# Mysql终端操作

查看版本：登入后，select version()

8.0.17

！！！如果没有设置环境变量，那么需要到bin下面登入

# 忘记密码

第一步，关掉系统服务

net stop mysql

第二步，创建一个文本文件，内含一条密码修改命令

ALTER USER 'root'@'localhost' IDENTIFIED BY '';

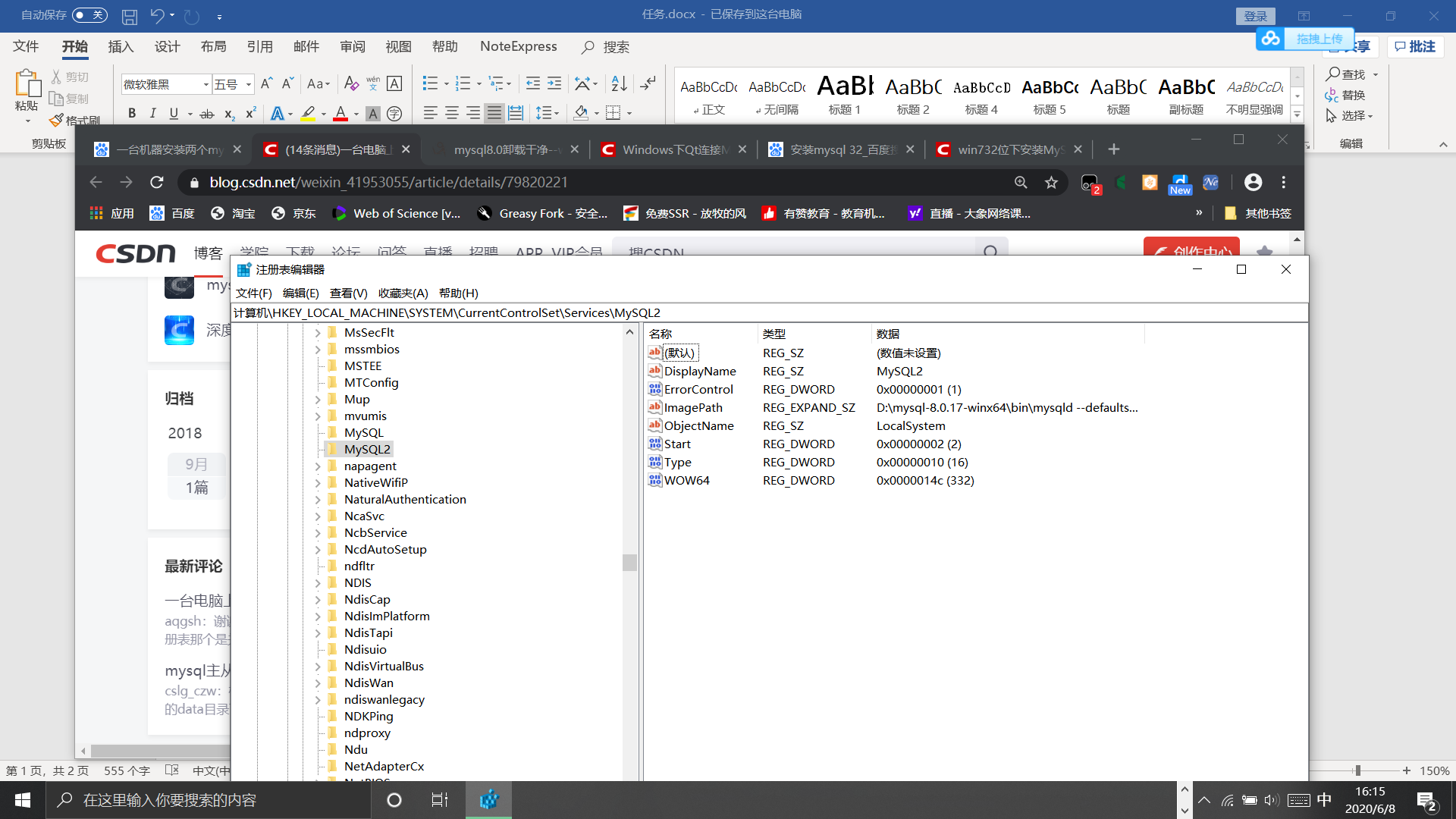
第三步：命令行方式启动服务器，指定启动时执行上述的密码修改命令文件

mysqld --init-file=d:mysqlc.txt --console

# 删除mysql

在cmd中，输入sc delete mysql，删除服务

清除注册表中的mysql记录



# 允许非本机用户访问

第二步：开放mysql的全向

打开mysql的命令窗口，修改访问权限为all，即set host='%' where host= 'localhost'

