



Electronic Health Records

DATABASE ASSIGNMENT

CS4D2A INFORMATION MANAGEMENT II

ALEXANDRU SULEA
D STREAM
#12315152
13 DECEMBER 2016

Contents

List of Tables	1
1 Brief Description	1
2 Entity Relationship Diagram and a Relational Schema	2
3 Functional dependency diagram	3
4 Explanation of semantic constraints	4
5 Examples of:	5
5.1 View creation	5
5.2 Relational Select	5
5.3 Update operations	5
5.4 Triggers	5
6 Additional Features	5
7 APPENDIX (Listing of Database Content):	5
7.1 Listing of data definition commands (create) used to create your database - including integrity constraints, e.g. table checks etc.	5
7.2 Listing of database population commands (Insert) used to populate your database	8

List of Tables

1 Brief Description

The following database project was inspired by the idea of the growing trend of E health in Ireland. The following database is modeled along the lines of a medical clinic such as a GP or dentist office.

While many new offices have already been integrated with cloud systems, many rural and regional offices may have a minimal amount of database applications due to poor Internet availability or simply a staff unwillingness to adapt new technologies.

The following entities were chosen. Branch: The branch office which corresponds to a physical building. The branch office has a many-to-many relationship to the equipment entity. This is due to the fact that many types of equipment (heart monitor, thermometer etc.) can be in many buildings. Should a disaster take place or an area become more busy than others, equipment should be able to be moved around between locations. The details associated with the branch entity are the address of the branch which can be multi varied and split up into Street, neighborhood, city.

The link between branch and equipment is equipping which uses the primary keys of equipment and branch, the equipping has a date associated to it so as to be able to quickly know on what date, what components were supplied by whom. Helping with expenses and stock documentation.

Equipment entity should be able to be supplied by multiple vendors to multiple branches of the clinic.

The equipment entity simply has the equipment type and supplier id along with a unique id associated with each individual medical product. A serial code of such. The equipment entity is linked to the supplier by the supply verb. This supply verb will also have a date associated with it , thus denoting when the order was placed for the equipment.

The supplier entity will have the suppliers available for equipping the branch. These suppliers can have their addresses, the telephone numbers, emails and as their primary key we can use the company registration number which is publicly available for all globally registered limited liability companies.

On the other side of the graph there is the relationship between staff and branches. Normally there would be multiple staff to one branch, but for the interest of work flexibility and the possibility of having staff transfer around to different places the relationship has to be many to many.

The table between branch and staff will have the date at which the certain staff was assigned to a branch, thus the movement of staff to different branches can be tracked by time and or location.

