

Australian Standard™

**Information technology—Database
languages—SQL**

**Part 10: Object Language Bindings
(SQL/OLB)**



**STANDARDS
AUSTRALIA**

This Australian Standard was prepared by Committee IT-027, Data Management and Interchange. It was approved on behalf of the Council of Standards Australia on 16 May 2005. This Standard was published on 16 June 2005.

The following are represented on Committee IT-027:

Association of Superannuation Funds of Australia
Australia Post
Australia Bankers Association
Australian Bureau of Statistics
Australian Computer Society
Australian Customs Service
Australian Electoral Commission
Australian Electric and Electronic Manufacturers Association
Australian Industry Group
Australian Information Industry Association
Australian Institute of Health and Welfare
Australian Taxation Office
Centrelink
Department of Defence
Department of Immigration, Multicultural and Indigenous Affairs
Health Insurance Commission
Data Management Association Australia

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about Standards can be found by visiting the Standards Web Shop at www.standards.com.au and looking up the relevant Standard in the on-line catalogue.

Alternatively, the printed Catalogue provides information current at 1 January each year and the monthly magazine, *The Global Standard*, has a full listing of revisions and amendments published each month.

Australian Standards™ and other products and services developed by Standards Australia are published and distributed under contract by SAI Global, which operates the Standards Web Shop.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Contact us via email at mail@standards.org.au, or write to the Chief Executive, Standards Australia, GPO Box 5420, Sydney, NSW 2001.

Australian Standard™

**Information technology—Database
languages—SQL**

**Part 10: Object Language Bindings
(SQL/OLB)**

First published as AS ISO/IEC 9075.10—2005.

COPYRIGHT

© Standards Australia

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Published by Standards Australia GPO Box 5420, Sydney, NSW 2001, Australia

ISBN 0 7337 6769 9

PREFACE

This Standard was prepared by the Standards Australia Committee IT-027, Data Management and Interchange.

This Standard is identical with, and has been reproduced from ISO/IEC 9075-10:2003, *Information technology—Database languages—SQL—Part 10: Object Language Bindings (SQL/OLB)*.

The objective of this Standard is to provide database designers, administrators and developers extension definitions of database language SQL to support embedding of SQL statements into the programs written in the Java(tm) programming language.

This Standard is Part 10 of AS 9075, *Information technology—Database languages—SQL*, which is published in parts as follows:

- Part 1: Framework (SQL/Framework)
- Part 2: Foundation (SQL/Foundation)
- Part 3: Call-Level Interface (SQL/CLI)
- Part 4: Persistent Stored Modules (SQL/PSM)
- Part 5: Host Language Bindings (SQL/Bindings)
- Part 9: Management of External Data (SQL/MED)
- Part 10: Object Language Bindings (SQL/OLB) (this Standard)
- Part 11: Information and Definition Schemas (SQL/Schemas)
- Part 13: SQL Routines and Types Using the Java™ Programming Language (SQL/JRT)
- Part 14: XML-Related Specifications (SQL/XML)

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover.
- (b) In the source text ‘this part of ISO/IEC 9075’ should read ‘this Australian Standard’.
- (c) A full point substitutes for a comma when referring to a decimal marker.

References to International Standards should be replaced by references to Australian or Australian/New Zealand Standards, as follows:

<i>Reference to International Standard</i>	<i>Australian Standard</i>
ISO/IEC	AS
9075 Information technology—Database language—SQL	9075 Information technology—Database languages—SQL
9075-1 Part 1: Framework (SQL/Framework)	9075.1 Part 1: Framework (SQL/Framework)
9075-2 Part 2: Foundation (SQL/Foundation)	9075.2 Part 2: Foundation (SQL/Foundation)
9075-4 Part 4: Persistent Stored Modules (SQL/PSM)	9075.4 Part 4: Persistent Stored Modules (SQL/PSM)
9075-9 Part 9: Management of External Data (SQL/MED)	9075.9 Part 9: Management of External Data (SQL/MED)
9075-11 Part 11: Information and Definition Schemas (SQL/Schemas)	9075.11 Part 11: Information and Definition Schemas (SQL/Schemas)

Only referenced documents that have been adopted as Australian or Australian/New Zealand Standards have been listed.

CONTENTS

Page

1	Scope	1
2	Normative references	3
2.1	JTC 1 standards	3
2.2	Other international standards	3
3	Definitions, notations, and conventions	5
3.1	Definitions	5
3.1.1	Definitions provided in Part 10	5
3.2	Notation	6
3.3	Conventions	7
3.3.1	Use of terms	7
3.3.1.1	Other terms	7
3.3.2	Specification of translator-generated classes	8
4	Concepts	11
4.1	Introduction	11
4.2	Embedded syntax	11
4.3	SQL constructs	12
4.4	Character strings	12
4.4.1	Unicode support	12
4.4.2	Character sets	13
4.5	Host variables	13
4.6	Host expressions	13
4.7	SQLJ clauses	14
4.8	Database connection context	14
4.9	Default connection context	15
4.10	Schema checking using exemplar schemas	15
4.11	Using multiple SQLJ contexts and connections	16
4.12	Dynamic SQL and JDBC/SQLJ Connection interoperability	16
4.12.1	Creating an SQLJ ConnectionContext from a JDBC connection	16
4.12.2	Obtaining a JDBC connection from an SQLJ ConnectionContext	17
4.12.3	Connection sharing	17
4.12.4	Connection resource management	17
4.13	SQL execution control and status	18
4.14	Iterators	19
4.15	Input and output assignability	20

