## 实验二 MySQL建表实验-电子杂志订阅表

一、实验目的

能根据文字提示，完成对应数据表的创建，并可以对数据表中的数据进行简单的增、删、改、查操作

二、实验需求

1.在mysql数据库中新建一张电子杂志订阅表（subscribe）

2.电子杂志订阅表中要包含4个字段，分别为编号（id）、订阅右键的邮箱地址（email）、用户是否确认订阅（status，使用数字1表示已确认，0表示未确认）、邮箱确认的验证码（code）。

3.为杂志订阅表添加5条测试数据，见课本中的表。

1. 实验要求

1.动手完成实验

2.完成实验报告

1. 实验过程及结果

**步骤1:**启动数据库，输入net start mysql 就能启动数据库。启动的结果如下图1所示。

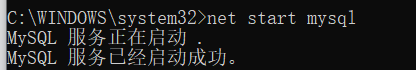


图1 数据库启动成功

**步骤2：**登录数据库，输入mysql -u root -p 就能进入数据库登录验证步骤，之后在其上输入登陆密码，能能够成功登入数据库。登陆成功截图如下图2所示。

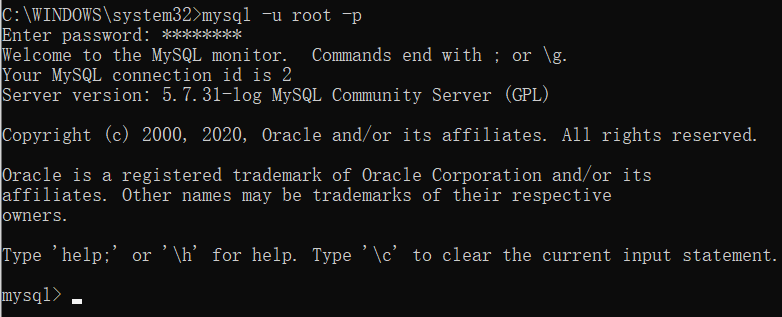


图2 登录数据库成功

**步骤3**：创建数据库，如果要打开的的数据库不存在，则需要创建。由于本人数据库并未创建过，因此没有现存数据库供选择，因此需要先创建数据库。创建的命令为：CREATE DATABASE mydb;但是考虑到如果本人忘记了，以前的确创建了数据库，而且名字也叫mydb，此时为了避免程序报错，需要做一个万全考虑，因此在创建的时候加一个判断mydb是否存在的语句，若不存在就创建mydb，命令为：CREATE DATABASE IF NOT EXISTS mydb;不报错表示已经成功。成功的截图如下图3所示。



图3 创建mydb数据库成功

**步骤4：**使用数据库。就是选择数据库到当前操作的对象。比如当数据库中有两个对象，一个是mydb和mydb1，如果当前是在mydb1的状态下，需要修改mydb的内容，就需要修改数据库状态。命令为USE + 数据库名。如下图4所示。提示Database changed表示切换成功。



图4 切换数据库成功

**步骤5：**创建电子杂志订阅表，此订阅表中包括四个字段。通过“CRAEATE TABLE+表格名称”的命令完成创建。格式为字段代表的含义、邮箱地址、状态（是否确认）、验证码，最后需要确定编码方式为UTF-8，否则有可能发生乱码。没有报错表示已经创建成功。当我继续创建subscribe为名称时发生了系统报错“Table 'subscribe' already exists”，这是由于我之前创建过subscribe表，因此在这次复现过程中，将该表格命名为subscribe1以示区别。创建成功的截图如下图5所示。

