

ANE-MySQL 数据库设计规范 SOP

上海安能聚创供应链管理有限公司

2017-05-12

目录

文档信息..... 2

文档修改记录..... 3

三. 前言..... 4

 1.文档简介..... 4

 2.适用环境..... 4

四、 规范..... 5

 4.1 数据库和表设计..... 5

 4.2 索引..... 6

 4.3 sql 脚本..... 8

 4.4 安全..... 8

文档信息

文档编号:		文档名称:	ANE-MySQL 数据库设计规范 SOP
文档说明:			
编 撰:	杨彪	编撰日期:	2017-05-12
审 核:		审核日期:	

文档修改记录

版本号	版本发布日	变更说明	编写者	审核者
1.0	2017-05-12	初版	杨彪	

三. 前言

1.文档简介

该文档主要用于 mysql 中的设计规范。

2.适用环境

本文档适用于所有使用 mysql 为后台数据库的开发人员。

四、规范

4.1 数据库和表设计

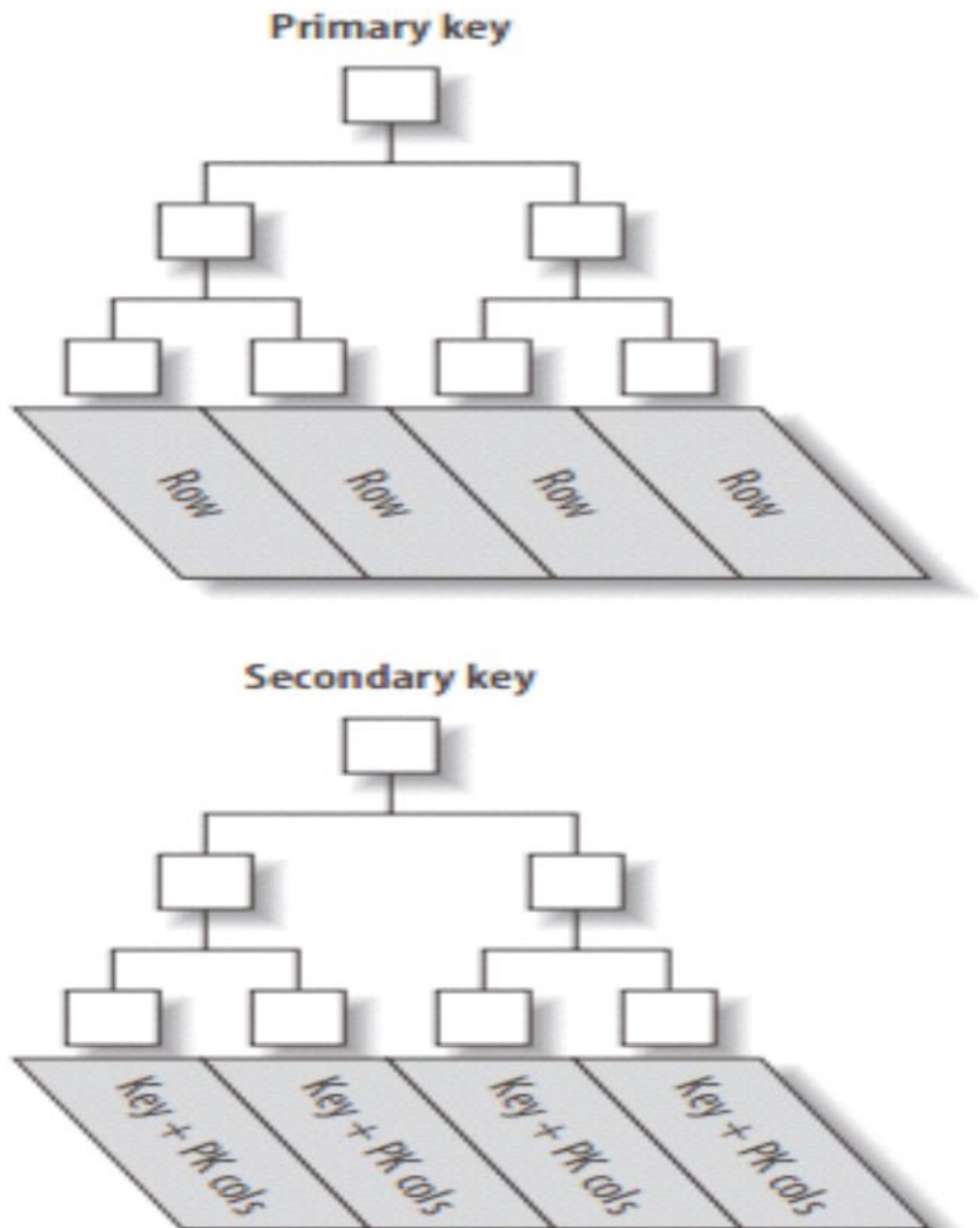
1. 库名、表名和字段名必须使用小写字母，并采用“_”进行分割
 - 1) Mysql 参数 `lower_case_table_names`，参数设置为 1，大小写不敏感。
 - 2) 如果大小写混用，会导致多表共存并且混乱，如 ABC，abc，Abc
 - 3) 字段显示区分大小写，使用不区分，不能同时创建如 ABC 和 abc 的字段
 - 4) 统一规范全部使用小写
2. 库名、表名和字段名不允许超过 20 个字符
库名、表名和字段名支持最多 64 个字符，为了易于辨识，控制字符数。
3. 库名、表名和字段名见名知意
能够从字段名或者表名直接看出什么意思。
