day14-MySQL基础

学习目标

1. 能够理解数据库的概念
2. 能够安装MySQL数据库
3. 能够启动,关闭及登录MySQL
4. 能够使用SQL语句操作数据库
5. 能够使用SQL语句操作表结构
6. 能够使用SQL语句进行数据的添加修改和删除的操作
7. 能够使用SQL语句简单查询数据

一,数据库概述

1.数据的存储方式

存数据: 集合(内存), 文件(流)

存储位置	优点	缺点
内存	速度快	不能永久保存，数据是临时状态。
文件	数据可以永久保存	操作数据不方便，特别是查询某个数据。
数据库	1.数据可以永久保存 2.查询速度快 3.对数据的管理方便	占用资源，有些服务需要购买。

2.什么是数据库?

数据库（DataBase，DB）：指长期保存在计算机的存储设备(磁盘)上，按照一定规则组织起来，可以被各种用户或应用共享的数据集合.还是以文件的方式存在服务器的电脑上的。

- 用我们自己的话来说: 就是数据的仓库, 用来存数据的

3.常见的关系型数据库

- MYSQL ：开源免费的数据库，中小型的数据库，已经被Oracle收购了。MySQL6.x版本也开始收费。后来Sun公司收购了MySQL，而Sun公司又被Oracle收购
- Oracle：收费的大型数据库.Oracle公司的产品.Oracle收购SUN公司，收购MYSQL.
- DB2：IBM公司的数据库产品,收费的.银行系统中.
- SQLServer：MS公司.收费的中型的数据库.
- SyBase：已经淡出历史舞台.提供了一个非常专业数据建模的工具PowerDesigner.
- SQLite: 嵌入式的小型数据库,应用在手机端.

二,数据库的安装和卸载

具体参考文档: `MySQL安装图解.docx` 和 `MySQL卸载手册.doc`

1.安装需要注意的地方

- 安装路径不要有空格和中文

2.卸载需要注意的地方

- 去360或者控制面板卸载
- 一定要删除两个文件夹（数据库安装路径和数据存放路径，这两个文件夹在配置文件里面my.ini）

```
#Path to installation directory. All paths are usually resolv
basedir="E:/worksoft/mysql/"  数据库软件安装路径

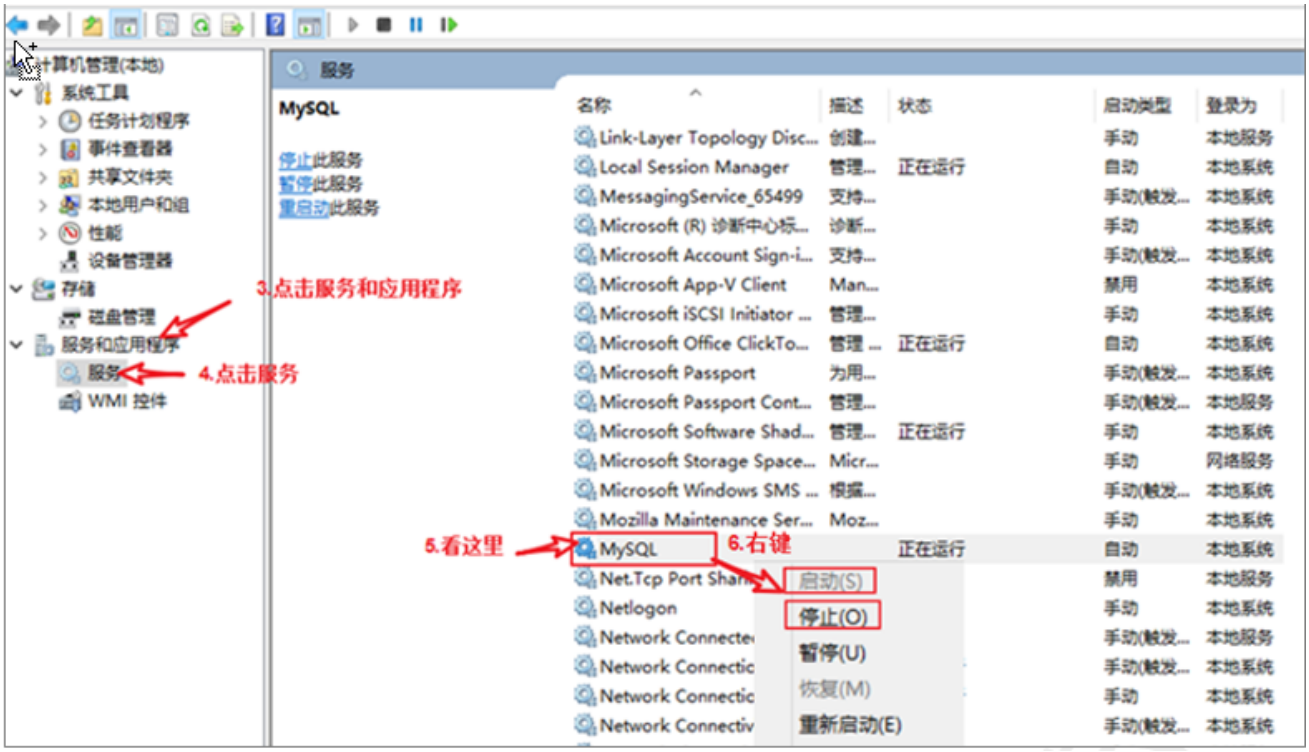
#Path to the database root
datadir="C:/ProgramData/MySQL/MySQL Server 5.5/Data/"
```

