**第一部分 挂号**

输入：

患者身份证号，患者姓名，性别，出生日期，年龄，年龄类型，家庭住址，本次看诊日期，午别，本次挂号科室ID，本次挂号医生ID，本次挂号级别ID，结算类别ID，病历本要否，挂号员ID

输出：

操作成功（success）/ 操作失败（failure）

功能描述：

1. 获取当前时间用于作为本次挂号时间字段
2. 查找可用病历号
3. 将挂号信息保存到挂号表中

IN patientIDNumber1 char(19),IN patientName1 text CHARACTER SET UTF8,IN patientSex1 char(10),IN patientBirthday1 date,IN patientAge1 int(10),IN patientAgeType1 text CHARACTER SET UTF8,IN patientAddress1 text CHARACTER SET UTF8,IN dateOfConsultation1 date,IN noon1 char(10) CHARACTER SET UTF8,IN departmentID1 char(30),IN doctorID1 char(30),IN registeredGradeID1 char(30),IN settlementCategory1 char(30),IN medicalRecordNeeded1 char(10),IN registeredStaffID1 char(30),OUT result char(30)

'350801199902026637','王益','1','1999-02-07',20,'岁','福州','2019-07-16','上','1','2','3','2','1','138',@result

BEGIN

#Routine body goes here...

DECLARE medicalRecordNumber1 int(30);

DECLARE deletedMark1 char(10);

DECLARE registrationTime1 timestamp;

DECLARE statusOfConsultation1 char(10);

DECLARE s1 char(30);

DECLARE s2 char(30);

SET s1='success!';

SET s2='failure!';

#获取当前时间用于作为本次挂号时间字段

select FROM\_UNIXTIME(UNIX\_TIMESTAMP(NOW()),'%Y-%m-%d %H:%i:%S') INTO registrationTime1;

#默认deletedMark1为1

SET deletedMark1='1';

#初始化statusOfConsultation1为1

SET statusOfConsultation1='1';

#查询该患者信息是否存在，存在则取病历号并刷新患者信息，不存在则自动生成病历号并插入患者信息到01patients

IF EXISTS(SELECT medicalRecordNumber FROM 01patients WHERE patientIDNumber LIKE patientIDNumber1)THEN

SET medicalRecordNumber1=(SELECT medicalRecordNumber FROM 01patients WHERE patientIDNumber LIKE patientIDNumber1);

UPDATE 01patients SET medicalRecordNumber=medicalRecordNumber1,patientIDNumber=patientIDNumber1,patientName=patientName1,patientSex=patientSex1,patientBirthday=patientBirthday1,patientAge=patientAge1,patientAgeType=patientAgeType1,patientAddress=patientAddress1,deletedMark=deletedMark1 WHERE medicalRecordNumber=medicalRecordNumber1;

ELSE SET medicalRecordNumber1=(SELECT medicalRecordNumber FROM 01patients ORDER BY medicalRecordNumber DESC LIMIT 1)+1;

INSERT INTO 01patients VALUES(NULL,medicalRecordNumber1,patientIDNumber1,patientName1,patientSex1,patientBirthday1,patientAge1,patientAgeType1,patientAddress1,deletedMark1);

END IF;

#插入挂号信息到14registrationForm

INSERT INTO 14registrationForm VALUES(NULL,medicalRecordNumber1,dateOfConsultation1,noon1,departmentID1,doctorID1,registeredGradeID1,settlementCategory1,medicalRecordNeeded1,registrationTime1,registeredStaffID1,statusOfConsultation1);

SET result=s1;

END

**第二部分 退号**

输入：

挂号ID

输出：

操作成功（success）/ 操作失败（failure）

功能描述：

1. 查询挂号状态
2. 条件判断是否可以退号

3-1）如果患者已经就诊，不能退号。 如果患者已经退号，不能二次退号。

3-2）如果患者未就诊，可以退号，执行退号操作，更新挂号表中的挂号状态

IN registrationID1 char(30),OUT result char(30)

206,@result

BEGIN

#Routine body goes here...

