

范式概念：

1) **1NF**：目标就是表中每列都不可分割；

2) **2NF**：目标就是表中的每行都是有标识的。前提是满足了 **1NF**。当关键字为单 *field* 时，一定满足 **2NF**。当关键字为组合 *field* 时（即超过一个 *field*），不能存在组合关键字中有某个字段能够决定非关键字段的某部分。非主 *field* 非部分依赖于主 *field*，即非关键字段必须完全依赖于一组 组合关键字，而不是组合关键字的某一部分。

3) **3NF**：目标是一个 *table* 里面所有的列不依赖于另外一个 *table* 里面非关键的列。前提是满足了 **2NF**，不存在某个非关键字段决定另外一个非关键字段。即：不存在传递依赖（关键字 $x \rightarrow$ 非关键属性 $y \rightarrow$ 非关键属性 z ）

4) **BCNF**：前提是满足了 **2NF**，不存在某个非关键字段决定另外一个非关键字段。也不存在某个关键字段决定另外一个关键字段。即：在 **3NF** 基础上，加上约束：不存在某个关键字段决定另外一个关键字段。

1 第一范式（1NF）

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**。所谓第一范式（**1NF**）是指数据库表的每一列都是不可分割的基本数据项，同一列中不能有多值，即实体中的某个属性不能有多个值或者不能有重复的属性。如果出现重复的属性，就可能需要定义一个新的实体，新的实体由重复的属性构成，新实体与原实体之间为一对多关系。在第一范式（**1NF**）中表的每一行只包含一个实例的信息。例如，对于图 3-2 中的员工信息表，不能将员工信息都放在一列中显示，也不能将其中的两列或多列在一列中显示；员工信息表的每一行只表示一个员工的信息，一个员工的信息在表中只出现一次。**简而言之，第一范式就是无重复的列。**

2 第二范式（2NF）

第二范式（**2NF**）是在第一范式（**1NF**）的基础上建立起来的，即满足第二范式（**2NF**）必须先满足第一范式（**1NF**）。第二范式（**2NF**）要求数据库表中的每个实例或行必须可以被惟一地区分。为实现区分通常需要为表加上一个列，以存储各个实例的惟一标识。如图 3-2 员工信息表中加上了员工编号（*emp_id*）列，因为每个员工的员工编号是惟一的，因此每个员工可以被惟一区分。这个惟一属性列被称为主关键字或主键、主码。第二范式（**2NF**）要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列，以存储各个实例的惟一标识。**简而言之，第二范式就是非主属性非部分依赖于主关键字。**

3 第三范式（3NF）

满足第三范式（**3NF**）必须先满足第二范式（**2NF**）。简而言之，第三范式（**3NF**）要求一个数据库表中不包含已在其它表中已包含的非主关键字信息。例如，存在一个部门信息表，其中每个部门有部门编号（*dept_id*）、部门名称、部门简介等信息。那么在图 3-2 的员工信息表中列出部门编号后就不能再将部门名称、部门简介等与部门有关的信息再加入员工信息表中。如果不存在部门信息表，则根据第三范式（**3NF**）也应该构建它，否则就会有大量的数据冗余。**简而言之，第三范式就是属性不依赖于其它非主属性。**

例子：

第一范式（**1NF**）：数据库表中的字段都是单一属性的，不可再分。这个单一属性由基本类型构成，包括整型、实数、字符型、逻辑型、日期型等。

例如，如下的数据库表是符合第一范式的：字段 1 字段 2 字段 3 字段 4

而这样的数据库表是不符合第一范式的：字段 1 字段 2 字段 3 字段 4 字段 31 字段 32

很显然，在当前的任何关系数据库管理系统（S）中，傻瓜也不可能做出不符合第一范式的数据库，因为这些 S 不允许你把数据库表的一列再分成二列或多列。因此，**你想在现有的 S 中设计出不符合第一范式的数据库都是不可能的。**

第二范式（2NF）：数据库表中不存在非关键字段对任一候选关键字的部分函数依赖（部分函数依赖指的是存在组合关键字中的某些字段决定非关键字段的情况），也即所有非关键字段都完全依赖于任意一组候选关键字。

假定选课关系表为 Ss(学号, 姓名, 年龄, 课程名称, 成绩, 学分), 关键字为组合关键字(学号, 课程名称), 因为存在如下决定关系：

(学号, 课程名称) → (姓名, 年龄, 成绩, 学分)

这个数据库表不满足第二范式，因为存在如下决定关系：

(课程名称) → (学分)

(学号) → (姓名, 年龄)

