Oracle XMLDB

Building Blocks & Best Practices

Marco Gralike – UKOUG, 5th Dec, 2007

- Started as DBA with Oracle 7 on Windows NT 3.1 (1994)
- Experienced with Oracle 7.x / 8.x / 9.x / 10.x and 11.1
- Started with Oracle XMLDB in 2004 (Oracle 10.1.3)
- Oracle 11g Beta tester on Oracle XMLDB (mainly storage)
- Active Oracle OTN XMLDB Forum Member

Overview

- WORST PRACTICES
- XML SCHEMATA
- STORAGE STRUCTURES
- RECAP

Worst Practices !

What Inspired Me To Do This Presentation

Mr. Thomas Kyte:

"**Never Question Authority**"

Notorious Presentation from 2006: http://asktom.oracle.com;

"Worst Practices For Developers and DBAs Alike"

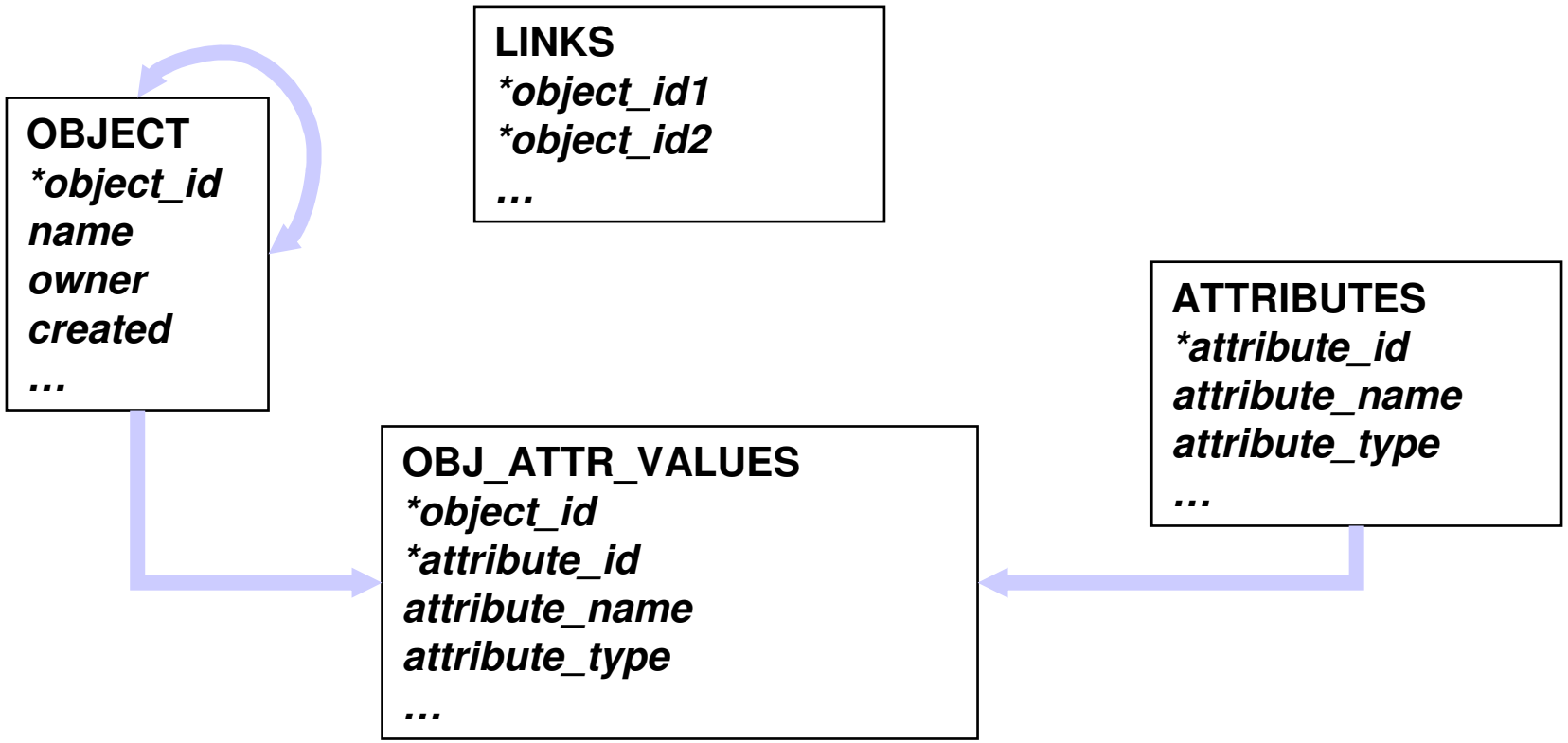
Probably You Do Not Need to Actually Design Anything

Quickly Answer:

- How many tables do you ***really*** need?

