
**Information technology — Open Systems
Interconnection — Remote Database
Access —**

**Part 2:
SQL Specialization**

*Technologies de l'information — Interconnexion de systèmes ouverts
(OSI) — Accès aux bases de données à distance —*

Partie 2: Spécialisation SQL

Contents

1.	GENERAL.....	1
1.1	Scope	1
1.2	Normative references.....	1
1.3	Compatibility	2
1.3.1	SQL Standards Supported.....	2
1.3.2	Upward Compatibility.....	2
1.4	Definitions	2
1.4.1	Terms defined in ISO/IEC 9075 (Database Language SQL).....	2
1.4.2	Terms defined in this part of ISO/IEC 9579	4
1.4.2.1	SQL database resource	4
1.4.2.2	RDA SQL statement.....	4
1.5	Abbreviations.....	5
1.6	Conventions.....	5
2.	MODEL.....	6
2.1	The RDA SQL Specialization Service model	6
2.1.1	Mapping to the general model of the RDA Service.....	6
2.1.2	Mapping to the Concepts of Database Language SQL	6
3.	SERVICE.....	7
3.1	The RDA SQL Specialization Service	7
3.1.1	RDA Dialogue Management services	7
3.1.1.1	RDA Dialogue Initialization functional unit.....	7
3.1.1.1.1	R-Initialize Service	7
3.1.1.1.2	RDA Dialogue Termination functional unit	7
3.1.1.1.2.1	R-Terminate Service.....	7
3.1.2	RDA Transaction Management services.....	8
3.1.2.1	RDA Transaction Management functional unit	8
3.1.2.1.1	R-BeginTransaction Service.....	8
3.1.2.1.2	R-Commit Service.....	8
3.1.2.1.3	R-Rollback Service.....	8
3.1.3	RDA Control Services.....	8
3.1.3.1	Cancel functional unit	8
3.1.3.1.1	R-Cancel Service	8
3.1.3.2	Status functional unit.....	8
3.1.3.2.1	R-Status Service.....	8
3.1.4	Resource Handling services	8
3.1.4.1	Resource Handling functional unit.....	8
3.1.4.1.1	R-Open Service.....	8
3.1.4.1.2	R-Close Service	11
3.1.5	Database Language Services.....	11
3.1.5.1	Immediate Execution DBL functional unit	11
3.1.5.1.1	R-ExecuteDBL Service.....	11
3.1.5.2	Stored Execution DBL functional unit.....	14
3.1.5.2.1	R-DefinedBL Service.....	14

© ISO/IEC 1998

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

3.1.5.2.2 R-InvokeDBL Service.....	15
3.1.5.2.3 R-DropDBLService:	17
3.1.6 SQL statements, arguments and results:	17
3.1.6.1 sQLDBLStatement:	17
3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification.....	18
3.1.6.3 sQLDBLArgumentValues and sQLDBLResultValues.....	23
3.1.6.4 Referencing to parameters.....	24
3.2 Sequencing rules.....	24
3.2.1.1 RDA client sequencing rules.....	24
3.2.2 RDA Server Sequencing Rules	25
4. PROTOCOL	26
4.1 The RDA SQL Specialization client execution rules.....	26
4.1.1 RDA SQL Specialization dialogue-state model.....	26
4.1.1.1 SQL Transaction Entity.....	26
4.1.2 General client execution rules.....	26
4.1.2.1 Generation of the SQL transaction entity.....	26
4.1.2.2 Execution of an SQL-statement	26
4.1.3 <set transaction statement>	26
4.1.4 <connect statement>	27
4.1.5 <set connection statement>	27
4.1.6 <commit statement>.....	27
4.1.7 <rollback statement>.....	27
4.2 The RDA SQL Specialization server execution rules.....	27
4.2.1 RDA SQL Specialization dialogue-state model.....	28
4.2.1.1 RDA dialogue entity.....	28
4.2.1.2 Opened data resource entity	28
4.2.1.3 Defined DBL Entity	28
4.2.1.4 RDA operation entity	29
4.2.2 General server execution rules.....	29
4.2.3 RDA Dialogue Management services.....	29
4.2.3.1 RDA Dialogue Initialization functional unit.....	29
4.2.3.1.1 R-Initialize Service	29
4.2.3.2 RDA Dialogue Termination functional unit	29
4.2.3.2.1 R-Terminate Service	29
4.2.4 RDA Transaction Management services.....	29
4.2.4.1 RDA Transaction Management functional unit	29
4.2.4.1.1 R-Begin Transaction Service	29
4.2.4.1.2 R-Commit Service	29
4.2.4.1.3 R-Rollback Service	29
4.2.5 RDA Control services	30
4.2.5.1 Cancel functional unit	30
4.2.5.1.1 R-Cancel Service	30
4.2.5.2 Status functional unit.....	30
4.2.5.2.1 R-Status Service.....	30
4.2.6 Resource Handling services	30
4.2.6.1 Resource Handling functional unit.....	30
4.2.6.1.1 R-Open Service.....	30
4.2.6.1.2 R-Close Service	31
4.2.7 Database Language services	31
4.2.7.1 Immediate Execution DBL functional unit	31
4.2.7.1.1 R-ExecuteDBL service	31
4.2.7.2 Stored Execution DBL functional unit.....	36
4.2.7.2.1 R-define DBL Service	36
4.2.7.2.2 R-InvokeDBL Service.....	39
4.2.7.2.3 R-DropDBL Service	43

4.3 Structure and Encoding of RDA SQL APDUs.....	43
4.3.1 Abstract Syntax Name.....	43
4.3.2 ASN.1 Module for RDA SQL Specialization ASE	43
4.4 Conformance.....	56
4.4.1 Static conformance.....	56
4.4.2 Dynamic Conformance	57
5. THE RDA SQL SPECIALIZATION APPLICATION-CONTEXTS	58
5.1 The RDA SQL Basic application-context	58
5.1.1 RDA SQL Basic application-context object identifier	58
5.1.2 Additional constraints on the RDA SQL ASE datatypes.....	58
5.1.3 Additional sequencing rules.....	58
5.1.4 Additional conformance requirements.....	58
5.1.4.1 Static conformance	58
5.1.4.2 Dynamic conformance	58
5.2 The RDA SQL TP application-context	58
5.2.1 RDA SQL TP application-context object identifier.....	58
5.2.2 Additional constraints on the RDA SQL ASE datatypes.....	58
5.2.3 Additional sequencing rules.....	58
5.2.4 Additional conformance requirements.....	58
5.2.4.1 Static conformance	58
5.2.4.2 Dynamic conformance	58
Annex A (informative): Encoding of Multiple Rows	59

IECNORM.COM : Click to view the full PDF of ISO/IEC 9579-2:1998

Figures

Figure 1. Usage of Embedded SQL Variables	24
---	----

Tables

Table 1. R-Initialise SQL Specific Service Parameters	7
Table 2. R-Cancel SQL Specific Service Parameters	8
Table 3. R-Status SQL Specific Service Parameters.....	8
Table 4. R-Open SQL Specific Service Parameters.....	9
Table 5. R-Close SQL Specific Service Parameters	11
Table 6. R-ExecuteDBL SQL Specific Service Parameters.....	12
Table 7. R-Define DBL SQL Specific Service Parameters.....	14
Table 8. R-InvokeDBL SQL Specific Service Parameters.....	15
Table 9. sQLDBLStatement	17
Table 10. sQLDBLArgumentSpecification and sQLDBLResultSpecification.....	19
Table 11. sQLDBLArgumentValues and sQLDBLResultValues.....	23
Table 12. R-ExecuteDBL use of SQL argument and result parameters.....	33
Table 13. R-DefinedBL use of SQL argument and result parameters.....	37
Table 14. R-InvokeDBL use of SQL argument and result parameters	40

IECNORM.COM : Click to view the full PDF of ISO/IEC 9579-2:1998

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. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9579-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 21, *Open systems interconnection, data management and open distributed processing*.

This second edition cancels and replaces the first edition (ISO/IEC 9579-2:1993), which has been technically revised. It also incorporates Technical Corrigendum 1:1995.

ISO/IEC 9579 consists of the following parts, under the general title *Information technology — Open Systems Interconnection — Remote Database Access*:

- *Part 1: Generic Model, Service and Protocol*
- *Part 2: SQL Specialization*

Annex A of this part of ISO/IEC 9579 is for information only.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9579-2:1998

Introduction

Remote Database Access (RDA) International Standards are members of a set of International Standards produced to facilitate the interworking of computer systems. The RDA International Standards are positioned in the Application Layer of the Reference Model of Open Systems Interconnection (OSI) and are related to other Open Systems International Standards in the set, as defined in ISO 7498, *OSI - Basic Reference Model*.

The goal of Remote Database Access is to allow, with a minimum of technical agreement outside the interconnection standards, the interconnection of applications and database systems:

- from different manufacturers;
- under different managements;
- of different levels of complexity;
- exploiting different technologies.

An application may itself be a database system and therefore an RDA Specialization standard can be used to support multi-database system interworking.

This part of ISO/IEC 9579 is to be used together with ISO/IEC 9579-1 to provide remote data access to a database management system conforming to ISO/IEC 9075 (Database Language SQL).

This part of ISO/IEC 9579 is provided for implementators of the protocol and service defined in ISO/IEC 9579-2:1993 who wish to upgrade their implementation to support those features of ISO/IEC 9075:1992 which are not supported by ISO/IEC 9579-2:1993. A third edition of this part of ISO/IEC 9579 is in preparation which will provide enhanced support more appropriate for new implementations wishing to support ISO/IEC 9075:1992 in conjunction with ISO/IEC 9075-3:1995 (SQL/CLI).

IECNORM.COM : Click to view the full PDF of ISO/IEC 9579-2:1998

This page intentionally left blank

IECNORM.COM : Click to view the full PDF of ISO/IEC 9579-2:1998

Information technology – Open Systems Interconnection – Remote Database Access –

Part 2:

SQL Specialization

1 GENERAL

1.1 Scope

This part of ISO/IEC 9579 specifies the functionality of a database server within a distributed open systems environment and specifies the communication service and protocol for accessing its capabilities. The communications capabilities are positioned in the Application Layer of the Reference Model for Open Systems Interconnection (OSI).

This part of ISO/IEC 9579 complements ISO/IEC 9579-1 (RDA Generic) in order that the two parts together:

- a) define the capabilities of an RDA SQL database server supporting dialogues with clients.
- b) define a model of dialogues between the RDA SQL database server and remote users.
- c) define a model of a dialogue between an RDA client and an RDA server.
- d) define an abstract service for the RDA SQL ASE, which models the communications facilities supporting interaction between the RDA client and RDA server.
- e) define the RDA SQL ASE protocol to support the RDA SQL Service.
- f) define the characteristics of application-contexts which include the RDA SQL ASE.
- g) define application contexts that support remote database access using SQL:
 - 1) RDA Basic application-context
 - 2) RDA TP application-context

This part of ISO/IEC 9579 does not specify individual implementations or products, nor does it constrain the implementation of entities and interfaces within a computer system.

This part of ISO/IEC 9579 does not define a programmatic interface. The RDA server includes a database capability as defined in ISO/IEC 9075 (Database Language SQL).

Notes:

- 1 The RDA client may contain an SQL application program but there is no requirement that the RDA client shall be an application program written to the ISO/IEC 9075 (Database Language SQL) application program interface.
- 2 ISO/IEC 9075:1989, the former International Standard for Database Language SQL, has been replaced by ISO/IEC 9075:1992. Both the current and former standards contain conformance rules, and the RDA SQL Specialization allows an RDA client to specify the desired level of conformance which it expects the RDA Server to support. Throughout this part of ISO/IEC 9579, "SQL" refers to the language statements permitted by the appropriate standard at the requested level of conformance.

1.2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9579. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9579 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*.

ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.

ISO/IEC 9075:1992, *Information technology – Database languages – SQL*.

ISO/IEC 9075-3:1995, *Information technology – Database languages – SQL – Part 3: Call-Level Interface (SQL/CLI)*.

ISO/IEC 9075-4:1996, *Information technology – Database languages – SQL – Part 4: Persistent Stored Modules*.

ISO/IEC 9579-1:1993, *Information technology – Open Systems Interconnection – Remote Database Access – Part 1: Generic Model, Service and Protocol*.

1.3 Compatibility

1.3.1 SQL Standards Supported

This part of ISO/IEC 9579 is fully compatible with the following editions and parts of the SQL International Standard:

- ISO/IEC 9075:1987 *SQL87*
- ISO/IEC 9075:1989 *SQL89*
- ISO/IEC 9075:1992 *SQL92*
- ISO/IEC 9075-3:1995 *SQL/CLI*
- ISO/IEC 9075-4:1996 *SQL/PSM*

1.3.2 Upward Compatibility

This edition of ISO/IEC 9579-2 is upward compatible with ISO/IEC 9579-2:1993. Where services or protocols are provided by both this edition and ISO/IEC 9579-2:1993, these services or protocols behave in the same manner and provide the same functionality.

SQL application programs which conform to SQL 87, SQL 89 or SQL 92 entry level should be able to correctly execute without change in the following configurations:

- ISO/IEC 9579-2:1993 client and ISO/IEC 9579-2:1993 server - This is the reference configuration against which the other cases are compared.
- ISO/IEC 9579-2:1993 client and ISO/IEC 9579-2:1997 server - The ISO/IEC 9579-2:1997 server behaves as an ISO/IEC 9579-2:1993 server.
- ISO/IEC 9579-2:1997 client and ISO/IEC 9579-2:1993 server - The ISO/IEC 9579-2:1997 client behaves as an ISO/IEC 9579-2:1993 client.
- ISO/IEC 9579-2:1997 client and ISO/IEC 9579-2:1997 server - This is the configuration provided for by this standard.

1.4 Definitions

For the purposes of this part of ISO/IEC 9579, the definitions given in ISO/IEC 9579-1 and the following definitions apply.

1.4.1 Terms defined in ISO/IEC 9075 (Database Language SQL)

This part of ISO/IEC 9579 makes use of the following terms defined in ISO/IEC 9075:1989 (Database Language SQL) or ISO/IEC 9075:1992 (Database Language SQL).

applicable privileges

catalog

cluster of catalogs

Embedded SQL

ISOLATION LEVEL SERIALIZABLE

READ WRITE

SQLCODE

SQLSTATE

SYSTEM_USER

In addition, this part of ISO/IEC 9579 references the following non-terminal production symbols from the SQL syntax, which represent a valid string of characters that can be derived from these production symbols according to the syntax rules in ISO/IEC 9075 (Database Language SQL) at the appropriate level of conformance.

