DDL：数据库定义语言 database definition language

对于数据库来说，

可食用CREATE 来创建一个数据库或者是一个数据表

DROP 来删除一个数据库或者一个表

ALTER 来修改数据库，如名字、类型

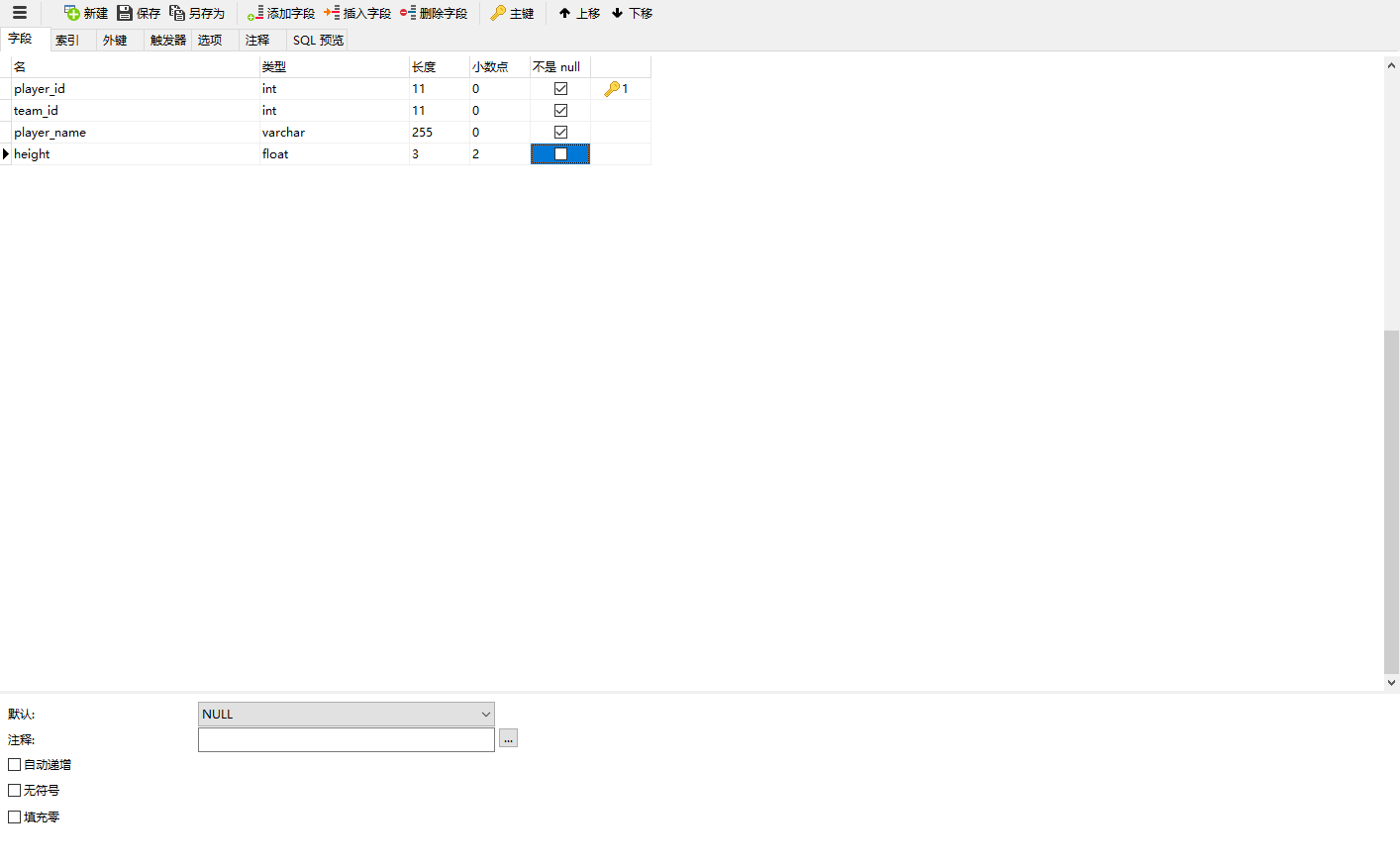
简洁快捷的方法是使用navicat来创建一个数据结构，可视化、设置方便

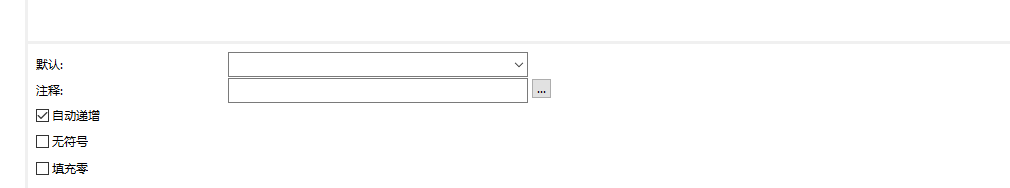
1.需求创建如下player的数据表



2.实现方法

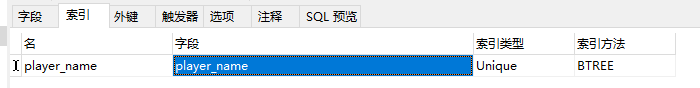
直接新建表格，并设置字段





自动递增在下面设置，主键player\_id需要设置成自动递增

设置索引类型



先设立一个名（相当于一个可变量），字段可以自己从…选择，也可以用’ ’来敲

Unique表示具有唯一性的约束，**BTREE索引方法未知，留待后面讲述**

设置要点：player\_id是设为主键，且递增

player\_name具有唯一性约束

数据存储引擎默认InnoDB

**数据表常见的约束**

约束的目的在于包住RDBMS里面数据的准确性和一致性。

比如有两个表(员工\职位),员工表中有员工代码、姓名、职位代码等属性，职位表中有职位代码、职位名称、职位等级等属性。你在其中员工表中进行了插入操作,你插入了一个新员工的信息，而这个新员工的职位是公司新创建的一个职位。如果没有一致性的保证，就会出现有这么一个员工，但是不知道他到底担当什么职责！这个只是它的一个小小方面。（一致性解释）ps：本应该需要在员工表中插一个职位代码

数据库一致性（Database Consistency）是指事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。数据库里表本应是一一对应的，更改后，数据库里的表还是要一一对应

准确性靠字段的属性来保证，比如是字符类型还是整数类型还是float类型，允许存在几位数字，是否存在空值

