

Mysql数据库

目前关系数据库有六种范式：

第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、巴斯-科德范式（BCNF）、第四范式(4NF)和第五范式（5NF，又称完美范式）。满足最低要求的范式是第一范式（1NF）。在第一范式的基础上进一步满足更多规范要求的称为第二范式（2NF），其余范式以次类推。一般说来，数据库只需满足第三范式(3NF)就行了。所以这里就只记录三范式相关的知识。

一 范式

1NF:字段不可分; 2NF:有主键，非主键字段依赖主键; 3NF:非主键字段不能相互依赖;

解释: 1NF:原子性 字段不可再分,否则就不是关系数据库; 2NF:唯一性 一个表只说明一个事物; 3NF:每列都与主键有直接关系，不存在传递依赖;

第一范式（1NF）

即表的列的具有原子性,不可再分解，即列的信息，不能分解,只要数据库是关系型数据库(mysql/oracle/db2/informix/sysbase/sql server)，就自动的满足1NF。数据库表的每一列都是不可分割的原子数据项，而不能是集合，数组，记录等非原子数据项。如果实体中的某个属性有多个值时，必须拆分为不同的属性。通俗理解即一个字段只存储一项信息。



关系型数据库: mysql/oracle/db2/informix/sysbase/sql server 非关系型数据库: (特点: 面向对象或者集合) NoSql数据库: MongoDB/redis(特点是面向文档)

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我们设计一个主键来实现(这里的主键不包含业务逻辑)。

即满足第一范式前提，当存在多个主键的时候，才会发生不符合第二范式的情况。比如有两个主键，不能存在这样的属性，它只依赖于其中一个主键，这就是不符合第二范式。通俗理解是任意一个字段都只依赖表中的同一个字段。（涉及到表的拆分）

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键字段。就是说，表的信息，如果能够被推导出来，就不应该单独的设计一个字段来存放(能尽量外键join就用外键join)。很多时候，我们为了满足第三范式往往会把一张表分成多张表。

即满足第二范式前提，如果某一属性依赖于其他非主键属性，而其他非主键属性又依赖于主键，那么这个属性就是间接依赖于主键，这被称作传递依赖于主属性。通俗解释就是一张表最多只存两层同类型信息。



巴斯-科德范式（BCNF）

Boyce-Codd Normal Form（巴斯-科德范式）

在3NF基础上，任何非主属性不能对主键子集依赖（在3NF基础上消除对主码子集的依赖）

巴斯-科德范式（BCNF）是第三范式（3NF）的一个子集，即满足巴斯-科德范式（BCNF）必须满足第三范式（3NF）。通常情况下，巴斯-科德范式被认为没有新的设计规范加入，只是对第二范式与第三范式中设计规范要求更强，因而被认为是修正第三范式，也就是说，它事实上是对第三范式的修正，使数据库冗余度更小。这也是BCNF不被称为第四范式的原因。某些书上，根据范式要求的递增性将其称之为第四范式是不规范，也是更让人不容易理解的地方。而真正的第四范式，则是在设计规范中添加了对多值及依赖的要求

第四范式

对于第四范式，从理论层面来讲是，关系模式 $R \in 1NF$,如果对于R对于R的每个非平凡多值依赖 $X \twoheadrightarrow Y$ (Y 不属于 X), X 都含有候选码，则 $R \in 4NF$ 。4NF就是限制关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。显然一个关系模式是4NF，则必为BCNF。

也就是说，当一个表中的非主属性互相独立时（3NF），这些非主属性不应该有多值。若有多值就违反了第四范式。

第五范式

不得存在不遵循键约束的非平凡连接依赖。如果且只有一个表符合4NF，同时其中的每个连接依赖被候选键所包含，此表才符合第五依赖。

二 反三范式

没有冗余的数据库未必是最好的数据库，有时为了提高运行效率，提高读性能，就必须降低范式标准，适当保留冗余数据。具体做法是：在概念数据模型设计时遵守第三范式，降低范式标准的工作放到物理数据模型设计时考虑。降低范式就是增加字段，减少了查询时的关联，提高查询效率，因为在数据库的操作中查询的比例要远远大于DML的比例。但是反范式化一定要适度，并且在原本已满足三范式的基础上再做调整的