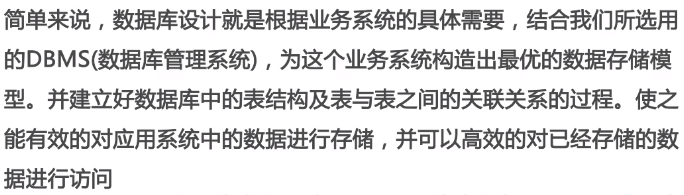
# 需求分析

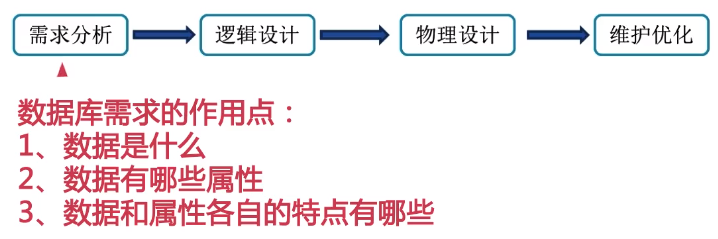
## 概述

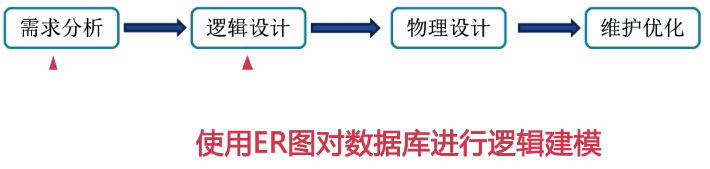
 什么是数据库设计？

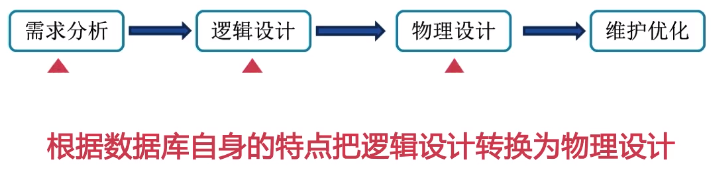
为什么需要数据库设计？

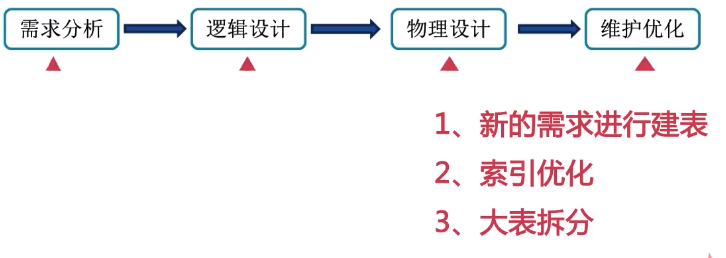


## 步骤



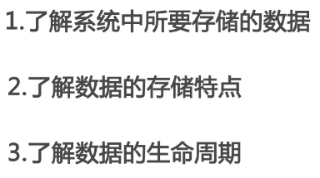






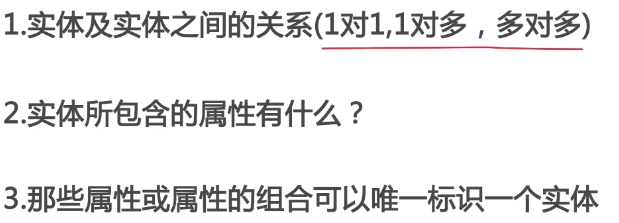
## 重要性

为什么需要需求分析？



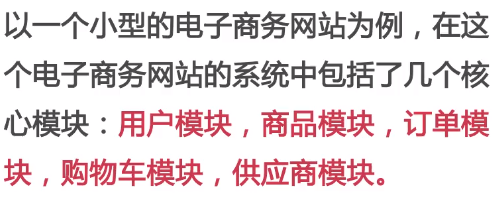
注：了解需要存储的数据类型，比如有的数据是有时效性的，可以采用过期清理和归档的方式处理；还有一些数据增长很快，数据量很大，但是不是核心数据，可以采用分库分表的方式存储；日志一般不适合存储在数据库中，如果要存储在数据库中需要先设计好归档方案。

