**第3章\_保存数据的完整性笔记**

**1. 为什么要保存数据的完整性**

之前我们讲过，用execl或者word软件也同样可以保存数据，但是很容易造成数据的冗余（重复），所以我们使用RDBMS来存储我们的数据

总之，我们在数据库里面存储数据的时候，也必须保证我们的数是准确和有效

有一个公式：精确性+可靠性=数据完整性

**2. 数据的完整性分别有哪些？**

数据完整性如果分类的话，大致可以分为4类：实体完整性，域完整性，引用完整性，自定义完整性

1. 实体完整性（域就是一行）

所谓实体完整性，就是指我们的每一个实体是要精确有效的

举例：

|  |  |  |
| --- | --- | --- |
| 姓名 | 性别 | 年龄 |
| 张三 | 男 | 20 |
| 张三 | 男 | 20 |

无法区分该张三是几个张三，所以这个时候需要我们在前面再添加一个字段，如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 年龄 |
| 1 | 张三 | 女 | 18 |
| 2 | 张三 | 女 | 18 |

这个唯一标识字段往往被称为主键

实现实体完整性的方法有两种：一是添加主键，二是添加唯一约束

1. 添加主键（主键 = 非空 + 唯一）

添加主键的关键字为 primary key

例如：create table 表名（

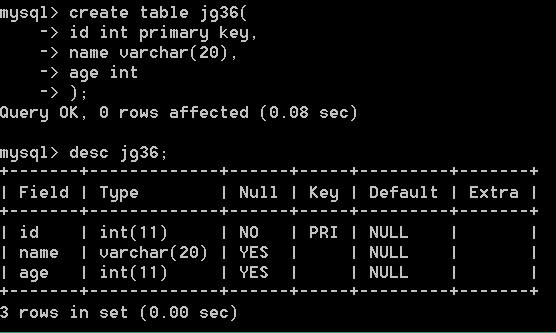
字段名 数据类型 primary key，

字段名 数据类型，

….

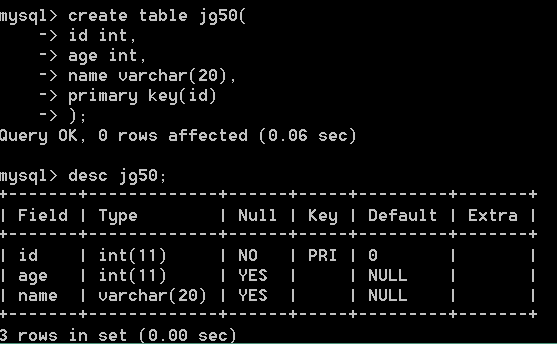
)；

举例：



还有一种方法，就是写完所有的字段之后，再写primary key（字段名）

举例：



2.唯一约束

关键字为unique

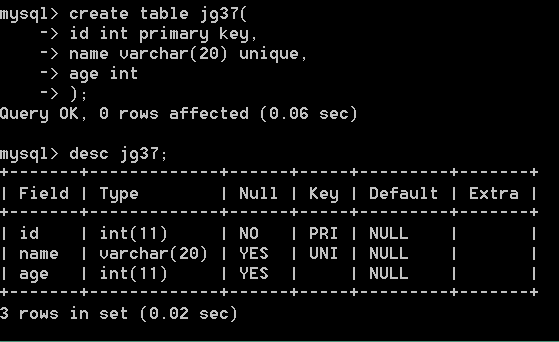
例如：create table 表名（

字段名 数据类型 primary key，

字段名 数据类型 unique，

….

