

3天挑战架构师级MySQL海量数据设计与实践

内容介绍

Day1 MySQL架构体系设计深入剖析篇

MySQL架构体系拆解以及设计深度剖析;
MySQL存储引擎原理拆解以及设计深度剖析;
MySQL锁实现原理拆解以及设计深度剖析;
MySQL事务实现原理拆解以及设计深度剖析;

Day 2 企业千亿级海量数据并发分库分表设计方法论提炼篇

千亿级海量数据高并发场景分库分表设计方法论;
千亿级海量数据高并发场景主键设计选择;
千亿级海量数据高并发场景分片键设计选择;
千亿级海量数据高并发场景分库实践落地方案;
千亿级海量数据高并发场景分表实践落地方案;

Day 3 企业千亿级海量数据真实案例设计与实战篇

万亿级微信消息垂直拆分真实案例实战;
企业级数据库Sharding Sphere分库分表应用设计实战;
企业级分布式事务阿里巴巴Seata应用设计实战;
企业级MySQL数据库高可用应用设计与实战;



陈东

前58集团架构师，前转转公司架构平台部负责人、高级架构师、技术委员会核心成员，主导了转转基础架构部门从0到1的建设工作，负责转转RPC框架和服务治理生态的落地、消息队列的研发和多元化存储体系的建设，以及众多核心基础组件的设计研发和产品化工作，擅长后端架构、中间件、服务治理、存储等技术方向，对即时通讯系统有深刻的研究。

今晚20:00准时开课，请稍后...

奈学教育 出品

- 千亿级海量数据高并发场景主键设计选择;
- 千亿级海量数据高并发场景分库分表设计方法论;
- 千亿级海量数据高并发场景分库分表实践落地方案;