DECLARE a char(30);

DECLARE s1 char(30);

DECLARE s2 char(30);

SET s1='success!';

SET s2='failure!';

#查询挂号状态

IF EXISTS(SELECT statusOfConsultation FROM 14registrationForm WHERE registrationID=registrationID1) THEN (SELECT statusOfConsultation INTO a FROM 14registrationForm WHERE registrationID=registrationID1);

#条件判断是否可以退号

IF a LIKE '1' THEN

UPDATE 14registrationForm SET statusOfConsultation='4' WHERE registrationID=registrationID1;

SET result=s1;

ELSE SET result=s2;

END IF;

ELSE SET result=s2;

END IF;

END

**第三部分** **医生看诊**

输入：

挂号ID，诊断信息

输出：

操作成功（success）/ 操作失败（failure）

功能描述

1）根据挂号ID得到病历ID

2）保存诊断信息到病历表。

3）更新挂号的诊断状态为已诊断。

IN registrationID1 int (30),IN chiefComplaint1 char(100),IN diagnosisResult1 char(100),OUT result char(30)

206, '头疼','感冒',@result

BEGIN

#Routine body goes here...

DECLARE medicalRecordNumber1 int(30);

DECLARE medicalRecordID1 int (30);

DECLARE statusOfMedicalRecord1 char(10);

DECLARE statusOfConsultation1 char(10);

DECLARE a char(30);

DECLARE s1 char(30);

DECLARE s2 char(30);

SET s1='success!';

SET s2='failure!';

#查询挂号状态

IF EXISTS(SELECT statusOfConsultation FROM 14registrationForm WHERE registrationID=registrationID1) THEN (SELECT statusOfConsultation INTO a FROM 14registrationForm WHERE registrationID=registrationID1);

#条件判断是否可以就诊

IF a LIKE '1' THEN

#根据挂号ID得到病历号

SET medicalRecordNumber1=(SELECT medicalRecordNumber FROM 14registrationForm where registrationID=registrationID1);

#更新病历状态为已诊断，并保存诊断信息到病历表

SET statusOfMedicalRecord1='3';

INSERT INTO 15medicalRecordForm VALUES(NULL,medicalRecordNumber1,registrationID1,NULL,diagnosisResult1,NULL,statusOfMedicalRecord1);

#根据挂号ID得到病历ID,并保存信息到病历首页

SET medicalRecordID1=(SELECT medicalRecordID FROM 15medicalRecordForm WHERE registrationID=registrationID1);

INSERT INTO 16homePageOfMedicalRecord VALUES(medicalRecordID1,registrationID1,chiefComplaint1,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);

#更新挂号的诊断状态为已诊断

SET statusOfConsultation1='3';

UPDATE 14registrationForm SET statusOfConsultation=statusOfConsultation1 WHERE registrationID=registrationID1;

SET result=s1;

ELSE SET result=s2;

END IF;

ELSE SET result=s2;

END IF;

END

**第四部分 医生开药**

输入：

挂号ID，处方名称，药品列表（可以以某种格式拼接成字符串作为入参）

输出：

操作成功（success）/ 操作失败（failure）

功能描述

1）将处方的基本信息保存到处方表，包括病历ID，挂号ID，开立医生ID，处方名称，开立时间等

2）将每一条药品的划价信息保存到划价明细表中，包括划价ID, 药品ID, 药品价格，药品数量等。

IN registrationID1 int (30),IN prescriptionName1 char(30),IN drugIDList1 char(100),IN drugUsageList1 char(100),IN drugDosageList1 char(100),IN frequencyList1 char(100),IN quantityList1 char (100),OUT result char(30)

206,'病毒性感冒','1,2,3','口服,注射,冲泡','10g,10ml,100g','一日两次,一次,一日三次','4,1,6',@result

BEGIN

#Routine body goes here...

DECLARE medicalRecordID1 int(30);

DECLARE doctorID1 int(30);

DECLARE prescriptionID1 int(30);

DECLARE prescriptionTime1 timestamp;

DECLARE prescriptionStatus1 char(10);

DECLARE drugID1 int(30);

DECLARE drugUsage1 char(30);

DECLARE drugDosage1 char(30);

DECLARE frequency1 char(30);

DECLARE quantity1 int(10);

DECLARE drugStatus1 char(10);

DECLARE len int(10);

DECLARE a char(30);

DECLARE s1 char(30);

DECLARE s2 char(30);

SET s1='success!';

SET s2='failure!';

#查询挂号状态

IF EXISTS(SELECT statusOfConsultation FROM 14registrationForm WHERE registrationID=registrationID1) THEN (SELECT statusOfConsultation INTO a FROM 14registrationForm WHERE registrationID=registrationID1);

#条件判断是否可以开药

IF a LIKE '3' THEN

#根据挂号ID得到medicalRecordID1和doctorID1

SELECT doctorID INTO doctorID1 FROM 14registrationForm WHERE registrationID=registrationID1;

SELECT medicalRecordID INTO medicalRecordID1 FROM 15medicalrecordForm WHERE registrationID=registrationID1;

#获取当前时间用于作为本次开立时间字段

SELECT FROM\_UNIXTIME(UNIX\_TIMESTAMP(NOW()),'%Y-%m-%d %H:%i:%S') INTO prescriptionTime1;

#设定处方和处方明细的状态为开立

SET prescriptionStatus1='1';

SET drugStatus1='1';

#将处方的基本信息保存到处方表

INSERT INTO 17prescription VALUES(NULL,medicalRecordID1,registrationID1,doctorID1,prescriptionName1,prescriptionTime1,prescriptionStatus1);

SELECT prescriptionID INTO prescriptionID1 FROM 17prescription WHERE registrationID=registrationID1;

#划分字符串，并将处方明细保存到处方明细表

SELECT LENGTH(drugIDList1)-LENGTH(REPLACE(drugIDList1,',',''))+1 INTO len;

WHILE len>0 DO

START TRANSACTION;

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(drugIDList1,',',-len),',',1) INTO drugID1;

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(drugUsageList1,',',-len),',',1) INTO drugUsage1;

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(drugDosageList1,',',-len),',',1) INTO drugDosage1;

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(frequencyList1,',',-len),',',1) INTO frequency1;

SELECT SUBSTRING\_INDEX(SUBSTRING\_INDEX(quantityList1,',',-len),',',1) INTO quantity1;

SELECT CONVERT(drugID1,UNSIGNED INTEGER) INTO drugID1;

SELECT CONVERT(quantity1,UNSIGNED INTEGER) INTO quantity1;

INSERT INTO 18prescriptionDetails VALUES(NULL,prescriptionID1,drugID1,drugUsage1,drugDosage1,frequency1,quantity1,drugStatus1);

SET len=len-1;

END WHILE;

SET result=s1;

ELSE SET result=s2;

END IF;

ELSE SET result=s2;

END IF;

END

**第五部分 患者缴费**

输入：

处方ID，操作员ID

输出：

操作成功（success）/ 操作失败（failure）

功能描述

1. 医生需要先开立，用户再缴费，根据处方ID得到药品明细。
2. 修改成药处方表处方明细表缴费状态
3. 获取可用发票号
4. 将每一条药品的缴费信息保存到缴费明细表中，包括缴费ID, 药品ID, 药品价格，药品数量等。
5. 新增发票记录

IN prescriptionID1 char(30),IN chargingOrRefundStaffID1 char(30),IN chargingMethod1 char(30),OUT result char(30)

111,'1','1',@result

BEGIN

#Routine body goes here...

DECLARE prescriptionDetailID1 int(30);

DECLARE prescriptionStatus1 char(10);

DECLARE drugStatus1 char(10);

DECLARE invoiceAmount1 double(20,2);

DECLARE chargingOrRefundTime1 timestamp;

DECLARE registrationID1 char(30);

DECLARE unitPriceOfDrugs1 double(20,2);

DECLARE quantity1 int(10);

DECLARE invoiceNumber1 int(30);

DECLARE itemID1 int(30);

DECLARE itemCategory1 char(30);

DECLARE itemName1 char(30);

DECLARE executiveDepartmentsID1 char(30);

DECLARE a char(30);

DECLARE num int(10);

DECLARE s1 char(30);

DECLARE s2 char(30);

SET s1='success!';

SET s2='failure!';

#初始化发票金额为0.00

SET invoiceAmount1=0.00;

#获取当前时间用于作为本次收/退费时间字段

SELECT FROM\_UNIXTIME(UNIX\_TIMESTAMP(NOW()),'%Y-%m-%d %H:%i:%S') INTO chargingOrRefundTime1;

#根据处方ID获得挂号ID

SELECT registrationID INTO registrationID1 FROM 17prescription WHERE prescriptionID=prescriptionID1;

#查询处方状态

IF EXISTS(SELECT prescriptionStatus FROM 17prescription WHERE prescriptionID=prescriptionID1) THEN (SELECT prescriptionStatus INTO a FROM 17prescription WHERE prescriptionID=prescriptionID1);

#条件判断是否可以缴费

IF a LIKE '1' THEN

#更新处方为已缴费

SET prescriptionStatus1='3';

UPDATE 17prescription SET prescriptionStatus=prescriptionStatus1 WHERE prescriptionID=prescriptionID1;

#查询该处方ID对应的处方明细ID的数量

SET num=(SELECT count(prescriptionDetailID) FROM 18prescriptionDetails WHERE prescriptionID=prescriptionID1)-1;

WHILE num>=0 DO

#根据处方ID得到处方明细ID,明细状态

SELECT prescriptionDetailID,drugStatus INTO prescriptionDetailID1,drugStatus1 FROM 18prescriptionDetails WHERE prescriptionID=prescriptionID1 LIMIT num,1;

IF drugStatus1='1' THEN

#更新处方明细表相关药品的状态为已缴费

SET drugStatus1='3';

UPDATE 18prescriptionDetails SET drugStatus=drugStatus1 WHERE prescriptionDetailID=prescriptionDetailID1;

#获得该条处方明细的药品单价和数量

SELECT drugID,quantity INTO itemID1,quantity1 FROM 18prescriptionDetails WHERE prescriptionDetailID=prescriptionDetailID1;

SELECT unitPriceOfDrugs INTO unitPriceOfDrugs1 FROM 06drugs where drugID=itemID1;

#更新发票金额

SET invoiceAmount1=invoiceAmount1+unitPriceOfDrugs1\*quantity1;

ELSE SET result=s2;

END IF;

SET num=num-1;

END WHILE;

#新增发票记录

INSERT INTO 22invoices VALUES(NULL,invoiceAmount1,chargingOrRefundTime1,chargingOrRefundStaffID1,registrationID1,chargingMethod1,NULL,'0','1');

#将每一条药品的缴费信息保存到缴费明细表中

SET executiveDepartmentsID1='132';

SELECT invoiceNumber INTO invoiceNumber1 FROM 22invoices WHERE registrationID=registrationID1 AND chargingOrRefundTime=chargingOrRefundTime1;

SET num=(SELECT count(prescriptionDetailID) FROM 18prescriptionDetails WHERE prescriptionID=prescriptionID1)-1;

WHILE num>=0 DO

SELECT prescriptionDetailID INTO prescriptionDetailID1 FROM 18prescriptionDetails WHERE prescriptionID=prescriptionID1 LIMIT num,1;

SELECT drugID,quantity INTO itemID1,quantity1 FROM 18prescriptionDetails WHERE prescriptionDetailID=prescriptionDetailID1;

