CSC370 - Assignment 1 - Daniel Dubichev

Project Overview, E/R Diagram & Journal.

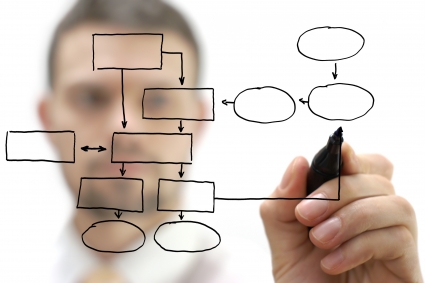


Daniel Dubichev – V00877776 – CSC370 – Assignment/Project Phase One.

***Introduction***

This document describes the process of creating a database, unequivocally one of the most sought out and demanding tasks in the IT industry today. It is split into three main parts To begin, I will post some requirement analysis from scenario 2 (my selection), analyze the requirements and elaborate on my design and decision making process for the database. Following requirements analysis, my E/R diagram will become the focus. Accordingly, I will clarify each entity, relationship and attribute as I did in my journal, and then put all entities together using draw.io. Lastly, I will photocopy my personal journal and attach it at the end of this document, preceding my conclusion. It is worthwhile to note that during my analyzation and requirements analysis of the database I take many notes and references from my journal.

Requirement Analysis & Initial Design Ideas



One of the key aspects in database design is to specify a schema. From the schema, a basis is formed from the designers perspective which can be translated and tweaked to project specifications. However, In order design a proper initial schema, there must be some requirements set out at first. This is where our assignment PDF came into the light. I haven’t done any E/R diagrams in the past, and option #2 was more interesting to me. Also because it is a viable business idea and isn’t constrained to mathematical subjects like object #1. It was important to set out a few specifications/mandatory constraints from the PDF description. The most important ones are listed below, with my own thoughts and ideas and how to implement them.

***Quote by quote requirements analysis from assignment pdf***

**Quote (Q):**

“A non-profit organization wants to develop a website which will assist their users in online learning by using freely available material on the internet. The users who want to use the website need to register themselves before accessing the material on site/make their recommendations.”

**Thoughts & Analysis (T&A):**

The first thought that came to mind is to have users register themselves somehow. However, using the keep-it-simple-stupid paradigm, I **assume** it would be best to just have an attribute to the user entity titled “registered-date” to prove the user was in deed registered and eligible to use the website. It also became clear to me that users would have multiple tasks to do on this website, thus resulting in a large E/R diagram.

**Quote (Q):**

“Develop a database for this organization’s website which assists in online learning by giving the sequence of learning material (academic or otherwise: freely downloadable books/academic articles, etc.) for a beginner to the professional in a specific subject area.”

**Thoughts & Analysis (T&A):**

My authentic impression of this quote was how to develop a database that can hold sequences of course materials. I was used to working with pre defined data sets in all my previous experience, so this was new to me. It was the keyword “sequences” that struck me, and I would have to keep a counter or iterator to ORDER and SORT each material with it’s corresponding sequence (1 for the first in the sequence, “n” for last). Also, how were the sequences generated? Randomly or by users? On the other hand, the quote “Beginner to professional in a specific subject area” did not seem too troublesome. I interpreted them as attributes. But, I began thinking of eliminating attributes and replacing them with entity sets, as with the normalization approach. Although, this translation would have to be implemented in relational (SQL) tables, not E/R diagram.

**Quote (Q):**

“Should give links to online material (available options: MOOCs, course material of top universities, textbooks used in each class, topics covered and to what depth, available homework/practice or exam material, etc.) and store the sequence in which that material should be followed.”

**Thoughts & Analysis (T&A):**

I will create a material entity that holds links to a very large selection of all the available options listed from the quote alongside attributes for the materials. Attributes would include links to the online resources, title of the resource, difficulty, which institution it belongs too, what grade or university year it would correspond too (etcetera). Primary key would be a incremental ID valued from (0 – total # of materials present)

**Quote (Q):**

“The original material should be stored by produced date or when the course/MOOC was offered, and users should be able to query the database by institution, topic, date, MOOC/other academic course, view the material of their choice and store their preferred possible sequences (there might be a few) in which the material could be followed. The sequence to follow for learning/to gain competence could be from different courses (but valid sequences as defined by the user according to taste, difficulty, etc.)”

