Technical Report

ISO/IEC TR 19583-21

Second edition
2025-03

Information technology — Concepts and usage of metadata —

Part 21: 11179-3, -31, -32 Data model in SQL

Technologies de l'information — Concepts et utilisation des métadonnées —

Partie 21: Modèle de données en SQL 11179-3, -31, -32

Reference number
ISO/IEC TR 19583-21:2025(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

This second edition cancels and replaces the first edition (ISO/IEC 19583-21:2022), which has been technically revised.

The main changes are as follows:

— examples have been added to instatiate the metamodel of the latest version of the ISO/IEC 11179 series (Annexes B and C have been added).

A list of all parts in the ISO/IEC 19583 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

The ISO/IEC 11179 series[1] provides a specification for a registry in which information about metadata can be recorded and maintained.

The metamodel to instantiate such a registry is expressed in text as a conceptual model. This conceptual model is illustrated with a series of diagrams which use the class diagram notation from the Unified Modeling Language (UML).[2][3]

Implementers and users of the registries described in ISO/IEC 11179 require further guidance to turn the conceptual models into concrete instantiations. This document provides a possible instantiation of the registry metamodel specified in ISO/IEC 11179-3[4], ISO/IEC 11179-31[5] and ISO/IEC 11179-32[6] using the SQL database language as specified in the ISO/IEC 9075 series.[7]

This specimen instantiation is provided to increase the understanding of ISO/IEC 11179-3, ISO/IEC 11179-31 and ISO/IEC 11179-32, and hence, to promote its adoption.

This document is not intended to replace the UML version but rather serves as a complement, providing SQL statements that describe the ISO/IEC 11179 metamodel. It aims to facilitate the application of ISO/IEC 11179 in a database or related environment.

Information technology — Concepts and usage of metadata —

Part 21:
11179-3, -31, -32 Data model in SQL

1 Scope

This document provides a possible instantiation of the registry metamodel specified in ISO/IEC 11179-3, ISO/IEC 11179-31, ISO/IEC 11179-32 using the SQL database language as specified in ISO/IEC 9075-2.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

4 The relationship among ISO/IEC 11179-3, ISO/IEC 11179-31 and ISO/IEC 11179-32

4.1 Overview

The relationships among these parts are mutually dependent, and they collectively form a complete metadata registry system. ISO/IEC 11179-3 provides infrastructure support for the entire registry. ISO/IEC 11179-31 and ISO/IEC 11179-32 extend the functionality of the metadata registry by registering data specifications and concept systems, further enriching the metadata information stored in the registry. These parts work together to enable the metadata registry to effectively support data management and the discovery, understanding, and utilization of data assets.

4.2 ISO/IEC 11179-3 — Metamodel for registry common facilities

ISO/IEC 11179-3 specifies the core metamodel for a Metadata Registry (MDR), but it does so in a generic way, so that the resulting registry could be used to register anything, not just metadata.

The registry metamodel is not a complete description of all the metadata an organization might wish to record. Therefore, the model is designed to be extended if required. However, extensions are, by their nature, not part of ISO/IEC 11179-3.

Some extensions to the ISO/IEC 11179-3 metamodel are specified in other parts of ISO/IEC 11179, such as:

— ISO/IEC 11179-31: Metamodel for data specification registration;

— ISO/IEC 11179-32: Metamodel for concept system registration;

— ISO/IEC 11179-33: Metamodel for data set registration;

— ISO/IEC 11179-34: Metamodel for computable object registration;

— ISO/IEC 11179-35: Metamodel for model registration.

4.3 ISO/IEC 11179-30 — Basic attributes of metadata

ISO/IEC 11179-30[8] describes basic attributes for data elements and related metadata for use in contexts where a full metadata registry is not appropriate, such as standards documents. It is limited to a set of basic attributes for: data elements, data element concepts, value domains, conceptual domains and other related classes.

4.4 ISO/IEC 11179-31 — Metamodel for data specification registration

ISO/IEC 11179-31 provides a specification for an extension to a Metadata Registry (MDR), as specified in ISO/IEC 11179-3, in which metadata that describe data elements and associated concepts, such as data element concepts, conceptual domains and value domains can be registered.

The specification in ISO/IEC 11179-31, together with the relevant clauses of the specification in ISO/IEC 11179-3, provides the ability to record metadata about:

— data elements, units of measure and derivation rules;

— data element concepts and associated object classes and properties;

— conceptual domains and value meanings;

— value domains, value domain subsets and permissible values.

ISO/IEC 11179-31 is applicable to the formulation of data representations, concepts, meanings and relationships to be shared among people and machines, independent of the organization that produces the data. It is not applicable to the physical representation of data as bits and bytes at the machine level.

4.5 ISO/IEC 11179-32 — Metamodel for concept system registration

ISO/IEC 11179-32 provides a specification for an extension to a Metadata Registry (MDR), as specified in ISO/IEC 11179-3, in which metadata that describe concept systems can be registered.

The specification in ISO/IEC 11179-32, together with the relevant clauses of the specification in ISO/IEC 11179-3, provides the ability to record the following metadata:

— concept systems and associated concepts;

— relations among concepts in a concept system;

— assertions about concepts in a concept system.

The metamodel in ISO/IEC 11179-32 is intended to support the full description of a concept system, including ontologies.

5 Overview of the relationship between UML Class Diagrams and SQL

The Unified Modeling Language (UML) provides a family of graphical notations that can be used in the analysis and design of software systems. The UML is under the control of the Object Management Group (OMG) and, as such, it is (a) a relatively 'open' standard, and (b) firmly rooted in the object-oriented paradigm for software engineering. The UML is now at Version 2 and is the subject of two international standards: ISO/IEC 19505-1 and ISO/IEC 19505-2.

Within the UML, the Class Diagram notation is used to represent information (and, hence, data) requirements for a particular 'universe of discourse', a business area or the scope of a proposed information system.

A UML Object is often defined as a:

construct within a system for which a set of attributes and operations can be specified.

Whilst this is a reasonable definition within the context of object-oriented system development, a more appropriate definition of an Object for the purposes of this document is a:

representation of something of interest within the universe of discourse about which information needs to be recorded.

An Object Class in both contexts can then defined as a:

definition of a set of Objects that share the same attributes, associations, and operations.

NOTE Although UML Class diagrams allow the specification of operations on the classes, the ISO/IEC 11179 series data models do not specify operations on their classes, so this document does not specify any operations either.

The Database Language SQL is a, largely, declarative language used to manage structured data held in a database under the control of a Relational Database Management System (RDBMS). As such it was originally based on Edgar F. Codd's relational model of data published in 1970,[9] but its scope has grown over the years. SQL is the subject of the ISO/IEC 9075 series. Most commercial SQL products, however, deviate from the standards to some extent, some more than others.

6 Generating the SQL for the metamodel

6.1 Overview

The UML (and the Class Diagrams, in particular) and the SQL database language exist in two separate programming paradigms and there is, therefore, no direct translation from one (the UML) to the other (SQL). There are, however, approaches that can be taken to achieve a translation. This document uses one of those approaches to generate a set of SQL statements to instantiate the metadata registry metamodel, where the SQL statements enable easy reference back to the original UML Class Diagram and text of the metamodel. This is achieved by using names for the SQL objects that reflect the names of the UML artefacts and, also, by embedding comments referencing the metamodel within the SQL statements.

6.2 General principles for the translation of a UML Class diagram into SQL statements

It is good practice to distinguish between SQL keywords and the names given to the SQL objects. One convention is to use UPPER CASE for the keywords and lower_case (using snake case) for the object names.

Each UML class is represented by an SQL table. To make correlation to the model easier, the name of a table that represents a class is the same as that of the class.

Each composite datatype is also represented by an SQL table. To make correlation to the model easier, the name of the table is the same as that of the datatype.

Each single-valued attribute of a class or a composite datatype is represented by a column in the appropriate table. The name of this column is the same as that of the attribute in the class or datatype. The datatype of an attribute column is intuitively selected to be similar to that of the datatype of the attribute.

If the datatype of an attribute of a class is another class or a composite datatype, the column that represents that attribute is additionally declared as a foreign key column referencing the relevant table that represents the class or composite datatype.In this case, the column name is suffixed with "_id" to reflect the fact that it contains a reference to the data, not the data itself.

Where an attribute is multivalued (that is, it has a multiplicity of [0..*] or [1..*] in the UML diagram) there are two possible instantiations available. These are the following.

a) Use one of the collection types, MULTISET or ARRAY, available in SQL.

b) Create a new table, a characteristic table, to hold the multiple values, with each row in the table having a foreign key referencing the kernel (prime) table and one of the values.

In object-orientation, and, hence, UML, there is no equivalent of the SQL primary key, so each table that represents a class or a composite datatype has an additional column that is used as a surrogate identifier. This column then becomes the primary key for the table.

UNIQUE or CHECK constraints may be added to a table where required. The latter are used, for example, to control the valid values for a column or to control which columns should, or should not, take values in different circumstances.

Specialization hierarchies (superclasses and their subclasses) can be instantiated in one of two ways using SQL structures.

1) It is possible to integrate all the classes (the superclass and its subclasses) into one table, named after the superclass, which includes a column for each attribute of the superclass, with those columns representing mandatory attributes being declared NOT NULL, and columns for each of the attributes of the subclasses. None of these subclass columns are declared as NOT NULL, irrespective of whether they are mandatory or optional within their subclass. An additional column, often called a discriminator column, with possible values representing each of the subclasses and a CHECK constraint are provided to manage which columns are populated for each subclass. Where a subclass is related to another class via a one-to-many association, such that there is a foreign key in the table representing the other class, a cross-table constraint is needed to ensure that a row in the other table will only exist if the value of the discriminator column represents the relevant subclass in the inheritance relationship.

2) Another possibility is the creation of a separate table for the superclass and the creation of additional tables for each of the subclasses. When the hierarchy is complete, disjoint and static (the normal situation in most models), a fully mandatory 'one-to-one' relationship is provided between the table representing the superclass and each table representing the subclasses. To achieve this, a subclass type column is added, which has a fixed value in each subclass type, but the value of which in the superclass is set to match the corresponding subclass. Although this column does not form part of the primary key of either the superclass or subclass tables, it is appended to the primary key in the referential constraints that enforce the subclassing. This ensures that for each row in the superclass table there is a corresponding row in one of the subclass tables, and that there is no duplication of primary key values across the tables representing the subclasses.

There are a number of different approaches that can be used when translating UML Class Diagram associations into SQL. Since each association in the metadata registry metamodel is named, the approach used in this document is to create a table for each association, with the table named with the name of the association.

Some many-to-many associations are annotated with association classes. These association classes also become tables.

Since the ISO/IEC 11179 metamodel does not specify operations on the UML classes, this document does not specify how operations can be specified in SQL.

6.3 Specific approaches taken for the translation of the metadata registry metamodel

6.3.1 Overview

The following subclauses provide specific detail about the translation of the metamodel artefacts, where the information in 6.2 is either not applicable or insufficient.

6.3.2 Obligations

In the metamodel the obligations applicable to each attribute or association are described as one of "Mandatory", "Conditional" or "Optional", with these obligations being enforced if, and only if, the Registration Status of the associated metadata item is Recorded or higher, that is, if the Registration Status of the associated item is one of "Recorded", "Qualified", "Standard" or "Preferred Standard". The obligations are not enforced if the Registration Status of the associated item is one of "Candidate", "Incomplete", "Retired" and "Superseded".

Any registry instantiation has to be able to register items with a lower Registration Status than “Recorded”, and the obligations cannot, therefore, be simply enforced.

The example SQL instantiation allows the attributes and associations to be optional so that items with a Registration Status lower than “Recorded” can be accommodated. The obligations applicable to items with a Registration Status of “Recorded” or higher will, therefore, need to be enforced in the registry application as opposed to the register (the database) itself.

6.3.3 Translation of datatypes

The datatypes used in the metamodel can be considered to be ‘primitive’ or more complex.

The primitive datatypes are translated as described in Table 1.

