Introduction to database system : Deliverable 1

Introduction

A database system is a software application that allows users to store, organize, and retrieve data. In this project, we will be designing a database system for businesses, including their attributes and schedules. Our goal is to create a system that is efficient, easy to use, and can handle a large amount of data.
To achieve this goal, we have made several assumptions about the data. For example, we assume that friends are not always a reciprocal relation, and that a business can have several overlapping opening hours in a day. We also assume that there are only six attributes for a business, which we call categories, and that there is no fixed number of category names possible for a business.
In this report, we will present our Entity Relationship Schema and Relational Schema, as well as discuss our data cleaning and transformation process. We will also provide general comments on our team's dynamic and communication during the project.

Assumptions

1) Friends is not always a reciprocal relation. 2) A business can have, the same day, several and overlaping opening hours. Eg an hotel can have the breakfeast starting at 7 and the pool opening at 8 while the breakfeast is still opened. 3) There are only 6 attributes to a business : business parking, good for meal, ambiance, noise level, music, and dietary restrictions. We called these attributes categories. 4) There are not a reasonable fixed number of category name possible for a business. 5) In business hours, open now is not a relevant feature because it depends of the time data have been extracted. That's why we don't mention it in our work. 6) A business could have no attributes. 7) A business could have no schedule.

Entity Relationship Schema

Schema

Description

- Elite is associated to a user, so it is a weak regarding users table.
  A tip is given by a user to a business, so it is weak regarding these two entities.
  Same reasoning for review table.
  A schedule is dependant of a business, so it is also a weak entity.
- The FRIENDS table should be a many-to-many relationship, due to assumption 1
- We splitted the weekly schedule of a business into daily schedules, to make it more exploitable. As a business could open and close twice in a day (eg a restaurant), we created an Schedule ID (SID) to identify each time slot. This manages assumption 2.
- We didn't do a ISA for CATEGORIES of a business because there are too many possibilities for each one (assumption 4)
- Assumptions 6 and 7 justifies the thin lines between business and 'has' linked to ATTRIBUTES and SCHEDULES tables

Relational schema

- CATEGORIES table have for each business the same number of raws that the number of categories associated with the business. This will create a long table with PK (Bid, Category) but very practical to ask queries related.
- Weak entities and their relationship table to their owner entity can be merged into one table in the RM. eg. Was-Elite, Rates-Review, Gives-Tip, Has-Schedule.
- For all entities, when an attribute is a key (usually a primary or foreign key), it has a condition NOT NULL. There are no conditions on other non-necessary attributes.
- The ATTRIBUTES table pertains to the six categories of attributes that are associated with a business. Each category contains multiple attributes. Initially, we considered implementing attributes as an ISA relationship, but ultimately opted

for a simpler approach. All relevant information is consolidated within the ATTRIBUTES table, which is considered weak. For instance, each attribute that a business possesses is represented by a row in the table, containing the business ID, attribute category, and attribute itself. Since no two attributes are repeated across the six categories, the combination of business ID and attribute can serve as the primary key. Although this results in a lengthy table, it is highly efficient for all types of queries related to the attributes of a business.

- A benefit from some of our design choices as for CATEGORIES and ATTRIBUTES tables is that we don't have sparse data, or to store a lot True/False boolean. We just keep information that is relevant.

Data cleaning and transformation discussion

- Replace nulls/nones with 0's or NA's when applies (e.g counts).
- Remove exact duplicates if found.
- If not an int => put a 0
- Ignore non essential attributes that are null.
- We used for address, schedule, text of the tip, open and close time, etc CHARVAR to allow flexibility.
- Split attributes or cat names from dictionary to smaller strings

General comments

The team had a good dynamic, we all worked well on it. We split the work into 3 (design ER, write the RM, and write the report), so we had to communicate a lot to avoid misunderstandings.