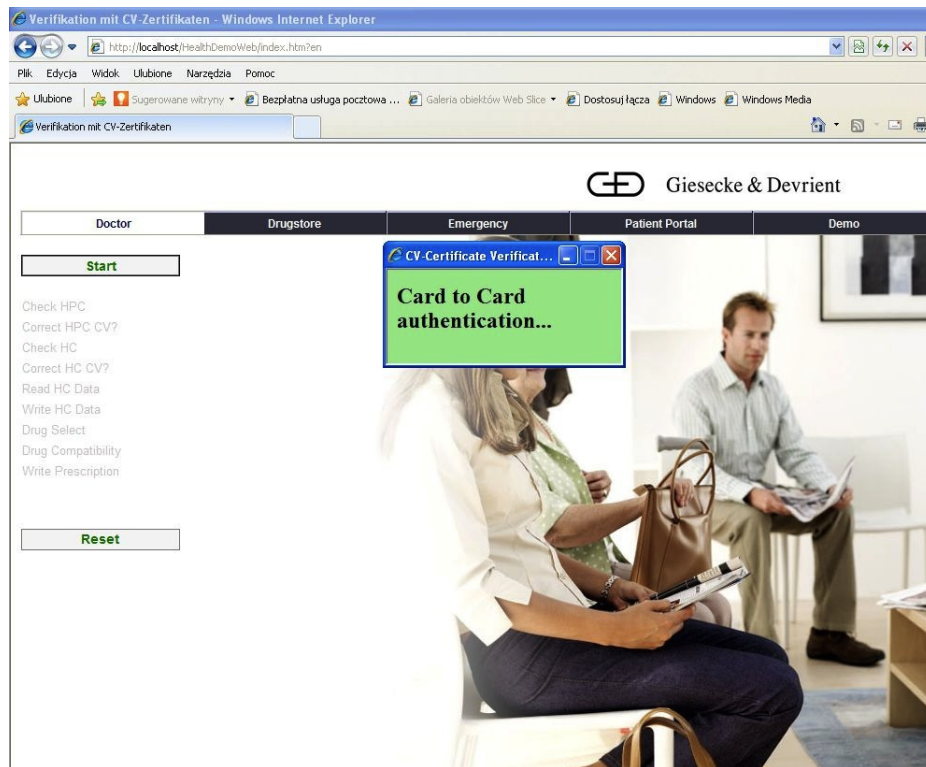
Prescription fee:

Rozważmy teraz przykład, kiedy do szpitala trafia nieprzytomny pacjent posiadający swoją kartę. Wówczas w oddziale ratunkowym lekarz odczytuje dane pacjenta z karty działając w trybie „emergency” uwierzytelniając się swoją kartą lekarza. Dostęp do danych pacjenta jest w trybie tylko do odczytu. Oczywiście operacja jest możliwa tylko jeśli zweryfikowana będzie para kart.

W celu demonstracji działania obu kart należy:

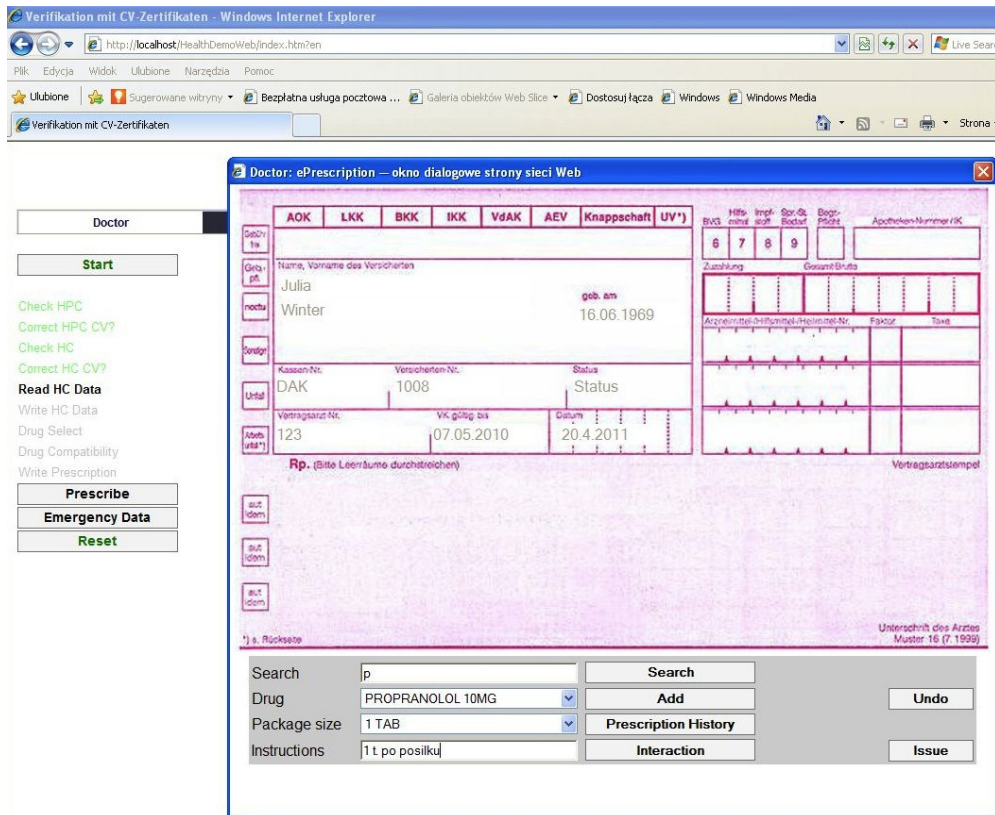
- włożyć tylko kartę pacjenta do czytnika kart,
- uruchomić oprogramowanie eHealthDemo (English),
- wybrać tryb „emergency”,
- wybrać polecenie „Start”.

W wyniku tych działań nie uzyskamy dostępu do danych pacjenta ponieważ NIE UDAŁA się weryfikacja dwóch kart. Musimy włożyć kartę lekarza i ponowić operację (Start). Pojawi się wówczas komunikat „Card to card verification” i jeśli zakończy się sukcesem, otrzymamy dostęp do danych pacjenta. Pokazano to na poniższych rysunkach.



