

Phase III. Implementation

Vinit PatelMadhav Patel

vnp130130@utdallas.edu mmp130630@utdallas.edu

(Week 11-16: OCT.29-DEC.02)

0. Pre-Illumination

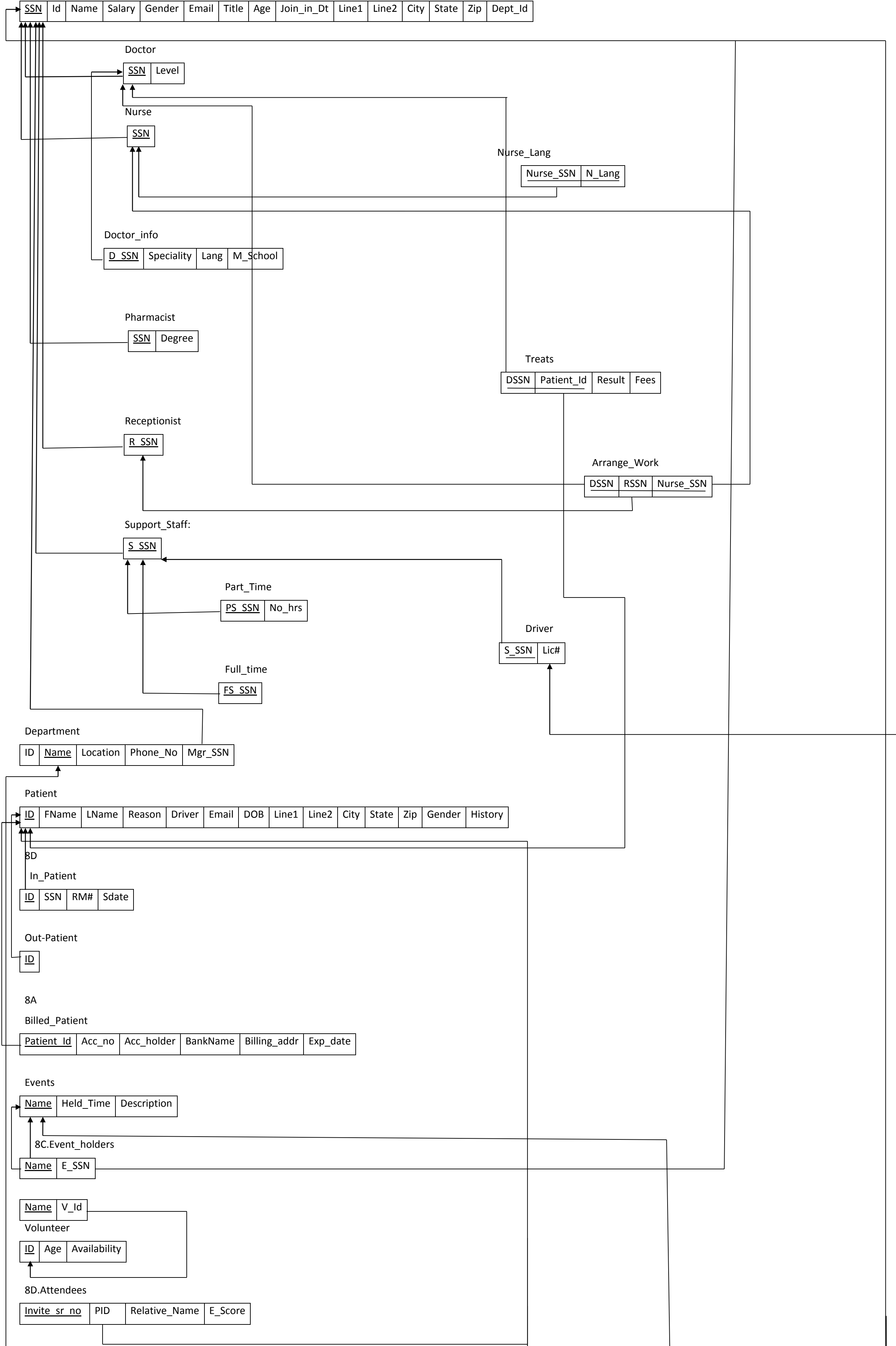
For clearly describing the implementation of our database, we separate this report into four sections. In Section 1 we normalize the original relational schema into third normal form and changed part of our relational schema because of some requirement from Phase III. We then explained what are changed. In Section 2 we drew a dependency diagram for each relation table one by one. In Section 3 we began our process of building a database in Oracle using SQL statements, which contains three parts. Part one is the creation of database, including tables, all other structures as well as data type and format, Part two is the creation of views corresponding to five distinct requirements from Question d, and Part three is the creation of Queries to satisfy 14 requirements from Question e. Finally, a short summary is given at the end of this report.

1. Modified Relational Schema

Firstly, according to the requirement of phase III and with purpose to simplify the relation model for this database, we changed three things respect to original relational models. We will list them as follows

(See next page for modified Relational Schema)

8A.Employee:



Appointment

<u>Date</u>	Time	PID
-------------	------	-----

Medicines

<u>ID</u>	Name	Price	Stored_in_dt	Type	Qty
-----------	------	-------	--------------	------	-----

Insurance

<u>Cmpny Name</u>	Policy#	PID	Phone	Exp_dt
-------------------	---------	-----	-------	--------

Ambulance

<u>Lic#</u>	Stored_dt	Location
-------------	-----------	----------

8B

Call

<u>Date</u>	Time	PID	Treatment_type	Doc_name	Receptionist_info
-------------	------	-----	----------------	----------	-------------------

Online

<u>Date</u>	Time	PID	Treatment_type	Doc_Name
-------------	------	-----	----------------	----------

Relatives

<u>P_ID</u>	Rel_Name
-------------	----------

Bill

<u>B_No</u>	P_ID	Status	Bed	Prescription	Ambulance	Pharmacy
-------------	------	--------	-----	--------------	-----------	----------

Event_City

<u>Event_Name</u>	E_City
-------------------	--------

E_DepName

<u>ESSN</u>	Dept_Name
-------------	-----------

PATIENT_PH_NO

<u>P_ID</u>	PH_NO
-------------	-------

Works_at

<u>D_LIC#</u>	<u>A_LIC#</u>
---------------	---------------

The modified relational schema is shown in Figure 1.

2. Dependency Diagram

We now draw a dependency diagram for each table from Figure 1 as follows:

2.1 Employee

There is only one attribute on left hand side which is SSN and that is the primary key and all other attributes are deoendent on SSN.

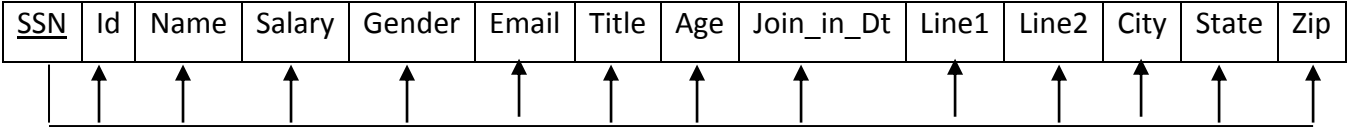
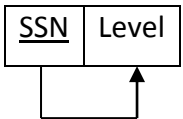


Figure 3. Dependency Diagram of Hospital Personnel

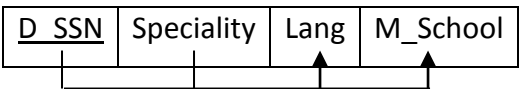
2.2 Doctor

Here level is dependent on SSN.



