学习方法

1. 建立逻辑思维
   1. 多分析
   2. 理解逻辑算法
   3. 多练、多敲、多读
2. 编程习惯
3. 大胆动手
4. 保持好奇好
5. 乐观面对错误

就业能力

1. 分析能力
2. 独立思考能力
3. 程序阅读能力
4. 程序排错能力
5. 独立编写程序能力
6. 团队协作能力
7. 学习能力
8. 善于总结的能力

专业技能

mysql数据库（单表）

1. 数据库的概念
   1. 概念：存储数据的仓库，是一个很广义的概念。任何能存储数据的容器，都可以被称之为数据库。  
      一个MP3，一个U盘，一个文件夹，甚至一个txt文档，都可以叫做数据仓库。一般情况下我们所说的数据库文件，指的是被特定软件结构化的数据文件。
2. 结构化数据与传统数据的区别
   1. 什么是结构化的数据？
      1. 如：我是一个热爱生活的人，很乐观。我今年刚从清华大学毕业。我学的专业是前端开发，我目前还是单身，并没有女朋友，欢迎大家跟我交朋友，谢谢。
      2. 仔细阅读上面的这段话，如果让你提取这段文字的中心意思，你能做到吗？我相信对一个正常人来说都不是难事。可是，这段话如果交给计算机去处理呢？  
         计算机没有人类的思维，这件事对它来讲恐怕就太难了。  
         因此，有时候我们才需要将数据结构化。结构化的数据是什么样子呢？  
         以上面那段话为例，我们将它的信息提取出来，以表格的形式展现：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | 年龄 | 性别 | 毕业院校 | 所在地 | 性格 | 专业 |
| 张三 | 23 | 男 | 清华 | 北京 | 开朗 | 前端 |

这样一来，我们把一段复杂的信息，进行了结构化处理，瞬间显得清爽多了。计算机处理起来也相当方便。因此，这样的数据，我们就称之为结构化数据。

1. 关系型数据库
   1. 数据库管理软件是什么？
      1. 一种对数据库文件进行管理的软件，通常简称数据库。
   2. 常见的数据库管理软件有哪些？
      1. Oracle、MySQL、SQLServer、DB2……  
         这些都是我们所谓的关系型数据库，还有非关系型的数据库。  
         例如：Redis,Tokyo Cabinet,Cassandra,Voldemort,MongoDB，Dynomite，HBase,CouchDB,Hypertable,Riak,Tin,Flare,Lightcloud,KiokuDB,Scalaris,Kai,ThruDB……
   3. 那么关系型数据库和非关系型数据库又是指什么呢？
      1. 我们仍然以结构化的数据为例来说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 身份证 | 银行卡号 | 余额 | 消费记录 |
| 张三 | 11011211111 | 111111111 | 111111.0 | 2017.01.01 14：20：56-10000RMB买大象 2017.01.03 19：10：11-200RMB看电影…… |

* + 1. 这样的数据看起来是不是很别扭呢？因为它查询起来极不方便。  
       因此我们把它改造一下：

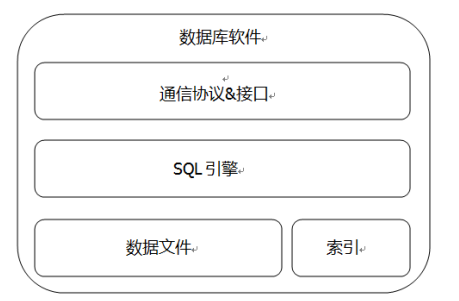
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 姓名 | 身份证 | 银行卡号 | 余额 |
| 1 | 张三 | 111111 | 1111111 | 9999999.0 |
| 2 | 李四 | 222222 | 2222222 | 8888888.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 时间 | 消费金额 | 备注 | 用户编号 |
| 1 | 2017.01.01 | 10000 | 买大象 | 1 |
| 2 | 2017.01.02 | 260 | 看电影 | 2 |
| 3 | 2017.01.03 | 16000 | 首饰 | 1 |

我们把用户账号的信息，和消费记录信息分离开来，通过用户编号进行了关联，既把不同的数据进行了分离，使得查询数据更加方便。同时又保证了它们关系的正确性。这就是传统的关系型数据库。其中用户编号我们称之为外键。

非关系型数据库是一种泛指。  
KV（键值）数据库，例如缓存  
列存储数据库，例如分布式文件系统等等

1. 结构化查询语言介绍
   1. 为什么数据库软件需要端口号？为什么说它是一种服务？
      1. 首先，你要理解数据库软件最大的作用是什么。  
         它最大的功能在于，将数据结构化以后，可以对大量的数据提供非常便利的查询方式。也就是我们接下来要讲的这门查询语言：Struct Query Language  
         是的，为了更方便的查询，它单独发明了一种查询语言，就是我们常说的SQL语言。  
         目前也是W3C的规范之一。  
         不过这种语言及其的简单，号称最接近自然语言的编程，因此不会带很大的学习成本。
      2. 一个关系型数据库软件的结构大致如下：