01. 千亿级海量数据高并发场景主键设计选择

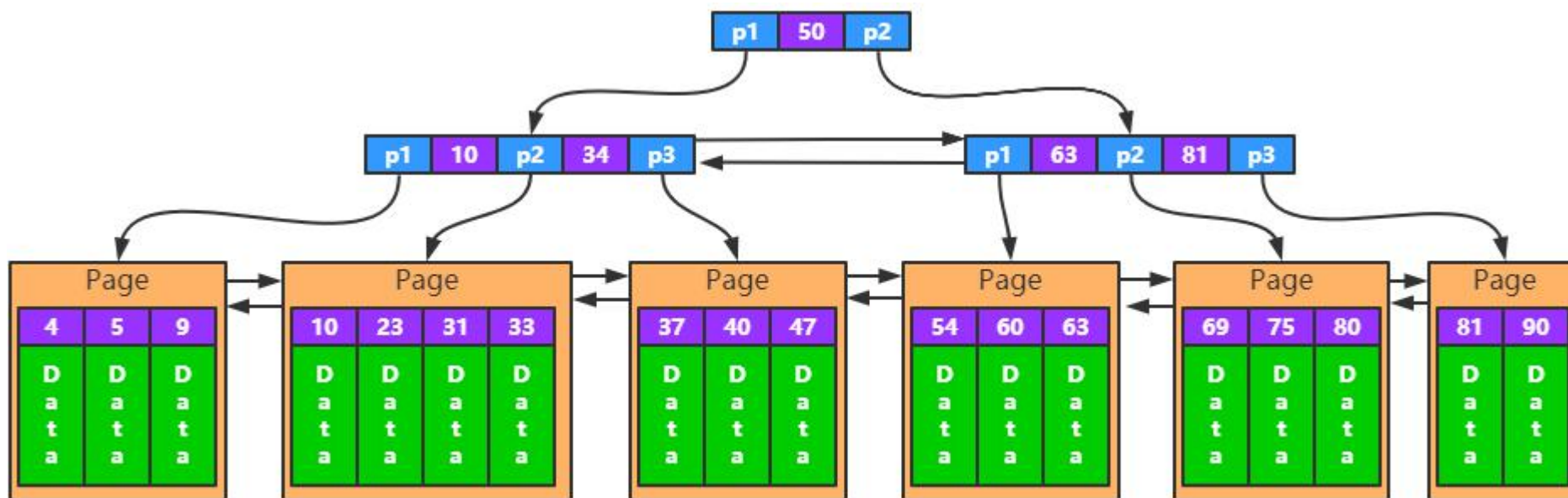
01. 千亿级海量数据高并发场景主键设计选择

索引原理分析

- 聚簇索引
- 二级索引
- 联合索引

索引原理分析

- 聚簇索引
- 二级索引
- 联合索引



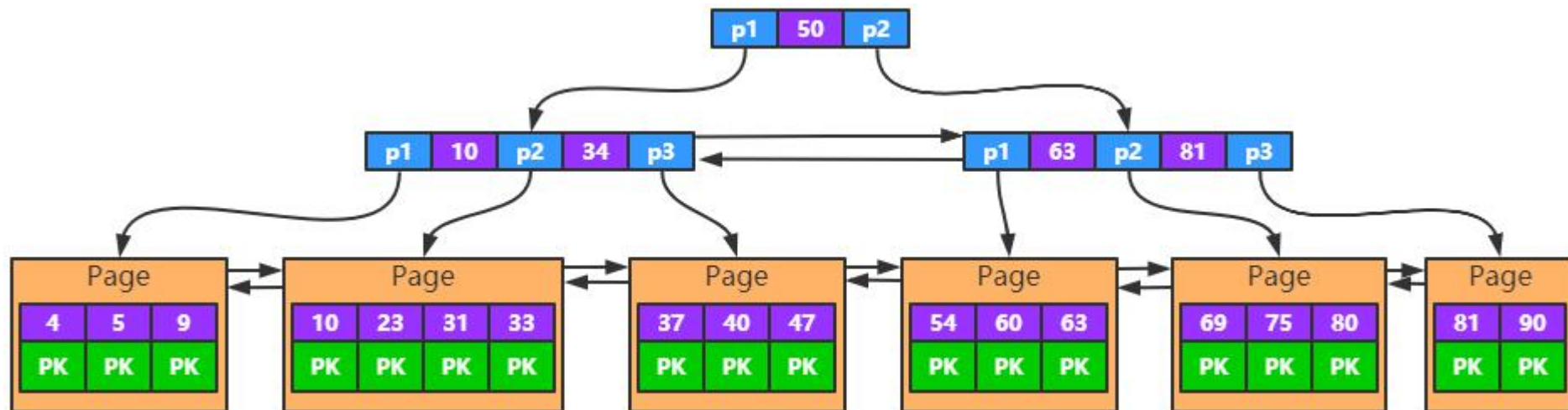
聚簇索引

- 数据存储在主键索引中
- 数据按主键顺序存储

自增主键 VS 随机主键

索引原理分析

- 聚簇索引
- **二级索引**
- 联合索引



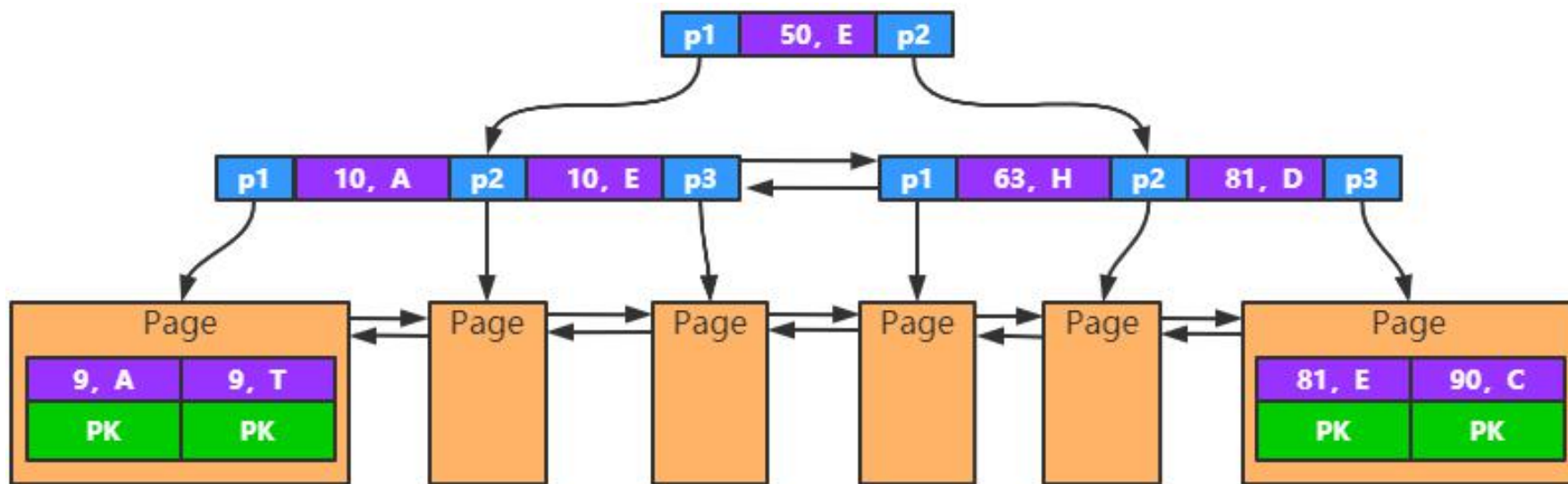
01. 千亿级海量数据高并发场景主键设计选择

二级索引

- 除主键索引以外的索引
- 叶子中存储主键值
- 一次查询需要走两遍索引
- 主键大小会影响所有索引的大小

索引原理分析

- 聚簇索引
- 二级索引
- **联合索引**



01. 千亿级海量数据高并发场景主键设计选择

联合索引

- Key由多个字段组成
- 最左匹配原则
- 一个索引只创建一棵树
- 按第一列排序，第一列相同按第二列排序

如果不是按照最左开始查找，无法使用索引
不能跳过中间列
某列使用范围查询，后面的列不能使用索引

01. 千亿级海量数据高并发场景主键设计选择

索引使用优化分析

- 存储空间
- 主键选择
- 联合索引使用
- 字符串索引

索引使用优化分析

➤ 存储空间

- 索引文件大小;
- 字段大小->页内节点个数->树的层数;

➤ 主键选择

➤ 联合索引使用

➤ 字符串索引

BIGINT类型主键3层可以存储约10亿条数据

$16KB / (8B(key) + 8B(指针)) = 1K$

$10^3 * 10^3 * 10^3 = 10\text{亿}$

32字节主键3层可以存储6400W

索引使用优化分析

➤ 存储空间

➤ 主键选择

- 自增主键，顺序写入，效率高;
- 随机主键，结点分裂、数据移动;

➤ 联合索引使用

➤ 字符串索引

自增主键：写入磁盘利用率高，每次查询走两级索引；

随机主键：写入磁盘利用率低，每次查询走两级索引；