<1987>
<1989>
<1992>
<allocate descriptor statement>
<alter domain statement>
<alter table statement>
<assert definition>
<assignment statement>
<call statement>
<case statement>
<catalog name>
<character set definition>
<close statement>
<collation definition>
<commit statement>
<compound statement>
<condition declaration>
<connect statement>
<cursor name>
<deallocate descriptor statement>
<declare cursor>
<delete statement: positioned>
<delete statement: searched>
<direct implementation-defined statement>
<domain definition>
<drop assertion statement>
<drop character set statement>
<drop collation statement>
<drop domain statement>
<drop module statement>
<drop routine statement>
<drop schema statement>
<drop table statement>
<drop translation statement>
<drop view statement>
<embedded variable name>
<fetch statement>
<for statement>
<get descriptor statement>
<get diagnostics statement>
<grant statement>
<host identifier>
<if statement>
<indicator variable>

<insert statement>
 <level of isolation>
 <loop statement>
 <module authorization identifier>
 <module character set specification>
 <module name>.
 <open statement>
 <repeat statement>
 <resignal statement>
 <revoke statement>
 <rollback statement>
 <routine invocation>
 <schema definition>
 <schema name>
 <select statement: single row>
 <set catalog statement>
 <set connection statement>
 <set constraints mode statement>
 <set descriptor statement>
 <set path statement>
 <set schema statement>
 <set transaction statement>
 <signal statement>
 <SQL conformance>
 <SQL data statement>
 <SQL edition>
 <SQL variable declaration>
 <SQL-invoked routine>
 <SQL-server module definition>
 <table definition>
 <temporary table declaration>
 <transaction access mode>
 <translation definition>
 <update statement: positioned>
 <update statement: searched>
 <variable specification>
 <view definition>
 <while statement>

1.4.2 Terms defined in this part of ISO/IEC 9579

1.4.2.1 SQL database resource

An SQL cluster of catalogs together with the data described by the schemas of the catalogs of that cluster.

1.4.2.2 RDA SQL statement

One of the non-terminals <schema definition>, <table definition>, <view definition>, <drop schema statement>, <alter table statement>, <drop table statement>, <drop view statement>, <domain definition>, <alter domain statement>, <drop domain statement>, <grant statement>, <SQL data statement>, <commit statement>, <rollback statement> and <declare cursor> as defined in ISO/IEC 9075 (Database Language SQL).

1.5 Abbreviations

For the purposes of this part of ISO/IEC 9579, the abbreviations given in ISO/IEC 9075 and ISO/IEC 9579-1 apply.

1.6 Conventions

This part of ISO/IEC 9579 adopts the conventions established in ISO/IEC 9579-1.

The RDA SQL Specialization is formally defined in an ASN.1 Module that is derived from the RDA Specialization module template defined in ISO/IEC 9579-1. The RDA SQL Specialization module provides definitions for those types listed in the template as being undefined by the Generic part of ISO/IEC 9579.

The structure of this part of ISO/IEC 9579 follows the requirements defined in ISO/IEC 9579-1 for defining specializations of the RDA Generic Service and Protocol.

In this part of ISO/IEC 9579, the RDA Generic parameters and data types named `specificXxxx...` and `SpecificXxxx...` are renamed to `sQLXxxx...` and to `SQLXxxx.....`, respectively.

In the service parameter tables in 3.1, parameters which are defined by this specialization have values entered in the Req, Ind, Rsp or Cnf columns. Other service parameters in these tables which are copied from the RDA Generic to improve readability do not have entries in these columns.

The RDA SQL Specialization defines a means of communicating SQL database language statements and their parameters from an RDA client to an RDA server, and of returning the results of those statements. Database Language SQL is supported at various levels of conformance, determined by an Object Identifier defined in ISO/IEC 9075

In this part of ISO/IEC 9579, the reference "**ISO/IEC 9075 (Database Language SQL)**" means any International Standard carrying the designation ISO/IEC 9075 and year of approval. The specific version (or versions) of that standard that is intended in each instance is determined, for the RDA client, by the SQL Conformance Level requested either when the RDA dialogue is initialized or when a data resource is opened, and, for the RDA server, by the SQL Conformance Level (or Levels) claimed in the Protocol Implementation Conformance Statement.

When the SQL Conformance Level specifies a year value of 1987 or 1989, the relevant International Standard is ISO/IEC 9075:1989; when the SQL Conformance Level specifies a year value of 1992, the relevant International Standard is ISO/IEC 9075:1992.

The term "**SQL**" is used throughout in a generic sense, intending to cover Database Language SQL at a particular conformance level as well as valid statements written within that conformance level.

2 MODEL

2.1 The RDA SQL Specialization Service model

2.1.1 Mapping to the general model of the RDA Service

This subclause relates the relevant concepts defined in Section 2 of ISO/IEC 9579-1 to this RDA SQL Specialization.

The term **data resource**, defined in ISO/IEC 9579-1, corresponds to an **SQL database resource** as defined in this part of ISO/IEC 9579.

The RDA client gains access to an SQL database resource at the RDA server by opening it. It is then available for use in requests for Database Language Services (3.1.5 "Database Language Services" on page 11). Nested SQL database resources are not supported.

Closing an SQL database resource causes it to be made inaccessible to the RDA client; that is, unavailable for subsequent use in requests for Database Language Services.

An implementor shall provide an RDA server at which one or multiple SQL database resources are available.

2.1.2 Mapping to the Concepts of Database Language SQL

This subclause relates the relevant Concepts defined in clause 4 of ISO/IEC 9075 (Database Language SQL) to the database model contained in the RDA server.

RDA SQL statements are executed by the database server exactly as if they were embedded in a host program local to the SQL database resource. Any exception condition or completion condition raised by the database server is returned to the RDA client.

Note: A <declare cursor> RDA SQL statement must have been executed or invoked at the RDA server prior to executing or invoking any <open statement> that uses the same cursor name. A <dynamic declare cursor> or <allocate cursor statement> must have been executed or invoked at the RDA server prior to executing or invoking any <dynamic open statement> that uses the same <dynamic cursor name>.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9579-2:1998

3 SERVICE

3.1 The RDA SQL Specialization Service

This clause contains the expansion of the RDA Generic Service parameters which are specialization defined. These specifications are in addition to those specified in ISO/IEC 9579-1.

Note: The subclauses in this clause are elaborations of the corresponding subclauses in ISO/IEC 9579-1 and are numbered accordingly.

3.1.1 RDA Dialogue Management services

3.1.1.1 RDA Dialogue Initialization functional unit

3.1.1.1.1 R-Initialize Service

Table 1 lists the R-Initialize SQL Specific Service Parameters.

	Req	Ind	Rsp	Cnf
<u>Request Parameters</u>				
sQLInitializeArgument	U	C(=)		
sQLConformanceLevelDefault	U	C(=)		
userData	C	C(=)		
<u>Result Parameters</u>				
sQLInitializeResult			U	C(=)
userData			C	C(=)
<u>Error Parameters</u>				
sQLInitializeError			S	S(=)
characterSetNotSupported			S	S(=)

Request Parameters

sQLInitializeArgument:

This argument is used to negotiate the level of support desired by the RDA client.

sQLConformanceLevelDefault:

This parameter identifies the characteristics ("SQL level") of the Database Language SQL requested by the RDA client. The object identifiers specifying those characteristics are defined in ISO/IEC 9075:1992. This is the default SQL Conformance Level requested by the RDA client. If this parameter is omitted, there is no default established for the RDA client.

userData:

The meaning of this parameter is defined by the implementor of the RDA server.

Result Parameters

sQLInitialize Result:

This parameter contains information returned by the RDA server whose meaning is specific to this RDA SQL Specialization

userData:

The meaning of this parameter is defined by the implementor of the RDA server.

Error Parameters

sQLInitializeError:

This parameter is used by the RDA server to report errors.

characterSetNotSupported:

This parameter is returned by the RDA server when the RDA server does not support a character set associated with a character string value received from an RDA client.

3.1.1.2 RDA Dialogue Termination functional unit

3.1.1.2.1 R-Terminate Service

Note: None of the specific request, result or error parameters for this service are used in this Specialization.

3.1.2 RDA Transaction Management services

3.1.2.1 RDA Transaction Management functional unit

3.1.2.1.1 R-BeginTransaction Service

Note: There are no specific request, result or error parameters.

3.1.2.1.2 R-Commit Service

Note: There are no specific request, result or error parameters.

3.1.2.1.3 R-Rollback Service

Note: There are no specific request, result or error parameters.

3.1.3 RDA Control Services

3.1.3.1 Cancel functional unit

3.1.3.1.1 R-Cancel Service

Table 2 lists the R-Cancel SQL Specific Service Parameters.

	Req	Ind	Rsp	Cnf
<u>Error Parameters</u>				
sQLCancelError			S	S(=)
characterSetNotSupported			S	S(=)

Error Parameters

sQLCancelError:

This parameter is used by the RDA server to report errors.

characterSetNotSupported:

This parameter is returned by the RDA server when the RDA server does not support a character set associated with a character string value received from an RDA client.

3.1.3.2 Status functional unit

3.1.3.2.1 R-Status Service

Table 3 lists the R-Status SQL Specific Service Parameters.

	Req	Ind	Rsp	Cnf
<u>Error Parameters</u>				
sQLStatusError			S	S(=)
characterSetNotSupported			S	S(=)

Error Parameters

sQLStatusError:

This parameter is used by the RDA server to report errors.

characterSetNotSupported:

This parameter is returned by the RDA server when the RDA server does not support a character set associated with a character string value received from an RDA client.

3.1.4 Resource Handling services

3.1.4.1 Resource Handling functional unit

3.1.4.1.1 R-Open Service

Table 4 lists the R-Open SQL Specific Service Parameters.

Table 4. R-Open SQL Specific Service Parameters				
	Req	Ind	Rsp	Cnf
<u>Request Parameters</u>				
sQLAccessControlData	U	C(=)		
sQLUsageMode	U	C(=)		
sQLOpenArgument	U	C(=)		
charSet	U	C(=)		
sQLConformanceLevel	U	C(=)		
sQLDiagnosticsRequested	U	C(=)		
sQLDescriptorStatements	U	C(=)		
<u>Result Parameters</u>				
sQLOpenResult			U	C(=)
charSet			U	C(=)
charSetNotSupported			U	C(=)
sQLConformanceLevel			C	C(=)
<u>Error Parameters</u>				
sQLOpenError			S	S(=)
invalidSQLConformanceLevel			S	S(=)
rDATransactionOpen			S	S(=)
sQLAccessControlViolation			S	S(=)
sQLDatabaseResourceAlreadyOpen			S	S(=)
characterSetNotSupported			S	S(=)

Request Parameters

dataResourceName:

This parameter (defined in ISO/IEC 9579-1) specifies the <catalog name> of a catalog belonging to an SQL database resource as defined in this part of ISO/IEC 9579.

sQLAccessControlData:

Data supplied by the RDA client to authenticate the right to open the SQL database resource for the required usage.

sQLUsageMode:

This argument specifies the access mode to the SQL database resource. If "retrieval" mode is chosen then all the objects within that SQL database resource can be accessed for read-only purposes and any attempt to modify any object within that SQL database resource causes an error usageModeViolation. If "update" mode is chosen then update, insert and delete of the objects within that SQL database resource are permitted. The applicable privileges as defined in ISO/IEC 9075:1992 may further restrict the access to any of the objects within that SQL database resource irrespective of the sQLUsageMode. The default value for sQLUsageMode is "retrieval".

Note: The RDA service does not enforce consistency between the sQLUsageModeValue sent on R-Open and the access mode attributes on a <set transaction statement>.

sQLOpenArgument:

This parameter contains information sent by the RDA client whose meaning is specific to SQL.

charSet:

This parameter uniquely identifies the specification of a coded character set. The character repertoire for character data associated with (described by) this parameter is the character repertoire specified in the identified coded character set specification. The character set identified by that specification shall be used by the RDA client as the default in this RDA dialogue for all RDA SQL Statements and their associated character data arguments accessing the SQL database resource opened by this RDA service.

Note: This default can be overridden by explicit specification, in an sQLDBLArgumentSpecification or an sQLDBLStatement, of a character set for a particular argument value or RDA SQL statement.

If this parameter is omitted, there is no default established for the RDA client.

Note: This means that a charSet parameter must be specified for every character data argument and sQLDBLStatement sent by the RDA client.

sQLConformanceLevel:

This parameter identifies the characteristics ("SQL level") of the Database Language SQL requested by the RDA client. The object identifiers specifying those characteristics are defined in ISO/IEC 9075:1992. This is the sQLConformanceLevel which the RDA client requests for all RDA SQL statements for this SQL database resource. This parameter shall be provided if no default was established by the RDA client on R-Initialize.

sQLDiagnosticsRequested:

This parameter specifies the level of diagnostics information requested by the RDA client. If "always" is chosen, then the RDA client desires that diagnostics information always be included in the results of subsequent R-ExecuteDBL and R-InvokeDBL requests. If "onRequest" is chosen, then the RDA client desires that diagnostics information be included in the results of a subsequent R-ExecuteDBL and R-InvokeDBL request only when specifically requested on that individual R-ExecuteDBL or R-InvokeDBL request. If "never" is chosen, then the RDA client does not desire that diagnostics information be included in the results of subsequent R-ExecuteDBL and R-InvokeDBL requests. The default value for sQLDiagnosticsRequested is "never".

sQLDescriptorStatements:

This parameter specifies whether the client will flow the descriptor statements <allocate descriptor statement>, <deallocate descriptor statement>, <get descriptor statement> and <set descriptor statement>. If «always» is chosen, then the RDA client will flow these statements in subsequent R-ExecuteDBL requests. If «never» is chosen, then the RDA client will not flow these statements on subsequent R-Execute requests. The default value of «sQLDescriptorStatements» is always.

Note: If the client chooses an sQLDescriptorStatements value of «never» then it is the responsibility of the client to maintain local descriptor areas, for example by caching the relevant values received on R-ExecuteDBL results.

parentDataResourceHandle:

This parameter is not used in this Specialization.

Result Parameters**sQLOpenResult:**

This parameter contains information returned by the RDA server whose meaning is specific to SQL.

charSet:

This parameter uniquely identifies the specification of a coded character set. The character repertoire for character data associated with (described by) this parameter is the character repertoire specified in the identified coded character set specification. The character set identified by that specification shall be used by the RDA server as the default in this RDA dialogue for all character data results that are associated with DBL statements accessing the SQL database resource opened by this RDA service.

If the default character set declared by the RDA client in the charSet argument parameter is also supported for result parameters by the RDA server, then the RDA server declares the same default character set for itself by responding with the same character set identification.

Note: This default can be overridden by explicit specification, in an sQLDBLResultSpecification, of a character set for a particular result value.

If this parameter is omitted, there is no default established for the RDA server.

Note: This means that a charSet parameter must be specified for every character data result sent by the RDA server.

charSetNotSupported:

The character set declared by the RDA client is not supported by the RDA server.

Note: If the RDA server returns this parameter the RDA client should close the SQL database resource and propose another default character set.

sQLConformanceLevel:

If the RDA server cannot provide the SQL level requested by the RDA client then this parameter identifies the characteristics ("SQL level") of the Database Language SQL that the RDA server will provide. The object identifiers specifying those characteristics are defined in ISO/IEC 9075:1992. If the RDA server does not support the sQLConformanceLevel requested by the RDA client, the RDA server returns the sQLConformanceLevel which the RDA server does support in this result parameter.

