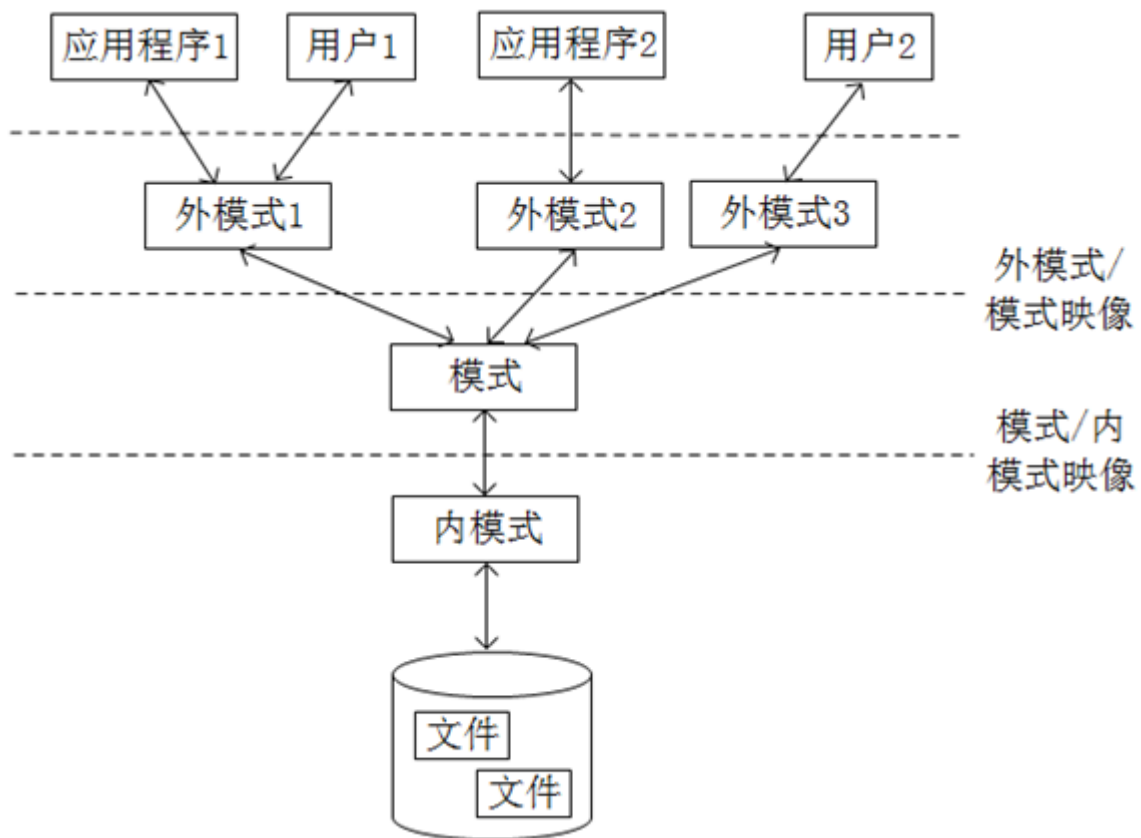


要点	内容
三级模式和两级映像	三级模式、两级映像、数据独立性
关系模型	关系数据结构、完整性约束、关系操作
函数依赖	完全依赖、部分依赖、传递依赖
范式	1NF、2NF、3NF、BCNF

三级模式和两级映像



1.三级模型

1.1 模式

模式也称为概念模式或逻辑模式。它是数据库中全体数据的逻辑结构和特征的描述。一个数据库只有一个模式，它处于数据库系统三级模式结构的中间，与数据的物理存储细节无关，也与具体的应用程序和用户无关。

1.2 外模式

外模式也称子模式或用户模式，它是用户所看到和使用的局部数据的逻辑结构和特征的描述。它由模式导出，是模式的一个子集。

1.3 内模式

内模式也称存储模式。它是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。一个数据库只有一个内模式，内模式由数据库管理系统提供的内模式描述语言定义。