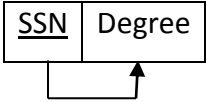
2.3 Doctor_info

Here D_SSN and Speciality uniquely identifies Lang and M_School.



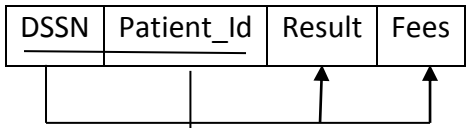
2.4 Pharmacist

Here Degree is dependent on pharmacist SSN



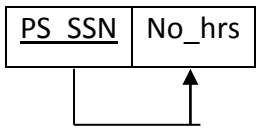
2.5 Treats

Here DSSN and Patient_id uniquely identifies which doctor treats which patient and what is the patient charged.



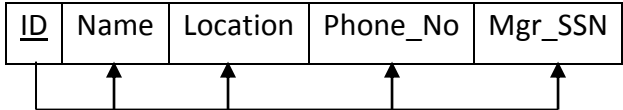
2.6 Part_Time

Here no_hrs is dependent on PS_SSN



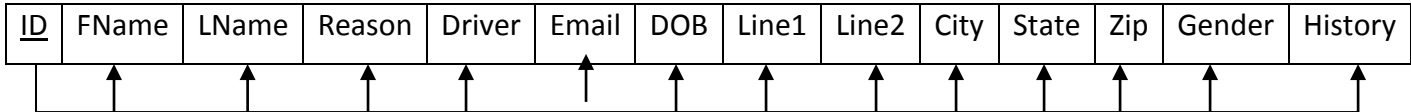
2.7 Department

Here ID uniquely identifies all the attributes



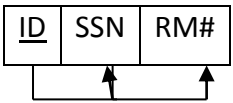
2.8 Patient

Here ID of patient is primary key which uniquely identifies all the other attributes which are dependent on ID.



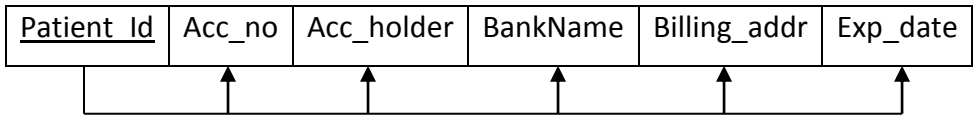
2.9 In_Patient

Here ID determines SSN of in-patient.



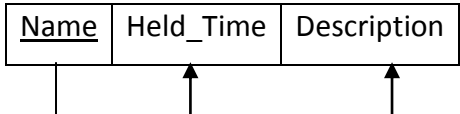
2.10 Billed_Patient

Here Patient_id uniquely identifies the all details of billed patient.



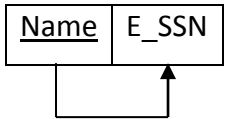
2.11 Events

Here name of event is primary key which uniquely determines Held_time and description of event.



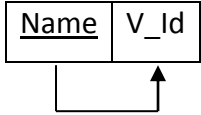
2.12 Event_Hold_Emp

Here name of Event determines which employee holds it.



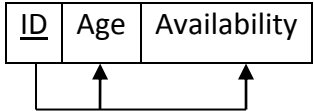
2.13 Event_Hold_Vol

Here name of volunteer determines which volunteer holds the event.



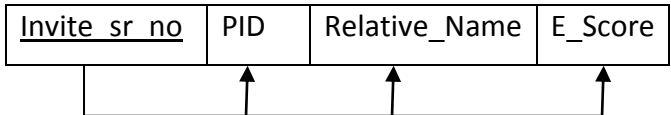
2.14 Volunteer

The ID o volunteer is primary key on which other attributes depend.



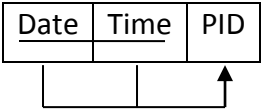
2.15 Attendees

Here Invite_sr_no uniquely determines all the information of invitees.



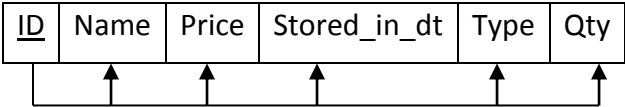
2.16 Appointment

Here Date and time are composite primary keys which identifies P_id to know who has made a appointment.



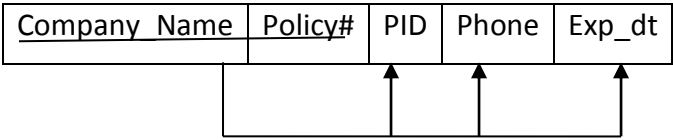
2.17 Medicines

Here medicines’ id determines all info about medicine.



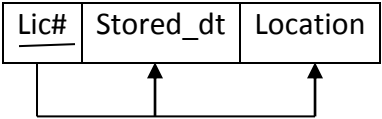
2.18 Insurance

Here Company name and policy# are composute primary key which uniquely identifies all attributes of patient to which policy belongs.



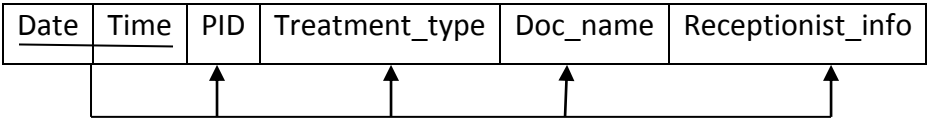
2.19 Ambulance

Here lic# determines the lic# of driver driving the ambulance.



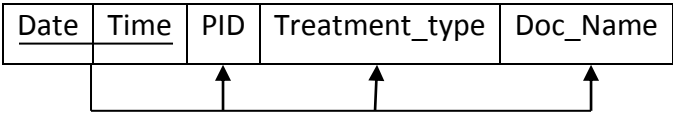
2.20 Call

Here Date and Time uniquely identifies the patient attributes who called to make an appointment.



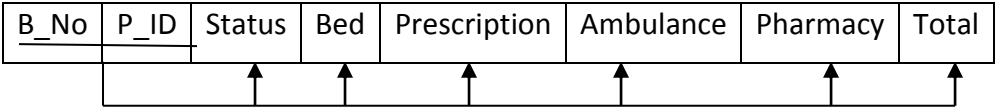
2.21 Online

Here Date time identifies patient who made online appointment.



2.22 Bill

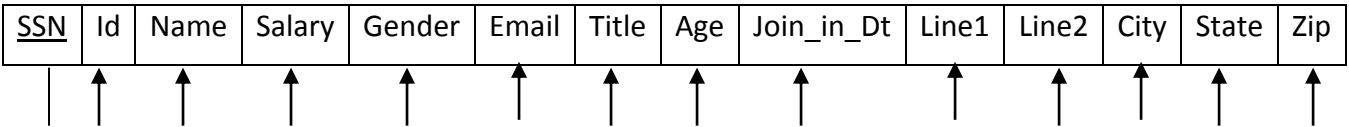
Here B_no(Bill number) and P_id determines the bill attributes which are inturn dependent on the key.