业务主键：写入、查询磁盘利用率都高，可以使用一级索引；

联合主键：影响索引大小，不易维护，不建议使用；

01. 千亿级海量数据高并发场景主键设计选择

索引使用优化分析

- 存储空间
- 主键选择
- **联合索引使用**
 - 按索引区分度排序
 - 覆盖索引
 - 索引下推
- 字符串索引

01. 千亿级海量数据高并发场景主键设计选择

索引使用优化分析

- 存储空间
- 主键选择
- 联合索引使用
- **字符串索引**
 - 设置合理长度
 - 不支持%开头的模糊查询

索引失效问题分析

什么情况会导致索引失效？

索引失效问题分析

➤ $A=XX \text{ OR } B=XX$

5.1开始，引入Index merge技术，对同一个表可以使用多个索引分别扫描

索引失效问题分析

- 隐式类型转换

WHERE moblie = '135XXXX';
WHERE mobile = 135XXX;

01. 千亿级海量数据高并发场景主键设计选择

索引失效问题分析

- 索引列包含计算

WHERE name-20 = 30;

索引失效问题分析

- 数据范围影响

**索引区分度过低
条件超出索引范围**

01. 千亿级海量数据高并发场景主键设计选择

MySQL库表设计实践

- 联合索引：优于多列独立索引
- 索引顺序：选择性高的在前面
- 覆盖索引：Key里面包含要查询的数据
- 索引排序：索引同时满足查询和排序

01. 千亿级海量数据高并发场景主键设计选择

MySQL库表设计实践

- 数据库字符集使用utf8mb4;
- VARCHAR按实际需要分配长度;
- 文本字段建议使用VARCHAR;
- 时间字段建议使用long;
- bool字段建议使用tinyint;
- 枚举字段建议使用tinyint;
- 交易金额建议使用long;
- 禁止使用 “%” 前导的查询;
- 禁止在索引列进行数学运算，会导致索引失效;
 - `select * from t1 where id+1 >1121` 不会使用索引
 - `select * from t1 where id >1121 - 1` 会使用索引