2. 两级映像

2.1 外模式/模式映像

外模式/模式映像定义了外模式与模式之间的对应关系，也就是数据的局部逻辑结构与全局逻辑结构之间的关系。

2.2 模式/内模式映像

模式/内模式映像定义了模式与内模式之间的对应关系，也就是数据全局逻辑结构与存储结构之间的对应关系

3. 数据独立性

数据独立性是指应用程序与数据库的数据逻辑组织以及存储结构之间的相互独立。

3.1 逻辑数据独立性

逻辑数据独立性是指外模式不会随着模式的改变而改变。

3.2 物理数据独立性

物理数据独立性是指模式不会随着内模式的改变而改变。

关系模型

关系模型由关系数据结构、关系操作和关系完整性约束所组成

1 关系数据结构

- 关系(Relation)：一张规范化的二维表，如读者表

- 元组(Tuple): 表中的一行元组的集合构成关系
- 属性(Attribute): 表中的一列
- 域(Domain): 属性的取值范围
- 分量: 元组中的一个属性值
- 候选码(Candidate Key): 能够唯一标识一个元组的某个属性或属性组

主码(Primary Key): 也称为主键。若一个关系有多个候选码, 则选择其中一个为主码, 用于标识该关系中元组

主属性(Prime attribute): 包含在候选码中的属性

非主属性(Non-prime attribute): 不包含任何候选码中的属性

2.关系的完整性约束

2.1 实体完整性规则:

若属性(或属性组)A是基本关系R的主属性, 则A不能取空值。

2.2 参照完整性规则:

基本关系R的属性(或属性组)F是另一个关系S的主码, 但F不是R本身的主码, 则称F是基本关系R的外码或外键。基本关系R为参照关系, 基本关系S为被参照关系。对于R中的每个元组在F上的值要么取空值, 要么等于S中某个元组的主码值。

2.3 用户自定义的完整性规则:

不同的应用还要满足不同的约束, 这些约束反映了具体应用所涉及数据的特定语义要求。

3.关系操作

主要有查询和更新两大类

3.1 选择 (Selection)

选择又称为限制 (Restriction) 。用于选取某个关系中满足给定条件的元组。

$$\sigma_F(R) = \{t | t \in R \wedge F(t) = \text{'真'}\}$$

3.2 投影 (Projection)

投影运算用于从某个关系中选取若干属性列组成新的关系, 并重新安排列的顺序。

$$\Pi_A(R) = \{t[A] \mid t \in R\}$$

A为关系R中的属性组, $t[A]$ 表示元组t在属性组A上的分量。

3.3 连接 (Join)

连接(也称为 θ 连接)是从两个关系的笛卡尔积中选取属性间满足一定条件的元组。

$$R \bowtie S = \{trts \mid tr \in R \wedge ts \in S \wedge tr[A] \theta ts[B]\}$$

A和B为R和S上度数相等且可比的属性组, θ 为比较运算符。

3.4 等值连接

连接运算符是“=”的连接运算, 它是从关系R与S的广义笛卡尔积中选取A, B属性值相等的那些元组

3.5 自然连接

是一种特殊的等值连接, 它要求两个关系中进行比较的分量必须是相同的属性组, 并且在结果中把重复的属性列去掉。

3.6 外连接

如果在连接时将不匹配的元组也保存在关系中, 并将元组中那些没有匹配值的分量置为空值, 则这种连接称为外连接。如果只把左边关系中不匹配的元组保留就叫左外连接, 如果只把右边关系中不匹配的元组保留就叫右外连接。

4. 关系代数表达式优化

- (1) 尽可能早地进行选择运算, 以便减少中间结果的大小。
- (2) 尽可能早地进行投影运算, 以便减少中间结果的大小。
- (3) 将多个选择和投影运算同时执行, 以便在一次扫描关系时完成所有这些运算, 从而避免重复扫描关系。
- (4) 尽量避免直接做笛卡尔积运算, 而是将笛卡尔积前后的一连串选择和投影运算合并起来一起操作。

