

# 数据库设计

## 1. 数据库设计三范式

### 1.1 什么是数据库设计范式？

数据库表的设计依据。教你怎么进行数据库表的设计。

### 1.2 数据库设计范式共有？

3个。

第一范式：要求任何一张表必须有主键，每一个字段原子性不可再分。

第二范式：建立在第一范式的基础之上，要求所有非主键字段完全依赖主键，不要产生部分依赖。

第三范式：建立在第二范式的基础之上，要求所有非主键字段直接依赖主键，不要产生传递依赖。

声明：三范式是面试官经常问的，所以一定要熟记在心！

### 1.3 数据库范式的作用

设计数据库表的时候，按照以上的范式进行，可以避免表中数据的冗余，空间的浪费。

### 1.4 第一范式

是最核心，最重要的范式，所有表的设计都需要满足第一范式。

内容：要求任何一张表都必须有主键，并且每一个字段都是原子性不可再分。

示例：

学生编号	学生姓名	联系方式
1001	张三	zs@gmail.com,1359999999
1002	李四	ls@gmail.com,1369999999
1001	王五	ww@163.net,1348888888

以上是学生表，满足第一范式吗？

答：不满足，第一：没有主键。第二：联系方式可以分为邮箱地址和电话

应该修改为：

学生编号(pk)	学生姓名	邮箱地址	联系电话
1001	张三	zs@gmail.com	1359999999
1002	李四	ls@gmail.com	1369999999
1003	王五	ww@163.net	1348888888

# 1.5 第二范式

内容：建立在第一范式的基础之上，要求所有非主键字段必须完全依赖主键，不要产生部分依赖。

看是否满足第二范式时，要先看看是否已经满足第一范式。

第二范式解决了多对多关系的表的设计问题。

示例：

学生编号	学生姓名	教师编号	教师姓名
1001	张三	001	王老师
1002	李四	002	赵老师
1003	王五	001	王老师
1001	张三	002	赵老师

这张表是用来描述学生和老师的关系，而学生和老师的关系是：1个学生可能有多个老师，1个老师可能有多个学生

这是非常典型的：多对多关系！

分析以上的表是否满足第一范式？

不满足第一范式。

怎么满足第一范式呢？

修改：学生编号 教师编号，两个字段联合做主键，复合主键（PK：学生编号+教师编号）。

学生编号 + 教师编号(pk)	学生姓名	教师姓名
1001 001	张三	王老师
1002 002	李四	赵老师
1003 001	王五	王老师
1001 002	张三	赵老师

经过修改之后，以上的表满足了第一范式。但是满足第二范式吗？

不满足，因为非主键字段部分依赖主键而不是完全依赖。

“张三”依赖的是主键里的1001部分，而不是整个主键，“王老师”依赖的只是主键里的001部分，只是部分依赖。

产生部分依赖有什么缺点？

数据冗余了。空间浪费了。

例如本表中的学生姓名字段中的“张三”重复了，教师姓名字段中的“王老师”重复了。

为了让以上的表满足第二范式，你需要这样设计：使用三张表来表示多对多的关系！！

学生表：

学生编号(pk)	学生名字
1001	张三
1002	李四
1003	王五

教师表：

教师编号(pk)	教师姓名

001	王老师
002	赵老师

学生教师关系表

id(pk)	学生编号(fk)	教师编号(fk)
1	1001	001
2	1002	002
3	1003	001
4	1001	002

## 背口诀(需要记住! ):

多对多怎么设计?

多对多, 三张表, 关系表两个外键!!!

## 实际的工作中是如何设计表的:

实际工作中, 在做表的设计的时候, 不用去想范式, 而是直接想实体之间是什么关系, 关系应该怎么设计。

## 1.6 第三范式

内容: 第三范式建立在第二范式的基础之上, 要求所有非主键字段必须直接依赖主键, 不要产生传递依赖。

第三范式解决了一对多关系的表的设计问题。

学生编号 (PK)	学生姓名	班级编号	班级名称
1001	张三	01	一年一班
1002	李四	02	一年二班
1003	王五	03	一年三班
1004	赵六	03	一年三班

以上表的设计是描述: 班级和学生的关系。一个班级可能对应多个学生, 一个学生只对应一个班级, 所以是1对多关系。

分析以上表是否满足第一范式?

满足第一范式, 有主键。

分析以上表是否满足第二范式?

满足第二范式, 因为主键不是复合主键, 没有产生部分依赖。主键是单一主键。

分析以上表是否满足第三范式?

第三范式要求: 不要产生传递依赖!

一年一班依赖01, 01依赖1001, 产生了传递依赖。不符合第三范式的要求。产生了数据的冗余。

例如本表的班级名称字段中的“一年三班”重复了。

为了让以上的表满足第三范式, 你需要这样设计来表示一对多的关系:

班级表: 一方

班级编号(pk)	班级名称
01	一年一班
02	一年二班
03	一年三班

学生表：多方

学生编号 (PK)	学生姓名	班级编号 (fk)
1001	张三	01
1002	李四	02
1003	王五	03
1004	赵六	03

## 背口诀(需要记住! ):

一对多怎么设计?

一对多，两张表，多的表加外键!!!

## 1.7 如何判断两个实体间的关系

判断实体A和实体B的关系。先分析：A能关联几个B， B能关联几个A。然后看表述中的"多个"后面跟着谁，谁就是多方。不是多方的，就是1方。

示例1:

班级和学生的关系判断:

一个班级能关联多个学生。

一个学生只能关联1个班级。

"多个"后面跟着学生，所以学生是多方，可以得出班级是1方。

示例2:

教师和学生的关系判断:

一个教师能关联多个学生。

一个学生能关联多个教师。

"多个"后面跟着学生，也跟着老师。所以学生是多方，老师也是多方。

## 1.8 一对一关系的表的设计

一对一：

一对一放到一张表中就行了。

但在实际的开发中，可能存在一张表字段太多的情况，太庞大。这个时候要拆分表。

当一对一的表太庞大，拆分表后，分成了多张表，此时该怎么设计？

没有拆分表之前：一张表

t\_user

id	login_name	login_pwd	real_name	email	address
.....					
1	zhangsan	123	张三	zhangsan@xxx	
2	lisi	123	李四	lisi@xxx	
...					

这种庞大的表建议拆分为两张：

t\_login 登录信息表

id(pk)	login_name	login_pwd
1	zhangsan	123
2	lisi	123

t\_user 用户详细信息表 （设置外键列login\_id，并给外键列加上唯一性约束）

id(pk)	real_name	email	address.....	login_id(fk+unique)
100	张三	zhangsan@xxx		1
200	李四	lisi@xxx		2

背口诀：

一对一，外键唯一！！

意思是：在一对一关系拆分表之后，在其中的一张表中设置外键，并给外键列设置唯一性约束。

## 1.9 嘱咐一句话：

数据库设计三范式是理论上的。

实践和理论有的时候有偏差。

最终的目的都是为了满足客户的需求，有的时候会拿冗余(即拿存储空间)换执行速度。

因为在**sql**当中，表和表之间连接次数越多，效率越低。（笛卡尔积）

有的时候可能会存在冗余，但是为了减少表的连接次数，这样做也是合理的，并且对于开发人员来说，这样做，**sql**语句的编写难度也会降低。

面试的时候把这句话上：他就不会认为你是初级程序员了！