INTERNATIONAL STANDARD

ISO/IEC 10728

First edition
1993-04-15

Information technology — Information Resource Dictionary System (IRDS) Services Interface

Technologies de l'information — Interface de services du gestionnaire de ressources du système d'informations (IRDS)

Reference number
ISO/IEC 10728:1993(E)

Table of Contents

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 10728 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Sub-Committee SC 21, *Information retrieval, transfer and management for open systems interconnection (OSI).*

Annex A forms an integral part of this International Standard. Annex B is for information only.

Introduction

This International Standard is one of a series of International Standards on Information Resource Dictionary Systems. ISO/IEC 10027 defines the context within which this International Standard is to be applied.

Information technology – Information Resource Dictionary System (IRDS) Services Interface

1 Scope

The IRDS series of International Standards specifies a software tool that can be used to describe and potentially control an enterprise's information resources. It defines the structure and part of the content of the data to be maintained at the IRD Definition Level, and the structure of the data to be maintained at the IRD Level. It also defines the services to be provided for maintaining and retrieving data at both levels. Further details of the IRDS series of standards are to be found in ISO/IEC 10027.

This International Standard specifies a Services Interface that gives any program full access to all IRDS services, through whatever external call interface is provided by the language in which the program is written. The body of this International Standard defines the semantics of this interface, and also specifies the language bindings for ISO Pascal (ISO 7185). Language bindings for other ISO standard programming languages are provided as separate standards.

This International Standard makes no assumptions about an implementation environment, and assumes no specific run-time or compile-time interfaces.

2 Normative references

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

ISO 3166: 1988; *Codes for the representation of names of countries*

ISO 7185: 1990; *Information Technology - Programming languages - Pascal.*

ISO/IEC 9075: 1992; *Information Technology - Database Languages - SQL.*

ISO/IEC 10027: 1990; *Information Technology - Information Resource Dictionary System (IRDS) - Framework.*

ISO/IEC 10032: 1993; *Information Technology - Reference Model of Data Management*

3 Definitions and abbreviations

NOTE - SQL terms are not defined here. When used in this International Standard, they have the meanings ascribed to them in ISO/IEC 9075. All IRDS terms used in this International Standard are fully defined here or in ISO/IEC 10027.

3.1 Terms defined or referenced in the IRDS Framework (ISO/IEC 10027) and used in this International Standard

The following terms are defined (or referenced) and used in the IRDS Framework. They are used in the same way in this International Standard.

3.1.1 client

3.1.2 Information Resource Dictionary (IRD)

3.1.3 Information Resource Dictionary System (IRDS)

3.1.4 IRD definition

3.1.5 IRD definition level

3.1.6 IRD definition schema

3.1.7 IRD level

3.1.8 IRD schema

3.1.9 level pair

3.1.10 real system

3.1.11 service

3.2 Terms defined in this International Standard

Where each term listed in this clause is introduced in a later clause of this International Standard, it is printed in bold type.

3.2.1 active: A state in which a dictionary is accessible to all relevant IRDS services. When an IRD is not active, only the Reactivate IRD service is applicable.

3.2.2 archived: A content status class that indicates data that is no longer in active use.

3.2.3 attribute: A characteristic of an object.

3.2.4 common table: A table that exists in each IRD Definition and each IRD.

3.2.5 content module: A collection of objects introduced into an IRD Definition or IRD at the same time and from the source, identified by a module name that indicates the source of the module.

3.2.6 context: A working set established by default or by user request within which IRDS services are performed.

3.2.7 controlled: A content status class that indicates data that is stable and not subject to change.

3.2.8 definition object: An object recorded at the IRD definition level that controls the data which may be present at the IRD level.

3.2.9 dictionary: An IRD Definition or IRD

3.2.10 environment table: A table that exists once in each IRD Definition, controlling the services provided on that IRD Definition and any associated IRDs.

3.2.11 implementation-defined: Behaviour not defined by this International Standard, but which shall be precisely defined by any conforming implementation

3.2.12 implementation-dependent: Behaviour not defined by this International Standard, and which an implementation is not required to define. Further, there is no requirement that such behaviour be consistent from case to case.

3.2.13 internal table: A table that exists once in each IRD Definition and each IRD, rows in which cannot be accessed by the object-related services in clause 9.

