

TECHNICAL REPORT

ISO/IEC TR
19075-6

First edition
2017-03

Information technology — Database languages — SQL Technical Reports —

Part 6: SQL support for JavaScript Object Notation (JSON)

iTeh STANDARD REVIEW
Technologies de l'information — Langages de base de données — SQL rapport techniques —
(standards.iteh.ai)
Partie 6. Support de SQL pour JavaScript Object Notation (JSON)

[ISO/IEC TR 19075-6:2017](#)
<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>



Reference number
ISO/IEC TR 19075-6:2017(E)

© ISO/IEC 2017

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC TR 19075-6:2017](#)

<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, 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
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword.....	ix
Introduction.....	x
1 Scope.....	1
2 Normative references.....	3
2.1 ISO and IEC standards.....	3
2.2 Other international standards.....	3
3 JavaScript Object Notation (JSON).....	5
3.1 What is JSON?.....	5
3.2 Representations of JSON data.....	6
3.2.1 Avro.....	6
3.2.2 BSON.....	7
3.3 Schemas.....	7
3.3.1 JSON schemata and validity.....	7
3.3.2 Avro schemata.....	ISO/IEC TR 19075-6:2017
3.3.3 BSON schemata.....	https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017
3.4 Why does JSON matter in the context of SQL? What is JSON's relationship to NoSQL?.....	9
3.5 JSON terminology.....	10
3.6 Use cases for JSON support in SQL.....	11
3.6.1 JSON data ingestion and storage.....	11
3.6.2 JSON data generation from relational data.....	12
3.6.3 Querying JSON as persistent semi-structured data model instances.....	12
3.7 What features address those use cases?.....	12
3.7.1 Storing JSON data in an SQL table.....	12
3.7.2 Generating JSON in an SQL query.....	13
3.7.3 Querying JSON data in SQL tables using SQL.....	13
4 The SQL/JSON data model.....	15
4.1 SQL/JSON items.....	16
4.1.1 Atomic values.....	17
4.1.2 SQL/JSON arrays.....	18
4.1.3 SQL/JSON objects.....	18
4.2 SQL/JSON sequences.....	19
4.3 Parsing JSON.....	19
4.4 Serializing JSON.....	19
5 SQL/JSON functions.....	21
5.1 Handle JSON using built-in functions.....	21

5.2	JSON API common syntax.....	21
5.2.1	JSON value expression.....	22
5.2.2	Path expression.....	22
5.2.3	PASSING clause.....	22
5.2.4	JSON output clause.....	23
5.2.5	ON ERROR and ON EMPTY syntax.....	23
5.3	Query functions.....	24
5.3.1	JSON_EXISTS.....	24
5.3.2	JSON_VALUE.....	27
5.3.3	JSON_QUERY.....	32
5.3.4	JSON_TABLE.....	35
5.3.4.1	COLUMNS clause that is not nested.....	36
5.3.4.2	Nested COLUMNS clause.....	39
5.3.4.3	PLAN clause.....	40
5.3.5	Conformance features for query operators.....	44
5.4	Constructor functions and IS JSON predicate.....	46
5.4.1	JSON_OBJECT.....	47
5.4.2	JSON_OBJECTAGG.....	48
5.4.3	JSON_ARRAY.....	50
5.4.4	JSON_ARRAYAGG.....	50
5.4.5	IS JSON predicate.....	52
5.4.6	Handling of JSON nulls and SQL nulls.....	53
5.4.7	Conformance features for constructor functions.....	53
6	SQL/JSON path language.....	55
6.1	Overview of SQL/JSON path language.....	55
6.2	Objectives for the SQL/JSON path language.....	57
6.3	Modes.....	58
6.3.1	Example of strict vs lax.....	59
6.4	Lexical issues.....	61
6.5	Syntax summary.....	62
6.6	Formal semantics.....	63
6.6.1	Notational conventions.....	63
6.7	Primitive operations.....	64
6.7.1	Concatenation.....	64
6.7.2	unwrap().....	64
6.7.3	wrap().....	65
6.8	Mode declaration.....	65
6.9	<JSON path primary>.....	66
6.9.1	Literals.....	66
6.9.2	Variables.....	67
6.9.3	Parentheses.....	68
6.10	Accessors.....	68
6.10.1	Member accessor.....	69