4. 临时库、表名必须以 tmp 为前缀，并以日期为后缀
5. 备份库、表名必须以 bak 为前缀，并以日期为后缀
6. 所有表都需要加注释
7. 单表数据量建议控制在 5000w 以内
8. 建议使用 InnoDB 存储引擎（根据业务需要）
9. 库名、表名和字段名禁止使用 MySQL 保留字
10. 不建议使用分区表
11. 存储精度浮点数建议使用 decimal 代替 float 和 double
12. 建议使用 unsigned 存储非负数值
13. 建议使用 INT UNSIGNED 来存储 IPV4
14. 整型定义中建议不要添加长度，因为所占字节都是固定的
 - 1) Tinyint 占用 1 字节，smallint 占用 2 字节，int 占用 4 字节，bigint 占用 8 字节
 - 2) 括号里的数字只是显示宽度，和占用字节无关
15. 使用 TINYINT 来代替 ENUM 类型
 - 1) Enum 和 set 类型添加新值需要重新 DDL
 - 2) 默认值问题
 - 3) 索引值问题
16. 尽量不使用 text 和 blob 类型，如果必须使用，请单独存放
 - 1) 索引排序，只能使用 max_sort_length 的长度或者手工指定 order by substring(column,length)
 - 2) Memory 引擎不支持
 - 3) 浪费更多空间
 - 4) 可能无法使用 hash index
 - 5) 导致使用 where 没有索引的语句变慢
17. VARCHAR(N)，N 表示字符数，请根据实际宽度合理选择，尽可能小，因为在排序等操作中 MYSQL 会使用 N 长度的内存
18. 如果是固定字符数，请使用 CHAR(N)
增加检索性能
19. 库和表的字符集请选择 UTF8（数据库参数设置）

