

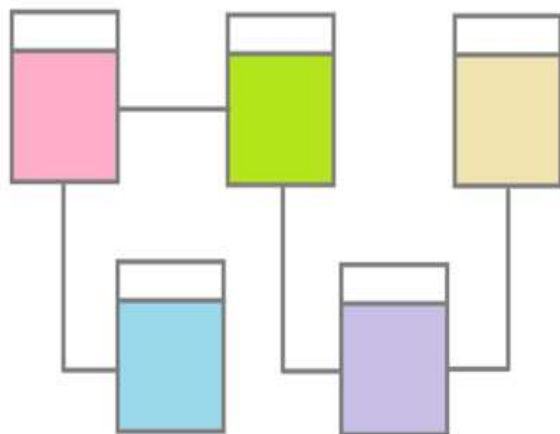
SQL/JSON ORACLE

Графеева НГ

2020

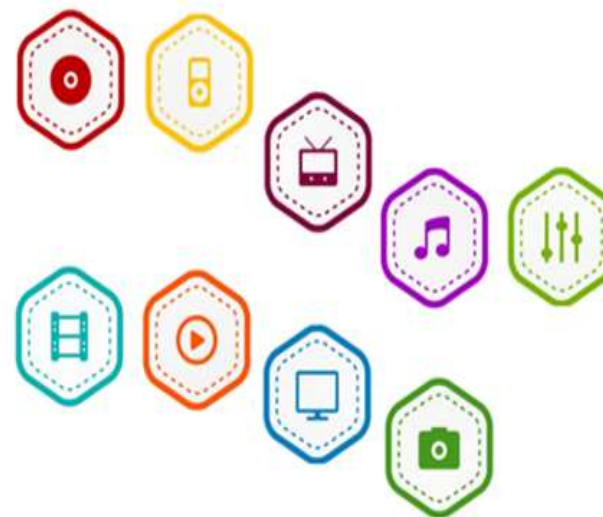
SQL

Жесткая структура



NoSQL

Слабоструктурированные данные



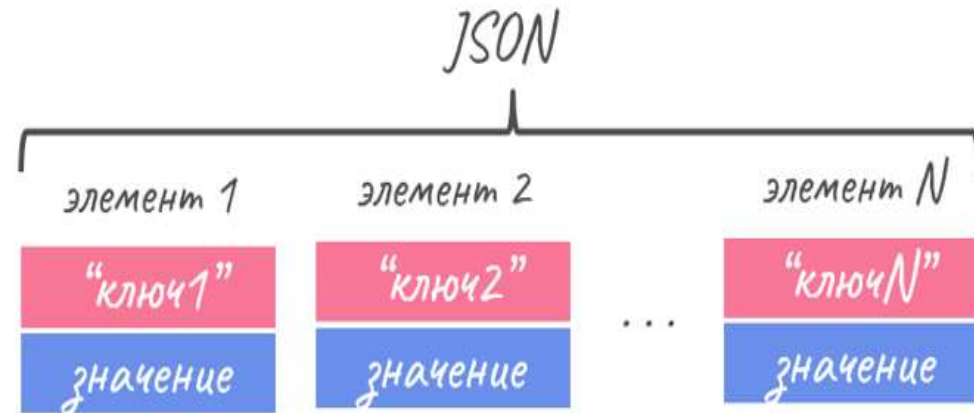
Примеры слабоструктурированных данных



XML	JSON
<pre data-bbox="955 439 1607 1228"><Cartoon> <CartoonName>The Lion King</CartoonName> <Year>1994</Year> <Country>USA</Country> <Duration>88</Duration> <FilmDirector> <Name>Roger Allers</Name> <Name>Rob Minkoff</Name> </FilmDirector> </Cartoon></pre>	<pre data-bbox="1694 425 2181 911">{ "CartoonName": "The Lion King", "Year": 1994, "Country": "USA", "Duration": 88, "FilmDirector": ["Roger Allers", "Rob Minkoff"] }</pre>

Структура JSON

JavaScript Object Notation



Шаблон структуры JSON

```
{ "ключ1": значение1,  
  "ключ2": значение2,  
  ....  
  "ключN": значениеN }
```

