Atitit 跨库join解决方案

目录

[1.1. 同实例跨库直接join 1](#_Toc23303)

[1.2. 全局表 1](#_Toc23128)

[1.3. 字段冗余 1](#_Toc27432)

[1.4. 数据同步 2](#_Toc16608)

[1.5. OLAP数据仓库+es 2](#_Toc18925)

[1.5.1. 上面举例的都太简单了，我们的后台报表系统中join的表都有n个了， 分库后该怎么查？ 2](#_Toc26647)

[1.6. Feb存储邀请 2](#_Toc6615)

[1.7. 系统层组装 join rowSEt 2](#_Toc30538)

[1.8. 引用层 变化子查询组装即可 2](#_Toc137)

跨库Join的几种解决思路

## 同实例跨库直接join

## 全局表

所谓全局表，就是有可能系统中所有模块都可能会依赖到的一些表。比较类似我们理解的“数据字典”。为了避免跨库join查询，我们可以将这类表在其他每个数据库中均保存一份。同时，这类数据通常也很少发生修改（甚至几乎不会），所以也不用太担心“一致性”问题。

## 字段冗余

这是一种典型的反范式设计，在互联网行业中比较常见，通常是为了性能来避免join查询。  
举个电商业务中很简单的场景：

“订单表”中保存“卖家Id”的同时，将卖家的“Name”字段也冗余，这样查询订单详情的时候就不需要再去查询“卖家用户表”。

字段冗余能带来便利，是一种“空间换时间”的体现。但其适用场景也比较有限，比较适合依赖字段较少的情况。最复杂的还是数据一致性问题，这点很难保证，可以借助数据库中的触发器或者在业务代码层面去保证。当然，也需要结合实际业务场景来看一致性的要求。就像上面例子，如果卖家修改了Name之后，是否需要在订单信息中同步更新呢？

## 数据同步

定时A库中的tab\_a表和B库中tbl\_b有关联，可以定时将指定的表做同步。当然，同步本来会对数据库带来一定的影响，需要性能影响和数据时效性中取得一个平衡。这样来避免复杂的跨库查询。笔者曾经在项目中是通过ETL工具来实施的。

## OLAP数据仓库+es

### 上面举例的都太简单了，我们的后台报表系统中join的表都有n个了， 分库后该怎么查？

有很多朋友跟我提过类似的问题。其实互联网的业务系统中，本来就应该尽量避免join的，如果有多个join的，要么是设计不合理，要么是技术选型有误。请自行科普下OLAP和OLTP，报表类的系统在传统BI时代都是通过OLAP数据仓库去实现的（现在则更多是借助离线分析、流式计算等手段实现），而不该向上面描述的那样直接在业务库中执行大量join和统计。

## Feb存储邀请

## 系统层组装 join rowSEt

在系统层面，通过调用不同模块的组件或者服务，获取到数据并进行字段拼装。说起来很容易，但实践起来可真没有这么简单，尤其是数据库设计上存在问题但又无法轻易调整的时候。

具体情况通常会比较复杂。下面笔者结合以往实际经验，并通过伪代码方式来描述。

## 引用层 变化子查询组装即可

应用对子对象。。级联查询即可。。循环foreach，，增加为子对象即可。。简单。。

## 解析sql ，跨库join 子查询json模式挂载返回即可