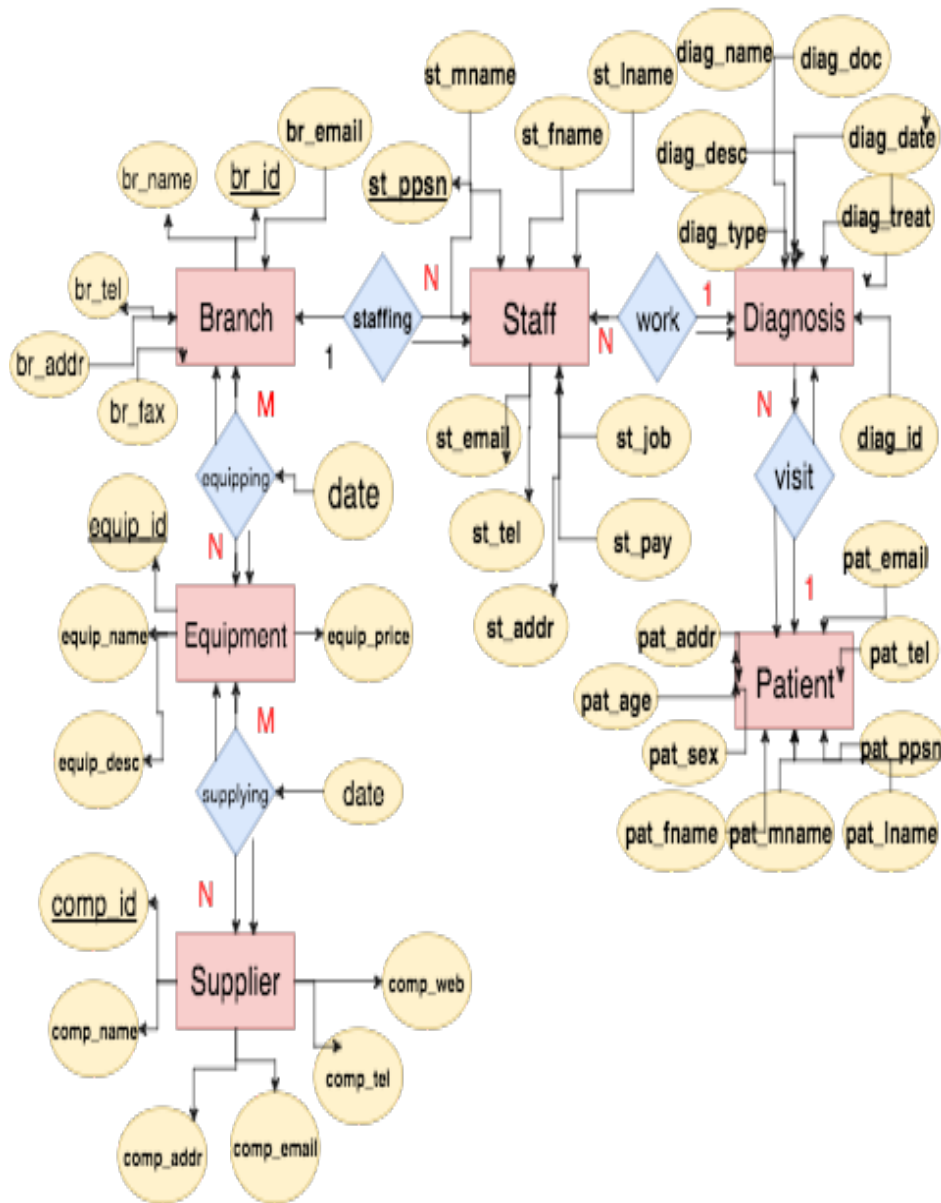
The staff entity will have the staff ppsn number as its primary key. The first second and last name of the staff member. Their qualification as a practitioner. Their addresses, telephone numbers and emails. This will also have their salary information.

The other entity is the the diagnosis entity. The diagnosis entity corresponds to the diagnosis given to each patient . As its primary key, it will use the patient ppsn numbers as a diagnosis is only given to a patient at a time by 1 staff member at a time. Again the diagnosis entity can be administered by only one staff member but staff members can administer many diagnosis .

Different diagnosis, and many different diagnoses can be made to patients. But only one diagnosis can be made per patient at any one time.

The last entity in the database is the patient's entity. This entity will store all the relevant patient information. The primary key will be the patient ppsn , other details will be the patient first,second and middle names. The patient's home address. The patient's phone and email details. The patient's age and the clinic that they are regionally associated to. The patient age and gender will denote to other doctors why a certain treatment was chosen above others.