Error Parameters**sQLOpenError:**

This parameter is used by the RDA server to report errors.

invalidSQLConformanceLevel:

The value of sQLConformanceLevel is not allowed. If the sQLConformanceLevel request parameter was supplied on the R-Open request, then the error refers to the value in that parameter. Otherwise it refers to the sQLConformanceLevelDefault request parameter value as supplied on the previous R-Initialize request.

rDATransactionOpen

The opening of an SQL database resource is not allowed within a transaction.

sQLAccessControlViolation:

The RDA client does not have the required authority to open the SQL database resource for the requested sQLUsageMode.

sQLDatabaseResourceAlreadyOpen

There is already an SQL database resource open.

characterSetNotSupported:

This parameter is returned by the RDA server when the RDA server does not support a character set associated with a character string value received from an RDA client.

3.1.4.1.2 R-Close Service

Table 5 lists the R-Close SQL Specific Service Parameters.

	Req	Ind	Rsp	Cnf
<i>Error Parameters</i>				
sQLCloseError			S	S(=)
rDATransactionOpen			S	S(=)

Error Parameters**sQLCloseError:**

This parameter is used by the RDA server to report errors.

rDATransactionOpen

The closing of an SQL database resource is not allowed within a transaction.

3.1.5 Database Language Services

There are no constraints on intermixing invocation (through R-InvokeDBL operations) and execution (through R-Execute DBL operations) of RDA SQL statements.

3.1.5.1 Immediate Execution DBL functional unit**3.1.5.1.1 R-ExecuteDBL Service**

Table 6 lists the R-ExecuteDBL SQL Specific Service Parameters.

Table 6. R-ExecuteDBL SQL Specific Service Parameters				
	Req	Ind	Rsp	Cnf
<u>Request Parameters</u>				
sQLDBLStatement	M	M(=)		
sQLDBLArgumentSpecification	U	C(=)		
sQLDBLResultSpecification	U	C(=)		
dBLArguments				
singleArgument				
sQLDBLArgumentValues	U	C(=)		
multipleArgument				
listOfSQLDBLArgumentValues	U	C(=)		
returnSQLDiagnostics	U	C(=)		
<u>Result Parameters</u>				
sQLDBLResultSpecification			C	C(=)
listOfResultValues				
sQLDBLException			M	M(=)
sQLSTATE			C	C(=)
sQLCODE			C	C(=)
sQLErrorText			U	C(=)
sQLDiagnostics			U	C(=)
sQLDBLResultValues			C	C(=)
<u>Error Parameters</u>				
sQLExecuteDBLError			S	S(=)
hostIdentifierError			S	S(=)
rDATransactionNotOpen			S	S(=)
sQLDBLArgumentCountMismatch			S	S(=)
sQLDBLArgumentTypeMismatch			S	S(=)
sQLDBLNoCharSet			S	S(=)
sQLDBLStatementNotAllowed			S	S(=)
sQLUsageModeViolation			S	S(=)
characterSetNotSupported			S	S(=)
Note: The specific TerminationCode result parameter is not used in this Specialization.				

Request Parameters**sQLDBLStatement:**

This parameter contains the RDA SQL statement to be executed and is fully defined in "3.1.6.1 sQLDBLStatement:" on page 17.

sQLDBLArgumentSpecification:

This parameter consists of a list of SQL datatype descriptors and is fully defined in "3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification" on page 18.

sQLDBLResultSpecification:

This parameter consists of a list of SQL datatype descriptors and is fully defined in "3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification" on page 18.

sQLDBLArgumentValues:

This parameter consists of a list of the values of the argument parameters defined in the sQLDBLArgumentSpecification and is fully defined in "3.1.6.3 sQLDBLArgumentValues and sQLDBLResultValues" on page 23.

listOfSQLDBLArgumentValues:

This parameter consists of multiple lists of the values of the argument parameters defined in the sQLDBLArgumentSpecification and is fully defined in "3.1.6.3 sQLDBLArgumentValues and sQLDBLResultValues" on page 23.

returnSQLDiagnostics

This parameter specifies the level of diagnostics information requested by the RDA client when

'onRequest' was chosen by the RDA Client for the sQLDiagnosticsRequested parameter on R-Open. If "true" is chosen, then the RDA client desires that diagnostics information be included in the results of the request. If "false" is chosen, then the RDA client desires that diagnostics information not be included in the results of the request. The default value for returnSQLDiagnostics is "false".

Result Parameters

sQLDBLResultSpecification:

This parameter consists of a list of SQL datatype descriptors and is fully defined in "3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification" on page 18.

sQLDBLException:

This defines the completion code for the execution of the RDA SQL statement. It is the method used by the RDA server to return SQL exceptions (exception conditions or completion conditions).

sQLSTATE:

An SQLSTATE value as defined in ISO/IEC 9075:1992 or by the RDA server in conformance with that specification.

sQLCODE:

An SQLCODE value as defined in ISO/IEC 9075 (Database Language SQL).

sQLErrorText:

Supplemental information describing the error or warning corresponding to the sQLSTATE value or the sQLCODE value.

sQLDiagnostics

The semantics of the components of sQLDiagnostics are specified in clause 18 (Diagnostics Management) of ISO/IEC 9075:1992. An optional parameter may be omitted if its value is null.

sQLDBLResultValues:

This parameter consists of a list of the values of the result parameters defined in the sQLDBLResultSpecification and is fully defined in "3.1.6.3 sQLDBLArgumentValues and sQLDBLResultValues" on page 23.

Error Parameters

sQLExecuteDBLError:

This parameter is used by the RDA server to report errors.

hostIdentifierError:

The RDA server detected an error in an SQL variable name in an RDA SQL statement.

rDATransactionNotOpen:

No transaction is open and an attempt was made to execute an operation containing an RDA SQL statement.

sQLDBLArgumentCountMismatch:

The number of entries in sQLDBLArgumentValues parameter is not the same as in its associated sQLDBLArgumentSpecification parameter.

sQLDBLArgumentTypeMismatch:

The type of one or more entries in sQLDBLArgumentValues parameter is not the same as in its associated sQLDBLArgumentSpecification parameter.

sQLDBLNoCharSet

There is no default character set for the SQL database resource and either the sQLDBLArgumentSpecification contains a descriptor for character data without the explicit specification of its character set or the sQLDBLStatement contains character data without the explicit specification of its character set.

sQLDBLStatementNotAllowed:

The content of sQLDBLStatement is an SQL statement that is not permitted by the RDA SQL Specialization.

sQLUsageModeViolation:

The access mode of the SQL database resource associated with this request is in conflict with the RDA SQL statement specified.

characterSetNotSupported:

This parameter is returned by the RDA server when the RDA server does not support a character set associated with a character string value received from an RDA client.

3.1.5.2 Stored Execution DBL functional unit**3.1.5.2.1 R-DefinedDBL Service**

Table 7 lists the R-DefinedDBL SQL Specific Service Parameters.

	Req	Ind	Rsp	Cnf
<u>Request Parameters</u>				
sQLDBLStatement	M	M(=)		
sQLDBLArgumentSpecification	U	C(=)		
sQLDBLResultSpecification	U	C(=)		
<u>Result Parameters</u>				
sQLDBLResultSpecification			C	C(=)
sQLDBLException			C	C(=)
sQLSTATE			C	C(=)
sQLCODE			C	C(=)
sQLErrorText			U	C(≠)
<u>Error Parameters</u>				
sQLDefineDBLError			S	S(=)
hostIdentifierError			S	S(=)
sQLDBLNoCharSet			S	S(=)
sQLDBLStatementNotAllowed			S	S(=)
sQLUsageModeViolation			S	S(=)
characterSetNotSupported			S	S(=)

Request Parameters**sQLDBLStatement:**

This parameter contains the RDA SQL statement to be defined and is fully defined in "3.1.6.1 sQLDBLStatement:" on page 17.

sQLDBLArgumentSpecification:

This parameter consists of a list of SQL datatype descriptors and is fully defined in "3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification" on page 18.

sQLDBLResultSpecification:

This parameter consists of a list of SQL datatype descriptors and is fully defined in "3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification" on page 18.

Result Parameters**sQLDBLResultSpecification:**

This parameter consists of a list of SQL datatype descriptors and is fully defined in "3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification" on page 18.

sQLDBLException:

This defines the completion code for the execution of the RDA SQL statement. It is the method used by the RDA server to return SQL exceptions (exception conditions or completion conditions).

sQLSTATE:

An SQLSTATE value as defined in ISO/IEC 9075:1992 or by the RDA server in conformance with that specification.

sQLCODE:

An SQLCODE value as defined in ISO/IEC 9075 (Database Language SQL).

sQLErrorText

Supplemental information describing the error or warning corresponding to the sQLSTATE value or the sQLCODE value.

Error Parameters**sQLDefinedDBLError:**

This parameter is used by the RDA server to report errors.

hostIdentifierError:

The RDA server detected an error in an SQL variable name in an SQL statement.

sQLDBLNoCharSet

There is no default character set for the SQL database resource and either the sQLDBLArgumentSpecification contains a descriptor for character data without the explicit specification of its character set or the sQLDBLStatement contains character data without the explicit specification of its character set.

sQLDBLStatementNotAllowed:

The content of sQLDBLStatement is an SQL statement that is not permitted by the RDA SQL Specialization.

sQLUsageModeViolation:

The access mode of the SQL database resource associated with this request is in conflict with the RDA SQL statement specified.

characterSetNotSupported:

This parameter is returned by the RDA server when the RDA server does not support a character set associated with a character string value received from an RDA client.

3.1.5.2.2 R-InvokeDBL Service

Table 8 lists the R-InvokeDBL SQL Specific Service Parameters.

Table 8. R-InvokeDBL SQL Specific Service Parameters				
	Req	Ind	Rsp	Cnf
<u>Request Parameters</u>				
dBL Arguments				
singleArgument				
sQLDBLArgumentValues	U	C(=)		
multipleArgument				
listOfSQLDBLArgumentValues	U	C(=)		
returnSQLDiagnostics	U	C(=)		
<u>Result Parameters</u>				
sQLDBLResultSpecification			C	C(=)
listOfResultValues				
sQLDBLException			M	M(=)
sQLSTATE			C	C(=)
sQLCODE			C	C(=)
sQLErrorText			U	C(=)
sQLDiagnostics			U	C(=)
sQLDBLResultValues			C	C(=)
<u>Error Parameters</u>				
sQLInvokeDBLError			S	S(=)
rDA TransactionNotOpen			S	S(=)
sQLDBLArgumentCountMismatch			S	S(=)
sQLDBLArgumentTypeMismatch			S	S(=)
sQLUsageModeViolation			S	S(=)
characterSetNotSupported			S	S(=)
Note: The specific TerminationCode result parameter is not used in this Specialization.				

Request Parameters**sQLDBLArgumentValues:**

This parameter consists of a list of the values of the argument parameters defined in the sQLDBLArgumentSpecification and is fully defined in "3.1.6.3 sQLDBLArgumentValues and

sQLDBLResultValues" on page 23.

listOfSQLDBLArgumentValues:

This parameter consists of multiple lists of the values of the argument parameters defined in the sQLDBLArgumentSpecification and is fully defined in "3.1.6.3 sQLDBLArgumentValues and sQLDBLResultValues" on page 23.

returnSQLDiagnostics:

This parameter specifies the level of diagnostics information requested by the RDA client when 'onRequest' was chosen by the RDA Client for the sQLDiagnosticsRequested parameter on R-Open. If "true" is chosen, then the RDA client desires that diagnostics information be included in the results of the request. If "false" is chosen, then the RDA client desires that diagnostics information not be included in the results of the request. The default value for returnSQLDiagnostics is "false".

Result Parameters**sQLDBLResultSpecification:**

This parameter consists of a list of SQL datatype descriptors and is fully defined in "3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification" on page 18.

sQLDBLException:

This defines the completion code for the execution of the RDA SQL statement. It is the method used by the RDA server to return SQL exceptions (exception conditions or completion conditions).

sQLSTATE:

An sQLSTATE value as defined in ISO/IEC 9075:1992 or by the RDA server in conformance with that specification.

sQLCODE:

An sQLCODE value as defined in ISO/IEC 9075 (Database Language SQL).

sQLErrorText:

Supplemental information describing the error or warning corresponding to the sQLSTATE value or the sQLCODE value.

sQLDiagnostics

The semantics of the components of sQLDiagnostics are specified in clause 18 (Diagnostics Management) of ISO/IEC 9075:1992. An optional parameter may be omitted if its value is null.

sQLDBLResultValues:

This parameter consists of a list of the values of the result parameters defined in the sQLDBLResultSpecification and is fully defined in "3.1.6.3 sQLDBLArgumentValues and sQLDBLResultValues" on page 23.

Error Parameters**sQLInvokeDBLError:**

This parameter is used by the RDA server to report errors.

rDATransactionNotOpen:

No transaction is open and an attempt was made to execute an operation containing an RDA SQL statement.

sQLDBLArgumentCountMismatch:

The number of entries in sQLDBLArgumentValues parameter is not the same as in its associated sQLDBLArgumentSpecification parameter.

sQLDBLArgumentTypeMismatch:

The type of entries in sQLDBLArgumentValues parameter is not the same as in its associated sQLDBLArgumentSpecification parameter.

sQLUsageModeViolation:

The access mode of the SQL database resource associated with this request is in conflict with the RDA SQL statement specified.

characterSetNotSupported:

This parameter is returned by the RDA server when the RDA server does not support a character set associated with a character string value received from an RDA client.

3.1.5.2.3 R-DropDBLService:

Note: None of the specific request, result or error parameters for this Service are used in this Specialization.

3.1.6 SQL statements, arguments and results:

3.1.6.1 sQLDBLStatement:

Table 9 lists the sQLDBLStatement parameter.

	Req	Ind	Rsp	Cnf
statementText	M	M(=)		
charSet	U	C(=)		
sQLContextSpecifier	U	C(=)		
contextName	U	C(=)		
charSet	U	C(=)		
schemaName	U	C(=)		
authorizationIdentifier	U	C(=)		
systemUser	U	C(=)		

statementText:

This parameter is an encoded string containing the RDA SQL statement. The encoding of the RDA SQL statement shall be the encoding specified in the coded character set specification identified by the corresponding charSet parameter.

charSet:

This parameter uniquely identifies the specification of a coded character set. The character repertoire for the RDA SQL statement associated with (described by) this parameter is the character repertoire specified in the identified coded character set specification. If the parameter is omitted, then the character set is the default established by declaration during the execution of the R-Open service that opened the associated SQL database resource. If no default was established, then this charSet parameter must be specified.

sQLContextSpecifier:

This parameter specifies the context defined by the SQL module within which the statement is defined, namely the <module character set specification>, <schema name>, <module authorisation identifier>, SYSTEM_USER and within which temporary table references are resolved, which affects the execution of statements within the module. If this parameter is not provided, then values from the most recent previous sQLDBLStatement within the opened data resource are used. If this is the first sQLDBLStatement within the opened data resource and no contextSpecifier is provided the default values are used as defined for modules in ISO 9075

This parameter must not be present if the sQLConformanceLevel parameter provided to R_Open for the opened data resource specified any of the following:

- an <SQL edition> value of <1987>
- an <SQL edition> value of <1989>
- an <SQL edition> value of <1992> and <SQL conformance> value of <low>

contextName:

This parameter uniquely identifies the context within the session. If this parameter is not present the default context is identified.

