Ubuntu MySQL 手册

1. 安装 MySQL

<1> sudo apt-get install mysql-server (在安装的时候会要求输入 root 密码)

sudo apt-get install mysql-client

2. 启动，停止，重启 MySQL?

<1> /etc/init.d/mysql start

/etc/init.d/mysql stop

/etc/init.d/mysql restart

(修改了配置文件需要重启服务)

3. MySQL 配置文件的存放位置:

<1> /etc/mysql/my.cnf

4. MySQL 数据文件的存放位置:

<1> 这个值保存在配置文件中，通常是:

datadir = /var/lib/mysql

如果我们创建了一个数据库 testdb,这个数据库通常保存在/var/lib/mysql/testdb 下面

5. 使用 MySQL 命令行?

<1> mysql -u root -p

Enter password: 输入密码登录

mysql>

6. 创建，删除 MySQL 数据库?

<1> create database test;

drop database test;

<2> 有时候用 drop 命令删除了数据库之后，通过 show databases; 命令仍然能看到刚才删除的数据库。这通常是因为数据库目录中存在残留的文件。如果出现这种情况，你需要先停止 MySQL 服务器，然后手动将 /var/lib/mysql 下指定数据库的目录删除。

7. 如何创建，删除用户?

<1> 创建用户

insert into mysql.user(host, user, password)
values ("%", "user123", password("user@123"));
成功后根据 8 分配用户一定的权限.

<2> 删除用户

delete from mysql.user where user = 'uer123' and host = '%';
flush privileges;

<3> 修改用户密码

update mysql.user set password = password('newpwd')
where user = 'user123' and host = '%';
flush privileges;

8. 为数据库用户设置权限?

<1> 格式:

grant [权限] on [数据库名].[表名] to ['用户名']@['服务器地址']
identified by ['密码']

服务器地址 = localhost, 只有本机的用户可以访问.
服务器地址 = %, 通配符，任意客户机的 IP 都可以匹配.
服务器地址 = Ip 地址

<2> 例子:

赋予用户 user1 访问 test database 的所有权限
grant all privileges on test.* to 'user1'@'%' identified by 'pwd'
赋予用户 user1 访问所有数据库的所有权限
grant all privileges on *.* to 'user1'@'%' identified by 'pwd'
赋予用户 user1 访问 test 数据库的所有权限，但只限定在本地访问。
grant all privileges on test.* to 'user1'@'localhost' identified by 'pwd'

<3> flush privileges

9. 执行脚本？

<1> mysql -u root -p testdb < adddata.script
mysql -u root -p test < /usr/local/etc/example.script

10. 查看数据?

<1>

mysql>show databases; - 查看所有数据库
mysql>show tables; - 查看表

mysql>describe tabelname; - 查看表结构
mysql>use testdb;　　- 选择数据库

11. 数据库备份和恢复?
<1> 备份
　　mysqldump --add-drop-table -u [username] -p [database] > [backup_file]
　　mysqldump -add-drop-table -u root -p test > /home/kexiao/tmp/testbackup.sql

<2> 恢复
　　mysql -u [username] -p [database] < [backup_file]
　　mysql -u root -p test < /home/kexiao/tmp/testbackup.sql

12. 常用的 SQL 语句:
<1> 创建表格:
　　CREATE TABLE test.documents (
　　id INTEGER PRIMARY KEY NOT NULL AUTO_INCREMENT,
　　title　VARCHAR(255) NOT NULL,
　　content TEXT NOT NULL);

<2> 插入
　　INSERT INTO documents (title, content)　VALUES ('title1', 'content1')

<3> 删除
　　DELETE FROM documents WHERE id = 1 AND content = 'content1'

<4> 修改
　　UPDATE documents SET title = 'titlenew' WHERE id = 1 AND title = 'title1'

<5> 查询
　　SELECT * FROM documents
　　SELECT id, content FROM documents
　　SELECT id, content FROM documents WHERE id = 23 AND content = 'content23'

<6> 联合查询
　　(SELECT A.username, B.username FROM A INNER JOIN B ON A.id = B.id)
　　(SELECT A.username, B.username FROM A LEFT　JOIN B ON A.id = B.id)
　　(SELECT A.username, B.username FROM A RIGHT JOIN B ON A.id = B.id)

13. 测试联合查询

测试数据:

tbl_a

id username

1, 'liqiang'

2, 'huangwei'

3, 'sunchen'

4, 'wuyuehan'

tbl_b

id username

3, 'user3'

4, 'user4'

5, 'user5'

<1> 内联结 (取'交集')

首先可以使用

(SELECT A.username, B.username FROM A, B WHERE A.id = B.id), 这个是隐式的内联结，

例如下面语句:

```
select tbl_a.id, tbl_a.username, tbl_b.username from tbl_a, tbl_b
  where tbl_a.id = tbl_b.id
```

实质上和下面语句作用是一样的:

```
(SELECT A.username, B.username FROM A INNER JOIN B ON A.id = B.id)
select tbl_a.id, tbl_a.username, tbl_b.username from tbl_a
  inner join tbl_b on tbl_a.id = tbl_b.id
```

结果是:

```
id    username    username
3,    'sunchen',  'user3'
4,    'wuyuehan', 'user4'
```

<2> 左联合

查询得到的结果将会是保留所有 tbl_a 表中联结字段的记录，若无与其相对应的 tbl_b 表中的字段记录则留空

```
select tbl_a.id, tbl_a.username, tbl_b.username from tbl_a left join tbl_b
  on tbl_a.id = tbl_b.id
```

结果是:

```
id    username    username
1,    'liqiang',  'NULL'
```

2, 'huangwei', 'NULL'
3, 'sunchen', 'user3'
4, 'wuyuehan', 'user4'

<3> 右联合

查询得到的结果将会是保留所有 tbl_b 表中联结字段的记录，若无与其相对应的 tbl_a 表中的字段记录则留空

```
select tbl_a.id, tbl_a.username, tbl_b.username from tbl_a right join tbl_b
  on tbl_a.id = tbl_b.id
```

结果是:

```
id    username     username
3,    'sunchen',   'user3'
4,    'wuyuehan',  'user4'
NULL, 'NULL',      'user5'
```