Recently, covid-19 has been developing rapidly, which poses a great challenge to medical resources around the country.In order for both sides of doctors and patients to clearly understand the information, medical resource data is needed.Please give an ODL and E/R design for the medical resource database, including the following information:

①doctor: name, ID, gender, position,workday, tel,the department that doctor *works in*.

②department: name, number, director,the hospital that the department is *part of.*

③hospital: name, address,level,size,the region that hospital *belongs to*.

④region: name, code, city.

⑤patient: name, identity, gender, tel, age, disease,the department that patient *chooses*,the hospital that patient *visits* .

Please select and specify keys for your ODL design and convert your ODL design to relational database schemes.

Please draw an E/R diagram for the former database, as well.

ODL

不考虑一家医院有分院区，即不考虑多个地址

医院之间之间没有同名情况

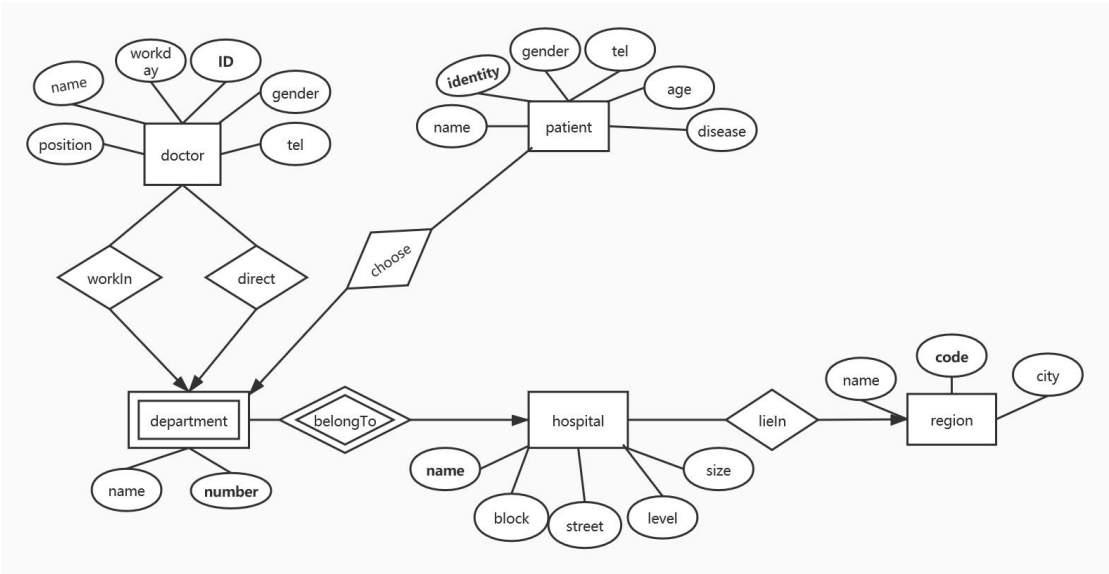
一个医生只工作于一个部门，最多领导一个部门

假设患者只在一家医院的一个部门就诊（即不考虑过往就诊记录，只记录当前）

```
interface doctor {
    attribute string ID;
    attribute string name;
    attribute char gender;
    attribute string position;
    attribute string workday;
    attribute string tel;
    relationship department workIn
        inverse department :: workers;
    relationship department direct
        inverse department :: directors;
}

interface department {
    attribute string name;
    attribute string number;
    relationship hospital belongTo
        inverse hospital :: departments;
}

interface hospital{
    attribute string name;
    attribute Struct Addr{string block, string street} address;
    attribute string level;
```

```
    attribute string size;
    relationship Set<region> lieIn
        inverse region :: hospitals;
}

interface region{
    attribute string name;
    attribute string code;
    attribute string city;
}

interface patient{
    attribute string name;
    attribute string identity;
    attribute char gender;
    attribute int age;
    attribute string disease;
    relationship department choose
        inverse department :: treat;
    relationship hospital visit
        inverse hospital :: receive;
}
```

ER（导出后下划线无法显示，所以键码加粗表示）



From ODL to RM

doctor (ID, name, gender, position, workday, tel, hospitalName, workDeptNum, directDeptNum)

department (name, number, hospitalName)

hospital (name, city, street, level, size, regionCode)

region (name, code, city)

patient (name, identity, gender, age, disease, hospitalName, deptNum)

