河南工程学院

《工程师应用能力综合训练》实习实训

成果报告

惠普实习报告

学生学号： 201410913152

学生姓名： 陈双杰

学 院： 计算机学院

专业班级： 计算机科学与技术1441

专业课程： 工程师应用能力综合训练

指导教师： 李华

1. 年 12 月 5 日

# 惠普实习报告

# 实习目的和安排

随着数字时代的来临，高薪酬、好就业等多个优点，计算机行业越来越受到求职者的青睐，对于我们计算机专业的学生来说，竞争越来越大，企业对我们的要求也越来越高，所以，我们必须拥有比别人更好地技术，才能在众多的求职者中脱颖而出。

本次实习的地点是洛阳惠普软件基地，实习期为四个月，目的是让即将毕业的应届生们把从学校学到的理论知识应用到实践中来，培养学生们的实践能力，为毕业做准备。

我所在的班级是开发班，课程安排主要是java语言、oracle数据库，课程共分两个阶段，第一阶段主要学习基础，第二阶段是项目开发，就业方向是java开发工程师。目前，我们还在进行第一阶段的基础学习，为第二阶段做准备。

# Java课程

## Java初体验

Java语言的特点：是一种面向对象的语言，是一种平台无关的语言提供程序运行的解释环境，多线程，安全性。

开发环境的下载、安装、配置：下载并安装JDK，选择合适的文本编辑器或使用IDE(集成开发环境)，配置path&classpath，验证：运行命令：java，javac，java -version(查看版本)。

使用Eclipse开发java程序：集成开发环境，集成开发环境（IDE）是一类软件，它将程序开发，环境和程序调试环境集合在一起，帮助程序员开发软件。

使用Eclipse开发简单java程序步骤：在Eclipse中创建java项目，在项目中创建java类，在main()方法中编写java代码，在Eclipse中运行程序。

Java的运行机制：java语言编写.java文件，java编译器翻译为jvm可以理解的.class字节码文件，java解释器读取文件翻译为计算机可以执行的代码。

## **数据类型与运算符**

标识符：标识符是在程序中自定义的一些名称，由大小写字母[a-zA-Z]，数字[0-9]，下划线[ \_ ]，特殊字符[$]组成。数字不可以开头，不可以使用关键字，严格区分大小写。

关键字：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [abstract](http://baike.baidu.com/view/122814.htm" \t "_blank) | [assert](http://baike.baidu.com/view/653925.htm" \t "_blank) | [boolean](http://baike.baidu.com/view/1229867.htm" \t "_blank) | break | [byte](http://baike.baidu.com/view/44243.htm" \t "_blank) |
| case | [catch](http://baike.baidu.com/view/1908899.htm" \t "_blank) | [char](http://baike.baidu.com/view/1006519.htm" \t "_blank) | [class](http://baike.baidu.com/view/76711.htm" \t "_blank) | const |
| continue | [default](http://baike.baidu.com/view/1109945.htm" \t "_blank) | [do](http://baike.baidu.com/view/126968.htm" \t "_blank) | [double](http://baike.baidu.com/view/860124.htm" \t "_blank) | [else](http://baike.baidu.com/view/2249891.htm" \t "_blank) |
| [enum](http://baike.baidu.com/view/827326.htm" \t "_blank) | [extends](http://baike.baidu.com/view/745501.htm" \t "_blank) | [final](http://baike.baidu.com/view/2116821.htm" \t "_blank) | [finally](http://baike.baidu.com/view/137061.htm" \t "_blank) | float |
| [for](http://baike.baidu.com/view/124948.htm" \t "_blank) | goto | [if](http://baike.baidu.com/view/127156.htm" \t "_blank) | [implements](http://baike.baidu.com/view/2424683.htm" \t "_blank) | [import](http://baike.baidu.com/view/2117022.htm" \t "_blank) |
| [instanceof](http://baike.baidu.com/view/1989052.htm" \t "_blank) | [int](http://baike.baidu.com/view/804413.htm" \t "_blank) | [interface](http://baike.baidu.com/view/334756.htm" \t "_blank) | long | native |
| new | [package](http://baike.baidu.com/view/1006568.htm" \t "_blank) | [private](http://baike.baidu.com/view/51615.htm" \t "_blank) | [protected](http://baike.baidu.com/view/833512.htm" \t "_blank) | [public](http://baike.baidu.com/view/1022383.htm" \t "_blank) |
| [return](http://baike.baidu.com/view/1350512.htm" \t "_blank) | [strictfp](http://baike.baidu.com/view/1866622.htm" \t "_blank) | [short](http://baike.baidu.com/view/981206.htm" \t "_blank) | [static](http://baike.baidu.com/view/536145.htm" \t "_blank) | [super](http://baike.baidu.com/view/496937.htm" \t "_blank) |
| [switch](http://baike.baidu.com/view/600161.htm" \t "_blank) | [synchronized](http://baike.baidu.com/view/1207212.htm" \t "_blank) | [this](http://baike.baidu.com/view/626297.htm" \t "_blank) | [throw](http://baike.baidu.com/view/3019126.htm" \t "_blank) | [throws](http://baike.baidu.com/item/throws" \t "_blank) |
| [transient](http://baike.baidu.com/view/1698173.htm" \t "_blank) | try | [void](http://baike.baidu.com/view/1004734.htm" \t "_blank) | [volatile](http://baike.baidu.com/view/608706.htm" \t "_blank) | [while](http://baike.baidu.com/view/1455003.htm" \t "_blank) |