SELECT unitPriceOfDrugs,drugType,drugName INTO unitPriceOfDrugs1,itemCategory1,itemName1 FROM 06drugs where drugID=itemID1;

INSERT INTO 23costDetails VALUES(NULL,registrationID1,invoiceNumber1,itemID1,itemCategory1,itemName1,unitPriceOfDrugs1,quantity1,executiveDepartmentsID1,chargingOrRefundTime1,chargingOrRefundStaffID1,chargingOrRefundTime1,chargingOrRefundStaffID1,chargingMethod1,NULL);

SET num=num-1;

END WHILE;

SET result=s1;

ELSE SET result=s2;

END IF;

ELSE SET result=s2;

END IF;

END

**第六部分 患者退费**

输入：

处方ID，发票号，操作员ID

输出：

操作成功（success）/ 操作失败（failure）

功能描述

1）更新成药处方明细表状态为已退费

2）在缴费明细表中添加若干明细记录，包括缴费ID, 药品ID, 药品价格，药品数量等, 其中药品价格为负数。

3）新增冲正发票记录

IN prescriptionID1 int(30),IN invoiceNumber1 int(30),IN chargingOrRefundStaffID1 char(30),IN chargingMethod1 char(30),OUT result char(30)

111,800902,'1','1',@result

BEGIN

#Routine body goes here...

DECLARE costDetailID1 int (30);

DECLARE prescriptionDetailID1 int(30);

DECLARE prescriptionStatus1 char(10);

DECLARE drugStatus1 char(10);

DECLARE invoiceAmount1 double(20,2);

DECLARE invoiceAmount2 double(20,2);

DECLARE chargingOrRefundTime1 timestamp;

DECLARE registrationID1 char(30);

DECLARE unitPriceOfItems1 double(20,2);

DECLARE quantity1 int(10);

DECLARE itemID1 int(30);

DECLARE itemCategory1 char(30);

DECLARE itemName1 char(30);

DECLARE executiveDepartmentsID1 char(30);

DECLARE a char(30);

DECLARE num int(10);

DECLARE s1 char(30);

DECLARE s2 char(30);

SET s1='success!';

SET s2='failure!';

#查询处方状态

IF EXISTS(SELECT prescriptionStatus FROM 17prescription WHERE prescriptionID=prescriptionID1) THEN (SELECT prescriptionStatus INTO a FROM 17prescription WHERE prescriptionID=prescriptionID1);

#条件判断是否可以退费

IF a LIKE '3' THEN

#初始化重开发票金额为0.00

SET invoiceAmount2=0.00;

#获取当前时间用于作为本次退费时间字段

SELECT FROM\_UNIXTIME(UNIX\_TIMESTAMP(NOW()),'%Y-%m-%d %H:%i:%S') INTO chargingOrRefundTime1;

#更新处方为已退费

SET prescriptionStatus1='4';

UPDATE 17prescription SET prescriptionStatus=prescriptionStatus1 WHERE prescriptionID=prescriptionID1;

#查询该处方ID对应的处方明细ID的数量

SET num=(SELECT count(prescriptionDetailID) FROM 18prescriptionDetails WHERE prescriptionID=prescriptionID1)-1;

WHILE num>=0 DO

#根据处方ID得到处方明细ID,状态,项目ID

SELECT prescriptionDetailID,drugStatus,drugID INTO prescriptionDetailID1,drugStatus1,itemID1 FROM 18prescriptionDetails WHERE prescriptionID=prescriptionID1 LIMIT num,1;

IF drugStatus1='3' THEN

#更新处方明细表相关药品的状态为已退费

SET drugStatus1='4';

UPDATE 18prescriptionDetails SET drugStatus=drugStatus1 WHERE prescriptionDetailID=prescriptionDetailID1;