From ER to RM

doctor (ID, name, gender, position, workday, tel)

patient (name, identity, gender, age, disease)

hospital (name, city, street, level, size)

region (name, code, city)

department (name, number, hospitalName)

workIn (ID, deptNum, hospitalName)

direct (ID, deptNum, hospitalName)

choose (identity, deptNum, hospitalName)

lieIn (hospitalName, regionCode)

下面是李子怡同学根据题目建立的数据库

需求分析

需求分析

1. 导入地区数据
   a. https://github.com/kbdxbt/area_sql
2. 注册医院
   a. 创建触发器生成默认科室
3. 添加医护人员信息
4. 患者就医（通过存储过程实现）
   a. 注册
   b. 充值
   c. 消费
   d. 注销

其他更新：
根据疫情情况，可创建

A. 医疗队
B. 临时增援医护人员

我们可以建立一个个人信息表，将医生与患者的个人信息集中放置（医生也有做为患者的可能）
Resident

ID	NAME	XXX

模拟实际就医体验，患者可在多个医院就诊注册

- 注册时产生医院唯一卡号
  Patient

ID	卡号	注册时间	医院ID

- 充值和消费以流水记录
  流水

医院ID	卡号	金额	时间	柜台（充值处/科室）

- 注销：删除注册记录，但流水依然会存在
  - 问题：如果患者选择了注销，医院是否保留患者的注册信息？

过程

CREATE DATABASE

数据库名称为 `medicalSystem`

```
DECLARE @databaseName varchar(20);
DECLARE @createSql varchar(40);
set @databaseName = 'medicalSystem'
if exists (SELECT * FROM sys.databases
   WHERE name = @databaseName)
   print 'database '+@databaseName+' already exists';
else
BEGIN
 set @createSql='CREATE DATABASE '+@databaseName
 exec(@createSql);
 print @createSql;
END
```

CREATE TABLE

1. 将 `COMMENT` 注释掉
2. `AUTO_INCREMENT` 对应SQL SERVER中的 `IDENTITY` （为了插入数据方便，我去掉了这个规则）
3. 注释掉建表最后的要求 `--ENGINE=InnoDB AUTO_INCREMENT=3750 DEFAULT CHARSET=utf8;`

```
CREATE TABLE "hy_area" (
  "id" int NOT NULL ,-- COMMENT 'ID',--AUTO_INCREMENT
  "pid" int DEFAULT NULL ,--COMMENT '父id',
  "shortname" varchar(100) DEFAULT NULL ,--COMMENT '简称',
  "name" varchar(100) DEFAULT NULL ,--COMMENT '名称',
  "merger_name" varchar(255) DEFAULT NULL ,--COMMENT '全称',
  "level" tinyint DEFAULT NULL ,--COMMENT '层级 0 1 2 省市区县',
  "pinyin" varchar(100) DEFAULT NULL ,--COMMENT '拼音',
  "code" varchar(100) DEFAULT NULL ,--COMMENT '长途区号',
  "zip_code" varchar(100) DEFAULT NULL ,--COMMENT '邮编',
  "first" varchar(50) DEFAULT NULL ,--COMMENT '首字母',
  "lng" varchar(100) DEFAULT NULL ,--COMMENT '经度',
  "lat" varchar(100) DEFAULT NULL ,--COMMENT '纬度',
  PRIMARY KEY ("id")
) --ENGINE=InnoDB AUTO_INCREMENT=3750 DEFAULT CHARSET=utf8;
INSERT INTO "hy_area" VALUES ('1', '0', '北京', '北京', '中国,北京', '1', 'beijing', '', '', 'B', '116.405285', '39.9
INSERT INTO "hy_area" VALUES ('2', '1', '北京', '北京市', '中国,北京,北京市', '2', 'beijing', '010', '100000', 'B', '1
INSERT INTO "hy_area" VALUES ('3', '2', '东城', '东城区', '中国,北京,北京市,东城区', '3', 'dongcheng', '010', '100010'
INSERT INTO "hy_area" VALUES ('4', '2', '西城', '西城区', '中国,北京,北京市,西城区', '3', 'xicheng', '010', '100032',
INSERT INTO "hy_area" VALUES ('5', '2', '朝阳', '朝阳区', '中国,北京,北京市,朝阳区', '3', 'chaoyang', '010', '100020',
```

