# 数据库表设计文档

### **环境说明**

该项目数据库采用了mysql数据库管理系统建立并维护，采用了InnoDb数据库引擎，其中数据库的用户名以及密码等均可采用自己的数据库密码设置，端口号为3306。

### 数据库表设计都有哪些

1. 数据库表设计有一对一、一对多、多对多
2. 一对一：一张表的一条记录一定只能与另外一张表的一条记录进行对应
3. 一对多：子表中的某一个字段是父表中的ID的字段，一张表中有一条记录可以对应另外一张表中的多条记录；但是反过来，另外一张表的一条记录，只能对应第一张表的一条记录，这种关系就是一对多或多对一。
4. 多对多：一对表中A的一条记录能够对应另外一张表B中的多条记录；同时B表中的一条记录，也能对应A表中的多条记录

### 三、这个项目采用哪种设计

注！该表是本地测试用表只有一张单表，方便测试使用，正常生产环境中情况复杂多变，要考虑很多情况，在这里只分析该表在实际的使用中应该怎么进行优化。

sql语句：

create table courses

(

id int auto\_increment

primary key,

courses\_id varchar(256) not null,

courses\_name varchar(256) not null,

price varchar(16) not null,

teacherName varchar(256) null,

title varchar(256) null,

link varchar(256) null,

pub\_time timestamp default CURRENT\_TIMESTAMP not null on update CURRENT\_TIMESTAMP,

create\_time timestamp default CURRENT\_TIMESTAMP not null

);

这里设置id为自增主键还有两个时间字段一个pub\_time记录如有updata操作就会更新时间。另一个create\_time记录创建词条记录的时间updata操作并不会更新此字段

### 四、有哪些可以优化的地方

优化设计1：该表主键为自增主键，在实际的项目中可以用课程ID作为主键，这样一对一或者一对多和多对多的设计就可以对该表进行关联。

优化设计2：教师姓名可以单独创建一张表，在创建一张教师与课程关系表，以及爬取的课程信息表，课程信息表中可能有多位教师授课，所以这里就要用到教师与课程关系表来进行关联，比如教师与课程关系表主键可以设置成课程表课程ID以及教师表中教师ID，这样就可以实现多对多的关系。