
ORACLE®

免責事項

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント（確約）するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

Oracle Database 12*c* Release 1 (12.1.0.2) CoreTech Seminar

JSON対応

日本オラクル株式会社
データベース事業統括 製品戦略統括本部
データベースエンジニアリング本部 Database & Exadata技術部
井上 克己
2014/08/18

Agenda

1. JSONデータのOracle Databaseへの格納
2. 実装 – 関数、条件
3. JSONデータの索引付け

JSONとは

- JavaScript Object Notationの頭文字
- 右下は最新の RFC より http://tools.ietf.org/html/rfc7159
- 構成要素
  - スカラー値
  - オブジェクト
  - 配列(array)

JSON can represent four primitive types (strings, numbers, booleans, and null) and two structured types (objects and arrays).

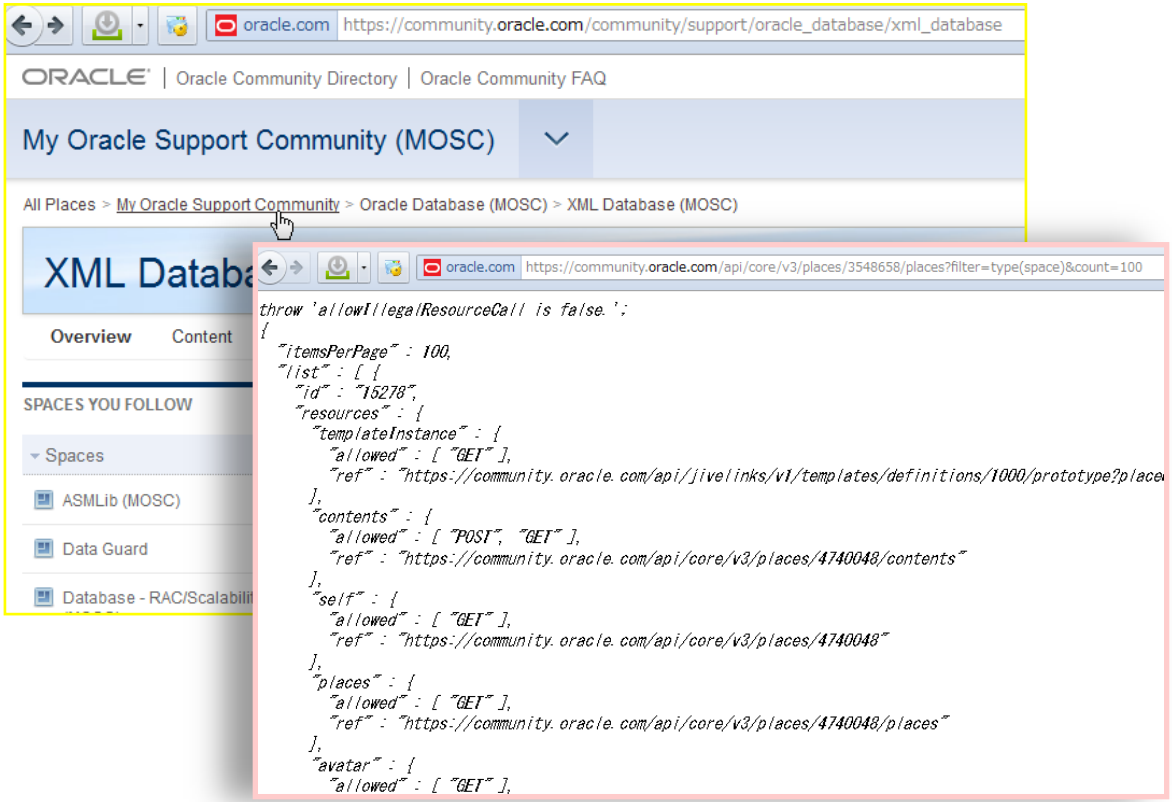
JSONドキュメント例: Purchase Order(発注書)

```
{
  "PONumber" : 1600,
  "Reference" : "ABULL-20140421“,
  "Requestor" : "Alexis Bull“,
  "User" : "ABULL“,
  "CostCenter" : "A50“,
  "ShippingInstructions" : {
      "name" : "Alexis Bull“,
      "Address" : { ... },
      "Phone" : [ ... ]
  },
  "Special Instructions" : null,
  "AllowPartialShipment" : true,
  "LineItems" : [{
      "ItemNumber" : 1,
      "Part" : {
          "Description" : "One Magic Christmas“,
          "UnitPrice" : 19.95,
          "UPCCode" : 13131092899
      },
      "Quantity" : 9
  },
     {...}
  ]
}
```