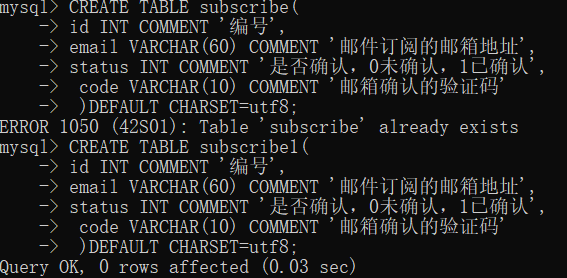


图5 表格创建成功

**步骤6：**向步骤5创建好的表格中添加数据，由于上述表格中规定了四个字段，因此添加数据的时候也需要对应着添加四个字段。添加的内容必须要和字段需要的类型一致，否则会报错。如图6是添加字段时，由于需要字段类型不一致报错的截图，步骤5中要求的是第三个字段必须是整型数字类型，而在实际输入中不是；第四个字段要求的是字符串类型，但是实际输入的并不是字符串（字符串要用英文引号”引起来），因此会图6的错误。若字段超出了关于规定的字段时，也会报错。如图7是添加的字段长度大于规定的字段长度时的报错。同理，当字段少于规定字段时也会报相同的错误。因此这一步要万分小心谨慎。最终没有报错表示输入成功，建表成功。如图8所示。

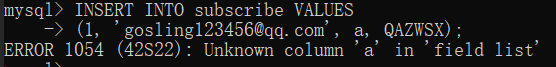


图6 添加字段类型不同时报错



图7 字段溢出报错

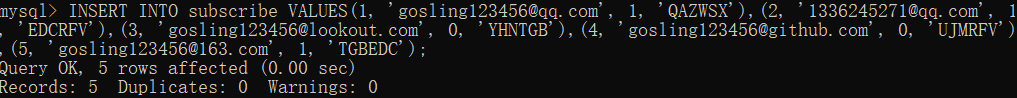


图8 向表格中添加数据成功

**步骤7：** 验证步骤6是否成功，可以通过查看表subscribe1的内容来完成。可以通过通配符\*来全选所有的数据。输入命令为：SELECT \* FROM subscribe1;结果如图9所示。从图表中能非常清晰的看出在步骤5创建的字段相当于第一行的所有列坐标。在步骤6中添加的数据是每一行数据，分别于步骤5的字段对应，规范整齐。

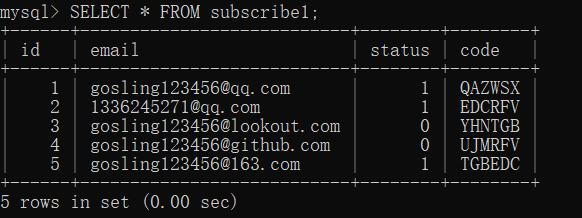


图9 数据添加成功后显示的表格

**步骤8：**实现查找的操作。在上述的表subscribe1中，已经有四个字段，因此可以通过四个字段和他们对应的值进行查找。其格式为“SELECT \* FROM + 表格名”。在这里，表二名为subscribe1。图10为按照status字段为1查找的订阅内容、图11为按照id字段为3查找的订阅内容、图12为按照email名称查找email名为[gosling123456@qq.com](mailto:gosling123456@qq.com)的订阅内容、图13为按照验证码code名称查找code名为EDCRFV的订阅内容。



图10 按照status查找status为1的订阅内容



图11 按照id查找id为3的订阅内容

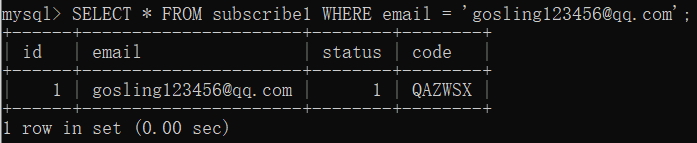


图12 [按照email名称查找email名为gosling23456@qq.com](mailto:按照email名称查找email名为gosling23456@qq.com)的内容

