

First edition
2021-08

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

Part 7: Polymorphic table functions

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

Partie 7: Fonctions de table polymorphes



Reference number
ISO/IEC 19075-7:2021(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

| Contents | Page |
|---|-------------|
| Foreword..... | xii |
| Introduction..... | xiv |
| 1 Scope..... | 1 |
| 2 Normative references..... | 2 |
| 3 Terms and definitions..... | 3 |
| 4 Introduction to polymorphic table functions..... | 4 |
| 4.1 Context of polymorphic table functions..... | 4 |
| 4.2 What is a polymorphic table function?..... | 4 |
| 4.3 Audiences..... | 4 |
| 4.4 Motivating examples..... | 5 |
| 4.4.1 Summary of motivating examples..... | 5 |
| 4.4.2 CSVreader..... | 5 |
| 4.4.3 Pivot..... | 6 |
| 4.4.4 Score..... | 8 |
| 4.4.5 TopNplus..... | 11 |
| 4.4.6 ExecR..... | 14 |
| 4.4.7 Similarity..... | 15 |
| 4.4.8 UDjoin..... | 17 |
| 4.4.9 MapReduce..... | 18 |
| 4.5 The life cycle of a PTF..... | 19 |
| 5 PTF processing model..... | 21 |
| 5.1 Introduction to the PTF processing model..... | 21 |
| 5.2 Processing phases..... | 21 |
| 5.3 Virtual processors..... | 21 |
| 5.4 PTF component procedures..... | 21 |
| 5.5 Input table characteristics..... | 22 |
| 5.6 Partitioning and ordering..... | 23 |
| 5.7 Flow of control..... | 24 |
| 5.8 Flow of information..... | 25 |
| 5.9 Flow of row types..... | 26 |
| 5.10 Pass-through columns..... | 27 |
| 5.11 Security model..... | 28 |
| 5.12 Conformance features..... | 29 |
| 6 Specification..... | 31 |
| 6.1 Introduction to the specification of PTFs..... | 31 |
| 6.2 Functional specification..... | 31 |
| 6.2.1 Introduction to the functional specification..... | 31 |
| 6.2.2 Parameter list..... | 31 |

| | | |
|----------|---|-----------|
| 6.2.3 | Input table semantics. | 32 |
| 6.2.4 | Prunability. | 33 |
| 6.2.5 | Pass-through columns. | 33 |
| 6.2.6 | Result row type. | 34 |
| 6.2.7 | Determinism. | 34 |
| 6.2.8 | SQL-data access. | 35 |
| 6.2.9 | Documenting the PTF to the query author. | 35 |
| 6.3 | Design specification. | 36 |
| 6.3.1 | Introduction to the design specification. | 36 |
| 6.3.2 | Name the component procedures. | 36 |
| 6.3.3 | Private data. | 36 |
| 6.3.4 | Routine characteristics of the component procedures. | 37 |
| 6.3.5 | Component procedure signatures. | 38 |
| 7 | Data definition language. | 43 |
| 7.1 | Introduction to data definition language. | 43 |
| 7.2 | PTF creation. | 43 |
| 7.3 | PTF component procedures. | 45 |
| 7.4 | Altering PTF component procedures and PTFs. | 45 |
| 7.5 | Dropping a PTF and its component procedures. | 45 |
| 8 | Implementation. | 46 |
| 8.1 | Introduction to PTF implementation. | 46 |
| 8.2 | PTF descriptor areas. | 46 |
| 8.2.1 | Introduction to PTF descriptor areas. | 46 |
| 8.2.2 | PTF descriptor area header. | 47 |
| 8.2.3 | SQL item descriptor areas for row types. | 48 |
| 8.2.4 | SQL item descriptor areas for partitioning. | 51 |
| 8.2.5 | SQL item descriptor areas for ordering. | 51 |
| 8.3 | PTF extended names. | 52 |
| 8.4 | Reading a PTF descriptor area. | 52 |
| 8.5 | Writing a PTF descriptor area. | 53 |
| 8.5.1 | Introduction to writing a PTF descriptor area. | 53 |
| 8.5.2 | Using DESCRIBE to populate a PTF descriptor area. | 53 |
| 8.5.3 | Using SET DESCRIPTOR to populate a PTF descriptor area. | 54 |
| 8.5.4 | Using COPY DESCRIPTOR to populate a PTF descriptor area. | 55 |
| 8.6 | Reading a PTF input cursor. | 55 |
| 8.7 | Outputting a row. | 56 |
| 9 | Invocation. | 58 |
| 9.1 | Introduction to PTF invocation. | 58 |
| 9.2 | <table primary>. | 58 |
| 9.3 | <PTF derived table>. | 58 |
| 9.4 | Proper result correlation name and proper result column naming. | 58 |
| 9.5 | <routine invocation>. | 59 |
| 9.6 | <table argument>. | 60 |
| 9.7 | <table argument proper>. | 60 |
| 9.7.1 | Types of table arguments. | 60 |
| 9.7.2 | <table or query name>. | 61 |

