COMP 214

SPRING FIELD

HOSPITAL DATABASE

PROFESSOR: LAXMI GOVINDA

COURSE: ADVANCE DATABASE CONCEPTS

SECTION 12

CENTENNIAL COLLEGE

The project that is currently being developed is a hospital database for medical professions. This database will have patients' profile ready at hand for medical professions and medical directors to view their patient general information to what ward, disease, treatment, and the recovery room where patient is located. This will allow all hospital staff to know the patient current health and determine the best course of action when they can see their patient profile that they can pull from the database.

Table of Contents

I. **Introduction** 1

II. **Noun Table** 3

III. **Business Information** 3

IV. **Business Roles** 4

V. **Detailed Problem Definition** 4

- Problem Definition
- Fact Finding
- Business Rules
- Database Users

VI. **Table Design** 8

- ERD

VII. **Table Layout** 9

VIII. **Table Database** 14

- Room Table
- Patient Table
- Hospital Table
- Disease Table
- Appointment Table
- Duty Table
- Speciality Table
- Admission Table
- Doctor Table
- Files Table
- Result Table
- Administration Table
- Admin Appointment Table
- Doctor Duty Table

IX. **Procedures** 22

- Procedure to get the history of patient
- Procedure to discharge a patient or show error if already discharged
- Procedure that shows analysis of diseases id entered

Procedure to show the patient name and bed number based on diseases id

X. **Triggers** **27**

Start date must be earlier than end date

Result must be Assigned to File

XI. **Views** **29**

Number of Doctors per Hospital

Total Patient Treated in one month

Number of Duty Performed by each employee in One Month

History for patients

Number of beds Available per each room.

Number patient per doctor

XII. **Function** **33**

Function Will return Number of free bed availabel in hospital

Function to see how many times a given patient was admitted

Team Members

Parth Chandgadhiya 300986134
Reneton Jayakumar 300939376
Agapito Feliciano 301033578
Kien Luong Ly 300925214

Noun Table

Noun	Description
Doctor	Someone who is medically trained and can operate on people medically
Director	A higher up who controls the flow of work activity of the doctors
Custodian	An employee who cleans up messes. Some clean up certain kind of messes that vary
Hospital	An institution that provides medical treatment
Disease	Sickness that may cause through biological or non-biological means
Health Card	A Photo I.D that grants public health in Canada
SIN	Social insurance number
Administrator	A person who controls and is responsible for running the business
Patient	An individual who may require or getting medical treatment
Room	Space where a patient can rest
Bed	An object where a patient can use to rest
Cure	Relieve of a disease or condition

Business Information

Our database system read, write, and store data from what doctors and medical personal put into the database. The system should be relevant, reliable, accurate, bug-free, and provide an outstanding user experience. The database is required to be user friendly that even someone who at least know how to input, and update data should be able to use the system without any frustration. All medical personal can update any patient profile

whenever they need too. System will constantly back up itself, so data will be lost or corrupted.

Business Roles

Appointment number

The appointment number is generated by the system. It's 6 digits and it's always unique with each appointment. Appointments can only be created if a person has a valid Ontario health card.

Administration

Administrators are people who verify that patients have a valid name, address, telephone, OHIP, and a SIN. Security on this level is a lot tighter and strict.

Patient

Someone receiving treatment from a doctor. A Patient needs a name, address, telephone, OHIP, SIN, and optional have benefits or health insurance

Medical Professions

The following medical professions are:

- Doctors
- Custodians
- Admins

Detailed Problem Definition

Problem Definition

Managing a hospital patient and retrieving their info can be quite difficult especially in an environment were every second count. It's time consuming if everything must be reviewed by looking through profiles rather than having a "straight to the point" system on a patient health.

Fact Finding

- New patients can be registered by the hospital registration system or having an auto fill registry
  if the patient comes in through an emergency vehicle.
- Doctors and patient should be able to create appointments through a doctor or visiting the hospital.
- Medical personal should be able to update their patient information. Patient can also request to have their information changed after proper verification.
- OHIP and SIN number always valid and stored in the database. The Ontario Healthcare can not run without these two vital features.
- The database is required to make backup of the current database at certain times. The back up data must be able to recover and restore easily.

Business Rules

General

1. A patient must have valid identification, health card (OHIP), and a SIN number.
2. Appointment can only generate if a patient has meet valid requirements to get one

3. All appointments must have a start date and an end date, and they end date can't be past the start date

4. All tables must be filled in any patient, doctor, hospital, room, and admin. The only exception is insurance since it's optional

5. All Tables except for appointment table and duty doctor table must have a unique ID

Database Users

In this system, they are only three database users:

- User
  - They have access to their patient profile
  - They can make update to their patient profile whenever they need

- Manager
  - Accessible to any information at any given time
  - Manages all patient information
  -

- Admin
  - Control over entire system
  - Manages the entire database
  - Access and change information with no restriction

Table Design

ERD

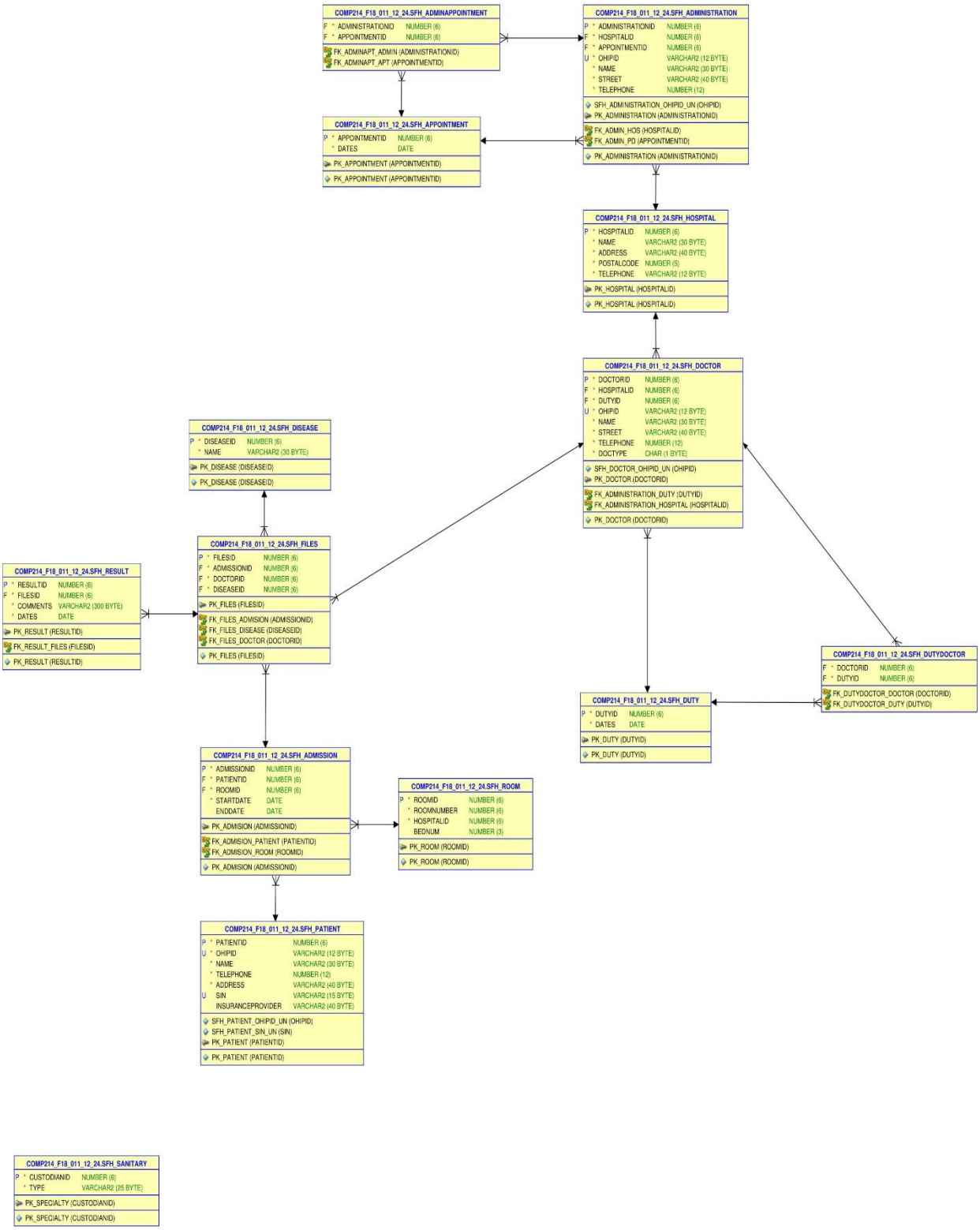


Table Layout

Room Table

Attribute	Description	Key Type	Data Type	Field Length
roomId	Unique Room Id to patient	Primary Key, Not Null	Number	6
roomNumber	Room Number related to the room Id	Not Null	Number	6
hospitalId	Hospital Identification number	Not Null	Number	6
BedNum	Bed Number		Number	3

