



Information Management

# Новые тенденции управления данными в продуктах компании IBM

*Nikolay\_Kulikov@ru.ibm.com*

IBM Software Group



**ON** DEMAND BUSINESS™

# Agenda

- **Part I – XML: Почему и Где**
- **Part II – Native XML в DB2 Viper**
- **Part III – Немного о других новых возможностях DB2 Viper**
- **Part III – Informix Dynamic Server 10**
- **Part IV – Вопросы и ответы**



# XML and Information on Demand

## ▪ Возрастающая важность XML

- 4 Формат обмена документами
- 4 Web Services
- 4 Обмен метаданными
- 4 Модель программирования

## ▪ Глубокая XML интеграция упрощает и объединяет инфраструктуру

- 4 Объединяет данные и Content
- 4 Расширяет использование активов

## ▪ Native XML хранилище

- 4 Оптимизация производительности
- 4 Предоставление гибкости
- 4 Поддержка природы XML
- 4 Сервисы БД



# Типичные сценарии использования XML

- Интеграция разрозненных источников данных
- Обмен информацией между приложениями и организациями
- eForms и системы workflow
- Управление наполнением и документами
- Транзакции основанные на сообщениях, web services, SOA
- XML документы как бизнес объекты/ транзакционные записи (цифровая подпись, аудит, соответствие регулируемому законодательству)
- XML как модель данных лучше подходит для (множественных значений, иерархий и комплексных данных)



# Пример 1: Финансовые данные (FIXML)

- Покупка 1000 акций компании IBM.

8=FIX.4.2^9=251^35=D^49=AFUNDMGR^56=ABROKER^34=2  
 ^52=20030615-01:14:49^11=12345^1=111111^63=0^64=2003  
 0621^21=3^110=1000^111=50000^55=IBM^48=459200101^22=  
 1^54=1^60=2003061501:14:4938=5000^40=1^44=15.75^15=USD  
 ^59=0^10=127

Старый протокол FIX

Новый FIXML

протокол

- Расширяемый
- Снижает затраты на поддержку и разработку

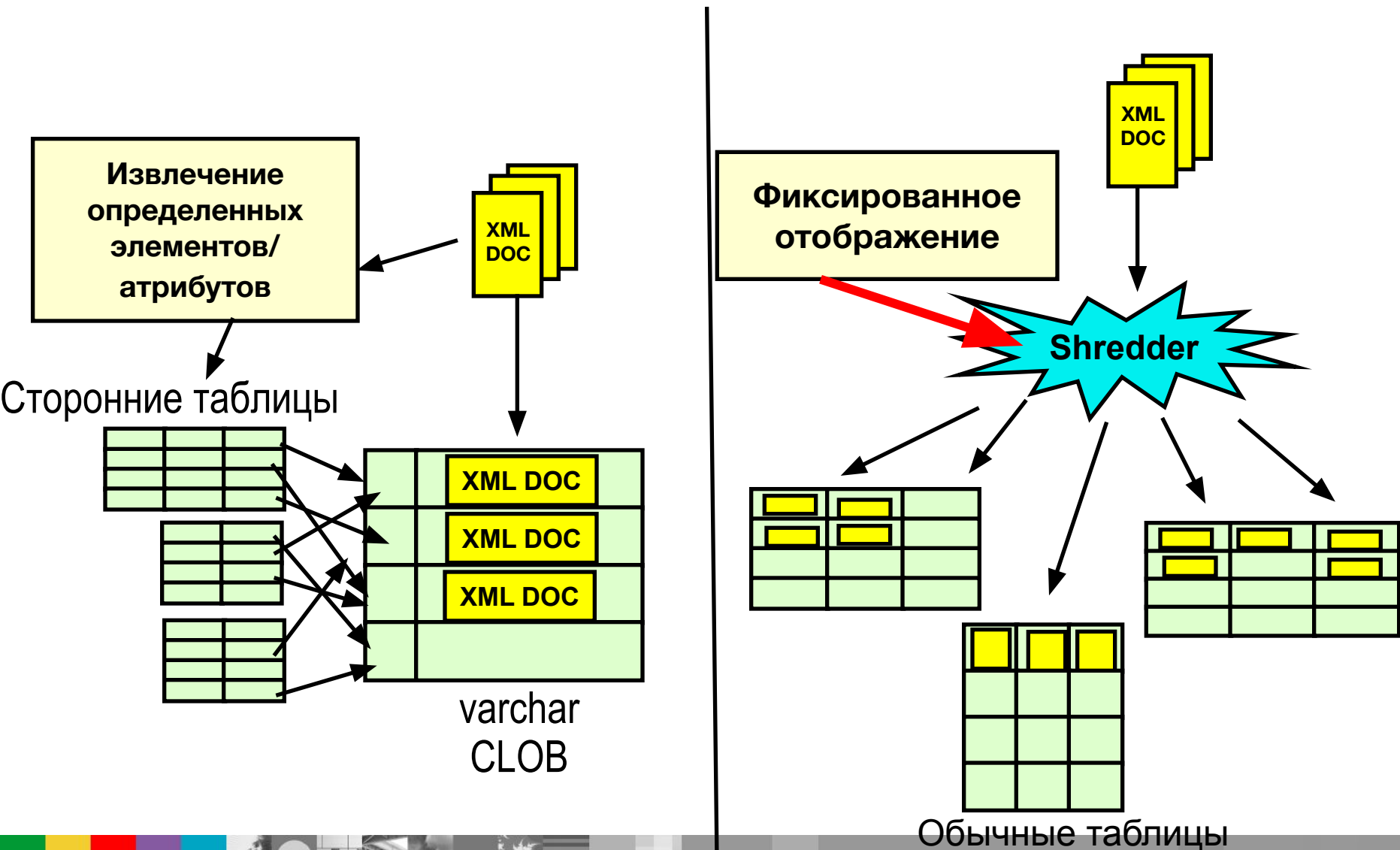
```
<FIXML >
  <NewOrdSingle  ClOrdID ="123456"
    Side ="2"
    TransactIm ="2003 -06 -15T01:14:49 -05:00"
    OrderType ="2"
    Price ="93.25"
    Acct ="26522154">
    < Header  Sent ="2001 -06 -21T01:31:28 -05:00"
      PosDup ="N"
      PosRsnd ="N"
      SeqNum ="521">
      < Sender  ID ="AFUNDMGR"/>
      < Target  ID ="ABROKER"/>
    </Header >
    <Instrument  Symbol ="IBM"
      ID ="459200101"
      IDSrc ="1"/>
    <OrderQuantity  Qty ="1000"  Cur ="USD"/>
  </NewOrdSingle >
</FIXML >
```

# Почему нужно использовать XML с БД ?

- **Управление большими объемами это проблема для БД!**
  - 4 Эффективный поиск и извлечение XML
  - 4 ACID
  - 4 Производительность, Масштабируемость
  - 4 *...те же причины что и для реляционных данных!*
  
- **Интеграция**
  - 4 Интеграция новых XML данных с существующими реляционными данными
  - 4 Публикация (реляционных) данных как XML
  - 4 Поддержка для web приложений, SOA, web служб (SOAP)



# XML-Enabled Databases: две опции



# Проблемы XML-enabled Databases

- Хранение как CLOB:
  - 4 Выполнение запроса требует разбора XML (Parsing) – медленно !
  
- Нарезка:
  - 4 Отображение из XML в реляционные таблицы часто слишком сложно
  - 4 Часто требует десятки, а то и сотни таблиц
  - 4 Требуются сложные Complex разнонаправленные соединения(joins) для восстановления XML документа
  - 4 При изменении схемы XML ломается mapping
    - Нет гибкости в схеме !
    - Например: изменение элемента, теперь он может встречаться в документе один и более раз. Это требует нормализации реляционной схемы и данных





# Нарезка: Простой пример

```
<DEPARTMENT deptid="15" deptname="Sales">
```

```
<EMPLOYEE>
```

```
<EMPNO>10</EMPNO>
```

```
<FIRSTNAME>CHRISTINE</FIRSTNAME>
```

```
<LASTNAME>SMITH</LASTNAME>
```

```
<PHONE>408-463-4963</PHONE>
```

```
<SALARY>52750.00</SALARY>
```

```
</EMPLOYEE>
```

```
<EMPLOYEE>
```

```
<EMPNO>27</EMPNO>
```

```
<FIRSTNAME>MICHAEL</FIRSTNAME>
```

```
<LASTNAME>THOMPSON</LASTNAME>
```

```
<PHONE>406-463-1234</PHONE>
```

```
<SALARY>41250.00</SALARY>
```

```
</EMPLOYEE>
```

```
</DEPARTMENT>
```

## Department

DEPTID	DEPTNAME
15	Sales

## Employee

DEPTID	EMPNO	FIRSTNAME	LASTNAME	PHONE	SALARY
15	27	MICHAEL	THOMPSON	406-463-1234	41250
15	10	CHRISTINE	SMITH	408-463-4963	52750

# Нарезка: Изменение схемы...

```

<DEPARTMENT deptid="15" deptname="Sales">
  <EMPLOYEE>
    <EMPNO>10</EMPNO>
    <FIRSTNAME>CHRISTINE</FIRSTNAME>
    <LASTNAME>SMITH</LASTNAME>
    <PHONE>408-463-4963</PHONE>
    <PHONE>415-010-1234</PHONE>
    <SALARY>52750.00</SALARY>
  </EMPLOYEE>
  <EMPLOYEE>
    <EMPNO>27</EMPNO>
    <FIRSTNAME>MICHAEL</FIRSTNAME>
    <LASTNAME>THOMPSON</LASTNAME>
    <PHONE>406-463-1234</PHONE>
    <SALARY>41250.00</SALARY>
  </EMPLOYEE>
</DEPARTMENT>

```

## Требует:

- Нормализации текущих данных !
- Изменение отображения
- Изменения приложения

## Phone

EMPNO	PHONE
27	406-463-1234
10	415-010-1234
10	408-463-4963

## Department

DEPTID	DEPTNAME
15	Sales

Costly!