{ … } (波括弧): オブジェクト
[…] (角括弧): 配列(array, collection)

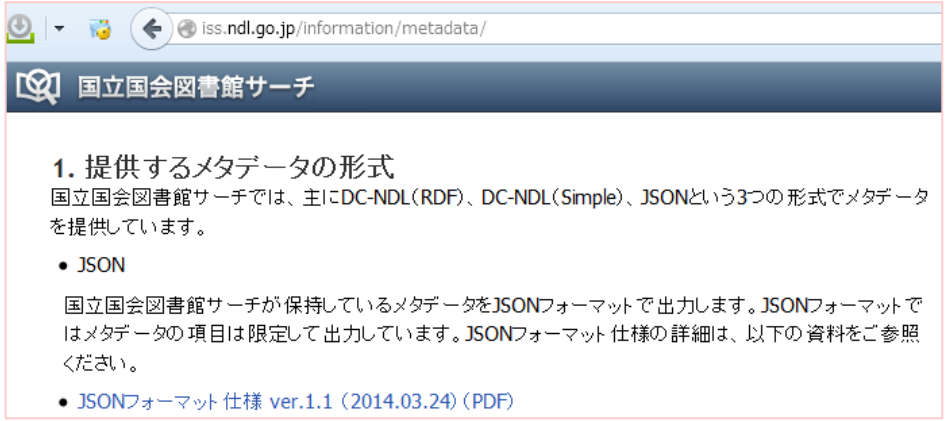
HTTPでのJSON使用例

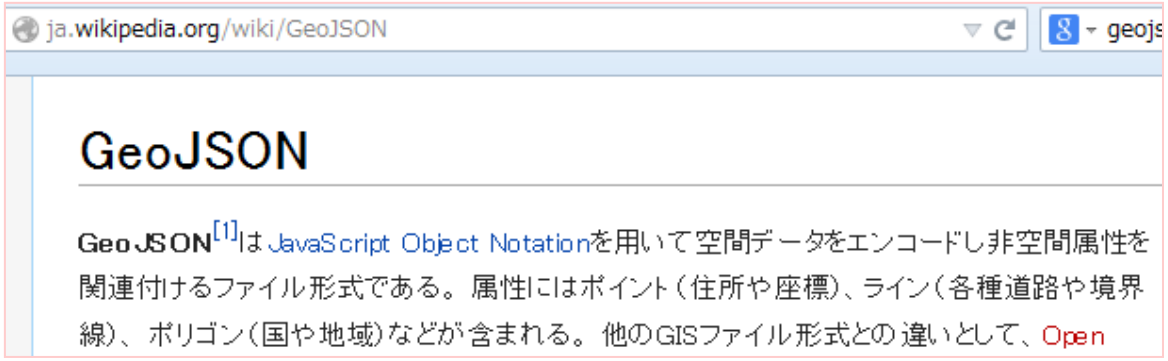
- JavaScriptはブラウザに実装されている
- ブラウザはJSONデータを受け取る



標準的なデータ交換フォーマットとしてのJSON

- XML、CSVなどの代替として使われるケースが多くなっている
- HDFSなどのファイルシステムに配置すれば外部テーブルとして参照可能



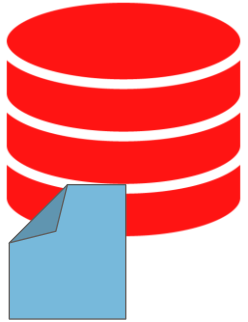


JSONデータのデータベース内のフォーマット

リレーショナル表 または JSON ドキュメント のまま

https://myhost/myapp/studentreg/1001

HTTP(s)