Patient Table

Attribute	Description	Key Type	Data Type	Field Length
patientId	Unique Patient id number	Primary Key, Not Null,	Number	6
ohipID	Unique OHIP card number	Not Null, Unique	Varchar2	12
name	Patient Name	Not Null	Varchar2	30
telephone	Patient Telephone	Not Null	Number	12
address	Patient Address	Not Null	Varchar2	40
sin	Unique Social Insurance Number	Unique	Varchar2	15
insuranceProvider	Private Insurance or Benefits is eligible		Varchar2	40

Hospital Table

Attribute	Description	Key Type	Data Type	Field Length
hospitalId	Unique hospital Id	Primary Key, Not Null	Number	6
name	Hospital Name	Not Null	Varchar2	30
address	Hospital Address	Not Null	Varchar2	40
postalCode	Hospital Postal Code	Not Null	Number	5

telephone	Hospital Phone number	Not Null	Varchar2	12

Disease Table

Attribute	Description	Key Type	Data Type	Field Length
diseaseId	Unique Disease Id	Primary Key, Not Null	Number	6
name	Disease Name	Not Null	Varchar2	30

Appointment Table

Attribute	Description	Key Type	Data Type	Field Length
appointmentId	Unique Date Id	Primary Key, Not Null	Number	6
date	Appointment Date	Not Null	Date	8

Duty Table

Attribute	Description	Key Type	Data Type	Field Length
dutyId	Unique Duty Id	Primary Key, Not Null	Number	6
dates	Duty Dates	Not Null	Date	8

Sanitary Table

Attribute	Description	Key Type	Data Type	Field Length
dutyId	Unique Custodian Id	Primary Key, Not Null	Number	6
Type	Custodian Cleaning Type	Not Null	Varchar2	25

Admission Table

Attribute	Description	Key Type	Data Type	Field Length
admissionId	Unique admission Id	Primary Key, Not Null	Number	6
patientID	Patient Id	Not Null, Foreign Key	Number	6
roomID	Room Identification	Not Null, Foreign Key	Number	6
startDate	Starting day of admission	Not Null	Date	8
endDate	Final day of Admission		Date	8
Admission_Dates	Check admission dates	Check	Date	16

Doctor Table

Attribute	Description	Key Type	Data Type	Field Length
doctorID	Unique Doctor Id	Primary Key, Not Null	Number	6
hospitalID	Hospital Id	Not Null, Foreign Key	Number	6
dutyID	Duty Code	Not Null, Foreign Key	Number	6
ohipID	OHIP ID	Not Null, Unique	Varchar2	12
name	Doctor Name	Not Null	Varchar2	30
street	Doctor Address	Not Null	Varchar2	40
telephone	Doctor Telephone	Not Null	Number	12
doctype	Kind of Doctor	Not Null, Check	Char	1

Files Table

Attribute	Description	Key Type	Data Type	Field Length

filesID	Unique File Id	Primary Key, Not Null	Number	6
admissionID	Admission Id	Not Null, Foreign Key	Number	6
doctorID	Doctor Id	Not Null, Foreign Key	Number	6
diseaseID	Disease Id name	Not Null, Foreign Key	Number	6

Result Table

Attribute	Description	Key Type	Data Type	Field Length
filesID	Unique File Id	Primary Key, Not Null	Number	6
admissionID	Admission Id	Not Null, Foreign Key	Number	6
doctorID	Doctor Id	Not Null, Foreign Key	Number	6
diseaseID	Disease Id name	Not Null, Foreign Key	Number	6

Administration Table

Attribute	Description	Key Type	Data Type	Field Length
administrationID	Unique Administration Id	Primary Key, Not Null	Number	6
hospitalID	Hospital Id	Not Null, Foreign Key	Number	6
appointmentID	Appointment Id	Not Null, Foreign Key	Number	6
ohipID	OHIP ID	Not Null, Unique	Varchar2	12
name	Administrator Name	Not Null	Varchar2	30
street	Administrator Street	Not Null	Varchar2	40
telephone	Administrator Telephone	Not Null	Number	12

Appointment Table

Attribute	Description	Key Type	Data Type	Field Length
adminstrationID	Administration Id	Not Null, Foreign Key	Number	6
appointmentID	Appointment Id	Not Null, Foreign Key	Number	6

Duty Doctor Table

Attribute	Description	Key Type	Data Type	Field Length
doctorID	Doctor Id	Not Null, Foreign Key	Number	6
dutyID	Duty Id	Not Null, Foreign Key	Number	6

```
drop TABLE SFH_dutydoctor CASCADE CONSTRAINTS;
drop TABLE SFH_adminappointment CASCADE CONSTRAINTS;
drop TABLE SFH_administration CASCADE CONSTRAINTS;
drop TABLE SFH_result CASCADE CONSTRAINTS;
drop TABLE SFH_files CASCADE CONSTRAINTS;
drop TABLE SFH_doctor CASCADE CONSTRAINTS;
drop TABLE SFH_admission CASCADE CONSTRAINTS;
drop TABLE SFH_specialty CASCADE CONSTRAINTS;
drop TABLE SFH_duty CASCADE CONSTRAINTS;
drop TABLE SFH_appointment CASCADE CONSTRAINTS;
drop TABLE SFH_disease CASCADE CONSTRAINTS;
drop TABLE SFH_hospital CASCADE CONSTRAINTS;
drop TABLE SFH_patient CASCADE CONSTRAINTS;
drop TABLE SFH_room CASCADE CONSTRAINTS;
```

Room Table

```
CREATE TABLE SFH_ROOM(
      roomId       NUMBER(6)    NOT NULL CONSTRAINT PK_ROOM PRIMARY KEY,
      roomNumber   NUMBER(6)    NOT NULL,
      hospitalId   NUMBER(6)    NOT NULL,
      bedNum NUMBER(3)
);

INSERT INTO SFH_ROOM(ROOMID, ROOMNUMBER, HOSPITALID, BEDNUM) VALUES(1, 25, 1, 4);
INSERT INTO SFH_ROOM(ROOMID, ROOMNUMBER, HOSPITALID, BEDNUM) VALUES(2, 26, 1, 2);
INSERT INTO SFH_ROOM(ROOMID, ROOMNUMBER, HOSPITALID, BEDNUM) VALUES(3, 27, 1, 3);
INSERT INTO SFH_ROOM(ROOMID, ROOMNUMBER, HOSPITALID, BEDNUM) VALUES(4, 28, 1, 2);
INSERT INTO SFH_ROOM(ROOMID, ROOMNUMBER, HOSPITALID, BEDNUM) VALUES(5, 29, 1, 5);
INSERT INTO SFH_ROOM(ROOMID, ROOMNUMBER, HOSPITALID, BEDNUM) VALUES(6, 25, 2, 1);
INSERT INTO SFH_ROOM(ROOMID, ROOMNUMBER, HOSPITALID, BEDNUM) VALUES(7, 26, 2, 2);
INSERT INTO SFH_ROOM(ROOMID, ROOMNUMBER, HOSPITALID, BEDNUM) VALUES(8, 24, 3, 3);
```

Patient Table

```
CREATE TABLE SFH_PATIENT(
      patientID    NUMBER(6)    NOT NULL CONSTRAINT PK_PATIENT PRIMARY KEY,
      ohipID       VARCHAR2(12) NOT NULL UNIQUE,
      name         VARCHAR2(30) NOT NULL,
      telephone    NUMBER(12)   NOT NULL,
      address      VARCHAR2(40) NOT NULL,
      sin          VARCHAR2(15) UNIQUE,
      insuranceProvider   VARCHAR2(40)
);
```

```
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS, SIN,
INSURANCEPROVIDER) VALUES(1, '71295830A', 'JOHN WALL', 4161234567, '258 Victoria Park
Ave', 912345678, 'Manulife Financial');
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS, SIN) VALUES(2,
'71295832A', 'JACK MURPHY', 6471112222, '5 King Street West', 922734351);
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS) VALUES(3,
'71295833A', 'ALAN RUSSELS', 4165556677, '7895 Queen Street');
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS, SIN) VALUES(4,
'71295834A', 'JUSTIN TRUDEAU', 6478661234, '1150 Bellamy Road', 9447343554);
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS, SIN,
INSURANCEPROVIDER) VALUES(5, '71295835A', 'LONZO LOPEZ', 6474445568, '89 Lawrance
West', 955573435,'Sun Life Financial');
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS, SIN) VALUES(6,
'71295836A', 'BENJAMIN AFFLECT', 6475883322, '463 Progress Ave', 966673435);
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS) VALUES(7,
'71295837A', 'VAN HELSING', 6476883322, '89 Tania Cres');
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS, SIN,
INSURANCEPROVIDER) VALUES(8, '71295838A', 'KEVIN GARNET', 4165689784, '7795
Eglinton', 957773435,'Sun Life Financial');
INSERT INTO SFH_PATIENT(PATIENTID, OHIPID, NAME, TELEPHONE, ADDRESS, SIN) VALUES(9,
'71295839A', 'JESS ALBERT', 4161556532, '4621 Amberjack Blvd', 999734352);
```

