Atitit 负载均衡 数据分片 切分一句 userid 时间字段 hashkey

目录

[1. 区分全局表和业务表 1](#_Toc2038)

[1.1. 只对业务表进行分片 1](#_Toc231)

[1.2. ．按功能划分 1](#_Toc23724)

[1.3. 按 userid 推荐 1](#_Toc29953)

[2. 方案三：区域 按数值范围 （不可可去） 1](#_Toc27333)

[3. 不推荐的切分法 2](#_Toc18465)

[3.1. 4．基于路由表的切分 2](#_Toc17215)

[3.2. 时间字段（推荐使用场景：日志记录） 2](#_Toc27328)

[3.3. 基于hash的切分（垃圾切分法不要用 扩容困难） 2](#_Toc29277)

系统数据的隔离性 ，业务一般是工具功能和用户来隔离数据的。。所以使用功能和用户key切分比较好。。 后台管理admin 的某些时间相关的数据，可以同步到一个从库，从库使用时间切分即可。。如果后台既要时间切片又要用户切片做groupby，建议同步到es等nosql库，使用groupby dsl 实现。。

多维度查询，可以同步到一个特定维度的从库，使用不同的切片规则。

也可以直接同步到all in one，大数据库，使用dsl groupby 查询即可。。

# 区分全局表和业务表

## 只对业务表进行分片

## ．按功能划分

## 按 userid 推荐

# 方案三：**区域** 按数值范围 （不可可去）

# 不推荐的切分法

## 4．基于路由表的切分

前面的几种方式都是根据应用的数据来决定操作的，基于路由表的切分是一种更加松散的方法。它单独维护一张路由表，根据用户的某一属性来查找路由表决定使用哪个数据库，这种方式是一种更加通用的方案

优点：id和库的mapping算法可以随意更改

缺点：可能引入额外的单点

## 时间字段（推荐使用场景：日志记录）

## 基于hash的切分（垃圾切分法不要用 扩容困难）