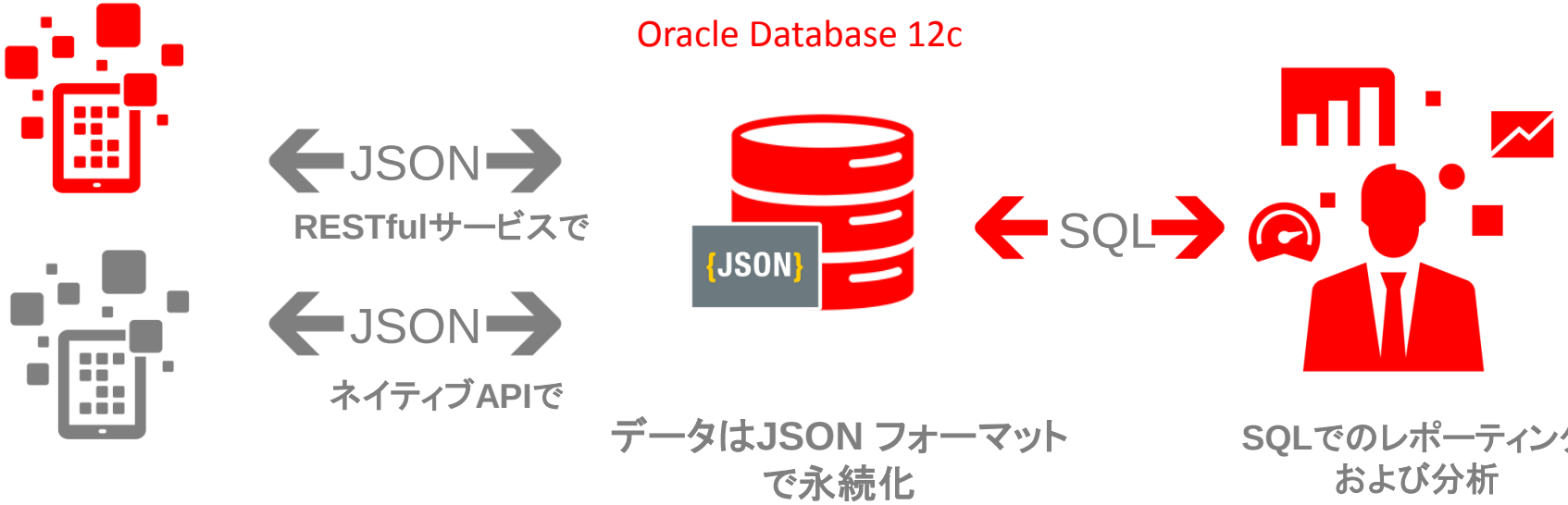
JSON

```
{ “studentno”:  1001,
   “name”:   “Scott King”,
   “address”:  “500 Main street, Innovation CA”,
   “registrations”: [ { “regnumber”:  404,
                        “regdate”: “Feb 27, 2014”,
                        “status”: “in process”},
                      { “regnumber”:  303,
                        “regdate”: “Feb 26, 2014”,
                        “status”: “in process”}] }
```

学生番号	名前	住所	その他
3134	John	Ave.	
834	Doe	Street	

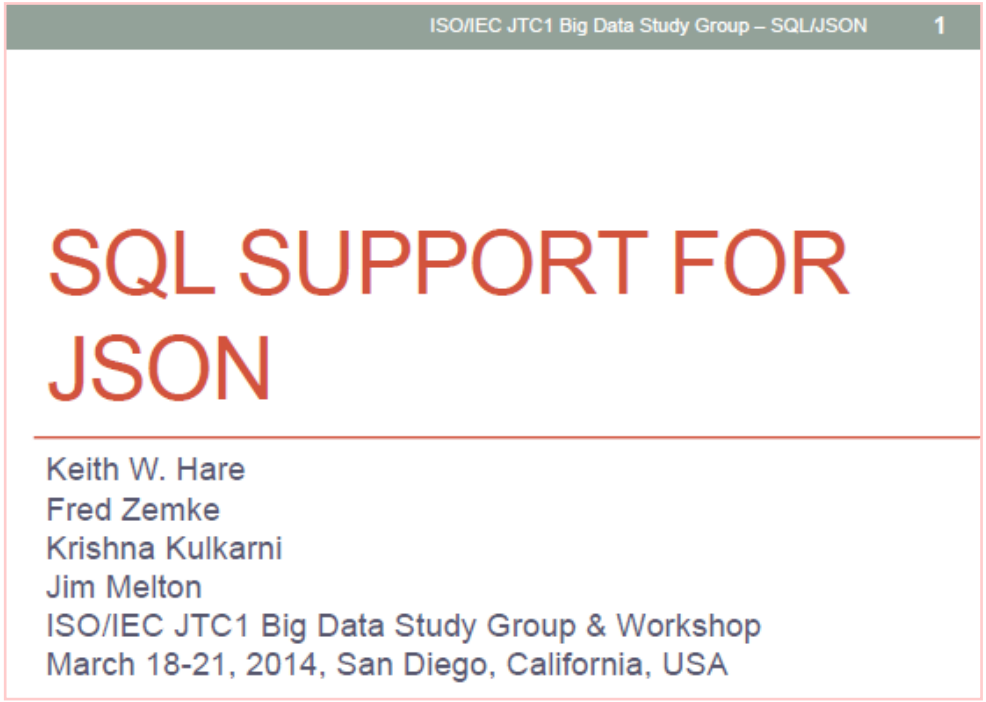
Oracle Database 12*c* 12.1.0.2 : JSONサポート