基本数据类型：数值型和非数值型。

数值型有四种：short、byte、int、long。非数值型有四种：double、float、char、boolean。

基本类型的级别：byte/short/char，三个平级，运算时自动提升为int；低到高排序：int->long->float->double。

自动类型转换：两种类型要兼容，数值类型（整型和浮点型）互相兼容，目标类型大于源类型。

强制类型转换：目标类型小于源类型。语法：（数据类型）表达式。

常量：常量是在程序运行中，值不能改变的量。

变量：声明变量，根据数据类型在内存申请空间；赋值，将数据存储至对应的内存空间；使用变量，获取数据值。

运算符：算数运算符、赋值运算符、比较运算符、逻辑运算符、位运算符、条件运算符、三目运算符。

运算符的优先级：算数运算符 > 位运算符 > 比较运算符 > 逻辑运算符 > 条件运算符 > 赋值运算符。

|  |  |  |
| --- | --- | --- |
| **优先级** | **运算符** | **描述** |
| 1 | （） | 括号 |
| 2 | +、- | 正负号 |
| 3 | ++、--、！ | 单目运算符 |
| 4 | \*、/、% | 乘除 |
| 5 | +、- | 加减 |
| 6 | <<、>>、>>> | 移位运算 |
| 7 | <、<=、>、>= | 比较大小 |
| 8 | ==、！= | 是否相等 |
| 9 | & | 按位与运算 |
| 10 | ^ | 按位异或运算 |
| 11 | | | 按位或运算 |
| 12 | && | 非简洁逻辑与运算 |
| 13 | || | 非简洁逻辑或运算 |
| 14 | = | 赋值运算 |

## **条件结构**

if、if-else条件选择结构:根据条件判断结果选择不同的处理。

switch选择结构:根据等值条件判断结果选择不同的处理。

switch和多重if选择结构的异同点如下：

相同点：都是用来处理多分支条件的结构。

不同点：switch选择结构，只能处理等值条件判断的情况；多重if选择结构，适合某个变量处于某个连续区间时的情况。

条件结构代码练习：

|  |
| --- |
| public class CondetionTest {  @Test  public void test3(){  int i=800;  int j=800;  if(i>j){  System.out.println(i+"大于"+j);  }else if(i<j){  System.out.println(i+"小于"+j);  }else if(i==j){  System.out.println(i+"等于"+j);  }  }  @Test  public void testSwitch1(){  Scanner input = new Scanner(System.in);  System.out.println("请输入一个月份：");  int month=input.nextInt();  switch(month){  case 3:  case 4:  case 5:  System.out.println("春季");  break;  case 6:  case 7:  case 8:  System.out.println("夏季");  break;  case 9:  case 10:  case 11:  System.out.println("秋季");  break;  case 12:  case 1:  case 2:  System.out.println("冬季");  break;  default:  System.out.println("月份输入有误！");  }  input.close();  }  //三目运算符  @Test  public void test4(){  int score =8;  String result=score>90?"Iphone8":"";  System.out.println(result);  }  } |

