



MySQL 超级入门教程（内含资源福利）



地球的外...

Linux云计算和Python推动市场提升的学习研究者。

174 人赞同了该文章

MySQL简介

1、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，它产生于距今六十多年前，随着信息技术和市场的发展，特别是二十世纪九十年代以后，数据管理不再仅仅是存储和管理数据，而转变成用户所需要的各种数据管理的方式。数据库有很多种类型，从最简单的存储有各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用。

主流的数据库有：sqlserver，mysql，Oracle、SQLite、Access、MS SQL Server等，本文主要讲述的是mysql

2、数据库管理是干什么用的？

- a. 将数据保存到文件或内存
- b. 接收特定的命令，然后对文件进行相应的操作

PS：如果有了以上管理系统，无须自己再去创建文件和文件夹，而是直接传递 命令 给上述软件，让其来进行文件操作，他们统称为数据库管理系统（DBMS，Database Management System）

MySQL安装

MySQL是一种开放源代码的关系型数据库管理系统（RDBMS），MySQL数据库系统使用最常用的数据库管理语言-结构化查询语言（SQL）进行数据库管理。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。

使用mysql必须具备一下条件

- a. 安装MySQL服务端
- b. 安装MySQL客户端
- c. 【客户端】连接【服务端】
- d. 【客户端】发送命令给【服务端MySQL】服务的接受命令并执行相应操作(增删改查等)

1、下载地址：dev.mysql.com/downloads...

2、安装

- windows安装请参考：cnblogs.com/lonelywolfm...
- linux下安装：cnblogs.com/chenjunbiao...

注：以上两个链接有完整的安装方式，撸主也是参考他的安装的，安装完以后mysql.server start 启动mysql服务

MySQL操作

一、连接数据库

```
mysql -u user -p
例：mysql -u root -p
```

常见错误如下：

```
ERROR 2002 (HY000): Can't connect to local MySQL server through socket
'/tmp/mysql.sock' (2), it means that the MySQL server daemon (Unix) or service (Windows)
is not running.
```

退出连接：

```
QUIT 或者 Ctrl+D
```

二、查看数据库，创建数据库，使用数据库查看数据库：

```
show databases;
```

默认数据库：

```
mysql - 用户权限相关数据
test - 用于用户测试数据
information_schema - MySQL本身架构相关数据
```

创建数据库：

```
create database db1 DEFAULT CHARSET utf8 COLLATE utf8_general_ci; # utf8编码

create database db1 DEFAULT CHARACTER SET gbk COLLATE gbk_chinese_ci; # gbk编码
```

使用数据库：

```
use db1;
```

显示当前使用的数据库中所有表：

```
SHOW TABLES;
```

三、用户管理

创建用户

```
create user '用户名'@'IP地址' identified by '密码';
```

删除用户

```
drop user '用户名'@'IP地址';
```

修改用户

```
rename user '用户名'@'IP地址'; to '新用户名'@'IP地址';
```

修改密码

```
set password for '用户名'@'IP地址' = Password('新密码');
```

注：用户权限相关数据保存在mysql数据库的user表中，所以也可以直接对其进行操作（不建议）

四、权限管理

mysql对于权限这块有以下限制：

```
all privileges: 除grant外的所有权限
select: 仅查权限
select,insert: 查和插入权限
...
usage: 无访问权限
alter: 使用alter table
alter routine: 使用alter procedure和drop procedure
create: 使用create table
create routine: 使用create procedure
create temporary tables: 使用create temporary tables
create user: 使用create user、drop user、rename user和revoke all privileges
create view: 使用create view
delete: 使用delete
drop: 使用drop table
execute: 使用call和存储过程
```

file: 使用select into outfile 和 load data infile
grant option: 使用grant 和 revoke
index: 使用index
insert: 使用insert
lock tables: 使用lock table
process: 使用show full processlist
select: 使用select
show databases: 使用show databases
show view: 使用show view
update: 使用update
reload: 使用flush
shutdown: 使用mysqladmin shutdown(关闭MySQL)
super: 使用change master、kill、logs、purge、master和set global。还允许mysqladmin调试
replication client: 服务器位置的访问
replication slave: 由复制从属使用

