

Database Management System Assignment

Problem statement - Detection of fake profiles.

Entities :

(i) GOVERNMENT DATA :

Attributes :

- C_ID (Citizen Identification ID) -> *primary key*
- C_Name (Citizen Name)
- P_Num (Citizen Phone Number) -> *multivalued attribute*

(ii) PROFILES :

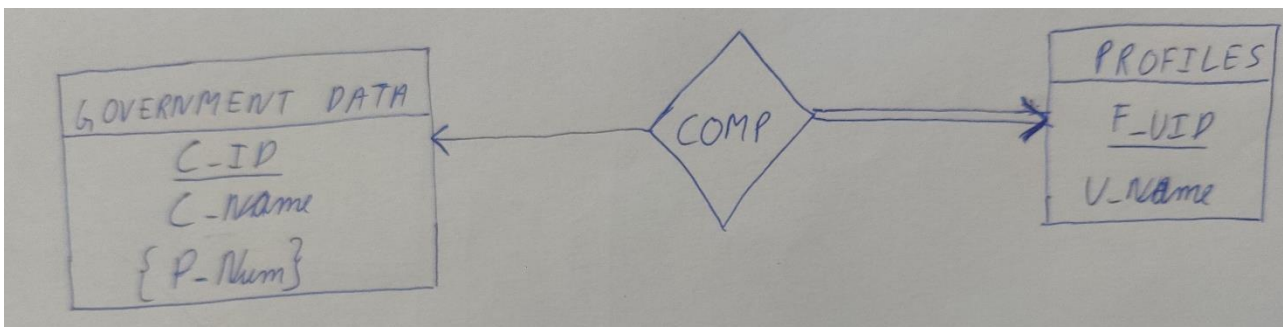
Attributes :

- F_UID (Fixed User ID)-> *primary key*
- U_Name (Profile UserName)

Relationships :

(i) COMP: One to one relationship between GOVERNMENT DATA and PROFILES, with total participation from PROFILES.

1. E-R diagram :



2. E-R diagram to relational schema :

(i) GOVERNMENT DATA:

ATTRIBUTES	DATA_TYPE	CONSTRAINTS
C_ID	NUMBER(15)	Primary Key
C_Name	VARCHAR(30)	NOT NULL
P_Num	NUMBER(10)	

(ii) PROFILES:

ATTRIBUTES	DATA_TYPE	CONSTRAINTS
F_UID	NUMBER(15)	Primary Key
U_Name	VARCHAR(30)	NOT NULL

(iii) COMP:

ATTRIBUTES	DATA_TYPE	CONSTRAINTS
C_ID	NUMBER(15)	Primary Key, Foreign Key
F_UID	NUMBER(15)	Primary Key, Foreign Key

3. DDL Commands :

(i)CREATE TABLE GOVERNMENT DATA (
C_ID NUMBER (15) PRIMARY KEY ,
C_Name VARCHAR(30) NOT NULL);

(ii)CREATE TABLE PROFILES (
F_UID NUMBER(15) PRIMARY KEY ,
U_Name VARCHAR(20) NOT NULL);

(iii)CREATE TABLE COMP(
C_ID NUMBER(15) references GOVERNMENT DATA (C_Id),
F_UID NUMBER(15) references PROFILES (C_Id)
PRIMARY KEY(P_ID, C_ID));

4. Queries to solve the problem :

(i) To see the citizen details from social profile username:

Select C_Name, P_Num
From GOVERNMENT DATA A and COMP B on A.C_ID = B.C_ID
where U_Name = x;

Problem statement - Tracking and dispensing intravenous fluids.

IV fluids may need to be given urgently to restore circulation to vital organs following loss of intravascular volume due to bleeding, plasma loss, or excessive external fluid and electrolyte loss, usually from the gastrointestinal (GI) tract, or severe internal losses. In ICU and wards, patients on IV are required to be monitored constantly for the rate of flow of IV and completion of IV dispensing. Improper attachments may lead to bleeding in some cases. It is required to Design an effective, low cost device to monitor problems related to dispensing of IV.