| | | |
|-----------|---|-----------|
| 9.7.3 | <table subquery>..... | 61 |
| 9.7.4 | Nested table function invocation. | 62 |
| 9.8 | Table argument correlation name. | 62 |
| 9.9 | Table argument column renaming. | 62 |
| 9.10 | Range variables and column renaming in nested PTF. | 63 |
| 9.11 | Partitioning. | 63 |
| 9.12 | Pruning. | 64 |
| 9.13 | Ordering. | 64 |
| 9.14 | Co-partitioning. | 65 |
| 9.15 | Cross products of partitions. | 66 |
| 9.16 | <descriptor argument>..... | 66 |
| 10 | Compilation..... | 67 |
| 10.1 | Introduction to PTF compilation. | 67 |
| 10.2 | Calling the describe component procedure. | 67 |
| 10.3 | Inside the describe component procedure. | 67 |
| 10.4 | Using the result of describe. | 68 |
| 11 | Optimization..... | 69 |
| 12 | Execution..... | 70 |
| 12.1 | Introduction to PTF execution. | 70 |
| 12.2 | Partitions and virtual processors. | 70 |
| 12.3 | Calling the start component procedure. | 71 |
| 12.4 | Inside the start component procedure. | 71 |
| 12.5 | Calling the PTF fulfill component procedure. | 72 |
| 12.6 | Inside the PTF fulfill component procedure. | 72 |
| 12.7 | Closing cursors. | 72 |
| 12.8 | Calling the PTF finish component procedure. | 72 |
| 12.9 | Inside the PTF finish component procedure. | 73 |
| 12.10 | Collecting the output. | 73 |
| 12.11 | Cleanup on a virtual processor. | 73 |
| 12.12 | Final result. | 73 |
| 13 | Examples..... | 74 |
| 13.1 | Introduction to the examples. | 74 |
| 13.2 | Projection. | 75 |
| 13.2.1 | Overview. | 75 |
| 13.2.2 | Functional specification of Projection. | 75 |
| 13.2.3 | Design specification for Projection. | 76 |
| 13.2.4 | Projection component procedures. | 76 |
| 13.2.5 | Invoking Projection. | 78 |
| 13.2.6 | Calling Projection_describe. | 78 |
| 13.2.7 | Inside Projection_describe. | 80 |
| 13.2.8 | Result of Projection_describe. | 82 |
| 13.2.9 | Virtual processors for Projection. | 82 |
| 13.2.10 | Calling Projection_fulfill. | 83 |
| 13.2.11 | Inside Projection_fulfill. | 84 |
| 13.2.12 | Collecting the results. | 84 |
| 13.2.13 | Cleanup. | 85 |

