VICTORIA S. PLANGE

ALY 6030

DATA WAREHOUSING AND SQL

PROFESSOR SEAN CORNELIUS

WINTER TERM PART B SEC 02

03/03/2018



NORMALIZATION, JOINTS AND INDEXING

Problem 1. Normalization

This problem will use the file techcrunch.csv in the Supporting Materials. This dataset consists of company funding records reported by TechCrunch. Each row represents one funding event for a company. You can assume that the (composite) key is {company, round} where round denotes the given round of funding.

1. Why is {company, round} the key? Why doesn’t {company} work?

The data set is exploring company funding records and without ‘round’ as a part of the primary key, we cannot determine what type of funding stage the company is at. If company is a primary key it should have a unique ID so each company name should belong to a round of funding. Because a company can have several rounds of funding as seen in the data set it is no longer unique. Composite key means a primary key must have more than one column to identify the rows in a table. However, our composite key does not satisfy this thus it is also not unique.

1. Do the data satisfy 1NF? Why or why not?

The data does not satisfy 1NF because there are two rows which are duplicated thus it is not unique. As seen in figure 1.0 below, the company “zoom info” is duplicated within the data, which makes each of these rows not unique.

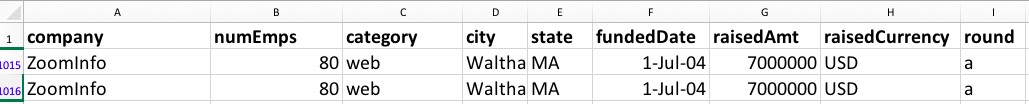


Fig 1.0

1. Does it satisfy 2NF? Why or why not?

No because since it does not satisfy 1NF, it automatically cannot be 2NF. Additionally, the rule states that non- primary keys should only depend on the primary key and not on another non-primary key. However, in our data not all non-primary keys depend on the primary key. Our primary key in this scenario is “Company -round”. However, numEmps does not depend on “company-round”, it only depends on company. In order for it to satisfy 2NF, we would have to create two new columns named “company id” and “round id” and merge those two by creating a bridge table under a new id “round company\_id” so that our a non-primary key such as numEmps would now have a primary key to depend on named “round company\_id” as seen in Figure 1.2.

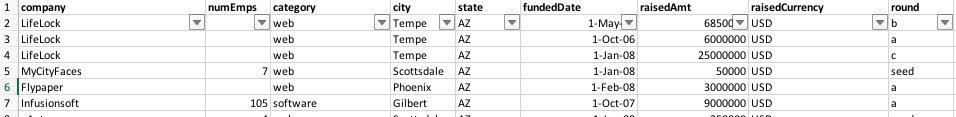


Fig 1.1

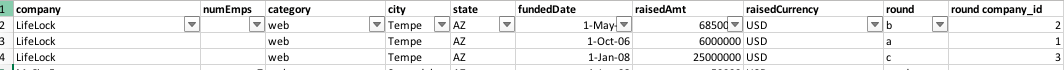


Fig 1.2

1. Does it satisfy 3NF? Why or why not?

No because if it is not in 1NF it cannot be in 3NF.As seen in Fig 1.0, there are duplicates in the data. In addition, the rule states that all non- primary key attributes do not depend on other non-primary key attributes. But city depends on state as seen in all above diagrams. Since the data set fails to illustrate the normal forms, below is a sketched entity-relationship diagram which should bring the dataset into 3NF.

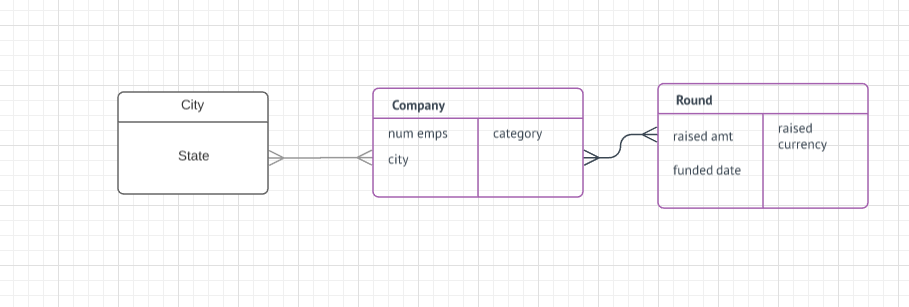


Fig 1.3

It is possible to have the same company in different cities and it also possible for one company to have more than one round of funding. i.e seed, angel, series a, b or c. Thus, it is also possible for one company to have several funding dates.

