

7 Database Design

Database design plays a major role in application systems. Almost all courses/programs, which are related to software development and information technology, include at least one module on database systems. Like other sections of this book, the intent of the book is not to repeat the concepts that you might have studied thoroughly, rather to show you the way that you can adapt your knowledge about the subject and utilize it properly in your final year project.

7.1 Database Management Systems

Database Management System (DBMS) is a software system, which manages database construction, update, and access control, and maintenance. I assume that you are familiar with some DBMSs such as My SQL, MS SQL, MS Access, Oracle, etc. Some DBMSs are licensed software and some others are open source. You have to choose a DBMS as the database tool for your final year project if you are going to develop a database application.

Below you can find some tips on selecting a DBMS.



Tips on DBMS section

- Choose a DBMS that you have already your hands dirty with it.
- Unless you have strong reasons don not go for small/medium-scale databases such as MS Access.
- Choose among large-scale databases e.g. My SQL or MS SQL.

7.2 Relational vs. Object Oriented Database

The most common DBMSs in the market are Relational Database Management Systems (RDBMS). Their popularity, both in the market and education, make them the first choice for most of the final year projects. However, other types of databases, particularly Object Oriented Database Management Systems (OODBMS), have steadily been growing during the past decade. In fact, you can download and use some of them such as ObjectStore and combine them with your programming environments i.e. MS Visual Studio.

So, which one is preferable? RDBMS or OODBMS? My quick answer to this question is “if you are about to do an ordinary application and you do not have previous background on OODBMS, and particularly if your methodology is not object-oriented do not think about using an OODBMS. In other words, unless you have a strong reason and desire to use OODBMS, from one side, and it can give more facility and robustness to your system that is not achievable using RDBMS, choose one of the well-known RDBMSs, especially among those with which you have already had some experience”.

Finally, some RDBMSs support Object Relational Models (ORM), which allow developers to map their object models into relational models. Although, the process of mapping your class model into an ERD might persuade you to use ORM, but you should be careful about the concepts behind each one. To clarify, you can use ORM only if the DBMS supports it. In this case, you should develop your data layer in your system architecture according to this decision. However, if you are not planning to use ORM, then you have to map your persistent classes into an ERD, and then apply normalization techniques on the result to obtain your relational database design. In this case, you have to develop your data layer, accordingly, which might not be the same as you could have in the previous situation.

7.3 Data Modeling

Data modeling, which is also called conceptual database design, is a core activity in database applications. It is comparable with the importance and the role of foundations of constructions in civil engineering. I assume you have had at least one module/course about database design before you start your final year project. However, this familiarity cannot guaranty that your database design is proper. Below you can find some tips on data modeling.

Tips on Data Modeling

- Use an automated tool, preferably the one that your chosen DBMS provides, for your data model.
- Apply normalization rules and normalize your database design at least to the 3rd Normalization form or to BCNF.
- Use understandable names for relations/tables and attributes/fields.
- Use proper data types and apply DBMS provided restrictions and rules such as Not Null.
- Use proper naming for Foreign Keys and Relationships.
- Check your design by providing data through DBMS interface and control the expected outputs by running some Structured Query Language (SQL) commands on the provided data.

If you follow an ordinary (structured) methodology, then you would provide an Entity Relationship Diagram (ERD). You can use this diagram to provide a normalized data model, afterwards. However, if you follow object-oriented methodology then you should know how you could map your class model to an ERD. Most of IDEs and development tools are providing tools, add-ons, and facilities to help in the mapping process. If you are using one of these tools then your job is easier. Simply use the facilities that they provide. However, if you do not have an automated tool to do so, below you can find some rules of thumb for mapping a class model into an ERD, manually.

Mapping a Class Diagram into an ERD

- Indicate your persistent class, e.g. by using a persistent stereotype.
- Create an Entity for any persistent class, giving the name of the class.
- Map all data members of the class as attributes of the created Entity.
- Give the same, or equivalent, data type to the attributes.
- Make the class identity (ID) the Primary Key for the Entity.
- Map associations, aggregations, and compositions to relationships and use class IDs as Foreign Keys.
- Change generalization/specialization relations of the class model to superclass/subclass relations and apply ERD rules, accordingly.
- Map multiplicities of associations to the equivalent cardinalities in the ERD.