Quickly Answer:

- How many tables do you ***really*** need?
- **FOUR** at *most!*



Quickly Answer:

- How many tables do you ***really*** need?
- But of course **ONE** is *best!*
- And you are industry standard as well!

```
Create table Object
( object_id number primary key,
  data      xmltype );
```

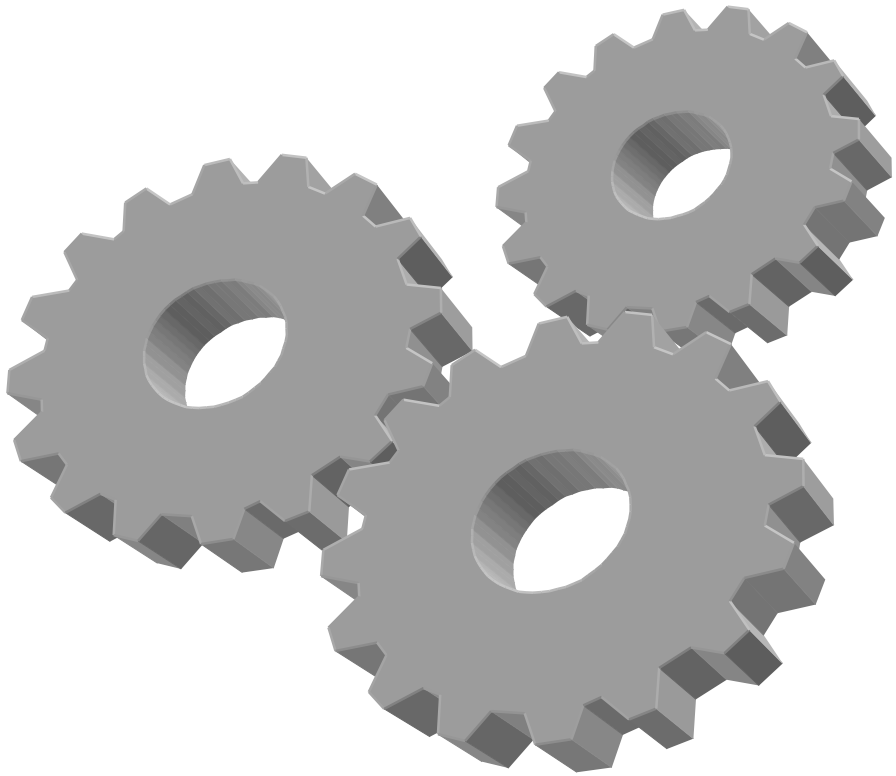
Brilliant Stuff, This XMLDB Database Thingy !

(You only need ONE TABLE !)

Mr. Tom Kyte Has a Point Though…

Logical Design?

Physical Design?



The "XML Relational" Mindset

```
Create table Object
( data  xmltype );
```

Here is your Design !

“Hidden” Oracle Design Features

```
CREATE TABLE "MARCO".“OBJECT" OF "SYS"."XMLTYPE"
  XMLSCHEMA "http://www.myserver.com/myapp/1.0/xsd/object.xsd"
ELEMENT “ROOT" ID 4733
PCTFREE 10 PCTUSE                                    GGING
  STORAGE(INITIA
   2147483645
  PCTINCREASE 0                                      AULT)
  TABLESPACE "USERS"
VARRAY "XMLEXTRA"."NAMESPACES" STORE AS BASICFILE LOB
   "NAMESPACES660_L"
  (ENABLE STORAGE IN ROW CHUNK 8192 PCTVERSION 10 CACHE
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
   2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT))
VARRAY "XMLEXTRA"."EXTRADATA" STORE AS BASICFILE LOB
   "EXTRADATA659_L"
  (ENABLE STORAGE IN ROW CHUNK 819                     CACHE
  STORAGE(INITIAL 65536 NEXT 10                        TENTS
   2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT))
```

