数据库myself

应用系统的首要目标：满足业务需求

1. 视图
   1. 只是一个查看数据的窗口，视图并不拥有数据，只体现一个结构；所以基本表中的数据发生变化，从视图中查出的数据也随之改变
   2. 语法
   3. Pro：
      1. 视图使用户能以多种角度看待统一数据，当许多不同种类的用户共享同一个数据库时，灵活性

CREATE VIEW 视图名称（字段值）

AS

SELECT 对应表中的相应字段

FROM 表

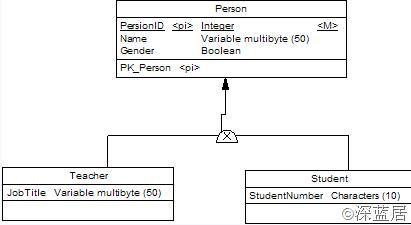
WHERE 限制条件

* + 1. 视图对重构数据库提供了一定的逻辑独立性（当数据库重构时，用户的应用程序不会受影响），例如将学生关系拆为sx和sy两个表，这是原表为sx何sy自然连接的结果，如果建立一个student的视图，尽管数据库的逻辑结构改变了，但应用程序通过视图仍然能够查找数据

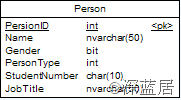
1. 数据库规范化
   1. 1NF：数据库中的所有表都满足这个特性，意即数据库表中的每个字段都是原子字段，不能进行拆分
   2. 2NF：数据库中不存在非关键属性对码的部分函数依赖，即码的真子集并不能决定一个非关键属性
   3. 3NF：数据库表中不存在非主属性对码的传递函数依赖
   4. BCNF：数据库表中不存在 主属性 对码的部分函数依赖与传递函数依赖

<https://www.zhihu.com/question/24696366>

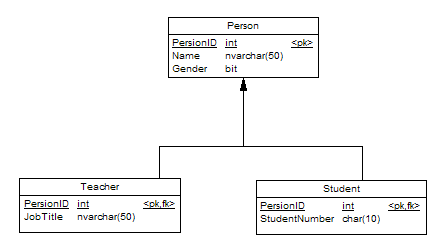
1. 数据库中的继承



* 1. 具体表继承：不建立父对象，将父对象的所有属性转移到子对象中，为每个子对象建立对应的表
  2. 单表继承，在一个宽表中列出所有父对象和子对象的属性，同时用一个标识列表示该行数据存储的是那个子类的数据



* 1. 多表继承：为父对象和每个子对象建立一个对应的表，然后在子表中设置该字表的主键为与父表关联的外键



可以看出，teacher的主键和student的主键同时又是外表的外键，连接到person表

1. ??数据库的设计原则？？