Entities :

(i) PATIENT :

Attributes :

- P_ID (Patient ID)-> *primary key*
- P_Name (Patient Name)

(ii) CONDITION :

Attributes :

- C_Id (Condition's ID) -> *primary key*
- C_Name (Condition's Name)

(iii) IV (*weak entity set*) :

Attributes :

- Min (Minimum Value)
- Low_Problem (Problem on low IV supply)
- Max (Maximum Value)
- High_Problem (Problem on high IV supply)

Relationships :

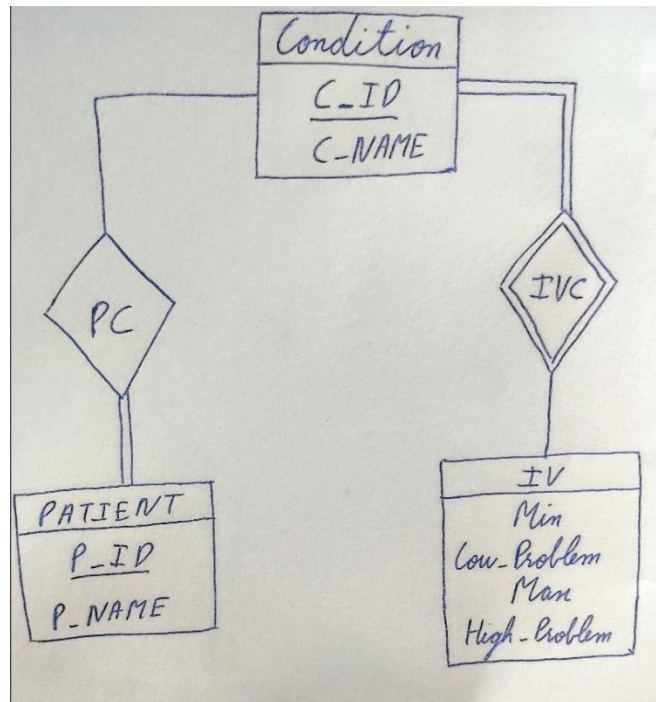
(i) PC:

Many to many relationship between PATIENT and CONDITION with total participation from PATIENT entity.

(ii) IVC:

Many to many relationship between weak entity set IV and CONDITION with total participation from CONDITION entity.

1. E-R diagram :



2. E-R diagram to relational schema :

(i) PATIENT:

ATTRIBUTES	DATA_TYPE	CONSTRAINTS
P_ID	NUMBER(15)	Primary Key
P_Name	VARCHAR(30)	NOT NULL

(ii) CONDITION:

ATTRIBUTES	DATA_TYPE	CONSTRAINTS
C_ID	NUMBER(15)	Primary Key
C_Name	VARCHAR(100)	NOT NULL

(iii) PC:

ATTRIBUTES	DATA_TYPE	CONSTRAINTS
P_ID	NUMBER(15)	Primary Key, Foreign Key
C_ID	NUMBER(15)	Primary Key, Foreign Key

(iv) IVC:

ATTRIBUTES	DATA_TYPE	CONSTRAINTS
C_ID	NUMBER(15)	Primary Key, Foreign Key
Min	NUMBER(15,2)	
Low_Problem	VARCHAR(100)	
Max	NUMBER(15,2)	
High_Problem	VARCHAR(100)	

3. DDL Commands :

(i) CREATE TABLE PATIENT(
P_ID NUMBER(15) PRIMARY KEY ,
P_Name VARCHAR(30) NOT NULL);

(ii) CREATE TABLE CONDITION(
C_ID NUMBER(15) PRIMARY KEY ,
C_Name VARCHAR(100) NOT NULL);

(iii)CREATE TABLE PC(
P_ID NUMBER(15) references PATIENT(P_ID),
C_ID NUMBER(15) references CONDITION(C_ID),
PRIMARY KEY(P_ID, C_ID));

(iv)CREATE TABLE IVC(
C_ID NUMBER(15) references CONDITION(C_Id),
Min NUMBER(15,2),
Low_Problem VARCHAR(100),
Max NUMBER(15,2),
High_Problem VARCHAR(100),
PRIMARY KEY(C_ID));

4. Queries to solve the problem :

(i) To show minimum and maximum IV requirement for a condition C_Name=x

```
SELECT A.C_Name, B.Min, B.Max, from IVC B  
INNER JOIN CONDITION A on A.C_ID = B.C_ID
```

where A.C_Name = x;

(ii) To show problems if not injected in the given range for a condition C_Name=x

```
SELECT A.C_Name, B.Low_Problem, B.High_Problem, from IVC B
INNER JOIN CONDITION A on A.C_ID = B.C_ID
where A.C_Name = x;
```
