**DATABASE PROJECT**

On this project you can work in pairs. Due December 1st.

Design a database for storing information about medical doctors, their patients, and illnesses:

1. Draw its E-R diagram.

Simplify

Treat

Suffer

Consult

Illness

Patient

Doctor

2. Give a relational representation of the E-R diagram that allows for 2 functional preserving and lossless join decompositions. Describe its functional dependencies.

Original (d.ddl)

Relations:

Doctor (Doctor\_ID, Doctor\_Name, Phone, Address, Birthday, Experiences Years, Department)

Patient (Patient\_ID, Patient\_Name, Phone, Address, Birthday, Gender, Allergies History, Doctor\_ID, Illness\_ID)

Illness (Illness\_ID, Illness\_Name, Department, Symptoms, Emergency Level)

**Doctor**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Doctor\_ID | Doctor\_Name | Phone | Address | Birthday | Experiences Years | Department |

P.K

**Patient**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Patient\_ID | Patient\_Name | Phone | Address | Birthday | Gender | Allergies History | Doctor\_ID | Illness\_ID |

P.K F.K F.K

**Illness**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Illness\_ID | Illness\_Name | Department | Symptoms | Emergency Level |

P.K

Decompositions

Version 2(d2.ddl)

Relations:

Doctor (Doctor\_ID, Doctor\_Name, Phone, Address, Birthday, Experiences Years, Department)

Functional Dependency: Doctor\_Name, Phone -- > Doctor\_ID, Department, Experiences Years

Doctor\_ID -- > Doctor\_Name, Address, Phone, Birthday

Decompositions:

R1: Doctor Contact Info (Doctor\_Name, Phone, Address, Department)

R2: Doctor Personal Info (Doctor\_ID, Doctor\_Name, Phone, Department, Birthday)

Version 3(d3.ddl)

Relations:

Patient (Patient\_ID, Patient\_Name, Phone, Address, Birthday, Gender, Allergies History, Doctor\_ID, Illness\_ID)

Functional Dependency: Patient\_Name, Phone -- > Patient\_ID, Address, Birthday, Allergies History,

Patient\_ID -- > Patient\_Name, Doctor\_ID, Illness\_ID, Phone, Gender

Decompositions:

R1: Patient Contact Info (Patient\_Name, Phone, Address, Patient\_ID)

R2: Patient Personal Info (Patient\_Name, Phone, Gender, Birthday, Allergies History, Doctor\_ID, Illness\_ID)

Additional work (We don’t need that, actually)

Relations:

Illness (Illness\_ID, Illness\_Name, Department, Symptoms, Emergency Level)

Functional Dependency: Illness\_Name -- > Illness\_ID, Department, Symptoms

Illness\_ID -- > Illness\_Name, Emergency Level

Decompositions:

R1: Illness Describe (Illness\_Name, Illness\_ID, Symptoms)

R2: Illness Emergency Level (Illness\_Name, Emergency Level, Department)

3. Give the two decompositions and prove that they are lossless join and functional preserving.

References: http://www.geeksforgeeks.org/lossless-join-and-dependency-preserving-decomposition/

D2.ddl

**Relation:**

Doctor (Doctor\_ID, Doctor\_Name, Phone, Address, Birthday, Experiences Years, Department)

Functional Dependency: Doctor\_Name, Phone -- > Doctor\_ID, Department, Experiences Years

Doctor\_ID -- > Doctor\_Name, Address, Phone, Birthday

Decompositions:

R1: Doctor Contact Info (Doctor\_Name, Phone, Address, Department)

R2: Doctor Personal Info (Doctor\_ID, Doctor\_Name, Phone, Department, Birthday)

**Prove:**

First condition holds true as Att(R1) U Att(R2) = (Doctor\_Name, Phone, Address, Department) U (Doctor\_ID, Doctor\_Name, Phone, Department, Birthday) = (Doctor\_ID, Doctor\_Name, Phone, Address, Birthday, Experiences Years, Department) = Att(R).

Second condition holds true as Att(R1) ∩ Att(R2) = (Doctor\_Name, Phone, Address, Department) ∩ (Doctor\_ID, Doctor\_Name, Phone, Department, Birthday) = (Doctor\_Name, Phone) ≠ Φ

Third condition holds true as Att(R1) ∩ Att(R2) = (Doctor\_Name, Phone, Department) is a key of R1 (Doctor\_Name, Phone, Address, Department) because FD’s is given.

Furthermore, all dependencies of R either can be a part of R1 or R2 or must be derivable from combination of FD’s of R1 and R2.

d3.ddl

**Relations:**

Patient (Patient\_ID, Patient\_Name, Phone, Address, Birthday, Gender, Allergies History, Doctor\_ID, Illness\_ID)

Functional Dependency: Patient\_Name, Phone -- > Patient\_ID, Address, Birthday, Allergies History,

Patient\_ID -- > Patient\_Name, Doctor\_ID, Illness\_ID, Phone, Gender

Decompositions:

R1: Patient Contact Info (Patient\_Name, Phone, Address, Gender)

R2: Patient Personal Info (Patient\_Name, Phone, Patient\_ID, Birthday, Allergies History, Doctor\_ID, Illness\_ID)

**Prove:**

First condition holds true as Att(R1) U Att(R2) = (Patient\_Name, Phone, Address, Gendder) U (Patient\_Name, Phone, Patient\_ID, Birthday, Allergies History, Doctor\_ID, Illness\_ID) = (Patient\_ID, Patient\_Name, Phone, Address, Birthday, Gender, Allergies History, Doctor\_ID, Illness\_ID) = Att(R).

Second condition holds true as Att(R1) ∩ Att(R2) = (Patient\_Name, Phone, Address,Gender) ∩ (Patient\_Name, Phone, Patient\_ID, Birthday, Allergies History, Doctor\_ID, Illness\_ID) ≠ Φ

Third condition holds true as Att(R1) ∩ Att(R2) = (Patient\_Name, Phone) is a key of R1(Patient\_Name, Phone, Address, Patient\_ID) FD’s is given.

Furthermore, all dependencies of R either can be a part of R1 or R2 or must be derivable from combination of FD’s of R1 and R2.

4. Give the SQL DDLs for creating the three versions of the database.

5. Fill the first database with data (at least 100 rows in each table).

6. Give the SQL queries to copy the data from the first version into its 2 decompositions.

7. Propose in English 3 queries that require at least 2 table joins each and such that all tables are involved in at least 2 queries.

1. Give Doctor\_Name = “Taren Batarse”, find all his patients Patient\_ID and Patient\_Name.

(Relate to Doctor, Patient table)

2. Give Patient\_Name is “Erika Heuberger” and Phone is” 2449717107”, find Doctor\_Name, and her symptom.

(Relate to Doctor, Illness, Patient table)

3. Give Illness\_ID = ”1”, find patient Patient\_Name has that illness and their doctor’s name.

(Relate to Doctor, Illness, Patient table)

8. Propose SQL implementations of the 3 queries on all three versions of the database.

9. Test the time in ns for executing the 3 queries on each database, by running each of them 1000 times.

Submit the databases populated with data, a file for each ddl (d.ddl, d1.ddl, d2.ddl), and a

report with a separate section for which of the above points.