Hospital Table

```
CREATE TABLE SFH_HOSPITAL(
      hospitalID   NUMBER(6)    NOT NULL CONSTRAINT PK_HOSPITAL PRIMARY KEY,
      name         VARCHAR2(30) NOT NULL,
      address      VARCHAR2(40) NOT NULL,
      postalCode   VARCHAR2(10) NOT NULL,
      telephone    VARCHAR2(12) NOT NULL
);

INSERT INTO SFH_HOSPITAL(HOSPITALID, NAME, ADDRESS, POSTALCODE, TELEPHONE) VALUES(1,
'North Spring Field Hospital', '4001 Leslie St', 'M2K 1E1', 416756123);
INSERT INTO SFH_HOSPITAL(HOSPITALID, NAME, ADDRESS, POSTALCODE, TELEPHONE) VALUES(2,
'South Spring Field Hospital', ' 2867 Ellesmere Rd', 'M1E 4B9', 4162848131);
INSERT INTO SFH_HOSPITAL(HOSPITALID, NAME, ADDRESS, POSTALCODE, TELEPHONE) VALUES(3,
'East Spring Field Hospital', '825 Coxwell Ave', 'M4C 3E7', 4164164116);
INSERT INTO SFH_HOSPITAL(HOSPITALID, NAME, ADDRESS, POSTALCODE, TELEPHONE) VALUES(4,
'West Spring Field Hospital', ' 381 Church St', 'L3P 7P3', 6474727111);
```

Disease Table

```
CREATE TABLE SFH_DISEASE(
      diseaseID    NUMBER(6)    NOT NULL CONSTRAINT PK_DISEASE PRIMARY KEY,
      name         VARCHAR2(30) NOT NULL
);

INSERT INTO SFH_DISEASE(DISEASEID, NAME) VALUES(1,'Ischemic heart');
INSERT INTO SFH_DISEASE(DISEASEID, NAME) VALUES(2,'Lower Respiratory Infections');
INSERT INTO SFH_DISEASE(DISEASEID, NAME) VALUES(3,'DIARRHEA');
INSERT INTO SFH_DISEASE(DISEASEID, NAME) VALUES(4,'Tuberculosis');
```

```
INSERT INTO SFH_DISEASE(DISEASEID, NAME) VALUES(5,'Diabetes');
INSERT INTO SFH_DISEASE(DISEASEID, NAME) VALUES(6,'Cirrhosis');
INSERT INTO SFH_DISEASE(DISEASEID, NAME) VALUES(7,'Dehydration due to diarrheal');
```

Appointment Table

```
CREATE TABLE SFH_APPOINTMENT(
      appointmentID      NUMBER(6)    NOT NULL CONSTRAINT PK_APPOINTMENT PRIMARY
KEY,
      dates              DATE         NOT NULL
);

INSERT INTO SFH_APPOINTMENT(APPOINTMENTID, DATES) VALUES(1, '15/Dec/2017');
INSERT INTO SFH_APPOINTMENT(APPOINTMENTID, DATES) VALUES(2, '16/Dec/2017');
INSERT INTO SFH_APPOINTMENT(APPOINTMENTID, DATES) VALUES(3, '17/Dec/2017');
INSERT INTO SFH_APPOINTMENT(APPOINTMENTID, DATES) VALUES(4, '18/Dec/2017');
INSERT INTO SFH_APPOINTMENT(APPOINTMENTID, DATES) VALUES(5, '19/Dec/2017');
INSERT INTO SFH_APPOINTMENT(APPOINTMENTID, DATES) VALUES(6, '20/Dec/2017');
INSERT INTO SFH_APPOINTMENT(APPOINTMENTID, DATES) VALUES(7, '21/Dec/2017');
INSERT INTO SFH_APPOINTMENT(APPOINTMENTID, DATES) VALUES(8, '22/Dec/2017');
```

Duty Table

```
CREATE TABLE SFH_DUTY(
      dutyID       NUMBER(6)    NOT NULL CONSTRAINT PK_DUTY PRIMARY KEY,
      dates              DATE         NOT NULL
);

INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(1, '15/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(2, '16/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(3, '17/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(4, '18/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(5, '19/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(6, '20/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(7, '15/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(8, '16/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(9, '17/Dec/2017');
INSERT INTO SFH_DUTY(DUTYID, DATES) VALUES(11, '19/Dec/2017');
```

Speciality Table

```
CREATE TABLE SFH_SPECIALTY(
      doctorID     NUMBER(6)    NOT NULL CONSTRAINT PK_SPECIALTY PRIMARY KEY,
      specialty    VARCHAR2(25) NOT NULL
);

INSERT INTO SFH_SPECIALTY(DOCTORID, SPECIALTY) VALUES(1, 'flu a');
INSERT INTO SFH_SPECIALTY(DOCTORID, SPECIALTY) VALUES(2, 'flu b');
INSERT INTO SFH_SPECIALTY(DOCTORID, SPECIALTY) VALUES(3, 'flu c');
```

```
INSERT INTO SFH_SPECIALTY(DOCTORID, SPECIALTY) VALUES(4, 'flu d');
```

Admission Table

```
CREATE TABLE SFH_ADMISSION(
      admissionID  NUMBER(6)    NOT NULL CONSTRAINT PK_ADMISION PRIMARY KEY,
      patientId    NUMBER(6)    NOT NULL,
      roomID       NUMBER(6)    NOT NULL,
      startDate    DATE         NOT NULL,
      endDate             DATE,
      CONSTRAINT   CK_ADMISION_DATES   CHECK (startDate<=endDate),
      constraint   FK_ADMISION_PATIENT FOREIGN KEY (patientId) REFERENCES
SFH_patient,
      constraint   FK_ADMISION_ROOM    FOREIGN KEY (roomId) REFERENCES SFH_room
);


INSERT INTO SFH_ADMISSION(ADMISSIONID, PATIENTID, ROOMID, STARTDATE, ENDDATE)
VALUES(1, 1, 1, '15/feb/2017', '26/feb/2017');
INSERT INTO SFH_ADMISSION(ADMISSIONID, PATIENTID, ROOMID, STARTDATE, ENDDATE)
VALUES(2, 2, 2, '20/feb/2017', '26/feb/2018');
INSERT INTO SFH_ADMISSION(ADMISSIONID, PATIENTID, ROOMID, STARTDATE, ENDDATE)
VALUES(3, 3, 4, '14/feb/2017','25/feb/2018');
INSERT INTO SFH_ADMISSION(ADMISSIONID, PATIENTID, ROOMID, STARTDATE, ENDDATE)
VALUES(4, 2, 6, '14/feb/2017','26/feb/2018');
INSERT INTO SFH_ADMISSION(ADMISSIONID, PATIENTID, ROOMID, STARTDATE, ENDDATE)
VALUES(5, 1, 1, '14/feb/2017','28/feb/2018');
INSERT INTO SFH_ADMISSION(ADMISSIONID, PATIENTID, ROOMID, STARTDATE, ENDDATE)
VALUES(6, 4, 7, '01/feb/2017', '26/feb/2018');
INSERT INTO SFH_ADMISSION(ADMISSIONID, PATIENTID, ROOMID, STARTDATE, ENDDATE)
VALUES(7, 5, 8, '04/feb/2017', '26/feb/2018');
INSERT INTO SFH_ADMISSION(ADMISSIONID, PATIENTID, ROOMID, STARTDATE, ENDDATE)
VALUES(8, 4, 7, '20/feb/2017','09/feb/2018');
```

Doctor Table

```
CREATE TABLE SFH_Doctor(
      doctorID           NUMBER(6)    NOT NULL CONSTRAINT PK_DOCTOR PRIMARY KEY,
      hospitalID         NUMBER(6)    NOT NULL,
      dutyID       NUMBER(6)    NOT NULL,
      ohipID             VARCHAR2(12) NOT NULL UNIQUE,
      name               VARCHAR2(30) NOT NULL,
      street             VARCHAR2(40) NOT NULL,
      telephone          NUMBER(12)   NOT NULL,
      doctortype   CHAR(1)            NOT NULL CHECK ( doctortype IN ('M', 'P')),
      constraint   FK_ADMINISTRATION_HOSPITAL FOREIGN KEY (hospitalID) REFERENCES
SFH_hospital,
      constraint   FK_ADMINISTRATION_DUTY    FOREIGN KEY (dutyID) REFERENCES
SFH_duty
);
```

```
INSERT INTO SFH_DOCTOR(DOCTORID, HOSPITALID, DUTYID, OHIPID, NAME, STREET, TELEPHONE,
DOCTORTYPE) VALUES(1,1,1,'71295566Z', 'Steven GRINCH', '1 Lawrence Ave', 6477888333,
'M');
INSERT INTO SFH_DOCTOR(DOCTORID, HOSPITALID, DUTYID, OHIPID, NAME, STREET, TELEPHONE,
DOCTORTYPE) VALUES(2,1,1,'72295566Z', 'Michael Love', '245 Spadina Ave', 6476888444,
'M');
INSERT INTO SFH_DOCTOR(DOCTORID, HOSPITALID, DUTYID, OHIPID, NAME, STREET, TELEPHONE,
DOCTORTYPE) VALUES(3,2,1,'73295566Z', 'Eileen Murdoc', '99 Dundas St', 6475888666,
'M');
INSERT INTO SFH_DOCTOR(DOCTORID, HOSPITALID, DUTYID, OHIPID, NAME, STREET, TELEPHONE,
DOCTORTYPE) VALUES(4,2,1,'74295566Z', 'Mary WILLIAMS', '465 Parktree Cres',
4168888555, 'M');
INSERT INTO SFH_DOCTOR(DOCTORID, HOSPITALID, DUTYID, OHIPID, NAME, STREET, TELEPHONE,
DOCTORTYPE) VALUES(5,2,1,'75295566Z', 'Miriam Lopez ', '55 SpringSide Rd',
4160888777, 'P');
INSERT INTO SFH_DOCTOR(DOCTORID, HOSPITALID, DUTYID, OHIPID, NAME, STREET, TELEPHONE,
DOCTORTYPE) VALUES(6,3,1,'76295566Z', 'Paul GEORGE', '456 Avenue', 6471888888, 'P');
INSERT INTO SFH_DOCTOR(DOCTORID, HOSPITALID, DUTYID, OHIPID, NAME, STREET, TELEPHONE,
DOCTORTYPE) VALUES(7,4,1,'77295566Z', 'James LOGANS', '7866 Lakeshore', 4163888999,
'P');
```

