

资产卡片数据规范应用数据库示例说明

本说明展示了在非关系数据库 MongoDB 系统中建立资产卡片数据库的方法，并提供资产数据添加的流程，供资产卡片管理系统的设计者和开发者参考。对于本示例有以下几点说明：

- (1) 为了便于表述和理解本附录仅选取具有代表性的字段进行说明，集合的具体字段见正文。
- (2) 本附录所给出的 json 全部为示例仅是为了方便说明所展示的一条信息。
- (3) 基于 MongoDB 的插入数据即建立集合的特性，本示例省略数据库建立步骤，如使用其它数据库系统建立数据库请参考 A1 数据库各集合说明自行摸索数据库建立方法。
- (4) 数据集合中的关联指针（关联字段）可以用被关联的集合的任何一个字段，但要保证被关联字段的唯一性。并且在同一个集合中关联字段要保持一致。建议采用 id 关联，但本附录为了便于理解都采用关联名称的方式进行关联。

A1 数据库各集合说明

使用一维编码多维分类的资产描述方法,在非关系数据库系统中建立数据库共需建立四个集合，分别为实际资产信息集合、资产类别信息集合、元属性集合、维度信息集合。其关系如图 A1-1 所示。

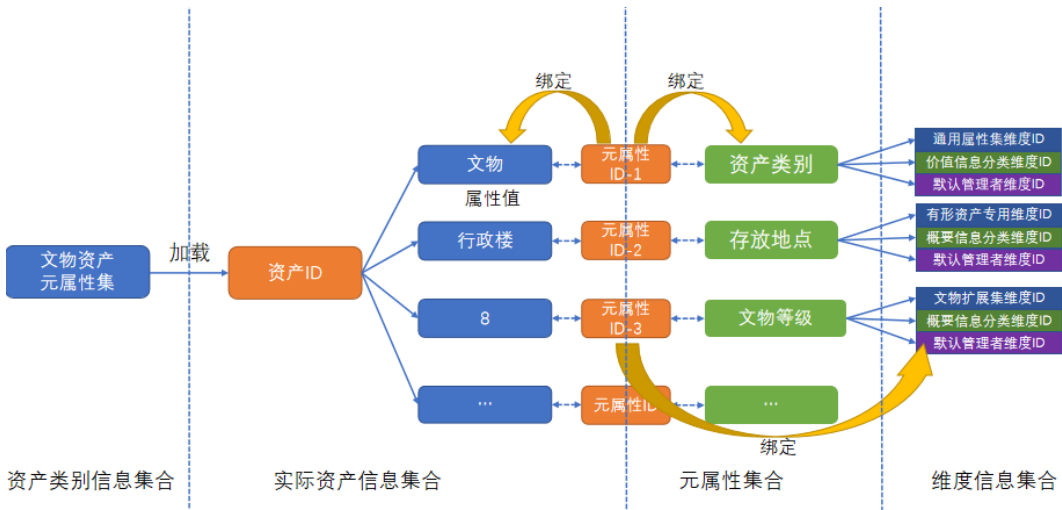


图 A1-1 实际资产信息与各集合之间映射表

A1.1 实际资产信息集合

【实际资产信息集合】用于存储实际资产的全部信息。其 json 表示如下（以一条实际资产为例）：

```
{  "_id" : ObjectId("5cda8b03135b111ab69ece4d"),
  "Asset_Coding" : {    "v" : "a600001",    "mp" : "资产代码"  },
  "Asset_Type" : {    "v" : "文物",    "mp" : "资产类别"  },
```

```

"Unit_of_measurement" : {      "v" : "个",      "mp" : "计量单位"  },
"Book_Value" : {      "v" : "15000",      "mp" : "账面价值"  },
"User" :      {      "v" : "xxx",      "mp" : "使用人"  },
"ZY_SX" : [{      "v" : "20*45",      "mp" : "规格"  },
              {      "v" : "行政楼",      "mp" : "存放地点"  }],
"KJ_SX" : [{      "v" : "8",      "mp" : "文物等级"  },
              {      "v" : "1995",      "mp" : "藏品年代"  }],
"bz" : "" }

```

其中\_id 为资产 id。Asset\_Coding、Asset\_Name、Unit\_of\_measurement、Book\_Value、User 为资产通用属性集的部分属性。ZY\_SX 为资产专用属性集列表，用于存储资产专用属性信息。KJ\_SX 为资产扩展属性集合列表，用于存储资产扩展属性信息。bz 是用于存放备注信息的字段。

