

# 运营支持部数据库应用规范

2010-04-26

Taomee 淘米

## 目 录

运营支持部数据库应用规范.....	1
一、关于本文档.....	1
1.1 修改历史.....	1
1.2 参考文档.....	1
1.3 编写目的.....	1
二、规范内容.....	1
2.1 数据库相关命名.....	1
2.2 关于类型选择.....	1
2.3 数据库账户及其授权.....	2
2.4 设计原则.....	2
2.5、SQL 语句编码.....	3
2.6 关于数据库设计文档说明.....	3

Taomee 淘米

## 一、关于本文档

### 1.1 修改历史

版本	作者	修改日期	修改概要	备注
V1.0.0	Aceway	2010-04-27		创建初版

### 1.2 参考文档

- 1、《MySQL5 权威指南》。
- 2、《性能调优与架构设计》。

### 1.3 编写目的

规范部门项目的数据库设计，为项目开发，维护提供统一的参考。

目的在于：

强化、统一数据库设计习惯，给新同事明确的要求，从数据库设计方面加强团队建设。

## 二、规范内容

这里的数据库指关系型数据库，重点是针对 MySQL。本文档内容用于规范数据库的设计，包括数据相关命名规则，数据类型选择，同时对运营支持相关系统里的数据库账户，权限设置做一些指导。

### 2.1 数据库相关命名

- 1、库、表的命名仅使用小写字母、下划线、数字。项目中已经存在的的大写字母命名的数据库除外；严禁大、小写字母混合使用。
- 2、字段命名建议使用字母、下划线，限制使用数字；
- 3、**新项目**中的库名称加“db\_”前缀；表名称建议加“t\_”前缀，名称的后缀部份留给数字使用；**已有项目**的数据库建议遵循其已有的命名规则。
- 4、库、表、字段命名尽量反映其业务含义，名称长度建议不超过 32 个字符；禁止使用复数名词作为库、表、字段的名称，统一使用单数形式的名词。
- 5、数据库账户命名建议和项目业务结合。

## 2.2 关于类型选择

- 1、用尽量少的存储空间来存数一个字段的数：  
能用 int 的就不用 char 或者 varchar；能用 tinyint 的就不用 int；能用 varchar(20)的就不用 varchar(255)；时间戳字段尽量用 int 型；
- 2、字符类型一定要统一。在操作数据库的链接里一定明确指定字符类型，建议统一使用 UTF8；数据库内部存储的字符集类型建议使用 UTF8；
- 3、**要求事务安全的系统其数据库选择 InnoDB 为表引擎；**

## 2.3 数据库账户及其授权

- 1、根据项目建立项目账户，项目中使用的数据库账户严格禁止使用 root；
- 2、分配账户时根据需要限制账户是从 localhost 还是从指定 IP 访问；
- 3、根据项目对数据库的访问需要定制其数据库账户的权限，遵循**权限够用同时最小**的原则。严格禁止为了方便授权所有库、所有表的 UPDATE，INSERT，DELETE 等操作，特殊的系统应根据需要按字段授权给账户；
- 4、对于一般项目不需要的修改库、表等结构的权限，请慎重考虑授权给项目的数据库账户；

## 2.4 设计原则

- 1、**尽量遵循三范式：**

**第一范式：**数据库表的每一列都是不可分割的基本数据项，同一列中不能有多值；

如下表是符合第一范式：

字段 1	字段 2	字段 3	字段 4
?	?	?	?

如下表是不符合第一范式的：

字段 1	字段 2	字段 3		字段 4
?	?	字段 3.1	字段 3.2	?