Table 1 — Translation of the primitive datatypes

Metadata registry metamodel datatype	Examples or Comment	SQL Datatype
Boolean		This simply translates as a column of type BOOLEAN
Date		This simply translates as a column of type DATE
Datetime		This simply translates as a column of type TIMESTAMP
Integer		This simply translates as a column of type INTEGER
Notation	XCL Common Logic, OWL-DL XML	This translates as a column of type CHARACTER VARYING (2500)
Sign	This could be a bit string, but, at a minimum, String must be supported.	This translates as a column of type CHARACTER VARYING (2500). If the SQL implementation supports the BINARY LARGE OBJECT type, this could be used instead.
String		This translates as a column of type CHARACTER VARYING (255)
Text		This translates as a column of type CHARACTER VARYING (2500). If the SQL implementation supports the CHARACTER LARGE OBJECT type, this could be used instead.

The more complex datatypes are translated as described in Table 2.

Table 2 — Translation of the more complex datatypes

Metadata registry metamodel datatype	Examples or Comment	SQL Datatype
Natural_Range	0, 1, 2, 1..2, 2..8, 0..*, 3..*	This is instantiated with three columns, one INTEGER column for the lower bound, one INTEGER column for the fixed upper bound, and a CHARACTER column, defaulting to ‘many’, for the many upper bound. The columns are then managed with a CHECK constraint.
Value	This represents a value of any of the types listed above	This is instantiated using many different columns, one, or more, for each of the datatypes listed and then a CHECK constraint implemented to ensure that only one datatype is represented.
Phone_Number		A table is created (named cdt_phone_number) with columns as specified in ISO/IEC 19773[10]; the table has a surrogate primary key of datatype INTEGER. Tables representing classes that have attributes specified with this datatype have foreign keys that reference cdt_phone_number.
Postal_Address		A table is created (named cdt_postal_address) with columns as specified in ISO/IEC 19773[10]; the table has a surrogate primary key of datatype INTEGER. Tables representing classes that have attributes specified with this datatype have foreign keys that reference cdt_postal_address.

6.3.4 Translation of the basic classes

Each basic class in the metamodel is represented by a table, with the name prefixed by cls_. Each table representing a basic class has a surrogate primary key of type INTEGER.

6.3.5 Translation of the remaining classes

Each of the remaining classes in the metamodel is represented by a table, with the name prefixed by cls_. Each table representing a class has a surrogate primary key of type INTEGER.

6.3.6 Translation of specialization hierarchies

Each superclass and its subclasses are represented by separate tables. Those tables that represent subclasses inherit the surrogate primary key of the superclass, either as the primary key or as an additional column, with that inherited primary key also being declared as a foreign key. Keys and constraints are normally included when the tables are created.

6.3.7 Translation of the association classes

Each association class in the metamodel is represented by a table, with the name prefixed by asscls_. Each table representing an association class has a surrogate primary key of type INTEGER, a column, or columns representing the attributes of the association class, and two additional columns: one each for the roles of the association class, both of which have datatype INTEGER and are specified as foreign keys referencing the tables representing the relevant classes.

6.3.8 Translation of the attributes of the classes

Where the metamodel datatype of the attribute is listed in the first column of Table 1, the attribute is translated as a single column of the relevant table representing the parent class, with a datatype as specified in the third column of Table 1.

Where the metamodel datatype of the attribute is listed in the first column of Table 2, the attribute is instantiated as specified in the third column of Table 2.

Where the attribute is a multivalued attribute, there are three options available.

— Where the metamodel datatype is specified in Table 1 and the values are unordered, a separate characteristic table, prefixed mva_, is created with a column for the multivalued attribute specified with the SQL datatype identified in Table 1. The second column in Table 1 is a foreign key column referencing the table representing the class that specifies the attribute. Both columns are declared as the primary key. If the SQL implementation supports the MULTISET collection type, this can be used instead of creating the characteristic table.

— Where the datatype is specified in Table 1 and the values are specified as being ordered, or it makes sense to order them, a separate characteristic table, prefixed mva_, is created with a column for the multivalued attribute specified with the SQL datatype identified in Table 1. The second column in Table 1 is a column of datatype INTEGER named ‘priority’ to indicate the order of the value. The third column in this table is a foreign key column referencing the table representing the class that specifies the attribute. All three columns are declared as the primary key. If the SQL implementation supports the ARRAY collection type, this could be used instead of creating the characteristic table.

— Where the datatype is specified in Table 2, a separate characteristic table, prefixed mva_, is created with a column for the multivalued attribute specified with the datatype INTEGER and as a foreign key referencing the table representing the complex datatype. The second column in this table is a foreign key column referencing the table representing the class that specifies the attribute. Both columns are declared as the primary key.

Where the datatype of the attribute is specified as another class specified in the metamodel, the attribute is instantiated as a column specified with the datatype INTEGER and as a foreign key referencing the table representing the class that is specified as the datatype in the metamodel. Where feasible, this column is

included in the CREATE TABLE statement. If this is not feasible, later ALTER TABLE statements are used to add the column and the foreign key.

6.3.9 Translation of the associations

Each association is one of three basic types, as follows:

— A zero or one-to-many association: In this case, a column is included in the table representing the class at the 'many' end of the association, with a referential constraint to the table representing the class at the other end of the association. The referential constraint is named with the name of the association prefixed by ass_.

— A many-to-many association: In this case, a separate association table is created, named with the name of the association prefixed by ass_, with two columns: one each for the roles of the association, both of which have datatype INTEGER and are specified as foreign keys referencing the tables representing the relevant classes. Both columns are declared as the primary key of the association table.

— A one-to-one association: In this case, a separate association table is created as for the many-to-many association, but, in addition, each column is declared with a UNIQUE constraint to enforce the one-to-oneness.

6.3.10 Cross-table constraints

Because obligations specified in the metamodel are only applicable if and only if, the Registration Status of the associated metadata item is Recorded or higher, very few of the constraints needed to enforce the obligations are included in the example SQL.

No cross-table constraints are specified in this document.

7 Example SQL for instantiation of the metamodel

The complete set of SQL statements needed to provide this example SQL instantiation is contained in Annexes A, B and C.

The statements are provided in the following order.

a) CREATE TABLE statements to create the tables to represent the two complex datatypes, the Phone_Number class and the Postal_Address class.

b) CREATE TABLE statements to create the tables to represent the basic classes, with additional characteristic tables to represent multi-valued attributes where needed.

c) CREATE TABLE statements to create the tables to represent the remaining classes. These tables are created grouped by the metamodel regions of the ISO/IEC 11179 part from which they are sourced. Additional characteristic tables are created to represent multi-valued attributes where needed.

d) CREATE TABLE statements to create the tables to represent the association classes. These tables are specified following the class tables within each metamodel region grouping, and are specified in alphabetical order within each group. Additional characteristic tables are created to represent multi-valued attributes where needed.

e) CREATE TABLE statements to create the tables to represent the metamodel associations.

This set of SQL statements does not provide the most optimal instantiation of a database for a metadata registry, but it does provide an instantiation that can easily be traced back to the metamodel.

Annex A
(informative)

Example SQL to instantiate the ISO/IEC 11179-3 metamodel

A.1 Predefined types metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for predefined types                                       */
/* ------------------------------------------------------------------------ */
CREATE TABLE cdt_phone_number
  (
    /* surrogate primary key */
    phone_number_id                     INTEGER PRIMARY KEY                ,
    /* columns to represent attributes */
    international_numbering_plan_prefix CHARACTER VARYING(255)             ,
    country_code                        CHARACTER VARYING(255)             ,
    city_code                           CHARACTER VARYING(255)             ,
    local_number                        CHARACTER VARYING(255) NOT NULL    ,
    extension                           CHARACTER VARYING(255)
  );

