Unstrukturierte und semistrukturierte Daten in der Oracle Datenbank – ein Überblick

Marc Lieber
Principal Consultant
AGENDA

1. Basic storage types
   - LOBs
   - XML
   - JSON

2. MDSYS: Oracle Spatial & Graphs data storage
   - Oracle Spatial
   - Network graphs
   - RDF semantic graphs

3. Oracle Big Data SQL
   - NoSQL and SQL
   - Hadoop and SQL
Basic Storage Types
LOBs

- **CLOB** = Character large objects
  - Text files: log files, SQL scripts, RTF documents...

- **BLOB** = Binary large objects
  - Binary files: Excel, MS Word, PDFs, pictures, movies...

- **Bfiles**
  - Binary files not stored in the database but linked

Dedicated APIs and Functions

- **DBMS_LOB** package
- SQL functions regexp... to handle CLOBs
- Oracle Text support for full text search
**LOBs**

- Query example on CLOB:

```sql
SELECT textdoc,
       dbms_lob.getlength(textdoc) length,
       regexp_instr(textdoc, 'v_$session') locate
FROM t_documents
where regexp_like (textdoc, 'v_$session');
```
XML storage

- XML datatype: **XMLTYPE** with or without XSD Validation
- Schema **XDB** is an XML repository that implements a file system paradigm. It can be also used as a
  - Content Management System: any type of document + XDBUriType
  - Native Web Service provider
  - HTTP server for APEX
- XML implementation has gone through lots of changes since first Oracle 8i Java APIs. **XQUERY** is now the Standard
  - → C APIs → XDB → XQuery functions → XMLType Binary storage → XML Indexes, XQuery Update, XQuery full text search
  - Oracle supported XQuery functions: XMLQuery, XMLTable, XMLExists, XMLCast, XMLSerialize, XMLParse, XMLIndex
SQL Function XMLTable example

```sql
SELECT xt.deptno, xt.dname, vt.ename, vt.empno, vt.job, a.SYS_NC_ROWINFO$
FROM T_XMLDOCS_BINARY a,
     XMLTABLE('**'
     PASSING OBJECT_VALUE
     COLUMNS
     deptno INTEGER PATH '@Deptno',
     dname VARCHAR(40) PATH 'DepartmentName',
     xmlresult XMLTYPE PATH './Employees/Employee'
 ) xt
,
     XMLTABLE('**'
     PASSING xt.xmlresult
     COLUMNS
     ename VARCHAR(30) PATH 'Lastname/text()',
     empno INTEGER PATH '@Empno',
     job VARCHAR(30) PATH 'Job'
 ) vt;
```

Query Result:

<table>
<thead>
<tr>
<th>DEPTNO</th>
<th>DNAME</th>
<th>ENAME</th>
<th>EMPNO</th>
<th>JOB</th>
<th>SYS_NC_ROWINFO$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ACCOUNTING</td>
<td>CLARK</td>
<td>7782</td>
<td>MANAGER</td>
<td>&lt;Department Deptno=&quot;10&quot;&gt; &lt;DepartmentName&gt;ACCOUNTING&lt;/DepartmentName&gt; &lt;Loca</td>
</tr>
<tr>
<td>2</td>
<td>ACCOUNTING</td>
<td>KING</td>
<td>7839</td>
<td>PRESIDENT</td>
<td>&lt;Department Deptno=&quot;10&quot;&gt; &lt;DepartmentName&gt;ACCOUNTING&lt;/DepartmentName&gt; &lt;Loca</td>
</tr>
<tr>
<td>3</td>
<td>ACCOUNTING</td>
<td>MILLER</td>
<td>7934</td>
<td>CLERK</td>
<td>&lt;Department Deptno=&quot;10&quot;&gt; &lt;DepartmentName&gt;ACCOUNTING&lt;/DepartmentName&gt; &lt;Loca</td>
</tr>
</tbody>
</table>
Support for XQuery Update

