1，命名规范

库名、表名、字段名必须使⽤小写字母，并采⽤用下划线分割

库名、表名、字段名禁止超过32个字符。须见名之意

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

2，基础规范

使用INNODB存储引擎

表字符集使用UTF8

所有表都需要添加注释，及查询业务注释

单表数据量建议控制在5000W以内

不在数据库中存储图片、文件等大数据

禁止在线上做数据库压力测试

禁止直接连接线上环境，查询都不可以(可以在254从库做相关查询)

3，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

按日期时间分表需符合YYYY[MM][DD][HH]格式

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

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

使用INT UNSIGNED存储IPV4

使用VARBINARY存储大小写敏感的变长字符串

禁止在数据库中存储明文密码

4,主键准则

表必须有主键

不使用更新频繁的列

尽量不选择字符串列

默认使用非空的唯一键

建议选择自增或发号器

5,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

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online DDL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buffer pool利用率降低）

6,sql设计

使用预编译语句，只传参数，比传递SQL语句更高效，一次解析，多次使用，降低SQL注入概率

避免隐式转换，会导致索引失效

充分利用前缀索引，必须是最左前缀

避免使用存储过程、触发器、UDF、events等，让数据库做最擅长的事

避免使用大表的JOIN

避免在数据库中进行数学运算（MySQL不擅长数学运算和逻辑判断，无法使用索引）

合理的使用分页，（限制分⻚页展示的页数，只能点击上一页、下一页，采用延迟关联)

拒绝大SQL，将其拆分成小SQL（充分利用QUERY CACHE，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()

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 \*

7,行为规范

批量导入、导出数据必须提前通知DBA协助观察

禁止在线上从库执行后台管理和统计类查询

产品出现非数据库导致的故障时及时通知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