Project for Database Design

Phase III. Implementation

Group 14

Haoran Lou, Yao Li, Yongliang Yang

1. **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 e, 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.

# Modified Relational Schema

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

* We assume that one person has one phone number. Then the phone number relation no longer exists.
* We give doctor, nurse, receptionist each of them an id attribute.
* We modify the payment relation, we add the cash relation and insurance relation in payment into one relation.
* We add an Doc\_ID to the Class\_2\_patient relation.

The relation schema is below:



# Dependency Diagram

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

## Hospital Personnel

There is only one attribute in the left-hand side of the functional dependencies, which is the key of relational schema Hospital Personnel, Person\_ID. Therefore, every other attribute of this relational schema is functionally dependent on Person\_ID. The dependency diagram is shown as Figure 1.



Figure 1. Dependency Diagram of Hospital Personnel

## Class\_1\_Patient

There is only one attribute in the left-hand side of the functional dependencies, which is the key of relational schema Class\_1\_Patient, Patient\_ID. Therefore, every other attribute of this relational schema is functionally dependent on Patient\_ID. The dependency diagram is shown as Figure 2.



Figure 2. Dependency Diagram of Class\_1\_Patient

## Hospital Employee

There is only one attribute in the left-hand side of the functional dependencies, which is the key of relational schema Hospital Employee,Employee\_num. Therefore, every other attribute of this relational schema is functionally dependent on Employee\_num. The dependency diagram is shown as Figure 3.



Figure 3. Dependency Diagram of Employee

## Hospital Doctor

There is only one attribute in the left-hand side of the functional dependencies, which is DOC\_ID. The dependency diagram is shown as Figure 4.



Figure 2. Dependency Diagram of Doctor

## Relation Access

There are two attribute in the left-hand side of the functional dependencies, which are Medicine code and Treatment ID. The dependency diagram is shown as Figure 5.



Figure 5. Dependency Diagram of Access

## Hospital Nurse

The dependency diagram is shown as Figure 6.



Figure 6. Dependency Diagram of Nurse

## Hospital Room

The dependency diagram is shown as Figure 7.



Figure 7. Dependency Diagram of Room

## Hospital Receptionist

The dependency diagram is shown as Figure 8.



Figure 8. Dependency Diagram Receptionist

## Patient Records

The dependency diagram is shown as Figure 9.



Figure 9. Dependency Diagram of Records

## Payment Information

The dependency diagram is shown as Figure 10.



Figure 10. Dependency Diagram of Payment

## Medical Information

The medicine code is depend on both T\_ID and Class\_2\_ID, thus the primary of this relation is T\_ID together with Class\_2\_ID. The dependency diagram is shown as Figure 11.



Figure 11. Dependency Diagram

## Hospital Pharmacy

The dependency diagram is shown as Figure 12.



Figure 12. Dependency Diagram

## Hospital Treatment

The dependency diagram is shown as Figure 13.



Figure 13. Dependency Diagram

## Class 2 Patient

The dependency diagram is shown as Figure 14.



Figure 14. Dependency Diagram

## Visitor Log

The dependency diagram is shown as Figure 15.



Figure 15. Dependency Diagram of Hospital Personnel

## 2.Final Results

After drawing the dependency diagrams one after another, Figure 16 shows the final results for the whole database including the ones who do not have any functional dependencies.





Figure 16. Whole Dependency Diagram for Dallas Care Database

# Implementation of Database

## 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 15 tables as follows:

* **ACCSEE1:**

CREATE TABLE ACCESS1

(

DOC\_ID VARCHAR(200) NOT NULL,

MEDICINE\_CODE VARCHAR(200) NOT NULL,

T\_ID VARCHAR(200) NOT NULL,

CLASS\_2\_ID VARCHAR(200) NOT NULL,

primary key (DOC\_ID, MEDICINE\_CODE, T\_ID, CLASS\_2\_PATIENT\_ID),

foreign key (MEDICINE\_CODE, T\_ID, CLASS\_2\_ID) references MEDICAL\_INFORMATION(MEDICINE\_CODE, T\_ID, CLASS\_2\_ID),

foreign key (DOC\_ID) references DOCTOR(DOC\_ID)

);

