In this assignment, you're asked to come up with a database to support a business domain of your choosing.

The database should have at least four tables, where all of the tables are connected to at least one other table (by one-to-many, many-to-many relationships).

Each table should be populated with at least a few rows, so that you can demonstrate the referential integrity behaviors that you implement.

You should choose at least two different behaviors. (The hands on lab demonstrated three different behaviors: the default REFERENCES clause with no reference options, ON UPDATE CASCADE, and ON DELETE CASCADE).

You should deliver code that demonstrates the behaviors that you've implemented.

In the following week's class, you'll be asked to present your chosen business domain and code.

You may work with a partner on this assignment.
You only need to create a small number of columns for each of these tables, since the purpose of this exercise is to develop a test-bed for studying the different referential integrity behaviors. You should be able to defend the referential integrity behaviors that you choose, based on reasonable business assumptions.

While you are strongly encouraged to come up with your own business domain, if you're not able to do so, you may use this example: You're a data analyst working for the City of New York, monitoring building energy consumption. You'll need these four tables:

- One table provides a list of the buildings being monitored,
- A second table that shows monthly energy consumption for each building,
- A third table lists the meetings that your organization has had with each building's management over time, and
- A fourth table lists the different types of buildings (examples: Low to Medium-Rise Commercial, High-Rise Commercial, Multi-Family, Industrial, Mixed Use).

Show reasonable behavior when:

- You combine Low to Medium Rise Commercial and High-Rise Commercial building types into a single "Commercial Category"; and,
- You delete a building from the database.