Designing Relational Databases -- Step 1.
Requirements Gathering – What Data Are Needed?

- Course Title
- Term (fall, winter or spring)
- Meeting days
- Course number
- Number of credits
- Activity (seminar, lecture)
- Department name
- Section number
- Session number
- Catalog number
- Department code
- Building
- Room
- Meeting Time

Step 2.
Organize or Group Data “Fields” or “Attributes” by 1-1 Relationships
Designing Relational Databases -- Step 1.
Requirements Gathering – What Data Are Needed?

Course Title
Term (fall, winter or spring)
Meeting days
Course number
Number of credits
Activity (seminar, lecture)
Department name
Section number
Session number
Catalog number
Department code
Building
Room
Meeting Time

Step 2.
Organize or Group Data “Fields” or “Attributes” by 1-1 Relationships

Department name
Department code
Building
Assumes that each department has only one building

Course title
Course number
Number of credits
Activity (seminar, lecture)

Section number
Session number
Catalog number
Meeting days
Term
Meeting time
Room
Building (may be different than department building)
Step 2.
Organize or Group Data “Fields” or “Attributes” by 1-1 Relationships

- Department name
- Department code
- Building
- Course title
- Course number
- Number of credits
- Activity (seminar, lecture)
- Section number
- Catalog number
- Meeting days
- Term
- Meeting time
- Room
- Building (may be different than department building)

Step 3.
Draw Entity/Table Relationship Diagram

Looking at Step 2, what are the entities?
What relationships do they have?

Draw an entity-relationship diagram.
Step 3.
Draw Entity/Table Relationship Diagram

While Sections do have an association with Departments, there isn’t a direct relationship. The relationship is really THROUGH Courses. So you don’t need to model this relationship directly.

THIS RELATIONSHIP NOT NEEDED.
Step 3.
Draw Entity/Table Relationship Diagram

Step 4.
Draw Primary Keys (fields that make a record unique) on Entity/Table Relationship Diagram
Step 4.
Draw Primary Keys (fields that make a record unique) on Entity/Table Relationship Diagram

- **Departments**
  - Department code (primary key)
  - M
- **Courses**
  - Department code
  - Course number
  - 2 fields together make up the primary key
  - M
- **Sections**
  - Department code
  - Course number
  - Section number
  - 4 fields together make up the primary key
  - M
Step 4.
Draw Primary Keys (fields that make a record unique) on Entity/Table Relationship Diagram

Step 5. Where will joining of tables be needed?

The arrows in your database design will help you identify where joins between tables will take place. To prepare for a join, you need to store the primary key of the “one side table” in the “many side table.”

A. Look to your entity-relationship diagram. What are the tables that have direct relationships?

B. For each relationship (arrow), which is the "many" table?

C. Do you need to add the primary key of the “one table” as a foreign key to the “many table”? Or is it already in the many table as part of its own primary key?
Step 5.
Where are joins needed? Make sure “many side” table has the primary key of the “one side” in it to allow for joining.
Step 5.
Where are joins needed? Make sure “many side” table has the primary key of the “one side” in it to allow for joining.

Step 6. Add Other Attributes Identified in Step 2
(You can do this on the diagram or in a word processor as an attachment)
Step 6. Add Other Attributes Identified in Step 2
(You can do this on the diagram or in a word processor as an attachment)
Step 7. Create Database Tables in Access

Step 8. Set Up Table Relationships

Step 9: Queries

Step 10: Data Entry Forms

Step 11: Reports

Step 12: Miscellaneous (installation, teaching end users, security, backup)
Defining the section table

<table>
<thead>
<tr>
<th>Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Department code</strong></td>
</tr>
<tr>
<td><strong>Course number</strong> (primary key, but also foreign key for joining)</td>
</tr>
<tr>
<td><strong>Section number</strong> (primary key)</td>
</tr>
</tbody>
</table>

Format - @ @ - @ @

Catalog number

Format – upper case (>)
Meeting days
Term
Meeting time
Room
Building (may be different than department building)