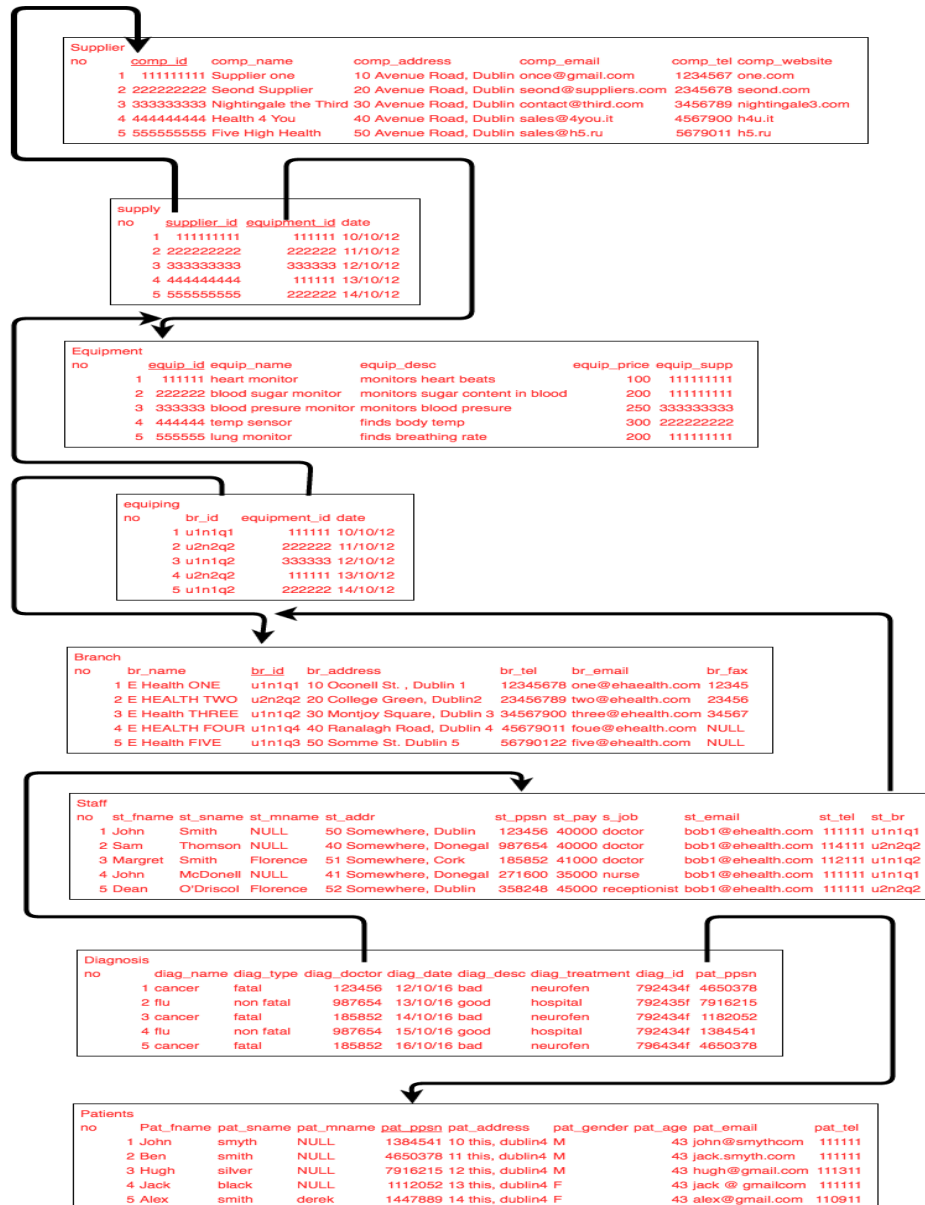
2 Entity Relationship Diagram and a Relational Schema



(a) The image shows the relationships between entities and attributes

Figure 1: The figure shows Relationship Diagram

3 Functional dependency diagram



(a) The image shows the relationships between entities and attributes

Figure 2: The figure shows Functional Diagram

4 Explanation of semantic constraints

As with many things there are tables with m to n and n to 1 dependencies as well as well:

The supplier entity will have 6 attributes attached to it, those being *company_reg*, which will act as the primary key, *company_name*, *comp_address*, *comp_email*, *comp_tel* and *comp_wbsite*. The supplier entity will have an n to m relationship with equipment. The relationship will also be that of total participation as a supplier needs to supply equipment to exist.

The connection between supplier and equipment entities is managed by the supply table. Due to the relationship between the two being m to n a middle table is needed which will hold the date at which the supplier was added to the table and also the supplier and equipment primary keys.

The equipment entity will have 5 attributes associated with it. These being *equip_id* which will act as primary key as all medical equipment come with their own uniquely identifiable id. *Equip_name*, *equip_adesc*, *equip_price*, *equip_supplier*. Equipment has an m to n relationship with branch as many branches can be supplied simultaneously by many suppliers with lots of equipment. The relationship with supplier is also as lots of equipment can be supplied by lots of suppliers simultaneously. The relationship between equipment and supplier is that of partial participation as equipment could just sit in storage or not be assigned a department while it is in transit, branches should not need equipment to be operational.

Due to the relationship between branch and equipment also being m to n there is a middle table to help manage the data. This table will store or point to the equipment and branch primary keys as well as the date at which equipment was stored in a branch.

The branch entity represents the physical medical branches, the brick and mortar buildings. The entity has 6 entities. The *branch_name*, *branch_address*, *branch_tel*, *branch_fax*, *branch_email* and as a primary key, the *branch_id*.

The branch has a 1 to n relationship with the staff entity. That is due to the fact that many staff can work in a branch, but many staff cannot be in all the branches simultaneously. Or even one staff member cannot be in all the branches simultaneously.

The relationship between branch and staff is also one of partial participation as a branch should not need staff to exist. Especially if a branch would be closed down temporarily and staff would have to be reallocated to a new branch.