* **CLASS\_1\_PATIENT**

CREATE TABLE CLASS\_1\_PATIENT

(

PATIENT\_ID VARCHAR(255) NOT NULL,

PERSON\_ID VARCHAR(255) NOT NULL,

CLASS\_2\_ID VARCHAR(255),

EMPLOYEE\_NUM VARCHAR(255),

primary key (PATIENT\_ID),

foreign key (PERSON\_ID) references PERSON(PERSON\_ID)

);

alter table CLASS\_1\_PATIENT

add DOC\_ID varchar(255) null;

alter table CLASS\_1\_PATIENT

add foreign key (DOC\_ID) references DOCTOR(DOC\_ID);

* **CLASS\_2\_PATIENT**

CREATE TABLE CLASS\_2\_PATIENT

(

T\_ID VARCHAR(200),

EMPLOYEE\_NUM VARCHAR(200),

CLASS\_2\_ID VARCHAR(200) NOT NULL,

DATE\_OF\_ADMITTED DATE,

ROOM\_ID VARCHAR(255),

primary key(CLASS\_2\_ID),

foreign key(T\_ID) references TREATMENT(T\_ID),

foreign key(ROOM\_ID) references ROOM(ROOM\_ID)

);

alter table CLASS\_2\_PATIENT

add DOC\_ID varchar(255) null;

alter table CLASS\_2\_PATIENT

add foreign key (DOC\_ID) references DOCTOR(DOC\_ID);

* **DOCTOR**

CREATE TABLE DOCTOR

(

EMPLOYEE\_NUM VARCHAR(255) NOT NULL,

DOC\_ID VARCHAR(255),

DOC\_TYPE VARCHAR(255),

primary key(DOC\_ID),

foreign key (EMPLOYEE\_NUM) references EMPLOYEE(EMPLOYEE\_NUM)

);

* **EMPLOYEE**

CREATE TABLE EMPLOYEE

(

PERSON\_ID VARCHAR(255) NOT NULL,

EMPLOYEE\_NUM VARCHAR(255) NOT NULL,

START\_DATE DATE,

CLASS\_2\_ID VARCHAR(255),

primary key (EMPLOYEE\_NUM),

foreign key (PERSON\_ID) references PERSON(PERSON\_ID)

);

* **MEDICAL\_INFORMATION**

CREATE TABLE MEDICAL\_INFORMATION

(

MEDICINE\_CODE VARCHAR(200) NOT NULL,

T\_ID VARCHAR(200) NOT NULL,

CLASS\_2\_ID VARCHAR2(200) NOT NULL,

primary key (MEDICINE\_CODE,T\_ID,CLASS\_2\_ID),

foreign key(T\_ID) references TREATMENT(T\_ID),

foreign key(CLASS\_2\_ID) references CLASS\_2\_PATIENT(CLASS\_2\_ID)

);

* **NURSE**

CREATE TABLE NURSE

(

EMPLOYEE\_NUM VARCHAR(255) NOT NULL,

NURSE\_ID VARCHAR(255) NOT NULL,

primary key(NURSE\_ID),

foreign key (EMPLOYEE\_NUM) references EMPLOYEE(EMPLOYEE\_NUM)

);

* **PYMENT**

CREATE TABLE PAYMENT

(

PAYMENT\_ID VARCHAR2(20) NOT NULL,

PAYMENT\_DATE DATE,

TOTAL\_AMOUNT\_DUE VARCHAR(20),

RECEPTIONIST\_ID VARCHAR(20),

PATIENT\_ID VARCHAR(40),

T\_ID VARCHAR(20),

CASH\_AMOUNT VARCHAR(100),

I\_ID VARCHAR(60),

I\_PROVIDERID VARCHAR(100),

I\_COVERAGE VARCHAR(250),

I\_AMOUNT VARCHAR(100),

primary key (PAYMENT\_ID),

foreign key (RECEPTIONIST\_ID) references RECEPTIONIST(RECEPTIONIST\_ID),

foreign key (PATIENT\_ID) references CLASS\_1\_PATIENT(PATIENT\_ID)

);

alter table PAYMENT

add CLASS\_2\_ID varchar(255) null;

alter table PAYMENT

add foreign key (CLASS\_2\_ID) references CLASS\_2\_PATIENT(CLASS\_2\_ID);

alter table PAYMENT

add foreign key (T\_ID) references TREATMENT(T\_ID);

