

PD ISO/IEC TR 19075-7:2017



BSI Standards Publication

Information technology — Database languages — SQL Technical Reports

Part 7: Polymorphic table functions in SQL

National foreword

This Published Document is the UK implementation of ISO/IEC TR 19075-7:2017.

The UK participation in its preparation was entrusted to Technical Committee IST/40, Data management and interchange.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2017.

Published by BSI Standards Limited 2017

ISBN 978 0 580 95286 9

ICS 35.060

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 30 April 2017.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

**Information technology — Database
languages — SQL Technical Reports —
Part 7:
Polymorphic table functions in SQL**

*Technologies de l'information — Langages de base de données — SQL
rapport techniques —*

Partie 7: Fonctions de table polymorphes dans SQL



Contents

Page

Foreword.....	xi
Introduction.....	xii
1 Scope.....	1
2 Normative references.....	3
2.1 ISO and IEC standards	3
3 Introduction to Polymorphic Table Functions.....	5
3.1 Audiences.....	5
3.2 Motivating examples.....	6
3.2.1 CSVreader.....	6
3.2.2 Pivot.....	7
3.2.3 Score.....	9
3.2.4 TopNplus.....	12
3.2.5 ExecR.....	15
3.2.6 Similarity.....	16
3.2.7 UDjoin.....	18
3.2.8 MapReduce.....	19
3.3 The life cycle of a PTF.....	20
4 PTF processing model.....	23
4.1 Processing phases	23
4.2 Virtual processors	23
4.3 PTF component procedures	23
4.4 Input table characteristics	24
4.5 Partitioning and ordering	25
4.6 Flow of control	26
4.7 Flow of information	27
4.8 Flow of row types.....	28
4.9 Pass-through columns	30
4.10 Security model	30
4.11 Conformance features.....	31
5 Specification.....	35
5.1 Functional specification	35
5.1.1 Parameter list	35
5.1.2 Input table semantics	36
5.1.3 Prunability.....	37
5.1.4 Pass-through columns.....	37

5.1.5	Result row type	38
5.1.6	Determinism	38
5.1.7	SQL-data access	39
5.1.8	Documenting the PTF to the query author	39
5.2	Design specification	40
5.2.1	Name the component procedures	40
5.2.2	Private data	40
5.2.3	Routine characteristics of the component procedures	41
5.2.4	Component procedure signatures	42
6	Data definition language	47
6.1	PTF creation	47
6.2	PTF component procedures	49
6.3	Altering PTF component procedures and PTFs	49
6.4	Dropping a PTF and its component procedures	50
7	Implementation	51
7.1	PTF descriptor areas	51
7.1.1	PTF descriptor area header	52
7.1.2	SQL item descriptor areas for row types	53
7.1.3	SQL item descriptor areas for partitioning	57
7.1.4	SQL item descriptor areas for ordering	57
7.2	PTF extended names	57
7.3	Reading a PTF descriptor area	58
7.4	Writing a PTF descriptor area	58
7.4.1	Using DESCRIBE to populate a PTF descriptor area	59
7.4.2	Using SET DESCRIPTOR to populate a PTF descriptor area	59
7.4.3	Using COPY DESCRIPTOR to populate a PTF descriptor area	60
7.5	Reading a PTF input cursor	61
7.6	Outputting a row	62
8	Invocation	65
8.1	<table primary>	65
8.2	<PTF derived table>	65
8.3	Proper result correlation name and proper result column naming	65
8.4	<routine invocation>	66
8.5	<table argument>	67
8.6	<table argument proper>	68
8.6.1	<table or query name>	68
8.6.2	<table subquery>	68
8.6.3	Nested table function invocation	69
8.7	Table argument correlation name	69
8.8	Table argument column renaming	70
8.9	Range variables and column renaming in nested PTF	70
8.10	Partitioning	70
8.11	Pruning	71

