This assignment has a reading part and a development part. Hand both parts in together at the beginning of class on the due date. By the end of the assignment, you should be familiar with the following terms and concepts.

```
<table>
<thead>
<tr>
<th>Physical Level</th>
<th>Logical Level</th>
<th>View Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Data Independence</td>
<td>Relational Model</td>
<td>Table</td>
</tr>
<tr>
<td>Relation Schema</td>
<td>Relation Instance</td>
<td>Tuple</td>
</tr>
<tr>
<td>Attribute</td>
<td>Domain</td>
<td>Database Schema</td>
</tr>
<tr>
<td>Database Instance</td>
<td>Primary Key</td>
<td>Candidate Key</td>
</tr>
<tr>
<td>Superkey</td>
<td>Foreign Key</td>
<td>Referencing Relation</td>
</tr>
<tr>
<td>Referenced Relation</td>
<td>Referential Integrity</td>
<td>Schema Diagram</td>
</tr>
</tbody>
</table>
```

**Part 1: Reading Assignment.** The goal of the reading assignments is to give you an opportunity to read and reflect on the material covered in the textbook. Each assignment will consist of questions about the reading. Reading assignments are assessed on how fully you answer the questions and whether you demonstrate that you understand the reading material. Answers must be clear, organized, and neatly written (preferably typed).

Read the following sections in the textbook and answer the questions below.

- Ch. 1: 1.1, 1.2, 1.3
- Ch. 2: 2.1, 2.2, 2.3, 2.4

1. Section 1.1 of the book lists various types of application areas (e.g., Online Retailers, Banking, Airlines, etc.) and the general type of information they store. List 3 actual software applications you use on a regular basis (be specific) and briefly describe the type of information you think each application is storing.

2. In your own words, briefly describe the three levels of data abstraction described in the book.

3. Using a programming language you are familiar with, describe how relational tables compare to data structures within the programming language. In particular, identify one or more built-in data structures that could be used to represent a table. Briefly describe how a table could be represented in the language and give a concrete example (code) that captures a simple table.

4. In your own words, describe the concept of a “primary key”.

5. In your own words, describe the concept of a “foreign key”.

**Part 2: Design a Database.** Your job is to define a set of relational tables for storing basic information about tech startup companies (similar in spirit to Angel List). Your tables must be designed to store information about companies, technical areas companies are focused on, investments in companies, company founders, and company employees. Specifically, your tables should be designed to support the following types of queries.
Q1: Find the name and location (US city) of all companies founded in a specific year.

Q2: For companies in a particular funding stage (either Seed, Series A, Series B, Series C, or Acquired) and of a particular type (either Startup, VC Firm, Incubator, or Private), find the total amount of money each company has raised.

Q3: Find all companies in a particular area. An area is just a tag like “Retail”, “Mobile”, “Education”, “Advertising”, “Games”, and so on. Note that any particular company can be associated with one or more areas.

Q4: For a particular company, find all of its investors and the amount and year of the investments. Note that a company can have both other companies (e.g., venture-capital firms) and individuals as investors, and many companies will have multiple investors of both types.

Q5: Find all of the companies that an individual has been a founder of. For each such company, find the position they held as founder and the year they left the company (if they have left).

Q6: Find all of the employees of a company within a range of years (e.g., between 2012 and 2016). For each employee, find the employee’s name and the position they last held in the company.

Q7: Find the names of all the individuals stored in the database based on the name of last school they attended. For each individual, find their name and the last degree they earned at the school.

Note that your task is to design tables that can support these types of queries, and not to write the queries themselves. Hand in your answers for each of the following questions. Note that you will use your design in subsequent assignments, so please make sure you keep a copy of your answers (e.g., type up your answers or if you write out your answers make a photo-copy). Be sure to print a cover sheet, write an assignment reflection, and turn both your reading and design questions in together with the cover sheet attached to the front. You do not need to submit your assignment online.

1. Give the schema (table and attribute names) for your company database. Indicate which attributes are primary and foreign keys. For foreign keys, state which attributes the foreign keys refer to. You do not need to give the datatypes (domains) for table attributes.

2. Draw a “schema diagram” (see Fig. 2.8 in the textbook) for your answer in Question 1.

3. Construct instances of your schema (i.e., tables) such that each of the above queries (Q1–Q7) would return at least two rows.

4. List what the queries should return when run against your database instance.

5. We typically think of a “well designed” schema as one with no redundancy (note that there are times when having some redundancy can help performance). As an extreme example, representing the above information in a single table would result in considerable redundancy. For instance, the same company information would be repeated for every employee of the company. Based on the idea of limiting redundancy, rate how “good” you think your schema design is and why. Be specific (e.g., use examples) when describing your rating. Note that you won’t be penalized for having redundancy—the goal is to recognize whether there is redundancy and where the redundancy can occur.