Files Table

```
CREATE TABLE SFH_FILES(
      filesID      NUMBER(6)    NOT NULL CONSTRAINT PK_FILES PRIMARY KEY,
      admissionID  NUMBER(6)    NOT NULL,
      doctorID     NUMBER(6)    NOT NULL,
      diseaseID    NUMBER(6)    NOT NULL,
      constraint   FK_FILES_ADMISION   FOREIGN KEY (admissionID) REFERENCES
SFH_admission,
      constraint   FK_FILES_DOCTOR           FOREIGN KEY (doctorID) REFERENCES
SFH_doctor,
      constraint   FK_FILES_DISEASE    FOREIGN KEY (diseaseID) REFERENCES
SFH_disease
);

INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(1,1,1,1);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(2,1,3,3);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(3,1,2,5);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(4,2,1,2);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(5,2,2,4);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(6,3,1,7);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(7,3,3,5);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(8,5,2,6);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(9,4,2,2);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(10,7,1,1);
INSERT INTO SFH_FILES(FILESID, ADMISSIONID, DOCTORID, DISEASEID) VALUES(11,8,3,6);
```

Result Table

```
CREATE TABLE SFH_RESULT(
	resultID	NUMBER(6)	NOT NULL CONSTRAINT PK_RESULT PRIMARY KEY,
	filesID			NUMBER(6)	NOT NULL,
	comments	VARCHAR2(300)NOT NULL,
	dates		DATE		NOT NULL,
	constraint	FK_RESULT_FILES		FOREIGN KEY (filesID) REFERENCES SFH_FILES
);

INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(1,1,'1-Having side
effects of pill', '13/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(2,1,'2-Pill abuse',
'01/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(3,2,'3-Blood test
needed', '23/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(4,2,'4-Getting
better', '01/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(5,3,'5-Patient in
recovering condition', '14/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(6,4,'6-Take flu
shot tomorrow morning', '07/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(7,5,'7-Need special
treatment', '09/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(8,6,'8-In
recovering process', '11/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(9,7,'9-Surgery
needed', '12/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(10,8,'10-Exchange
to special treatment dpt', '13/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(11,9,'11-In good
health condition', '05/feb/2017');
INSERT INTO SFH_RESULT(RESULTID, FILESID, COMMENTS, DATES) VALUES(12,10,'12-Re-
examination next month', '27/feb/2017');
```

Administration Table

```
CREATE TABLE SFH_ADMINISTRATION(
	administrationID		NUMBER(6)	NOT NULL CONSTRAINT PK_ADMINISTRATION
PRIMARY KEY,
	hospitalID		NUMBER(6)	NOT NULL,
	appointmentID NUMBER(6)	NOT NULL,
	ohipID			VARCHAR2(12) NOT NULL UNIQUE,
	name			VARCHAR2(30) NOT NULL,
	street			VARCHAR2(40) NOT NULL,
	telephone		NUMBER(12)	NOT NULL,
	constraint	FK_ADMIN_HOSPITAL	FOREIGN KEY (hospitalID) REFERENCES
SFH_HOSPITAL,
	constraint	FK_ADMIN_APPOINTMENT		FOREIGN KEY (appointmentID) REFERENCES
SFH_appointment
);

INSERT INTO SFH_ADMINISTRATION(ADMINISTRATIONID, HOSPITALID, APPOINTMENTID, OHIPID,
NAME, STREET, TELEPHONE) VALUES(1,1,1,'71295566Z', 'Julian Calton', '12 King East',
6477888999);
```

```
INSERT INTO SFH_ADMINISTRATION(ADMINISTRATIONID, HOSPITALID, APPOINTMENTID, OHIPID,
NAME, STREET, TELEPHONE) VALUES(2,1,1,'72295566Z', 'Sam Jobs', '456 Queen St',
4166888999);
INSERT INTO SFH_ADMINISTRATION(ADMINISTRATIONID, HOSPITALID, APPOINTMENTID, OHIPID,
NAME, STREET, TELEPHONE) VALUES(3,2,1,'73295566Z', 'Will Gates', '4612 McCowan Rd',
4165888999);
INSERT INTO SFH_ADMINISTRATION(ADMINISTRATIONID, HOSPITALID, APPOINTMENTID, OHIPID,
NAME, STREET, TELEPHONE) VALUES(4,2,1,'74295566Z', 'Ivan Zhang', '216 Ellesmere Rd',
4168888999);
INSERT INTO SFH_ADMINISTRATION(ADMINISTRATIONID, HOSPITALID, APPOINTMENTID, OHIPID,
NAME, STREET, TELEPHONE) VALUES(5,2,1,'75295566Z', 'Lorenzo Lopez ', '136 McNicoil',
6470888999);
INSERT INTO SFH_ADMINISTRATION(ADMINISTRATIONID, HOSPITALID, APPOINTMENTID, OHIPID,
NAME, STREET, TELEPHONE) VALUES(6,3,1,'76295566Z', 'Juan Gimenez', '846 Bathurt',
4161888999);
INSERT INTO SFH_ADMINISTRATION(ADMINISTRATIONID, HOSPITALID, APPOINTMENTID, OHIPID,
NAME, STREET, TELEPHONE) VALUES(7,4,1,'77295566Z', 'Philip Albertson', '656 Warden',
6473888999);
```

Admin Appointment Table

```
CREATE TABLE SFH_ADMINAPPOINTMENT(
      administrationID    NUMBER(6)    NOT NULL,
      appointmentID       NUMBER(6)    NOT NULL,
      constraint   FK_ADMINAPT_ADMIN   FOREIGN KEY (administrationID) REFERENCES
SFH_administration,
      constraint   FK_ADMINAPT_APT           FOREIGN KEY (appointmentID) REFERENCES
SFH_appointment
);

INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(1, 1);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(1, 2);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(1, 3);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(1, 4);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(2, 5);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(2, 6);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(3, 2);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(4, 8);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(5, 7);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(6, 7);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(6, 6);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(6, 5);
INSERT INTO SFH_ADMINAPPOINTMENT(ADMINISTRATIONID, APPOINTMENTID) VALUES(7, 6);
```

Doctor Duty Table

```
CREATE TABLE SFH_DUTYDOCTOR(
      doctorID     NUMBER(6)    NOT NULL,
      dutyID NUMBER(6)    NOT NULL,
```

```
      constraint   FK_DUTYDOCTOR_DOCTOR      FOREIGN KEY (doctorID) REFERENCES
SFH_doctor,
      constraint   FK_DUTYDOCTOR_DUTY  FOREIGN KEY (dutyID) REFERENCES SFH_duty
);

INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(1, 1);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(1, 2);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(1, 3);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(1, 4);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(2, 5);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(2, 6);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(2, 7);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(2, 8);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(3, 9);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(3, 9);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(3, 11);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(3, 1);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(4, 3);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(4, 5);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(4, 5);
INSERT INTO SFH_DUTYDOCTOR(DOCTORID, DUTYID) VALUES(4, 8);

commit;
```