CREATE TABLE cdt_postal_address
  (
    /* surrogate primary key */
    postal_address_id                   INTEGER PRIMARY KEY                ,
    /* columns to represent attributes */
    sub_building_name                   CHARACTER VARYING(255)             ,
    building_name                       CHARACTER VARYING(255)             ,
    throughfare                         CHARACTER VARYING(255)             ,
    dependent_locality                  CHARACTER VARYING(255)             ,
    post_town                           CHARACTER VARYING(255)             ,
    region                              CHARACTER VARYING(255)             ,
    postcode                            CHARACTER VARYING(255)             ,
    country                             CHARACTER VARYING(255)
  );
```

A.2 Basic classes metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for basic classes                                          */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_individual
  (
    /* surrogate primary key */
    individual_id                       INTEGER PRIMARY KEY                ,
    /* columns to represent attributes */
    name                                CHARACTER VARYING(255) NOT NULL    ,
    "title"                             CHARACTER VARYING(255)             ,
    mail_address_id                     INTEGER
      REFERENCES cdt_postal_address (postal_address_id)
    /* the multi-valued attributes email_address and phone_number */
    /* instantiated as the characteristic tables mva_individual_email_addresses */
    /* and mva_individual_phone_numbers */
  );

CREATE TABLE mva_individual_email_addresses
  (
    /* column to represent multi-valued attribute */
    email_address                       CHARACTER VARYING(255)             ,
    /* identification of the individual this email address belongs to */
    owning_individual_id                INTEGER
      REFERENCES cls_individual (individual_id)                            ,
```

```
    PRIMARY KEY (email_address, owning_individual_id)
  );

CREATE TABLE mva_individual_phone_numbers
  (
    /* column to represent multi-valued attribute */
    phone_number_id                 INTEGER
      REFERENCES cdt_phone_number (phone_number_id)                    ,
    /* identification of the individual this phone number belongs to */
    owning_individual_id            INTEGER
      REFERENCES cls_individual (individual_id)                        ,
    PRIMARY KEY (phone_number_id, owning_individual_id)
  );

CREATE TABLE cls_organization
  (
    /* surrogate primary key */
    organization_id                 INTEGER PRIMARY KEY              ,
    /* columns to represent attributes */
    mail_address_id                 INTEGER
      REFERENCES cdt_postal_address (postal_address_id)              ,
    web_address                     CHARACTER VARYING(255)           ,
    /* the multi-valued attributes name, email_address and phone_number are */
    /* instantiated as the characteristic tables mva_organization_names, */
    /* mva_organization_email_addresses and mva_organization_phone_numbers */
  );

CREATE TABLE mva_organization_names
  (
    /* column to represent multi-valued attribute */
    name                            CHARACTER VARYING(2500)          ,
    /* identification of the organization this name belongs to */
    owning_organization_id          INTEGER
      REFERENCES cls_organization (organization_id)                  ,
    PRIMARY KEY (name, owning_organization_id)
   );

CREATE TABLE mva_organization_email_addresses
  (
    /* column to represent multi-valued attribute */
    email_address                   CHARACTER VARYING(255)           ,
    /* identification of the organization this email address belongs to */
    owning_organization_id          INTEGER
      REFERENCES cls_organization (organization_id)                  ,
    PRIMARY KEY (email_address, owning_organization_id)
  );

CREATE TABLE mva_organization_phone_numbers
  (
    /* column to represent multi-valued attribute */
    phone_number_id                 INTEGER
      REFERENCES cdt_phone_number (phone_number_id)                  ,
    /* identification of the organization this phone number belongs to */
    owning_organization_id          INTEGER
      REFERENCES cls_organization (organization_id)                  ,
    PRIMARY KEY (phone_number_id, owning_organization_id)
  );

CREATE TABLE cls_role
  (
    /* surrogate primary key */
    role_id                         INTEGER PRIMARY KEY              ,
    /* columns to represent attributes */
    role_name                       CHARACTER VARYING(255)           ,
    mail_address_id                 INTEGER
      REFERENCES cdt_postal_address (postal_address_id)
    /* the multi-valued attributes email_address and phone_number are */
    /* instantiated as the characteristic tables mva_role_email_addresses */
    /* and mva_role_phone_numbers */
  );

CREATE TABLE mva_role_email_addresses
```

```
  (
    /* column to represent multi-valued attribute */
    email_address                       CHARACTER VARYING(255)            ,
    /* identification of the role this email address belongs to */
    owning_role_id                      INTEGER
      REFERENCES cls_role (role_id)                                       ,
    PRIMARY KEY (email_address, owning_role_id)
  );

CREATE TABLE mva_role_phone_numbers
  (
    /* column to represent multi-valued attribute */
    phone_number_id                     INTEGER
      REFERENCES cdt_phone_number (phone_number_id)                       ,
    /* identification of the role this phone number belongs to */
    owning_role_id                      INTEGER
      REFERENCES cls_role (role_id)                                       ,
    PRIMARY KEY (phone_number_id, owning_role_id)
  );

CREATE TABLE cls_contact
  (
    /* surrogate primary key */
    contact_id                          INTEGER PRIMARY KEY               ,
    /* columns to represent attributes */
    organization_id                     INTEGER NOT NULL
      REFERENCES cls_organization (organization_id)                       ,
    individual_id                       INTEGER
      REFERENCES cls_individual (individual_id)                           ,
    role_id                             INTEGER
     REFERENCES cls_role (role_id)                                        ,
    /* constraint to check that individual_id and role_id attributes */
    CONSTRAINT contact_constraint
      CHECK
        ( ( individual_id IS NOT NULL ) OR ( role_id IS NOT NULL ) )
  );

CREATE TABLE cls_document_type
  (
    /* surrogate primary key */
    document_type_id                    INTEGER PRIMARY KEY               ,
    /* columns to represent attributes */
    identifier                          CHARACTER VARYING(255)            ,
    description                         CHARACTER VARYING(2500)           ,
    scheme_reference                    CHARACTER VARYING(2500)           ,
    /* constraint to check identified and described attributes */
    CONSTRAINT document_type_constraint1
      CHECK
        ( ( identifier IS NOT NULL ) OR ( description IS NOT NULL ) )      ,
    /* constraint to check identifier and scheme_reference attributes */
    CONSTRAINT document_type_constraint2
      CHECK
        ( ( ( identifier IS NOT NULL ) AND ( scheme_reference IS NOT NULL ) )
         OR ( ( identifier IS NULL ) AND ( scheme_reference IS NULL ) ) )
  );

CREATE TABLE cls_language_identification
  (
    /* surrogate primary key */
    language_identification_id          INTEGER PRIMARY KEY               ,
    /* columns to represent attributes */
    language_identifier                 CHARACTER VARYING(255)  NOT NULL  ,
    script_identifier                   CHARACTER VARYING(255)            ,
    geopolitical_territory_identifier   CHARACTER VARYING(255)            ,
    private_use_qualifier               CHARACTER VARYING(255)
    /* the multi-valued attributes variant_identifiers and extension_identifiers */
    /* are instantiated as the characteristic tables */
    /* mva_language_identification_variant_identifiers and */
    /* mva_language_identification_extension_identifiers */
  );

CREATE TABLE mva_language_identification_variant_identifiers
```

```
  (
    /* column to represent multi-valued attribute */
    variant_identifier                  CHARACTER VARYING(255)            ,
    /* column to represent the priority of this multi-valued attribute */
    priority                            INTEGER                           ,
    /* identification of the language identification this */
    /* variant identifier belongs to */
    owning_language_identification_id   INTEGER
      REFERENCES cls_language_identification (language_identification_id) ,
    PRIMARY KEY (variant_identifier, priority, owning_language_identification_id)
  );

CREATE TABLE mva_language_identification_extension_identifiers
  (
    /* column to represent multi-valued attribute */
    extension_identifier                CHARACTER VARYING(255)            ,
    /* column to represent the priority of this multi-valued attribute */
    priority                            INTEGER                           ,
    /* identification of the language identification this */
    /* extension identifier belongs to */
    owning_language_identification_id   INTEGER
      REFERENCES cls_language_identification (language_identification_id) ,
    PRIMARY KEY (extension_identifier, priority, owning_language_identification_id)
  );

CREATE TABLE cls_reference_document
  (
    /* surrogate primary key */
    reference_document_id               INTEGER PRIMARY KEY               ,
    /* columns to represent attributes */
    identifier                          CHARACTER VARYING(255) NOT NULL   ,
    type_description_id                 INTEGER
      REFERENCES cls_document_type (document_type_id)                     ,
    notation                            CHARACTER VARYING(2500)           ,
    "title"                             CHARACTER VARYING(2500)           ,
    uri                                 CHARACTER VARYING(255)
    /* the multi-valued attributes language and provider are instantiated as the
characteristic tables */
    /* mva_reference_document_language and mva_reference_document_organization */
  );

CREATE TABLE mva_reference_document_language
  (
    /* column to represent multi-valued attribute */
    language_id                         INTEGER
      REFERENCES cls_language_identification (language_identification_id) ,
    /* identification of the cls_reference_document this language belongs to */
    owning_reference_document_id        INTEGER
      REFERENCES cls_reference_document (reference_document_id)           ,
    PRIMARY KEY (language_id, owning_reference_document_id)
  );

CREATE TABLE mva_reference_document_organization
  (
    /* column to represent multi-valued attribute */
    provider_id                         INTEGER
      REFERENCES cls_organization(organization_id)                        ,
    /* identification of the cls_reference_document this provider belongs to */
    owning_reference_document_id       INTEGER
      REFERENCES cls_reference_document (reference_document_id)           ,
    PRIMARY KEY (provider_id, owning_reference_document_id)
  );

CREATE TABLE cls_registration_authority_identifier
  (
    /* surrogate primary key */
    registration_authority_identifier_id
                                        INTEGER PRIMARY KEY               ,
    /* columns to represent attributes */
    international_code_designator       CHARACTER VARYING(255) NOT NULL   ,
    organization_identifier             CHARACTER VARYING(255) NOT NULL   ,
    organization_part_identifier        CHARACTER VARYING(255)            ,
```

```
    OPI_source                          CHARACTER VARYING(255)            ,
    /* constraint to check that IF the organization_part_identifier */
    /* is present, then the organization_part_identifier_source     */
    /* shall be present                                             */
    CONSTRAINT organization_part_identifier_source_presence
      CHECK
        ( ( ( organization_part_identifier IS NOT NULL )
         AND ( organization_part_identifier_source IS NOT NULL ) ) OR
         ( ( organization_part_identifier IS NULL ) ) )
  );

CREATE TABLE cls_datetime_period
  (
    /* surrogate primary key */
    datetime_period_id                  INTEGER PRIMARY KEY               ,
    /* columns to represent attributes */
    start_datetime                      TIMESTAMP                         ,
    end_datetime                        TIMESTAMP                         ,
    CONSTRAINT datetime_period_constraint
      CHECK ( ( start_datetime IS NOT NULL OR end_datetime IS NOT NULL) )
  );
```

A.3 Core metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for core                                                   */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_item
  (
    /* surrogate primary key */
    item_id                             INTEGER PRIMARY KEY               ,
    /* add column used to enforce disjointness of item subtypes */
    item_type_id                        INTEGER NOT NULL                  ,
    /* The following UNIQUE constraint may or may not be needed. */
    /* Logically it is redundant because item_id is unique */
    /* as the primary key, but the DBMS may require it. */
    UNIQUE (item_id,item_type_id)                                         ,
    /* Check that item_type_id has a valid value. */
    /* The upper bound depends on the number of subclass types */
    /* defined and will need to be updated as new subclasses are added. */
    CHECK ( item_type_id > 0 AND item_type_id < 35 )
  );

CREATE TABLE cls_concept
  (
    /* cls_concept is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY               ,
    item_type_id                        INTEGER DEFAULT 1 NOT NULL /* 1 identifies subclass
cls_concept */ ,
    CONSTRAINT cls_concept_is_a_subclass_of_cls_item
      FOREIGN KEY ( item_id, item_type_id )
      REFERENCES cls_item (item_id, item_type_id)                         ,
    /* add column used to enforce disjointness of concept subtypes */
    concept_type_id                     INTEGER NOT NULL                  ,
    /* The following UNIQUE constraint may or may not be needed. */
    /* Logically it is redundant because item_id is unique */
    /* as the primary key, but the DBMS may require it. */
    UNIQUE (item_id, concept_type_id)                                     ,
    /* Check that concept_type_id has a valid value. */
    /* The upper bound depends on the number of subclass types */
    /* defined and will need to be updated as new subclasses are added. */
    CHECK ( concept_type_id > 0 AND concept_type_id < 9 )                 ,
    /* column to represent attribute */
    source_uri                          CHARACTER VARYING(255)
  );

CREATE TABLE cls_context
  (
    /* cls_context is a subclass of cls_item */
```

```
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY                ,
    item_type_id                     INTEGER DEFAULT 2 NOT NULL /* 2 identifies subclass
cls_context */ ,
    CONSTRAINT cls_context_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)
  );

CREATE TABLE cls_slot
  (
    /* surrogate primary key */
    slot_id                          INTEGER PRIMARY KEY                ,
    /* columns to represent attributes */
    name                             CHARACTER VARYING(255) NOT NULL    ,
    type                             CHARACTER VARYING(255)             ,
    /* the multi-valued attribute value is instantiated */
    /* as the characteristic table mva_slot_values */
    /* column to represent association */
    item_id                          INTEGER NOT NULL                   ,
    CONSTRAINT ass_item_slot
      FOREIGN KEY (item_id)
      REFERENCES cls_item (item_id)                                     ,
    CONSTRAINT slot_name_unique_within_item
      UNIQUE (item_id, name)
  );

CREATE TABLE mva_slot_values
  (
    /* column to represent multi-valued attribute */
    "value"                          CHARACTER VARYING(255)             ,
    /* column to represent the priority of this multi-valued attribute */
    priority                         INTEGER                            ,
    /* identification of the slot this value belongs to */
    owning_slot_id                   INTEGER
      REFERENCES cls_slot (slot_id)                                     ,
    PRIMARY KEY ("value", priority, owning_slot_id)
  );
```

A.4 Identification metamodel

```
/* ------------------------------------------------------------------- */
/* create tables for identification                                    */
/* ------------------------------------------------------------------- */
CREATE TABLE cls_identified_item
  (
    /* surrogate primary key */
    identified_item_id               INTEGER PRIMARY KEY
  );

CREATE TABLE cls_namespace
  (
    /* cls_namespace is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY                ,
    item_type_id                     INTEGER DEFAULT 3 NOT NULL /* 3 identifies subclass
cls_namespace */ ,
    CONSTRAINT namespace_is_a_subclass_of_item
      FOREIGN KEY (item_id,item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                       ,
    /* columns to represent native attributes */
    naming_authority_id              INTEGER
      REFERENCES cls_organization(organization_id)                      ,
    one_name_per_item_indicator      BOOLEAN NOT NULL                   ,
    one_item_per_name_indicator      BOOLEAN NOT NULL                   ,
    mandatory_naming_convention_indicator
                                     BOOLEAN NOT NULL                   ,
    shorthand_prefix                 CHARACTER VARYING(255)             ,
    scheme_reference                 CHARACTER VARYING(2500)
```

```
  );

