# 为什么要分库分表

## 数据库分性能瓶颈

连接数

处理能力

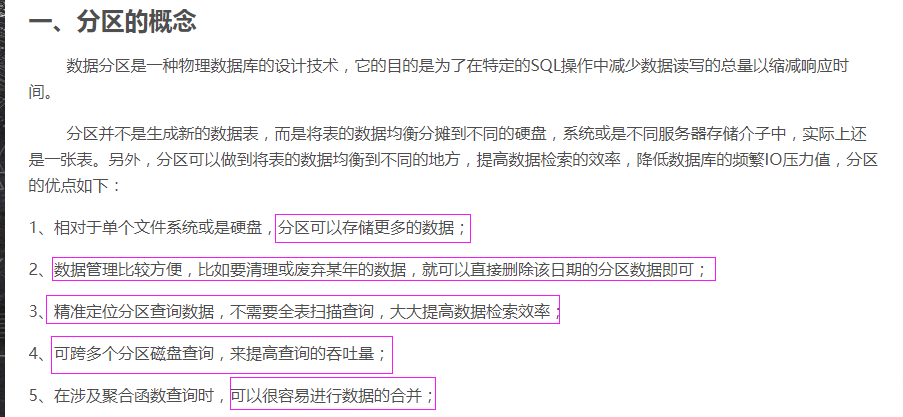
存储容量

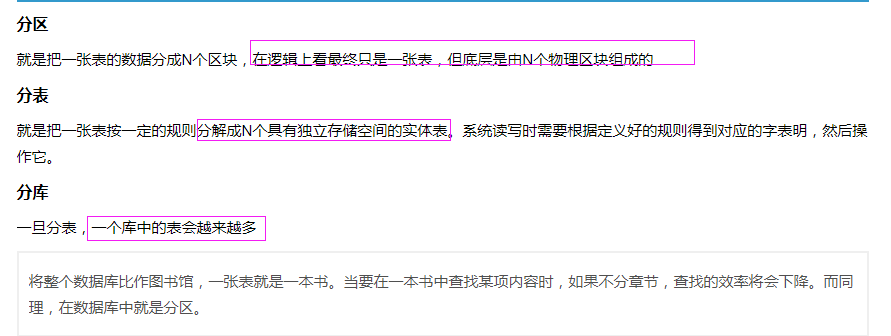
## 数据库优化方案

### sql优化（尽量走索引）

### 分区

分区（就是把一张表的数据分成N个区块，在逻辑上看最终只是一张表，但底层是由N个物理区块组成的）





### 主从复制，读写分离

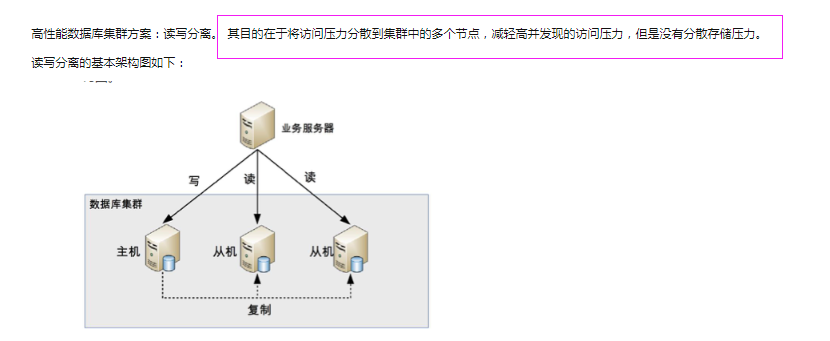
* 数据热备
* 读写分离（提高并发）

主从复制是异步的（通过数据冗余去实现高可用。复制中没有任何共享。每台机器都是独立且完整的系统。

### 缓存

缓存数据库查询结果，加快访问速度，缓解数据库压力

### 集群



### 主从复制和集群的区别

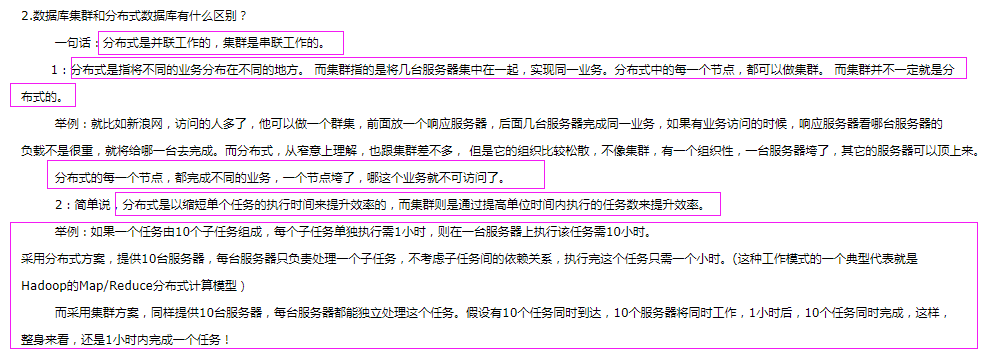