4.16	Calls to stored procedures and functions.	31
4.17	Multiple JDBC ResultSet objects from stored procedure calls.	31
4.17.1	Resource management with multiple results.	32
4.18	JDBC/SQLJ ResultSet interoperability.	32
4.18.1	Creating an SQLJ iterator from a JDBC ResultSet object.	32
4.18.2	Obtaining a JDBC ResultSet object from an SQLJ iterator object.	33
4.18.3	Obtaining a JDBC ResultSet object from an untyped iterator object.	33
4.18.4	Iterator and JDBC ResultSet resource management.	33
4.19	Multi-threading considerations.	34
4.20	User-defined data types.	34
4.21	Batch updates.	35
4.21.1	Batchable statements and batch compatibility.	35
4.21.2	Statement batching API.	36
4.21.3	Execution status and update counts.	36
4.21.4	Program semantics and exceptions.	37
4.21.5	Batch cancellation and disabling.	38
4.21.6	Specification of a batching limit.	38
4.22	SQLJ language elements.	39
4.22.1	<cursor name>.	39
4.22.2	SQL schema, data, and transaction statements.	40
4.22.3	<SQL dynamic statement>.	40
4.22.4	<SQL connection statement>.	40
4.22.5	<host variable definition>.	40
4.22.6	<embedded exception declaration>.	41
4.22.7	<SQL diagnostics statement>.	41
4.22.8	Cursor declaration.	41
4.22.9	Input parameters to SQL statements.	42
4.22.10	Extracting column values from SQLJ iterators.	42
4.22.11	<open statement> and cursors.	42
5	Lexical elements.	43
5.1	<SQL terminal character>.	43
5.2	<token> and <separator>.	44
6	Scalar expressions.	45
6.1	<value specification> and <target specification>.	45
7	Additional common elements.	47
7.1	<routine invocation>.	47
8	Embedded SQL.	49
8.1	<embedded SQL host program>.	49
8.2	<embedded SQL Java program>.	52
9	Binary portability.	53
9.1	Components of binary portable applications.	53

9.2	Naming runtime components.....	54
9.3	Binary portability requirements.....	54
9.4	Profile overview.....	55
9.5	Profile generation and naming.....	56
9.5.1	Example.....	57
9.6	SQLJ application packaging.....	57
9.6.1	Example.....	58
9.7	Profile customization overview.....	58
9.7.1	Profile customization process.....	59
9.7.2	Profile customization utilities.....	59
9.7.3	Profile customizer interface.....	60
9.8	Customization interface.....	60
9.8.1	Customization usage.....	61
9.8.2	Customization registration.....	62
9.9	EntryInfo overview.....	63
9.10	TypeInfo overview.....	65
9.11	SQLJ datatype properties.....	67
10	SQLJ grammar constructs.....	71
10.1	SQLJ reserved names.....	71
10.1.1	Temporary variable names.....	71
10.1.2	Class and resource file names.....	71
10.1.2.1	Internal classes.....	72
10.1.2.2	Resource files and profiles.....	72
10.2	Common subelements.....	73
10.2.1	<modifiers>.....	73
10.2.2	<java class name>.....	73
10.2.3	<java id>.....	74
10.2.4	<java datatype>.....	75
10.2.5	<java constant expression>.....	75
10.2.6	<embedded Java expression>.....	76
10.2.7	<implements clause>.....	78
10.2.8	<declaration with clause>.....	79
11	<SQLJ specific clause> and contents.....	83
11.1	<SQLJ specific clause>.....	83
11.2	<connection declaration clause>.....	84
11.3	Generated connection class.....	85
11.4	<iterator declaration clause>.....	90
11.5	<positioned iterator>.....	92
11.6	Generated positioned iterator class.....	93
11.7	<named iterator>.....	96
11.8	Generated named iterator class.....	97
11.9	<executable clause>.....	99

