Atitit join的实现

作者：聿明leslie  
链接：https://www.zhihu.com/question/68258877/answer/264097272  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

但是确实大多数业务都会考虑把这种合并操作放到service层，我觉得有几方面考虑：  
第一：单机数据库计算资源很贵，数据库同时要服务写和读，都需要消耗CPU，为了能让数据库的吞吐变得更高，而业务又不在乎那几百微妙到毫秒级的延时差距，业务会把更多计算放到service层做，毕竟计算资源很好水平扩展，数据库很难啊，所以大多数业务会把纯计算操作放到service层做，而将数据库当成一种带事务能力的kv系统来使用，这是一种重业务，轻DB的架构思路  
第二：很多复杂的业务可能会由于发展的历史原因，一般不会只用一种数据库，一般会在多个数据库上加一层中间件，多个数据库之间还能做毛的join，自然业务会抽象出一个service层，降低对数据库的耦合。  
第三：对于一些大型公司由于数据规模庞大，不得不对数据库进行分库分表，这个问题我在《阿里为什么要禁用三表以上的join》上也回答过，对于分库分表的应用，使用join也受到了很多限制，除非业务能够很好的根据sharding key明确要join的两个表在同一个物理库中。而中间件一般对跨库join都支持不好。举一个很常见的业务例子，在分库分表中，要同步更新两个表，这两个表位于不同的物理库中，为了保证数据一致性，一种做法是通过分布式事务中间件将两个更新操作放到一个事务中，但这样的操作一般要加全局锁，性能很捉急，而有些业务能够容忍短暂的数据不一致，怎么做？让它们分别更新呗，但是会存在数据写失败的问题，那就起个定时任务，扫描下A表有没有失败的行，然后看看B表是不是也没写成功，然后对这两条关联记录做订正，这个时候同样没法用join去实现，只能将数据拉到service层应用自己来合并了。

邀。我姑且认为题主说的是内连接。处理内连接肯定是放在数据库上做，毫无疑问。

从内连接的角度来讲，数据库通常会实现2种执行计划。第一种是NESTED LOOP。这种执行计划通常用于小表关联，能够最大限度的减少内存开销并尽快返回结果。第二种是HASH。这种执行计划通常用于小表和大表关联，或者大表和大表关联，会有一些内存开销，但是大数据量下的处理速度是NESTED LOOP不可比拟的。如果2张表都是十几万条记录的话，数据库的查询优化器应该会选择HASH JOIN来实现内连接操作。而如果关联操作放在应用里做，几乎所有程序员都会选择2重循环去实现，这种逻辑是被HASH JOIN完爆的。

还有就是放在应用层做还有很大的网络开销，主要是数据库服务器和应用服务器之间的开销。显然，从网络开销的角度来说，也应该把内连接放在数据库上实现。

作者：徐志斌  
链接：https://www.zhihu.com/question/68258877/answer/272939933  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。