Graduate Student Project

If you are registered for the graduate version of the course, 565, then you will be implementing the case study described in Appendix B.I entitled "The University Accommodation Office Case Study." The book's description of the case study is by necessity somewhat brief. You will probably have questions about certain parts of the case study. Please post your questions to piazza, so I can share the answers with everyone. You may be frustrated about the need to ask questions, but in the real world you will need to ask your user/clients questions, so get used to it. Further, make sure you do not start the project at the last minute because you may find that you need answers to questions, and you may have to go through piazza to find answers, or ask questions if you cannot locate the information on piazza. I will not be repeating answers to questions that have already been asked.

You will need to

1. Design and implement the database using mysql. Please use the database account that you were assigned for this course.

2. To design and implement the database, you will need to design and implement a set of relations, populate them with example data, and test the sample query transactions in B.1.2. Make sure your database works with all of these sample query transactions as the TA may test any or all of them. In addition, make sure that your database can answer the following query:

   > List the names of students, their apartment number, and their bedroom number renting an apartment at 1678 Cardiff Rd.

3. You should submit:

   a. an ER diagram for your DB,
   b. an English language description about the tables you used, including
      - their names
      - their attributes,
      - whether these attributes are mandatory or optional (can be NULL)
      - their primary and foreign keys,
      - any constraints you used. Even though mySql does not support constraints, you should try to identify constraints that you think should hold, based on the books description.
   c. a source file containing the SQL DDL code you used to create your database. I do not want to see the code you used to populate your relations with concrete data.
   d. a separate source file containing all the sample queries from B.1.2 plus the query listed above.

4. You will submit your database project by demoing it for the TA. The demo will consist of showing him the sample data in your relations and demoing the queries from B.1.2. You will need to see him during office hours or set up an appointment with him to demo the database. You should print out and bring your descriptive document with you to the demo and give it to the TA at that time.

Here are some clarifying assumptions about the project:

1. ignore the student program of study at the end of the project description--it has already been specified as a major with the student's information

2. A place number uniquely identifies a room in the entire residence system, hence you may assume that if a room in a flat has a particular place number, then no room in the residence halls will have that place number.

3. I have not been able to think of a clean way to create a single table for apartment and residence halls. I think you need separate tables to handle them well. One possible solution is to place common information for the two into one relation (this relation would hold information about both apts and halls), and then create two other relations, one for apts and one for halls, that contains additional information that is unique to each. It is up to you though to figure out the best way to handle the situation, and your solution may vary, depending on how you design the rest of the database. You are certainly welcome to introduce primary keys or other information as you deem fit, just make sure you can justify it.

4. A course title consists of two values, a name and a year. They are two separate values. For example:

```
title     year
COSC 465  2011
COSC 465  2012
```

5. It is okay to assume either that an invoice must be paid in full, or that it may be paid in installments. State your assumption and then design your DB accordingly. Your DB design will be somewhat different depending on whether or not you assume that an invoice must be paid in full, or can be paid in installments. As for the determination of the invoice amounts, I am again willing to entertain reasonable assumptions. Both the 4-4-4 and the 5-5-2 ones I've seen mentioned are reasonable ones.

   As for determining total rent, the spec indicates that payment amounts are recorded. Hence somewhere in your DB you will store information about these payments and a query should be able to sum them up using an aggregate operation.