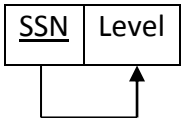
2.1 Final Results

After drawing the dependency diagrams one after another, Figure 2.1 shows the final results for the whole database including the ones who do not have any functional dependencies.

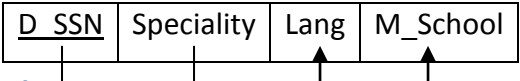
2.1 Employee



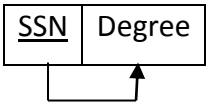
2.2 Doctor



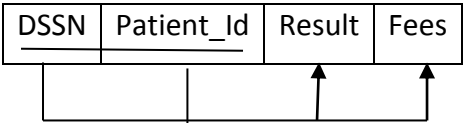
2.3 Doctor_info



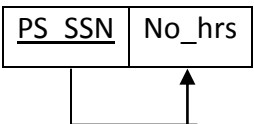
2.4 Pharmacist



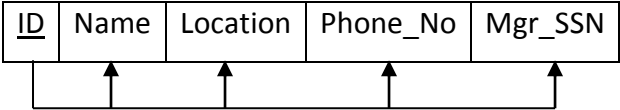
2.5 Treats



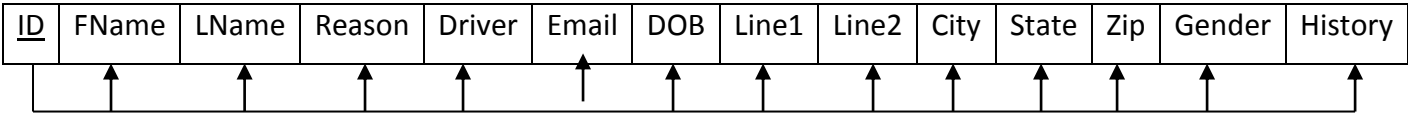
2.6 Part_Time



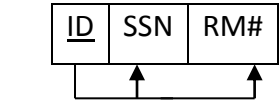
2.7 Department



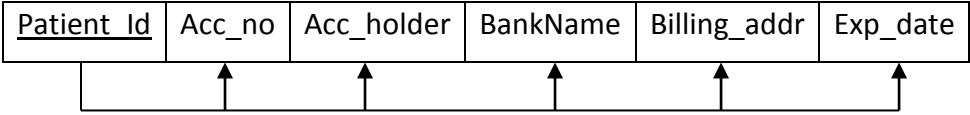
2.8 Patient



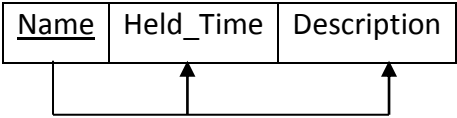
2.9 In_Patient



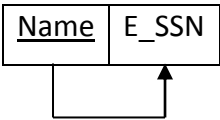
2.10 Billed_Patient



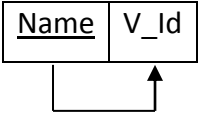
2.11 Events



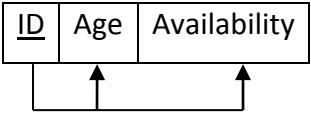
2.12 Event_Hold_Emp



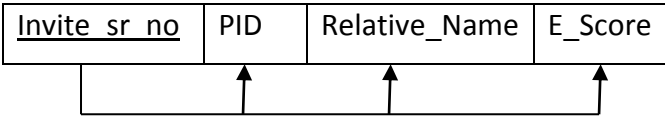
2.13 Event_Hold_Vol



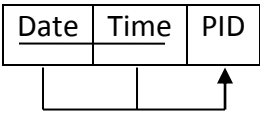
2.14 Volunteer



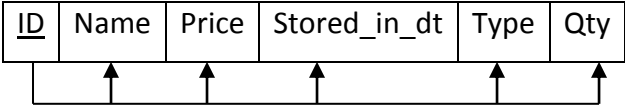
2.15 Attendees



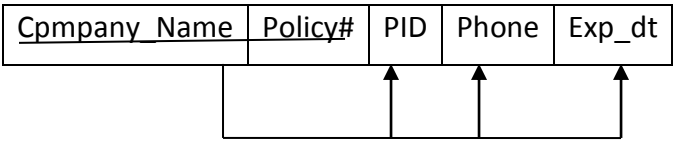
2.16 Appointment



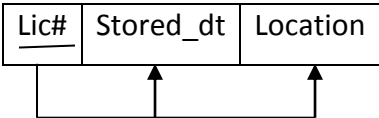
2.17 Medicines



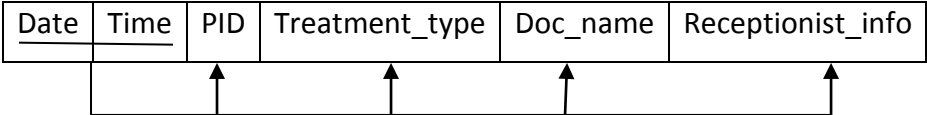
2.18 Insurance



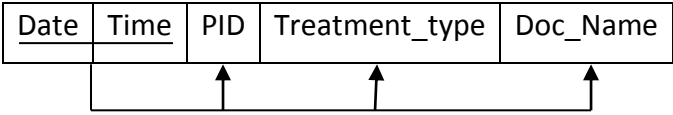
2.19 Ambulance



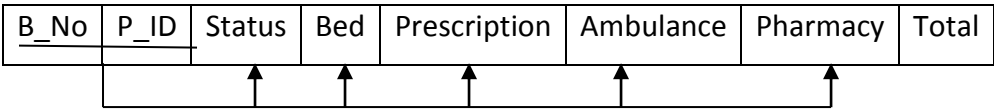
2.20 Call



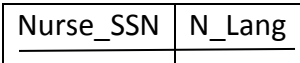
2.21 Online



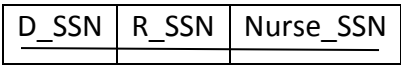
2.22 Bill



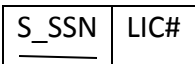
2.23Nurse_lang:



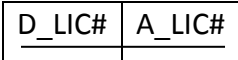
2.24Arrange_work



2.25Driver



2.26Works_at



2.27Relatives

<u>P_ID</u>	Rel_Name
-------------	----------

2.28Event_City

<u>Event_Name</u>	E_City
-------------------	--------

2.29E_DepName

<u>ESSN</u>	Dept_Name
-------------	-----------

2.30Receptionist

<u>R_SSN</u>

2.31Nurse

<u>SSN</u>

2.32Out-Patient

<u>ID</u>

2.33Full_time

<u>FS_SSN</u>

2.34Support_staff

<u>S_SSN</u>

2.35 PATIENT_PH_NO

<u>P_id</u>	PH_NO
-------------	-------

Figure 2.1. Whole Dependency Diagram for Hospital Database

3. Implementation of Database

3.1 Creation of Database with SQL Statements

After normalizing every relational schema into third normal form and modifying some details, it is the time to implement our database using SQL languages into Oracle.

