

# 9 高性能数据库开发原则

---为性能而设计

南京大学 软件学院



# 数据的关系视图

- 数据库只是对现实世界的有限描述。对特定的业务活动的描述不止一种
- “关系模型” 中的 “关系” 的含义
- 关系模型的一致性
  - 只要遵守关系理论，可以保证基于数据库的任何查询结果与原始数据具有同样的有效性
  - 关系理论包括
    - 关系不包含重复数据
    - 关系理论保证无限数据的正确性
    - 记录之间没有顺序



# 规范化 (Normalization)

- 表结构没有规范化会面临很大的风险吗?
  - 1NF 确保原子性 (Atomicity)
  - 2NF 检查对键的完全依赖
  - 3NF 检查属性的独立性
- 过分精益求精会使精力分散
- 规范化的价值
  - 合理规范化的模型可应对需求变更
  - 规范化数据重复降至最少



# 有值、无值、空值

- 表中的每一条记录都应该是特定“事物”的状态描述，如果大部分特征信息都显示“我们不知道”，无疑大大降低了信息可信性
- 存在空值意味着关系模型存在严重的问题，动摇了查询优化的基础
- 空值对程序逻辑是危险的
  - 必须使用空值的话，一定要清楚它在特定情况下的影响。



# 限用Boolean型字段

- SQL中并不存在Boolean类型
- 实现flag表示标志位的Y/N或T/F
  - 例如: order\_completed
  - 但是...往往增加信息字段能包含更多的信息量
  - 例如: completion\_date completion\_by
  - 或者增加order更多状态标示
- 极端的例子: 四个属性取值都是T/F, 可以用0-15这16个数值代表四个属性所有组合状态
  - 技巧可能违反了原子性的原则
  - 为数据而数据, 是通向灾难之路



# 理解子类型 (SubType)

- 表过“宽”（有太多属性）的另一个原因，是对数据项之间的关系了解不够深入
- 一般情况下，给子类型表指定完全独立于父表主键的主键，是极其错误的



# End

下一讲再见