## Employee

DEPTID	EMPNO	FIRSTNAME	LASTNAME	PHONE	SALARY
15	27	MICHAEL	THOMPSON	406-463-1234	41250
15	10	CHRISTINE	SMITH	408-463-4963	52750

# Зачем native XML хранилище?

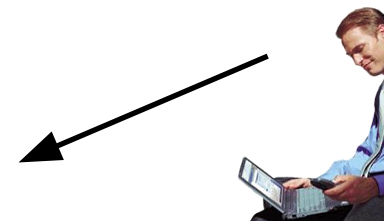
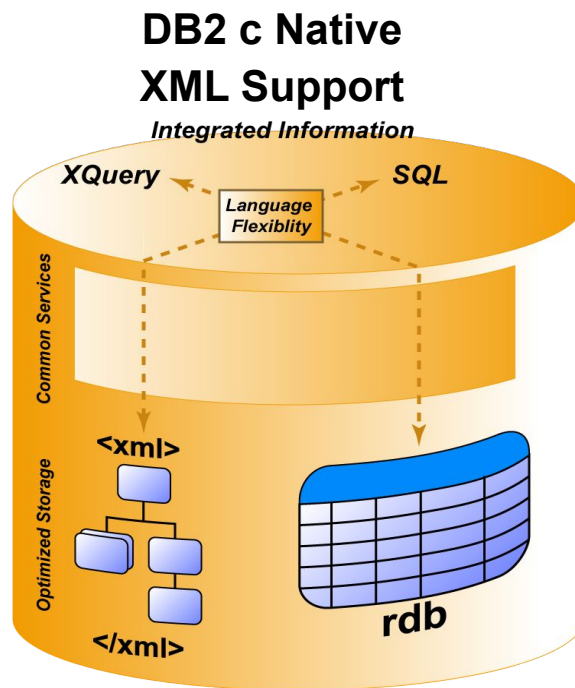
- Много данных в реляционных БД
  - 4 И это не будет меняться... так как SQL DBs быстрые
  
- XML формат обмена
  - 4 Web Services везде
- XML как артефакт транзакции
  - 4 Все больше и больше данных представляются в формате XML с самого начала...
  
- Держать XML как XML
  - 4 Каждая трансформация (нарезка) затрата
  - 4 И потенциально ведет к потерям информации
  - 4 XML может быть слишком сложным для нарезки
  - 4 Иерархический XML более адаптирован к человеческому мышлению



## XML в DB2



**SQL Developer...** "Я вижу комплексную РСУБД которая так же поддерживает XML"



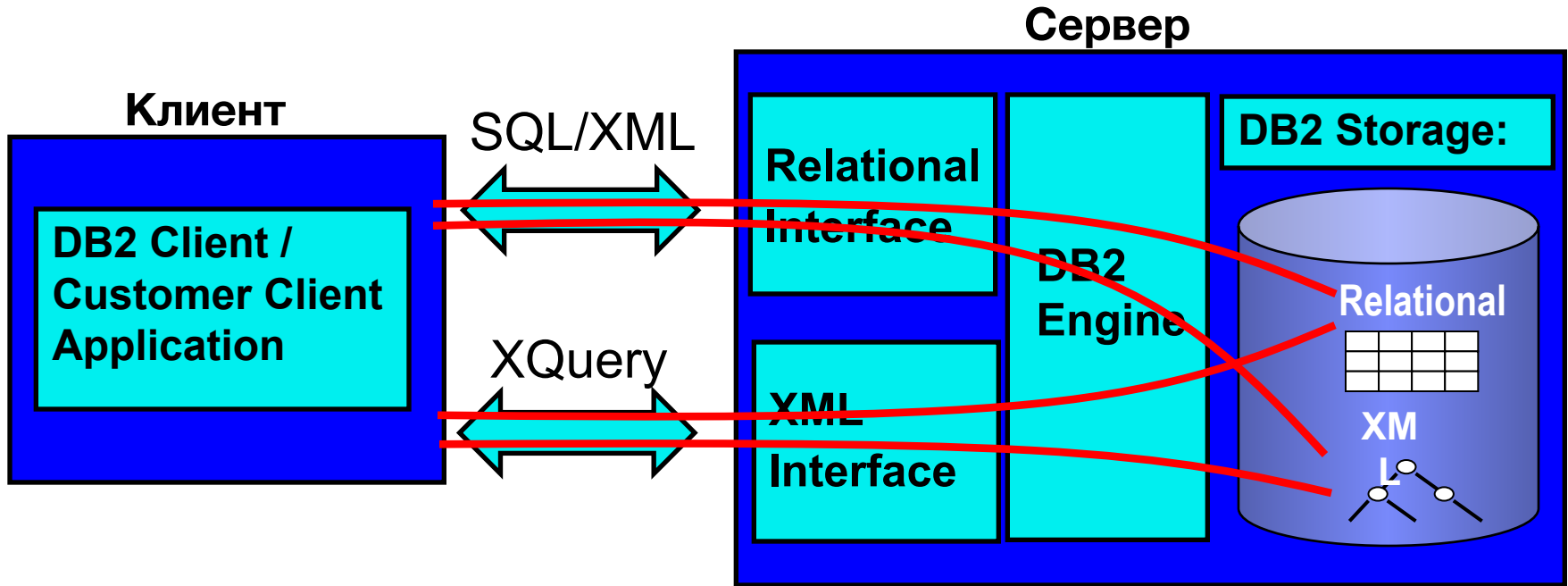
**XML Developer...** "Я вижу комплексный XML репозиторий XML который так же поддерживает SQL"

**XML интегрирован во все аспекты DB2!**



## XML в DB2

## SQL и XML одинаковы в DB2

**XML возможности встроены в DB2**

Глубока интеграция XML подразумевает высокую производительность!

**Оптимизированное хранение данных**

Новое хранилище и индексы для XML

**Множество интерфейсов для запросов**

- SQL/XML и XQuery
- ⇒ Оба языка имеют полный доступ ко всем хранимым данным
- ⇒ Выбор представления данных которое больше подходит для приложения

# Обзор Интерфейса

- Определение данных  
`create table dept(deptID int, deptdoc xml);`
- Вставка  
`insert into dept(deptID, deptdoc) values (?,?)`
- Индексирование  
`create index xmlindex1 on dept(deptdoc)`  
`generate key using xmlpattern '/dept/name' as varchar(30);`
- Извлечение  
`select deptdoc from dept where deptID = ?`
- Запросы  
`select deptID, xmlquery('$d/dept/name' passing deptdoc as "d")`  
`from dept where deptID <> "PR27";`

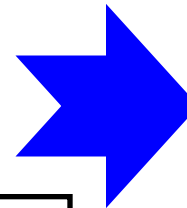
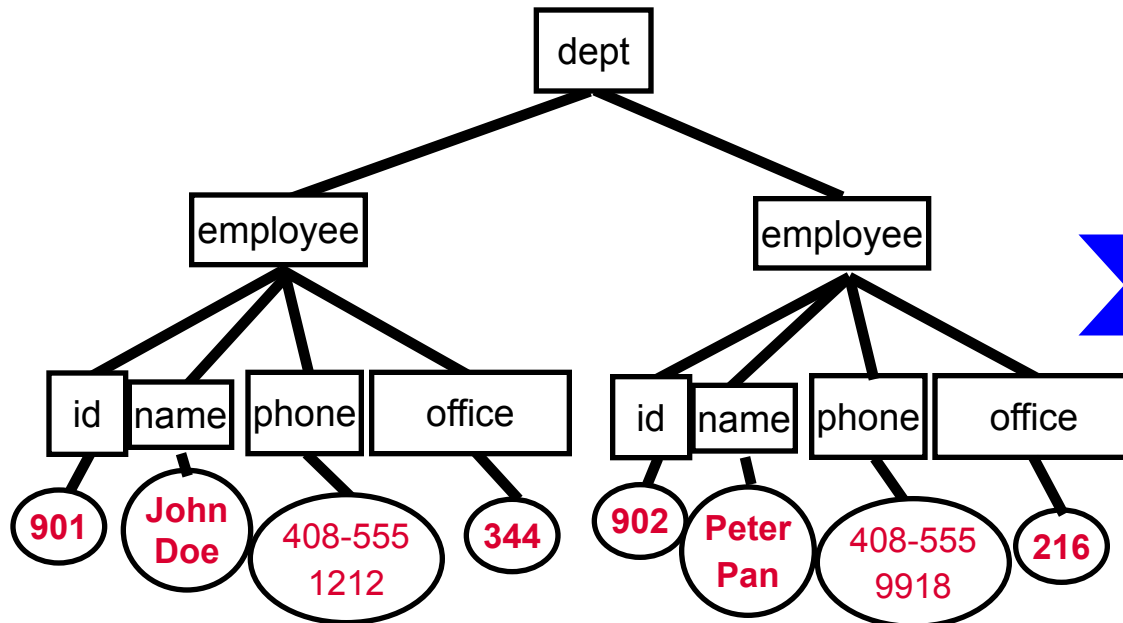