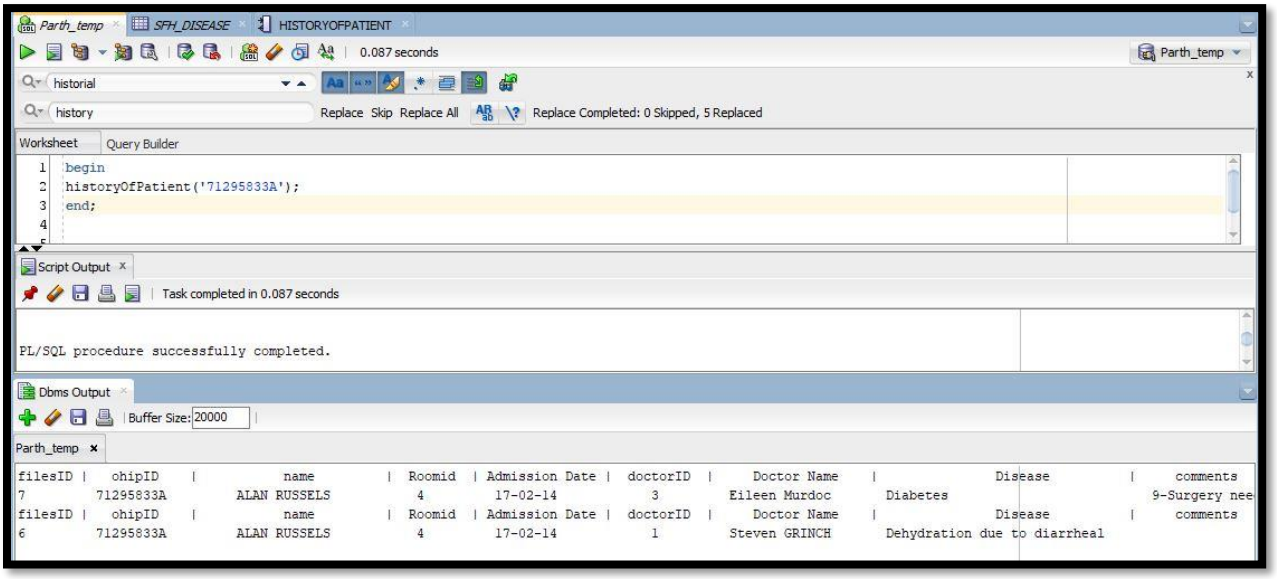
Queries

Procedures

Procedure to get the history of patient

```
CREATE OR REPLACE
PROCEDURE historyOfPatient(p_ohipID VARCHAR)
AS
	v_filesID SFH_files. filesID%TYPE;
	v_ohipID SFH_Patient.ohipID%TYPE;
	v_name SFH_Patient.name%TYPE;
	v_roomID SFH_admission.roomID%TYPE;
	v_startDate SFH_admission.startDate%TYPE;
	v_endDate SFH_admission.endDate%TYPE;
	v_doctorID SFH_files.doctorID%TYPE;
	v_docname SFH_doctor.name%TYPE;
	v_diname SFH_Disease.name%TYPE;
	v_comments SFH_result.comments%TYPE;
	v_dates SFH_result.dates%TYPE;

	cursor history is select * from patienthistory h where h.ohipID = p_ohipID
order by "Admission Date";
begin
	open history;
	loop
		fetch history into v_filesID, v_ohipID, v_name, v_roomID,v_startDate,
v_endDate, v_doctorID, v_docname, v_diname,v_comments, v_dates;
		exit when history%notfound;
		dbms_output.put_line('filesID |   ohipID    |           name         |
Roomid  | Admission Date |  doctorID  |     Doctor Name    |               Disease
|     comments                                |   Result Date');
		dbms_output.put_line(v_filesID ||'         ' || v_ohipID ||'         '||
v_name ||'           '|| v_roomID ||'         '|| v_startDate||'            ' ||
v_doctorID ||'         ' || v_docname ||'       '|| v_diname ||'
'|| v_comments ||'                   '|| v_dates);
	end loop;
	close history;
end historyOfPatient;


/*------FUNCTION query---------*/
```

```
begin
historyOfPatient('71295833A');
end;
```




```
/*-------------------------------------------------------------------------------------------------------------------------*/
```

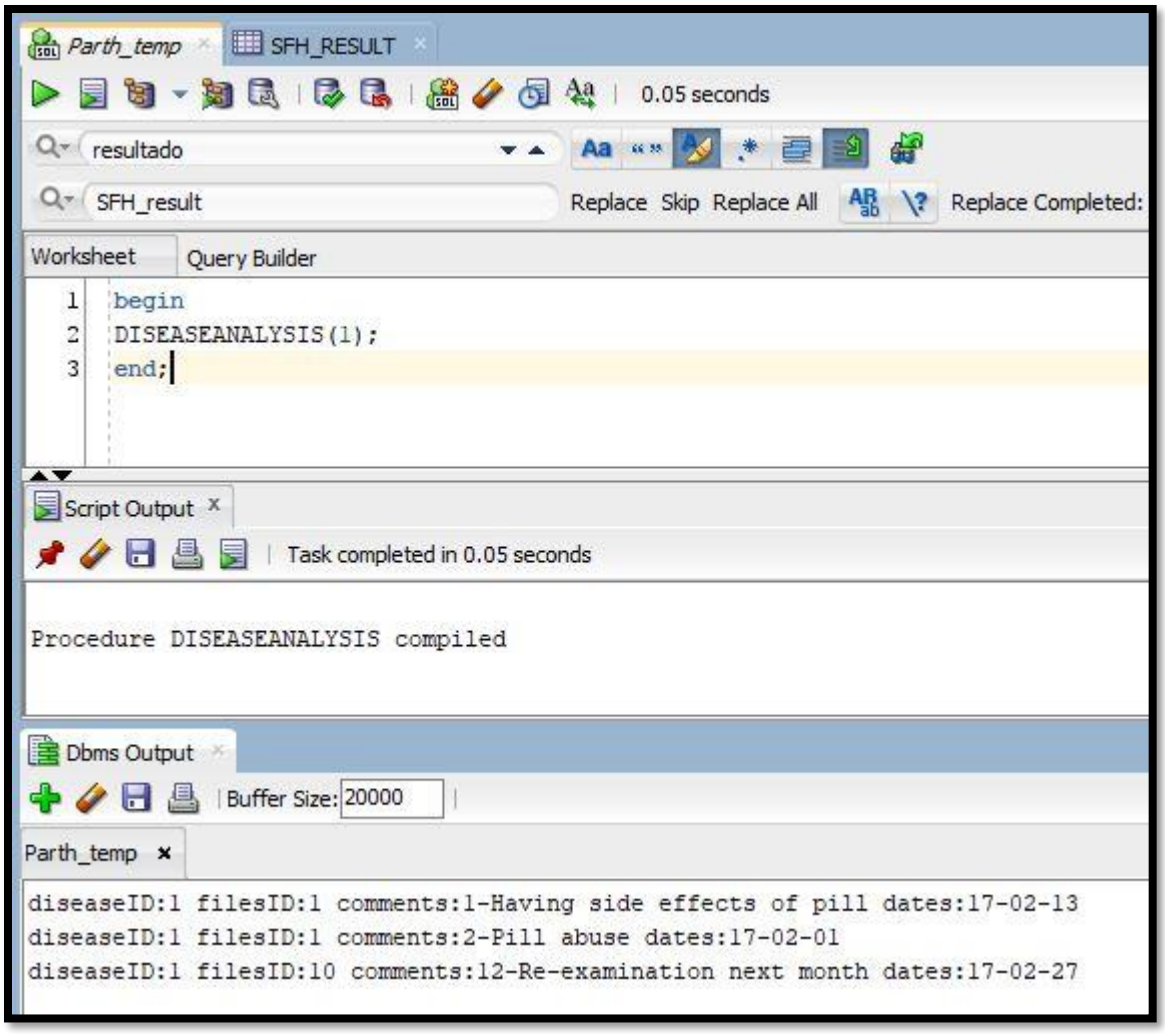
Procedure to discharge a patient or show error if already discharged

```
CREATE OR REPLACE
PROCEDURE dischargePatient(p_ohipID VARCHAR)
AS
	v_cont NUMBER(2);
	v_patientID SFH_Patient.patientID%TYPE;
	v_endDate SFH_admission.endDate%TYPE;
	v_discharged NUMBER(1);

begin
	select patientID into v_patientID from SFH_Patient where ohipID = p_ohipID;
	select count(*) into v_discharged from SFH_admission a, SFH_Patient p where
p.ohipID = p_ohipID and a.patientID = p.patientID and endDate is null;

	if (v_discharged>0) then
		update SFH_admission set endDate = CURRENT_DATE where patientID =
v_patientID;
			commit;
			dbms_output.put_line('Discharged');
	end if;
end dischargePatient;

/*------FUNCTION query---------*/

begin
DISCHARGEPATIENT('71295837A');
end;
```

```
/*-------------------------------------------------------------------------------------------------------------------------------------------*/
```

Procedure that shows analysis of diseases id entered

```
CREATE OR REPLACE
PROCEDURE diseaseAnalysis(p_diseaseID NUMBER)
AS
	v_cont NUMBER(2);
	v_filesID SFH_files.filesID%TYPE;
	v_comments VARCHAR2(300);
	v_dates DATE;

	cursor filesID is select filesID from SFH_files f where f.diseaseID =
p_diseaseID;
	cursor SFH_result (p_filesID in number) is
			select comments, dates from SFH_result where filesID = p_filesID;
begin

	open filesID;
	loop
		fetch filesID into v_filesID;
		exit when filesID%notfound;

			open SFH_result(v_filesID);
			loop
				fetch SFH_result into v_comments, v_dates;
				exit when SFH_result%notfound;
				dbms_output.put_line('diseaseID:'||p_diseaseID||'
filesID:'||v_filesID||' comments:'||v_comments||' dates:'||v_dates);
			end loop;
			close SFH_result;
	end loop;
	close filesID;
end diseaseAnalysis;

/*------FUNCTION query---------*/

begin
DISEASEANALYSIS(1);
end;
```

