

在 Oracle 12c 使用 JSON





演讲人: John Jay King

下载地址: <http://www.kingtraining.com>



- 理解Oracle 12c 如何支持 JSON
- 在数据库中存储JSON 数据
- 熟悉JSON 的条件和功能
- 使用 JSON 条件和功能查询JSON数据
- 在JSON_TABLE上使用 SQL 查询JSON 数据



- John King –King Training Resources的股东
- Oracle Ace 总监 
- Oak Table Network成员 
- 给Oracle 和IT 社团提供培训, 有超过25年的工作经验, 网址是 <http://www.kingtraining.com>
- 对Oracle, ADF, SQL, Java, 和PL/SQL (还有一些其他课题) 了如指掌的“技术专家”。
- AZORA, ODTUG, IOUG, 和 RMOUG的成员



- JSON (JavaScript Object Notation)是一种独立于语言, 开放式的 (www.json.org)
- 在名值对中储存文本
- 最初出现在JavaScript, 现在也出现于: Java, R, .NET, PHP, Python, Node.js, 和 Oracle中
- 常用于数据互换
- 频繁用于在REST-style 网络服务中来回传递数据
- 因大数据的持久性而广受欢迎



- 12c patch-set 2 (12.1.0.2) 添加了JSON
- JSON 文件以VARCHAR2, CLOB, 或BLOB 数据类型进行存储
- JSON 数据可以和Oracle 现有包括SQL和 Analytics的所有特性一起工作
- 12c 支持 JSON data 基于路径的查询, 这些数据存储在数据库, JSON Path Language, 和JSON Path Expressions中
- 通过SQL/JSON 视图, 在SQL中使用JSON
- JSON 文件可以被索引



- JSON 就像XML格式一样, 只是文本, 因此它是进行数据互换的一种有效媒介
- JSON 和 XML 是“人们可读的”并且能够进行“自描述”(有点...)
- JSON 和XML 是分层的(数据集合安置在 数据集内)
- JSON 和XML 提供了验证功能; 现在XML的功能 愈加成熟



- XML 比较冗长, JSON 相对更精简
- JSON 没有结束标记, 而XML中则需要有结束标记
- JSON 在读写速度方面更快
- 读取XML 文件时需要“遍历DOM” – JSON 则不需要
- 与AJAX一起运行时, 相对XML来说, JSON 运行起来更加便捷快速
- 在处理数据之前, XML文件必须进行“语法规则”测试



- 使用逗点隔开的名/值对来存储数据
 - 字段名/键 (由双引号括起来)
 - 冒号“:”,
 - 值(由双引号括起的字符串, 不需要括起来的数值, 布尔值 true/false/null)
- 对象包含在大括号内 “{“ & “}”

```
{ "lastName": "King" }
```

- 数组 使用方括号 “[“ & “]” 和 逗号

```
[ { "lastName": "King" },  
  { "lastName": "Manzo" } ]
```




```
<?xml version="1.0"?>
<myBooks>
  <book>
    <name>Learning XML</name>
    <author>Eric T. Ray</author>
    <publisher>O'Reilly</publisher>
  </book>
  <book>
    <name>XML Bible</name>
    <author>Elliotte Rusty Harold</author>
    <publisher>IDG Books</publisher>
  </book>
  <book>
    <name>XML by Example</name>
    <author>Sean McGrath</author>
  </book>
</myBooks>
```



```
{ "myBooks" :  
  [ { "book" : {  
      "name" : "Learning XML",  
      "author" : "Eric T. Ray",  
      "publisher" : "O'Reilly" }  
    },  
    { "book" : {  
      "name" : "XML Bible",  
      "author" : "Elliotte Rusty Harold",  
      "publisher" : "IDG Books" }  
    },  
    { "book" : {  
      "name" : "XML by Example",  
      "author" : "Sean McGrath",  
      "publisher" : "Prentice-Hall" }  
    }  
  ]  
}
```



- JSON 文件使用现有的数据类型存储到数据库中
 - VARCHAR2, CLOB 和 BLOB 针对 JSON 字符模式
 - 外部的JSON 数据资源(包括HDFS 文件系统的那些资源)可以通过外部表来获取。
- 基于JSON_TABLE的相关视图, 使用SQL来获取JSON数据
- Oracle JSON 文件可以被索引; JSON 路径可以使用功能索引



