

关系数据库的设计理论

笔记本: database_theory

创建时间: 2021/6/29 21:18

更新时间: 2021/6/29 22:29

作者: 134exetj717

- 设计关系数据库的过程中, 我们会存在哪些异常呢?
 - 插入异常
 - 元组插不进去
 - 插入一个元组, 却要求插入多个元组
 - 删除异常
 - 删除时也删掉了其他信息
 - 删除一个元组却删除了多个元组
 - 冗余
 - 某种信息在关系中存储多次
 - 更新异常
 - 修改一个元组, 却要求 修改多个元组
- 为什么会产生这些异常现象呢?
 - 因为我们将多种数据集中于一个关系模式中, 使得关系模式中属性间存在这样或那样复杂的依赖关系
- 解决异常的方法是什么呢?
 - 利用关系数据库规范化理论, 对关系模式进行相应的分解, 使得每一个关系模式表达的概念单一, 属性间的数据依赖关系单纯话, 从而消除异常
- 接下来, 我们需要引进函数依赖的概念

函数依赖定义: 设R是一个关系模式, U是R的属性集合, X和Y是U的子集。对于R的任意实例r, r中任意两个元组 t_1 和 t_2 , 如果 $t_1[X]=t_2[X]$ 则 $t_1[Y]=t_2[Y]$, 那么称X**函数地决定**Y, 或Y**函数地依赖于**X, 记作: $X \rightarrow Y$, X称为**决定子(Determinant)**/决定属性集。

平凡函数依赖：如果 $Y \subseteq X$ ，则 $X \rightarrow Y$ 称为平凡函数依赖。平凡函数依赖不反映新的语义。

非平凡函数依赖：如果 $X \rightarrow Y$ ，且 Y 不是 X 的子集，则称 $X \rightarrow Y$ 是非平凡函数依赖。如不特别声明，一般总是讨论非平凡函数依赖。

决定属性集/决定子：如果 $X \rightarrow Y$ ，则称 X 为该函数依赖的决定属性集。

XY等价：如 $X \rightarrow Y$ ，且 $Y \rightarrow X$ ，则 X 与 Y 一一对应，记作 $X \leftrightarrow Y$ 。

完全函数依赖：设 R 是一个具有属性集合 U 的关系模式，如果 $X \rightarrow Y$ ，并且对于 X 的任何一个真子集 Z ， $Z \rightarrow Y$ 都不成立，则称 Y 完全函数依赖于 X ，记作： $X \twoheadrightarrow Y$ 。

$X=(a, b), Y=(c)$ ，如果 $a \not\rightarrow c, b \not\rightarrow c$ ，但 $(a, b) \rightarrow c$ ，则 $X \twoheadrightarrow Y$ 。

部分函数依赖：若 $X \rightarrow Y$ ，但 Y 不完全函数依赖于 X ，则称 Y 部分函数依赖于 X ，记作： $X \rightharpoonup Y$ 。

$X=(a, b), Y=(c)$ ，如果 $a \rightarrow c$ 或 $b \rightarrow c$ ，则 $X \rightharpoonup Y$ 。

传递函数依赖：设 R 是一个具有属性集合 U 的关系模式， $X, Y, Z \subseteq U$ ， X, Y, Z 是不同的属性集。如果 $X \rightarrow Y$ ， $Y \rightarrow X$ 不成立， $Y \rightarrow Z$ ，则称 Z 传递地函数依赖于 X 。