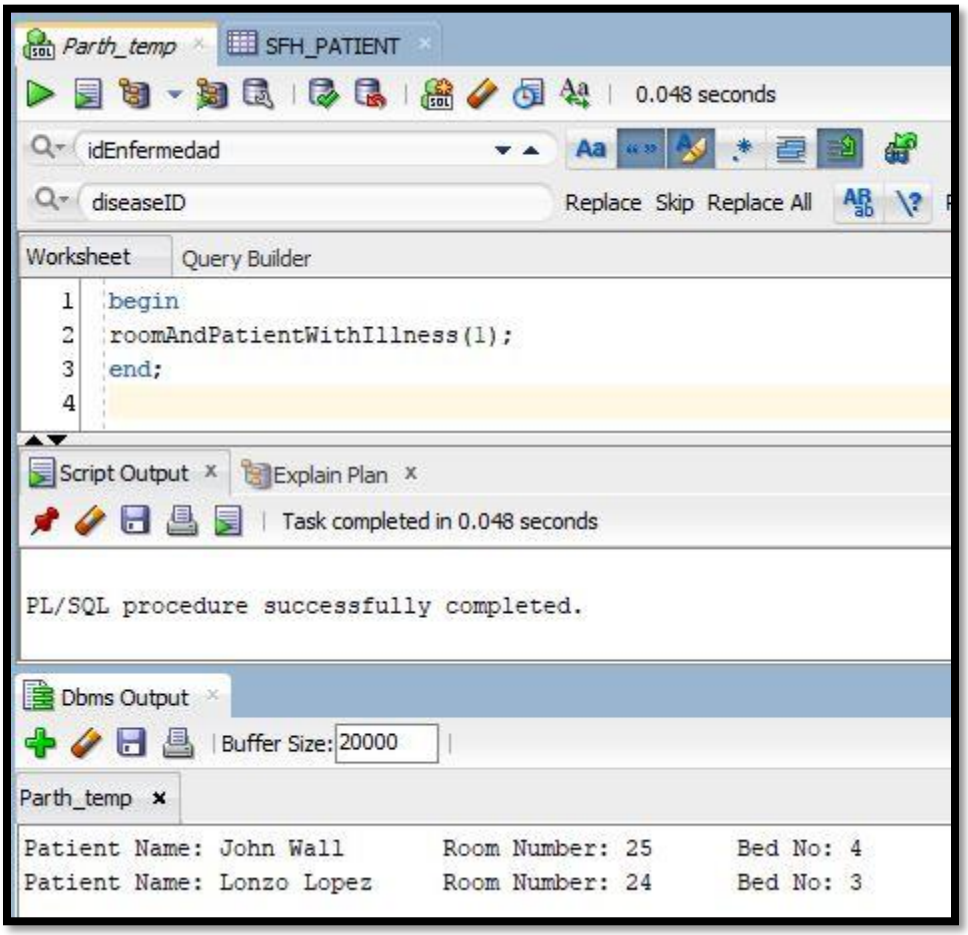


/*----------*/

Procedure to show the patient name and bed number based on diseases id

```
CREATE OR REPLACE
PROCEDURE roomAndPatientWithIllness(p_diseaseID NUMBER)
AS
	v_cont NUMBER(2);
	v_admissionID SFH_admission.admissionID%TYPE;

	v_roomNumber SFH_ROOM.roomNumber%TYPE;
	v_bedNum SFH_ROOM.bedNum%TYPE;
	v_name SFH_Patient.name%TYPE;

	cursor admissionID is select admissionID from SFH_files f where f.diseaseID =
p_diseaseID;
	cursor dataValue (p_admissionID in number) is
			select r.roomNumber, r.bedNum, p.name from SFH_Patient p,
SFH_admission a, SFH_ROOM r
			where a.admissionID = p_admissionID
```

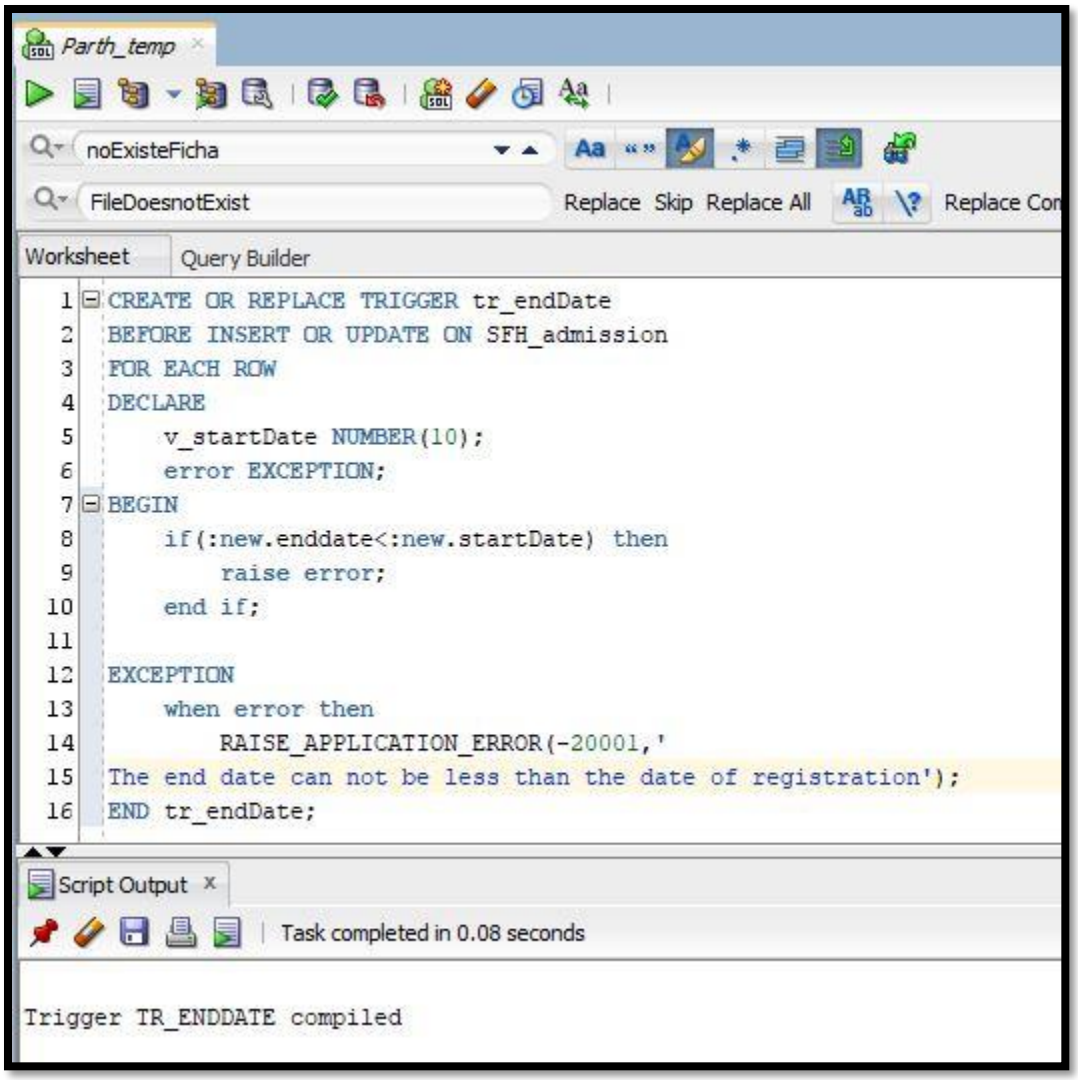
```
            and p.patientID = a.patientID
            and a.roomID = r.roomID
            and a.endDate is null;
begin
       open admissionID;
       loop
              fetch admissionID into v_admissionID;
              exit when admissionID%notfound;

                     open dataValue(v_admissionID);
                     loop
                            fetch dataValue into v_roomNumber, v_bedNum, v_name;
                            exit when dataValue%notfound;
                            dbms_output.put_line('Patient Name:'||v_name|| ' Room
Number: '||v_roomNumber||' Bed No: '||v_bedNum);
                     end loop;
                     close dataValue;
       end loop;
       close admissionID;
end roomAndPatientWithIllness;
```