フレキシブルなアプリ開発 とSQLでの分析の両方 を可能に



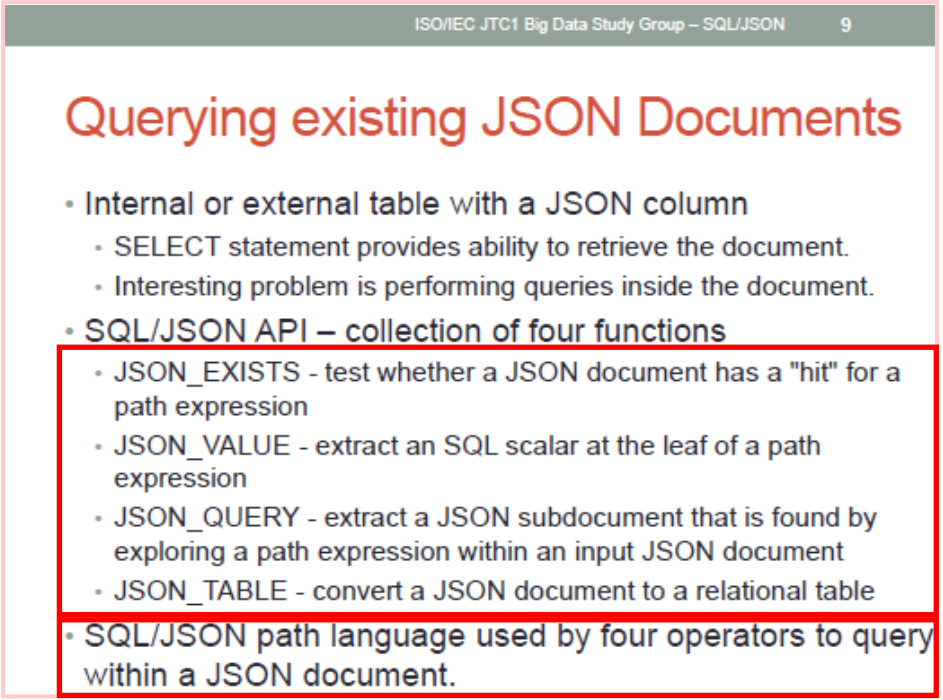
SQL/JSON: 標準として提唱中

- http://jtc1bigdatasg.nist.gov/_workshop/08_SQL_Support_for_JSON_abstract.pdf
  - 2014年3月に提出されたもの
- XMLでのSQL/XMLに相当



プロポーザル内容

- 4つの関数をOracle Database 12*c*(12.1.0.2)で同名で実装
- SQL/JSON Path



Agenda

1. JSONデータのOracle Databaseへの格納
2. 実装 – 関数、条件
3. JSONデータの索引付け

JSONドキュメントストアとしてのOracle Database

- JSONは既存のデータ型を使用
  - VARCHAR2、CLOB、BLOB
    - 内部的にはUnicode(UTF8)で処理される
    - BLOBではキャラクタセット変換が発生しない
  - JSON用新データ型はない
    - XML DBでのXMLType相当のものは存在しない

```
create table J_PURCHASEORDER (
  ID           RAW(16) NOT NULL,
  DATE_LOADED   TIMESTAMP(6) WITH TIME ZONE,
  PO_DOCUMENT CLOB CHECK (PO_DOCUMENT IS JSON)
)
```

条件、関数 一覧

条件または関数名	機能
IS JSON 条件	JSON として整形式かどうかをチェック
JSON_EXISTS()	JSON の要素が存在するかを確認するSQL関数
JSON_VALUE()	JSON の条件に合致する特定の要素を取得するSQL関数
JSON_QUERY()	JSON から条件に合致する全ての要素を取得するSQL関数
JSON_TABLE()	JSON から表形式でデータを取得するSQL関数
JSON_TEXTCONTAINS()	JSON の特定の要素に指定したテキストが含まれるかどうかを確認するSQL関数

Oracle JSON Path(記法/表現)

