# **Photon Server游戏服务器教程（三）MySQL数据库基础及数据操作**

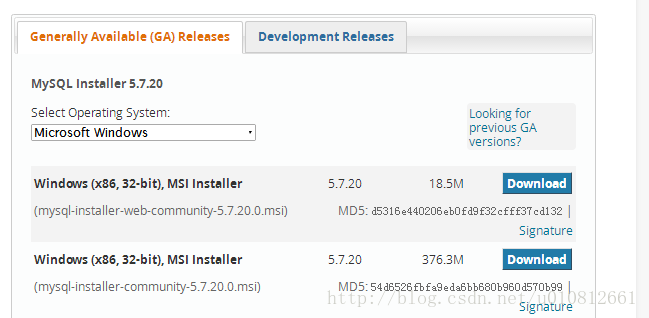
**概述**

MySQL是一个开放源码的小型关联式数据库管理系统，开发者为瑞典MySQL AB公司。目前MySQL被广泛地应用在Internet上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库。

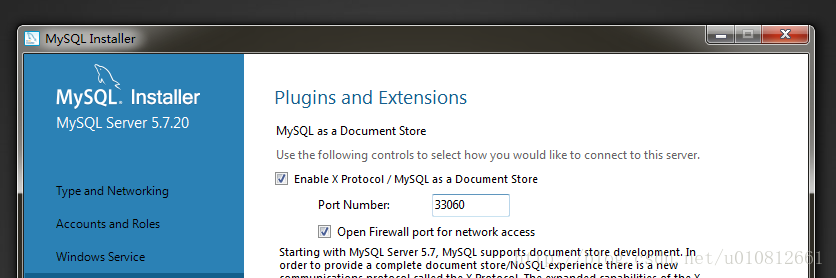
MySQL是一种关系型数据库，即建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示。关系模型是由埃德加·科德于1970年首先提出的，并配合”科德十二定律”。现如今虽然对此模型有一些批评意见，但它还是数据存储的传统标准。标准数据查询语言SQL就是一种基于关系数据库的语言，这种语言执行对关系数据库中数据的检索和操作。 关系模型由关系数据结构、关系操作集合、关系完整性约束三部分组成。

**一、MySQL下载与安装**

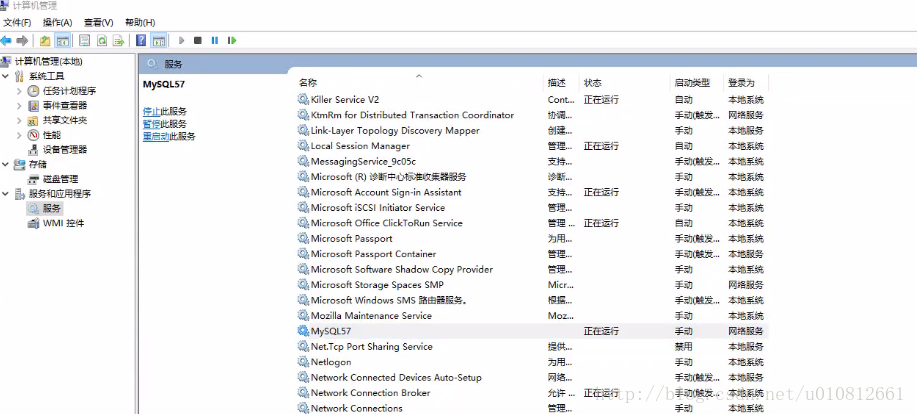
官网 https://dev.mysql.com/downloads/installer/ ，点击Download下载即可。