/*---------- --------*/

Start date must be earlier than end date

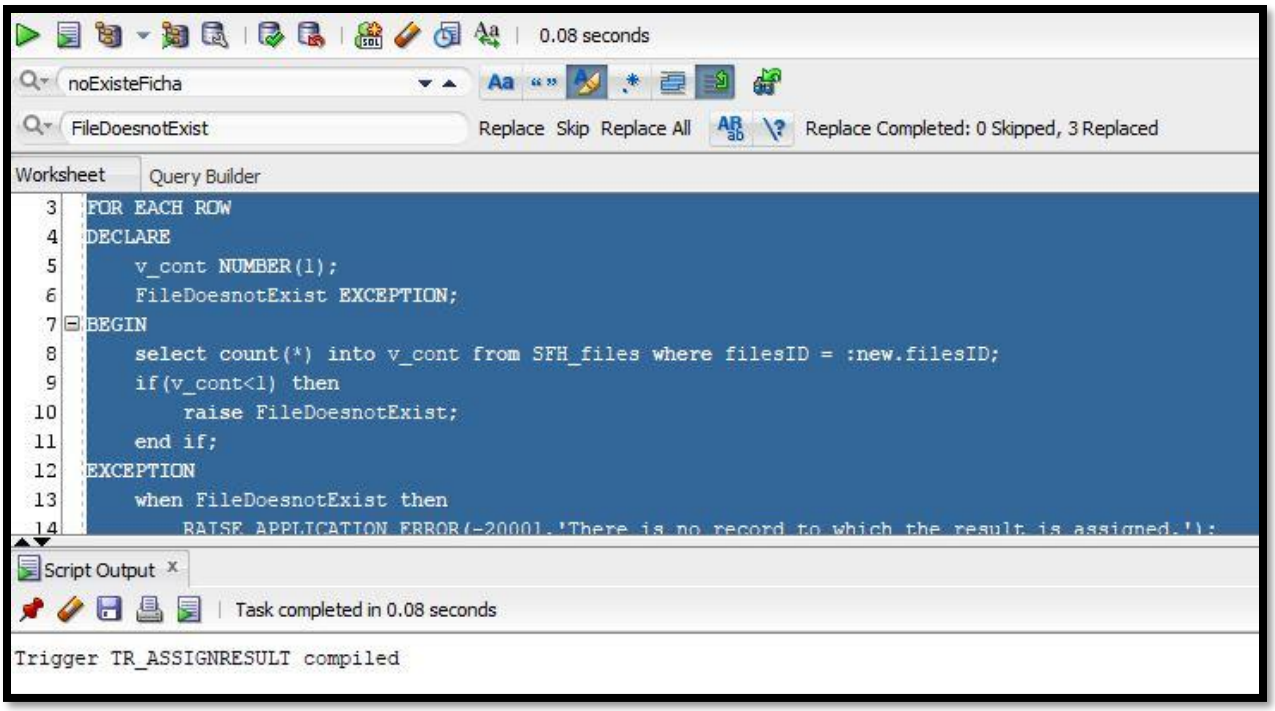
```
CREATE OR REPLACE TRIGGER tr_endDate
BEFORE INSERT OR UPDATE ON SFH_admission
FOR EACH ROW
DECLARE
        v_startDate NUMBER(10);
        error EXCEPTION;
BEGIN
        if(:new.enddate<:new.startDate) then
                raise error;
        end if;

EXCEPTION
        when error then
                RAISE_APPLICATION_ERROR(-20001,'
The end date can not be less than the date of registration');
END tr_endDate;
```



/*---------- --------*/

Result must be Assigned to File

```
CREATE OR REPLACE TRIGGER tr_AssignResult
BEFORE INSERT OR UPDATE ON SFH_result
FOR EACH ROW
DECLARE
        v_cont NUMBER(1);
        FileDoesnotExist EXCEPTION;
BEGIN
        select count(*) into v_cont from SFH_files where filesID = :new.filesID;
        if(v_cont<1) then
                raise FileDoesnotExist;
        end if;
EXCEPTION
        when FileDoesnotExist then
                RAISE_APPLICATION_ERROR(-20001,'There is no record to which the result is assigned.');
END tr_AssignResult;
```



Views

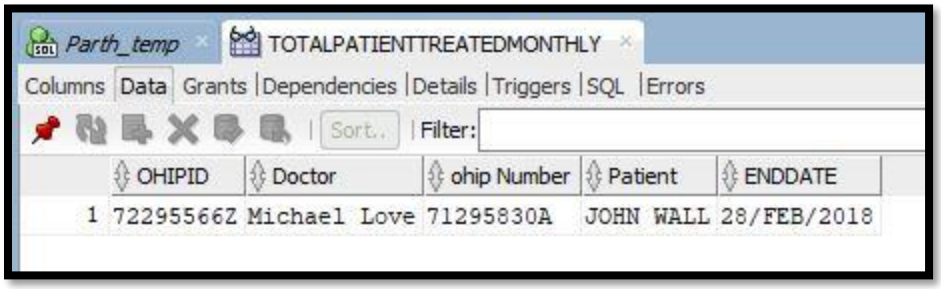
Number of Doctors per Hospital

```
CREATE VIEW DoctorsPerHospital AS
select h.hospitalID  as "hospital Id" , h.Name as "Hospital Name", count(*) as "Total
Number" from SFH_Hospital h, SFH_doctor s
where s.doctorType ='M' and s.hospitalID       = h.hospitalID
group by h.hospitalID     , h.Name
order by h.hospitalID;
```


```
/*-------------------------------------------------------------------------------------------------------------------------*/
```

Total Patient Treated in one month

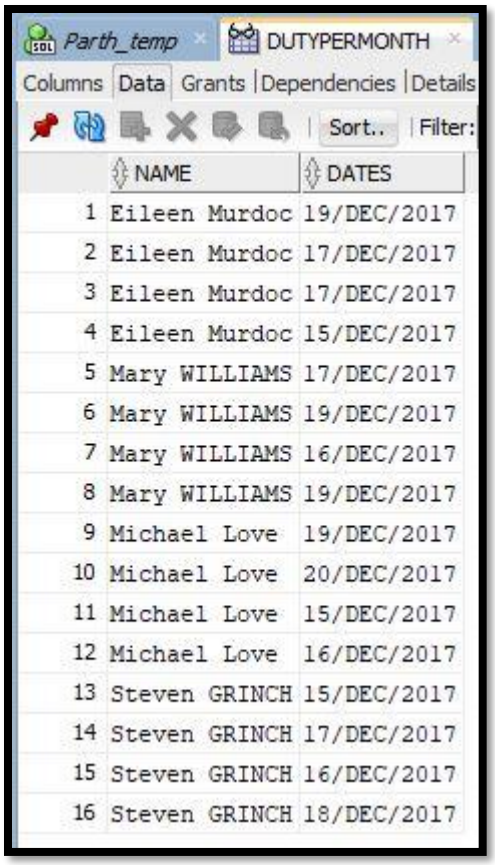
```
CREATE VIEW TotalPatientTreatedMonthly AS
SELECT SFH_doctor.ohipID, SFH_doctor.name as "Doctor", SFH_Patient.ohipID as "ohip
Number", SFH_Patient.name as "Patient", sfh_admission.enddate
FROM SFH_Patient,SFH_doctor, SFH_files, SFH_admission
WHERE SFH_doctor.doctortype = 'M'
and SFH_doctor.doctorID = SFH_files.doctorID
and SFH_admission.admissionID = SFH_files.admissionID
and SFH_Patient.patientID = SFH_admission.patientID
and SFH_admission.enddate = (select max(enddate) from SFH_admission)
order by SFH_doctor.doctorID desc;
```




```
/*-----------------------------------------------------------------------------------------------------------------------------------------*/
```

Number of Duty Performed by each employee in One Month

```
CREATE VIEW DutyPerMonth AS
select name, DATES from SFH_DOCTOR d,SFH_DUTYDOCTOR dd, SFH_DUTY du
where d.Doctortype = 'M'
and d.doctorID = dd.doctorID
and dd.dutyID = du.dutyID
order by name;
```



	NAME	DATES
1	Eileen Murdoc	19/DEC/2017
2	Eileen Murdoc	17/DEC/2017
3	Eileen Murdoc	17/DEC/2017
4	Eileen Murdoc	15/DEC/2017
5	Mary WILLIAMS	17/DEC/2017
6	Mary WILLIAMS	19/DEC/2017
7	Mary WILLIAMS	16/DEC/2017
8	Mary WILLIAMS	19/DEC/2017
9	Michael Love	19/DEC/2017
10	Michael Love	20/DEC/2017
11	Michael Love	15/DEC/2017
12	Michael Love	16/DEC/2017
13	Steven GRINCH	15/DEC/2017
14	Steven GRINCH	17/DEC/2017
15	Steven GRINCH	16/DEC/2017
16	Steven GRINCH	18/DEC/2017

```
/*--------------------------------------------------------------------------------------------------------------------------------*/
```

History for patients

```
CREATE VIEW PatientHistory AS
select f.filesID, p.ohipID, p.name, a.roomID as "Room", a.startDate as "Admission
Date", a.endDate as "End Date", f.doctorID as "Doctor Id", d.name as "Doctor Name",
di.name as "Disease", r.comments as "Remark", r.dates as "Result Date"
from SFH_Patient p, SFH_admission a, SFH_files f, SFH_result r,  SFH_doctor d,
SFH_Disease di
where p.patientID = a.patientID and
a.admissionID = f.admissionID and
r.filesID = f.filesID and
f.doctorID = d.doctorID and
f.diseaseID = di.diseaseID
order by p.name;
```

Parth_temp | PATIENTHISTORY

Columns | Data | Grants | Dependencies | Details | Triggers | SQL | Errors

Sort.. | Filter:

	FILESID	OHIPID	NAME	Room	Admission...	End Date	Doctor Id	Doctor Name	Disease	Remark	Result Date
1	7	71295833A	ALAN RUSSELS	4	14/FEB/2017	25/FEB/2018	3	Eileen Murdoc	Diabetes	9-Surgery needed	12/FEB/2017
2	6	71295833A	ALAN RUSSELS	4	14/FEB/2017	25/FEB/2018	1	Steven GRINCH	Dehydration due to diarrheal	8-In recovering process	11/FEB/2017
3	4	71295832A	JACK MURPHY	2	20/FEB/2017	26/FEB/2018	1	Steven GRINCH	Lower Respiratory Infections	6-Take flu shot tomorrow morning	07/FEB/2017
4	9	71295832A	JACK MURPHY	6	14/FEB/2017	26/FEB/2018	2	Michael Love	Lower Respiratory Infections	11-In good health condition	05/FEB/2017
5	5	71295832A	JACK MURPHY	2	20/FEB/2017	26/FEB/2018	2	Michael Love	Tuberculosis	7-Need special treatment	09/FEB/2017
6	8	71295830A	JOHN WALL	1	14/FEB/2017	28/FEB/2018	2	Michael Love	Cirrhosis	10-Exchange to special treatment dpt	13/FEB/2017
7	1	71295830A	JOHN WALL	1	15/FEB/2017	26/FEB/2017	1	Steven GRINCH	Ischemic heart	2-Pill abuse	01/FEB/2017
8	1	71295830A	JOHN WALL	1	15/FEB/2017	26/FEB/2017	1	Steven GRINCH	Ischemic heart	1-Having side effects of pill	13/FEB/2017
9	3	71295830A	JOHN WALL	1	15/FEB/2017	26/FEB/2017	2	Michael Love	Diabetes	5-Patient in recovering condition	14/FEB/2017
10	2	71295830A	JOHN WALL	1	15/FEB/2017	26/FEB/2017	3	Eileen Murdoc	DIARRHEA	4-Getting better	01/FEB/2017
11	2	71295830A	JOHN WALL	1	15/FEB/2017	26/FEB/2017	3	Eileen Murdoc	DIARRHEA	3-Blood test needed	23/FEB/2017
12	10	71295835A	LONZO LOPEZ	8	04/FEB/2017	26/FEB/2018	1	Steven GRINCH	Ischemic heart	12-Re-examination next month	27/FEB/2017

/*----------*/

Number of beds Available per each room.

```
CREATE VIEW AvailableAccommodation AS
select hospitalID, roomID, roomNumber, bedNum from SFH_ROOM
where roomID not in(select roomID from SFH_admission)
union
select hospitalID, SFH_ROOM .roomID, SFH_ROOM .roomNumber, SFH_ROOM .bedNum -
count(*) from SFH_admission, SFH_ROOM
where SFH_ROOM .roomID = SFH_admission.roomID
group by hospitalID, SFH_ROOM .roomID, SFH_ROOM .roomNumber, SFH_ROOM .bedNum;
```

Parth_temp | AVAILABLEACCOMMODATION

Columns | Data | Grants | Dependencies | Details | Triggers | SQL | Errors

Sort.. | Filter:

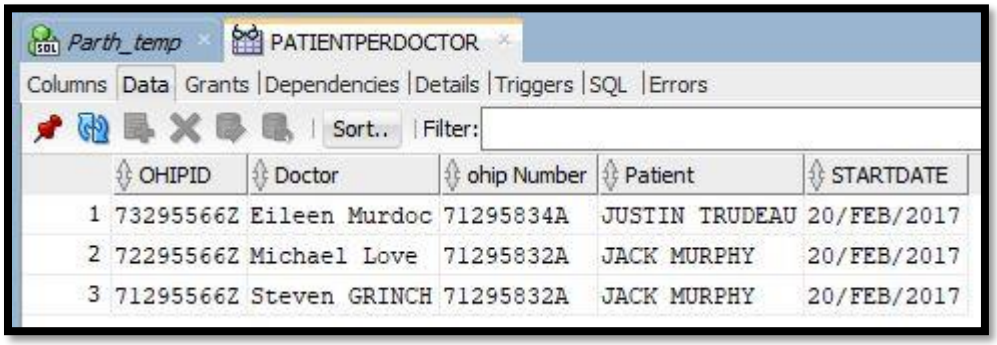
	HOSPITALID	ROOMID	ROOMNUMBER	BEDNUM
1	1	1	25	2
2	1	2	26	1
3	1	3	27	3
4	1	4	28	1
5	1	5	29	5
6	2	6	25	0
7	2	7	26	0
8	3	8	24	2

/*----------*/

Number patient per doctor

```
CREATE VIEW PatientPerDoctor AS
```

```
SELECT SFH_doctor.ohipID, SFH_doctor.name as "Doctor", SFH_Patient.ohipID as "ohip
Number", SFH_Patient.name as "Patient", sfh_admission.startdate
FROM SFH_Patient,SFH_doctor, SFH_files, SFH_admission
WHERE SFH_doctor.doctortype = 'M'
and SFH_doctor.doctorID = SFH_files.doctorID
and SFH_admission.admissionID = SFH_files.admissionID
and SFH_Patient.patientID = SFH_admission.patientID
and SFH_admission.startdate = (select max(startdate) from SFH_admission)
order by SFH_doctor.doctorID desc;
```

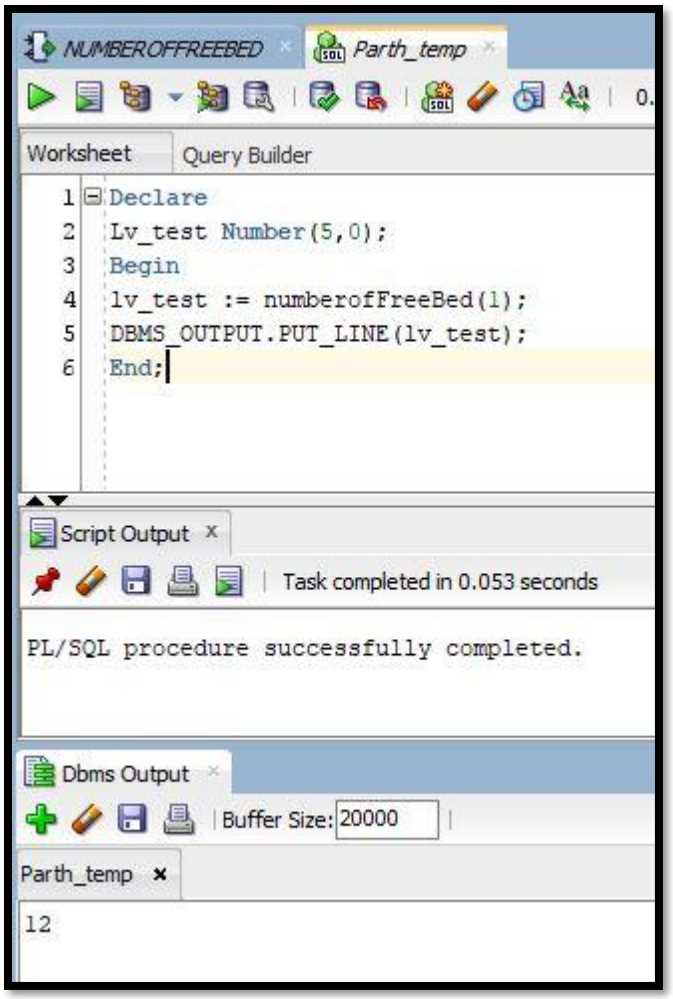


	OHIPID	Doctor	ohip Number	Patient	STARTDATE
1	73295566Z	Eileen Murdoc	71295834A	JUSTIN TRUDEAU	20/FEB/2017
2	72295566Z	Michael Love	71295832A	JACK MURPHY	20/FEB/2017
3	71295566Z	Steven GRINCH	71295832A	JACK MURPHY	20/FEB/2017

Function

Function Will return Number of free bed availabel in hospital

```
create or replace function numberOfFreeBed (v_hospitalID NUMBER)
return NUMBER
AS
      v_cont NUMBER(38);
BEGIN
            select sum(bedNum) into v_cont from AVAILABLEACCOMMODATION
       where hospitalID = v_hospitalID
       order by hospitalID;
      return v_cont;
exception
      WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Error:' ||SQLCODE||SQLERRM);
end numberOfFreeBed;


/*------FUNCTION query---------*/

Declare
Lv_test Number(5,0);
Begin
lv_test := numberofFreeBed(1);
DBMS_OUTPUT.PUT_LINE(lv_test);
End;
```



/*----------*/

Function to see how many times a given patient was admitted

```
CREATE OR REPLACE
function numberofTimeAdmitted (p_hospitalId NUMBER, p_patientID NUMBER)
return NUMBER
AS
       v_cont NUMBER(38);
BEGIN
             select count(*) into v_cont from SFH_admission, SFH_ROOM
        where SFH_ROOM.hospitalId = p_hospitalId
        and SFH_admission.patientID = p_patientID
        and SFH_ROOM.roomID = SFH_admission.roomID;
       return v_cont;

exception
       WHEN OTHERS THEN
             DBMS_OUTPUT.PUT_LINE('Error:' ||SQLCODE||SQLERRM);

end numberofTimeAdmitted;
```

```
/*------FUNCTION query---------*/

Declare
Lv_test Number(5,0);
Begin
lv_test := NUMBEROFTIMEADMITTED(1,1);
DBMS_OUTPUT.PUT_LINE(lv_test);
End;
```