XML Schema: your container for *Logical* and *Physical* Design

***Physical* Design**

My Relational Mindset

- **What goes IN, must come OUT (IN=OUT)**
  - Data Integrity, Meaning, Context
- Why do we **DESIGN** in the First Place?
  - ACID: Atomicity, Consistency, Isolation, Durability
  - Edgar F Codd's 12 Rules defining a "Relational Database"
  - Database Normalization (3NF)
  - Nullology
  - Imperfections of the (Procedural / Query) Language
- **Maintainability, Performance, Usability**
- **Decades of Innovation, Methods and Standards**

When you must store XML

- File Storage
- XML Database
  - Native
  - **XML Enabled**

Oracle XMLDB

- A World within a World
- More then one Option
- Complies to Standards
- Still License Free
- Core part of (OXJR)DBMS
- *The (Relational) foundation is there and can be used for solving XML imperfections*

XML Schemata

What should you take into account?

- **Logical Design**
  - XML Instance / Document
  - XML Schema (**XDB Annotations**), XML Evolution, Validation

- **Physical Design**
  - Oracle Database / Storage Design based on XML Logical Design
  - Binary Storage (Encrypted, Compressed, Validation, Evolution?)
  - **Indexes** (BTree, IOT, XMLIndex, Function Based, Text)

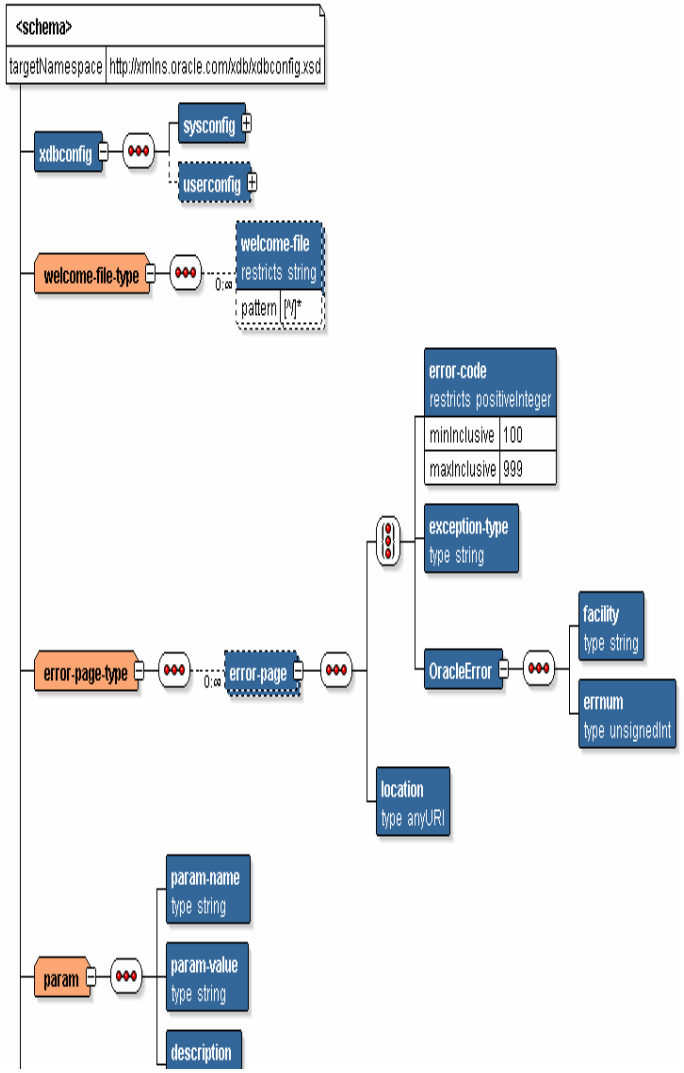
Is your XML data *CONTENT* or *DOCUMENT* driven?

