

一 . 安装与配置 **MYSQL**

二 . 常用 **mysql** 命令行命令

1 .**mysql** 的启动与停止

启动 **MYSQL** 服务 **net start mysql**

停止 **MYSQL** 服务 **net stop mysql**

2 . **netstat -na | findstr 3306** 查看被监听的端口 , **findstr** 用于查找后面的端口是否存在

3 . 在命令行中登陆 **MYSQL** 控制台 , 即使用 **MYSQL**
COMMEND LINE TOOL

语法格式 **mysql -user=root -password=123456 db_name**

或 **mysql -u root -p123456 db_name**

4 . 进入 **MYSQL** 命令行工具后 , 使用 **status;** 或 **\s** 查看运行环境信息

5 . 切换连接数据库的语法 : **use new_dbname;**

6 . 显示所有数据库 : **show databases;**

7 . 显示数据库中的所有表 : **show tables;**

8 . 显示某个表创建时的全部信息 : **show create table table_name;**

9 . 查看表的具体属性信息及表中各字段的描述

Describe table_name; 缩写形式 : **desc table_name;**

三 。 **MySql** 中的 **SQL** 语句

1 . 数据库创建 : **Create database db_name;**

数据库删除 : **Drop database db_name;** 删除时可先判断是否存在 , 写成 : **drop database if exists db_name**

2. 建表 : 创建数据表的语法 : create table table_name (字段 1 数据类型 , 字段 2 数据类型);

例 : create table mytable (id int , username char(20));

删表 : drop table table_name; 例 : drop table mytable;

8. 添加数据 : Insert into 表名 [(字段 1 , 字段 2 ,)] values (值 1 , 值 2 ,);

如果向表中的每个字段都插入一个值,那么前面 [] 括号内字段名可写也可不写

例 : insert into mytable (id,username) values (1,'zhangsang');

9. 查询 : 查询所有数据 : select * from table_name;

查询指定字段的数据 : select 字段 1 , 字段 2 from table_name;

例 : select id,username from mytable where id=1 order by desc;多表查询语句-----参照第 17 条实例

10. 更新指定数据 , 更新某一个字段的数据 (注意, 不是更新字段的名称)

Update table_name set 字段名='新值' [, 字段 2 ='新值' ,][where id=id_num] [order by 字段 顺序]

例 : update mytable set username='lisi' where id=1;

Order 语句是查询的顺序 , 如 : order by id desc(或 asc) , 顺序有两种 : desc 倒序(100—1,即从最新数据往后查询),asc(从 1-100), Where 和 order 语句也可用于查询 select 与删除 delete

11. 删除表中的信息：

删除整个表中的信息：**delete from table_name;**

删除表中指定条件的语句：**delete from table_name
where 条件语句**；条件语句如：**id=3;**

12. 创建数据库用户

一次可以创建多个数据库用户如：

**CREATE USER username1 identified BY 'password',
username2 IDENTIFIED BY 'password'....**

13. 用户的权限控制：**grant**

库，表级的权限控制：将某个库中的某个表的控制权赋予某个用户

**Grant all ON db_name.table_name TO user_name
[identified by 'password'];**

14. 表结构的修改

(1) 增加一个字段格式：

alter table table_name add column (字段名 字段类型);
----此方法带括号

(2) 指定字段插入的位置：

**alter table table_name add column 字段名 字段类型 after
某字段;**

删除一个字段：

alter table table_name drop 字段名;

(3) 修改字段名称/类型

**alter table table_name change 旧字段名 新字段名 新字
段的类型;**

(4) 改表的名字

alter table table_name rename to new_table_name;

(5) 一次性清空表中的所有数据

truncate table table_name; 此方法也会使表中的取号器 (ID) 从 1 开始

15. 增加主键，外键，约束，索引。。。。(使用方法见 17 实例)

① 约束 (主键 **Primary key**、唯一性 **Unique**、非空 **Not Null**)

② 自动增长 **auto_increment**

③ 外键 **Foreign key**-----与 **reference table_name(col_name 列名)** 配合使用，建表时单独使用

④ 删除多个表中有关联的数据----设置 **foreign key** 为 **set null** ---具体设置参考帮助文档

16. 查看数据库当前引擎

SHOW CREATE TABLE table_name;

修改数据库引擎

ALTER TABLE table_name ENGINE=MyISAM |

InnoDB;

17. SQL 语句运用实例:

--1 建 **users** 表

**create table users (id int primary key
auto_increment,nickname varchar(20) not null unique,password
varchar(100) not null,address varchar(200), reg_date timestamp
not null default CURRENT_TIMESTAMP);**

--2 建 **articles** 表,在建表时设置外键

**create table articles (id int primary key
auto_increment,content longtext not null,userid int,constraint
foreign key (userid) references users(id) on delete set null);**