## **循环结构**

判断结构包括 while语句、do-while语句、for语句。

while和do while的区别：while是先判断后执行，而do while是先执行后判断，就是说，使用do while时，无论条件是否成立，循环体都会执行一次。

结束循环的方法包括break、continue。

break和continue的区别：break是跳出改循环继续执行循环后边的内容，如果是多层循环只能跳出本层循环；continue结束本次循环，直接执行下次循环。

循环结构代码练习：

|  |
| --- |
| public class Test {  //输出会员卡的各个位数，并计算各个位数的总和  @Test  public void card(){  Scanner scanner=new Scanner(System.in);  System.out.println("请输入四位数的会员卡号:");  String str=scanner.next();  System.out.println("会员卡号是:"+str);  char[] num=str.toCharArray();  System.out.print("千位数是："+num[0]+",");  System.out.print("百位数是："+num[1]+",");  System.out.print("十位数是："+num[2]+",");  System.out.print("个位数是："+num[3]+","+"\n");  System.out.print("会员卡号");  for(int i=0;i<num.length;i++){  System.out.print(num[i]);  }  int sum = 0;  for(int i=0;i<num.length;i++){  sum+=num[i]-48; //num是一个字符，要转成int ‘1’=49  System.out.println(sum);  }  System.out.print("各位数之和是："+sum);  }  //九九乘法表  @Test  public void test3(){  for(int i=1;i<=9;i++){  for(int j=1;j<=i;j++){  System.out.print(j+"\*"+i+"="+(i\*j)+"\t");  }  System.out.println();  }  }  //水仙花123  @Test  public void test4(){  int gw=0,sw=0,bw=0;  for(int i=100;i<=999;i++){  gw=i%10;  sw=i/10%10;  bw=i/100;  if(i==(bw\*bw\*bw+sw\*sw\*sw+gw\*gw\*gw)){  System.out.println(i);  }  }  }  //星号输出直角三角形  @Test  public void test5(){  Scanner input=new Scanner(System.in);  System.out.println("请输入一个整数：");  int num=input.nextInt();  for(int i=num;i>=0;i--){  for(int j=0;j<i;j++){  System.out.print("\*");  }  System.out.println();  }  }  } |

## **数组**

数组的定义：数组是一个变量，存储相同数据类型的一组数据。

数组的结构和基本要素：标识符，数组的名称，用于引用不同的数组；数组元素，向数组中存放的数据；元素下标，对数组元素进行编号，数组下标从0开始；元素类型，数组元素的数据类型。

Arrays类：Arrays类提供了操作数组的一些static方法。

代码练习：

|  |
| --- |
| public class ArrayTest {  //计算五个学生的总分、平均分、最低分、最高分、从大到小排序  @Test  public void test1(){  int arr[] = new int[5];  Scanner input=new Scanner(System.in);  for(int i=0;i<5;i++){  System.out.println("请输入第"+(i+1)+"个学生的成绩：");  arr[i]=input.nextInt();  }  int sum=0;  for(int s:arr){  sum+=s;  }  System.out.println("总成绩："+sum);  //最低分、最高分  int max=0,min=100;  for(int j=0;j<arr.length;j++){  if(max<arr[j]){  max=arr[j];  }  if(min>arr[j]){  min=arr[j];  }  }  System.out.println("最高分："+max+",最低分："+min);  //冒泡排序  for(int x=0;x<arr.length-1;x++){  for(int y=0;y<arr.length-1-x;y++){  if(arr[y]<arr[y+1]){  int temp=arr[y];  arr[y]=arr[y+1];  arr[y+1]=temp;  }  }  }  System.out.print("排序结果为：");  for(int array:arr){  System.out.print(array+",");  }  }  } |