01. 千亿级海量数据高并发场景主键设计选择

MySQL库表设计实践

- 表必须有主键，建议使用业务主键；
- 单张表中索引数量不超过5个；
- 单个索引字段数不超过5个；
- 字符串索引使用前缀索引，前缀长度不超过10个字符；

01. 千亿级海量数据高并发场景主键设计选择

MySQL库表设计实践

- 是否分表
 - 建议单表不超过1KW
- 分表方式
 - 取模：存储均匀&访问均匀
 - 按时间：冷热库
- 分库
 - 按业务垂直分
 - 水平查分多个库



02. 千亿级海量数据高并发场景分库分表设计方法论

分布分表方案

- 垂直拆分
- 水平拆分

分布分表方案

- 垂直拆分
 - 宽表拆多个表
- 水平拆分

分布分表方案

- 垂直拆分
- 水平拆分
 - 读写均匀
 - 冷热库拆分



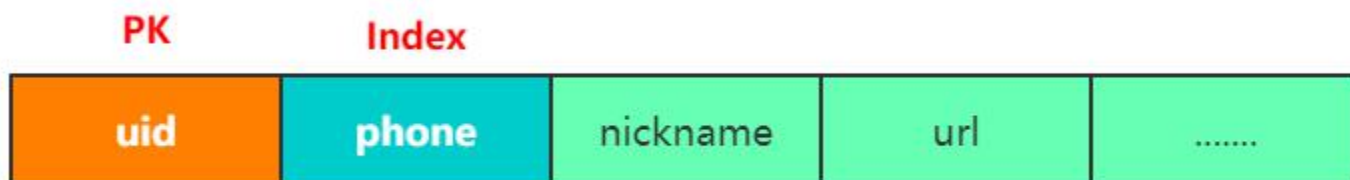
03. 千亿级海量数据高并发场景分库分表实践落地方案

分库分表—案例分享

- 用户库分表
- 商品库分表
- 系统消息库分表

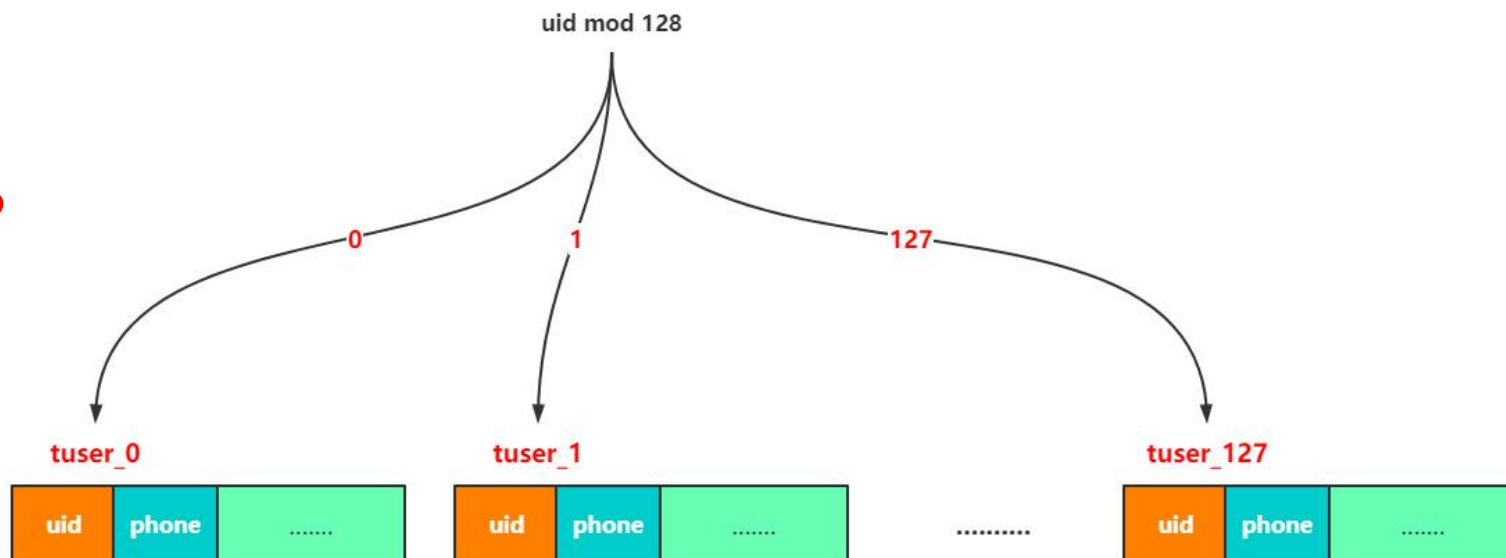
分库分表—案例分享

- 用户库分表
- 商品库分表
- 系统消息库分表



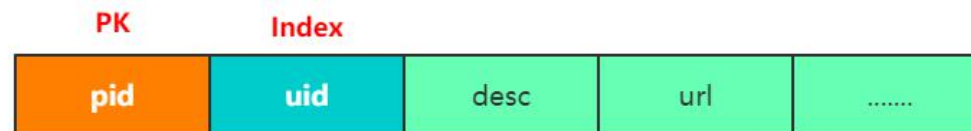
存储均匀&访问均匀 -> 取模

手机号查询怎么办?

