恒大智慧小区项目

**G1项目组**

**数据库设计规范**

目 录

1. 文档说明 4

1.1 版本修订记录 4

1.2 相关读者 4

1.3 术语定义 4

2. 设计规范 5

2.1 基本规范 5

2.1.1 命名规范 5

2.1.2 对象命名一般规则 5

2.1.3 对象类型命名 6

2.1.4 字符集 6

2.1.5 注释规范 6

2.1.6 生产账号权限 7

2.1.7 数据字典 7

2.2 表设计规范 7

2.2.1 必有字段 7

2.2.2 考虑过期策略 7

2.2.3 数据的删除 8

2.2.4 关联表数据的变更 8

2.2.5 业务约束 8

2.3 列设计规范 8

2.3.1 字段类型 8

2.3.2 规范字段长度 8

2.3.3 编码字段规范 9

2.3.4 常见字段编码 9

2.3.5 冗余字段 9

2.4 索引规范 9

2.4.1 主键设计 9

2.4.2 索引规约 9

2.5 公共数据字典 10

2.6 版本机制 11

附录1：业务命名 13

附录2：常见字段名称 14

附录3：保留字与关键字 15

# 文档说明

## 版本修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 文档版本 | 修订描述 | 文档作者 | 审批人 |
| 2017/11/09 | 1.0 | 初始版本 | 程立强 |  |
| 2017/12/06 | 1.1 | 增加部分内容 | 陆润 |  |
| 2018/1/10 | 1.2 | 整理部分内容 | 马军辉 |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 相关读者

|  |  |
| --- | --- |
| 读者分类 | 目的 |
| 架构人员 |  |
| 设计人员 |  |
| 开发人员 |  |
| 测试人员 |  |

## 术语定义

|  |  |
| --- | --- |
| 术语 | 描述 |
|  |  |
|  |  |
|  |  |
|  |  |

# 设计规范

## 基本规范

### 命名规范

**库命名**

采用应用英文缩写且数据库名不能超过8个字符。

**长度**

对于所有的数据库对象标识符名称不能超过30个字符，即数据库的表名、字段名、函数名、过程名、触发器名、序列名、视图名等长度都不能超过30个字符。

**大小写**

数据库中所有对象名称(表名,字段名,过程名,视图名等)全部统一使用小写，禁止对任何名称添加双引号 "" 来设定字符的大小写形式。

**业务命名**

相同的业务应该命名一致，这部分内容维护在附录中。

### 对象命名一般规则

* 只能由小写字母，数字和下划线组成，并且不能以数字开头
* 统一采用英文单词，禁止使用中文拼音
* 使用下划线“\_”分割名称中各个业务单词
* 禁止使用数据库的关键字或者保留字命名

### 对象类型命名

规则如下：

|  |  |
| --- | --- |
| **对象类型** | **命名方法** |
| 表 | 业务名称\_扩展  (对于表名的命名，字符尽量短的原则，避免以后索引的命名，约束的命名容易超30个字符的范围) |
| 视图 | v\_ + 业务名称 |
| 物化视图 | mv\_ + 业务名称 |
| 序列 | seq\_ + 表名 |
| 存储过程 | p\_ + 业务名称 |
| 函数 | f\_ + 业务名称 |
| 触发器 | trig\_ + 业务名称 |
| 主键约束 | pk\_ + 表名 |
| 外键约束 | fk\_ ＋ 外键表名(可简写) + 主键表名(可省略) + 关联字段(可省略) |
| 普通索引 | idx\_ + 表名(可简写) + 索引字段(可省略) |
| 位图索引 | bidx\_ +表名(可简写) + 索引字段(可省略) |
| 唯一索引 | uidx\_ + 表名(可简写) + 索引字段(可省略) |
| 函数索引 | fidx\_ +表名(可简写) + 索引字段(可省略) |
| 备份表 | bak\_+表名 |
| 临时表 | tmp\_ + 业务名称 |
| JOB名称 | job\_ + 业务名称 + 业务功能 |