需要搞清楚这样的一些关系：

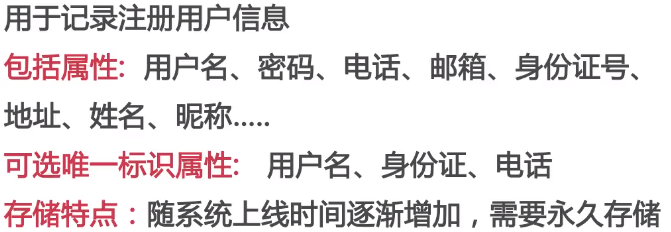


注：了解实体的特性和增长量，对于增长快的可以采用归档和分库分表。

## 实例

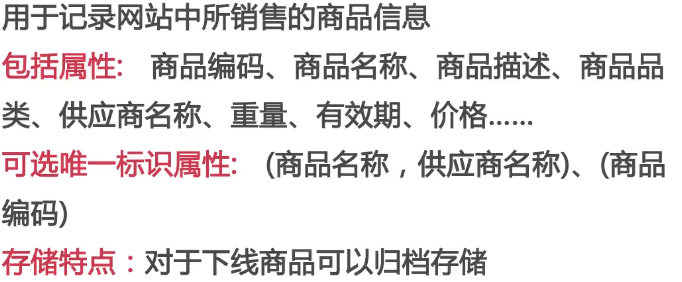


用户模块



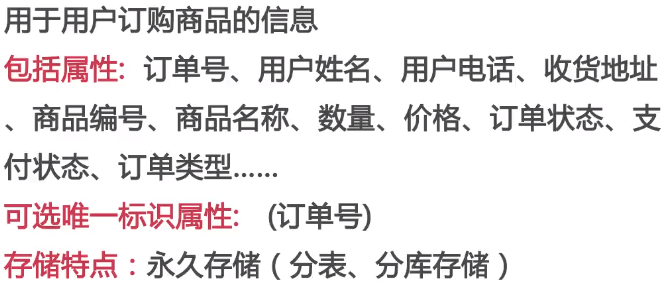
注：对于用户是不能做数据删除和归档的，因此需要永久保存。

商品模块



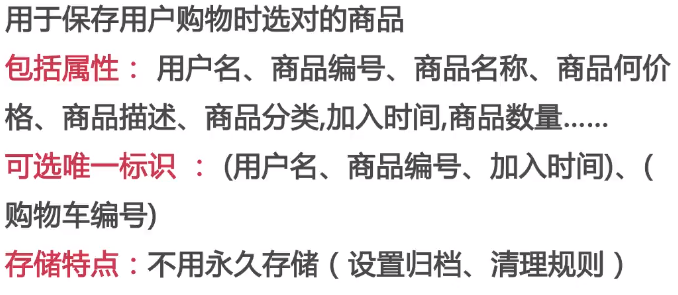
注：商品是越来越多的，因此不可以删除，但是有些商品可以下线，可以做归档（但是不是删除），需要永久存储的。

