

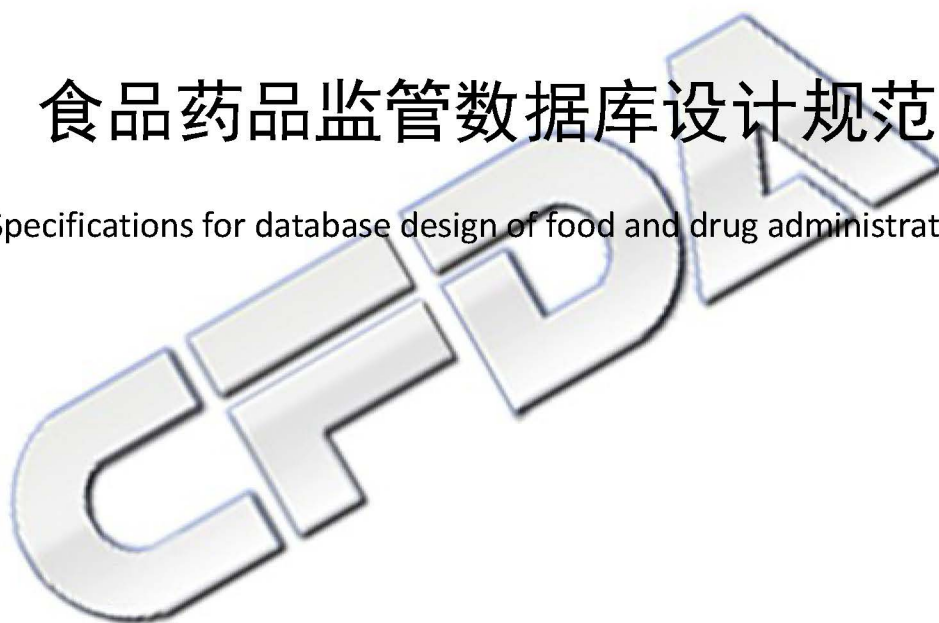
CFDAB

国家食品药品监管信息化标准

CFDAB/T 0501—2014

食品药品监管数据库设计规范

Specifications for database design of food and drug administration



2014-02-14 发布

2014-02-14 实施

国家食品药品监督管理总局 发布

目 次

前 言.....	IV
1 范围.....	1
2 规范性引用文件.....	1
3 术语、定义和缩略语.....	1
3.1 术语和定义.....	1
3.2 缩略语.....	2
4 设计原则.....	2
4.1 设计规范化原则.....	2
4.2 数据完整性原则.....	3
4.3 数据安全性原则.....	3
4.4 分级设计原则.....	3
4.5 高内聚原则.....	3
4.6 低耦合原则.....	3
4.7 高效性原则.....	3
5 数据库命名.....	3
5.1 系统标识.....	3
5.2 基本命名规范.....	4
5.2.1 语言.....	4
5.2.2 大小写.....	4
5.2.3 保留字.....	4
5.2.4 命名长度.....	4
5.2.5 字段名称.....	4
5.3 数据库服务名称.....	4
5.4 数据库对象命名规范.....	4
5.4.1 表.....	4
5.4.2 视图.....	4
5.4.3 存储过程.....	4
5.4.4 函数.....	5
5.4.5 触发器.....	5

5.4.6 自定义数据类型	5
5.4.7 主键	5
5.4.8 外键	5
5.4.9 索引	5
6 数据库结构设计	5
6.1 需求分析	5
6.1.1 基本要求	5
6.1.2 需求分析主要任务	5
6.1.3 需求收集内容	6
6.1.4 需求分析表达	6
6.2 概念设计	6
6.3 逻辑设计	6
6.3.1 数据项	6
6.3.2 主键约束	6
6.3.3 外键关联	7
6.3.4 NULL 值	7
6.3.5 触发器	7
6.3.6 注释	7
6.4 物理设计	7
7 数据库操作规范	7
7.1 表	7
7.2 视图	7
7.3 触发器	8
7.4 存储过程	8
7.5 索引	8
8 数据类型选择	8
8.1 字符型	8
8.2 数字型	9
8.3 日期和时间	9
8.4 大字段	9
8.5 唯一键	9
9 数据操作要求	9
9.1 查询数据	9