成功导入

```
USE medicalSystem
GO

SELECT TOP(20)* FROM hy_area
```

121 %

结果 消息

	id	pid	shortname	name	merger_name	level	pinyin	code	zip_code	first	lng	lat
1	1	0	北京	北京	中国,北京	1	beijing			B	116.405285	39.904989
2	2	1	北京	北京市	中国,北京,北京市	2	beijing	010	100000	B	116.405285	39.904989
3	3	2	东城	东城区	中国,北京,北京市,东城区	3	dongcheng	010	100010	D	116.41005	39.93157
4	4	2	西城	西城区	中国,北京,北京市,西城区	3	xicheng	010	100032	X	116.36003	39.9305
5	5	2	朝阳	朝阳区	中国,北京,北京市,朝阳区	3	chaoyang	010	100020	C	116.48548	39.9484
6	6	2	丰台	丰台区	中国,北京,北京市,丰台区	3	fengtai	010	100071	F	116.28625	39.8585
7	7	2	石景山	石景山区	中国,北京,北京市,石景山区	3	shijingshan	010	100043	S	116.2229	39.90564
8	8	2	海淀	海淀区	中国,北京,北京市,海淀区	3	haidian	010	100089	H	116.29812	39.95931
9	9	2	门头沟	门头沟区	中国,北京,北京市,门头沟区	3	mentougou	010	102300	M	116.10137	39.94043
10	10	2	房山	房山区	中国,北京,北京市,房山区	3	fangshan	010	102488	F	116.14257	39.74786
11	11	2	通州	通州区	中国,北京,北京市,通州区	3	tongzhou	010	101149	T	116.65716	39.90966
12	12	2	顺义	顺义区	中国,北京,北京市,顺义区	3	shunyi	010	101300	S	116.65417	40.1302
13	13	2	昌平	昌平区	中国,北京,北京市,昌平区	3	changping	010	102200	C	116.2312	40.22072
14	14	2	大兴	大兴区	中国,北京,北京市,大兴区	3	daxing	010	102600	D	116.34149	39.72668
15	15	2	怀柔	怀柔区	中国,北京,北京市,怀柔区	3	huairou	010	101400	H	116.63168	40.31602

导入结果

```
if exists (select * from sysobjects where name = 'Hospital')
    PRINT 'ALREADY EXISTS Hospital'
else
BEGIN
CREATE TABLE [dbo].[Hospital](
 医院名称 nvarchar(40) NOT NULL UNIQUE,
 医院等级 nvarchar(4) NULL,
医院类型 nvarchar(4) NULL,
省 nvarchar(10) NULL,
市 nvarchar(10) NULL,
县 nvarchar(20) NULL,
床位数 int NULL,
医院地址 nvarchar(60) NULL,
--邮编 char(6) NOT NULL,
constraint pk_Hospital PRIMARY KEY
          NONCLUSTERED(医院名称),
--CONSTRAINT fk_Hospital FOREIGN KEY
--          REFERENCES Region(邮编)
)
```

%

结果 消息

医院名称	医院等级	医院类型	省	市	县	床位数	医院地址
河北大学附属医院	三级甲等	综合医院	河北省	保定市	莲池区	2300	保定市莲池区裕华东路212号
保定市第一中心医院	三级甲等	综合医院	河北省	保定市	莲池区	2100	保定市莲池区长城北大街320号
沧州市人民医院	三级甲等	综合医院	河北省	沧州市	新华区	3000	沧州市清池大道7号
沧州市中心医院	三级甲等	综合医院	河北省	沧州市	新华区	4700	沧州市新华中路201号
承德医学院附属医院	三级甲等	综合医院	河北省	承德市	双桥区	2400	承德市南营子大街36号
兰州大学第二附属医院	三级甲等	综合医院	甘肃省	兰州市	城关区	3500	城关区萃英门80号
大连医科大学附属第一医院	三级甲等	综合医院	辽宁省	大连市	沙河口区	3700	大连市中山路222号
恩施州中心医院	三级甲等	综合医院	湖北省	恩施土家族苗族自治州	恩施市	2300	恩施市舞阳大街158号(西医部)；恩施市航空大道178号(中医部)
佛山市第一人民医院	三级甲等	综合医院	广东省	佛山市	南海区	2500	佛山市大佛南路3号

触发器创建默认科室

默认科室意味着每增添一个新的医院，就为其创建几个默认的科室。原本想到通过游标遍历表的方法实现，但在网上看到人们推荐SQL使用集合形式实现，而减少使用游标。
因此我首先创建了一个初始科室表：