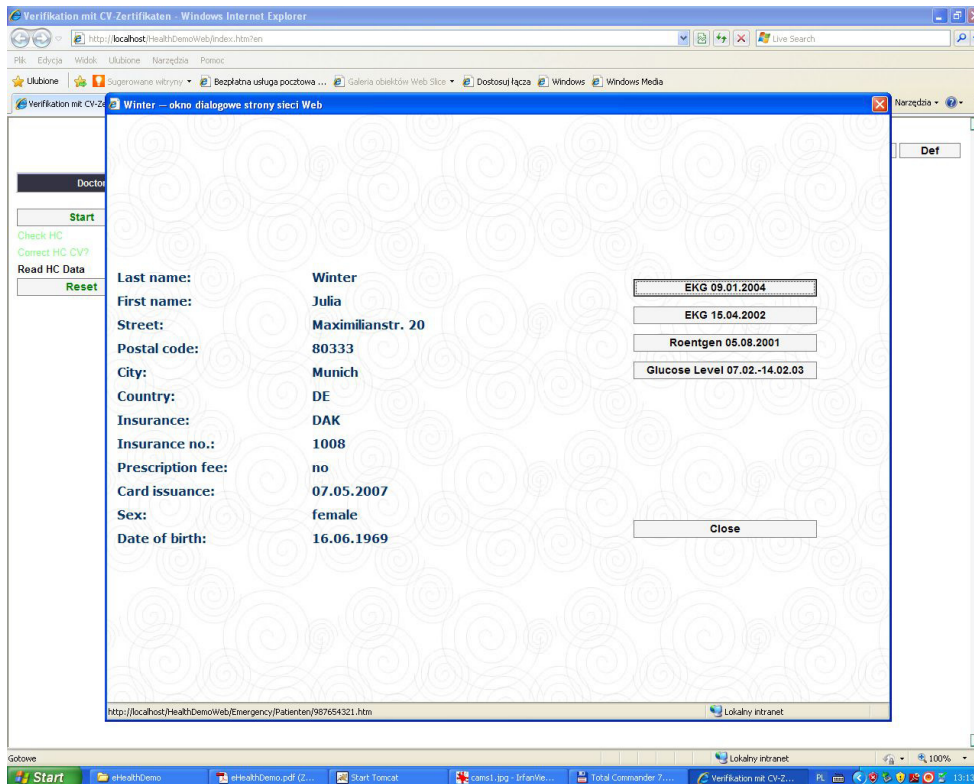
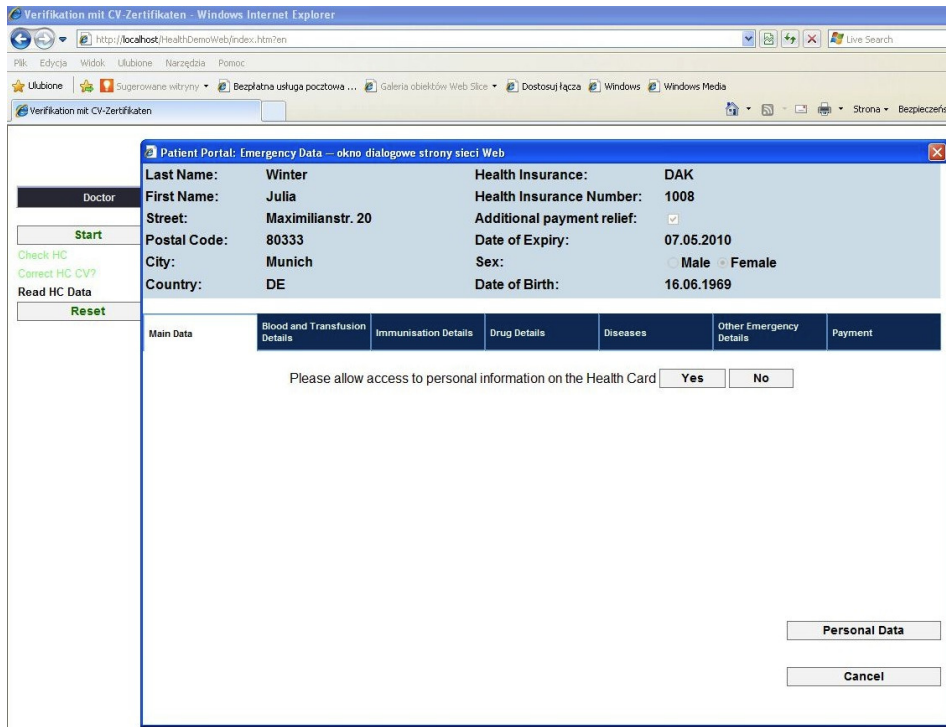
Kolejna demonstracja ilustruje możliwość wystawienia elektronicznej recepty przez lekarza. W tym celu wkładamy obie karty do czytników i uruchamiamy oprogramowanie eHealthDemo. Następnie wybieramy opcję „Doctor” oraz polecenie „Start”. Pojawi się wówczas dostęp do pozycji menu „Prescribe”, po wybraniu której zobaczymy formularz recepty z dodatkowymi polami umożliwiającymi wybór zapisywanego leku. W tym celu w polu „Search” wpisujemy np. literę „p” i wciskamy przycisk rozpoczynający akcję przeszukiwania katalogu leków. Z listy wybieramy „propranolol”, „Package size 1

tab”, a polu „Instructions” wpisujemy krótką informację o dawkowaniu. Następnie wystawiamy receptę (zapisujemy na kartę pacjenta) poprzez wciśnięcie przycisku „Issue”. Operację zilustrowano na poniższym rysunku.



Ostatnią częścią demonstracji będzie pokazanie możliwości odczytu medycznych danych pacjenta. W tym celu NIE WKŁADAMY ŻADNEJ karty, a następnie uruchamiamy program eHealthDemo. Wybieramy opcję „Patient Portal”, po czym uruchamiamy opcję „Start”. Pojawi się komunikat o błędzie – nie ma bowiem karty pacjenta. Wkładamy teraz kartę pacjenta i ponownie wciskamy przycisk „Start”. Możemy mieć teraz dostęp do danych medycznych. Wciskamy przycisk „Personal Data” (bez wciskania YES/NO w części Please allow access to...). Zilustrowano to na kolejnych rysunkach.





Wybierając przycisk EKG otrzymamy (oczywiście tylko jako ilustrację) zapis przebiegu EKG wykonany w dniu X.