* **PERSON**

CREATE TABLE PERSON

(

PERSON\_ID VARCHAR(255) NOT NULL,

F\_NAME VARCHAR(255) NOT NULL,

M\_NAME VARCHAR(255),

L\_NAME VARCHAR(255) NOT NULL,

ADDRESS VARCHAR(255),

GENDER VARCHAR(255),

BITH\_DATE DATE,

PHONE\_NUMBER VARCHAR(255),

primary key (PERSON\_ID)

);

* **PHARMACY**

CREATE TABLE PHARMACY

(

MEDICINE\_CODE VARCHAR(200) NOT NULL,

P\_PRICE VARCHAR(200),

P\_NAME VARCHAR(200),

P\_QUANTITY VARCHAR(200),

P\_EXPIRECTION\_DATE DATE,

primary key(MEDICINE\_CODE)

);

* **RECEPTIONIST**

CREATE TABLE RECEPTIONIST

(

RECEPTIONIST\_ID VARCHAR(255) NOT NULL,

EMPLOYEE\_NUM VARCHAR(255) NOT NULL,

primary key (RECEPTIONIST\_ID),

foreign key (EMPLOYEE\_NUM) references EMPLOYEE(EMPLOYEE\_NUM)

);

* **RECORDS**

CREATE TABLE RECORDS

(

RECORD\_ID VARCHAR(255) NOT NULL,

RECEPTIONIST\_ID VARCHAR(255),

PATIENT\_ID VARCHAR(255),

VISIT\_DATE DATE,

APPOINTMENT DATE,

RECORD\_DESCRIPTION VARCHAR(255),

primary key(RECORD\_ID),

foreign key (RECEPTIONIST\_ID) references RECEPTIONIST(RECEPTIONIST\_ID),

foreign key (PATIENT\_ID) references CLASS\_1\_PATIENT(PATIENT\_ID)

);

* **ROOM**

CREATE TABLE ROOM

(

ROOM\_ID VARCHAR(255) NOT NULL,

ROOM\_TYPE VARCHAR(255),

ROOM\_DURATION VARCHAR(255),

NURSE\_ID VARCHAR(255) NOT NULL,

primary key(ROOM\_ID),

foreign key (NURSE\_ID) references NURSE(NURSE\_ID)

);

* **TREATMENT**

CREATE TABLE TREATMENT

(

T\_ID VARCHAR(200) NOT NULL,

T\_NAME VARCHAR(250),

T\_DURATION VARCHAR(200),

primary key(T\_ID)

);

* **VISITOR\_LOG**

CREATE TABLE VISITOR\_LOG

(

CLASS\_2\_ID VARCHAR(200),

VISITOR\_ID VARCHAR(200) NOT NULL,

VISITOR\_NAME VARCHAR(200),

V\_ADDRESS VARCHAR(200),

V\_CONTACT\_INFO VARCHAR(200),

primary key(VISITOR\_ID),

foreign key (CLASS\_2\_ID) references CLASS\_2\_PATIENT(CLASS\_2\_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 PERSON

--------------------------------------------------------------------------------------------------------

insert into PERSON

values ('10000', 'Emily', 'A', 'Navathe', ‘2665 Main St., Denton, TX 75083’, ‘Female’, '1980-04-30',’ 2144567626’

---------------------------------------------------------------------------------------------------------------------------

Table shows the states for PERSON database schemas.