9.2 新增数据.....	9
9.3 更新数据.....	10
9.4 删除数据.....	10



前 言

本标准依据 GB/T 1.1—2009 给出的规则起草。

本标准由国家食品药品监督管理总局信息中心提出。

本标准由国家食品药品监督管理总局科技和标准司归口。

本标准起草单位：国家食品药品监督管理总局信息中心、中科软科技股份有限公司、广东省食品药品监督管理局。

本标准主要起草人：陈锋、张原、陆颖、刘靓、赵坤、李宗波、张翔、刘吕昕、史先东、李建魁。



食品药品监管数据库设计规范

1 范围

本标准规定了食品药品监管信息化工程各应用系统数据库的设计原则、数据库命名、数据库结构设计、数据库操作规范、数据类型选择、数据操作要求等方面内容，明确了应用系统数据库设计的要求。

本标准适用于规范数据库的建设，并对食品药品监管信息化工程各应用系统的数据库建设起指导作用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 5271.4—2000 信息技术 词汇 第4部分 数据的组织

GB/T 8567—2006 计算机软件文档编制规范

GB/T 11457—2006 信息技术 软件工程术语

GB/T 17532—2005 术语工作计算机应用词汇

CFDAB/T 0102.1—2014 食品药品监管信息化基础术语 第1部分：信息技术

CFDAB/T 0301—2014 食品药品监管信息基础数据元

3 术语、定义和缩略语

3.1 术语和定义

CFDAB/T 0102.1—2014 界定的以及下列术语和定义适用于本文件。

3.1.1

数据库 database

为多个独立用户接受存储和提供所需数据的一种数据结构。

[GB/T 11457—2006，定义 2.390]

3.1.2

数据元 data element

一种数据单位，在某个上下文中视为不可分。例：“人的年龄”的数据元，其值由三个十进制数字的所有组合组成。

[GB/T 5271.4—2000，定义 04.07.01]

3.1.3

表 list

数据元的有序集合。

[GB/T 5271.4—2000, 定义 04.08.01]

3.1.4

视图 view

某一些模型的一种如下投影：从一定的透视或有利位置观看，且略去与此透视无关的实体。

[GB/T 11457—2006, 定义 2.1833]

3.1.5

索引 index

（自然语言）字符串以及标示着其在文本或文本语料库中所处位置的标引与键值的列表。

[GB/T 17532—2005, 定义 5.5]

3.1.6

实体关系图 entity-relationship diagram; E-R

规定实际的实体集和它们之间的逻辑联系的图。

[GB/T 11457—2006, 定义 2.553]

3.1.7

数据库管理系统 database management system

是一整套计算机程序，它提供为建立、修改、使用和完整性维护一个数据库所需的功能。

[GB/T 8567—2006, 定义 3.15]

3.2 缩略语

DBMS: 数据库管理系统 (Database Management System)

E-R: 实体关系图 (Entity-Relationship Diagram)

4 设计原则

4.1 设计规范化原则

数据库的设计应采用 E-R 模型设计思想。

数据库设计应简洁且结构明晰，同时，不应发生插入 (insert)、删除 (delete) 和更新 (update) 操作的异常。

数据库设计应尽量保证数据的一致性，减少数据冗余。对于有较高性能及其他特殊要求的情况，可

以考虑以“空间换时间”的设计方式。一个数据库表中不包含在其他表中已包含的非主关键字信息。

4.2 数据完整性原则

数据库设计应考虑数据完整性，参照完整性约束，加强数据库的关系结构，避免数据库中存在不符合语义规定的数据库，防止因错误信息的输入输出造成无效操作或信息错误，保障数据库中的信息不被篡改。

4.3 数据安全性原则

数据库设计时应充分考虑数据安全性，根据用户的权限对数据进行过滤，做到数据权限的严格控制，防止由于非法用户使用数据库或合法用户非法使用数据库造成的数据泄露、更改或破坏。

4.4 分级设计原则

数据库分级设计遵循以下原则：

- a) 需求分析阶段：综合并分析用户的应用需求。
- b) 概念设计阶段：形成独立于机器特点，独立于各 DBMS 产品的 E-R 图。
- c) 逻辑设计阶段：首先将 E-R 图转换成数据库产品支持的数据模型，形成数据库逻辑模式，然后根据用户处理的要求、安全性的考虑，在基本表的基础上建立必要的视图。
- d) 物理设计阶段：根据 DBMS 特点，进行物理存储安排，建立索引，形成统一、稳定、可扩展及安全的数据库方案。

4.5 高内聚原则

将数据库中使用的基础数据元 and 值域代码等公共内容建立基础数据库，以数据库授权等形式访问基础数据库，避免重复设计和存储。对基础数据进行统一管理，保证数据的一致性。