#根据项目ID和发票号找到缴费明细中的对应项目，并在缴费明细表中添加明细记录

SELECT costDetailID,registrationID,itemCategory,itemName,unitPriceOfItems,quantity,executiveDepartmentsID INTO costDetailID1,registrationID1,itemCategory1,itemName1,unitPriceOfItems1,quantity1,executiveDepartmentsID1 FROM 23costDetails WHERE invoiceNumber=invoiceNumber1 AND itemID=itemID1;

INSERT INTO 23costDetails VALUES(NULL,registrationID1,invoiceNumber1,itemID1,itemCategory1,itemName1,-1\*unitPriceOfItems1,quantity1,executiveDepartmentsID1,chargingOrRefundTime1,chargingOrRefundStaffID1,chargingOrRefundTime1,chargingOrRefundStaffID1,chargingMethod1,costDetailID1);

ELSEIF drugStatus1='5' THEN

#获得该条处方明细的药品单价和数量

SELECT drugID,quantity INTO itemID1,quantity1 FROM 18prescriptionDetails WHERE prescriptionDetailID=prescriptionDetailID1;

SELECT unitPriceOfItems INTO unitPriceOfItems1 FROM 06drugs where drugID=itemID1;

#更新发票金额

SET invoiceAmount2=invoiceAmount2+unitPriceOfItems\*quantity1;

END IF;

SET num=num-1;

END WHILE;

#新增冲红发票

SELECT invoiceAmount,registrationID INTO invoiceAmount1,registrationID1 FROM 22invoices WHERE invoiceNumber=invoiceNumber1;

INSERT INTO 22invoices VALUES(NULL,-1\*invoiceAmount1,chargingOrRefundTime1,chargingOrRefundStaffID1,registrationID1,chargingMethod1,invoiceNumber1,'0','1');

#新增重开发票记录

IF invoiceAmount2>0 THEN

INSERT INTO 22invoices VALUES(NULL,invoiceAmount2,chargingOrRefundTime1,chargingOrRefundStaffID1,registrationID1,chargingMethod1,NULL,'0','1');

END IF;

#原发票有效状态更新为报废

UPDATE 22invoices SET effectiveStatus='2' WHERE invoiceNumber=invoiceNumber1;

SET result=s1;

ELSE SET result=s2;

END IF;

ELSE SET result=s2;

END IF;

END

**第七部分 药房发药**

输入：

病历ID

输出：

操作成功（success）/ 操作失败（failure）

功能描述

1）根据病历ID查询处方ID

2）更新处方明细表相关药品的状态为已发药

IN medicalRecordNumber1 int(30),OUT result char(30)

27,@result

BEGIN

#Routine body goes here...

DECLARE prescriptionID1 char(30);

DECLARE prescriptionStatus1 char(10);

DECLARE drugStatus1 char(10);

DECLARE a char(30);

DECLARE num int(10);

DECLARE s1 char(30);

DECLARE s2 char(30);

SET s1='success!';

SET s2='failure!';

SET result=s1;

#查询该病历ID对应的处方ID的数量

SET num=(SELECT count(prescriptionID) FROM 17prescription WHERE medicalRecordID=medicalRecordID1)-1;

WHILE num>=0 DO

#根据病历ID查询处方ID

SELECT prescriptionID INTO prescriptionID1 FROM 17prescription WHERE medicalRecordID=medicalRecordID1 LIMIT num,1;

#查询处方状态

SELECT prescriptionStatus INTO a FROM 17prescription WHERE prescriptionID=prescriptionID1;

#条件判断是否可以发药

IF a LIKE '3' THEN

#更新处方为已处理

SET prescriptionStatus1='5';

UPDATE 17prescription SET prescriptionStatus=prescriptionStatus1 WHERE prescriptionID=prescriptionID1;

#更新处方明细表相关药品的状态为已发药

SET drugStatus1='5';

UPDATE 18prescriptionDetails SET drugStatus=drugStatus1 WHERE prescriptionID=prescriptionID1;

ELSE SET result=s2;

END IF;

SET num=num-1;

END WHILE;

END