在传统的项目中，我们的读和写是放在一个应用中，这给我们带来了如下问题

1. 大多少情况下我们的写是比读要少很多，但我们无法针对读或写进行单独的扩容
2. 我们的应用程序会面向不同的人群，其所关注的数据也各不相同，为了查询更加方便，我们会引入各种关联，这会破坏我们的模型

CQRS就是解决如上问题的，CQRS衍生于领域驱动设计，其所关注的是读写分离，按照CQRS的方法，我们可以将项目划分为读项目和写项目，读项目连接从数据库，使用贫血模型，读项目一般使用mvc架构，而写项目连接主数据库，使用充血模型，架构为领域驱动设计

CQRS的另一个优点就是读项目设计的领域逻辑很少，可以分配给不了解领域的人员或者外包。

相对于领域驱动设计，CQRS引入了命令/事件总线。领域逻辑是由命令驱动的，在执行领域逻辑后会产生领域事件，而领域事件又会被发送到总线上。

CQRS下可以引入Saga模式，Saga代表长时间活动的事务，可以理解为一个流程，其包含一个流程的所有操作以及终止流程的补偿操作，如下单Saga会包含下单处理器，检查库存处理器，支付处理器，以及一些补偿处理器等，如果用户在未完成下单前取消了订单，那么这些补偿处理器就会运行，saga一般持久化到数据库中