Note: There is not necessarily a correspondance between contextName and the SQL <module name>.

charSet:

If this parameter is present it corresponds to the SQL <module character set specification> of the module that defines this context.

schemaName:

If this parameter is present it corresponds to the SQL <schema name> of the module that defines this context.

authorisationIdentifier:

If this parameter is present it corresponds to the SQL <module authorisation identifier> of the module that defines this context.

systemUser:

If this parameter is present it corresponds to the value of SYSTEM_USER associated with this module.

3.1.6.2 sQLDBLArgumentSpecification and sQLDBLResultSpecification

Table 10 lists the sQLDBLArgumentSpecification and sQLDBLResultSpecification parameters.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9579-2:1998

Table 10. sQLDBLArgumentSpecification and sQLDBLResultSpecification				
	Req	Ind	Rsp	Cnf
listOfSQLDataTypeDescriptor	U	C(=)	U	C(=)
nullable			M	M(=)
colName			M	M(=)
typeDescriptor	M	M(=)	M	M(=)
characterType	S	S(=)	S	S(=)
charSet	U	C(=)	U	C(=)
length	M	M(=)	M	M(=)
fixedLengthEncoding	M	M(=)	M	M(=)
characterSetCatalog	U	C(=)	U	C(=)
characterSetSchema	U	C(=)	U	C(=)
characterSetName	U	C(=)	U	C(=)
collationCatalog	U	C(=)	U	C(=)
collationSchema	U	C(=)	U	C(=)
collationName	U	C(=)	U	C(=)
numericType	S	S(=)	S	S(=)
precision	M	M(=)	M	M(=)
scale	M	M(=)	M	M(=)
decimalType	S	S(=)	S	S(=)
precision	M	M(=)	M	M(=)
scale	M	M(=)	M	M(=)
integerType	S	S(=)	S	S(=)
precision	M	M(=)	M	M(=)
precisionBase	M	M(=)	M	M(=)
smallIntType	S	S(=)	S	S(=)
precision	M	M(=)	M	M(=)
precisionBase	M	M(=)	M	M(=)
floatType	S	S(=)	S	S(=)
mantissaPrecision	M	M(=)	M	M(=)
maxExponent	M	M(=)	M	M(=)
realType	S	S(=)	S	S(=)
mantissaPrecision	M	M(=)	M	M(=)
maxExponent	M	M(=)	M	M(=)
doublePrecisionType	S	S(=)	S	S(=)
mantissaPrecision	M	M(=)	M	M(=)
maxExponent	M	M(=)	M	M(=)
dateTimeType	S	S(=)	S	S(=)
dateTimeQualifier	M	M(=)	M	M(=)
fractionalSecondsPrecision	U	C(=)	U	C(=)
intervalType	S	S(=)	S	S(=)
intervalQualifier	M	M(=)	M	M(=)
leadingFieldPrecision	M	M(=)	M	M(=)
fractionalSecondsPrecision	U	C(=)	U	C(=)
varcharType	S	S(=)	S	S(=)
charSet	U	C(=)	U	C(=)
length	M	M(=)	M	M(=)
characterSetCatalog	U	C(=)	U	C(=)
characterSetSchema	U	C(=)	U	C(=)
characterSetName	U	C(=)	U	C(=)
collationCatalog	U	C(=)	U	C(=)
collationSchema	U	C(=)	U	C(=)
collationName	U	C(=)	U	C(=)
bitType	S	S(=)	S	S(=)
length	M	M(=)	M	M(=)
bitVaryingType	S	S(=)	S	S(=)
length	M	M(=)	M	M(=)

Note: The Req and Ind columns apply when the listOfSQLDataTypeDescriptor is part of an Request parameter; the Rsp and Cnf columns apply when the list is part of a Result parameter.

listOfSQLDataTypeDescriptor:

Each item in the listOfSQLDataTypeDescriptor parameter defines a separate occurrence of an input or

output <variable specification> within the sQLDBLStatement. The nth member of the list defines the SQL data type of the nth occurrence of a <variable specification> within the sQLDBLStatement.

nullable:

This parameter is "true" if the column is possibly nullable, "false" if it is known not nullable.

colName:

This parameter specifies the name of the result column. The colName parameter occurs within items in the listOfSQLDataTypeDescriptors that are returned by an RDA server as part of sQLDBLResultSpecification.

typeDescriptor:

This parameter defines the data type of the corresponding item in an sQLDBLArgumentValues or sQLDBLResultValues parameter. The nth descriptor refers to the nth data item.

characterType:

This parameter describes an item of character data.

charSet:

This parameter uniquely identifies the specification of a coded character set. The character repertoire for character data associated with (described by) this parameter is the character repertoire specified in the identified coded character set specification. If the parameter is omitted, then the character set is the default established by the declaration during the execution of the R-Open service that opened the associated database resource. If no default was established, then this parameter must be specified.

length:

This parameter specifies the maximum number of characters allowed for the corresponding character data item.

fixedLengthEncoding:

This parameter specifies whether an sQLValue of the characterType has SPACES removed from the end of the string for transmission. When this parameter has a value of "true" it specifies a fixed length encoding and no SPACES are removed from the corresponding sQLValue. When it has a value of "false" it specifies a variable length encoding and SPACES may be removed from the corresponding sQLValue. If a variable length encoding is specified, then the corresponding sQLValue transmitted is equivalent to the same sQLValue as if the sQLValue had been transmitted as a fixed length encoding and had been padded with SPACES at the end of the string.

characterSetCatalog:

This parameter is the name of the catalog within which the character set is defined.

characterSetSchema:

This parameter is the name of the schema within which the character set is defined.

characterSetName:

This parameter is the name of the character set of the character item.

collationCatalog:

This parameter is the name of the catalog within which the collation sequence is defined.

collationSchema:

This parameter is the name of the schema within which the collation sequence is defined.

collationName:

This parameter is the name of the collation sequence of the character item.

numericType:

This parameter describes an item of numeric data.

precision:

For numeric Type, specifies the number of digits to a decimal base needed to represent the value transmitted.

scale:

For numeric Type, specifies a non-negative integer. If the scale is S then the value transmitted is multiplied by 10^{-S} to obtain the actual value.

decimalType:

This parameter describes an item of decimal data.

precision:

For decimalType, specifies the number of digits to a decimal base needed to represent the value transmitted.

scale:

For decimalType, specifies a non-negative integer. If the scale is S then the value transmitted is multiplied by 10^S to obtain the actual value.

integerType:

This parameter describes an item of integer data.

precision:

For integerType, specifies the number of digits to the base as specified in the precisionBase parameter needed to represent the value transmitted.

precisionBase:

For integerType, specifies the base used to count the number of digits when determining the precision which is represented.

smallIntType:

This parameter describes an item of small integer data.

precision:

For smallIntType, specifies the number of digits to the base as specified in the precisionBase parameter needed to represent the value transmitted.

precisionBase:

For smallIntType, specifies the base used to count the number of digits when determining the precision which is represented.

floatType:

This parameter describes an item of floating point data.

mantissaPrecision:

For floatType, a positive integer that specifies the number of significant binary digits in the mantissa.

maxExponent:

For floatType, a non-negative integer that specifies the maximum absolute value of the exponent. If maxExponent is E then the value of the exponent can range from -E to +E.

realType:

This parameter describes an item of real data.

mantissa Precision:

For realType, a positive integer that specifies the number of significant binary digits in the mantissa.

maxExponent:

For realType, a non-negative integer that specifies the maximum absolute value of the exponent. If maxExponent is E then the value of the exponent can range from -E to +E.

doublePrecisionType:

This parameter describes an item of double precision data.

mantissaPrecision:

For doublePrecisionType, a positive integer that specifies the number of significant binary digits in the mantissa.

maxExponent:

For doublePrecisionType, a non-negative integer that specifies the maximum absolute value of the exponent. If maxExponent is E then the value of the exponent can range from -E to +E.

dateTimeType:

This parameter describes an item of datetime type.

dateTimeQualifier:

The dateTimeQualifier parameter corresponds to the DATETIME_INTERVAL_CODE parameter

defined in ISO/IEC 9075:1992 (SQL).

fractionalSecondsPrecision:

For `dateTimeType` and `intervalType` the `fractionalSecondsPrecision` parameter corresponds to the `<time fractional seconds precision>` or `<interval fractional seconds precision>` respectively defined in ISO/IEC 9075:1992 (SQL).

intervalType:

This parameter describes an item of interval type.

intervalQualifier:

The `intervalQualifier` parameter corresponds to the `DATETIME_INTERVAL_CODE` parameter defined in ISO/IEC 9075:1992 (SQL).

leadingFieldPrecision:

The `leadingFieldPrecision` parameter corresponds to the `<interval leading field precision>` parameter defined in ISO/IEC 9075:1992 (SQL).

fractionalSecondsPrecision:

The `fractionalSecondsPrecision` parameter corresponds to the `<interval fractional seconds field precision>` parameter defined in ISO/IEC 9075:1992 (SQL).

varcharType:

This parameter describes an item of varchar type.

charSet:

This parameter uniquely identifies the specification of a coded character set. The character repertoire for character data associated with (described by) this parameter is the character repertoire specified in the identified coded character set specification. If the parameter is omitted, then the character set is the default established by the declaration during the execution of the R-Open service that opened the associated database resource. If no default was established, then this parameter must be specified.

length:

This parameter specifies the maximum number of characters allowed for the corresponding character data item.

characterSetCatalog:

This parameter is the name of the catalog within which the character set is defined.

characterSetSchema:

This parameter is the name of the schema within which the character set is defined.

characterSetName:

This parameter is the name of the character set of the character item.

collationCatalog:

This parameter is the name of the catalog within which the collation sequence is defined.

collationSchema:

This parameter is the name of the schema within which the collation sequence is defined.

collationName:

This parameter is the name of the collation sequence of the character item.

bitType:

This parameter describes an item of bit type.

length:

The `length` parameter for `bitType` specifies the maximum number of bits allowed for the corresponding argument or result parameter.

bitVaryingType:

This parameter describes an item of bit type.

length:

The `length` parameter for `bitVaryingType` specifies the maximum number of bits allowed for the corresponding argument or result parameter.

3.1.6.3 sQLDBLArgumentValues and sQLDBLResultValues

Table 11 lists the sQLDBLArgumentValues and sQLDBLResultValues parameters.

	Req	Ind	Rsp	Cnf
listOfSQLValue	U	C(=)	U	C(=)
dataItem	U	C(=)	U	C(=)
characterItem	S	S(=)	S	S(=)
numericItem	S	S(=)	S	S(=)
decimalItem	S	S(=)	S	S(=)
integerItem	S	S(=)	S	S(=)
smallIntItem	S	S(=)	S	S(=)
floatItem	S	S(=)	S	S(=)
realItem	S	S(=)	S	S(=)
doublePrecisionItem	S	S(=)	S	S(=)
dateTimeItem	S	S(=)	S	S(=)
intervalItem	S	S(=)	S	S(=)
varcharItem	S	S(=)	S	S(=)
bitItem	S	S(=)	S	S(=)
bitVarItem	S	S(=)	S	S(=)
indicator	U	C(=)	U	C(=)

Note: The Req and Ind columns apply when the listOfSQLValue is part of an Request parameter; the RSp and Cnf columns apply when the list is part of a Result parameter.

listOfSQLValue:

Each item in the listOfSQLValue parameter contains the value of a separate <variable specification> within the RDA SQL statement.

dataItem:

This parameter contains the value of the Data Variable (as defined in 3.1.6.4, "Referencing to Parameters" on page 19).

characterItem:

This parameter contains the value of a character Data Variable. The encoding of the character data shall be the encoding specified in the coded character set specification identified by the corresponding charSet parameter.

numericItem:

This parameter contains the value of a numeric Data Variable.

decimalItem:

This parameter contains the value of a decimal Data Variable.

integerItem:

This parameter contains the value of an integer Data Variable.

smallIntItem:

This parameter contains the value of a small integer Data Variable.

floatItem:

This parameter contains the value of a floating point Data Variable.

realItem:

This parameter contains the value of a real Data Variable.

doublePrecisionItem:

This parameter contains the value of a double precision Data Variable.

dateTimeItem:

This parameter contains the value of a date time Data Variable. The values for dateTimeItem are restricted to those allowed for <date string>, <time string> and <timestamp string> excluding the surrounding <quote>s as defined in ISO/IEC 9075:1992 (SQL).

intervalItem:

This parameter contains the value of an interval Data Variable. The values for intervalItem are restricted to those allowed for <interval string> excluding the surrounding <quote>s as defined in ISO/IEC 9075:1992 (SQL).

varcharItem:

This parameter contains the value of a varying character Data Variable.

bitItem:

This parameter contains the value of a bit Data Variable.

bitVarItem:

This parameter contains the value of a varying bit Data Variable.

indicator:

This parameter contains the value of the <indicator variable>

3.1.6.4 Referencing to parameters

An RDA SQL statement may contain references to the parameters in the request and/or result response. Each parameter reference in the RDA SQL statement is replaced in the sQLDBLStatement by the <embedded variable name> ":H".

Figure 1 below shows how actual data is referenced:

sQLDBLStatement:

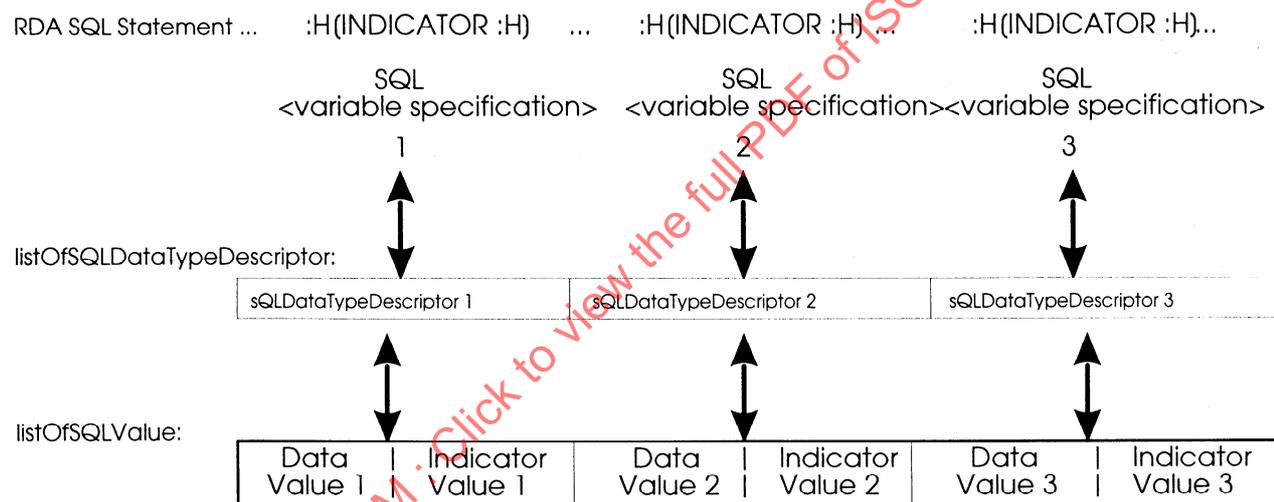


Figure 1. Usage of Embedded SQL Variables

A <variable specification> contains either one <embedded variable name>, if no <indicator variable> is included, or two <embedded variable name>s, if an <indicator variable> is included. The first or only <embedded variable name> corresponds to the value of the Data Variable (**Data Value** in Figure 1), the second corresponds to the value of the <indicator variable> (**Indicator Value** in Figure 1).

