

数据库规范化三个范式应用实例 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/273/2021_2022__E6_95_B0_E6_8D_AE_E5_BA_93_E8_c97_273108.htm 规范化为什么重要？目前很多的数据库由于种种原因还没有被规范化。本文中解释了其中一些原因，并用不同形式的范式（normal form）规范化了一个保险公司的理赔表。在这个过程中表的改变以及添加的一些附加表使数据库效率更高、错误更少、更容易维护。数据库的规范化是优化表的结构和把数据组织到表中的实践，这样做数据才能更明确。规范化使你能够改变业务规则、需求和数据而不需要重新构造整个系统。通过改变存储数据的方式--仅仅改变一丁点--并改变访问这些信息的程序，你就可以消除很多错误或垃圾数据出现的机会并减轻更新信息所必要的工作量。公司现实存在的一个问题可以用一句话概括"我们一般都这样做"。我们一般像采用那种方式存储信息；我们一般允许人们把任何信息写入；我们一般采用那种方式编程。这通常是一件坏事，特别是对于年轻的和正在学习的公司来说。但是，当有新的系统和更好的完成任务的途径的时候，有时"采用那种方式任务完成得很好"这句话可能需要重新探讨和修改。规范化数据就是公司常常采用的有益的方式之一。尽管对于COBOL程序（例如任何COBOL程序员都熟悉的文件布局）使用数据来说，把它们（数据）存储在关系数据库中存储与存储在平面文件中很相似，但是存储在平面文件中的方法并不是完成任务的必要的最好的途径，特别是由于你不了解两者之间的差别或害怕改变，而简单地把过去的观念带入到现在的方式。注意：Dictionary.com是这样定

义规范化的：“使其标准，特别使导致它符合某种标准或规范。”或“某种标准的强制接受”。Webopedia认为规范化是“在关系数据库设计中，组织数据以最小化冗余的过程。规范化通常包括把一个数据库分成两个或多个表并定义表之间的关系。其目标是隔离数据，这样添加、删除和修改某个字段只需要在一个表中进行，接着可以通过定义的关系传递到数据库中剩余的表中”。我更喜欢这个定义。术语在你了解现实世界中的一个保险公司的例子之前，你需要了解一些在讨论中会用到的术语。处理数据库的时候，特别是在处理规范化问题的时候，下面一部分讲到的一组新的关键字很有作用：

关系（Relation）：从本质上说，关系是一个包含行和列的二维表或数组。

关联（Relationship）：关联是不同表之间的数据彼此联系的方法。关联同时存在于形成不同实体的数据项之间和表实体本身之间，构成了数据库规范化的基本核心问题。数据关联有三种基本的类型，对它们有所了解是很重要的：

一对一（1:1）：一对一关联意味着任何给定的每个（而不是大多数）实例严密地与另一个实体的一个实例对应。每个人只有一个正确的指纹就是唯一的。每个电话号码准确地与一个付帐的独立私人客户对应（不是公司）。美国的每个人都只有一个社会保障号码。

一对多（1:M）：一对多关联意味着给定实体的一个实例可以可以与另一个实体的零个实例、一个实例或者多个实例关联。每个人可能没有小孩、有一个小孩或多个小孩。每个人可能没有汽车、有一辆汽车或多辆汽车。

多对多（M:N）：多对多关联（给定实体的零个、一个或多个实例与另一个实体的零个、一个或多个实例关联）是一种直接模拟很复杂的关联，它经常被分解为多个1:M关

联。由于多个家庭混合在一起，一个或多个小孩可能没有父母亲（孤儿）、一个父母（单亲家庭），多于一个父母（两个仍然在一起或者离婚的两个父母、或者离婚了又复婚了的父母）。房屋或财产可以转让给一个人或多个人，而这些人（一个或多个）在遗嘱上可能又一个或多个房屋或财产。

属性（Attribute）：属性被认为是程序或数据库中的某些组件的可以修改的特性或特征，它可以被设置为不同值或者关系或表中的列。

Tuple：Tuple是关系数据库或非关系数据库中的排序了的一组值或值属性：关系中的一行。

删除异常：删除异常指由于其它数据故意的删除而导致的数据矛盾或未预料到的数据（信息）丢失。

插入异常：插入异常指由于数据的缺少或缺乏导致没有能力把信息添加到数据库。

更新异常：更新异常指由于数据冗余或者冗余数据的不完整更新造成的数据矛盾。

关系的分解：关系的分解指把一个关系分解成多个关系，从而使关系符合更高的范式。

数据冗余：数据冗余指数据库中没有必要的重复数据。

数据完整性：数据完整性指数据库中数据的一致性。保证数据完整性很重要，只有这样用户才知道他们依赖的数据是正确的、他们查询的结果以及程序才是精确的和符合期望的。

原子值：原子值是一个值，它既不是能被进一步拆分的一组值，也不是一个重复的组。每个列都有一个完整的值，但是只有一个值--这个值不能被分解为多个部分，它要么被数据库使用，要么被使用数据库的用户访问的信息。

