**Over tracking system on our Training and the candidate enrollments**

I came across this situation in my daily work at UNO. We as an organization provide multiple trainings such as Synchronous online trainings, Asynchronous training, Supervision and Exam Prep to multiple working professionals. Please go through the below details of each process and their workflow.

Synchronous Training:

There are multiple trainings on different topis given in different dates. Candidates are registered into these trainings online depending on their choice. Candidates can enroll into multiple trainings and a training can have multiple unique students.

Handouts are provided to enrolled students before the starting of training for more context on the topics that they are going to learn in sessions. These handouts are provided by Speakers. Speaker can take multiple trainings.

A quiz is conducted at the end of training to test the knowledge that each candidate has gained. Depending on the scores they gain they are awarded with CEUs. Candidates can take multiple attempts to clear the test to get their CEUs.

Asynchronous Training:

Once Synchronous training is completed it is turned into Asynchronous training and kept open of all the candidates who are willing to watch is anytime anywhere. A by default quiz is created for the candidates who were willing to have CEUs and once they are qualified, they were provided the CEUs. Candidates can take multiple attempts to clear the test to get their CEUs.

Supervision and Exam Prep are the two supporting programs that will help the candidates with their professional experiences and their exams in this process. Candidates will register into the online sessions and the candidate will be trained and supervised on the topic they are willing to learn for.

For each training and supervision there are certain fixed amount of prices for which candidate needs to pay while they enroll into a session.

Below is the data model for tracking end to end process of our trainings. We have used Oracle Data Modeler to do this task.

Diagram

Description automatically generated

*If the diagram is not clear, please refer to the file in this git page:* [*https://github.com/AnuragPanithi/DM-mid-term-project*](https://github.com/AnuragPanithi/DM-mid-term-project)

*File name:* ***mid term.dmd***

**1. Discuss any choices you had to make in modeling the data. What alternatives did**

**you consider, and why did you choose the alternative you did?**

The choices that I had to make while modelling this data are we offer sessions with same name multiple times in a year, or we keep on updating the training data and resources with day-to-day life this makes our training content change after certain amount of period. To make sure we have the track of old data and new data and to have control over entity I have created a Unique Key for each of the training, depending on the key and date we can identify which version of the training it is and how many people attended it.

Currently we are accepting payments in real time after completion of quiz, which is during CEU collection, if a candidate needs CEU they have to pay. The model here will track the people who paid fees and received their CEUs and by eliminating them we can see people who qualified the test and not requiring the CEUs.

**2. Consider the data from the perspective of ethics as described in your text (Chapter**

**2). Identify a guiding principle, risk, practice, and control for some data entity in**

**your scenario (see pages 61-62 for an example).**

Keeping the guiding principle in consideration the handouts entity is designed in such a way that it can only be identified by speaker, this is because the speaker is the only one who will create the handouts and these handouts are given only for the Synchronous Trainings.

The entity enrollment is the main base here as there are other entities that are pointing into this. There will be large dependency on all the other tables this can be one of the effected entity due to data redundancy. But in my model as the data from other entities is not mandatory this is particular entity; we can further proceed with making reports and all.

*To find the reports and all other related files for this project please refer to the GitHub link below:*

[*https://github.com/AnuragPanithi/DM-mid-term-project*](https://github.com/AnuragPanithi/DM-mid-term-project)

**SQL Queries using Assignment 1 data base:**

/\* For each challenge, list the challenge name, the nickname of the member accepting that challenge, and the day the challenge was accepted along with the total no of hours passed since it has been accepted. Narrow down the results to Location Nebraska and sort the hours from largest to smallest.\*/

***SELECT*** *challenge.challenge\_name, member.member\_nickname, acceptance.acceptance\_date, (sysdate-acceptance.acceptance\_date) hours\_acceptance , location\_state*

***FROM*** *member*

***LEFT******JOIN*** *acceptance* ***ON*** *member.member\_id* ***=*** *acceptance.member\_id*

***LEFT******JOIN*** *challenge* ***ON*** *challenge.challenge\_id* ***=*** *acceptance.challenge\_id*

***LEFT******JOIN*** *location* ***ON*** *member.location\_id* ***=*** *location.location\_id*

***WHERE*** *location\_state* ***=*** *'Nebraska'*

***ORDER******BY*** *hours\_acceptance* ***DESC****;*

/\* For each challenge, list the challenge name, the nickname of the member accepting that challenge, Narrow down the results to Location to city that starts with letter 'O'.\*/

***SELECT*** *challenge.challenge\_name, member.member\_nickname, location\_city*

***FROM*** *member*

***LEFT JOIN*** *acceptance* ***ON*** *member.member\_id* ***=*** *acceptance.member\_id*

***LEFT JOIN*** *challenge* ***ON*** *challenge.challenge\_id* ***=*** *acceptance.challenge\_id*

***LEFT JOIN*** *location* ***ON*** *member.location\_id* ***=*** *location.location\_id*

***WHERE*** *location\_city* ***like*** *'O%'; -- this will give all the results that start with 'O'.*

/\* this is something i wanted to try. I wanted to limit rows like SQL server but in Oracle it sues fetch condition to complete the task.\*/

***SELECT*** *\**

***FROM*** *acceptance*

***FETCH NEXT 4 ROWS ONLY****;* -- you can use fetch next (no of rows) percent rows only as well to limit x% of fata in the sample.