Cheatography

First Scope of Json Cheat Sheet

by Jianmin Feng (taotao) via cheatography.com/79308/cs/19247/

What's JSON?

Java**S**cript **O**bject **N**otation

Language independent data format for storing and sharing

Lightweight,human readable,easy parsed/generated

Name is misleading somehow

JSON vs XML

Shorter,more lightweight

Wirteable, more readable

More easier to parse by programming

Easier to work with arrays

No namespace, comments not allowed

XML like a truck, suitable for large scale file based data exchange, while JSON like a compact car, good for small flexible data exchange over network, such as ajax

Examples XML vs JSON

```
<departments>
<department>
  <dept_id>1 </dept_id>
  <manager>Josh</manager>
  <employees>
    <employee>
      <id>7839</id>
      <name>John</name>
      <jobs>
        <job>
          <Title>Clerk</Tile>
        </job>
        <job>
          <Title>dba</Tile>
        </job>
      </jobs>
   </employee>
   <employee>
     <id>7698</id>
     <name>Tom</name>
     <jobs>
```

Examples XML vs JSON (cont)

```
        <job>
          <Title>Clerk</Tile>
        </job>
         <job>
          <Title>Analyst</Tile>
        </job>
      </jobs>
    </employee>
  </employees>
</department>
</departments>
{departments:[{
 "department":{
  "dept_id":"1",
  "employees":[{
   "employee":{
     "id":66",
     "name":"John",
     {jobs:["Clerk","dba"]}},
   "employee":{
    "id":88,
    "name":"Tom",
    {jobs:["Analyst"]}}
  ]}
}]}
```

JSON Structure

Object	{string1:value1,string2:value2,...}
array	[value1,value2,...]
value	string,number,true/false,null,object,array

Oracle JSON Support

12.1.0.2	store in varchar2,clob,blob w/ check constraint
Apex 5.0	Apex_JSON
pljson	github.com/pljson/pljson

NoSql - RDBMS - blender with JSON support

Load JSON By External Table

```
CREATE TABLE JSON_DEMO( id
number, json_data clob)
ORGANIZATION EXTERNAL
  ( TYPE ORACLE_LOADER
    DEFAULT DIRECTORY EXT_DIR
    ACCESS PARAMETERS
    ( records delimited by
newline
      fields terminated by
0X'09'
      missing field values are
null
      ( id,fname )
      COLUMN TRANSFORMS (
         json_data FROM LOBFIL-
E(fname) FROM (EXT_DIR) )
    )
    LOCATION (EXT_DIR:'json_-
file_list_to_be_load.txt')
  ) REJECT LIMIT UNLIMITED;
```

Content in json_file_list_to_be_load.txt
1 jason_data.json

Parse with APEX_JSON

1 select json_clob into v_json_data from json_demo where id=1;

2 APEX_JSON.parse(v_json_data); -- g_value variable hold the data

3 APEX_JSON.get_varchar2(v_json_data,'-dept');

4. APEX_JSON.get_count(v_json_data,'-dept.emp');

5 loop based on count

By **Jianmin Feng** (taotao)
cheatography.com/taotao/

Not published yet.
Last updated 22nd April, 2019.

Oracle JSON Dot Notation

```
select
jd.json_doc.departments[0].employees[0].name
from json_demo jd;
--dot notation:table_alias.json_col.key1.k-
ey2[0,1..],
--table must be aliased
--col must be checked IS JSON, no need to
enable
--key is case-senstive
```

dot notion is slower and easily confused, use json_table instead.

Oracle 12.1.0.2 JSON Condition

```
1. IS JSON
create table json_demo
(
 json_doc clob not null,
 ts date default sysdate
);
alter table json_demo
add constriant ck_is_json
check (json_doc IS JSON WITH UNIQUE KEYS
(STRICT));
--WITH UNIQUE KEYS: use care (performance)
--(STRICT): force strict over Lax syntax,
such as case sensitive," not ' etc
--ORA-02290: check constraint (TEST.DOCUME-
NT_JSON) violated
2. JSON_EXISTS(json_doc,'$.employee_no'
FALSE ON ERROR);
```

JSON doc saved to VARCHAR2, CLOB or BLOB with conditions

Oracle JSON Path

```
$.Title.Songs[*].producer[0].
```

array index could be * or ordered index, no duplicate

Oracle JSON_VALUE: retrieve scalar value

```
select JSON_VALUE(json_data,
'$.Title.Songs[].producer[0].',RETURNING
VARCHAR2,ERROR ON ERROR )
from json_demo;
--Mandatory: table & Col name, add
FORMAT JASON for BLOB
--Optional: Returning type - varchar2,
number,date, timestamp, 18c suppoert
CLOB,BLOB
--Optional: error NULL ON ERROR(def-
ault), ERROR ON ERROR
```

Oracle JSON_QUERY: wrap return into array

```
select JSON_QUERY(json_data,
'$.Songs[0].producer[*].Name WITH
WRAPPER) from json_demo;
--return type: only varchar2
--WITH WRAPPER: default WITOUT WRAPPER
--EMPTY ON ERROR(default)
```

Oracle JASON_TABLE:JSON doc --> table

```
select t.*
from json_demo,
  JSON_TABLE ( json_doc, '$.departmen-
ts[*]'
   COLUMNS (
    row_number FOR ORDINALITY,
    song varchar2(50) PATH '$.Title',
    artist varchar2(50) PATH '$.Artist,
    ESTED PATH '$.producer[*]'
    columns(
       company varchar2(50) path '$.com-
pany'
        )
 ))
AS t;
--original table must included in from
--Column path is from table path
--FOR CARDINALITY: get row number
```

NESTED PATH only available for 12.2 above

FORMAT JSON

implict for all json column
explictly specify if column type is BLOB

{USER|ALL|DBA}_-JSON_COLUMNS Views

```
SELECT table_nam-
e,column_name,f-
ormat,data_type
FROM user_json_co-
lumns;
```

JSON_TEXTCONTAINS

```
1. context search index(full text search or json
search idnex
--12.1
CREATE INDEX json_docs_search_idx ON json_doc(-
data)
   INDEXTYPE IS CTXSYS.CONTEXT
   PARAMETERS ('section group CTXSYS.JSON_SECTIO-
N_GROUP SYNC (ON COMMIT)');
--12.2
CREATE SEARCH INDEX json_docs_search_idx ON
json_doc(data) FOR JSON;
EXEC DBMS_STATS.gather_table_stats(USER, 'JSON_-
DOCUMENTS');
2. query
SELECT COUNT(*) FROM json_doc
WHERE JSON_TEXTCONTAINS(data, '$.ContactDetails.E-
mail', 'john.doe@example.com');
SELECT COUNT(*) FROM json_documents
WHERE JSON_EXISTS(data, '$.ContactDetails');
```