- JSON 内容可以从SQL中获取，而且可以视为JSON 来使用：
 - JSON 条件运算符
 - JSON 功能
- JSON 运算符和功能函数使用 JSON Path 语言来引导JSON 对象



- JSON 内容可以经过新的条件运算符从SQL 中
 - IS JSON 验证JSON, 通常是在 CHECK 限制和 WHERE 中 (IS / IS NOT)
 - JSON_EXISTS 如果JSON 路径 在文件中存在则为True
 - JSON_TEXTCONTAINS 如果在 JSON 的属性值 中发现文本字符串则为True (使用 Oracle 文本)



- JSON 内容也可以通过新的功能 来获取
JSON_VALUE 用来从JSON 文件中查询
标量值
 - JSON_QUERY 用来查询所有或者部分JSON
文件内容
 - JSON_TABLE 用来查询JSON文件
并且创建相关格式的列



- 如果特定的表达是一个有效的JSON文件，那么IS JSON 会返回TRUE

```
**expr**      IS JSON | IS NOT JSON  
              FORMAT JSON  
              STRICT | LAX  
              WITH | WITHOUT UNIQUE KEYS
```

- 如果是有效的JSON，则为IS true，如果不是JSON，则为IS NOT true
- FORMAT JSON (只是满足BLOB 数据的需要)
- STRICT / LAX 严格检查级别
- 如果需要主键，则为WITH / WITHOUT



```
create table deptj
(id raw(16) not null,
 dept_info clob constraint deptjson
                check (dept_info is json)
);
```

```
create table deptj
(id raw(16) not null,
 dept_info clob constraint deptjson
                check (dept_info is json strict)
);
```




```
select id,dept_info
from deptj
where dept_info is json
```

```
select id,dept_info
from deptj
where dept_info is json strict;
```

```
select id,dept_info
from deptj
where dept_info is json format json strict;
```



```
insert into deptj values
(sys_guid(),
'{"departments":{
  "DEPTNO": 10, "DNAME": "ACCOUNTING", "LOC": "NEW YORK",
  "deptemps": [
    { "EMPNO": 7782,
      "ENAME": "CLARK",
      "JOB": "MANAGER",
      "MGR": 7839,
      "HIREDATE": "09-JUN-81",
      "pay": {
        "SAL": 2450,
        "COMM": null},
      "DEPTNO": "10"
    },
    /* more */
```



```
select dept_info  
from deptj;
```

DEPT_INFO

```
{"departments": {  
  "DEPTNO": 10,  
  "DNAME": "ACCOUNTING",  
  "LOC": "NEW YORK",  
  "deptemps": [  
    {  
      "EMPNO": 7782,  
      "ENAME": "CLARK",  
      **** more ****
```



- JSON 文件中的\$.jsonpath Identifies 路径
 - \$. (需要的)
 - Jsonpath (如果不需要整个文件的返回, 那么就指定查找JSON 文件中的部分来进行搜寻)
- 由JSON_QUERY, JSON_VALUE, JSON_TABLE, JSON_EXISTS, JSON_TEXTCONTAINS使用的JSON path



- JSON_VALUE 在JSON 文件中发现了一个值, 并将它作为 VARCHAR2 或者 NUMBER 返回给SQL

```
JSON_VALUE (expr | expr FORMAT JSON,  
            '$.jsonpath'  
            RETURNING VARCHAR2 (n) | NUMBER (n, n)  
            ERROR | NULL | DEFAULT xxx ON ERROR  
)
```



```
select json_value(dept_info, '$.departments.DNAME')  
from deptj;
```

DNAME

ACCOUNTING

RESEARCH

SALES

OPERATIONS



- JSON_VALUE 从JSON 数据中返回标量值

```
select json_value(dept_info
                  , '$.departments.DNAME')
from deptj;
```

```
select json_value(dept_info
                  , '$.departments.deptemps[0].ENAME')
from deptj;
```

```
select dept_info
from deptj
where json_value(dept_info,
                 '$[0].departments.DEPTNO') = '10'
```