3.1.1 Table Creation

Using SQL statement, we created 35 tables as follows:

- EMPLOYEE

```
CREATE TABLE EMPLOYEE
(SSN          VARCHAR(10)          NOT NULL,
ID            VARCHAR(11)          NOT NULL,
NAME          VARCHAR(15)          NOT NULL,
SALARY        VARCHAR(8)           ,
GENDER VARCHAR(8),
EMAIL         VARCHAR(20),
TITLE         VARCHAR(3),
JOIN_IN_DATE  DATE                ,
LINE          VARCHAR(20),
LINE2         VARCHAR(20),
CITY          VARCHAR(20),
STATE         VARCHAR(20),
ZIP           VARCHAR(10),
DEPT_ID       VARCHAR (11),
PRIMARY KEY (SSN));
```

- DOCTOR

```
CREATE TABLE DOCTOR
(SSN          VARCHAR(10)          NOT NULL,
```

```
D_L          INT,  
  
PRIMARY KEY(SSN),  
  
FOREIGN KEY(SSN) REFERENCES EMPLOYEE(SSN)  
  
CHECK(D_L>0 AND D_L<5));
```

- NURSE_LANG

CREATE TABLE NURSE_LANG

(N_SSN VARCHAR(10) NOT NULL,

N_LANG VARCHAR(20),

PRIMARY KEY(N_SSN),

FOREIGN KEY(N_SSN) REFERENCES NURSE(SSN));

- NURSE

CREATE TABLE NURSE

(SSN VARCHAR(10) NOT NULL,

PRIMARY KEY(SSN),

FOREIGN KEY(SSN) REFERENCES EMPLOYEE(SSN));

- DOCTOR_INFO

CREATE TABLE DOCTOR_INFO

(D_SSN VARCHAR(10) NOT NULL,

SPECIALITY VARCHAR(10) NOT NULL,

LANG VARCHAR(10) NOT NULL,

M_SCHOOL VARCHAR(15) NOT NULL,

PRIMARY KEY(D_SSN),

FOREIGN KEY(D_SSN) REFERENCES DOCTOR(SSN));

- PHARMACIST

CREATE TABLE PHARMACIST

(P_SSN VARCHAR(10) NOT NULL,

DEGREE VARCHAR(10) NOT NULL,

PRIMARY KEY(P_SSN),

FOREIGN KEY(P_SSN) REFERENCES EMPLOYEE(SSN));

- TREATS

CREATE TABLE TREATS

(D_SSN VARCHAR(10) NOT NULL,

P_ID VARCHAR(11) NOT NULL,

RESULT VARCHAR(10),

FEES VARCHAR(10),

PRIMARY KEY(D_SSN),

FOREIGN KEY(D_SSN) REFERENCES DOCTOR(SSN),

FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));

- RECEPTIONIST

CREATE TABLE RECEPTIONIST

(R_SSN VARCHAR(10) NOT NULL,

PRIMARY KEY(R_SSN),

FOREIGN KEY(R_SSN) REFERENCES EMPLOYEE(SSN));

- ARRANGE_WORK

CREATE TABLE ARRANGE_WORK

(D_SSN VARCHAR(10) NOT NULL,

R_SSN VARCHAR(10) NOT NULL,

N_SSN VARCHAR(10) NOT NULL,

PRIMARY KEY (D_SSN,R_SSN,N_SSN),

FOREIGN KEY(D_SSN) REFERENCES DOCTOR(SSN),

FOREIGN KEY(R_SSN) REFERNCES RECEPTIONIST(R_SSN),

FOREIGN KEY(N_SSN) REFERENCES NURSE(SSN));

- SUPPORT_STAFF

```
CREATE TABLE SUPPORT_STAFF
(S_SSN          VARCHAR(10)          NOT NULL,
PRIMARY KEY(S_SSN),
FOREIGN KEY(S_SSN) REFERENCES EMPLOYEE(SSN));
```

- PART_TIME

```
CREATE TABLE PART_TIME
(PS_SSN         VARCHAR(10)          NOT NULL,
NO_HOURS        VARCHAR(5),
PRIMARY KEY(PS_SSN),
FOREIGN KEY(PS_SSN) REFERENCES SUPPORT_STAFF(S_SSN));
```

- FULL_TIME

```
CREATE TABLE FULL_TIME
(FS_SSN         VARCHAR(10)          NOT NULL,
PRIMARY KEY(FS_SSN),
FOREIGN KEY(FS_SSN) REFERENCES SUPPORT_STAFF(S_SSN));
```

- DRIVER

```
CREATE TABLE DRIVER
(S_SSN          VARCHAR(10)          NOT NULL,
LIC#            VARCHAR(10)          NOT NULL,
PRIMARY KEY(S_SSN),
FOREIGN KEY(S_SSN) REFERENCES SUPPORT_STAFF(S_SSN),
FOREIGN KEY(LIC#) REFERENCES WORKS_AT(D_LIC#));
```

- DEPT

```
CREATE TABLE DEPT
(ID             VARCHAR(10)          NOT NULL,
NAME           VARCHAR(15)          NOT NULL,
LOC            VARCHAR(20)          NOT NULL,
PHONE_NO       VARCHAR(10),
MGR_SSN        VARCHAR(10)          NOT NULL,
PRIMARY KEY(ID),
FOREIGN KEY(MGR_SSN) REFERENCES EMPLOYEE(SSN));
```

- IN_PATIENT

```
CREATE TABLE IN_PATIENT
(P_ID          VARCHAR(10)          NOT NULL,
SSN            VARCHAR(10)          NOT NULL,
Rm#           VARCHAR(5),
PRIMARY KEY(P_ID),
FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));
```

- PATIENT

```
CREATE TABLE PATIENT
(P_ID          VARCHAR(11)          NOT NULL,
FNAME          VARCHAR(10)          NOT NULL,
LNAME          VARCHAR(10),
REASON         VARCHAR(25),
DRIVER         VARCHAR(10),
EMAIL_ID       VARCHAR(20),
DOB            DATE,
LINE           VARCHAR(20),
LINE2          VARCHAR(20),
```

```
CITY          VARCHAR(15),

STATE        VARCHAR(15),

GENDER       VARCHAR(5),

HISTORY      VARCHAR(25),

PRIMARY KEY(P_ID));
```

• OUT_PATIENT

```
CREATE TABLE OUT_PATIENT

(P_ID          VARCHAR(10)          NOT NULL,

PRIMARY KEY(P_ID),

FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));
```

BILLED_PATIENT

```
CREATE TABLE BILLED_PATIENT

(P_ID          VARCHAR(11)          NOT NULL,

ACC_NO        VARCHAR(10),

ACC HOLDER    VARCHAR(15),

BANK_NAME     VARCHAR(20),

BILLINH_ADDR  VARCHAR(25),

EXP_DATE      DATE,

PRIMARY KEY(PID),

FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));
```

• EVENTS

```
CREATE TABLE EVENTS

(NAME          VARCHAR(10)          NOT NULL,

HELD_TIME     DATE                  NOT NULL,

DESCRIPTION    VARCHAR(40)          ,

PRIMARY KEY(NAME));
```

• EVENT_HOLD_EMP

```
CREATE TABLE EVENT_HOLD_EMP

(NAME          VARCHAR(10)          NOT NULL,

E_SSN         VARCHAR(10)          NOT NULL,

PRIMARY KEY(NAME),

FOREIGN KEY(NAME) REFERENCES EVENTS(NAME));
```