- 1) 不会有乱码风险
- 2) 支持几乎所有的语言
- 3) 汉字占用 3 个字节，ascii 是 1 个字节
20. 使用 `VARBINARY` 来存储大小写敏感的变长字符串或二进制文件
21. 存储年用 `YEAR` 类型，存储日期用 `DATE` 类型，存储到秒用 `TIMESTAMP`
 - 1) `Datetime` 和 `timestamp` 都是精确到秒，优先使用 `timestamp`，因为 `timestamp` 占用 4 个字节，`datetime` 占用 8 个字节
 - 2) `Timestamp` 具有自动赋值以及自动更新的特性，`current_timestamp`
22. 建议字段定义为 `NOT NULL`，并设置默认值
 - 1) 如果 `NULL` 字段被索引，需要额外的 1 个字节
 - 2) 如果有 `null` 则使索引和索引统计的值变的复杂
 - 3) 整型用 0 代替 `null`，字符型用''代替 `null`
 - 4) 索引字段一定要设置为 `not null`
23. 将大字段、访问频率低的字段放到单独表中，冷热数据分离
采用分布式文件系统更高效
24. 采用合适的分库分表策略
25. 数据库变更需要通知 DBA

4.2 索引

1. 索引名称必修使用小写
2. 一般索引名称需要按照 `idx_字段名[_字段名]` 来进行命名
3. 唯一索引名称需要按照 `uni_字段名[_字段名]` 来进行命名
4. 索引内的字段数建议不超过 5 个
5. 单表内的索引数建议不超过 5 个
在 `update`，`insert` 和 `delete` 的时候会重新对 `b+tree` 进行调整，降低更新的速度
6. 唯一键用 3 个以下字段组成，并且字段都为整型
7. 使用自增 ID 作为主键，建议与业务逻辑不相关的字段
8. 唯一键和主键不要重复
9. `ORDER BY, GROUP BY, DISTINCT` 必须用上索引
10. 使用 `EXPLAIN` 进行判断是否使用索引
能够检查出 SQL 语句是否为最优，是否使用索引，是否全表扫描等等问题
11. `Update`，`delete` 语句需要使用 `where` 添加索引字段
提升更新的速度
12. 多表 `join` 的字段需要用索引
13. 不建议使用 `like %` 前缀查询，这样不能使用索引
14. 合理创建联合索引，避免冗余，如 `(a,b,c)` 相当于 `(a) (a,b) (a,b,c)`
15. 合理使用覆盖索引
 - 1) 非主键索引没有存储行地址，只存了主键值，所以必须要先查找到主键，再从主键定位到行数据。
 - 2) 覆盖索引的概念就是只通过一个索引完成行的定位
 - 3) 主键是天生的覆盖索引
 - 4) 覆盖索引即是通过 `where` 的条件直接检索需要的字段，然后创建条件+字段的索引

- 5) 举例:select a,b from c where b=xxx,如果 b 不是主键, 可以创建索引(b,a)提升性能
16. 不在低基数列上建索引, 例如性别
17. 不在索引列上进行数学运算和函数运算
18. 尽量不使用外键, 外键可在业务端实现并且对父表和子表的操作会相互影响
- 1) 外键是用来保护参照的完整性, 可在业务端实现
 - 2) 对父表和子表的操作会相互影响
 - 3) Innodb 本身对 online DDL 的限制
19. 变更索引需要通知 DBA
20. 索引差别



4.3 sql 脚本

1. sql 语句中 in 和 exists 需要合理使用
2. update 和 delete 不使用 limit
会造数据的不一致性
3. where 条件中必须使用合适的类型，避免隐式转换
4. where 条件中的转换必须放在值里面，而不要放到字段上
不要使用 where date(c1),而要使用 where c1=date(c1)
5. select 只提取需要的字段，不使用 select *
6. 使用 union all 而不使用 union
7. select into 必须指明字段，不要使用 select * into table
8. where 条件后的非等式(in,between,<,>,>=,<=)会导致使用不了索引
9. 拆分复杂 sql 为小 sql，避免大事务
 - 1) 简单的 sql 容易使用到 query cache
 - 2) 减少锁表时间，特别是 myisam
 - 3) 可以使用多核 CPU
10. 不使用负向查询，如 not in，这样无法使用索引，导致全表扫描
11. 使用预编译语句，只传参数，一次编译，多次使用，并且避免注入
12. 充分利用前缀索引，必须是最左前缀
13. 避免使用存储过程，触发器和函数
14. 避免使用大表 join，能在一张表上完成的 sql 不要去 join 两张表，join 消耗内存并且产生临时表
15. 减少与数据库的交互
16. 合理使用分页

4.4 安全

1. 禁止在生产库上做压力测试
2. 禁止从测试和开发环境直接连接数据库
3. 禁止在数据库中保存明文密码
4. 程序应该有捕捉 sql 异常的机制
5. 导出导入数据必须提前通知 DBA
6. 禁止有 super 的账号存在
7. 每个项目一个用户，每个项目一个数据库
8. 出现非数据库导致的故障及时通知 DBA 协助排查
9. 数据库数据丢失，及时通知 DBA 进行修复
10. 项目的数据库选型和设计必须提前通知 DBA 进行参与
11. 对重要的库和表通知 DBA 确定维护和备份的优先级
12. 不在业务高峰期进行批量更新
13. 提交线上建表改表需求，详细注明相关 SQL