IBM Software Group

Разработка Приложений с помощью pureXML

Николай Куликов, IBM EE/A
Nikolay_Kulikov@ru.ibm.com

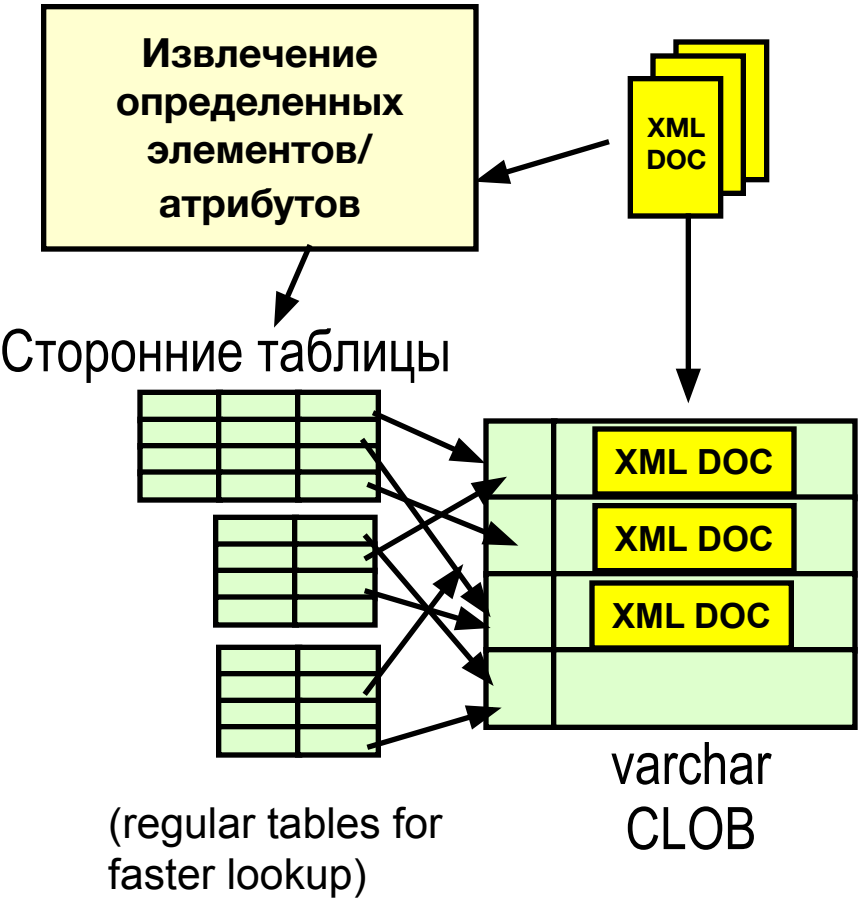
e business on demand software

Agenda

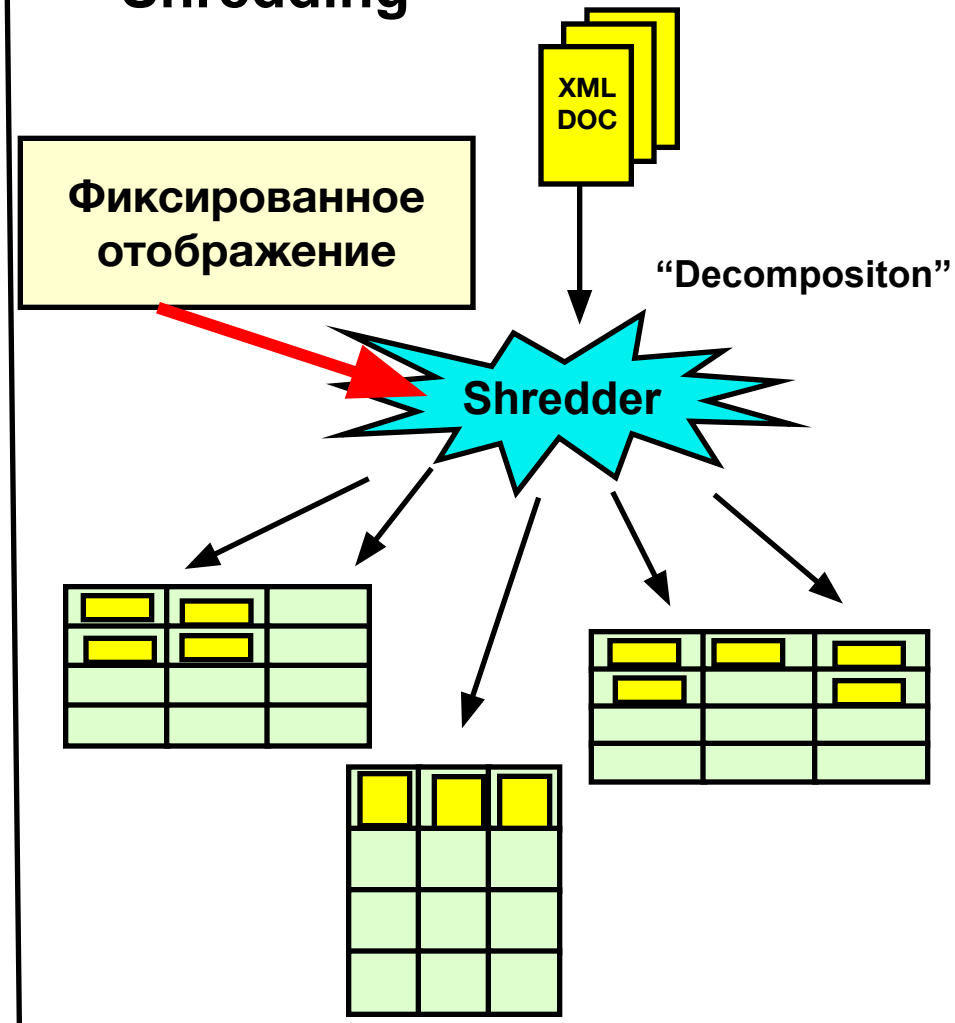
- Немного основ
- Интеграция в различные среды/языки разработки
  - Java
  - .Net
  - Ruby, PHP, Python
- Рекомендации по использованию pureXML
- Планы на будущее Viper II

