INTERNATIONAL STANDARD

ISO/IEC 19075-6

First edition
2021-08

Information technology — Guidance for the use of database language SQL —

Part 6: Support for JSON

Technologies de l'information — Recommandations pour l'utilisation du langage de base de données SQL —

Partie 6: Support pour JSON

Reference number
ISO/IEC 19075-6:2021(E)

© ISO/IEC 2021



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2021

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2021 – All rights reserved

Contents

Page

Foreword . . . ix
Introduction . . . xi
1 Scope . . . 1
2 Normative references . . . 2
3 Terms and definitions . . . 3
4 JavaScript Object Notation (JSON) . . . 6
4.1 Context for JSON . . . 6
4.2 What is JSON? . . . 6
4.3 Representations of JSON data . . . 7
4.3.1 Introduction to representations of JSON data . . . 7
4.3.2 Avro . . . 7
4.3.3 BSON . . . 8
4.4 Schemas . . . 8
4.4.1 JSON schemata and validity . . . 8
4.4.2 Avro schemata . . . 9
4.4.3 BSON schemata . . . 9
4.5 Why does JSON matter in the context of SQL? What is JSON's relationship to NoSQL? . . . 10
4.6 Use cases for JSON support in SQL . . . 11
4.6.1 Introduction to the use cases . . . 11
4.6.2 JSON data ingestion and storage . . . 11
4.6.3 JSON data generation from relational data . . . 11
4.6.4 Querying JSON as persistent semi-structured data model instances . . . 12
4.7 What features address those use cases? . . . 12
4.7.1 Addressing the use cases . . . 12
4.7.2 Storing JSON data in an SQL table . . . 12
4.7.3 Generating JSON in an SQL query . . . 12
4.7.4 Querying JSON data in SQL tables using SQL . . . 13
5 The SQL/JSON data model . . . 14
5.1 Introduction to the SQL/JSON data model . . . 14
5.2 SQL/JSON items . . . 15
5.2.1 Definition of SQL/JSON items . . . 15
5.2.2 Atomic values . . . 16
5.2.3 SQL/JSON arrays . . . 17
5.2.4 SQL/JSON objects . . . 17
5.3 SQL/JSON sequences . . . 17
5.4 Parsing JSON . . . 18
5.5 Serializing JSON . . . 18
6 SQL/JSON functions . . . 19

© ISO/IEC 2021 – All rights reserved

6.1 Introduction to SQL/JSON functions. . . . 19
6.2 Handle JSON using built-in functions. . . . 19
6.3 JSON API common syntax. . . . 19
6.3.1 Introduction to JSON API common syntax. . . . 19
6.3.2 JSON value expression. . . . 20
6.3.3 Path expression. . . . 20
6.3.4 PASSING clause. . . . 20
6.3.5 JSON output clause. . . . 21
6.3.6 ON ERROR and ON EMPTY syntax. . . . 21
6.4 Query functions. . . . 22
6.4.1 The four query functions. . . . 22
6.4.2 JSON_EXISTS. . . . 22
6.4.3 JSON_VALUE. . . . 25
6.4.4 JSON_QUERY. . . . 29
6.4.5 JSON_TABLE. . . . 33
6.4.5.1 Introduction to JSON_TABLE. . . . 33
6.4.5.2 COLUMNS clause that is not nested. . . . 34
6.4.5.3 Nested COLUMNS clause. . . . 36
6.4.5.4 PLAN clause. . . . 37
6.4.6 Conformance features for query operators. . . . 41
6.5 Constructor functions and IS JSON predicate. . . . 43
6.5.1 Tables used to illustrate constructor functions and the IS JSON predicate. . . . 43
6.5.2 JSON_OBJECT. . . . 44
6.5.3 JSON_OBJECTAGG. . . . 45
6.5.4 JSON_ARRAY. . . . 46
6.5.5 JSON_ARRAYAGG. . . . 47
6.5.6 IS JSON predicate. . . . 49
6.5.7 Handling of JSON nulls and SQL nulls. . . . 49
6.5.8 Conformance features for constructor functions. . . . 50
7 SQL/JSON path language. . . . 51
7.1 Overview of SQL/JSON path language. . . . 51
7.2 Objectives for the SQL/JSON path language. . . . 52
7.3 Modes. . . . 53
7.3.1 Introduction to modes. . . . 53
7.3.2 Example of strict *vs* lax. . . . 54
7.4 Lexical issues. . . . 56
7.5 Syntax summary. . . . 58
7.6 Formal semantics and notational conventions. . . . 58
7.7 Primitive operations in formal semantics. . . . 59
7.7.1 Concatenation. . . . 59
7.7.2 unwrap(). . . . 59
7.7.3 wrap(). . . . 60
7.8 Mode declaration. . . . 60
7.9 <JSON path primary>. . . . 61
7.9.1 Introduction to JSON path primaries. . . . 61
7.9.2 Literals. . . . 61
7.9.3 Variables. . . . 62