对于数据库及内部其他权限如下：

数据库名.*	数据库中的所有
数据库名.表	指定数据库中的某张表
数据库名.存储过程	指定数据库中的存储过程
.	所有数据库

对于用户和IP的权限如下：

用户名@IP地址	用户只能在改IP下才能访问
用户名@192.168.1.%	用户只能在改IP段下才能访问(通配符%表示任意)
用户名@%	用户可以再任意IP下访问(默认IP地址为%)

1、查看权限：

```
show grants for '用户'@'IP地址'
```

2、授权

```
grant 权限 on 数据库.表 to '用户'@'IP地址'
```

3、取消授权

```
revoke 权限 on 数据库.表 from '用户名'@'IP地址'
```

授权实例如下：

```
grant all privileges on db1.tb1 TO '用户名'@'IP'
```

```
grant select on db1.* TO '用户名'@'IP'
```

```
grant select,insert on *.* TO '用户名'@'IP'
```

```
revoke select on db1.tb1 from '用户名'@'IP'
```

MySQL表操作

一、查看表

```
show tables; # 查看数据库全部表
```

```
select * from 表名; # 查看表所有内容
```

二、创建表

```
create table 表名(  
    列名 类型 是否可以为空,  
    列名 类型 是否可以为空  
)ENGINE=InnoDB DEFAULT CHARSET=utf8
```

来一个实例好详解

```
CREATE TABLE `tab1` (  
    `nid` int(11) NOT NULL auto_increment,  
    `name` varchar(255) DEFAULT zhangyanlin,  
    `email` varchar(255),  
    PRIMARY KEY (`nid`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

注：

- 默认值，创建列时可以指定默认值，当插入数据时如果未主动设置，则自动添加默认值
- 自增，如果为某列设置自增列，插入数据时无需设置此列，默认将自增（表中只能有一个自增列）注意：1、对于自增列，必须是索引（含主键）2、对于自增可以设置步长和起始值
- 主键，一种特殊的唯一索引，不允许有空值，如果主键使用单个列，则它的值必须唯一，如果是多列，则其组合必须唯一。

三、删除表

```
drop table 表名
```

四、清空表内容

```
delete from 表名  
truncate table 表名
```

五、修改表

添加列：

```
alter table 表名 add 列名 类型
```

删除列：

```
alter table 表名 drop column 列名
```

修改列：

```
alter table 表名 modify column 列名 类型; -- 类型  
alter table 表名 change 原列名 新列名 类型; -- 列名, 类型
```

添加主键：

```
alter table 表名 add primary key(列名);
```

删除主键：

```
alter table 表名 drop primary key;  
alter table 表名 modify 列名 int, drop primary key;
```

添加外键：

```
alter table 从表 add constraint 外键名称 (形如: FK_从表_主表) foreign key 从表(外键字段)
```

删除外键:

```
alter table 表名 drop foreign key 外键名称
```

修改默认值:

```
ALTER TABLE testalter_tbl ALTER i SET DEFAULT 1000;
```

删除默认值:

```
ALTER TABLE testalter_tbl ALTER i DROP DEFAULT;
```

对于上述这些操作是不是看起来很麻烦, 很浪费时间, 别慌! 有专门的软件能提供这些功能, 操作起来非常简单, 这个软件名字叫Navicat Premium, 大家自行在网上下载, 练练手, 但是下面的即将讲到表内容操作还是建议自己写命令来进行

六、基本数据类型

MySQL的数据类型大致分为: 数值、时间和字符串

`bit[(M)]`

二进制位 (101001), m表示二进制位的长度 (1-64), 默认m=1

`tinyint[(m)] [unsigned] [zerofill]`

小整数, 数据类型用于保存一些范围的整数数值范围:

有符号:

-128 ~ 127.

无符号:

0 ~ 255

特别的: MySQL中无布尔值, 使用tinyint(1)构造。

`int[(m)] [unsigned] [zerofill]`

整数, 数据类型用于保存一些范围的整数数值范围:

有符号:

-2147483648 ~ 2147483647

无符号:

0 ~ 4294967295

特别的：整数类型中的m仅用于显示，对存储范围无限制。例如： int(5),当插入数据2时，

bigint[(m)][unsigned][zerofill]

大整数，数据类型用于保存一些范围的整数数值范围：

有符号：

-9223372036854775808 ~ 9223372036854775807

无符号：

0 ~ 18446744073709551615

decimal[(m[,d))][unsigned][zerofill]

准确的小数值，m是数字总个数（负号不算），d是小数点后个数。 m最大值为65，d最大值为

特别的：对于精确数值计算时需要用此类型

decimal能够存储精确值的原因在于其内部按照字符串存储。

FLOAT[(M,D)][UNSIGNED][ZEROFILL]

单精度浮点数（非准确小数值），m是数字总个数，d是小数点后个数。

无符号：

-3.402823466E+38 to -1.175494351E-38,

0

1.175494351E-38 to 3.402823466E+38

有符号：

0

1.175494351E-38 to 3.402823466E+38

**** 数值越大，越不准确 ****

DOUBLE[(M,D)][UNSIGNED][ZEROFILL]

双精度浮点数（非准确小数值），m是数字总个数，d是小数点后个数。

无符号：

-1.7976931348623157E+308 to -2.2250738585072014E-308

0

2.2250738585072014E-308 to 1.7976931348623157E+308

有符号：

0

2.2250738585072014E-308 to 1.7976931348623157E+308

**** 数值越大，越不准确 ****

char (m)

char数据类型用于表示固定长度的字符串，可以包含最多达255个字符。其中m代表字符串的长度。
PS：即使数据小于m长度，也会占用m长度

varchar(m)

varchar数据类型用于变长的字符串，可以包含最多达255个字符。其中m代表该数据类型所允许的最大长度。

注：虽然varchar使用起来较为灵活，但是从整个系统的性能角度来说，char数据类型的处理效率更高。

text

text数据类型用于保存变长的大字符串，可以组多到65535 ($2^{16} - 1$)个字符。

mediumtext

A TEXT column with a maximum length of 16,777,215 ($2^{24} - 1$) characters.

longtext

A TEXT column with a maximum length of 4,294,967,295 or 4GB ($2^{32} - 1$) characters.

enum

枚举类型，

An ENUM column can have a maximum of 65,535 distinct elements. (The maximum length of an ENUM value is 65,535 characters.)
示例：

```
CREATE TABLE shirts (  
    name VARCHAR(40),  
    size ENUM('x-small', 'small', 'medium', 'large', 'x-large')  
);  
INSERT INTO shirts (name, size) VALUES ('dress shirt','large'),
```

set

集合类型

A SET column can have a maximum of 64 distinct members.

示例：

```
CREATE TABLE myset (col SET('a', 'b', 'c', 'd'));  
INSERT INTO myset (col) VALUES ('a,d'), ('d,a'), ('a,d,a'), ('a,
```

DATE

YYYY-MM-DD (1000-01-01/9999-12-31)

TIME

HH:MM:SS ('-838:59:59'/'838:59:59')

YEAR

YYYY (1901/2155)

DATETIME

YYYY-MM-DD HH:MM:SS (1000-01-01 00:00:00/9999-12-31 23:59:59 Y)

TIMESTAMP

YYYYMMDD HHMMSS (1970-01-01 00:00:00/2037 年某时)

MySQL表内容操作

表内容操作无非就是增删改查，当然用的最多的还是查，而且查这一块东西最多，用起来最难，当然对于大神来说那就是so easy了，对于我这种小白还是非常难以灵活运用的，下面咱来一一操作一下

一、增

```
insert into 表 (列名,列名...) values (值,值,...)
insert into 表 (列名,列名...) values (值,值,...),(值,值,值,...)
insert into 表 (列名,列名...) select (列名,列名...) from 表
例：
insert into tab1(name,email) values('zhangyanlin','zhangyanlin8851@163.com')
```

二、删