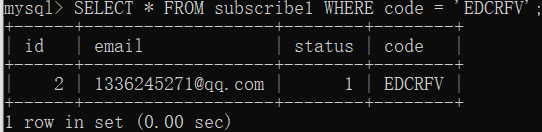


图13 按照code名称查找code为EDCRFV的订阅内容

**步骤9：**实现修改操作。在输入时有个在email字段下的订阅邮箱输入错了，其原本的形式应该是’[gosling123456@outlook.com](mailto:gosling123456@outlook.com)’，但是在第6步，向建好的表里面添加字段的时候输入成’[gosling123456@lookout.com](mailto:gosling123456@lookout.com)’，因此需要将其改正过来。因此需要选择数据库中的UPDATA关键词命令。输入UPDATE subscribe1 SET +“需要修改的数据属于的字段”= +“要修改为的字段的内容” + WHERE + “主键的字段” = + “需要修改的对象主键字段的内容”。例如需要将email字段内容为’[gosling123456@lookout.com](mailto:gosling123456@lookout.com)’的订阅邮箱修改为’[gosling123456@outlook.com](mailto:gosling123456@outlook.com)’的操作为：UPDATE subscribe1 SET email = 'gosling123456@outlook.com' WHERE email = 'gosling123456@lookout.com'。修改前后的截图如下图14和图15所示，其中图14为修改前的subscribe1表，图15为修改后的subscribe1表。



图14修改前的subscribe1表

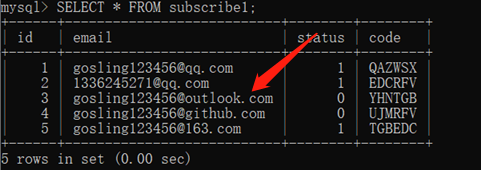


图15 修改后的subscribe1表

**步骤10：** 实现删除操作。删除的命令为DELETE FROM + 表名 + 对应数据的相应字段和内容。例如如果要删除subscribe1表中的第5个元素，命令为DELETE FROM subscribe1 WHERE id = 5这样就能删除掉该组数据，结果的截图如图16所示; 如果要删除subscribe1表中的email为’1336245271@qq.com’的元素，命令为DELETE FROM subscribe1 WHERE email = ’[1336245271@qq.com](mailto:1336245271@qq.com)’、这样就能删除掉该组数据，结果的截图如图17所示; 如果要删除subscribe1表中的status为1的元素，命令为DELETE FROM subscribe1 WHERE status = 1,此时所有status为1的元素都会被删除这样就能删除掉该组数据，结果的截图如图18所示；如果要删除subscribe1表中的code为QAZWSX的元素，命令为DELETE FROM subscribe1 WHERE code = QAZWSX这样就能删除掉该组数据，结果的截图如图19所示;



图16 删除id为5的结果

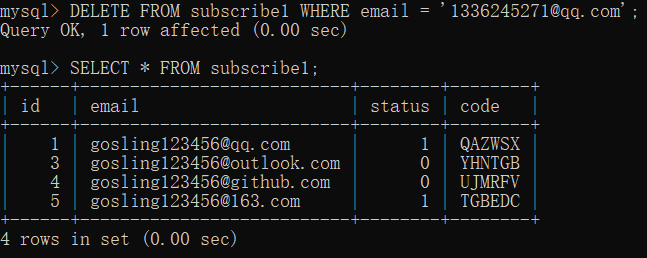


图17 删除email为'1336245271@qq.com'的结果



图18 删除status为1的结果

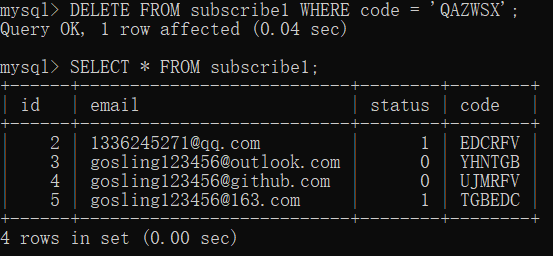


图19 删除code为QAZWSX的结果

**步骤11：**扩展操作，实现数据库中特定关键字的排序，升序和降序。排序是利用SORT关键词命令来实现的，其格式为：SORT \* FROM + 表名 ORDER BY + 关键字段 + （升序asc或降序desc）。下图分别是通过对id、状态排序（升序和降序的操作图和操作后的表）。