XML-Enabled Databases: две опции

CLOB/Varchar

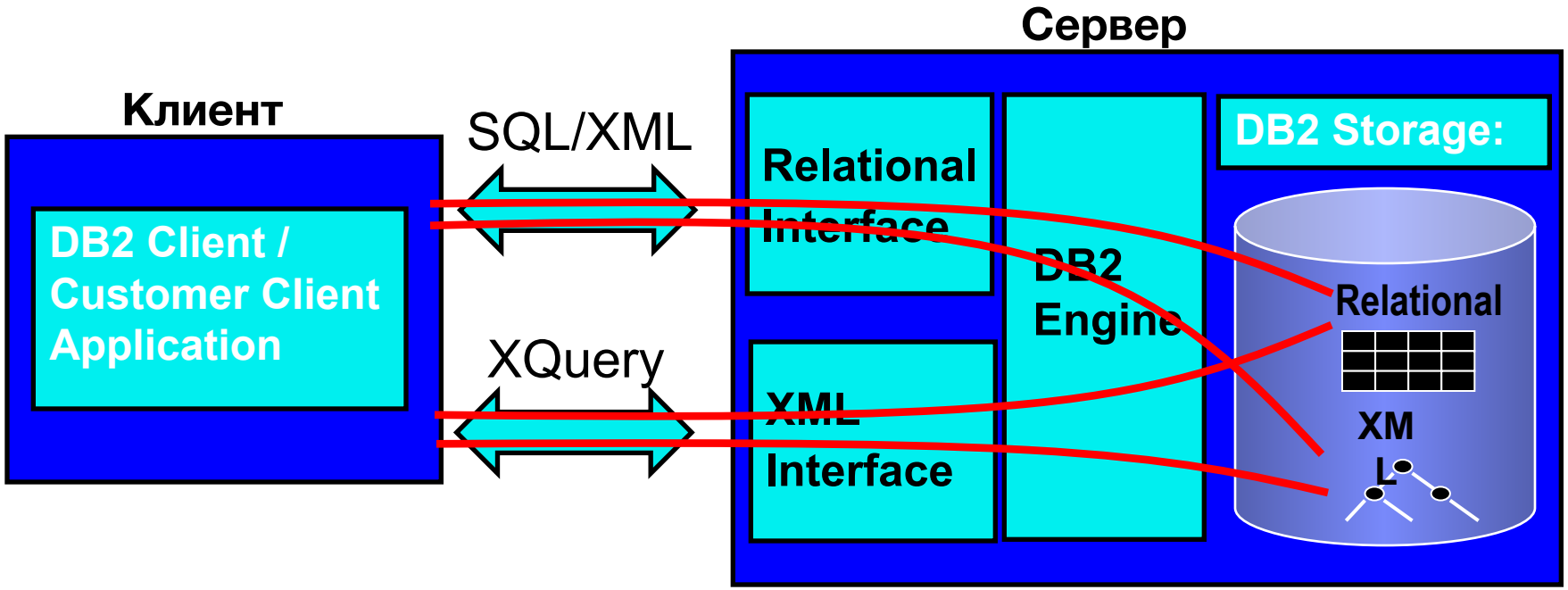


Shredding

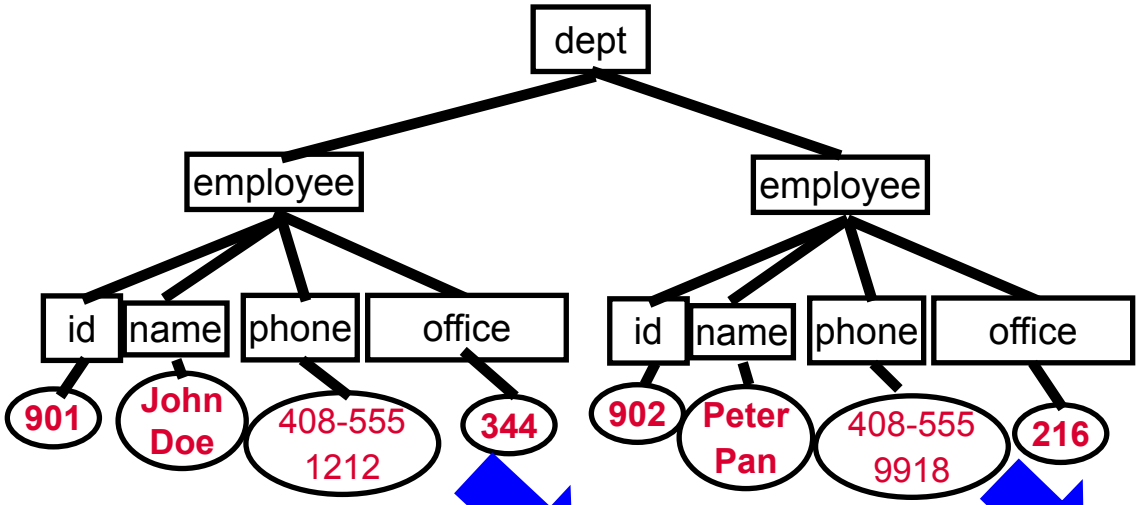


XML в DB2

SQL и XML одинаковы в DB2



Таблицы строк и путей



- Strings & Paths table per database
- Database wide dictionary…
- …for all documents in all XML columns

Strings table

0	dept
4	employee
1	name
5	id
2	phone
3	office

SYSIBM.SYSXMLSTRINGS

Paths table

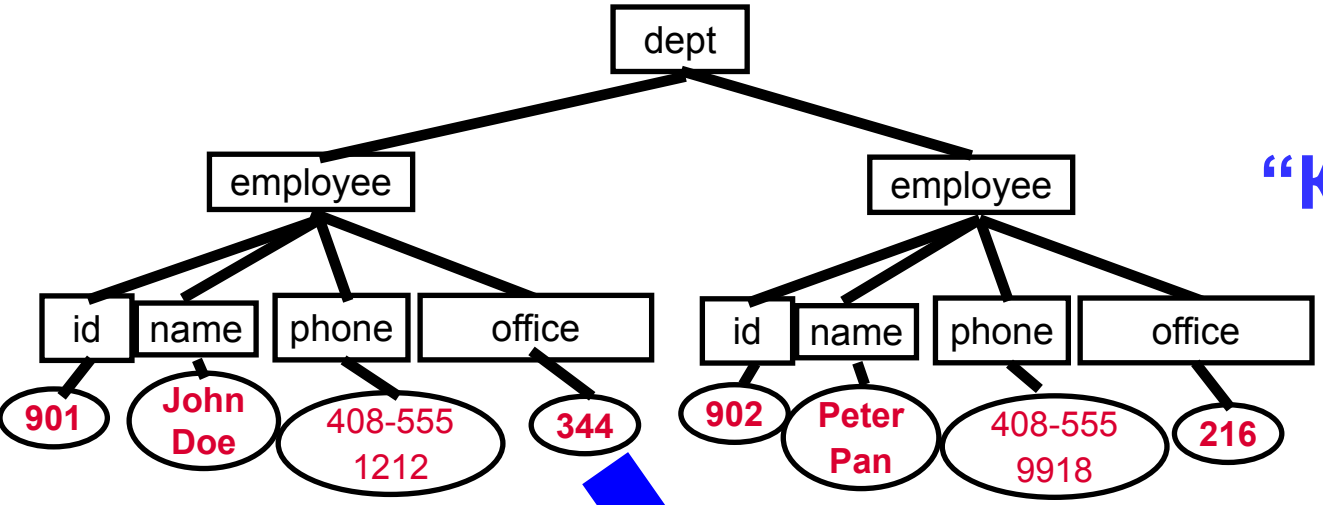
0	/
1	/dept
2	/dept/employee
3	/dept/employee/@id
4	/dept/employee/name
5	/dept/employee/phone
6	/dept/employee/office

