Creating Structure in Unstructured Data

What is possible, today...?
Hotsos Revisited 2013

Van 3 tot en met 7 maart vindt in Irving, Texas, het internationale Oracle Performance Symposium Hotsos plaats. Dit jaar belooft het symposium een garantie op inhoudelijk hoogstaande presentaties en discussies, want naast presentaties van Tom Kyte, Cary Millsap, Maria Colgan en Steven Feuerstein over performance, worden er ook onderwerpen behandeld zoals Big Data, noSQL, XML, Statistische toepassingen met betrekking tot performance, beheer in de (Oracle) Cloud, Exadata en Oracle 12c onderwerpen.

Dit jaar is het vijf Nederlanders gelukt om mee te mogen doen en door de zware abstract criteria heen te komen. De heren Toon Koppelaars, Gerwin Hendriksen, Jacco Landlust, Frits Hoogland en Marco Graalke hebben niet alleen het genoegen om ter plekke te zijn, maar geven ook zelf een presentatie over hun favoriete onderwerpen.

In alfabetische volgorde:

- Marco Graalke - Creating Structure in Unstructured Data
- Gerwin Hendriksen - "Method GAPP" Used to Mine OEM 12c Repository and AWR Data
- Frits Hoogland - About Multiblock Read
- Toon Koppelaars - SQL Plan Management
- Jacco Landlust - Lessons Learned while Pushing the Limits of SecureFiles

Op dinsdagavond 2 april organiseert AMIS 'Hotsos Revisited 2013' waarin de bovenstaande Nederlandse sprekers hun gegeven presentaties herhalen. Voor een terugblik van voorgaande jaren, zie ook bijvoorbeeld de hieronder vermeldde AMIS Technology Blog artikelen of "Hotsos Revisited, Revisited".

Je bent vanaf 16.30 uur van harte welkom. Als altijd is het diner en de drankjes gratis.

Voor het aanmelden voor deze avond, gebruiken de volgende link: "Hotsos Revisited 2013"

Related posts:
Hotsos Revisited 2013

Van 3 tot en met 7 maart vindt in Irving, Texas, het internationale Oracle Performance Symposium Hotsos plaats. Dit jaar belooft het symposium een garantstelling voor inhoudelijk hoogstaande presentaties en discussies, want naast presentaties van Tom Kyte, Cary Millsap, Maria Colgan en Steven Feuerstein over performance, worden er ook onderwerpen behandeld zoals Big Data, noSQL, ...

Tags: Frits Hoogland, genot en bijdragen, hotsos, hotsos 2013, Jacco Landlust, Marco Gralike, Toon Coppelaers

Basisregistraties Adressen en Gebouwen - Het importeren van Kadaster BAG data in een Oracle Database

Vorig jaar heb ik behoorlijk wat vragen gekregen over of er een tool was of een methodiek om BAG data van het Nederlandse Kadaster in een Oracle database te krijgen voor allerlei doeleinden. BAG data (Basisregistraties Adressen en Gebouwen) wordt, zo ver ik weet, onder andere uitgeleverd door het Kadaster in XML bestanden waarin alle...
“Big Data” = XML?
Challenges are!
Ahum, the problems are!
WikiPedia

- One string of XML data with structured and unstructured data sections
- Language: English
- Size: 42,15 GB
- Pages: 12,961,997
- Date: 21 Dec 2012
Adventures into the unknown...?
Setup

• VirtualBox VM
  – OEL 5U8 (64)
  – 8 GB RAM
• LaCie Little Big Disk
  – RAID 0
  – Thunderbolt
• Database
  – SGA 4GB
  – PGA 2GB
My new LaCie LBD is really fast - 😊
Defeat?! - 1.000.000 pages only
Status of Technology used
XML - Where are we...?
Achieved...?

XPath, XQuery, XSLT

XQuery 1.0 → XQuery 3.0
XPath Full Text
XPath 1.0 → XPath 2.0 → XPath 3.0

XSLT 1.0 → XSLT 2.0 → XSLT 3.0

1999 → 2010 → 2011 → Now

exerts
uses
extends
XQuery Update
On the Horizon!

