第7章 数据库设计

金培权（jpq@ustc.edu.cn）

3、关系数据库模式的规范化

- 确定范式级别
- 实施规范化处理

（1）确定范式级别

- 范式级别的确定
  - 根据数据依赖确定已有的范式级别
    - 根据需求写出数据库模式中存在的所有函数依赖
    - 消除冗余数据依赖，求出最小的依赖集
    - 确定范式级别

（1）确定范式级别

- 范式级别的确定
  - 根据实际应用的需要（处理需求）确定要达到的范式级别
    - 时间效率和模式设计问题之间的权衡
      - 范式越高，模式设计问题越少，但连接运算越多，查询效率越低
      - 如果应用对数据只是查询，没有更新操作，则非BCNF范式也不会带来实际影响
      - 如果应用对数据更新操作较频繁，则要考虑高一级范式以避免数据不一致
    - 实际应用中一般以3NF为最高范式

（2）规范化处理

- 确定了初始数据模式的范式，以及应用要达到的范式级别后
- 按照规范化处理过程，分解模式，达到目标范式
  - “模式设计”部分的内容

4、模式评价

- 检查规范化后的数据库模式是否完全满足用户需求，并确定要修正的部分
  - 功能评价：检查数据库模式是否支持用户所有的功能要求
    - 必须包含用户要存取的所有属性
    - 如果某个功能涉及多个模式，要保证无损连接性
  - 性能评价：检查查询响应时间是否满足规定的需求。
    - 由于模式分解导致连接代价
    - 如果不满足，要重新考虑模式分解的适当性
    - 可采用模拟的方法评价性能

5、模式修正

- 根据模式评价的结果，对已规范化的数据库模式进行修改
  - 若功能不满足，则要增加关系模式或属性
  - 若性能不满足，则要考虑
    - 逆规范化
    - 分库分表
    - 使用存储过程
    - 缓存加速
    - ……

常用优化策略

- 逆规范化
- 分库分表
- 使用存储过程
- 缓存加速

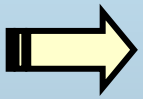
1、逆规范化

- 逆规范化
  - 模式合并
  - 增加冗余属性

分行号	名称	电话	存款总额	贷款总额
001	北京分行	6360111	123456	54321
002	上海分行	6360111	22344	112334
003	南京分行	6360111	4564	2123
004	重庆分行	6360222	1236788	88723
005	天津分行	6360222	345612	2312
006	广州分行	6360333	5788622	675323

2、分库分表

学号	……	所在系
01		1
02		2
03		6
……		……



学号	……	所在系
01		1
12		1
13		1
……		……

学号	……	所在系
02		2
18		2
45		2
……		……

……

2、分库分表

- 冷热数据划分
  - 80/20原则：一个关系经常被使用的数据只占20%
  - 将20%热数据单独划分为一个模式，使得大部分的查询都可以在较小规模的数据集上执行
  - 减少了应用处理的数据量，提高效率

2、分库分表

- 把关系模式按属性集垂直分解为多个模式
- 在实际中，应用可能经常存取的只是关系的某几个列，可考虑将这些经常访问的列单独拿出组成一个关系模式
- 若一个关系中，某几个属性的值重复较多，并且值较大，可考虑将这些属性单独组成关系模式，以降低存储空间