 © ISO/IEC 2021 – All rights reserved

7.9.4 Parentheses. . . . 63
7.10 Accessors. . . . 63
7.10.1 Introduction to accessors. . . . 63
7.10.2 Member accessor. . . . 64
7.10.3 Member wildcard accessor. . . . 67
7.10.4 Element accessor. . . . 68
7.10.5 Element wildcard accessor. . . . 70
7.10.6 Sequence semantics of the accessors. . . . 71
7.11 Item methods. . . . 71
7.11.1 Introduction to item methods. . . . 71
7.11.2 type(). . . . 72
7.11.3 size(). . . . 72
7.11.4 Numeric item methods (double, ceiling, floor, abs). . . . 73
7.11.5 datetime(). . . . 73
7.11.6 keyvalue(). . . . 73
7.12 Arithmetic expressions. . . . 75
7.12.1 Introduction to arithmetic expressions. . . . 75
7.12.2 Unary plus and minus. . . . 75
7.12.3 Binary operations. . . . 76
7.13 Filter expression. . . . 76
7.13.1 Introduction to filter expressions. . . . 76
7.13.2 `true`/`false` and *True*/*False* 77
7.13.3 `null` and *Unknown*. . . . 78
7.13.4 Error handling in filters. . . . 78
7.13.5 Truth tables. . . . 81
7.13.6 Comparison predicates. . . . 82
7.13.7 `like_regex` predicate. . . . 83
7.13.8 `starts with` predicate. . . . 84
7.13.9 `exists` predicate. . . . 84
7.13.10 `is unknown` predicate. . . . 86
7.14 Conformance features for SQL/JSON path language. . . . 87
Bibliography. . . . 88
Index. . . . 89

© ISO/IEC 2021 – All rights reserved

Tables

Table

Page

1 JSON, SQL/JSON, and SQL (other than SQL/JSON values and counterparts). . . . 14
2 Parallels between JSON text and SQL/JSON data model. . . . 18
3 JSON_EXISTS sample data. . . . 22
4 Result of the sample query. . . . 23
5 Accessor example. . . . 24
6 Result 1. . . . 26
7 Result 2. . . . 26
8 Result 3. . . . 27
9 Result 4. . . . 27
10 Result 5. . . . 28
11 Result 6. . . . 29
12 JSON_EXISTS results. . . . 30
13 ON EMPTY results. . . . 31
14 ARRAY WRAPPER results. . . . 31
15 Illustrating differences. . . . 32
16 Comparison of wrapper options. . . . 32
17 JSON_TABLE sample data in a book recommendation table. . . . 33
18 Unnested query result. . . . 35
19 Nested query result. . . . 36
20 PLAN query result. . . . 39
21 Second PLAN query result. . . . 39
22 Third PLAN query result. . . . 40
23 DEPTS table. . . . 43
24 JOBS table. . . . 43
25 EMPLOYEES table. . . . 44
26 The JSON object returned. . . . 45
27 Returned JSON object with the corresponding job sequence number. . . . 46
28 ARRAYAGG query result. . . . 48
29 Second ARRAYAGG query result. . . . 48
30 Third ARRAYAGG query result. . . . 49
31 Three aspects of path evaluation governed by modes. . . . 54
32 Example of strict *vs* lax. . . . 55
33 Features of the SQL/JSON path language. . . . 58
34 Data used by unwrap() example. . . . 59
35 Data used by wrap() example. . . . 60
36 Examples of atomic values in the SQL/JSON path language 61
37 Examples of the escaping rules. . . . 61
38 Evaluation of `'$.phones.type'` in lax mode. . . . 65
39 Intermediate step. . . . 65
40 Evaluation of`'$.phones[*].type'` 66
41 Evaluation of `'$.phones[*] ? (exists (@.type)).type'` 66
42 Evaluation of `'$.phones.*'` in lax mode. . . . 67
43 Evaluation of `'$.phones[*].*'` 67
44 Evaluation of `'lax $.sensors.*[0, last, 2]'` 69
45 The step in the evaluation 71
46 Result of the query with the sample data 74
47 Evaluation of `'lax -$.readings.floor()'` 75

© ISO/IEC 2021 – All rights reserved

48 Evaluation of 'lax (-$.readings).floor()' . . . 76
49 Table T with two rows . . . 78
50 Computation on row K=102 . . . 79
51 Modified table T . . . 79
52 Computation on row K=102 in modified table T. . . . 79
53 Computation of 'lax $? (@.hours > 9)' on row K=102 . . . 80
54 Computation of 'strict $? (@.hours > 9)' in strict mode . . . 80
55 Result of &&. . . . 81
56 Result of || . . . 81
57 Result of ! . . . 81
58 Supported comparisons . . . 82
59 Final result . . . 83
60 A table with JSON column . . . 85
61 Evaluation of 'strict $? (exists (@.name)).name' on row K=201 . . . 85
62 Evaluation of 'strict $? (exists (@.name)).name' on row K=202 . . . 85