| | | |
|---------|--|-----|
| 13.3 | CSVreader..... | 86 |
| 13.3.1 | Overview..... | 86 |
| 13.3.2 | Functional specification of CSVreader..... | 86 |
| 13.3.3 | Design specification for CSVreader..... | 86 |
| 13.3.4 | CSVreader component procedures..... | 87 |
| 13.3.5 | Implementation of CSVreader..... | 88 |
| 13.3.6 | Invoking CSVreader..... | 88 |
| 13.3.7 | Calling CSVreader_describe..... | 89 |
| 13.3.8 | Inside CSVreader_describe..... | 91 |
| 13.3.9 | Result of CSVreader_describe..... | 92 |
| 13.3.10 | Virtual processor for CSVreader..... | 94 |
| 13.3.11 | Calling CSVreader_start..... | 94 |
| 13.3.12 | Inside CSVreader_start..... | 95 |
| 13.3.13 | Calling CSVreader_fulfill..... | 95 |
| 13.3.14 | Inside CSVreader_fulfill..... | 95 |
| 13.3.15 | Collecting the output..... | 96 |
| 13.3.16 | Calling CSVreader_finish..... | 96 |
| 13.3.17 | Inside CSVreader_finish..... | 96 |
| 13.3.18 | Cleanup..... | 97 |
| 13.4 | Pivot..... | 98 |
| 13.4.1 | Overview..... | 98 |
| 13.4.2 | Functional specification of Pivot..... | 98 |
| 13.4.3 | Design specification for Pivot..... | 98 |
| 13.4.4 | Pivot component procedures..... | 99 |
| 13.4.5 | Invoking Pivot..... | 100 |
| 13.4.6 | Calling Pivot_describe..... | 100 |
| 13.4.7 | Inside Pivot_describe..... | 104 |
| 13.4.8 | Result of Pivot_describe..... | 105 |
| 13.4.9 | Virtual processors for Pivot..... | 107 |
| 13.4.10 | Calling Pivot_fulfill..... | 108 |
| 13.4.11 | Inside Pivot_fulfill..... | 108 |
| 13.4.12 | Collecting the results..... | 109 |
| 13.4.13 | Cleanup..... | 109 |
| 13.5 | Score..... | 110 |
| 13.5.1 | Overview..... | 110 |
| 13.5.2 | Functional specification of Score..... | 110 |
| 13.5.3 | Design specification for Score..... | 110 |
| 13.5.4 | Score component procedures..... | 111 |
| 13.5.5 | Invoking Score..... | 112 |
| 13.5.6 | Calling Score_describe..... | 112 |
| 13.5.7 | Inside Score_describe..... | 115 |
| 13.5.8 | Result of Score_describe..... | 116 |
| 13.5.9 | Virtual processors for Score..... | 117 |
| 13.5.10 | Calling Score_fulfill..... | 119 |
| 13.5.11 | Inside Score_fulfill..... | 119 |
| 13.5.12 | Collecting the output..... | 120 |
| 13.5.13 | Cleanup..... | 121 |

| | | |
|---------|---|-----|
| 13.6 | TopNplus. | 122 |
| 13.6.1 | Overview. | 122 |
| 13.6.2 | Functional specification of TopNplus. | 122 |
| 13.6.3 | Design specification for TopNplus. | 122 |
| 13.6.4 | TopNplus component procedures. | 123 |
| 13.6.5 | Invoking TopNplus. | 124 |
| 13.6.6 | Calling TopNplus_describe. | 124 |
| 13.6.7 | Inside TopNplus_describe. | 127 |
| 13.6.8 | Result of TopNplus_describe. | 129 |
| 13.6.9 | Virtual processors for TopNplus. | 129 |
| 13.6.10 | Calling TopNplus_fulfill. | 131 |
| 13.6.11 | Inside TopNplus_fulfill. | 131 |
| 13.6.12 | Collecting the output. | 132 |
| 13.6.13 | Cleanup. | 132 |
| 13.6.14 | TopNplus using pass-through columns. | 132 |
| 13.7 | ExecR. | 135 |
| 13.7.1 | Overview. | 135 |
| 13.7.2 | Functional specification of ExecR. | 135 |
| 13.7.3 | Design specification for ExecR. | 135 |
| 13.7.4 | ExecR component procedures. | 136 |
| 13.7.5 | Invoking ExecR. | 137 |
| 13.7.6 | Calling ExecR_describe. | 137 |
| 13.7.7 | Inside ExecR_describe. | 139 |
| 13.7.8 | Result of ExecR_describe. | 140 |
| 13.7.9 | Virtual processors for ExecR. | 140 |
| 13.7.10 | Calling ExecR_start. | 141 |
| 13.7.11 | Inside ExecR_start. | 141 |
| 13.7.12 | Calling ExecR_fulfill. | 142 |
| 13.7.13 | Inside ExecR_fulfill. | 142 |
| 13.7.14 | Collecting the output. | 142 |
| 13.7.15 | Calling ExecR_finish. | 143 |
| 13.7.16 | Inside ExecR_finish. | 143 |
| 13.7.17 | Cleanup. | 143 |
| 13.8 | Similarity. | 144 |
| 13.8.1 | Overview. | 144 |
| 13.8.2 | Functional specification of Similarity. | 144 |
| 13.8.3 | Design specification for Similarity. | 144 |
| 13.8.4 | Similarity component procedures. | 145 |
| 13.8.5 | Invoking Similarity. | 145 |
| 13.8.6 | Calling Similarity_describe. | 146 |
| 13.8.7 | Inside Similarity_describe. | 148 |
| 13.8.8 | Result of Similarity_describe. | 149 |
| 13.8.9 | Virtual processors for Similarity. | 149 |
| 13.8.10 | Calling Similarity_fulfill. | 153 |
| 13.8.11 | Inside Similarity_fulfill. | 153 |
| 13.8.12 | Collecting the output. | 154 |
| 13.8.13 | Cleanup. | 154 |