初始科室

复制

```
CREATE TABLE defaultDepartment(
number int NOT NULL IDENTITY(001,1),
name nvarchar(20) NOT NULL
CONSTRAINT pk_defaultDepartment PRIMARY KEY( number)
)
END
INSERT INTO defaultDepartment VALUES
 ('门诊部'),
 ('住院部'),
 ('急诊部'),
 ('药房'),
 ('收费室'),
 ('化验室'),
 ('放射科'),
 ('手术室'),
 ('B超室'),
 ('行政楼'),
 ('后勤科室')
```

科室表

```
CREATE TABLE Department(
number int NOT NULL,
name nvarchar(20) NOT NULL,
hospitalName  nvarchar(40) NOT NULL
CONSTRAINT pk_department PRIMARY KEY( number,hospitalName),
CONSTRAINT fk_Hospital FOREIGN KEY(hospitalName)
  REFERENCES Hospital(医院名称) ON DELETE CASCADE
)
END
```

复制

CREATE TRIGGER

触发器例题分析：数据库每日一题（3）触发器

```
CREATE TRIGGER  trigger_defaultDepartment ON Hospital
AFTER INSERT
AS
BEGIN
 INSERT INTO Department
  SELECT D.number,D.name,I.医院名称 FROM defaultDepartment D, inserted I
END
```

实现展示

	number	name	hospitalName
1	1	门诊部	保定市第一中心医院
2	1	门诊部	沧州市人民医院
3	1	门诊部	沧州市中心医院
4	1	门诊部	承德医学院附属医院
5	1	门诊部	河北大学附属医院
6	2	住院部	保定市第一中心医院
7	2	住院部	沧州市人民医院
8	2	住院部	沧州市中心医院
9	2	住院部	承德医学院附属医院

CREATE PROC

```
CREATE PROC findZipCode
AS
BEGIN
 UPDATE Hospital
 SET 邮编= zip_code
 FROM Hospital left join hy_area
  on 市=name
END
GO
```

实现思路

通过对应医院信息中的市名与地区信息中的name

109 %

结果 消息

	医院名称	医院等级	医院类型	省	市	县	床位数	医院地址	邮编
1	河北大学附属医院	三级甲等	综合医院	河北省	保定市	莲池区	2300	保定市莲池区裕华东路212号	NULL
2	保定市第一中心医院	三级甲等	综合医院	河北省	保定市	莲池区	2100	保定市莲池区长城北大街320号	NULL
3	沧州市人民医院	三级甲等	综合医院	河北省	沧州市	新华区	3000	沧州市清池大道7号	NULL
4	沧州市中心医院	三级甲等	综合医院	河北省	沧州市	新华区	4700	沧州市新华中路201号	NULL
5	承德医学院附属医院	三级甲等	综合医院	河北省	承德市	双桥区	2400	承德市南营子大街36号	NULL

	医院名称	省	市	县	shortname	name	zip_code
1	保定市第一中心医院	河北省	保定市	莲池区	保定	保定市	071052
2	沧州市人民医院	河北省	沧州市	新华区	沧州	沧州市	061001
3	沧州市中心医院	河北省	沧州市	新华区	沧州	沧州市	061001
4	承德医学院附属医院	河北省	承德市	双桥区	承德	承德市	067000
5	河北大学附属医院	河北省	保定市	莲池区	保定	保定市	071052

另：可能会出现没有匹配结果的情况，可以通过 `EXCEPT` 找到相应条目并手动更新，或通过游标遍历，使用算法寻找。

结果展示

	医院名称	医院等级	医院类型	省	市	县	床位数	医院地址	邮编
1	河北大学附属医院	三级甲等	综合医院	河北省	保定市	莲池区	2300	保定市莲池区裕华东路212号	NULL
2	保定市第一中心医院	三级甲等	综合医院	河北省	保定市	莲池区	2100	保定市莲池区长城北大街320号	NULL
3	沧州市人民医院	三级甲等	综合医院	河北省	沧州市	新华区	3000	沧州市清池大道7号	NULL
4	沧州市中心医院	三级甲等	综合医院	河北省	沧州市	新华区	4700	沧州市新华中路201号	NULL
5	承德医学院附属医院	三级甲等	综合医院	河北省	承德市	双桥区	2400	承德市南营子大街36号	NULL