数据存放的路径

三,数据库服务的启动和登录

1.数据库服务的启动

1.1 方式一通过界面启动



1.2方式二通过DOS命令方式启动

2.使用命令窗口连接登录和退出

MySQL是一个需要账户名密码登录的数据库，登陆后使用，它提供了一个默认的root账号，使用安装时设置的密码即可登录.

2.1使用命令行连接

- 格式一

```
mysql -u用户名 -p       #然后再输入密码.
mysql.exe -hlocalhost -P3306 -uroot -p
```

- 格式二

```
mysql -hip地址 -u用户名 -p密码     注: 127.0.0.1 代表本机的IP地址
```

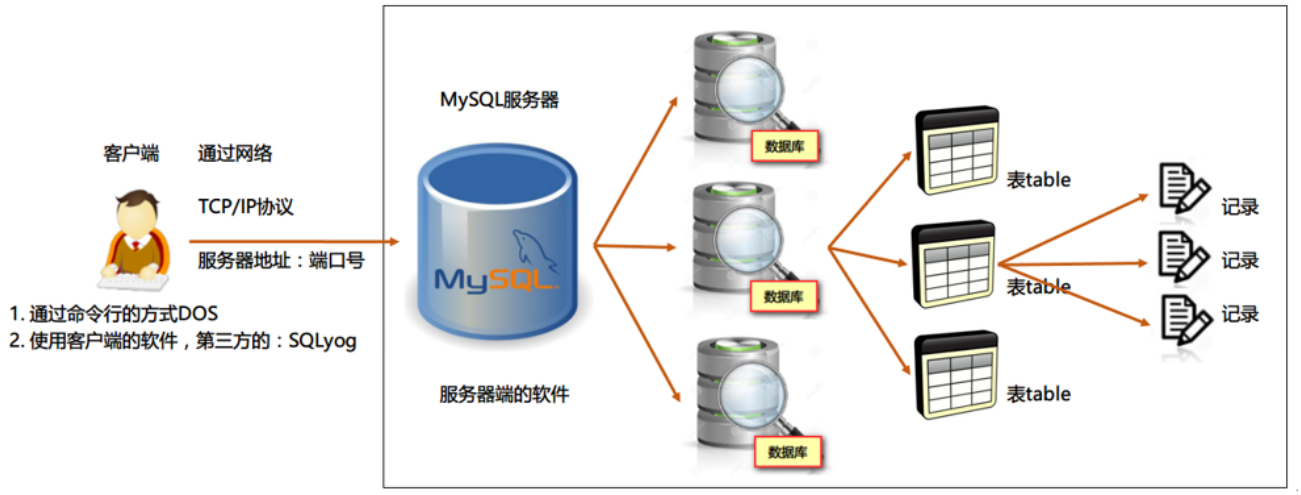
2.2退出

```
输入: quit或exit或\q
```

四,数据库结构【重点】

数据库管理程序(DBMS)可以管理多个数据库，一般开发人员会针对每一个应用创建一个数据库。为保存应用中实体的数据，一般会在数据库创建多个表，以保存程序中实体User的数据。

数据库管理系统、数据库和表的关系如图所示：

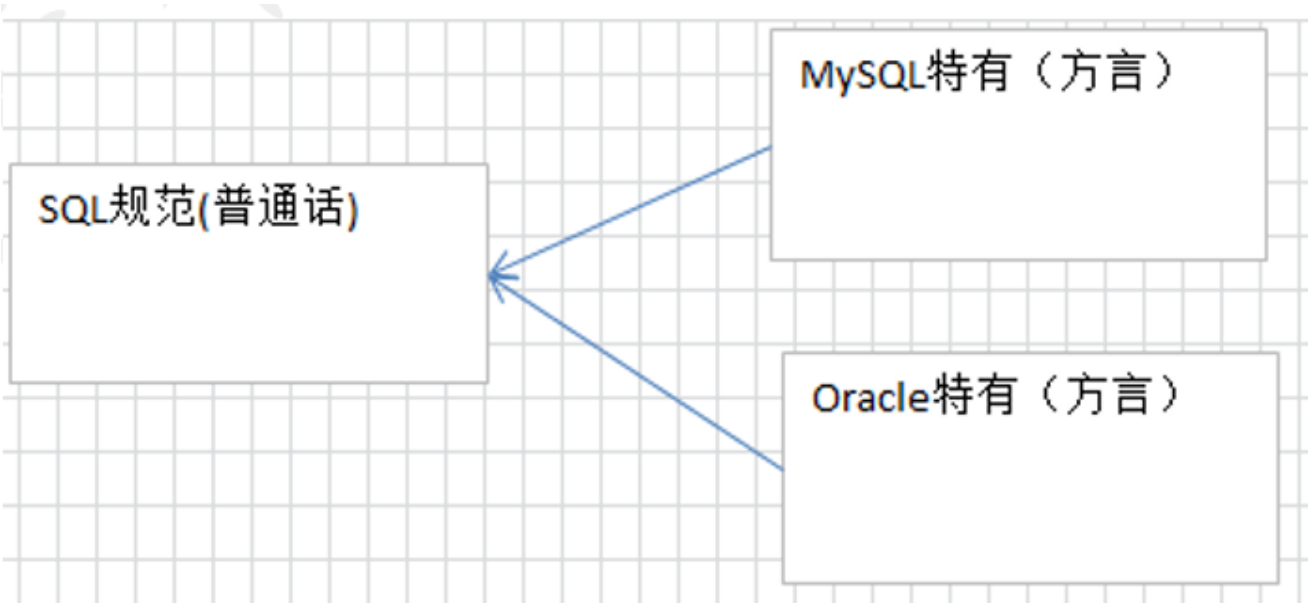


- 一个数据库服务器包含多个库
- 一个数据库包含多张表
- 一张表包含多条记录

五,sql概述

1.什么是sql?

- SQL：Structure Query Language。（结构化查询语言）,通过sql语法操作数据库
- SQL被美国国家标准局（ANSI）确定为关系型数据库语言的美国标准，后来被国际化标准组织（ISO）采纳为关系数据库语言的国际标准
- 各数据库厂商(mysql,oracle,sql server)都支持ISO的SQL标准。
- 各数据库厂商在标准的基础上做了自己的扩展。 各个数据库自己特定的语法



2.sql的语法

- 每条语句以分号结尾(命令行里面需要, 今天需要)，如果在navicat中不是必须加的。
- SQL在window中不区分大小写，关键字中认为大写和小写是一样的

3.sql的分类

- Data Definition Language (DDL数据定义语言) 如：操作数据库，操作表
- DataManipulation Language(DML数据操纵语言)，如：对表中的记录操作增删改
- Data QueryLanguage(DQL 数据查询语言)，如：对表中的查询操作
- Data Control Language(DCL 数据控制语言)，如：对用户权限的设置

六,对数据库的CRUD

1,创建数据库

1.1语法

```
create database 数据库名 [character set 字符集][collate  校对规则]     注: []意思是可选的意思
```

字符集(charset)：是一套符号和编码。

1.2练习

- 创建一个web14_1的数据库（默认字符集,）
- ```
  create database web14_1;
  ```
- 创建一个web14_2的数据库,指定字符集为gbk
- ```
  create database web14_2 character set gbk;
  ```

2.查看数据库

2.1查看所有的数据库

- 语法

```
show databases;
```

2.2查看数据库的定义结构

- 语法

```
show create database 数据库名;
```

3.删除数据库

- 语法

```
drop database 数据库名;
```

4.修改数据库

- 语法

```
alter database 数据库名 character set 字符集;
- 把web14_1的数据库的字符集改成gbk
alter database web14_1 character set gbk;
```

注意：

- 是utf8，不是utf-8
- 不是修改数据库名

5.其它操作

- 切换数据库, 选定哪一个数据库

```
use 数据库名;            //注意: 在创建表之前一定要指定数据库. use 数据库名
```

- 查看正在使用的数据库

```
select database();
```

七,对表的CRUD

1.创建表

1.1语法

```
create table 表名(
    列名(字段) 类型 [约束],
    列名 类型 [约束],
    列名 类型 [约束]
);
```

1.2 类型

分类	数据类型	说明
数值类型	TINYINT [UNSIGNED] [ZEROFILL] BOOL，BOOLEAN SMALLINT [UNSIGNED] [ZEROFILL] INT [UNSIGNED] [ZEROFILL] BIGINT [UNSIGNED] [ZEROFILL] FLOAT[(*M*,*D*)] [UNSIGNED] [ZEROFILL] DOUBLE[(*M*,*D*)] [UNSIGNED] [ZEROFILL]	带符号的范围是-128到127。无符号0到255。 使用0或1表示真或假 2的16次方 2的32次方 2的64次方 M指定显示长度，d指定小数位数 表示比float精度更大的小数
文本、二进制类型	CHAR(size) char(20) VARCHAR(size) varchar(20) BLOB LONGBLOB TEXT(clob) LONGTEXT(longclob)	固定长度字符串 可变长度字符串 二进制数据 大文本
时间日期	DATE/DATETIME/TimeStamp	日期类型(YYYY-MM-DD) (YYYY-MM-DD HH:MM:SS)，TimeStamp表示时间戳，它可用于自动记录insert、update操作的时间

整型:一般 int

浮动型: float, float(8,2); double

eg: grade float 或者 grade float(4,2); 一共是4位(其中小数位占2位), -99.99~99.99之间

字符串: char(20), 最大存放20个字符. 固定长度; 'aaa', 本来只占3个字符的空间, 后面的17个自动用""补上去. 效率高

varchar(20),最大存放20个字符. 可变长度; 'aaa', 本来只占3个字符的空间,就分配三个, 不会补全. 效率低一点

一般情况下用varchar(), 什么情况下情况用char() ? eg: 数据是确定了长度. 身份证号码, 手机号码

二进制: 我们工作里面开发, 一般不会存文件的内容(二进制) ,存文件的路径

时间类型: TimeStamp 时间戳. 如果你把当前的列设置成了TimeStamp 类型, 插入数据库的时候不需要插入的(null),

mysql会自动的把系统的时间自动赋值

1.3 约束

- 即规则,规矩 限制;
- 作用：保证用户插入的数据保存到数据库中是符合规范的

约束名	约束关键字
主键	primary key
唯一	unique
非空	not null

约束种类:

- not null; 非空 ; eg: name varchar(40) not null, name这个列里面的数据不可以有null值
- unique；唯一约束, 后面的数据不能和前面重复; eg: cardNum varchar(200) unique
- primary key；主键约束(非空+唯一); 一般用在表的id列上面. 一张表基本上都有id列的, id列作为唯一标识的

- auto_increment;自动增长列 , id int primary key auto_increment; 当前的id自动增长的, 设置为null, 从1开始自动给你赋值

  注意: 1.先设置主键再有自动增长

  2.只有加了自动增长,才可以插入null,系统自动维护.

