

XML w bazach danych.

XML a relacyjne bazy danych

- Korzyści:
 - integracja aplikacji, wymiana danych,
 - łatwe transformacje danych,
 - prezentacja danych.
- Problemy:
 - czy i jak przechowywać dokumenty XML w bazie danych?
 - metody dostępu (zadawania zapytań),
 - efektywność.

XML a relacyjne bazy danych

- Przechowywanie XML-a w relacyjnych bazach danych:
 - elementy dokumentu XML jako pola tabeli bazodanowej (dokument „rozłożony na czynniki pierwsze”),
 - dokument XML w całości przechowywany w polu bazy danych.
- Sposoby wspierania XML-a przez systemy zarządzania bazami danych:
 - generowanie XML-a na podstawie zawartości bazy danych,
 - wypełnianie zawartości bazy na podstawie zawartości dokumentu XML,
 - specjalne indeksowanie pól zawierających XML,
 - wbudowane parsery XML i procesory XSLT,
 - integracja z serwerem WWW.

Klasyfikacja wsparcia dla XML-a w bazach danych

- Wg wewnętrznej reprezentacji danych?
 - *XML-enabled* – przechowująca dokumenty w postaci zdekomponowanej,
 - *native XML* – przechowująca dokumenty w „naturalnej” postaci.
- Lepiej wg dostępnych operacji i zastosowania:
 - (relacyjna) baza danych:
 - konfiguracja struktur danych przy pomocy tabel i relacji,
 - umożliwia eksport i import danych w postaci dokumentów XML,
 - struktura dokumentów XML pochodną relacyjnych struktur danych,
 - zastosowanie: integracja, wymiana danych;
 - XML-owa baza danych:
 - przechowuje dokumenty XML,
 - konfiguracja struktur danych przy pomocy DTD/XML Schema,
 - Indeksowanie elementów, atrybutów, wyrażeń XPath,
 - wyszukiwanie z użyciem XQuery,
 - zastosowanie: przechowywanie i przetwarzanie dokumentów strukturalnych.

XML w Oracle 10g

- Parsery XML dostarczane przez Oracle:
 - pozwalają na wykorzystanie XML-a we własnych aplikacjach korzystających z bazy,
 - dostępne dla PL-SQL-a, Javy i C++.
- XML-SQL Utility:
 - generowanie XML-a bezpośrednio z bazy przy pomocy specjalnych zapytań,
 - wypełnianie bazy na podstawie zawartości dokumentu XML.

Oracle XML-SQL Utility

- Przykład zapytania:

```
SELECT xmlgen.getXML('select * from emp') FROM dual;
```
- Domyślne użycie standardowej struktury, np.

```
<rowset>
  <row id="1">
    <empno>10</empno>
    <name>Scott Tiger</name>
    <title>specialist</title>
  </row>
  ...
</rowset>
```
- Możliwość przededefiniowania elementów.

XML w Oracle 10g (c.d.)

- XML-SQL Servlet:
 - ułatwia budowanie aplikacji internetowych opartych na Oracle'u,
 - umożliwia przesyłanie XML-a z i do bazy za pośrednictwem protokołu HTTP,
 - możliwość korzystania ze skryptów XSQL Pages, np:

```
<xsql:query connection="demo"
  xmlns:xsql="urn:oracle-xsql">
  SELECT * FROM emp
</xsql:query>
```
- XMLType – specjalny typ danych:
 - używany do deklarowania kolumn, tabel, perspektyw, zmiennych,
 - indeksowanie zawartości XML,
 - zapytania XQuery,
 - kontrola poprawności strukturalnej względem XML Schema,
 - przekształcenia XSLT.

XMLType w Oracle 10g

- Specjalne operatory:
 - extract,
 - extractValue,
 - existsNode,
 - transform,
 - updateXML,
 - XMLSequence.
- XPath Rewrite – przekształcanie ścieżek XPath w równoważne konstrukcje SQL na wewnętrznej reprezentacji strukturalnej XMLType.