主从复制和集群的区别：集群是同步的（集群通过负载提高性能），主从复制是异步的（通过数据冗余去实现高可用）集群是共享存储的。复制中没有任何共享。每台机器都是独立且完整的系统。

向上扩展（扩展硬件）

分库分表

大数据

## 数据库集群和分布式数据库有什么区别



## 什么时候考虑切分

能不切分尽量不要切分

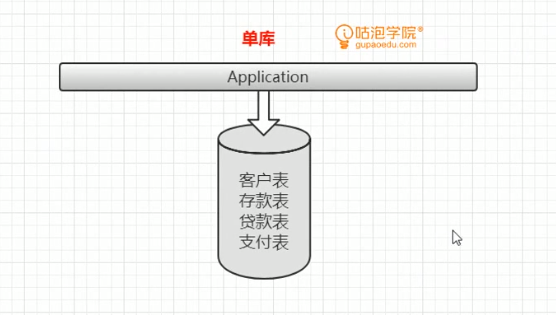
数据量过大（500w~1000w可以接受） 数据量不要超过2G

数据增长过快（阿里巴巴规约）

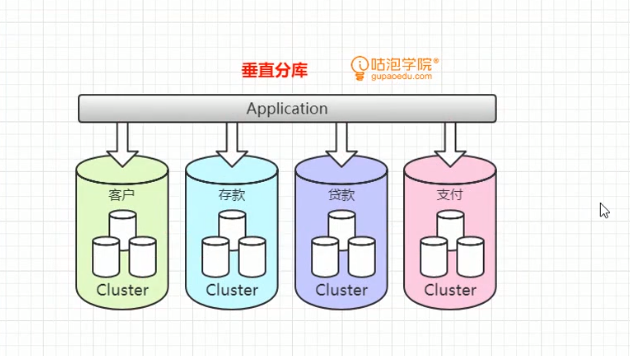
需要对某些字段进行切分

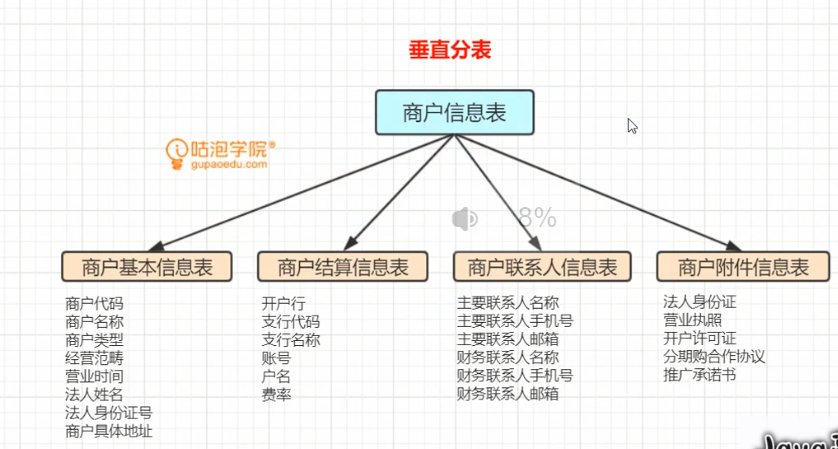
出于安全性和可用性的考虑（为什么做了切分就安全了（切分之后，可以为每个地区创建一个用户来控制安全性））

