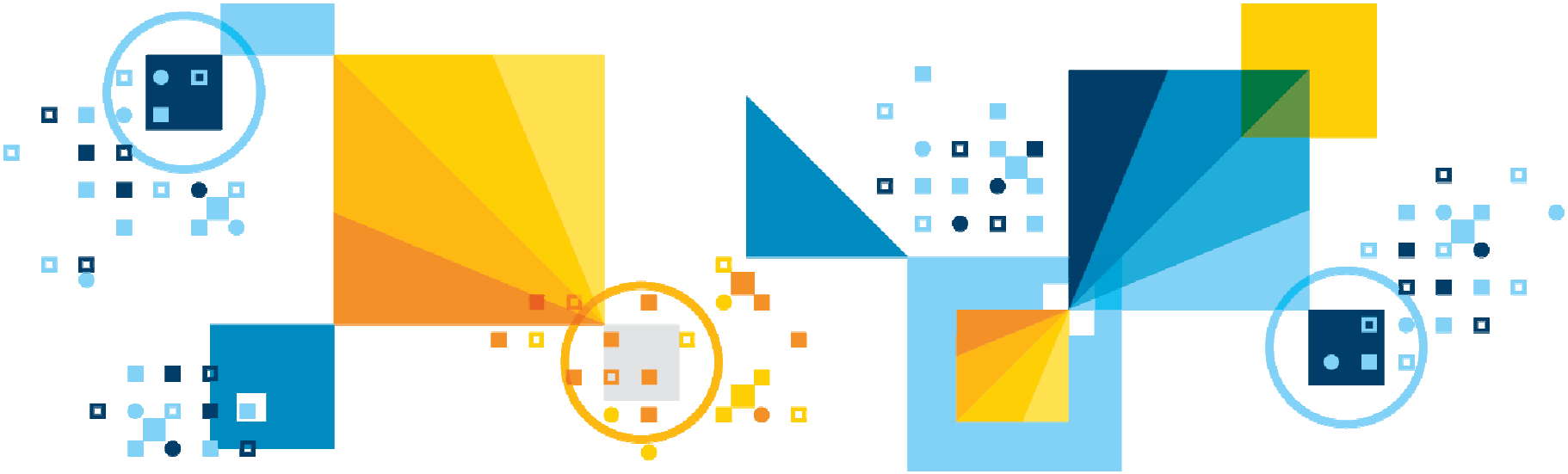
Paul Bird

June 2018

Db2 = JSON + SQL



Safe Harbor Statement

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON CURRENT THINKING REGARDING TRENDS AND DIRECTIONS, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. FUNCTION DESCRIBED HEREIN MY NEVER BE DELIVERED BY I BM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

Agenda

- A brief history
- A look into the future
- Request for your feedback

A BRIEF HISTORY

Db2 and JSON (A timeline)

- **August 2013**
  - Db2 JSON NoSQL support introduced in Db2 10.5 FP1
- **June 2016**
  - Proprietary Db2 JSON SQL functions revealed with Db2 11.1 GA
- **June 2017**
  - Proprietary Db2 JSON SQL functions integrated with Db2 11.1.2.2

Db2 JSON NoSQL support introduced (Db2 10.5 FP1)

- **Focused on enabling Db2 to participate in the NoSQL paradigm**

- **Db2 NoSQL JSON support consists of:**
  - IBM NoSQL Wire Listener
  - JSON Java APIs
  - JSON command line

- **More insight available in developerWorks:**
  - DB2 JSON capabilities, Part 1: Introduction to DB2 JSON
    - https://www.ibm.com/developerworks/data/library/techarticle/dm-1306nosqlforjson1/index.html
  - DB2 JSON capabilities, Part 2: Using the command-line processor
    - https://www.ibm.com/developerworks/data/library/techarticle/dm-1306nosqlforjson2/index.html?ca=drs-
  - DB2 JSON capabilities, Part 3: Writing applications with the Java API
    - https://www.ibm.com/developerworks/data/library/techarticle/dm-1307nosqlforjson3/index.html?ca=drs-
  - DB2 JSON capabilities, Part 4: Using the IBM NoSQL Wire Listener for DB2
    - https://www.ibm.com/developerworks/data/library/techarticle/dm-1306nosqlforjson4/index.html?ca=drs-

