

MySQL

什么是MySQL?

MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。

什么是数据库?

数据库 (Database) 是按照数据结构来组织、存储和管理数据的仓库。

每个数据库都有一个或多个不同的 API 用于创建，访问，管理，搜索和复制所保存的数据。

我们也可以将数据存储于文件中，但是在文件中读写数据速度相对较慢。

所以，现在我们使用关系型数据库管理系统 (RDBMS) 来存储和管理大数据量。所谓的关系型数据库，是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。

RDBMS 即关系数据库管理系统(Relational Database Management System)的特点：

- 1.数据以表格的形式出现
- 2.每行为各种记录名称
- 3.每列为记录名称所对应的数据域
- 4.许多的行和列组成一张表单
- 5.若干的表单组成database

MySQL数据库

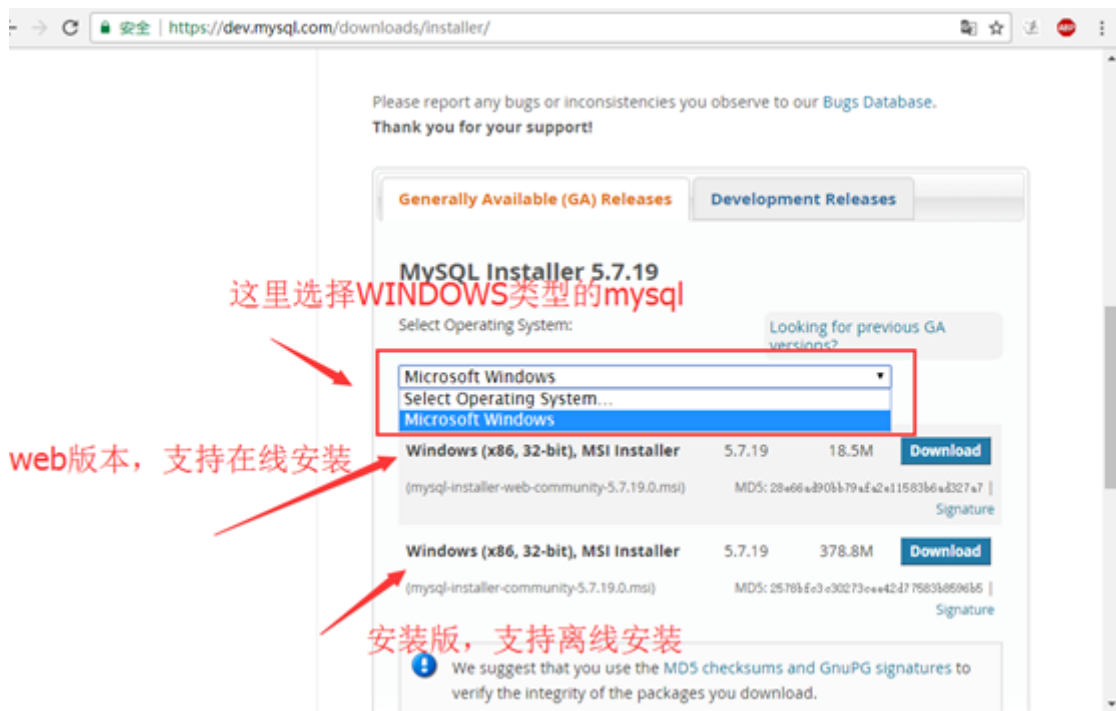
MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

- MySQL 是开源的，目前隶属于 Oracle 旗下产品。
- MySQL 支持大型的数据库。可以处理拥有上千万条记录的大型数据库。
- MySQL 使用标准的 SQL 数据语言形式。
- MySQL 可以运行于多个系统上，并且支持多种语言。这些编程语言包括 C、C++、Python、Java、Perl、PHP、Eiffel、Ruby 和 Tcl 等。
- MySQL 对PHP有很好的支持，PHP 是目前最流行的 Web 开发语言。
- MySQL 支持大型数据库，支持 5000 万条记录的数据仓库，32 位系统表文件最大可支持 4GB，64 位系统支持最大的表文件为8TB。
- MySQL 是可以定制的，采用了 GPL 协议，你可以修改源码来开发自己的 MySQL 系统。

