1、学校图书馆借书信息管理系统建立三个表：

学生信息表：student

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型** | **说明** |
| stuID | char(10) | 学生编号，主键 |
| stuName | Varchar(10) | 学生名称 |
| major | Varchar(50) | 专业 |

图书表：book

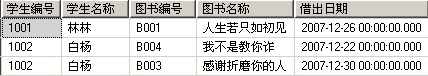
|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型** | **说明** |
| BID | char(10) | 图书编号，主键 |
| title | char(50) | 书名 |
| author | char(20) | 作者 |

借书信息表：borrow

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型** | **说明** |
| borrowID | char(10) | 借书编号，主键 |
| stuID | char(10) | 学生编号，外键 |
| BID | char(10) | 图书编号，外键 |
| T\_time | datetime | 借书日期 |
| B\_time | datetime | 还书日期 |

请编写SQL语句完成以下的功能：

1. 查询“计算机”专业学生在“2007-12-15”至“2008-1-8”时间段内借书的学生编号、学生名**称、图书编号、图书名称、借出日期；参考查询结果如**下图所示：



SELECT

b.stuID AS '学生编号',

c.stuName AS '学生姓名',

b.BID AS '图书编号',

a.title AS '书名',

b.T\_time AS '借出日期'

FROM

book a,

borrow b,

student c

WHERE

b.BID = a.BID

AND b.stuID = c.stuID

AND T\_time BETWEEN '2007-12-15'

AND '2008-1-8'

ORDER BY

b.stuID ASC

1. 查询所有借过图书的学生编号、学生名称、专业；参考查询结果如下图所示：



SELECT b.stuId AS '学生编号',b.stuName AS '学生姓名',b.major AS '专业' FROM(

(SELECT DISTINCT stuID FROM borrow) a

INNER JOIN

(SELECT stuID,stuName,major FROM student) b

ON a.stuID = b.stuID)

1. 查询借过作者为“安意如”的图书的学生姓名、图书名称、借出日期、归还日期；参考查询结果如下图所示：



1. 查询目前借书但未归还图书的学生名称及未还图书数量；参考查询结果如下图所示：



USE master

GO

/\*$$$$$$$$$$$$$建库$$$$$$$$$$$$$$$$$$$$$$$$\*/

--检验数据库是否存在，如果为真，删除此数据库--

IF exists(SELECT \* FROM sysdatabases WHERE name='BOOK')

DROP DATABASE BOOK

GO

CREATE DATABASE BOOK

GO

--建数据表--

USE BOOK

GO

CREATE TABLE student --学生信息表

(

stuID CHAR(10) primary key, --学生编号

stuName CHAR(10) NOT NULL , --学生名称

major CHAR(50) NOT NULL --专业

)

GO

CREATE TABLE book --图书表

(

BID CHAR(10) primary key, --图书编号

title CHAR(50) NOT NULL, --书名

author CHAR(20) NOT NULL, --作者

)

GO

CREATE TABLE borrow --借书表

(

borrowID CHAR(10) primary key, --借书编号

stuID CHAR(10) foreign key(stuID) references student(stuID), --学生编号

BID CHAR(10) foreign key(BID) references book(BID),--图书编号

T\_time datetime NOT NULL, --借出日期

B\_time datetime --归还日期

)

GO

--学生信息表中插入数据--

INSERT INTO student(stuID,stuName,major)VALUES('1001','林林','计算机')

INSERT INTO student(stuID,stuName,major)VALUES('1002','白杨','计算机')

INSERT INTO student(stuID,stuName,major)VALUES('1003','虎子','英语')

INSERT INTO student(stuID,stuName,major)VALUES('1004','北漂的雪','工商管理')

INSERT INTO student(stuID,stuName,major)VALUES('1005','五月','数学')

--图书信息表中插入数据--

INSERT INTO book(BID,title,author)VALUES('B001','人生若只如初见','安意如')