下载完成后点击安装，其中只需设置超级管理员root密码，勾选Plugins and Extensions 项，其它按默认设置即可。

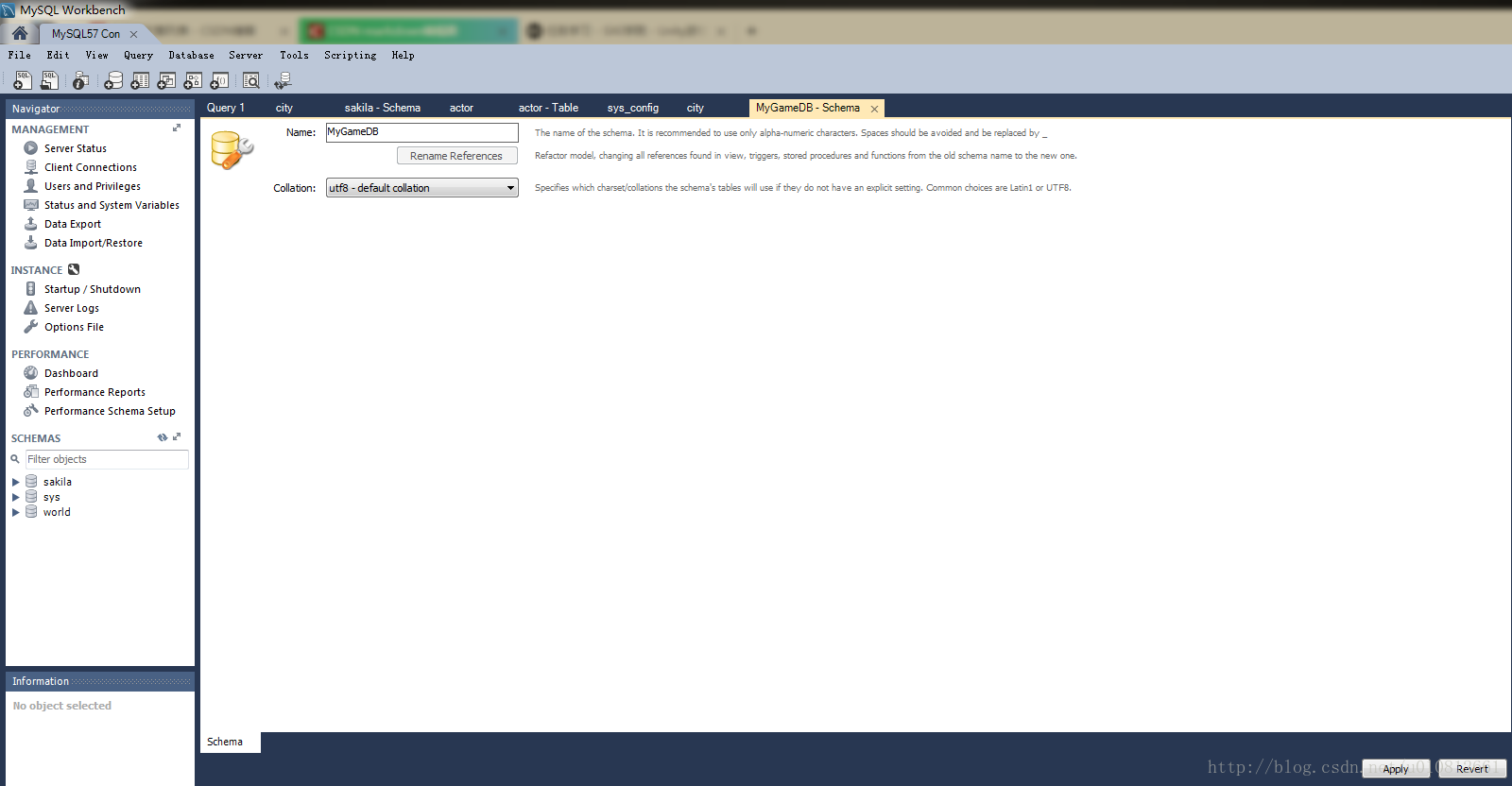


启动MySQL，点击我的电脑>管理>服务和应用>服务，找到MySQL57右键启动。

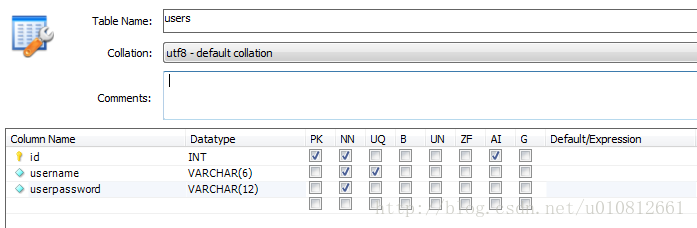


**二、查看和创建数据库**

在MySQL中可以通过MySQL Workbench可视化编辑界面来查看和创建数据库。在这里我们点击新建数据库图标创建一个可支持中文Name为MyGameDB（不区分大小写），collation为utf8的数据库。



创建user表



**三、数据类型**

1、整型

|  |  |
| --- | --- |
| MySQL数据类型 | 含义（有符号） |
| tinyint(m) | 1个字节 范围(-128~127) |
| smallint(m) | 2个字节 范围(-32768~32767) |
| mediumint(m) | 3个字节 范围(-8388608~8388607) |
| int(m) | 4个字节 范围(-2147483648~2147483647) |
| bigint(m) | 8个字节 范围(+-9.22\*10的18次方) |

2、浮点型(float和double)

|  |  |
| --- | --- |
| MySQL数据类型 | 含义 |
| float(m,d) | 单精度浮点型 8位精度(4字节) m总个数，d小数位 |
| double(m,d) | 双精度浮点型 16位精度(8字节) m总个数，d小数位 |

3、字符串(char,varchar,\_text)

|  |  |
| --- | --- |
| MySQL数据类型 | 含义 |
| char(n) | 固定长度，最多255个字符 |
| varchar(n) | 固定长度，最多65535个字符 |
| tinytext | 可变长度，最多255个字符 |
| text | 可变长度，最多65535个字符 |