### 字符集

字符集采用UTF-8。

### 注释规范

**表注释**

必须有注释，简要说明表的用途。

**列注释**

每个表及字段都需要有注释，说明其表示的具体含义,有枚举值需要全部列出，列出的值使用分号隔离，不要使用换行符；对于冗余字段还要特别说明其维护方法。

### 生产账号权限

最少授权，只授予最基础权限的需求，遵循够用就好的原则。

### 数据字典

数据字典表结构应该统一维护，且字典编码已有国家标准的沿用国标编码。

## 表设计规范

对于业务数据表，需要考虑一定的设计原则。

### 必有字段

业务数据表必须要有以下字段，以用于数据的上传。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段注释 | 是否为空 |
| uuid | varchar(32) | 表记录UUID | 不为空 |
| court\_uuid | varchar(32) | 小区UUID | 不为空 |
| create\_time | timestamp | 创建时间 | 不为空 |
| update\_time | timestamp | 更新时间 | 不为空 |
| create\_user | varchar(32) | 创建人 | 不为空 |
| update\_user | varchar(32) | 修订人 | 不为空 |

注：以上的时间字段，采用数据库时间。创建人及修订人为人员UUID，非实际的人员名称。

### 考虑过期策略

需要明确数据生命周期的管理，必要时，需要考虑分区或者是分表，需要给出配置过期数据策略的方法。

### 数据的删除

各业务表数据应该明确数据的删除方法，物理删除还是逻辑删除。

### 关联表数据的变更

应该特别注意有主外键关系的数据删除与更新，必要时请增加”cascade”关键字。

### 业务约束

* 业务及组件不允许对数据库进行DDL操作。
* 业务及组件不允许进行公共字典的变更。

## 列设计规范

### 字段类型

* 应该选用合适的数据类型，避免数据库资源的浪费。一般情况下，数值的类型比字符的类型高效。
* 不同表相同的业务字段采用相同的字段类型，这样在查询时可以避免隐式转换。

### 规范字段长度

无特殊情况，数据类型建议选用以下的类型。

|  |  |
| --- | --- |
| 数值型 | smallint、integer、bigint、number(p,s) |
| 字符型 | varchar(16) 、varchar(32) 、varchar(64) 、varchar(256) 、text |
| 日期型 | date、timestamp |
| 布尔型 | boolean |

### 编码字段规范

编码ID、CODE、UUID统一规范如下：

|  |  |
| --- | --- |
| 编码名称 | 用途 |
| id | 无业务逻辑的编码 |
| code | 特有规则，有业务逻辑且需要关联的编码 |
| uuid | 程序生成，需要保证全局唯一且需要关联的编码 |

### 常见字段编码

常见字段编码与字段类型如附录2

### 冗余字段

由于大表增加字段会消耗大量的数据库资源和锁表，设计时需要适当考虑冗余字段。字段名称使用attr\_1、attr\_2 、attr\_3…。字段类型采用varchar，长度自行决定。

## 索引规范

### 主键设计

* 业务表都应该建立主键

### 索引规约

* 原则上来讲，一张表的索引不允许超过4个。当表的数据量很大时,将会极大的影响该表的更新和插入,因此在表中创建索引时需要谨慎考虑。只创建业务真正需要的索引。
* 原则上来讲，复合索引字段应该不超过3个，字段太多会造成数据库维护复合索引时消耗的资源过多。
* 基数(Cardinality)很低的字段不建议创建索引。
* 业务上具有唯一特性的字段，即使是多个字段的组合，也必须建成唯一索引。
* 页面搜索严禁左模糊或者全模糊，如果需要请走搜索引擎来解决。
* 外键需要创建索引。
* 在OLTP场景中，尽量使用B-tree索引，禁止使用位图索引，位图索引常见于OLAP中（小区平台为OLTP场景）。

