

# Assignment 4

CS- 548

Due Date: 10/6/2017

Submitted By-

Rajat kinkhabwala

Stevens ID-10414081

Canvas Id- rkinkhab

**Provide the data model for a domain-driven design for a clinical information system. You will define a data model using JPA-annotated Java code, and use Eclipse tools to configure database tables for storing the data. In the next assignment, you will add domain-specific logic to the Java classes.**

## **Documentation-**

1. The domain of the clinical entities consists for patient entities, for the patients that are treated at that clinic and the provider entities for the health providers that work at the clinic.  
The class name for the clinic is ClinicGateway. The Factory class for this is called ClinicGatewayFactory which implements IClinicGateway Interface.
2. In Patient Entity Class, the @Entity Notation with several others JPA notations were used. There is also a primary key annotation for Id which is auto-generated by the database.  
List of all treatments are also initialized and are set with One to Many relationships between Patient and Treatment Class. Each patient entity also has a patient identifier, that uniquely identifies them in the health system, a patient name and date of birth.
3. The set of possible treatments, and therefore the possible forms of the treatment entities, is open-ended. Every treatment entity includes a provider administering the treatment, and a diagnosis of the condition for which the treatment is prescribed. There is also a class which specifies the value of the fields and column of the table for various types of treatment types. Several types of treatment types are:-
  - (i) Drug Treatment Type
  - (ii) Surgery Type
  - (iii) Radiology Type
4. Date fields in Patient (for Date of Birth), in Surgery (for Date of Surgery) and in Radiology (List of Dates of Radio Treatment) are specified with @Temporal Annotation. Although in case of Radiology, list of dates is combined using @ElementCollection annotation to handle collection of radiology dates. Further ahead, all the treatment types are the subclasses of the Treatment class file which follows the @inheritance strategy.

5. A health provider has a Provider Identifier (ProviderId), a name and an indication of their specialization (surgery, radiology, oncology, etc.). A provider will be related to patients by the treatments that she performs on them. There is a one to many relationships between providers and treatments. Each provider is related to each of the treatments for which they are responsible, and therefore they are indirectly related to the patient being treated.
6. In both Patient and Provider entities, the list of treatments should be initialized to the empty list. There is also a fix for a problem which happens if a patient is deleted or a provider is deleted, all the treatment records associated with them are also removed. This is done by setting cascading option to 'REMOVE'.

The Several SQL SCRIPTS that were generated using the JPA CODE are below-:

- (i) Script to create the Patient Table  

```
CREATE TABLE patient (  
    id INT8 NOT NULL,  
    birthdate DATE, name  
    VARCHAR(255),  
    patientid INT8  
);
```
- (ii) Script to create the Provider Table  

```
CREATE TABLE provider (  
    id INT8 NOT NULL,  
    providerid INT8, name  
    VARCHAR(255),  
    specialization VARCHAR(255)  
);
```
- (iii) Script to create Treatment Table  

```
CREATE TABLE treatment (  
    id INT8 NOT NULL, ttype  
    VARCHAR(31), diagnosis  
    VARCHAR(255),  
    patient_fk INT8,  
    provider_fk INT8  
);
```
- (iv) Script to create a Drug Treatment Record  

```
CREATE TABLE drugtreatment (  
    id INT8 NOT NULL,  
    dosage FLOAT8, drug  
    VARCHAR(255) );
```
- (v) Script to create a Surgery Record  

```
CREATE TABLE surgery (  
    id INT8 NOT  
    NULL, date DATE  
);
```

(vi) Script to create a Radiology Record

```
CREATE TABLE radiology (  
    id INT8 NOT NULL  
);
```

(vii) Table Radiology Dates for collection of dates in radiology treatment

```
CREATE TABLE radiologydates (  
    radiology_id INT8,  
    dates DATE  
);
```

```
CREATE UNIQUE INDEX provider_pkey ON provider (id ASC);
```

```
CREATE UNIQUE INDEX patient_pkey ON patient (id ASC);
```

```
CREATE UNIQUE INDEX drugtreatment_pkey ON drugtreatment (id ASC);
```

```
CREATE UNIQUE INDEX radiology_pkey ON radiology (id ASC);
```

```
CREATE UNIQUE INDEX treatment_pkey ON treatment (id ASC);
```

```
CREATE UNIQUE INDEX surgery_pkey ON surgery (id ASC);
```

```
ALTER TABLE drugtreatment ADD CONSTRAINT drugtreatment_pkey PRIMARY KEY (id);
```

```
ALTER TABLE provider ADD CONSTRAINT provider_pkey PRIMARY KEY (id);
```

```
ALTER TABLE radiology ADD CONSTRAINT radiology_pkey PRIMARY KEY (id);
```

```
ALTER TABLE treatment ADD CONSTRAINT treatment_pkey PRIMARY KEY (id);
```

```
ALTER TABLE surgery ADD CONSTRAINT surgery_pkey PRIMARY KEY (id);
```

```
ALTER TABLE patient ADD CONSTRAINT patient_pkey PRIMARY KEY (id);
```

```
ALTER TABLE surgery ADD CONSTRAINT fk_surgery_id FOREIGN KEY (id)  
REFERENCES treatment (id);
```

```
ALTER TABLE radiologydates ADD CONSTRAINT  
fk_radiologydates_radiology_id FOREIGN KEY (radiology_id)  
REFERENCES treatment (id);
```

```
ALTER TABLE drugtreatment ADD CONSTRAINT fk_drugtreatment_id FOREIGN KEY (id)  
REFERENCES treatment (id);
```

```
ALTER TABLE treatment ADD CONSTRAINT fk_treatment_provider_fk FOREIGN  
KEY (provider_fk)  
REFERENCES provider (id);
```

```
ALTER TABLE radiology ADD CONSTRAINT fk_radiology_id FOREIGN KEY (id)  
REFERENCES treatment (id);
```

```
ALTER TABLE treatment ADD CONSTRAINT fk_treatment_patient_fk FOREIGN KEY  
(patient_fk)  
REFERENCES patient (id);
```

The above script commands set the primary key(s)/secondary key(s) indexes and constraints wherever required.

**After Unzipping the archive, In the root folder, there is a Project Folder named “ClinicalDomain” which is the project exported from eclipse, a video file (.mp4) which shows the generation of the tables and this PDF named “README.pdf”**