Person

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Person\_ID | F\_Name | M\_Name | L\_Name | Address | Gender | Birth\_Date | Phone\_Num |
| 10000 | Emily | A | Navathe | 2665 Main St., Denton, TX 75083 | Female | 1980/4/30 | 2.14E+09 |
| 10001 | Tom | B | Brown | 263 Green St., Dallas, TX 75076 | Male | 1956/1/12 | 2.14E+09 |
| 10002 | Jimmy | C | Johnson | Apt.14, 3663 Beltline Blvd., Dallas, TX 75034 | Male | 1980/2/3 | 4.7E+09 |
| 10003 | Sally | D | Smith | 744 Walnut St., Dallas, TX 75074 | Female | 1976/3/26 | 2.14E+09 |
| 10004 | Jeniffer | E | Smack | 467 Parker St., Plano, TX 75076 | Female | 1957/4/5 | 2.15E+09 |
| 10005 | Smuel | F | Sunder | 18675 Chase Oak St., Frisco, TX 75034 | Male | 1997/5/20 | 9.72E+09 |
| 10006 | Raja | G | Farage | 556 Spring St., Mosquite, TX 75087 | Male | 2000/6/3 | 9.73E+09 |
| 10007 | Kenneth | H | Chenault | 2445 Wolf Creek St., Greenvill, TX 75056 | Male | 1979/7/16 | 2.14E+09 |
| 10008 | Brett | I | Cotton | 24567 Walnut St., The Colony, TX 75032 | Male | 1956/8/19 | 4.69E+09 |
| 10009 | Adam | J | Daley | 865 Park St., Garland, TX 75073 | Male | 1935/9/24 | 4.69E+09 |
| 10010 | George | K | Cobb | 263 Beltline Ave., Carleton, TX 75008 | Male | 1945/1/12 | 4.7E+09 |
| 10011 | Ivor | L | Page | 1247 Floyd Rd., Richardson, TX 75075 | Male | 1943/8/19 | 9.73E+09 |
| 10012 | Joseph | M | Tomason | 9454 RoyleLine Blvd., Irving, TX 75042 | Male | ######## | 9.73E+09 |
| 10013 | Sara | N | Gaddis | 345 King St., Fort Worth, TX 75023 | Female | 1974/4/27 | 9.72E+09 |
| 10014 | Aaron | A | Lee | 346 King St., Fort Worth, TX 75023 | Male | 1980/2/5 | 9.72E+09 |
| 10015 | Adolph |  | Young | 347 King St., Fort Worth, TX 75023 | Male | 1980/2/6 | 9.72E+09 |
| 10016 | Alan |  | King | 348 King St., Fort Worth, TX 75023 | Male | 1980/2/7 | 9.72E+09 |
| 10017 | Albert |  | Hall | 349 King St., Fort Worth, TX 75023 | Male | 1980/2/8 | 9.72E+09 |
| 10018 | Alcander |  | Scott | 301 King St., Fort Worth, TX 75023 | Male | 1980/2/9 | 9.72E+09 |
| 10019 | Alvin |  | Roberts | 302 King St., Fort Worth, TX 75023 | Male | 1980/2/10 | 9.72E+09 |
| 10020 | Andy |  | Phillips | 303 King St., Fort Worth, TX 75023 | Male | 1980/2/11 | 9.72E+09 |
| 10021 | Angus |  | Cook | 304 King St., Fort Worth, TX 75023 | Male | 1980/2/12 | 9.72E+09 |
| 10022 | Anker |  | Bell | 305 King St., Fort Worth, TX 75023 | Male | 1980/2/13 | 9.72E+09 |
| 10023 | Anthony |  | Richardson | 306 King St., Fort Worth, TX 75023 | Male | 1980/2/14 | 9.72E+09 |
| 10024 | Asher |  | Howard | 307 King St., Fort Worth, TX 75023 | Male | 1980/2/15 | 9.72E+09 |
| 10025 | August |  | Gray | 308 King St., Fort Worth, TX 75023 | Male | 1980/2/16 | 9.72E+09 |
| 20026 | Bali |  | Johnson | 309 King St., Fort Worth, TX 75023 | Male | 1980/2/17 | 9.72E+09 |
| 20027 | Barclay |  | Williams | 310 King St., Fort Worth, TX 75023 | Male | 1980/2/18 | 9.72E+09 |
| 20028 | Barnett |  | Jones | 311 King St., Fort Worth, TX 75023 | Male | 1980/2/19 | 9.72E+09 |
| 20029 | Barney |  | Brown | 312 King St., Fort Worth, TX 75023 | Male | 1980/2/20 | 9.72E+09 |
| 20030 | Baron |  | Davis | 313 King St., Fort Worth, TX 75023 | Male | 1980/2/21 | 9.72E+09 |
| 20031 | Barrett |  | Miller | 314 King St., Fort Worth, TX 75023 | Male | 1980/2/22 | 9.72E+09 |
| 20032 | Barth |  | Wilson | 315 King St., Fort Worth, TX 75023 | Male | 1980/2/23 | 9.72E+09 |
| 20033 | Beck |  | Moore | 316 King St., Fort Worth, TX 75023 | Male | 1980/2/24 | 9.72E+09 |
| 30034 | Ben |  | Taylor | 317 King St., Fort Worth, TX 75023 | Male | 1980/2/25 | 9.72E+09 |
| 20035 | Benson |  | Anderson | 318 King St., Fort Worth, TX 75023 | Male | 1980/2/26 | 9.72E+09 |
| 30036 | Berkeley |  | Thomas | 319 King St., Fort Worth, TX 75023 | Male | 1980/2/27 | 9.72E+09 |
| 30037 | Bern |  | Jackson | 320 King St., Fort Worth, TX 75023 | Male | 1980/2/28 | 9.72E+09 |
| 30038 | Bert |  | White | 321 King St., Fort Worth, TX 75023 | Male | 1980/2/29 | 9.72E+09 |
| 30039 | Bill |  | Harris | 322 King St., Fort Worth, TX 75023 | Male | 1980/3/1 | 9.72E+09 |
| 30040 | Billy |  | Martin | 323 King St., Fort Worth, TX 75023 | Male | 1980/3/2 | 9.72E+09 |