• EVENT_HOLD_VOL

```
CREATE TABLE EVENT_HOLD_VOL

(NAME          VARCHAR(10)          NOT NULL,

V_ID          VARCHAR(11)          NOT NULL,

PRIMARY KEY(NAME),

FOREIGN KEY(NAME) REFERENCES EVENTS(NAME),

FOREIGN KEY(V_ID) REFERENCES VOLUNTEER(ID));
```

• VOLUNTEER

```
CREATE TABLE VOLUNTEER

(ID            VARCHAR(11)          NOT NULL,

AGE           VARCHAR(3),

AVAILVARCHAR(3),

PRIMARY KEY(ID),

CHECK(AGE>0 AND AGE<75));
```

• ATTENDEES

```
CREATE TABLE ATTENDEES

(INVITE_SR_NO  VARCHAR(10)          NOT NULL,

P_ID          VARCHAR(11)          NOT NULL,

RELATIVE_NAME VARCHAR(20),
```

E_SCORE VARCHAR(3) NOT NULL,

PRIMARY KEY(INVITE_SR_NO),

FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID),

CHECK(E_SCORE>0 AND E_SCORE<100));

• APPOINTMENT

CREATE TABLE APPOINTMENT

(DT DATE NOT NULL,

TIME DATE NOT NULL,

P_ID VARCHAR(11) NOT NULL,

PRIMARY KEY(DT,TIME),

FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));

• WORKS_AT

CREATE TABLE WORKS_AT

(D_LIC# VARCHAR(10) NOT NULL,

A_LIC# VARCHAR(10) NOT NULL,

PRIMARY KEY(D_LIC#,A_LIC#),

FOREIGN KEY(D_LIC#) REFERENCES DRIVER(S_SSN),

FOREIGN KEY(A_LIC#) REFERENCES AMBULANCE(LIC_NO));

• AMBULANCE

CREATE TABLE AMBULANCE

(LIC_NO VARCHAR(10) NOT NULL,

STORED_DT DATE ,

LOC VARCHAR(20),

PRIMARY KEY(LIC_NO));

• MEDICINES

CREATE TABLE MEDICINES

(ID VARCHAR(10) NOT NULL,

NAME VARCHAR(15) NOT NULL,

PRICE VARCHAR(4) NOT NULL,

STORED_IN_DT DATE ,

TYPE VARCHAR(10),

QTY VARCHAR(4) ,

PRIMARY KEY(ID));

• INSURANCE

CREATE TABLE INSURANCE

(C_NAME VARCHAR(10) NOT NULL,

POLICY_NO VARCHAR(10) NOT NULL,

P_ID VARCHAR(11) NOT NULL,

PHONE VARCHAR(10),

EXP_DT DATE,

PRIMARY KEY(C_NAME,POLICY_NO),

FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));

• CALL

CREATE TABLE CALL

(DT DATE NOT NULL,

TIME DATE NOT NULL,

P_ID VARCHAR(11) NOT NULL,

TREAT_TYPE VARCHAR(10),

DOC_NAME VARCHAR(20) NOT NULL,

REC_INFO VARCGAR(20) NOT NULL,

PRIMARY KEY(DT,TIME),

FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));

- ONL

CREATE TABLE ONL

(DT DATE NOT NULL,
TIME DATE NOT NULL,
P_ID VARCHAR(11) NOT NULL,
TREAT_TYPE VARCHAR(10),
DOC_NAME VARCHAR(20) NOT NULL,
PRIMARY KEY(DT,TIME),
FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));

- RELATIVES

CREATE TABLE RELATIVES

(P_ID VARCHAR(11) NOT NULL,
REL_NAME VARCHAR(20) NOT NULL,
PRIMARY KEY(P_ID,REL_NAME),
FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));

- BILL

CREATE TABLE BILL

(B_NO VARCHAR(10) NOT NULL,
P_ID VARCHAR(11) NOT NULL,
STATUS VARCHAR(10),
BED VARCHAR(10),
PRESCRIPTION VARCHAR(10),
AMBULANCE VARCHAR(10),
PHARMACY VARCHAR(5),
TOTAL VARCHAR(10), NOT NULL,
PRIMARY KEY(B_NO),
FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));

- EVENT_CITY

CREATE TABLE EVENT_CITY

(EVENT_NAME VARCHAR(20) NOT NULL,
EVENT_CITY VARCHAR(20) NOT NULL,
PRIMARY KEY(EVENT_NAME,EVENT_CITY),
FOREIGN KEY(EVENT_NAME) REFERENCES EVENTS(NAME));

- E_DEPTNAME

CREATE TABLE E_DEPTNAME

(E_SSN VARCHAR(10) NOT NULL,
DEPT_NAME VARCHAR(20) NOT NULL,
PRIMARY KEY(E_SSN,DEPT_NAME),
FOREIGN KEY(E_SSN) REFERENCES EMPLOYEE(SSN),
FOREIGN KEY(DEPT_NAME) REFERENCES DEPT(NAME));

- PATIENT_PH_NO

CREATE TABLE PATIENT_PH_NO

(P_ID VARCHAR(11) NOT NULL,
PH_NO VARCHAR(10) NOT NULL,
PRIMARY KEY(P_ID,PH_NO),
FOREIGN KEY(P_ID) REFERENCES PATIENT(P_ID));

3.1.3 A Database State

We insert some values into the database in order to test our SQL create view and query statement. Here we just give one example of insertions as follows:

➤ INSERTION OF TABLE EMPLOYEE

1. insert into Employee

values ('123674356','01', 'Rohan', '30,000', 'Male', 'r@gmail.com','Doctor','24','24-May-10','546 XYZ Apt','Courts mccallum','Dallas','Texas','75252');

2. insert into Employee

values ('123674389','02', 'Balika', '35,000', 'Female', 'b@gmail.com','Nurse','27','12-Aug-08','532 ABC Appt','Glenn mccallum','Dallas','Texas','75252');

3. insert into Employee

values ('123674334','03', 'Vinit', '50,000', 'Male', 'v@hotmail.com','Receptionist','20','29-Jan-11','4578 Meadows','Drive way','Chicago','Illinois','68133');

4. insert into Employee

values ('123672389','04', 'Jem', '37,000', 'Male', 'j@gmail.com','Pharmacist','30','06-Jul-05','4135 Downwards Appt','Dentour','San antanio','California','43167');

5. insert into Employee

values ('489957634','05', 'Maddy', '25,000', 'Male', 'm@gmail.com','Driver','52','14-Jul-10','7777 McCallum Blvd','Ashwood','Detroit','Michigan','48188');

6 insert into Employee

values ('378959208','06', 'Ashtha', '21,000', 'Female', 'a@yahoo.com','Driver','24','29-Sep-11','4315 Sherwood CIR','Akshar','Los Angelos','California','43986');