3.2.14 IRD-specific table: A table that exists only in the IRD Definition or a specific IRD, as part of the representation of the data structuring rules of a defined data modelling facility.

3.2.15 IRD content status: A user-defined attribute of a working set. Every value of IRD content status belongs to one of the three predefined content status classes. Each object version takes its IRD content status from the working set that contains it.

3.2.16 IRD content status class: One of three predefined sets of IRD content statuses: uncontrolled, controlled and archived.

3.2.17 IRD object: An object recorded at the IRD level.

3.2.18 IRD Schema Group: A collection of one or more IRD Schemas that completely defines what may exist at any time in an IRD.

3.2.19 IRDS database: An IRD definition and zero or more IRDs.

3.2.20 IRDS environment: An operational instance of an implementation of the IRDS Services Interface managing an IRDS database.

3.2.21 IRDS name: A name optionally assigned when an object is added to an IRD or a definition object is added to an IRD definition. If specified, the combination of IRDS name, variation name, working set name and working set version name shall be unique.

3.2.22 IRDS session: A temporary association between an IRDS user and an IRDS environment, during which the former requests services and the latter performs them.

3.2.23 IRDS user: An individual or group authorized to use the IRDS.

3.2.24 level independent service: A service that is equally applicable to the IRD Definition level and the IRD level.

3.2.25 level specific service: A service which only applies either to the IRD Definition level or to the IRD level, but not both.

3.2.26 materialization (of a working set): That collection of object versions that can be in the working table of a cursor opened on that working set.

3.2.27 name: A string of characters used to distinguish objects, either alone or in combination with other names.

3.2.28 non-versionable (of an object type): Indicates that only one version of an object of that type may exist at any time, in a non-versionable working set; of a working set, indicates that the working set may not be based on another working set or be used as the basis for another working set.

3.2.29 object: Any concept or thing of interest to an enterprise.

3.2.30 object type: A class of objects all of whose attributes belong to a common set of attribute types.

3.2.31 object version: A record of the information about an object that is current for some period of time in some information processing context.

3.2.32 reference path: A directed association from one working set to another which allows an object version in the first working set to reference object versions in the second working set. A reference path allows references only in the direction in which it is specified.

3.2.33 referenced table: The table to which the reference is made in a referential constraint.

3.2.34 referencing table: The table from which the reference is made in a referential constraint.

3.2.35 subtable: Let SUBT and SUPERT be tables. SUBT is defined to be a subtable of SUPERT if and only if every row of SUBT corresponds to one and only one row of SUPERT, and each row of SUPERT corresponds to at most one row of SUBT. SUPERT is defined to be a supertable of SUBT.

3.2.36 supertable: A table that has at least one subtable (see definition of subtable).

3.2.37 uncontrolled: A content status class that indicates data that is not stable.

3.2.38 variation name: A name used to identify significantly different variants of objects with the same IRDS name.

3.2.39 versionable (of an object type): Indicates that more than one version of an object of that type may exist at the same time, in different working sets; of a working set, indicates that the working set may not contain objects of non-versionable object types, may be based on another working set and may be used as the basis for another working set.

3.2.40 working set: A collection of versions of either definition objects or IRD objects, defined by an IRDS user as a unit for the purposes of change management, content status specification and access control.

3.3 Data Item Name abbreviations

The following list describes all of the abbreviations that are used in naming the columns of the SQL tables and the corresponding Pascal constants, types and variables:

Add = added
Arch = archived
Atr = attribute
Cls = class
Cntl = controlled
Col = column
Cur = cursor
Curr = current
Dcs = IRD content status
Def = definition
Dflt = default
Dflts = defaults
Dic = dictionary
Dom = domain
Id = identifier
Imp = implementation
Ind = indicator
Inst = installation
Int = integer
Ird = Information Resource Dictionary
Len = length
Lim = limit
Maint = maintained
Max = maximum
Min = minimum
Mod = modified
Nat = national
Num = number
Obj = object
Ref = reference
Ret = return
Scm = schema
Sep = separator
Sess = session
Spec = specification
Srv = service
Str = string
Tran = transaction
Txt = text
Ucntl = uncontrolled
Val = value
Var = variation
Ver = version
Wkg = working
Ws = working set

4 Conventions

This clause describes the conventions used within this International Standard to describe the facilities of an IRDS. These conventions are not themselves a subject of this standard. Several different specification conventions are used, for different purposes.

