INTERNATIONAL STANDARD ISO/IEC 9075-10:2000
TECHNICAL CORRIGENDUM 1

Published 2003-06-01

Information technology — Database languages — SQL —

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

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Langages de base de données — SQL —

Partie 10: Liaisons de langage objet (SQL/OLB)

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 9075-10:2000 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

Statement of purpose for rationale:

A statement indicating the rationale for each change to ISO/IEC 9075 is included. This is to inform the users of that standard as to the reason why it was judged necessary to change the original wording. In many cases the reason is editorial or to clarify the wording; in some cases it is to correct an error or an omission in the original wording.

Notes on numbering:

Where this Corrigendum introduces new Syntax, Access, General and Conformance Rules, the new rules have been numbered as follows:
Rules inserted between, for example, Rules 7) and 8) are numbered 7.1), 7.2), etc. [or 7) a.1), 7) a.2), etc.]. Those inserted before Rule 1) are numbered 0.1), 0.2), etc.
Where this Corrigendum introduces new Subclauses, the new subclauses have been numbered as follows:
Subclauses inserted between, for example, Subclause 4.3.2 and 4.3.3 are numbered 4.3.2a, 4.3.2b, etc.
Those inserted before, for example, 4.3.1 are numbered 4.3.0, 4.3.0a, etc.

ICS 35.060 **Ref. No. ISO/IEC 9075-10:2000/Cor.1:2003(E)**

Contents

Information technology — Database languages — SQL —

Part 10: Object Language Bindings (SQL/OLB)

TECHNICAL CORRIGENDUM 1

3.3.2.1 Clause, subclause and table relationships

1. *Rationale: Remove incorrect replacement paragraph.*
   Source: ICN-054R2

Replace the following row from Table 1 "Clause, Subclause, and Table relationships":

Clause, Subclause, or Table in this part of ISO/IEC 9075	**Corresponding Clause, Subclause, or Table from another part**	**Part containing correspondence**
Subclause 4.4.2, ''Named character sets''	Subclause 4.2.4, "Named character sets"	ISO/IEC 9075-2

with:

Clause, Subclause, or Table in this part of ISO/IEC 9075	**Corresponding Clause, Subclause, or Table from another part**	**Part containing correspondence**
Subclause 4.4.2, ''Character repertoires''	Subclause 4.2.3a, ''Character repertoires''	ISO/IEC 9075-2

4.4.1 UNICODE support

1. *Rationale: Correct reference to character set.*

Replace the 1st paragraph with:

Java relies on the UNICODE character set [UNICODE] (also known as ISO/IEC 10646-1 [UCS]) for String data and for identifiers. That allows Java to represent most character data in a uniform way. [SQL] defines support for UNICODE through its UTF8, UTF16, and UTF32 character sets, which represent different encodings for UNICODE character data. When character data is moved between an SQL-server and an SQL/OLB host program, an SQL/OLB implementation that provides SQL character set support for UTF8, UTF16, and/or UCS2 is required to support implicit conversion between Java string data and the supported UNICODE encodings. Any support for implicit conversions to and from character sets other than UNICODE is implementation-defined. Because of Java's reliance on UNICODE as an internal representation for character data, the SQL/OLB specification does not define support for host variables that hold character data based on character sets other than UNICODE.

4.4.2 Named character sets

1. *Rationale: Remove incorrect replacement paragraph.*

Replace the subclause with:

4.4.2 Character repertoires

— Augment 1st bullet of the 3rd paragraph If <embedded SQL host program> immediately contains an <embedded SQL Java program>, then <SQL special character> shall include the <number sign> (#) in addition to the characters that it otherwise required to contain.

4.22.8 Cursor declaration

1. *Rationale: Remove the concept of returnability that could not be implemented because it was in conflict with concepts within JDBC.*

Replace the 3rd paragraph with:

When <assignment spec clause> immediately contains <query clause>, the <query clause> provides the implicit <declare cursor>'s <query expression> and optional <order by clause>. An implicit <declare cursor>'s <cursor returnability> is always WITH RETURN. The <Lval expression> ***LV*** immediately contained in <assignment clause> either refers to an object of a generated iterator class or to an object the type of which implements sqlj.runtime.ResultSetIterator. When ***LV*** refers to an object of a generated iterator class, the associated <iterator declaration clause>'s <declaration with list> specification of the iterator's sensitivity, scrollability, holdability, and updateColumns respectively provide the implicit <declare cursor>'s <cursor sensitivity>, <cursor scrollability>, <cursor holdability>, and update <column name list> specifications.

5.1 <SQL terminal character>

Replace Syntax Rule 2) with:

2) Insert this SR If the character set SQL_TEXT does not include <number sign>, then <number sign> shall be immediately contained in an <SQL prefix> that is contained in an <embedded SQL Java program>.

10.6.8 <declaration with clause>

1. *Rationale: Remove the concept of returnability that could not be implemented because it was in conflict with concepts within JDBC.*

In the Format, replace the production for <predefined iterator with keyword> with:

```
<predefined iterator with keyword> ::=
       sensitivity
     | holdability
     | updateColumns
```