```
delete from 表 # 删除表里全部数据
delete from 表 where id=1 and name='zhangyanlin' # 删除ID =1 和name='zhangyanlin'
```

三、改

```
update 表 set name = 'zhangyanlin' where id>1
```

四、查

```
select * from 表  
select * from 表 where id > 1  
select nid,name,gender as gg from 表 where id > 1
```

查这块的条件太多太多我给列举出来至于组合还得看大家的理解程度哈

a、条件判断where

```
select * from 表 where id > 1 and name != 'aylin' and num = 12;  
select * from 表 where id between 5 and 16;  
select * from 表 where id in (11,22,33)  
select * from 表 where id not in (11,22,33)  
select * from 表 where id in (select nid from 表)
```

b、通配符like

```
select * from 表 where name like 'zhang%' # zhang开头的所有（多个字符串）  
select * from 表 where name like 'zhang_' # zhang开头的所有（一个字符）
```

c、限制limit

```
select * from 表 limit 5;           - 前5行  
select * from 表 limit 4,5;        - 从第4行开始的5行  
select * from 表 limit 5 offset 4  - 从第4行开始的5行
```

d、排序asc, desc

```
select * from 表 order by 列 asc      - 根据“列”从小到大排列  
select * from 表 order by 列 desc    - 根据“列”从大到小排列  
select * from 表 order by 列1 desc,列2 asc - 根据“列1”从大到小排列，如果相同则按列2
```

e、分组group by