MySQL 安装

下载地址为：[MySQL下载](#)。

Windows 上安装 MySQL



下载完后，我们将 zip 包解压到相应的目录，这里我将解压后的文件夹放在 **C:\web\mysql-8.0.11** 下。

接下来我们需要配置下 MySQL 的配置文件

打开刚刚解压的文件夹 **C:\web\mysql-8.0.11**，在该文件夹下创建 **my.ini** 配置文件，编辑 **my.ini** 配置以下基本信息：

```
[client]
# 设置mysql客户端默认字符集
default-character-set=utf8

[mysqld]
# 设置3306端口
port = 3306
# 设置mysql的安装目录
basedir=C:\\web\\mysql-8.0.11
# 设置 mysql数据库的数据的存放目录，MySQL 8+ 不需要以下配置，系统自己生成即可，否则有可能报错
# datadir=C:\\web\\sqldata
# 允许最大连接数
max_connections=20
# 服务端使用的字符集默认为8比特编码的latin1字符集
character-set-server=utf8
# 创建新表时将使用的默认存储引擎
default-storage-engine=INNODB
```

接下来我们来启动下 MySQL 数据库：

配置环境变量，将mysql/bin加入到PATH中

MYSQL_HOME: C:\software\mysql-8.0.23-winx64

Path: C:\software\mysql-8.0.23-winx64\bin

以管理员身份打开 cmd 命令行工具，切换目录：

```
cd C:\web\mysql-8.0.11\bin
```

输入以下安装命令：

```
mysqld install
```

初始化数据库：

```
mysqld --initialize --console
```

执行完成后，会输出 root 用户的初始默认密码，如：

```
...
2018-04-20T02:35:05.464644Z 5 [Note] [MY-010454] [Server] A temporary password is
generated for root@localhost: APWCY5ws&hjQ
...
```

APWCY5ws&hjQ 就是初始密码，后续登录需要用到，你也可以在登陆后修改密码。

启动输入以下命令即可：

```
启动: net start mysql
关闭: net stop mysql
```

注意: 在 5.7 需要初始化 data 目录：

```
cd C:\web\mysql-8.0.11\bin
mysqld --initialize-insecure
```

初始化后再运行 net start mysql 即可启动 mysql。

登录 MySQL

当 MySQL 服务已经运行时，我们可以通过 MySQL 自带的客户端工具登录到 MySQL 数据库中，首先打开命令提示符，输入以下格式的命名：

```
mysql -h 主机名 -u 用户名 -p
```

参数说明：

- **-h**：指定客户端所要登录的 MySQL 主机名，登录本机(localhost 或 127.0.0.1)该参数可以省略；
- **-u**：登录的用户名；
- **-p**：告诉服务器将会使用一个密码来登录，如果所要登录的用户名密码为空，可以忽略此选项。

如果我们要登录本机的 MySQL 数据库，只需要输入以下命令即可：

```
mysql -u root -p
```

按回车确认，如果安装正确且 MySQL 正在运行，会得到以下响应：

```
Enter password:
```

若密码存在, 输入密码登录, 不存在则直接按回车登录。登录成功后你将会看到 Welcome to the MySQL monitor... 的提示语。

然后命令提示符会一直以 **mysql>** 加一个闪烁的光标等待命令的输入, 输入 **exit** 或 **quit** 退出登录。

首先安装后, 执行任何指令都会提示:

```
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
```

可以用以下指令修改你密码为 123456。

```
ALTER USER 'root'@'localhost' IDENTIFIED BY '123456' PASSWORD EXPIRE NEVER;
```

之后使用以下指令刷新权限:

```
flush privileges;
```

注意指令末尾的分号。

退出后重新登陆。

安装Navicat (key)

连接 MySQL

mysql 报错Error: ER_NOT_SUPPORTED_AUTH_MODE: Client does not support authentication protocol requested by server; consider upgrading MySQL client

起因: mysql8.0加密方式的原因报错。

解决办法:

执行指令

```
mysql -u root -p
```

```
123456
```

```
use mysql;
```

```
alter user 'root'@'localhost' identified with mysql_native_password by '123456';
```

```
flush privileges;
```

MySQL 数据类型

MySQL中定义数据字段的类型对你数据库的优化是非常重要的。

MySQL支持多种类型, 大致可以分为三类: 数值、日期/时间和字符串(字符)类型。

数值类型

MySQL支持所有标准SQL数值数据类型。

这些类型包括严格数值数据类型(INTEGER、SMALLINT、DECIMAL和NUMERIC)，以及近似数值数据类型(FLOAT、REAL和DOUBLE PRECISION)。