- XMLでの"XPath"に相当
- "$"がJSONドキュメント全体のルート
- . (ドット) を階層区切りとする記法
  - XMLでは '/' (スラッシュ)
- 配列部分のインデックスには数字、*、範囲を指定可能
  - 例: $.LineItems[2], $.LineItems[3 to 8]
- JSON関数の多くがJSON Pathを引数とする
  - SQL文中には直接記述することができない
- "SQL/JSON Path"として提案されているものに準ずる

Oracle JSON Pathの使用例

JSON Path表現	型	内容
$.Reference	String	"ABULL-20120421"
$.ShippingInstructions.Address.zipcode	Number	99236
$.ShippingInstructions.Address	Object	{ "street": "200 Sporting Green", "city": "South San Francisco", "state": "CA", "zipCode": 99236, "country": "United States of America" }
$.LineItems[1]	Object	{ "ItemNumber" : 2, "Part" : { "Description" : "Lethal Weapon", "UnitPrice" : 19.95, "UPCCode" : 85391628927 }, "Quantity" : 5 }
$.LineItems[*].UPCCode	Array	[13131092899, 85391628927]

より簡易な . (ドット)記法

- . (ドット)を階層区切りとする記法
- SQL文中に直接記述可能
- FROM句のテーブルにエイリアスを指定
- ADT(Abstract Data Type、ユーザー定義型)使用時と同じ記法

```
select   j.PO_DOCUMENT.Reference,
         j.PO_DOCUMENT.Requestor,
         j.PO_DOCUMENT.CostCenter,
         j.PO_DOCUMENT.ShippingInstructions.Address.city
  from J_PURCHASEORDER      j
 where j.PO_DOCUMENT.PONumber = 1600
```

表別名

カラム名

ドット記法でのJSONデータへのクエリー実行例

アプリケーション開発者:

RESTful APIでJSONを永続化

```
PUT /my_database/my_schema/customers HTTP/1.0
Content-Type: application/json
Body:
{
 "firstName": "John",
 “lastName”: "Smith",
 "age": 25,
 "address": {
            "streetAddress": "21 2nd Street",
            "city": "New York",
            "state": "NY",
            "postalCode": "10021“,
            "isBusiness" : false  },
  "phoneNumbers": [
            {"type": "home",
             "number": "212 555-1234“ },
            {"type": "fax",
             "number": "646 555-4567“ }  ]
}
```

分析ツールおよびビジネスユーザー:
JSONをSQLで問い合わせ

```
select
  c.document.firstName,
  c.document.lastName,
  c.document.address.city
from customers c;
```

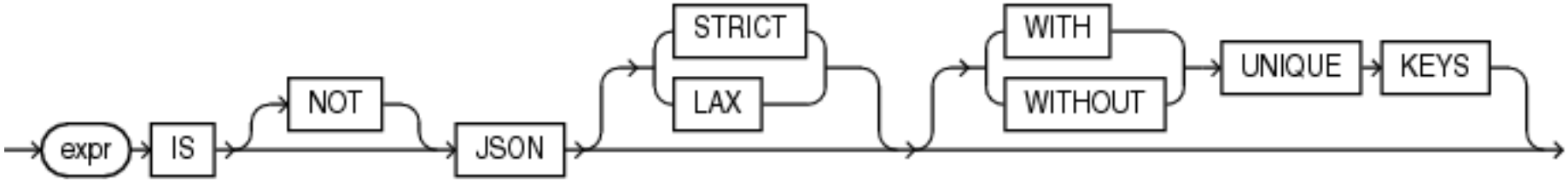
firstName	lastName	address.city
“John”	“Smith”	“New York”

IS JSON条件

- "IS NULL"と同様に使用することが可能

Contents

- Oracle Database SQL Language Reference
  - Preface
  - Changes in This Release for Oracle Database SQL Language Reference
  - Introduction to Oracle SQL
  - Basic Elements of Oracle SQL



IS JSON条件

- 厳密なフォーマット仕様に沿っているかどうかのオプション
  - IS JSON (STRICT)
  - IS JSON (LAX): "許容" オプション
    - こちらがデフォルト
- 例:
  - STRICT 追加により 2行 除外される

```
SELECT col1
  FROM t
  WHERE col1 IS JSON  STRICT

COL1
--------------------------------------------
[ "LIT192", "CS141", "HIS160" ]
{ "Name": "John" }
{ "Grade Values" : { A : 4.0, B : 3.0, C : 2.0 } }
{ "isEnrolled" : true }
{ "isMatriculated" : False }
```

