

2.6. SPECIFICATION OF A RELATIONAL DICTIONARY DEFINITION LANGUAGE (DDL2)

Because special rules exist in the STAR grammar for the specification of data items that belong to a common category, the organization of data items within categories has a significant influence on how these items may be expressed in a data file. For example, a data category may be specified only once within a STAR data block or save frame, and at any level of a STAR loop structure only data items of a common category may appear.

2.6.6.2.2. *CATEGORY_EXAMPLES*

The category named *CATEGORY_EXAMPLES* holds examples that apply to an entire category. This typically includes a complete specification of the category with annotations. An example specification consists of the text of the example, *_category_examples.case*, and an optional comment item, *_category_examples.detail*, which can be used to qualify the example. The key for this category includes the items *_category_examples.id* and *_category_examples.case*. The former is completely defined in the parent category named *CATEGORY*.

2.6.6.2.3. *CATEGORY_GROUP*

The category *CATEGORY_GROUP* names the category groups to which a category belongs. The assignment of a category to a category group is made when the category is defined. Each category group that is specified in this category must also be defined in the parent category, *CATEGORY_GROUP_LIST*. The basis for this category also includes the category identifier *_category_group.category_id*, which is completely defined in the parent category named *CATEGORY*.

2.6.6.2.4. *CATEGORY_GROUP_LIST*

The DDL category *CATEGORY_GROUP_LIST* holds data items that define category groups. Category groups are collections of related categories. Parent-child relationships may be defined for these groups. The specification of category groups and the relationships between these groups allow a complicated collection of categories to be organized into a hierarchy of more relevant groups. This higher level of structure is essential for large application dictionaries that may contain hundreds of category definitions.

The category *CATEGORY_GROUP_LIST* holds the description of each category group, *_category_group_list.description*, and an optional identifier of the parent group, *_category_group_list.parent_id*. Category groups can be formed from collections of base categories, those categories that hold data. Category groups can also be formed from collections of base categories and category groups.

Example 2.6.6.2 illustrates the category groups that are defined in this DDL. These include the group of categories that define categories, the group of categories defining data items and the group of categories that define properties of the dictionary. An additional compliance group is also defined for categories that are included specifically for compliance with previous versions of DDL. Each of these category groups is defined as a child of the group named *ddl_group* to which all of the base DDL categories belong.

2.6.6.2.5. *CATEGORY_KEY*

The category *CATEGORY_KEY* identifies the data items within a category that form the basis for the category. The category basis uniquely identifies each group or tuple of items in the category. In the analogy of the category as a table, no row in a table may have duplicate values for its key data items.

Example 2.6.6.2. *Category groups defined in the DDL2 dictionary.*

```

loop_
  _category_group_list.id
  _category_group_list.parent_id
  _category_group_list.description

  'ddl_group'          .
;
Component categories of the macromolecular DDL.
;
'datablock_group'    'ddl_group'
;
Categories that describe the characteristics of
data blocks.
;
'category_group'     'ddl_group'
;
Categories that describe the characteristics of
categories.
;
'sub_category_group' 'ddl_group'
;
Categories that describe the characteristics of
subcategories.
;
'item_group'         'ddl_group'
;
Categories that describe the characteristics of
data items.
;
'dictionary_group'   'ddl_group'
;
Categories that describe the dictionary.
;
'compliance_group'   'ddl_group'
;
Categories that are retained specifically for
compliance with older versions of the DDL.
;

```

The choice of basis has important consequences in the specification of a category. It is important to ensure that the key items that form the category basis can unambiguously identify any tuple of data items within the category. If this is not the case, then it may not be possible to reliably recover data items that are stored in the category. Because key items are required to address each tuple of items in a category, key items are considered mandatory items in the category.

It is interesting to note how the key data items have been selected for the categories that define the DDL, and how this choice of key items influences the structure of the DDL dictionary. In the DDL category *CATEGORY_KEY*, the basis includes both the identifier for the category, *_category_key.id*, and the name of the key data item, *_category_key.name*. This choice of basis allows for any unique groups of items in a category to be defined as key items. Duplicate key-item values within a category are forbidden by the data model. In the DDL category *ITEM_TYPE*, the basis includes only the identifier for the item name, *_item_type.name*. This choice of basis has the desired effect of limiting the specification of item data type, *_item_type.code*, to a single choice for each data item.

2.6.6.2.6. *CATEGORY_METHODS*

The *CATEGORY_METHODS* category is used to associate method identifiers with categories. Any number of unique method identifiers may be associated with a category. The method identifiers reference the full method definitions in the parent *METHOD_LIST* category.