INSERT INTO book(BID,title,author)VALUES('B002','入学那天遇见你','晴空')

INSERT INTO book(BID,title,author)VALUES('B003','感谢折磨你的人','如娜')

INSERT INTO book(BID,title,author)VALUES('B004','我不是教你诈','刘庸')

INSERT INTO book(BID,title,author)VALUES('B005','英语四级','白雪')

--借书信息表中插入数据--

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T001','1001','B001','2007-12-26',null)

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T002','1004','B003','2008-1-5',null)

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T003','1005','B001','2007-10-8','2007-12-25')

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T004','1005','B002','2007-12-16','2008-1-7')

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T005','1002','B004','2007-12-22',null)

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T006','1005','B005','2008-1-6',null)

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T007','1002','B001','2007-9-11',null)

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T008','1005','B004','2007-12-10',null)

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T009','1004','B005','2007-10-16','2007-12-18')

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T010','1002','B002','2007-9-15','2008-1-5')

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T011','1004','B003','2007-12-28',null)

INSERT INTO borrow(borrowID,stuID,BID,T\_time,B\_time)VALUES('T012','1002','B003','2007-12-30',null)

# 题目2

程序员工资表：ProWage

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型** | **说明** |
| ID | int | 自动编号，主键 |
| PName | Char(10) | 程序员姓名 |
| Wage | int | 工资 |

1. 创建一个存储过程，对程序员的工资进行分析，月薪1500到10000不等，如果有百分之五十的人薪水不到2000元，给所有人加薪，每次加100，再进行分析，直到有一半以上的人大于2000元为止，存储过程执行完后，最终加了多少钱？

BEGIN

set @total = 0;

set @addNum = 100;

WHILE(2\*(SELECT count(1) FROM ProWage WHERE Wage < 2000) <= SELECT count(1) FROM ProWage)

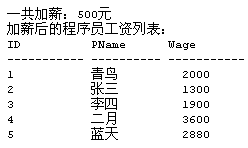
DO

UPDATE ProWage SET Wage = @addNum + Wage , @total = @total + @addNum;

END WHILE;

END

例如：如果有百分之五十的人薪水不到2000，给所有人加薪，每次加100元，直到有一半以上的人工资大于2000元，调用存储过程后的结果如图：



请编写存储过程来实现如下功能：

2.创建存储过程，查询是否有一半程序员的工资在2200、3000、3500、4000、5000或6000元之上，如果不到分别每次给每个程序员加薪100元，至之一半程序员的工资达到2200，3000，3500，4000，5000或6000元。

3.创建存储过程，查询程序员平均工资在4500元，如果不到则每个程序员每次加200元，至到所有程序员平均工资达到4500元。

|  |
| --- |
| 1. USE master 2. GO 3. /\*$$$$$$$$$$$$$建库$$$$$$$$$$$$$$$$$$$$$$$$\*/ 4. --检验数据库是否存在，如果为真，删除此数据库-- 5. IF exists(SELECT \* FROM sysdatabases WHERE name='Wage') 6. DROP DATABASE Wage 7. GO 8. CREATE DATABASE Wage 9. GO 10. --建数据表-- 11. USE Wage 12. GO 13. CREATE TABLE ProWage --程序员工资表 14. ( 15. ID int identity(1,1) primary key, --工资编号 16. PName CHAR(10) NOT NULL , --程序员姓名 17. Wage int NOT NULL --工资 18. ) 19. GO 20. --插入数据-- 21. INSERT INTO ProWage(PName,Wage)VALUES('青鸟',1900) 22. INSERT INTO ProWage(PName,Wage)VALUES('张三',1200) 23. INSERT INTO ProWage(PName,Wage)VALUES('李四',1800) 24. INSERT INTO ProWage(PName,Wage)VALUES('二月',3500) 25. INSERT INTO ProWage(PName,Wage)VALUES('蓝天',2780) |