

< 2018年12月 >

| 日 | 一 | 二 | 三 | 四 | 五 | 六 |
|----|----|----|----|----|----|----|
| 25 | 26 | 27 | 28 | 29 | 30 | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 1 | 2 | 3 | 4 | 5 |

公告

昵称：MacoLee

园龄：2年7个月

粉丝：36

关注：8

+加关注

搜索

找找看

谷歌搜索

- 常用链接
- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

- 随笔分类
- [Git\(1\)](#)
- [python\(23\)](#)
- [python之路\(1\)](#)
- [Web前端\(5\)](#)
- [算法\(1\)](#)

MySQL使用说明

一、概述

什么是数据库？

答：数据的仓库，如：在ATM的示例中我们创建了一个 db 目录，称其为数据库

什么是 MySQL、Oracle、SQLite、Access、MS SQL Server等？

答：他们均是一个软件，都有两个主要的功能：

- 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：如果有了以上软件，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）

什么是SQL？

答：上述提到MySQL等软件可以接受命令，并做出相应的操作，由于命令中可以包含删除文件、获取文件内容等众多操作，对于编写的命令就是是SQL语句。SQL是结构化语言（Structured Query Language）的缩写，SQL是一种专门用来与数据库通信的语言。

二、下载安装

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 旗下公司。MySQL 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。

想要使用MySQL来存储并操作数据，则需要做几件事情：

- 安装MySQL服务端
- 安装MySQL客户端

运维(38)

随笔档案

2018年8月 (1)
2017年4月 (1)
2017年3月 (1)
2017年2月 (1)
2017年1月 (1)
2016年12月 (2)
2016年11月 (3)
2016年10月 (1)
2016年9月 (15)
2016年8月 (14)
2016年7月 (20)
2016年6月 (14)

积分与排名

积分 - 85799
排名 - 5054

最新评论

1. Re:Pycharm如何导入Django项目
@dalyday首先，你要在pycharm中打开你的django项目，然后再操作...
--MacoLee

2. Re:Pycharm如何导入Django项目
第二步箭头1处不显示项目名称，麻烦能指点下
--dalyday

3. Re:Ansible之playbook天道酬勤，-i的用法
--Michael2397

4. Re:Python之Fabric模块
@runs_once #查看本地系统信息，当有多台主机时只运行一次
这句没看懂 查看本地系统信息，怎么会有多台主机呢？？？
--Lemon_乐

5. Re:HAProxy安装配置详解赞
--_BLUE

阅读排行榜

1. linux下进程、端口号相互查看方法(36700)

b. 【客户端】连接【服务端】
c. 【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)



下载
http://dev.mysql.com/downloads/mysql/
安装
windows:
点点点
Linux:
yum install mysql-server
Mac:
点点点



三、数据库操作

1、显示数据库

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

2、使用数据库

USE db_name;

3、显示所有表

SHOW TABLES;

- 2. HAProxy安装配置详解(28053)
- 3. Nmap命令的29个实用范例(20559)
- 4. LVS安装使用详解(18316)
- 5. linux下查看进程运行的时间(18280)
- 6. linux系统中rsync+inotify实现服务器之间文件实时同步(10439)
- 7. Pycharm如何导入Django项目(8909)
- 8. Django数据库设计中字段为空的方式(8327)
- 9. Saltstack系列3: Saltstack常用模块及API(7642)
- 10. Keepalived安装使用详解(5999)

评论排行榜

- 1. Pycharm如何导入Django项目(2)
- 2. HAProxy安装配置详解(1)
- 3. Python之Fabric模块(1)
- 4. Ansible之playbook(1)

推荐排行榜

- 1. HAProxy安装配置详解(5)
- 2. LVS安装使用详解(4)
- 3. Nmap命令的29个实用范例(3)
- 4. linux下查看进程运行的时间(2)
- 5. linux下进程、端口号相互查看方法(2)
- 6. Python之Rpyc模块(1)
- 7. Saltstack系列3: Saltstack常用模块及API(1)
- 8. CentOS下puppet安装(1)

4、用户授权

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');
```

授权管理:

```
show grants for '用户'@'IP地址'      -- 查看权限
grant  权限 on 数据库.表 to    '用户'@'IP地址'  -- 授权
revoke 权限 on 数据库.表 from '用户'@'IP地址'  -- 取消权限
```

```
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 | <input type="checkbox"/> <input type="checkbox"/> 使用change master、kill、logs、purge、master和set global 。还允许 |
| mysqladmin <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 调试登陆 | |
| replication client | 服务器位置的访问 |
| replication slave | 由复制从属使用 |

对于权限



对于目标数据库以及内部其他:

| | |
|-----------|-------------|
| 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 |



| | |
|-----------------|---------------------------|
| 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%) |



```
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'
```



