# ****有利网MYSQL开发规范****

# ****1 命名规范****

## 1.1  命名规范

* 库名、表名、字段名必须使用小写字母，并采用下划线进行分割。
* 库名、表名、字段名禁止超过32个字符。须见名知意。
* 库名、表名、字段名禁止使用MySQL保留字。
* 临时库、表名必须以tmp为前缀，并以日期为后缀。
* 备份库、表必须以bak为前缀，并以日期为后缀。
* 索引命名以idx\_为前缀。唯一索引以uniq\_为前缀

# ****2 基础规范****

## 2.1   基础规范

1. 使用INNODB存储引擎
2. 表字符集使用UTF8
3. 所有表都需要添加注释
4. 所有的列都需要添加注释
5. 单表数据量建议控制在5000W以内
6. 不在数据库中存储图片、文件等大数据
7. 禁止在线上做数据库压力测试
8. 禁止从测试、开发环境直连数据库

# ****3 库表设计****

## 3.1  库表设计

* 禁止使用分区表
* 拆分大字段和访问频率低的字段，分离冷热数据
* 用HASH进行散表，表名后缀使用十进制数, 下标从0开始
* 按日期时间分表需符合YYYY[MM][DD][HH]格式
* 采用合适的分库分表策略。例如一库十表、十库百表等
* 建表时自增列的类型为int或者bigint，指定初始值为1
* 自增列必须是无符号型   unsigned

## 3.2   字段设计

* 尽可能不使用TEXT、BLOB类型
* 用DECIMAL代替FLOAT和DOUBLE存储精确浮点数
* 将字符转化为数字
* 使用TINYINT来代替ENUM类型
* 存储 “hello”时VARCHAR(5) VSVARCHAR(200)
* 所有字段均定义为NOT NULL
* 使用UNSIGNED存储非负整数
* INT类型固定占用4字节存储
* 使用timestamp存储时间
* 禁止在数据库中存储明文密码

## 3.3索引设计

### 3.3.1索引的用途

* 去重
* 加速定位
* 避免排序
* 覆盖索引

### 3.3.2索引数量的控制

* 单张表中索引数量不超过8个
* 单个索引中的字段数不超过5个
* 对字符串使用前缀索引，前缀索引长度不超过8个字符
* 建议优先考虑前缀索引，必要时可添加伪列并建立索引

### 3.3.3 主键的准则

* 表必须有主键
* 不使用更新频繁的列
* 尽量不选择字符串列
* 不使用UUID MD5 HASH
* 默认使用非空的唯一键
* 建议选择自增或发号器

### 3.3.4 索引设计准则

* 重要的SQL必须被索引
* UPDATE、DELETE语句的WHERE条件列
* ORDER BY、GROUP BY、DISTINCT的字段

多表JOIN的字段

区分度最大的字段放在前面

核心SQL优先考虑覆盖索引

避免冗余和重复索引

索引不是越多越好

综合评估数据密度和分布

考虑查询和更新比例



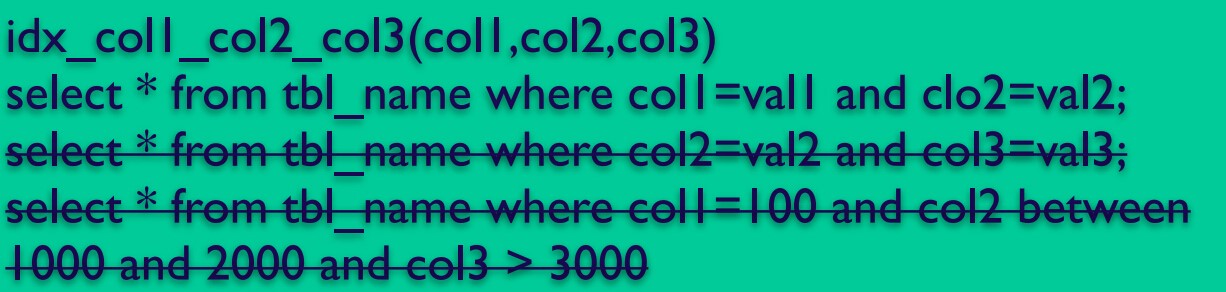
### 3.3.5  ****索引禁忌****

* 不在低基数列上建立索引，例如“性别”
* 不在索引列进行数学运算和函数运算
* 尽量不使用外键
* 外键用来保护参照完整性，可在业务端实现
* 对父表和子表的操作会相互影响，降低可用性
* INNODB本身对onlineDDL的限制
* 不使用%前导的查询，如like “%ab”
* 不使用负向查询，如not in/like
* 无法使用索引，导致全表扫描
* 全表扫描导致bufer pool利用率降低

