Model semistrukturalny

- standaryzacja danych z różnych źródeł
- realizacja złożonej struktury zależności, wielokrotne zagnieżdżania
- zobrazowane przez grafy skierowane
- model samoopisujący się
- wielkości i typy atrybutów encji z tej samej grupy mogą się różnić
- wykorzystywany do wymiany i integracji danych z źródeł heterogenicznych
- struktury danych nie muszą być regularne
- łatwość wyszukiwania, bez wiedzy o typach atrybutów
- trudniejsza optymalizacja

XML (ang. Extensible Markup Language – Rozszerzalny Język Znaczników)

- to nie język programowania
- standard rekomendowany oraz specyfikowany przez W3C
- to sposób zapamiętywania danych wraz z ich strukturą w dokumencie tekstowym
 - otwarty, elastyczny, bezpłatny, nadmiarowy
 - niezależny od platformy sprzętowej

XML to rama składniowa do tworzenia języków specyficznych dla wielu zastosowań

XML a model semistrukturalny

różnice:

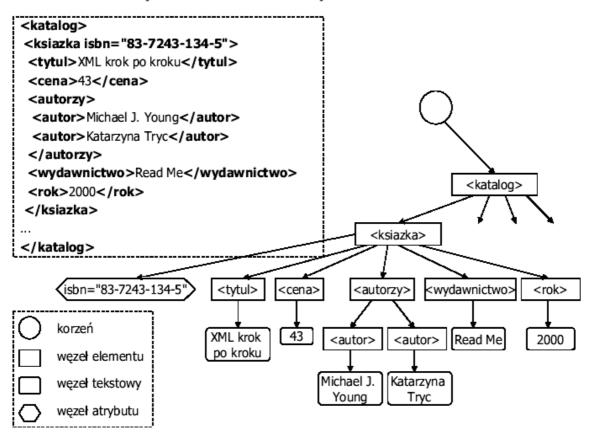
- w modelu danych semistrukturalnych można wyrazić dowolny graf skierowany, w XML tylko drzewo
- w dokumentach XML kolejność występowania elementów w ramach elementu nadrzędnego jest ważna. W modelu semistrukturalnym krawędzie nie są porządkowane
- w dokumencie XML istnieje kilka typów składników struktury, spośród których podstawowe to elementy i atrybuty. W modelu danych semistrukturalnych nie istnieje takie rozróżnienie

Document Object Model (DOM)

• Document Object Model jest standardem modelowania dokumentów XML przy użyciu struktury drzewa – znaczniki XML i ich zawartości są modelowane przez węzły drzewa; zagnieżdżanie znaczników służy za podstawę do konstruowania hierarchii.

- Document Object Model jest wykorzystywany jako forma reprezentacji dokumentów XML w pamięci komputera
- transformacja dokumentu XML do postaci DOM jest realizowana automatycznie przez parser DOM
- standard W3C DOM definiuje zespół klas i interfejsów, pozwalających na dostęp do struktury dokumentów oraz jej modyfikację

Przykład struktury drzewa DOM



Rysunek pozyskano z materiałów II Szkoły PLOUG

Poprawność dokumentu w XML

Dokument dobrze sformułowany (ang. well – formed) zgodność z regułami skład XML

- niepusta zawartość
- wszystkie konstrukcje rozpoczęte muszą być jawnie zakończone
- każda konstrukcja zawarta w innej musi być w niej zawarta w całości

Przeznaczenie: tylko wyświetlanie

Dokument prawidłowy (ang. valid) poprawność składniowa

- jest well formed
- powiązany i zgodny z istniejącym DTD lub XML Schema

Przeznaczenie: wyświetlanie i przetwarzanie

Dokument w XML

- plik tekstowy
 - znaczniki początkowe ujęte w znaki <>
 - końcowe ujęte w </>
- budowa dokumentu prolog, element główny, elementy podrzędne
- znakowanie znaczeniowe
- nie ma predefiniowanych znaczników
- znaczniki odzwierciedlają strukturę dokumentu/danych
- znacznikom nie jest z góry przypisany sposób prezentacji
- do formatowania wizualnego konieczne jest użycie arkusza stylistycznego

Język Xpath (ang. XML Path Language, język scieżek)

To specyfikacja języka służącego do adresowania, odczytywania i przeszukiwania drzew DOM dokumentów XML.

Xpath

- odgrywa podobną rolę w stosunku do drzew DOM, jak język SQL w środowisku relacyjnych baz danych
- stosuje notację przypominającą ścieżki dostępu w systemach plików
- zawiera bibliotekę standardowych funkcji
- pozwala stosować dwa rodzaje zapisu wyrażeń: skrócony, pełny lub mieszany
- jest rekomendowany przez W3C

przykłady wyrażeń:

```
/ katalog / ksiazka / autorzy

// wydawnictwo

// ksiazka [ 1 ] // autor [ 2 ]

// ksiazka [ @isbn = "83-7279-149-X " ]

// ksiazka [ contains ( tytul , " XML ") ]
```