11.10	<context clause>.....	106
11.11	<statement clause>.....	108
11.12	<delete statement: positioned>.....	110
11.13	<update statement: positioned>.....	112
11.14	<select statement: single row>.....	114
11.15	<fetch statement>.....	117
11.16	<assignment statement>.....	122
11.17	<savepoint statement>.....	124
11.18	<release savepoint statement>.....	125
11.19	<commit statement>.....	126
11.20	<rollback statement>.....	127
11.21	<set transaction statement>.....	128
11.22	<call statement>.....	129
11.23	<assignment clause>.....	131
11.24	<query clause>.....	133
11.25	<function clause>.....	139
11.26	<iterator conversion clause>.....	142
11.27	<compound statement>.....	145
12	Package sqlj.runtime.....	147
12.1	SQLJ runtime interfaces.....	147
12.1.1	sqlj.runtime.ConnectionContext.....	147
12.1.1.1	Variables.....	148
12.1.1.1.1	CLOSE_CONNECTION.....	148
12.1.1.1.2	KEEP_CONNECTION.....	149
12.1.1.2	Methods.....	149
12.1.1.2.1	close ()......	149
12.1.1.2.2	close (boolean).....	149
12.1.1.2.3	getConnectionProfile (Object).....	150
12.1.1.2.4	getConnection ()......	151
12.1.1.2.5	getExecutionContext ()......	151
12.1.1.2.6	getTypeMap ()......	151
12.1.1.2.7	isClosed ()......	152
12.1.2	sqlj.runtime.FetchUpdate.....	152
12.1.2.1	Methods.....	152
12.1.2.1.1	getCursorName ()......	152
12.1.3	sqlj.runtime.NamedIterator.....	153
12.1.4	sqlj.runtime.PositionedIterator.....	153
12.1.4.1	Methods.....	154
12.1.4.1.1	endFetch ()......	154
12.1.5	sqlj.runtime.ResultSetIterator.....	154
12.1.5.1	Variables.....	154
12.1.5.1.1	ASENSITIVE.....	154

12.1.5.1.2	FETCH_FORWARD.....	155
12.1.5.1.3	FETCH_REVERSE.....	155
12.1.5.1.4	FETCH_UNKNOWN.....	155
12.1.5.1.5	INSENSITIVE.....	155
12.1.5.1.6	SENSITIVE.....	155
12.1.5.2	Methods.....	155
12.1.5.2.1	clearWarnings ()......	156
12.1.5.2.2	close ()......	156
12.1.5.2.3	getFetchSize ()......	156
12.1.5.2.4	getResultSet ()......	157
12.1.5.2.5	getRow ()......	157
12.1.5.2.6	getSensitivity ()......	158
12.1.5.2.7	getWarnings ()......	158
12.1.5.2.8	isClosed ()......	159
12.1.5.2.9	next ()......	159
12.1.5.2.10	setFetchSize (int).....	160
12.1.6	sqlj.runtime.Scrollable.....	161
12.1.6.1	Variables.....	161
12.1.6.2	Methods.....	161
12.1.6.2.1	absolute (int).....	161
12.1.6.2.2	afterLast ()......	162
12.1.6.2.3	beforeFirst ()......	162
12.1.6.2.4	first ()......	162
12.1.6.2.5	getFetchDirection ()......	162
12.1.6.2.6	isAfterLast ()......	163
12.1.6.2.7	isBeforeFirst ()......	163
12.1.6.2.8	isFirst ()......	164
12.1.6.2.9	isLast ()......	164
12.1.6.2.10	last ()......	164
12.1.6.2.11	previous ()......	165
12.1.6.2.12	relative (int).....	165
12.1.6.2.13	setFetchDirection (int).....	166
12.2	SQLJ Runtime Classes.....	167
12.2.1	sqlj.runtime.AsciiStream.....	167
12.2.1.1	Constructors.....	167
12.2.1.1.1	AsciiStream (InputStream).....	167
12.2.1.1.2	AsciiStream (InputStream, int).....	168
12.2.2	sqlj.runtime.BinaryStream.....	168
12.2.2.1	Constructors.....	169
12.2.2.1.1	BinaryStream (InputStream).....	169
12.2.2.1.2	BinaryStream (InputStream, int).....	169
12.2.3	sqlj.runtime.DefaultRuntime.....	169

12.2.3.1	Constructors.....	170
12.2.3.1.1	DefaultRuntime ()......	170
12.2.3.2	Methods.....	170
12.2.3.2.1	getDefaultConnection ()......	170
12.2.3.2.2	getLoaderForClass (Class).....	170
12.2.4	sqlj.runtime.ExecutionContext.....	171
12.2.4.1	Variables.....	172
12.2.4.1.1	ADD_BATCH_COUNT.....	172
12.2.4.1.2	AUTO_BATCH.....	172
12.2.4.1.3	EXEC_BATCH_COUNT.....	172
12.2.4.1.4	EXCEPTION_COUNT.....	173
12.2.4.1.5	NEW_BATCH_COUNT.....	173
12.2.4.1.6	QUERY_COUNT.....	173
12.2.4.1.7	UNLIMITED_BATCH.....	174
12.2.4.2	Constructors.....	174
12.2.4.2.1	ExecutionContext ()......	174
12.2.4.3	Methods.....	174
12.2.4.3.1	cancel ()......	174
12.2.4.3.2	execute ()......	175
12.2.4.3.3	executeBatch ()......	176
12.2.4.3.4	executeQuery ()......	177
12.2.4.3.5	executeUpdate ()......	177
12.2.4.3.6	getBatchLimit ()......	178
12.2.4.3.7	getBatchUpdateCounts ()......	179
12.2.4.3.8	getFetchDirection ()......	179
12.2.4.3.9	getFetchSize ()......	179
12.2.4.3.10	getMaxFieldSize ()......	180
12.2.4.3.11	getMaxRows ()......	180
12.2.4.3.12	getNextResultSet ()......	181
12.2.4.3.13	getNextResultSet (int).....	181
12.2.4.3.14	getQueryTimeout ()......	182
12.2.4.3.15	getUpdateCount ()......	183
12.2.4.3.16	getWarnings ()......	183
12.2.4.3.17	isBatching ()......	183
12.2.4.3.18	registerStatement (ExecutionContext, Object, int).....	184
12.2.4.3.19	releaseStatement ()......	185
12.2.4.3.20	setBatching (boolean).....	186
12.2.4.3.21	setBatchLimit (int).....	186
12.2.4.3.22	setFetchDirection (int).....	187
12.2.4.3.23	setFetchSize (int).....	187
12.2.4.3.24	setMaxFieldSize (int).....	188
12.2.4.3.25	setMaxRows (int).....	188