# Разбор документа

XML Documents парсится в соответствии с XQuery Data Model

Создаются дополнительная информация в каталоге

- Строки и пути хранятся в каталоге DB2
- Компрессия строк сохраняет место



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

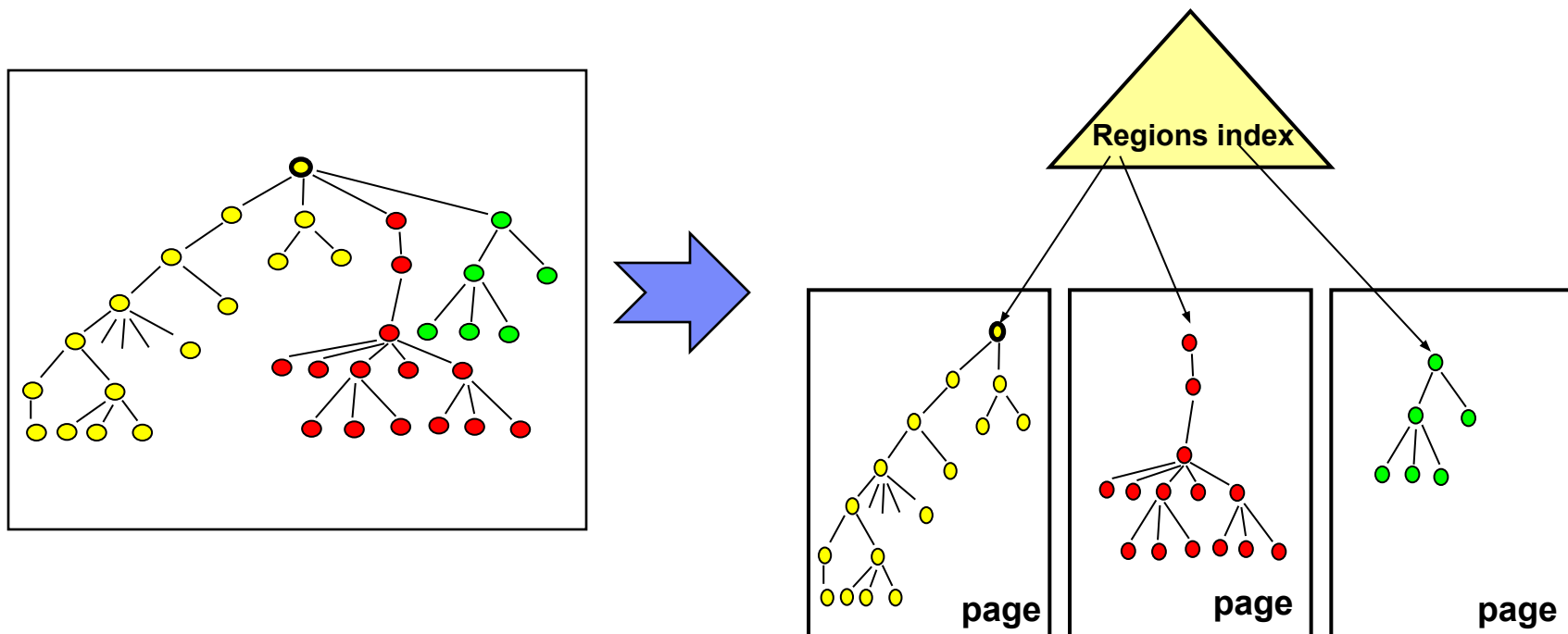
SYSIBM.SYSXMLSTRINGS

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

SYSIBM.SYSXMLPATHS

# Хранение документа

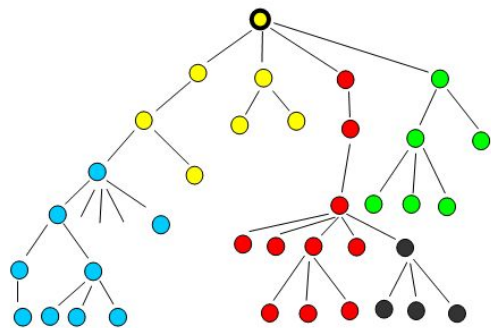
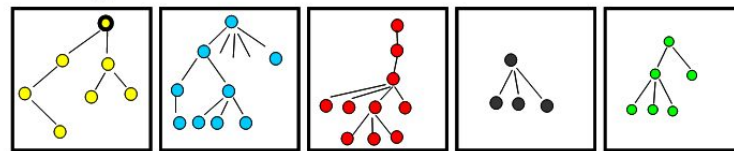
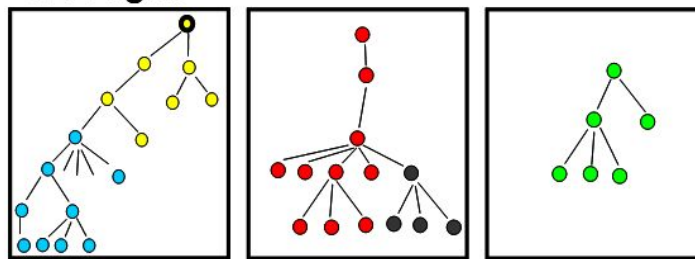
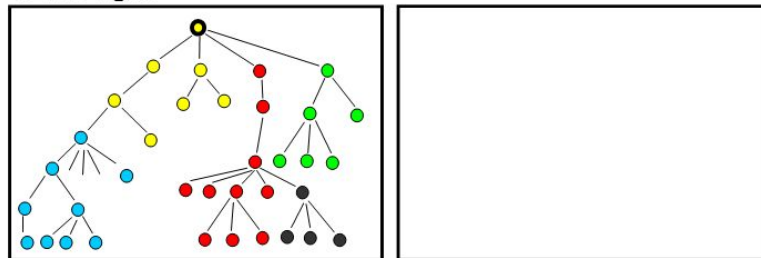
- Иерархия узлов XML Документа хранится на страницах DB2
- Если документ не помещается на одну страницу он разбивается на страницы/регионы
- Управляемый системой индекс регионов указывает на части документа





# Размер страницы для хранения XML

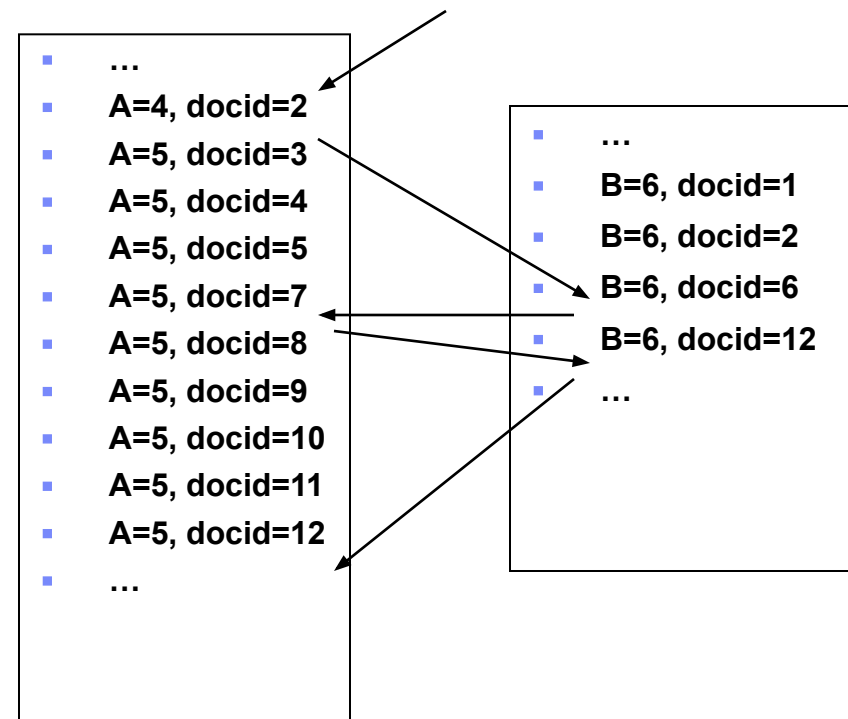
- Большие документы разбиваются на регионы
- Максимальный размер документа 2GB

**4K Pages****8K Pages****32k Pages**

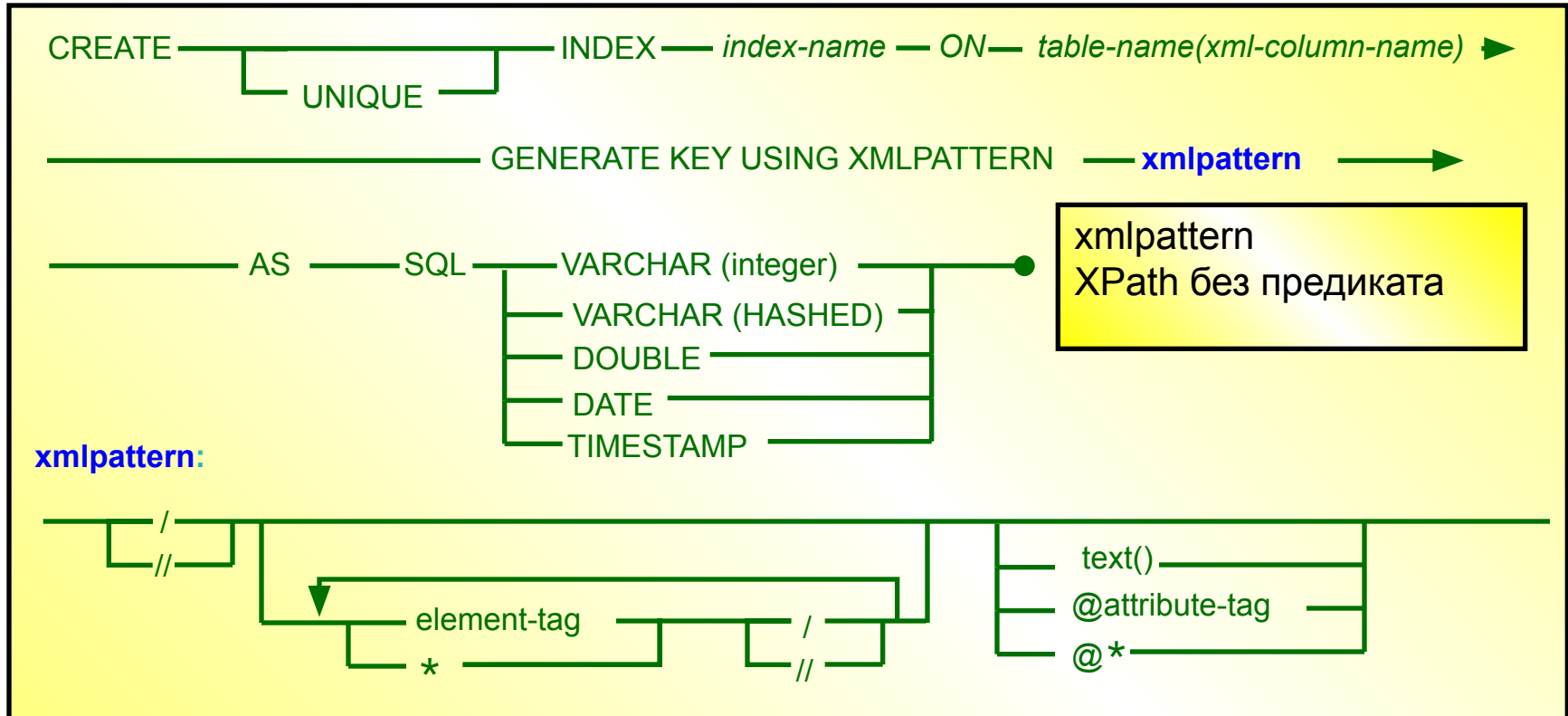
# Индексирование

- В-tree индексы расширены для управления XML документами
- Дополнительные поля
  - 4 PATHID: путь к значению
  - 4 VALUE: индексированное значение
  - 4 DOCID: Идентификатор документа
  - 4 NODEID: узел в котором хранится значение
- Дополнительные техники при поиске по индексам
  - 4 Pivot Join (Zig-Zag join)
  - 4 Конкурентная проверка предикатов “and” и “or”
  - 4 Множество курсоров на индексе каждый проверяет предикат

Find docids where **x/y[A=5 and B=6]**



# Индексирование



**create index idx1 on dept(deptdoc) generate key using xmlpattern '/dept/employee/name' as sql varchar(35)**

## Document Retrieval using SQL

- Retrieve XML documents

Select i, **deptdoc** from dept

- Retrieve some XML documents based on SQL Predicate

Select deptnum, **deptdoc** from dept

**Where deptnum like 'PR%'**



# Поиск по документу используя XQuery

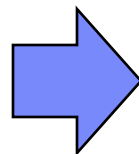
- Полная поддержка XQuery и XPath 2.0
  - 4 Включая FLWOR встроенный (nested) FLWOR
- XQuery по всем документам в столбце
  - 4 FOR \$d in **db2-fn:xmlcolumn**('dept.deptdoc')...
- XQuery по документам используя SQL предикаты
  - 4 FOR \$d in **db2-fn:sqlquery**("select deptdoc from dept where deptID LIKE 'PR%' ")...