**Thoughts & Analysis (T&A):**

The original material should always be stored by produced date. This is odd. I don’t know what jurisdiction this would have over the data itself. Why order material by date released instead of popularity or academic rating? I guess date will have to be an attribute and we will **assume** that all the data is ordered in chronological order by date ascending/descending. Querying by the attributes like institution, date, and more will be no problem as per the previous quote’s thoughts and analysis. Storing preferred sequences for a user implies that a user will be able to curate their own sequences / customized courses and store them. Knowing this I **assume** there will be a relationship, binary or ternary (to be discussed later) between the three major entities; user, material and a new defined sequence entity. According to the last line from the quote, attributes will be present in the sequence entity.

**Quote (Q):**

Other users visiting the website should be able to view these stored sequences recommended by other learners with their comments and store their own recommended sequences. Users should be able to recommend links to original course material available on the web; which they think are suitable for a specific course, if that link is already not listed on the website

**Thoughts & Analysis (T&A):**

Immediate **assumption:** Other users means other registered accounts. Sequences are viewable by other accounts. Sequences and materials can have recommendations, comments and ratings. Users have the ability to recommend courses they are fond of, which must be added to the database and approved by staff if the data/article/material doesn’t exist yet.

***Requirement Analysis Conclusion***

After analyzing and fully indulging the requirements set out in the assignment PDF, I am comfortable with beginning to shape the E/R Diagram. While these requirements are somewhat brief, they do contain assumptions that make my database unique and are integral in the understanding of my databases functionality/design.

E/R Diagram implementation and design



**Entities & Attributes**

My strategy for implementing my E/R Diagram began with identifying key nouns in our requirement analysis, and comparing them with my thoughts, intuition and assumptions.

**Users**

Logically, when a user is registered they will need some data to accompany their account. All this data could be organized into columns in a relation, so the data is portrayed as attribute in my diagram. The data for users I chose was:

* userID: A primary key for each user to be denoted by an incrementing counter
* userName: A unique string “user name” for each user, key credential
* userPass: A non-unique sequence of characters used with userName to access the websites features
* userRegisterDate: Datetime value indicating the user is registered and able to use the website
* userEmail: A string containing a users email address. In case they forget their password

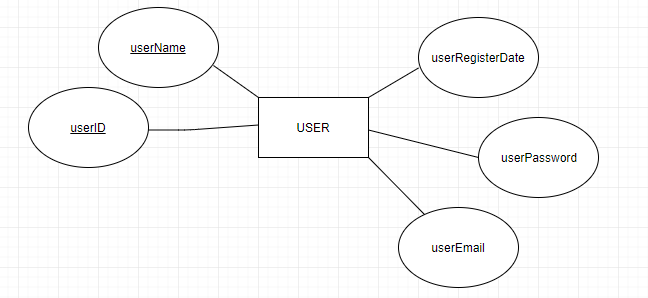


Figure 1: User Entity with no subclasses.

After establishing my basic user entity, I thought of some more logic corresponding to a user. What I came up with was administration. In the specifications, admins should be able to add to materials entity after a regular user requests to add a material that is not yet present in the database. So I split my user entity into two subclasses, with user being the superclass (Figure 2).

The Staff subclass will contain some personal data such as staffPhone and staffEmail in case of emergency contact via other staff members, while the learner subclass contains informational data about the specific user such as their favorite topic, preferred difficulty, a bio about themselves and the last time they logged on. I imagine the database would be implemented alongside a customer relationship management system, so having the “learnerLastLog” attribute would prove valuable for reconnecting with old or inactive customers.