Over time, customer requirements began to change…

- **Our customers began to ask for native SQL support of JSON in Db2**
  - These customers and their applications are comfortable with SQL

- **Primary usage scenarios:**
  1. Legacy applications accessing JSON data stored by new applications
     - Using Db2 as the "mission critical" repository for JSON documents in environments with high performance and concurrency requirements
  2. New JSON applications could access legacy data stored within Db2 production systems

- **SQL support for JSON has become a fundamental requirement for Db2**

Proprietary Db2 JSON SQL functions revealed (Db2 11.1 GA)

- **To provide immediate relief, we decided to disclose to our customers the existing internal, proprietary SQL JSON functions used by our DB2 10.5 NoSQL interfaces**
  - These functions are now officially supported by Db2

- **We began to communicate this information through a number of channels**
  - Primary evangelist is George Baklarz (webcasts, conferences, blogs, etc.)
  - DB2night webcast:
    - Episode #187, 9 December 2016, FREE Resources to help you succeed with DB2 LUW V11!
  - Chapter 12 “Experimental JSON Functions“ in DB2 11.1 eBook by George Baklarz and Enzo Cialini
    - https://ibm.ent.box.com/v/DB2v11eBook

Existing, proprietary Db2 JSON SQL functions

Schema	Name	Comments
SYSTOOLS	BSON2JSON	Convert BSON formatted document into JSON strings
SYSTOOLS	BSON_VALIDATE	Checks to make sure that a BSON field in a BLOB object is in a correct format
SYSTOOLS	JSON2BSON	Convert JSON strings into a BSON document format
SYSTOOLS	JSON_GET_POS_ARR_INDEX	Find a value within an array
SYSTOOLS	JSON_LEN	Returns the count of elements in an array type inside a document
SYSTOOLS	JSON_TABLE	Returns a table of values for an array field
SYSTOOLS	JSON_TYPE	Returns the data type of a specific field within a JSON document
SYSTOOLS	JSON_UPDATE	Update a field or document using set syntax
SYSIBM	JSON_VAL	Extracts data from a JSON document into SQL data types

Some characteristics of these JSON SQL functions

- **Located in the SYSTOOLS schema***
  - Must explicitly qualify all references to these functions or add SYSTOOLS schema to function path
  - Must grant EXECUTE to any user wanting to call these functions

- **Require:**
  1. JSON must be stored in a BLOB column as BSON
  2. JSON must be converted to BSON using the SYSTOOLS.JSON2BSON function
     - This (internal) BSON is not compatible with external BSON products

* The one exception is SYSIBM.JSON_VAL

Proprietary Db2 JSON SQL functions integrated (Db2 11.1.2.2)

- **To simplify the customer experience, we formally added these JSON SQL functions to the product**
  - Automatically created in SYSTOOLS schema for a new database or added to an existing one when updated to Db2 11.1.2.2
  - Documented functions in the Db2 knowledge center under a section called "SQL access to JSON documents"
    - https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.swg.im.dbclient.json.doc/doc/c0070285.html

Querying JSON documents

- **Use SYSIBM.JSONVAL or SYSTOOLS.JSON_TABLE**

- **Given this JSON document:**

```
{"empno":"200170",
 "firstname":"KIYOSHI",
 "lastname":"YAMAMOTO",
 "workdept":"D11",
 "pay":{"salary":64680.00,
        "bonus":500.00,
        "comm":1974.00}}
```

- **To retrieve the three "pay“ fields for a specific employee:**

```
SELECT JSON_VAL(EMP_DATA,'pay.salary','i'),
       JSON_VAL(EMP_DATA,'pay.bonus','i'),
       JSON_VAL(EMP_DATA,'pay.comm','i')
FROM JSON_EMP
WHERE
  JSON_VAL(EMP_DATA,'empno','s:6') = '200170';
```

Publishing JSON documents

- **Stored BSON**
  - Use SYSTOOLS.BSON2JSON to convert internal BSON back to JSON and return result

- **All other relational data**
  - ***RYO***
    - Need to construct JSON objects yourself using existing SQL capabilities (e.g. concatenation, data conversion, etc.)
    - Must convert from Db2 data types and formats to JSON equivalents