# Выражение FLWOR

- **F**OR: проходит по последовательности документов,
- **L**ET: привязка переменных к элементам
- **W**HERE: фильтрация элементов итерации
- **O**RDER: перегруппирует элементы итерации
- **R**ETURN: конструирует результаты запроса

```
FOR $movie in
db2-fn:xmlcolumn('table1.movies')
LET $actors := $movie//actor
WHERE $movie/duration > 90
ORDER by $movie/@year
RETURN <movie>
      {$movie/title, $actors}
      </movie>
```



```
<movie>
  <title>Chicago</title>
  <actor>Renee Zellweger</actor>
  <actor>Richard Gere</actor>
  <actor>Catherine Zeta-Jones</actor>
</movie>
```

# Выражение FLWOR

dept

xml,

```
for $d in xmlcolumn('deptdoc')/dept
where $d/@bldg = 101
return <namelist>
    {$d/employee/name}
</namelist>
```

```
<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>
```

```
for $d in xmlcolumn('deptdoc')/dept
where $d/@bldg = 101
return $d/employee/name
```

```
for $d in xmlcolumn('deptdoc')/dept
where $d/@bldg = 101
return $d/employee/name/text()
```

```
<name>John
Doe</name>
<name>Peter
Pan</name>
```

```
ohn
Doe</name>
<name>Peter
Pan
```

```
John
Doe
Peter
Pan
```

Результат не обязательно XML !

## XQuery с объединением

```
for $book in db2-fn:xmlcolumn('BOOKS')/book
  for $entry in db2-fn:xmlcolumn('REVIEWS')/entry
  where $book/title = $entry/title
  return
    <review>
      {$entry/review/text()}
    </review>;
```



# SQL/XML – Лучшее от общих от обоих миров





- Все возможности SQL для обработки реляционных столбцов
- Все возможности XQuery и XPath 2.0 для адресации и трансформации XML документов
- Объединение XML документов и таблиц
- Предикаты как SQL так и XML
- Создание XML из структурированных полей
- Материализация таблиц и представлений из XML документов



```
select d.deptID , u.headcount, xmlquery('$deptdoc/dept/name' passing d.deptdoc as "deptdoc")
  from dept d, unit u
  where d.deptID = u.unitID and u.headcount > 200
  and xmlquery('$deptdoc/dept/@bldg' passing d.deptdoc as "deptdoc") = u.bldg
  and xmlexists('$deptdoc/dept/employee/name' passing d.deptdoc as "deptdoc")
```

## SQL/XML

Dept

deptID	unitID	deptdoc
100	10	
110	10	
220	20	
240	20	
310	30	

```

<dept bldg="G">
  <name>Engineering</name>
  <manager>Anjul</manager>
  <backup>Susan</backup>
  <admin>Mary</admin>
  <chargecode>CW345</chargecode>

```

Unit

unitID	Empcount	bldg
10	234	G
20	123	H
30	32	H

```

select d.deptID, xmlquery('$deptdoc/dept/name' passing d.deptdoc as "deptdoc"), u.empcount
from dept d, unit u
where d.unitID = u.unitID and u.empcount > 200
and xmlquery('$deptdoc/dept/@bldg' passing d.deptdoc as "deptdoc") = u.bldg
and xmlexists('$deptdoc/dept/name' passing d.deptdoc as "deptdoc")

```

# SQL/XML – публикация XML документов

FIRSTNAME	LASTNAME	DEPARTMENT
SEAN	LEE	A00
MICHAEL	JOHNSON	B01
VINCENZO	BARELLI	A00



**SELECT**

```
XMLELEMENT (NAME "Department",  
XMLATTRIBUTES (e.department AS "name" ),  
XMLAGG ( XMLELEMENT(NAME "emp", e.firstname))) AS "department_list"
```

```
FROM employee e GROUP BY e.department;
```



```
department_list
```

```
<Department name="A00">  
  <emp>VINCENZO </emp>  
  <emp>SEAN</emp>  
</Department>  
<Department name="B01">  
  <emp>MICHAEL</emp>  
</Department>
```

# XMLTABLE - XML->relational

```

SELECT X.* from
XMLTABLE ('db2-fn:xmlcolumn("POORDERS.PO")//customer'
COLUMNS
  "CID"          INTEGER          PATH '@id',
  "Name"        VARCHAR(30)      PATH 'name',
  "ZipType"     CHAR(2)          PATH 'zip/@type',
  "Zip"         XML              PATH 'zip'
) AS "X"

```

CID	Name	ZipType	Zip
1325	Bobby	US	<zip>33129<zip>
4711	null	US	<zip>95023<zip>

# Функции публикации SQL/XML

- Скалярные функции
  - 4 **XMLELEMENT** – generate XML element
  - 4 **XMLATTRIBUTES** - used within XMLELEMENT, specifies attributes
  - 4 **XMLFOREST** - produces a forest of XML elements from SQL values
  - 4 **XMLCONCAT** - concatenates a variable number of XML values
  - 4 **XMLNAMESPACES** – produces a namespace declarations
- Функции агрегации
  - 4 **XMLAGG** - to group or aggregate XML data
- Табличные
  - 4 **XMLTABLE** – materializes a table from XML documents
- Функции преобразования типов
  - 4 **XMLCAST** - converts between XML data type and standard relational types
  - 4 **XMLSERIALIZE** – converts XML data type to serialized XML as a char/varchar/clob/blob

# JDBC API Enhancements for XML



- Поддержка нового XML типа
  - Поддержка нового интерфейса `com.ibm.db2.jcc.DB2Xml`
  - Поддержка XQuery
  - Поддержка регистрации XML Schema
- 4 Type Code `java.sql.Types.OTHER` до тех пор пока не появится новый тип JDBC `java.sql.Types.XML`

# Sample Java Program

create table dept (id char(8), doc xml);



```
PreparedStatement stmt1 = con.prepareStatement("Select doc from dept where id = '001' ");
ResultSet rs = stmt1.executeQuery();
rs.next();
```

```
String xmlString = rs.getString(1);
InputStream is = rs.getBinaryStream(1);
```

```
com.ibm.db2.jcc.DB2Xml xml = (com.ibm.db2.jcc.DB2Xml) rs.getObject (1);
String xmlString = xmlOut.getDB2String();
InputStream is = xmlOut.getDB2XmlBinaryStream("ISO-10646-UCS-2");
```

```
... = xmlOut.getDOM() or xmlOut.getSAX(); //future
```

```
PreparedStatement stmt2 = con.prepareStatement("update dept set doc = ? where id = '001'");
```

```
stmt2.setObject(1, xmlOut);
stmt2.setBinaryStream(1, new FileInputStream(file), (int)file.length());
stmt2.setString(1, xmlString);
```

```
stmt2.executeUpdate();
```

# Sample Java Program – XQuery



```
create table customer (customerinfo xml);
```

```
PreparedStatement stmt=null;
ResultSet rs;
String sqls = "XQUERY "+
    "for $info in
    db2-fn:sqlquery('SELECT info FROM
    CUSTOMER WHERE cid > ?') "+
    "where
    $info/*:customerinfo/*:addr[@country=\"E
    ngland\"] "+
    "return
    $info/*:customerinfo/*:phone/text()";
stmt = conn.prepareStatement(sqls);
stmt.setInt(1, cidToFilter);
rs = stmt.executeQuery();
```



# DB2.NET XML Support

- DB2 .NET Provider
- DB2Xml - Инкапсулирует тип XML
  - 4 Любой доступ к XML типу осуществляется через другие объекты основанные на XML (XmlReader, XPathDocument)
  - 4 Методы доступа к XML столбцам
    - DB2Xml.GetXmlReader
    - DB2Xml.GetString
    - DB2Xml.GetBytes
- DB2DataReader
- DB2Command
- XML Input и Output Parameters
  - 4 DB2Type.Xml
- Поддержка XQuery
  - 4 DB2XmlCommand
  - 4 DB2XmlAdapter
- Поддержка XmlSchema

# Sample .NET Program - XQuery

create table **employee** (**empinfo XML**)

```
DB2XmlCommand xcmd = new DB2XmlCommand();  
//Retrieve the name of all employees in department #100  
xcmd.CommandText =  
“for $e in db2-fn:xmlcolumn ('EMPLOYEE.EMPINFO)/employee  
where $e/deptno = 100  
return {$b/name}”  
  
//Set the root tag  
xcmd.RootTag = “deptlist”;  
  
//Retrieve the result of the xquery expression as an XmlReader  
//the result will be wrapped with <deptlist> </deptlist>  
XmlReader xrdr = xcmd.ExecuteXmlReader();  
  
//Retrieve the result of the xquery expression as a Stream  
Stream xmlstream = xcmd.ExecuteStream()
```

# Проверка с помощью XML Schemas

- Проверка опциональна и осуществляется на уровне д
- 4 Без проверки
  - `insert into dept(deptdoc) values (?)`
- 4 С проверкой
  - `insert into dept(deptdoc) values (xmlvalidate(?))`
- Схема может быть перезаписана и указывать на схему в репозитории DB2
  - 4 `insert into dept(deptdoc) values (xmlvalidate(? according to xmlschema id "ibm.invoice"))`
  - 4 `insert into dept(deptdoc) values (xmlvalidate(? according to xmlschema uri 'http://my.world.com'))`



# Управление XML Schema

- Репозиторий XML Schema (XSR)
  - 4 Хранит зарегистрированные схемы
  - 4 Управляется как часть каталога DB2
  - 4 Таблицы и представления создаются автоматически
    - SYSCAT.XSROBJECTS, SYSCAT.XSROBJECTCOMPONENTS
    - SYSCAT.XSROBJECTAUTH, SYSCAT.XSROBJECTHIERARCHIES
  - 4 Интерфейс командной строки, API, Хранимые процедуры

	Хранимые процедуры
Регистрация	XSR_REGISTER( <i>rschema, name, schemalocation, content, docproperty</i> )
Add	XSR_ADDSCHEMADOC( <i>rschema, name, schemalocation, content, docproperty</i> )
Complete	XSR_COMPLETE( <i>rschema, name, schemaproperties, isusedforshred</i> )

# Аннотация схемы для Нарезки

- Отображение из XML в реляционные таблицы
  - 4 Отображение контролируется аннотациями XML в DB2 XSR

Аннотация	Цель
<b>db2-xdb:defaultSQLSchema</b>	The default schema for the table names
<b>db2-xdb:rowSet</b>	The table name the element/attribute should be mapped to
<b>db2-xdb:column</b>	The column name of the table the item should be mapped to
<b>db2-xdb:contentHandling</b>	Text value, string value or xml fragment to be treated as the content to be inserted into the database
<b>db2-xdb:truncate</b>	Truncate content if the size is greater than the specified size in the database
<b>db2-xdb:normalization</b>	White space treatment
<b>db2-xdb:expression</b>	Custom expressions to apply to the data before insert
<b>db2-xdb:condition</b>	Conditions to apply to rows before insert
<b>db2-xdb:tableMapping</b>	Multiple mappings, to same or different tables, for an element/attribute
<b>db2-xdb:dependence</b>	Referential constraints

```

<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