Пример записи структуры JSON

```
{  
  "CartoonName": "The Lion King",  
  "Year": 1994,  
  "Country": "USA",  
  "Duration": 88,  
  "FilmDirector": [  
    "Roger Allers",  
    "Rob Minkoff"  
  ]  
}
```

строка

число

строка

число

массив

Пример JSON коллекции

```
{  
  "CartoonName": "The Lion King",  
  "Year": 1994,  
  "Country": "USA",  
  "Duration": 88,  
  "FilmDirector": [  
    "Roger Allers",  
    "Rob Minkoff"  
  ]  
}
```



```
{  
  "CartoonName": "Cinderella",  
  "Year": 1950,  
  "Country": "USA",  
  "Duration": 75  
}
```



```
{  
  "CartoonName": "Ugly Duckling",  
  "Year": 1956,  
  "Country": "USSR",  
  "Duration": 18  
}
```

JSON в ORACLE?

- Данные из простых реляционных таблиц могут выгружаться в формате JSON. Для этого есть специальное расширение SQL (**SQL/JSON**).
- Данные из JSON документов могут загружаться в структуры ORACLE (оператор INSERT или SQL Workshop -> UTILITES-> Data Workshop->Load).
- В таблицах реляционной базы ORACLE можно создавать поля для структур JSON.
- Для хранения таких столбцов (пока!) используются типы VARCHAR, BLOB, CLOB.
- Структура (формат) таких данных поддерживается специфическим правилом целостности (IS JSON).
- Доступ к данным внутри JSON документа обеспечивается специальными операторами SQL (расширение).

Как создать таблицу с полем JSON?

```
CREATE TABLE js_table  
  (ID NUMBER PRIMARY KEY,  
   DDATE DATE,  
   js_document VARCHAR2 (3200)  
   CONSTRAINT ensure_json CHECK (js_document IS JSON));
```


Как добавить значение в таблицу с полем JSON?

```
INSERT INTO js_table
VALUES (1,
       sysdate,
       '{ "PONumber"           : 1600,
         "Reference"          : "ABULL-20140421",
         "Requestor"          : "Alexis Bull",
         "User"                : "ABULL",
         "CostCenter"          : "A50",
         "Special Instructions" : null,
         "AllowPartialShipment" : true
       }'
);
```

Как выбрать из таблицы JSON-документ?

```
SELECT js_document FROM js_table
```

```
20  
21  
22
```

```
SELECT js_document FROM js_table
```

Results Explain Describe Saved SQL History

JS_DOCUMENT

```
{"PONumber" : 1600, "Reference" : "ABULL-20140421", "Requestor" : "Alexis Bull", "User" : "ABULL", "CostCenter" : "A50", "Special Instructions" : null, "AllowPartialShipment" : true }
```

1 rows returned in 0.02 seconds [Download](#)

Как выбрать из JSON документа отдельное значение?

```
SELECT js_document.PONumber  
FROM js_table where id = 1
```

нет!!!

```
SELECT p1.js_document.PONumber  
FROM js_table p1 where id = 1
```

да!!!

Как редактировать содержимое JSON документов?

```
UPDATE js_table
SET js_document =
    json_mergepatch(js_document,
        '{"Special Instructions":"DO NOT TOUCH!"}')
WHERE ID = 1;
```

Как генерировать JSON-документы из обыкновенных таблиц?

```
SELECT json_object
('name' VALUE ename ,
 'hasCommission' VALUE
CASE WHEN comm IS NULL THEN 'false' ELSE 'true'
END FORMAT JSON
) js_doc
FROM emp ;
```

```

29
30 SELECT json_object('name'          VALUE ename ,
31                   'hasCommission' VALUE
32                   CASE WHEN comm IS NULL THEN 'false' ELSE 'true'
33                   END FORMAT JSON) js_doc
34 FROM emp ;
35
36

```

Results

Explain

Describe

Saved SQL

History

JS_DOC

```
{"name":"KING","hasCommission":false}
```

```
{"name":"BLAKE","hasCommission":false}
```

```
{"name":"CLARK","hasCommission":false}
```

```
{"name":"JONES","hasCommission":false}
```

```
{"name":"SCOTT","hasCommission":false}
```

```
{"name":"FORD","hasCommission":false}
```

```
{"name":"SMITH","hasCommission":false}
```

