

# Oracle XML DB

Tomáš Nykodým  
xnykodym@fi.muni.cz

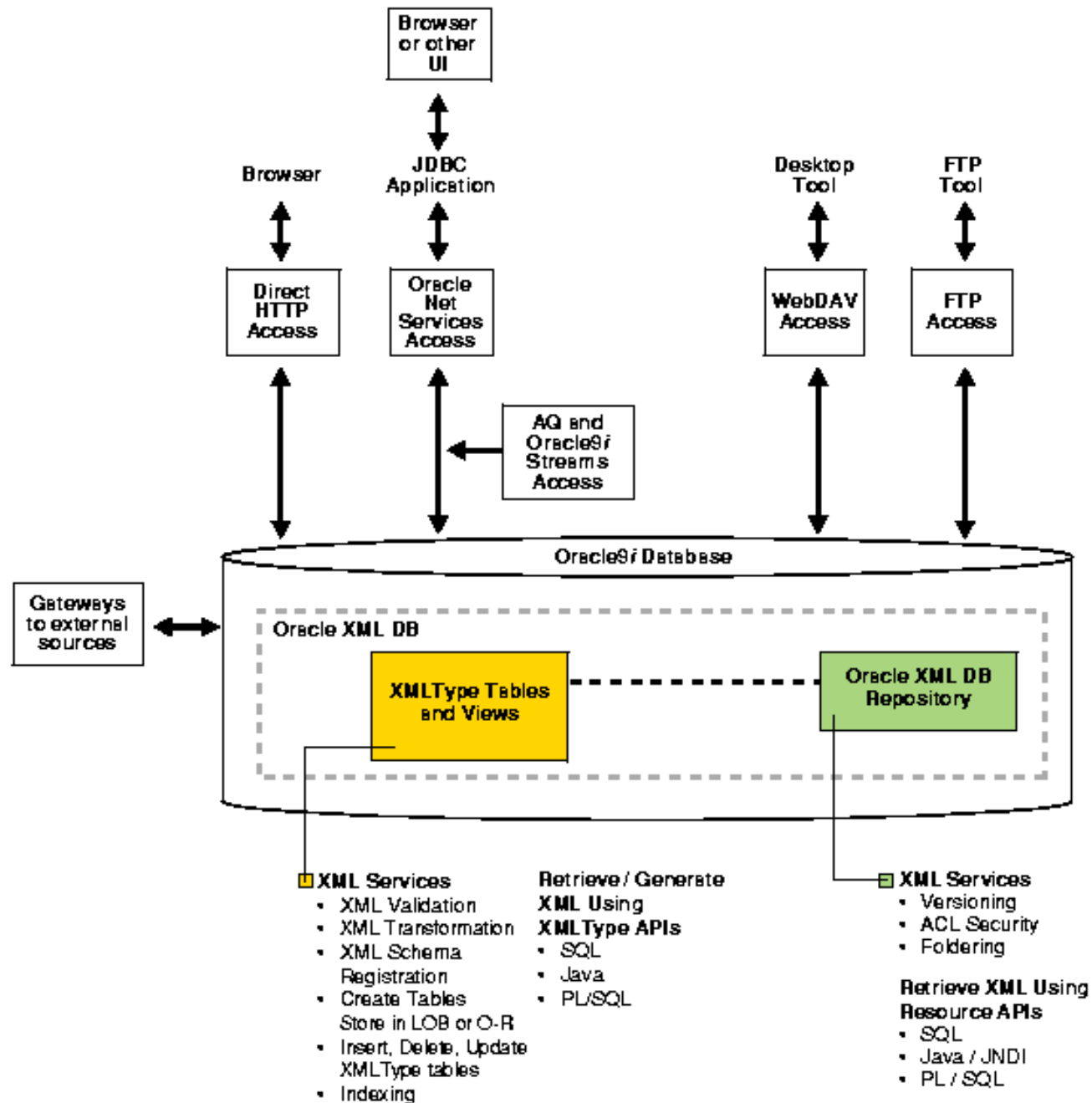
# Osnova

- Oracle XML DB
- Architektura Oracle XML DB
- Hlavní rysy Oracle XML DB
- Hlavní rysy Oracle XML DB - pokračování
- XMLType
- XML Repository
- Využívání databázových možností
- Využívání XML možností
- Kdy použít Oracle XML DB?
- Použití, příklady, odkazy

# Oracle XML DB

- množina zabudovaných, vysoce výkonných úložných a vyhledávacích technologií zaměřených na XML
- plně podporuje [W3C](#) XML datový model
- spojuje všechny výhody relačních databází a technologie XML současně
- XML/SQL dualita – XML operace nad SQL daty, SQL operace nad XML daty
- možnost ukládat a spravovat strukturovaná i nestrukturovaná data
- podpora otevřených standardů – [XPath](#), [SQL/XML](#)

