

1、表的设计范式

设计范式是设计表的依据，按照三范式设计的表不会出现数据的冗余

2、设计三范式

--第一范式：任何一张表都应该有主键，并且每一个字段原子性不可再分

不可再分表示该字段不应该可以自成一表，例如学生和班级，如果放进一张表，班级这个字段还是可以再分成班级表，该表中有班级编号和班级名字，所以这些个字段不能放在一张表中，每个字段是原子性
一个表不能没有主键。

--第二范式：是在第一范式的基础上，所有非主键完全依赖一个主键字段，不能有部分依赖

例子：一张表中有学生编号sno，学生名sname，教师编号tno，教师名tname，该表中将字段sno和tno联合起来做一个复合主键，此时就有sname依赖sno，又有tname依赖tno，这时数据就冗余了
这种该怎么设计呢？

因为每一个学生有多名教师，每位教师又有多位学生，是多对多的关系

--所以多对多，三张表，其中关系表有两个外键约束

学生表

sno(pk) sname

1	张三
2	李四
3	王五

教师表

tno(pk) tname

1	李老师
2	王老师
3	陈老师

教师与学生的关系表

id(pk) sno(fk) tno(fk)

1	1	2
2	1	3
3	2	1
4	2	2
5	2	3

....

--第三范式：在第二范式的基础上，所有非主键字段依赖主键，不能产生传递依赖

例如，学生和班级，一张表中，学号、姓名、班号、班名，此时仅将学号定为主键，此时班号依赖学号主键，但是班名却依赖班号，然而班号却不是主键，这是不行的。

一个班级有多名学生，这是一对多的关系

--所以这样设计，一对多，两张表，多的表加外键

班级t_class表

cno(pk) cname

1	banji1
2	banji2

学生表t_student

sno(pk)	sname	cno(fk)

1	zz	1
2	ss	2
3	ww	1
4	ee	2

--实际开发中，以客户需求为准，有时候会拿冗余换执行速度

3、一对一怎么设计？

一张用户表**t_user**一般有用户名和密码，当然还有其他信息，此时我们将这张大表拆分成两张表：

用户登录表**t_ulogin**：其中保存用户名和密码

用户详细信息表**t_udetail**：保存有用户名，住址等其他信息

这时有两种设计方案：

--主键共享

将**t_udetail**表的主键用**t_ulogin**表的主键约束

t_ulogin表

id(pk)	uname	password
--------	-------	----------

1	zz	123
2	xx	123

t_udetail表

id(pk+fk)	realname	adress	phonenum
-----------	----------	--------	----------

1	杂杂	xcccv	123365
2	学习	csdsd	462616

--外键唯一

t_udetail表中的一个userid字段用**t_ulogin**表的主键，并且该字段**unique**约束

t_ulogin表

id(pk)	uname	password
--------	-------	----------

1	zz	123
2	xx	123

t_udetail表

id(pk)	realname	adress	phonenum	userid(fk+unique)
--------	----------	--------	----------	-------------------

1	杂杂	xcccv	123365	1
2	学习	csdsd	462616	2

--一对多的变形，多的表中外键加了**unique**约束