If the **Data Value** is null, then it is omitted and the **Indicator Value** is included. If the **Indicator Value** is zero, then it shall be omitted and the **Data Value** is included.

There is a one-to-one correspondence between items in the listOfSQLDataTypeDescriptor and items in the listOfSQLValue. The nth item in the listOfSQLDataTypeDescriptor (the list of descriptors) corresponds to the nth item in the listOfSQLValue (the list of values). For each corresponding pair of items, the data type specified by the "...Type" subparameter of the typeDescriptor parameter corresponds with the data type specified by the "...Item" subparameter of the dataItem parameter; for example, characterType corresponds to characterItem, numericType corresponds to numericItem, etc.

3.2 Sequencing rules

3.2.1.1 RDA client sequencing rules

In addition to the sequencing rules defined in ISO/IEC 9579-1, the following sequencing rules apply:

- a) If the RDA client has successfully opened an SQL database resource, then the RDA client shall successfully close the SQL database resource before issuing another R-Open request on the RDA dialogue.
- b) If the RDA client requests the execution of an RDA SQL statement then there shall be an RDA transaction open on the RDA dialogue.

3.2.2 RDA Server Sequencing Rules

There are no additional SQL specific sequencing rules for the RDA server.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9579-2:1998

4 PROTOCOL

4.1 The RDA SQL Specialization client execution rules

This clause specifies entities, entity attributes, and client execution rules required by this part of ISO/IEC 9579 which are in addition to those defined in ISO/IEC 9579-1.

4.1.1 RDA SQL Specialization dialogue-state model

The RDA SQL Specialization dialogue-state model is the RDA dialogue-state model defined in ISO/IEC 9579-1 with additional SQL Specialization entities and additional attributes in the dialogue-state model entities. This subclause of the RDA SQL Specialization defines the additional entities and entity attributes.

4.1.1.1 SQL Transaction Entity

An SQL transaction entity always exists at an RDA client.

The SQL transaction entity has the following attributes:

- a) **defaultTransactionMode**: a boolean attribute that indicates whether the default transaction mode settings of SQL92 are in effect within the current transaction.
 - «false»: the transaction mode settings in effect within the current transaction are the settings specified by the last <set transaction statement> executed by the SQL-agent.
 - «true»: the <level of isolation> ISOLATION LEVEL SERIALIZABLE and the <transaction access mode> READ WRITE are in effect within the current transaction.
- b) **setTransactionStatement**: the last <set transaction statement> executed by the SQL-agent.
- c) **transactionAccessMode**: the transaction access mode setting in effect within the current transaction:
 - «read only»
 - «read write»

4.1.2 General client execution rules

This subclause describes the general client execution rules which apply to requests received from an SQL-client to execute SQL-statements. The general rules provide for specific types of augmentation based upon the particular type of request received from an SQL-client. Those augmentations are defined in subsequent subclauses in this subclause (of this part of ISO/IEC 9579). Additional augmentations are specified by individual application contexts (in this part of ISO/IEC 9579).

4.1.2.1 Generation of the SQL transaction entity

An SQL transaction entity always exists at an RDA client with the following constraints on the initial attribute values:

<i>Attribute</i>	<i>Initial value</i>
defaultTransactionMode	«true»
setTransactionStatement	«null»
transactionAccessMode	«read write»

4.1.2.2 Execution of an SQL-statement

The client execution rules for a particular SQL-statement include two general types of rules. These are:

Entity manipulation rules: Entity manipulation rules specify constraints on the modification of the entities and attributes which comprise the RDA dialogue state. No modifications are allowed except when explicitly stated.

Request rules: Request rules specify actions that an RDA client must take upon receipt of a request to execute the SQL-statement.

4.1.3 <set transaction statement>

Entity manipulation rules

The SQL transaction entity shall be modified according to the following table:

<i>Attribute</i>	<i>New value</i>
defaultTransactionMode	«false»
setTransactionStatement	the text of the <set transaction statement>
transactionAccessMode	the <transaction access mode> of the <set transaction statement>

Request rules

The RDA client shall take the following actions:

If there is a current SQL-connection, then issue an R-ExecuteDBL request on the RDA dialogue that supports the SQL-connection with an sQLDBLStatement value equal to the setTransactionStatement value of the SQL transaction entity.

4.1.4 <connect statement>*Entity manipulation rules*

This part of ISO/IEC 9579 does not specify any entity manipulation rules for this SQL-statement.

Request rules

The RDA client shall take the following actions:

If the value of the defaultTransactionMode attribute of the SQL transaction entity is «false», then issue an R-ExecuteDBL request on the RDA dialogue that supports the SQL-connection established by the <connect statement> with an sQLDBLStatement value equal to the setTransactionStatement value of the SQL transaction entity.

4.1.5 <set connection statement>*Entity manipulation rules*

This part of ISO/IEC 9579 does not specify any entity manipulation rules for this SQL-statement.

Request rules

The RDA client shall take the following actions:

If the value of the defaultTransactionMode attribute of the SQL transaction entity is «false», then issue an R-ExecuteDBL request on the RDA dialogue that supports the SQL-connection selected by the <set connection statement> with an sQLDBLStatement value equal to the setTransactionStatement value of the SQL transaction entity.

4.1.6 <commit statement>*Entity manipulation rules*

The SQL transaction entity shall be modified according to the following table:

<i>Attribute</i>	<i>New value</i>
defaultTransactionMode	«true»
setTransactionStatement	«null»
transactionAccessMode	«read write»

Request rules

This part of ISO/IEC 9579 does not specify any request rules for this SQL-statement.

4.1.7 <rollback statement>*Entity manipulation rules*

The SQL transaction entity shall be modified according to the following table:

<i>Attribute</i>	<i>New value</i>
defaultTransactionMode	«true»
setTransactionStatement	«null»
transactionAccessMode	«read write»

Request rules

This part of ISO/IEC 9579 does not specify any request rules for this SQL-statement.

4.2 The RDA SQL Specialization server execution rules

This subclause specifies entity attributes and server execution rules required by the Specialization which

are in addition to those defined in ISO/IEC 9579-1:1993 (RDA Part 1:Generic).

4.2.1 RDA SQL Specialization dialogue-state model

The RDA SQL Specialization dialogue-state model is the RDA dialogue-state model defined in ISO/IEC 9579-1 with additional SQL Specialization attributes in the dialogue-state model entities. This subclause of the RDA SQL Specialization defines the additional entity attributes.

4.2.1.1 RDA dialogue entity

In addition to the attributes defined in ISO/IEC 9579-1 this subclause defines the following attributes for the RDA dialogue entity:

sSQLConformanceLevelDefault:

An identifier, defined in ISO/IEC 9075:1992 used to identify the Database Language SQL International Standard for the DBL statements allowed to be used in the context of the RDA dialogue. This is the default SQL Conformance Level requested by the RDA client.

userData:

This attribute and any references to it from the server execution rules are defined by the implementor of the RDA server.

4.2.1.2 Opened data resource entity

In addition to the attributes defined in ISO/IEC DIS 9579-1 (RDA Part 1: Generic), this subclause defines the following attributes for the opened data resource entity:

sSQLConformanceLevel:

An identifier, defined in ISO/IEC 9075:1992, used to identify the Database Language SQL International Standard for the DBL statements allowed to be used in the context of the RDA dialogue. This is the SQL Conformance Level which will be valid for all RDA DBL operations for this open SQL database resource.

sSQLUsageMode:

This attribute specifies the access mode to the SQL database resource.

retrieval: The SQL database resource can only be accessed for read-only purposes.

update: The SQL database resource can be accessed for update, insert and delete of objects within the SQL database resource.

sSQLDiagnosticsRequested:

This attribute specifies the level of diagnostics information requested by the RDA client.

always: The RDA client desires that diagnostics information always be included in the results of subsequent R-ExecuteDBL and R-InvokeDBL requests.

onRequest: The RDA client desires that diagnostics information be included in the results of a subsequent R-ExecuteDBL and R-InvokeDBL request only when specifically requested on that individual R-ExecuteDBL or R-InvokeDBL request.

never: The RDA client does not desire that diagnostics information be included in the results of subsequent R-ExecuteDBL and R-InvokeDBL requests.

clientDefCharSet:

This attribute specifies the default character set to be used by the RDA client for RDA SQL statements and for character data arguments that are associated with RDA SQL statements accessing the SQL database resource.

serverDefCharSet:

This attribute specifies the default character set to be used by the RDA server for character data results that are associated with RDA SQL statements accessing the SQL database resource.

4.2.1.3 Defined DBL Entity

In addition to the attributes defined in ISO/IEC 9579-1:1993 (RDA part 1:Generic), this subclause defines the following attributes for the defined DBL entity:

sSQLDBLArgumentSpecification:

The SQL data types of the arguments of the RDA SQL statement.

sQLResultSpecification:

The SQL data types of the results of the RDA SQL statement.

sQLDBLStatement:

The RDA SQL statement defined.

4.2.1.4 RDA operation entity

There are no additional operation entity attributes required for the RDA SQL Specialization.

4.2.2 General server execution rules

There are no additional general server execution rules required for the RDA SQL Specialization.

4.2.3 RDA Dialogue Management services**4.2.3.1 RDA Dialogue Initialization functional unit****4.2.3.1.1 R-Initialize Service****Entity Manipulation Rules**

The following defines the constraints on the initial values of the additional SQL Specialization attributes for the RDA dialogue entity:

Attribute	Initial Value
sQLConformanceLevelDefault	The sQLConformanceLevelDefault parameter value on the R-Initialize indication primitive if this parameter is provided. Otherwise, the initial value of the sQLConformanceLevelDefault attribute shall be set to "null".
userData	The userData parameter value on the R-Initialize indication primitive if this parameter is provided and it is supported by the RDA server. Otherwise the initial value of the userData attribute shall be set to a value supported by the RDA server.

Result Rules

If a result is returned then the result parameters shall satisfy the following constraints:

Result Parameter	Constraints
userData	The value of the userData attribute in the RDA dialogue entity if it does not equal the value of the userData parameter in the R-Initialize indication. Otherwise, the userData parameter shall be omitted.

Error Rules

No additional error rules are required for the RDA SQL Specialization.

4.2.3.2 RDA Dialogue Termination functional unit**4.2.3.2.1 R-Terminate Service**

No additional server execution rules are required for the RDA SQL Specialization.

4.2.4 RDA Transaction Management services**4.2.4.1 RDA Transaction Management functional unit****4.2.4.1.1 R-Begin Transaction Service**

No additional server execution rules are required for the RDA SQL Specialization.

4.2.4.1.2 R-Commit Service

No additional server execution rules are required for the RDA SQL Specialization.

4.2.4.1.3 R-Rollback Service

No additional server execution rules are required for the RDA SQL Specialization.

4.2.5 RDA Control services**4.2.5.1 Cancel functional unit****4.2.5.1.1 R-Cancel Service**

No additional server execution rules are required for the RDA SQL Specialization.

4.2.5.2 Status functional unit**4.2.5.2.1 R-Status Service**

No additional server execution rules are required for the RDA SQL Specialization.

4.2.6 Resource Handling services**4.2.6.1 Resource Handling functional unit****4.2.6.1.1 R-Open Service****Entity Manipulation Rules**

The following defines the constraints on the initial values of the additional SQL Specialization attributes for the opened data resource entity:

Attribute	Initial Value
sQLConformanceLevel	The sQLConformanceLevel parameter value on the R-Open indication primitive if this parameter is provided and it identifies an SQL level that is supported by the RDA server; else the sQLConformanceLevelDefault attribute value in the RDA Dialogue entity if it identifies an SQL level that is supported by the RDA server; otherwise, the initial value of the sQLConformanceLevel attribute shall be set to a value supported by the RDA server.
sQLDiagnosticsRequested	The sQLDiagnosticsRequested parameter value on the R-Open indication primitive if provided. Otherwise, this attribute contains the default value "never".
sQLUsageMode	The sQLUsageMode parameter value on the R-Open indication primitive if provided. Otherwise, this attribute contains the default value "retrieval".
clientDefCharSet	The charSet parameter value on the R-Open indication primitive. If the charSet parameter on the R-Open indication primitive is not provided, the initial value of the clientDefCharSet attribute shall be "null" (to indicate each descriptor for a character string and each RDA SQL statement sent by the RDA client shall include a character set identifier).
serverDefCharSet	If the RDA server can support for result parameters the character set identified by the value of the charSet parameter of the R-Open indication primitive, then that value; else if the RDA server decides to use some other character set that it supports as its default, then an object identifier identifying that character set; else "null" (to indicate that each descriptor for a character string returned by the RDA server shall include a character set identifier).

Result Rules

If a result is returned then the result parameters shall satisfy the following constraints:

Result Parameter	Constraints
sQLConformanceLevel	If the sQLConformanceLevel parameter was provided in the R-Open indication, the value of the sQLConformanceLevel attribute in the opened data resource entity if this value does not equal the value of the sQLConformanceLevel parameter in the R-Open indication; else, if the sQLConformanceLevel parameter was not provided in the R-Open indication, the value of the sQLConformanceLevel attribute in the opened data resource entity if this value does not equal the value of the sQLConformanceLevelDefault attribute in the RDA dialogue entity; otherwise, the sQLConformanceLevel parameter shall be omitted.
charSetNotSupported	"true" if the RDA server cannot support the character set identified by the charSet parameter in the R-Open indication primitive, otherwise "false".
charSet	If the serverDefCharSet attribute value of the opened data resource entity, whose RDA dialogueID attribute identifies the current RDA dialogue and whose dataResourceHandle attribute equals the dataResourceHandle parameter of the R-Open indication primitive, is not "null", then the value of the attribute. Otherwise, this parameter shall be omitted.

Error Rules

Error	Type	Predicate
invalidSQLConformanceLevel	M+P	The sQLConformanceLevel parameter value on the R-Open indication primitive, or if none was provided the sQLConformanceLevelDefault attribute value in the RDA dialogue entity, does not conform to one of the object identifier values specified in ISO/IEC 9075:1992
rDATransactionOpen	P	The rDATransactionStatus attribute of the RDA dialogue entity for the current dialogue has the value of rDATransactionOpen.
sQLAccessControlViolation	M+P	The sQLAccessControlData parameter of the R-Open indication primitive, when used in conjunction with the identifyOfUser attribute of the RDA dialogue entity for the current RDA dialogue, fails to authorize the RDA client to open the SQL database resource specified by the dataResourceName parameter for the usage mode specified by the sQLUsageMode parameter.
sQLDatabaseResourceAlreadyOpen	M+P	There exists an opened data resource entity whose dialogueID attribute is that of the current dialogue.

4.2.6.1.2 R-Close Service**Error Rules**

Error	Type	Predicate
rDATransactionOpen	P	The rDATransactionStatus attribute of the RDA dialogue entity for the current RDA dialogue has the value of rDATransactionOpen.