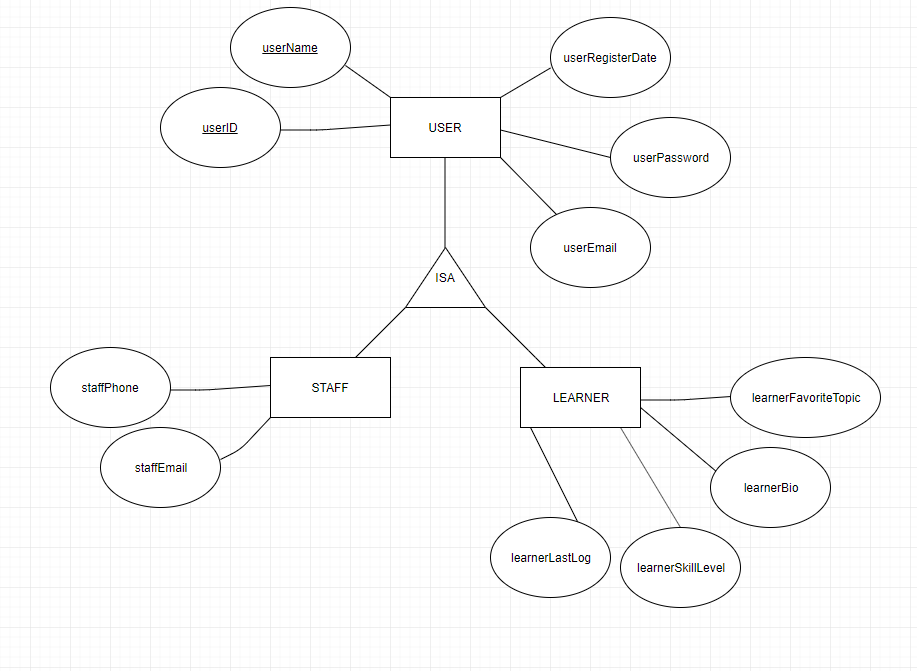


Figure 2: User superclass with subclasses

**Material**

The material entity set consists of unique rows describing a learning resource on the internet. Many attributes are specified to be included, however the PDF in assignment one applies to all of the following attributes (Figure 3). Here are the attributes and their meanings:

* materialID: Primary key for the material entity. Each row may be uniquely identified using this key.
* materialSubject: A string which displays the subject corresponding to the current row/entry. For example, a subject may range alphabetically anywhere from:

Advanced Algebra – Zoology.

* materialGrade: A string which indicates what grade the material is indicated for. Such as for grade school education the range would be between Kindergarten to Grade 12, and for post secondary education the range would be from First Year College/University to Graduate school material.
* materialType: Entry will be a string which indicates what type of material corresponds to the column. Whether it may be an “Academic Article”, “Online Video” Or “MOOC” it will be defined here.
* materialDifficulty: This will be a decimal number, and is ranged between 0-10. This field indicates how difficult the material is on a scale of one to ten.
* materialLink: The physical link/url that is stored for the resource/material.
* materialDateProduced: The date the material was published or made public to the internet. Due to project specifications in the requirement analysis, the order of this attribute will determine the way all material is stored, from oldest date to earliest date, which is the reasoning of this attribute existing.
* materialSource: What university or publishing company owns the material/resource.

There will also be a relationship between the **textbook** entity and **material** entity, which is an entity that holds textbook information. Some MOOC’s or online courses may require multiple textbooks, so instead of having textbook as an ATTRIBUTE, It would be its own entity. Textbook has some similar attributes to that of material as you can see in the figure below.

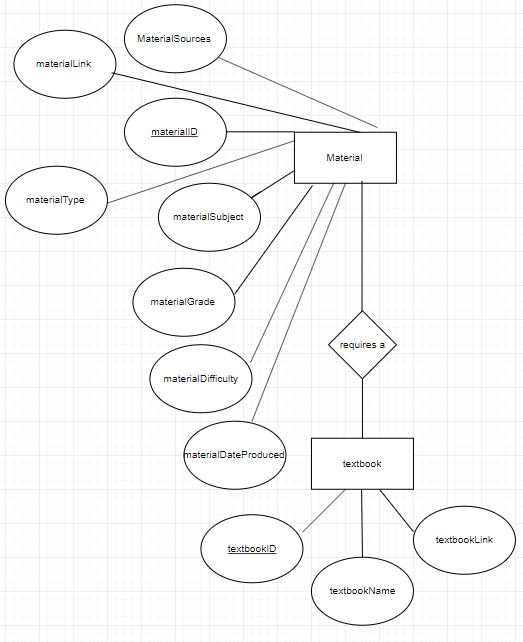


Figure 3: Material and Textbook entities. Notice relationship is many to many. Many textbooks for one material (MOOC), but also many course (intro to something) may use one universal textbook.