SYSIBM.SYSXMLPATHS

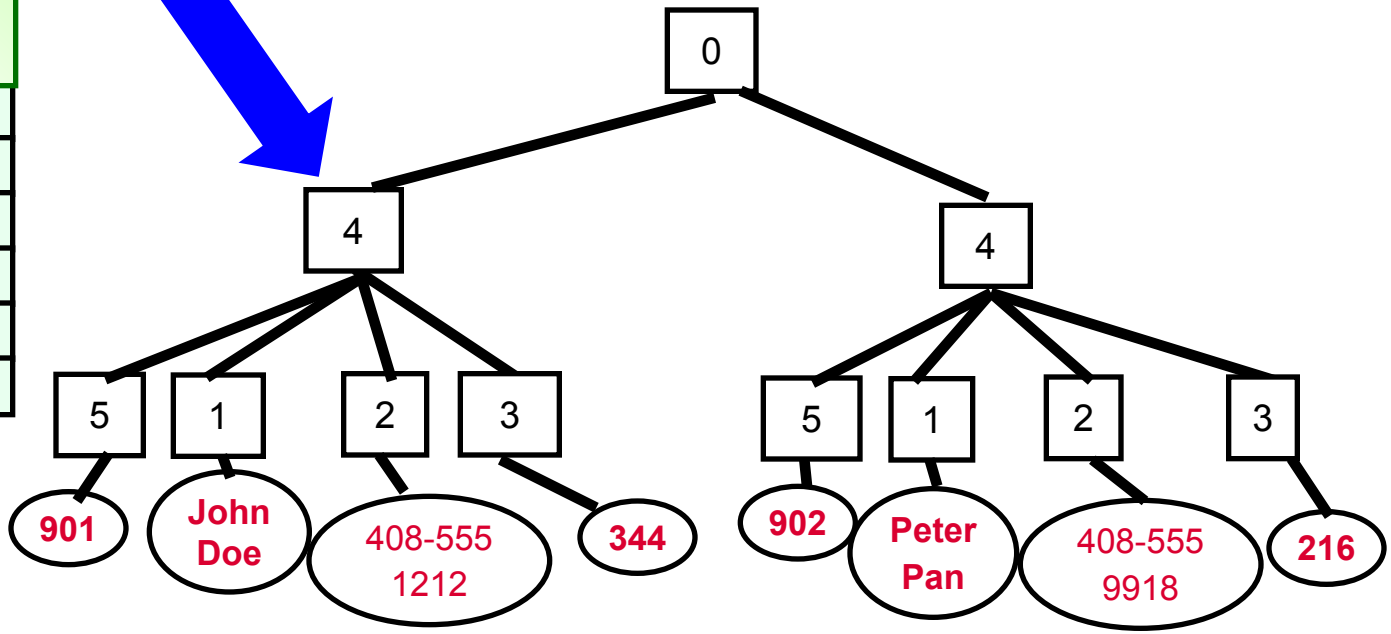
Paths table

0	/
1	/0
2	/0/4
3	/0/4/5
4	/0/4/1
5	/0/4/2
6	/0/4/3

“Компрессия"

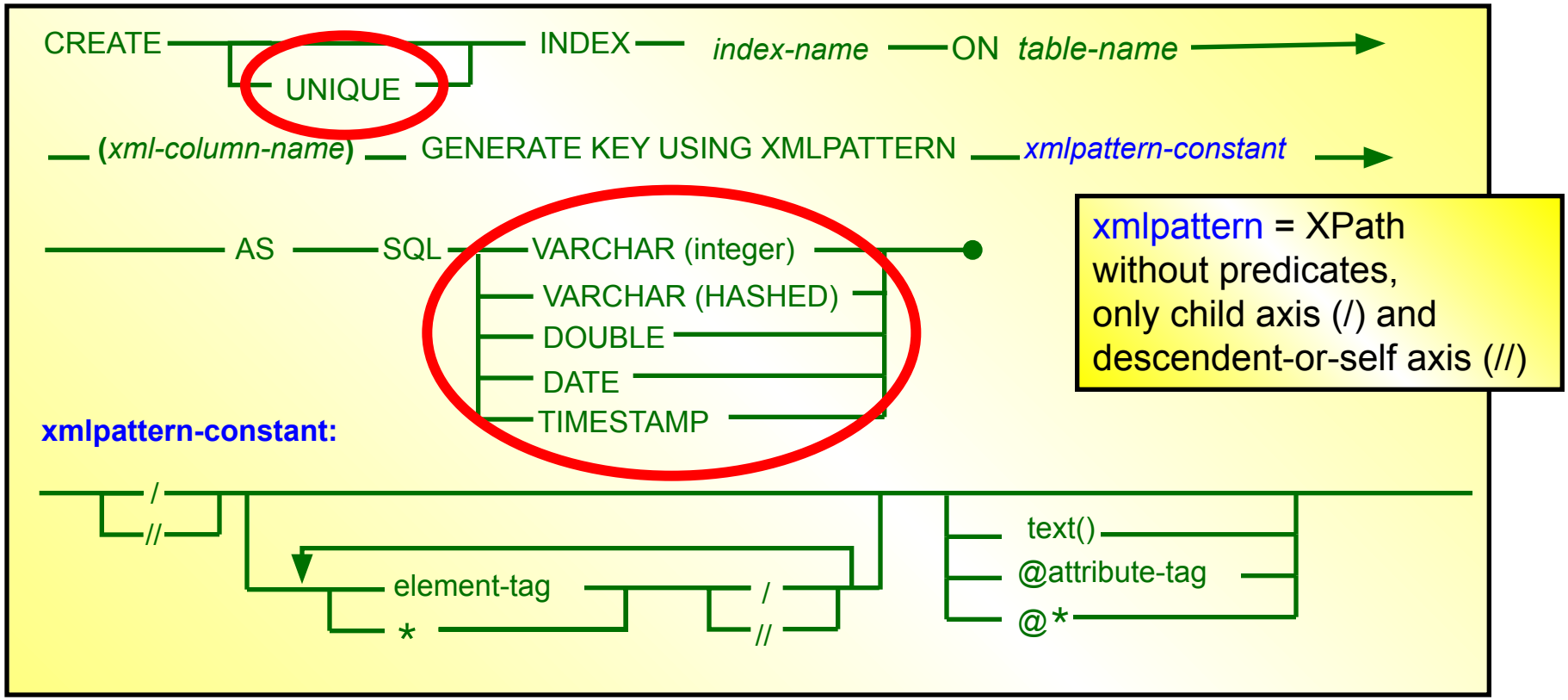


Strings table	
0	dept
4	employee
1	name
5	id
2	phone
3	office



DDL for Index on XML column

create index idx1 on T(xmlcol)
generate key using xmlpattern '/a/b/@c'
as sql date