12.2.4.3.26	setQueryTimeout (int).....	189
12.2.5	sqlj.runtime.RuntimeContext.....	189
12.2.5.1	Variables.....	189
12.2.5.1.1	DEFAULT_DATA_SOURCE.....	189
12.2.5.1.2	DEFAULT_RUNTIME.....	190
12.2.5.1.3	PROPERTY_KEY.....	190
12.2.5.2	Constructors.....	190
12.2.5.2.1	RuntimeContext ()......	190
12.2.5.3	Methods.....	190
12.2.5.3.1	getDefaultConnection ()......	190
12.2.5.3.2	getLoaderForClass (Class).....	191
12.2.5.3.3	getRuntime ()......	191
12.2.6	sqlj.runtime.StreamWrapper.....	192
12.2.6.1	Constructors.....	192
12.2.6.1.1	StreamWrapper (InputStream).....	192
12.2.6.1.2	StreamWrapper (InputStream, int).....	193
12.2.6.2	Methods.....	193
12.2.6.2.1	getInputStream ()......	193
12.2.6.2.2	getLength ()......	193
12.2.6.2.3	setLength (int).....	194
12.2.7	sqlj.runtime.UnicodeStream.....	194
12.2.7.1	Constructors.....	195
12.2.7.1.1	UnicodeStream (InputStream).....	195
12.2.7.1.2	UnicodeStream (InputStream, int).....	195
12.2.8	sqlj.runtime.CharacterStream.....	195
12.2.8.1	Constructors.....	196
12.2.8.1.1	CharacterStream (Reader).....	196
12.2.8.1.2	CharacterStream (Reader, int).....	196
12.2.8.2	Methods.....	196
12.2.8.2.1	getReader ()......	196
12.2.8.2.2	getLength ()......	197
12.2.8.2.3	setLength (int).....	197
12.2.9	sqlj.runtime.SQLNullException.....	197
12.2.9.1	Constructors.....	198
12.2.9.1.1	SQLException ()......	198
13	Package sqlj.runtime.profile.....	199
13.1	SQLJ sqlj.runtime.profile Interfaces.....	199
13.1.1	sqlj.runtime.profile.BatchContext.....	199
13.1.1.1	Methods.....	199
13.1.1.1.1	clearBatch ()......	199
13.1.1.1.2	executeBatch ()......	199
13.1.1.1.3	setBatchLimit (int).....	200

13.1.2	sqlj.runtime.profile.ConnectedProfile	200
13.1.2.1	Methods	202
13.1.2.1.1	close ()	202
13.1.2.1.2	getConnection ()	202
13.1.2.1.3	getProfileData ()	202
13.1.2.1.4	getStatement (int, Map)	203
13.1.2.1.5	getStatement (int, BatchContext, Map)	203
13.1.3	sqlj.runtime.profile.Customization	204
13.1.3.1	Methods	205
13.1.3.1.1	acceptsConnection (Connection)	205
13.1.3.1.2	getProfile (Connection, Profile)	205
13.1.4	sqlj.runtime.profile.Loader	206
13.1.4.1	Methods	206
13.1.4.1.1	getResourceAsStream (String)	206
13.1.4.1.2	loadClass (String)	207
13.1.5	sqlj.runtime.profile.RTResultSet	207
13.1.5.1	Methods	211
13.1.5.1.1	clearWarnings ()	211
13.1.5.1.2	close ()	211
13.1.5.1.3	findColumn (String)	211
13.1.5.1.4	getArray (int)	212
13.1.5.1.5	getAsciiStreamWrapper (int)	213
13.1.5.1.6	getBigDecimal (int)	214
13.1.5.1.7	getBinaryStreamWrapper (int)	214
13.1.5.1.8	getBlob (int)	215
13.1.5.1.9	getBooleanNotNull (int)	216
13.1.5.1.10	getBooleanWrapper (int)	217
13.1.5.1.11	getByteNotNull (int)	218
13.1.5.1.12	getBytes (int)	219
13.1.5.1.13	getByteWrapper (int)	219
13.1.5.1.14	getCharacterStreamWrapper (int)	220
13.1.5.1.15	getClob (int)	221
13.1.5.1.16	getColumnCount ()	222
13.1.5.1.17	getCursorName ()	222
13.1.5.1.18	getDate (int)	223
13.1.5.1.19	getDoubleNotNull (int)	224
13.1.5.1.20	getDoubleWrapper (int)	225
13.1.5.1.21	getFloatNotNull (int)	225
13.1.5.1.22	getFloatWrapper (int)	226
13.1.5.1.23	getIntNotNull (int)	227
13.1.5.1.24	getIntWrapper (int)	228
13.1.5.1.25	getJDBCResultSet ()	229