This dataset is also the source of other tables.

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

## 3.2 Creation of Views (Answer for Question d)

#### 3.2.1 Employees-Hired

Use the Create View statement to create the following views:

1. TopDoctor- This view returns the First Name, Last Name and Date of Joining

of those doctors who have made more than 5 Class 1 patients and over 10

Class 2 patients.

CREATE VIEW TopDoctor as

Select P.FName,P.LName,E.Start\_Date ,Doc\_ID

 From Person as P,Employee E, Doctor D

Where E.Person\_ID =P.Person\_ID and D.Employee\_Num = E.Employee\_Num and E.Employee\_Num in

(

(select count(\*)

From class \_1\_patient as c1p,

Group by c1p.Employee\_Num

HAVING count(\*) > 5)

and

(select count(\*)

From Class\_2\_Patient as c2p

Group by c2p.Employee\_Num

HAVING count(\*)>10)

);

2 TopTreatment- This view returns the treatment name of the most common

treatment in Dallas Care along with the bill payment amount when a person

receives that treatment.

CREATE VIEW TopTreatment as

Select T.name, T.T\_ID

From Treatment T,Payment P,

Where P.T\_ID=T.T\_ID and T.T\_ID exist

(select P.T\_ID ,max(count(\*))

From Payment P

Group by P.T\_ID

)

3 ReorderMeds- This view returns the medicines that need to be reordered. A

medicine needs to be reordered if the expiration date is 1 month from current

date or quantity is less than 1000.

CREATE VIEW ReorderMeds as

Select Medicine\_Code

From Pharmacy

Where ((to\_char(sysdate,'mm') - (to\_char(P\_Expirection\_Date,'mm')))<=1 or (P\_Quantity<1000)

4. PotentialPatient- This view returns the name, phone number and ID of patients who visited the hospital more than 3 times as a Class 1 patient but has not been admitted yet.

CREATE VIEW PotentialPatient as

Select F\_name, L\_name, Phone\_Number,Patient\_ID,Person\_ID

From Record r, Person p,Class\_1\_Patient c1p

Where r.Patient\_ID =c1p.Patient\_ID and c1p.person\_ID =p.Person\_ID and c1p.class\_2\_ID = Null and p.Patient\_ID in

(

Select r.patient\_ID,count(\*)

Form Record

Group by r.patient\_ID

HAVING count(\*) >3

)

5. MostFrequentIssues - This view returns the maximum frequency of the reason that patients visit the hospital for and the associated treatment for the same. For example, if patients visit the hospital mostly complaining about heart issues then what are the treatment associated with heart issues.

CREATE VIEW PotentialPatient as

View5:

CREATE VIEW FrequentIssues AS

select T.T\_Name ,count(\*)

from Treatment T

Group by distinct T.T\_name

order by count(\*) desc

limit 1

## 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.For each Doctor class, list the start date and specialization of the doctor.

Select E.start\_date,D.type

From Doctor D,Employee E

Where D.Employee\_Num = E.Employee\_Num

2. Find the names of employees who have been admitted to the hospital within 3

months of joining.

Select P.F\_name ,P.L\_name

From Person p,Employee e

Where p.person\_ID = e.person\_ID and e.start\_date =(sysdate -e.start\_date)<90)

3. Find the age and class (trainee, visiting or permanent) of top 5 doctors in the hospital.

SELECT F\_name,L\_name,((to\_char(sysdate,'yyyy'))-to\_char(E.birth\_date,'yyyy'))) as age,Doc\_type

FROM TopDoctor as T, Person P, Employee E, Doctor D

WHERE T.Doc\_ID=D.Doc\_ID

4. Find the name of medicines associated with the most common treatment in the

hospital.