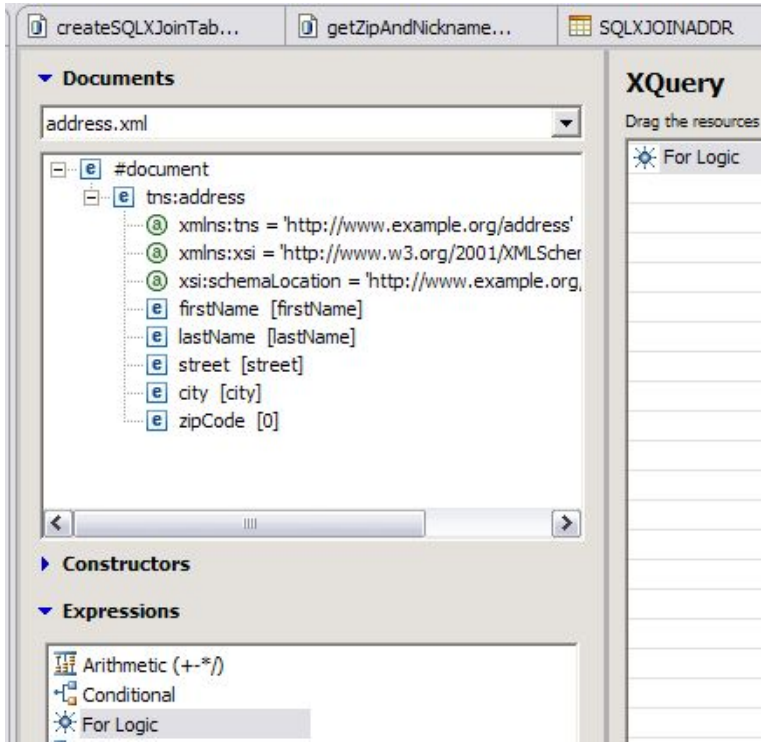


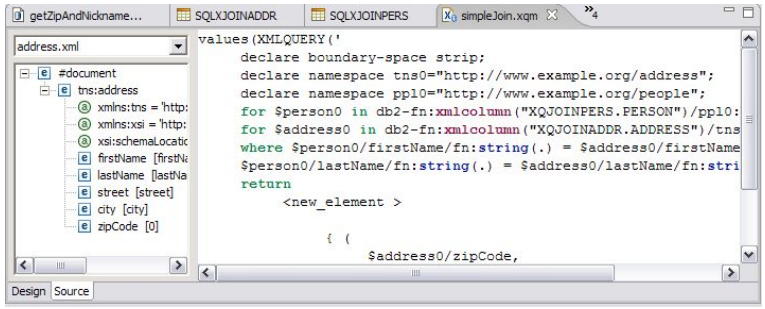
- Declaration & use of namespace prefix supported (not shown above)

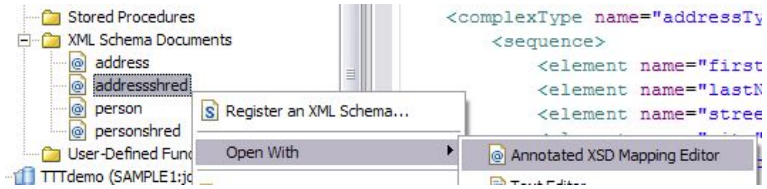
Требования разработчиков

- Быстрая разработка приложений БД
  - Работа в рамках одной среды Eclipse/Visual Studio/Zend Core/Ruby
  - Эффективное использования API надстроек над ним
  - Разработка и отладка клиентской и серверной логики в рамках среды Eclipse/.NET/Zend Core/Ruby
  - Управление новыми требованиями к данным
  - Интеграция с гетерогенными системами

