

<b>Imię i nazwisko:</b>	
<b>Numer indeksu:</b>	

<b>Zad:</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7 st.d.</b>	<b>8 st.d.</b>	<b>Suma</b>	<b>Ocena</b>
<b>Punkty</b>										
<b>Max</b>	<b>5</b>	<b>5</b>	<b>4</b>	<b>6</b>	<b>6</b>	<b>5</b>	<b>5</b>	<b>7</b>	<b>43/31</b>	

**Egzamin z przedmiotu**  
**Przetwarzanie Danych Semistrukturalnych**  
(SUM i Studia dzienne)  
Grupa 4  
2007-01-19

Ocena	St. dzienne		SUM	
	Od	Do	Od	Do
bdb	39,5	43	28	31
db plus	34,5	39	25	27,5
db	30,5	34	22	24,5
dst plus	26,5	30	19	21,5
dst	22	26	16	18,5
ndst	0	21,5	0	15,5

**Zadanie 1 (5 pkt)**

W relacyjnej bazie danych w schemacie SCOTT znajduje się tabela PROGNOZY. Polecenie ją tworzące oraz DTD zgodne z mapowaniem SQL/XML bazy danych ze schematem SCOTT przedstawiono poniżej. Wszystkie elementy, które w DTD nie posiadają definicji to elementy proste (#PCDATA).

```
create table PROGNOZY (
  DZIEIN      DATE NOT NULL,
  MIEJSCE     VARCHAR2(20) NOT NULL,
  POGODA      CHAR(6) NOT NULL
  CHECK(POGODA IN ('DESZCZ','SŁONCE','CHMURY','SNIEGI'))
  CONSTRAINT PRO_PK PRIMARY KEY (DZIEIN, MIEJSCE) );
<!ELEMENT DB          (SCOTT)>
<!ELEMENT SCOTT      (PUDELKA)>
<!ELEMENT PROGNOZY (ROW*)>
<!ELEMENT ROW (DZIEIN, MIEJSCE,
               POGODA)>
```

Zdefiniuj poniżej dokument XML Schema, który będzie zgodny z przedstawionym DTD. Schemat ma uwzględniać wszelkie reguły obowiązujące w schemacie relacyjnym. W szczególności odwzoruj typy, ograniczenia integralnościowe i jeśli to możliwe ich nazwy. Wykorzystaj poniższe informacje i przykłady.

**Ograniczenie Opis**

enumeration Definiuje kolejne akceptowalne wartości elementu lub atrybutu  
length Określa dokładną dozwoloną liczbę znaków lub elementów na liście. Musi być równe lub większe od zera  
maxInclusive Definiuje górną granicę dla wartości numerycznych (wartość musi być mniejsza lub równa)  
maxLength Określa maksymalną dozwoloną liczbę znaków lub elementów na liście. Musi być równe lub większe od zera  
minInclusive Definiuje dolną granicę dla wartości numerycznych (wartość musi być większa lub równa)  
minLength Określa minimalną dozwoloną liczbę znaków lub elementów na liście. Musi być równe lub większe od zera  
totalDigits Definiuje dokładną ilość liczb, musi być większa od zera

```
<xs:key name="regKey">
  <xs:selector xpath="//vehicle"/>
  <xs:field xpath="@state"/>
  <xs:field xpath="@plateNumber"/>
</xs:key>
<xs:keyref name="carRef" refer="regKey">
  <xs:selector xpath="//car"/>
  <xs:field xpath="@regState"/>
  <xs:field xpath="@regPlate"/>
</xs:keyref>
```

**Zadanie 2 (6 pkt)**

W katalogu istnieją trzy pliki XML Schema przedstawione poniżej. Zbuduj dokument XML będący instancją poniższych schematów i wykorzystujący definicje zawarte we wszystkich trzech schematach.

dni\_ladne.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="ladne"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="DNI_LADNE">
    <xs:complexType>
      <xs:choice>
        <xs:element name="DZIEIN" type="xs:string" />
        <xs:element name="MIEJSCE" type="xs:string" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

prognoza.xsd:
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="prognoza"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="PROGNOZA">
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace="##other"/>
        <xs:any namespace="##any"/>
      </xs:sequence>
      <xs:anyAttribute/>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

```

dni_brzydkie.xsd:
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="brzydkie"
  elementFormDefault="unqualified" attributeFormDefault="qualified">
  <xs:element name="DNI_BRZYDKIE">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="DZIEN" type="xs:string"/>
        <xs:element name="MIEJSCE" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="POGODA" type="xs:string"/>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

## ROZWIĄZANIE:

```

<?xml version="1.0" encoding="UTF-8"?>
<. . . . . xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
. . . . .
. . . . . >

```

### Zadanie 3 (4 pkt) (Studia dzienne)

Dla dokumentów XML zgodnych przedstawionym poniżej DTD przedstaw schemat relacji pozwalający na ich przechowywanie. Zastosuj podejście SHARED. Rozrysuj odpowiedni graf i schematy relacji. Wszystkie elementy, które w DTD nie posiadają definicji to elementy proste (#PCDATA).

```

<!ELEMENT PROGNOZY      (DNI_LADNE, DNI_BRZYDKIE)
<!ELEMENT DNI_LADNE    (DZIEN, MIEJSCE, DNI_LADNE?)
<!ELEMENT DNI_BRZYDKIE (DZIEN, MIEJSCE, DNI_BRZYDKIE?)>

```

### Zadanie 3 (4 pkt) (SUM)

Zakładając, że w bazie danych SQL Server istnieje schemat zgodny z wcześniej przedstawionym schematem SCOTT'a napisz polecenie SQL, którego wynikiem będzie dokument XML zgodny z poniższym DTD. Wszystkie elementy, które w DTD nie posiadają definicji to elementy proste (#PCDATA).

```

<!ELEMENT PROGNOZY      (DNI_LADNE*, DNI_BRZYDKIE*)
<!ELEMENT DNI_LADNE    (DZIEN, MIEJSCE)
<!ELEMENT DNI_BRZYDKIE (DZIEN, MIEJSCE)>

```

Do elementów DNI\_LADNE wstaw elementy powstałe z krotek, które w atrybucie POGODA mają wartość 'SLONCE'. Elementy utworzone z pozostałych krotek umieść w elemencie DNI\_BRZYDKIE.

## Zadanie 4 (6 pkt)

Napisz zapytanie w języku XML-QL (Studia dzienne) lub XQuery (SUM), które na podstawie dokumentu XML zgodnego z DTD **prognozy.dtd** (przedstawionym poniżej)

### prognozy.dtd:

```
<!ELEMENT PROGNOZY      (DNI_LADNE, DNI_BRZYDKIE)
<!ELEMENT DNI_LADNE     (DZIEEN, MIEJSCE, POGODA, DNI_LADNE?)
<!ELEMENT DNI_BRZYDKIE (DZIEEN, MIEJSCE, POGODA, DNI_BRZYDKIE?)>
```

stwórz dokument, w którym pogrupowane zostaną dni o takiej samej pogodzie. Wynik ma być zgodny z DTD **pogoda.dtd** (przedstawionym poniżej)

### pogoda.dtd:

```
<!ELEMENT PROGNOZY      (POGODA*)
<!ELEMENT POGODA        (NAZWA, DZIEEN*)
<!ELEMENT DZIEEN        (DATA, MIEJSCE)
```

Wszystkie elementy, które w DTD nie posiadają definicji to elementy proste (#PCDATA).

Przykładowo:

Dokument XML:

```
<PROGNOZY>
  <DNI_LADNE>
    <DZIEEN>01.01.2005</DZIEEN>
    <MIEJSCE>POZNAŃ</MIEJSCE>
    <POGODA>SLONCE</POGODA>
  </DNI_LADNE>
  <DNI_LADNE>
    <DZIEEN>01.02.2005</DZIEEN>
    <MIEJSCE>ŚREM</MIEJSCE>
    <POGODA>SLONCE</POGODA>
  </DNI_LADNE>
</DNI_LADNE>
  <DNI_BRZYDKIE>
    <DZIEEN>01.07.2005</DZIEEN>
    <MIEJSCE>POZNAŃ</MIEJSCE>
    <POGODA>DESZCZ</POGODA>
  </DNI_BRZYDKIE>
  <DNI_BRZYDKIE>
    <DZIEEN>02.02.2005</DZIEEN>
    <MIEJSCE>ŚREM</MIEJSCE>
    <POGODA>SNIEGI</POGODA>
  </DNI_BRZYDKIE>
</DNI_BRZYDKIE>
</PROGNOZY>
```

Wynik zapytania:

```
<PROGNOZY>
  <POGODA>
    <NAZWA>SLONCE</NAZWA>
    <DZIEEN>
      <DATA>01.01.2005</DATA>
      <MIEJSCE>POZNAŃ</MIEJSCE>
    </DZIEEN>
    <DZIEEN>
      <DATA>01.02.2005</DATA>
      <MIEJSCE>ŚREM</MIEJSCE>
    </DZIEEN>
  </POGODA>
  <POGODA>
    <NAZWA>DESZCZ</NAZWA>
    <DZIEEN>
      <DATA>01.07.2005</DATA>
      <MIEJSCE>POZNAŃ</MIEJSCE>
    </DZIEEN>
  </POGODA>
  <POGODA>
    <NAZWA>SNIEGI</NAZWA>
    <DZIEEN>
      <DATA>02.02.2005</DATA>
      <MIEJSCE>ŚREM</MIEJSCE>
    </DZIEEN>
  </POGODA>
</PROGNOZY>
```

## Zadanie 5 (6 pkt)

Napisz zapytanie w języku X-Query, które na podstawie dokumentu XML zgodnego z dokumentem  **pogoda.dtd** (przedstawionym wcześniej) utworzy dokument XML będący listą miejsc wraz z statystyką dotyczącą pogody. Wszystkie elementy, które w DTD nie posiadają definicji to elementy proste (#PCDATA).

```
<!ELEMENT LISTA_MIEJSC (MIEJSCE*)>
<!ELEMENT MIEJSCE (NAZWA_MIEJSCA, POGODA*)>
<!ELEMENT POGODA (TYP, LICZBA_DNI)>
```

Przykładowo:

Dokument XML:

```
<PROGNOZY>
  <POGODA>
    <NAZWA>SLONCE</NAZWA>
    <DZIEN>
      <DATA>01.01.2005</DATA>
      <MIEJSCE>POZNAŃ</MIEJSCE>
    </DZIEN>
    <DZIEN>
      <DATA>01.02.2005</DATA>
      <MIEJSCE>ŚREM</MIEJSCE>
    </DZIEN>
    <DZIEN>
      <DATA>02.02.2005</DATA>
      <MIEJSCE>ŚREM</MIEJSCE>
    </DZIEN>
  </POGODA>
</PROGNOZY>
```

Wynik zapytania:

```
<LISTA_MIEJSC>
  <MIEJSCE>
    <NAZWA>ŚREM</NAZWA>
    <POGODA>
      <TYP>SLONCE</TYP>
      <LICZBA_DNI>2</LICZBA_DNI>
    </POGODA>
  </MIEJSCE>
  <MIEJSCE>
    <NAZWA>POZNAŃ</NAZWA>
    <POGODA>
      <TYP>SLONCE</TYP>
      <LICZBA_DNI>1</LICZBA_DNI>
    </POGODA>
  </MIEJSCE>
  <MIEJSCE>
    <NAZWA>DESZCZ</NAZWA>
    <POGODA>
      <TYP>DESZCZ</TYP>
      <LICZBA_DNI>1</LICZBA_DNI>
    </POGODA>
  </MIEJSCE>
</LISTA_MIEJSC>
```

## Zadanie 6 (2+3 pkt) (Studia dzienne)

Na dokumencie  **prognozy.xml** przedstawionym poniżej

**prognozy.xml:**

```
<?xml version="1.0"?>
<POGODA>
  <NAZWA>SLONCE</NAZWA>
  <DZIEN>
    <DATA>01.01.2005</DATA>
    <MIEJSCE>POZNAŃ</MIEJSCE>
  </DZIEN>
  <DZIEN>
    <DATA>01.02.2005</DATA>
    <MIEJSCE>ŚREM</MIEJSCE>
  </DZIEN>
  <DZIEN>
    <DATA>02.02.2005</DATA>
    <MIEJSCE>ŚREM</MIEJSCE>
  </DZIEN>
</POGODA>
```

Napisz polecenia XUpdate, które pozwolą na poniższe zmiany. Wykorzystaj przykłady poleceń dostępnych w ostatnim zadaniu, oraz poniższe informacje.

Element xupdate:modifications może zawierać następujące elementy: insert-before, insert-after, append, update, remove, rename, variable, value-of, if. Elementy insert-before i insert-after mogą zawierać elementy tworzące odpowiedniego rodzaju węzeł, który ma zostać wstawiony: element, attribute, text, processing-instruction, comment.

- Zamień nazwę POGODA na PROGNOZY
- Usuń element NAZWA znajdujący się na poziomie elementu PROGNOZY i wstaw go jako element POGODA do wnętrza każdego z elementów DZIEN

### Zadanie 6 (5 pkt) (SUM)

W tabeli PROGNOZY\_XML posiadającej kolumnę DOC typu XMLType znajdują się dokumenty zgodne z poniższym DTD