1. Allows fragment-level updates on XML content

```sql
UPDATE t_xmltable a
SET object_value = XMLQuery
('copy $NEWXML := $XML modify {
  for $EMP in $NEWXML/ROWSET/ROW return {
    replace value of node $EMP/ENAME with concat($USERID,$EMP/ENAME),
    replace value of node $EMP/SAL with xs:integer(fn:number($EMP/SAL)+$NEWSAL),
    delete nodes $EMP/MGR,
    insert node $NEWNODE after $EMP/ENAME |
  }
}) return $NEWXML' passing object_value AS "XML",
'Mr' AS "USERID",
11 as "NEWSAL",
xmType('<Updated>'||sysdate||'</Updated>'
) AS "NEWNODE"
returning content )
where xmlexists('/ROWSET' passing object_value);
```

- In this example, we did two updates, one delete and one insert on one single XML document stored in an XMLType table.
Oracle RDBMS support of JSON

- New feature Oracle 12.1.0.2.0: Support for storing, querying and indexing JSON data in the database
- SQL Query JSON data using a PATH based notation
- IS JSON Constraint ensures that the column contains valid JSON documents → no new JSON data type

```
create table t_countries
  ( id          RAW (16) NOT NULL,
    shortName  varchar2(3 CHAR),
    countryName VARCHAR2(50 CHAR),
    region VARCHAR2(50 CHAR),
    date_loaded TIMESTAMP WITH TIME ZONE,
    JsonDoc CLOB,
    CONSTRAINT ensure_json_c CHECK (JsonDoc IS JSON));
```
JSON SQL functions

- JSON_VALUE, JSON_QUERY, JSON_TABLE, JSON_EXISTS
- Example with JSON_TABLE

```
SELECT t.shortname, t.countryname, jt.*
FROM t_countries t,
    json_table(jsondoc, '$'
    COLUMNS (currency VARCHAR2(32 CHAR) PATH '$.currency[*]',
             NESTED PATH '$.borders[*]' COLUMNS (border VARCHAR2(32 CHAR) PATH '$')))) AS "JT"
WHERE shortname='DEU';
```

- Full text search support using JSON_TEXTCONTAINS()
JSON_QUERY example

- Returns an ARRAY or OBJECT

```
SELECT t.shortname, t.countryname,
   json_query(jsondoc, '$.translations.deu' WITH WRAPPER) german_Translation,
   json_query(jsondoc, '$.name' PRETTY WITHOUT WRAPPER) name
FROM t_countries t where region='Europe' ;
```
Database Schema MDSYS

Oracle Spatial and Graph
Spatial Features, Network Data Model Graph
& RDF Semantic Graph
Oracle Spatial and Graph option

- “Points”
- “Lines”
- “Polygons”

Web Services (OGC)  
SPARQL End Point  
Rasters  
Network Graphs  
Topologies  
RDF Semantic Graphs  
Geocoding  
Routing  
Inferencing  
3D
Oracle Spatial

- A spatial data management platform for Spatial Analysis
  - Support complex geographic information systems (GIS) applications
  - Geocoding for mining functions
  - GeoRaster for satellite imagery
  - Topology Data Model
  - Mapviewer

- Example of application
  - Transportation: Tracking & Logistics
  - Utilities: gas, electric, pipeline, and water agencies
  - Defense & Commercial Location intelligence
  - Customer Relationship Management (Sales, Marketing, Call Centers)
SQL Spatial Operators

- **Topological operators**
  - Inside Contains
  - Touch Disjoint Interact
    - SDO_ANYINTERACT
  - Covers Covered By
  - Equal Overlap Boundary

- **Distance operators**
  - Within Distance
    - SDO_WITHIN_DISTANCE
  - Nearest Neighbor
Oracle Spatial example

Save the coordinates of one restaurant:

```sql
INSERT INTO us_restaurants VALUES
( 1,'PIZZA HUT',
  SDO_GEOMETRY(2001, NULL,
    SDO_POINT_TYPE -- coordinates of the point
    ( -87, -- longitude dimension
    38, -- latitude dimension
    NULL), NULL, NULL));
```