7.4 Database Design

The next step is to prepare a database design, which is also called physical database design, based on the data model. This model is quite similar to your data model. However, data model is a DBMS independent model while database design should be prepared according to a specific DBMS. The reason is in database design you have to specify the attribute types, length, key types, attributes' default values, and other parameters, which may be slightly different from one DBMS to another, yet this slight difference can make your design to work on one DBMS, but not the other.

7.5 Using Stored-Procedures and Triggers

Stored-procedure is a specific procedure that is written in Standard Query Language (SQL) as a database item. Stored-procedures can accomplish database related requirements efficiently. Although they are part of database, however, many experts categorize them under business logic layer in a three-tier/multi-tier software architecture. Stored-procedures are independent of specific tables.

Trigger is a piece of SQL, which can perform a sequence of actions if an action such as Create, Update, and Delete happens on specific table.

If your project were a database application, and if your DBMS supported stored procedures and triggers, it would be a good practice to use these features. In this case, do not forget to document stored procedures and triggers properly and provide them as part of supplementary documents in your final report.

7.5.1 XML

Extensible Markup Language (XML) is a markup language that you have studied during modules related to database or web technology. I advise you to consider XML as a data container in your final year project if one or more of the following conditions are applicable. Be careful about the term “data container” that I used. Differentiation between a “data container” and a “database” is important. Several arguments and discussions can be found on whether you can use XML as a database or not. However, many DBMSs provide diverse number of utilities to facilitate using XML with databases. This phenomenon, in its essence, shows that the arguments against using skeletal format of XML as a database have been widely accepted. On the other hand, providing specific tools regarding securing the XML files and treating them as database, particularly for semi-structured or unstructured data is becoming very popular.

If you use XML in the context of DBMS, then you have to follow and apply specific rules that are pertinent to using XML in that context. You can find several resources on this in the bibliography section. Anyway, I will leave you in peace by avoiding more technical (even philosophical!) discussions on this matter, rather, I provide with some tips on using XML in your final year project.

Tips on using XML

- XML is mainly a technology to be used to organize/standardize data exchanges
- Consider using XML as a data container in your final year project if:
 - Your data is semi-structured or unstructured
 - You do not have heavy update on your database
 - Security is not a main concern
 - You do not deal with a large amount of data
- Ignore all the above and use XML if you are not obliged to use a traditional database! This is an opportunity for you to practice an amazing software technology.

7.6 Summary

Most of the final year projects are dealing with databases one way or another. However, some projects might not use any specific database or database systems. If your project includes a database or if it is a database application, then you have to design a proper database. Although recently Object Oriented Database Management Systems (OODBMS) have started to play a greater role than they were playing, however, Relation Database Management Systems (RDBMS) are still the most popular databases, both in the market and education. Therefore, whether you use an object-oriented or traditional approach, it is most probable that you would use a RDBMS in order to implement your database.

If you follow a traditional approach, you are required to create an Entity Relationship Diagram (ERD) during your database design. However, if you follow an object-oriented approach you do not have this diagram, hence, some straightforward rules were presented in this chapter, using which you can create an ERD from your Class Model. Then, you can follow the normal process of normalization and database design as you would do it in the traditional approach. Finally, the importance of choosing the proper RDBMS was discussed and some guidelines were provided in order to help you in finding suitable RDBMS for your final year project.



www.sylvania.com

We do not reinvent the wheel we reinvent light.

Fascinating lighting offers an infinite spectrum of possibilities: Innovative technologies and new markets provide both opportunities and challenges. An environment in which your expertise is in high demand. Enjoy the supportive working atmosphere within our global group and benefit from international career paths. Implement sustainable ideas in close cooperation with other specialists and contribute to influencing our future. Come and join us in reinventing light every day.

Light is OSRAM

OSRAM SYLVANIA 