6.10.2	Member wildcard accessor	72
6.10.3	Element accessor	73
6.10.4	Element wildcard accessor	75
6.10.5	Sequence semantics of the accessors	76
6.11	Item methods	77
6.11.1	type()	77
6.11.2	size()	78
6.11.3	Numeric item methods (double, ceiling, floor, abs)	78
6.11.4	datetime()	78
6.11.5	keyvalue()	79
6.12	Arithmetic expressions	80
6.12.1	Unary plus and minus	81
6.12.2	Binary operations	82
6.13	Filter expression	82
6.13.1	true/false and <u>True/False</u>	83
6.13.2	null and <u>Unknown</u>	83
6.13.3	Error handling in filters	83
6.13.4	Truth tables	86
6.13.5	Comparison predicates	87
6.13.6	like_regex predicate	89
6.13.7	starts with predicate	90
6.13.8	exists predicate	90
6.13.9	is unknown predicate	92
6.14	Conformance features for SQL/JSON path language	93
Bibliography	95
Index	97

Tables

Table		Page
1 Terms and definitions		10
2 JSON, SQL/JSON, and SQL (other than SQL/JSON values and counterparts)		15
3 Parallels between JSON text and SQL/JSON data model		19
4 JSON_EXISTS sample data		24
5 Result of the sample query		25
6 Accessor example		26
7 Result 1		28
8 Result 2		29
9 Result 3		29
10 Result 4		30
11 Result 5		31
12 Result 6		31
13 Comparison of wrapper options		35
14 JSON_TABLE sample data in a book recommendation table		36
15 Query result	iTeh STANDARD PREVIEW	38
16 Query result	(standards.iteh.ai)	39
17 Query result	(standards.iteh.ai)	42
18 Query result	(standards.iteh.ai)	42
19 Query result	(standards.iteh.ai)	44
20 DEPTS table	ISO/IEC TR 19075-6:2017	46
21 JOBS table	https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91ef-6746c105b9a8/iso-iec-tr-19075-6-2017	47
22 EMPLOYEES table	(standards.iteh.ai)	47
23 The JSON object returned		48
24 Returned JSON object with the corresponding job sequence number		49
25 Query result		51
26 Query Results		51
27 Query Results		52
28 Three aspects of path evaluation governed by modes		58
29 Example of strict vs lax		59
30 Features of the SQL/JSON path language		62
31 Data used by unwrap() example		64
32 Data used by wrap() example		65
33 Examples of atomic values in the SQL/JSON path language		66
34 Examples of the escaping rules		66
35 Evaluation of '\$.phones.type' in lax mode		70
36 Intermediate step		70
37 Evaluation of '\$.phones[*].type'		71
38 Evaluation of '\$.phones[*] ? (@.type).type'		71
39 Evaluation of '\$.phones.*' in lax mode		72
40 Evaluation of '\$.phones[*].*'		73
41 Evaluation of 'lax \$.sensors.*[0, last, 2]'		75
42 The step in the evaluation		76
43 Result of the query with the sample data		80

44	Evaluation of lax <code>-\$readings.floor()</code>	81
45	Evaluation of 'lax <code>(-\$readings).floor()</code> '.....	81
46	Table T with two rows.....	84
47	Computation on row K=102	84
48	Modified table T	85
49	Computation on row K=102 in modified table T.....	85
50	Computation of 'lax \$? (@.hours > 9)' on row K=102	86
51	Computation of 'strict \$? (@.hours > 9)' in strict mode	86
52	Result of &&.....	87
53	Result of 	87
54	Result of !.....	87
55	Supported comparisons	88
56	Final result	89
57	A table with JSON column	91
58	Evaluation of 'strict \$? (exists (@.name)).name' on row K=201.....	91
59	Evaluation of 'strict \$? (exists (@.name)).name' on row K=202.....	91

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC TR 19075-6:2017](#)

<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>

Figures

Figure	Page
1 Relationships between “JSON” and “SQL/JSON”.....	15
2 The SQL/JSON path language architecture.....	55

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC TR 19075-6:2017](https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017)
<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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).

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).

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

iTeh STANDARD PREVIEW

([standards.iteh.ai](https://standards.iteh.ai/catalog/standards/sist/iso/19075-6))

For an explanation on 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 the following URL: www.iso.org/iso/foreword.html.

<https://standards.iteh.ai/catalog/standards/sist/iso/19075-6>

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

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

NOTE 1 — The individual parts of multi-part technical reports are not necessarily published together. New editions of one or more parts can be published without publication of new editions of other parts.