对于实际资产信息集合中的每一个字段都含有两个键值对为 V 和 mp，其中 V 为属性值、mp 用于存放指向元属性的关联字段。这里为了便于表述和查询关联了元属性表的 Name 字段，也可以关联\_id 字段。

#### A1.2 资产类别信息集合

【资产类别信息集合】与实例资产信息集合类似，但不含有资产的通用集信息和属性值。资产类别信息集合是实际资产信息的模板。其 json 表示如下：

```

{  "_id" : ObjectId("5cda899a135b111ab69ecd3d"),
  "Name" : "文物",
  "ZY_SX" : ["规格", "存放地点"],
  "KZ_SX" : ["文物等级", "藏品年代"],
  "bz" : "" }

```

资产类别信息集合共包含五个字段，分别为\_id、Name、ZY\_SX、KZ\_SX、bz。其中\_id 为资产类别 id，Name 为资产类别名称，ZY\_SX 为存储了专用属性关联指针列表，用于关联到元属性集合中的 Name 字段，KZ\_SX 存储了扩展属性的列表，列表中的每一项都是一个关联指针，用于关联元属性集合中的 Name 字段。bz 是用于存放备注信息的字段

#### A1.3 元属性集合

【元属性集合】用于存放元属性的全部信息。其 json 表示如下：

```

{  "_id" : ObjectId("5cda872d135b111ab69ecbd3"),
  "Name" : "资产代码",
  "Data_Type" : "整型",
  "Unit" : "",
  "Length" : "30",
  "Edition" : "v0.1",
  "bz" : "",

```

```

"Dimension_dic" : {
    "资产类别维度" : "通用集",
    "管理者维度" : "默认管理者",
    "应用特性维度" : "概要信息"
}

```

其中\_id 为元属性 id。Name 为元属性名称。Data\_Type 为元属性值的数据类型。Unit 为元属性计量单位。Length 为元属性值得字符长度。Edition 为元属性版本号。bz 存储备注信息。Dimension\_dic 是一个维度信息字典其键为维度类别名称，值存储的是维度信息名称，值是一个关联指针用于关联维度信息的 Name 字段。

#### A1.5 维度信息集合

【维度信息集合】用于存储关于维度的全部信息。这里给出两条 json 分别为维度信息和维度类别，具体如下：

##### 1、维度类别

```

{
  "_id" : "0001",
  "Name" : "资产类别维度",
  "TYPE" : "all",
  "F_ID" : "",
  "bz" : ""
}

```

##### 2、维度信息

```

{
  "_id" : "1100",
  "Name" : "有形资产专用集",
  "TYPE" : "资产类别维度",
  "F_ID" : "",
  "bz" : ""
}

```

维度信息集合共五个字段，分别为\_id、Name、TYPE、F\_ID、bz。\_id 为维度 id 与前几个集合不同这个 id 采用自定义的方式其编码方法见正文。Name 表示维度名称/维度信息名称。TYPE 表示这条信息所在的维度，当其为”all”时表示这条信息是维度类别信息，否则 TYPE 为维度类别名称。F\_ID 存储父类 id 信息。bz 存储备注信息。

#### A2 资产卡片信息查询

资产卡片信息的查询包括实际资产查询、资产类别信息查询、元属性查询、维度信息查询。其中维度信息查询就是简单的数据库查询，查询方法简单，不再赘述。实际资产查询、资产类别信息查询、元属性查询都涉及到 MongoDB 的聚合管道功能，在 A1 数据库各集合说明中已经指出其关联的表和字段，本附录以查询实际资产信息集合给出一个示例，其余两个集合的查询与之相同。

示例（查询所有资产中资产名称为“元青花瓷瓶”的全部信息（包括元属性信息））：

```

db.getCollection('ZC_data').aggregate([
  {

```

```

$match:{
    'Asset_Name.V':'元青花瓷瓶',
}
},{
$lookup:
{
    from: 'Meta_attribute',
    localField: 'KJ_SX.mp',
    foreignField: 'Name',
    as: 'KJ_SX'
}
})

