

数据库的设计范式是数据库设计所需要满足的规范，满足这些规范的数据库是简洁的、结构明晰的，同时，不会发生插入（insert）、删除（delete）和更新（update）操作异常。反之则是乱七八糟，不仅给数据库的编程人员制造麻烦，而且面目可憎，可能存储了大量不需要的冗余信息。

范式说明

1.1 第一范式（1NF）无重复的列

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。简而言之，第一范式就是无重复的列。

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。

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 3.1	字段 3.2

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。很显然，在当前的任何关系数据库管理系统（DBMS）中，傻瓜也不可能做出不符合第一范式的数据库，因为这些 DBMS 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 DBMS 中设计出不符合第一范式的数据库都是不可能的。

1.2 第二范式（2NF）属性完全依赖于主键 [消除部分子函数依赖]

如果关系模式 R 为第一范式，并且 R 中每一个非主属性完全函数依赖于 R 的某个候选键，则称为第二范式模式。

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。这个惟一属性列被称为主关键字或主键、主码。

例如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。

简而言之，第二范式（2NF）就是非主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（设有函数依赖 $W \rightarrow A$ ，若存在 XW ，有 $X \rightarrow A$ 成立，那么称 $W \rightarrow A$ 是局部依赖，否则就称 $W \rightarrow A$ 是完全函数依赖）。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。

假定选课关系表为 SelectCourse(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:

$(\text{学号, 课程名称}) \rightarrow (\text{姓名, 年龄, 成绩, 学分})$

这个数据库表不满足第二范式, 因为存在如下决定关系:

$(\text{课程名称}) \rightarrow (\text{学分})$

$(\text{学号}) \rightarrow (\text{姓名, 年龄})$

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 m 门课程, 姓名和年龄就重复了 $m-1$ 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。这样, 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 SelectCourse 改为如下三个表：

学生：Student(学号, 姓名, 年龄)；

课程：Course(课程名称, 学分)；

选课关系：SelectCourse(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。

1.3 第三范式 (3NF) 属性不依赖于其它非主属性 [消除传递依赖]

如果关系模式 R 是第二范式，且每个非主属性都不传递依赖于 R 的候选键，则称 R 为第三范式模式。

满足第三范式 (3NF) 必须先满足第二范式 (2NF)。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。

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。简而言之 , 第三范式就是属性不依赖于其它非主属性。

所谓传递函数依赖 , 指的是如果存在 " $A \rightarrow B \rightarrow C$ " 的决定关系 , 则 C 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 \rightarrow 非关键字段 $x \rightarrow$ 非关键字段 y

假定学生关系表为 Student(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) \rightarrow (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) \rightarrow (所在学院) \rightarrow (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。