  
因为实际当中，数据库都是通过网络远程访问的，它必须以提供服务的方式，让我们可以本地或远程连接

1. 数据库管理软件如何安装？
2. MySQL介绍
   1. 一种开源的数据库软件，市场占有率很高，现在属于Oracle公司
   2. mysql默认端口3306
   3. mysql数据库如何存储数据？
      1. 以库为单位，在库中，以表格为单位，在表中分为行和列
   4. 如何使用Mysql呢？
      1. 安装完成后，打开Mysql的命令行工具窗口，输入你安装时设定的密码
   5. 常见命令：查看目前为止，有多少库存在？
      1. show databases;
   6. 了解数据类型
      1. 创建表格的时候，每一列都有数据类型的要求，数据库中类型跟编程中叫法不太一样，但意思是一样的。
         1. int 整数
         2. varchar 字符串
         3. blob 二进制数据
         4. date 日期
   7. SQL语句（struct query language）结构化查询语言
      1. 这种语言的发明，给我们查询数据提供了极大的便利，它也是一种标准，从1992到1999版，因此，所有的数据库提供的SQL语言，基本都是一致的。
      2. 为什么说基本呢？因为SQL标准当中提供的功能实在太少，因此对于非标准之外的其他功能，数据库之间就各不相同了。
3. 建库
   1. create database zzl;
4. 查看所有数据库
   1. show databases;
5. 打开某一个库
   1. use zzl;
6. 创建一张表
   1. create table students{  
       id int unsigned not null auto\_increment primary key,  
       name char(8) not null,  
       sex char(4) not null,  
       age tinyint unsigned not null,  
       tel char(13) null default “-“  
       };
7. 查看所有表
   1. show tables;
8. 查看某张表的结构
   1. desc students;
9. 插入一条记录insert
   1. insert [into] 表名 [(列名，列名2，列名3，…)] values(值1，值2，值3,…);
   2. insert into students values(NULL, "王刚", "男", 20, "13811371377");  
      insert into students (name, sex, age) values("孙丽华", "女", 21);
10. 删除delete
    1. delete from 表名称 where 删除条件
    2. delete from students;  
       delete from students where id=2;  
       delete from students where age<20;
11. 修改update
    1. update表名称 set 列名称=新值 where 更新条件;
    2. update students set tel=default where id=5;  
       update students set age=age+1;  
       update students set name="张伟鹏", age=19 where tel="13288097888";
12. 查询select
    1. select 列名称 from 表名称[查询条件];
    2. select name, age from students;  
       select \* from students;  
       select \* from students where age > 21;  
       select \* from students where name like "%王%";  
       select \* from students where id<5 and age>20;
13. 练习  
    create table EMP  
    (   
    EMPNO numeric(4) PRIMARY KEY,  
    ENAME VARCHAR(10),  
    JOB VARCHAR(9),  
    MGR numeric(4),  
    HIREDATE DATE,  
    SAL numeric(7,2),  
    COMM numeric(7,2),  
    DEPNO numeric(4)  
    );  
    CREATE TABLE Dept(  
    DEPTNO numeric(4),  
    DNAME VARCHAR(14),  
    LOC VARCHAR(13)  
    );  
    CREATE TABLE Salgrade  
    (   
    GRADE numeric,  
    LOSAL numeric,  
    HISAL numeric  
    );  
    INSERT INTO Dept VALUES (10,'ACCOUNTING','NEW YORK');  
    INSERT INTO Dept VALUES (20,'RESEARCH','DALLAS');  
    INSERT INTO Dept VALUES (30,'SALES','CHICAGO');  
    INSERT INTO Dept VALUES (40,'OPERATIONS','BOSTON');  
    INSERT INTO EMP VALUES  
    (7369,'SMITH','CLERK',7902,'1980-12-17',800,null,20);  
    INSERT INTO EMP VALUES  
    (7499,'ALLEN','SALESMAN',7698,'1981-02-20',1600,300,30);  
    INSERT INTO EMP VALUES  
    (7521,'WARD','SALESMAN',7698,'1981-02-22',1250,500,30);  
    INSERT INTO EMP VALUES  
    (7566,'JONES','MANAGER',7839,'1981-04-02',2975,NULL,20);  
    INSERT INTO EMP VALUES  
    (7654,'MARTIN','SALESMAN',7698,'1981-09-28',1250,1400,30);  
    INSERT INTO EMP VALUES  
    (7698,'BLAKE','MANAGER',7839,'1981-05-01',2850,NULL,30);  
    INSERT INTO EMP VALUES  
    (7782,'CLARK','MANAGER',7839,'1981-06-09',2450,NULL,10);  
    INSERT INTO EMP VALUES  
    (7839,'KING','PRESIDENT',NULL,'1981-11-17',5000,NULL,10);  
    INSERT INTO EMP VALUES  
    (7844,'TURNER','SALESMAN',7698,'1981-09-08',1500,0,30);  
    INSERT INTO EMP VALUES  
    (7900,'JAMES','CLERK',7698,'1981-12-03',950,NULL,30);  
    INSERT INTO EMP VALUES  
    (7902,'FORD','ANALYST',7566,'1981-12-03',3000,NULL,20);  
    INSERT INTO EMP VALUES  
    (7934,'MILLER','CLERK',7782,'1982-01-23',1300,NULL,10);  
    INSERT INTO SALGRADE VALUES (1,700,1200);  
    INSERT INTO SALGRADE VALUES (2,1201,1400);  
    INSERT INTO SALGRADE VALUES (3,1401,2000);  
    INSERT INTO SALGRADE VALUES (4,2001,3000);  
    INSERT INTO SALGRADE VALUES (5,3001,9999);  
    #要求列出每个雇员的姓名及年薪  
    SELECT ename,sal\*12 FROM emp;  
    #查看每月可以得到奖金的雇员信息  
    SQL> SELECT \* FROM emp WHERE comm is NOT NULL;  
    #要求基本工资大于1500，同时可以领取奖金的雇员信息  
    SQL> SELECT \* FROM emp WHERE sal>1500 AND comm is NOT NULL;  
    #查询在1981年雇佣的全部雇员信息，BETWEEN .. AND 包含等于的情况  
    SQL> SELECT \* FROM emp WHERE hiredate BETWEEN '01-JAN-81' AND '31-DEC-81';  
    #要求查询出雇员编号不是 7369、7499的雇员信息  
    SQL> SELECT \* FROM emp WHERE empno NOT IN(7369,7499);  
    #SQL中LIKE语句要注意通配符 % 和 \_  
    SQL> SELECT \* FROM emp WHERE hiredate LIKE '%81%';  
    #要求对雇员的工资由低到高进行排序，升序为默认(ASC)，降序(DESC)  
    SQL> SELECT \* FROM emp ORDER BY sal;  
    找出佣金高于薪金的60%的员工  
    SQL> SELECT \* FROM emp WHERE comm>sal\*0.6  
    找出部门10中所有经理(MANAGER)和部门20中所有办事员(CLERK)的详细资料  
    SQL> SELECT \* FROM emp WHERE (deptno=20 AND job='MANAGER') OR (deptno=10 AND job='CLERK');  
    找出既不是经理又不是办事员但其薪金大于或等于2000的所有员工的资料  
    SQL> SELECT \* FROM emp WHERE job NOT IN('MANAGER','CLERK') AND sal >= 2000;  
    显示不带有"R"的员工的姓名  
    SQL> SELECT ename FROM emp WHERE ename NOT LIKE '%R%';