Przykłady

```
SELECT extract(value(X),
  '/PurchaseOrder/LineItems/LineItem/Description')
FROM XMLTABLE X;
```

zwraca:

```
<Description>The Ruling Class</Description>
<Description>Diabolique</Description>
```

```
SELECT extractValue(value(t), '/Description')
FROM XMLTABLE X,
TABLE (xmlsequence(extract(value(X),
  '/PurchaseOrder/LineItems/LineItem/Description'))) t;
```

zwraca:

```
The Ruling Class
Diabolique
```

Źródło: Oracle 9i XML Database Developer's Guide,
www.leidenuniv.nl/awcourse/oracle/appdev.920/a96620/toc.htm

XML Schema w Oracle 10g

- Rejestrowanie schematów.
 - automatycznie tworzone tabele dla elementów globalnych.
- Przekształcanie schematów:
 - rejestrowanie nowej wersji schematu,
 - przekształcanie XSLT dostosowujące zastane dokumenty do nowej wersji schematu.
- Adnotowanie schematów:
 - kontrola nazw tabel, obiektów i atrybutów SQL generowanych dla schematu,
 - niestandardowe odwzorowanie typów XML Schema na typy SQL,
 - wskazywanie sposobu dekompozycji kolekcji elementów XML:
 - kolumna CLOB,
 - tablica obiektów w kolumnie LOB,
 - zagnieżdżona tabela (Index Organized Nested Table),
 - osobna tabela XMLType.

Podejście alternatywne: Tamino

- Transaction Architecture for the Management of Internet Objects.
- Pierwszy serwer „bazodanowy” przechowujący dane w XML-u.
- Komunikacja:
 - za pośrednictwem protokołu HTTP, bezpośrednio przez URL,
 - moduł X-Node, zapewniający integrację z innymi źródłami danych:
 - ODBC, OLE DB,
 - system plików.
- Platforma dla:
 - aplikacji internetowych typu B2C,
 - elektronicznej wymiany dokumentów:
 - nowość: wsparcie dla XML Signature;
 - systemów zarządzania treścią:
 - nowości: wersjonowanie, scalanie, indeksowanie dokumentów nie-XML.

 www.softwareag.com/tamino
Number one in XML management

Baza danych w Tamino

- Definicja tabeli
- Wiersz tabeli
- Pole tabeli
- Relacja
- Kwerenda
- XML-Schema
- Element (z podelementami) w dokumencie XML.
- Element lub atrybut XML
- Odpowiednie atrybuty w XML-Schema
- Zapytanie w języku XQuery

XML a bazy danych – przegląd

Relacyjne bazy danych
ze wsparciem dla XML-a:

- Oracle 8i / 9i / 10g
- Microsoft SQL Server 2000
- DB2, IBM
- Sybase ASE 12.5

XML-owe bazy danych:

- Tamino, Software AG
- TEXTML Server, InxiaSoft
- dbXML (open source)
- eXist (open source)
- Xindice, Apache Software Foundation (open source)

Gdzie szukać dalej

- Bourret, R., *XML and Databases*
📄 www.rpbourret.com/xml/
- Ogrodniczuk, M., *XML w bazie danych*
📄 Software 2.0, 12/2001, Wydawnictwo Software
- *Research note: The demise of the XML database*
📄 www.it-analysis.com/article.php?articleid=11287
- *Oracle XML DB. An Oracle Technical White Paper*
📄 www.oracle.com/technology/tech/xml/xmldb/Current/TWP.pdf
- *Oracle 9i XML Database Developer's Guide*
📄 www.lc.leidenuniv.nl/awcourse/oracle/appdev.920/a96620/toc.htm



Systemy zarządzania treścią Część 1

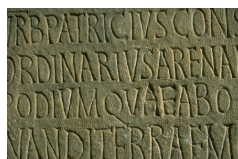
Statystyka

• **90%** zasobów informacyjnych firm
jest przechowywanych w dokumentach
a nie w bazach danych (Deloitte & Touche)

• **92 miliardy** dokumentów
tworzonych co roku (AIM)

Technologia

- Miejsce:
 - ryłca,
 - dłuta,
 - pióra,
 - maszyny do pisaniazajął komputer.
- Ale czy w istocie wiele się zmieniło?
 - kalka → *copy*
 - gumka → *back-space*

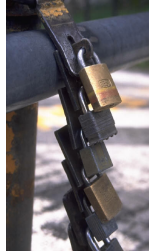


Najczęściej stosowane „systemy” zarządzania dokumentami

- Tradycyjny system obiegu dokumentów papierowych (szafy, segregatory, asystentka, goniec).
- Poczta elektroniczna, wymiana przy pomocy dyskietek, pen-drive'ów, itp.
- Współdzielony system plików (dysk sieciowy).