## 公共数据字典

由主数据提供公共字典接口，表数据由各业务主键进行提交。

表结构如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **公共字典主表：dict\_code\_type** | | | | |
| 字段名称 | 字段类型 | 说明 | 是否可空 | 备注 |
| type\_code | varchar(32) | 类型编码 | 否 | 公共字典大类编码 |
| type\_name | varchar(64) | 类型名称 | 否 |  |
| enable\_flag | smallint | 是否启用:0-禁用，1-启用 | 否 | 默认启用 |
| create\_time | timestamp | 创建时间 | 否 |  |
| update\_time | timestamp | 更新时间 | 否 |  |
| create\_user | varchar(32) | 创建人 | 否 |  |
| update\_user | varchar(32) | 修订人 | 否 |  |
| memo | varchar(256) | 描述 | 是 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **公共字典子表：dict\_code** | | | | |
| 字段名称 | 字段类型 | 说明 | 是否可空 | 备注 |
| item\_code | varchar(32) | 项目编码 | 否 | 1. 同一个大类下的CODE应该唯一，CODE内容不做限制，可以是数字、字符等 2. 与PARENT\_CODE构成唯一键 |
| item\_name | varchar(64) | 项目名称 | 否 |  |
| parent\_code | varchar(32) | 父编码 | 是 | 关联本表ITEM\_CODE，需与ITEM\_TYPE配合使用 |
| item\_type | varchar(32) | 类型编码,关联数据字典主表 | 否 | 关联主表TYPE\_CODE |
| sort | smallint | 排序码 | 是 |  |
| enable\_flag | smallint | 是否启用:0-禁用，1-启用 | 否 | 默认启用 |
| create\_time | timestamp | 创建时间 | 否 |  |
| update\_time | timestamp | 更新时间 | 否 |  |
| create\_user | varchar(32) | 创建人 | 否 |  |
| update\_user | varchar(32) | 修订人 | 否 |  |
| memo | varchar(256) | 描述 | 是 |  |

公共数据字典表结构应非常稳定，因为任何修改都会影响到所有业务组件的正常使用。公共数据字典的数据应该严格把控，维护进去的数据不应该再删除。

业务平台不允许对公共字典CODE进行变更，以及对公共字典进行添加条目。

## 版本机制

建议在数据库中引入版本控制机制来确定使用中的数据库的版本。时间一长，用户的需求总是会改变的。最终可能会要求修改数据库结构,虽然可以通过检查新字段或者索引来确定数据库结构的版本，但把版本信息直接存放到数据库中会更方便。

表结构如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **组件版本表：module\_version** | | | | |
| 字段名称 | 字段类型 | 说明 | 是否可空 | 备注 |
| id | smallint | ID | 否 |  |
| module\_name | varchar(32) | 组件名称 | 否 |  |
| module\_version | varchar(32) | 组件版本号 | 否 |  |
| update\_date | timestamp | 更新时间 | 否 |  |

# 附录1：业务命名

注：英文太长的尽量使用缩写

|  |  |  |  |
| --- | --- | --- | --- |
| 名词 | 英文名 | 名词 | 英文名 |
| 房屋 | house |  |  |
| 图片 | pic |  |  |
| 用户 | user |  |  |
| 证件 | iden |  |  |
| 组织 | org |  |  |
| 父级 | parent |  |  |
| 小区 | court |  |  |

# 附录2：常见字段名称

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字段编码 | 字段类型 | 注释 |
| 删除标记 | delete\_flag | smallint | 逻辑删除标记： 0-已删除 1-未删除（默认） |
| 用户类型编码 | user\_type | varchar | 用户类型（**不需要再额外增加code**） |
| 是否禁用 | enable\_flag | smallint | 是否启用:0-禁用，1-启用（默认） |
| 排序码 | sort | smallint | 用于页面的排序 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

注：如不指定长度，那么长度请自行判断。

# 附录3：保留字与关键字

以下列举的单词不应该使用在数据库的对象命名中、

|  |  |  |
| --- | --- | --- |
| ALL | DISTINCT | ONLY |
| ANALYSE | DO | OR |
| ANALYZE | ELSE | ORDER |
| AND | END | PLACING |
| ANY | EXCEPT | PRIMARY |
| ARRAY | FALSE | REFERENCES |
| AS | FETCH | RETURNING |
| ASC | FOR | SELECT |
| ASYMMETRIC | FOREIGN | SESSION\_USER |
| BOTH | FROM | SOME |
| CASE | GRANT | SYMMETRIC |
| CAST | GROUP | TABLE |
| CHECK | HAVING | THEN |
| COLLATE | IN | TO |
| COLUMN | INITIALLY | TRAILING |
| CONSTRAINT | INTERSECT | TRUE |
| CREATE | INTO | UNION |
| CURRENT\_CATALOG | LATERAL | UNIQUE |
| CURRENT\_DATE | LEADING | USER |
| CURRENT\_ROLE | LIMIT | USING |
| CURRENT\_TIME | LOCALTIME | VARIADIC |
| CURRENT\_TIMESTAMP | LOCALTIMESTAMP | WHEN |
| CURRENT\_USER | NOT | WHERE |
| DEFAULT | NULL | WINDOW |
| DEFERRABLE | OFFSET | WITH |
| DESC | ON |  |