Select ph.name

From Medical\_infomation me,Pharmacy as ph,TopTreatment as top

Where me.medicine\_code = ph.medicine\_code and top.T\_ID = me.T\_ID

5. Find all the doctors who have not had a patient in the last 5 months. (Hint:

Consider the date of payment as the day the doctor has attended a patient/been

consulted by a patient.)

Select F\_name ,L\_name

From Person p ,Employee e ,Doctor d

Where p.Person\_ID =e.Person\_ID and d.Employee\_Num=e.Employee.Num and (D.doc\_ID ,D.Employee\_Num) in

(

Select d.Doc\_ID ,d.Employee\_Num,

From class\_2\_patient c2p, class\_1patient c1p ,payment p

Where c1p.class\_2\_ID=c2p.class\_2\_ID and c2p.doc\_ID=D.doc\_ID and ((to\_char(sysdate,'mm') - (to\_char(Date\_of\_admitted,'mm')))>5 and ((to\_char(sysdate,'mm') - (to\_char(payment\_date,'mm')))>5

)

6. Find the total number of patients who have paid completely using insurance

and the name of the insurance provider.

Select I\_Provider\_name ,count(\*)

From Class\_1\_Patient c1p, Class\_2\_Patient c2p,Payment p

Where c1p.Class\_2\_ID= c2p.Class\_2\_ID and p.Patient\_ID = c1p.Patient\_ID and p.I\_ID <> null

Group by p.I\_Provider\_name

7.Find the most occupied room in the hospital and the duration of the stay.

Select r.Room\_ID ,r.duration

From Room r

Where duration = max(r.duration)

8. Find the year with the maximum number of patient visiting the hospital and

the reason for their visit.

Select r.Record\_Description, to\_char(Visit\_Date,'yyyy') as year

From Records r

Where year in

(

Select year ,r.Patient\_ID, max(count(\*))

From Records r

Group by r.Patient\_ID

)

9 Find the duration of the treatment that is provided the least to patients.

SELECT Name

FROM TREATEMENT

WHERE duration=

(

SELECT Min(Duration)

FROM TREATEMENT

)

10. List the total number of patients that have been admitted to the hospital after

the most current employee has joined.

SELECT COUNT(C.Class\_2\_ID) as COUNTNUMBER

FROM Class\_2\_Patient C

WHERE C.Date\_Of\_Admitted>(

SELECT MAX(Start\_date)

FROM DOCTOR,EMPLOYEE

WHERE DOCTOR.Employee\_Num=EMPLOYEE.Employee\_Num

)

11. List all the patient records of those who have been admitted to the hospital

within a week of being consulted by a doctor.

SELECT DISTINCT t3.\*

FROM Class\_1\_Patient t1 LEFT JOIN Class\_2\_Patient t2 on t2.Class\_2\_ID=t1.Class\_2\_ID

LEFT JOIN RECORDS t3 on t1.Patient\_ID=t3.Patient\_ID

WHERE (to\_char(t2.Date\_Of\_admitted,'yyyymmdd') - (to\_char(t3.Visit\_Date,'yyyymmdd'))<7

12.Find the total amount paid by patients for each month in the year 2017.

Select sum(Total\_Amount\_Due)

From Payment

Where Payment\_Date = to\_char(Payment\_Date,'2017')

Group by (Payment\_ID)

13. Find the name of the doctors of patients who have visited the hospital only

once for consultation and have not been admitted to the hospital.

Select F\_name,L\_name

From Person p,Doctor d, Class\_1\_Patient as c1p, Employee e

Where p.person\_id =e.person\_id and e.Employee\_Num = d.Employee\_Num and c1p.Doc\_ID =d.Doc\_ID and c1p.Class\_2\_ID = null and (c1p.Patient\_ID) exist

(

Select r.Patient\_ID,count(Visit\_Date)

From Records r

Where count(Visit\_Date)=1

Group by r.patient\_id

14. Find the name and age of the potential patients in the hospital.

Select F\_name, L\_name,,((to\_char(sysdate,'yyyy'))-to\_char(E.birth\_date,'yyyy'))) as age

From PotentialPatient po,Person p

Where po.Person\_ID= p.Person\_ID

# Conclusion

In this report we modified the EER diagram and relational schemas for Dallas Care 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.