```

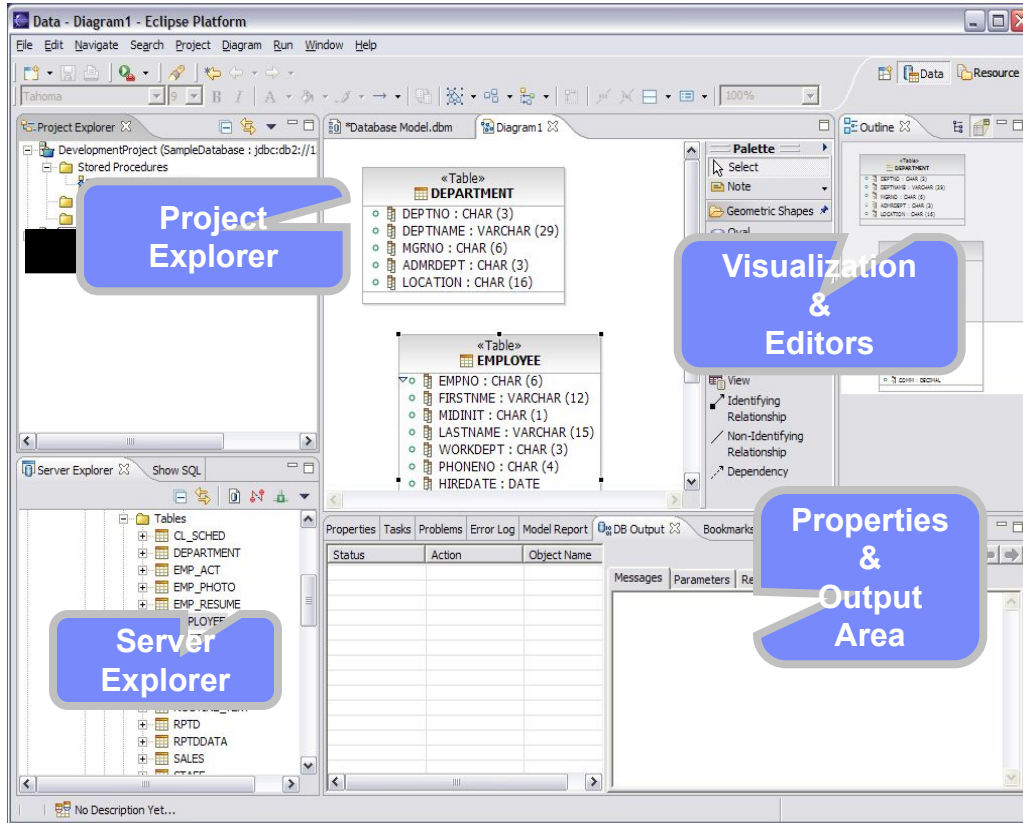
...<sequence>
<element name="productName" type="string"/>
<element name="quantity"
type="ipo:derivedPositiveIntegerType"
db2-xdb:rowSet="ELECTRONICITEMS"
db2-xdb:column="QTY"/>
.....<!-- ignoring mapping of PRICE --> ....
  <attribute name="partNum" type="ipo:SKU"
db2-xdb:rowSet="ELECTRONICITEMS"
db2-xdb:column="PARTNUM"
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>
    
```



# DB2 Development Workbench



- Eclipse based
- Support for XML type in tables, views, SPs and UDFs
- Support for XML Index
- Support for XML type in SQL Query Builder
- Graphical XML Query Builder
- XSD (XML Schema) Editor
- XML Editor
- XML viewer
- Support for XML Schema registration
- XML & XSL Parsers
- XML<->Relational mapping

# XQuery Builder

The screenshot displays the Eclipse XQuery Builder interface. The main window is titled "Resource - My1.xquery - Eclipse Platform". It features a Navigator on the left showing a project structure with files like "aaa.xq", "karen.xquery", "My1.xquery", and "My1.xquery". The central area is divided into several panes:

- Documents:** Shows the XML document structure for "C:\corona\xquery\customer.xml", including elements like #document, customerinfo, name, addr, and phone.
- XQuery:** A graphical editor for building XQuery statements. It includes sections for:
  - For each:** A table for defining iterations over sequences.
  - Let:** A section for variable declarations.
  - where:** A table for defining filter conditions (Operand 1, Operator, Operand 2, Co...).
  - Order By:** A table for defining sorting criteria (Value- node, expression, function, variable, Direction).
- Functions:** A list of available XQuery functions like fn:string, fn:compare, etc.
- Outline:** A tree view of the query's logical structure.

At the bottom, there is a "Design" and "Source" view. The "Source" view shows the XQuery code generated from the graphical editor:

```

for $customerinfo in db2-fn:xmlcolumn("schema.table.column")/custo
where /customerinfo/@Cid=1000
order by /customerinfo/name ascending
return
(
  for $addr in $customerinfo/addr
  where $addr/@country="us"
  return (
    ($addr/street) ,
    ($addr/city)
  )
)

```

- Graphical XQuery builder
- Supports selecting from actual documents to build query
- Can view in progress XQuery statement
- Save Statements for re-use
- Execute and view results



# MS Visual Studio .NET – DB2 XML add-in

**.NET**

The screenshot displays the Microsoft Visual Studio .NET environment with the DB2 XML add-in. The interface includes a menu bar (File, Edit, View, Project, Build, Debug, Data, Tools, Window, Help), a toolbar, and several panes:

- Server Explorer:** Shows a data connection to 'hoymich@SAMPLE [DB2/NT 08.C]' with a tree view of tables (DEPARTMENT, EMPLOYEE, etc.), views, and procedures.
- Design View:** Displays two data adapters: 'TBDEPTTableAdapter' (with fields DEPTID, DEPTNAME, TIME1) and 'TBEMPLOYEETableAdapter' (with fields EMPID, DEPTID, VARCHAR1, CLOB1, DATE1, TIMESTAMP1). A relationship line labeled 'FK\_TBDEPT\_TBEMPLOYEE' connects the 'DEPTID' field of TBDEPT to the 'DEPTID' field of TBEMPLOYEE.
- Solution Explorer:** Shows the project structure for 'WindowsApplication4', including 'Properties', 'References', 'app.config', 'Form1.cs', 'Program.cs', and 'sampleDataSet.xsd'.
- Properties Window:** Shows details for 'HOYMICH.DEPARTMENT', including 'Colcount: 5', 'Name: DEPARTMENT', 'Schema: HOYMICH', and 'Type: T'.

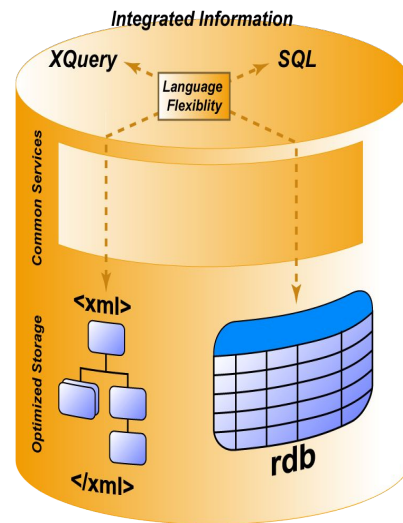
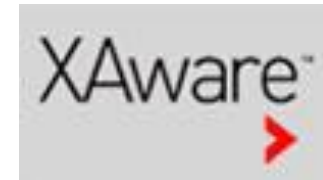
- Server Explorer
- XML Editor
- XML Schema Editor
- XSR Registration
- XML Index Builder

# DB2 Viper в Storebrand



- Лидирующий игрок, на рынке страхования здоровья и жизни, управления активами и ценными бумагами в Норвегии.
- Пионеры в адаптации SOA, Web Services и XML в информационных системах
- Senior Enterprise Architect Thore Thomassen тесно работал IBM на протяжении Viper Alpha
- Преимущество Native XML в сравнительных тестах
