

5. MySQL

CONTENTS

- 简介
- 数据库列类型
- SQL语句语法
- 导入导出



1: MySQL 简介

- MySQL是一种关系数据库管理系统，是一种开源软件
- 由瑞典MySQL AB公司开发，2008年1月16号被Sun公司收购。2009年，SUN又被Oracle收购
- MySQL软件采用双授权政策，分为社区版和商业版。由于体积小、速度快、总体拥有成本低，尤其是开放源码特点，一般中小型网站的开发都选择MySQL作为网站数据库。
- 由于其社区版的性能卓越，搭配PHP和Apache可组成良好的开发环境。
- MySQL能够工作在众多不同的平台上



1:MySQL简介

- 端口：3306
- 默认用户：root
- 字符集：默认字符集latin1，应设置为gbk或utf-8
- 安装时可以启动远程登录、创建匿名账户
- 可安装图形化操作界面，方便操作
- 安装mysql错误解决方案
 - 手动删除mysql安装目录
 - 重新运行配置向导MySQLInstanceConfig.exe
 - 删除C:\ProgramData\MySQL目录



1:MySQL简介

- 启动MySQL服务
 - `net start mysql`
- 连接MySQL
 - `mysql -h127.0.0.1 -uroot -p`
- 退出MySQL
 - `exit`
 - `Quit`
- 关闭MySQL服务
 - `net stop mysql`



2. MySQL列类型

- 数值类型：
 - SMALLINT: 2个字节
 - INT: 4个字节 // age int(10)
 - INTEGER:INT的同义词
 - BIGINT : 8个字节

 - FLOAT : 4个字节
 - DOUBLE : 8个字节 //score float(10,2)
- MySQL支持选择在该类型关键字后面的括号内指定整数值的显示宽度(例如, INT(4))。显示宽度并不限制可以在列内保存的值的范围,也不限制超过列的指定宽度的值的显示



2. MySQL列类型

- 字符串(字符)类型
 - CHAR: 固定长度字符串 sex char(2)
 - VARCHAR: 可变长度字符串 name varchar(20)
 - VARCHAR使用起来较为灵活, CHAR处理速度更快
- TEXT: 非二进制大对象 (字符)
- BLOB: 二进制大对象 (非字符)



2. MySQL列类型

- 日期/时间类型：
 - DATE: YYYY-MM-DD
 - DATETIME: YYYY-MM-DD HH:MM:SS
 - TIMESTAMP: YYYY-MM-DD HH:MM:SS
 - TIME: HH:MM:SS
 - YEAR: YYYY
- 主键自增
 - 不使用序列，通过auto_increment



3. SQL语句语法

- SQL语言包含4个部分：
 - 数据定义语言（如create,drop,alter等语句）
 - 数据查询语言（select语句）
 - 数据操纵语言（insert,delete,update语句）
 - 数据控制语言（如grant,revoke,commit,rollback等语句）
- 数据操纵语言针对表中的数据，而数据定义语言针对数据库或表



3. SQL语句语法

- 数据定义语言

```
create database school; /*创建数据库*/
```

```
show databases;      /*显示所有数据库select database()*/
```

```
use school;          /*指定默认数据库*/
```

```
create table student( /*创建表*/
```

```
    id int(10) primary key auto_increment, /*主键，自增*/
```

```
    name varchar(8),
```

```
    sex char(1),
```

```
    score float(6,2)
```

```
);
```



3. SQL语句语法

- 数据定义语言

show tables; /*显示当前库中表清单*/

/* 显示指定表结构show columns from student;*/

describe student;

show create table student; /*显示建表sql语句*/

drop table student; /*删除表*/

drop database school; /*删除数据库*/



3. SQL语句语法

- 数据操纵语言

```
insert into student values("张三", "t",87.5);
```

```
insert into student values(null,"张三","t",87.5);
```

```
//insert into student (name,sex,score) values("张三","t",87.5);
```

```
select * from student;
```

```
insert into student values(null,"李四","男",89);
```

```
alter table student modify sex char(2);
```

```
insert into student values(null,"李四","男",89);
```

```
select * from student;
```



3. SQL语句语法

- 数据操纵语言

```
update student set sex ="女";
```

```
select * from student;
```

```
update student set sex="男" where id=1;
```

```
select * from student;
```

```
update student set name="王五",score=100 where id=2;
```

```
select * from student;
```

```
delete from student where name="王五";
```

```
select * from student;
```

```
delete from student;
```

```
select * from student;
```



3. SQL语句语法

- 数据查询语言（select语句）
 - select * from student;
 - select id,name,score from student;
 - select * from student where id<5 and sex="男" ;
 - select count(*) from student ;
 - select max(score) from student;
 - select id ,name,score from student order by score desc;
 - select * from stu where name like "张%";



3. SQL语句语法

- 更改表结构语句

- alter table student add birth date;
- insert into student values(null,"赵六","男",100, now());
- insert into student values(null,"赵六","男",100, "1980-12-23");

- alter table student change birth birthday date ;
- alter table student modify sex char(4) not null ;
- alter table student modify sex char(2) after birthday;

- alter table student rename as stu;
- alter table stu drop birthday;



3. SQL语句语法

- 多表查询和外键关联

```
create table person( id int primary key, name varchar(6), sex char(2), age int(10) );
```

```
insert into person values(1,"张三","男",45);
```

```
insert into person values(2,"李四","男",32);
```

```
insert into person values(3,"王五","女",32);
```

```
create table pet(  
    id int auto_increment ,  
    name varchar(6),  
    masterid int,  
    primary key(id),  
    constraint fk foreign key (masterid) references person(id)  
);
```




3. SQL语句语法

- 多表查询和外键关联

- insert into pet (name,masterid) values("happy",1);
- insert into pet (name,masterid) values("lucky",1);
-
- select * from person p1,pet p2 where p1.id=p2.masterid;
- select * from person p1,pet p2 where p1.id=p2.masterid and p1.id=1;

- alter table pet drop foreign key fk;
- alter table pet add constraint fk foreign key (masterid) references person(id) on delete cascade on update cascade;
- update person set id=6 where id=2;
- delete from person where id=6;



3. SQL语句语法

- 分页语句
 - `select * from table limit (start-1)*limit,limit;`
 - 其中start是页码
 - limit是每页显示的条数。



4. 导入导出

- 命令行操作
 - 导入
 - 方法1: `mysql>source d:\dbname.sql`
 - 方法2: `mysql -u root -p 数据库名 < dbname.sql`
 - 导出
 - 导出整个数据库
 - `mysqldump -u root -p 数据库名 > dbname.sql`
 - 只导出一个表
 - `mysqldump -u root -p 数据库名 表名 > dbname.sql`
- 图形界面操作
 - Navicat
 - 导入: 运行SQL文件
 - 导出: 转储SQL文件