4.1 Specification of concepts and facilities

Within clause 5, diagrams are used, in conjunction with English description, to introduce some of the concepts and facilities that are formally defined later in this International Standard. Two types of diagram are used:

a) Data Structure Diagrams, described in 4.6;

b) Working Set Diagrams, described in 4.8.

4.2 Specification of data structures

In clause 6, the data structures to be managed by an IRDS Services Processor are specified using Database Language SQL (ISO/IEC 9075).

In general, this involves the use of tables to represent object types and columns to represent attribute types; certain data is also required for control purposes.

The use of SQL as a definition formalism in this International Standard does not imply any specific implementation approach. The user of the services provided at the IRDS Services Interface sees the data only in the terms defined in clause 8.

4.3 Specification of constraints - overview

Constraints on the possible values, or combinations of values, which may appear in the IRD are specified once only wherever possible. In order of preference, each constraint is specified:

a) within the formal data specification, in clause 6;

b) in the description of the relevant table, in clause 6;

c) in the description of the services that can be used to change the data, in clause 9.

The specification of constraints is described in detail in 4.7.

4.4 Specification of service data structures

In clause 8, the formats of the data structures associated with each service are described using Pascal (ISO 7185).

4.5 Specification of services

In clause 9, the services to be supported by an IRDS are specified using a combination of Pascal (ISO 7185) and English.

4.6 Data Structure Diagrams

The Data Structure Diagrams used in clause 5 illustrate tables and the constraints between the tables. A table is represented by a rectangular box.

Constraints between the tables are represented by lines between the boxes. Each line is considered to have two halves, each half associated with the box to which it is directly connected.

4.7 Specification of constraints - detail

4.7.1 Types of constraint

The following types of constraint are used within the formal data specification in clause 6:

a) Primary key constraints - which identify the primary key of each table.

b) Uniqueness constraints - which identify combinations of columns within a table whose values must be unique, when not wholly or partially null.

c) Referential constraints - which identify the dependencies of one table on another.

d) Check constraints - which allow the specification of many additional general constraints on the values or combinations of values which may appear in one or more rows in one or more tables.

In addition to their specification in clause 6, referential constraints are also illustrated diagrammatically in clause 5. The rest of this clause describes the diagramming conventions used for different types of referential constraint and the SQL syntax used to represent each type.

4.7.2 Overview of referential constraints

It is necessary to explain the type of referential constraints used before describing their representation.

A referential constraint exists between two tables A and B if at all times the values of some specified column or combination of columns, when not null, in each row in table B shall be equal to the values of a corresponding combination (in terms of number and data type) of columns in one and only one row in table A. Such constraints are further qualified as follows:

a) One-to-one - if only a single row in table B may reference any one row in table A.

b) One-to-many - if more than one row in table B can reference the same row in table A.

Orthogonal to the above classification of constraints, the following classification may also be made:

c) Optional at referencing table - if any of the columns in table B, which are used to reference table A, may be null; a row in which any of these columns is null is considered not to reference table A;

d) Optional at referenced table - if a row may exist in table A without being referenced from any row in table B;

e) Required at referencing table - if the referencing columns in table B may not be null. Every row in table B must reference a row in table A.

f) Required at referenced table - if a row in table A may exist only if there is a corresponding row in table B that references it.

The following basic constructs are used within the Data Structure Diagram to illustrate these constraints. Each half of the line that represents the constraint is treated separately in illustrating the characteristics of the constraint at each table.

a) Solid line - the constraint is required at that table;

b) Dashed line - the constraint is optional at that table;

c) "Crow's feet" at end of line - represents a one-to-many constraint. (Crow's feet are not allowed at both ends of a line.)

d) No "crow's feet" at either end - represents a one-to-one constraint.

The diagrammatic representations of some of the various constraint types listed above are now illustrated. In each case, an example is also given of how the relevant constraint would be expressed in clause 6, using SQL with or without accompanying English description.

4.7.3 Optional one-to-many referential constraint

Figure 1 illustrates the simplest (in terms of its SQL specification) constraint - an optional one-to-many referential constraint.

