**Software Developer Course Assessment**

**Quantitative Assessment Practice #2**

**Course Name: Database Programming and Data Processing**

**Current Week:** 06 Feb 2022

**Introduction:**

The purpose of this assessment is to help us understand how the class is doing in terms of the review material that we have covered during the previous couple of weeks. The **only** purpose of this assessment is for us to improve our approach to review and ensure that what we’re currently doing is an effective strategy. Completion of this assessment is **mandatory - if you don’t submit a solution, it will be marked as incomplete. You must complete a minimum of 80% of your assigned QAPs per course – otherwise you will be marked as incomplete for that course no matter how good your other grades are.** If you do submit a solution, it will be marked as complete, as you will receive full marks no matter what your actual performance was – again this is a participation grade.  
  
**Again, the goal here is to help you** all in the best way that we can, so please do be honest when answering the questions related to how long it took, which resources you used, etc. And please ensure that you do your **own** work – don't just copy off a friend to get it done, earnestly do your best with it. If you can’t get it completely working, give us what you have. While it will be graded, the grade will not count against you, it’s just a way for us to see where everybody is, and to know which concepts, if any, we, as a class, may be struggling with.

**Deadline:** You will have until the end of the day on Tuesday (23:59 pm) to submit your assessment solutions. Please ensure you answer all the questions outlined in the instructions portion of this document as well in your submission.

**Marking:** In this program core evaluation is marked with one of three possible marks: *Incomplete, Pass, Pass Outstanding.* For QAPs, though, where incomplete marks are more important for our own information as well as for the information of the student, we wanted to increase the resolution of our grading system. Therefore, QAPs are marked on a scale of 1-5. The details of this marking system are summarized in the table below.

|  |  |
| --- | --- |
| **Grade** | **Meaning** |
| 1 | *Incomplete.* Student shows severe lack of understanding of the material – solution is heavily incomplete, non-functional, or completely off base of what the assignment was asking for. |
| 2 | *Partially Complete.* Students show some understanding of the material. Solution may be non-functional or partially functional, but the approach is correct, albeit with some major bugs or missing features. |
| 3 | *Mostly Complete.* Student demonstrates understanding of the major ideas of the assignment. Solution is mostly working, albeit with a few small bugs or significant edge cases which were not considered. Shows a good understanding of the correct approach, and is either nearly a feature-complete solution, or is a feature-complete solution with some bugs. |
| 4 | *Complete (Equivalent to: Pass.)* Student shows complete understanding of assigned work and implemented all necessary features. Any bugs that are present are insignificant (for example aesthetic bugs when testing the functionality of code) and do not impact the core functionality in a significant way. All necessary objectives for the assignment are completed, and the student has delivered something roughly equivalent to the canonical solution in terms of features and approach. |
| 5 | *Complete with Distinction (Equivalent to: Pass Outstanding)* The student demonstrates a clear mastery of the subject matter tested by the QAP. The solution goes above and beyond in some way, makes improvements on the canonical solution, or otherwise demonstrates the student’s mastery of the subject matter in some way. A solution in this category would consider all reasonable edge cases and implement more than the necessary functionality required by the assignment. |

**Instructions:**

You are allowed to complete the assessment problems below in whatever way you can but please answer the following questions/points as part of your submission:

1. How many hours did it take you to complete this assessment? (Please keep try to keep track of how many hours you have spent working on each individual part of this assessment as best you can - an estimation is fine; we just want a rough idea.)
2. What online resources you have used? (My lectures, YouTube, Stack overflow etc.)
3. Did you need to ask any of your friends in solving the problems. (If yes, please mention name of the friend. They must be amongst your class fellows.)
4. Did you need to ask questions to any of your instructors? If so, how many questions did you ask (or how many help sessions did you require)?
5. Rate (subjectively) the difficulty of each question from your own perspective, and whether you feel confident that you can solve a similar but different problem requiring some of the same techniques in the future now that you’ve completed this one.

**Exercise One:**

The ability to analyze and design databases is a desired technical skill. The ability to identify the entities and their attributes from real-world business scenarios is essential for programmers and database professionals.

The problems for this QAP come as exercises in the realm of designing database schemas from written scenarios. There are five written scenarios, complete two of the five scenarios to the best of your abilities.

Continuing to develop an exemplary understanding of SQL will serve you very well as a software developer or a database administrator. During this selection of exercises, you will also be asked to write SQL CREATE statements to implement the designs from each completed exercise. Take the time to consider the data types best suited for each attribute or field within each entity.

**Complete two of the five written scenarios:**

1. Exercise1CityProvince – cities within their provinces
2. Exercise2DesigningEntities – designing basic database entities
3. Exercise3PublicationUsage – dealing with data over time
4. Exercise4SimpleNormalization – getting to third normal form
5. Exercise5StudentClasses – students and their grades

**BONUS:** Complete all five exercises including their respective database table CREATEs.

**Approach:**

I suggest you work in a software tool that can create diagrams; the choice of tool is yours. The diagram files should be submitted as jpg or pdf file(s). You can also create the diagram(s) on paper with pen or pencil and scan to a pdf or jpg. Iterate your designs; have peers review them and make suggestions, implement the diagrams as SQL CREATE and learn if they work. Sometimes implementing entities (tables) as SQL CREATE adjusts the design. Become comfortable in creating tables and then dropping (deleting) them and starting again.

Once you have completed your design diagrams implement them by either writing the SQL CREATE statements or using the create table… dialog in pgAdmin. Once the CREATE statements are successfully completed save each to disk files in preparation for assignment submission.

**Exercise Two:**

Find online or write your own simple business scenario that would require a database to store information. This description should be one or two paragraphs in length.

Read through the paragraph description you have found or created and identify the business entities that could become database tables. Create a written list of the entities, with each having a brief description of purpose.

**Example:**

**Scenario:** A mobile catering truck sells coffee break and lunchtime meals to businesses during the weekdays. On Fridays and Saturday nights they set up in the bar district to sell evening snacks. They require a point of sale (PoS) system to keep track of the meal sales, process the financial transactions, and keep basic inventory of what is sold at each location and the time of the sale. They do not have the same menu every day and at every location. They want to have variety as part of their offering. The business is new, and the owners want to track what is sold when and where. They will use this information to improve their food inventory purchasing and keep the menu active to increase sales at each location.

**Entities:**

* Location – location of the truck while selling meals.
* Receipt – basic information for the sale; total amount, payment method, datetime, etc.
* Meal-Items – list of all items in the meal sale, consider meal-items as the detail records of the Receipt entity.
* Menu – all items available for sale on a given day
* Inventory – a complete list of every meal item they could have for sale

**Overall Deliverables:**

**Exercise One:**

1. One set of schema diagrams for each exercise completed. Each diagram(s) should have fully defined entities and entity relationships (ER Diagram). You may do one or more diagrams for each exercise; the choice is yours. These diagrams can be done with a software diagramming tool or drawn on paper with pencil and scanned. The diagram should be submitted as either **pdf or jpg** file(s).
2. One set of CREATE statements for all the designed entities and attributes. Be sure to set the Primary Key (PK) CONSTRAINT. Bonus marks for also setting the Foreign Key (FK) CONSTRAINTs.
3. Note: each exercise should have an ER Diagram and the respective table CREATE statements.

**Exercise Two:**

1. A paragraph (or two) describing the business scenario.
2. A list of entities derived from the business scenario.

**Project Submission:**

Store all your answer files into a single working directory (folder) and archive them into a single zip file. Submit/Upload/Attach this zip file to your QAP2 assignment in Microsoft Teams.