4.2.7 Database Language services**4.2.7.1 Immediate Execution DBL functional unit****4.2.7.1.1 R-ExecuteDBL service**

The following rules shall be observed by the RDA server:

- An error shall be returned if no execution of the RDA SQL statement completes. Otherwise, the result shall be returned. The result includes one or more SQLDBLExceptions depending on the

repetitionCount or the number of sQLDBLArgumentValues in the listOfSQLDBLArgumentValues.

- The effect of execution of RDA SQL statements by the database server is exactly the same as if they were embedded in a host program local to the SQL database resource. In addition, a <declare cursor> RDA SQL statement containing a <cursor name> must have been executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) at the RDA server prior to executing any <open statement> that uses the same <cursor name>.
- For each execution of the RDA SQL statement, an sQLDBLException returned by the SQL database resource shall be included in the result. The sQLDBLResultValues, if returned by the SQL database resource, shall also be included.

Entity Manipulation Rules

No additional entity manipulation rules are required for the RDA SQL Specialization.

Result Rules

If a result is returned then the result parameters shall satisfy the following constraints:

Result Parameter	Constraints
charSet	If the serverDefCharSet attribute of the open data resource entity is "null", each character string in the sQLDBLResultValues parameter of the R-ExecuteDBL response primitives shall be accompanied by an object identifier in the charSet parameter in the corresponding item of listOfSQLDataTypeDescriptor.
sQLDBLException	<p>If the RDA SQL statement contained in the sQLDBLStatement parameter is not permitted by an implementation claiming conformance to ISO/IEC 9075 (Database Language SQL) at the level of conformance specified by the sQLConformanceLevel attribute of the opened data resource entity, an sQLDBLException may be returned.</p> <p>If the sQLConformanceLevel attribute of the opened data resource entity has an <SQL edition> value of <1987> or <1989>, an sQLDBLException shall contain an sQLCODE parameter, otherwise it shall contain an sQLSTATE parameter. Applicable sQLCODE values of the sQLCODE parameter are defined in conformance with ISO/IEC 9075:1989, applicable sQLSTATE values of the sQLSTATE parameter are defined in conformance with ISO/IEC 9075:1992.</p> <p>If the sQLDiagnosticsRequested attribute of the opened data resource entity has a value of "always" and the sQLConformanceLevel attribute of the opened data resource entity has an <SQL conformance> value of 1, Intermediate(1), 2, or High(2), then sQLDiagnostics shall be returned by the RDA server. If the sQLDiagnosticsRequested attribute of the opened data resource entity has a value of onRequest and the sQLConformanceLevel attribute of the opened data resource entity has an <SQL conformance> value of 1, Intermediate(1), 2, or High(2), and the returnSQLDiagnostics parameter of the R-ExecuteDBL indication has a value of "true", then sQLDiagnostics shall be returned by the RDA server. In all other cases, sQLDiagnostics shall not be returned.</p>
colName	This parameter shall be specified within each item of listOfSQLDataTypeDescriptor that is part of an sQLDBLResultSpecification, otherwise it shall not be specified.

The following table defines R-ExecuteDBL use of SQL argument and result parameters.

Table 12. R-ExecutedDBL use of SQL argument and result parameters				
RDA SQL statement to be Executed	ArgSpec	ArgVal	ResSpec	ResVal
<close statement>				
<commit statement> ¹				
<declare cursor>	C→S(H) ⁶			
<delete statement: positioned>				
<delete statement: searched>	C→S(H)	C→S(H)		
<fetch statement>			C→S ²	C←S
<insert statement>	C→S(H)	C→S(H)		
<open statement>	C→S(H) ⁵	C→S(H)	C←S ³	
<rollback statement> ¹				
<select statement: single row>	C→S(H)	C→S(H)	C←→S ⁴	C←S
<update statement: positioned>	C→S(H)	C→S(H)		
<update statement: searched>	C→S(H)	C→S(H)		
<schema definition>				
<table definition>				
<view definition>				
<drop schema statement>				
<alter table statement>				
<drop table statement>				
<drop view statement>				
<domain definition>				
<alter domain statement>				
<drop domain statement>				
<assertion definition>				
<drop assertion statement>				
<revoke statement>				
<temporary table declaration>				
<grant statement>				
<allocate cursor statement>	C→S(H)	C→S(H)		
<allocate descriptor statement> ¹²	C→S(H)	C→S(H)		
<deallocate descriptor statement> ¹²	C→S(H)	C→S(H)		
<deallocate prepared statement>	C→S(H)	C→S(H)		
<describe statement>	C→S(H)	C→S(H)	C←S ⁸	
<dynamic close statement>	C→S(H)	C→S(H)		
<dynamic declare cursor>				
<dynamic delete statement: positioned>	C→S(H)	C→S(H)		
<dynamic fetch statement>	C→S(H)	C→S(H)	C→S ⁹	C←S
<dynamic open statement>	C→S(H) ⁷	C→S(H)	C←S ¹⁰	
<dynamic update statement: positioned>	C→S(H)	C→S(H)		
<execute statement>	C→S(H)	C→S(H)	C→S(H)	C←S
<execute immediate statement>	C→S(H)	C→S(H)		
<get descriptor statement> ¹²	C→S(H)	C→S(H)	C←S	
<preparable dynamic delete statement: positioned> ¹				
<preparable dynamic update statement: positioned> ¹				
<prepare statement>	C→S(H)	C→S(H)		
<set descriptor statement> ¹²	C→S(H)	C→S(H)		
<connect statement> ¹¹				
<set connection statement> ¹¹				
<disconnect statement> ¹¹				
<get diagnostics statement> ¹				

<character set definition>				
<drop character set statement>				
<collation definition>				
<drop collation statement>				
<translation definition>				
<drop translation statement>				
<set catalog statement>				
<set schema statement>	C→S(H)	C→S(H)		
<set constraints mode statement>				
<set transaction statement>				
<direct implementation-defined statement>	C→S(H)	C→S(H)	C→S	C←S
<SQL-server module definition>				
<drop module statement>				
<SQL-invoked routine>				
<drop routine statement>				
<set path statement>	C→S(H)	C→S(H)		
<call statement>	C→S(H)	C→S(H)	C→S	C←S
<compound statement>	C→S(H)	C→S(H)	C→S	C←S
<condition declaration>	C→S(H)	C→S(H)		
<SQL variable declaration>				
<assignment statement>	C→S(H)	C→S(H)	C→S	C←S
<case statement>	C→S(H)	C→S(H)	C→S	C←S
<if statement>	C→S(H)	C→S(H)	C→S	C←S
<loop statement>	C→S(H)	C→S(H)	C→S	C←S
<while statement>	C→S(H)	C→S(H)	C→S	C←S
<repeat statement>	C→S(H)	C→S(H)	C→S	C←S
<for statement>	C→S(H)	C→S(H)	C→S	C←S
<signal statement>	C→S(H)	C→S(H)		
<resignal statement>	C→S(H)	C→S(H)		
Legend:				
ArgSpec The sQLDBLArgumentSpecification parameter.				
ArgVal The sQLDBLArgumentValues parameter.				
ResSpec The sQLDBLResultSpecification parameter.				
ResVal The sQLDBLResultValues parameter.				
C→S Parameter supplied by the RDA client.				
C←S Parameter returned by the RDA server.				
(H) The RDA client must set this parameter if the RDA SQL statement contains host variables, except as noted below.				
blank cells unspecified (i.e. not subject to conformance testing)				

Notes:

1. See the sQLDBLStatementNotAllowed error under the Error Rules below.
2. The RDA client may optionally send the ResSpec, which will supercede the ResSpec sent on a previously executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <fetch statement> using the same cursor or received from the RDA server on the corresponding <open statement>.
3. The ResSpec describes the ResVal that the subsequently executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <fetch statement> using this cursor will return, but any ResSpec specified with such a <fetch statement> overrides this ResSpec.
4. The RDA client may optionally send the ResSpec, otherwise the RDA server shall return the ResSpec which describes the ResVal, if any.
5. The RDA client may optionally send the ArgSpec. If it is not specified, the RDA server shall use the ArgSpec sent on the previously executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <declare cursor> statement that used the same cursor name.
6. The RDA client may optionally send the ArgSpec. If it is not specified, the RDA client shall send the ArgSpec on the subsequently executed (via an R-Execute DBL operation) or invoked (via an R-InvokeDBL operation) <open statement> that uses the same cursor name.
7. The RDA client may optionally send the ArgSpec. If it is not specified, the RDA server shall use the ArgSpec sent on the previously executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <allocate cursor statement> or <dynamic declare cursor> that used the same <dynamic cursor name>.
8. If the <describe statement> is a <describe output statement>, then the RDA server shall return a ResSpec only if the <SQL statement name> in the <describe statement> refers to a prepared statement that is a <dynamic select statement> or a <dynamic single row select statement>. The i-th sQLDataTypeDescriptor of the ResSpec shall contain the description of the i-th column of the table defined by the prepared statement.
If the <describe statement> is a <describe input statement>, then the RDA server shall return a ResSpec only if the <SQL statement name> in the <describe statement> refers to a prepared statement that contains one or more <dynamic parameter specification>s. The i-th sQLDataTypeDescriptor of the ResSpec shall contain the description of the i-th <dynamic parameter specification> of the prepared statement.
9. The RDA client may optionally send the ResSpec, which will supercede the ResSpec sent on a previously executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <dynamic fetch statement> using the same cursor or received from the RDA server on the corresponding <dynamic open statement>.
10. The ResSpec describes the ResVal that a subsequently executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <dynamic fetch statement> using this cursor will return, but any ResSpec specified with such a <dynamic fetch statement> overrides this ResSpec.
11. If the content of sQLDBLStatement performs connection management (eg. a <connect statement>, <set connection statement> or <disconnect statement>), then the RDA server shall return the error sQLDBLStatementNotAllowed. A <connect statement> or <disconnect statement> may result in the RDA client invoking the Dialogue Management services provided by RDA or TP depending on the application context chosen, and the Resource Handling services provided by RDA. A <set connection statement> may result in the RDA client invoking the Resource Handling services provided by RDA.
12. If the opened data resource was opened with an sQLDescriptorStatements value of «never» then the RDA server shall return the error sQLDBLStatementNotAllowed

Error Rules

Error	Type	Predicate
hostIdentifierError	M+P	The <host Identifier> of an <embedded variable name> of the RDA SQL statement is not "H".
rDATransactionNotOpen	M+P	The RDA client requested the execution of an RDA SQL statement and the rDATransactionStatus attribute of the RDA dialogue entity for the current dialogue has the value of rDATransactionNotOpen.
sQLDBLArgumentCountMismatch	M+P	The number of items in listOfSQLValue within the sQLDBLArgumentValues parameter is not the same as the number of items in listOfSQLDataTypeDescriptor within the sQLDBLArgumentSpecification parameter.
sQLDBLArgumentTypeMismatch	M+P	The type of one or more items in listOfSQLValue within the sQLDBLArgumentValues parameter is not the same as the corresponding item (or items) in listOfSQLDataTypeDescriptor within the sQLDBLArgumentSpecification parameter.
sQLDBLNoCharSet	M+P	The sQLDBLArgumentSpecification or the sQLDBLResultSpecification contains a descriptor for character data without the explicit specification of its character set or the sQLDBLStatement contains no explicit specification of its character set, and the charSet attribute of the opened data resource entity identified by the dataResourceHandle parameter is "null".
sQLDBLStatementNotAllowed	M+P	The RDA client requested the execution of an RDA SQL statement that is not permitted by the RDA SQL Specialization as specified in Table 12 on page 33.
sQLUsageModeViolation	M+P	The sQLUsageMode attribute of the opened data resource entity identified by the dataResourceHandle parameter has a value of "retrieval" and the RDA SQL statement will modify one or more objects within the SQL database resource.

4.2.7.2 Stored Execution DBL functional unit**4.2.7.2.1 R-define DBL Service****Entity Manipulation Rules**

The following defines the constraints on the initial values of the additional SQL Specialization attributes for the defined DBL entity.

Attribute	Initial Value
sQLDBLStatement	The sQLDBLStatement parameter value in the R-DefineDBL indication primitive.
sQLDBLArgumentSpecification	The sQLDBLArgumentSpecification parameter value on the R-DefineDBL indication primitive.
sQLDBLResultSpecification	The sQLDBLResultSpecification parameter value on the R-DefineDBL indication primitive.

Result Rules

If a result is returned then the result parameters shall satisfy the following constraints:

Result Parameter	Constraints
sQLDBLException	<p>If the RDA SQL statement contained in the sQLBLStatement parameter is not permitted by an implementation claiming conformance to ISO/IEC 9075 (Database Language SQL) at the level of conformance specified by the sQLConformanceLevel attribute of the opened data resource entity, an sQLDBLException may be returned.</p> <p>If the sQLConformanceLevel attribute of the opened data resource entity has an <SQL edition> value of <1987> or <1989>, an sQLDBLException shall contain an sQLCODE parameter, otherwise it shall contain an sQLSTATE parameter. Applicable sQLCODE values of the sQLCODE parameter are defined in conformance with ISO/IEC 9075:1989, applicable sQLSTATE values of the sQLSTATE parameter are defined in conformance with ISO/IEC 9075:1992.</p>

The following table defines R-DefinedDBL use of SQL argument and result parameters.

Table 13. R-DefinedDBL use of SQL argument and result parameters		
RDA SQL statement to be Defined	ArgSpec	ResSpec
<close statement>		
<commit statement> ¹		
<declare cursor>	C→S(H) ⁴	
<delete statement: positioned>		
<delete statement: searched>	C→S(H)	
<fetch statement>		C→S ²
<insert statement>	C→S(H)	
<open statement>	C→S(H) ³	
<rollback statement> ¹		
<select statement: single row>	C→S(H)	C→S ²
<update statement: positioned>	C→S(H)	
<update statement: searched>	C→S(H)	
<schema definition>		
<table definition>		
<view definition>		
<drop schema statement>		
<alter table statement>		
<drop table statement>		
<drop view statement>		
<domain definition>		
<alter domain statement>		
<drop domain statement>		
<assertion definition>		
<drop assertion statement>		
<revoke statement>		
<temporary table declaration>		
<grant statement>		
<allocate cursor statement>	C→S(H)	
<allocate descriptor statement> ⁶	C→S(H)	
<deallocate descriptor statement> ⁶	C→S(H)	
<deallocate prepared statement>	C→S(H)	

<describe statement>	C→S(H)	
<dynamic close statement>	C→S(H)	
<dynamic declare cursor>		
<dynamic delete statement: positioned>	C→S(H)	
<dynamic fetch statement>	C→S(H)	C→S ²
<dynamic open statement>	C→S(H) ⁴	
<dynamic update statement: positioned>	C→S(H)	
<execute statement>	C→S(H)	C→S(H)
<execute immediate statement>	C→S(H)	
<get descriptor statement> ⁶	C→S(H)	
<preparable dynamic delete statement: positioned> ¹		
<preparable dynamic update statement: positioned> ¹		
<prepare statement>	C→S(H)	
<set descriptor statement> ⁶	C→S(H)	
<connect statement> ⁵		
<set connection statement> ⁵		
<disconnect statement> ⁵		
<get diagnostics statement> ¹		
<character set definition>		
<drop character set statement>		
<collation definition>		
<drop collation statement>		
<translation definition>		
<drop translation statement>		
<set catalog statement>		
<set schema statement>	C→S(H)	
<set constraints mode statement>		
<set transaction statement>		
<direct implementation-defined statement>	C→S(H)	C←S
Legend:		
ArgSpec The sQLDBLArgumentSpecification parameter.		
ResSpec The sQLDBLResultSpecification parameter.		
C→S Parameter supplied by the RDA client.		
C←S Parameter returned by the RDA server.		
(H) The RDA client must set this parameter if the RDA SQL statement contains host variables, except as noted below.		
blank cells unspecified (i.e. not subject to conformance testing)		
Notes:		
1. See the sQLDBLStatementNotAllowed error under the Error Rules below.		
2. The RDA client may optionally send the ResSpec to be used when the RDA SQL statement is subsequently invoked (via an R-InvokeDBL operation).		
3. The RDA client may optionally send the ArgSpec. If it is not specified, the RDA server shall use the ArgSpec sent on the previously executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <declare cursor> statement that used the same cursor name.		
4. The RDA client may optionally send the ArgSpec. If it is not specified, the RDA server shall use the ArgSpec sent on the previously executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <allocate cursor statement> or <dynamic declare cursor> that used the same <dynamic cursor name>.		