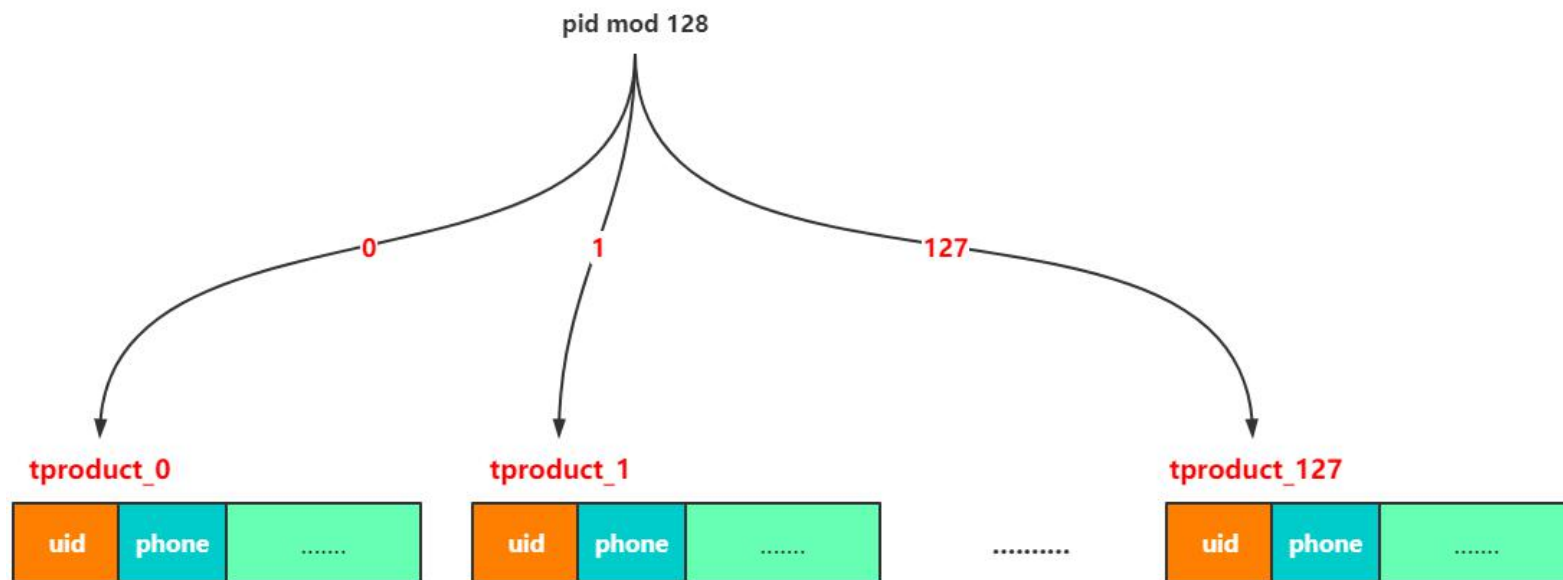


分库分表—案例分享

- 用户库分表
- **商品库分表**
- 系统消息库分表



查询自己发布的商品



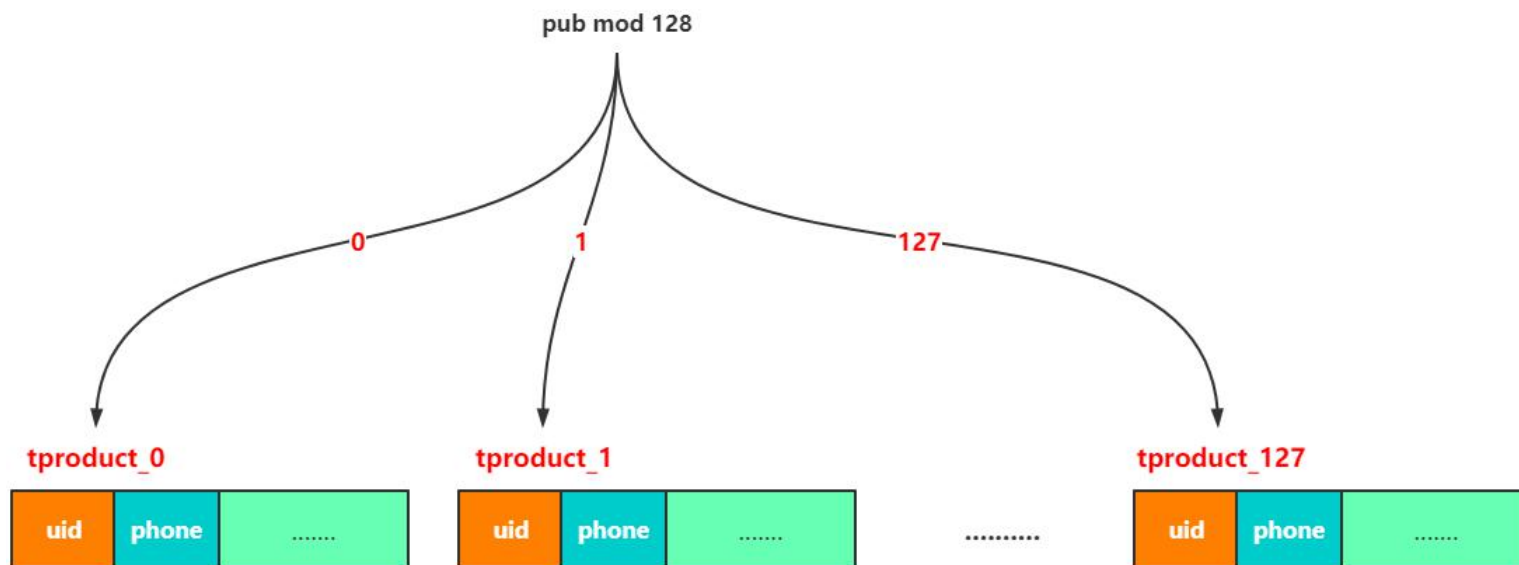
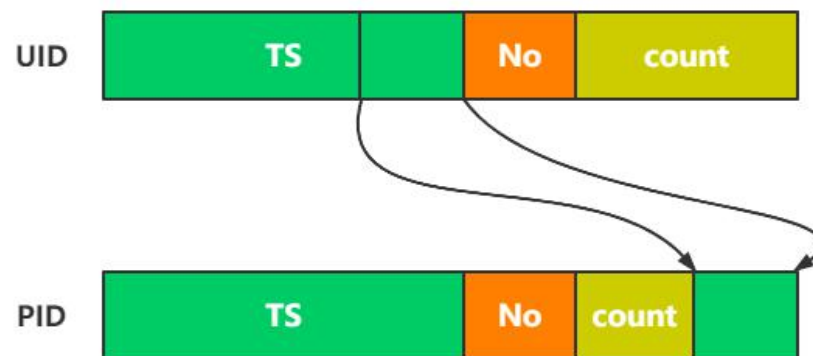
分库分表—案例分享