Problem 2. Case study

1) Write a set of queries to return all information on a specific recipe including main details,

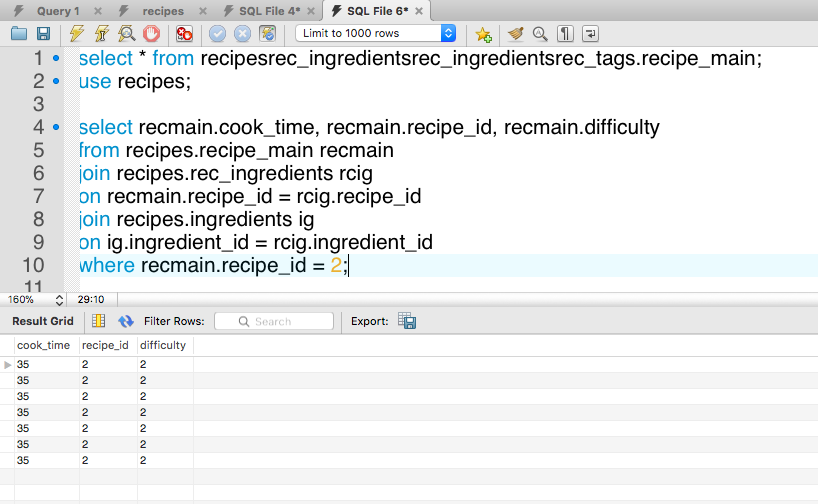
ingredients, recipe tags, nutrition, comments, food warnings and any available substitutions. Use

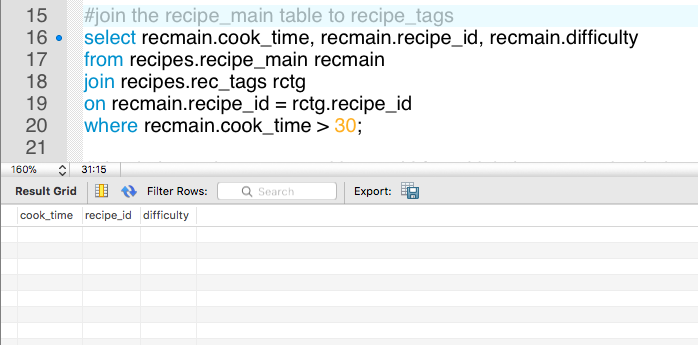
as few queries as possible. Your set of queries should be designed with creating something like

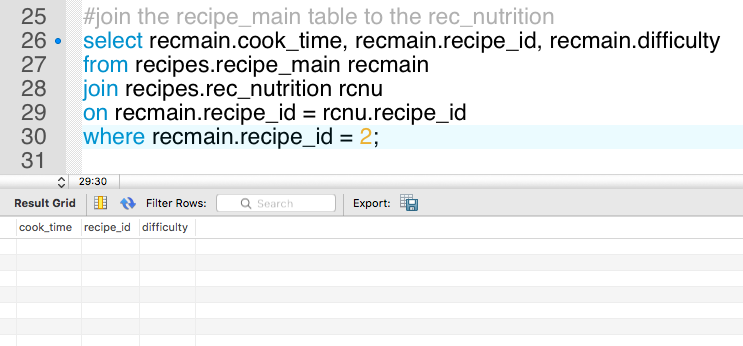
Figure 9.14 in mind—a complete recipe page (in that case, Chicken Marsala) that might appear on a website. Remember to use aliases on the field names that are returned so that the raw query

