Cheatography

Oracle XML support Cheat Sheet

by Jianmin Feng (taotao) via cheatography.com/79308/cs/19476/

Why Oracle XML

XML

Extensible Markup Language	User defined tags
tags associated w/ the storage of data	
xml:Nothing but the presentation of data	HTML:present of web page
syntax	
<?xml version="1.0" encoding="ETF-8"?>	
tags: <Product> .... </Product>	
tags: case sensitive, attribute,val is not require	
free tool: html-kit	tag align, indent etc

Oracle XML DB

stand database feature	native support by XMLType
XMLPath and XQuery	XML<->relational data
File repository	
XMLType limitations	
performance Hit	used only when required
new API and design	

Relational data -> XML

XMLELEMNT	get one XMLType instance
XMLType.getcolbval	char format of xmltype data
XMLAttributes	set attribute
XMLForest	deal with col
XMLAgg	deal wit row
XMLElent(XMLForest)	get parent tag for group of col
XMLPi	xml processing instruction
XMLCommnet	add comment line

DBMS_XMLGen

```
--avoid complex syntax of using XML function to
create xml doc
-- create object view:s implify sql
  CREATE OR REPLACE VIEW VIEW_NAME OF OBJECT _TYPE
WITH OBJECT IDENTIFIER (COLUM N(s)) as SELECT
--using dbms_x mlgen to generate file and save
--dire ctory: pre-de fined & ALL CAPS
out_file utl_fi le.f il e_type;
out_file := utl_fi le.f open ('MY_X ML_ FIL -
ES' ,'f ile.xm l', 'W');
QueryC ontext dbms_x mlg en.c tx handle;
QueryC ontext := dbms_x mlg en.n ew context
('sql_ que ry');

dbms_x mlg en.s et row settag (Query Con text,
'xxx');
dbms_x mlg en.s et rowtag (Query Con text, 'xxx');

my_clob := dbms_x mlg en.g et xml (Qu ery Con -
text);
--write that XML to our external file
utl_fi le.put (out_file, my_clob);
utl_fi le.f close (out_f ile);
dbms_x mlg en.c lo sec ont ext (Qu ery Con text);
```

xml --> relational table

external.xml file	xmltype col type
get external file into col: xmltype(bfilename('DIR','fname.xml")	
Query xml col	
extract(xml_doc,'Path')	return xml
extract(xml_doc,'Path/text()')	return value only
existnode(xml_doc,'path').getdtringval	same as text()
existsnode(xml_doc, 'path')	if exists =1, else 0
dot notation	nested element
[i] to access array element	

to avoid access violation error(11g):
select XMLSERIALIZE (CONTENT xmldocument AS CLOB INDENT SIZE = 2) from xmldocuments

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Oracle XML support Cheat Sheet

by Jianmin Feng (taotao) via cheatography.com/79308/cs/19476/

XML_TABLE

```
VARIABLE v_job VARCHAR2(10);
EXEC :v_job := 'CLERK';
SELECT xt.*
FROM xml_tab x,
              XML TAB LE( '/e mpl oye es/ emp loy -
ee[ job =$job]'
                  PASSING x.xml_ data, :v_job AS
" job "
                  COLUMNS
                      empno VARCHA R2(4) PATH
'empno',
                      ename VARCHA R2(10) PATH
'ename',
                     job VARCHA R2(9) PATH 'job',
                      hir edate VARCHA R2(11) PATH
'hiredate,
                     id VARCHA R2(10) PATH '@id','
                ) xt;
-- :v_job AS " job "
--The variable must be aliases using AS and double
quoted to make sure the name and case matches that
of the variable in the XPath expres sion.
--
SELECT x.xml_ dat a.g etC lob Val()
FROM xml_tab x;
--nested XML_table
SELECT xt.*
FROM xml_tab x,
        XML TAB LE( '/d epa rtm ent s/d epa rtment'
        PASSING x.xml_data
          COLUMNS
                  deptno VARCHA R2(4) PATH
'dept_no',
                  XML TAB LE( '/e mpl oye es/ emp -
loyee'
                  PASSING dd.emp loyees
                  COLUMNS
                          empno varcha r2(4) PATH
'emp_no',
                          ename varcha r2(4) PATH
'emp_name'
```

XML_TABLE (cont)

```
>       )
 ) xt
--form xmltype in variable
PASSING xmltype(v_varchar2)
PASSING v_xml
```

XDB

Resource_view	
res	xmltype
any_path	varchar2
resid	raw
Path_view	
path	varchar2
res	xmltype
link	xmltype
resid	raw
xml function	

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com