4.6 低耦合原则

除基础数据库以外的应用系统数据库的设计，应避免直接调用其他应用系统的数据库，进行独立建库和维护，降低因为其他应用系统数据库变更而引起的威胁。

4.7 高效性原则

数据库设计时，要注意提高数据库的效率。

5 数据库命名

5.1 系统标识

食品药品监管信息化工程的应用系统应对所建数据库进行系统标识。

5.2 基本命名规范

5.2.1 语言

语言规则包括：

- 命名宜使用拼音的首字母大写缩写，不宜使用中文或者特殊字符。
- 当出现对象名重名且不同类型对象时，宜增加类型前缀或后缀以示区别。

5.2.2 大小写

名称宜大写，方便不同数据库移植，避免程序调用问题。

5.2.3 保留字

命名不宜使用 SQL 保留字。

5.2.4 命名长度

表名、字段名、视图名长度应限制在 20 个字符内（含前缀）。

5.2.5 字段名称

相同的字段在不同的数据表中应该采用同样的名称和字段类型定义。

5.3 数据库服务名称

数据库服务名称是唯一标识数据库的符号，命名长度不应超过 5 个字符。对于单节点数据库，建议以字母开头的 5 个长度以内字符串作为数据库的名称。对于集群数据库，当定义数据库名称后，各节点服务名称命名增加递增的序号，例如节点的序号：1、2、3、...，则节点服务名称为：abc1、abc2、abc3、...。

5.4 数据库对象命名规范

5.4.1 表

5.4.1.1 实体（表）

实体（表）名称前缀为系统标识，<系统标识>_<表标识>，表标识建议为拼音的首字母大写缩写。

5.4.1.2 属性（字段）

属性（字段）名称应用字母开头，建议为拼音的首字母大写缩写，不宜用双引号包含。

5.4.2 视图

视图名称前缀为 V_，V_[<系统标识>]_<视图标识>，视图标识建议为拼音的首字母大写缩写。

5.4.3 存储过程

存储过程名称前缀为 P_，P_[<系统标识>]_<存储过程标识>，存储过程标识建议为拼音的首字母大

写缩写。

5.4.4 函数

函数名称前缀为 F_，F_[<系统标识>]_<函数标识>，函数标识建议为拼音的首字母大写缩写。

5.4.5 触发器

触发器名称通过前缀来区分数据操作之前（before）和数据操作之后（after）的操作，数据操作之前名称 TB_<表标识>_<I,U,D 的任意组合>。数据操作之后名称为 TR_<表标识>_<I,U,D 的任意组合>。

5.4.6 自定义数据类型

数据自定义类型名称为 UD_<自定义数据类型标识>_<数据类型>，自定义数据类型标识建议为能够描述该数据类型的大写英文单词，数据类型为数据库数据类型的名称大写。

5.4.7 主键

主键名称前缀为 PK_，PK_<表标识>_<主键标识>，主键标识建议为 UUID、ID 或拼音的首字母大写缩写。

5.4.8 外键

外键名称前缀为 FK_，FK_<表标识>_<主表标识>_<外键标识>，外键标识建议为 UUID、ID 或拼音的首字母大写缩写。

5.4.9 索引

索引名称前缀为 IDX_，IDX_<表标识>_<构成的字段名>。如果复合索引的构成字段较多，则只包含第一个字段，并添加序号。

6 数据库结构设计

6.1 需求分析

6.1.1 基本要求

设计性能良好的数据库系统，应该把对用户需求的收集和分析作为数据库设计的第一步，明确应用环境对系统的要求。

6.1.2 需求分析主要任务

需求分析的主要任务是通过详细调查要处理的对象，包括某个组织、某个部门的业务管理流程等，充分了解原手工或原计算机系统的工作概况及工作流程，明确用户的各种需求，产生数据流图，然后在此基础上确定新系统的功能，并产生《软件需求规格说明》。新系统应充分考虑今后可能的扩充和改变，

不宜仅仅按当前应用需求来设计数据库，应留有一定的扩展空间。

需求分析的重点是调查、收集和分析用户数据管理中的信息需求、处理需求、安全性与完整性要求。信息需求是指用户需要从数据库中获得的信息的内容和性质，由用户的信息需求可以导出数据需求，即在数据库中应该存储的数据。处理需求是指用户需要的处理功能，对处理要求的响应时间。安全性与完整性要求在需求分析的非功能要求中提出，描述数据库设计的数据安全与数据完整性要求。