关键字INT是INTEGER的同义词，关键字DEC是DECIMAL的同义词。

BIT数据类型保存位字段值，并且支持MyISAM、MEMORY、InnoDB和BDB表。

作为SQL标准的扩展，MySQL也支持整数类型TINYINT、MEDIUMINT和BIGINT。下面的表显示了需要的每个整数类型的存储和范围。

类型	大小	范围（有符号）	范围（无符号）	用途
TINYINT	1 byte	(-128, 127)	(0, 255)	小整数值
SMALLINT	2 bytes	(-32 768, 32 767)	(0, 65 535)	大整数值
MEDIUMINT	3 bytes	(-8 388 608, 8 388 607)	(0, 16 777 215)	大整数值
INT或 INTEGER	4 bytes	(-2 147 483 648, 2 147 483 647)	(0, 4 294 967 295)	大整数值
BIGINT	8 bytes	(-9,223,372,036,854,775,808, 9 223 372 036 854 775 807)	(0, 18 446 744 073 709 551 615)	极大整数值
FLOAT	4 bytes	(-3.402 823 466 E+38, -1.175 494 351 E-38), 0, (1.175 494 351 E-38, 3.402 823 466 351 E+38)	0, (1.175 494 351 E-38, 3.402 823 466 E+38)	单精度浮点数值
DOUBLE	8 bytes	(-1.797 693 134 862 315 7 E+308, -2.225 073 858 507 201 4 E-308), 0, (2.225 073 858 507 201 4 E-308, 1.797 693 134 862 315 7 E+308)	0, (2.225 073 858 507 201 4 E-308, 1.797 693 134 862 315 7 E+308)	双精度浮点数值
DECIMAL	对 DECIMAL(M,D) , 如果M>D, 为M+2否则为 D+2	依赖于M和D的值	依赖于M和D 的值	小数值

日期和时间类型

表示时间值的日期和时间类型为DATETIME、DATE、TIMESTAMP、TIME和YEAR。

每个时间类型有一个有效值范围和一个"零"值，当指定不合法的MySQL不能表示的值时使用"零"值。

TIMESTAMP类型有专有的自动更新特性，将在后面描述。

类型	大小 (bytes)	范围	格式	用途
DATE	3	1000-01-01/9999-12-31	YYYY-MM-DD	日期值
TIME	3	'-838:59:59'/'838:59:59'	HH:MM:SS	时间值或持续时间
YEAR	1	1901/2155	YYYY	年份值
DATETIME	8	1000-01-01 00:00:00/9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS	混合日期和时间值
TIMESTAMP	4	1970-01-01 00:00:00/2038结束时间是第 2147483647 秒，北京时间 2038-1-19 11:14:07 ，格林尼治时间 2038年1月19日凌晨 03:14:07	YYYYMMDD HHMMSS	混合日期和时间值，时间戳

字符串类型

字符串类型指CHAR、VARCHAR、BINARY、VARBINARY、BLOB、TEXT、ENUM和SET。该节描述了这些类型如何工作以及如何在查询中使用这些类型。

类型	大小	用途
CHAR	0-255 bytes	定长字符串
VARCHAR	0-65535 bytes	变长字符串
TINYBLOB	0-255 bytes	不超过 255 个字符的二进制字符串
TINYTEXT	0-255 bytes	短文本字符串
BLOB	0-65 535 bytes	二进制形式的长文本数据
TEXT	0-65 535 bytes	长文本数据
MEDIUMBLOB	0-16 777 215 bytes	二进制形式的中等长度文本数据
MEDIUMTEXT	0-16 777 215 bytes	中等长度文本数据
LOBLOB	0-4 294 967 295 bytes	二进制形式的极大文本数据
LONGTEXT	0-4 294 967 295 bytes	极大文本数据

MySQL 创建数据库

我们可以在登陆 MySQL 服务后，使用 **create** 命令创建数据库，语法如下：

```
CREATE DATABASE 数据库名；
show databases； 查看数据库
status； 查看mysql数据库的运行状态
```

1.information_schema

information_schema 提供了数据库访问元数据的方式（元数据，可以理解为关于数据的数据，比如记录数据库名称或者表名的数据就可以称为元数据，就像数据字典和数据目录一般）。information_schema是MySQL的信息库，里面保存了所有数据库的相关信息，比如数据库名、数据库表、表的数据类型、访问权限等。