**Sequence**

This is the last main entity, and everything else in the E/R Diagram will mostly be relationship based, however after explaining the attributes of sequence, I will talk about the entry entity and how it is related to sequence (Illustrated in Figure 4)

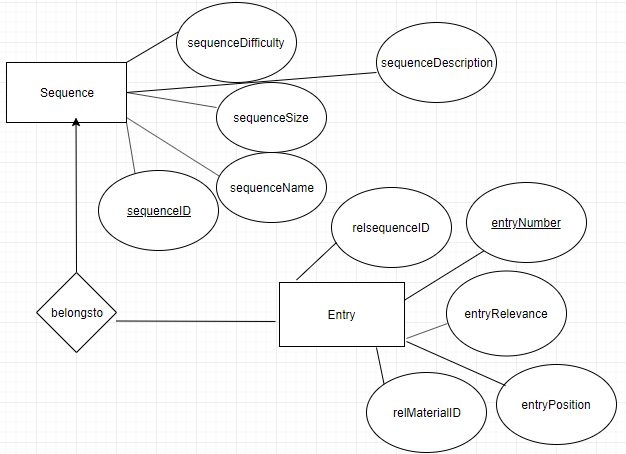


Figure 4: Sequence and entry entities and their attributes. Notice the many to one relationship, so multiply entries only relate to one and one sequence only.

Attributes of sequence and their meanings:

* sequenceID: Primary key of a sequence created by a use
* sequenceDifficulty: A string indicating whether the sequence is amateur, beginner, intermediate, etcetera
* sequenceSize: The amount of entries in a sequence.
* sequenceName: What the learner user decides to title the sequence.
* sequenceDescription: A text field that users/learners can use to market their specific sequence to many different users.

The sequence mentioned in the requirement analysis consisted of entries.

To save space in the Sequence entity, I chose to make a weak entity, entry, which contains information for entries corresponding with at most one sequence. We save space by using the entry entity by not including every description or sequence title for each entity, we eliminate data redundancy. If we had 1000 entries in a specific sequence, would we want to repeat 213 characters for the sequence description for each entry in the sequence table? No! That’s why the entry entity is essential to store sequence entries separate from the potentially detailed sequence itself.

Attributes of entry

* relSequenceID: relates the entry to the sequence it belongs to. Used for a reference to know which entries belong to what sequence
* entryNumber: essentially ID for entries. For example if you have 14 sequences and 1000 entries between all of them, there will be an entry number with entry 1000. Primary Key
* entryPosition: the number in which the entry is order in the sequence. Say if you have 1 sequence with three entries, then you would have three rows in the entries table with entryPosition having “1” “2” and “3”. What position each sequence entry is in is determined by the user who creates the sequence.
* relMaterialID: the material referenced by the entry.
* entryRelevance: small text, maybe 30-50 characters max describing why this certain entry is in the sequence. Otherwise how would a sequence user know what the entry is about?

**Big three Relationship (User, Material, Sequence)**

My relationship between user, material and sequence is split into some logical reasoning (Figure 5)

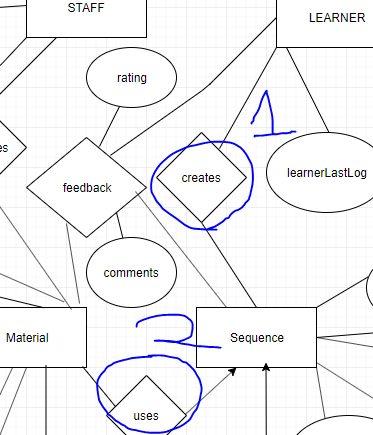


Figure 5: Big three relationship.

**1)** Learner is connected to “sequence” via a many to one relationship named “creates”. I believe it is appropriate to have a many to many relationship here because a learner/user can create a set of sequences, and many of those sequences can be recreated by learner/user/people.