CREATE TABLE cls_scoped_identifier
  (
    /* surrogate primary key */
    scoped_identifier_id                INTEGER PRIMARY KEY               ,
    /* columns to represent attributes */
    identifier                          CHARACTER VARYING(255) NOT NULL   ,
    version                             CHARACTER VARYING(255) NOT NULL   ,
    short_expansion                     CHARACTER VARYING(255)            ,
    /* association with identified_item */
    identified_item_id                  INTEGER                           ,
    CONSTRAINT ass_identification
      FOREIGN KEY (identified_item_id)
      REFERENCES cls_identified_item (identified_item_id)                 ,
    /* association with identifier_scope */
    namespace_item_id                   INTEGER NOT NULL                  ,
    CONSTRAINT ass_identifier_scope
      FOREIGN KEY (namespace_item_id)
      REFERENCES cls_namespace (item_id)
  );
```

A.5 Designation and Definition metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for designation and definition                             */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_designation
  (
    /* surrogate primary key */
    designation_id                      INTEGER PRIMARY KEY               ,
    /* columns to represent attributes */
    sign                                CHARACTER VARYING(2500) NOT NULL  ,
    language_id                         INTEGER
      REFERENCES cls_language_identification (language_identification_id) ,
    source_id                           INTEGER
      REFERENCES cls_reference_document (reference_document_id)           ,
    /* reference to item being designated */
    designated_item_id INTEGER NOT NULL                                   ,
    CONSTRAINT ass_item_designation
      FOREIGN KEY (designated_item_id)
      REFERENCES cls_item (item_id)
  );

CREATE TABLE cls_definition
  (
    /* surrogate primary key */
    definition_id                        INTEGER PRIMARY KEY              ,
    /* column to represent attribute */
    text                                 CHARACTER VARYING(2500) NOT NULL ,
    language_id                          INTEGER
      REFERENCES cls_language_identification(language_identification_id)  ,
    source_id                            INTEGER
      REFERENCES cls_reference_document(reference_document_id)            ,
    /* reference to item being defined */
    defined_item_id                      INTEGER NOT NULL                 ,
    CONSTRAINT ass_item_definition
      FOREIGN KEY (defined_item_id)
      REFERENCES cls_item(item_id)
  );

CREATE TABLE cls_designation_definition_pairing
  (
    /* columns to represent associations */
    definition_heading_id               INTEGER NOT NULL                  ,
    CONSTRAINT ass_paired_designation
      FOREIGN KEY (definition_heading_id)
      REFERENCES cls_designation (designation_id)                         ,
    specific_definition_id              INTEGER NOT NULL                  ,
    CONSTRAINT ass_paired_definition
```

```
      FOREIGN KEY (specific_definition_id)
      REFERENCES cls_definition (definition_id)                              ,
    context_id                          INTEGER NOT NULL                     ,
    CONSTRAINT ass_context_for_pairing
      FOREIGN KEY (context_id)
      REFERENCES cls_context (item_id)                                       ,
    PRIMARY KEY (definition_heading_id, specific_definition_id, context_id),
    /* column to represent attribute */
    acceptability                       CHARACTER VARYING(255)               ,
    CONSTRAINT designation_definition_pairing_acceptability_enumeration
      CHECK ( acceptability IN
            ( 'preferred', 'admitted', 'deprecated',
              'superseded','obsolete' ) )
  );

CREATE TABLE cls_naming_convention
  (
    /* cls_naming_convention is a subclass of cls_item */
    /* Inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY                  ,
    item_type_id                        INTEGER DEFAULT 4 NOT NULL /* 4 identifies cls_maming_
convention */ ,
    CONSTRAINT cls_naming_convention_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                            ,
    /* columns to represent attributes */
    scope_rule                          CHARACTER VARYING(2500) NOT NULL     ,
    authority_rule                      CHARACTER VARYING(2500)              ,
    semantic_rule                       CHARACTER VARYING(2500)              ,
    syntactic_rule                      CHARACTER VARYING(2500)              ,
    lexical_rule                        CHARACTER VARYING(2500)
  );

CREATE TABLE asscls_definition_context
  (
    /* surrogate primary key */
    definition_context_id               INTEGER PRIMARY KEY                  ,
    /* column to represent attribute */
    acceptability                       CHARACTER VARYING(255)               ,
    /* columns and foreign keys of the association class */
    definition_id                       INTEGER NOT NULL
      REFERENCES cls_definition(definition_id)                               ,
    context_id                          INTEGER NOT NULL
      REFERENCES cls_context(item_id)                                        ,
    CONSTRAINT definition_context_acceptability_enumeration
      CHECK ( acceptability IN
        ( 'preferred', 'admitted', 'deprecated', 'superseded', 'obsolete' ) )
  );

CREATE TABLE asscls_designation_context
  (
    /* surrogate primary key */
    designation_context_id              INTEGER PRIMARY KEY                  ,
    /* column to represent attribute */
    acceptability                       CHARACTER VARYING(255)               ,
    /* columns and foreign keys of the association class */
    designation_id                       INTEGER NOT NULL
      REFERENCES cls_designation(designation_id)                             ,
    context_id                          INTEGER NOT NULL
      REFERENCES cls_context(item_id)                                        ,
    CONSTRAINT designation_context_acceptability_enumeration
      CHECK ( acceptability IN
       ( 'preferred', 'admitted', 'deprecated', 'superseded', 'obsolete' ) )
  );
```

A.6 Registration metamodel

```
/* ------------------------------------------------------------------------  */
/* create tables for registration                                            */
/* ------------------------------------------------------------------------  */
```

```
CREATE TABLE cls_constraint_set
  (
    /* cls_constraint_set is a subclass of cls_item */
    /* Inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY                ,
    item_type_id                     INTEGER DEFAULT 5 NOT NULL /* 5 identifies cls_
constraint_set */ ,
    CONSTRAINT cls_constraint_set_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                        ,
constraint_set_type_id              INTEGER                          ,
    /* The following UNIQUE constraint may or may not be needed. */
    /* Logically it is redundant because item_id is unique */
    /* as the primary key, but the DBMS may require it. */
    UNIQUE (item_id, constraint_set_type_id)                           ,
    /* Check that constraint_set_type_id has a valid value. */
    /* The upper bound depends on the number of subclass types */
/* defined and will need to be updated as new subclasses are added. */
    CHECK (constraint_set_type_id > 0 AND constraint_set_type_id <14)  ,
    /* columns to represent attributes */
    item_shall_be_designated_indicator  BOOLEAN NOT NULL               ,
    item_shall_be_defined_indicator     BOOLEAN NOT NULL               ,
    item_shall_be_classified_indicator  BOOLEAN NOT NULL
);

CREATE TABLE cls_registered_item
  (
    /* cls_registered_item is a subclass of cls_identified_item */
    /* inherited surrogate primary key */
    identified_item_id               INTEGER PRIMARY KEY                ,
    registered_item_type_id          INTEGER NOT NULL                   ,
    UNIQUE (identified_item_id, registered_item_type_id)                ,
    CHECK (registered_item_type_id IN (1,2))                           ,
    CONSTRAINT cls_registered_item_is_a_subclass_of_cls_identified_item
      FOREIGN KEY(identified_item_id)
      REFERENCES cls_identified_item(identified_item_id)               ,
    /* columns to represent associations */
    constraint_set_id                INTEGER                            ,
    CONSTRAINT ass_registered_item_constraint_set
      FOREIGN KEY (constraint_set_id)
      REFERENCES cls_constraint_set(item_id)
  );

CREATE TABLE cls_registration_state
  (
    /* surrogate primary key */
    registration_state_id            INTEGER PRIMARY KEY                ,
    /* columns to represent attributes */
    registration_status              CHARACTER VARYING(255) NOT NULL    ,
    effective_date                   TIMESTAMP NOT NULL                 ,
    until_date                       TIMESTAMP                          ,
    administrative_note              CHARACTER VARYING(2500)            ,
    unresolved_issue                 CHARACTER VARYING(2500)            ,
    administrative_status            CHARACTER VARYING(255)             ,
    /* this reflexive association to registration is one-to-one */
    previous_state_id                INTEGER UNIQUE
      REFERENCES cls_registration_state (registration_state_id)        ,
    CONSTRAINT registration_status_enumeration
      CHECK ( registration_state IN
        ( 'Incomplete', 'Candidate', 'Recorded', 'Qualified',
          'Standard', 'Preferred Standard', 'Superseded',
          'Retired', 'Historical', 'Application' ) )
  );

CREATE TABLE cls_stewardship_record
  (
    /* surrogate primary key */
    stewardship_record_id            INTEGER PRIMARY KEY                ,
    /* columns to represent attributes */
    organization_id                  INTEGER NOT NULL
      REFERENCES cls_organization (organization_id)                    ,
    contact_id                       INTEGER NOT NULL
```

```
      REFERENCES cls_contact (contact_id)
  );

CREATE TABLE cls_registration_authority
  (
    /* cls_registration_authority is a subclass of cls_organization */
    /* inherited surrogate primary key */
    organization_id                     INTEGER PRIMARY KEY                ,
    CONSTRAINT cls_registration_authority_is_a_subclass_of_cls_organization
      FOREIGN KEY (organization_id)
      REFERENCES cls_organization (organization_id)                        ,
    /* columns to represent attributes */
    registration_authority_identifier_id
                                        INTEGER UNIQUE NOT NULL
      REFERENCES cls_registration_authority_identifier (registration_authority_identifier_id)
    /* the multi-valued attributes documentation_language_identifier */
    /* instantiated as the characteristic tables mva_registration_authority_documentation_
language_identifier */
  );

CREATE TABLE cls_administered_item
  (
    /* cls_administered_item is a subclass of cls_registered_item */
    /* inherited surrogate primary key */
    identified_item_id                  INTEGER PRIMARY KEY                ,
    registered_item_type_id             INTEGER DEFAULT 1 NOT NULL /* 1 identifies cls_
administered_item */ ,
    CONSTRAINT cls_administered_item_is_a_subclass_of_cls_registered_item
      FOREIGN KEY (identified_item_id, registered_item_type_id)
      REFERENCES cls_registered_item(identified_item_id, registered_item_type_id),
    /* columns to represent attributes */
    creation_datetime                   TIMESTAMP NOT NULL                 ,
    last_change_datetime                TIMESTAMP                          ,
    change_description                  CHARACTER VARYING(2500)            ,
    explanatory_comment                 CHARACTER VARYING(2500)            ,
    origin                              CHARACTER VARYING(2500)            ,
    registration_state_id               INTEGER                            ,
      REFERENCES cls_registration_state(registration_state_id)             ,
    /* columns to represent associations */
    registration_authority_id           INTEGER NOT NULL                   ,
    CONSTRAINT ass_registration
      FOREIGN KEY (registration_authority_id)
      REFERENCES cls_registration_authority (organization_id)              ,
    stewardship_record_id               INTEGER NOT NULL                   ,
    CONSTRAINT ass_stewardship
      FOREIGN KEY (stewardship_record_id)
      REFERENCES cls_stewardship_record (stewardship_record_id)
  );

CREATE TABLE cls_attached_item
  (
    /* cls_attached_item is a subclass of cls_registered_item */
    /* inherited surrogate primary key */
    identified_item_id                  INTEGER PRIMARY KEY                ,
    registered_item_type_id             INTEGER DEFAULT 2 NOT NULL /* 2 identifies cls_
attached_item */ ,
    /* foreign key to instantiate subclass */
    CONSTRAINT cls_attached_item_is_a_subclass_of_cls_registered_item
      FOREIGN KEY  (identified_item_id, registered_item_type_id)
      REFERENCES cls_registered_item (identified_item_id, registered_item_type_id),
    /* column to represent association */
    administered_item_id                INTEGER NOT NULL                   ,
    CONSTRAINT ass_attachment
      FOREIGN KEY (administered_item_id)
      REFERENCES cls_administered_item (identified_item_id)
  );

CREATE TABLE mva_registration_authority_documentation_language_identifier
  (
    /* column to represent multi-valued attribute */
    documentation_language_identifier_id   INTEGER NOT NULL
      REFERENCES cls_language_identification(language_identification_id)  ,
```

```
    /* identification of the cls_registration_authority this documentation_language_identifier
belongs to */
    owning_organization_id              INTEGER NOT NULL
      REFERENCES cls_registration_authority(organization_id)              ,
    PRIMARY KEY(documentation_language_identifier_id,owning_organization_id)
  );

CREATE TABLE cls_registrar
  (
    /* cls_registrar is a subclass of cls_contact */
    /* inherited surrogate primary key */
    contact_id                          INTEGER PRIMARY KEY                 ,
    CONSTRAINT cls_registrar_is_a_subclass_of_cls_contact
      FOREIGN KEY (contact_id)
      REFERENCES cls_contact (contact_id)                                   ,
    /* column to represent native attribute */
    identifier                          CHARACTER VARYING(255)              ,
    /* column to represent association */
    registration_authority_id           INTEGER NOT NULL                    ,
    CONSTRAINT ass_registration_authority_registrar
      FOREIGN KEY (registration_authority_id)
      REFERENCES cls_registration_authority (organization_id)
  );

CREATE TABLE cls_submission_record
  (
    /* surrogate primary key */
    submission_record_id                INTEGER PRIMARY KEY                 ,
    /* columns to represent attributes */
    organization_id                     INTEGER NOT NULL
      REFERENCES cls_organization (organization_id)                         ,
    contact_id                          INTEGER NOT NULL
      REFERENCES cls_contact (contact_id)
  ) ;

CREATE TABLE cls_registry_specification
  (
    /* surrogate primary key */
    registry_specification_id           INTEGER PRIMARY KEY                 ,
    /* columns to represent native attributes */
    name                                CHARACTER VARYING(2500) NOT NULL    ,
    "comment"                           CHARACTER VARYING(2500)             ,
    web_address                         CHARACTER VARYING(255)              ,
    default_language_id                 INTEGER
      REFERENCES cls_language_identification(language_identification_id)    ,
    character_repertoire                CHARACTER VARYING(255) NOT NULL     ,
    reference_document_identifier_kind  CHARACTER VARYING(255)              ,
    representation_class_scheme_id      CHARACTER VARYING(255)              ,
    context_id                          INTEGER
      REFERENCES cls_context(context_id)                                    ,
    "standard"                          CHARACTER VARYING(255)              ,
    degree_of_conformance               CHARACTER VARYING(255)              ,
    conformance_level                   CHARACTER VARYING(255)              ,
    conformance_label                   CHARACTER VARYING(255)              ,
    included_features                   CHARACTER VARYING(255)              ,
    excluded_features                   CHARACTER VARYING(255)              ,
    default_time_zone                   TIMESTAMP                           ,
    CHECK ( ( degree_of_conformance IS NULL )
      OR ( "standard" IS NOT NULL AND degree_of_conformance IN ( 'conforming', 'strictly
conforming' ) ) ) ,
    CHECK (conformance_level IS NULL OR "standard" IS NOT NULL)               ,
    CHECK (conformance_label IS NULL OR "standard" IS NOT NULL)               ,
    CHECK (included_features IS NULL OR "standard" IS NOT NULL)               ,
    CHECK (excluded_features IS NULL OR "standard" IS NOT NULL)
    /* the multi-valued attributes alternative_language and reference_document_identifier_kind
*/
    /* instantiated as the characteristic tables mva_registry_specification_alt ernative_
language */
    /* and mva_registry_specification_reference_document_identifier_kind */
  );

CREATE TABLE mva_registry_specification_alternative_language
```

```
  (
    /* column to represent multi-valued attribute */
    language_identification_id          INTEGER NOT NULL
      REFERENCES cls_language_identification(language_identification_id)  ,
    /* identification of the cls_registry_specification this alternative_language belongs to
*/
    owning_registry_specification_id    INTEGER NOT NULL
      REFERENCES cls_registry_specification(registry_specification_id)    ,
    PRIMARY KEY(language_identification_id,owning_registry_specification_id)
  );

CREATE TABLE mva_registry_specification_reference_document_identifier_kind
  (
    /* column to represent multi-valued attribute */
    reference_document_identifier_kind    CHARACTER VARYING(255) NOT NULL ,
    /* identification of the cls_registry_specification this reference_document_identifier_
kind belongs to */
    owning_registry_specification_id      INTEGER NOT NULL
      REFERENCES cls_registry_specification(registry_specification_id)    ,
    PRIMARY KEY(reference_document_identifier_kind,owning_registry_specification_id)
  );
```