Think in: Access Paths - The Optimizer - Performance

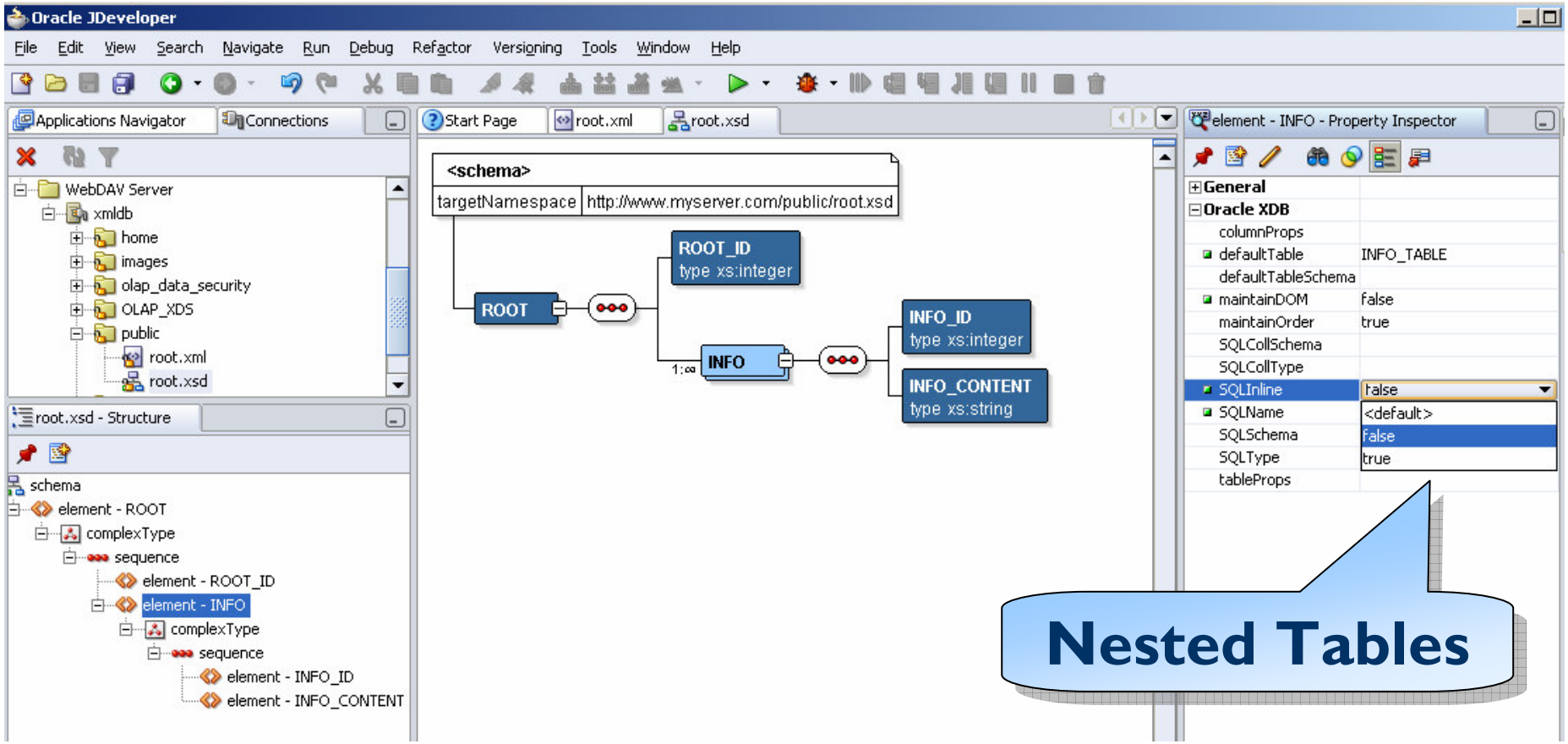
Controle Your Design

- **Maintained** via the **Repository**
  - Storage, Evolution, Security
- Storage can be a **Black Box**

  (without schemata)
  - CLOB
  - Binary XML (ALLOW NONSCHEMA)
- **Namespaces** (recursiveness ?!)
- **Overhead** (due to parsing)
- **XDB Annotations** have influence on:
  - Storage, Behaviour, Name