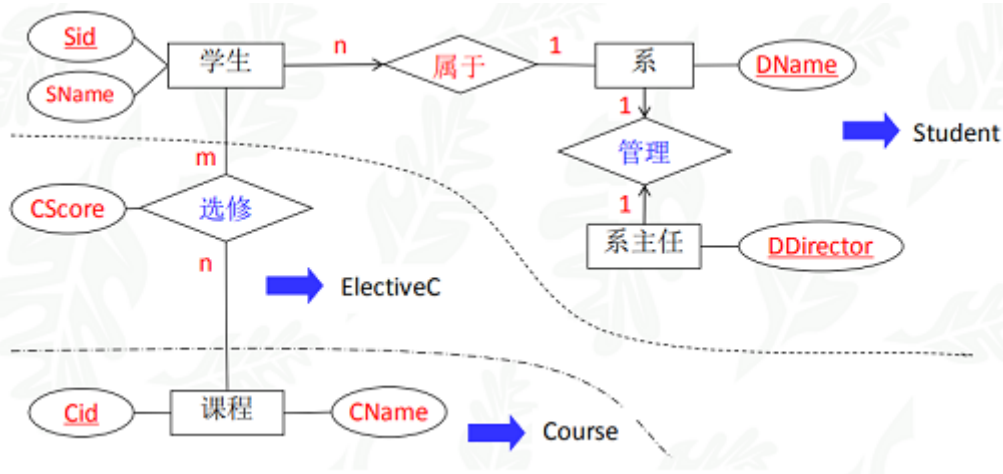
- 其实函数依赖，就类似于离散数学的传递之类的，最接近于函数 $y = f(x)$ ，其中决定子是 X ，另一个是 Y ，因为 1 个 Y 对应着多个 X

- 那么有了函数依赖，以其为基础的范式又有哪些呢？

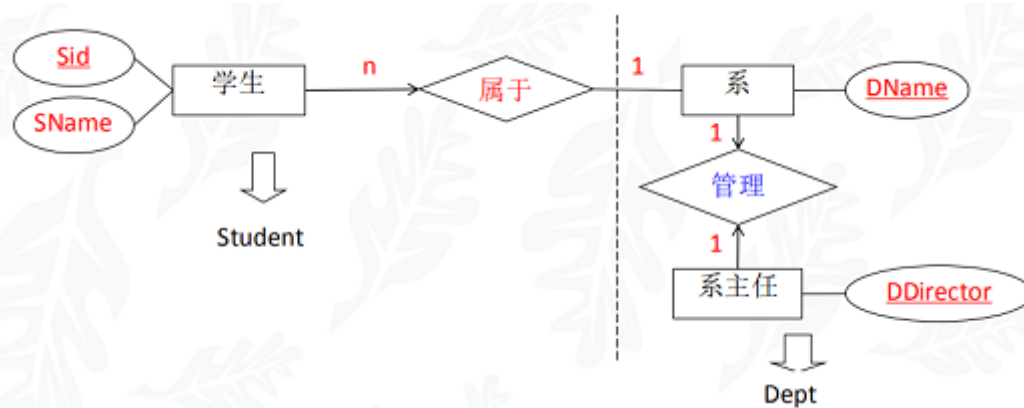
范式之间存在的关系或级别：



- 我们对于规范化的理解是什么呢？
 - 定义：将一个给定的关系模式转化为某种范式的过程，规范化过程实际上就是对关系模式不断分解的过程
 - 目的：消除异常
- 规范化程度是不是越高越好呢？
 - 答案不是的。当我们对关系模式不断分解，但是当我们需要查询这些有联系的数据时，我们需要联结运算，因此开销增大
- 第一范式：
 - 每个属性的值域，都是不可分的简单数据项的集合（即每个属性都不是多值属性），简单来说，要求表中每一列与每一行的交叉处只有一个值
- 第二范式：
 - R 是 1NF
 - 每个非键属性都完全函数依赖于 R 的键，消除其中的部分函数依赖



- 第三范式:
 - R是2NF
 - 任何一个非键属性都不传递依赖于R的任何候选键



- BCNF范式: 如果对于R的每个函数依赖 $X \rightarrow Y$, X 必为候选键, 则R为BCNF范式
 - 所有非键属性都完全函数依赖于每个候选键
 - 所有键属性都完全函数依赖于每个不包含它的候选键
 - 没有任何属性完全函数依赖于非键的任何一组属性
- 了解了以上四个常用范式, 我们具体的规范化步骤是什么呢?

规范化的实质: 概念的单一化。即: 一个关系只描述一个概念、一个实体或实体间的一种联系, 若多于一个概念就应将其它概念分离出去。

规范化基本步骤:

