

5.1 日目标

5.2 [了解]数据库

- 介绍:

存储和管理数据的仓库

- 分类:

关系型数据库:

Oracle

Microsoft SQL Server

MySQL

SQLite

非关系型数据库:

MongoDB

Redis

- 作用:

存储和管理数据的

- 特点:

1.持久化数据

2.读写速度极快

3.保证数据的有效性

5.3 [了解]关系型数据库管理系统

- 关系型数据库管理系统:

- 概念: **为管理关系型数据库而设计的软件系统**

- 组成:

- 服务端

- 客户端

- SQL介绍: 客户端和服务端通讯的语言

- 主要分成:

- DQL: 数据查询语言 select

- DML: 数据管理语言 insert update delete

- DDL: 数据定义语言 create drop

5.4 [重点]MySQL数据库

- 介绍: MySQL是一个关系型数据库管理系统

- 服务器端

- 安装: `sudo apt-get install mysql-server`

- 使用: 下面命令是在ubuntu下使用的

- 开启:

```
sudo service mysql start
```

- 停止

```
sudo service mysql stop
```

- 重启

```
sudo service mysql restart
```

- 查看状态

```
sudo service mysql status
```

- 配置文件

```
/etc/mysql/mysql.conf.d/mysqld.cnf
```

- 客户端

- 安装: `sudo apt-get install mysql-client`

- 图形化安装和使用

- 命令行安装和使用

5.5 [重点]数据类型和约束

- 作用: 准确的存储数据, 保证数据的正确有效

- 数据类型

- int, bit

- decimal

- varchar, char

- date, time, datetime

- enum

- 数据约束

- primary key

- not null

- unique

- default

- foreign key

5.6 [了解]图形化客户端Navicat

5.7 [重点]命令行客户端MySQL的使用-登陆和登出

- 登陆

```
mysql -uroot -pmysql
```

- 登出

```
quit / exit / Ctrl + D(win下快捷键无效)
```

5.8 [重点]命令行客户端MySQL的使用-数据库操作sql语句

```
-- sql语句最后需要有分号;结尾
-- 显示时间
select now();

-- 查看所有数据库
show databases;

-- 创建数据库(python_db)
-- create database 数据库名 charset=utf8;
create database python39;

-- 指定编码的数据库创建
create database python39 charset=utf8;

-- 使用数据库
-- use 数据库的名字
use python39;

-- 查看当前使用的数据库
select database();

-- 删除数据库
-- drop database 数据库名;
drop database python39;
```

5.9 [重点]命令行客户端MySQL的使用-表结构操作sql语句

```
--02 数据表的操作

-- 查看当前数据库中所有表
show tables;

-- 创建表
-- int unsigned 无符号整形
-- auto_increment 表示自动增长
-- not null 表示不能为空
-- primary key 表示主键
-- default 默认值
```

```
-- create table 数据表名字 (字段 类型 约束[, 字段 类型 约束]);

-- 创建 classes 表(id, name)
create table classes(
    id int unsigned auto_increment not null primary key,
    name varchar(20) not null
);

-- 查看表结构
-- desc 数据表的名字;
desc classes;

-- 创建 students 表(id, name, age, height (decimal)、gender (enum)、cls_id)
create table students(
    id int unsigned not null primary key auto_increment,
    name varchar(20) not null,
    age tinyint unsigned default 0,
    height decimal(5, 2),
    gender enum('男', '女', '保密') default '保密',
    cls_id tinyint unsigned default 1
);

-- 修改表-添加字段 生日 datetime
-- alter table 表名 add 列名 类型;
alter table students add birthday datetime;

-- 修改表-修改字段: 不重命名版
-- alter table 表名 modify 列名 类型及约束;
alter table students modify birthday date not null;

-- 修改表-修改字段: 重命名版
-- alter table 表名 change 原名 新名 类型及约束;
alter table students change birthday birth datetime not null;

-- 修改表-删除字段
-- alter table 表名 drop 列名;
alter table students drop birth;

-- 查看表的创建语句
-- show create table 表名字;
show create table students;

-- 查看库的创建语句
-- show create database 库名字;
show create database python39;
```

```
-- 删除表
-- drop table 表名;
drop table students;
-- drop database 数据库;
drop database python39;
```