订单模块

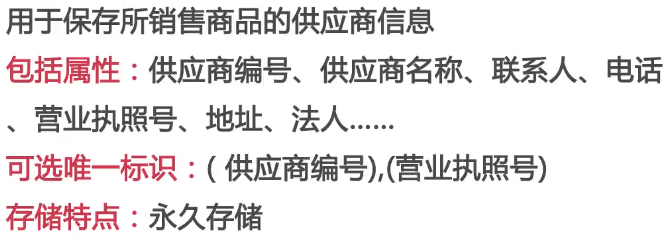


注：订单数量大，需要永久存储。

购物车模块



供应商模块

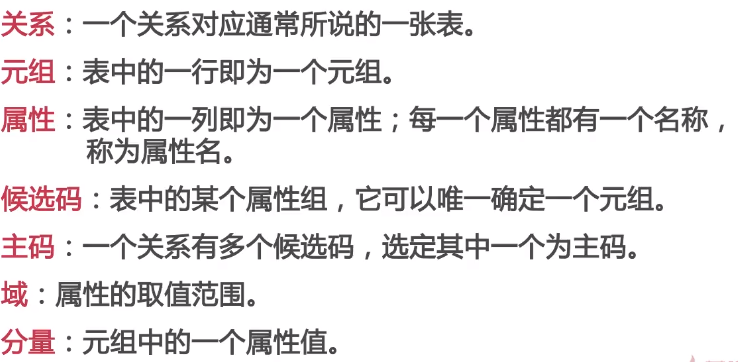


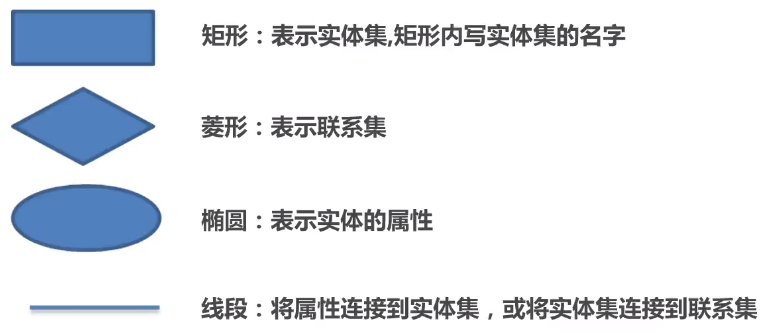
模块间关系

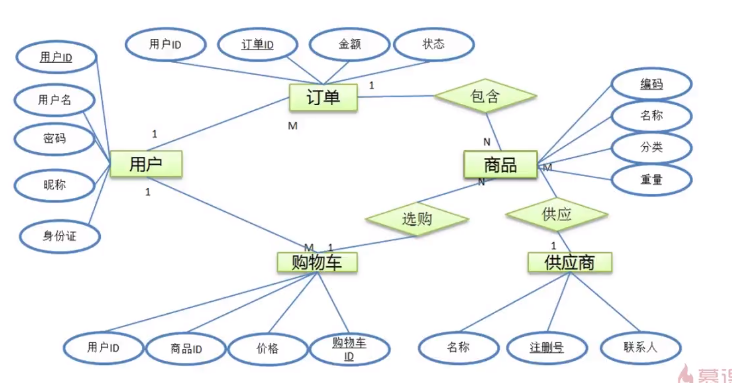


# 逻辑设计

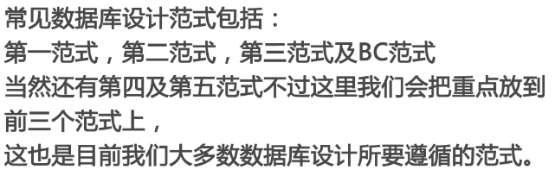