• **JSoniq**
• **Zorba**
Building (streaming) Bridges

XQuery
Uniform Interface

Oracle
Content Management
MongoDB

CVS
PDF
Oracle XML DB

- **NO** cost option
- **C** (native / embedded kernel)
- (XQuery) Standards
- Code maintained by Oracle
XMLType Abstraction

DB XQuery

XQuery Rewrite

Procedural XQuery

XVM
(use “no query rewrite”)

SQL Execution

Relational Access Methods

Streaming XPath Evaluation

XMLIndex

DOM Tree Model

Object-Relational

Relational Storage

Binary XML

Secure Files

Source: S317428: Building Really Scalable XML Applications with Oracle XML DB and Oracle Text
So about what are we talking?
Andrea Andreani (1540–1623) was an Italian engraver on wood, who was among the first printmakers in Italy to use chiaroscuro, which required multiple colours.

Born and generally active in Mantua about 1540 (Brullot says 1560) and died at Rome in 1623. His engravings are scarce and valuable, and are chiefly copies of Mantegna, Albrecht Dürer, Parmigianino and Titian. The most remarkable of his works are Mercury and Ignorance, the Deluge, Pharaoh's Host Drowned in the Red Sea (after Titan), the Triumph of Caesar (after Mantegna), and Christ retiring from the judgment-seat of Pilate after a relief by Giambologna. He was active 1584-1610 in Florence.[1]

References

1. ^ ULAN
   • This article incorporates text from a publication now in the public domain: Chisholm, Hugh, ed. (1911). *Encyclopedia Britannica* (11th ed.). Cambridge University Press.
   • Getty ULAN entry.
   • artnet.

Rate this page

What's this?

- [ ] Trustworthy
- [ ] Objective
- [ ] Complete
- [ ] Well-written

- [ ] I am highly knowledgeable about this topic (optional)

Submit ratings

Categories: 1540 births | 1623 deaths | Italian engravers | People from the Province of Mantua
Andrea Andrea (1540-1623) was an Italian engraver on wood, who was among the first printmakers in Italy to use chiaroscuro, which required multiple colours. Born and generally active in Mantua, approximately 1540 (Bruliot says 1560) and died in Rome in 1623. His engravings are scarce and valuable, and are often copies of Andrea Mantegna, Albrecht Dürer, Parmigianino, and Titian. The most remarkable of his works are Mercury and Ignorance, Host Drowned in the Red Sea, Triumph of Caesar, Hostess retiring from the judgment-seat of Pilate, after Titian, and after Mantegna, and after Christ retiring from the judgment-seat of Pilate. He was active 1584-1610 in Florence.
Born and generally active in [[Mantua]] about 1540 (Brulliot says 1560) and died at [[Rome]] in 1623. His engraving style was influenced by the work of [[Sebastiano Serlio]], and he collaborated with [[Benvenuto Cellini]] on the execution of the bronze door of the [[Sant'Agnese in Agone]]. Andreani's engravings are notable for their fine detail and intricate compositions, which were often based on classical and mythological themes.

References:

- {{EB1911 poster|Andreani, Andrea}}

WikiPedia

- **Structured & Unstructured** bits and pieces
- A lot of "unbounded" elements
- Not a lot of restrictions
- The bit with value is in element "tekst"
How do we get this Structured?
Strings = small & defined (12c?)

Ename $\rightarrow$ pointer += 100;
<string1/> <string2/> <string3/>
Flexible, **Humans**
No **Design** Patterns
<small/> <verybigggr/> <bigger/>
<verybigggr>
  <empno>1</empno><ename>Marco</ename>
  <empno>2</empno>
</verybigggr>
We need options!
“XMLType” Container

In Memory (document)

Object Relational (data)

CLOB (document)

Binary XML (data)
XMLType

In Memory
(document)

XOB

XML Schema
XMLType

Binary XML Securefile
(document/content)

Post Parse

LOB Index
XMLType

Object Relational
(content)

Fully Shredded

Indexes
Something else to Realize!
“What is the *fastest* way to get this stuff in the database...?”
“...it depends...”
“So what is the fastest way to get XML in the database...?”
“...it depends...”
“So what is the fastest way to get XML in the database...