- JSON_TABLE 把JSON 数据匹配到 相关的行和列中

```

JSON_TABLE ( expr | expr FORMAT JSON, '$.jsonpath'
            ERROR | NULL | DEFAULT xxx ON ERROR
            COLUMNS (
                colname1 datatype
                    EXISTS PATH '$.jsonpath'
                    ERROR|NULL|DEFAULT xxx ON ERROR,
                colname2 datatype FORMAT JSON
                    jsonquerywrapper PATH '$.jsonpath'
                    ERROR|NULL|DEFAULT xxx ON ERROR,
                colname3 datatype PATH '$.jsonpath'
                    ERROR|NULL|DEFAULT xxx ON ERROR,
                NESTED PATH '$.jsonpath' COLUMNS (...),
                colname4 FOR ORDINALITY
            )
    
```




```
select dname,ename,job,sal
from deptj, json_table(dept_info,'$.departments'
    columns (dname varchar2(15) path '$.DNAME'
    ,nested path '$.deptemps[*]'
        columns (ename varchar2(20) path '$.ENAME'
        ,job varchar2(20) path '$.JOB'
        ,nested path '$.pay'
            columns (sal number path '$.SAL')
        )
    )
);
```

DNAME	ENAME	JOB	SAL
-----	-----	-----	-----
ACCOUNTING	CLARK	MANAGER	2450
ACCOUNTING	KING	PRESIDENT	5000
**** more ****			



- Oracle 12c 及时的对JSON进行了支持, 并非常有用
- 相信很快JSON 就会被得到应用
- JSON 是:
 - 与AJAX相互的一种最常见的机制
 - 成为网络服务数据最常见的机制(尤其是基于服务的REST/HTTP API)
 - 几种“大数据”数据存储的基石
- 花费时间来了解JSON, JSON 句法, 和 Oracle的实现

RMOUG Training Days 2016

2016年2月9-11日

(周二-周四)

丹佛会展中心



Tracks

- Application Development
- Business Intelligence
- Database Administration
- DBA Deep Dive
- Database Tools of the Trade
- Hyperion
- Middleware
- Professional Empowerment

PHOTO CREDIT: Mike Landrum, SQL Developer and the "Data Tsunami" from i-Behavior

www.rmoug.org



COLLABORATE 16 – IOUG 论坛, 2016年4月10-14日

**Mandalay Bay
Las Vegas, NV**



ODTUG Kscope16

SAVE THE DATE
CHICAGO, ILLINOIS
JUNE 26-30
www.kscope16.com





在Oracle 12c中使用JSON

联系作者:

John King

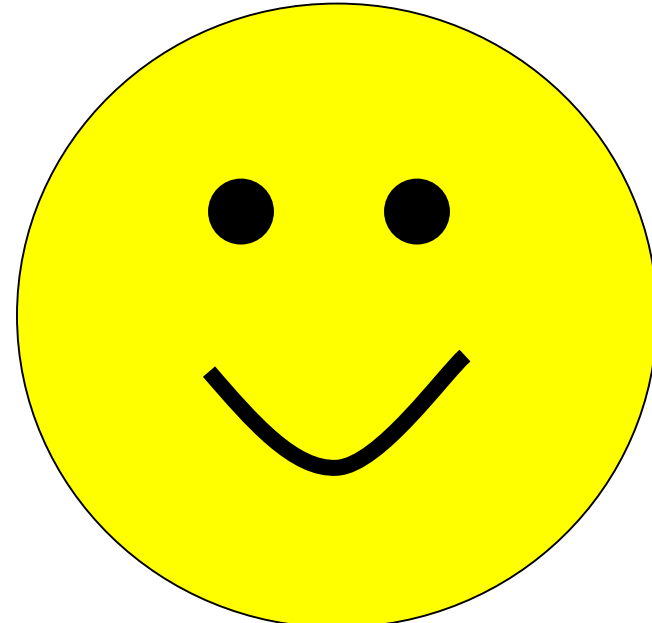
King Training Resources

P. O. Box 1780

Scottsdale, AZ 85252 USA

1.800.252.0652 - 1.303.798.5727

Email: john@kingtraining.com



感谢收听

可在该网站查看今天的幻灯片和例子:

<http://www.kingtraining.com>



- 結束