XDB Annotations



Mixing Logical and Physical Design

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:xdb="http://xmlns.oracle.com/xdb"
           xmlns="http://www.myserver.com/public/root.xsd"
           targetNamespace="http://www.myserver.com/public/root.xsd"
           elementFormDefault="qualified" attributeFormDefault="unqualified"
           xdb:storeVarrayAsTable="true">
  <xs:element name="ROOT" xdb:defaultTable="ROOT_TABLE" xdb:maintainDOM="false">
    <xs:complexType xdb:maintainDOM="false">
      <xs:sequence>
        <xs:element name="ROOT_ID" type="xs:integer" xdb:SQLName="ID"/>
        <xs:element name="INFO" xdb:SQLName="INFO" xdb:SQLInline="false"
                    xdb:defaultTable="INFO_TABLE" xdb:maintainDOM="false"
                    maxOccurs="unbounded">
          <xs:complexType xdb:maintainDOM="false">
            <xs:sequence>
              <xs:element name="INFO_ID" type="xs:integer" xdb:SQLName="INFO_ID"/>
              <xs:element name="INFO_CONTENT" xdb:SQLName="INFO_CONTENT"
                          type="xs:string" xdb:maintainDOM="false"
                          xdb:SQLType="VARCHAR2"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

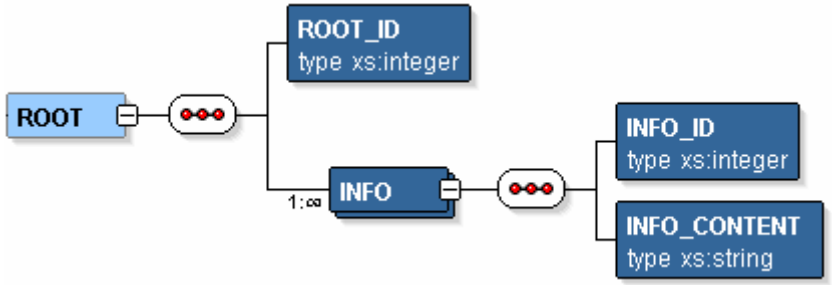
Generate or Manual Creation

- **Generate via DBMS_XMLSCHEMA**
  - XML Schema Based
  - XDB Annotations
  - JDeveloper or XMLSpy

- **Manual Creation**
  - DBMS_METADATA.GET_DDL
  - EVENT setting
    - alter session set events='31098 trace name context forever';
    - /oracle/diag/rdbms/xmldb/XMLDB/trace
  - Use Oracle 11g generation to apply in 10gR2 (IOT <> BTREE)

Creating Storage Structures (OR)

```
SQL> BEGIN
       DBMS_XMLSCHEMA.registerSchema
         (SCHEMAURL => 'http://www.myserver.com/root.xsd',
          SCHEMADOC => xdbURIType('/public/root.xsd').getClob(),
          LOCAL     => TRUE,   -- local
          GENTYPES  => TRUE,   -- generate object types
          GENBEAN   => FALSE,  -- no java beans
          GENTABLES => TRUE,   -- generate object tables
          OWNER     => USER);
       END;
     /
```

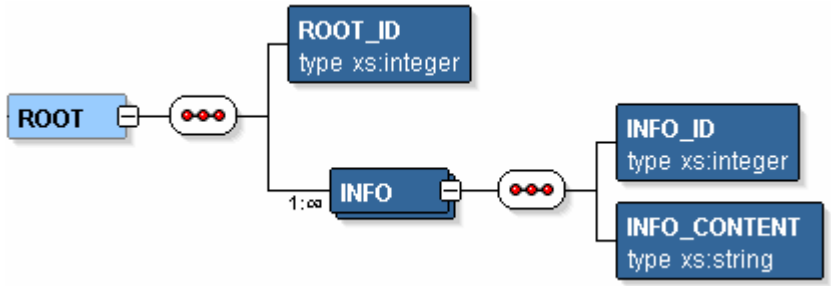


Result Object Relational Storage

```
SQL> select * from tab;

TNAME                          TABTYPE
------------------------------ -------
INFO_TABLE                     TABLE
ROOT_TABLE                     TABLE

SQL> select dbms_metadata.get_ddl('TABLE','ROOT_TABLE') from dual;

DBMS_METADATA.GET_DDL('TABLE','ROOT_TABLE')
--------------------------------------------------------
CREATE TABLE "MARCO"."ROOT_TABLE" OF "SYS"."XMLTYPE"
XMLSCHEMA "http://www.myserver.com/root.xsd" ELEMENT "ROOT"
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"
VARRAY "XMLEXTRA"."NAMESPACES" STORE AS BASICFILE LOB "NAMESPACES780_L"
 ...
VARRAY "XMLDATA"."INFO"."SYS_XDBPD$" STORE AS BASICFILE LOB "SYS_XDBP..."
DISALLOW NONSCHEMA
...
```