第二节：应用

1. 掌握安装、建库、建表
2. 掌握基本增删改查

第三节

1. 主键及外键
   1. 主键(primary key) 能够唯一标识表中某一行的属性或属性组。一个表只能有一个主键，但可以有多个候选索引。主键常常与外键构成参照完整性约束，防止出现数据不一致。主键可以保证记录的唯一和主键域非空,[数据库](http://lib.csdn.net/base/mysql)管理系统对于主键自动生成唯一索引，所以主键也是一个特殊的索引。
   2. 外键（foreign key） 是用于建立和加强两个表数据之间的链接的一列或多列。外键约束主要用来维护两个表之间数据的一致性。简言之，表的外键就是另一表的主键，外键将两表联系起来。一般情况下，要删除一张表中的主键必须首先要确保其它表中的没有相同外键（即该表中的主键没有一个外键和它相关联）。
2. SQL条件查询
   1. 查询数值型数据:  
       SELECT \* FROM tb\_name WHERE sum > 100;  
       查询谓词:>,=,<,<>,!=,!>,!<,=>,=<
   2. 询字符串  
       SELECT \* FROM tb\_stu  WHERE sname  =  '小刘'
   3. 查询逻辑型数据  
       SELECT \* FROM tb\_name WHERE type = 'T'  
       SELECT \* FROM tb\_name WHERE type = 'F'  
       逻辑运算符:and or not

第四节：应用

1. 掌握商品表SQL的增删改查

第五节：综合应用

1. 了解数据库的使用
2. 了解SQL语句的编写
3. 了解PHP连接数据库