| | | |
|---------|--|------------|
| 13.9 | UDjoin..... | 155 |
| 13.9.1 | Overview..... | 155 |
| 13.9.2 | Functional specification of UDjoin..... | 155 |
| 13.9.3 | Design specification for UDjoin..... | 155 |
| 13.9.4 | UDjoin component procedures..... | 156 |
| 13.9.5 | Invoking UDjoin..... | 156 |
| 13.9.6 | Calling UDjoin_describe..... | 157 |
| 13.9.7 | Inside UDjoin_describe..... | 157 |
| 13.9.8 | Result of UDjoin_describe..... | 158 |
| 13.9.9 | Virtual processors for UDjoin..... | 158 |
| 13.9.10 | Calling UDjoin_fulfill..... | 158 |
| 13.9.11 | Inside UDjoin_fulfill..... | 159 |
| 13.9.12 | Collecting the output..... | 159 |
| 13.9.13 | Cleanup..... | 159 |
| 13.10 | Nested PTF invocation..... | 160 |
| 13.10.1 | Introduction to nested PTF invocation..... | 160 |
| 13.10.2 | Nested PTF syntax and semantics..... | 160 |
| 13.10.3 | Nested PTF compilation..... | 162 |
| 13.10.4 | Nested PTF execution..... | 164 |
| 13.10.5 | The PTF author's view of nested PTF invocations..... | 165 |
| | Bibliography..... | 166 |
| | Index..... | 167 |

Tables

| Table | Page | |
|--------------|--|-----|
| 1 | Results of CVSreader sample query. | 6 |
| 2 | Sample data for pivot example. | 8 |
| 3 | Pivot example result with sample data. | 8 |
| 4 | Score example: contents of Model table. | 10 |
| 5 | Score example: data to be scored. | 10 |
| 6 | Score example: result. | 10 |
| 7 | TopNplus example: input data. | 12 |
| 8 | TopNplus example: East input data. | 12 |
| 9 | TopNplus example: West input data. | 12 |
| 10 | TopNplus example: result. | 13 |
| 11 | TopNplus example: result with statistics. | 14 |
| 12 | Similarity example: result. | 17 |
| 13 | UDjoin example: result. | 18 |
| 14 | Primary audiences for Clauses and Subclauses in this document. | 19 |
| 15 | Examples and table parameter characteristics. | 23 |
| 16 | Schematic execution plan. | 24 |
| 17 | PTF routine characteristics. | 37 |
| 18 | Table parameter semantics. | 39 |
| 19 | Corresponding PTF component procedure parameters. | 40 |
| 20 | PTF descriptor area. | 47 |
| 21 | PTF descriptor area header. | 47 |
| 22 | Relevant SQL item descriptor components. | 48 |
| 23 | Scenarios illustrated by the examples. | 74 |
| 24 | Deriving parameter names. | 77 |
| 25 | Descriptor for Emp's row type. | 78 |
| 26 | Input table's requested row type. | 79 |
| 27 | Descriptor generated from query's argument. | 79 |
| 28 | Descriptor of initial result row. | 80 |
| 29 | Populated input request descriptor. | 82 |
| 30 | Populated initial result row descriptor. | 82 |
| 31 | Input cursor's row type descriptor. | 83 |
| 32 | PTF descriptor area for floats. | 90 |
| 33 | PTF descriptor area for dates. | 90 |
| 34 | PTF descriptor area for initial result. | 90 |
| 35 | PTF descriptor area after describe of initial result. | 92 |
| 36 | Row type corresponding to descriptor. | 93 |
| 37 | Pivot: row type descriptor. | 101 |
| 38 | Pivot: requested row type descriptor. | 102 |
| 39 | Pivot: descriptor for first argument. | 102 |
| 40 | Pivot: descriptor for second argument. | 103 |
| 41 | Pivot: descriptor for third argument. | 103 |
| 42 | Pivot: initial row type descriptor. | 104 |
| 43 | Pivot: requested row type descriptor. | 106 |
| 44 | Pivot: initial result row type descriptor. | 106 |
| 45 | Pivot: columns of the result row type. | 107 |
| 46 | Score: first argument full row type descriptor. | 112 |
| 47 | Score: first argument requested row type descriptor. | 113 |