--2.1 建 articles 表,建表时不设置外键

**create table articles (id int primary key
auto_increment,content longtext not null,userid int);**

--2.2 给 articles 表设置外键

**alter table articles add constraint foreign key (userid)
references users(id) on delete set null;**

--3. 向 users 表中插入数据,同时插入多条

**insert into users (id,nickname,password,address) values
(1,'lyh1','1234',null),(10,'lyh22','4321','湖北武汉'),
(null,'lyh333','5678','北京海淀');**

--4. 向 article 中插入三条数据

**insert into articles (id,content,userid) values
(2,'hahahahahaha',11),(null,'xixixixixix',10),
(13,'aiaiaiaiaiaiaiaiaiaiaiaia',1),(14,'hohoahaoaoooooooooooo',10);**

**--5. 进行多表查询，选择 users 表中 ID=10 的用户发布的
的所有留言及该用户的所有信息**

**select articles.id,articles.content,users.* from users,articles
where users.id=10 and articles.userid=users.id order by
articles.id desc;**

--6. 查看数据库引擎类型

show create table users;

--7. 修改数据库引擎类型

alter table users engine=MyISAM; ---因为 **users** 表中 **ID** 被设置成外键，执行此句会出错

--8. 同表查询,已知一个条件的情况下.查询 **ID** 号大于用户 **lyh1** 的 **ID** 号的所有用户

select a.id,a.nickname,a.address from users a,users b where b.nickname='lyh1' and a.id>b.id;

-----也可写成

select id,nickname,address from users where id>(select id from users where nickname='lyh1');

9. 显示年龄比领导还大的员工：

select a.name from users a,users b where a.managerid=b.id and a.age>b.age;

查询编号为 **2** 的发帖人：先查 **articles** 表,得到发帖人的编号,再根据编号查 **users** 得到的用户名。

接着用关联查询。

select * from articles,users 得到笛卡儿积,再加 **order by articles.id** 以便观察

使用 **select * from articles,users where articles.id=2** 筛选出 **2** 号帖子与每个用户的组合记录

再使用 **select * from articles,users where articles.id=2 and**

articles.userid=users.id 选出 **users.id** 等于 2 号帖的发帖人 **id** 的记录.

只取用户名:**select user where user.id=(select userid from articles where article.id =2)**

找出年龄比小王还大的人:假设小王是 28 岁,先想找出年龄大于 28 的人

select * from users where age>(select age from users where name='xiaowang');

*****要查询的记录需要参照表里面的其他记录:

select a.name from users a,users b where b.name='xiaowang' and a.age>b.age

表里的每个用户都想 **pk** 一下.**select a.nickname,b.nickname from users a,users b where a.id>b.id ;**

更保险的语句:**select a.nickname,b.nickname from (select * from users order by id) a,(select * from users order by id) b where a.id>b.id ;**

再查询某个人发的所有帖子.

select b.* from articles a , articles b where a.id=2 and a.userid=b.userid

说明:表之间存在着关系, **ER** 概念的解释, 用 **access** 中的示例数据库演示表之间的关系.只有 **innodb** 引擎才支持 **foreign key**, **mysql** 的任何引擎目前都不支持 **check** 约束。

四、字符集出现错误解决办法

出现的问题:

```
mysql> update users
```

```
-> set username='关羽'
```

```
-> where userid=2;
```

ERROR 1366 (HY000): Incorrect string value:

'\xB9\xD8\xD3\xF0' for column 'usern

ame' at row 1

向表中插入中文字符时，出现错误。

```
mysql> select * from users;
```

```
+-----+-----+
```

```
| userid | username |
```

```
+-----+-----+
```

```
| 2 | ??? |
```

```
| 3 | ??? |
```

```
| 4 | ?í?ù |
```

```
+-----+-----+
```

3 rows in set (0.00 sec)

表中的中文字符位乱码。

解决办法:

使用命令:

```
mysql> status;
```

```
-----
```

mysql Ver 14.12 Distrib 5.0.45, for Win32 (ia32)

Connection id: 8

Current database: test

Current user: root@localhost

SSL: Not in use

Using delimiter: ;
Server version: 5.0.45-community-nt MySQL Community
Edition (GPL)
Protocol version: 10
Connection: localhost via TCP/IP
Server charset: latin1
Db charset: latin1
Client charset: gbk
Conn. charset: gbk
TCP port: 3306
Uptime: 7 hours 39 min 19 sec
Threads: 2 Questions: 174 Slow queries: 0 Opens: 57 Flush
tables: 1 Open ta
bles: 1 Queries per second avg: 0.006

查看 **mysql** 发现 **Server charset**, **Db charset** 的
字符集设成了 **latin1**, 所以出现中文乱码。

```
mysql> show tables;  
+-----+  
| Tables_in_test |  
+-----+  
| users |  
+-----+  
1 row in set (0.00 sec)
```

更改表的字符集。

```
mysql> alter table users character set GBK;  
Query OK, 3 rows affected (0.08 sec)
```

Records: 3 Duplicates: 0 Warnings: 0