... and useful in my case...?”
Garbage IN – Garbage OUT
WikiPedia

- SQL*Loader
- Parallel or Direct
- Securefile LOB Column
- 2.5 hours

And no (performant) way to get the details out...
a.k.a “completely useless”
WikiPedia

- SQL*Loader
- Parallel or Direct
- Securefile Binary XML
- ...2.5 hours ???
XML Parsing

- **SAX** - Simple API for XML
- **DOM** - Document Object Module
XML Partitioning

• Object Relational Partitioning
  – Equi-Partitioning since version Oracle 11.1.0.7.0
• Binary XML Partitioning
  – Range, List, Hash
• Local partitioned XMLIndex
  – LOCAL keyword in XMLIndex create syntax
• Partition Key on virtual column (Binary XML)
• Partition Key on column (Object Relational)
XMLType

Binary XML Securefile
(document/content)

Post Parse

LOB Index
Driving access on CONTENT

- Bookstore
- Title
- Author
- Chapter
- Content

Unstructured XML Index

- Oracle XML Text Index

Structured XML Index

- BTree Index
- Function based Index (XPath)

- Whitepaper
- Title
- Author
- Id
- Paragraph
- Structured Content
Structured Data
Unstructured XMLIndex (UXI)

- PATH TABLE
- Use Path Subsetting
  - Full Blown XMLIndex can be BIG
- Token Tables (XDB.X$......)
  - Query re-write on Tokens
  - Fuzzy Searches, //
  - Optimizer Statistics
- Can be maintained manually
  - Recorded in Pending Table
- Secondary indexes possible
Describe PATH TABLE

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RID</td>
<td></td>
<td>ROWID</td>
</tr>
<tr>
<td>PATHID</td>
<td></td>
<td>RAW(8)</td>
</tr>
<tr>
<td>ORDER_KEY</td>
<td></td>
<td>RAW(1000)</td>
</tr>
<tr>
<td>LOCATOR</td>
<td></td>
<td>RAW(2000)</td>
</tr>
<tr>
<td>VALUE</td>
<td></td>
<td>VARCHAR2(4000)</td>
</tr>
</tbody>
</table>

```sql
SQL> select VALUE from UXI_RANGE_PATH_TABLE;
select VALUE from UXI_RANGE_PATH_TABLE
  *
ERROR at line 1:
ORA-30967: operation directly on the Path Table is disallowed
```
What’s hidden...

```
SQL> SELECT * from UXI_RANGE_PATH_TABLE where ROWNUM <= 10;

<table>
<thead>
<tr>
<th>RID</th>
<th>PATHID</th>
<th>ORDER_KEY</th>
<th>LOCATOR</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>2FD0</td>
<td>0202</td>
<td>010800180402001D000000000000076DF</td>
<td><a href="http://www">http://www</a> mediawiki org/xml/export-0.8/</td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>47B4</td>
<td>0204</td>
<td>0108001804020020000000000000076DF</td>
<td><a href="http://www">http://www</a> w3 org/2001/XMLSchema-instance</td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>7EE9</td>
<td>0206</td>
<td>010800180402002300000000000004120</td>
<td>Ronny and the Daytonas</td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>3BB7</td>
<td>0208</td>
<td>010800180402003D0000000000000A50 0</td>
<td></td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>0936</td>
<td>020A</td>
<td>01080018040200420000000000000008C5 167379</td>
<td></td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>15FC</td>
<td>020C02</td>
<td>010800180402004F00000000000002D6</td>
<td>Ronny &amp; the Daytonas</td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>3ABC</td>
<td>020C</td>
<td>011000180402004C00000000000007861</td>
<td></td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>2668</td>
<td>020E02</td>
<td>010800180402006B000000000000008C5 16054053</td>
<td></td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>7878</td>
<td>020E04</td>
<td>01080018040200770000000000000673 2003-01-09T23:56:47Z</td>
<td></td>
</tr>
<tr>
<td>AAAcB3AAABAARJdAAC</td>
<td>41A2</td>
<td>020E0602</td>
<td>010800180402009200000000000038D3 TUF-KAT</td>
<td></td>
</tr>
</tbody>
</table>

10 rows selected.
```
PATH TABLE

INDEXED COLUMNS

PATH INDEX
  ▪ (PATHID, RID), BTREE

ORDER INDEX
  ▪ (RID, ORDER_KEY), BTREE

VALUE INDEX
  ▪ (SUBSTRB("VALUE",1,1599))
  ▪ FUNCTION BASED

Not Indexed: LOCATOR column, pointer to XML fragments (XDB.X$...)

