MySQL JSON

New Data Type JSON in MySQL >= 5.7.8

Sergej Kurakin

Sergej Kurakin

Age: 36

Company: NFQ Technologies

Position: Software Engineer

MySQL 5.7 is in “General Availability”

MySQL released version 5.7.9 and marked it with “General Availability” on October 21, 2015.

Today is August 3, 2017...

So why am I making this talk about MySQL JSON support 2 years after release?

Reasons

Yes, I’ve missed the release.

My project upgraded to MySQL 5.7 just few months ago.

Not all PHP libraries and frameworks support JSON as data type.

Not all “stable” Linux distributions are already upgraded to MySQL 5.7.

I think there are more people who missed the release of MySQL 5.7.

I think there are more people who missed new MySQL type JSON.

What does official documentation say?

MySQL from version 5.7.8 supports new Data Type: JSON

That new feature gives us some advantages!

Automatic validation of JSON documents stored in JSON columns.

Invalid documents produce an error.

Now we might meet a project that validates JSON using calls to MySQL.

I hope we not.

Optimized storage format.

JSON documents stored in JSON columns are converted to an internal format that permits quick read access to document elements.

The space required to store a JSON document is roughly the same as for LONGBLOB or LONGTEXT

The size of any JSON document stored in a JSON column is limited to the value of the *max_allowed_packet* system variable.

A JSON column cannot have a default value, so default values is always *NULL*.

JSON columns are not indexed directly.

MySQL handles strings used in JSON context using the *utf8mb4* character set and *utf8mb4_bin* collation.

Table Example

```
CREATE TABLE jsonexample (
    id MEDIUMINT NOT NULL AUTO_INCREMENT,
    c JSON,
    PRIMARY KEY (id)
);
```

JSON Functions

JSON Creation Functions

JSON_ARRAY([val[, val] ...])

JSON_OBJECT([key, val[, key, val] ...])

JSON_QUOTE(string)

CAST(value AS JSON)

JSON Search Functions

JSON_CONTAINS(target, candidate[, path])

JSON_CONTAINS_PATH(json_doc, one_or_all, path[, path] ...)

JSON_EXTRACT(json_doc, path[, path] ...)

column->path

column->>path

JSON_KEYS(json_doc[, path])

JSON_SEARCH(json_doc, one_or_all, search_str[, escape_char[, path] ...])

JSON Modification Functions

JSON_APPEND(json_doc, path, val[, path, val] ...)

JSON_ARRAY_APPEND(json_doc, path, val[, path, val] ...)

JSON_ARRAY_INSERT(json_doc, path, val[, path, val] ...)

JSON_INSERT(json_doc, path, val[, path, val] ...)

JSON_MERGE(json_doc, json_doc[, json_doc] ...)

JSON Modification Functions

JSON_REMOVE(json_doc, path[, path] ...)

JSON_REPLACE(json_doc, path, val[, path, val] ...)

JSON_SET(json_doc, path, val[, path, val] ...)

JSON_UNQUOTE(json_val)

JSON Attribute Functions

JSON_DEPTH(json_doc)

JSON_LENGTH(json_doc[, path])

JSON_TYPE(json_val)

JSON_VALID(val)

JSON Path Syntax

$.a

$.a.d

$[*]

$[*][0].k

$**.k

Spatial GeoJSON Functions

ST_AsGeoJSON(g [, max_dec_digits [, options]])

ST_GeomFromGeoJSON(str [, options [, srid]])

Indexing JSON Column

Indexing JSON Column

InnoDB supports secondary indexes on virtual generated columns.

When a secondary index is created on a virtual generated column, generated column values are materialized in the records of the index.

Indexing JSON Column

```
CREATE TABLE jsonindex (
    c JSON,
    g INT GENERATED ALWAYS AS (c->"$.id"),
    INDEX i (g)
);
```

Usage Examples

Key-Value Store

Metadata Column

More about JSON related stuff

Other Databases and Storages

MongoDB uses JSON-like documents with schemas from February 2009.

PostgreSQL introduced JSON type in version 9.2 on September 2012.

IBM DB2 introduced JSON type on June 2013 (10.5).

Oracle Database supports JSON type from July 2014 (12.1.0.2).

Microsoft SQL Server 2016 supports JSON.

MariaDB added JSON functions on May 2017.

** MySQL introduced JSON type on October 2015. Slide updated at July 29, 2017*

Library Support

Doctrine DBAL supports native JSON mapping from July 23, 2017 (2.6.0).

No stable Doctrine ORM has been released yet.

Eloquent ORM (Laravel) from September 2016 (5.3, if I got it right).

No information about ADOdb was found.

** Slide updated at July 29, 2017.*

Linux OS MySQL versions

Debian Stretch: MySQL 5.5.x.

Ubuntu 16.04, 17.04: MySQL 5.7.x.

OpenSUSE: no information was found.

Fedora, RedHat, CentOS: no information was found.

** Based on https://distrowatch.com/ information 2017-07-29*

Discussion

Sergej Kurakin

Work @mail: sergej.kurakin@nfq.lt

Personal @mail: sergej@kurakin.info

https://www.linkedin.com/in/sergejkurakin