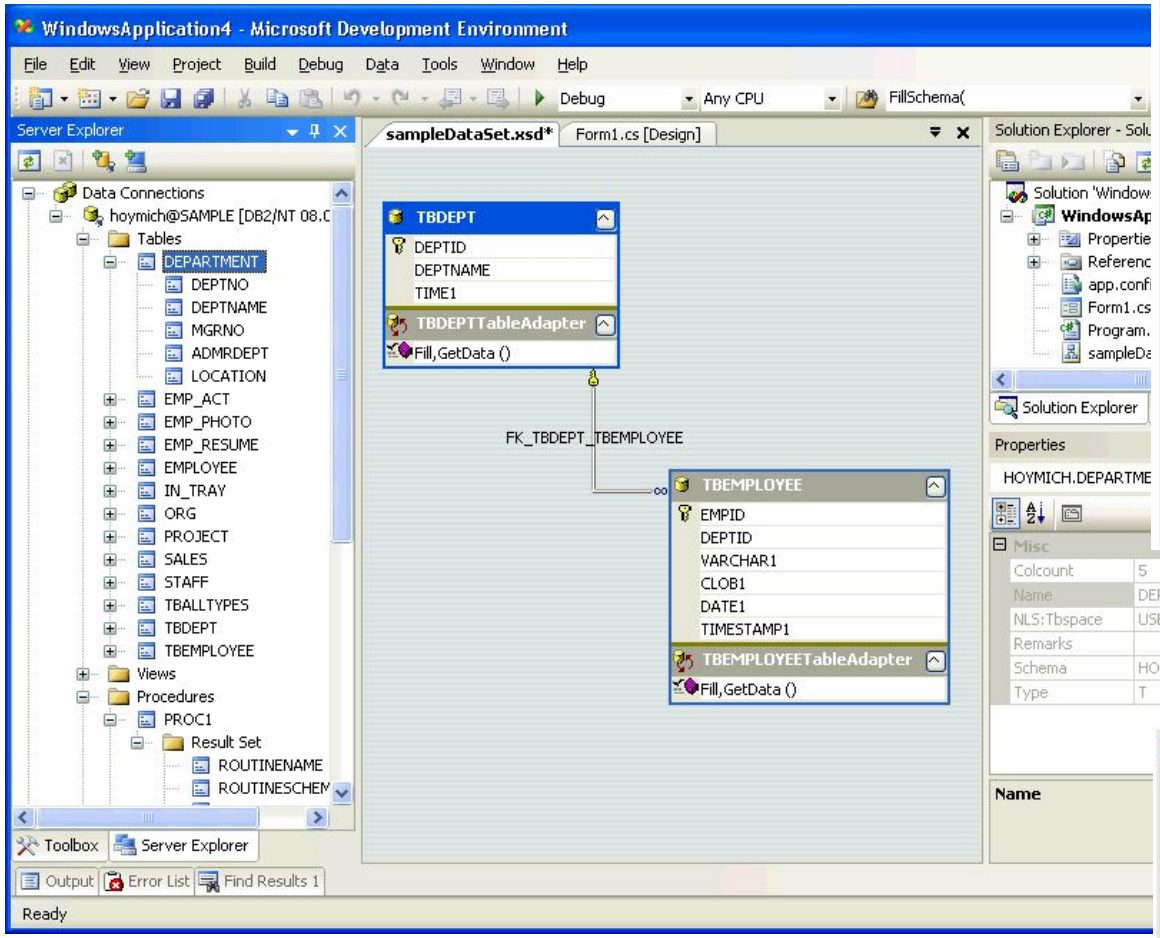
Интеграция в среды Eclipse

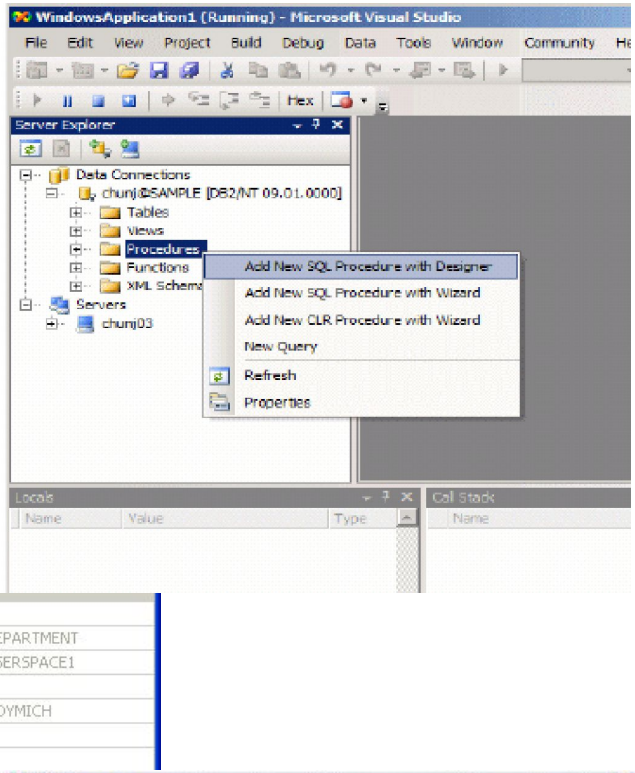


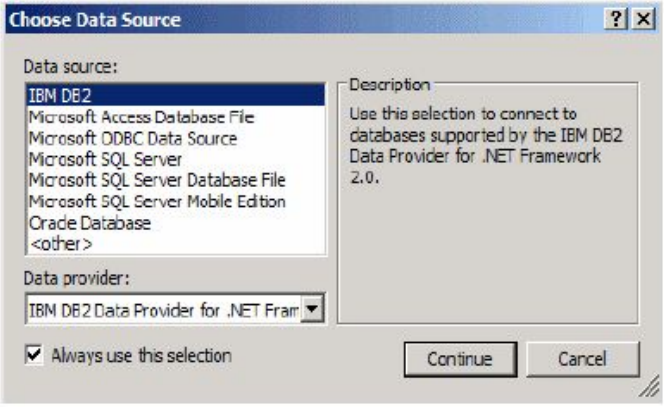




Интеграция в среды .Net







Новые возможности Add-ins для MS VS 2005

- Возможность генерации DB2 Web Services без кодирования
- Возможность создания и отладки SQL и CLR процедур
- Регистрация и работа со схемами в DB2
- Поддержка других серверов БД DB2 Informix IDS, DB2/iSeries, DB2 z/Series.

DB2 data provider for .NET Version 2.0

- Поддержка базовых классов System.Data.Common
- Эквивалентность типов данных DB2 и .NET
- Поддержка 64-bit
- Scrollable and updateable result sets
- Data Paging
  - Класс DB2Command теперь имеет ExecutePageReader
- Bulk Data Copy
- Update batch size
  - Позволяет приложению определять сколько операторов за раз будет отправлятся на сервер DB2 для обработки

Поддержка репозитория XML схем (XSR)

- Новый узел в.Net server explorer
- Полная интеграция с редактором XML схем в .NET
- Регистрация XML в XSR
- Удаление XML из XSR
- Модификация или аннотация XML схемы IBM DB2 XML

```
<ipo:purchaseOrder ….. orderDate="1999-12-01“
  orderID=“19991201-AZFG”>  ……..
   <item partNum="833-AA">
        <productName>Phone </productName>
        <quantity>1</quantity>
        <USPrice>132.95</USPrice>
        <shipDate>1999-12-05</shipDate>
    </item>
   <item partNum=“533-AC">
        <productName>Cycle</productName>
        <quantity>1</quantity>
        <USPrice>149.95</USPrice>
        <shipDate>1999-12-05</shipDate>
     </item></items></ipo:purchaseOrder>
```

ELECTRONICITEMS

ORDERID	PARTNUM	QTY	PRICE
19991201-AZFG	**833-AA**	1	132.95

```
…<sequence>
<element name="productName“ type="string“/>
<element name="quantity“
type=“ipo:derivedPositiveIntegerType”
db2-xdb:rowSet=“ELECTRONICITEMS”
db2-xdb:column=“QTY”/>
…….<!– ignoring mapping of PRICE --> ….
   <attribute name="partNum" type="ipo:SKU“
  db2-xdb:rowSet =” ELECTRONICITEMS”
  db2-xdb:column=“PARTNUM”
  db2-xdb:expr =“udf_convertToInternalPart($SELF)”
  db2-xdb:cond=“udf_isElectronicItem($SELF) =
                                    ‘true’/>……..
```

```
<attribute name="orderID" type=“xs:string“
  db2-xdb:rowSet =  “PURCHASE_ORDER”
  db2-xdb:column      = “ORDERID” >
 <annotation>
    <appinfo>
      <db2-xdb:tableMapping>
        <db2-xdb:rowSet>ELECTRONICITEMS</db2-xdb:rowSet>
        <db2-xdb:column>ORDERID</db2-xdb:column>
      </db2-xdb:tableMapping>
    </appinfo>
   </annotation>
</attribute>
```

Concurrent Data Readers

```
//Open connection to the database
DB2Connection conn = new DB2Connection ("database=sample;");
conn.Open();
//Create 2 commands from the same connection
DB2Command cmd1 = conn.CreateCommand();
DB2Command cmd2 = conn.CreateCommand();
cmd1.CommandText = "SELECT SALES_PERSON, SALES FROM SALES";
cmd2.CommandText  = "SELECT DEPTNUMB, DEPTNAME FROM ORG"
//Execute the first query
DB2DataReader dr1 = cmd1.ExecuteReader();
//Execute the second query while the first one is still open
DB2DataReader dr2 = cmd2.ExecuteReader();
//Get results from both data readers
while (dr1.Read())
{
   Console.WriteLine(" Sales Person {0}, Sales {1}"
  dr1.GetString(0), dr1.GetInt16(1));
}
while (dr2.Read())
{
   Console.WriteLine(" Department Number {0}, Department Name {1}"
  dr2.GetInt16(0), dr2.GetString(1));
}
//Close both data readers
dr1.Close();
dr2.Close();
conn.Close();
```