  3. 一般用在整型上面, 字符串不可以设置

id列:

1. 给id设置为int类型, 添加主键约束, 自动增长
2. 或者给id设置为字符串类型,添加主键约束,

1.4练习

- -创建一张学生表(含有id字段,姓名字段,性别字段. id为主键自动增长)
- ```
  create table student(
  id int primary key auto_increment,
  name varchar(40) not null,
  sex int
  );
  ```

2.查看表

2.1查看所有的表

```
show tables;
```

2.2查看表的定义结构

```
desc  表名；
```

3.修改表

3.1语法

- 增加一列；alter table 表 add 字段 类型 约束；
- 修改列的类型约束； alter table 表 modify 字段 类型 约束 ；
- 修改列的名称，类型，约束；alter table 表 change 旧列 新列 类型 约束;
- 删除一列； alter table 表名 drop 列名；
- 修改表名 ; rename table 旧表名 to 新表名；

3.2练习

- 给学生表增加一个grade字段

```
alter table student add grade double;
```
```

- 给学生表的sex字段改成字符串类型

```
alter table student modify sex varchar(10);
```

- 给学生表的grade字段修改成class字段

```
alter table student change  grade class varchar(40);
```

- 把class字段删除

```
alter table student drop class;
```

- 把学生表修改成老师表

```
rename table student to teacher;
```

4.删除表

```
drop table 表名；
```

八,对表里面数据的CRUD【重点】

- 准备工作:

  创建一张商品表(商品id,商品名称,商品价格,商品数量,商品类型.)

1.插入数据

- 方式一:

```
insert into 表(列，列..) values(值，值..)；     注：插入指定的列,没有插入的列会自动置为null;
```

- 方式二

```
insert into 表 values(值，值....);            注:插入所有的列
```

注意：

- 方式一没有赋值的列,系统自动赋为null(前提是当前列没有设置not null 约束)
- 列名与列值的类型、个数、顺序要一一对应。
- 值不要超出列定义的长度。
- 插入的日期和字符串，使用引号括起来。

命令行插入中文数据报错:

```
mysql> use day01;
Database changed
mysql> insert into student values(2,'张三');
ERROR 1366 (HY000): Incorrect string value: '\xD5\xC5\xC8\xFD' for column 'name'
 at row 1
mysql>
```

- 关闭服务, net stop mysql
- 在数据库软件的安装目录下面, 修改配置文件 my.ini中客户端的编码为gbk

```
[client]

port=3306

[mysql]

default-character-set=utf8
```

- 重新打开命令行,开启服务, net start mysql

2.更新记录

2.1语法

```
update 表 set 列 = 值 ,列= 值 [where 条件]
```

2.2练习

- 将所有商品的价格修改为5000元
- 将商品名是Mac的价格修改为18000元
- 将商品名是Mac的价格修改为17000,数量修改为5
- 将商品名是方便面的商品的价格在原有基础上增加2元

```
创建一张商品表(商品id,商品名称,商品价格,商品数量,商品类型.)
create table product(
    id int primary key auto_increment,
    name varchar(40),
    price double,
    num int,
    cno int   //类型的编号(我随便取的)
);

1. 插入记录
- 插入指定的列 id, name, price

insert into product(id,name,price) values(null,'iPhonex',8000.0);
```

```
- 插入所有的列
insert into product values(null,'苹果电脑',18000,10,1);
insert into product values(null,'iPhone8s',5500,100,1);
insert into product values(null,'iPhone7',5000,100,1);
insert into product values(null,'iPhone6s',4500,1000,1);
insert into product values(null,'iPhone6',3800,200,1);
insert into product values(null,'iPhone5s',2000,10,1);
insert into product values(null,'iPhone4s',18000,1,1);

insert into product values(null,'方便面',4.5,1000,2);
insert into product values(null,'咖啡',10,100,2);
insert into product values(null,'矿泉水',2.5,100,2);