SECONDARY INDEXES
Structured XMLIndex (SXI)

- **CONTENT TABLE(s)**
- Based on **XMLTABLE** syntax
- XMLTable construct can be nested:
  - VIRTUAL column alias
- Can be maintained **manually**
- **Secondary** indexes possible
Describe CONTENT TABLE

• A “regular” heap table with columns...
• Ideal for secondary indexes, if needed.
CONTENT TABLE(s)

Structured XMLIndex

$ f(x) $

INDEXED COLUMNS

**KEY INDEX**
- (KEY), Unique BTREE

**RID INDEX**
- (RID), Non-Unique BTREE

Indexes needed for combined XMLIndex Types
Mixing Unstructured and Structured XMLIndexes

Your defined columns
Secondary indexes
```
SELECT PAGE_ID, -- NO XMLINDEX --
       PAGE_TITLE,
       PAGE_REV_TIMESTAMP
FROM BINARYXML_SECUREFILE_XSD t1,
    XMLTABLE (xmlnamespaces (default 'http://www.mediawiki.org/xml/export-0.6/'),
               '/*page'
               PASSING t1.content
               COLUMNS
               PAGE_ID NUMBER(9) PATH 'id'
               PAGE_TITLE VARCHAR2(100) PATH 'title'
               , PAGE_REV_TIMESTAMP TIMESTAMP(6) WITH TIME ZONE PATH 'revision/timestamp'
)
WHERE PAGE_TITLE='Andrea Andreani'
;
```
Binary XML + XMLIndex (SXI)

```
SELECT PAGE_ID, PAGE_TITLE, PAGE_REV_TIMESTAMP
FROM BINARYXML_TABLE_SECUREFILE t1,
XMLTABLE (xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/'), '/page',t1.content)
PASSING t1.content
COLUMNS
  PAGE_ID NUMBER(9) PATH 'id'
, PAGE_TITLE VARCHAR2(100) PATH 'title'
, PAGE_REV_TIMESTAMP TIMESTAMP(6) WITH TIME ZONE PATH 'revision/timestamp'
WHERE PAGE_TITLE='Andrea Andreani'
;
```
SELECT PAGE_ID, PAGE_TITLE, PAGE_REV_TIMESTAMP
FROM BINARYXML_TABLE_SECUREFILE t1,
XMLTABLE (xmlns:default 'http://www.mediawiki.org/xml/export-0.8/' )
  , '/page'
  PASSING t1.content
COLUMNS
    PAGE_ID NUMBER(9) PATH 'id'
, PAGE_TITLE VARCHAR2(100) PATH 'title'
, PAGE_REV_TIMESTAMP TIMESTAMP(6) WITH TIME ZONE PATH 'revision/timestamp'
WHERE PAGE_ID=1754
AND PAGE_TITLE='Andrea Andreani'
;
SELECT PAGE_ID, PAGE_TITLE, PAGE_REV_TIMESTAMP
FROM BINARYXML_TABLE_SECUREFILE t1,
XMLTABLE (xmlnamespaces('http://www.mediawiki.org/xml/export-0.8/','page')
PASSING t1.content
COLUMNS
    PAGE_ID NUMBER(9) PATH 'id'
    , PAGE_TITLE VARCHAR2(100) PATH 'title'
    , PAGE_REV_TIMESTAMP TIMESTAMP(6) WITH TIME ZONE PATH 'revision/timestamp'
WHERE PAGE_ID=1754
AND PAGE_TITLE='Andrea Andreani'
Un-Structured Data
XML Full Tekst Index

- Based on Oracle Text Index, XQuery Full Text
- XML Namespace Aware
- XML Semantic aware full text search
  - Full-Tekst Selection Expression – contains text
  - Logical Full Text Operator – ftor, ftand, ftMildNot
  - Context Aware full text search

```sql
SELECT po.id
FROM purchaseorder po
WHERE XMLEXISTS ('$src/purchaseOrder/billingInstruction/Address
  [.contains text '{$PHRASE1} ftand {$PHRASE2} using stemming]
  PASSING po.x,
  'Science' as "PHRASE1",
  'Magdalen' as "PHRASE2"
')
```
SELECT xt1.PAGE_TEXT
FROM BINARYXML_RANGE_PART_NORMAL t1,
XMLTABLE(xml.namespaces(default 'http://www.mediawiki.org/xml/export-0.8/'),
'SP/page/revision/text'
PASSING t1.content as 'P'
COLUMNS
PAGE_TEXT varchar2(4000) PATH '.
) xt1
WHERE XMLExists('xquery version "1.0";
  declare default element namespace 'http://www.mediawiki.org/xml/export-0.8/'; (: ;)
  $P/page/revision/text[], contains text {$PHRASE1} ftand {$PHRASE2} using stemming window 2 words'
PASSING
  t1.content as 'P',
  'oracle' as 'PHRASE1',
  'fusion' as 'PHRASE2'
  AND rownum <= 10
/
-- An XQuery Full-Text "contains text" search on a fragment using the ftand operator.
-- The index is used since the "contains text" comparison is case insensitive.
-- The Window clause specifies that the words must appear with 2 words of each other.