➤ 用户库分表

➤ 商品库分表

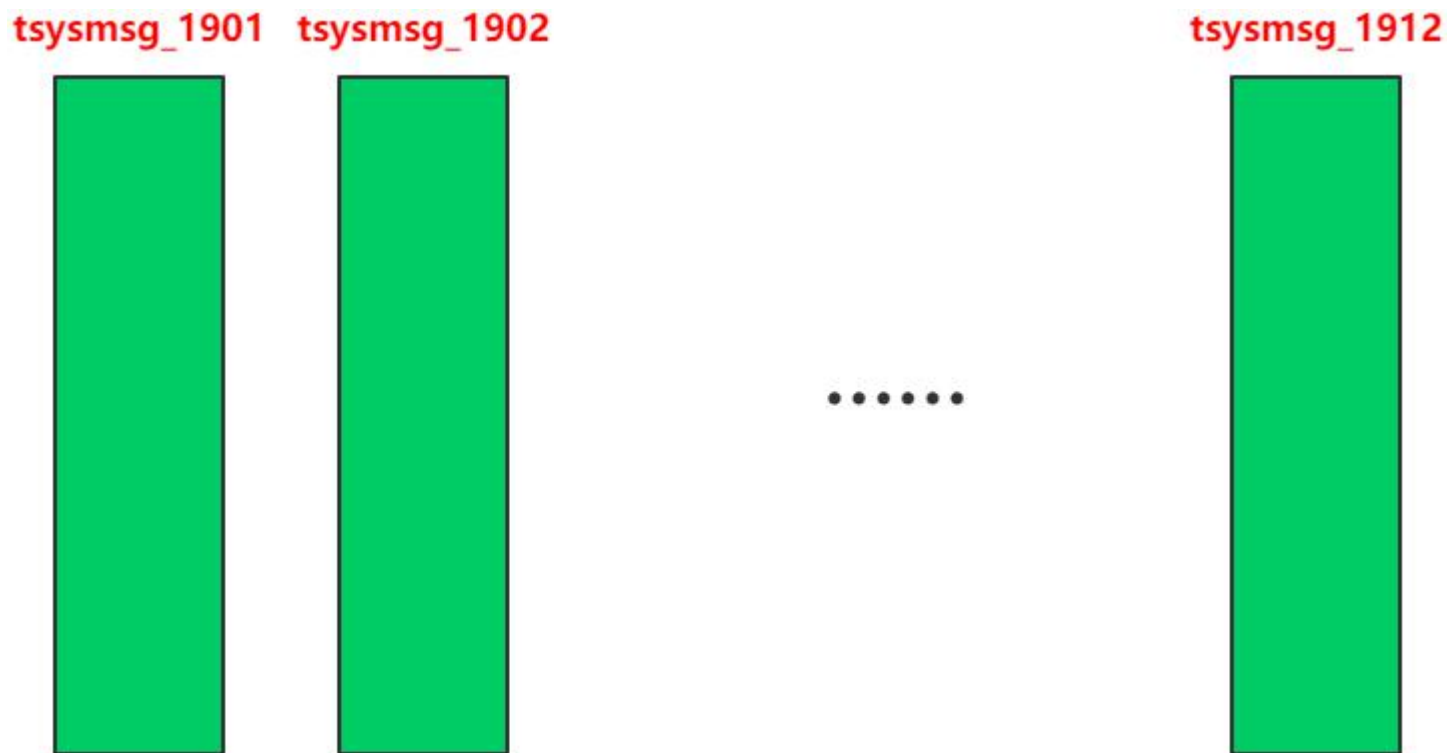
- 两个维度查询必须满足
- 映射表太重

➤ 系统消息库分表