A.7 Classification metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for classification                                         */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_classification_scheme
  (
    /* cls_classification_scheme is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY                 ,
    item_type_id                        INTEGER DEFAULT 6 NOT NULL /* 6 identifies cls_
classification_scheme */ ,
    CONSTRAINT cls_classification_scheme_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)
  );

CREATE TABLE cls_classification_scheme_item
  (
    /* cls_classification_scheme_item is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY                 ,
    item_type_id                        INTEGER DEFAULT 7 NOT NULL /* 7 identifies cls_
classification_scheme_item */ ,
    CONSTRAINT cls_classification_scheme_item_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                           ,
    /* columns to represent native attributes */
    classification_scheme_item_concept_id INTEGER
      REFERENCES cls_concept(item_id)                                       ,
    classification_scheme_item_sign     CHARACTER VARYING(2500)             ,
    hierarchy_level                     INTEGER                             ,
    CHECK (classification_scheme_item_concept IS NOT NULL OR classification_scheme_item_sign
IS NOT NULL )
  );

CREATE TABLE cls_classification_scheme_item_relationship_type
  (
    /* cls_classification_scheme_item_relationship_type is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY                 ,
    item_type_id                        INTEGER DEFAULT 8 NOT NULL /* 8 identifies cls_
classification_scheme_item_relationship_type */ ,
    CONSTRAINT cls_classification_scheme_item_relationship_type_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)
  );
```

```
CREATE TABLE cls_classification_scheme_item_relationship
  (
    /* cls_classification_scheme_item_relationship is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY ,
    item_type_id                         INTEGER DEFAULT 9 NOT NULL /* 9 identifies cls_
classification_scheme_item_relationship */ ,
    CONSTRAINT cls_classification_scheme_item_relationship_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                        ,
    /* column to represent association */
    classification_scheme_item_relationship_type_id INTEGER NOT NULL     ,
    CONSTRAINT ass_classification_scheme_item_relationship_categorization
      FOREIGN KEY (classification_scheme_item_relationship_type_id)
      REFERENCES cls_classification_scheme_item_relationship_type (item_id)
  );
```

A.8 Item Mapping metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for item  mapping                                          */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_item_mapping
  (
    /* cls_item_mapping is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY                ,
    item_type_id                         INTEGER DEFAULT 10 NOT NULL /* 10 identifies cls_item_
mapping */ ,
    CONSTRAINT cls_item_mapping_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                        ,
    /* columns to represent native attributes */
    degree                               CHARACTER VARYING(255) NOT NULL   ,
    description                          CHARACTER VARYING(2500)           ,
    transformation_rule                  CHARACTER VARYING(2500)           ,
    transformation_rule_notation         CHARACTER VARYING(2500)           ,
    CHECK ( degree IN
     ( 'related_to', 'same_as', 'semantically_equivalent', 'semantically_similar',
        'derived_from', 'has_derivation', 'broader_than', 'narrower_than',
        'child_of', 'parent_of', 'instance_of', 'has_instance', 'superset_of',
        'subset_of', 'supertype_of', 'subtype_of', 'contains', 'contained_in' ) ),
    /* column to represent association */
    item_mapping_id                      INTEGER                           ,
    CONSTRAINT ass_mapping_hierarchy
      FOREIGN KEY (item_mapping_id)
      REFERENCES cls_item_mapping (item_id)
  );

CREATE TABLE asscls_subject_mapping
  (
    /* surrogate primary key */
    subject_mapping_id                   INTEGER PRIMARY KEY               ,
    /* columns to represent native attributes */
    mapping_variable_name                CHARACTER VARYING(255)            ,
    /* columns and foreign keys of the association */
    item_mapping_id                      INTEGER
      REFERENCES cls_item_mapping (item_id)                              ,
    item_id                              INTEGER
      REFERENCES cls_item (item_id)
  );

CREATE TABLE asscls_object_mapping
  (
    /* surrogate primary key */
    object_mapping_id                    INTEGER PRIMARY KEY               ,
    /* columns to represent native attributes */
    mapping_variable_name                CHARACTER VARYING(255)            ,
    /* columns and foreign keys of the association */
    item_mapping_id                      INTEGER
```

```
      REFERENCES cls_item_mapping (item_id)                              ,
    item_id                             INTEGER
      REFERENCES cls_item (item_id)
  );
```

A.9 Associations in metamodels

```
/* ------------------------------------------------------------------------ */
/* create tables for the identification associations                        */
/* ------------------------------------------------------------------------ */
CREATE TABLE ass_item_identification
/* a one-to-one association */
  (
    /* columns and foreign keys of the association */
    identified_item_id                  INTEGER UNIQUE
      REFERENCES cls_identified_item (identified_item_id)               ,
    item_id                             INTEGER UNIQUE
      REFERENCES cls_item (item_id)                                     ,
    PRIMARY KEY (identified_item_id, item_id)
  );

/* ------------------------------------------------------------------------ */
/* create tables for the designation and definition associations            */
/* ------------------------------------------------------------------------ */
CREATE TABLE ass_designation_namespace
/* a many-to-many association */
  (
    /* columns and foreign keys of the association */
    designation_id                      INTEGER
      REFERENCES cls_designation (designation_id)                       ,
    namespace_id                        INTEGER
      REFERENCES cls_namespace (namespace_id)                           ,
    PRIMARY KEY (designation_id, namespace_id)
  );

CREATE TABLE ass_naming_convention_conformance
/* a many-to-many association */
  (
    /* columns and foreign keys of the association */
    naming_convention_id                INTEGER
      REFERENCES cls_naming_convention (item_id)                        ,
    designation_id                      INTEGER
      REFERENCES cls_designation (designation_id)                       ,
    PRIMARY KEY (naming_convention_id, designation_id)
  );

CREATE TABLE ass_naming_convention_utilization
/* a many-to-many association */
  (
    /* columns and foreign keys of the association */
    naming_convention_id                INTEGER
      REFERENCES cls_naming_convention (item_id)                        ,
    namespace_id                        INTEGER
      REFERENCES cls_namespace (item_id)                                ,
    PRIMARY KEY (naming_convention_id, namespace_id)
  );

/* ------------------------------------------------------------------------ */
/* create tables for the registration associations                          */
/* ------------------------------------------------------------------------ */
CREATE TABLE ass_reference
/* a many-to-many association */
  (
    /* columns and foreign keys of the association */
    reference_document_id               INTEGER
      REFERENCES cls_reference_document (reference_document_id)         ,
    registered_item_id                  INTEGER
      REFERENCES cls_registered_item (identified_item_id)               ,
    PRIMARY KEY (reference_document_id, registered_item_id)
  );
```

```
CREATE TABLE ass_registration_authority_namespace
/* a one-to-one association */
  (
    /* columns and foreign keys of the association */
    registration_namespace_id           INTEGER UNIQUE
      REFERENCES cls_namespace (item_id)                                ,
    registration_authority_id           INTEGER UNIQUE
      REFERENCES cls_registration_authority (registration_authority_id)   ,
    PRIMARY KEY (registration_namespace_id, registration_authority_id)
  );

CREATE TABLE ass_submission
/* a many-to-many association */
  (
    /* columns and foreign keys of the association */
    submission_record_id                INTEGER
      REFERENCES cls_submitted_record (submitted_record_id)             ,
    registered_item_id                  INTEGER
      REFERENCES cls_registered_item (identified_item_id)               ,
    PRIMARY KEY (submission_record_id, registered_item_id)
  );

/* ------------------------------------------------------------------- */
/* create tables for the classification associations                   */
/* ------------------------------------------------------------------- */
CREATE TABLE ass_classification_scheme_membership
/* a many-to-many association */
  (
    /* columns and foreign keys of the association */
    classification_scheme_item_id       INTEGER
      REFERENCES cls_classification_scheme_item (item_id)               ,
    classification_scheme_id            INTEGER
      REFERENCES cls_classification_scheme (item_id)                    ,
    PRIMARY KEY (classification_scheme_id, classification_scheme_item_id)
  );

CREATE TABLE ass_subject_classification_scheme_item
/* a many-to-many association */
  (
    /* columns and foreign keys of the association */
    classification_scheme_item_relationship_id
                                        INTEGER
      REFERENCES cls_classification_scheme_item_relationship(item_id)   ,
    classification_scheme_item_id       INTEGER
      REFERENCES cls_classification_scheme_item (item_id)               ,
    PRIMARY KEY (classification_scheme_item_id, classification_scheme_item_relationship_id)
  );

CREATE TABLE ass_object_classification_scheme_item
/* a many-to-many association */
  (
    /* columns and foreign keys of the association */
    classification_scheme_item_relationship_id
                                        INTEGER
      REFERENCES cls_classification_scheme_item_relationship (item_id)          ,
    classification_scheme_item_id       INTEGER
      REFERENCES cls_classification_scheme_item (item_id)               ,
    PRIMARY KEY (classification_scheme_item_id, classification_scheme_item_relationship_id)
  );
```