)；



1. 域完整性（域就是一列）

所谓域完整性，就是指一列必须是完整。

实现域完整性的方法有 1. 数据类型 2.非空约束 3.默认约束 4. 检查约束（MySQL不支持检查约束）

1. 数据类型

我们在创建表的时候就创建了该字段的数据类型

比如说我们在创建表的时候，id这一字段的数据类型为int的话，那我们就不能插入int以外的数据，从而在一定程度上保证了我们域的完整性

1. 非空约束

所谓非空约束，就是指该字段上的数据不能为空。使用关键字not null（null就是‘空’的意思）

例如：create table 表名（

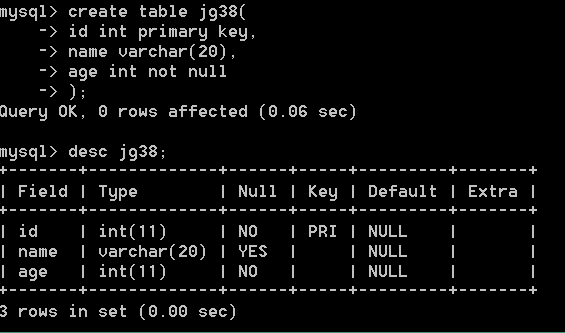
字段名 数据类型 primary key，

字段名 数据类型 not null，

….

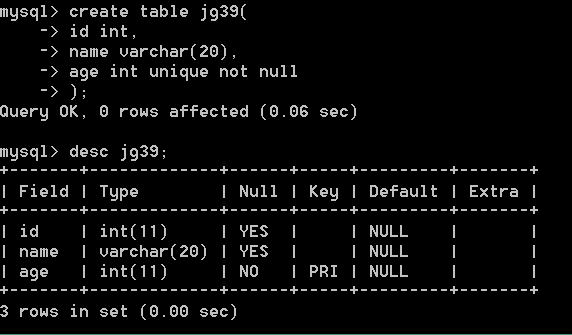
)；

举例：



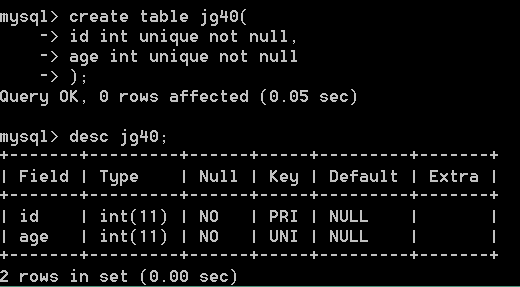
如果我们在一个字段上既设置非空约束又设置唯一约束，那么它变成主键

例如：



但是，如果我们在多个字段上都同时设置非空约束和唯一约束的话，那么只有第一个被设置的才会变成主键

例如：



1. 默认约束

理解起来也很简单，就是给该字段一个默认值，关键字为default

例如：create table 表名（

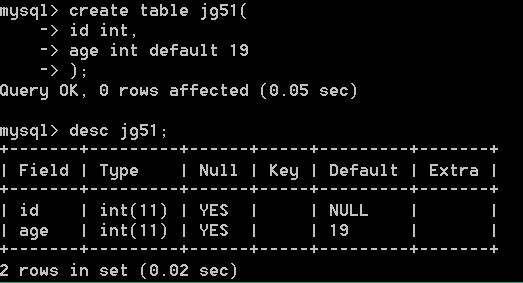
字段名 数据类型 primary key，

字段名 数据类型 default 默认值，

….

)；

举例：



1. 检查约束（MySQL不支持检查）

自己设定条件，然后当插入数据的时候会进行检查，看是否符合你自己设定的这个条件。使用关键字check来设置检查约束。

例如：create table 表名（

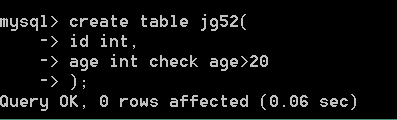
字段名 数据类型 primary key，

字段名 数据类型 check 条件设置，

….

)；

举例：



虽然这里建表成功了， 但是该检查约束并不会执行，当你在age字段插入小于20的数据时，也能插入成功。

1. 引用完整性

所谓引用完整性，确保表之间的关系不会被破坏。（后面将连接查询的时候会详细介绍， 这里只是简单提及一下）

1. 自定义完整性

所谓自定义完整性，就是用户自己来定义约束条件，就称之为自定义完整性

补充知识：

设置字段自动增加

如果我们不给该字段设置值，该字段就会自动增加。关键字为 auto\_increment（需要和主键配合使用）

例如：create table 表名（

字段名 数据类型 auto\_increment,

字段名 数据类型，

….

)；

举例：

