

COURSE: DATABASES

ASSIGNMENT 9

ENTITY-RELATIONSHIP MODELING

USING UML DIAGRAMMATIC NOTATION

Theory part:

Entity-Relationship Modeling using Unified Modeling Language (UML) diagrammatic notation. Explain the basics of ER modeling in database environment using UML.

One of the most difficult aspects of database design is the fact that designers, programmers, and end-users tend to view data and its use in different ways. Unfortunately, unless you gain a common understanding that reflects how the enterprise operates, the design you produce will fail to meet the users' requirements. To ensure that you get a precise understanding of the nature of the data and how it is used by the enterprise, you need a model for communication that is nontechnical and free of ambiguities. The Entity-Relationship (ER) model is one such example. ER modeling is a top-down approach to database design that begins by identifying the important data called entities and relationships between the data that must be represented in the model. You then add more details, such as the information you want to hold about the entities and relationships called attributes and any constraints on the entities, relationships, and attributes. ER modeling is an important technique for any database designer to master.

Practical part:

Create an ER model for the following description:

A company sells books on the net. There are several book Publishers included. The information related to a publisher is street address, postal town (town/city), postal code, country name, and URL.

If a book has only one Author, the information related to an author is street address, postal town (town/city), postal code, and country name. Sometimes, a book can consist of several articles. In those cases, Editor information is again street address, postal town (town/city), postal code, and country name.

A Book has a name and it is categorized by Subject (Life Sciences, Education, Medicine, Mathematics, ...), publication date is known and the price of the book. Book entity must also include Publisher information (who published), Author information (who wrote), Editor information (who was the editor, if any), and Subject information.

Customer information includes name, street address, postal town (town/city), postal code, country name, and email address.

Order information includes Customer information (who ordered), Book information (what was ordered), date of ordering, how many books has been ordered, how many has been delivered, and when they have been delivered.

Build one Word document where you copy and paste all the source code (from the final solution) you have generated during solving the task. Your word document should also show the functional details of the solution. One generic example from functional details: if the task is to make some calculations with the user input, use print screens to show one successful use case where the input is received and calculations will be completed. Your document need not be a complete road map from each individual step, but it should still be understandable and show street credibility to the outside reader. Use exactly the same format you would use with the thesis document (not description sheet, please!), or alternatively use the shorter report template. You can find the instructions from the student intranet.

Theory part can be at the beginning of or at the end of your word document where you have your **Practical part**. Remember to add all the references used in your document!

Take your script files together with your Word file and zip it into one file. Return that composite zip file to Moodle.

Assessment: Half from the points come from **Theory part**, another half from **Practical part**.

Submit your task before deadline! It is not possible to return this task after the deadline.