

建议收藏 | 专业的MySQL开发规范

python爬虫人工智能大数据 今天

文章转载自公众号  运维咖啡吧，作者 37°Y37

为了项目的稳定，代码的高效，管理的便捷，在开发团队内部会制定各种各样的规范

这里分享一份我们定义的MySQL开发规范，欢迎交流拍砖

数据库对象命名规范

数据库对象

命名规范的对象是指数据库SCHEMA、表TABLE、索引INDEX、约束CONSTRAINTS等的命名约定

数据库对象命名原则

1. 命名使用具有意义的英文词汇，词汇中间以下划线分隔
2. 命名只能使用英文字母、数字、下划线
3. 避免用MySQL的保留字如：call、group等
4. 所有数据库对象使用小写字母

数据库命名规范

1. 数据库名不能超过30个字符
2. 数据库命名必须为项目英文名称或有意义的简写

3. 数据库创建时必须添加默认字符集和校对规则子句。默认字符集为UTF8(已迁移dumbo的使用utf8mb4)
4. 命名应使用小写

表命名规范

1. 同一个模块的表尽可能使用相同的前缀，表名称尽可能表达含义
2. 多个单词以下划线 (_) 分隔
3. 表名不能超过30个字符
4. 普通表名以t_开头，表示为table，命名规则为t_模块名 (或有意义的简写) _+table_name
5. 临时表(运营、开发或数据库人员临时用作临时进行数据采集用的中间表)命名规则：加上tmp前缀和8位时间后缀 (tmp_test_user_20181109)
6. 备份表 (DBA备份用作保存历史数据的中间表) 命名规则：加上bak前缀和8位时间后缀 (bak_test_user_20181109)
7. 命名应使用小写

字段命名规范

1. 字段命名需要表示其实际含义的英文单词或简写，单词之间用下划线 (_) 进行连接
2. 各表之间相同意义的字段必须同名
3. 字段名不能超过30个字符

用户命名规范

1. 生产使用的用户命名格式为 code_应用
2. 只读用户命名规则为 read_应用

数据库对象设计规范

存储引擎的选择

如无特殊需求，必须使用innodb存储引擎

字符集的选择

如无特殊要求，必须使用utf8或utf8mb4

表设计规范

1. 不同应用间所对应的数据库表之间的关联应尽可能减少，不允许使用外键对表之间进行关联，确保组件对应的表之间的独立性，为系统或表结构的重构提供可能性
2. 表设计的角度不应该针对整个系统进行数据库设计，而应该根据系统架构中组件划分，针对每个组件所处理的业务进行数据库设计
3. 表必须要有PK
4. 一个字段只表示一个含义
5. 表不应该有重复列
6. 禁止使用复杂数据类型(数组,自定义等)
7. 需要join的字段(连接键),数据类型必须保持绝对一致，避免隐式转换
8. 设计应至少满足第三范式,尽量减少数据冗余。一些特殊场景允许反范式化设计，但在项目评审时需要对冗余字段的设计给出解释
9. TEXT字段必须放在独立的表中,用PK与主表关联。如无特殊需要，禁止使用TEXT、BLOB字段
10. 需要定期删除(或者转移)过期数据的表,通过分表解决
11. 单表字段数不要太多，建议最多不要大于50个

12. MySQL在处理大表时，性能就开始明显降低，所以建议单表物理大小限制在16GB，表中数据控制在2000W内
13. 如果数据量或数据增长在前期规划时就较大，那么在设计评审时就应加入分表策略
14. 无特殊需求，严禁使用分区表

字段设计规范

INT：如无特殊需要，存放整型数字使用UNSIGNED INT型。整型字段后的数字代表显示长度

DATETIME：所有需要精确到时间(时分秒)的字段均使用DATETIME, 不要使用TIMESTAMP类型

VARCHAR：所有动态长度字符串 全部使用VARCHAR类型,类似于状态等有限类别的字段,也使用可以比较明显表示出实际意义的字符串,而不应该使用INT之类的数字来代替；VARCHAR(N)，N表示的是字符数而不是字节数。比如VARCHAR(255)，可以最大可存储255个字符（字符包括英文字母，汉字，特殊字符等）。但N应尽可能小，因为MySQL一个表中所有的VARCHAR字段最大长度是65535个字节，且存储字符个数由所选字符集决定。如UTF8存储一个字符最大要3个字节，那么varchar在存放占用3个字节长度的字符时不应超过21845个字符。同时，在进行排序和创建临时表一类的内存操作时，会使用N的长度申请内存。（如无特殊需要，原则上单个varchar型字段不允许超过255个字符）