**主键约束**：

主键作用：唯一标志一条记录，不重复，不为空值。（UNIQUE + NOT NULL）

一个数据表中，主键只能有一个，主键可以使一个字段，也可以是复合层的字段（CONCAT）

**外键约束**：

外键确保表与表之间引用的完整性，

**字段约束**

**唯一性约束**：

表示该字段只能存在唯一值，唯一性约束相当于创建了一个约束和普通索引，目的是保证字段的正确性而普通索引只是提升数据检索的速度，并不对字段的唯一性进行约束

**NOT NULL 约束**：

表示该字段不存在空值

**DEFAULT 约束**：

表明了字段的默认值。如果在插入数据的时候，这个字段没有取值，就设置为默认值

**CHECK 约束**：

用来检查特定字段取值范围的有效性，CHECK 约束的结果不能为 FALSE，比如我们可以对身高 height 的数值进行CHECK 约束，必须≥0，且＜3，即CHECK(height>=0 AND height<3)。

CHECK约束和DEFAULT约束navicat无法实现，只能通过SQL语言实现

CHECK 在ALTER里使用

ALTER TABLE player

ADD CONSTRAINT chk\_player CHECK(player\_age >18 AND player\_money <150000);

撤销

ALTER TABLE player

DROP CHECK chk\_player

Constraint 后面紧接着约束的名字，后续想删除这个约束的时候直接找这个名就好了

**数据设计表的原则**

三少一多原则：

数据表个数越少越好

数据表里字段越少越好

联合主键的字段个数越少越好：多个字段复合起来成为主键称为联合主键

使用主键和外键越多越好

“三少一多”原则的核心就是简单可复用。简单指的是用更少的表、更少的字段、更少的联合主键字段来完成数据表的设计。可复用则是通过主键、外键的使用来增强数据表之间的复用率。因为一个主键可以理解是一张表的代表。键设计得越多，证明它们之间的利用率越高

冗余度：表里可通过其他字段计算推演出来的字段的多少；

数据冗余是指数据在存储器中的不必要的多次重复存储。