# Architektura Oracle XML DB



# Hlavní rysy Oracle XML DB

- **XMLType**

- úsekové aktualizace dokumentu XML (**XPath**)
- XPath vyhledávání
- XML indexy – rychlý přístup k XML obsahu
- XSL transformace pro XMLType
- XML pohledy
- SQLX operátory – ANSI SQLX

- **XML Schema**

- věrnost reprodukce DOM (DOM Fidelity)
- věrnost reprodukce dokumentu (včetně bílých znaků)
- plná podpora standardu W3C - **XML Schema**
  - validace podle XML Schema

# Hlavní rysy Oracle XML DB - pokračování

- **XML DB Repository**
  - vyhledávání v XML Repository
  - zabezpečení přístupu k datům – ACL
  - správa verzí
- WebDAV, HTTP, FTP přístup k datům
- **přístupové XML API: C, PL/SQL, Java**
  - servlets/JSP (JDBC)
  - XSL/XSPs (XML dokumenty + DOM)
- **XML/SQL dualita**

# XMLType

- datový typ pro ukládání XML dat – typ sloupce nebo celé tabulky
- způsoby ukládání:
  - **LOB** (Large Object)
    - zachová přesnou reprezentaci dokumentu
    - velká přizpůsobivost změně schématu
  - **objektově-relační reprezentace**
    - omezená přizpůsobivost změně schématu
    - vypuštění whitespace znaků, komentářů
    - vysoký DML výkon
    - přístupnost existujících SQL rysů (indexy, integritní omezení,...)

# XML Repository

- sklad pro správu a přístup k XML dokumentům
- části:
  - seznam a správa přístupových práv (ACL) – i vlastní práva
  - správa složek
  - WebDAV a FTP protokol pro přístup ke složkám
  - SQL vyhledávání v XML Repository
  - API pro práci s XML Repository
  - přístup pomocí Java servletu – manipulace s objekty



# Využívání databázových možností

- bez podpory XML se XML data ukládala jako nestrukturovaná data (CLOB) nebo v XML souborech, kde nebylo možno využít některé klíčové vlastnosti databází:
  - **indexace a vyhledávání** – Oracle XML DB umožňuje efektivní strukturované vyhledávání nad XML daty
  - **aktualizace a transakční zpracování**
  - **více pohledů na data** – relační pohledy, kombinování dat
  - **výkonnost a škálovatelnost** – rychlé ukládání, dotazování a aktualizace XML dat

# Využívání XML možností

- **strukturální nezávislost** – otevřený obsahový model nelze jednoduše zachytit pomocí čistého světa tabulek a sloupců  
=> XML Schema
- **nezávislost na uložení** – XMLType umožňuje psát aplikace bez přesné znalosti kde jsou data uložena, v jakém formátu, v jaké tabulce, jaké jsou vztahy mezi tabulkami, dovoluje administrátorovi mapovat strukturovaná data na fyzické tabulky a sloupce
- **jednoduchost prezentace** – podpora XML v prohlížečích, desktopových aplikacích. Relační data vyžadují programování vrstvy pro zpřístupnění dat uživateli. XML DB ukládá data jako XML a také vyvolává XML data
- **jednoduchost výměny** – XML je jazykem B2B datové výměny. XML DB umožňuje čistou výměnu, bez několika-násobného překládání (ztrácí se informace)

# Kdy použít Oracle XML DB?

- XML DB je vhodné pro každou aplikaci, ve které jsou některá nebo všechna data reprezentována pomocí XML
- XML DB zabezpečuje vysokou výkonnost uložení, správy a získání XML dat. Podporuje možnost rychlé a jednoduché tvorby XML z relačních dat.
- typické aplikace:
  - B2B nebo A2A integrace
  - internetové aplikace
  - Content-management aplikace
  - Messaging
  - Web Services

# Použití Oracle XML DB

- Uložení dokumentu XML v Oracle DB použitím datového typu XMLType – uložení ve sloupci/tabulce
- Příklad:

```
CREATE TABLE PURCHASE_ORDER_TABLE  
(  
    po_number          number(16),  
    purchase_order     xmltype  
);
```

```
CREATE TABLE XMLTABLE OF XMLType;
```

