

**Assignment 1 – Due end of Week 2**

**Problem 1. Normalization**

This problem will use the file techcrunch.csv. This dataset consists of company funding records reported by TechCrunch, a website that reports on the business of technology, startups, venture capital funding, and Silicon Valley. Each row represents one funding event for a company.

1) What is a good choice for a primary key here? In contrast, give an example of an attribute (or composite) that would ***not*** be a valid primary key.

**Be careful**. You can check that your proposal really satisfies the definition of a primary key by sorting your data by the relevant column(s) in Excel.

2) Does the table satisfy 1NF? Why or why not?

3) Does the table satisfy 2NF? Why or why not?

4) Does the table satisfy 3NF? Why or why not?

**To get full credit**, if the dataset fails to be in any of the normal forms above, you should document all the ways in which it so fails. For example if you think the data are not in 1NF, it does not suffice to say “the data aren’t in 2NF because you can’t be in 2NF without being in 1NF”. You need to be more specific.

5) Sketch a proposed Entity-Relationship diagram that would bring this dataset into 3NF. If you answered “yes” to (4), for example, your ERD would just be the raw data table with no changes. If however your ERD requires multiple tables to be in 3NF, you should draw all relationships between them and indicate their type (one-to-one, one-to-many, etc.)

Problem 2. Case study (adapted from Comeau, Chapter 9)

This question is designed to exercise your ability to use JOIN statements to “undo” the often-complex relationships that result from normalization. We will use the Recipe Database case study from the textbook.

The associated **recipes.sql** file contains code that creates a new schema and populates it with associated tables and a small amount of example data.

1) Once you run the sql code you’ll notice that each table has data populated for two recipes, Chicken Marsala and Absolute Brownies.

Use the INSERT INTO statement to insert new information about **two (2) completely new recipes of your choosing into the database**.

You can make up your own recipes or copy them from a website. **Be sure to insert values into each of the tables generated by the recipe.sql file**: **recipe\_main, rec\_ingredients, ingredients, and categories.**

Use Chicken Marsala and Absolute Brownies as your guides for the problem as the instructions follow similar steps as what was already done in the code, but remember you need to do the same thing for two new recipes.

2) Write **only one SQL query** that returns **all** information on **only the two new recipes** you inserted from all the tables you created in step 1 above.

Your query should show all relevant information from all four of the tables from step 1.

Don’t worry that the output table may duplicate rows, for example the recipe\_main table will duplicate rows for each of the ingredients you enter which is ok. The idea is to get you comfortable with joins.

Show the results of your query as screenshots pasted in below:

3) Write a SELECT query that identifies the recipe name, category name, and ingredient name, and ingredient amount. No other variables should be included.

Your output should be sorted first by descending category name, then by ascending recipe name, followed by descending ingredient name.

Recall that string variables are sorted alphabetically (ascending only) when you check your query results.

**Submit one MS Word Document and one SQL program file into the Assignment 1 drop box in week 1.**

**Make sure your name is in both of your filenames**

**Example:**

**Yoni\_Dvorkis\_Assignment\_1.doc, and Yoni\_Dvorkis\_Assignment\_1.sql for your code file**