Identifying All Restaurants in a 50km Radius around I-95

```sql
SELECT POI_NAME
FROM us_interstates I,
  us_restaurants P
WHERE SDO_ANYINTERACT(
  P.location,
  SDO_GEOM.SDO_BUFFER(
    I.geom, 50, 0.5, 'UNIT=KM')) = 'TRUE'
AND I.interstate='I95';
```
Network Data Model (NDM)

- Link-node Graphs to represent physical and logical networks used in transportation, utilities, energy and communications
  - PL/SQL API for managing network data in the database
  - Java APIs to perform analysis
  - Traffic patterns recording
  - Network analysis such as:
    - shortest Path
    - nearest neighbors
  - Spatial data is optional
**Network Data Model**

- Connectivity defined with nodes and links
  - Each link has a start and an end node
  - Links & nodes have costs
  - Links can be one-way or bi-directed

SQL> desc MYNETWORK_NODE$;
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE_ID</td>
<td>NUMBER (Primary Key)</td>
</tr>
<tr>
<td>COST</td>
<td>NUMBER</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>VARCHAR2(1)</td>
</tr>
<tr>
<td>GEOMETRY</td>
<td>SDO_GEOMETRY</td>
</tr>
</tbody>
</table>

SQL> desc MYNETWORK_LINK$;
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINK_ID</td>
<td>NUMBER (Primary Key)</td>
</tr>
<tr>
<td>START_NODE_ID</td>
<td>NUMBER</td>
</tr>
<tr>
<td>END_NODE_ID</td>
<td>NUMBER</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>VARCHAR2(1)</td>
</tr>
<tr>
<td>LINK_LEVEL</td>
<td>NUMBER</td>
</tr>
<tr>
<td>COST</td>
<td>NUMBER</td>
</tr>
<tr>
<td>GEOMETRY</td>
<td>MDSYS.SDO_GEOMETRY</td>
</tr>
</tbody>
</table>
RDF Semantic technologies

- Techniques for finding signal in large or complex data sources
  - Link Analysis
  - Distance analysis
  - Pattern discovery
  - Detect anomalies
  - Complex search

- Data federation using an open standard: SPARQL

- Reasoning concepts based on predefined rules such as OWL, RDFS for data discovery
  - Example of use: fraud detection, Intelligence services, Pharmaceutical industries
Oracle Spatial & Graphs; RDF support

- An RDF Triple Store running inside RDBMS
  - SEM_API and SEM_PERF PL/SQL Packages for admin
  - Query language is SPARQL
    - Native W3C Query and UID language available through Java APIs
    - Embedded SPARQL queries within SQL : SEM_MATCH
SPARQL example

Find all neighbors of Switzerland:

```
SELECT x, cname, pop, gdp, neighbor
FROM TABLE(SEM_MATCH(
  '{?x rdf:type :Country .
   ?x :name ?cname filter (sameTerm(?cname,"Switzerland")) .
   ?x :population ?pop .
   ?x :neighbor ?o .
   ?o :name ?neighbor}',
  SEM_Models('VIRT_MODEL_MONDIAL'),
  SEM_Rulebases(''),
  SEM_ALIASES(SEM_ALIAS('','http://www.semwebtech.org/mondial/meta#'))),
null,null,
'HINT0={LEADING(t0 t1) USE_NL(?x ?y) GET_CANON_VALUE(?x ?y)}'));
```

<table>
<thead>
<tr>
<th>x</th>
<th>CNAME</th>
<th>POP</th>
<th>GDP</th>
<th>NEIGHBOR</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.semwebtech.org/mondial/10/countries/CH/Switzerland">http://www.semwebtech.org/mondial/10/countries/CH/Switzerland</a> 7207060 158500 Germany</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.semwebtech.org/mondial/10/countries/CH/Switzerland">http://www.semwebtech.org/mondial/10/countries/CH/Switzerland</a> 7207060 158500 Italy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.semwebtech.org/mondial/10/countries/CH/Switzerland">http://www.semwebtech.org/mondial/10/countries/CH/Switzerland</a> 7207060 158500 France</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.semwebtech.org/mondial/10/countries/CH/Switzerland">http://www.semwebtech.org/mondial/10/countries/CH/Switzerland</a> 7207060 158500 Liechtenstein</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.semwebtech.org/mondial/10/countries/CH/Switzerland">http://www.semwebtech.org/mondial/10/countries/CH/Switzerland</a> 7207060 158500 Austria</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Semantic Web query federation