不满足第一范式的数据库不是关系型的数据库；

**第二范式：**属性完全依赖于主键；

假设存在下列学生选课表：

**(学号，学生姓名，年龄，性别，课程，课程学分，学科成绩，系别，系办地址，系办电话)：**

表中“学号”为主键，那么其中，“课程”，“课程学分”，“学科成绩”不完全依赖于“学号”，即确定了“学号”并不能完全确定该条纪录的“课程”，“课程需分”，“学科成绩”内容，即一个学生可以选不止一门课程；对于一个学生只许从属于一个系别，表中“系别”，“系办地址”，“系办电话”完全依赖于“学号”主键，即一旦确定了表中一个“学号”，那么就能确定其“系别”，“系办地址”，“系办电话”内容。因此根据第二范式重新设计成三个表：

**学生表：(学号，学生姓名，年龄，性别，系别，系办地址，系办电话)**此表“学号”为主键；

**课程表：(课程，课程学分)**此表“课程”为主键；

**选课表：(学号，课程，学科成绩)**此表“学号”和“课程”一起作为联合主键；

**第三范式：**属性不依赖于其它非主属性；

如上例重新设计的“学生表”中的“系办地址”，“系办电话”依赖于主键“学号”的同时依赖于“系别”，它们必须与“系别”保持一致，仍然需要一个系别信息表纪录每个系别（dept\_id）、系别名称、系别地址，系办电话等信息。根据第三范式原学生选课表重新设计成四个表：

**学生表：(学号，学生姓名，年龄，性别，系别)**此表“学号”为主键；

**系别表：(系别，系办地址，系办电话)**此表“系别”为主键；

**课程表：(课程，课程学分)**此表“课程”为主键；

**选课表：(学号，课程，学科成绩)**此表“学号”和“课程”一起作为联合主键；

实际设计中需要对三范式权衡：它最大的不利是把信息放置在不同的表中，增加了操作的难度，同时把多个表连接在一起的花费也是巨大的。因为表和表的连接操作是做两个关系的笛卡儿积，必然会产生大量无用甚至无效的记录，性能的代价是巨大的。

规范化优点：避免了大量的数据冗余，节省了空间，保持了数据的一致性，如果完全达到 3NF，就不会在超过一个地方更改同一个值。

- 2、每张表必须设计主键，主键一定放在所属表的前排字段，联合主键的字段尽量相邻；
- 3、积极设计并使用索引，但不滥用索引，禁止给大型字段设置索引；
- 4、存储过程、视图、触发器的使用需经过三人以上的共同审核通过；
- 5、避免长事务；
- 6、对于一个数据库内设计的表数量建议控制在 300 个以内，否则设计分库存储；建议单表纪录数控制在 3 千万以内，对于性能有较高要求的建议控制在 1 千万以内，否则设计分表存储；建议单表的字段数量控制在 32 个左右；

## 2.5、SQL 语句编码

- 1、所有的 SQL 关键字大写；
- 2、如果 SQL 语句要分行，从 SQL 关键字处分；
- 3、不允许写 SELECT \* FROM ...，必须指明需要读取的具体字段；
- 4、对于 UPDATE，DELETE 语句必须带上 WHERE 条件，必要时加上 LIMIT 限制；
- 5、连接符 OR、IN、AND、=、<=、>=、!=、<>等前后加上一个空格；
- 6、大量的排序操作影响系统性能，所以尽量减少 ORDER BY 和 GROUP BY 排序操作，排序尽量建立在有索引的列上；
- 7、索引中，尽量避免使用 NULL，对于索引的比较，尽量避免使用 <> (!=)；
- 8、尽量避免相同语句由于书写格式的不同，而导致多次语法分析；
- 9、对较为复杂的 SQL 语句加上注释，说明算法、功能；
- 10、对数据量大的表不要将复杂的数学计算放入 SQL 语句，尽量只使用简单的加、减、乘、除；
- 11、用执 EXPLAIN 分析 SQL 性能，根据分析结果对语句进行优化；

## 2.6 关于数据库设计文档说明

- 1、在项目的数据库设计文档里，必须清楚描述数据库的表组成，每张表的字段组成。
- 2、数据库详细设计格式请使用统一模板《XXX 数据库设计文档 V1.0.0--模板.doc》。

Taomee 淘米