	医院名称	医院等级	医院类型	省	市	县	床位数	医院地址	邮编
1	河北大学附属医院	三级甲等	综合医院	河北省	保定市	莲池区	2300	保定市莲池区裕华东路212号	071052
2	保定市第一中心医院	三级甲等	综合医院	河北省	保定市	莲池区	2100	保定市莲池区长城北大街320号	071052
3	沧州市人民医院	三级甲等	综合医院	河北省	沧州市	新华区	3000	沧州市清池大道7号	061001
4	沧州市中心医院	三级甲等	综合医院	河北省	沧州市	新华区	4700	沧州市新华中路201号	061001
5	承德医学院附属医院	三级甲等	综合医院	河北省	承德市	双桥区	2400	承德市南营子大街36号	067000

过程

随机身份数据

为了快速将数据导入数据库中测试代码效果，在GitHub上找到随机生成身份信息的项目：https://github.com/gh0stkey/RGPerson，是python代码，运行效果如下：

```
[17]: run RGPerson.py
```

```
姓名：袁霏咏
年龄：49
性别：女
身份证：430181197107141205
手机号：15266685736 联通
组织机构代码：66835524-X
统一社会信用代码：A91101113788602560
单位性质：外国常驻新闻机构
```

为了方便插入数据，我们一方面通过循环获取多条随机身份信息；另一方面直接以SQL语句的格式输出，便可以直接通过复制粘贴放到SQL中执行。修改后的主函数如下：

复制

```
if __name__ == '__main__':
for i in range(1000):
        age = random.randint(16,60) #可调整生成的年龄范围（身份证），这边是16-60岁
        gender = random.randint(0,1)
        sex = u"男" if gender == 1 else u"女"
        print("INSERT INTO Resident VALUES ('{0}',{1} ,'{2}','{3}','{4}','{5}','{6}','{7}','{8}')".format(genName(),
```

打印结果如下：

```
run RGPerson.py
INSERT INTO Resident VALUES ('罗鞠',59 ,'男','141024196106185155','18456634850','移动','37831221-
7','52110101141045026K','个体工商户')
INSERT INTO Resident VALUES ('殷唉',58 ,'女','620602196212042543','17887826850','联通','19361359-
0','12110229775829623E','外国常驻新闻机构')
INSERT INTO Resident VALUES ('窦饶翌',50 ,'男','510421197001027818','18732404220','移动','84080838
-7','A11101063613750518','港澳台地区旅游部门常驻内地（大陆）代表机构')
INSERT INTO Resident VALUES ('邹兼',31 ,'男','210200198912278539','14548316995','电信','70153429-
1','53130000266001051C','军队事业单位')
INSERT INTO Resident VALUES ('罗刍',41 ,'女','211081197907250228','17877755518','电信','38651717-
5','49130302268215221L','基层工会')
```

此外，为符合医疗场景，我们将单位性质生成函数作如下修改，把原本为**其他**的一些条目改为**医护工作者**：

```
# 生成统一社会信用代码
def genCreditCode():
    orgCode = {"1":"机构编制","2":"外交","3":"教育","4":"公安","5":"民政","6":"司法","7":"交通运输","8":"文化","9":"工商","A":"中央军委改革和编制办公室","N":"农业","Y":"其他"}

    icCode = {"1":{"1":"机关","2":"事业单位","3":"中央编办直接管理机构编制的群众团体","9":"医院：护士长"},"2":{"1":"外国常驻新闻机构","9":"医院：医师"},"3":{"1":"律师执业机构","2":"公证处","3":"基层法律服务所","4":"司法鉴定机构","5":"仲裁委员会","9":"医院：院长"},"4":{"1":"外国在华文化中心","9":"医院：药剂师"},"5":{"1":"社会团体","2":"民办非企业单位","3":"基金会","9":"医院：主治医师"},"6":{"1":"外国旅游部门常驻机构代表机构","2":"港澳台地区旅游部门常驻内地（大陆）代表机构","9":"医院：会计"},"7":{"1":"宗教活动场所","2":"宗教院校","9":"医院：检验员"},"8":{"1":"基层工会","9":"其他"},"9":{"1":"企业","2":"个体工商户","3":"农民专业社","9":"医院：主任医师"},"A":{"1":"军队事业单位","9":"其他"},"N":{"1":"组级集体经济组织","2":"村级集体经济组织","3":"乡镇集体经济组织","9":"医院：副主任医师"},"Y":{"1":"医院：护士"}}
```