Creating Storage Structures (Binary)

```
SQL> BEGIN
       DBMS_XMLSCHEMA.registerSchema
         (SCHEMAURL => 'http://www.myserver.com/root.xsd',
          SCHEMADOC => xdbURIType('/public/root.xsd').getClob(),
          LOCAL     => TRUE,   -- local
          GENTYPES  => FALSE,  -- generate object types
          GENBEAN   => FALSE,  -- no java beans
          GENTABLES => TRUE,   -- generate object tables
          OPTIONS   => DBMS_XMLSCHEMA.REGISTER_BINARYXML,
          OWNER     => USER);
       END;
     /
```



Result Binary XML Storage

```
SQL> select * from tab;

TNAME                          TABTYPE
------------------------------ -------
ROOT_TABLE                     TABLE

SQL> select dbms_metadata.get_ddl('TABLE','ROOT_TABLE') from dual;

DBMS_METADATA.GET_DDL('TABLE','ROOT_TABLE')
--------------------------------------------
CREATE TABLE "MARCO"."ROOT_TABLE" OF "SYS"."XMLTYPE"
XMLTYPE STORE AS BASICFILE BINARY XML  (
  TABLESPACE "USERS" ENABLE STORAGE IN ROW CHUNK 8192 PCTVERSION 10
  NOCACHE LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT))
XMLSCHEMA "http://www.myserver.com/root.xsd" ELEMENT "ROOT"
DISALLOW NONSCHEMA
...
TABLESPACE "USERS"
```

XML Validation

- **Functions**
  - XMLisValid()
  - isSchemaValid()
  - isSchemaValidated()
- **Procedures**
  - setSchemaValidated()
  - schemaValidate()

Oracle 11gR1

- **Binary XML Storage Option**
  - *) XMLSCHEMA
  - *) ALLOW ANYSCHEMA
  - (DIS)ALLOW NONSCHEMA