|  |
| --- |
| //分别计算三个班学生的成绩  @Test  public void test7(){  int[][] array={{80,90,100},{100,82,90},{90,90,80}};  for(int i=0;i<array.length;i++){  System.out.print((i+1)+"班");  Arrays.sort(array[i]);  System.out.println("成绩排序后：");  for(int j=0;j<array[i].length;j++){  System.out.println(array[i][j]);  }  }  }  //打印扑克牌  @Test  public void test8(){  String[] types={"黑桃","红桃","梅花","方块"};  String[] values={"A","2","3","4","5","6","7","8","9","10","J","Q","K"};  for(int i=0;i<types.length;i++){  for(int j=0;j<values.length;j++){  System.out.print(types[i]+values[j]+"\t");  }  System.out.println();  }  }  //Array  @Test  public void test8(){  int[] scores1={100,2,35,43};  int[] scores2={1,2,3,43,1};  System.out.println(Arrays.equals(scores1, scores2));  System.out.println(Arrays.toString(scores1));  System.out.println(Arrays.toString(scores2));  Arrays.sort(scores1);  Arrays.sort(scores1);  System.out.println(Arrays.toString(scores1));  System.out.println("================================");  System.out.println(Arrays.toString(scores2));  Arrays.fill(scores2, 1);  System.out.println(Arrays.toString(scores2));  System.out.println("---------------------------------");  int[] b=Arrays.copyOf(scores1, 10);  System.out.println(Arrays.toString(b));  System.out.println("---------------------------------");  System.out.println(Arrays.toString(scores1));  int index=Arrays.binarySearch(scores1, 36);  System.out.println(index);  } |

## **面向对象**

对象是java语言的核心，在java程序中“万事万物皆对象”。对象可以看成是静态属性（成员变量）和动态属性（方法）的封装体类是用来创建同一类型的对象的模板，在一个类中该类对象具有的成员变量及方法。J2SDK提供了很多类供编程人。

成员变量可以使用java语言中任何一种数据类型（包括基本数据类型和引用类型），定义成员变量时可以对其进行初始化，如果没有初始化，java会默认进行初始，成员变量的作用范围为整个类体。

对象的创建和使用：必须使用new关键字创建对象，使用对象（引用）成员变量或来引用对象的成员变量，使用对象（引用）方法（参数列表）来调用对象的方法，同一类的每个对象有不同的成员变量储存空间，同一类的每个对象共享该类的方法。

成员变量和局部变量的区别：作用域不同，局部变量的作用域仅限于定义它的方法，成员变量的作用域在整个类内部都是可见的。初始值不同，Java会给成员变量一个初始值，Java不会给局部变量赋予初始值。

方法的重载是指一个类中可以定义相同的名字，但参数不同的对个方法。调用时，会根据不同的参数表选择对应的方法。

访问控制修饰符：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 访问修饰符 | 本类 | 同包 | 子类 | 其他 |
| private | √ |  |  |  |
| 默认 | √ | √ |  |  |
| protected | √ | √ | √ |  |
| public | √ | √ | √ | √ |

抽象类：抽象类可以将已经实现的方法提供给其子类使用，使代码可以被复用，抽象类中的抽象方法在子类中重写，保证了子类还具有自身的独特性，通过抽象类指向其子类的对象，可以实现多态。用abstract关键字来修饰一个类时，这个类叫抽象类；用abstract来修饰一个方法时，该方法叫抽象方法。含有抽象方法的类必须被声明为抽象类，抽象类必须被继承，抽象方法必须被重写，抽象类不能被实例化。抽象方法只需声明，而不需实现。

面向对象编程：

|  |
| --- |
| //计算坐标中两点的距离  public class MyPoint {  private double x;  private double y;  public MyPoint() {  this.x=0;  this.y=0;  }  public MyPoint(double x, double y) {  this.x = x;  this.y = y;  }  public static double distance1(MyPoint p){  return Math.hypot(p.x, p.y);  }  public double distance(MyPoint p1,MyPoint p2){  return Math.hypot(p1.x-p2.x, p1.y-p2.y);  }  public static void main(String[] args) {  MyPoint m0=new MyPoint();  MyPoint m=new MyPoint(10,9);  double a0=m0.distance1(m);  System.out.println(a0);  double a=m.distance(m,m0);  System.out.println(a);  }  }  //计算图形的面积  public class GeoObjectTest {  public static void main(String[] args) {  Scanner input=new Scanner(System.in);  Circle c1=new Circle();  System.out.println("请输入圆的半径：");  double r=input.nextDouble();  c1.setRadius(r);  c1.getArea();  c1.getPerimeter();  System.out.println("请输入圆的半径：");  double r2=input.nextDouble();  System.out.println("请输入圆的颜色：");  String color=input.next();  Circle c2=new Circle(r2,color,true);  c2.getArea();  c2.printInfo();  }  }  public class Circle extends GeoObject{  private double radius;  private final double pi=3.14;  //计算面积  public double getArea(){  return pi\*radius\*radius;  }  //计算周长  public double getPerimeter(){  return pi\*2\*radius;  }  //计算直径  public double getDiameter(){  return 2\*radius;  }  //输出圆的信息  public void printCircle(){  System.out.println("圆的创建时间："+new Date());  System.out.println("是否填充颜色："+filled);  System.out.println("圆的颜色："+color);  System.out.println("圆的半径："+radius);  }  public Circle(double radius,String color,boolean filled) {  super();  this.radius = radius;  this.color=color;  this.filled=filled;  }  public Circle(double radius) {  super();  this.radius=radius;  }  public double getRadius() {  return radius;  }  public void setRadius(double radius) {  this.radius = radius;  }  } |

# 三、Oracle课程

## **Oracle基础**

1.数据库

数据库概念：数据库是一种软件产品，是用于存放数据，管理数据的存储仓库，是有效组织在一起的数据集合。

数据库的分类：关系型数据库和非关系型数据库。

常用数据库软件：大型数据库oracle、中小型数据库Sql Server Microsoft SQL Server、mysql、小型数据库Access Microsoft Office Access。

2.Oracle 开发工具

slqplus是 oracle 自带的工具软件，pl/sql developer：属于第三方软件，主要用于开发，测试，优化 oracle pl/sql 的存储。

3.Oracle用户管理

sys 用户：超级管理员（群主），权限最大，有create database 的权限。

system 用户：普通管理员用户（普通群管理员），没有create database 的权限。

scott 用户：普通用户（普通的群成员）在默认的情况下，scott 用户是锁定状态(lockuser)，一般我们启用它。如果安装的时候，忘记对scott用户解锁，比如scott可以通过 system来对该用户解锁。

创建用户:在oracle中要创建一个新的用户使用create user语句，一般是具有 dba(数据库管理员)的权限才能使用。

4.Oracle权限管理

权限分为系统权限与对象权限。

系统权限：是数据库管理相关的权限： create session(登录权限)、 create table(创建表权限)、create index(创建索引权限)、create view(创建视图权限)、create sequence(创建序列权限)、create trriger(创建触发器权限)

对象权限： 是用户操作数据对象相关的权限。比如对表的增删改查（insert 增、delete 删、update 改、select查）。

创建的新用户是没有任何权限的，甚至连登录的数据库的权限都没有，需要为其指定相应的权限。所以我们需要给用户分配权限，命令：grant create session（登录权限）to 用户名；也可以按角色给用户分配权限，命令：grant resource（角色名）to 用户名；

5.Oracle角色管理

角色分为：预定义角色和自定义角色。

预定义角色：把常用的权限集中起来，形成角色。常见的角色有：dba、 connect、resource等。dba 角色：是授予系统管理员的，拥有该角色的用户就能成为系统管理员了，它拥有所有的系统权限。connnect 角色：是授予用户的最基本的权利，能够连接到数据库中，并在对其他用户的表有访问权限时。resource 角色：具有创建表、序列、视图等权限。