  - 4 Сокращение времени разработки внутренних отчетов с более чем 1 day до 10 минут
  - 4 Сокращение составляющей I/O для Web services в среднем на ан 65% и сокращение времени поддержки на 20%
  - 4 Реализация изменения схемы до нескольких минут с целого дня прототипирования и тестирования и целой недели внедрения
  - 4 Реализация поиска и извлечения документов в соответствии с новыми требованиями в 30 минут, ранее требовалось 2 часа with decomposition and 8 hours with CLOB.

# Partner Solutions



## Немного о других НОВЫХ ВОЗМОЖНОСТЯХ

- STTM – самонастраиваемая и конфигурируемая память DBMS
- LBAC – Ограничение доступа на уровне строк
- Hybrid Partitioning
- Компрессия данных
- Улучшения в Резервном копировании etc...



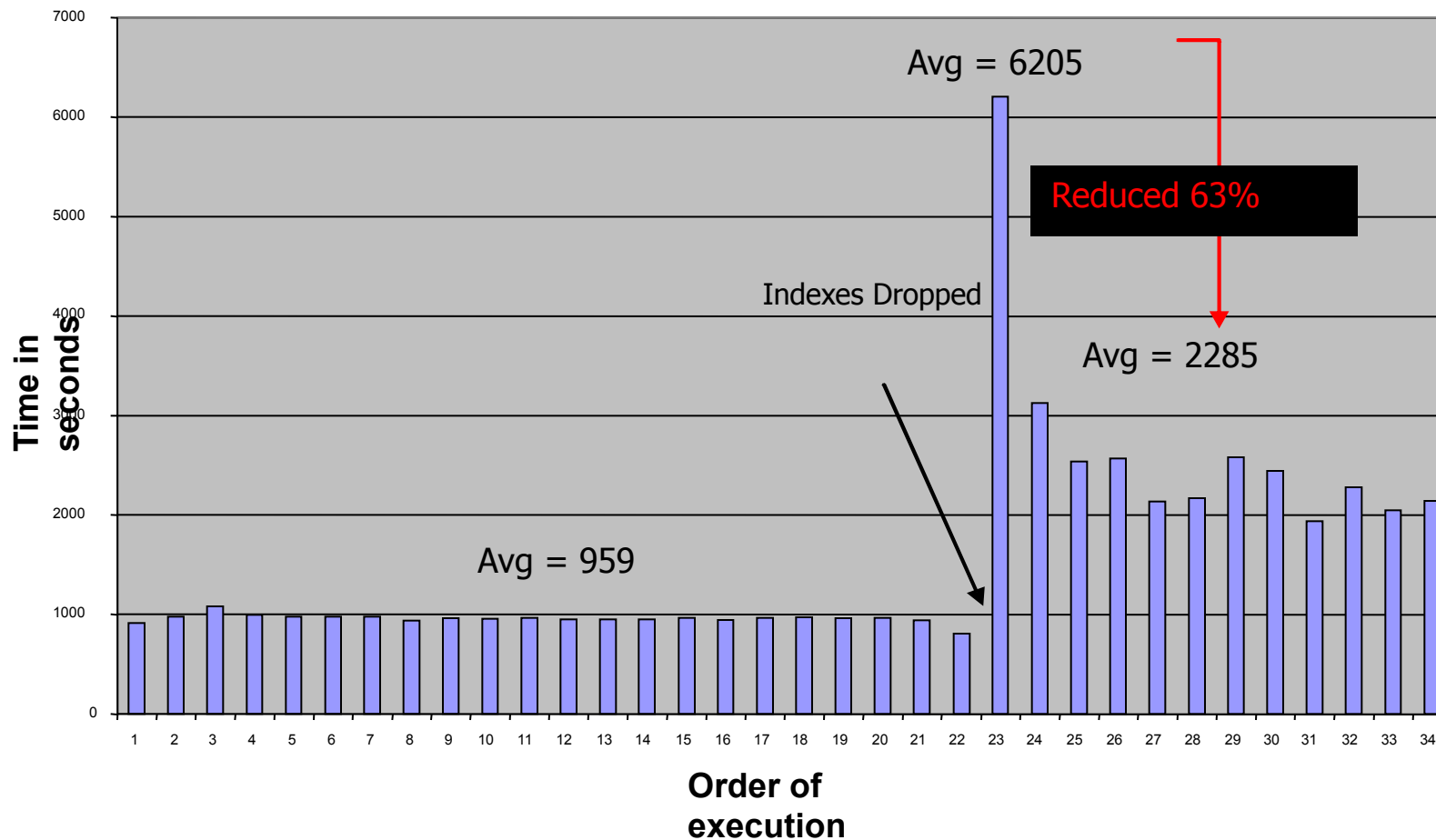
# Автоматизация автоматически!

- Включение множества автономных возможностей по умолчанию.
- Примеры:
  - 4 Configuration Advisor (2 second tuning)
  - 4 Adaptive Self Tuning Memory
  - 4 Автоматический сбор статистики.
- Автоматическое вычисление параметров I/O
  - 4 Значения вычисляются во время запуска
    - Основываются на кол-ве CPU расположении дисков



# STMM в действии – Удаление важного индекса

## TPCH Query 21 - After drop index - Average times for the 10 streams

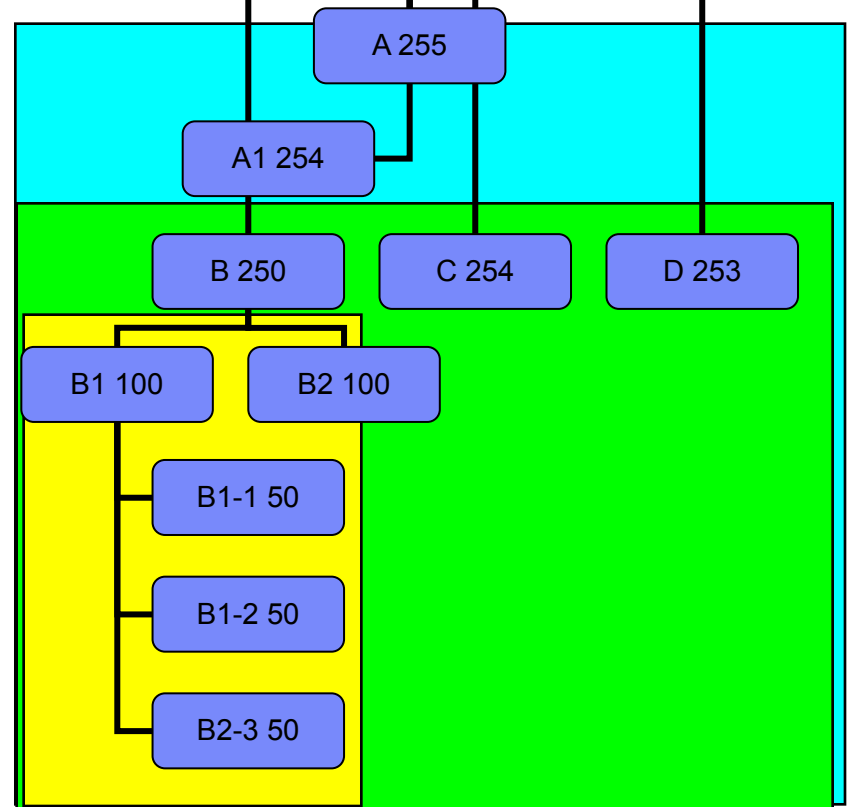
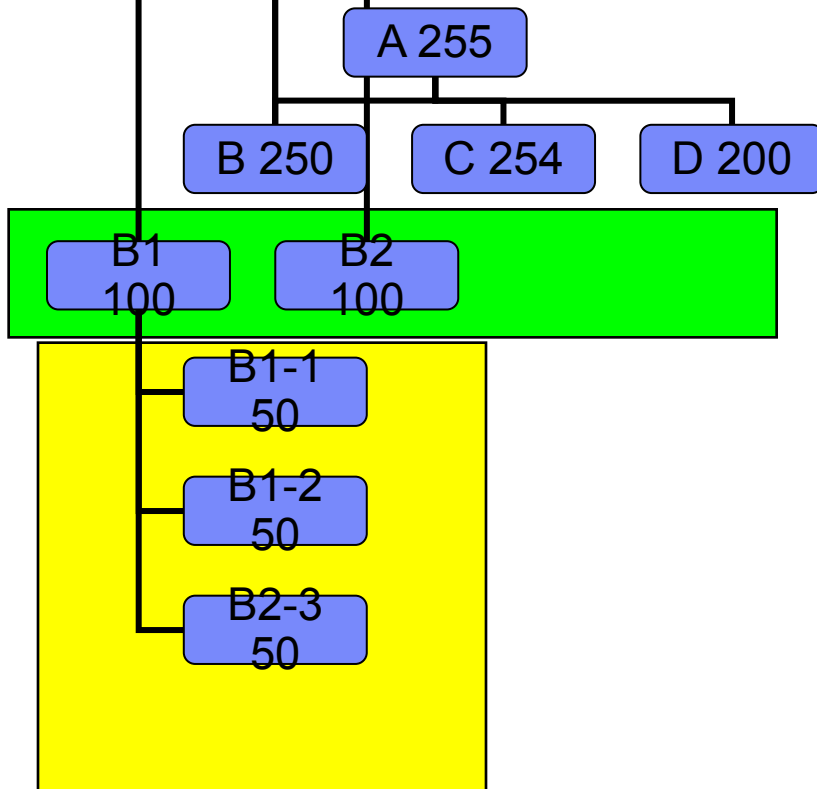


# Security - Label Based Access Control

- Label Based Access Control (LBAC)
  - 4 “label” ассоциирован как с пользовательской сессией так и со строками
  - 4 Правила (Rules) проверяют метки пользователей и строк для выявления прав
- Labels могут состоять из множества компонентов
  - 4 Иерархия, группа и массив
  - 4 Метка на строку есть дополнительный вне зависимости от количества компонентов метки
  - 4 Пользовательские метки выдаются security administrator
- Похожа на поддержку LBAC в DB2 для z/OS v8

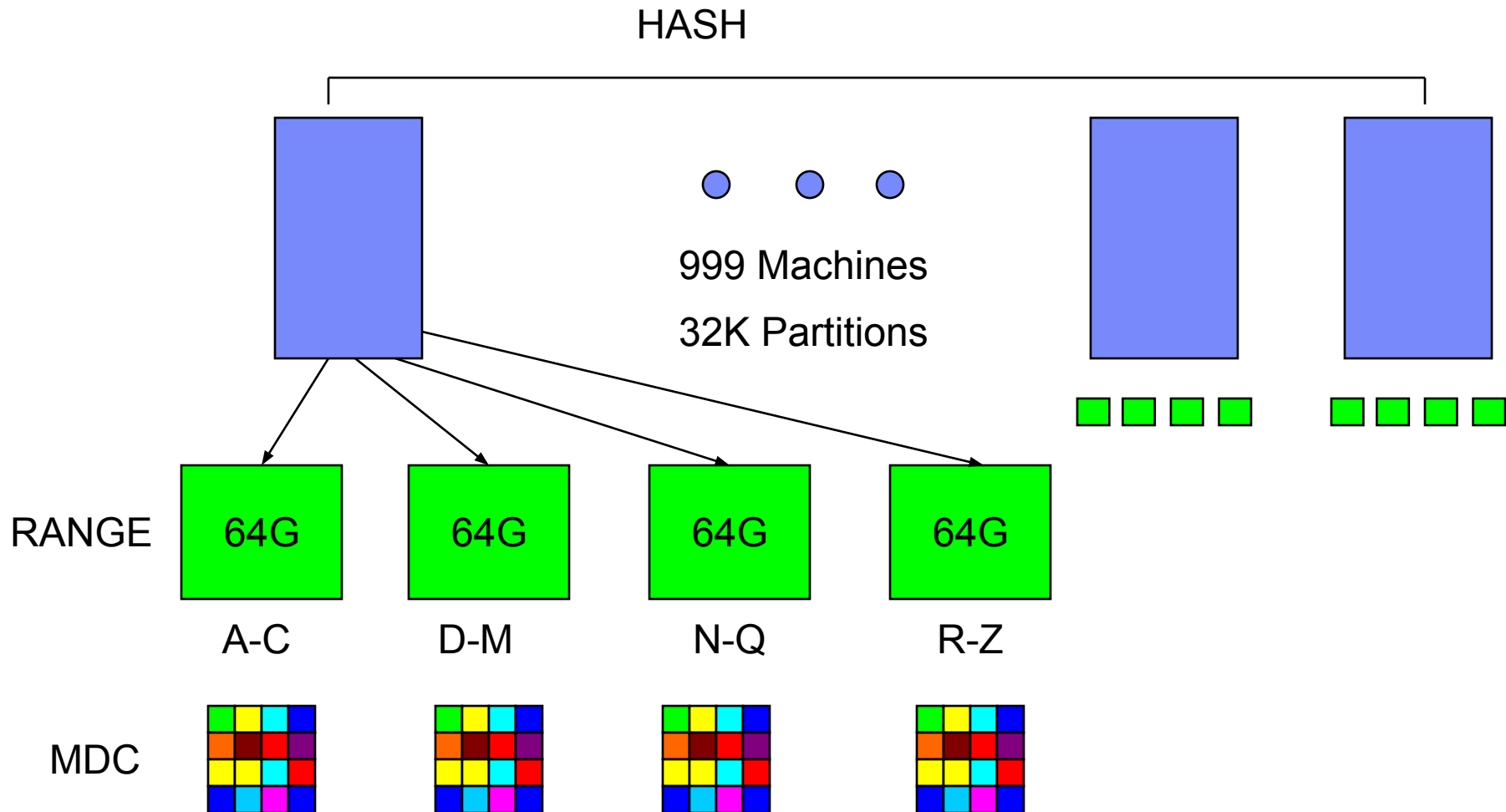


# LBAC Hierarchy Update/Read



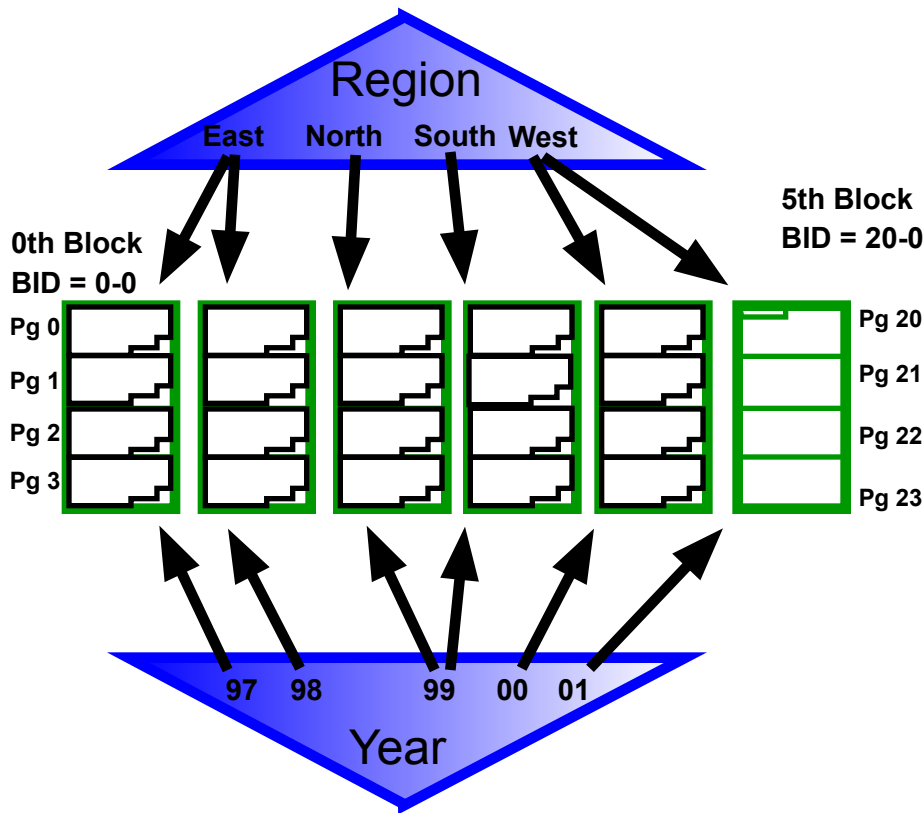
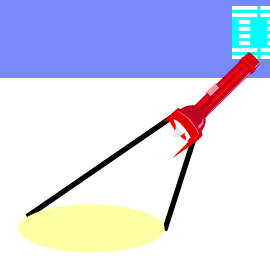


# Hybrid Partitioning



# MDC изнутри

## Примеры обработки запросов



### Block INDEX ANDing