Bazy danych a XML

- bazy danych umożliwiające przechowywanie danych w formacie XML (ang. XML enabled database systems) postrelacyjne, obiektowo-relacyjne systemy zarządzania, które na wejściu i wyjściu akceptują i generują dane w postaci XML.
- bazy danych dokumentów XML natywne bazy danych XML (ang. native XML database systems) zorientowane na przetwarzanie dokumentów XML (podstawowy typ przechowywania)

Standard SQL/XML został zdefiniowany w postaci specyfikacji przygotowanej przez nieformalną grupę SQLX. W pracach nad nimi brali udział przedstawiciele takich firm jak: DataDirect Technologies, Hewlett-Packard, IBM, Microsoft, Oracle, Sybase i innych. SQL/XML to ogólny standard dotyczący zastosowania SQL w ogólnie rozumianym przetwarzaniu dokumentów XML. Jego pierwsze implementacje zostały stworzone przez Oracle, IBM i DataDirect Technologies.

Pod koniec roku 2003 SQL/XML stał się częścią standardu języka SQL.

Zakres standardu SQL/XML definiuje

- mapowanie pomiędzy SQL a XML
- typ danych XML
- funkcje

Typ danych XML

W większości systemów zarządzania bazą danych użytkownicy mogą przechowywać dokumenty XML za pomocą typów VARCHAR, CLOB lub też w postaci dekomponowanej na kolumny typu prostego znajdujące się w jednej lub wielu tabelach. Wszystkie te metody mają swoje wady, najważniejszą jest to, że SZBD " nie wie", że przetwarza dokumenty XML. W związku z tym sposób przetwarzania nie uwzględnia specyfiki struktur oraz metod przetwarzania XML.

SQL/XML wprowadza nowy typ danych mający na celu zwiększenie funkcjonalności i wydajności

przetwarzania dokumentów XML. Typ ten wg standardu można wykorzystywać do definiowania kolumn tabel, zmiennych, oraz parametrów procedur i funkcji.

W przypadku bazy danych Oracle to typ XMLType

Jego obiektowy charakter umożliwia wykorzystywane wartości typu XMLType nie tylko w przypadkach narzucanych przez standard SQL/XML

```
przykłady:
1)
CREATE TABLE Produkty xml
(Kod NUMBER(5),
Opis XMLType);
INSERT INTO Produkty xml VALUES (125,
XMLType ( '< produkt> < nazwa> Komputer </nazwa > < cena > 1975 </cena> </produkt> '));
2)
CREATE OR REPLACE FUNCTION Fun_xml
RETURN XMLType IS
BEGIN
RETURN XMLType('<maly>dokument</maly>');
END;
3)
CREATE TABLE Cv_xml OF XMLTYPE;
SELECT * FROM Cv xml ;
<zbior>
 <cv>
 <dane osobowe>
  <imie> Święty </imie> <nazwisko> Mikołaj </nazwisko>
  <adres> <uli> <hoinkowa 113 </ulica> <miasto> Chmura Śniegowa </miasto> </adres>
  <telefon>nieznany</telefon>
  <data urodzenia>1573</data urodzenia>
  </dane osobowe>
```

Przeszukiwanie dokumentów XML

dokumenty zapisane w bazie danych jako obiekty XMLType mogą być łatwo przeszukiwane z poziomu SQL dzięki specjalnym funkcjom:

- ExistsNode(,)
- Extract(,)
- Extractvalue(,)

Najbardziej ogólna składnia tych funkcji;

```
<funkcja>(< obiekt xmltype >, ' < wyrażenie xpath >')
```

- funkcja **existsNode()** sprawdza, czy wynik ewaluacji podanego wyrażenia Xpath zwraca przynajmniej jeden element. Jeśli tak, wówczas wynikiem jej działania jest wartość 1, w przeciwnym przypadku 0.
- funkcja **extract()** zwraca fragment dokumentu XML wyznaczony przez podane wyrażenie Xpath.
- funkcja extractValue() zwraca zawartość tekstową węzła wyznaczonego przez podane wyrażenie Xpath. Wyrażenie to musi wskazywać na nie więcej niż jeden węzeł, inaczej funkcja nie zadziała.

```
przykłady:
```

```
SELECT Kod, extract (Opis, '/produkt/ nazwa[cena > 1500] ') FROM Produkty_xml;
odp:
```

```
125 <nazwa> Komputer </nazwa>
```

SELECT extract (VALUE (k), '//autor') FROM Ksiazki_xml k; odp: <autor>Stephen Forte</autor> <autor>Thomas Howe</autor> <autor>Kurt Wall</autor> <autor>Paul Cassel</autor> <autor>Craig Eddy</autor> <autor>Jon Price </autor> <autor>Scott Worley</autor> <autor>Fabio Arciniegas</autor> ... <autor>Elliotte Rusty Harold</autor> <autor>Robert Eckstein</autor> SELECT k.existsNode('//ksiazka[cena=31]') FROM Ksiazki_xml k; odp: 1 SELECT extractvalue(value(k),'//ksiazka[cena=31]/tytul') FROM Ksiazki_xml k; odp: Dane w sieci WWW

SELECT extract(value(k),'//ksiazka[cena=31]/tytul') FROM Ksiazki xml k;

odp:

<tytul>Dane w sieci WWW</tytul>