实现结果

最终导入数据库中执行语句如下：

145 %

结果 消息

	姓名	年龄	性别	身份证	手机号	运营商	组织机构代码	统一社会信用代码	单位性质
1	姚拾镭	35	女	110105198507090981	13026341062	联通	86642621-3	34130105769712963F	医院：医师
2	葛晢	22	女	120110199808261729	18075460813	联通	65819320-8	341302301922659266	事业单位
3	窦燎侄	29	女	120114199106252188	17699749022	移动	10369057-7	131301834019344290	医院：检验员
4	尤汩	16	男	120225200407305474	17811531733	移动	21624262-3	531100006362424005	宗教活动场所
5	施竦	47	女	130131197307315607	17522495440	移动	75125356-X	791304064326169760	外国常驻新闻机构
6	伍婶	60	男	130207196009093795	14721140980	移动	44408294-0	N111010485321790 6R	基层法律服务所

医生信息

CREATE TABLE Doctor

doctor (***ID***, position, workday,***hospitalName***, ***workDeptName***,directDeptNumber)

```
CREATE TABLE Doctor
(
ID char(18) NOT NULL ,
position nvarchar(20) NULL,
workday nvarchar(20) NULL,
hospitalName nvarchar(40) NOT NULL ,
workDeptNumber int NOT NULL,
directDeptNumber int NULL,
 CONSTRAINT pk_doctor PRIMARY KEY(ID,workDeptNumber,hospitalName),
 CONSTRAINT fk_doctor_ID FOREIGN KEY(ID)
 REFERENCES Resident(身份证) ON DELETE CASCADE,
 CONSTRAINT fk_doctor_workDept FOREIGN KEY(workDeptNumber,hospitalName)
 REFERENCES Department(number,hospitalName) ON DELETE CASCADE
)
GO
```

导入数据

为简化过程，我们把此数据库中所有的医护工作者都分配到同一家医院。并采用随机函数 `rand()` 将他们随机分配至不同科室。

```
INSERT INTO Doctor (ID ,position,hospitalName,workDeptNumber)
SELECT 身份证,substring(单位性质,4,len(单位性质)),hospitalName,number
FROM Department ,Resident
WHERE
hospitalName='河北大学附属医院' and
convert(int,substring(身份证,10,3))%11+1=number and
单位性质 like '医院%'
```

120 %

结果 消息

	ID	position	workday	hospitalName	workDeptNumber	directDeptNumber
1	110105198507090981	医师	NULL	河北大学附属医院	2	NULL
2	120114199106252188	检验员	NULL	河北大学附属医院	8	NULL
3	130503199502264572	主任医师	NULL	河北大学附属医院	8	NULL
4	130600197312201041	护士	NULL	河北大学附属医院	5	NULL
5	130721196707248541	主任医师	NULL	河北大学附属医院	4	NULL
6	130981197305038164	主任医师	NULL	河北大学附属医院	9	NULL
7	140109199306106903	副主任医师	NULL	河北大学附属医院	10	NULL
8	140110196107064554	护士	NULL	河北大学附属医院	9	NULL
9	140202197910233388	医师	NULL	河北大学附属医院	9	NULL
10	140603196610084521	护士长	NULL	河北大学附属医院	6	NULL

“升职”

我们默认主任医师都是所在科室的管理者

```
UPDATE Doctor SET directDeptNumber=workDeptNumber
WHERE position='主任医师'
```

结果 消息

	ID	position	workday	hospitalName	workDeptNumber	directDeptNumber
1	110105198507090981	医师	NULL	河北大学附属医院	2	NULL
2	120114199106252188	检验员	NULL	河北大学附属医院	8	NULL
3	130503199502264572	主任医师	NULL	河北大学附属医院	8	8
4	130600197312201041	护士	NULL	河北大学附属医院	5	NULL
5	130721196707248541	主任医师	NULL	河北大学附属医院	4	4
6	130981197305038164	主任医师	NULL	河北大学附属医院	9	9
7	140109199306106903	副主任医师	NULL	河北大学附属医院	10	NULL
8	140110196107064554	护士	NULL	河北大学附属医院	9	NULL
9	140202197910233388	医师	NULL	河北大学附属医院	9	NULL

“排班”