分库分表—案例分享

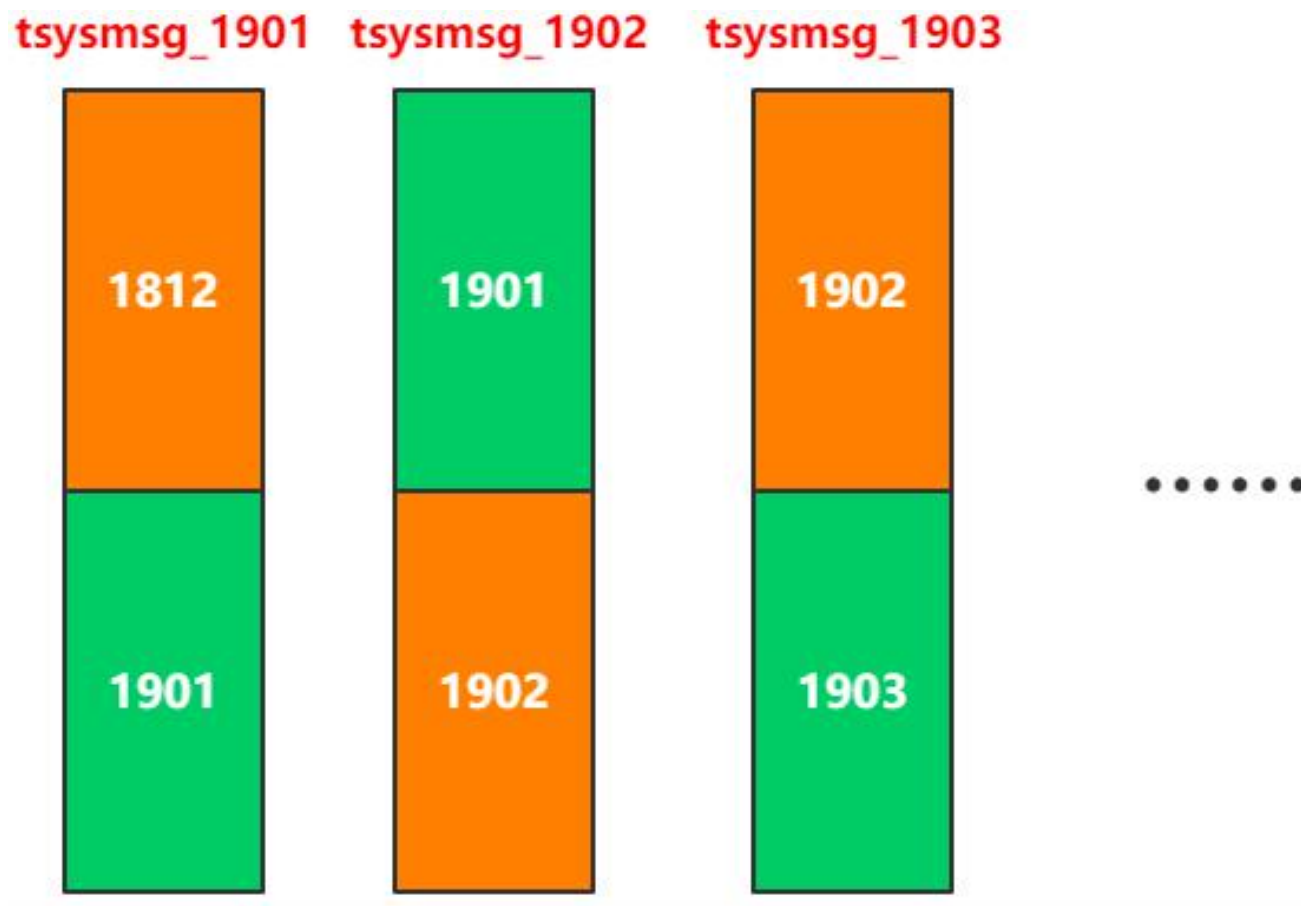
- 用户库分表
- 商品库分表
- **系统消息库分表**
 - 时效性强
 - 冷热数据拆分