- Linked Open Data Initiative: SPARQL 1.1 to federate local data with remote **SPARQL End Points**
- Search over multiple Datasets with one Query

```sparql
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX gp: <http://www4.wiwi.uni-berlin.de/gutendata/resource/people/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

WHERE
{
  SERVICE <http://DBpedia.org/sparql>
  { SELECT ?dbpProperty ?dbpValue
      WHERE
      { <http://dbpedia.org/resource/Joseph_Hocking> ?dbpProperty
      }
  }
  SERVICE <http://www4.wiwi.uni-berlin.de/gutendata/sparql>
  { SELECT ?gutenProperty ?gutenValue
      WHERE
      }
  }
}
```
Data Federation: RDBMS mapping to RDF

- Purpose: federate data stored in relational Table with RDF data
- Oracle native R2RML and RDB2RDF implementation
- External tools: Ontop, D2RQ, ...
Hadoop and Nosql Database mapping in RDBMS
Oracle Big Data SQL to federate data

- Oracle strategy on data federation is to use SQL as a common querying tool
- not available for Oracle Databases running on non-Exadata hardware
Oracle NoSQL Database Enterprise Edition for document storage

1. Scalable, highly available, high performance Key Value store
   - Massive horizontal scalability – petabytes of data

2. Storage options:
   - Key values
   - RDF Graph using Jena or Sesame APIs
   - Table based model: Specific APIs + use of JSON to load and retrieve data
     - Can be seen as an external table in Oracle RDBMS

3. Advantage Oracle NoSQL
   - Oracle Support
   - Is an Enterprise solution
Oracle NoSQL: external tables

Oracle loader type mapping:

```sql
CREATE TABLE CUSTOMER (CUSTID VARCHAR2(13), EMAIL VARCHAR2(30), AGE NUMBER(3), FIRST VARCHAR2(16), MIDDLE VARCHAR2(8), LAST VARCHAR2(16), GENDER VARCHAR2(1), PASSWORD VARCHAR2(32))
ORGANIZATION EXTERNAL (type oracle_loader
    default directory ext_tab
    access parameters (records delimited by newline
        preprocessor nosql_bin_dir:'nosql_stream'
        fields terminated by ',')
    LOCATION ('user.dat')) PARALLEL;
```

SQL> select custid, email from customer;

<table>
<thead>
<tr>
<th>CUSTID</th>
<th>EMAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><a href="mailto:john.parker@email.com">john.parker@email.com</a></td>
</tr>
<tr>
<td>12</td>
<td><a href="mailto:susan.wang@email.com">susan.wang@email.com</a></td>
</tr>
</tbody>
</table>
Oracle Big Data SQL

- Combine data from Oracle Database, Hadoop and NoSQL in a single SQL query
  - Only available on Oracle Exadata Big Data Appliance
- Extend security and access policies from Oracle Database to data in Hadoop and NoSQL
- Uses external table types with Hadoop or NoSQL keywords:
  - data in Hadoop and NoSQL is exposed to Oracle Database users
Oracle Big Data SQL: external tables

- example

```sql
CREATE TABLE access_per_post_categories(
    hostname varchar2(100),
    request_date varchar2(100),
    post_id varchar2(10),
    title varchar2(200),
    author varchar2(100),
    category varchar2(100),
    ip_integer number)
ORGANIZATION EXTERNAL (type oracle_hive
    default directory default_dir
    access parameters (com.oracle.bigdata.tablename =
    default.access_per_post_post_categories));
```
Fragen und Antworten...

Marc Lieber

marc.lieber@trivadis.com