自定义角色：按需定制一定权限形成角色，该角色初始没有任何权限。

6.代码练习

|  |
| --- |
| 1.解锁scott  alter user scott account unlock;  2.创建 qc，并赋予 connect 和 resource权限  create user qc identified by 123456;  grant connect,resource to qc;  3.切换用户  conn qc;  4.qc修改密码  alter user qc identified by 123;  5.回收权限  revoke connect,resource from qc;  6.删除用户  drop user qc[cascade]; |

## **表的操作**

1.Oracle 数据类型

char(size):存放字符串，它最大可以存放 2000 个字符，是定长.

varchar2(size):存放字符串，它最大可以存放 4000 个字符，是变长。

nchar(n):以 Unicode 编码来存放字符串，它最大可以存放 2000 个字符，是定长。

nvarchar(n):以 Unicode 编码来存放字符串，它最大可以存放 4000 个字符，是变长。

clob 型:字符型大对象，它最大可以存放 8TB，是变长。

blob 型：二进制数据，可存放图片、声音，它最大可以存放 8TB，是变长。

number 型：存放整数，也可存放小数，是变长。

date 类型：用于表示时间， (年/月/日/时/分/秒)，是定长。

2.表空间（Oracle 独有）

表空间： 表存在的空间，一个表空间是指向具体的数据文件。创建表空间的目的：主要是为了提高数据库的管理性能。

表空间属性： 一个数据库可以包含多个表空间，一个表空间只能属于一个数据库；一个表空间包含多个数据文件，一个数据文件只能属于一个表空间；表空间可以划分成更细的逻辑存储单元。

|  |
| --- |
| --创建表空间  create tablespace qc\_tp  datafile 'D:\csj.dbf'  size 20m  autoextend on next 5m  maxsize unlimited; |

3.表结构操作

|  |
| --- |
| --建表  create table student(  id number,  name varchar2(20),  sex char(2),  birthday date,  score number(3,1),  resume clob  );  --修改表结构  --给学生表添加班级编号classid  alter table student add(classid number);  --学生姓名变成 varchar2(30)  alter table student modify(name varchar2(30));  --删除学生表的 score 字段  alter table student drop (score,resume);  --修改表的名称  rename student to stu;  --修改列名  alter table student rename column birthday to birth; |

## **Oracle约束**

1.约束的定义和功能

约束的定义：是强加在表上的规则或条件。确保数据库满足业务规则。保证数据的完整。

约束的功能：实现一些业务规则，防止无效的垃圾数据进入数据库，维护数据库的完整性。从而使数据库的开发和维护都更加容易。

2.约束的分类

约束分为：非空（NOT NULL）约束、唯一（UNIQUE）约束、主键（PRIMARY KEY）约束、外键（FOREIGN KEY）约束、条件（CHECK）约束。

|  |
| --- |
| create table customer(  customerId number primary key,  name varchar2(32) not null,  address varchar2(64),  email varchar2(64) unique,  sex char(2) default'男' check(sex in('男','女')),  idCard varchar2(20)  ); |

3.维护约束

|  |
| --- |
| --增加 purchase 表主键；  alter table purchase add constraint PK\_PURCHASE\_PID primary key (cId,gId);  --客户的姓名不能为空值； --增加商品名也不能为空  alter table goods modify goodsName not null;  --邮箱不能够重复； --增加身份证也不重复  alter table customer add constraint unique\_cardId unique(cardId);  alter table customer add constraint UK\_EMAIL unique(email);  --增加客户的住址只能是(伊滨区， 老城区，洛龙区)  alter table customer add constraint check\_address check(address in('伊滨区','老城区','洛龙区')); |