2. 更新记录 update 表 set 列 = 值 ,列= 值 [where 条件]
- 将所有商品的价格修改为5000元
update product set price = 5000;
- 将商品名是iPhonex的价格修改为18000元
update product set price = 18000 where name = 'iPhonex';
- 将商品名是iPhonex的价格修改为17000,数量修改为5
update product set price = 17000, num = 5 where name ='iPhonex';
- 将商品名是方便面的商品的价格在原有基础上增加2元
update product set price = price + 2 where name ='方便面';
```

3.删除记录

3.1delete

```
delete from 表 [where条件];     注意: 删除数据用delete,不用truncate
```

3.2truncate

```
truncate table 表;
```

delete 和truncate区别

- DELETE 删除表中的数据，表结构还在;删除后的数据可以找回,一条一条的删除.
- TRUNCATE 删除是把表直接DROP掉，然后再创建一个同样的新表。删除的数据不能找回。执行速度比DELETE快。

3.3练习

- 删除表中名称为'苹果电脑'的记录
- 删除价格小于100的商品记录
- 删除表中的所有记录

```
- 删除表中名称为’苹果电脑’的记录
delete from product where name = '苹果电脑';

- 删除价格大于5000的商品记录
delete from product where price > 5000;

- 删除表中的所有记录
delete from product;
```

4.查询记录【重点】

4.1语法

```
select  [*] [列名,列名] [distinct 字段] from 表名 [where 条件]
```

4.2简单查询

4.2.1 查询所有的的记录

- 语法

```
select * form 表
```

4.2.2查询某张表特定列的记录

- 语法

```
select 列名,列名 from 表
```

4.2.3 去重查询

- 语法

```
SELECT DISTINCT 字段名 FROM 表名;   //要数据一模一样才能去重
```

4.2.4 别名查询

- 语法

```
select 列名 as 别名 ,列名 from 表   //列别名
select 别名.* from 表 as 别名      //表别名 这里别名不带单引号
```

- 查询商品名称和商品价格，商品价格通过别名‘价格’来显示

4.2.5运算查询(+,-,*,/等)

- 把商品名，和商品价格+10查询出来

注意：

- 运算查询字段,字段之间是可以的
- 字符串等类型可以做运算查询，但结果没有意义

```
-  查询所有商品记录
select * from product;
- 查询商品的名字和价格
select name, price from product;
- 去重查询(查询商品的价格)
select distinct price from product;
- 别名查询
select name, price '价格' from product;
- 把商品名，和商品价格+10查询出来
select name, price+10 from product;
```

4.3条件查询

4.3.1语法

```
SELECT 字段名 FROM 表名 WHERE 条件;  //取出表中的每条数据，满足条件的记录就返回，不满足条件的记录不返回
```

比较运算符	> < <= >= = <>	大于、小于、大于(小于)等于、不等于
	between .. and ..	显示在某一区间的值
	in(set)	显示在in列表中的值，例：in(100,200)
	like ‘张pattern’	模糊查询
	Is null	判断是否为空
逻辑运算符	and	多个条件同时成立
	or	多个条件任一成立
	not	不成立，例：where not(salary>100);

1. between 小的值 and 大的值 , 闭区间 ;

   eg: price between 3000 and 6000 ---> 3000<= price <= 6000

2. 字段 like '模糊的值'; like通常和占位符一起使用 _ 占一位, % 占多位(0~n)

   查询 姓张的用户 uname like '张_'

   3.in(set)用法 select name from product where price in(2000,6000,3800,5000);

   4.like用法

   select name from product where name like 'ip%'; select name from product where name like 'ip_';

4.3.2练习

- 查询商品价格>3000的商品
- 查询价格在3000到6000之间的商品
- 查询id在1，5，10，15范围内的商品
- 查询商品名以iP开头的商品

- 查询商品价格大于3000并且数量大于20的商品
- 查询pid为1或者价格小于3000的商品