```
<!ELEMENT LISTA_MIEJSC (MIEJSCE*)>
<!ELEMENT MIEJSCE (NAZWA_MIEJSCA, POGODA*)>
<!ELEMENT POGODA (TYP, LICZBA_DNI)>
```

Napisz zapytanie, które wyświetli miejsca, w których pogoda o wartości SLONCE trwała przez ponad 10 dni. Wynik ma być zgodny z poniższym DTD:

```
<!ELEMENT POGODNE_MIEJSKA (MIEJSCE*)>
<!ELEMENT MIEJSCE (NAZWA_MIEJSCA, POGODA*)>
<!ELEMENT POGODA (TYP, LICZBA_DNI)>
```

Element MIEJSCE ma zawierać pełną oryginalną jego postać.

Wszystkie elementy, które w DTD nie posiadają definicji to elementy proste (#PCDATA).

### Zadanie 7 (5 pkt) (Studia dzienne)

- a) Co to jest XML:DB API?
- b) W bazach danych dokumentów XML wyróżniamy trzy sposoby modyfikacji dokumentów XML:
- Modyfikacja za pomocą operacji DOM
  - Modyfikacje przy wykorzystaniu XPath, które wskazują węzły, po czym następuje: wstawienie węzła przed lub po, modyfikacja węzła, usunięcie węzła
  - Modyfikacja za pomocą rozszerzeń języka XQuery

Jakiego rodzaju modyfikacją jest XUpdate?

Które z metod dostępne są w Oracle'u i za pomocą czego?

- c) W bazach danych dokumentów XML wyróżniamy dwa typy kolekcji:
- Podobne do katalogów w systemie plików (dokumenty mogą mieć dowolny schemat, kolekcje mogą być zagnieżdżone)
  - Podobne do tabel w systemie relacyjnym (dokumenty są ograniczone do określonego schematu, umożliwiają indeksowanie i optymalizację zapytań)

Które z kolekcji dostępne są w Oracle'u w zapytaniach XQuery i za pomocą jakich funkcji?

- d) Funkcja XMLQuery pozwala w bazie danych Oracle na wykonywanie zapytań XQuery. W ramach jakiego standardu związanego z XMLem ta funkcja występuje?

### Zadanie 8 (7 pkt) (Studia dzienne)

Na dokumencie **prognozy.xml** przedstawionym w zadaniu 6 dwie transakcje usiłują wykonać operacje. W celu weryfikacji kompatybilności tych transakcji zastosuj algorytm Path Lock Propagation. Rozrysuj odpowiednią strukturę i zaznacz na niej blokady założone przez obie transakcje, opisz wyraźnie wartości poszczególnych blokad. Czy są one w konflikcie? Jeśli tak zaznacz blokady, które o tym decydują.

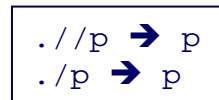
T1:

```
<xu:modifications version="1.0" xmlns:xu="http://www.xmldb.org/xupdate">
  <xupdate:variable name=nazwa select="/POGODA//NAZWA"/>
  <xu:append select="/POGODA/DZIEN[DATA='01.01.2005']">
    <xupdate:value-of select="$nazwa "/>
  </xu:append>
</xu:modifications>
```

T2:

```
<xu:modifications version="1.0" xmlns:xu="http://www.xmldb.org/xupdate">
  <xu:remove select="//NAZWA" />
</xu:modifications>
```

Wnioskowanie:



Propagacja:

blokada na rodzicu	typ dziecka	nazwa dziecka	blokada na dziecku
./p	element	-	./p
t/p	element	t	p
t//p	element	t	./p
*/p	element	-	p
*/p	element	-	./p
@a/p	atrybut	a	p
@*/p	atrybut	-	p
τ/p	tekst	-	p