

6.三大范式

——数据库设计规范

CONTENTS

- 引入三大范式
- 第一范式
- 第二范式
- 第三范式
- 范式的优缺点
- 总结

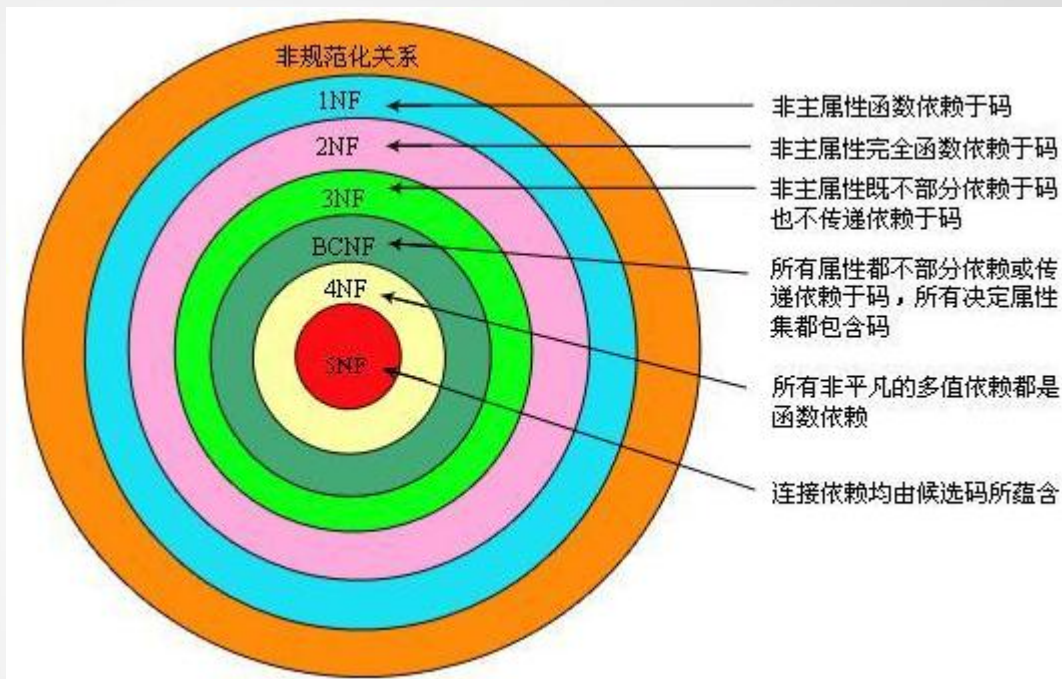


引入三大范式

- 必须保证数据库设计的合理性
 - 数据库设计关系整个系统的架构，关系到后续的开发效率和运行效率
 - 数据库的设计主要包含了设计表结构和表之间的联系
- 如何是合理数据库
 - 结构合理
 - 冗余较小
 - 尽量避免插入删除修改异常
- 如何才能保证数据库设计水平
 - 遵循一定的规则
 - 在关系型数据库中这种规则就称为范式
- 什么是范式 (NF= NormalForm)
 - 范式是符合某一种设计要求的总结。
 - 要想设计一个结构合理的关系型数据库，必须满足一定的范式。

• 范式分类

- 第一范式
- 第二范式
- 第三范式
 - Boyce Codd范式=NCNF
 - 由Boyce和Codd提出的,
 - 比3NF又进了一步
 - 通常认为是修正的第三范式.
- 第四范式
- 第五范式
- 各个范式是依次嵌套包含的
- 范式越高, 设计质量越高, 在现实设计中也越难实现
- 一般数据库设计, 只要达到第三范式, 即可避免异常的出现





第一范式

- 要求
 - 最基本的范式
 - 数据库表每一列都是不可分割的基本数据项，同一列中不能有多个值
 - 简单说就是要确保每列保持原子性
 - 第一范式的合理遵循需要根据系统的实际需求来定
- 示例
 - 用户表（用户名，家庭地址）
 - 用户表（用户名，省，城市，详细地址）
 - 系（系名称，系主任，系高级职称人数）
 - 系（系名称，系主任，系教授人数，系副教授人数）