```
- 查询商品价格>3000的商品
select * from product where price > 3000;
- 查询 商品名字不是 苹果电脑 的商品
select * from product where name <> '苹果电脑';
- 查询价格在3000到6000之间的商品
select * from product where price between 3000 and 6000;
-查询id=1，5，10，15的商品
select * from product where id = 1;
select * from product where id = 5;
select * from product where id = 10;
select * from product where id = 15;

select * from product where id in(1,5,10,15);
- 查询商品名以iP开头的商品(查询iPhone系列)
select * from product where name like 'iP%';
- 查询商品价格大于3000并且数量大于20的商品
select * from product where price > 3000 and num > 20;
-查询id为1或者价格小于3000的商品
select * from product where id = 1 or price < 3000;
```

数据库

除了use使用数据库外其他均带database

```
增
create database student
删
drop database student
改
alter database student charset utf8;
查
show databases;
show create database student
select database();
use student
```

表

除了描述desc之外其他全带table

```
增
create table student(
id int primary key auto_increment,
name varchar(6),
age char(2)
```

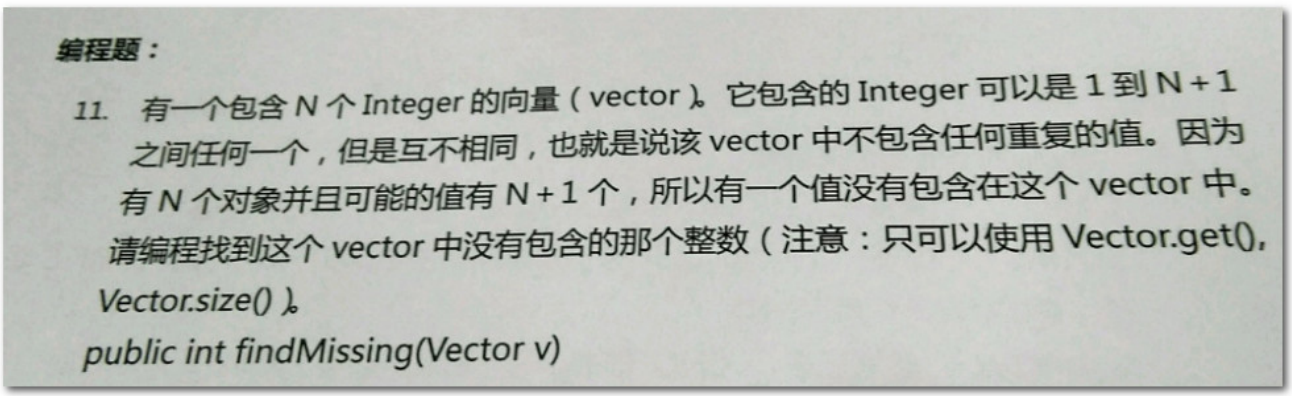
```
);
删
drop table student;
truncate table student;
改
alter table student add sex int;
alter table student modify name int;
alter table student change name mingzi varchar(10) unique;
alter table student drop mingzi;
rename table student to teacher;
查
show tables;
desc student;
```

数据

```
数据库的所有操作均不带table
数据的删除为delete from且不带*
数据的增为insert into values
数据的改为update set
数据的查为select from
```

```
增
insert into student values(null,13,25);
insert into student (id,sex) values(null,10);
删
delete from student where sex=25;
改
update student set sex=3 where id=2;
查
select * from student;
select distinct name from student;
```

面试题：



答案

```
public int find(Vector<Integer> v){
int n = v.size();
int result = 0;
for(int i=1;i<=n+1;i++){
boolean isExist = false;
for(int j=0;j<n;j++){
if(i == v.get(j)){
isExist = true;
break;
}
}
if(isExist == false){
result = i;
break;
}
}
return result; //返回0 证明传入的参数不符合规定或N+1个值都包含在vector中
}
```