5. If the content of sQLDBLStatement performs connection management (eg. a <connect statement>, <set connection statement> or <disconnect statement>), then the RDA server shall return the error sQLDBLStatementNotAllowed. A <connect statement> or <disconnect statement> may result in the RDA client invoking the Dialogue Management services provided by RDA or TP depending on the application context chosen, and the Resource Handling services provided by RDA. A <set connection statement> may result in the RDA client invoking the Resource Handling services provided by RDA.
6. If the opened data resource was opened with an sQLDescriptorStatements value of «never» then the RDA server shall return the error sQLDBLStatementNotAllowed

Error Rules

Error	Type	Predicate
hostIdentifierError	M+P	The <host Identifier> of an <embedded variable name> of the RDA SQL statement is not "H".
sQLDBLNoCharSet	M+P	The sQLDBLArgumentSpecification or the sQLDBLResultSpecification contains a descriptor for character data without the explicit specification of its character set or the sQLDBLStatement contains no explicit specification of its character set, and the charSet attribute of the opened data resource entity identified by the dataResourceHandle parameter is "null".
sQLDBLStatementNotAllowed	M+P	The RDA client requested the execution of an RDA SQL statement that is not permitted by the RDA SQL Specialization, as specified in Table 13 on page 37.
sQLUsageModeViolation	M+P	The sQLUsageMode attribute of the opened data resource entity identified by the dataResourceHandle parameter has a value of «retrieval» and the RDA SQL statement will modify one or more objects within the SQL database resource.

4.2.7.2.2 R-InvokeDBL Service

The following rules shall be observed by the RDA server:

- Execution of the operation uses the sQLDBLStatement, sQLDBLArgumentSpecification and sQLDBLResultSpecification attributes of the defined DBL entity identified by the commandHandle parameter.
- An error shall be returned if no execution of the RDA SQL statement completes. Otherwise, the result shall be returned. The result includes one or more sQLDBLExceptions depending on the repetitionCount or the number of sQLDBLArgumentValues in the listOfSQLDBLArgumentValues.
- The effect of execution of RDA SQL statements by the database server is exactly the same as if they were embedded in a host program local to the SQL database resource. In addition, a <declare cursor> RDA SQL statement containing a <cursor name> must have been executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) at the RDA server prior to invoking any <open statement> that uses the same <cursor name>.
- For each execution of the RDA SQL statement, an sQLDBLException returned by the SQL database resource shall be included in the result. The sQLDBLResult values, if returned by the SQL database resource, shall also be included.

Entity Manipulation Rules

No additional entity manipulation rules are required for the RDA SQL Specialization.

Result Rules

If a result is returned then the result parameters shall satisfy the following constraints:

Result Parameter	Constraints
charSet	If the serverDefCharSet attribute is "null", each character string in the sQLDBLResultValues parameter of the R-InvokeDBL response primitives shall be accompanied by an object identifier in the charSet parameter in the corresponding item of listOfSQLDataType Descriptor.
sQLDBLException	<p>If the RDA SQL statement contained in the sQLBLStatement attribute of the defined DBL entity is not permitted by an implementation claiming conformance to ISO/IEC 9075 (Database Language SQL) at the level of conformance specified by the sQLConformanceLevel attribute of the opened data resource entity, an sQLDBLException may be returned.</p> <p>If the sQLConformanceLevel attribute of the RDA opened data resource entity has an <SQL edition> of <1987> or <1989>, an sQLDBLException shall contain an sQLCODE parameter, otherwise it shall contain an sQLSTATE parameter. Applicable sQLCODE values of the sQLCODE parameter are defined in conformance with ISO/IEC 9075:1989, applicable sQLSTATE values of the sQLSTATE parameter are defined in conformance with ISO/IEC 9075:1992.</p> <p>If the sQLDiagnosticsRequested attribute of the opened data resource entity has a value of "always" and the sQLConformanceLevel attribute of the opened data resource entity has an <SQL conformance> value of 1, Intermediate(1), 2, or High(2), then sQLDiagnostics shall be returned by the RDA server. If the sQLDiagnosticsRequested attribute of the opened data resource entity has a value of onRequest and the sQLConformanceLevel attribute of the opened data resource entity has an <SQL conformance> value of 1, Intermediate(1), 2, or High(2), and the returnSQLDiagnostics parameter of the R-InvokeDBL indication has a value of "true", then sQLDiagnostics shall be returned by the RDA server. In all other cases, sQLDiagnostics shall not be returned.</p>
colName	This parameter shall be specified within each item of listOfSQLDataTypeDescriptor that is part of an sQLDBLResultSpecification, otherwise it shall not be specified.

The following table defines R-InvokeDBL use of SQL argument and result parameters.

RDA SQL statement to be Invoked	ArgVal	ResSpec	ResVal
<close statement>			
<commit statement>			
<declare cursor>			
<delete statement: positioned>			
<delete statement: searched>	C→S(H)		
<fetch statement>		1	C←S
<insert statement>	C→S(H)		
<open statement>	C→S(H)	C→S ³	
<rollback statement>			
<select statement: single row>	C→S(H)	C→S ²	C←S

<update statement: positioned>	C→S(H)		
<update statement: searched>	C→S(H)		
<schema definition>			
<table definition>			
<view definition>			
<drop schema statement>			
<alter table statement>			
<drop table statement>			
<drop view statement>			
<domain definition>			
<alter domain statement>			
<drop domain statement>			
<assertion definition>			
<drop assertion statement>			
<revoke statement>			
<temporary table declaration>			
<grant statement>			
<allocate cursor statement>	C→S(H)		
<allocate descriptor statement>			
<deallocate descriptor statement>			
<deallocate prepared statement>	C→S(H)		
<describe statement>	C→S(H)	C←S ⁴	
<dynamic close statement>	C→S(H)		
<dynamic declare cursor>			
<dynamic delete statement: positioned>	C→S(H)		
<dynamic fetch statement>	C→S(H)	⁵	C←S
<dynamic open statement>	C→S(H)	C←S ⁶	
<dynamic update statement: positioned>	C→S(H)		
<execute statement>	C→S(H)	C→S(H)	C←S
<execute immediate statement>	C→S(H)		
<allocate descriptor statement> ⁷		C→S(H)	
<deallocate descriptor statement> ⁷		C→S(H)	
<get descriptor statement> ⁷		C→S(H)	C←S
<set descriptor statement> ⁷		C→S(H)	
<preparable dynamic delete statement: positioned>			
<preparable dynamic update statement: positioned>			
<prepare statement>	C→S(H)		
<set descriptor statement>			
<character set definition>			
<drop character set statement>			
<collation definition>			
<drop collation statement>			
<translation definition>			
<drop translation statement>			
<set catalog statement>			
<set schema statement>	C→S(H)		
<set constraints mode statement>			
<set transaction statement>			
<direct implementation-defined statement>	C→S(H)	C→S	C←S

<p>Legend: ArgVal The sQLDBLArgumentValues parameter. ResSpec The sQLDBLResultSpecification parameter. ResVal The sQLDBLResultValues parameter. C→S Parameter supplied by the RDA client. C←S Parameter returned by the RDA server. (H) The RDA client must set this parameter if the RDA SQL statement contains host variables. blank cells unspecified (i.e. not subject to conformance testing)</p>		
<p>Notes:</p> <ol style="list-style-type: none"> The RDA server does not return any ResSpec, but uses the sQLDBLResultSpecification attribute of the defined DBL entity identified by the commandHandle parameter. If that attribute is "null", then the ResSpec returned on the execution (via an R-ExecuteDBL operation) or invocation (via an R-InvokeDBL operation) of the corresponding <open statement> is used. The ResSpec describes the ResVal that the subsequently executed (via an R-ExecuteDBL operation) or invoked (via an R-InvokeDBL operation) <fetch statement> using this cursor will return, but any ResSpec specified with the <fetch statement> overrides this ResSpec. If the sQLDBLResultSpecification attribute of the defined DBL entity identified by the commandHandle parameter is "null", then the RDA server shall return the ResSpec which describes the ResVal, if any. If the <describe statement> is a <describe output statement>, then the RDA server shall return a ResSpec only if the <SQL statement name> in the <describe statement> refers to a prepared statement that is a <dynamic select statement> or a <dynamic single row select statement>. The i-th sQLDataTypeDescriptor of the ResSpec shall contain the description of the i-th column of the table defined by the prepared statement. If the <describe statement> is a <describe input statement>, then the RDA server shall return a ResSpec only if the <SQL statement name> in the <describe statement> refers to a prepared statement that contains one or more <dynamic parameter specification>s. The i-th sQLDataTypeDescriptor of the ResSpec shall contain the description of the i-th <dynamic parameter specification> of the prepared statement. The RDA server does not return any ResSpec but uses the sQLDBLResultSpecification attribute of the defined DBL entity identified by the commandHandle parameter. If that attribute is null, then the ResSpec returned on the execution (via an R-ExecuteDBL operation) or invocation (via an R-InvokeDBL operation) of the corresponding <dynamic open statement> is used. The ResSpec describes the ResVal that subsequent execution (via an R-ExecuteDBL operation) or invocation (via an R-InvokeDBL operation) of a <dynamic fetch statement> using this cursor will return, but any ResSpec specified with the <dynamic fetch statement> on the R-DefinedDBL request overrides this ResSpec. If the opened data resource was opened with an sQLDescriptorStatements value of «never» then the RDA server shall return the error sQLDBLStatementNotAllowed 		

Error Rules

Error	Type	Predicate
rDATransactionNotOpen	M+P	The RDA client requested the execution of an RDA SQL statement and the rDATransactionStatus attribute of the RDA dialogue entity for the current RDA dialogue has the value of rDATransactionNotOpen.
sQLDBLArgumentCountMismatch	M+P	The number of items in listOfSQLValue within the sQLDBLArgumentValues parameter is not the same as the number of items in listOfSQLDataTypeDescriptor within the sQLDBLArgumentSpecification attribute of the defined DBL entity identified by the commandHandle parameter.

SQLDBLArgumentTypeMismatch	M+P	The type of one or more items in listOfSQLValue within the sQLDBLArgumentValues parameter is not the same as the corresponding item (or items) in listOfSQLDataTypeDescriptor within the sQLDBLArgumentSpecification attribute of the defined DBL entity identified by the commandHandle parameter.
SQLUsageModeViolation	M+P	The sQLUsageMode attribute of the opened data resource entity identified by the dataResourceHandle parameter has a value of «retrieval» and the RDA SQL statement will modify one of more objects within the SQL database resource.

4.2.7.2.3 R-DropDBL Service

No additional server execution rules are required for the RDA SQL Specialization.

4.3 Structure and Encoding of RDA SQL APDUs

The RDA SQL data types are defined in this subclause using the ASN.1 notation specified in ISO/IEC 8824:1990 (ASN.1). The set of these RDA SQL APDUs defines the abstract syntax of the RDA SQL Presentation context. The transfer syntax for these RDA SQL APDUs is defined in the presentation context for the particular presentation connection.

4.3.1 Abstract Syntax Name

For support of the levels of ISO/IEC 9075:1992 having an object identifier with an <SQL edition> value of <1987>, with an <SQL edition> value of <1989>, or with an <SQL edition> value of <1992> and an <SQL conformance> value of <low>, this part of ISO/IEC 9579 assigns the ASN.1 object identifier value { iso standard rda (9579) part-2 (2) abstract-syntax (1) version-1 (1) }

as an abstract syntax name for the set of presentation data values, each of which is a value of the ASN.1 type ISO9579-RDASQL.RDA-APDU. The corresponding ASN.1 object descriptor value shall be «RDA-SQL-ABSTRACT-SYNTAX-V1»

For support of the levels of ISO/IEC 9075:1992 having an object identifier with an <SQL edition> value of <1992> and an <SQL conformance> value of <intermediate> or <high>, this part of ISO/IEC 9579 assigns the ASN.1 object identifier value

{ iso standard rda (9579) part-2 (2) abstract-syntax (1) version-2 (2) }

as an abstract syntax name for the set of presentation data values, each of which is a value of the ASN.1 type ISO9579-RDASQL.RDA-APDU.

Note: The abstract syntax name { iso standard rda (9579) part-2 (2) abstract-syntax (1) version-2 (2) } is also valid for support of the levels of ISO/IEC 9075:1992 having an object identifier with an <SQL edition> value of <1987>, with an <SQL edition> value of <1989>, and with an <SQL edition> value of <1992> and an <SQL conformance> value of <low>. It is intended that the abstract syntax name { iso standard rda (9579) part-2 (2) abstract-syntax (1) version-1 (1) } will be removed at a later date from a revised version of this international standard.

The corresponding ASN.1 object descriptor value shall be «RDA-SQL-ABSTRACT-SYNTAX-V2»

The ASN.1 object identifier and object descriptor values

{ joint-iso-ccitt asn1 (1) basic-encoding (1) }

and

«Basic Encoding of a single ASN.1 type»

(assigned to an information object in ISO 8825:1990) can be used as a transfer syntax name with this abstract syntax name.

4.3.2 ASN.1 Module for RDA SQL Specialization ASE

The following represents the ASN.1 Module for the RDA SQL Specialization ASE and spans a number of pages.