Как генерировать JSON-документы из обыкновенных таблиц?

```
SELECT json_objectagg(dname VALUE deptno)  
dept_list  
FROM dept;
```

```
36 | SELECT json_objectagg(dname VALUE deptno) dept_list FROM dept;  
37  
38
```

Results[Explain](#)[Describe](#)[Saved SQL](#)[History](#)**DEPT_LIST**

```
{"NOT ACC":10,"RESEARCH":20,"SALES":30,"OPERATIONS":40}
```


Как генерировать JSON-документы из нескольких таблиц?

```
SELECT json_object
( 'name'      VALUE ename ,
  'salary'    VALUE sal,
  'hiredate'  VALUE hiredate,
  'hasCommission' VALUE
(CASE WHEN comm IS NULL THEN 'false' ELSE 'true' END) ,
  'dname'     VALUE
(select dname from dept where dept.deptno=emp.deptno)
  FORMAT JSON) js_doc
FROM emp ;
```

Как генерировать JSON-документы из нескольких таблиц?

```
SELECT json_object
('name' VALUE ename ,
 'salary' VALUE sal,
 'hiredate' VALUE hiredate,
 'hasCommission' VALUE
(CASE WHEN comm IS NULL THEN 'false' ELSE 'true' END) ,
 'department' VALUE
json_object('dname' VALUE dname, 'deptno' VALUE dept.deptno)
FORMAT JSON) js_doc
FROM emp,dept WHERE emp.deptno = dept.deptno ;
```

```

37
38     SELECT json_object('name' VALUE ename ,
39                        'salary' VALUE sal,
40                        'hiredate' VALUE hiredate,
41                        'hasCommission' VALUE (CASE WHEN comm IS NULL THEN 'false' ELSE 'true' END) ,
42                        'department' VALUE json_object('dname' VALUE dname, 'deptno' VALUE dept.deptno)
43                        FORMAT JSON) js_doc
44 FROM emp,dept WHERE emp.deptno = dept.deptno ;
45
46

```

Results

[Explain](#)
[Describe](#)
[Saved SQL](#)
[History](#)

JS_DOC

```
{"name":"MILER","salary":1477,"hiredate":"1982-01-23T00:00:00","hasCommission":"false","department":{"dname":"NOT ACC","deptno":10}}
```

```
{"name":"CLARK","salary":2627,"hiredate":"1981-06-09T00:00:00","hasCommission":"false","department":{"dname":"NOT ACC","deptno":10}}
```

```
{"name":"KING","salary":5177,"hiredate":"1981-11-17T00:00:00","hasCommission":"false","department":{"dname":"NOT ACC","deptno":10}}
```

```
{"name":"ANDREW","salary":5207,"hiredate":"2016-02-01T00:00:00","hasCommission":"false","department":{"dname":"NOT ACC","deptno":10}}
```

```
{"name":"SMITH","salary":977,"hiredate":"1980-12-17T00:00:00","hasCommission":"false","department":{"dname":"RESEARCH","deptno":20}}
```

```
{"name":"FORD","salary":3192,"hiredate":"1981-12-03T00:00:00","hasCommission":"false","department":{"dname":"RESEARCH","deptno":20}}
```

```
{"name":"JONES","salary":3152,"hiredate":"1981-04-02T00:00:00","hasCommission":"false","department":{"dname":"RESEARCH","deptno":20}}
```

```
{"name":"ADAM","salary":1277,"hiredate":"1983-01-12T00:00:00","hasCommission":"false","department":{"dname":"RESEARCH","deptno":20}}
```

```
{"name":"SCOTT","salary":7692,"hiredate":"1982-12-09T00:00:00","hasCommission":"false","department":{"dname":"RESEARCH","deptno":20}}
```

Зачетное задание 12

- Сгенерировать JSON документ по данным своей предметной области.
- Создать приложение, которое демонстрирует данные о 5 любых фильмах (актеры, режисеры, год выпуска и т.п). Данные про фильмы загрузить как JSON-документы.