13.1.5.1.26	getLongNotNull (int).....	229
13.1.5.1.27	getLongWrapper (int).....	230
13.1.5.1.28	getObject (int, Class).....	231
13.1.5.1.29	getRef (int).....	232
13.1.5.1.30	getShortNotNull (int).....	233
13.1.5.1.31	getShortWrapper (int).....	234
13.1.5.1.32	getString (int).....	234
13.1.5.1.33	getTime (int).....	235
13.1.5.1.34	getTimestamp (int).....	236
13.1.5.1.35	getUnicodeStreamWrapper (int).....	237
13.1.5.1.36	getURL (int).....	238
13.1.5.1.37	getWarnings ()......	238
13.1.5.1.38	isClosed ()......	239
13.1.5.1.39	isValidRow ()......	239
13.1.5.1.40	next ()......	240
13.1.6	sqlj.runtime.profile.RTStatement.....	240
13.1.6.1	Methods.....	245
13.1.6.1.1	cancel ()......	245
13.1.6.1.2	clearWarnings ()......	246
13.1.6.1.3	execute ()......	246
13.1.6.1.4	executeComplete ()......	246
13.1.6.1.5	executeRTQuery ()......	247
13.1.6.1.6	executeUpdate ()......	248
13.1.6.1.7	getArray (int).....	248
13.1.6.1.8	getBatchContext ()......	249
13.1.6.1.9	getBigDecimal (int).....	250
13.1.6.1.10	getBlob (int).....	250
13.1.6.1.11	getBooleanNotNull (int).....	251
13.1.6.1.12	getBooleanWrapper (int).....	252
13.1.6.1.13	getByteNotNull (int).....	253
13.1.6.1.14	getBytes (int).....	254
13.1.6.1.15	getByteWrapper (int).....	254
13.1.6.1.16	getClob (int).....	255
13.1.6.1.17	getDate (int).....	256
13.1.6.1.18	getDoubleNotNull (int).....	257
13.1.6.1.19	getDoubleWrapper (int).....	257
13.1.6.1.20	getFloatNotNull (int).....	258
13.1.6.1.21	getFloatWrapper (int).....	259
13.1.6.1.22	getIntNotNull (int).....	260
13.1.6.1.23	getIntWrapper (int).....	261
13.1.6.1.24	getJDBCCallableStatement ()......	261
13.1.6.1.25	getJDBCPreparedStatement ()......	262

13.1.6.1.26	getLongNotNull (int).....	262
13.1.6.1.27	getLongWrapper (int).....	263
13.1.6.1.28	getMaxFieldSize ()......	264
13.1.6.1.29	getMaxRows ()......	264
13.1.6.1.30	getMoreResults (int).....	265
13.1.6.1.31	getObject (int, Class).....	265
13.1.6.1.32	getQueryTimeout ()......	267
13.1.6.1.33	getRef (int).....	267
13.1.6.1.34	getResultSet ()......	268
13.1.6.1.35	getShortNotNull (int).....	268
13.1.6.1.36	getShortWrapper (int).....	269
13.1.6.1.37	getString (int).....	270
13.1.6.1.38	getTime (int).....	271
13.1.6.1.39	getTimestamp (int).....	272
13.1.6.1.40	getUpdateCount ()......	272
13.1.6.1.41	getURL ()......	273
13.1.6.1.42	getWarnings ()......	274
13.1.6.1.43	isBatchable ()......	274
13.1.6.1.44	isBatchCompatible ()......	275
13.1.6.1.45	getArray (int, Array).....	276
13.1.6.1.46	setAsciiStreamWrapper (int, AsciiStream).....	277
13.1.6.1.47	setBigDecimal (int, BigDecimal).....	277
13.1.6.1.48	setBinaryStreamWrapper (int, BinaryStream).....	278
13.1.6.1.49	setBlob (int, Blob).....	279
13.1.6.1.50	setBoolean (int, boolean).....	279
13.1.6.1.51	setBooleanWrapper (int, Boolean).....	280
13.1.6.1.52	setByte (int, byte).....	281
13.1.6.1.53	setBytes (int, byte).....	282
13.1.6.1.54	setByteWrapper (int, Byte).....	282
13.1.6.1.55	setCharacterStreamWrapper (int, CharacterStream).....	283
13.1.6.1.56	setClob (int, Clob).....	284
13.1.6.1.57	setDate (int, Date).....	284
13.1.6.1.58	setDouble (int, double).....	285
13.1.6.1.59	setDoubleWrapper (int, Double).....	286
13.1.6.1.60	setFloat (int, float).....	287
13.1.6.1.61	setFloatWrapper (int, Float).....	287
13.1.6.1.62	setInt (int, int).....	288
13.1.6.1.63	setIntWrapper (int, Integer).....	289
13.1.6.1.64	setLong (int, long).....	289
13.1.6.1.65	setLongWrapper (int, Long).....	290
13.1.6.1.66	setMaxFieldSize (int).....	291
13.1.6.1.67	setMaxRows (int).....	291