```
select num from 表 group by num
select num,nid from 表 group by num,nid
select num,nid from 表 where nid > 10 group by num,nid order nid desc
select num,nid,count(*),sum(score),max(score),min(score) from 表 group by num,nid
select num from 表 group by num having max(id) > 10
```

特别的：group by 必须在where之后，order by之前

好了，以上我们就完成了MySQL的入门。经过练习之后，一些简单的增删改查任务已经不成问题，这时候就应该要考虑进阶了。

限于本文的主题和篇幅，在这里我们就不多说了，如果有需要的同学可以前往 MySQL 的官方文档：dev.mysql.com/doc/ 进行查看，英文不太好的同学也可以去菜鸟教程：runoob.com/mysql/mysql-...。

MySQL 资源大全

除了一份入门的教程，小编还给大家准备了一份礼物，那就是 MySQL 资源大全。

工欲善其事必先利其器，有了这些工具，我们和大神的差距就会缩小许多，而且很多工具也会为我们的日常提升提供很大帮助——有了工具包，妈妈再也不用担心你的学习！

分析工具

性能，结构和数据分析工具

- Anemometer – 一个 SQL 慢查询监控器。
- innodb-ruby – 一个对 InnoDB 格式文件的解析器，用于 Ruby 语言。
- innotop – 一个具备多种特性和可扩展性的 MySQL 版 ‘top’ 工具。
- pstop – 一个针对 MySQL 的类 top 程序，用于收集，汇总以及展示来自 performance_schema 的信息。
- mysql-statsd – 一个收集 MySQL 信息的 Python 守护进程，并通过 StatsD 发送到 Graphite。

备份

备份/存储/恢复 工具

- MyDumper – 逻辑的，并行的 MySQL 备份/转储工具。
- MySQLDumper – 基于 web 的开源备份工具-对于共享虚拟主机非常有用。
- mysqldump-secure – 将加密，压缩，日志，黑名单和 Nagios 监控一体化的 mysqldump 安全脚本。
- Percona Xtrabackup – 针对 MySQL 的一个开源热备份实用程序——在服务器的备份期间不会锁定你的数据库。

性能测试

给你的服务器进行压测的工具

- iibench-mysql -基于 Java 的 MySQL/Percona/MariaDB 索引进行插入性能测试工具。
- Sysbench – 一个模块化，跨平台以及多线程的性能测试工具。

聊天应用

集成进聊天室的脚本

- Hubot MySQL ChatOps

配置

MySQL 配置实例及指导

- mysql-compatibility-config – 使 MySQL 配置起来更像新的（或先前）的 MySQL 版本。

连接器

多种编程语言的 MySQL 连接器

- Connector/Python – 一个对于 Python 平台和开发的标准化数据库驱动程序。
- go-sql-driver – 一个 Go 语言的轻量级、极速的 MySQL 驱动程序。
- libAttachSQL – libAttachSQL 是 MySQL 服务器的一个轻量级，非阻塞的 C 语言 API。
- MariaDB Java Client – 针对 Java 应用且经过 LGPL 许可的 MariaDB 客户端库。
- MySQL-Python – 一个 Python 语言的 MySQL 数据库连接器。
- PHP mysqlnd – 针对 MySQL 的 MySQL 本地驱动，弃用过时的 libmysql 基础驱动。

开发

支持 MySQL 相关开发的工具

- Flywaydb – 数据库迁移;任意情况下轻松可靠地演变你的数据库版本。
- Liquibase – 对你的数据库进行源代码控制。
- Propagator – 集中模式和数据部署在一个多维拓扑上。

GUI

前端和应用的 GUI

- Adminer – 一个 PHP 编写的数据库管理工具。
- HeidiSQL – Windows 下的 MySQL 图形化管理工具。
- MySQL Workbench – 提供给数据库管理员和开发人员进行数据库设计和建模的集成工具环境;SQL 开发;数据库管理。
- phpMyAdmin – 一个 PHP 写成的开源软件，意图对 web 上的 MySQL 进行管理。
- SequelPro – 一个 mac 下运行 MySQL 的数据库管理应用程序。
- mycli – 一个带自动补全和语法高亮的终端版 MySQL 客户端

HA

高可用解决方案

- Galera Cluster – 一个基于同步复制的多主机集群方案。
- MHA – 针对 MySQL 的优秀高可用管理器及工具
- MySQL Fabric – 一个用于管理 MySQL 服务器场（Server Farms）的可扩展框架。
- Percona Replication Manager – 针对 MySQL 的异步复制管理代理。支持以文件和 GTID 为基础的复制，使用 booth 实现的地理分布式集群。

代理

MySQL 代理

- MaxScale – 开源，以数据库为中心的代理。
- Mixer – Go 实现的一个 MySQL 代理，目的为 MySQL 分片提供一个简单的解决方案。
- MySQL Proxy – 一个处于你的客户端和 MySQL 服务端之间的简单程序，它可以检测、分析或者改变它们的通信。
- ProxySQL – 高性能的 MySQL 代理。

复制

复制相关的软件

- orchestrator – 对 MySQL 复制拓扑管理并可视化的工具。
- Tungsten Replicator – MySQL 的一个高性能、开源、数据复制引擎。

模式

附加模式

- common_schema – MySQL DBA 的框架， 提供一个具有函数库、视图库和查询脚本的解释器。
- sys – 一个视图、函数和过程的集合，以帮助 MySQL 管理人员更加深入理解 MySQL 数据库的使用。

服务器

MySQL server flavors

- MariaDB – MySQL server 的一个由社区开发的分支。
- MySQL Server & MySQL Cluster – Oracle 官方的 MySQL server 和 MySQL 集群分布。
- Percona Server – 一个加强版的 MySQL 替代品
- WebScaleSQL – WebScaleSQL， 5.6 版本，基于 MySQL 5.6 社区版本。

分片

分片解决方案/框架

- vitess – 对于大规模的 web 服务，vitess 提供服务 and 工具以便于 MySQL 数据库的缩放。
- jetpants – 一个自动化套件，用于管理大规模分片集群，由 Tumblr 开发。

工具包

工具包，通用脚本

- go-mysql – 一个纯 go 的库，用于处理 MySQL 的网络协议和复制。
- MySQL Utilities – 一个命令行实用程序的集合，Python 语言编写，用于维护和管理单一或多层的 MySQL。
- Percona Toolkit – 一个先进的命令行工具集，用于执行对于 MySQL 服务器和系统过于困难或复杂的任务。
- openark kit – 一组实用的工具，解决日常的维护工作，包括一些复杂的或需徒手操作的，用 Python 语言编写。
- UnDROP – 一个用来恢复删除或损坏的 InnoDB 表中数据的工具。