## ER图

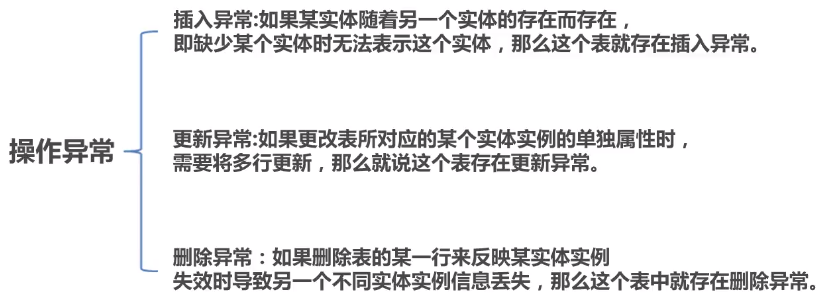


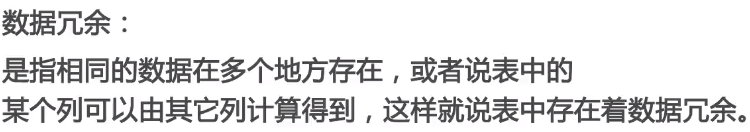




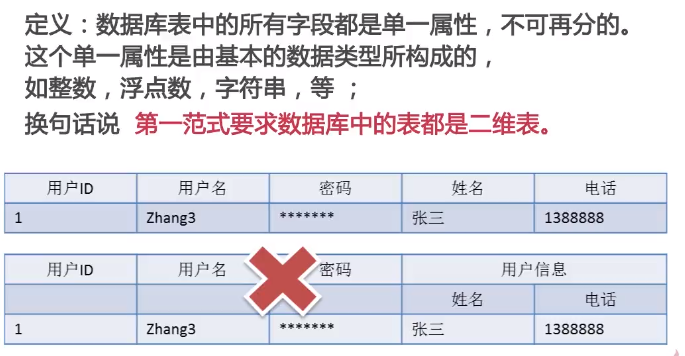
## 设计范式



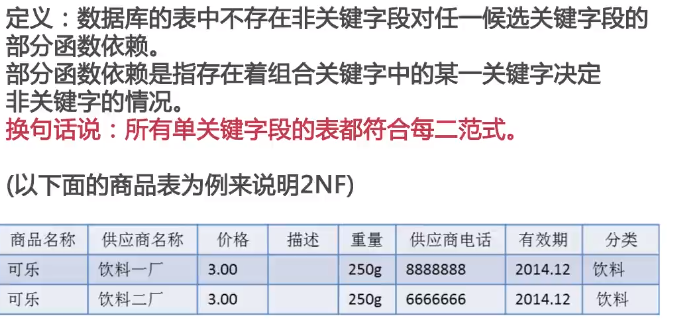


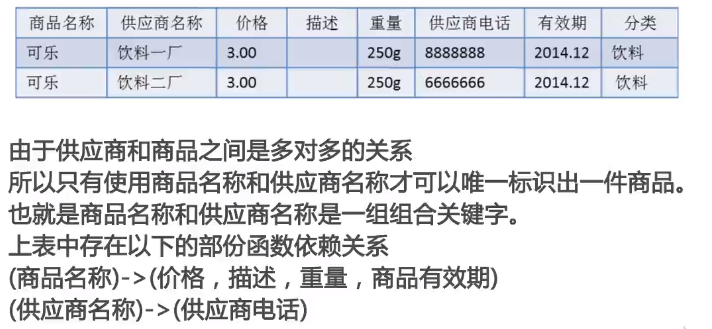


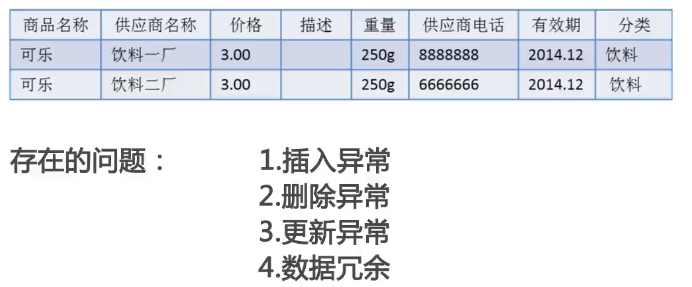
## 第一范式(1NF)