8.12	Ordering.	71
8.13	Copartitioning.	72
8.14	Cross products of partitions	73
8.15	<descriptor argument>.	74
9	Compilation.	75
9.1	Calling the describe component procedure.	75
9.2	Inside the describe component procedure.	75
9.3	Using the result of describe	76
10	Optimization.	77
11	Execution.	79
11.1	Partitions and virtual processors	79
11.2	Calling the start component procedure	80
11.3	Inside the start component procedure	81
11.4	Calling the PTF fulfill component procedure.	81
11.5	Inside the PTF fulfill component procedure	81
11.6	Closing cursors	81
11.7	Calling the PTF finish component procedure.	81
11.8	Inside the PTF finish component procedure.	82
11.9	Collecting the output.	82
11.10	Cleanup on a virtual processor	82
11.11	Final result	82
12	Examples.	83
12.1	Projection.	84
12.1.1	Overview.	84
12.1.2	Functional specification of Projection	84
12.1.3	Design specification for Projection.	85
12.1.4	Projection component procedures.	85
12.1.5	Invoking Projection.	87
12.1.6	Calling Projection_describe	87
12.1.7	Inside Projection_describe	89
12.1.8	Result of Projection_describe	91
12.1.9	Virtual processors for Projection	91
12.1.10	Calling Projection_fulfill	92
12.1.11	Inside Projection_fulfill	93
12.1.12	Collecting the results	93
12.1.13	Cleanup.	94
12.2	CSVreader.	95
12.2.1	Overview.	95
12.2.2	Functional specification of CSVreader.	95
12.2.3	Design specification for CSVreader.	95
12.2.4	CSVreader component procedures	96
12.2.5	Implementation of CSVreader	97
12.2.6	Invoking CSVreader	98

12.2.7	Calling CSVreader_describe	98
12.2.8	Inside CSVreader_describe	100
12.2.9	Result of CSVreader_describe	101
12.2.10	Virtual processor for CSVreader	103
12.2.11	Calling CSVreader_start	103
12.2.12	Inside CSVreader_start	104
12.2.13	Calling CSVreader_fulfill	104
12.2.14	Inside CSVreader_fulfill	104
12.2.15	Collecting the output	105
12.2.16	Calling CSVreader_finish	105
12.2.17	Inside CSVreader_finish	106
12.2.18	Cleanup	106
12.3	Pivot	107
12.3.1	Overview	107
12.3.2	Functional specification of Pi vot	107
12.3.3	Design specification for Pivot	108
12.3.4	Pivot component procedures	108
12.3.5	Invoking pi vot	109
12.3.6	Calling Pi vot_describe	109
12.3.7	Inside Pi vot_describe	113
12.3.8	Result of Pi vot_describe	115
12.3.9	Virtual processors for Pivot	117
12.3.10	Calling Pi vot_fulfill	117
12.3.11	Inside Pi vot_fulfill	118
12.3.12	Collecting the results	118
12.3.13	Cleanup	119
12.4	Score	120
12.4.1	Overview	120
12.4.2	Functional specification of Score	120
12.4.3	Design specification for Score	120
12.4.4	Score component procedures	121
12.4.5	Invoking Score	122
12.4.6	Calling Score_describe	122
12.4.7	Inside Score_describe	125
12.4.8	Result of Score_describe	126
12.4.9	Virtual processors for Score	127
12.4.10	Calling Score_fulfill	129
12.4.11	Inside Score_fulfill	129
12.4.12	Collecting the output	130
12.4.13	Cleanup	131
12.5	TopNplus	132
12.5.1	Overview	132
12.5.2	Functional specification of TopNplus	132
12.5.3	Design specification for TopNplus	132

12.5.4	TopNplus component procedures	133
12.5.5	Invoking TopNplus	134
12.5.6	Calling TopNplus_describe	134
12.5.7	Inside TopNplus_describe	137
12.5.8	Result of TopNplus_describe	139
12.5.9	Virtual processors for TopNplus	139
12.5.10	Calling TopNplus_fulfill	141
12.5.11	Inside TopNplus_fulfill	141
12.5.12	Collecting the output	142
12.5.13	Cleanup	142
12.5.14	TopNplus using pass-through columns	143
12.6	ExecR	145
12.6.1	Overview	145
12.6.2	Functional specification of ExecR	145
12.6.3	Design specification for ExecR	145
12.6.4	ExecR component procedures	146
12.6.5	Invoking ExecR	147
12.6.6	Calling ExecR_describe	147
12.6.7	Inside ExecR_describe	149
12.6.8	Result of ExecR_describe	150
12.6.9	Virtual processors for ExecR	150
12.6.10	Calling ExecR_start	151
12.6.11	Inside ExecR_start	152
12.6.12	Calling ExecR_fulfill	152
12.6.13	Inside ExecR_fulfill	152
12.6.14	Collecting the output	153
12.6.15	Calling ExecR_finish	153
12.6.16	Inside ExecR_finish	153
12.6.17	Cleanup	153
12.7	Similarity	154
12.7.1	Overview	154
12.7.2	Functional specification of Similarity	154
12.7.3	Design specification for Similarity	154
12.7.4	Similarity component procedures	155
12.7.5	Invoking Similarity	156
12.7.6	Calling Similarity_describe	156
12.7.7	Inside Similarity_describe	158
12.7.8	Result of Similarity_describe	159
12.7.9	Virtual processors for Similarity	159
12.7.10	Calling Similarity_fulfill	163
12.7.11	Inside Similarity_fulfill	163
12.7.12	Collecting the output	164
12.7.13	Cleanup	164
12.8	UDjoin	165