IS JSONチェック制約

- SQL文でJSONデータを挿入する場合の例

```
create table J_PURCHASEORDER (
  ID                    RAW(16) NOT NULL,
  PO_DOCUMENT    CLOB
  CHECK (PO_DOCUMENT    IS JSON)
)
```

```
insert into J_PURCHASEORDER values('0x1','{Invalid JSON Text}');
ERROR at line 1:
ORA-02290: チェック制約(SYS.JSON_CONSTRAINT)に違反しました
```

SQL/JSON関数: JSON_TABLE()

- JSONデータ を リレーショナル なフォーマットへ 切り出し
- JSON配列 または JSONオブジェクト から 行 を生成
- JSONオブジェクトのプロパティを カラム へ マップ
- VIEW作成、他の表とのジョイン操作など可能
- 'NESTED PATH'句により深くネストされた場所にある配列からも切り出し可能

JSON_TABLE() 使用例

```
SQL> select M.*
  2   from J_PURCHASEORDER p,
  3       JSON_TABLE(
  4        p.PO_DOCUMENT,
  5        '$'
  6        columns
  7           PO_NUMBER          NUMBER(10)          path  '$.PONumber',
  8           REFERENCE          VARCHAR2(30 CHAR)   path  '$.Reference',
  9           REQUESTOR          VARCHAR2(32 CHAR)   path  '$.Requestor',
 10           USERID             VARCHAR2(10 CHAR)   path  '$.User',
 11           COSTCENTER         VARCHAR2(16)        path  '$.CostCenter'
 12       ) M
 13   where PO_NUMBER > 1600 and PO_Number < 1605
```

JSON ドキュメント カラム

JSON Path 表現

JSON_TABLE() 出力例

1発注につき1行出力

{"PONumber":1600,"Reference":"ABULL-20140421","Requestor":"Alexis Bull","User":"ABULL","CostCenter":"A50","ShippingInstructions":{"name":"Alexis Bull","Address":{"street":"200 Sporting Green","city":"South San Francisco","state":"CA","zipCode":99236,"country":"United States of America"},"Phone":[{"type":"Office","number":"909-555-7307"},{"type":"Mobile","number":"415-555-1234"}]},"Special Instructions":null,"AllowPartialShipment":true,"LineItems":[{"ItemNumber":1,"Part":{"Description":"One Magic Christmas","UnitPrice":19.95,"UPCCode":13131092899},"Quantity":9.0},{"ItemNumber":2,"Part":{"Description":"Lethal Weapon","UnitPrice":19.95,"UPCCode":85391628927},"Quantity":5.0}]}

PO_NUMBER	REFERENCE	REQUESTOR	USERID	COSTCENTER
1600	ABULL-20140421	Alexis Bull	ABULL	A50
1601	ABULL-20140423	Alexis Bull	ABULL	A50
1602	ABULL-20140430	Alexis Bull	ABULL	A50
1603	KCHUNG-20141022	Kelly Chung	KCHUNG	A50
1604	LBISSOT-20141009	Laura Bissot	LBISSOT	A50

JSON_TABLE() 使用例：NESTED PATH句

```
SQL> select D.*
  2   from J_PURCHASEORDER p,
  3       JSON_TABLE(
  4         p.PO_DOCUMENT,
  5         '$'
  6         columns(
  7           PO_NUMBER          NUMBER(10)                path  '$.PONumber',
  8           NESTED PATH                                        '$.LineItems[*]'
  9           columns(
 10             ITEMNO           NUMBER(16)                path '$.ItemNumber',
 11             UPCCODE          VARCHAR2(14 CHAR)         path '$.Part.UPCCode' ))
 12       ) D
 13   where PO_NUMBER = 1600 or PO_NUMBER = 1601
```

JSON_TABLE出力

各明細の配列エントリ毎に1行 出力


```
{"PONumber":1600,"Reference":"ABULL-20140421","Requestor":"Alexis
Bull","User":"ABULL","CostCenter":"A50","ShippingInstructions":{"name":"Alexis
Bull","Address":{"street":"200 Sporting Green","city":"South San
Francisco","state":"CA","zipCode":99236,"country":"United States of
America"},"Phone":[{"type":"Office","number":"909-555-
7307"},{"type":"Mobile","number":"415-555-1234"}]},"Special
Instructions":null,"AllowPartialShipment":true,"LineItems":[
{"ItemNumber":1,"Part":{"Description":"One Magic
Christmas","UnitPrice":19.95,"UPCCode":13131092899},"Quantity":9.0},
{"ItemNumber":2,"Part":{"Description":"Lethal
Weapon","UnitPrice":19.95,"UPCCode":85391628927},"Quantity":5.0}]}
```