## **（四）Oracle查询基础**

|  |
| --- |
| --显示每个雇员的年工资  select ename, sal\*13+comm\*13 from emp;  --nvl()去空函数nvl(comm,0)  select ename, sal\*13+nvl(comm,0)\*13 from emp;  --使用列的别名  select ename, sal\*13+nvl(comm,0)\*13 年薪 from emp;  select ename, sal\*13+nvl(comm,0)\*13 as 年薪 from emp;  select ename, sal\*13+nvl(comm,0)\*13 as "年薪" from emp;  --如何显示 empno 为 123,345,800...的雇员情况  select \* from emp where empno in (7369,7788,7698);  --如何显示没有上级的雇员的情况  select ename from emp where mgr is null;  select ename from emp where mgr is not null;  select ename ,sal from emp;  --显示部门平均工资低于2000的部门号和它的平均工资  select deptno,avg(sal) from emp group by deptno having avg(sal) < 2000; |

## **（五）Oracle常用函数**

1.字符函数

|  |
| --- |
| --把字符转换成对应的十进制数  select ascii('a') space from dual;  --输入整数，返回对应的字符  select chr(54740) c1,chr(47600) c2 from dual;  --连接两个字符串  select concat('hello','world') from dual;  --返回字符串并将字符串的第一个字母变成大写  select initcap('hello') upp from dual;  --在一个字符串中搜索指定的字符，返回发现指定的字符的位置；  select instr('helloworld','o',1,2) instring from dual;  --返回字符串的长度  select length(ename),sal,length(to\_char(sal)) from emp where ename='SMITH';  --返回字符串，并将所有的字符小写；  select lower('HELLO') from dual;  --返回字符串，并将所有的字符大写；  select upper('hello') from dual;  --rpad 和 lpad(粘贴字符)  select lpad(rpad('hello',10,'\*'),15,'\*') from dual;  --除去字符，ltrim 和 rtrim、trim  select ltrim(rtrim(' hello ',' '),' ') from dual;  select trim(' hello ') from dual;  --取字符串'world'  select substr('helloworld',6,5) from dual;  --替换  select replace('helloworld','hello','你好') from dual; |

2.数学函数

|  |
| --- |
| --向上取整  select ceil(44.4) from dual;  select ceil(-44.4) from dual;  --向下取整  select floor(44.4) from dual;  select floor(-44.4) from dual;  --按照指定的精度截取一个数  select trunc(333.499,2) from dual;  --按照指定的精度进行舍入  select round(44.6),round(-44.6) from dual;  --返回指定值的绝对值  select abs(100),abs(-100) from dual; |

3.日期函数

|  |
| --- |
| --当前月份增加两个月  select add\_months(sysdate,2) from dual;  --请查找最近 350 个月入职的员工  select ename,hiredate from emp where add\_months(hiredate,350)>=sysdate;  --按一定的格式返回当前日期和下一天  select to\_char(sysdate,'yyyy-mm-dd'),to\_char((sysdate)+1,'yyyy-mm-dd') from dual;  --返回本月的最后一天  select to\_char(last\_day(sysdate),'yyyy-mm-dd') from dual;  --两个时间中的月份  select months\_between('19-10月-2017','19-10月-1996') mon\_between from dual;  --计算下一个星期一是几号  select next\_day('21-11月-2017','星期一') next\_day from dual;  --当前是周几  select to\_char(sysdate,'day') from dual; |

4.转换类型函数

|  |
| --- |
| --日期类型转换成字符串格式  select to\_char(sysdate,'yyyy-mm-dd hh24:mi:ss') from dual;  --将字符串转换成日期  select to\_date('2017-11-21','yyyy-mm-dd') from dual;  --将给出的数字类型的字符转换为数字  select to\_number('2017') year from dual; |

5.聚合函数

|  |
| --- |
| --求所有部门平均工资  select avg(all sal) from emp;  --求有部门的总工资  select sum(all sal) from emp;  --所有员工工资最高的  select ename,max(sal) from emp;  --所有员工中工资最低的  select ename,min(sal) from emp;  --员工表中总记录数  select count(\*) from emp;  --分组查询  select deptno,count(\*),sum(sal) from emp group by deptno having count(\*)>=5; |