SSN	ID	NAME	SALARY	GENDER	EMAIL	TITLE	AGE	JOIN_IN_D	LINE	LINE2	CITY
STATE		ZIP									
123674356 TX	01	Rohan 75252	30000	Male	r@gmail.com	Dr.	24	24-MAY-10	546xyz Apt	Courts of McCallum	Dallas
123674334 Illionis	03	Unit 68133	50000	Male	v@hotmail.com	Receptionist	20	29-JAN-11	4578 Meadows	Drive Way	Chicago
123674389 TX	02	Balika 75252	35000	Female	b@gmail.com	Nurse	27	12-AUG-08	532abc Apt	Glen McCallum	Dallas
123672389 California	04	Jem 43167	37000	Male	j@gmail.com	Pharmacist	30	06-JUL-05	4135 Downwards Apt	Dentour	San Anta
489957634 Michigan	05	Maddy 48188	25000	Male	m@gmail.com	Driver	52	14-JUL-10	7777 McCallum Blvd.,	Ashwood	Detroit
378959208 California	06	Ashtha 43986	21000	Female	a@yahoo.com	driver	24	29-SEP-11	4315 Sherwood Cir	Akshar	Los Ange
6 rows selected.											
SQL>											

➤ INSERTION OF TABLE Doctor

1. insert into Doctor

Values ('123674356','3');

```
SQL> insert into doctor values(<'123674356','3'>);
1 row created.
SQL> select * from doctor;
SSN          D_L
-----
123674356      3
SQL>
```

➤ INSERTION OF TABLE Nurse_Lang

1. insert into Nurse_Lang

Values ('123674389','English');

```
SQL> insert into nurse_lang values(<'123674389','English'>);
1 row created.
SQL> select * from nurse_lang;
N_SSN      N_LANG
-----
123674389  English
SQL>
```

➤ INSERTION OF TABLE Nurse

1. insert into Nurse

Values('123674389');

```
SQL> insert into nurse values(<'123674389'>);
1 row created.
SQL> select * from nurse;
SSN
-----
123674389
SQL>
```

➤ INSERTION OF TABLE Doctor_info

1. insert into Doctor_info

Values ('123674356','Cardiologist','English','UT Dallas');

```
SQL> insert into doctor_info values(<'123674356','Cardiologst','English','UTDallas'>);
1 row created.
SQL> select * from doctor_info;
D_SSN      SPECIALITY LANG      M_SCHOOL
-----
123674356   Cardiologst English   UTDallas
SQL>
```

➤ INSERTION OF TABLE Pharmacist

1. insert into Pharmacist
- Values ('123672389','Pharm.D.');

```
SQL> insert into pharmacist values(<'123672389','Pharm.D.'>);
1 row created.
SQL> select * from pharmacist;
P_SSN      DEGREE
-----
123672389   Pharm.D.
SQL>
```

➤ INSERTION OF TABLE Treats

1. insert into Treats
- Values ('123674356','100','Positive','1200');

```
SQL> insert into treats values(<'123674356','100','Positive','1200'>);
1 row created.
SQL> select * from treats;
D_SSN      P_ID      RESULT      FEES
-----
123674356   100       Positive    1200
SQL>
```

➤ INSERTION OF TABLE Receptionist

1. insert into Receptionist
- Values ('123674334');

```
SQL> insert into receptionist values(<'123674334'>);
1 row created.
SQL> select * from receptionist;
R_SSN
-----
123674334
SQL>
```

➤ INSERTION OF TABLE Arrange_work

1. insert into Arrange_work
- Values ('123674356', '123674334', '123674389');

```
SQL> insert into arrange_work values(<'123674356','123674334','123674389'>);
1 row created.
SQL> select * from arrange_work;
D_SSN      R_SSN      N_SSN
-----
123674356   123674334   123674389
SQL>
```

➤ INSERTION OF TABLE Support_staff

1. insert into Support_staff
- Values ('489957634');
2. insert into Support_staff
- Values ('378959208');

```
SQL> insert into support_staff values(<'489957634'>);
1 row created.
SQL> insert into support_staff values(<'378959208'>);
1 row created.
SQL> select * from support_staff;
S_SSN
-----
378959208
489957634
SQL>
```

➤ INSERTION OF TABLE Part_time

1. insert into Part_time
- Values ('378959208','45');

```
SQL> insert into part_time values(&'378959208',&'45');
1 row created.
SQL> select * from part_time;
PS_SSN      NO_HO
-----
378959208   45
SQL>
```

➤ INSERTION OF TABLE Full_time

1. insert into Full_time
Values ('489957634');

```
SQL> insert into full_time values(&'489957634');
1 row created.
SQL> select * from full_time;
FS_SSN
-----
489957634
SQL>
```

➤ INSERTION OF TABLE Driver

1. insert into Driver
Values ('489957634','376AP189X8');
2. insert into Driver
Values ('378959208','961MD375W8');

```
SQL> insert into driver values(&'489957634',&'376AP189X8');
1 row created.
SQL> insert into driver values(&'378959208',&'961MD375W8');
1 row created.
SQL> select * from driver;
S_SSN      LIC#
-----
489957634   376AP189X8
378959208   961MD375W8
SQL>
```

➤ INSERTION OF TABLE Department

1. insert into Dept
Values (&'CA123',&'Cardiology',&'Hill Street',&'9728830023', &'123674356');

```
SQL> select * from dept;
ID      NAME      LOC      PHONE_NO  MGR_SSN
-----
CA123    Cardiology  HillStreet  9728830023 123674356
SQL>
```

➤ INSERTION OF TABLE Patient

1. insert into Patient
Values (&'100',&Arpit,&Patel,&Back Pain,&Maddy,&arp@gmail.com,&'28-Feb-71',&2214 HillCrest Drive,&Northern Bound,&Dallas,&Texas,&'75254',&Male,&Admitted on 11-12-2012.);
2. insert into Patient
Values (&'101',&Hema,&Dowson,&Lever Pain,&Ashtha,&hem@yahoo.com,&'21-Aug-68', &2498 Coit road, &Bishop Ave, &Las Vegas,&Califonia,&'37965',&Female,&Admitted on 1st October 2013.);

```
SQL> select * from patient;
P_ID      FNAME      LNAME      REASON      DRIVER      EMAIL_ID      DOB      LINE      LINE2      CITY      STATE
-----
HISTORY      ZIP
100      Arpit      Patel      BackPain      Maddy      arp@gmail.com      28-FEB-71 2214 HillCrest Drive Northern Bound  DALLAS      Texas
Admitted on 12 Nov-2012  75254
101      Hema      Dowson      LiverPain      AAstha      hem@yahoo.com      21-AUG-68 2498 Coit road  Bisop Ave  Las Vegas  California
Admitted on 1 october13  37965
SQL>
```

➤ INSERTION OF TABLE In_Patient

1. insert into In_Patient
Values (&'100',&2479063589,);
2. insert into In_patient
Values(&'101',&5632489632,&'20');

```
SQL> INSERT INTO IN_PATIENT VALUES(&'101',&'5632489632',&'20');
1 row created.
SQL> SELECT * FROM IN_PATIENT;
P_ID      SSN      RM#
-----
100      2479063589
101      5632489632  20
SQL>
```

➤ INSERTION OF TABLE Out_Patient

1. insert into Out_Patient

Values ('101');

```
SQL> insert into out_patient values(<'101'>);
1 row created.
SQL> select * from out_patient;
P_ID
-----
101
SQL>
```