TEXT：仅仅当字符数量可能超过20000个的时候,才可以使用TEXT类型来存放字符类数据,因为所有MySQL数据库都会使用UTF8字符集。所有使用TEXT类型的字段必须和原表进行分拆,与原表主键单独组成另外一个表进行存放。如无特殊需要，严禁开发人员使用MEDIUMTEXT、TEXT、LONGTEXT类型

对于精确浮点型数据存储，需要使用DECIMAL，严禁使用FLOAT和DOUBLE

如无特殊需要，严禁开发人员使用BLOB类型

如无特殊需要，字段建议使用NOT NULL属性，可用默认值代替NULL

自增字段类型必须是整型且必须为UNSIGNED，推荐类型为INT或BIGINT，并且自增字段必须是主键或者主键的一部分

索引设计规范

1. 索引必须创建在索引选择性较高的列上,选择性的计算方式为: `select count(distinct(col_name))/count(*) from tb_name;` 如果结果小于0.2,则不建议在此列上创建索引,否则大概率会拖慢SQL执行
2. 组合索引的首字段,必须在where条件中,对于确定需要组成组合索引的多个字段,建议将选择性高的字段靠前放
3. 禁止使用外键
4. Text类型字段如果需要创建索引,必须使用前缀索引
5. 单张表的索引数量理论上应控制在5个以内。经常有大批量插入、更新操作表,应尽量少建索引
6. ORDER BY, GROUP BY, DISTINCT的字段需要添加在索引的后面,形成覆盖索引
7. 尽量使用Btree索引,不要使用其它类型索引

约束设计规范

1. PK应该是有序并且无意义的,尽量由开发人员自定义,且尽可能短,使用自增序列
2. 表中除PK以外,还存在唯一性约束的,可以在数据库中创建以“uidx_”作为前缀的唯一约束索引
3. PK字段不允许更新
4. 禁止创建外键约束,外键约束由应用控制
5. 如无特殊需要,所有字段必须添加非空约束,即not null
6. 如无特殊需要,所有字段必须有默认值

SQL编写规范

1. 尽量避免使用 `select *`, join语句使用 `select *` 可能导致只需要访问索引即可完成的查询需要回表取数

2. 严禁使用 `select * from table` 而不加任何where条件
3. MySQL中的text类型字段存储的时候不是和由其他普通字段类型的字段组成的记录存放在一起,而且读取效率本身也不如普通字段块。如果不需要取回text字段,又使用了 `select *`,会让完成相同功能的sql所消耗的io量大很多,而且增加部分的io效率也更低下
4. 在取出字段上可以使用相关函数,但应尽可能避免出现 `now()`, `rand()`, `sysdate()`, `current_user()` 等不确定结果的函数,在Where条件中的过滤条件字段上严禁使用任何函数,包括数据类型转换函数
5. 所有连接的SQL必须使用 `Join ... On ...` 方式进行连接,而不允许直接通过普通的Where条件关联方式。外连接的SQL语句,可以使用 `Left Join On` 的Join方式,且所有外连接一律写成 `Left Join`,而不要使用 `Right Join`
6. 分页查询语句全部都需要带有排序条件,除非应用方明确要求不要使用任何排序来随机展示数据
7. WHERE条件中严禁在索引列上进行数学运算或函数运算
8. 用 `in()` / `union` 替换or,并注意in的个数小于300
9. 严禁使用%前缀进行模糊前缀查询:如: `select id,val from table where val like '%name'`; 可以使用%模糊后缀查询如: `select id,val from table where val like 'name%'`
10. 严禁使用 `INSERT ON DUPLICATE KEY UPDATE`、`REPLACE INTO`、`INSERT IGNORE`

精选好文：

重磅！开源一个机器学习/情感分析实战项目（附源码/教程）

分享Django+Linux+Uwsgi+Nginx项目部署文档

python自动化测试技术打造升职加薪利器

110道python面试题

我去面试python岗位了

一幅漫画看透零基础转Python学习路线

工作必备，耗时3天总结踩坑写的超实用前端教程

python爬虫人工智能大数据公众号