思考：有效期30天

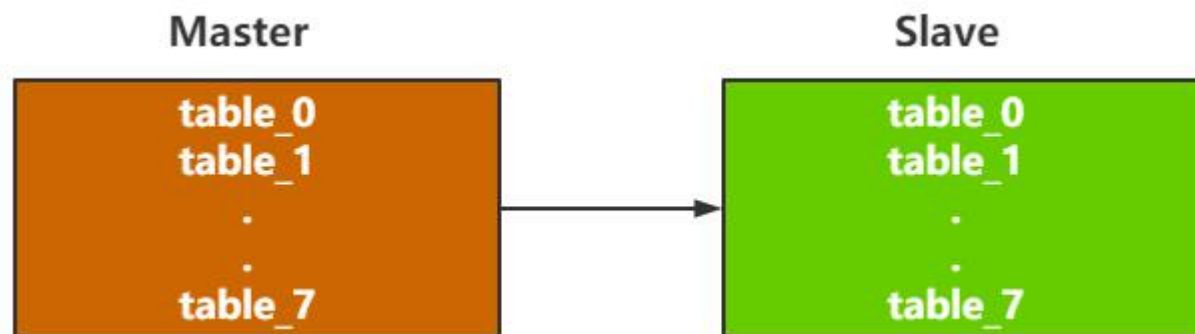
分库分表—案例分享

- 用户库分表
- 商品库分表
- **系统消息库分表**
 - 时效性强
 - 冷热数据拆分

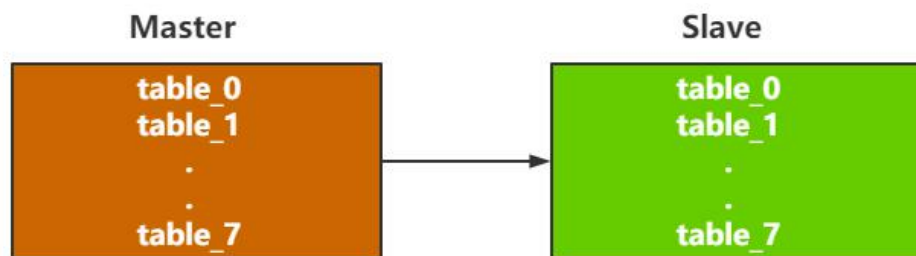


分库分表—案例分享

➤ 分表分少了怎么办？



分库分表—案例分享



业务修改路由算法，后台清理数据

NiX 奈学教育



欢迎关注本人公众号
“架构之美”