# 安装第二mysql，登入

原来的mysql是64位的，只能再安装一个32位的5.7.25

D:\mysql-5.7.25-win32\bin

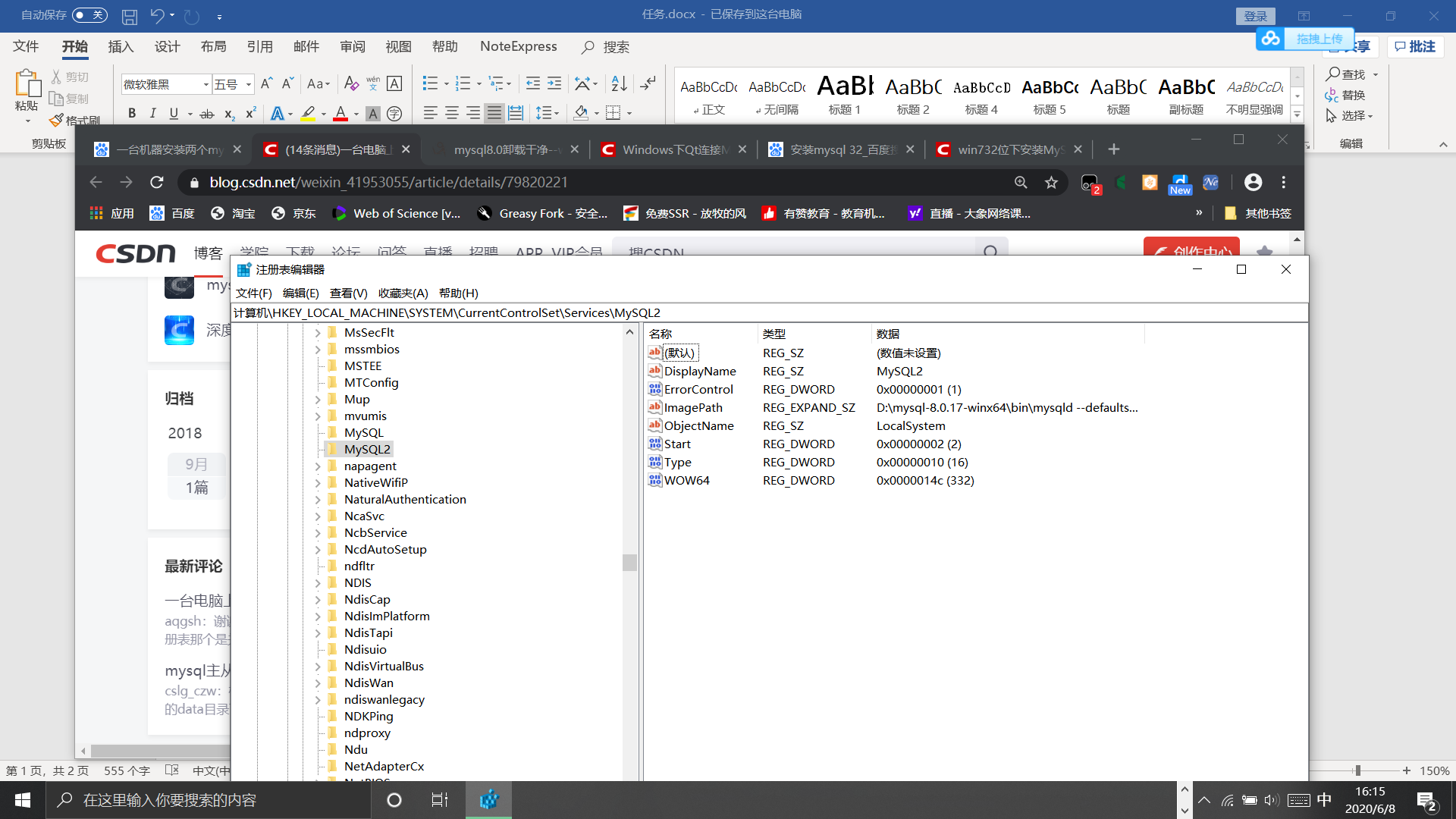
初始化mysql，使用命令

mysqld --defaults-file= D:\mysql-5.7.25-win32\my.ini --initialize --console

安装mysql2

mysqld install MySQL2 --defaults-file= D:\mysql-5.7.25-win32\my.ini

设置端口为3307安装成功



更改imagePath的值，修改default前的路径

这个方法要是不成功就官网下载msi进行，安装后登入

mysql -P 3307 -u root -p

查看端口号

mysql> show global variables like 'port';