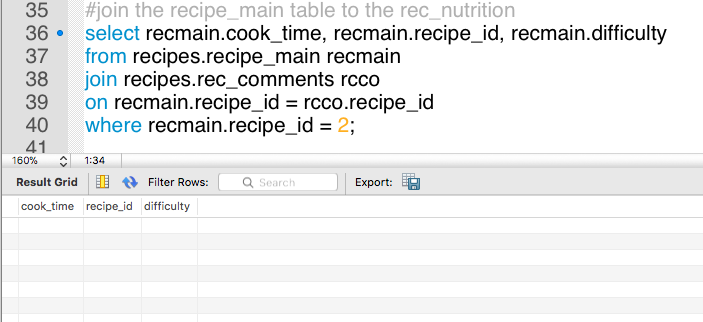
results will be more readable.

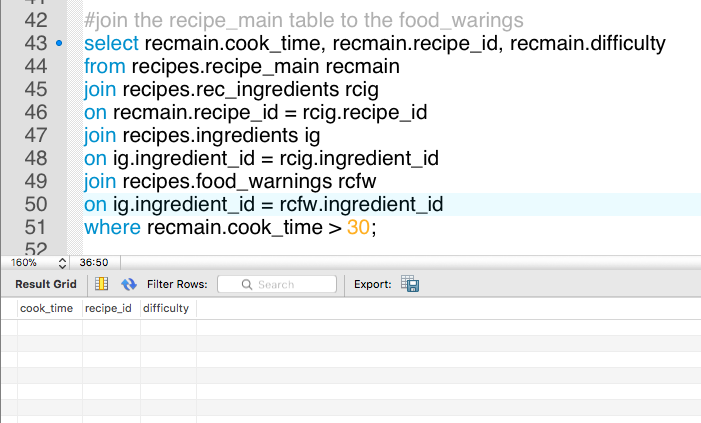
Show the results of your set of queries for one (1) recipe of your choosing.

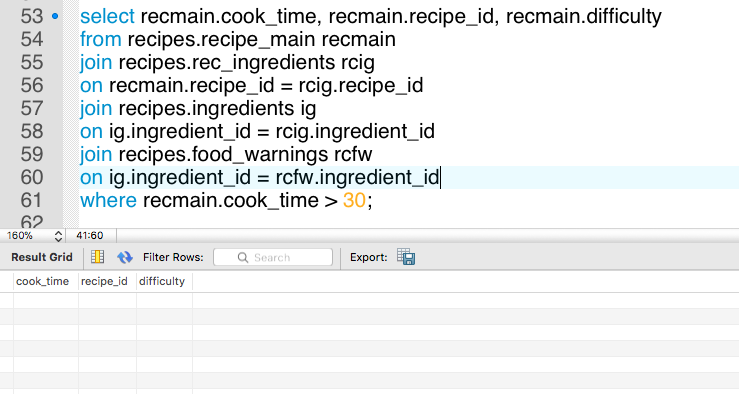




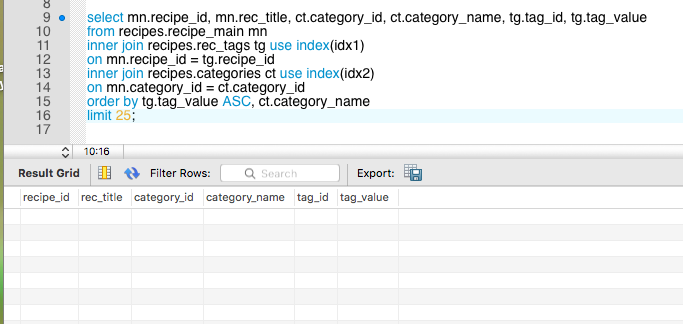








2) Write a SELECT query that would supply enough information to create an index of all recipes in the system with recipe name, category and all tag values. The index should be sorted first by tag value and then by category. Show the first 25 rows of the result of your query.



References

* Comeau, A. (2016). MySQL explained: your step-by-step guide to database design. Bradenton, FL: OSTraining.