| mediumtext | 可变长度，最多2的24次方-1个字符 |
| longtext | 可变长度，最多2的32次方-1个字符 |

4、二进制数据(\_Blob)

.\_BLOB和\_text存储方式不同，\_TEXT以文本方式存储，英文存储区分大小写，而\_Blob是以二进制方式存储，不分大小写。

.\_BLOB存储的数据只能整体读出。

.\_TEXT可以指定字符集，\_BLO不用指定字符集。

5、日期时间类型

|  |  |
| --- | --- |
| MySQL数据类型 | 含义 |
| date | 日期‘2008-12-2’ |
| time | 时间‘12:25:36’ |
| datetime | 日期时间‘2008-12-2 22:06:44’ |
| timestamp | 自动存储记录修改时间 |

若定义一个字段为timestamp，这个字段里的时间数据会随其他字段修改的时候自动刷新，所以这个数据类型的字段可以存放这条记录最后被修改的时间。

数据类型的属性

|  |  |
| --- | --- |
| MySQL关键字 | 含义 |
| NULL | 数据列可包含NULL值 |
| NOT NULL | 数据列不允许包含NULL值 |
| DEFAULT | 默认值 |
| PRIMARY KEY | 主键 |
| AUTO\_INCREMENT | 自动递增，适用于整数类型 |
| UNSIGNED | 无符号 |
| CHARACTER SET name | 指定一个字符集 |

**四、一些常用的SQL语句**

选择：select \* from table1 where 范围

插入：insert into table1(field1,field2) values(value1,value2)

删除：delete from table1 where 范围

更新：update table1 set field1=value1 where 范围

查找：select \* from table1 where field1 like ’%value1%’ ---like的语法很精妙，查资料!