```

查询结果如图 A2-1 所示。

▼ (1) ObjectId("5cda8b03135b111ab69ece4d")	{ 9 fields }	Object
_id	ObjectId("5cda8b03135b111ab69ece4d")	ObjectId
> Asset_Coding	{ 2 fields }	Object
▼ Asset_Name	{ 2 fields }	Object
V	元青花瓷瓶	String
mp	资产名称	String
> Unit_of_measurement	{ 2 fields }	Object
> Book_Value	{ 2 fields }	Object
> User	{ 2 fields }	Object
> Specifications	{ 2 fields }	Object
> Storage_location	{ 2 fields }	Object
▼ KJ_SX	[ 2 elements ]	Array
▼ [0]	{ 9 fields }	Object
_id	ObjectId("5cda8848135b111ab69ecc83")	ObjectId
ID	008	String
Name	藏品年代	String
Data_Type	整型	String
Unit	年	String
Length	64	String
Edition	v1.1	String
Remarks		String
> Dimension_dic	{ 3 fields }	Object
> [1]	{ 9 fields }	Object

图 A2-1 查询结果图

### A3 资产卡片信息添加

资产卡片信息添加是一个复杂的过程需要考虑各个表的关系,并且涉及到每个表信息的添加与查询,其流程图如图 A3-1 所示:

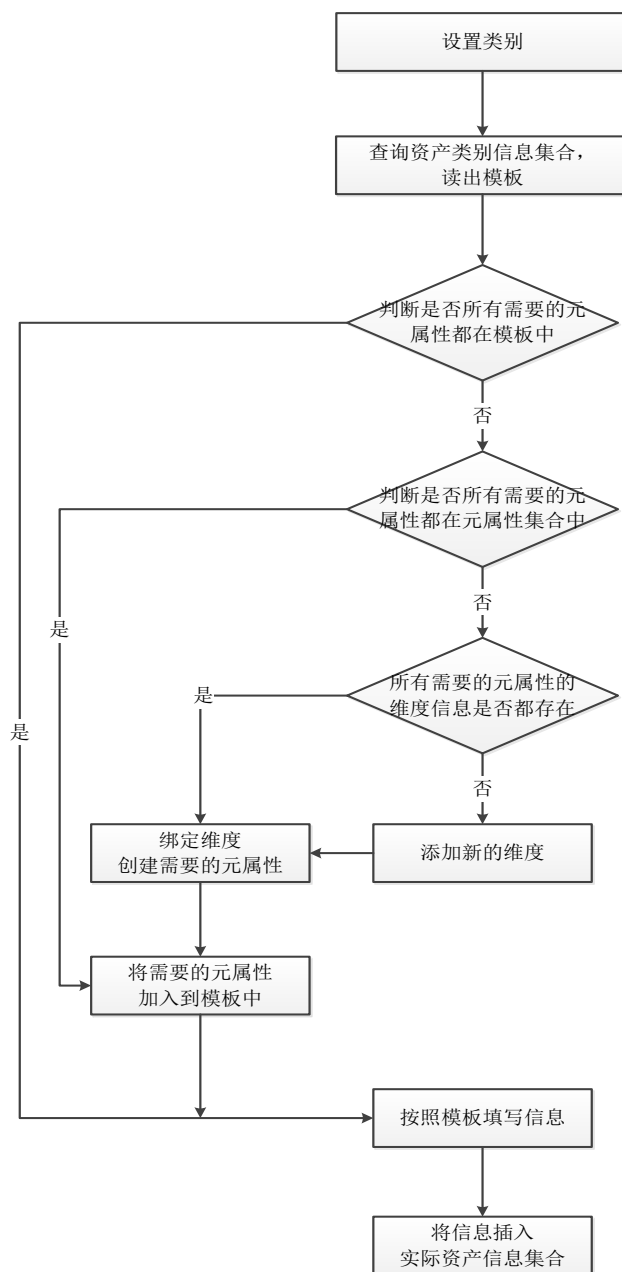


图 A3-1 实际资产添加流程图

其中查询部分已经在 A2 中稍加说明，各个表的数据添加也仅是 MongoDB 基础内容，本附录给出一条添加维度信息的例子，其余各集合添加信息与之类似，不再赘述。

示例（添加一条资产类别维度的通用集维度信息，其 id 为 1000）：

```

db.getCollection('Dimension_Information').insert({
  "_id" : "1000",
  "NAME" : "通用集",
  "TYPE" : "资产类别维度",
  "F_ID" : "",
  "Remarks" : ""})
  
```