A LOOK INTO THE FUTURE

A public standard for JSON SQL emerges

- **ISO finalized the SQL 2016 standard which includes a set of JSON SQL functions**
  - Information technology -- Database languages -- SQL Technical Reports -- Part 6: SQL support for JavaScript Object Notation (JSON)
    - https://www.iso.org/contents/data/standard/06/73/67367.html?browse=tc

- **The focus of this report is on querying and publishing JSON data using SQL**
  - No proposal made for modifying JSON documents internally using JSON

The Db2 plan

- **We will be delivering the ISO JSON SQL functions in stages**
  - Some function in Db2 11.1.4.4 and some in the next version
- **Early access to work-in-progress will come through the Early Access Program (EAP)**
  - https://db2-beta.mybluemix.net/
  - If you are interested in playing with Db2 JSON, sign up!
- **The original SYSTOOLS functions will be phased out**
  - They will remain active and supported for quite some time to come
  - They will be de-emphasized in the Db2 documentation as of Db2 11.1.4.4
  - They will be deprecated at some point after the ISO JSON SSQL functions are completed (i.e. in the next version of Db2)
    - They will then go back to being hidden (and undocumented)

Proposed set of new (ISO) JSON SQL functions

Schema	Name	Comments
SYSIBM	BSON_TO_JSON	Convert BSON formatted document into JSON strings
SYSIBM	JSON_TO_BSON	Convert JSON strings into a BSON document format
SYSIBM	JSON_EXISTS	Determine whether a JSON object contains the desired JSON value
SYSIBM	JSON_ARRAY	Creates JSON array from input key value pairs
SYSIBM	JSON_ARRAYAGG	Creates aggregate JSON array from input key value pairs
SYSIBM	JSON_OBJECT	Creates JSON object from input key value pairs
SYSIBM	JSON_OBJECTAGG	Creates aggregate JSON object from input key value pairs
SYSIBM	JSON_TABLE	Creates relational output from a JSON object
SYSIBM	JSON_QUERY	Extract a JSON object from a JSON object
SYSIBM	JSON_VALUE	Extract an SQL scalar value from a JSON object
(predicate)	IS JSON	Validates that input value is a valid JSON object

Some highlights

- **Easier to use**
  - No need to qualify or add SYSTOOLS to function path
  - No need to grant EXECUTE privilege

- **Flexible storage options**
  - You choose the stored format: JSON or BSON
    - Internal BSON requirement removed (format is tolerated but not required)
  - You choose the table organization: row or column
  - You choose the column data type
    - BLOB, CHAR, CLOB, VARBINARY, VARCHAR
  - Normal Db2 mechanisms are used to load JSON (or BSON) data into tables (e.g. INSERT, Load, etc..)
    - Complimentary (but optional) conversion functions are provided

Querying JSON documents (scalar)

- **New JSON scalar functions can be used to extract JSON objects or SQL values from JSON data**
  - JSON_QUERY returns a JSON object or array value
  - JSON_VALUE returns an SQL scalar value

- **Examples:**
  - Return the JSON object associated with the name key from JSON data:

    VALUES (JSON_QUERY('{"id":"701", "name":{"first":"John", "last":"Doe"}}', '$.name');

    The result is the following string that represents a JSON object:

    {"first":"John", "last":"Doe"}

  - Return a value from JSON data as an integer.

    VALUES (JSON_VALUE('{"id":"987"}', '$.id' RETURNING INTEGER));

    The result is 987.

Querying JSON documents (table)

- **New JSON_TABLE function outputs the contents of a JSON document as a relational result set**
  - Similar to what XML_TABLE does for XML

- **Example:**
  - List the employee id, first name, last name, and first phone type and number from JSON data stored in a column of EMPLOYEE_TABLE:

```
SELECT U."id", U."first name",U."last name",U."phone type",U."phone number“
FROM EMPLOYEE_TABLE E,
        JSON_TABLE(E.jsondoc,
                        ’lax $’
                        COLUMNS( "id" INTEGER,
                                        "first name" VARCHAR(20) PATH ’lax $.name.first’,
                                        "last name" VARCHAR(20) PATH ’lax $.name.last’,
                                        "phone type" VARCHAR(20) PATH ’lax $.phones[0].type’,
                                        "phone number" VARCHAR(20) PATH ’lax $.phones[0].number’)) AS U
```

  - Returns:

```
id first name  last name phone type phone number
--- ---------- --------- ---------- ------------
901 John       Doe       home       555-3762
887 Mary       Brown     home       555-2732
891 Qi         Cheng     work       555-8377
```