RDA SQL Specialization ASE ASN.1 Module:

ISO9579-RDASQL { iso standard rda (9579) part-2 (2) module (0) version-2 (2) }

```

-- *****
--
DEFINITIONS IMPLICIT TAGS ::= BEGIN
-- *****
--
IMPORTS
--
  AP-title,
  AE-qualifier,
  AP-invocation-identifier,
  AE-invocation-identifier
  FROM ACSE-1 { joint-iso-ccitt standard acse (8650) }
--
;
-- *****
--
-- RDA SQL APDU's
--
-- top level APDU CHOICE
RDA-APDU ::= CHOICE
{
  r-Initialize-RI [0] R-Initialize-RI,
  r-Initialize-RC [1] R-Initialize-RC,
  r-Synchronize-RI [2] R-Synchronize-RI,
  r-Terminate-RI [3] R-Terminate-RI,
  r-Terminate-RC [4] R-Terminate-RC,
  r-BeginTransaction-RI [5] R-BeginTransaction-RI,
  r-BeginTransaction-RC [6] R-BeginTransaction-RC,
  r-Commit-RI [7] R-Commit-RI,
  r-Commit-RC [8] R-Commit-RC,
  r-Rollback-RI [9] R-Rollback-RI,
  r-Rollback-RC [10] R-Rollback-RC,
  r-Cancel-RI [11] R-Cancel-RI,
  r-Cancel-RC [12] R-Cancel-RC,
  r-Status-RI [13] R-Status-RI,
  r-Status-RC [14] R-Status-RC,
  r-Open-RI [15] R-Open-RI,
  r-Open-RC [16] R-Open-RC,
  r-Close-RI [17] R-Close-RI,
  r-Close-RC [18] R-Close-RC,
  r-ExecuteDBL-RI [19] R-ExecuteDBL-RI,
  r-ExecuteDBL-RC [20] R-ExecuteDBL-RC,
  r-DefineDBL-RI [21] R-DefineDBL-RI,
  r-DefineDBL-RC [22] R-DefineDBL-RC,
  r-InvokeDBL-RI [23] R-InvokeDBL-RI,
  r-InvokeDBL-RC [24] R-InvokeDBL-RC,
  r-DropDBL-RI [25] R-DropDBL-RI,
  r-DropDBL-RC [26] R-DropDBL-RC,
}
-- individual APDU definitions.
R-Initialize-RI ::= SEQUENCE
{
  operationID
  r-Initialize-req
}
R-Initialize-RC ::= SEQUENCE
{
  operationID
  res-or-err
  {
    r-Initialize-res
    r-Initialize-err
  }
}
R-Synchronize-RI ::= SEQUENCE
{
}
R-Terminate-RI ::= SEQUENCE
{
  operationID
  r-Terminate-req
}

```

R-Terminate-RC { operationID res-or-err { r-Terminate-res r-Terminate-err } }	::= SEQUENCE OperationID, CHOICE [0] R-Terminate-Result, [1] R-Terminate-Error
R-BeginTransaction-RI { operationID r-BeginTransaction-req }	::= SEQUENCE OperationID, [0] NULL
R-BeginTransaction-RC { operationID r-BeginTransaction-err }	::= SEQUENCE OperationID, [0] R-BeginTransaction-Error
R-Commit-RI { operationID r-Commit-req }	::= SEQUENCE OperationID, [0] NULL
R-Commit-RC { operationID res-or-err { r-Commit-res r-Commit-err } }	::= SEQUENCE OperationID, CHOICE [0] R-Commit-Result, [1] R-Commit-Error
R-Rollback-RI { operationID r-Rollback-req }	::= SEQUENCE OperationID, [0] NULL
R-Rollback-RC { operationID res-or-err { r-Rollback-res r-Rollback-err } }	::= SEQUENCE OperationID, CHOICE [0] R-Rollback-Result, [1] R-Rollback-Error
R-Cancel-RI { operationID r-Cancel-req }	::= SEQUENCE OperationID, [0] R-Cancel-Request
R-Cancel-RC { operationID res-or-err { r-Cancel-res r-Cancel-err } }	::= SEQUENCE OperationID, CHOICE [0] R-Cancel-Result, [1] R-Cancel-Error
R-Status-RI { operationID r-Status-req }	::= SEQUENCE OperationID, [0] R-Status-Request
R-Status-RC { operationID res-or-err { r-Status-res r-Status-err } }	::= SEQUENCE OperationID, CHOICE [0] R-Status-Result, [1] R-Status-Error
R-Open-RI { operationID r-Open-req }	::= SEQUENCE OperationID, [0] R-Open-Request
R-Open-RC { operationID res-or-err { r-Open-res r-Open-err } }	::= SEQUENCE OperationID, CHOICE [0] R-Open-Result, [1] R-Open-Error

}		
R-Close-RI	::= SEQUENCE	
{ operationID	OperationID,	
r-Close-req	[0] R-Close-Request	
}		
R-Close-RC	::= SEQUENCE	
{ operationID	OperationID,	
res-or-err	CHOICE	
{ r-Close-res	[0] R-Close-Result,	
r-Close-err	[1] R-Close-Error	
}		
}		
R-ExecuteDBL-RI	::= SEQUENCE	
{ operationID	OperationID,	
r-ExecuteDBL-req	[0] R-ExecuteDBL-Request	
}		
R-ExecuteDBL-RC	::= SEQUENCE	
{ operationID	OperationID,	
res-or-err	CHOICE	
{ r-ExecuteDBL-res	[0] R-ExecuteDBL-Result,	
r-ExecuteDBL-err	[1] R-ExecuteDBL-Error	
}		
}		
R-DefineDBL-RI	::= SEQUENCE	
{ operationID	OperationID,	
r-DefineDBL-req	[0] R-DefineDBL-Request	
}		
R-DefineDBL-RC	::= SEQUENCE	
{ operationID	OperationID,	
res-or-err	CHOICE	
{ r-DefineDBL-res	[0] R-DefineDBL-Result,	
r-DefineDBL-err	[1] R-DefineDBL-Error	
}		
}		
R-InvokeDBL-RI	::= SEQUENCE	
{ operationID	OperationID,	
r-InvokeDBL-req	[0] R-InvokeDBL-Request	
}		
R-InvokeDBL-RC	::= SEQUENCE	
{ operationID	OperationID,	
res-or-err	CHOICE	
{ r-InvokeDBL-res	[0] R-InvokeDBL-Result,	
r-InvokeDBL-err	[1] R-InvokeDBL-Error	
}		
}		
R-DropDBL-RI	::= SEQUENCE	
{ operationID	OperationID,	
r-DropDBL-req	[0] R-DropDBL-Request	
}		
R-DropDBL-RC	::= SEQUENCE	
{ operationID	OperationID,	
res-or-err	CHOICE	
{ r-DropDBL-res	[0] R-DropDBL-Result,	
r-DropDBL-err	[1] R-DropDBL-Error	
}		
}		

R-Initialize-Request	::= SEQUENCE	
{ dialogueIDSuffix	[0] DialogueIDSuffix,	
identityOfUser	CHOICE	
{ vstring	[1] VisibleString,	
rstring	[5] RDACCharString	
}		
}		
userAuthenticationData	[2] AuthenticationData OPTIONAL,	
controlServiceDataRequested	[3] BOOLEAN DEFAULT FALSE,	
functionalUnitsRequested	[4] FunctionalUnits,	
sqlInitializeArgument	[30] SQLInitializeArgument OPTIONAL	
}		

R-Initialize-Result	::= SEQUENCE
{ controlServiceData	[0] SEQUENCE
{ controlServicesAllowed	[0] BOOLEAN DEFAULT TRUE,
controlAuthenticationData	[1] AuthenticationData OPTIONAL
}	OPTIONAL,
functionalUnitsAllowed	[1] FunctionalUnits,
sQLInitializeResult	[30] SQLInitializeResult OPTIONAL
}	
R-Initialize-Error	::= CHOICE
{ accessControlViolation	AccessControlViolation,
duplicateDialogueID	DuplicateDialogueID,
invalidSequence	InvalidSequence,
operationAborted	OperationAborted,
userAuthenticationFailure	UserAuthenticationFailure,
characterSetNotSupported	CharacterSetNotSupported
}	
.. *****	
R-Terminate-Request	::= NULL
R-Terminate-Result	::= NULL
R-Terminate-Error	::= CHOICE
{ duplicateOperationID	DuplicateOperationID,
invalidSequence	InvalidSequence,
operationAborted	OperationAborted,
serviceNotNegotiated	ServiceNotNegotiated
}	
.. *****	
R-BeginTransaction-Error	::= CHOICE
{ duplicateOperationID	DuplicateOperationID,
invalidSequence	InvalidSequence,
operationAborted	OperationAborted,
serviceNotNegotiated	ServiceNotNegotiated
}	
.. *****	
R-Commit-Result	::= SEQUENCE
{ transactionResult	[0] ENUMERATED
{ committed	(0),
rolledBack	(1)
},	
sQLCommitException	[1] SQLDBLException OPTIONAL
}	
R-Commit-Error	::= CHOICE
{ duplicateOperationID	DuplicateOperationID,
invalidSequence	InvalidSequence
}	
.. *****	
R-Rollback-Result	::= SEQUENCE
{ sQLRollbackException	[1] SQLDBLException OPTIONAL
}	
R-Rollback-Error	::= CHOICE
{ duplicateOperationID	DuplicateOperationID,
invalidSequence	InvalidSequence
}	
.. *****	
R-Cancel-Request	::= SEQUENCE
{ controlledDialogue	[0] SEQUENCE
{ dialogueIDClientInvocation	[0] DialogueIDClientInvocation OPTIONAL,
dialogueIDSuffix	[1] DialogueIDSuffix,
controlAuthenticationData	[2] AuthenticationData
}	OPTIONAL,
listOfOperationID	[1] SEQUENCE OF OperationID OPTIONAL
}	
R-Cancel-Result	::= NULL
R-Cancel-Error	::= CHOICE
{ controlAuthenticationFailure	ControlAuthenticationFailure,
controlServicesNotAllowed	ControlServicesNotAllowed,
dialogueIDUnknown	DialogueIDUnknown,

```

duplicateOperationID          DuplicateOperationID,
invalidSequence              InvalidSequence,
operationAborted             OperationAborted,
serviceNotNegotiated         ServiceNotNegotiated,
characterSetNotSupported     CharacterSetNotSupported
}
-- *****

R-Status-Request              ::= SEQUENCE
{
  controlledDialogue          [0] SEQUENCE
  {
    dialogueIDClientInvocation OPTIONAL,
    dialogueIDSuffix          [1] DialogueIDSuffix,
    controlAuthenticationData [2] AuthenticationData
  }
  listOfOperationID          [1] SEQUENCE OF OperationID OPTIONAL
}

R-Status-Result              ::= SEQUENCE
{
  listOfStatusInformation    [0] SEQUENCE OF StatusInformation OPTIONAL
}

R-Status-Error               ::= CHOICE
{
  controlAuthenticationFailure,
  controlServicesNotAllowed,
  dialogueIDUnknown,
  duplicateOperationID,
  invalidSequence,
  operationAborted,
  serviceNotNegotiated,
  characterSetNotSupported
}
-- *****

R-Open-Request                ::= SEQUENCE
{
  dataResourceHandle          [0] DataResourceHandle,
  dataResourceName            CHOICE
  {
    vstring                   [2] VisibleString,
    rstring                   [5] RDACHarString
  }
  SQLAccessControlData        [3] AccessControlData OPTIONAL,
  SQLUsageMode                [4] SQLUsageMode DEFAULT retrieval,
  SQLOpenArgument             [30] SQLOpenArgument OPTIONAL
}

R-Open-Result                 ::= SEQUENCE
{
  sQLOpenResult               [30] SQLOpenResult OPTIONAL
}

R-Open-Error                  ::= CHOICE
{
  dataResourceAlreadyOpen,
  dataResourceNameNotSpecified,
  dataResourceNotAvailable,
  dataResourceUnknown,
  duplicateDataResourceHandle,
  duplicateOperationID,
  invalidSequence,
  operationAborted,
  operationCancelled,
  serviceNotNegotiated,
  sQLOpenError,
  characterSetNotSupported
}
-- *****

R-Close-Request               ::= SEQUENCE
{
  listOfDataResourceHandle    [0] SEQUENCE OF DataResourceHandle OPTIONAL
}

R-Close-Result                ::= SEQUENCE
{
  listOfCloseExceptions       [0] SEQUENCE OF CloseException OPTIONAL
}

R-Close-Error                 ::= CHOICE
{
  duplicateOperationID,
  invalidSequence,
  operationAborted,
  operationCancelled,
  serviceNotNegotiated
}

```

```

    sQLCloseError
  }
}
-- *****

R-ExecuteDBL-Request ::= SEQUENCE
{
  dataResourceHandle [0] DataResourceHandle OPTIONAL,
  sQLDBLStatement [1] SQLDBLStatement,
  sQLDBLArgumentSpecification [2] SQLDBLArgumentSpecification OPTIONAL,
  sQLDBLResultSpecification [3] SQLDBLResultSpecification OPTIONAL,
  dbLArguments CHOICE
  {
    singleArgument [4] SEQUENCE
    {
      repetitionCount [0] INTEGER DEFAULT 1,
      sQLDBLArgumentValues [1] SQLDBLArgumentValues OPTIONAL
    },
    multipleArgument [5] SEQUENCE
    {
      listOfSQLDBLArgumentValues [0] SEQUENCE OF SQLDBLArgumentValues
    }
  }
}
returnSQLDiagnostics [6] BOOLEAN DEFAULT FALSE
}

R-ExecuteDBL-Result ::= SEQUENCE
{
  sQLDBLResultSpecification [1] SQLDBLResultSpecification OPTIONAL,
  listOfResultValues [2] SEQUENCE OF ResultValues OPTIONAL
}

R-ExecuteDBL-Error ::= CHOICE
{
  badRepetitionCount,
  dataResourceHandleNotSpecified,
  dataResourceHandleUnknown,
  duplicateOperationID,
  invalidSequence,
  noDataResourceAvailable,
  operationAborted,
  operationCancelled,
  serviceNotNegotiated,
  transactionRolledBack,
  sQLExecuteDBLError,
  characterSetNotSupported
}
}
-- *****

R-DefineDBL-Request ::= SEQUENCE
{
  commandHandle [0] CommandHandle,
  dataResourceHandle [1] DataResourceHandle OPTIONAL,
  sQLDBLStatement [2] SQLDBLStatement,
  sQLDBLArgumentSpecification [3] SQLDBLArgumentSpecification OPTIONAL,
  sQLDBLResultSpecification [4] SQLDBLResultSpecification OPTIONAL
}

R-DefineDBL-Result ::= SEQUENCE
{
  sQLDBLResultSpecification [0] SQLDBLResultSpecification OPTIONAL,
  sQLDBLException [1] SQLDBLException OPTIONAL
}

R-DefineDBL-Error ::= CHOICE
{
  dataResourceHandleNotSpecified,
  dataResourceHandleUnknown,
  duplicateCommandHandle,
  duplicateOperationID,
  invalidSequence,
  noDataResourceAvailable,
  operationAborted,
  operationCancelled,
  serviceNotNegotiated,
  sQLDefineDBLError,
  characterSetNotSupported
}
}
-- *****

R-InvokeDBL-Request ::= SEQUENCE
{
  commandHandle [0] CommandHandle,
  dbLArguments CHOICE
  {
    singleArgument [1] SEQUENCE
    {
      repetitionCount [0] INTEGER DEFAULT 1,
      sQLDBLArgumentValues [1] SQLDBLArgumentValues OPTIONAL
    },
    multipleArgument [2] SEQUENCE

```