**2)** Material is connected to sequence via a many to one relationship. My decision for this is based off a sequence can use many material, but many material only uses one sequence.

**Other Relationships based on requirements**

Staff and Material have a relationship named updates (Figure 6), where staff can add materials that previously did not exist.

Learner, Material, And Sequence have a ternary relationship with attributes: “Feedback”, where users can provide insightful comments and fair ratings to other sequences and/or materials (Figure 7).

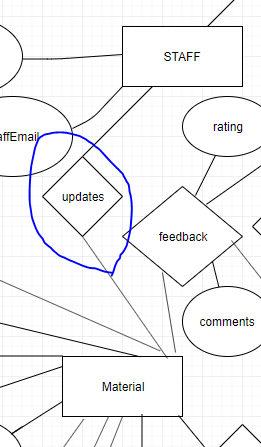


Figure 6: The updates relationship between Staff and Material as mentioned above. Add material if does not exist.

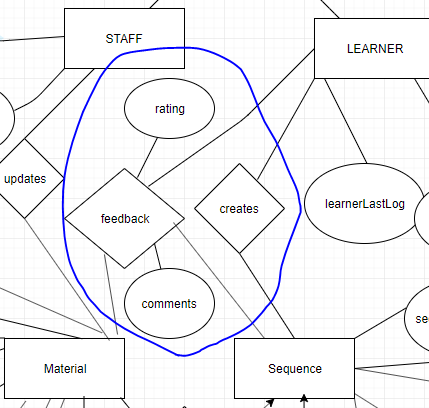


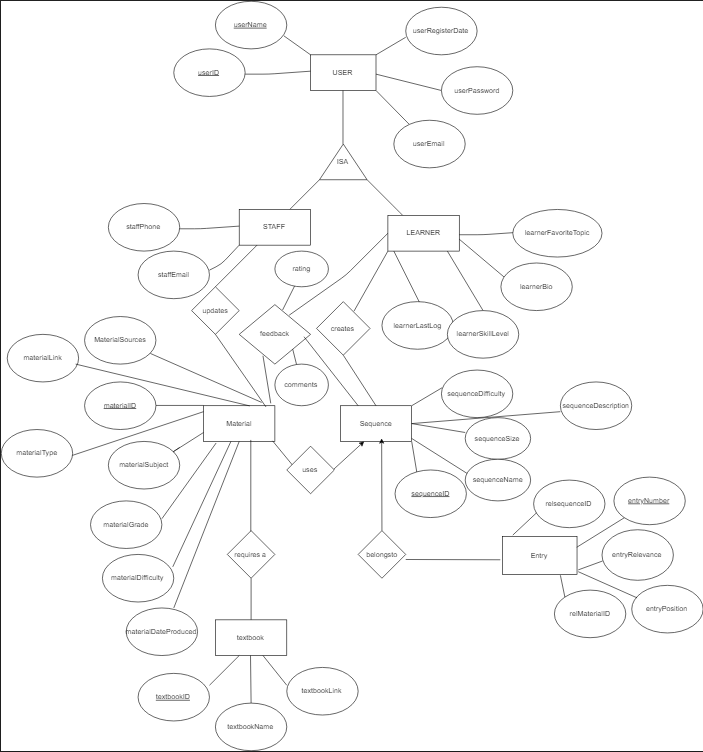
Figure 7: The feedback ternary relationship, including relationship attribute “Rating” and “comments”

**Final E/R Diagram thoughts**

Over my years of database administration and design, I have never encountered E/R diagrams. After a bit of a learning curve, I can now see clearly as to how useful they are for database design and implementation. I used to draw out relational tables one by one, but the E/R diagram would save time and resources. However, I feel like with the experience I still have from hand drawing relational table designs, that the E/R Diagram was a challenging yet manageable concept to me. Some things I am grateful for are the allowance on assumptions, which effectively allow me to design the database to my own requirement analysis standards. Overall, I am excited to begin the full implementation of my Database in SQL, and look forward to perfecting E/R diagrams in the future.

My draw.io link: https://drive.google.com/file/d/1SnGY2yOjkaxfYf0bp4QkvSEpXuj75WPH/view?usp=sharing

My draw.io final diagram:



Journal