函数依赖和范式

规范化设计的好处: 避免导致**数据不一致、插入异常、删除异常、更新异常、数据冗余**等问题

1. 函数依赖

1.1 部分函数依赖

部分函数依赖: 设X, Y是关系R的两个属性集合, 存在 $X \rightarrow Y$, 若 X' 是X的真子集, 存在 $X' \rightarrow Y$, 则称Y部分函数依赖于X。

例子: 学生基本信息表R中(学号, 身份证号, 姓名), 学号属性取值是唯一的, 在R关系中, (学号, 身份证号) \rightarrow (姓名), (学号) \rightarrow (姓名), (身份证号) \rightarrow (姓名); 所以姓名部分函数依赖于(学号, 身份证号);

1.2 完全函数依赖

设 X, Y 是关系 R 的两个属性集合, X' 是 X 的真子集, 存在 $X \rightarrow Y$, 但对每一个 X' 都有 $X' \not\rightarrow Y$, 则称 Y 完全函数依赖于 X 。

例子: 学生基本信息表 R (学号, 班级, 姓名) 假设不同的班级间学号有相同的, 班级内学号不能相同, 在 R 关系中, (学号, 班级) \rightarrow (姓名), 但是 (学号) \rightarrow (姓名) 不成立, (班级) \rightarrow (姓名) 不成立, 所以姓名完全函数依赖于 (学号, 班级);

1.3 传递函数依赖

设 X, Y, Z 是关系 R 中互不相同的属性集合, 存在 $X \rightarrow Y (Y \not\rightarrow X), Y \rightarrow Z$, 则称 Z 传递函数依赖于 X 。

例子: 在关系 R (学号, 系名, 宿舍) 中, (学号) \rightarrow (系名), 系名 \neq 学号, (系名) \rightarrow (宿舍), 宿舍 \neq 系名, 所以符合传递函数的要求;

2. 范式

1NF

第一范式是指数据库的每一列都是不可分割的基本数据项, 强调列的原子性, 即不能存在某一属性有多值情况
比如下表中联系电话属性可以继续划分, 所以不满足第一范式。

说明: 在任何一个关系数据库中, 第一范式 (1NF) 是对关系模式的基本要求, 不满足第一范式 (1NF) 的数据库就不是关系数据库。

ID	姓名	年龄	联系电话	
			移动电话	固定电话

2NF

若关系模式 $R(U, F) \in 1NF$, 且 R 中的每个非主属性完全函数依赖于 R 的某个候选码, 则 R 满足第二范式。

第二范式要求数据表每一个实例或者行必须被唯一标识。除满足第一范式外还有两个条件, 一是表必须有一个主键; 二是没有包含在主键中的列必须完全依赖于主键, 而不能只依赖于主键的一部分。

说明: 当数据表中是联合主键, 但是有的列只依赖联合主键中的一个或一部分属性组成的联合主键, 此时需要拆表才能复合第二范式。

例子: 在关系 R (学号, 学生所在系, 学生宿舍, 课程号, 成绩) 中, 非主属性={学生所在系, 学生宿舍, 成绩}。

存在依赖 $F=\{(学号, 课程号) \twoheadrightarrow 成绩, 学号 \twoheadrightarrow 学生所在系, 学号 \twoheadrightarrow 宿舍, 学生所在系 \twoheadrightarrow 学生宿舍\}$ 。

可知存在非主属性对联合主键的部分依赖{(学号, 课程号) \twoheadrightarrow 学生所在系, (学号, 课程号) \twoheadrightarrow 学生宿舍}, 所以不满足第二范式。

3NF

若某一范式是第二范式, 且每一个非主属性都不传递依赖于该范式的候选键, 则称为第三范式, 即不能存在: 非主键列 A 依赖于非主键列 B , 非主键列 B 依赖于主键的情况。

例子: 关系 (学号, 学生所在系, 学生宿舍), 存在传递依赖: 学号 \twoheadrightarrow 学生所在系, 学生所在系 \twoheadrightarrow 学生宿舍。