四、表操作

1、创建表

```
create table 表名(  
    列名 类型 是否可以空,  
    列名 类型 是否可以空  
)
```



是否可空，null表示空，非字符串

```
not null    - 不可空  
null        - 可空
```



#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

```
create table tb1(  
    nid int not null defalut 2,  
    num int not null  
)
```



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）

```
create table tb1(  
    nid int not null auto_increment primary key,  
    num int null  
)  
或  
create table tb1(  
    nid int not null auto_increment,  
    num int null,
```

```
        index(nid)
    )
# 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

show session variables like 'auto_inc%';
set session auto_increment_increment=2;
set session auto_increment_offset=10;

shwo global variables like 'auto_inc%';
set global auto_increment_increment=2;
set global auto_increment_offset=10;
```



#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。

```
create table tbl(
    nid int not null auto_increment primary key,
    num int null
)
#或
create table tbl(
    nid int not null,
    num int not null,
    primary key(nid,num)
)
```



#外键，一个特殊的索引，只能是指定内容

```
creat table color(
    nid int not null primary key,
    name char(16) not null
)

create table fruit(
    nid int not null primary key,
```

```
smt char(32) null ,
color_id int not null,
constraint fk_cc foreign key (color_id) references color(nid)
)
```



2、删除表

```
drop table 表名
```

3、清空表

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

4、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) references 主表(主键字段);
```

#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;
```



5、基本数据类型

MySQL的数据类型大致分为：数值、时间和字符串

<http://www.runoob.com/mysql/mysql-data-types.html>

五、基本操作

1、增

```
insert into 表 (列名,列名...) values (值,值,值...)
insert into 表 (列名,列名...) values (值,值,值...), (值,值,值...)
insert into 表 (列名,列名...) select (列名,列名...) from 表
```

2、删

```
delete from 表
delete from 表 where id=1 and name='alex'
```

3、改

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

4、查

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


5、其他



#1、条件

```
select * from 表 where id > 1 and name != 'alex' 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)
```

#2、通配符

```
select * from 表 where name like 'ale%' - ale开头的所有（多个字符串）
```

```
select * from 表 where name like 'ale_' - ale开头的所有（一个字符）
```

#3、限制

```
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
```

#4、排序

```
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从小到大排序
```

#5、分组

```
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

#6、连表

无对应关系则不显示

```
select A.num, A.name, B.name
```

```
from A,B
```

```
Where A.nid = B.nid
```

无对应关系则不显示

```
select A.num, A.name, B.name
```

```
from A inner join B
```

```
on A.nid = B.nid
```

A表所有显示, 如果B中无对应关系, 则值为null

```
select A.num, A.name, B.name
```

```
from A left join B
```

```
on A.nid = B.nid
```

B表所有显示, 如果B中无对应关系, 则值为null

```
select A.num, A.name, B.name
```

```
from A right join B
```

```
on A.nid = B.nid
```

#7、组合

组合, 自动处理重合

```
select nickname
```

```
from A
```

```
union
```

```
select name
```

```
from B
```

组合, 不处理重合

```
select nickname
```

```
from A
```

```
union all
```

```
select name
```

```
from B
```



分类: [运维](#)

标签: [数据库](#), [运维](#)

好文要顶

关注我

收藏该文



MacoLee

关注 - 8

粉丝 - 36

+加关注

« 上一篇: [使用Navicat连接数据库, 不能创建外键](#)

» 下一篇: [Nginx上传文件失败](#)

posted on 2016-11-24 17:39 [MacoLee](#) 阅读(378) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问](#) 网站首页。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!

【活动】华为云12.12会员节全场1折起 满额送Mate20

【活动】华为云会员节云服务特惠1折起

【推荐】服务器100%基准CPU性能, 1核1G首年168元, 限时特惠!



腾讯云AMD云服务器

节省IT成本30%

1核1G AMD机型0.57元/天起

立即抢购

相关博文：

- [mysql 使用说明-2](#)
- [Mysql 数据类型使用说明](#)
- [mysql 使用说明-1](#)
- 烂泥：[mysql帮助命令使用说明](#)
- [mysql免安装使用说明](#)



最新新闻：

- 如何在[Unity](#)实现从纹理中生成法线贴图？
 - 特斯拉与超级工厂生死与共
 - 前[NASA](#)工程师设计闪光弹陷阱 帮助打击包裹偷窃行为
 - 思科宣布以6.6亿美元收购光学芯片制造商[Luxtera](#)
 - 共享汽车途歌退押金难 退还规则生变
- » [更多新闻...](#)

Powered by:

[博客园](#)

Copyright © MacoLee