| | | |
|----|---|-----|
| 48 | Score: second argument full row type descriptor. | 113 |
| 49 | Score: second argument partitioning descriptor. | 114 |
| 50 | Score: second argument ordering descriptor. | 114 |
| 51 | Score: initial result row type descriptor. | 115 |
| 52 | Score: first input table requested row descriptors. | 116 |
| 53 | Score: second input table requested row descriptors. | 116 |
| 54 | Score: result row type. | 117 |
| 55 | Score: second input table data. | 117 |
| 56 | Score: first input table data. | 118 |
| 57 | Score: virtual processor cursors. | 118 |
| 58 | Score: full row type descriptor. | 125 |
| 59 | Score: partitioning descriptor. | 126 |
| 60 | Score: ordering descriptor. | 126 |
| 61 | Score: requested row type descriptor. | 126 |
| 62 | Score: intermediate row type descriptor. | 127 |
| 63 | Score: complete result row type. | 129 |
| 64 | Wcore: sample data. | 129 |
| 65 | ExecR: input table partitioning descriptor. | 138 |
| 66 | ExecR: fifth descriptor. | 138 |
| 67 | ExecR: sixth descriptor. | 139 |
| 68 | ExecR: result row type descriptor. | 140 |
| 69 | Similarity: full row type descriptor for Input1. | 146 |
| 70 | Similarity: partitioning descriptor for Input1. | 147 |
| 71 | Similarity: ordering descriptor for Input1. | 147 |
| 72 | Similarity: result row type. | 149 |
| 73 | Similarity: query result. | 149 |
| 74 | Similarity: master list query result. | 151 |
| 75 | Similarity: master list query result without co-partitioning. | 152 |
| 76 | Similarity: complete output. | 154 |
| 77 | UDjoin: result row type. | 158 |
| 78 | Nested PTF posited PTFs. | 160 |

Figures

| Figure | | Page |
|---------------|---|-------------|
| 1 | PTF information flow. | 26 |
| 2 | Row type relationships. | 27 |
| 3 | Nested PTF data flow. | 161 |
| 4 | Flow of row types. | 163 |
| 5 | Simplified flow of row types. | 164 |
| 6 | Net effect of complete compilation. | 164 |

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 19075-7 cancels and replaces ISO/IEC TR 19075-7:2017.

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.

This is a preview of "ISO/IEC 19075-7:2021". [Click here to purchase the full version from the ANSI store.](#)

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.

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, "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.
- 5) **Clause 5, "PTF processing model"**, describes the abstract processing model for polymorphic table functions in the context of an SQL-implementation.
- 6) **Clause 6, "Specification"**, describes the manner in which polymorphic table functions are specified in the SQL standard.
- 7) **Clause 7, "Data definition language"**, provides the syntax and semantics of the SQL statements that create, modify, and drop polymorphic table functions.
- 8) **Clause 8, "Implementation"**, guides authors of polymorphic table functions through the steps required to create all of the functions necessary to accomplish particular purposes.
- 9) **Clause 9, "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.
- 10) **Clause 10, "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.
- 11) **Clause 11, "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.
- 12) **Clause 12, "Execution"**, discusses the details of executing polymorphic table functions in the context of the processing model.
- 13) **Clause 13, "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.