Annex B
(informative)

Example SQL to instantiate the ISO/IEC 11179-32 metamodel

B.1 Concept System metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for Concept System metamodel                               */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_concept_system
  (
    /* cls_concept_system is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY
    item_type_id                     INTEGER DEFAULT 11 NOT NULL /* 11 identifies cls_
concept_system */ ,
    CONSTRAINT cls_concept_system_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                    ,
/* columns to represent attributes */
    notation                          CHARACTER VARYING(2500)       ,
    source_uri                        CHARACTER VARYING(255)
  );

CREATE TABLE cls_relation
  (
    /* cls_relation is a subclass of cls_concept which in turn is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY            ,
    concept_type_id                  INTEGER DEFAULT 1 NOT NULL /* 1 identifies cls_
relation */ ,
    CONSTRAINT cls_relation_is_a_subclass_of_cls_concept
      FOREIGN KEY (item_id, concept_type_id)
      REFERENCES cls_concept (item_id, concept_type_id)              ,
    /* columns to represent attribute with natural range datatype*/
    arity_lower_bound_natural_range   INTEGER                        ,
    arity_fixed_upper_bound_natural_range
                                      INTEGER                        ,
    arity_many_upper_bound_natural_range
                                      CHARACTER(4)                   ,
    CONSTRAINT relation_arity_natural_range_check
      CHECK
       ( ( ( arity_lower_bound_natural_range IS NULL )
             AND ( arity_fixed_upper_bound_natural_range IS NULL )
             AND ( arity_many_upper_bound_natural_range IS NULL ) )
         OR
         ( ( arity_lower_bound_natural_range IS NOT NULL )
             AND ( arity_lower_bound_natural_range >= 0 )
             AND ( arity_fixed_upper_bound_natural_range IS NULL )
             AND ( arity_many_upper_bound_natural_range IS NULL ) )
         OR
         ( ( arity_lower_bound_natural_range IS NOT NULL )
             AND ( arity_lower_bound_natural_range >= 0 )
             AND ( arity_fixed_upper_bound_natural_range IS NOT NULL )
             AND ( arity_fixed_upper_bound_natural_range >= arity_lower_bound_natural_range )
             AND ( arity_many_upper_bound_natural_range IS NULL ) )
         OR
         ( ( arity_lower_bound_natural_range IS NOT NULL )
             AND ( arity_lower_bound_natural_range >= 0 )
             AND ( arity_fixed_upper_bound_natural_range IS NULL )
             AND ( arity_many_upper_bound_natural_range IS NOT NULL ) ) )
  );
```

```
CREATE TABLE cls_relation_role
  (
    /* cls_relation_role is a subclass of cls_concept which in turn is a subclass of cls_item
*/
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY                ,
    concept_type_id                      INTEGER DEFAULT 2 NOT NULL /* 2 identifies cls_
relation_role */ ,
    CONSTRAINT cls_relation_role_is_a_subclass_of_cls_concept
      FOREIGN KEY (item_id, concept_type_id)
      REFERENCES cls_concept (item_id, concept_type_id)                    ,
    /* columns to represent attributes */
    multiplicity_lower_bound_natural_range
                                         INTEGER                            ,
    multiplicity_fixed_upper_bound_natural_range
                                         INTEGER                            ,
    multiplicity_many_upper_bound_natural_range
                                         CHARACTER(4)                       ,
    ordinal                              INTEGER                            ,
    /* columns to represent associations */
    relation_id                          INTEGER NOT NULL                   ,
    CONSTRAINT ass_relation_role_set
      FOREIGN KEY (relation_id)
      REFERENCES cls_relation (item_id)                                     ,
    CONSTRAINT relation_role_multiplicity_natural_range_check
      CHECK
       ( ( ( multiplicity_lower_bound_natural_range IS NULL )
            AND ( multiplicity_fixed_upper_bound_natural_range IS NULL )
            AND ( multiplicity_many_upper_bound_natural_range IS NULL ) )
         OR
         ( ( multiplicity_lower_bound_natural_range IS NOT NULL )
            AND ( multiplicity_lower_bound_natural_range >= 0 )
            AND ( multiplicity_fixed_upper_bound_natural_range IS NULL )
            AND ( multiplicity_many_upper_bound_natural_range IS NULL ) )
         OR
         ( ( multiplicity_lower_bound_natural_range IS NOT NULL )
            AND ( multiplicity_lower_bound_natural_range >= 0 )
            AND ( multiplicity_fixed_upper_bound_natural_range IS NOT NULL )
            AND ( multiplicity_fixed_upper_bound_natural_range >= multiplicity_lower_bound_
natural_range )
            AND ( multiplicity_many_upper_bound_natural_range IS NULL ) )
         OR
         ( ( multiplicity_lower_bound_natural_range IS NOT NULL )
            AND ( multiplicity_lower_bound_natural_range >= 0 )
            AND ( multiplicity_fixed_upper_bound_natural_range IS NULL )
            AND ( multiplicity_many_upper_bound_natural_range IS NOT NULL ) ) )
  );

CREATE TABLE cls_assertion
  (
    /* cls_assertion is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY                ,
    item_type_id                         INTEGER DEFAULT 12 NOT NULL /* 12 identifies cls_
assertion */ ,
    CONSTRAINT cls_assertion_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                           ,
    /* column to represent attribute */
    formula                              CHARACTER VARYING(2500)
  );

CREATE TABLE cls_link
/* subclass of cls_assertion */
  (
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY                ,
    CONSTRAINT cls_link_is_a_subclass_of_cls_assertion
      FOREIGN KEY (item_id)
      REFERENCES cls_assertion (item_id)                                    ,
    /* column to represent association */
    relation_id INTEGER NOT NULL                                            ,
```

```
    CONSTRAINT ass_relation_link
      FOREIGN KEY (relation_id)
      REFERENCES cls_relation (item_id)
  );

CREATE TABLE cls_link_end
  (
    /* cls_link_end is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY              ,
    item_type_id                     INTEGER DEFAULT 13 NOT NULL /* 13 identifies cls_link_
end */ ,
    CONSTRAINT cls_link_end_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
       REFERENCES cls_item (item_id, item_type_id)                    ,
    /* columns to represent associations */
    link_id                          INTEGER NOT NULL                 ,
    CONSTRAINT ass_link_has_link_end
      FOREIGN KEY (link_id)
      REFERENCES cls_link (item_id)                                   ,
    role_id                          INTEGER NOT NULL                 ,
    CONSTRAINT ass_link_end_role
      FOREIGN KEY (role_id)
      REFERENCES cls_relation_role (item_id)                          ,
    concept_id                       INTEGER NOT NULL                 ,
    CONSTRAINT ass_link_end_concept
      FOREIGN KEY (concept_id)
      REFERENCES cls_concept (item_id)
  );

CREATE TABLE cls_concept_constraint_set
  (
    /* cls_concept_constraint_set is a subclass of cls_constraint_set which in turn is a
subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY              ,
    constraint_set_type_id           INTEGER DEFAULT 1 NOT NULL /* 1 identifies cls_
concept_constraint_set */ ,
    CONSTRAINT cls_concept_constraint_set_is_a_subclass_of_cls_constraint_set
      FOREIGN KEY (item_id, constraint_set_type_id)
      REFERENCES cls_constraint_set (item_id, constraint_set_type_id)     ,
    /* column to represent attribute */
    concept_shall_be_member_of_concept_system_indicator
                                     BOOLEAN NOT NULL                 ,
    concept_shall_have_source_concept_system_indicator
                                     BOOLEAN NOT NULL
  );

CREATE TABLE cls_relation_constraint_set
  (
    /* cls_ relation_constraint_set is a subclass of cls_constraint_set which in turn is a
subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY              ,
    constraint_set_type_id           INTEGER DEFAULT 2 NOT NULL /* 2 identifies cls_
relation_constraint_set */,
    CONSTRAINT cls_relation_constraint_set_is_a_subclass_of_cls_constraint_set
      FOREIGN KEY (item_id, constraint_set_type_id)
      REFERENCES cls_constraint_set (item_id, constraint_set_type_id)     ,
    /* column to represent attribute */
count_of_Relation_Roles_shall_match_arity_indicator
                                     BOOLEAN NOT NULL                 ,
    arity_shall_be_greater_than_zero_indicator
                                     BOOLEAN NOT NULL
  );
```

B.2 Binary Relations metamodel

```
/* ------------------------------------------------------------------ */
/* create tables for Binary Relations metamodel                       */
```

```
/* ------------------------------------------------------------------- */
CREATE TABLE cls_binary_relation
/* subclass of cls_relation */
  (
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY             ,
    CONSTRAINT cls_binary_relation_is_a_subclass_of_cls_relation
      FOREIGN KEY (item_id)
      REFERENCES cls_relation (item_id)                                  ,
    /* columns to represent attributes */
    reflexivity                          CHARACTER VARYING(255)          ,
    symmetry                             CHARACTER VARYING(255)          ,
    transitivity                         CHARACTER VARYING(255)          ,
    /* column check constraint for enumeration of reflexivity */
    CONSTRAINT binary_relation_reflexivity_enumeration
        CHECK ( reflexivity IN
              ( 'reflexive' , 'irreflexive' , 'antireflexive' ) )        ,
    /* column check constraint for enumeration of symmetry */
    CONSTRAINT binary_relation_symmetry_enumeration
        CHECK ( symmetry IN
              ( 'symmetric' , 'asymmetric' , 'antisymmetric' ) )         ,
    /* column check constraint for enumeration of transitivity */
    CONSTRAINT binary_relation_transitivity_enumeration
        CHECK ( transitivity IN
              ( 'transitive' , 'intransitive' , 'antitransitive' ) )
  );
```

B.3 Associations in metamodels

```
/* ------------------------------------------------------------------------ */
/* create tables for the concept system associations                        */
/* ------------------------------------------------------------------------ */
CREATE TABLE ass_concept_system_membership
/* a many-to-many reflexive association */
(
    /* columns and foreign keys for the association */
    concept_id                           INTEGER
      REFERENCES cls_concept (item_id)                                   ,
    concept_system_id                    INTEGER
      REFERENCES cls_concept_system (concept_system_id)                  ,
    PRIMARY KEY (concept_id, concept_system_id)
  );

ALTER TABLE cls_concept
  ADD COLUMN
    concept_system_id                    INTEGER
  ;

ALTER TABLE cls_concept
  ADD CONSTRAINT ass_concept_source
    FOREIGN KEY (concept_system_id)
    REFERENCES cls_concept_system (item_id)
  ;

CREATE TABLE ass_concept_system_reference
/* a many-to-many reflexive association */
 (
    /* columns and foreign keys for the association */
    referenced_concept_system_id         INTEGER
      REFERENCES cls_concept_system (item_id)                            ,
    referencing_concept_system_id        INTEGER
      REFERENCES cls_concept_system (item_id)                            ,
    PRIMARY KEY (referenced_concept_system_id, referencing_concept_system_id)
  );

CREATE TABLE ass_concept_system_importation
/* a many-to-many reflexive association */
  (
    /* columns and foreign keys for the association */
    imported_concept_system_id           INTEGER
```

```
      REFERENCES cls_concept_system (item_id)                              ,
    importing_concept_system_id         INTEGER
      REFERENCES cls_concept_system (item_id)                              ,
    PRIMARY KEY (imported_concept_system_id, importing_concept_system_id)
  );

CREATE TABLE ass_assertion_inclusion
/* a many-to-many association */
  (
    /* columns and foreign keys for the association */
    concept_system_id                   INTEGER
      REFERENCES cls_concept_system (item_id)                              ,
    included_assertion_id               INTEGER
      REFERENCES cls_assertion (item_id)                                   ,
    PRIMARY KEY (concept_system_id, included_assertion_id)
  );

CREATE TABLE ass_assertion_about_concept
/* a many-to-many association */
  (
    /* columns and foreign keys for the association */
    assertion_id                        INTEGER
      REFERENCES cls_assertion (item_id)                                   ,
    concept_id                          INTEGER
      REFERENCES cls_concept (item_id)                                     ,
    PRIMARY KEY (assertion_id, concept_id)
  );

