1. 数据库表的设计

业务能够切分

核心系统是ERP

财务系统

网站

分单 第三方 自订

秒杀-

1. 数据库表结构设计 – 分层

基础数据

上层业务是根据基础数据操作和呈现

用户分为vip , 普通，超级vip

根据用户type先拆分，再取模，最后存储用户信息；可能会涉及多个数据库的join

1. 数据库表结构设计&拆分
2. mysql水平拆分（分片：单表限制1000万）涉及到分布式事务的问题
3. 分区（oracle每个分区可以放500万条）跨分区查询
4. 物化视图，解决多表join的问题：把视图变成一个实体

正常视图，还是走多表join

这两个视图，查询效率就差的非常大了

1. 中间表的解决方案，通过存储过程，或者程序写入到这个中间表，直接查这一张表，不会走数据量很大的基表
2. 设计的方案
3. 对于结构优化的设计
4. 建立索引

规则索引（前几位，中间几位，后几位）

Id 日期 类型

这个表数据量很大，除了要分区以外

UUID不能满足，id的前几位要什么样，中间几位，后几位要怎么样

预留字段，给后期扩展业务的表做外键

业务规则的索引：以身份证号举例

索引加函数会失效，所以规定规则的索引就是数值加上uuid

1. 建立规则索引
2. 建立普通索引
3. 建立复合索引

如果不建规则索引，这个查询人信息的表就要join区域表

比如说身份证的前几位是区域，中间是年月日。我想查河北省唐山市1992年6月出生的人那我的sql语句就可以写成：

select \* from data\_check where check\_no >= '13020519920600000000' and check\_no < '13020519920700000000'

通过**中间表**完成会员升级

LV1一张表，LV2一张表；一个用户刚开始是LV1，然后升级到LV2，这个时候在中间表中存放这个用户的记录；然后查询用户等级时一看这条记录有更新，就先从中间表里面读取这条记录；然后LV2定时的从中间表里更新数据；更新完成后再从LV1表中删除这条记录

不同语言系统之间数据交换

Java系统和ruby的mvc系统框架

**中间放一个数据库**（worker），ruby系统产生的数据都写入这个数据库。历史数据和应用数据

Java系统可以定时的job去拉取数据；也可以把公共的数据放到这个中间的数据库中

1. 数据规则（添加必要的扩展字段）

Create by create time update by update time

比如说做统计分析，数据有一条记录有更新，可以根据update time重新计算这一条记录对应的分析数据，更新数据

1. 预留字段（用于关联其他业务的，比如说留一个外键给扩展的表）
2. 做一些合理的冗余

比如说电商，供应商系统一个表，商品系统一张表，两个数据库

供应商表可以做一些合理的冗余比如说添加商品系统表中的一些字段，这样就避免两个数据库之间做join；单独连一个数据源就可以获取供应商和商品信息