PO_NUMBER	ITEMNO	UPCCODE
1600	1	13131092899
1600	2	85391628927
1601	1	97366003448
1601	2	43396050839
1601	3	13131119695
1601	4	25192032325

Agenda

1. JSONデータのOracle Databaseへの格納
2. 実装 – 関数、条件
3. JSONデータの索引付け

2種類の索引

- 定型検索のための索引
  - JSON_VALUE()関数 を使ったファンクション索引
    - Bツリー索引、または、ビットマップ索引
  - JSON_EXISTS()関数を 使ったビットマップ索引
  - IS JSONを使ったビットマップ索引
- アド・ホック検索のための索引
  - ドキュメント上では"JSON Search Index" と表記
  - JSONドキュメント全体のどこかに特定の文字列が含まれるか?
  - 任意のプロパティに特定の文字列が含まれるか?

JSON Search索引

- 実体はOracle Text索引
  - "ドメイン"索引: 特定の"領域"に特化した索引 例) 画像用
- JSON_TEXTCONTAINS()ではText索引がない場合はエラーが発生
  - Oracle TextのCONTAINS()に相当する関数

ORA-40467: JSON_TEXTCONTAINS() cannot be evaluated without JavaScript Object Notation (JSON) index

- JSON_EXISTS()、JSON_VALUE()でも使用される

Oracle Textの新機能

- JSONフォーマットを解釈し、索引付け

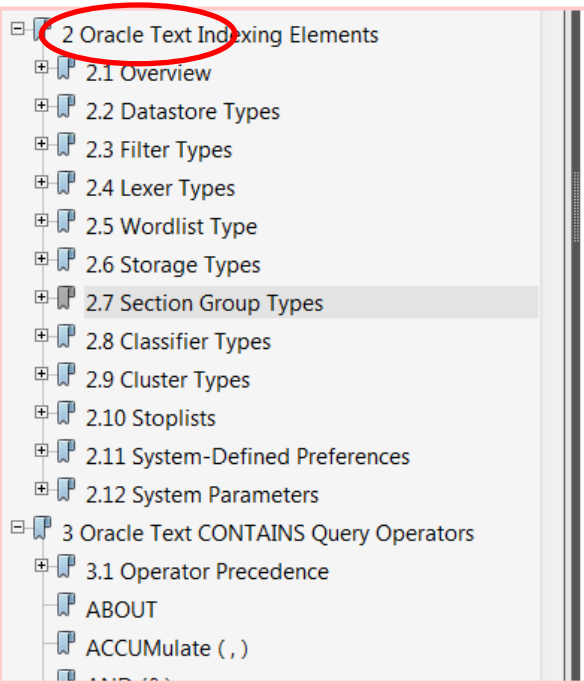


Table 2–42 Section Group Types

Type	Description
NULL_SECTION_GROUP	Use this group type when you define no sections or when you define *only* SENTENCE or PARAGRAPH sections. This is the default.
BASIC_SECTION_GROUP	Use this group type for defining sections where the start and end tags are of the form <A> and </A>. Note: This group type does not support input such as unbalanced parentheses, comments tags, and attributes. Use HTML_SECTION_GROUP for this type of input.
HTML_SECTION_GROUP	Use this group type for indexing HTML documents and for defining sections in HTML documents.
JSON_SECTION_GROUP	Use this group to create a JSON enabled context index. The JSON ENABLE attribute cannot be used with XML ENABLE. A section group can only be marked as JSON ENABLE. If it is already marked with XML ENABLE, then the path section group cannot be used for JSON ENABLE and vice versa.
XML_SECTION_GROUP	Use this group type for indexing XML documents and for defining sections in XML documents. All sections to be indexed must be manually defined for this group.