CREATE TABLE ass_assertion_of_predicate
/* a many-to-many association */
  (
    /* columns and foreign keys for the association */
    relation_id                         INTEGER
      REFERENCES cls_relation (item_id)                                    ,
    assertion_id                        INTEGER
      REFERENCES cls_assertion (item_id)                                   ,
    PRIMARY KEY (relation_id, assertion_id)
  );
```

Annex C
(informative)

Example SQL to instantiate the ISO/IEC 11179-31 metamodel

C.1 Data Element Concept metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for Data Element Concept metamodel                         */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_object_class
  (
    /* cls_object_class is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY
    item_type_id                     INTEGER DEFAULT 14 NOT NULL /* 14 identifies cls_
object_class */,
    CONSTRAINT cls_object_class_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)
  );

CREATE TABLE cls_property
  (
    /* cls_property is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY              ,
    item_type_id                     INTEGER DEFAULT 15 NOT NULL /* 15 identifies cls_
property */ ,
    CONSTRAINT cls_property_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)
  );

CREATE TABLE cls_data_element_concept
  (
    /* cls_data_element_concept is a subclass of cls_concept which in turn is a subclass of
cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY              ,
    concept_type_id                  INTEGER DEFAULT 3 NOT NULL /* 3 identifies cls_data_
element_concept */,
    CONSTRAINT cls_data_element_concept_is_a_subclass_of_cls_concept
      FOREIGN KEY (item_id, concept_type_id)
      REFERENCES cls_concept (item_id, concept_type_id)
  );

CREATE TABLE cls_data_element_concept_constraint_set
  (
    /* cls_data_element_concept_constraint_set is a subclass of cls_constraint_set which in
turn is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                          INTEGER PRIMARY KEY              ,
    constraint_set_type_id           INTEGER DEFAULT 3 NOT NULL /* 3 identifies cls_data_
element_concept_constraint_set */ ,
    CONSTRAINT cls_data_element_concept_constraint_set_is_a_subclass_of_cls_constraint_set
      FOREIGN KEY (item_id, constraint_set_type_id)
      REFERENCES cls_constraint_set (item_id, constraint_set_type_id)     ,
    /* columns to represent attributes */
    DEC_shall_have_Object_Class_indicator
                                     BOOLEAN NOT NULL                 ,
    DEC_shall_have_Property_indicator
                                     BOOLEAN NOT NULL                 ,
    DEC_shall_have_at_least_one_CD_indicator
```

```
                                        BOOLEAN NOT NULL                ,
    DEC_shall_have_at_most_one_CD_indicator
                                        BOOLEAN NOT NULL
  );
```

C.2 Measurement metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for Measurement metamodel                                  */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_unit_of_measure
  (
    /* cls_unit_of_measure is a subclass of cls_concept which in turn is a subclass of cls_
item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY             ,
    concept_type_id                     INTEGER DEFAULT 4 NOT NULL /* 4 identifies cls_unit_
of_measure */ ,
    CONSTRAINT cls_unit_of_measure_is_a_subclass_of_cls_concept
      FOREIGN KEY (item_id, concept_type_id)
      REFERENCES cls_concept (item_id, concept_type_id)
  );

CREATE TABLE cls_measure_class
  (
    /* cls_measure_class is a subclass of cls_concept which in turn is a subclass of cls_item
*/
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY             ,
    concept_type_id                     INTEGER DEFAULT 5 NOT NULL /* 5 identifies cls_
measure_class */ ,
    CONSTRAINT cls_measure_class_is_a_subclass_of_cls_concept
      FOREIGN KEY (item_id, concept_type_id)
      REFERENCES cls_concept (item_id, concept_type_id)
  );

CREATE TABLE cls_dimensionality
  (
    /* cls_dimensionality is a subclass of cls_concept which in turn is a subclass of cls_item
*/
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY             ,
    concept_type_id                     INTEGER DEFAULT 6 NOT NULL /* 6 identifies cls_
dimensionality */ ,
    CONSTRAINT cls_dimensionality_is_a_subclass_of_cls_concept
      FOREIGN KEY (item_id, concept_type_id)
      REFERENCES cls_concept (item_id, concept_type_id)             ,
    /* column to represent native attribute */
    coordinate_indicator                BOOLEAN
  );
```

C.3 Conceptual and Value_Domain metamodel

```
/* ------------------------------------------------------------------------ */
/* create tables for Conceptual and Value_Domain metamodel                  */
/* ------------------------------------------------------------------------ */
CREATE TABLE cls_conceptual_domain
  (
    /* cls_conceptual_domain is a subclass of cls_concept which in turn is a subclass of cls_
item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY             ,
    concept_type_id                     INTEGER DEFAULT 7 NOT NULL /* 7 identifies cls_
conceptual_domain */ ,
    CONSTRAINT cls_conceptual_domain_is_a_subclass_of_cls_concept
      FOREIGN KEY (item_id, concept_type_id)
      REFERENCES cls_concept (item_id, concept_type_id)             ,
    /* column to represent attribute */
    valid_period_id                     INTEGER
```

```
      REFERENCES cls_datetime_period (datetime_period_id)                ,
    dimensionality_id                   INTEGER
      REFERENCES cls_dimensionality (item_id)
  );

CREATE TABLE cls_described_conceptual_domain
  (
    /* cls_described_conceptual_domain is a subclass of cls_conceptual_domain */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY            ,
    CONSTRAINT cls_described_conceptual_domain_is_a_subclass_of_conceptual_domain
      FOREIGN KEY (item_id)
      REFERENCES cls_conceptual_domain (item_id)
  );

CREATE TABLE cls_enumerated_conceptual_domain
  (
    /* cls_enumerated_conceptual_domain is a subclass of cls_conceptual_domain */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY            ,
    CONSTRAINT cls_enumerated_conceptual_domain_is_a_subclass_of_cls_conceptual_domain
      FOREIGN KEY (item_id)
      REFERENCES cls_conceptual_domain (item_id)
  );

CREATE TABLE cls_local_enumerated_conceptual_domain
  (
    /* cls_local_enumerated_conceptual_domain is a subclass of cls_enumerated_conceptual_
domain */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY            ,
    CONSTRAINT cls_local_enumerated_conceptual_domain_is_a_subclass_of_cls_enumerated_
conceptual_domain
      FOREIGN KEY (item_id)
      REFERENCES cls_enumerated_conceptual_domain (item_id)
  );

CREATE TABLE cls_value_meaning
  (
    /* cls_value_meaning is a subclass of cls_concept */
    /* inherited surrogate primary key */
    item_id INTEGER PRIMARY KEY ,
    concept_type_id                     INTEGER DEFAULT 8 NOT NULL /* 8 identifies cls_value_
meaning */ ,
    CONSTRAINT cls_value_meaning_is_a_subclass_of_cls_concept
      FOREIGN KEY (item_id, concept_type_id)
      REFERENCES cls_concept (item_id, concept_type_id)
  );

CREATE TABLE cls_reference_enumerated_conceptual_domain
  (
    /* cls_reference_enumerated_conceptual_domain is a subclass of cls_enumerated_conceptual_
domain */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY            ,
    CONSTRAINT cls_reference_enumerated_conceptual_domain_is_a_subclass_of_cls_enumerated_
conceptual_domain
      FOREIGN KEY (item_id)
      REFERENCES cls_enumerated_conceptual_domain (item_id)
  );

CREATE TABLE cls_enumerated_conceptual_domain_definition
  (
    /* cls_enumerated_conceptual_domain_definition is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY            ,
    item_type_id                        INTEGER DEFAULT 16 NOT NULL /* 16 identifies cls_
enumerated_conceptual_domain_definition */ ,
    CONSTRAINT cls_enumerated_conceptual_domain_definition_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                        ,
    /* columns to represent attributes */
```

```
    expression                          CHARACTER VARYING(2500)          ,
    notation                            CHARACTER VARYING(2500)          ,
    source_uri                          CHARACTER VARYING(255)           ,
    /* column to represent association */
    enumerated_conceptual_domain_id     INTEGER                          ,
    CONSTRAINT ass_domain_definition
      FOREIGN KEY (enumerated_conceptual_domain_id)
      REFERENCES cls_enumerated_conceptual_domain (item_id)
 );

CREATE TABLE cls_datatype_scheme
  (
    /* cls_datatype_scheme is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY              ,
    item_type_id                        INTEGER DEFAULT 17 NOT NULL /* 17 identifies cls_
datatype_scheme */ ,
    CONSTRAINT cls_datatype_scheme_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                        ,
    /* column to represent attribute */
    scheme_reference                     CHARACTER VARYING(255)
  );

CREATE TABLE cls_datatype
  (
    /* cls_datatype is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY              ,
    item_type_id                        INTEGER DEFAULT 18 NOT NULL /* 18 identifies cls_
datatype */ ,
    CONSTRAINT cls_datatype_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                        ,
    /* column to represent attribute */
    annotation                          CHARACTER VARYING(2500)          ,
    /* column to represent association */
    datatype_scheme_id                   INTEGER                         ,
    CONSTRAINT ass_datatype_has_scheme_reference
      FOREIGN KEY (datatype_scheme_id)
      REFERENCES cls_datatype_scheme (item_id)
  );

CREATE TABLE cls_value_domain
  (
    /* cls_value_domain is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY              ,
    item_type_id                        INTEGER DEFAULT 19 NOT NULL /* 19 identifies cls_
value_domain */ ,
    CONSTRAINT cls_value_domain_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                        ,
    /* columns to represent attributes */
    datatype_id                         INTEGER
      REFERENCES cls_datatype (item_id)                                  ,
    valid_period_id                     INTEGER
      REFERENCES cls_datetime_period(datetime_period_id)                 ,
    "format"                            CHARACTER VARYING(255)           ,
    maximum_character_quantity          INTEGER                          ,
    unit_of_measure_id                  INTEGER
      REFERENCES cls_unit_of_measure (item_id)                           ,
    /* column to represent association */
    conceptual_domain_id                INTEGER                          ,
    CONSTRAINT ass_value_domain_meaning
      FOREIGN KEY (conceptual_domain_id)
      REFERENCES cls_conceptual_domain (item_id)
  );

CREATE TABLE cls_described_value_domain
  (
    /* cls_described_value_domain is a subclass of cls_value_domain which is a subclass of
```

```
cls_item */
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY                ,
    CONSTRAINT cls_described_value_domain_is_a_subclass_of_cls_value_domain
      FOREIGN KEY (item_id)
      REFERENCES cls_value_domain (item_id)                                  ,
    /* column to represent association */
    described_conceptual_domain_id       INTEGER                            ,
    CONSTRAINT ass_described_value_domain_meaning
      FOREIGN KEY (described_conceptual_domain_id)
      REFERENCES cls_described_conceptual_domain (item_id)
  );

CREATE TABLE cls_enumerated_value_domain
  (
    /* cls_enumerated_value_domain is a subclass of cls_value_domain which is a subclass of
cls_item */
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY                ,
    CONSTRAINT cls_enumerated_value_domain_is_a_subclass_of_cls_value_domain
      FOREIGN KEY (item_id)
      REFERENCES cls_value_domain (item_id)
  );

CREATE TABLE cls_local_enumerated_value_domain
  (
    /* cls_local_enumerated_value_domain is a subclass of cls_enumerated_value_domain which is
a subclass of cls_value_domain which is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY                ,
    CONSTRAINT cls_local_enumerated_value_domain_is_a_subclass_of_cls_enumerated_value_domain
      FOREIGN KEY (item_id)
      REFERENCES cls_enumerated_value_domain (item_id)
  );