SELECT xt1.PAGE_TEXT
FROM BINARYXML RANGE PART_NORMAL t1,
XMLTABLE(xmlnamespace('default' http://www.mediawiki.org/xml/export-0.8/'),
'SF/page/revision/text'
PASSING t1.content as 'P'
COLUMNS
PAGE_TEXT varchar2(4000) PATH '
)
xt1
WHERE XMLExists('xquery version "1.0"; (: :)
declare default element namespace 'http://www.mediawiki.org/xml/export-0.8/'; (: :)
'SF/page/revision/text[. contains text ($PHRASE1) ftand ($PHRASE2) using stemming window 2 words]
PASSING
t1.content as 'P',
'oracle' as 'PHRASE1',
'fusion' as 'PHRASE2'

AND rownum <= 10
SELECT count(*)
FROM BINARXML_RANGE_PART_NORMAL t1,
XMLTABLE(xmlnamespaces({default 'http://www.mediawiki.org/xml/export-0.8/' }),
  '$P/page/revision/text'
  PASSING t1.content as "P"
  COLUMNS
    PAGE_TEXT varchar2(4000) PATH "."
) xtl
WHERE XMLExists( 'xquery version "1.0"; (: :)
  declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: :)
  $P/page/revision/text[. contains text {$PHRASE1} ftand {$PHRASE2} using stemming window 2 words]
  PASSING
    t1.content as "P",
    'oracle' as "PHRASE1",
    'ellison' as "PHRASE2"
)
SELECT count(*)
FROM BINARYXML_RANGE_PART_NORMAL t1,
XMLTABLE(xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/'),
 'SP/page/revision/text'
PASSING t1.content as 'P'
COLUMNS
   PAGE_TEXT varchar2(4000) PATH '.
) xtl
WHERE XMLExists('xquery version '1.0'; (: ;)
declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: ;)
$SP/page/revision/text[], contains text {$PHRASE1} ftand {$PHRASE2} using stemming window 6 words]
PASSING
   t1.content as 'P',
'oracle' as 'PHRASE1',
'ellison' as 'PHRASE2'
)
How Structured is my Data?

Which Index will support my Needs best?

Document or Data Driven?

Which XML Storage Model?

Which Queries must I support?

Am I allowed to “Tweak” the Data Format?
How Structured is my Data?

Which Index will support my Needs best?

Document or Data Driven?

Which Queries must I support?

Which XML Storage Model?

Am I allowed to “Tweak” the Data Format?
How Structured is my Data?

Document or Data Driven?

Which XML Storage Model?

Which Queries must I support?

Am I allowed to “Tweak” the Data Format?
How Structured is my Data?

Document or Data Driven?

Which XML Storage Model?

Which Queries must I support?

Am I allowed to “Tweak” the Data Format?

Which Index will support my Needs best?
Balanced Design

- Inserts, Updates & Deletes
  - XML Future Changes
  - Index Maintenance
- Selects
  - In Memory
  - Via Indexes
- XML Validation
  - Strict, Lazy
  - Client Side Possibilities
Reward

- Optimal performance
- Out performing XML
- Proper design will give performance increase over XML handling...

...proper design is still key...
References

Oracle XML DB

– http://www.oracle.com/pls/db112/homepage

XML DB FAQ Thread


Personal Blog

– http://www.xmldb.nl
– http://technology.amis.nl
References

Daniela Florescu, Oracle Corporation

Advances in XML and XQuery

Sam Idicula, Oracle XML DB Development Team

Binary XML Storage and Query Processing in Oracle

Jinyu Wang, Scott Brewton

Making XML Technology Easier to Use

Joel Spolsky - Joel on Software

Back to Basics
References

Oracle XML DB Main page material

• [Oracle XML DB : Best Practices to Get Optimal Performance out of XML Queries (PDF)]
• [Oracle XML DB : Choosing the Best XMLType Storage Option for Your Use Case (PDF)]
• [A Request for Comments for the Oracle Binary XML Format]