# <https://www.runoob.com/php>

开启服务器net start sql

登录mysql：mysql -u root -p

退出mysql: quit

关闭mysql: net stop mysql

show databases;

use databasesName;

show tables

select \* from tableName;

create database databaseName;

DESCRIBE user;

create TABLE pet(

                   name VARCHAR(20),

                   owner VARCHAR(20),

                   specise VARCHAR(20),

                   sex CHAR(1),

                   brith DATE,

                   death DATE );

INSERT INTO pet VALUES('kk','cc','dog','1','1998-8-2',null);

DELETE FROM tablesName WHRER 条件;

UPDATE tableName SET 字段1=值1,字段2=值2 ... WHERE 条件;

## 1.主键约束:

   它能够**唯一确定**一张表中的一条记录,增加主键约束之后,就可以使得字段不重复而且不为空

create table user(

    id int PRIMARY KEY,

    name VARCHAR(20)

);

INSERT INTO user VALUES (1,'张三');

## 2.复合主键

CREATE TABLE user2(

    id INT,

    name VARCHAR(20),

    password VARCHAR(20),

    PRIMARY key(id,name)

);

INSERT INTO user2 VALUES (1,'老李','123456');

## 3.自增约束:

没有自定义id值 但是自动生成了id

CREATE TABLE user3(

    id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(20)

);

INSERT INTO user3(name) VALUES('张三');

## 4.唯一约束:

CREATE TABLE user5(

    id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(20)

);

新增name为唯一约束:

ALTER TABLE user5 ADD UNIQUE(name);

INSERT INTO user5(name) VALUES ('cc');

## 5.非空约束:

INSERT INTO user\_test (name,phone\_number) VALUES ('张三','12345678901');

## 6.默认约束

CREATE TABLE user6(

   id int PRIMARY KEY AUTO\_INCREMENT COMMENT'主键id',

    name VARCHAR(20) NOT NULL COMMENT'用户姓名不能为空',

    phone\_number VARCHAR(20) NOT NULL COMMENT'用户手机号,不能为空',

    status INT DEFAULT 0 COMMENT'用户状态0:启用 1:禁封 默认:0'

);

UPDATE user6 SET status = 1 WHERE phone\_number= '1234';

SELECT \* FROM user6;

## 7.外键约束

CREATE TABLE classes(

    id INT PRIMARY KEY AUTO\_INCREMENT COMMENT'班级表id',

    name VARCHAR(20) COMMENT'班级名称'

);

CREATE TABLE student(

   id INT PRIMARY KEY AUTO\_INCREMENT COMMENT'学生表id',

   name VARCHAR(20) COMMENT'学生姓名',

    class\_id int COMMENT'教室id,这张表中的class\_id是classes表中id的值',

    FOREIGN KEY (class\_id) REFERENCES classes(id)

);

## 8.如何建表之后添加主键约束

CREATE TABLE user4(

    id INT,

    name VARCHAR(20)

);

ALTER TABLE `表名` CHANGE id id int NOT NULL AUTO\_INCREMENT PRIMARY KEY;

alter table 表名 add primary key (字段名);  
例子： mysql> alter table employee add primary key(id);

 mysql> alter table 表名 add unique 索引名 (字段名);

例子： mysql> alter table employee add unique emp\_name2(cardnumber);

4) 删除某个索引  
   mysql> alter table 表名 drop index 索引名;  
  
例子： mysql>alter table employee drop index emp\_name;  
  
5) 增加字段  
    mysql> ALTER TABLE table\_name ADD field\_name field\_type;  
  
6) 修改原字段名称及类型  
    mysql> ALTER TABLE table\_name CHANGE old\_field\_name new\_field\_name field\_type;  
  
7) 删除字段  
    MySQL ALTER TABLE table\_name DROP field\_name;

# Php对mysql操作（mysqli）

## 连接数据库，在网页显示数据库内容

$link=mysqli\_connect("localhost","root","074024") or die("Cannot Connect to the database!");

mysqli\_select\_db($link,"department") or die ("Cannot select the database!");

$query="SELECT \* FROM students";

$resource=mysqli\_query($link,$query);

echo "

while($result=mysqli\_fetch\_array($resource))

{

echo "<tr><td>".$result[1]."</td><td>".$result[2]."</td><td>".$result[3]."</td><td>".$result[4]."</td></tr>";

} echo "</table>";

## 新建数据库

$sql = "create database stu\_01 default character set = 'utf8' ";

if(mysqli\_query($link,$query) === TRUE){

echo "数据库创建成功.";