CREATE TABLE cls_permissible_value
  (
    /* cls_permissible_value is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                              INTEGER PRIMARY KEY                ,
    item_type_id                         INTEGER DEFAULT 20 NOT NULL /* 20 identifies cls_
permissible_value */ ,
    CONSTRAINT cls_permissible_value_is_a_subclass_of_cls_item
      FOREIGN KEY (item_id, item_type_id)
      REFERENCES cls_item (item_id, item_type_id)                            ,
    /* the permitted_value attribute has datatype 'value' which can be of any */
    /* datatype - this is instantiated by a different column for each datatype */
    /* and a check constraint to ensure that only one 'datatype' has a value */
    boolean_permitted_value               BOOLEAN                           ,
    date_permitted_value                  DATE                              ,
    datetime_permitted_value              TIMESTAMP                         ,
    integer_permitted_value               INTEGER                           ,
    lower_bound_natural_range_permitted_value
                                          INTEGER                           ,
    fixed_upper_bound_natural_range_permitted_value
                                          INTEGER                           ,
    many_upper_bound_natural_range_permitted_value
                                          CHARACTER(4)                      ,
    notation_permitted_value              CHARACTER VARYING(2500)           ,
    phone_number_permitted_value          INTEGER
                          REFERENCES cdt_phone_number (phone_number_id)     ,
    postal_address_permitted_value        INTEGER
                    REFERENCES cdt_postal_address (postal_address_id)       ,
    sign_permitted_value                  CHARACTER VARYING(2500)           ,
    string_permitted_value                CHARACTER VARYING(255)            ,
    text_permitted_value                  CHARACTER VARYING(2500)           ,
    /* columns to represent other attributes */
    begin_date                            DATE                              ,
    end_date                              DATE                              ,
    /* check constraint to manage datatype columns */
    CONSTRAINT datatype_of_permitted_value
      CHECK
```

```
( ( ( boolean_permitted_value is NOT NULL )
      AND ( date_permitted_value IS NULL )
      AND ( datetime_permitted_value IS NULL )
      AND ( integer_permitted_value IS NULL )
      AND ( lower_bound_natural_range_permitted_value IS NULL )
      AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
      AND ( many_upper_bound_natural_range_permitted_value IS NULL )
      AND ( phone_number_permitted_value IS NULL )
      AND ( postal_address_permitted_value IS NULL )
      AND ( sign_permitted_value IS NULL )
      AND ( string_permitted_value IS NULL )
      AND ( text_permitted_value IS NULL ) )
  OR
  ( ( boolean_permitted_value is NULL )
      AND ( date_permitted_value IS NOT NULL )
      AND ( datetime_permitted_value IS NULL )
      AND ( integer_permitted_value IS NULL )
      AND ( lower_bound_natural_range_permitted_value IS NULL )
      AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
      AND ( many_upper_bound_natural_range_permitted_value IS NULL )
      AND ( phone_number_permitted_value IS NULL )
      AND ( postal_address_permitted_value IS NULL )
      AND ( sign_permitted_value IS NULL )
      AND ( string_permitted_value IS NULL )
      AND ( text_permitted_value IS NULL ) )
  OR
  ( ( boolean_permitted_value is NULL )
      AND ( date_permitted_value IS NULL )
      AND ( datetime_permitted_value IS NOT NULL )
      AND ( integer_permitted_value IS NULL )
      AND ( lower_bound_natural_range_permitted_value IS NULL )
      AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
      AND ( many_upper_bound_natural_range_permitted_value IS NULL )
      AND ( phone_number_permitted_value IS NULL )
      AND ( postal_address_permitted_value IS NULL )
      AND ( sign_permitted_value IS NULL )
      AND ( string_permitted_value IS NULL )
      AND ( text_permitted_value IS NULL ) )
  OR
  ( ( boolean_permitted_value is NULL )
      AND ( date_permitted_value IS NULL )
      AND ( datetime_permitted_value IS NULL )
      AND ( integer_permitted_value IS NOT NULL )
      AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
      AND ( many_upper_bound_natural_range_permitted_value IS NULL )
      AND ( phone_number_permitted_value IS NULL )
      AND ( postal_address_permitted_value IS NULL )
      AND ( sign_permitted_value IS NULL )
      AND ( string_permitted_value IS NULL )
      AND ( text_permitted_value IS NULL ) )
  OR
  ( ( boolean_permitted_value is NULL )
      AND ( date_permitted_value IS NULL )
      AND ( datetime_permitted_value IS NULL )
      AND ( integer_permitted_value IS NULL )
      AND ( lower_bound_natural_range_permitted_value IS NOT NULL )
      AND ( lower_bound_natural_range_permitted_value >= 0 )
      AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
      AND ( many_upper_bound_natural_range_permitted_value IS NULL )
      AND ( phone_number_permitted_value IS NULL )
      AND ( postal_address_permitted_value IS NULL )
      AND ( sign_permitted_value IS NULL )
      AND ( string_permitted_value IS NULL )
      AND ( text_permitted_value IS NULL ) )
  OR
  ( ( boolean_permitted_value is NULL )
      AND ( date_permitted_value IS NULL )
      AND ( datetime_permitted_value IS NULL )
      AND ( integer_permitted_value IS NULL )
      AND ( lower_bound_natural_range_permitted_value IS NOT NULL )
      AND ( lower_bound_natural_range_permitted_value >= 0 )
      AND ( fixed_upper_bound_natural_range_permitted_value IS NOT NULL )
```

```
            AND (fixed_upper_bound_natural_range_permitted_value >= lower_bound_natural_
range_permitted_value )
            AND ( many_upper_bound_natural_range_permitted_value IS NULL )
            AND ( phone_number_permitted_value IS NULL )
            AND ( postal_address_permitted_value IS NULL )
            AND ( sign_permitted_value IS NULL )
            AND ( string_permitted_value IS NULL )
            AND ( text_permitted_value IS NULL ) )
        OR
        ( ( boolean_permitted_value is NULL )
            AND ( date_permitted_value IS NULL )
            AND ( datetime_permitted_value IS NULL )
            AND ( integer_permitted_value IS NULL )
            AND ( lower_bound_natural_range_permitted_value IS NOT NULL )
            AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
            AND ( many_upper_bound_natural_range_permitted_value IS NOT NULL )
            AND ( phone_number_permitted_value IS NULL )
            AND ( postal_address_permitted_value IS NULL )
            AND ( sign_permitted_value IS NULL )
            AND ( string_permitted_value IS NULL )
            AND ( text_permitted_value IS NULL ) )
        OR
        ( ( boolean_permitted_value is NULL )
            AND ( date_permitted_value IS NULL )
            AND ( datetime_permitted_value IS NULL )
            AND ( integer_permitted_value IS NULL )
            AND ( lower_bound_natural_range_permitted_value IS NULL )
            AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
            AND ( many_upper_bound_natural_range_permitted_value IS NULL )
            AND ( phone_number_permitted_value IS NOT NULL )
            AND ( postal_address_permitted_value IS NULL )
            AND ( sign_permitted_value IS NULL )
            AND ( string_permitted_value IS NULL )
            AND ( text_permitted_value IS NULL ) )
        OR
        ( ( boolean_permitted_value is NULL )
            AND ( date_permitted_value IS NULL )
            AND ( datetime_permitted_value IS NULL )
            AND ( integer_permitted_value IS NULL )
            AND ( lower_bound_natural_range_permitted_value IS NULL )
            AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
            AND ( many_upper_bound_natural_range_permitted_value IS NULL )
            AND ( phone_number_permitted_value IS NULL )
            AND ( postal_address_permitted_value IS NOT NULL )
            AND ( sign_permitted_value IS NULL )
            AND ( string_permitted_value IS NULL )
            AND ( text_permitted_value IS NULL ) )
        OR
        ( ( boolean_permitted_value is NULL )
            AND ( date_permitted_value IS NULL )
            AND ( datetime_permitted_value IS NULL )
            AND ( integer_permitted_value IS NULL )
            AND ( lower_bound_natural_range_permitted_value IS NULL )
            AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
            AND ( many_upper_bound_natural_range_permitted_value IS NULL )
            AND ( phone_number_permitted_value IS NULL )
            AND ( postal_address_permitted_value IS NULL )
            AND ( sign_permitted_value IS NOT NULL )
            AND ( string_permitted_value IS NULL )
            AND ( text_permitted_value IS NULL ) )
        OR
        ( ( boolean_permitted_value is NULL )
            AND ( date_permitted_value IS NULL )
            AND ( datetime_permitted_value IS NULL )
            AND ( integer_permitted_value IS NULL )
            AND ( lower_bound_natural_range_permitted_value IS NULL )
            AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
            AND ( many_upper_bound_natural_range_permitted_value IS NULL )
            AND ( phone_number_permitted_value IS NULL )
            AND ( postal_address_permitted_value IS NULL )
            AND ( sign_permitted_value IS NULL )
            AND ( string_permitted_value IS NOT NULL )
```

```
            AND ( text_permitted_value IS NULL ) )
        OR
        ( ( boolean_permitted_value is NULL )
            AND ( date_permitted_value IS NULL )
            AND ( datetime_permitted_value IS NULL )
            AND ( integer_permitted_value IS NULL )
            AND ( lower_bound_natural_range_permitted_value IS NULL )
            AND ( fixed_upper_bound_natural_range_permitted_value IS NULL )
            AND ( many_upper_bound_natural_range_permitted_value IS NULL )
            AND ( phone_number_permitted_value IS NULL )
            AND ( postal_address_permitted_value IS NULL )
            AND ( sign_permitted_value IS NULL )
            AND ( string_permitted_value IS NULL )
            AND ( text_permitted_value IS NOT NULL ) ) )                    ,
    CONSTRAINT permissible_value_cannot_have_an_end_date_without_a_begin_date
      CHECK ( ( end_date IS NULL ) OR ( begin_date IS NOT NULL ) )
  );

CREATE TABLE cls_reference_enumerated_value_domain
  (
    /* cls_reference_enumerated_value_domain is a subclass of cls_value_domain */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY                 ,
    CONSTRAINT cls_reference_enumerated_value_domain_is_a_subclass_of_cls_value_domain
      FOREIGN KEY (item_id)
      REFERENCES cls_value_domain (item_id)                                 ,
    /* column to represent attribute */
    permitted_values_fieldname           CHARACTER VARYING(255)             ,
    /* column to represent association */
    reference_enumerated_conceptual_domain_id
                                        INTEGER                             ,
    CONSTRAINT ass_reference_enumerated_value_domain_meaning
      FOREIGN KEY (reference_enumerated_conceptual_domain_id)
      REFERENCES cls_reference_enumerated_conceptual_domain (item_id)
  );

CREATE TABLE cls_reference_enumerated_cd_constraint_set
  (
    /* cls_reference_enumerated_cd_constraint_set is a subclass of cls_constraint_set which in
turn is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY                 ,
    constraint_set_type_id              INTEGER DEFAULT 4 NOT NULL /* 4 identifies cls_
reference_enumerated_cd_constraint_set */ ,
    CONSTRAINT cls_reference_enumerated_cd_constraint_set_is_a_subclass_of_cls_constraint_set
      FOREIGN KEY (item_id, constraint_set_type_id)
      REFERENCES cls_constraint_set (item_id, constraint_set_type_id)     ,
    /* column to represent attribute */
    Reference_Enumerated_CD_shall_have_at_least_one_Enumerated_CD_Definition_indicator
                                        BOOLEAN NOT NULL
  );

CREATE TABLE cls_local_enumerated_cd_constraint_set
  (
    /* cls_local_enumerated_cd_constraint_set is a subclass of cls_constraint_set which in
turn is a subclass of cls_item */
    /* inherited surrogate primary key */
    item_id                             INTEGER PRIMARY KEY                 ,
    constraint_set_type_id              INTEGER DEFAULT 5 NOT NULL /* 5 identifies cls_local_
enumerated_cd_constraint_set */ ,
    CONSTRAINT cls_local_enumerated_cd_constraint_set_is_a_subclass_of_cls_constraint_set
      FOREIGN KEY (item_id, constraint_set_type_id)
      REFERENCES cls_constraint_set (item_id, constraint_set_type_id)      ,
    /* column to represent attribute */
    Local_Enumerated_CD_shall_have_at_least_one_Value_Meaning_indicator
                                        BOOLEAN NOT NULL
  );

CREATE TABLE cls_value_domain_constraint_set
  (
    /* cls_value_domain_constraint_set is a subclass of cls_constraint_set which in turn is a
subclass of cls_item */
```