13.1.6.1.68	setObject ()	292
13.1.6.1.69	setQueryTimeout (int)	292
13.1.6.1.70	setRef (int, Ref)	293
13.1.6.1.71	setShort (int, short)	294
13.1.6.1.72	setShortWrapper (int, Short)	294
13.1.6.1.73	setString (int, String)	295
13.1.6.1.74	setTime (int, Time)	296
13.1.6.1.75	setTimestamp (int, Timestamp)	296
13.1.6.1.76	setUnicodeStreamWrapper (int, UnicodeStream)	297
13.1.6.1.77	setURL (int, URL)	298
13.1.7	sqlj.runtime.profile.SerializedProfile	298
13.1.7.1	Methods	299
13.1.7.1.1	getProfileAsStream ()	299
13.2	SQLJ sqlj.runtime.profile Classes	300
13.2.1	sqlj.runtime.profile.DefaultLoader	300
13.2.1.1	Constructors	300
13.2.1.1.1	DefaultLoader (ClassLoader)	300
13.2.1.2	Methods	300
13.2.1.2.1	getResourceAsStream (String)	300
13.2.1.2.2	loadClass (String)	301
13.2.2	sqlj.runtime.profile.EntryInfo	302
13.2.2.1	Variables	302
13.2.2.1.1	BLOCK	302
13.2.2.1.2	CALL	302
13.2.2.1.3	CALLABLE_STATEMENT	303
13.2.2.1.4	COMMIT	303
13.2.2.1.5	EXECUTE	303
13.2.2.1.6	EXECUTE_QUERY	304
13.2.2.1.7	EXECUTE_UPDATE	304
13.2.2.1.8	ITERATOR_CONVERSION	304
13.2.2.1.9	NAMED_RESULT	305
13.2.2.1.10	NO_RESULT	305
13.2.2.1.11	OTHER	305
13.2.2.1.12	POSITIONED	306
13.2.2.1.13	POSITIONED_RESULT	306
13.2.2.1.14	PREPARED_STATEMENT	306
13.2.2.1.15	QUERY	307
13.2.2.1.16	QUERY_FOR_UPDATE	307
13.2.2.1.17	RELEASE_SAVEPOINT	307
13.2.2.1.18	ROLLBACK	308
13.2.2.1.19	SAVEPOINT	308
13.2.2.1.20	SET_TRANSACTION	308

13.2.2.1.21	SINGLE_ROW_QUERY.....	309
13.2.2.1.22	STATEMENT.....	309
13.2.2.1.23	UNTYPED_SELECT.....	309
13.2.2.1.24	VALUES.....	309
13.2.2.2	Constructors.....	310
13.2.2.2.1	EntryInfo ()......	310
13.2.2.3	Methods.....	310
13.2.2.3.1	executeTypeToString (int).....	310
13.2.2.3.2	getDescriptor ()......	310
13.2.2.3.3	getExecuteType ()......	311
13.2.2.3.4	getLineNumber ()......	312
13.2.2.3.5	getParamCount ()......	312
13.2.2.3.6	getParamInfo (int).....	312
13.2.2.3.7	getResultSetCount ()......	313
13.2.2.3.8	getResultSetInfo (int).....	313
13.2.2.3.9	getResultSetName ()......	314
13.2.2.3.10	getResultSetType ()......	314
13.2.2.3.11	getRole ()......	315
13.2.2.3.12	getSQLString ()......	316
13.2.2.3.13	getStatementType ()......	316
13.2.2.3.14	getTransactionDescriptor ()......	316
13.2.2.3.15	isDefinedRole (int).....	317
13.2.2.3.16	isValidDescriptor (Object, int).....	317
13.2.2.3.17	isValidExecuteType (int).....	318
13.2.2.3.18	isValidResultSetType (int).....	318
13.2.2.3.19	isValidRole (int).....	319
13.2.2.3.20	isValidStatementType (int).....	319
13.2.2.3.21	resultSetTypeToString (int).....	320
13.2.2.3.22	roleToString (int).....	320
13.2.2.3.23	statementTypeToString (int).....	321
13.2.2.3.24	validateObject ()......	321
13.2.3	sqlj.runtime.profile.Profile.....	321
13.2.3.1	Constructors.....	322
13.2.3.1.1	Profile (Loader).....	322
13.2.3.2	Methods.....	323
13.2.3.2.1	deregisterCustomization (Customization).....	323
13.2.3.2.2	getConnectedProfile (Connection).....	323
13.2.3.2.3	getContextName ()......	324
13.2.3.2.4	getCustomizations ()......	324
13.2.3.2.5	getJavaType (String).....	324
13.2.3.2.6	getJavaType (TypeInfo).....	325
13.2.3.2.7	getLoader ()......	326

