

## 2. CONCEPTS AND SPECIFICATIONS

Example 2.6.5.4. *Related categories linked by parent–child relationships.*

```

save_CITATION
_category.description
; Data items in the CITATION category record details
about the literature cited as being relevant to
the contents of the data block.
;
_category.id                citation
_category.mandatory_code    no
_category_key.name          '_citation.id'
_loop__category_group.id    'inclusive_group'
                           'citation_group'
# ----- abbreviated definition -----
save_

save__citation.id
_item.description.description
; The value of _citation.id must uniquely identify a
record in the CITATION list. The _citation.id
'primary' should be used to indicate the citation
that the author(s) consider to be the most
pertinent to the contents of the data block.
;
_loop__item.name
      _item.category_id
      _item.mandatory_code
'_citation.id'                citation          yes
'_citation_author.citation_id' citation_author yes
'_citation_editor.citation_id' citation_editor yes
'_software.citation_id'       software          yes
_item_aliases.alias.name     '_citation_id'
_item_aliases.dictionary      cif_core.dic
_item_aliases.version         2.0.1
_loop__item_linked.child_name
      _item_linked.parent_name
'_citation_author.citation_id' '_citation.id'
'_citation_editor.citation_id' '_citation.id'
'_software.citation_id'       '_citation.id'
_item_type.code              code
_loop__item_examples.case     'primary' '1' '2'
save_

save_CITATION_AUTHOR
_category.description
; Data items in the CITATION_AUTHOR category record
details about the authors associated with the
citations in the CITATION list.
;
_category.id                citation_author
_category.mandatory_code    no
_loop__category_key.name    '_citation_author.citation_id'
                           '_citation_author.name'
_loop__category_group.id    'inclusive_group'
                           'citation_group'
# ----- abbreviated definition -----
save_

save__citation_author.citation_id
_item.description.description
; This data item is a pointer to _citation.id in the
CITATION category.
;
_item.name                   '_citation_author.citation_id'
_item.mandatory_code         yes
_item_aliases.alias.name     '_citation_author.citation_id'
_item_aliases.dictionary      cif_core.dic
_item_aliases.version         2.0.1
save_

```

category identifier. The full name has been used here in order to provide compatibility with existing applications.

The item category also includes a code to indicate whether a data item is mandatory in a category and therefore must be included in any tuple of items in the category. This code, `_item.mandatory_code`, may have three values: `yes`, `no` and `implicit`. This last named value indicates that the item is manda-

tory, but that the value of this item may be derived from the context. In the case of an item name or a category identifier, these values can be obtained from the current save-frame name. Implicit specification dramatically simplifies the appearance of each dictionary definition because it avoids the repeated declaration of item names and category identifiers that are basis components or the unique identifiers for most categories.

Although the data item `_item.name` is the basis for all of the item-level categories, its definition and properties need only be specified at a single point. Here, the data items that occur in multiple categories are defined only in the parent category. In certain situations, a child data item may be used in a manner which requires a description distinct from the parent data item. For instance, `_item_linked.parent_name` and `_item_linked.child_name` are both data-item names as well as children of `_item.name`, but clearly the manner in which these items are used in the `ITEM_LINKED` category requires additional description. It is important to note that although the design of this DDL supports the definition of data items in multiple categories within the parent category, it is also possible to provide separate complete definitions within each category.

#### 2.6.6.1.2. *ITEM\_ALIASES*

The DDL category `ITEM_ALIASES` defines the alias names that can be substituted for a data-item name. The alias mechanism also provides a means of identifying items by names other than those that follow the naming conventions used in this DDL. This feature should be used primarily to guarantee the stability of names defined in previously published dictionaries. The items `_item_aliases.name`, `_item_aliases.dictionary` and `_item_aliases.version` form the key for this category. The items `_item_aliases.dictionary` and `_item_aliases.version` are provided to distinguish between dictionaries and different versions of the same dictionary. Any number of unique alias names can be defined for a data item.

#### 2.6.6.1.3. *ITEM\_DEFAULT*

The DDL category `ITEM_DEFAULT` holds default values assigned to data items. Default data values are specified in item `_item_default.value`. Default values are assigned to data items that are not declared within a category. The key item for this category, `_item_default.name`, is a child of `_item.name`. A single default value may be specified for a data item.

#### 2.6.6.1.4. *ITEM\_DEPENDENT*

The `ITEM_DEPENDENT` category defines dependency relationships among data items within a category. Each data item on which a particular data item depends is specified as an item `_item_dependent.dependent_name`. For a data item to be considered completely defined, each of its dependent data items must also be specified.

#### 2.6.6.1.5. *ITEM\_DESCRIPTION*

The DDL category `ITEM_DESCRIPTION` holds a description for each data item. The key item for this category is `_item_description.name`, which is defined in the parent category `ITEM`. The text of the item description is held by data item `_item_description.description`. A single description may be provided for each data item.