Mutually Exclusive

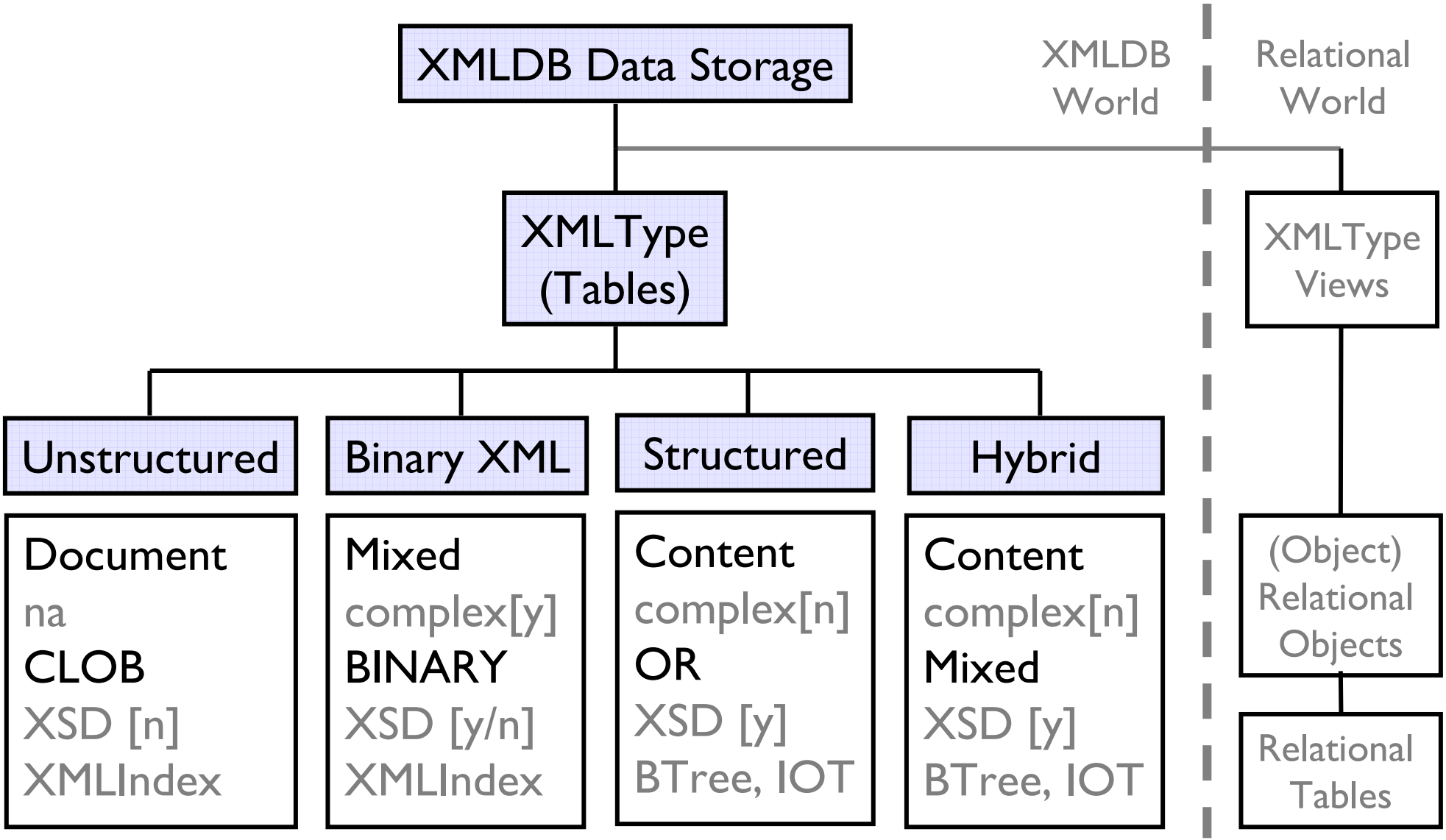
Syntax Example ;-)

```
SQL> create table "XML_XSD01" of XMLTYPE
  2  XMLTYPE STORE AS BASICFILE BINARY XML
  3  XMLSCHEMA "http://www.myserver.com/root.xsd"
  4    ELEMENT "ROOT"
  5  ALLOW NONSCHEMA;

Table created.

SQL> create table "XML_XSD02_BUG" of XMLTYPE
  2  XMLTYPE STORE AS BASICFILE BINARY XML
  3  XMLSCHEMA "http://www.myserver.com/root.xsd"
  4    ELEMENT "ROOT"
  5  ALLOW ANYSCHEMA
  6  DISALLOW NONSCHEMA;

Table created. Oops...
```

B.6396229

Storage Structures

CLOB, Object Relational, Binary XML



Some XML Storage Issues

	O.R.	CLOB	Binary XML
Schema Based	+	-	+
Whitespace	-	+	-
DML	++	-	+
XPath Performance	++	-	+
Throughput	-	+	++
Memory Management	++	-	+
XMLIndex	-	+	+
Function / Text	+	+	+
BTree (/ IOT)	+	-	-