13.2.3.2.8	<code>getProfileData ()</code>	326
13.2.3.2.9	<code>getProfileName ()</code>	326
13.2.3.2.10	<code>getTimestamp ()</code>	326
13.2.3.2.11	<code>instantiate (Loader, InputStream)</code>	327
13.2.3.2.12	<code>instantiate (Loader, String)</code>	327
13.2.3.2.13	<code>registerCustomization (Customization)</code>	329
13.2.3.2.14	<code>registerCustomization (Customization, Customization)</code>	329
13.2.3.2.15	<code>replaceCustomization (Customization, Customization)</code>	330
13.2.4	<code>sqlj.runtime.profile.ProfileData</code>	330
13.2.4.1	Constructors.....	331
13.2.4.1.1	<code>ProfileData ()</code>	331
13.2.4.2	Methods.....	331
13.2.4.2.1	<code>getEntryInfo (int)</code>	331
13.2.4.2.2	<code>getProfile ()</code>	331
13.2.4.2.3	<code>getSourceFile ()</code>	332
13.2.4.2.4	<code>size ()</code>	332
13.2.5	<code>sqlj.runtime.profile.SetTransactionDescriptor</code>	332
13.2.5.1	Variables.....	333
13.2.5.1.1	<code>READ_NONE</code>	333
13.2.5.1.2	<code>READ_ONLY</code>	333
13.2.5.1.3	<code>READ_WRITE</code>	333
13.2.5.2	Constructors.....	334
13.2.5.2.1	<code>SetTransactionDescriptor (int, int)</code>	334
13.2.5.3	Methods.....	334
13.2.5.3.1	<code>getAccessMode ()</code>	334
13.2.5.3.2	<code>getIsolationLevel ()</code>	335
13.2.6	<code>sqlj.runtime.profile.TypeInfo</code>	335
13.2.6.1	Variables.....	335
13.2.6.1.1	<code>IN</code>	335
13.2.6.1.2	<code>INOUT</code>	336
13.2.6.1.3	<code>OUT</code>	336
13.2.6.2	Constructors.....	336
13.2.6.2.1	<code>TypeInfo ()</code>	336
13.2.6.3	Methods.....	337
13.2.6.3.1	<code>getJavaTypeName ()</code>	337
13.2.6.3.2	<code>getMarkerIndex ()</code>	337
13.2.6.3.3	<code>getMode ()</code>	338
13.2.6.3.4	<code>getName ()</code>	338
13.2.6.3.5	<code>getSQLType ()</code>	339
13.2.6.3.6	<code>getSQLTypeName ()</code>	339
13.2.6.3.7	<code>isValidMode (int)</code>	340
13.2.6.3.8	<code>isValidSQLType (int)</code>	340

13.2.6.3.9	modeToString (int).....	341
13.2.6.3.10	SQLTypeToString (int).....	341
13.2.6.3.11	validateObject ()......	342
14	sqlj.runtime.profile.util.ProfileCustomizer.....	343
14.1	Methods.....	345
14.1.1	acceptsConnection (Connection).....	345
14.1.2	customize (Profile, Connection, ErrorLog).....	345
15	Status codes.....	347
15.1	SQLSTATE.....	347
16	Conformance.....	349
16.1	Claims of conformance to SQL/OLB.....	349
16.2	Additional conformance requirements for SQL/OLB.....	349
16.3	Implied feature relationships of SQL/OLB.....	349
Annex A	SQL Conformance Summary.....	351
Annex B	Implementation-defined elements.....	353
Annex C	Implementation-dependent elements.....	357
Annex D	SQL feature taxonomy.....	361
Annex E	SQLJ tutorial.....	363
E.1	Design goals.....	363
E.2	Advantages of SQLJ over JDBC.....	363
E.3	Consistency with existing embedded SQL languages.....	364
E.4	Examples.....	365
E.4.1	Host variables.....	365
E.4.2	Host expressions.....	365
E.4.3	SQLJ clauses.....	366
E.4.4	Database connection context.....	366
E.4.5	Default connection context.....	366
E.4.6	Iterators.....	367
E.4.6.1	Positional bindings to columns.....	367
E.4.6.2	Named bindings to columns.....	368
E.4.6.3	Providing names for columns of queries.....	369
E.4.7	Calls to stored procedures and functions.....	369
E.4.8	Using multiple SQLJ contexts and connections.....	370
E.4.9	SQL execution control and status.....	371
E.4.10	Multiple JDBC ResultSets from stored procedure calls.....	372
E.4.11	Creating an SQLJ iterator object from a JDBC ResultSet object.....	372
E.4.12	Obtaining a JDBC ResultSet object from an untyped iterator object.....	373
E.4.13	Working with user-defined types.....	373
E.4.14	Example program.....	375
E.4.15	Host variable definition.....	376