2.mysql

这个是MySQL的核心数据库，里面记录数据库用户、权限、关键字等核心的管理信息。

3.performance_schema

```
performance_table 为系统新增用以收集数据库服务器性能参数，其下的所有表的存储引擎均为
performance_schema
```

4.sys

sys 内的table大多承载了performance_schema的内容，它通过视图把information_schema和performance_schema结合起来，具体可以结合官方手册来看。

MySQL 删除数据库

使用普通用户登陆 MySQL 服务器，你可能需要特定的权限来创建或者删除 MySQL 数据库，所以我们这边使用 root 用户登录，root 用户拥有最高权限。

在删除数据库过程中，务必要十分谨慎，因为在执行删除命令后，所有数据将会消失。

drop 命令删除数据库

drop 命令格式：

```
drop database <数据库名>;
```

MySQL 选择数据库

在你连接到 MySQL 数据库后，可能有多个可以操作的数据库，所以你需要选择你要操作的数据库。

从命令提示窗口中选择MySQL数据库

在 mysql> 提示窗口中可以很简单选择特定的数据库。你可以使用SQL命令来选择指定的数据库。

实例

以下实例选取了数据库 RUNOOB:

```
[root@host]# mysql -u root -p
Enter password:*****
mysql> use RUNOOB;
Database changed
```

MySQL 创建数据表

创建MySQL数据表需要以下信息：

- 表名
- 表字段名
- 定义每个表字段

语法

以下为创建MySQL数据表的SQL通用语法：

```
CREATE TABLE table_name;
show tables;查看数据库中的表
desc table_name;查看表结构
```

以下例子中我们将在 RUNOOB 数据库中创建数据表runoob_tbl：

```
CREATE TABLE student(  
    `id` INT UNSIGNED AUTO_INCREMENT,  
    `title` VARCHAR(100) NOT NULL,  
    `author` VARCHAR(40) NOT NULL,  
    `date` DATE,  
    PRIMARY KEY ( `id` )  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
create table booktable(  
    `id` int UNSIGNED AUTO_INCREMENT,  
    `name` varchar(100) NOT NULL,  
    `author` varchar(30) NOT NULL,  
    `category` varchar(50) NOT NULL,  
    `description` varchar(100) DEFAULT NULL,  
    PRIMARY KEY(`id`)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

实例解析：

- 如果你不想字段为 **NULL** 可以设置字段的属性为 **NOT NULL**，在操作数据库时如果输入该字段的数据为**NULL**，就会报错。
- AUTO_INCREMENT定义列为自增的属性，一般用于主键，数值会自动加1。
- PRIMARY KEY关键字用于定义列为主键。您可以使用多列来定义主键，列间以逗号分隔。
- ENGINE 设置存储引擎，CHARSET 设置编码。

MySQL 删除数据表

MySQL中删除数据表是很容易操作的，但是你在进行删除表操作时要非常小心，因为执行删除命令后所有数据都会消失。

语法

以下为删除MySQL数据表的通用语法：

```
DROP TABLE table_name ;
```

在命令提示窗口中删除数据表

在mysql>命令提示窗口中删除数据表SQL语句为 **DROP TABLE**：

实例

以下实例删除了数据表runoob_tbl:

```
root@host# mysql -u root -p  
Enter password:*****  
mysql> use RUNOOB;  
Database changed  
mysql> DROP TABLE runoob_tbl  
Query OK, 0 rows affected (0.8 sec)
```

MySQL 插入数据

MySQL 表中使用 **INSERT INTO** SQL语句来插入数据。

你可以通过 `mysql>` 命令提示窗口中向数据表中插入数据，或者通过PHP脚本来插入数据。

语法

以下为向MySQL数据表插入数据通用的 **INSERT INTO** SQL语法：

```
INSERT INTO table_name ( field1, field2,...fieldN )
VALUES
( value1, value2,...valueN );
```

如果数据是字符型，必须使用单引号或者双引号，如："value"。

通过命令提示窗口插入数据

以下我们将使用 SQL **INSERT INTO** 语句向 MySQL 数据表 `runoob_tbl` 插入数据

实例

以下实例中我们将向 `runoob_tbl` 表插入三条数据:

```
root@host# mysql -u root -p password;
Enter password:*****
mysql> use RUNOOB;
Database changed
mysql> INSERT INTO student
-> (title, author, date)
-> VALUES
-> ("学习 mysql", "张三", NOW());
Query OK, 1 rows affected, 1 warnings (0.01 sec)
mysql> INSERT INTO student
-> (title, author, date)
-> VALUES
-> ("学习 MySQL", "李四", NOW());
Query OK, 1 rows affected, 1 warnings (0.01 sec)
```

读取数据表：