➤ INSERTION OF TABLE Billed_Patient

1. insert into Billed_patient
- Values ('100','1234567890','Maya','Bank of America',' 2214 HillCrest Dallas TX','04-Mar-14');

```
SQL> select * from billed_patient;
P_ID      ACC_NO      ACC HOLDER      BANK NAME      BILLING_ADDR      EXP DATE
-----
100      1234567890  Maya      Bank of America      2214 Hillcrest Dallas TX      04-MAR-14
SQL>
```

➤ INSERTION OF TABLE Events

1. insert into Events
- Values ('Cancer Aid','12-May-13','Awareness against cancer.');

```
SQL> select * from events;
NAME      HELD TIME      DESCRIPTION
-----
Cancer Aid 12-MAY-13 awareness against cancer
SQL>
```

➤ INSERTION OF TABLE Event_hold_emp

1. insert into Event_hold_emp
- Values ('Cancer Aid', '123674334');

```
SQL> insert into event_hold_emp values(<'Cancer Aid','123674334'>);
1 row created.
SQL> select * from event_hold_emp;
NAME      E_SSN
-----
Cancer Aid 123674334
SQL>
```

➤ INSERTION OF TABLE Event_hold_vol

1. insert into Event_hold_vol
- Values ('Cancer Aid','200');

```
SQL> insert into event_hold_vol values(<'Cancer Aid','200'>);
1 row created.
SQL> select * from event_hold_vol;
NAME      U_ID
-----
Cancer Aid 200
SQL>
```

1. insert into volunteer
- Values ('200','45','30');
2. insert into volunteer
- Values ('201','24','40');

```
SQL> insert into volunteer values(<'200','45','30'>);
1 row created.
SQL> insert into volunteer values(<'201','24','40'>);
1 row created.
SQL> select * from volunteer;
ID      AGE  AUA
-----
200      45   30
201      24   40
SQL>
```

➤ INSERTION OF TABLE Attendees

1. insert into attendees
- Values ('i1','100','Romil','75');
2. insert into attendees
- Values ('i2','101','Vasim','98');

```
SQL> insert into attendees values(<'i1','100','Romil','75'>);
1 row created.
SQL> insert into attendees values(<'i2','101','Vasim','98'>);
1 row created.
SQL> select * from attendees;
INVITE_SR_ P_ID      RELATIVE_NAME      E_S
-----
i1      100      Romil      75
i2      101      Vasim      98
SQL>
```

➤ INSERTION OF TABLE Appointment

1. insert into appointment
- Values ('12-04-2013','13:00','100');

Date	Time	P_ID
04-DEC-2013	13:00	100

➤ INSERTION OF TABLE Works_At

1. insert into works_at
- Values ('376AP189X8','AMB020');

```
SQL> insert into works_at values(<'376AP189X8','AMB020'>);
1 row created.
SQL> select * from works_at;
D_LIC#      A_LIC#
-----
376AP189X8  AMB020
SQL>
```

- . insert into ambulance
- Values ('AMB020','01-Apr-11','Drive Hill Parking');

➤ INSERTION OF TABLE Medicines

```
SQL> insert into ambulance values(<'AMB020','01-APR-11','DriveHill Parking'>);
1 row created.
SQL> select * from ambulance;
LIC_NO      STORED_DT LOC
-----
AMB020      01-APR-11 DriveHill Parking
SQL>
```

➤ INSERTION OF TABLE Medicines

1. insert into medicines
- Values ('M01','Stopache','5','12-Feb-13','Pain Killer','150');
2. insert into medicines
- Values ('M02','Metacin','10','19-Apr-13','Coriza','200');

```
SQL> insert into medicines values(<'M01','stopache','5','12-FEB-13','Painkiller','150'>);
1 row created.
SQL> insert into medicines values(<'M02','metacin','10','19-APR-13','Coriza','200'>
2  ;
1 row created.
SQL> select * from medicines;
ID      NAME      PRIC STORED_IN TYPE      QTY
-----
M01     stopache     5    12-FEB-13 Painkiller 150
M02     metacin     10    19-APR-13 Coriza    200
SQL>
```

➤ INSERTION OF TABLE Insurance

1. insert into insurance
- Values ('Bajaj','P9123','100','9722430010','16-Mar-16');

```
SQL> select * from insurance;
C_NAME      POLICY_NO P_ID      PHONE      EXP_DT
-----
Bajaj       P9123     100       9722430010 16-MAR-16
SQL>
```

➤ INSERTION OF TABLE Call

- 1 . insert into call
- Values ('23-Sept-13','10:00','100','Surgery','Rohan','Vinit, EmpID: 03');

Date	Time	P_id	Treat_type	Doc_name	Rec_info
23-SEP-13	10:00	100	Surgery	Rohan	Vinit,Empid:03

➤ INSERTION OF TABLE Online

1. insert into onl
- Values ('23-Sept-13','11:00','100','Operation','Rohan');

Date	Time	P_id	Treat_type	Doc_name
23-Sep-13	11:00	100	Operation	Rohan

➤ INSERTION OF TABLE Relatives

1. insert into relatives
- Values ('100','Romil');

```
SQL> insert into relatives values(<'100','Romil'>);
1 row created.
SQL> select * from relatives;
P_ID      REL_NAME
-----
100       Romil
SQL>
```

➤ INSERTION OF TABLE Bill

1. insert into Bill
Values ('B600','100','Paid','40','food & Amenities','100','75','215');

```
SQL> INSERT INTO BILL VALUES('B600','100','PAID','40','FOOD AND AMENITIES','100','75','215');
1 row created.
SQL> SELECT * FROM BILL;
B_NO      P_ID      STATUS    BED        PRESCRIPTION    AMBULANCE  PHARM  TOTAL
-----
B600      100       PAID      40         FOOD AND AMENITIES  100        75    215
SQL>
```

➤ INSERTION OF TABLE Event_City

1. insert into event_city
Values ('Cancer Aid','Dallas');

```
SQL> insert into event_city values('Cancer Aid','Dallas');
1 row created.
SQL> select * from event_city;
EVENT_NAME      EVENT_CITY
-----
Cancer Aid      Dallas
SQL>
```

➤ INSERTION OF TABLE E_DeptName

1. insert into E_DeptName
Values ('123674356','Cardiology');

```
SQL> insert into e_deptname values('123674356','Cardiology');
1 row created.
SQL> select * from e_deptname;
E_SSN      DEPT_NAME
-----
123674356  Cardiology
SQL>
```

➤ INSERTION OF TABLE PATIENT_PH_NO

1. insert into PATIENT_PH_NO
Values ('100','9723101775');

```
SQL> INSERT INTO PATIENT_PH_NO VALUES('100','9723101775');
1 row created.
SQL> SELECT * FROM PATIENT_PH_NO;
P_ID      PH_NO
-----
100       9723101775
SQL>
```

Till now we finished the process of creating tables and database states.