Publishing JSON documents

- **Stored JSON and BSON**
  - Retrieve from table using SQL and return as-is
  - Use complimentary BSON_TO_JSON or JSON_TO_BSON functions convert stored data to preferred output format

- **All other relational data**
  - New functions can be used to construct JSON from relational data:
    - JSON_ARRAY creates a JSON array from input key value pairs
    - JSON_ARRAYAGG creates aggregate JSON array from input key value pairs
    - JSON_OBJECT creates JSON object from input key value pairs
    - JSON_OBJECTAGG creates aggregate JSON object from input key value pairs

Publishing example

- **Generate a JSON object from a relational table containing the last name, date hired, and salary for the employee with an employee number of '000020'.**

```
SELECT JSON_OBJECT('Last name' : LASTNAME, 'Hire date' : HIREDATE, 'Salary' : SALARY)
FROM EMPLOYEE
WHERE EMPNO = '000020'
```

- **The result of this statement is the following JSON string:**

```
{"Last name":"THOMPSON","Hire date":"1973-10-10","Salary":41250.00}
```

Current outlook for Db2 11.1.4.4

Schema	Name	Comments
SYSIBM	BSON_TO_JSON	Convert BSON formatted document into JSON strings
SYSIBM	JSON_TO_BSON	Convert JSON strings into a BSON document format
SYSIBM	JSON_EXISTS	Determine whether a JSON object contains the desired JSON value
SYSIBM	JSON_ARRAY	Creates JSON array from input key value pairs
SYSIBM	JSON_ARRAYAGG	Creates aggregate JSON array from input key value pairs
SYSIBM	JSON_OBJECT	Creates JSON object from input key value pairs
SYSIBM	JSON_OBJECTAGG	Creates aggregate JSON object from input key value pairs
SYSIBM	JSON_TABLE	Creates relational output from a JSON object
SYSIBM	JSON_QUERY	Extract a JSON object from a JSON object
SYSIBM	JSON_VALUE	Extract an SQL scalar value from a JSON object
(predicate)	IS JSON	Validates that input value is a valid JSON object

REQUEST FOR YOUR FEEDBACK

Indexing JSON documents

- **Query within JSON is being done using open source libraries and not using Db2 operators**
  - Indexes are to help Db2 find the right JSON “payload” to query using the new JSON SQL functions
    - Indexes will not help with the actual JSON access itself

- **Currently, the proposal for indexing will be to do one of the following:**
  A. Pull out values from JSON into relational columns and apply normal indexing approaches
  B. Leverage index on expression to index on JSON document content directly
    - Only works for single value results (e.g. index on phone number value does not work if object has more than one phone number in it)

- **This approach also means indexes will not be updated when the document content is modified**

Editing JSON documents

- **One usage scenario not covered by the ISO JSON SQL functions is one where the JSON data is modified using SQL**
  - E.g. adding a new key:value pair to a JSON document stored in Db2

- **We are trying to understand the value and priority of this capability to our customers**
  - Does anyone use SYSTOOLS.JSON_UPDATE today?
  - Is this a scenario that you see yourself using/needing in the future?

Any feedback on these requests we have heard?

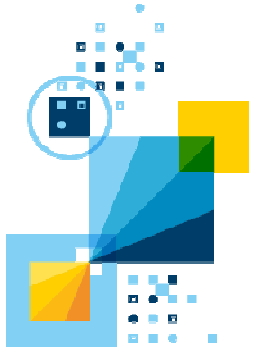
- Provide a way to define constraints on JSON documents (to help ensure document consistency)
- Provide a JSON data type perhaps as a wrapper around XML data type (to gain schema enforcement)
- (RFE) Provide a way to easily convert output from an SQL query directly to JSON (e.g. sort of a reverse JSON_TABLE)
- Provide a tool to migrate from XML to JSON
- Provide routines to help debug JSON construction errors

What will happen in Db2 next?

- **Our immediate plans in the next version of Db2 would be to finish the implementation of the remaining ISO JSON SQL functions**

- **After that, we would like to ensure that we have addressed any additional urgent JSON requirements**
  - We need your help to properly identify and prioritize those

- **Are there other requirements or needs out there?**
  - We would also like to hear about other scenarios or functions that need to be covered
  - Don’t be shy!



Questions