Dostępność

- Większość z istniejących dokumentów, mimo iż dostępnych elektronicznie, jest nadal nieużyteczna.
- Powody:
 - zamknięte i niekompatybilne formaty (n.p. DTP, Ms Word),
 - niedostępne/nieznanne miejsce przechowywania,
 - rozproszone, rozbieżne i nieaktualne wersje,
 - nieużywane już aplikacje.



Kiedy przestaje wystarczać system plików

- Zasoby informacyjne:
 - o dużej objętości,
 - o skomplikowanej strukturze i powiązaniach,
 - o dużej wartości,
 - o długim cyklu życia informacji,
 - o dużej częstotliwości aktualizacji informacji.
- Organizacja:
 - wieloosobowe zespoły,
 - wysoka specjalizacja członków zespołu,
 - rozproszenie geograficzne.

Kiedy przestaje wystarczać system plików

- Przykłady:
 - wydawnictwo encyklopedyczne,
 - wydawnictwo prawnicze,
 - wydawca czasopism,
 - koncern przemysłowy, producent zaawansowanych technicznie urządzeń,
 - operator rozległej sieci telekomunikacyjnej, energetycznej, ...,
 - organizacja oparta na wiedzy,
 - administracja państwowa.

Proste (i tanie) rozwiązania

- Centralne repozytoria, np.:
 - CVS (Concurrent Versions System),
 - SVN (Subversion).
- Typowe funkcje:
 - centralne składowanie dokumentów,
 - lokalne kopie, synchronizowane z repozytorium,
 - blokowanie dokumentów do edycji i zwalnianie blokady po edycji,
 - wersjonowanie dokumentów,
 - możliwość równoległej edycji dokumentów przez wiele osób i scalanie dokumentów.
- Rozwiązania typu wiki, np. MediaWiki, MoinMoin.

Budowa typowego systemu zarządzania treścią

- Repozytorium dokumentów.
- Warstwa aplikacji:
 - funkcjonalność biznesowa,
 - system przepływu prac,
 - silnik wyszukiwania,
 - silnik sprawdzający poprawność dokumentów,
 - silniki przekształceń,
 - system publikacyjny.
- Interfejs użytkownika:
 - system nawigacji,
 - system edycyjny.

Repozytorium

- Przechowywanie dokumentów:
 - dowolne typy dokumentów,
 - wersjonowanie,
 - blokowanie dokumentów do edycji,
 - specjalne wsparcie dla SGML-a/XML-a.
- Metainformacje:
 - informacje o dokumentach (np. autorzy, data publikacji, wersja),
 - przechowywane poza dokumentami (np. w bazie relacyjnej),
 - konfigurowalne w dowolny sposób,
 - metainformacje strukturalne (listy, struktury),
 - synchronizacja metainformacji z zawartością dokumentu,
 - wyszukiwanie w metainformacjach.

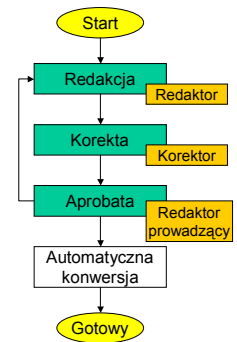
Metainformacje – przykład

```
<zeznanie-sprawcy>
Wypadek dnia <data>
13.01.2001</data>
o godzinie <godzina>13.13
</godzina> (<dzien-tygodnia>
piątek</dzien-tygodnia>) miał
miejsce nie z mojej winy,
<poszkodowany>Alojzy
M.</poszkodowany> nie miał
żadnego pomysłu w którą
stronę uciekać, więc go
przejechałem.
</zeznanie-sprawcy>
```

Nr protokołu: 1313/2001
Miejsce: Dołowice Górne
Data: 13.01.2001
Czas: 13.13
Rodzaj: potrącenie pieszego
Sprawca: Walenty Pechowy
Sprawę prowadzi: st. asp. Jan Łapówka

Przeływ prac

- Dwa podejścia:
 - tradycyjnie: działania osób „popychają” przepływ prac.
 - elektroniczny przepływ prac: steruje działaniami osób.



Interfejs użytkownika

- Nawigacja w zasobach repozytorium:
 - drzewiasta struktura katalogów,
 - wyświetlanie (wybranych) metainformacji,
 - struktura katalogów może przenieść istotną informację semantyczną o zależnościach między obiektami.
- Edycja dokumentów:
 - SGML/XML: edytor strukturalny,
 - obiekty dowolnego typu: aplikacje przeznaczone do ich edycji.