```
CREATE TABLE
  SALES (Customer VARCHAR(80),
        Region CHAR(5),
        Year INT)
  ORGANIZE BY DIMENSIONS (Region, Year)
```

```
SELECT * FROM SALES WHERE
  Region = 'West' AND Year = 00
```

- 'West' BIDs: 

16-0, 20-0
------------
- '00 BIDs: 

16-0
------
- Result of AND: 

16-0
------
- Retrieve block 16-0 in one I/O
- Mini-relation scan retrieves all records in block

### "AND"ing RID and Block INDEXes

```
CREATE INDEX i1 ON SALES (Customer)

SELECT * FROM SALES WHERE
  Region='East' AND Customer='Joe'
```

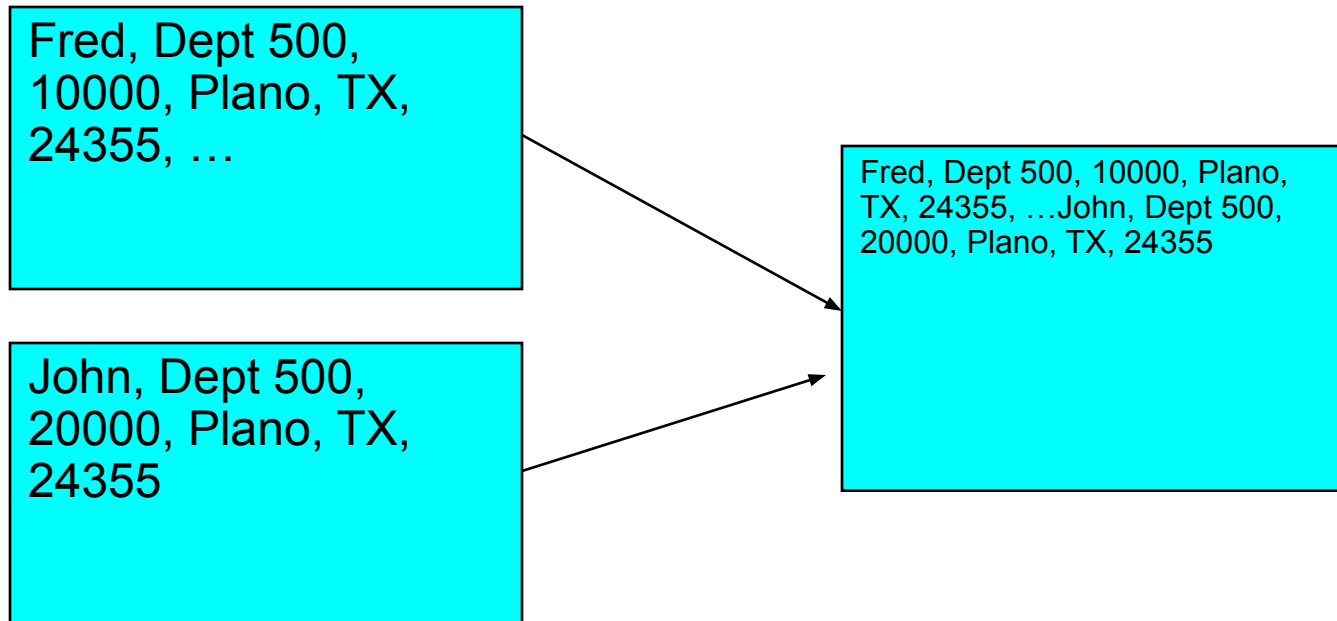
- 'East' BIDs: 0-0, 4-0 

--
- 'Joe' RIDs: 0-3, 0-211, 2-97, 7-1, 11-33 

--
- Filter out RIDs not in page range indicated by BIDs 

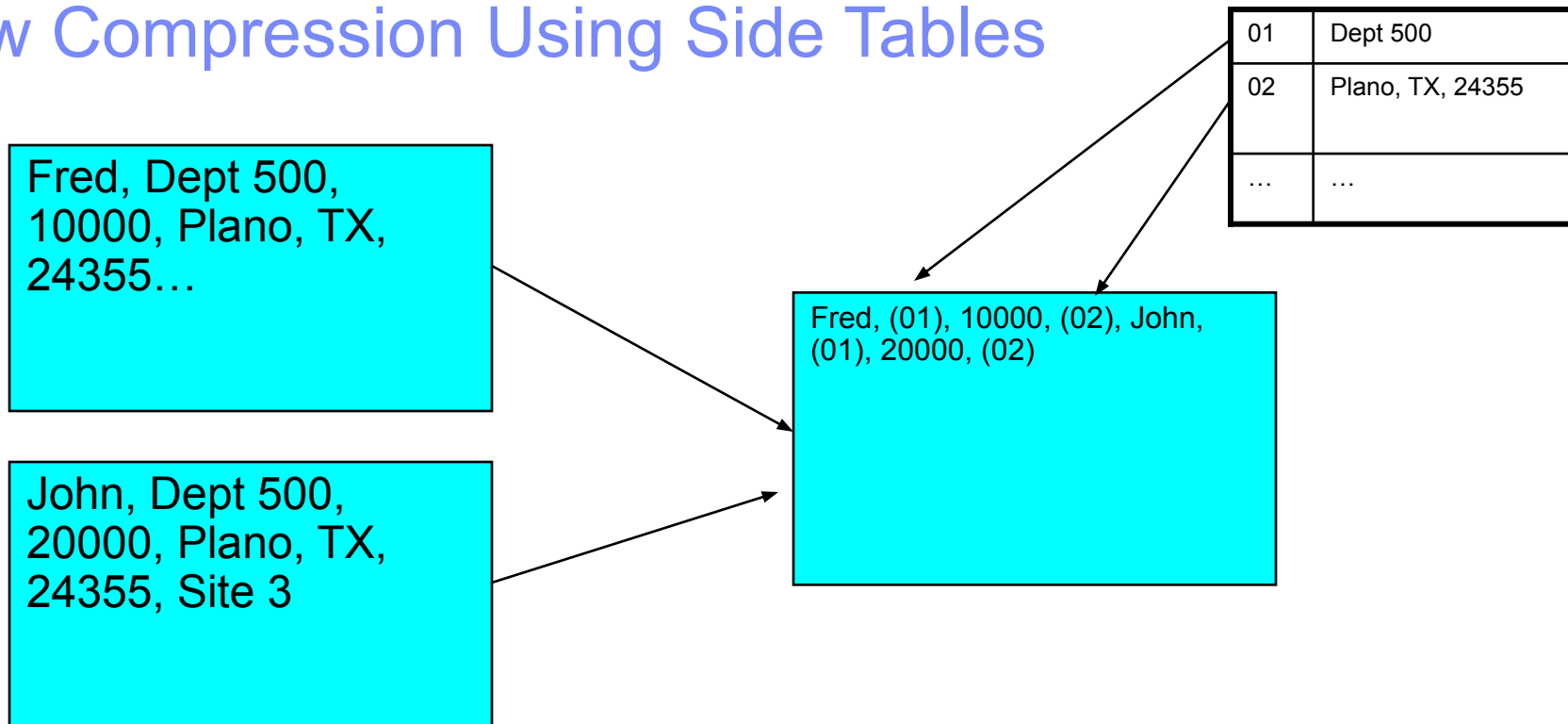
0-3, 0-211, 2-97, 7-1
-----------------------
- Directly fetch those records

## Row Compression используется LZV



Компрессия множества строк в одну страницу не эффективно с точки зрения БД.

# Row Compression Using Side Tables



Сторонние таблицы содержат повторяющуюся информацию из строк.

# DB2 - More Compression Ratios (Customer Data)

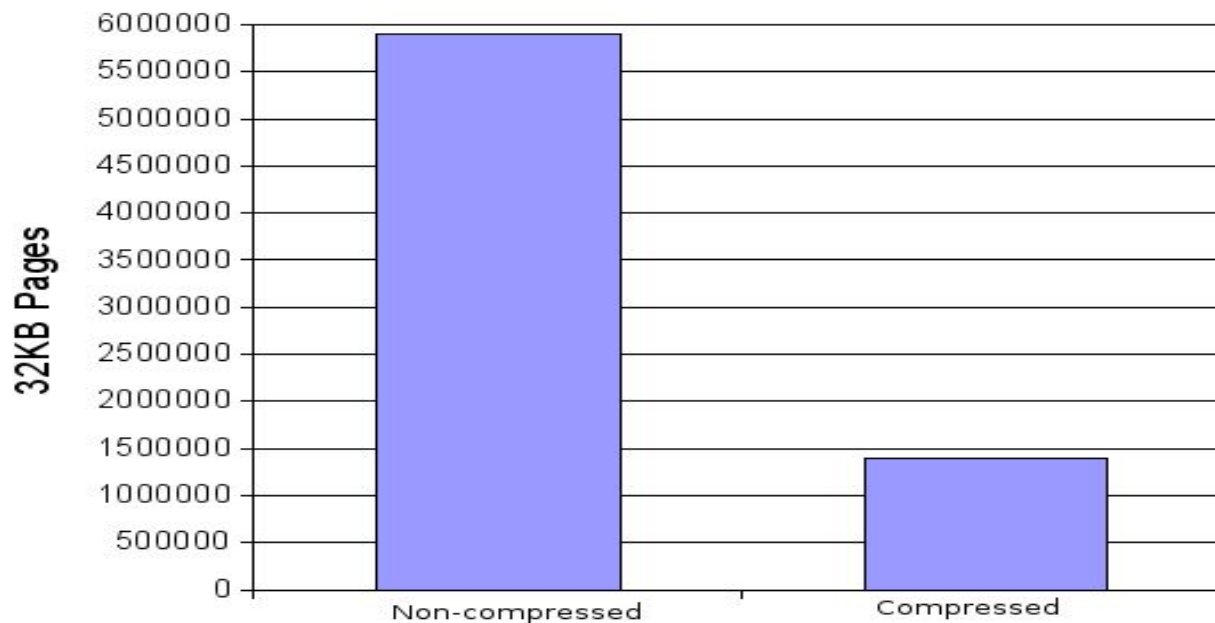
## Compression Type 32KB Page Count Space Required on Disk

No compression 5893888 179.9GB

Row compression 1392446 42.5GB

% Pages Saved: 76.4%

## T1 Compression - 179.9GB Initial Size





Information Management



# Informix Dynamic Server 10

IBM Software Group



# IBM Informix Dynamic Server Roadmap

IDS  
v9.40

IDS  
v10

IDS  
v10.5

IDS  
vNext

## Март 2003

- Производительность
- Backup & Restore
- HDR & ER сосуществование
- Удаление ограничений на размеры.
- Мониторинг, Утилиты

## Q1 2005

- Безопасность
- Устойчивость/Высокая доступность
- Разработка Соответствие стандартам
- Автономность/ Простота администрирования
- Интеграция с IBM SWG
- Запросы клиентов и партнеров

## 2H 2006

- Запросы рынка, Technology & заказчиков партнеров
- Интеграция с IBM Software
- Автономность (Online Table Reorg, GUI, SQL)

## ▪2008

- Запросы рынка, Technology & заказчиков партнеров
- Интеграция с IBM Software
- Улучшения Автономности
- MDC
- MQT

2003

2004

2005

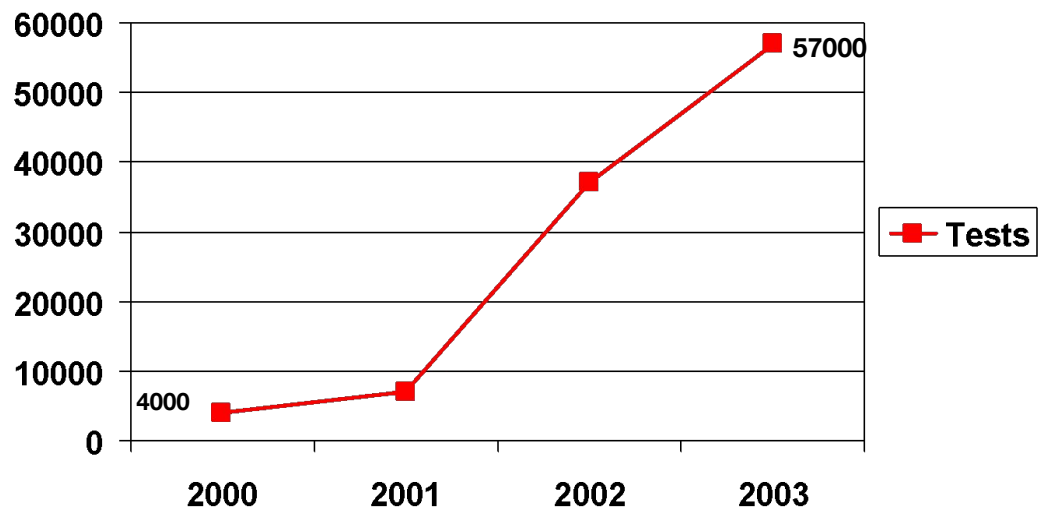
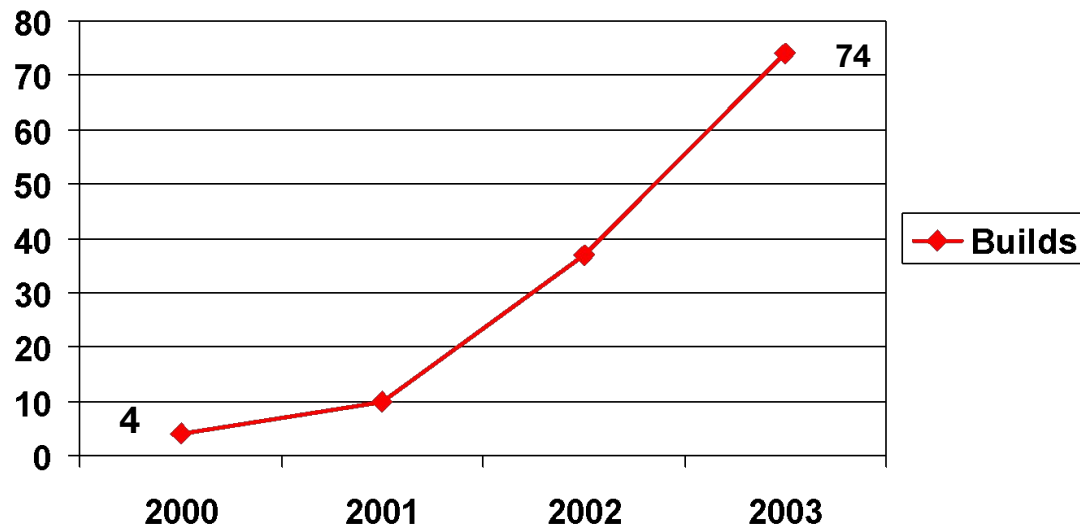
2006

2007

2008



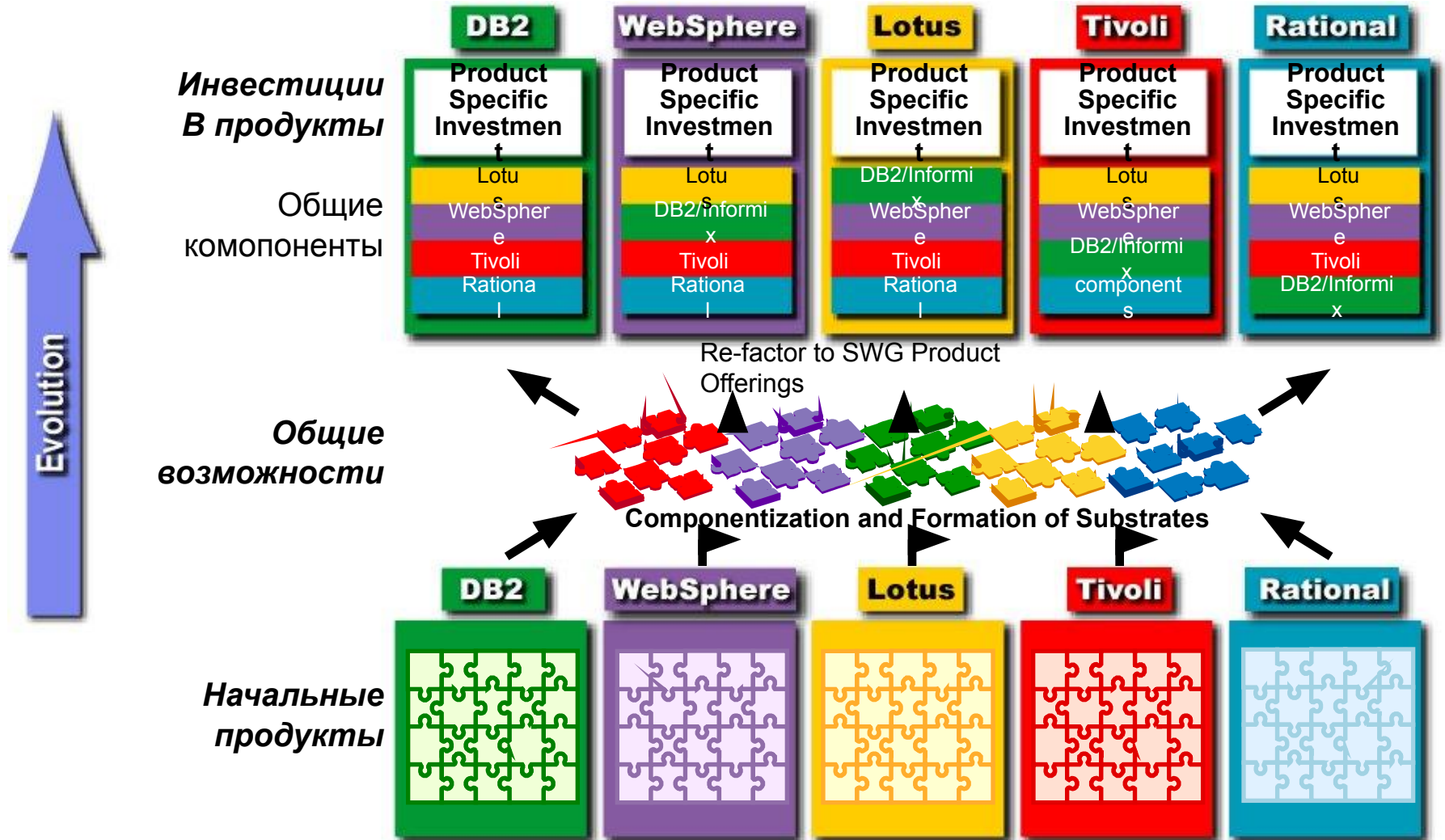
# Increase in Nightly Builds and Tests (IDS(4), CSDK, GLS, 4GL)







# IBM Software Portfolio – использование компонентов



Thank  
YOU