设定几个工作时间，用临时表存储，通过随机函数安排！

```
CREATE TABLE #workday
(
code int IDENTITY PRIMARY KEY,
workday nvarchar(20)
)
INSERT INTO #workday VALUES
 ( '一、三、五'),
 ( '二、四、六'),
 ('三、日'),
 ('四、日')
UPDATE Doctor SET workday=#workday.workday
FROM Doctor,#workday
WHERE
convert(int,substring(ID,10,3))%4+1=code
```

最终结果

结果 消息

	ID	position	workday	hospitalName	workDeptNumber	directDeptNumber
1	110105198507090981	医师	四、日	河北大学附属医院	2	NULL
2	120114199106252188	检验员	三、日	河北大学附属医院	8	NULL
3	130503199502264572	主任医师	三、日	河北大学附属医院	8	8
4	130600197312201041	护士	一、三、五	河北大学附属医院	5	NULL
5	130721196707248541	主任医师	四、日	河北大学附属医院	4	4
6	130981197305038164	主任医师	二、四、六	河北大学附属医院	9	9
7	140109199306106903	副主...	三、日	河北大学附属医院	10	NULL
8	140110196107064554	护士	四、日	河北大学附属医院	9	NULL
9	140202197910233388	医师	三、日	河北大学附属医院	9	NULL
10	140603196610084521	护士长	三、日	河北大学附属医院	6	NULL
11	140723197702104513	护士	三、日	河北大学附属医院	10	NULL
12	14103020010620632X	检验员	三、日	河北大学附属医院	8	NULL
13	150105199705185481	主治医师	二、四、六	河北大学附属医院	2	NULL

患者信息

CREATE TABLE Patient

Patient(***ID*** ,***hospitalName***, 卡号 ,注册时间)

```
CREATE TABLE Patient
(
ID char(18) NOT NULL ,
hospitalName nvarchar(40) NOT NULL,
卡号 nvarchar(40) NULL,
注册时间 datetime default getdate(),
CONSTRAINT pk_patient PRIMARY KEY(ID,hospitalName),
CONSTRAINT fk_patient_ID FOREIGN KEY(ID)
 REFERENCES Resident(身份证) ON DELETE CASCADE,
CONSTRAINT fk_patient_hospital FOREIGN KEY(hospitalName)
 REFERENCES Hospital(医院名称) ON DELETE CASCADE
)
```

消费记录

CREATE TABLE Tally

Tally(***hospitalName*** （医院） ,**卡号** ,金额 ,**时间**, ***departmentNumber*** （柜台:充值处/科室）)

```
CREATE TABLE Tally
(
hospitalName nvarchar(40) NOT NULL,
卡号 nvarchar(40) NOT NULL,
金额 money NULL,
时间 datetime default getdate(),
departmentNumber int NOT NULL,
CONSTRAINT pk_tally PRIMARY KEY(departmentNumber,hospitalName,卡号,时间),
CONSTRAINT fk_tally_department FOREIGN KEY(departmentNumber,hospitalName)
 REFERENCES Department(number,hospitalName) ON DELETE CASCADE,
)
```

患者操作

如之前所述，我们通过**存储过程** PROCEDURE实现，其实与编程语言中的函数概念相似，一旦生成，可以反复调用。
存储过程例题分析：数据库每日一题 （4）存储过程

注册

CREATE PROC proc_register

```
/**************注册*****************/
--提供个人姓名与身份证号与所在医院即可注册（选取Resident表中的数据）
CREATE PROC proc_register
@name nvarchar(20),
@ID char(18),
@hospitalName nvarchar(40)
AS
SET nocount ON --不返回计数
BEGIN
 if exists(SELECT * FROM Resident WHERE 身份证=@ID and 姓名=@name)
 BEGIN
  if exists(SELECT * FROM Patient WHERE ID=@ID)
   print @ID+'already registered'
  else
  BEGIN
   DECLARE @hospitalNumber int
   SELECT @hospitalNumber=ROW_NUMBER() over(order by 医院名称 desc)
   FROM Hospital
   INSERT INTO Patient(ID,hospitalName,卡号)
   VALUES(@ID,@hospitalName,convert(nvarchar(4),@hospitalNumber)+@ID)
  END
 END
```

实现展示

```
SELECT * FROM Patient
EXECUTE proc_register '郝掂','340321196507203290','河北大学附属医院'
SELECT * FROM Patient
```

ID	hospitalName	卡号	注册时间
340321196507203290	河北大学附属医院	5340321196507203290	2020-04-09 17:51:04.663