5.10 [重点]命令行客户端MySQL的使用-表数据操作sql语句(增删改查)

增加:

```
insert into 表名 values(值1, 值2, ...); # 全列插入
insert into 表名(列名1, 列名2) values(值1, 值2); # 部分插入
insert into 表名 values(值1, 值2, ...), (值1, 值2, ...); # 全列插入多条数据
insert into 表名(列名1, 列名2) values(值1, 值2), (值1, 值2); # 部分插入多条数据
```

修改:

```
update 表名 set 列1=值1, ... where 列1=值1;
```

查询:

```
select * from 表名;
select 列1, 列2, ... from 表名;
select 字段 as 别名, ... from 表名;
```

删除:

```
delete from 表名 where 列1=值1; # 物理删除
开发中通常都是逻辑删除: 先增加一个字段用来标记是否删除, 删除时其实是更改标记, 查询时排除标记
alter table 表名 add isdelete bit default 0;
update 表名 set isdelete = 1 where 列1 = 值1;
select * from 表名 where isdelete = 0; # 逻辑删除
```

5.11 [重点]as 和 distinct 关键字

as: 起别名

给字段起别名: select 字段 as 别名 from 表名;
给表起别名: select 表的别名.字段 from 表名 as 别名;

distinct : 去除重复数据行

```
select distinct 字段1, ... from 表名;
```

5.12 【记忆】where之比较运算

为了方便进行查询操作，已经提前准备好了SQL语句，可以导入SQL，实现快速创建表和添加数据的操作

导入数据：

1. 切换工作目录到python_test_1.sql文件所在目录：cd ~/Desktop/
2. 链接mysql服务端：sudo mysql -uroot -pmysql
3. 创建的数据库：create database python_test01 charset=utf8;
4. 使用自己创建的数据库：use python_test01;
5. 导入数据表和数据：source python_test_1.sql;
6. 查看数据表：show tables; （可以看到classes和students这两张表）

- >
- <
- >=或者 <=
- =
- != 或者 <>

```
select * from students where age > 18;
select * from students where age < 18;
select * from students where age <= 18;
select * from students where age = 18;
select * from students where age != 18;
```

5.13 【记忆】where之逻辑运算

```
select * from students where age > 18 and age < 28; # and
select * from students where age > 18 or height >= 180; # or
select * from students where not (age > 18 and gender = '女'); # not
```

5.14 【记忆】where之模糊查询

- like 表示模糊查询
 - %：任意多个任意字符
 - _：一个任意字符

```
select * from students where name like '___%'; # 查询至少有2个字的名字
```

5.15 【记忆】where之范围查询

- in 用于非连续的范围的查询
 - 相等于 多个值的 or（或）关系
- between 起始 and 终值
 - 连续，包含起始值和终值，是一种 and(与) 关系

```
select * from students where age in (18, 34); # 查询 年龄为18、34的姓名
select * from students where age between 18 and 34; # 查询 年龄在18到34之间的的信息
```

5.16 【记忆】where之空值判断

- 表示判断值为空:

```
is null
```

- 判断值非空:

```
is not null
```

- 错误写法: not is null

```
select * from students where height is null; # 查询身高为空的信息
select * from students where height is not null; # 判非空 is not null
```

5.17 【记忆】order排序

- 排序: order by 列1, 列2

- 排序规则:

- 升序

```
asc 默认, 不用谢
```

- 降序

```
desc
```

```
select * from students where age between 18 and 34 order by height desc, age desc, id desc;
```

5.18 今日知识总结