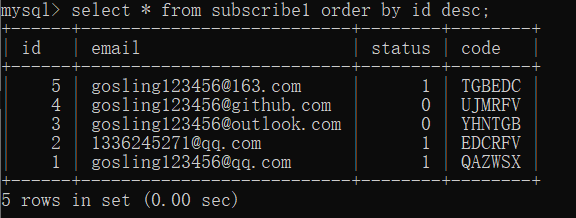


图20 id降序排序结果

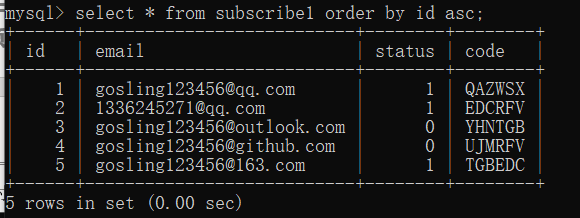


图21 id升序排序结果

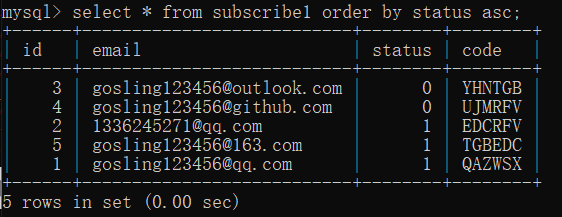


图22 status升序排序结果

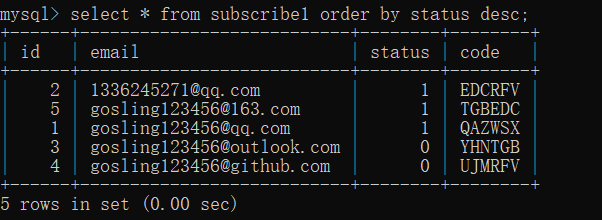


图23 status降序排序结果

1. 实验总结

本次实验我学会了数据库的创建、添加、删除、修改、查找和排序。本次实验主要是按照书上的步骤一个一个地敲命令，过程虽然有些繁琐，甚至还有一个步骤出现好多错误然后翻来覆去反反复复的修改，不过幸运的是，总算把所有功能都实现了，甚至还在查找资料的过程中发现了一个新的操作——排序，以及其实现方法。在敲命令的过程中，我越发觉得这和书上的内容遥相呼应，在我写到设置四个字段的时候，我想起来第一节理论课上的excel表的表头，当我写到SELECT \* FROM WHERE 的时候，我想到可以通过控制关键字的值（比如控制status为1）来控制视图等等，通过实验，我发现理论与实践结合的越发密切了。

虽说本次实验最终是完成了，但是一路上还是花了不少功夫的。虽然是对着课本内容敲命令，但是所谓学而不思则罔，我的目标不只是为了完成作业，而是要通过完成作业来夯实数据库的基础。在写命令的过程中，我明白了，数据库的语法规则，基本上是靠关键字来实现的，例如创建数据库就是CREATE，添加元素就是INSERT，删除元素就是DELETE，修改元素就是UPDATE，查找元素就是SELECT，其后缀都是用来确定元素的位置，以便于操作。在敲命令期间，我忘记开启大写锁定，但是写的命令还是通过了。于是我明白了，数据库的操作命令是不区分大小写的，那以后在写数据库命令的时候就不需要来回切换大小写了，也算是节省了精力。此外，我在敲命令的时候发现了\*的作用，我记得在python语言里面，\*的作用是通配符，在数据库中，\*代表的是全部的意思，也算是通配符的含义。

通过本次实验，我明白了数据库的学习在于多看多学多想多动手，要将理论与实践相结合才能学好数据库，真正将数据库变成自己最熟悉的工具。