# 数据库切分类型



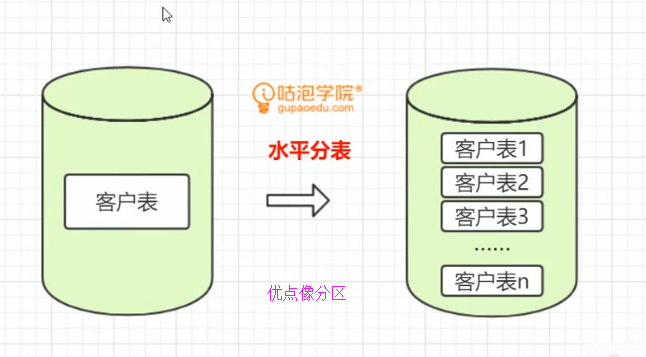
## 垂直

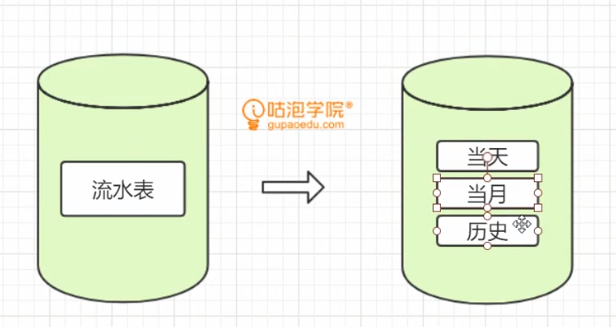


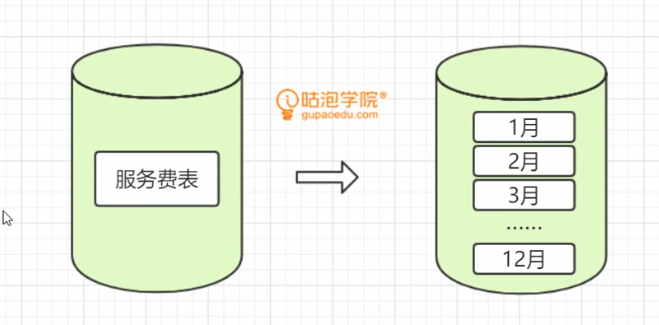


常用的字段放到一个表中去，不常用的放到令一个表中

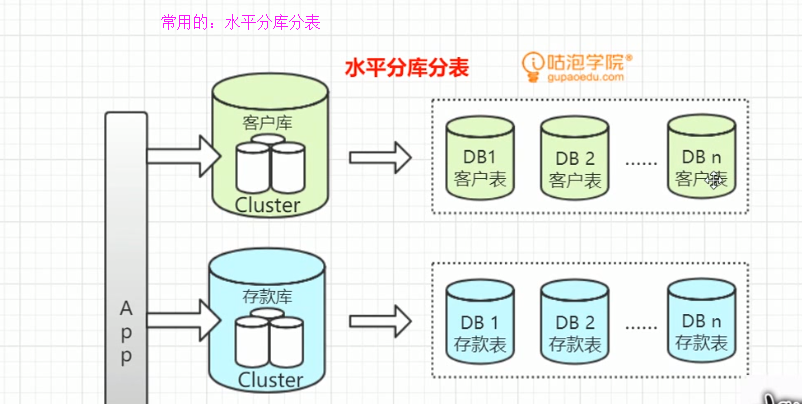
## 水平







## 推荐



# 数据切分以后的问题和解决方案

事物一致性问题，XA规范

跨结点的关联查询的问题join(全局表，字段冗余，er分片（订单表，订单明细表，这两个表落到一个结点上去），数据组装)。

跨结点排序，分页，函数，做结果汇总做二次计算

全局主键的问题：uuid（就算比较快，无序），开源的全局id生成系统，雪花算法

数据迁移和扩容的问题

# Mycat数据库分库分表中间件