The staff entity has 10 attributes associated with it. It follows the standard protocol of staff ppsn as the primary key, with the following attributes being staff first, second last names. The staff address phone no and email being part of the contact information. The last 3 attributes however outline the staff role which gives different permissions depending on whether they are a doctor or secretary. A secretary can make appointments but only a doctor can make diagnoses. The staff payment denotes how much each staff member is paid and the *staff_bbranch* points to the *branch_id* and denotes where each staff member works. Staff has a 1 to n relationship with diagnosis. That is because only one doctor should be required to make a diagnosis but doctors can make many diagnoses. The relationship is also of partial participation as staff do not need to diagnose to exist but a diagnosis needs a staff member to be created.

The diagnosis entity has 7 attributes associated with it, as a primary key it will hold a unique *diagnosis_id*. The rest of the attributes are the *diagnosis_name*, *diagnosis_type*, *diagnosis_adate*, *diagnosis_adetail* and chosen *diagnosis_treatment*. The *diagnosis_adoctor* attribute will point to the *staff_id* which should be that of a doctor which was responsible for making the diagnosis. Diagnosis has a 1 to n relationship with patient table as a patient can have many diagnoses at one time, but a diagnoses can not have more than one patient.

Diagnoses are patient specific. There is also a total participation with the patient entity as a patient cannot be a patient without a diagnosis and a diagnosis cannot exist without a patient.

The last entity is the patient entity, this entity will have 9 attributes. The patient PPSN being the primary key, then followed by first, second and middle names. The contact info will also be recorded in the form of address, email and telephone number. The last 2 entities will be of the patient age and the patient gender which are required many times for insight into why a specific diagnosis was chosen above other ones.

5 Examples of:

5.1 View creation

The doctors should be able to see all details, therefore for view creation they have super user constraints. The secretaries however should not be able to see the patient details such as age and gender, only their name and contact information, so for any staff which has a role of secretary, the view will be restricted to READ privilege for *pt_fname*, *pt_lname*, *pt_email*, *pat_addr* and *pt_tel*. this would generally be done by GRANT READ to secretary.

5.2 Relational Select

Relational privileges would mainly be between doctors and staff to patients and doctors to equipment. A system admin would first need to be instantiated. The sys adm would then grant select relational privileges to doctors and secretaries separately. Thus the sys adm might grant GRANT INSERT, SELECT, DELETE, MODIFY ON equipment, supplier TO doctor

and then
GRANT INSERT, SELECT, ON diagnosis TO doctor

and for secretary
GRANT INSERT, SELECT, MODIFY ON patients TO doctor

5.3 Update operations

The staff should be able to have modification privilege over *pat_tel* and *pat_email* and *pat_addr* so that if a patient wishes to update their details they can do so at the secretary office. Or the patient can do it in the doctors office. The doctor will also not have modification privileges over diagnosis, only creation ones. This will keep doctors from tampering with already recorded patient diagnosis info. One useful case of this would be if a doctor would wish to write their own prescription and then delete it from the system.

5.4 Triggers

6 Additional Features

Additional features would include preventing staff from writing their own subscriptions. This will keep a record of all transactions and activity in the organization without being tainted by corruption. Both legal and technical.