# Přístup k datům XML

- nespočet funkcí poskytovaných Oracle XML DB je založených na W3C standardu **XPath**
- SQL operátory pro práci s XML:
  - `existsNode()` - vyhodnocuje, zda daný dokument obsahuje uzel definovaný XPath výrazem (vrací 1 nebo 0)
  - `extractValue()` - operátor pro navrácení hodnoty textového uzlu nebo atributu asociovaného s XPath výrazem. Datový typ návratové hodnoty je převeden na odpovídající datový typ SQL.
  - `extract()` - používá se v případě, že XPath výraz vrátí množinu uzlů (vrácené uzly jsou typu `XMLType`). Výsledkem může být celý XML dokument nebo jeho část.
  - `value()` - tento operátor umožní návrat celého XML dokumentu

# Vložení záznamu do tabulky

```
INSERT INTO PURCHASE_ORDER_TABLE VALUES (1234,  
XMLTYPE (  
'<PurchaseOrder>  
  <Reference>BLAKE-2002</Reference>  
  <Actions/>  
  <Rejects/>  
  <Requestor>David E. Blake</Requestor>  
  <User>BLAKE</User>  
  <CostCenter>S30</CostCenter>  
  <LineItems>  
    <LineItem ItemNumber="1">  
      <Description>A Night to Remember</Description>  
      <Quantity>2</Quantity>  
    </LineItem>  
  </LineItems>  
</PurchaseOrder>' ) )
```

# Dotaz nad tabulkou

```
SELECT extractValue
      (P.Purchase_Order, 'PurchaseOrder/User')
FROM PURCHASE_ORDER_TABLE P
WHERE existsNode (P.Purchase_Order,
                  '/PurchaseOrder[CostCenter="S30"]') = 1
```

Výsledkem dotazu bude (s ohledem na předchozí příklad) řetězec 'BLAKE'

```
SELECT value (P.Purchase_Order) FROM
PURCHASE_ORDER_TABLE P
WHERE existsNode (P.Purchase_Order,
                  '/PurchaseOrder[Reference="BLAKE-2002"]') = 1
```

Návratovou hodnotou bude dokument vložený v předchozích příkladech.

# Aktualizace hodnot

Příklad změny hodnoty textového uzlu určeného XPath výrazem:

```
UPDATE PURCHASE_ORDER_TABLE P
SET value(P.Purchase_Order) =
    updateXML(value(P.Purchase_Order),
        '/PurchaseOrder/Reference/text()', 'MILLER-2002')
WHERE existsNode(value(P.Purchase_Order),
    '/PurchaseOrder[Reference="BLAKE-2002"]') = 1;
```



# Integritní omezení

unikátní index:

```
alter table PURCHASE_ORDER_TABLE  
add constraint REFERENCE_IS_UNIQUE  
unique (PurchaseOrder."PurchaseOrder/Reference")
```

cizí klíč:

```
alter table PURCHASE_ORDER_TABLE  
add constraint USER_EMP  
foreign key (PurchaseOrder."PurchaseOrder/User")  
references emp(name)
```

# Generování XML z SQL dotazu

Pro vytváření XML z SQL dotazu nad tabulkou (tabulkami) slouží operátor **sys\_xmlgen()**.

```
select sys_xmlgen(null, empno, empname, salary*100,  
    hiredate) as xml_doc from scott.emp e
```

Výsledkem budou (pokud v tabulce emp existují 2 záznamy) XML dokumenty:

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

```
<ROW>
```

```
    <EMPNO>20</EMPNO>
```

```
    <EMPNAME>John</EMPNAME>
```

```
    <HIREDATE>12-20-1999 20:23:23.0</HIREDATE>
```

```
</ROW>
```

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

```
<ROW>
```

```
    <EMPNO>30</EMPNO>
```

```
</ROW>
```

# Použití XSL transformace v XML DB

XSLT je v Oracle XML DB prováděna především pomocí operátorů **transform()** a **XMLTransform()**.

```
SELECT value (P.Purchase_Order) .transform(  
    xmltype (getDocument ('purchaseOrder.xml'))  
FROM PURCHASE_ORDER_TABLE P  
WHERE existsNode (value (P.Purchase_Order),  
    '/PurchaseOrder[Reference="BLAKE-2002"]') = 1;
```

vrátí (za předpokladu že toto provádí ona transformace):

```
<html>  
<head/>  
<body bgcolor="#003333" text="#FFFFCC" link="#FFCC00"  
    vlink="#66CC99" alink="#669999">  
<FONT FACE="Arial, Helvetica, sans-serif">  
<center>  
...  
</FONT>  
</body>  
</html>
```

# Odkazy

- <http://otn.oracle.com/tech/xml/xmlldb/index.html>
- <http://otn.oracle.com/tech/xml/xmlldb/Current/TWP.pdf>
- <http://www.sweb.cz/bielohlav/oracle/>