2、分库分表

50万种商品

商品号	……	商品说明书（4KB）

商品表约减少了
2GB数据量

商品号	……	说明编号（2B）

说明编号	商品说明书

3、使用存储过程

从A账号转帐

100到B账号

Total 1

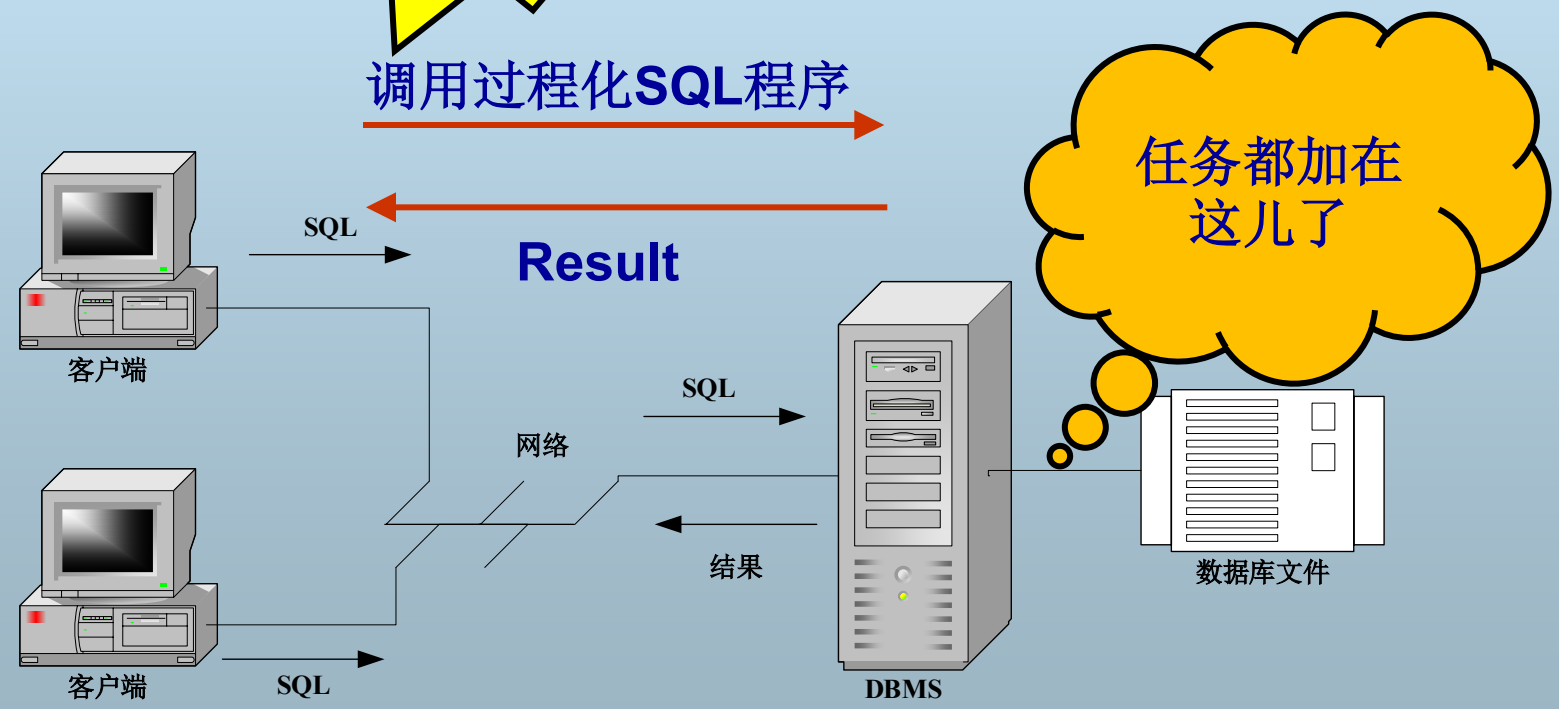
客户端更少计算

更少网络传输

客户机

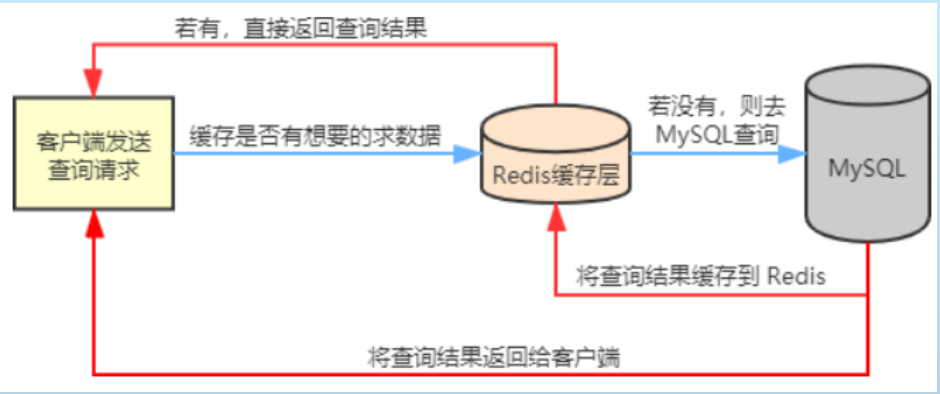
执行一个预先编写好并存储在服务器上的存储过程完成转帐

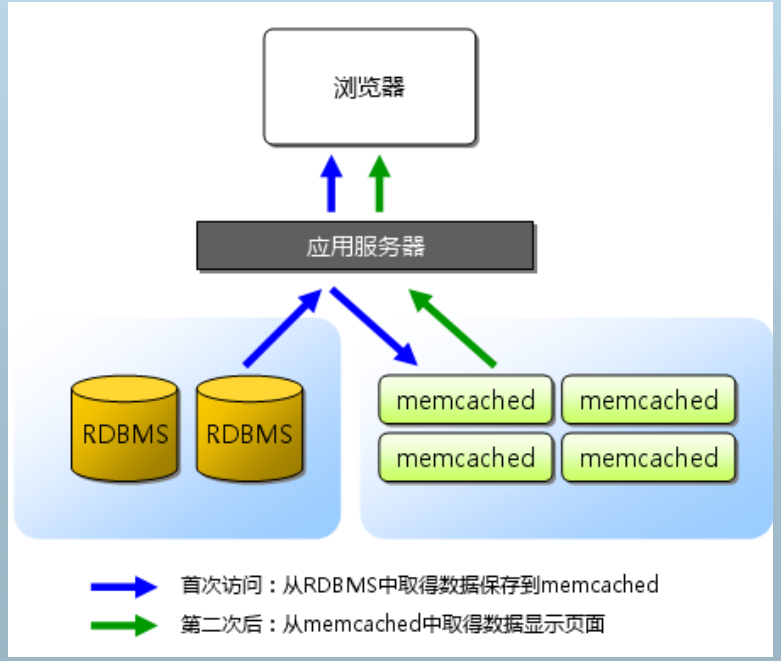
调用过程化SQL程序

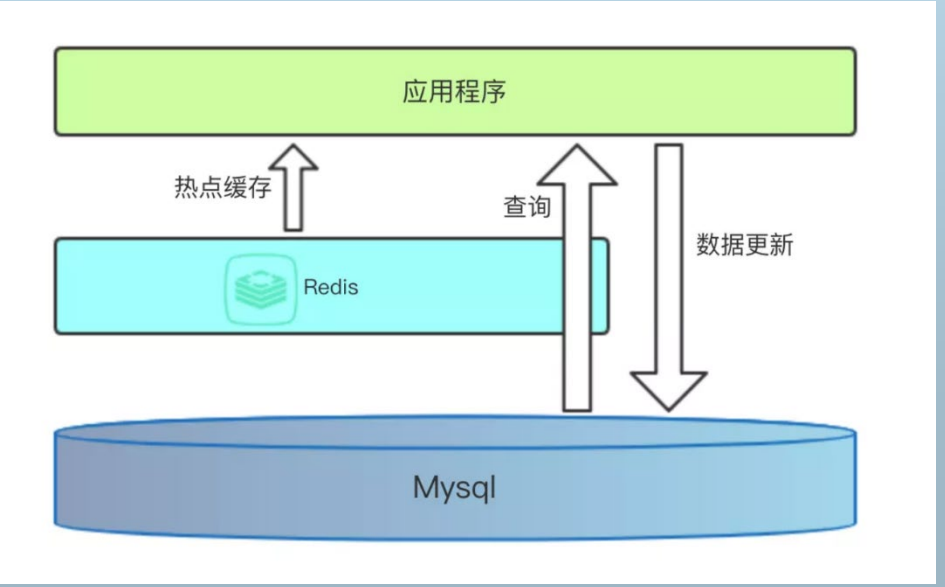


4、缓存加速

- 使用Key-Value数据库构建缓存层进行查询加速
  - Redis
  - Memcached

