CMS320 Pr.1: The Case for a Relational Database

Mark Wagner

Dear Client,

First off, thank you for hiring us to construct your fleet management software! Over the course of operating a rental car business, you will face many different types of people, different vehicles, and a myriad of situations that can occur and result in the generation of extra information that needs to be recorded, maintained, and eventually reported to insurers, prospective buyers/investors and state and local governments. Every interaction results in the creation of data, and every piece of data is in some way tied to another... Accident X involved Driver Y and Vehicle Z. In the past, for very simple constructs of linear data (looking at it from only two dimensions) a spreadsheet would have sufficed. This is no longer the case. Here are some reasons why.

Picture a square. It has a height and a width, just as a spreadsheet has columns and rows. It also has references, but without immense complexity, those references will still remain in either a column or a row. Similarly, you have vehicles and dates to occupy those columns/rows to record (for this example) income. Now as we know, this industry is not without turnover. Clients come and go, or maybe even switch cars. So you now can no longer record the vehicle in ‘A’, the car in ‘B’, and multiple dates in ‘C’ onward. You need to add a **third dimension** of depth. So let’s say you create a separate sheet for each month and assuming you have no mid-month turnover, this works for your vehicles. But now you have a problem, you are having to create many sheets over time, the size of the file is growing, each time you share it with someone, you must exercise version control, when you add vehicles or guests – every sheet must be updated individually. But you are moving along.

Then, an accident happens and extra insurance information needs to be stored on a vehicle. This accident puts you into a fifth dimension. You are beyond the functional capabilities of a spreadsheet. You need to be able to look up what parts were damaged on vehicle X in year K and find if guest 241 or 376 was involved and what his or her claim number was! You need a database.



A relational database put simply, is software that uses the processing capabilities of a computer to tie objects together using thousands to millions of different link references or relations. In the past, this required warehouses of paper and hundreds of employees to manage [Crosley, 2017] but now, it can all be done by means so advanced, many software engineers don’t even know specifically how it worksat the algorithmic level. Fortunately, the Relational Database Management Systems (RDBMS) software existing is so good, that designers don’t have to understand the inner workings in order to make you a robust, dynamic system for managing your data. It can even be integrated with your website so when someone books, their information is captured and added to the table (similar to an individual spreadsheet) where it belongs.

When an event happens (registration, rental, accident, repair), the website will send a command to the RDBMS to make a change to a table and manipulate (add, change, or remove) a piece of information. Each table can be visualized like an excel spreadsheet, but with some of the rows digitally referencing another spreadsheet in the background. The database is stored centrally on a server, which can be duplicated so that data loss will never occur. It can even be hosted on a small instance hosted by Amazon, saving you office space and providing enhanced security, performance and safety. You can even grant different levels of modification power to different users so that only managers can modify car properties but associates can change guest information. Credit card information can be protected from view by certain user groups, and while changes to or queries from the DB can be made directly using something called Structured Query Language (Indiana University, 2018), user-facing GUIs can be made to use that language automatically so the layman can add change the database without typing a thing!

When an associate adds a guest to a vehicle account, not only will the guest table increase by one, but the vehicle table will have an additional booking which, if queried, would reference back to that guest and so on and so forth. This has worked tremendously well for similar industries like Supply Chain Management (White, 2014) and a nearly identical model: the Hotel Industry. Rather than Guests, Vehicles, and Repairs, hotels record Guests, Rooms, and Renovations but the basic premise is the same: a Relational Database ties all the information about your business together based on those specific individual relationships between pieces of information, making them accessible to the human users behind your business.

[1] Crosley, T. (2017, August 29). How did people use to store data before the first generation of a computer? Retrieved January 23, 2019, from <https://www.quora.com/How-did-people-use-to-store-data-before-the-first-generation-of-a-computer>

[2] IU, I. (2018, January 18). What is SQL, and what are some example statements for retrieving data from a table? Retrieved January 23, 2019, from <https://kb.iu.edu/d/ahux>

[3] White, D. (2014, December 2). Dealing with Supply Chain Big Data. Retrieved January 23, 2019, from <https://logisticsviewpoints.com/2014/12/02/dealing-with-supply-chain-big-data/> Logistics ViewPoints