JDBC и XML обзор

- JDBC и XML сегодня
- Расширения DB2 для JDBC XML
- JDBC и XML Завтра (JDBC v4)
- Java's XML APIs
  - DOM, SAX, StAX, and Transform (XSLT)

JDBC and XML today

- JDBC сегодня имеет минимальные возможности по поддержке pureXML
  - Извлечение/обновление XML значений
  - XML как параметры
- SQL поддерживаются через JDBC
  - XQuery встроен в SQL
  - Параметры XQuery сначала проходят через SQL
- Java 1.4 и выше полностью Unicode
  - Нет проблем с перекодировкой

JDBC краткий обзор

1. Connection – соединение с БД

   connection = DriverManager.getConnection(url, user, pass);

2. Statement – Оператор для выполнения

   PreparedStatement stmt = connection.prepareStatement(sql);

3. ResultSet – результат выполнения

   ResultSet resultSet = stmt.executeQuery();

   4 Stream – значение XML

   InputStream inputStream = resultSet.getBinaryStream(1);

   4 ***или*** DB2XML ***или*** SQLXML

   DB2Xml db2xml = (DB2Xml) resultSet.getObject(1);

Расширения DB2's XML

- Функциональность совместима с будущим JDBC SQLXML
- XML LOB возвращается с ResultSet.getObject()
- Преобразование кодовых страниц
- Несколько методов для удобства как и в ResultSet

Выботка из XML столбца

```
String sql = "SELECT PID, DESCRIPTION from XMLPRODUCT
  where PID = ?";
PreparedStatement stmt =
  connection.prepareStatement(sql);
stmt.setString(1, "100-105-09");
ResultSet resultSet = stmt.executeQuery();
String xml = resultSet.getString("DESCRIPTION"); // or
InputStream inputStream =
  resultSet.getBinaryStream("DESCRIPTION"); // or
Reader reader =
  resultSet.getCharacterStream("DESCRIPTION"); // or
DB2Xml db2xml = (DB2Xml)
  resultSet.getObject("DESCRIPTION");
```

Вставка из файла XML

```
String sql = "INSERT INTO xmlproduct VALUES(?, ?)";
PreparedStatement stmt =
  connection.prepareStatement(sql);
stmt.setString(1, "100-105-09");
File binFile = new File("productBinIn.xml");
InputStream inBin = new FileInputStream(xmlFile);
stmt.setBinaryStream(2, inBin, (int)
  binFile.getLength());
stmt.execute();
```

JDBC и XML завтра (JDBC v4) (JSR 221)

- SQLXML объект добавлен к спецификации JDBC
  - getSQLXML() возвращает SQLXML объект
  - getObject() возвращает SQLXML объект
  - Объекты SQLXML являются также объектами DB2XML
- Представляет XML значение
- Похоже на BLOB/CLOB
- Доступ из ResultSet
  - SQLXML getSQLXML(int columnIndex)
  - SQLXML getSQLXML(String columnName)
  - void updateSQLXML(int columnIndex, SQLXML xmlObject)
  - void updateSQLXML(String columnName, SQLXML xmlObject)

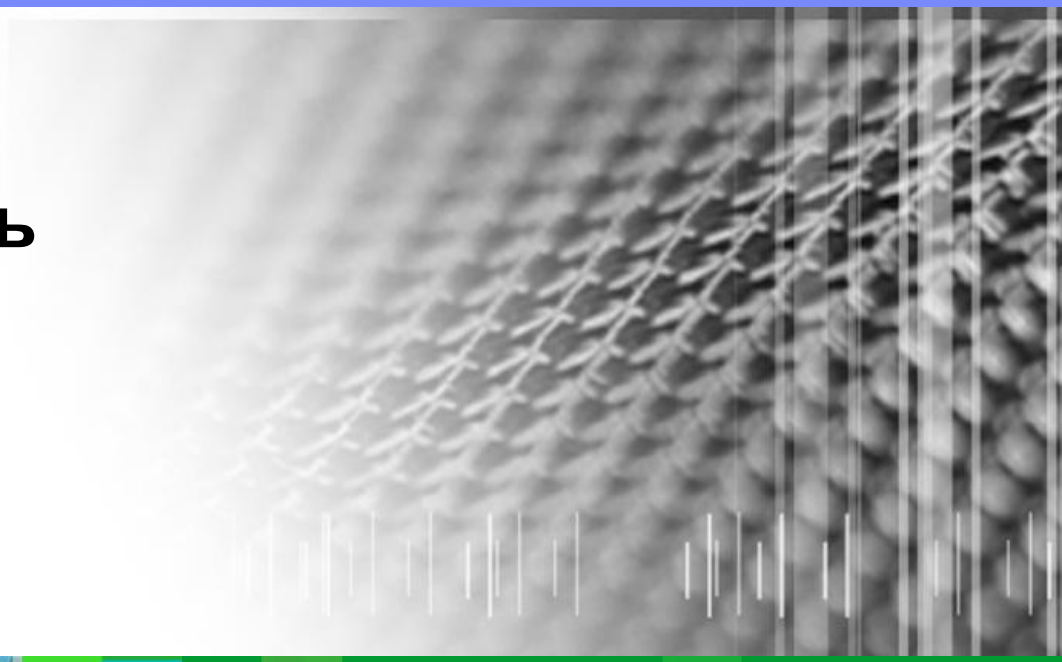
Source and Result Examples

- DOM get a Document
  - DOMSource domSource = sqlxml.getSource(DOMSource.class);
  - Document document = (Document) domSource.getNode();
- DOM set a Document
  - DOMResult domResult = sqlxml.setResult(DOMResult.class);
  - domResult.setNode(myNode);

- Run an XSLT on an XML result
  - File xsltFile = new File("my.xslt");
  - File myFile = new File("result.xml");
  - Transformer xslt = TransformerFactory.newInstance().newTransformer(new StreamSource(xsltFile));
  - Source source = sqlxml.getSource(null);
  - Result result = new StreamResult(myFile);
  - xslt.transform(source, result);