7 APPENDIX (Listing of Database Content):

7.1 Listing of data definition commands (create) used to create your database - including integrity constraints, e.g. table checks etc.

```
CREATE database EHealth; USE EHealth;
```

```
CREATE TABLE supplier(  
  comp_reg_no    NUMBER(9) PRIMARY KEY NOT NULL,  
  comp_name      VARCHAR2(25) NOT NULL,  
  comp_addr      VARCHAR2(25) NOT NULL,  
  comp_email     VARCHAR2(25),  
  comp_tel       NUMBER(5) NOT NULL,  
  comp_web       VARCHAR2(25));
```

```
CREATE TABLE equipment(  
  equip_id       NUMBER(9) PRIMARY KEY NOT NULL,  
  equip_name     VARCHAR2(25) NOT NULL,  
  equip_desc     VARCHAR2(25) NOT NULL,  
  equip_price    NUMBER(9) NOT NULL,);
```

```
CREATE TABLE supply(  
  equip_id       NUMBER(9) PRIMARY KEY NOT NULL,  
  comp_reg_no    NUMBER(9) PRIMARY KEY NOT NULL,  
  supply_date    NUMBER(8) NOT NULL,);
```

```
CREATE TABLE branch(  
  br_id         NUMBER(9) PRIMARY KEY NOT NULL,  
  br_name       VARCHAR2(25) NOT NULL,  
  br_addr       VARCHAR2(25) NOT NULL,  
  br_email      VARCHAR2(25),  
  br_tel        NUMBER(5) NOT NULL,  
  br_fax        NUMBER(11) NOT NULL);
```

```
CREATE TABLE supply(  
  equip_id       NUMBER(9) PRIMARY KEY NOT NULL,  
  br_id          NUMBER(9) PRIMARY KEY,  
  equip_date     NUMBER(8) NOT NULL);
```

```
CREATE TABLE staff(  
  st_ppsn       VARCHAR(9) PRIMARY KEY NOT NULL,  
  st_fname      VARCHAR2(25) NOT NULL,  
  st_lname      VARCHAR2(25) NOT NULL,  
  st_mname      VARCHAR2(25) ,  
  st_addr       VARCHAR2(25) NOT NULL,  
  st_email      VARCHAR2(25),  
  st_tel        NUMBER(11) NOT NULL,  
  st_pay        NUMBER(11) NOT NULL,  
  st_role       VARCHAR2(25),  
  st_br         VARCHAR(25)  
);
```

```
CREATE TABLE diagnosis(  
  diag_id       NUMBER(9) PRIMARY KEY NOT NULL,  
  diag_name     VARCHAR2(25) NOT NULL,
```



```
diag_desc      VARCHAR2(255) NOT NULL,  
diag_treatment VARCHAR2(255) NOT NULL,  
diag_type      VARCHAR2(25) NOT NULL,  
diag_doctor    NUMBER(11) NOT NULL,  
diag_date      NUMBER(8) NOT NULL  
pat_ppsn       VARCHAR(9) NOT NULL);
```

```
CREATE TABLE patient(  
pat_ppsn       VARCHAR(9) PRIMARY KEY NOT NULL,  
pat_fname      VARCHAR2(25) NOT NULL,  
pat_lname      VARCHAR2(25) NOT NULL,  
pat_mname      VARCHAR2(25) ,  
pat_addr       VARCHAR2(25) ,  
pat_email      VARCHAR2(25),  
pat_tel        NUMBER(11) NOT NULL,  
pat_age        NUMBER(3) NOT NULL  
pat_gender     VARCHAR2(3) NOT NULL,  
);
```

7.2 Listing of database population commands (Insert) used to populate your database

```
INSERT INTO supplier(COMP_REG_NO, COMP_NAME,  
COMP_ADDR, COMP_EMAIL,  
COMP_TEL, COMP_WEB) VALUES ('111111111',  
'Supplier ONE Health',  
'Main Street One', 'one@one.com',  
123456789, 'one.com');
```

```
INSERT INTO equipment(EQUIP_ID, EQUIP_NAME,  
EQUIP_DESC,  
EQUIP_PRICE, EQUIP_SUPP) VALUES (111111111,  
'Blood pressure monitor', 'a monotor for blood pressure',  
100000, '111111111');
```

```
INSERT INTO staff(ST_PPSN,ST_FNAME, ST_LNAME,ST_MNAME  
ST_ADDR, ST_ST_EMAIL,ST_TEL,ST_JOB, ST_PAY)  
VALUES (11141511C,  
'JOHN','Smyth',NULL,  
'Main Street One', 'john@gmail.com',097575321  
doctor, '100000.00');
```

```
INSERT INTO patient(PAT_PPSN,PAT_FNAME, PAT_LNAME,PAT_MNAME  
PAT_ADDR, PAT_EMAIL,PAT_TEL,PAT_SEX, PAT_AGE)  
VALUES (11146711A,  
'Derek','Allus',NULL,  
' 10 Leeson street Street, Dublin4', 'derek@gmail.com',0877569078,'MALE', '45');
```

```
INSERT INTO diagnosis(DIAG_NAME,DIAG_DESC,DIAG_TYPE,DIAG_DOCTOR,  
DIAG_DATE, DIAG_TREATMENT,DIAG_ID,DIAG_PAT)  
VALUES (11146711A,  
'CANCER','skin cancer','non fatal', ' 1104578C', ' 20/02/2016','CHEMO THERAMY','D573829',  
'11146711A');
```

```
INSERT INTO branch(BR_ID,BR_NAME  
BR_ADDR, BR_EMAIL,BR_TEL,BR_FAX)  
VALUES (BR101,  
'E health one',  
'Main Street One, Dublin 16', 'one@ehealthone.com',097575321  
, 4585432);
```