## **（六）Oracle复杂查询**

1.多表查询

|  |
| --- |
| --显示部门为 10 的部门名、员工名和工资  select d.dname,e.ename,e.sal from dept d,emp e where d.deptno=e.deptno and e.deptno=10;  --显示雇员名，雇员工资及所在部门的名字，并按部门排序  select e.ename,e.sal,d.dname from emp e,dept,d where d.deptno=e.deptno order by d.dname; |

2.自连接

|  |
| --- |
| 是指在同一张表的连接查询(把一张表看作两张表)  --显示员工的上级领导的姓名  select e1.ename,e2.ename from emp e1,emp e2 where e1.mgr=e2.empno;  --查询前五名员工的姓名和工资  select ename, sal , rownum from emp where rownum<=5; |

3.外连接(左外连接和右外连接)

|  |
| --- |
| 左外连接  --显示员工的上级领导的姓名  select e1.ename,e2.ename from emp e1,emp e2 where e1.mgr=e2.empno(+);  select e1.ename,e2.ename from emp e1 left join emp e2 on e1.mgr=e2.empno;  右外连接  --显示员工的上级领导的姓名  select e1.ename,e2.ename from emp e1,emp e2 where e2.empno=e1.mgr(+);  select e1.ename,e2.ename from emp e2 rigth join emp e1 on e2.empno=e1.mgr; |

4.子查询

|  |
| --- |
| --单行子查询：显示与 smith 同一部门的所有员工  select ename,deptno  from emp  where deptno=(select deptno from emp where ename='SMITH');  --多行子查询：如何查询和部门 10 的工作相同的雇员的名字、岗位、工资、部门号  select ename,job,sal,deptno  from emp  where job in (select distinct job from emp where deptno=10);  --多列子查询：查询与 smith 的部门和岗位完全相同的所有雇员  select \* from emp  where deptno=  (select deptno from emp where ename='SMITH')  and job=(select job from emp where ename='SMITH');  --显示高于自己部门平均工资的员工的信息  select \* from emp e,  (select avg(sal) mysal,deptno from emp group by deptno) t1  where e.deptno=t1.deptno and e.sal > t1.mysal  order by e.deptno; |

5.合并查询

|  |
| --- |
| --并集：工资大于2500和职位是MANAGER的员工名、工资、职位  select ename,sal,job from emp where sal>2500  union  select ename,sal,job from emp where job='MANAGER';  --取所有：工资大于2500和职位是MANAGER的员工名、工资、职位  select ename,sal,job from emp where sal>2500  union all  select ename,sal,job from emp where job='MANAGER'  --交集：工资大于2500并且职位是MANAGER的员工名、工资、职位  select ename,sal,job from emp where sal>2500  intersect  select ename,sal,job from emp where job='MANAGER';  --差集：工资大于2500并且职位不是MANAGER的员工名、工资、职位  select ename,sal,job from emp where sal>2500  minus  select ename,sal,job from emp where job='MANAGER'; |

6.分页查询

|  |
| --- |
| 分页查询，就是将过多的结果在有限的界面上分好多页来显示。  --以职位分组，找出平均工资最高的两种职位  select t2.\* from  (select t1.\*,rownum rn from  (select avg(sal) myavg from emp group by job order by avg(sal) desc)t1 )t2  where t2.rn<=2 ;  --查询薪水排名在3到7之间的员工  select t2.\* from  (select t1.\*,rownum rn from  (select ename,sal from emp order by sal asc) t1) t2  where rn >=3 and rn <=7; |

# 总结

通过这一个月的实习，我清楚的认识到在学校所学知识的重要性，现在回想起来觉得以前任课老师所讲的每一本书、带领我们做的每一次实训都是我们想要深入发展的基础，“万变不离其宗”，就是这个道理，以后所学都是建立在专业课的基础之上的。同时，我也认识到自己在实践方面是多么的欠缺，在学习的过程中会遇到一些问题，但是在老师和同学们的帮助下，都一一得到了解决，以至于让我在这里的每一天都有很大的收获。

十分感谢学校给了我们这个实习的机会，让我们有机会把在学校学习的基础知识应用于实践中来。未来的几个月，我会更加的努力，不辜负自己也不浪费学校给我们提供的这个机会。