6.1.3 需求收集内容

需求收集可采用以下方式：

- a) 了解组织机构的情况，调查这个组织的部门结构，与各部门的职责，为分析信息流程做准备。
- b) 了解各部门的业务活动情况，调查各部门输入和使用的数据，加工处理这些数据的方式。输出的信息，输出信息流向的部门，输出的格式等。在调查活动的同时，要注意对各种资料的收集，如票证、单据、报表、档案、计划等，要特别注意了解这些报表之间的关系，各数据项的含义等。
- c) 确定新系统的边界。确定由计算机完成或将来准备让计算机完成的功能，以及由人工完成的活动。由计算机完成的功能就是新系统应该实现的功能。

6.1.4 需求分析表达

需求分析中数据流向通过数据流图方式表达，数据流图表达了数据处理过程的关系，数据流图通过特定的符号来描述数据的流向，可通过常用的绘图工具绘制数据流图。

6.2 概念设计

概念结构设计的任务是在需求分析阶段产生的《软件需求规格说明》的基础上，按照特定的方法把业务需求抽象为不依赖于任何具体机器的数据模型。概念模型具有以下的要求：

- a) 概念模型是对现实世界的抽象和概括，应表达出对数据的处理要求。
- b) 概念模型应易于更新，当应用环境和应用要求改变时，容易对概念模型进行修改和扩充。

建议使用 E-R 图进行概念设计。E-R 模型是面向问题的概念模型，即用简单的图形方式（E-R 图）描述现实世界中的数据。这种描述不涉及数据在数据库中表示和存取方法。

6.3 逻辑设计

6.3.1 数据项

业务数据项设计内容应从“CFDAB/T 0301—2014”标准中获取相应的数据元。

其他数据项的命名应使用中文名称汉语拼音首字母大写。

6.3.2 主键约束

关联表的父表应有主键，主键字段或组合字段应满足非空属性和唯一性要求。对于数据量比较大的

父表，应指定索引段。

6.3.3 外键关联

对于关联两个表的字段，应该分别建立主键、外键。为了提高性能，数据量比较大的表应对外键建立索引。

6.3.4 NULL 值

对于字段能否 NULL，应该在 SQL 建表脚本中指明，不应为缺省值。

6.3.5 触发器

在选择触发器的 **before** 或 **after** 事务属性时，表操作的事务属性应与应用程序事务属性保持一致，以避免死锁发生。在大型数据表中，尽量避免使用触发器。

6.3.6 注释

表、字段等建议使用中文进行注释。

6.4 物理设计

物理设计将一个给定的逻辑结构实施到具体的工作环境中时，该环境提供了数据的存储结构与存取方法。物理设计应遵循以下要求：

- a) 应确定数据的存储结构，设计数据的存取路径。
- b) 应考虑数据库集群的要求，建议数据文件存取到独立的存储中。
- c) 应考虑数据库多国语言存储支持，建议采用 UTF8 字符集。

7 数据库操作规范

7.1 表

数据库表的基本要求如下：

- a) 避免使用业务主属性作为表的主键。
- b) 多表连接时，使用表的别名来引用列。

示例：

```
SELECT t1.abc002,t2.abd003
FROM ab001 t1,ab020 t2
WHERE t1.col2=t2.col3
```

.....

7.2 视图

数据库视图的基本要求如下：

- a) 使用静态视图，不宜使用动态视图。
- b) 视图中尽量避免使用 `Order By` 语句，`Order By` 语句会影响视图的效率。
- c) 视图中尽量避免使用 `union` 或 `union all` 语法。
- d) 视图编写宜使用大写英文字母。
- e) 经常使用的字段应放到视图中字段靠前位置。

7.3 触发器

数据库触发器的基本要求如下：

- a) 编写过程中不宜使用自动缩排，也不宜使用 `TAB` 键缩进。
- b) 变量名称命名方式参照 7.4 存储过程。
- c) 游标（Cursor）以 ‘`Cur_`’ + 名称 命名。
- d) 所有变量应加注释。
- e) 功能段之间应加注释。

7.4 存储过程

编写存储过程时，函数名称不能使用数据库系统的保留字、内置函数、预定义类型以及 `SQL` 关键字，表 1 为保留字、内置函数、预定义类型、`SQL` 关键字的举例。

表 1 保留字、内置函数、预定义类型、`SQL` 关键字的举例

类型	约定	举例
保留字	大写	<code>BEGIN</code> 、 <code>DECLARE</code> 、 <code>ELSIF</code>
内置函数	大写	<code>SUBSTR</code> 、 <code>COUNT</code> 、 <code>TO_NUMBER</code>
预定义类型	大写	<code>NUMBER(7,2)</code> 、 <code>BOOLEAN</code>
<code>SQL</code> 关键字	大写	<code>SELECT</code> 、 <code>INTO</code> 、 <code>WHERE</code>