## 3.4 SQL的设计

### 3.4.1 SQL设计

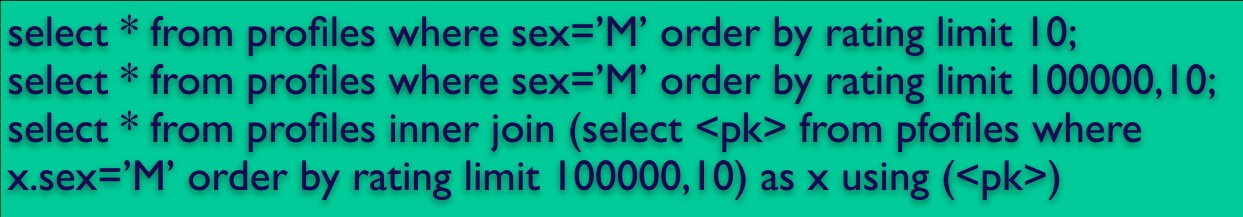
* DML语句（insert，update，delete）
  1. 禁止在DML语句中使用LIMIT
  2. 禁止在DML语句中使用order by
  3. DML语句中必须包含where条件
* 使用预编译语句
  1. 只传参数，比传递SQL语句更⾼效
  2. 一次解析，多次使用
  3. 降低SQL注入概率
* 避免隐式转换
  1. 会导致索引失效
* 充分利用前缀索引
  1. 必须是最左前缀
  2. 不可能同时用到两个范围条件



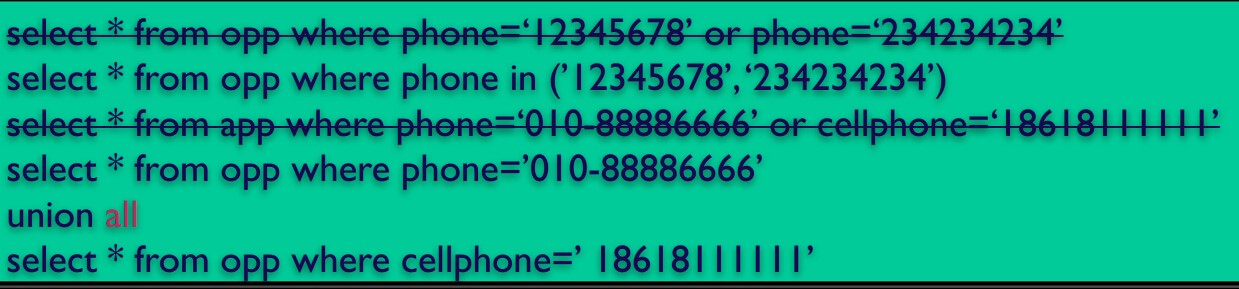
* 避免使用存储过程，触发器，UDF，events等
  1. 让数据库做最擅长的事
  2. 降低业务耦合度,为sacle out、sharding留有余地
  3. 避开BUG
* 避免使用大表的JOIN
  1. MySQL最擅长的是单表的主键/二级索引查询
  2. JOIN消耗较多内存，产生临时表
* 避免在数据库中进行数学运算
  1. MySQL不擅长数学运算和逻辑判断
  2. 无法使用索引



* 减少与数据库的交互次数
  1. INSERT ... ON DUPLICATE KEY UPDATE
  2. REPLACE INTO、INSERT IGNORE 、INSERT INTO VALUES(),(),()
  3. UPDATE … WHERE ID IN(10,20,50,…)
* 拒绝大SQL，拆分成小SQL
  1. 充分利用QUERY CACHE
  2. 充分利用多核CPU



* 使用in代替or, in的值不超过1000个
* 禁止使用order by rand()
* 使用EXPLAIN诊断，避免生成临时表
* 用union all而不是union
* 程序应有捕获SQL异常的处理机制
* 禁止单条SQL语句同时更新多个表
* 不使用select \*
  1. 消耗CPU和IO、消耗网络带宽
  2. 无法使用覆盖索引
  3. 减少表结构变更带来的影响
  4. 因为select/join 可能生成临时表



# 4       行为规范

## 4.1   行为规范

* 批量导入、导出数据必须提前通知DBA协助观察
* 禁止在从库上执行后台管理和统计类功能的查询
* 禁止有super权限的应用程序账号存在
* 产品出现非数据库导致的故障时及时通DBA协助排查
* 推广活动或上线新功能必须提前通知DBA进行流量评估
* 数据库数据丢失，及时联系DBA进行恢复
* 对单表的多次alter操作必须合并为一次操作
* 不在MySQL数据库中存放业务逻辑
* 重大项目的数据库方案选型和设计必须提前通知DBA参与
* 对特别重要的库表，提前与DBA沟通确定维护和备份优先级
* 不在业务高峰期批量更新、查询数据库
* 提交线上建表改表需求，必须详细注明所有相关SQL语句