## 第二范式(2NF)

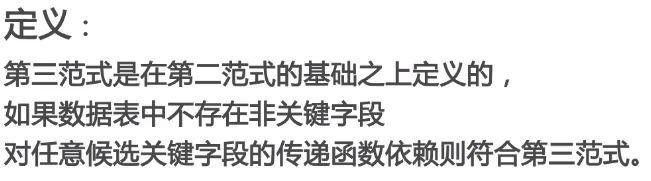


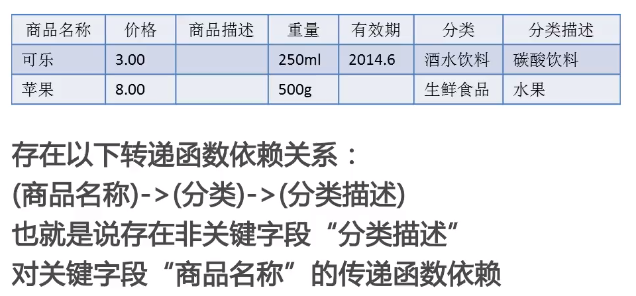


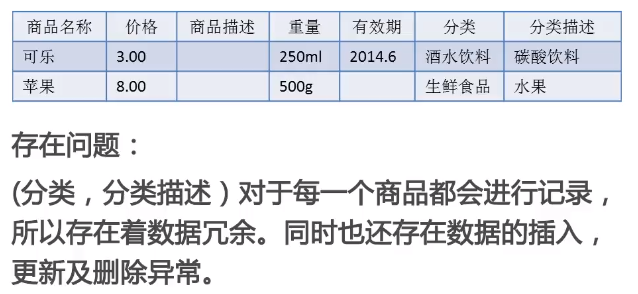


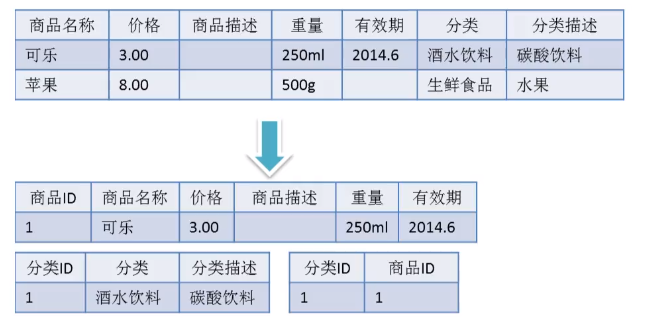


## 第三范式(3NF)



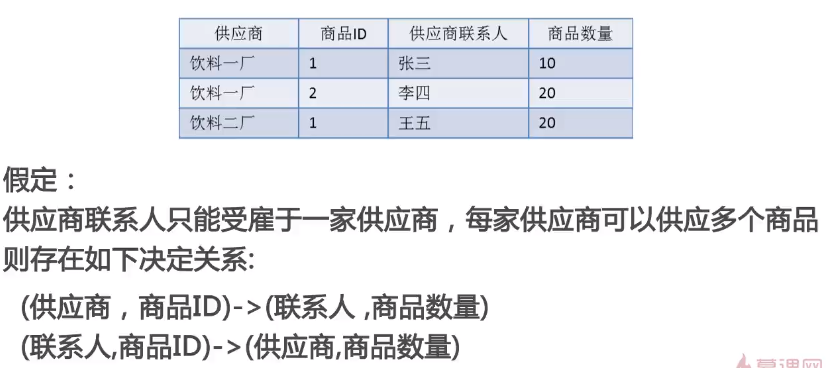


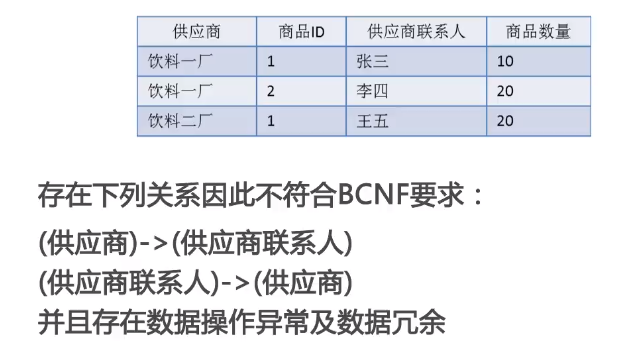


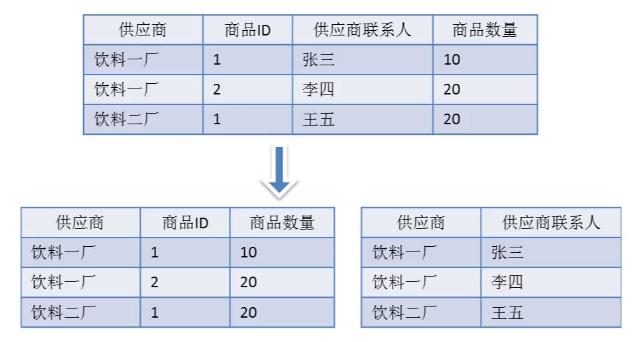


## BC范式





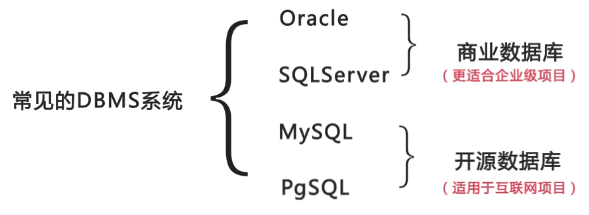




# 物理设计

## 概述

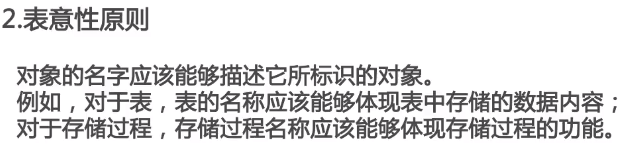
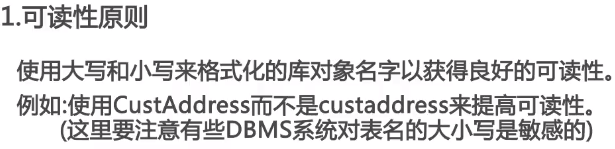
## 选型



## MySQL存储引擎



## 表/字段命名规则





## 字段类型选择

## 注意事项

## 反范式化表设计

# 维护优化

## 概述

## 数据字典

## 维护索引

## 适合的操作

## 垂直和水平拆分