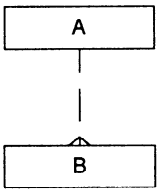

Figure 1 - Optional referential constraint

Table 1 shows the corresponding SQL syntax.

Table 1: SQL corresponding to Figure 1

```
CREATE TABLE A
( A1 IRDS_KEY PRIMARY KEY )

CREATE TABLE B
( B1 IRDS_KEY PRIMARY KEY,

  B2 IRDS_KEY

  CONSTRAINT constraint-name
    REFERENCES A
    ON DELETE referential-action )
```


In table 1 and similar tables, IRDS_KEY represents the name of a domain (see ISO/IEC 9075) which is used throughout the tables defined in clause 6. It is used here solely to make these examples look as similar as possible to the SQL used in clause 6.

The SQL syntax used in clause 6 will normally contain an ON DELETE clause, similar to that shown in Table 1. This clause specifies additional information that is not shown in the diagram - namely the action to be taken when an attempt is made to delete a row in table A that is referenced by a row in table B.

For an optional referential constraint, as shown in table 1, the referential action can be one of the following:

a) RESTRICT - (cannot be specified explicitly, but is implied if ON DELETE clause is not specified) which means the deletion will not be allowed;

b) CASCADE - which means the referencing rows will also be deleted.

c) SET NULL - which means the referencing columns will be nullified, thus removing the reference, but the row itself will not be deleted.

4.7.4 Required uni-directional one-to-many referential constraint

Figure 2 illustrates a required uni-directional one-to-many referential constraint. The solid half of the line connected to box B illustrates that this constraint is required from B to A.

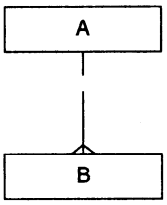

Figure 2 - Required referential constraint (one to many)

Table 2 shows an example of the SQL syntax to specify the constraint illustrated in figure 2. The addition of the NOT NULL clause on the referencing column B2 makes the constraint required from table B to table A.

Table 2 - SQL corresponding to figure 2

```
CREATE TABLE A
( A1 IRDS_KEY PRIMARY KEY )

CREATE TABLE B
( B1 IRDS_KEY PRIMARY KEY,

  B2 IRDS_KEY NOT NULL

  CONSTRAINT constraint-name
    REFERENCES A
    ON DELETE referential-action )
```

Because of the NOT NULL clause, SET NULL is not a valid referential action for a required referential constraint.

4.7.5 Required uni-directional one-to-one referential constraint

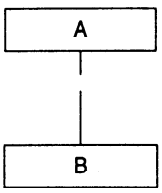

Figure 3 - Required referential constraint (one to one)

Figure 3 illustrates a required uni-directional one-to-one referential constraint. The removal of the crows feet from the line indicates in the diagram that this constraint is one-to-one.

Table 3 shows an example of the SQL syntax to specify the constraint illustrated in figure 3. The addition of a uniqueness constraint on column B2 enforces the singularity of the constraint.

Table 3 - SQL corresponding to figure 3

```
CREATE TABLE A
( A1 IRDS_KEY PRIMARY KEY )

CREATE TABLE B
( B1 IRDS_KEY PRIMARY KEY,

  B2 IRDS_KEY NOT NULL
  CONSTRAINT constraint-name
    REFERENCES A
    ON DELETE referential-action,

  UNIQUE (B2) )
```

4.7.6 Self-referencing tables

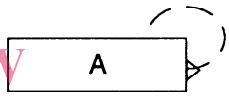

Figure 4 - Self-referencing constraint

Note that in any of the above cases A and B may be the same table, in which case the line representing the constraint is drawn as a circular arc between two points on the same box, as in figure 4, and the SQL referential constraint is between two columns, or groups of columns, in the same table, as in table 4.

Table 4: SQL corresponding to Figure 4

```
CREATE TABLE A
( A1 IRDS_KEY PRIMARY KEY

  A2 IRDS_KEY

  CONSTRAINT constraint-name
    REFERENCES A
    ON DELETE referential-action )
```

4.7.7 Required bi-directional referential constraint

Figure 5 illustrates the diagramming convention for a required bi-directional one-to-many referential constraint.