Introduction

The organization of this part of ISO/IEC 19075 is as follows:

- 1) Clause 1, “Scope”, specifies the scope of this part of ISO/IEC 19075.
- 2) Clause 2, “Normative references”, identifies additional standards that, through reference in this part of ISO/IEC 19075, constitute provisions of this part of ISO/IEC 19075.
- 3) Clause 3, “JavaScript Object Notation (JSON)”, introduces what is JSON.
- 4) Clause 4, “The SQL/JSON data model”, introduces the data model that is used by the SQL/JSON functions and the SQL/JSON path language.
- 5) Clause 5, “SQL/JSON functions”, introduces the SQL/JSON functions to query and construct JSON.
- 6) Clause 6, “SQL/JSON path language”, introduces the SQL/JSON path language.

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC TR 19075-6:2017](https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017)
<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>

Information technology — Database languages — SQL Technical Reports —

Part 6: SQL support for JavaScript Object Notation (JSON)

1 Scope

This Technical Report describes the support in SQL for JavaScript Object Notation.

This Technical Report discusses the following features of the SQL language:

- Storing JSON data.
- Publishing JSON data.
- Querying JSON data. **iTeh STANDARD PREVIEW**
(standards.iteh.ai)
- SQL/JSON data model and path language.

[ISO/IEC TR 19075-6:2017](#)

<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>

(Blank page)

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC TR 19075-6:2017](https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017)
<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>

2 Normative references

The following referenced documents are indispensable for the application 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.

2.1 ISO and IEC standards

[ISO9075-2] ISO/IEC 9075-2:2016, *Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation)*

2.2 Other international standards

iTeh STANDARD PREVIEW

[ECMAScript] ISO/IEC 16262:2011, Information technology — Programming languages, their environments and system software interfaces — ~~ECMAScript language specification~~; also available as *ECMAScript Language Specification*,

<http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf>

<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>

[Unicode] The Unicode Standard,

<http://unicode.org>

[RFC7159] Internet Engineering Task Force, RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014,

<https://tools.ietf.org/rfc/rfc7159.txt>

(Blank page)

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC TR 19075-6:2017](https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017)
<https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017>

3 JavaScript Object Notation (JSON)

3.1 What is JSON?

JSON (an acronym for “JavaScript Object Notation”) is both a notation (that is, a syntax) for representing data and an implied data model. JSON is not an object-oriented data model; that is, it does not define sets of classes and methods, type inheritance, or data abstraction. Instead, JSON “objects” are simple data structures, including arrays. [RFC7159] says that JSON is a text format for the serialization of structured data. Its initial intended use was as a data transfer syntax.

The complete syntax of JSON is specified in [RFC7159].

The first-class components of the JSON data model are JSON values. A JSON value is one of the following: JSON object, JSON array, JSON string, JSON number, or one of the JSON literals: true, false, and null. A JSON object is zero or more name-value pairs and is enclosed in curly braces — { }. A JSON array is an ordered sequence of zero or more values and is enclosed in square brackets — [].

Here is an example of a JSON object:

```
{ "Name" : "Isaac Newton",
  "Weight" : 80,
  "Famous" : true, https://standards.iteh.ai/catalog/standards/sist/eb83d5ab-4d72-4e77-91e1-c746c105b9a8/iso-iec-tr-19075-6-2017
  "Phone" : null }
```

The name-value pairs are separated by commas, and the names are separated from the values by colons. The names are always strings and are enclosed in (double) quotation marks.

Here is an example of a JSON array:

```
[ "Robert J. Oppenheimer", 98, false, "Beechwood 45789" ]
```

In a JSON array, the values are separated by commas. JSON arrays and objects are fully nestable. That is, values in both JSON objects and JSON arrays may be JSON strings, JSON numbers, JSON Booleans (represented by the JSON literals true and false), JSON nulls (represented by the JSON literal null), JSON objects, or JSON arrays.

JSON can be used to represent associative arrays — arrays whose elements are addressed by content, not by position. An associative array can be represented in JSON as a JSON object whose members are name-value pairs; the name is used as the “index” into the “array” — that is, to locate the appropriate member in the object — and the value is used as the content of the appropriate member. Here is such an associative array:

```
{ "Isaac Newton" : "apple harvester",
  "Robert J. Oppenheimer" : "security risk",
  "Albert Einstein" : "patent clerk",
  "Stephen Hawking" : "inspiration" }
```

An extremely important part of JSON’s design is that it is inherently schema-less. Any JSON object can be modified by adding new name-value pairs, even with names that were never considered when the object was