JSON Search索引の作成と使用

- JSONカラムにOracle Text索引作成

```
CREATE INDEX json_idx3 ON customersテスト (customer_infoテスト)
INDEXTYPE IS CTXSYS.CONTEXT PARAMETERS ('section group
CTXSYS.JSON_SECTION_GROUP')
```

- ドキュメント・ルートレベルの苗字プロパティ({"name_last":"Doe"}) でクエリー

```
SELECT customer_infoテスト FROM customersテスト
  WHERE JSON_VALUE(customer_infoテスト, '$.name_last') = 'Kochhar'
```

JSON Search索引が使用されたことを確認

- 実行計画を確認

```
| Id  | Operation                     | Name          | Rows| Bytes | Cost (%CPU)| Time|
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |               |    1 |   514 |    4   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS BY INDEX ROWID| CUSTOMERS _I|    1 |   514 |    4   (0)| 00:00:01 |
|*  2 |   DOMAIN INDEX                | JSON_IDX3     |      |       |    4   (0)| 00:00:01 |
```

- Predicate Informationを確認

```
1 - filter(JSON_VALUE("CUSTOMER_INFOテスト" FORMAT JSON , '$.name_last' RETURNING
    VARCHAR2(4000) NULL ON ERROR)='Kochhar')
2 - access("CTXSYS"."CONTAINS"("CUSTOMERSテスト"."CUSTOMER_INFOテスト",'{Kochhar}'
     INPATH(/name_last)')>0)
```

JSON Search索引使用時の注意点

- VARCHAR2、CLOBを使用時はキャラクターセットとしてUTF8の使用が必須
  - JA16SJIS、JA16EUC などのキャラクタセットには未対応

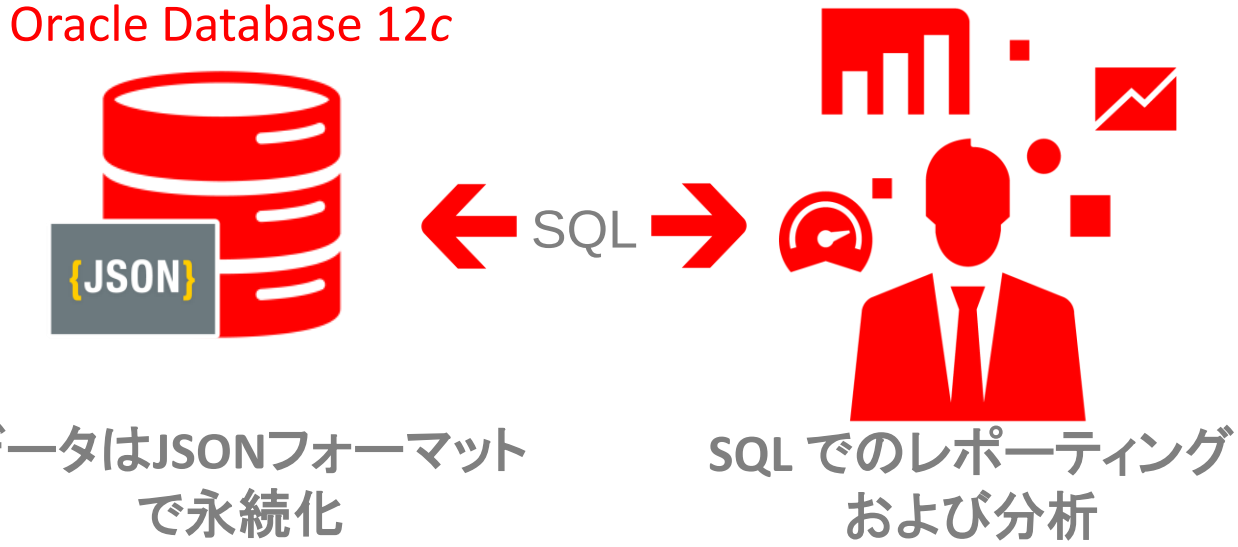
SQL> ALTER INDEX JSON_IDX REBUILD ONLINE;

…

ORA-20000: Oracle Text error:
ORA-06515: PL/SQL: unhandled exception json context index creation not supported in non-UTF8DB

まとめ

- JSONフォーマットデータにSQL でアクセスする機能が Oracle Database 12*c* (12.1.0.2)で実装



リファレンス

マニュアル・ドキュメント

- **Oracle® XML DB開発者ガイド 12cリリース1 (12.1)**
  - **39 Oracle DatabaseのJSON (新規に追加)**
    http://docs.oracle.com/cd/E57425_01/121/ADXDB/json.htm#CACGCBEG
- **Oracle® Textリファレンス 12cリリース1 (12.1)**
    http://docs.oracle.com/cd/E57425_01/121/CCREF/toc.htm
- **Oracle® Database SQL言語リファレンス 12cリリース1 (12.1)**
  - **JSON_QUERY**
    http://docs.oracle.com/cd/E57425_01/121/SQLRF/functions090.htm


ORACLE®