IBM Software Group

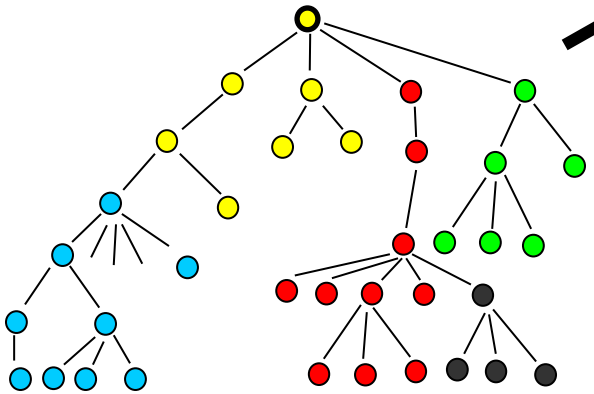
Производительность

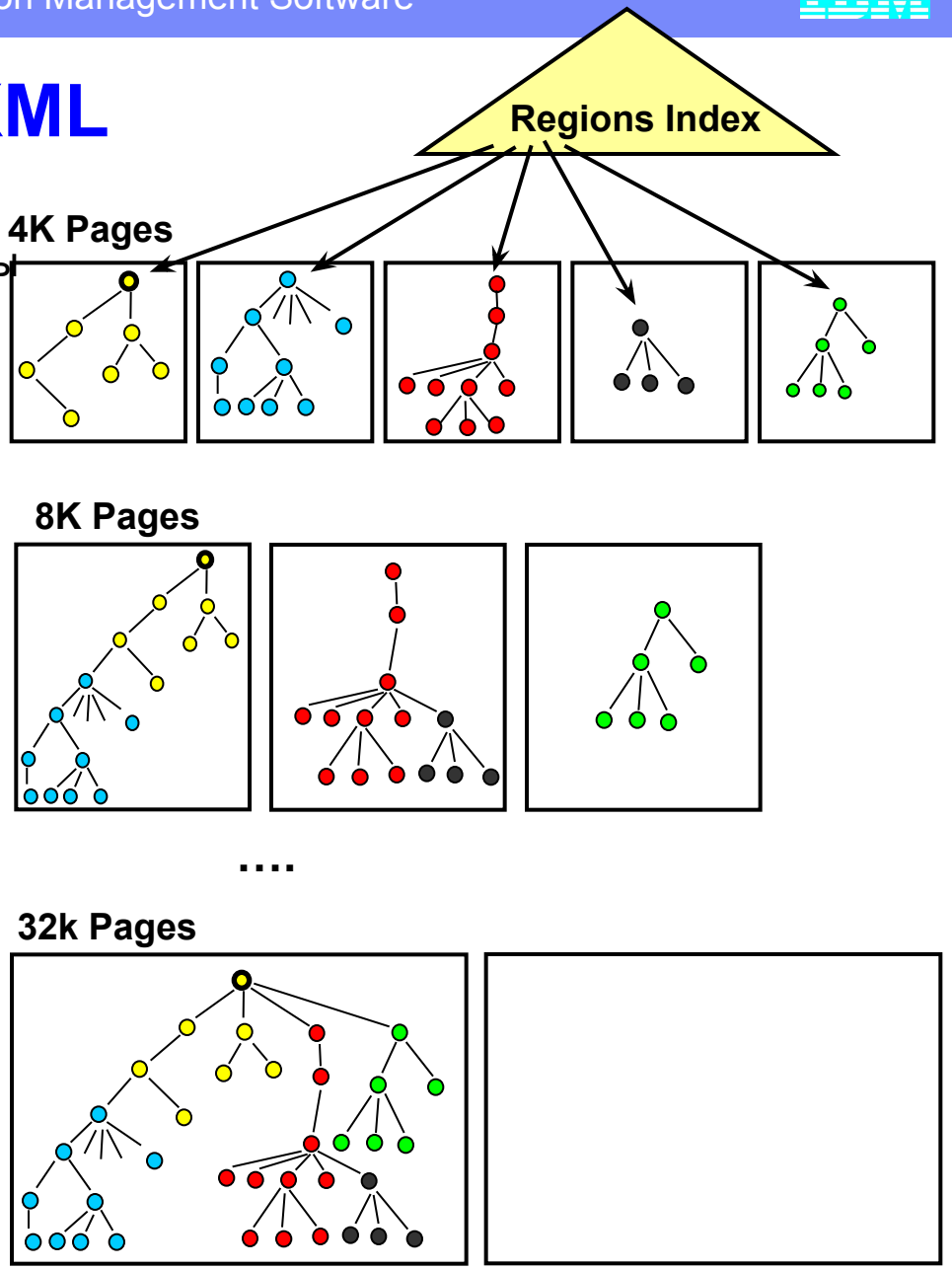


e business on demand software

Размер страницы для XML

- Большие документы разбиваются на регионы
- Макс док: 2GB, может находится на множестве страниц





Меньше регионов лучше производительность.

Выбирайте размер страницы в зависимости от размера документа !

Отдельные табличные пространства для XML

- Используйте **DMS** табличные пространства для лучше производительности !
- Используйте большие размеры страниц!

```
CREATE TABLE mytable(c1 integer, c2 char(8),…,c9 double,  c10 XML)
IN mytspace1
INDEX IN mytspace2
LONG IN mytspace3
```

Расположение XML данных и индексов в отдельных табличных пространствах позволяет делать более тонкую настройку…

…но только если это *действительно* нужно! Иначе пусть будет проще !

New Snapshot Monitor Counters for XML

```
Buffer pool data logical reads              = 253
Buffer pool data physical reads             = 70
Buffer pool temporary data logical reads    = 145
Buffer pool temporary data physical reads   = 0
Buffer pool data writes                     = 0
Buffer pool index logical reads             = 17275
Buffer pool index physical reads            = 0
Buffer pool temporary index logical reads   = 0
Buffer pool temporary index physical reads  = 0
Buffer pool index writes                    = 0
Buffer pool xda logical reads               = 2837
Buffer pool xda physical reads              = 174
Buffer pool temporary xda logical reads     = 0
Buffer pool temporary xda physical reads    = 0
Buffer pool xda writes                      = 0
```

Data Counters (relational)

Relational and XML Index Counters

XML Data Counters

New !

Проверяйте XDA счетчики для оценки XML активности !

Активности с XML Regions включается в index counters.

Проверка на основе XML схем

```
create table dept(deptID char(8),  deptdoc xml);
```

Validation is optional, and per document (per row):

```
insert into dept values (?, ?)
insert into dept values (?, xmlvalidate(?))
```

~~No Validation~~

~~With Validation~~

Валидация увеличивает процессорное время CPU для вставок, и уменьшает пропускную способность.