© ISO/IEC 2021 – All rights reserved

Figures

Figure

Page

1 Relationships between “JSON” and “SQL/JSON”. 14
2 The SQL/JSON path language architecture. 51

© ISO/IEC 2021 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents), or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

This first edition of ISO/IEC 19075-6 cancels and replaces ISO/IEC TR 19075-6:2017. Several small technical errors have been corrected,

This document is intended to be used in conjunction with the following editions of the parts of the ISO/IEC 9075 series:

— ISO/IEC 9075-1, sixth edition or later;

— ISO/IEC 9075-2, sixth edition or later;

— ISO/IEC 9075-3, sixth edition or later;

— ISO/IEC 9075-4, seventh edition or later;

— ISO/IEC 9075-9, fifth edition or later;

— ISO/IEC 9075-10, fifth edition or later;

— ISO/IEC 9075-11, fifth edition or later;

— ISO/IEC 9075-13, fifth edition or later;

— ISO/IEC 9075-14, sixth edition or later;

— ISO/IEC 9075-15, second edition or later;

— ISO/IEC 9075-16, first edition or later.

© ISO/IEC 2021 – All rights reserved

A list of all parts in the ISO/IEC 19075 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/-national-committees.

© ISO/IEC 2021 – All rights reserved

Introduction

The organization of this document is as follows:

1) Clause 1, “Scope”, specifies the scope of this document.

2) Clause 2, “Normative references”, identifies additional standards that, through reference in this document, constitute provisions of this document.

3) Clause 3, “Terms and definitions”, defines the terms and definitions used in this document.

4) Clause 4, “JavaScript Object Notation (JSON)”, introduces what is JSON.

5) Clause 5, “The SQL/JSON data model”, introduces the data model that is used by the SQL/JSON functions and the SQL/JSON path language.

6) Clause 6, “SQL/JSON functions”, introduces the SQL/JSON functions to query and construct JSON.

7) Clause 7, “SQL/JSON path language”, introduces the SQL/JSON path language.

© ISO/IEC 2021 – All rights reserved

Information technology — Guidance for the use of database language SQL —

Part 6:
Support for JSON

1 Scope

This document describes the support in SQL for JavaScript Object Notation.

This document discusses the following features of the SQL language:

— Storing JSON data.

— Publishing JSON data.

— Querying JSON data.

— SQL/JSON data model and path language.

© ISO/IEC 2021 – All rights reserved

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9075-1, *Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework)*

ISO/IEC 9075-2, *Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation)*

© ISO/IEC 2021 – All rights reserved

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 9075-1 and the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1
JSON text
sequence of tokens

Note 1 to entry: JSON text must be encoded in Unicode (UTF-8 by default); insignificant whitespace may be used anywhere in JSON text except within strings (where all whitespace is significant), numbers, and literals. JSON text is a single object or array

3.2
token
strings, numbers, literals, or one of six structural characters

Note 1 to entry: The six structural characters are "{", "}", "[", "]", ":", ","

3.3
value
object, array, number, string, or one of three literals

3.4
type
primitive type or structured type

3.5
primitive type
string, number, Boolean, or null

3.6
primitive value
value that is a string, number, Boolean, or null

3.7
structured type
object or array

3.8
structured value
value that is an object or an array

© ISO/IEC 2021 – All rights reserved

3.9
string
sequence of Unicode characters

Note 1 to entry: Some characters must be "escaped" by preceding them with a reverse solidus ("\"), while any or all characters can be represented in "Unicode notation" comprising the string '\u' followed by four hexadecimal digits or two such strings representing the UTF-16 surrogate pairs representing characters not on the Basic Multilingual Plane.

Note 2 to entry: Strings are surrounded by double-quote characters, which are not part of the value of the strings.

3.10
number
sequence comprising an integer part, optionally followed by a fractional part and/or an exponent part

Note 1 to entry: Non-numeric values, such as infinity and NaN are not permitted.

3.11
Boolean
literal "`true`" or literal "`false`"

3.12
null
literal "`null`"

3.13
object
structure represented by "`{`", zero or more members separated by "`,`", and "`}`"

3.14
member
string followed by a colon followed by a value in an object

Note 1 to entry: A member is also known as a "name-value pair"; the name is sometimes called a "key" and the second value is sometimes called a "bound value".

3.15
array
structure represented by a "`[`", zero or more elements separated by "`,`", and "`]`"

3.16
element
value in an array

3.17
field
member in an object, element in an array, name in a member, or value in a member

3.18
data model
<general> definition of what kinds of data belong to a particular universe of discourse, including the operations on those kinds of data

3.19
JSON data model
(implicit) data model associated with JSON

 © ISO/IEC 2021 – All rights reserved

3.20
SQL/JSON data model
data model created for operating on JSON data within the SQL language

© ISO/IEC 2021 – All rights reserved