即存在组合关键字中的字段决定非关键字的情况。

由于不符合 2NF，这个选课关系表会存在如下问题：1) 数据冗余：同一门课程由 n 个学生选修，“学分”就重复 n-1 次；同一个学生选修了门课程，姓名和年龄就重复了-1 次。2) 更新异常：若调整了某门课程的学分，数据表中所有行的“学分”值都要更新，否则会出现同一门课程学分不同的情况。3) 插入异常：假设要开设一门新的课程，暂时还没有人选修。由于还没有“学号”关键字，课程名称和学分也无法记录入数据库。4) 删除异常：假设一批学生已经完成课程的选修，这些选修记录就应该从数据库表中删除。但是，与此同时，课程名称和学分信息也被删除了。很显然，这也会导致插入异常。

把选课关系表 Ss 改为如下三个表：

学生：Sn(学号, 姓名, 年龄)；

课程：s(课程名称, 学分)；

选课关系：Ss(学号, 课程名称, 成绩)。

这样的数据库表是符合第二范式的，消除了数据冗余、更新异常、插入异常和删除异常。

另外，所有单关键字的数据库表都符合第二范式，因为不可能存在组合关键字。

第三范式（3NF）：在第二范式的基础上，数据表中如果不存在非关键字段对任一候选关键字段的传递函数依赖则符合第三范式。所谓传递函数依赖，指的是如果存在“A → B → C”的决定关系，则传递函数依赖于 A。因此，**满足第三范式的数据库表应该不存在如下依赖关系：关键字段 → 非关键字段 x → 非关键字段 y**

假定学生关系表为 Sn(学号, 姓名, 年龄, 所在[]学院[], 学院地点, 学院电话), 关键字为单一关键字“学号”，因为存在如下决定关系：

(学号) → (姓名, 年龄, 所在[]学院[], 学院[]地点, []学院[]电话)

这个数据库是符合 2NF 的，但是不符合 3NF，因为存在如下决定关系：

(学号) → (所在[]学院[]) → ([]学院[]地点, []学院[]电话)

即存在非关键字段“[]学院[]地点”、“[]学院[]电话”对关键字段“学号”的传递函数依赖。

它也会存在数据冗余、更新异常、插入异常和删除异常的情况，读者可自行分析得知。

把学生关系表分为如下两个表：

学生：(学号, 姓名, 年龄, 所在[]学院[])；

[]学院[]：([]学院[], 地点, 电话)。

这样的数据库表是符合第三范式的，消除了数据冗余、更新异常、插入异常和删除异常。

鲍依斯-科得范式（BCNF）：在第三范式的基础上，数据库表中如果不存在任何字段对任一候选关键字段的传递函数依赖则符合 BCNF。

假设仓库管理关系表为 Ssanag(仓库, 存储物品, 管理员, 数量), 且有一个管理员只在一个仓

库工作：一个仓库可以存储多种物品。这个数据库表中存在如下决定关系：

(仓库, 存储物品) → (管理员, 数量)

(管理员, 存储物品) → (仓库, 数量)

所以, (仓库, 存储物品)和(管理员, 存储物品)都是 *Ssanag* 的候选关键字, 表中的唯一非关键字段为数量, 它是符合第三范式的。但是, 由于存在如下决定关系:

(仓库) → (管理员)

(管理员) → (仓库)

即存在关键字段决定关键字段的情况, 所以其不符合 *BCNF* 范式。它会出现如下异常情况:

- 1) 删除异常: 当仓库被清空后, 所有"存储物品"和"数量"信息被删除的同时, "仓库"和"管理员"信息也被删除了。
- 2) 插入异常: 当仓库没有存储任何物品时, 无法给仓库分配管理员。
- 3) 更新异常: 如果仓库换了管理员, 则表中所有行的管理员都要修改。

把仓库管理关系表分解为二个关系表:

仓库管理: *Ssanag*(仓库, 管理员);

仓库: *Ss*(仓库, 存储物品, 数量)。

这样的数据库表是符合 *BCNF* 范式的, 消除了删除异常、插入异常和更新异常。

简言之数据库五大范式:

第一范式: 对于表中的每一行, 必须且仅仅有唯一的行值. 在一行中的每一列仅有唯一的值并且具有原子性。

(第一范式是通过把重复的组放到每个独立的表中, 把这些表通过一对多关联联系起来这种方式来消除重复组的)

第二范式: 第二范式要求非主键列是主键的子集, 非主键列活动必须完全依赖整个主键。主键必须有唯一性的元素, 一个主键可以由一个或更多的组成唯一值的列组成。一旦创建, 主键无法改变, 外键关联一个表的主键。主外键关联意味着一对多的关系。(第二范式处理冗余数据的删除问题。当某张表中的信息依赖于该表中其它的不是主键部分的列的时候, 通常会违反第二范式)

第三范式: 第三范式要求非主键列互不依赖。(第三范式规则查找以消除没有直接依赖于第一范式和第二范式形成的表的主键的属性。我们为没有与表的主键关联的所有信息建立了一张新表。每张新表保存了来自源表的信息和它们所依赖的主键)

第四范式: 第四范式禁止主键列和非主键列一对多关系不受约束

第五范式: 第五范式将表分割成尽可能小的块, 为了排除在表中所有的冗余。