XMLINDEX index

PATH INDEX

- (PATHID, RID)
- BTREE

ORDER INDEX

- (RID, ORDER_KEY)
- BTREE

VALUE INDEX

- (SUBSTRB("VALUE",1,1599))
- FUNCTION BASED

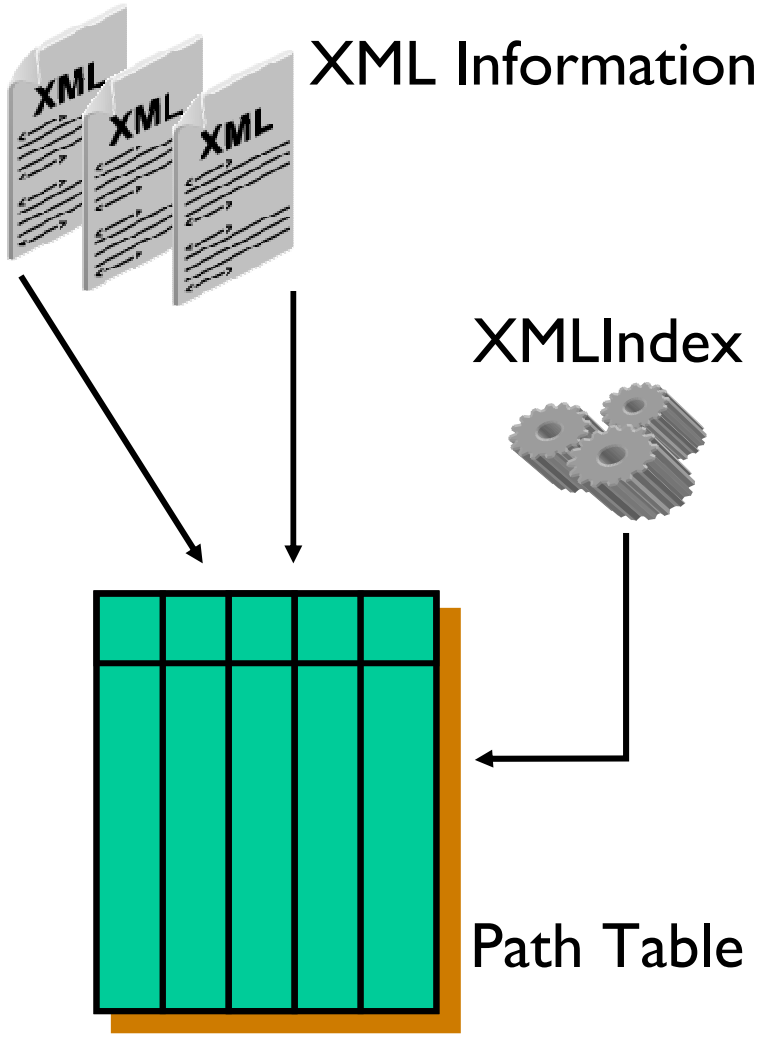
locator: pointer to XML fragments

PATH_TABLE

pathID	RID	order key	locator	value
RAW	RowID	RAW	RAW	varchar2 (4000)

XMLIndex came to the Rescue

- Logical Index designed for the XMLDB domain
- "Revival" of CLOB Storage (select)
- Use **Path Subsetting**
  - Full Blown XMLIndex can be BIG
- Token Tables (XDB.X$.....)
  - Query Rewrite on Tokens
  - Fuzzy Searches, //
- Can be maintained Manually
- Doesn't support all XPath expressions



XMLIndex Examples

```
SQL> CREATE INDEX XMLIX_INFO on INFO_TABLE (object_value)
  2  INDEXTYPE IS XDB.XMLIndex;

SQL> CREATE INDEX XMLIX_ROOT on ROOT_TABLE (object_value)
  2  INDEXTYPE IS XDB.XMLIndex
  3  PARAMETERS ('PATHS (INCLUDE (/ROOT/ID
  4                               /ROOT/INFO/INFO_ID
  5                              )
  6                    NAMESPACE MAPPING
                     (xmlns="http://www.myserver.com/root.xsd"))
  7                   PATH TABLE       XMLBIN_PATH_TABLE
  8                   PATH ID INDEX    XMLBIN_PATHID_IX
  9                   ORDER KEY INDEX XMLBIN_ORDERKEY_IX');
```

XML Binary Options

```
SQL> CREATE TABLE "ROOT_TABLE" OF "SYS"."XMLTYPE"
  2  XMLTYPE STORE AS SECUREFILE BINARY XML
  3  XMLSCHEMA
     "http://www.myserver.com/root.xsd" ELEMENT "ROOT"
  4  DISALLOW NONSCHEMA
  5  COMPRESS
  6  ;

SQL> alter table "ROOT_TABLE"
  2  modify lob (XMLDATA)
  3  (COMPRESS HIGH);

-- ALTER TABLE "ROOT" MODIFY LOG (XMLDATA)
-- COMPRESS | NOCOMPRESS | COMPRESS MEDIUM | COMPRESS HIGH
```

B.6364855

Recap…

- There are a lot of design decisions to make…
- There is more control then you would wish for…
- Design with future demands in mind !
- There is no universal solution for XML data, yet…

But what ever you do…

…you will have to go the “Full Monty”;

References

- XMLDB Developers Guide
- The XMLDB Forum
  - http://forums.oracle.com/forums/forum.jspa?forumID=34
- XML DB FAQ Thread
  - http://forums.oracle.com/forums/thread.jspa?threadID=410714
- Blog
  - http://technology.amis.nl/blog
  - http://blog.gralike.com
- Asktom at http://asktom.oracle.com ("Files" section)
- Very grateful for Mark Drake's (Oracle) efforts

Questions?

marco.gralike@amis.nl
http://www.amis.nl