即非主属性学生宿舍传递依赖主属性学号, 故不满足第三范式。

BCNF

在第三范式的基础上，数据库表中如果不存在任何属性对任一候选关键字段的传递函数依赖则符合第三范式。

- (1) 所有非主属性对每一个码都是完全函数依赖；
- (2) 所有的主属性对于每一个不包含它的码，也是完全函数依赖；
- (3) 没有任何属性完全函数依赖于非码的任意一个组合。

规范化过程

规范化就是对原关系进行投影，消除决定属性不是候选码的任何函数依赖。具体可以分为以下几步：

- (1) 对1NF关系进行投影，消除原关系中非主属性对码的部分函数依赖，将1NF关系转换成若干个2NF关系。
- (2) 对2NF关系进行投影，消除原关系中非主属性对码的传递函数依赖，将2NF关系转换成若干个3NF关系。
- (3) 对3NF关系进行投影，消除原关系中主属性对码的部分函数依赖和传递函数依赖，也就是说使决定因素都包含一个候选码，得到一组BCNF关系。
- (4) 对BCNF关系进行投影，消除原关系中属性间非平凡且非函数依赖的多值依赖，得到一组4NF关系。

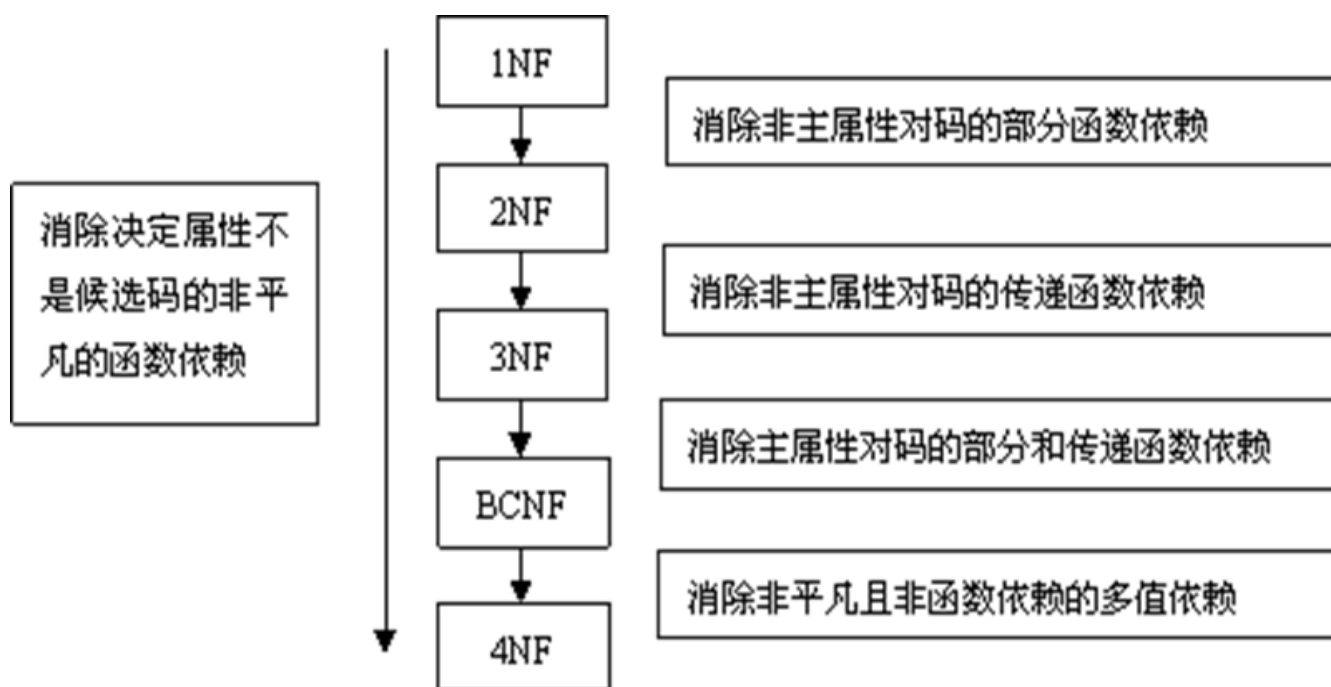


图 4-9 规范化过程