排序：select \* from table1 order by field1,field2 [desc]

总数：select count as totalcount from table1

求和：select sum(field1) as sumvalue from table1

平均：select avg(field1) as avgvalue from table1

最大：select max(field1) as maxvalue from table1

最小：select min(field1) as minvalue from table1

**五、数据操作**

1.在vs中对数据库的操作需添加对MySQL.Data的引用

2.与数据库建立连接

static void Main(string[] args) {

string connectStr = "server=127.0.0.1;port=3306;database=mygamedb;user=root;password=root;";

MySqlConnection conn = new MySqlConnection(connectStr);

try {

conn.Open();

//在这里执行其它操作

// Console.WriteLine("已经建立连接");

} catch (Exception e) {

Console.WriteLine(e.ToString());

}

finally {

conn.Close();

}

}

}

3.执行查询命令MySQLDataReader

string sql = "select \* from users";

MySqlConnection conn = new MySqlConnection(connectStr);

MySqlCommand cmd = new MySqlCommand(sql, conn);

//cmd.ExecuteReader();//执行一些查询

//cmd.ExecuteNonQuery();//插入 删除

//cmd.ExecuteScalar();//执行一些查询，返回一个单个的值

MySqlDataReader reader = cmd.ExecuteReader();

//reader.Read();//读取下一页数据，如果读取成功，返回true，如果没有下一页了，读取失败的话，返回false

while (reader.Read()){ }

//Console.WriteLine(reader[0].ToString() + reader[1].ToString() + reader[2].ToString());

//Console.WriteLine(reader.GetInt32(0) + " " + reader.GetString(1) + " " + reader.GetString(2));

Console.WriteLine(reader.GetInt32("id") + " " + reader.GetString("username") + " " + reader.GetString("password"));

4.对数据进行插入Insert操作

string sql = "insert into users(username,password,registerdate) values('csdFu','234','" + DateTime.Now + "')";

MySqlCommand cmd = new MySqlCommand(sql, conn);

int result = cmd.ExecuteNonQuery();//返回值是数据库中受影响的数据的行数

5.对数据进行删除Delete操作

string sql = "delete from users where id = 4";

MySqlCommand cmd = new MySqlCommand(sql, conn);

int result = cmd.ExecuteNonQuery();//返回值是数据库中受影响的数据的行数

1. 对数据进行更新Update

string sql = "update users set username='sdfsedfwer',password='23242342432' where id = 4";

MySqlCommand cmd = new MySqlCommand(sql, conn);

int result = cmd.ExecuteNonQuery();//返回值是数据库中受影响的数据的行数

7.QL中的一些函数介绍

SQL 语句学习网站 http://www.w3school.com.cn/sql/

//COUNT() 函数返回匹配指定条件的行数。

string sql = "select count(\*) from users";

MySqlCommand cmd = new MySqlCommand(sql, conn);

object o = cmd.ExecuteScalar();

int count = Convert.ToInt32(o.ToString());

Console.WriteLine(count);

//AVG 函数返回数值列的平均值。NULL 值不包括在计算中。

//FIRST() 函数返回指定的字段中第一个记录的值。

//提示：可使用 ORDER BY 语句对记录进行排序。

8.在查询的时候添加参数

static bool VerifyUser(string username, string password)

{

string connectStr = "server=127.0.0.1;port=3306;database=mygamedb;user=root;password=root;";

MySqlConnection conn = new MySqlConnection(connectStr);//并没有去跟数据库建立连接

try {

conn.Open();

//string sql = "select \* from users where username = '"+username+"' and password='"+password+"'";

//我们自己按照查询条件去组拼sql，当参数多时容易出错。

string sql = "select \* from users where username =@para1 and password = @para2 ";

MySqlCommand cmd = new MySqlCommand(sql, conn);

cmd.Parameters.AddWithValue("para1", username);

cmd.Parameters.AddWithValue("para2", password);

MySqlDataReader reader = cmd.ExecuteReader();

if (reader.Read()) return true;

}

catch (Exception e){

Console.WriteLine(e.ToString());

}

finally

{

conn.Close();

}

return false;

}

总结：以上是使用sql直接对数据库的增删改查操作，下一章我们将学习如何使用NHibernate对数据库的操作。

来自：https://blog.csdn.net/u010812661/article/details/78740337