}else {

echo "数据库创建失败 ，错误信息为:" . $link->error; // 如果失败输出错误信息

}

## 新建数据表

**面向对象编程**

<?php

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "myDB";

// 创建连接

$conn = new mysqli($servername, $username, $password, $dbname);

// 检测连接 if ($conn->connect\_error) { die("连接失败: " . $conn->connect\_error); } // 使用 sql 创建数据表

$sql = "CREATE TABLE MyGuests (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

firstname VARCHAR(30) NOT NULL, lastname VARCHAR(30) NOT NULL, email VARCHAR(50),

reg\_date TIMESTAMP )";

if ($conn->query($sql) === TRUE) { echo "Table MyGuests created successfully"; }

else { echo "创建数据表错误: " . $conn->error; } $conn->close();

?>

面向过程编程

<?php

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "myDB";

// 创建连接

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// 检测连接

if (!$conn) { die("连接失败: " . mysqli\_connect\_error()); }

// 使用 sql 创建数据表

$sql = "CREATE TABLE MyGuests (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

firstname VARCHAR(30) NOT NULL, lastname VARCHAR(30) NOT NULL,

email VARCHAR(50), reg\_date TIMESTAMP )";

if (mysqli\_query($conn, $sql)) { echo "数据表 MyGuests 创建成功"; }

else { echo "创建数据表错误: " . mysqli\_error($conn); } mysqli\_close($conn);

?>

## 表中插入数据

<?php

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "myDB";

// 创建连接

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// 检测连接

if (!$conn) { die("Connection failed: " . mysqli\_connect\_error()); }

$sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli\_query($conn, $sql)) { echo "新记录插入成功"; }

else { echo "Error: " . $sql . "<br>" . mysqli\_error($conn); }

mysqli\_close($conn);

?>

删除数据

$link=mysql\_connect("localhost","root","") or die("Cannot Connect to the database!");

mysql\_select\_db("department",$link) or die ("Cannot select the database!");

$query="DELETE FROM students WHERE id='".$id."'";

if(!mysql\_query($query,$link))

{die ("An unexpected error occured while <b>deleting</b> the record, Please try again!");}

else

{

echo "Record ".$id." removed successfully!";}

# QT操作mysql

## 连接mysql测试：

Pro中要加QT +=sql

#include <QSqlDatabase>

#include <QSqlQuery>

#include <QDebug>

QSqlDatabase db = QSqlDatabase::addDatabase("QMYSQL");

db.setHostName("localhost");

db.setPort(3307);

db.setDatabaseName("Qtest"); //这里输入你的数据库名，是已经存在的

db.setUserName("root");

db.setPassword("074024"); //这里输入你的密码

if (!db.open()) {

qDebug() << "连接数据库失败";

return false;

}

else{

qDebug() << "连接数据库成功";

}

// addDatabase("QMYSQL");表示我们需要使用的是哪种数据库

根据条件查询

//查询宠物信息

QString searchname = ui->SearchText->toPlainText();

qDebug()<<searchname;

QString str = QString("select \*from pet where name = '%1'").arg(searchname);

QSqlQuery query;

query.exec(str);

QString name=NULL;

QString owner;

int gender;

int i=0;

while (query.next()){

name = query.value(0).toString();

owner = query.value(1).toString();

gender= query.value(3).toInt();

QString str = QString("宠物名：%1 主人：%2 性别：%3").arg(name).arg(owner).arg(gender);

ui->ShowText->setText(str);

i++;

};

if(name == NULL)

{

QString a = QString("没有叫%1的宠物，请重新输入人名").arg(searchname);

ui-> SearchText->clear();

ui->ShowText->setText(a);

}

//查看全表

void MainWindow::**on\_ShowAll\_clicked**()

{

ui->ShowText->clear();

//查看全表

QString str = QString("select \*from pet");

QSqlQuery query;

query.exec(str);

QString name=NULL;

QString owner;

int gender;

int i=0;

while (query.next()){

name = query.value(0).toString();

owner = query.value(1).toString();

gender= query.value(3).toInt();

QString str = QString("宠物名：%1 主人：%2 性别：%3").arg(name).arg(owner).arg(gender);

ui->ShowText->append(str);

i++;

};

if(name == NULL)

{

QString a = QString("暂时没有数据记录");

ui-> SearchText->clear();

ui->ShowText->setText(a);

}

}

## 增删数据

<https://blog.csdn.net/kilotwo/article/details/79248657>

用QSqlTableModel

model = new QSqlTableModel(this);

model->setTable("course");

model->select();

在text显示：

ui->tableView->setModel(model);