Примеры XML индексов

```
create table customer( info XML);
```

```
create unique index idx1 on customer(info)
generate key using
xmlpattern '/customerinfo/@Cid'
as sql double;

create index idx2 on customer(info)
generate key using
xmlpattern '/customerinfo/name'
as sql varchar(40);

create index idx3 on customer(info)
generate key using
xmlpattern '//name'
as sql varchar(40);
```

```
<customerinfo Cid="1004">
  <name>Matt Foreman</name>
  <addr country="Canada">
      <street>1596 Baseline</street>
      <city>Toronto</city>
      <state>Ontario</state>
      <pcode>M3Z-5H9</pcode>
  </addr>
  <phone type="work">905-555-4789</phone>
  <phone type="home">416-555-3376</phone>
  <assistant>
      <name>Peter Smith</name>
      <phone type="home">416-555-3426</phone>
  </assistant>
</customerinfo>
```

XML Indexing Examples

```
create table customer( info XML);
```

```
create unique index idx1 on customer(info)
generate key using
xmlpattern '/customerinfo/@Cid'
as sql double;
```

Не индексируйте все!
Слишком дорого для
insert, update, delete !

```
create index idx2 on customer(info)
generate key using
xmlpattern '/customerinfo/name'
as sql varchar(40);
```

```
create index idx3 on customer(info)
generate key using
xmlpattern '//name'
as sql varchar(40);
```

```
create index idx4 on customer(info)
generate key using
xmlpattern '//text()'
as sql varchar(40);
```

```
<customerinfo Cid="1004">
   <name>Matt Foreman</name>
   <addr country="Canada">
      <street>1596 Baseline</street>
      <city>Toronto</city>
      <state>Ontario</state>
      <pcode>M3Z-5H9</pcode>
   </addr>
   <phone type="work">905-555-4789</phone>
   <phone type="home">416-555-3376</phone>
   <assistant>
      <name>Peter Smith</name>
      <phone type="home">416-555-3426</phone>
   </assistant>
</customerinfo>
```

Optimizer Tips

- Гибридный оптимизатор основан на стоимости

Вы должны запускать ‘runstats’ на ваших XML данных

- XML Статистика включает наиболее часто встречаемые пары path/value
- Проверяйте планы на основе db2exfmt, Visual Explain

Рекомендации по XML Запросам

- Если возможно полностью указывайте точный XPath, а не шаблон.
  - /customerinfo/phone вместо //phone
  - /customerinfo/addr/state вместо /customerinfo/*/state

```
<customerinfo Cid="1004">
   <name>Matt Foreman</name>
   <addr country="Canada">
        <street>1596 Baseline</street>
        <city>Toronto</city>
        <state>Ontario</state>
        <pcode>M3Z-5H9</pcode>
   </addr>
   <phone type="work">905-555-4789</phone>
   <phone type="home">416-555-3376</phone>
   <assistant>
        <name>Peter Smith</name>
        <phone type="home">416-555-3426</phone>
    </assistant>
</customerinfo>
```

create table customer(**info** XML);

SQL/XML с XMLEXISTS

Для каждой строки:

- XMLEXISTS возвращает FALSE если встроенный XQuery возвращает **Пустую последовательность**,
- иначе XMLEXISTS возвращает TRUE, и строка возвращается

Индекс!

select c.**info** from customer c

where **xmlexists**(‘$i/customerinfo[name = “Matt Foreman”]’ passing c.**info** as “i”)

- **Строка возвращается имя Matt Foreman:**
- **XQuery вернет пустую строку если [...] удовлетворено**

Без индекса!

~~select c.**info** from customer c~~

~~where **xmlexists**(‘$i/customerinfo/name = “Matt Foreman”’ passing c.**info** as “i”)~~

- **Все строки выбираются :**
- **Встроенный XQuery всегда вернет или Boolean вне зависимости путая последовательность или нет**

SQL/XML with XMLQUERY

Без индекса!

select **xmlquery**('$i/customerinfo[phone = "905-555-4789"]/name' passing c.**info** as "i")
from customer c;

customer table:

```
<customerinfo>
    <name>Matt Foreman</name>
    <phone>905-555-4789</phone>
</customerinfo>
```

```
<customerinfo>
    <name>Peter Jones</name>
    <phone>905-123-9065</phone>
</customerinfo>
```

```
<customerinfo>
    <name>Mary Poppins</name>
    <phone>905-890-0763</phone>
</customerinfo>
```

```
<name>Matt Foreman</name>
```

3 record(s) selected

```
<name>Matt Foreman</name>
```

1 record(s) selected

Индекс!

select **xmlquery**('$i/customerinfo/name' passing c.**info** as "i")
from customer c
where **xmlexists**('$i/customerinfo[phone = "905-555-4789"]' passing c.**info** as "i")

Viper II

- ***Ограничения целостности на XML***
- ***Передача параметров в SQLQuery***
  - –XQuery нет возможности передачи параметров в SQL

```
XQUERY
for $docid in (1,2,3),
$j in db2-fn:sqlquery('select xmlcol
 from t1 where docid = parameter(1)',
$docid)/bib/book
where count($j/author) > 1
return $j;
```

Viper II

- ***Репликация XML***
- **Поддержка триггеров для XML**
  - Before Trigger сможет работать перед валидацией на вставку
- **Обновление части документа**

```
update T set doc = XMLQuery ('
transform
copy $r := $doc
do
delete {$r/score},
replaceValue of {$r/salary} with $r/salary * 1.1,
insert {<d>Ph.D.</d>} into $r//emp,
rename ($r/status[1]) to "state"
return $r'
passing doc as "doc",
cast (? as double) as "raise",
id as "i");
```

Questions?

Nikolay_Kulikov@ru.ibm.com

New XML Statistics

- Gathered by runstats when run on XML column and XML index
- Cannot be modified via update

- For each of the most frequent paths in an XML column we collect
  - The total number of times the path is seen
    - Top-k Pathid node counts
  - In how many documents it is seen
    - Top-k Pathid doc counts

- Same is collected for most frequent paths that lead to values including the value itself
  - Top-k Pathid-Value node counts
  - Top-k Pathid-Value doc counts

- Catch all stats for paths that are not frequent

- New registry variables for fine tuning, db2cat for inspection

XANDOR Join Example:

New !

Query: /doc/A[B=5 and C/F=6]

XML Index 1
/doc/A/B = 5 ?
…
B=5, docid=3
B=5, docid=4
B=5, docid=5
B=5, docid=7
B=5, docid=8
B=5, docid=9
B=5, docid=10
B=5, docid=11
B=5, docid=**15**,
nodeID=1.1.1
B=5, docid=16
B=5, docid=17
B=5, docid=**19**
nodeID=1.1.1
…

XML Index 2
/doc/A/C/F = 6 ?
…
F=6, docid=1
F=6, docid=2
F=6, docid=6
F=6, docid=**15**,
nodeID=1.2.1.1,
F=6, docid=12
F=6, docid=13
F=6, docid=**19**
nodeID=1.1.2.1
…

docID=15

```
<doc>
  <A>
    <B>5</B>
  </A>
  <A>
    <C>
      <F>6</F>
    </C>
  </A>
<doc>
```

docID=19

```
<doc>
  <A>
    <B>5</B>
    <C>
      <F>6</F>
    </C>
  </A>
<doc>
```

IBM Confidential