```
select * from runoob_tbl;    返回数据表 runoob_tbl 的所有记录：
```

```
SELECT * from runoob_tbl WHERE author='张三';    返回数据表 runoob_tbl 的作者为张三记录：
```

MySQL UPDATE 更新

如果我们需要修改或更新 MySQL 中的数据，我们可以使用 SQL UPDATE 命令来操作。

语法

以下是 UPDATE 命令修改 MySQL 数据表数据的通用 SQL 语法：

```
UPDATE table_name SET field1=new-value1, field2=new-value2
```

实例

以下实例将更新数据表中 id 为 2 的 runoob_title 字段值：

SQL UPDATE 语句：

```
mysql> UPDATE student SET title='学习' WHERE id=2;  
Query OK, 1 rows affected (0.01 sec)
```

从命令行中删除数据

MySQL DELETE 语句

你可以使用 SQL 的 DELETE FROM 命令来删除 MySQL 数据表中的记录。

你可以在 **mysql>** 命令提示符或 PHP 脚本中执行该命令。

语法

以下是 SQL DELETE 语句从 MySQL 数据表中删除数据的通用语法：

```
DELETE FROM table_name [WHERE Clause]
```

- 如果没有指定 WHERE 子句，MySQL 表中的所有记录将被删除。
- 你可以在 WHERE 子句中指定任何条件
- 您可以在单个表中一次性删除记录。

当你想删除数据表中指定的记录时 WHERE 子句是非常有用的。

从命令行中删除数据

这里我们将在 SQL DELETE 命令中使用 WHERE 子句来删除 MySQL 数据表 runoob_tbl 所选的数据。

实例

以下实例将删除 runoob_tbl 表中 id 为 2 的记录：

DELETE 语句：

```
mysql> use RUNOOB;  
Database changed mysql  
> DELETE FROM runoob_tbl WHERE id=2;  
Query OK, 1 row affected (0.23 sec)
```

Node.js 连接 MySQL

mysql 报错Error: ER_NOT_SUPPORTED_AUTH_MODE: Client does not support authentication protocol requested by server; consider upgrading MySQL client

起因：mysql8.0加密方式的原因报错。

解决办法：

执行指令

```
mysql -u root -p
```

```
123456
```

```
use mysql;
```

```
alter user 'root'@'localhost' identified with mysql_native_password by '123456';
```

```
flush privileges;
```

卸载

**快捷键win+r输入regedit进入注册表，找到
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Eventlog\Application\MySQL文件夹删除**

删除

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Services\Eventlog\Application\MySQL文件夹。

删除

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\MySQL的文件夹。

注册表里没有这两个文件，就不用删除了

删除所有mysql文件

设计和创建用户表

**创建数据库=>shop=》字符集：utf-8=>排序规则=》utf8:
general_ci**

1.先从用户表设计

用户表

字段	类型	约束	描述
USER_ID	varchar(32)	NOT NULL	用户id
USER_NAME	varchar(20)	NOT NULL	用户名称
USER_PASSWORD	varchar(20)	NOT NULL	用户密码
USER_SEX	varchar(1)	NOT NULL	性别 T F
USER_BIRTHDAY	datetime	DEFAULT null	出生日期
USER_EMAIL	varchar(50)	DEFAULT null	电子邮箱
USER_MOBILE	varchar(11)	DEFAULT null	电话
USER_ADDRESS	varchar(200)	DEFAULT null	地址
USER_STATUS	decimal(6,0)(标记、状态位)	NOT NULL	用户类型1：用户2：管理员
USER_MEMO	varchar(60)	DEFAULT null	占位符

2.在数据库中创建用户表

```
drop table if exists `user`;

create table `user`(
  USER_ID varchar(32) not null,
  USER_NAME varchar(20) not null,
  USER_PASSWORD varchar(20) not null,
  USER_SEX varchar (1) not null,
  USER_BIRTHDAY datetime default null,
  USER_EMAIL varchar(50) default null,
  USER_MOBILE varchar(11) default null,
  USER_ADDRESS varchar(200) not null,
  USER_STATUS decimal(6, 0) not null,
  USER_MEMO varchar(60) default null,
  primary key(`USER_ID`)
)ENGINE=InnoDB default charset=utf8; //设置表引擎的默认字符编码
```