Index.....377

TABLES

Page

1	SQLJ output assignability.	20
2	SQLJ input assignability.	26
3	Association of roles with SQLJ <executable clause>s.	63
4	SQLJ type properties.	67
5	Methods retained from java.sql.ResultSet.	208
6	Methods not retained from java.sql.ResultSet.	209
7	Additional methods unique to RTResultSet.	210
8	Methods retained from java.sql.Statement.	241
9	Methods not retained from java.sql.Statement.	242
10	Methods retained from java.sql.PreparedStatement.	242
11	Methods not retained from java.sql.PreparedStatement.	243
12	Methods retained from java.sql.CallableStatement.	244
13	Methods not retained from java.sql.CallableStatement.	244
14	Additional methods unique to RTStatement.	245
15	Customize Result Interpretation.	344
16	SQLSTATE class and subclass values.	347
17	Implied feature relationships of SQL/OLB.	349
18	Feature taxonomy for optional features.	361

INTRODUCTION

The organization of this Part of this International Standard is as follows:

- 1) **Clause 1, “Scope”**, specifies the scope of this part of ISO/IEC 9075.
- 2) **Clause 2, “Normative references”**, identifies additional standards and publically-available specifications that, through reference in this part of ISO/IEC 9075, constitute provisions of this part of ISO/IEC 9075.
- 3) **Clause 3, “Definitions, notations, and conventions”**, defines the notations and conventions used in this part of ISO/IEC 9075.
- 4) **Clause 4, “Concepts”**, presents concepts used in the definition of the Object Language Bindings.
- 5) **Clause 5, “Lexical elements”**, defines the lexical elements of the language.
- 6) **Clause 6, “Scalar expressions”**, defines the elements of the language that produce scalar values.
- 7) **Clause 7, “Additional common elements”**, defines additional language elements that are used in various parts of the language.
- 8) **Clause 8, “Embedded SQL”**, defines the host language embeddings.
- 9) **Clause 9, “Binary portability”**, provides an overview of the binary portability and profile customization requirements for SQLJ.
- 10) **Clause 10, “SQLJ grammar constructs”**, defines the BNF conventions, common subelements, and datatype mappings for SQLJ.
- 11) **Clause 11, “<SQLJ specific clause> and contents”**, defines the syntax and rules for SQLJ constructs.
- 12) **Clause 12, “Package sqlj.runtime”**, specifies the SQLJ runtime package.
- 13) **Clause 13, “Package sqlj.runtime.profile”**, specifies the SQLJ runtime profile package.
- 14) **Clause 14, “sqlj.runtime.profile.util.ProfileCustomizer”**, specifies the SQLJ profile customizer class.
- 15) **Clause 15, “Status codes”**, defines SQLSTATE values related to Object Language Bindings.
- 16) **Clause 16, “Conformance”**, defines the criteria for conformance to this part of ISO/IEC 9075.
- 17) **Annex A, “SQL Conformance Summary”**, is an informative Annex. It summarizes the conformance requirements of the SQL language.
- 18) **Annex B, “Implementation-defined elements”**, is an informative Annex. It lists those features for which the body of this part of ISO/IEC 9075 states that the syntax, the meaning, the returned results, the effect on SQL-data and/or schemas, or any other behavior is partly or wholly implementation-defined.
- 19) **Annex C, “Implementation-dependent elements”**, is an informative Annex. It lists those features for which the body of this part of ISO/IEC 9075 states that the syntax, the meaning, the returned results, the effect on SQL-data and/or schemas, or any other behavior is partly or wholly implementation-dependent.

- 20) [Annex D, “SQL feature taxonomy”](#), is an informative Annex. It identifies features of the SQL language specified in this part of ISO/IEC 9075 by a numeric identifier and a short descriptive name. This taxonomy is used to specify conformance and might be used to develop other profiles involving the SQL language.
- 21) [Annex E, “SQLJ tutorial”](#), is an informative Annex. It contains tutorial information about the features of the SQL language that are specified in this part of ISO/IEC 9075.

In the text of this part of ISO/IEC 9075, Clauses begin a new odd-numbered page. Any resulting blank space is not significant.

All Clauses of this part of ISO/IEC 9075 are normative.

AUSTRALIAN STANDARD

Information technology — Database languages — SQL —

Part 10: Object Language Bindings (SQL/OLB)

1 Scope

This part of ISO/IEC 9075 defines extensions of Database language SQL to support embedding of SQL statements into programs written in the Java™ programming language (Java is a registered trademark of Sun Microsystems, Inc.). The embedding of SQL into Java is commonly known as “SQLJ”. This part of ISO/IEC 9075 specifies the syntax and semantics of SQLJ, as well as mechanisms to ensure binary portability of resulting SQLJ applications. In addition, it specifies a number of Java packages and their contained classes (including methods).

Throughout this part of ISO/IEC 9075, the terms "SQLJ" and "SQL/OLB" are used synonymously.