7.5 索引

在使用索引时，应注意以下几点：

- a) 逻辑主键应使用唯一的成组索引，外键采用非成组索引。
- b) 经常使用的键应创建索引，例如：主键、外键、常用作查询的键。
- c) 不宜索引大型字段（有很多字符的字段）。
- d) 不宜索引常用的小型表。

8 数据类型选择

8.1 字符型

固定长度的字符串类型采用 `char`，长度不固定的字符串类型采用 `varchar`。

8.2 数字型

在存储数字型数据时，应该充分考虑数据的长度，选择合适的类型进行存储，同时为数据的扩展保留一定的空间。

8.3 日期和时间

日期和时间类型选择时，参考以下要求：

- a) 系统时间：由数据库产生的系统时间首选数据库的日期型。
- b) 外部时间：由数据导入或外部应用程序产生的日期时间采用 `varchar` 类型，数据格式采用：
YYYYMMDDhhmmss。

8.4 大字段

如无特别需要，避免使用大字段（`blob`，`clob`，`long`，`text`，`image` 等）。

8.5 唯一键

对于数字型唯一键值，建议使用序列产生。

9 数据操作要求

9.1 查询数据

查询操作（Query）允许用户从数据库中获取指定的数据，查询操作中的 `Operate` 应指定操作的属性 ID，以便于在响应操作中进行关联，其中：

- a) 查询操作应指定 `From`：查询的数据表或数据集。
- b) 查询操作应指定 `Return`：返回的数据结构以及返回的数据最大个数。

其中最大个数通过指定属性 `maxCount` 值来标识，数据结构通过 `Structure` 来说明，数据结构可以自定义，或者引用一个定义的 `Schema`。

- c) 查询操作可以指定 `Conditions`：获取的条件，条件可以多个，如果不指定则选取全部数据。
 - 第一个条件应指定属性 `isFirst` 为“True”，`Relate` 应为空；
 - 非第一个条件应指定属性 `isFirst` 为“False”，`Relate` 应为“AND”或者“OR”；
 - 条件应指定 `Op` 操作类型（“EQUAL”等于、“GREAT”大于、“LESS”小于、“GREAT_EQUAL”大于等于、“LESS_EQUAL”小于等于）；
 - 条件应指定数据的名称 `DataName`；
 - 条件应指定数据的值 `DataValue`，数据的值来源于数据集或者固定值，如果来源于数据集，应用 `XPath` 来表达。

9.2 新增数据

新增操作（insert）允许用户向数据库存入指定的数据，新增操作中的 `Operate` 应指定操作的属

性 ID，以便于在响应操作中进行关联，其中：

- a) 新增操作应指定 **To**：存入的对象。
- b) 新增操作应指定 **Data**：新增的数据。

9.3 更新数据

更新数据操作（**Update**）允许用户更新数据库指定的数据，更新操作中的 **Operate** 应指定操作的属性 ID，以便于在响应操作中进行关联，其中：

- a) 更新操作应指定 **To**：存入的对象。
- b) 更新操作应指定 **Data**：更新的数据。
- c) 更新操作可以指定 **Conditions**：获取的条件，条件可以多个。
 - 第一个条件应指定属性 **isFirst** 为“False”，**Relate** 应为“AND”或者“OR”；
 - 条件应指定 **Op** 操作类型（“EQUAL”等于、“GREAT”大于、“LESS”小于、“GREAT_EQUAL”大于等于、“LESS_EQUAL”小于等于）；
 - 条件应指定数据的名称 **DataName**；
 - 条件应指定数据的值 **DataValue**，数据的值来源于数据集或者固定值，如果来源于数据集，应用 **XPath** 来进行表达。

9.4 删除数据

删除操作（**Delete**）允许用户删除数据库指定的数据，删除操作中的 **Operate** 应指定操作的属性 ID，以便于在响应操作中进行关联，其中：

- a) 删除操作应指定 **To**：存入的对象。
- b) 删除操作应指定 **Data**：删除的数据。
- c) 删除操作可以指定 **Conditions**：获取的条件，条件可以多个：
 - 第一个条件应指定属性 **isFirst** 为“False”，**Relate** 应为“AND”或者“OR”；
 - 条件应指定 **Op** 操作类型（“EQUAL”等于、“GREAT”大于、“LESS”小于、“GREAT_EQUAL”大于等于、“LESS_EQUAL”小于等于）；
 - 条件应指定数据的名称 **DataName**，根据具体的情况定义；
 - 条件应指定数据的值 **DataValue**，数据的值来源于数据集或者固定值，如果来源于数据集，应用 **XPath** 来进行表达。