3.2 Creation of Views (Answer for Question d)

1. View1: This view returns all patients related to any treatments in the system.

```
CREATE VIEW T_PATIENT
AS SELECT P.P_ID, P.FNAME, P.LNAME
FROM PATIENT P, TREATMENT T
WHERE P.P_ID=T.P_ID;
```

2. View2: This view returns all patients associated with any bills with a total amount more than \$10,000.

```
CREATE VIEW BILL_OF_PAT
AS SELECT P.P_ID, P.FNAME, P.LNAME
FROM PATIENT P, BILL B
WHERE P.P_ID=B.P_ID
AND B.TOTAL>10000;
```

3. View3: This view returns all events in Dallas.

```
CREATE VIEW DAL_EVENT
AS SELECT E.NAME
FROM EVENTS E, EVENT_CITY EC
WHERE E.NAME=EC.EVENT_NAME AND EC.EVENT_CITY='DALLAS';
```

4. View4: This view returns all senior doctors (Lev.4-5) worked on any surgery since 1/1/2013.

```
CREATE VIEW SENIOR_DOC
AS SELECT E.NAME, E.ID
FROM EMPLOYEE E, DOCTOR D, DOCTOR_INFO DI
WHERE D.SSN=E.SSN AND (D.D_L=4 OR D.D_L=5) AND DI.D_DDN=D.SSN AND DI.SPECIALITY='SURGEON' AND E.JOIN_IN_DATE>01-JAN-13;
```

3.3 Creation of SQL Queries (Answer for Question f)

Now we give out the SQL Queries for each of 14 questions listed in Question e as follows:

1. Retrieve the IDs and Names of all nurses who live in Dallas.

```
SQL> select e.id,e.name
  2  from employee e,nurse n
  3  where n.ssn=e.ssn and e.city='Dallas';

ID          NAME
-----
02          Balika

SQL>
```

2. Retrieve the Names and Addresses of all in-patients that lived in ward No.20 between 01/01/2013 and 03/31/2013.

```
SELECT P.LNAME, P.FNAME, P.LINE, P.LINE2, P.CITY
FROM PATIENT P, IN_PATIENT IP
WHERE P.P_ID=IP.P_ID
AND IP.RM#='20'
AND IP.SDATE BETWEEN 01-JAN-13 AND 31-MAR-13
```

3. Retrieve the IDs and Names of distinct employees whose salaries are higher than the average salary of all the employees in the same department.

```
SELECT DISTINCT E.ID, E.NAME
FROM EMPLOYEE E,E_DEPT D
WHERE E.SALARY>E.AVG (SALARY) AND E.DEPT_ID=D.ID
GROUPBY D.DEPT_NAME;
```

4. Retrieve the Names of patients who have surgery appointments with Dr. Gregory House on 01/01/2013.

```
SELECT P.NAME
FROM PATIENT P,CALL C
WHERE C.TREAT_TYPE='SURGERY' AND C.DOC_NAME='DR. GREGORY HOUSE' AND DT='01-JAN-13' AND C.P_ID=P.P_ID;
UNION
SELECT P.NAME
FROM PATIENT P, ONL O
WHERE O.TREAT_TYPE='SURGERY' AND O.DOC_NAME='DR. GREGORY HOUSE' AND DT='01-JAN-13' AND O.P_ID=P.P_ID;
```

5. Retrieve the Names and DL# of all drivers whose ages are older than 50.

```
SELECT E.NAME, D.LIC#
FROM EMPLOYEE E, DRIVER D
WHERE D.S_SSN=E.SSN AND E.AGE>50;
```

6. Retrieve the total number of ambulances dispatched during 2012.

```
SELECT COUNT (*)
FROM AMBULANCE
WHERE STORED_DT BETWEEN 01-JAN-12 AND 31-DEC-12;
```

7. Add a bank financial record to the patient ID “mxl123456”.

```
SQL> insert into billed_patient values('mxl123456','0932558964','Robert','Chase Bank','7740 McCallum,Dallas','2-JAN-14');
1 row created.

SQL>
```

8. Retrieve the SSNs, and Addresses of all in-patients whose insurances have expired today.

```
SELECT I.SSN,P.LINE,P.LINE2,P.CITY,P.STATE,P.ZIP
FROM IN_PATIENT I, PATIENT P, INSURANCE IN
WHERE I.P_ID=IN.P_ID AND I.P_ID=P.P_ID AND IN.EXP_DT=SYSDATE;
```

9. Retrieve the Name of the most popular Rx-medicine prescribed by senior doctors in "Cardiology" department and dispensed in 2013.

```
SELECT M.NAME, COUNT (M.ID)
FROM MEDICINE M, TREATS T, EMPLOYEE E, DEPT D
WHERE T.M_ID=M.ID
AND E.DEPT_ID=D.ID AND D.Name='CARDIOLOGY' AND M.DISPENSE_DT>12/31/2012 AND M.DISPENSE_DT<1/1/2014
GROUP BY M.ID
```

- 10.Retrieve the Names, SSNs and Emails of distinct patients who have any bill with a total amount bigger than \$10,000.

```
SELECT DISTINCT P.FNAME, P.LNAME, P.EMAIL_ID,IN.SSN
FROM IN_PATIENT IN, PATIENT P, BILL B
WHERE B.P_ID=IN.P_ID
AND B.TOTAL>10000;
```

11. Retrieve the Names, and Phone Numbers of all distinct patients that have appointments tested by all nurses in the department of Diagnostic Medicine.

```
SELECT DISTINCT P.FNAME, P.LNAME, PP.PH_NO
FROM PATIENT P, PATIENT_PH_NO PP, TREATS T, NURSE N, EMPLOYEE E, DEPT D
WHERE T.P_ID=P.P_ID
AND T.D_SSN=N.SSN
AND N.SSN=E.SSN
AND T.P_ID=PP.P_ID
AND E.DEPT_ID=D.ID
AND D.NAME='DIAGNOSTIC MEDICINE';
```

12. Retrieve the Gross Profit of the hospital between 01/01/2012 and 12/31/2012. (To put it simply, Gross Profit = total amount of all bills issued in 2012 – sum of all employees' salaries)

```
SELECT A.EXP-B.PRO FROM (SELECT SUM (TOTAL)EXP FROM BILL WHERE B_NO IN (SELECT B_NO FROM BILL WHERE P_ID IN (SELECT P_ID FROM PATIENT
WHERE ADMIT_DT LIKE'___/___/2012'))A,(SELECT SUM (SALARY) PRO FROM EMPLOYEE)B.
```

13. For each year, retrieve the Date and the largest number of visits to the hospital during that day.

```
SELECT COUNT(*) FROM APPOINTMENT GROUP BY DT;
```

14. Retrieve the average age of all patients who only choose online scheduling services.

```
SELECT AVG (SYSDATE-P.DOB) AS AVG_AGE
FROM PATIENT P, ONL O
WHERE O.P_ID=P.P_ID
```

15. Retrieve the attendees’ first names and last names who participate any event of all themes in Dallas and with the evaluation score bigger and equal to 80.

```
SELECT P.P_ID, R.REL_NAME, P.FNAME, P.LNAME
FROM ATTENDEES A, EVENT_CITY E, PATIENT P, RELATIVES R
WHERE A.P_ID=P.P_ID,
AND A.P_ID=R.P_ID,
AND A.EVENT_NAME=E.EVENT_NAME,
AND A.E_SCORE>=80;
```

4. Conclusion

In this report we modified the EER diagram and relational schemas for **HOSPITAL MANAGEMENT SYSTEM** Database according to the requirement of Phase III. We also give dependency diagram for each relational schema in database. Then we created tables for each relational schema and write the SQL statements for the views and queries listed in Question d and Question e.