充值

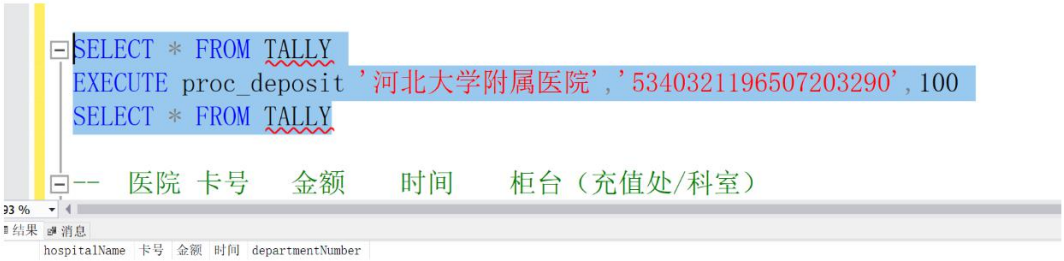
我们默认充值是从收费部进行，通过字符匹配找到相应的部门编号。`WHERE hospitalName=@hospitalName and name like '%费%' or name like '%值%'`

CREATE PROC proc_deposit

```
CREATE PROC proc_deposit
@hospitalName nvarchar(40),
@cardNumber nvarchar(40),
@money money
AS
SET nocount ON --不返回计数
BEGIN
 if exists (SELECT * FROM Patient
     WHERE 卡号=@cardNumber)
 BEGIN
 DECLARE @departmentNumber int
 SELECT @departmentNumber=number FROM department
  WHERE hospitalName=@hospitalName  and
   name like '%费%' or name like '%值%'
 --  SELECT @departmentNumber
 INSERT INTO Tally(hospitalName,卡号,金额,departmentNumber) VALUES
  (@hospitalName,@cardNumber,@money,@departmentNumber)
 END
END
```

实现展示

```
SELECT * FROM TALLY
EXECUTE proc_deposit '河北大学附属医院','5340321196507203290',100
SELECT * FROM TALLY
```



hospitalName	卡号	金额	时间	departmentNumber
河北大学附属医院	5340321196507203290	100.00	2020-04-09 18:20:18.010	5

消费

消费与充值基本一致，只是需要提供消费的部门名称

CREATE PROC proc_charge

```
CREATE PROC proc_charge
@hospitalName nvarchar(40),
@cardNumber nvarchar(40),
@money money,
@departmentName nvarchar(20)
AS
SET nocount ON --不返回计数
BEGIN
 if exists (SELECT * FROM Patient
     WHERE 卡号=@cardNumber)
 BEGIN
 DECLARE @departmentNumber int
 SELECT @departmentNumber=number FROM department
  WHERE hospitalName=@hospitalName  and
   name = @departmentName
 --  SELECT @departmentNumber
 INSERT INTO Tally(hospitalName,卡号,金额,departmentNumber) VALUES
  (@hospitalName,@cardNumber,@money,@departmentNumber)
 END
END
GO
```

注销

注销就相对简单，提供必要的信息，删除注册记录。

CREATE PROC proc_logout

```
CREATE PROC proc_logout
@name nvarchar(20),
@ID char(18),
@hospitalName nvarchar(40)
AS
SET nocount ON --不返回计数
BEGIN
 if exists(SELECT * FROM Resident WHERE 身份证=@ID and 姓名=@name)
 BEGIN
  DELETE FROM Patient
     WHERE ID =@ID and
      hospitalName=@hospitalName
 END
END
GO
```

实现展示

```
SELECT * FROM PATIENT
EXECUTE proc_logout'郝掂','340321196507203290','河北大学附属医院'
SELECT * FROM PATIENT
```

```
SELECT * FROM PATIENT
EXECUTE proc_logout'郝掂','340321196507203290','河北大学附属医院'
SELECT * FROM PATIENT
```

ID	hospitalName	卡号	注册时间
340321196507203290	河北大学附属医院	5340321196507203290	2020-04-09 17:51:04.663

ID	hospitalName	卡号	注册时间

数据库课设项目(上) 医院

https://blog.csdn.net/cascara/article/details/105404930

数据库课设项目(下) 医护人员与患者

https://blog.csdn.net/cascara/article/details/105418042

源代码

https://github.com/easilylazy/2020spring/tree/master/database/medicalSystem

以附件形式上传