参考完整性规则：参考完整性规则指存储在非空的外部键中的值必须是某种关系中的关键数据项。

外部键：外部键是一个关系中的一组属性（一个或多个列），它同时也是某种（相同的或其它的）关系中的主键。它

是关系之间的逻辑链接。参考自己关系的外部键称为递归外部键。 功能依赖：功能依赖意味着一行中某个属性的值由该行中另一个属性的值决定。这通常出现在主键（使某行唯一的信息片断）与该行的其它信息之间。城市和州的组合依赖于Zip（邮政）代码，即使给定的一个州中有很多Zip代码与某个城市关联。美国的每个合法的人员身份依赖于他的社会保障号码。 决定性：功能依赖左边的属性决定行中其它属性的值（Zip代码决定了城市和州；社会保障号码决定了人的身份；执照号码和州决定了汽车的拥有者）。 实体完整性规则：实体完整性规则指某一行的关键属性可能为空（如果你在某个城市就有一个Zip代码；如果你有一辆汽车就有一个执照号码）。 约束：约束是一种规则，它限定了数据库中的值。电话号码必须是数字的；美元数量必须是数字的；state必须是合法的州或省；country必须是合法的国家；日期不能是2月31号。现在你已经知道了很多相关的术语了，我们可以看看相关术语中规范会的意义了。下面的例子并不是典型的雇员—经理—部门示例，也不是学生—教授—课程提供示例。我将演示一个假设的保险公司的数据库。数据库中的表比本示例中用到的要复杂得多，但是与人们遇到的比较相近。图1显示了理赔（claim）表的非规范化定义。尽管在某个保险公司的数据库中的表比它多得多，但是这些表为我们提供了一些背景，通过它我们可以看到规范化和其分支。请记住每个章节中的示例都只有部分列，这样就简化了示例并使你轻易地看到发生变化的东西。 CLAIM_NUM、 OCCURANCE_NUM、 CLAIM_STATUS、 ACCDNT_YR、 ACCDNT_DT、 REPORTED_DT、 ENTERED_DT、

CLAIM_DT1、 CLAIM_DT2、 CLAIM_DT3、 CLAIM_DT4、
CLAIM_DT4、 CLAIM_DT5、 CLAIM_DT6、 CLAIM_DT7
、 CLAIM_DT8、 CLAIM_DT9、 CLAIM_DT10、
CLOSED_DT、 DEATH_DT、 ASSIGNED_DT、
ADJSTER_CD、 ADJUSTER_NAME、 AGENT_CD、
AWARD_CD、 CAUSE_CD、 CAUSE_DESC、 LOCATION
、 SITE、 COVERAGE_CD、 COVERAGE_DESC、
DED_RECOV、 DEDUCTIBLE_REMAIN、 PAID_1、
RESERVED_1、 PAID_2、 RESERVED_2、 PAID_3、
RESERVED_3、 PAID_4、 RESERVED_4、 PAID_5、
RESERVED_5、 PAID_6、 RESERVED_6、 PAID_7、
RESERVED_7、 PAID_8、 RESERVED_8、 PAID_9、
RESERVED_9、 PAID_10、 RESERVED_10、 LEGAL_FLG、
KEY1、 KEY2、 KEY3、 KEY4、 KEY5、 KEY6、 KEY7、
KEY8、 KEY9、 KEY10、 SEVERITY_CD、 POLICY_NUM、
PAYMENT_NUM、 SSN、 STATE、 ACTVY_DT、
ENTRY_DT、 ADMIN_CD、 ADMIN_DESC、 REOPEN_DT
、 INSURED_NAME、 INSURED_ADDRESS、
INSURED_PHONE、 INSURED_CITY、 INSURED_STATE、
INSURED_ZIP、 CLAIMANT_NAME、
CLAIMANT_ADDRESS、 CLAIMANT_CITY、
CLAIMANT_STATE、 CLAIMANT_ZIP、
CLAIMANT_PHONE、 SPECIAL_DT_1、 SPECIAL_DT_2、
SPECIAL_DT_3、 SPECIAL_DT_4、 SPECIAL_DT_5、
SPECIAL_DT_6、 SPECIAL_DT_7、 SPECIAL_DT_8
、 SPECIAL_DT_9、 SPECIAL_DT_10、 GROSS_PD、

POLICY_ID 图1：未规范化的理赔表的列 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com