Logo

Description automatically generated

**ALY 6030: Data Warehouse and SQL**

**Tech-Crunch**

**NUID: 002938209**

**Email: jain.abhin@northeastern.edu**

**Submitted by: Abhinav Jain**

**Date: 11/13/2022**

**Submitted to: Prof. Yoni Dvorkis**

**Introduction**

In this report, we are focusing on some of the concepts of normalization and a few SQL queries which will help in understanding how the data is handled in the organization while working on the dataset TechCrunch. This dataset consists of variables, or we can call it attributes are as follows fund id, name of the company, number of employees in the company, category type, city of the company, state in which that company established, funded date, amount raised for that company, and the currency on which that amount released, and the round of the company which depends on the concept of the company. Through this dataset, we can identify fund provided for the formation of the company to get the information of the company and funds provided to the 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.*

Ans1: According to the dataset analysis, the "Fund id" UNIQUE values will serve as the primary key because there are no NULL values observed there.

correct

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

Ans 2: It satisfied the requirements of 1 NF since each column has a distinct value according to their needs. The "Field ID" is the dataset's main key, and it links to each column. Because the data has been separated, the dataset will now be processed immediately in the second normal form.

correct

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

Ans 3: The dataset did not satisfy the condition of 2NF, because the dataset column needs to be separated to get the data into sets and split into multiple tables depending on the connecting records in each column.

For instance, since there are several categories of funds and firms, we may split the information into two tables to fulfill the two-table normal form (NF2) and provide the unique key (primary key) as fund\_id and which may have a foreign key in the company table and Company table can have company\_name as primary key and foreign key in Company category table.

correct

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

Ans 3: The dataset doesn't meet the 3NF since the dataset has to be divided and discarded in accordance with the main key. If we say that condition is satisfied for that we need to build the split dataset into multiple tables. To get the data into the normal form and eliminate redundancy, the data must be separated into linking tables with many-to-many, many-to-one, and one-to-many links. The information about the key should be represented by each attribute in a distinct table, though. To satisfy the condition of 3NF we have designed the ER diagram. correct

Question 5: Sketch a Proposed ER diagram that would bring this dataset into 3NF

Ans: Split the dataset into different characteristics so that we may connect them with the main and foreign keys to create the dataset in the normalization form, which will fulfill the requirement of 3NF. In this entity relationship diagram, the data is divided into two main categories: funds and company. This is done because the funds are allocated to the company based on the number of employees, the location of the company, and the state in which it was founded. However, we can also see how much money has been raised for the company's development and can determine the currency used depending on the region. Then, in order to meet all the requirements for funds raised, we have created another table that identifies the category of the company, whether it is web-based or software-based, and the round of the company in which category it lies.

Diagram

Description automatically generated

-6 this has the right idea but there are several issues:

1. every table needs to identify a PK

2. The PK in your table should join to an FK in the other table

3. Company can’t be a PK if fund ID is an FK that won’t work since there are by definitions more funds then companies such that company could never be unique in this Company table even though you flagged it as a PK.

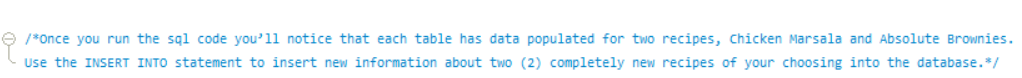
I’ll review solutions in class but feel free to speak to me after if you want more details

The ER diagram for the dataset provided by TechCrunch is shown above. In order to fulfill the three normalizations, the dataset was divided into four categories: money, fund\_raised, company, and category of the firms. We can plainly see that in this case, we built a one-to-many relationship between the funds and fund\_raised and the funds and the company, and we specified the primary key in the fund's table as fund id and the foreign key in the company table as fund id.

However, the foreign key fund id in that table, which can display one-to-many relations, is tied to the funds\_raised for the firm on the other side. Assigning the company\_name as the primary key to the business table, which is the foreign key in the company category database, creates the link between the company and the company category table and demonstrates the many-to-one relationship.

Part 2:

Question 1:



Answer: This portion created two recipes—a pizza recipe and a burger recipe—to the provided SQL database. The recipes were made from four tables—a category table, an ingredients table, a recipe ingredients table, and the main recipe—and included numerous properties. To achieve this result, more entries from each table were created, and the data were supplied in accordance with the necessary attribute to update the database with 2 new recipes.

Table

Description automatically generated with medium confidence

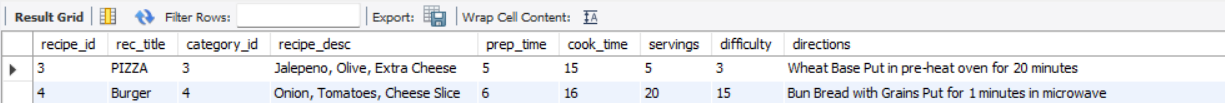
correct

Question 2:

Graphical user interface, text, application

Description automatically generated

Answer 2: In this solution, to showcase our 2 recipes which were created in the above query, using the left outer join with table 1 as recipe\_main left outer join with categories on category\_id matching in both the tables whereas define the alias recipe\_main as “r” and categories as “c”. In the second continuing left outer join is applied on the rec\_ingredients which has alias “ri” where recipe\_id are matching in both the tables on the other hand to get the join the ingredients table apply the left outer join on which ingredient alias as “i” and ingredients\_id are matching in both the tables. To get the result of our receips which were added to the recipe database by applying “where” clause where recipe\_id is greater than 2 will be displayed.



-5 there should be more ingredients displayed per recipe here

Question 3:

Graphical user interface, text, application, email

Description automatically generated

correct

Answer 3: To get the results for the recipe name, recipe category, and ingredients name, here also applied the left outer join to fetch the data from the database. In this, we have selected the recipe title, category name, ingredients name, recipe amount from the table 1 name recipe\_main which has alias r and table 2 which has alias c and furthermore table 3 rec\_ingredients, and table 4 ingredients to get the order in ascending and descending applied the order by clause where data fetch shows that category in descending order, recipe name in ascending order and ingredients name in descending order.

Graphical user interface, application, table

Description automatically generated