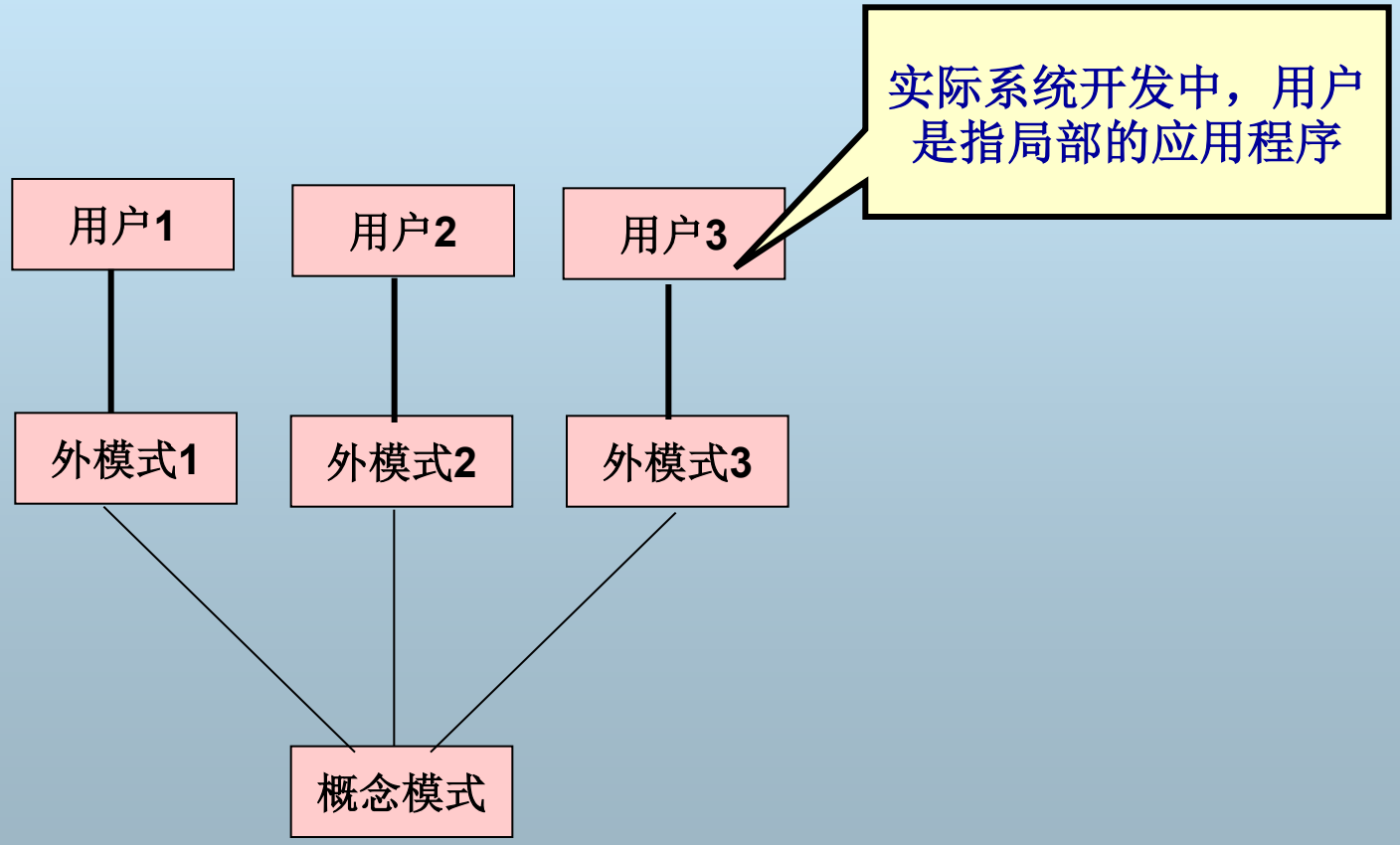






6、设计用户子模式（视图）

- 根据局部应用的需求，设计用户子模式



（1）设计用户子模式（视图）的考虑

- 使用更符合用户习惯的别名
  - ER图集成时要消除命名冲突以保证关系和属性名的唯一，在子模式设计时可以重新定义这些名称，以符合用户习惯
- 给不同级别的用户定义不同的子模式，以保证系统安全性
  - 产品(产品号，产品名，规格，单价，产品成本，产品合格率)
    - 为一般顾客建立子模式：产品1(产品号，产品名，规格，单价）
    - 为销售部门建立：产品2(产品号，名称，规格，单价，成本，合格率)
- 简化用户程序对系统的使用
  - 可将某些复杂查询设计为子模式以方便使用

六、数据库物理设计

- 设计数据库的物理结构
  - 为关系模式选择存取方法
  - 设计数据库的存储结构
- 物理设计的考虑
  - 查询时间效率
  - 存储空间
  - 维护代价
- 物理设计依赖于给定的计算机系统

1、选择存取方法

- 存取方法：数据的存取路径
  - 例如图书查询
- 存取方法的选择目的是加快数据存取的速度
  - 索引存取方法
  - 聚簇存取方法
  - 散列存取方法
- 索引设计
  - 哪些表上需要索引？
  - 哪些字段上需要索引？
  - 某个字段上的索引具体选择哪一种？

可以使用什么样的存取方法依赖于具体的DBMS

2、设计数据库的存储结构

- 确定数据的存放位置
  - 针对应用环境和DBMS特性，合理安排数据存储位置
    - 表和索引可考虑放在不同的磁盘上，使查询时可以并行读取
    - 日志文件和备份文件由于数据量大，而且只有恢复时使用，可放到磁带上
- 确定系统配置
  - 系统初始参数不一定适合应用
    - 并发用户数、同时打开的数据库对象数、缓冲区分配参数、物理块的大小等