第二范式

- 要求
 - 第二范式需要确保数据库表中的每一列都和主键相关，而不能只与主键的某一部分相关（主要针对联合主键而言）。
 - 即在一个数据库表中只能保存一种数据，不可以把多种数据保存在同一张数据库表中。
- 示例
 - 学号和课程编号作为联合主键
 - 课程名称只依赖于课程编号，而和学号没有关系

学号	学生姓名	课程编号	课程名称
023145	张三	988010	Java
023146	李四	988010	Java
023145	张三	988011	Oracle
023147	王五	988011	Oracle
023258	赵六	988010	Java



第二范式

- 解决
 - 提取出学生表
 - 提取成课程表
 - 提取选课表，存放选课记录

学号(主键)	学生姓名
023145	张三
023146	李四
023147	王五
023258	赵六

课程编号(主键)	课程名称
988010	Java
988011	Oracle

学号(主键)	教师编号(主键)
023145	988010
023146	988010
023145	988011
023147	988011
023258	988010



第三范式

- 要求
 - 确保数据表中的每一列数据都和主键直接相关，而不能间接相关
 - 属性不依赖于其他非主属性。

- 示例1: 学生班级表

学号(主键)	学生姓名	班级编号	班级名称	班级信息
023145	张三	987654	3班	特招班
023146	李四	987231	3班	特招班
023147	王五	987654	4班	普通班
023258	赵六	987654	3班	特招班

学号(主键)	学生姓名	班级编号
023145	张三	987654
023146	李四	987231
023147	王五	987654
023258	赵六	987654

班级编号(主键)	班级名称	班级信息
987654	3班	特招班
987655	4班	普通班



第三范式

- 示例2

- 订单明细表

编号(主键)	图书id	图书名称	价格	作者	出版社	出版日期	数量
023145	1	精通Java	6000	张三	清华大学出版社	2007	1
023146	2	Oracle宝典	6500	李四	机械出版社	2009	1
023147	3	JSP	87	王五	电子出版社	2014	3
023258	4	Struts2	56	赵六	清华大学出版社	2005	2

图书id	图书名称	价格	作者	出版社	出版日期
1	精通Java	6000	张三	清华大学出版社	2007
2	Oracle宝典	6500	李四	机械出版社	2009
3	JSP	87	王五	电子出版社	2014
4	Struts2	56	赵六	清华大学出版社	2005

编号(主键)	图书id	数量
023145	1	1
023146	2	1
023147	3	3
023258	4	2



范式的优缺点

- 优点
 - 结构合理
 - 冗余较小
 - 尽量避免插入删除修改异常
- 缺点
 - 性能降低
 - 多表查询比单表查询速度慢
- 数据库的设计应该根据当前情况和需求做出灵活的处理。
 - 在实际设计中，要整体遵循范式理论。
 - 如果在某些特定的情况下还死死遵循范式也是不可取的，因为可能降低数据库的效率，此时可以适当增加冗余而提高性能。



范式的优缺点

- 示例:

- 比如经常购物车条目的中除了条目编号, 商品编号, 商品数量外, 可以增加经常使用的商品名称, 商品价格等

- 商品表

图书id	图书名称	价格	作者	出版社	出版日期
1	精通Java	6000	张三	清华大学出版社	2007
2	Oracle宝典	6500	李四	机械出版社	2009
3	JSP	87	王五	电子出版社	2014
4	Struts2	56	赵六	清华大学出版社	2005

- 订单明细表

编号(主键)	图书id	图书名称	价格	数量
023145	1	精通Java	6000	1
023146	2	Oracle宝典	6500	1
023147	3	JSP	87	3
023258	4	Struts2	56	2



其他

- 数据库表之间的关系
 - 一对一：学生和学生证
 - 一对多：学生和班级
 - 多对多：学生和课程
- 如何表示数据库表之间的关系
 - 使用外键
- 数据库外键关系表示的其实是一种一对多关系
 - 一对一：外键+唯一
 - 多对多：引入中间表，把一个多对多表示为两个一对多



总结

- 范式是指导数据设计的规范化理论，可以保证数据库设计质量
- 第一范式：字段不能再分
- 第二范式：不存在局部依赖
- 第三范式：不含传递依赖（间接依赖）
- 使用范式可以减少冗余，但是会降低性能
- 特定表的的设计可以违反第三范式，增加冗余提高性能