12.8.1	Overview.	165
12.8.2	Functional specification of UDjoin.	165
12.8.3	Design specification for UDjoin.	165
12.8.4	UDjoin component procedures.	166
12.8.5	Invoking UDjoin.	167
12.8.6	Calling UDjoin_describe.	167
12.8.7	Inside UDjoin_describe.	168
12.8.8	Result of UDjoin_describe.	168
12.8.9	Virtual processors for UDjoin.	168
12.8.10	Calling UDjoin_fulfill.	168
12.8.11	Inside UDjoin_fulfill.	169
12.8.12	Collecting the output.	170
12.8.13	Cleanup.	170
12.9	Nested PTF in vocation.	171
12.9.1	Nested PTF syntax and semantics.	171
12.9.2	Nested PTF compilation.	174
12.9.3	Nested PTF execution.	175
12.9.4	The PTF author's view of nested PTF in vocations.	176
Bibliography.		177
Index.		179

Tables

Table	Page
1 Primary audiences for Clauses and Subclauses in this Technical Report.	20
2 PTF routine characteristics.	41
3 Table parameter semantics.	43
4 Corresponding PTF component procedure parameters	44
5 PTF descriptor area	52
6 PTF descriptor area header	52
7 Relevant SQL item descriptor components	53

Figures

Figure	Page
1 PTF information flow.	28
2 Row type relationships.	29
3 Nested PTF data flow.	172
4 Flow of row types.	174
5 Simplified flow of row types.	175
6 Net effect of complete compilation	175

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.

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.

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 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 may 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, “Introduction to Polymorphic Table Functions”**, provides an introduction to polymorphic table functions, the requirements leading to their incorporation into SQL, and illustrations of their use.
- 4) **Clause 4, “PTF processing model”**, describes the abstract processing model for polymorphic table functions in the context of an SQL-implementation.
- 5) **Clause 5, “Specification”**, describes the manner in which polymorphic table functions are specified in the SQL standard.
- 6) **Clause 6, “Data definition language”**, provides the syntax and semantics of the SQL statements that create, modify, and drop polymorphic table functions.
- 7) **Clause 7, “Implementation”**, guides authors of polymorphic table functions through the steps required to create all of the functions necessary to accomplish particular purposes.
- 8) **Clause 8, “Invocation”**, supplies the information necessary for application writers, especially SQL query authors, to take advantage of the polymorphic table functions that are available to them.
- 9) **Clause 9, “Compilation”**, is directed at the authors of polymorphic table functions and of SQL database systems to guide them in the steps required to compile polymorphic table functions in the context of a particular SQL-implementation.
- 10) **Clause 10, “Optimization”**, describes the various aspects of polymorphic functions of which the authors of such functions and the authors of SQL-implementations must be aware to adequately optimize the execution of such functions.
- 11) **Clause 11, “Execution”**, discusses the details of executing polymorphic table functions in the context of the processing model.
- 12) **Clause 12, “Examples”**, supplies numerous examples in detail with commentaries to explain the various use cases, the requirements that relate to polymorphic table functions, and the specifics of the solutions for each use case.

Information technology — Database languages — SQL Technical Reports —**Part 7:****Polymorphic table functions in SQL****1 Scope**

This Technical Report describes the definition and use of polymorphic table functions in SQL.

The Report discusses the following features of the SQL Language:

- The processing model of polymorphic table functions in the context of SQL.
- The creation and maintenance of polymorphic table functions.
- Issues related to methods of implementing polymorphic table functions.
- How polymorphic table functions are invoked by application programs.
- Issues concerning compilation, optimization, and execution of polymorphic table functions.