

## Question 1

- a) Impedance Mismatch is a set of problems regarding the differences between the model of the programming language and the model of the database. An example would be:- The attribute data type used in the programming language may not be the same as the data type for the same attribute in the database model.
- b)
  - i.) The main purpose of this program is to connect to the database and then it for all the records in the 'MEDICINE' table. It will then loop over each record and print out the name and concentration of each one. Finally, it will close the connection to the database.
  - ii.) The role of the driver manager class instantiated in line 5, is to help manage the set of JDBC drivers that are available to use.
  - iii.) The 'Connection' object is used to show the status of a connection to a database and whether it is available to access. The 'Statement' object is used to build and execute a SQL query as well as show the status of the query and return the result. The 'ResultSet' object is used to capture the returned results from the query and allow for the reading/interpretation of the data.

## Question 2

- a) Inheritance in the EER model is used to show how many entity types may be related and or could possibly be generalised into one entity type, whereas inheritance in the OO model is used to define a template like object that can then be used to create other objects of similar type, this can also happen more than once or can be selective such that the child object does not have to inherit all the parent properties.
- b) False. In the OODB each object is provided a system generated Object Identifier (OID), it is this identifier that is used to uniquely identify an object due to it never being used for more than one object as well as it being immutable.
- c) Persistent objects exist permanently in the database, while Transient objects exist only during the execution of the application.
- d) Structured data is stored in a very rigid format and thus comes with many constraints such that the data can be easily recognised and spilt into the appropriate types. An example would be a relational database, an excel spreadsheet or a simple table, where the data must be stored in a specific format.

Semi-Structured data is not as rigid, however still conforms to a certain amount of organisation such that the data can still be quickly analysed and interpreted. An example would be XML, where the data is still neatly organized using tags, and you can easily find what section of the data you are looking for, however the stored data has a lot more freedom than what can be stored in a table.

Unstructured data is not rigid at all, while it still conforms to some rule set defining how it must be laid out, this is mainly for a machine to interpret such that it can provide easier to understand output to the user, and not at all meant for manual interpretation. An example would be a pdf document or a picture, where there is certain headers used to describe document length, colour, size, data type, etc.
- e) Semi-structured.
- f) The use of tags in XML is used to define the tree like structure of the data for ease of interpretation as well as definitions of data, where as the use of tags in HTML is used to describe what type of element the following data will be affecting as well as where that data should be placed.

### Question 3

- a)
- b) If a patient is admitted several times, they will get multiple records inserted into the tree with lots of duplicate information.
- c) XML namespaces will solve the conflict. i.e. prefixing the tags with dc -- <dc:doctor>
- d) a
- e) name: This should stay as a xs:string.  
concentration: This should instead be a xs:decimal  
dosage: This should be a xs:decimal  
period: This should be a xs:time
- f)
- g)

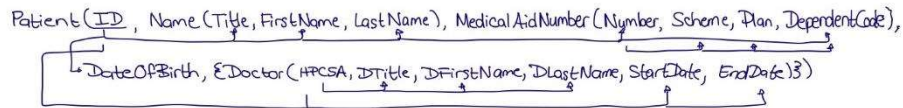
### Question 4

- a) i.) The primary key is {A, B}  
ii.) True. R is in 1NF due to there being no nested relations nor any multivalued relations.  
iii.) The highest normal form they are in is 2NF.  
iv.)  $R_4(\underline{A}, D, E)$  &  $R_5(\underline{D}, I, J)$
- b) This relation's next normal form will be 4NF, due to the fact that it meets all the requirements to be in 3NF, however it currently has some Multivalued Dependencies (MVD) that need to be decomposed.

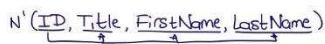
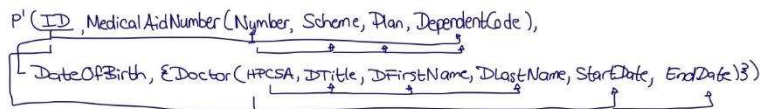
## Question 5

COS 210 Exam Opportunity 2 Dylan Kapnias (u18108467) Kapnias.

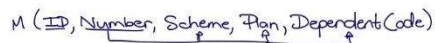
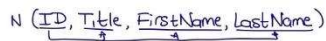
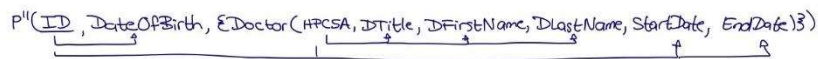
### Question 5



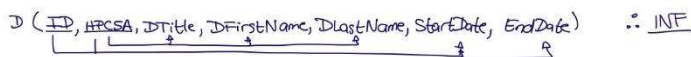
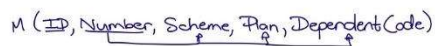
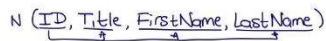
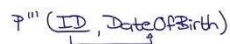
Name is nested:



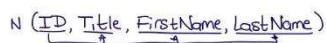
MedicalAidNumber is nested:



Doctor is nested:



Not in 2NF:



$M'(\underline{ID}, \text{Number})$

$M''(\text{Number}, \text{Scheme}, \text{Plan}, \text{Dependent Code})$

$D(\underline{ID}, \text{HPCSA}, \text{DTitle}, \text{DFirstName}, \text{DLastName}, \text{StartDate}, \text{EndDate})$

$D$  not in 2NF:

$P'''(\underline{ID}, \text{DateOfBirth})$

$N(\underline{ID}, \text{Title}, \text{FirstName}, \text{LastName})$

$M'(\underline{ID}, \text{Number})$

$M''(\text{Number}, \text{Scheme}, \text{Plan}, \text{Dependent Code})$

$D'(\underline{ID}, \text{HPCSA}, \text{StartDate}, \text{EndDate})$

$D''(\text{HPCSA}, \text{DTitle}, \text{DFirstName}, \text{DLastName})$

$\therefore$  2NF

Final Relations in 3NF

$P'''(\underline{ID}, \text{DateOfBirth})$

$N(\underline{ID}, \text{Title}, \text{FirstName}, \text{LastName})$

$M'(\underline{ID}, \text{Number})$

$M''(\text{Number}, \text{Scheme}, \text{Plan}, \text{Dependent Code})$

$D'(\underline{ID}, \text{HPCSA}, \text{StartDate}, \text{EndDate})$

$D''(\text{HPCSA}, \text{DTitle}, \text{DFirstName}, \text{DLastName})$

## Question 6

- a) Figure 1 → (f)  
Figure 2 → (b)  
Figure 3 → (e)  
Figure 4 → (a)  
Figure 5 → (d)  
Figure 6 → (c)
- b)