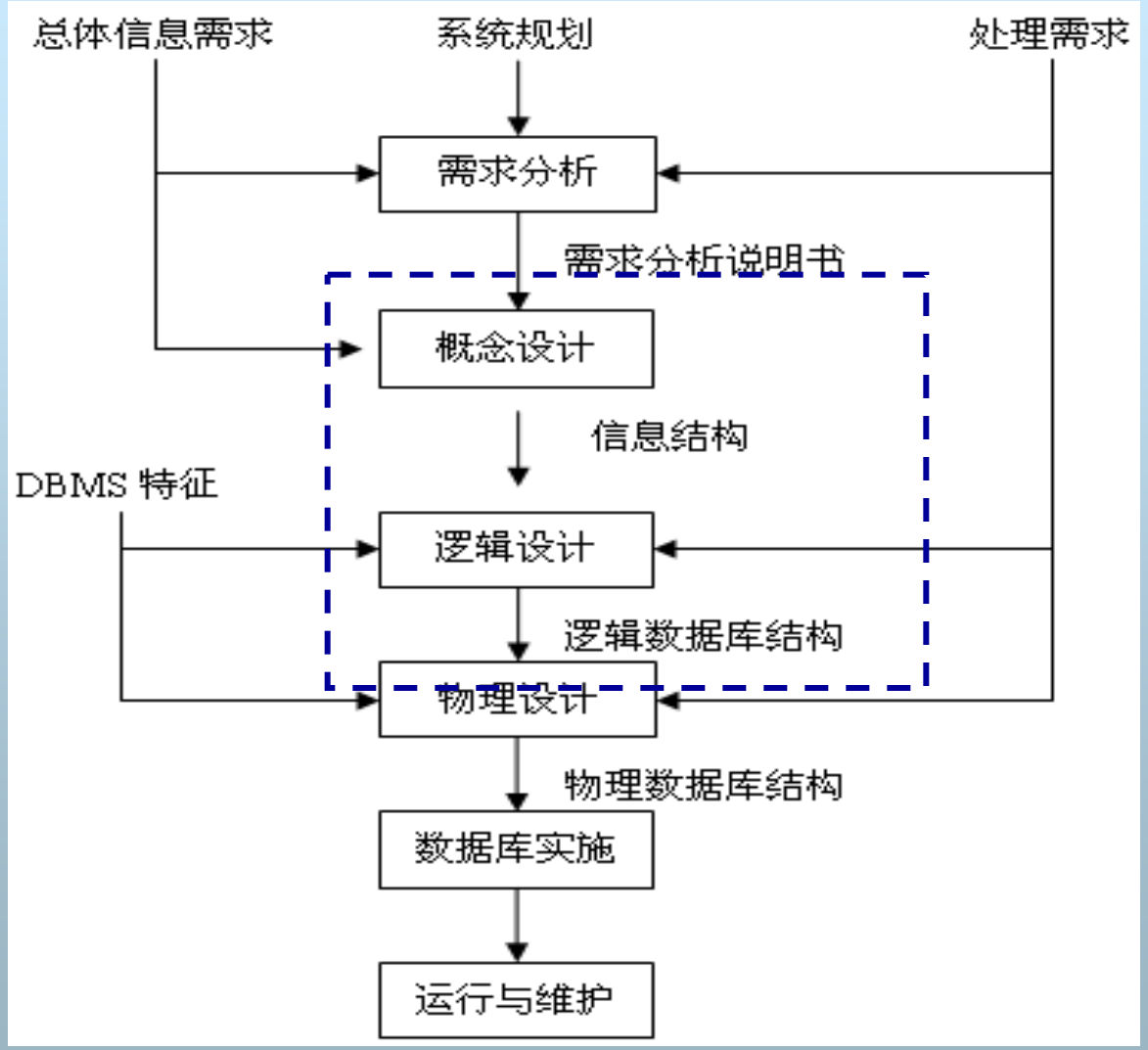
七、数据库实施

- 建立实际的数据库结构
  - CREATE TABLE
  - CREATE INDEX
  - ……
- 初始数据装入
- 安全性设计和故障恢复设计
- 应用程序的编码和调试

八、运行和维护

- 试运行
  - 根据初始数据对数据库系统进行联调
  - 执行测试：功能、性能
- 维护
  - 数据备份和恢复
  - 数据库安全性控制和完整性控制
  - 数据库性能的分析和改造
  - 数据库的重组织

本章小结


总体信息需求
系统规划
处理需求
需求分析
需求分析说明书
概念设计
信息结构
DBMS 特征
逻辑设计
逻辑数据库结构
物理设计
物理数据库结构
数据库实施
运行与维护