Spatial Big Data Management: Report

Student Number: 13173421

# Part A

The design for this relational database begins with an analysis of core entities and how they relate to each other. In this database concept there are four main entities to be considered: Clinics, doctors, patients and diseases. Understanding the relationships between these four entities identifies links between them. After identifying the entities and defining relationships between them an Entity-relationship (ER) model is created to visually plot their relationships (figure). This database utilized two of the main types of relationships, one to many and many to many.

Each entity typically contains attributes that help describe some of their characteristics. When creating an ER diagram the attributes are interpreted from the information given by the client and some can be unique.

When translating the ER diagram into the database schema, the entities can be thought of as the table names and the attributes are the columns that make up the table. The tables are then populated with the information provided about the attributes of each entity. This database also required an extra table to be created in order to accommodate a “many to many” relationship occurring between the patients and diseases. This table takes the primary keys from those tables and creates a unique primary key pair that allows for the two tables to be related. Similarly, the clinics and doctors share a “one to many” relationship, as well as the clinics and patients. To account for this, attributes are created as ‘foreign keys’ in the doctors and patients table that reference back to the primary key of the clinics table.

Several assumptions were required due to ambiguity or vagueness in the clients’ language. It is never made clear if the patients were visiting multiple clinics or just one. If this was interpreted to be a “many to many” relationship instead of “one to many” it would facilitate the need for another new table to create a primary key pair. There is never any relationship created between doctors and patients as well. Multiple doctors can work at the same clinic, so even when patients are related to a specific clinic it is not certain which doctor(s) they may have visited.

# Part B

This spatial database contains information about major hurricanes that effected the gulf states in the USA. Each hurricane contains many attributes that help explain its devastation on the region and the path it took. The gulf coast states are defined as the states bordering the Gulf of Mexico. They are made up of Texas, Louisiana, Alabama, Mississippi, and Florida. Within these states certain cities were chosen that were most heavily affected by the hurricanes; they were either the city where the hurricane made landfall or in a nearby location. Each of these hurricanes were forecasted by meteorologists as they were happening. The same storm could have multiple people forecasting it and each forecaster could have forecasted different storms in their life. They are assigned an accuracy percentage based on how well they forecasted the path, landfall area and intensity of the hurricanes. In the aftermath of the hurricanes, there were a contingent of people who were generous enough to donate money to the cities that were affected. Each city could receive donations from multiple people, but each donation could only go to one specific city.

With this database, many simple and complex queries can be made regarding hurricane activity in the gulf. It is possible to plot all the hurricane tracks and label them by category in order to show the paths they took (Figure 9), and which states may have encountered more hurricanes and what their intensity was (Figure 10). Monetary and death or injury information can be used to determine which states had the most population effected and how much money in damage the storms cost. Accuracy analysis of each meteorologist shows which meteorologists performed the best overall, or on individual storms. Sometimes there is a correlation between a storm’s forecast accuracy loss of life or injury. This type of database could be of interest to insurance companies to predict areas which are more prone to being hit with hurricanes and the amounts of damage caused. It can also be used by meteorologists or climatologists to better understand how hurricane paths and intensity vary year to year and enabling them to draw conclusions as to causality.

There are a few limitations to the database as it is currently designed. One issue is with the death or injury statistics. That number could be further detailed into more specific information about each city or state that had deaths. Each storm can pass through multiple cities and states, so one overall death toll doesn’t necessarily depict which states may have experienced more loss of life than others. Hurricanes cover very large areas and can be hard to define in the SQL geometry. Their paths move in line-strings but due to the sheer size of the storms they could be classified as polygons. This could increase the number of cities and population affected by each storm. This can be manually accounted for by the hurricanes\_cities table but is hard to represent spatially with the given geometries.

# Part C

The content of “university\_applicants” contains fictitious application information from throughout the United Kingdom. Creating a geometry column using the Easting and Northing columns relates this database spatially to the “uk\_lad” and “English\_regions” databases. This occurs by using an ST\_Contains or other similar function. The ST\_Contains function returns TRUE for the case when a point falls entirely within a polygon. Given that the university applicants table is in points and the region-based tables are polygons, this makes it suitable for doing a spatial join.

The ethnic and gender information in the applicants table make it possible to show relationships about where specific subsets of applicants are from. In the case of gender, points can be produced showing how the male applicants in the UK are distributed (figure 1). An interesting view of ethnic groups involves the distribution of each group. This shows where different groups may be clustered outside of a major metropolitan area like London (Figure 2). It is also possible to aggregate application information for regions or LAD’s in order to rank which areas had the most or least applicants. In Figure (3) the three regions with the lowest number of applicants is highlighted. Using this same technique can determine the 10 LAD’s with the greatest number of applicants (figure 4) and the least (figure 5). Lastly, the table can be used to combine both gender and ethnic group information, and then aggregate by region to determine which English region had the most female and non-white applicants outside of London (Figure 6).

# Part D

While there is no formal definition for “Big Data” it is commonly categorized in what is referred to as the 3 V’s model. The V’s in this model are constantly being updated to better understand Big Data, but the most commonly referred to ones are Volume, Velocity, and Variety (Edosio, 2014). Volume refers to the vast amount of data created every day. As of 2019, 2.5 quintillion bytes of data are produced every day and 90% of the data on the internet has been created since 2016 (Schultz, 2020). Velocity refers to the speed at which data can be generated and then analysed, while variety refers to the heterogenous nature of the data being consumed (Edosio, 2014).

The V’s point to several limitations with Big Data. With the ever-increasing volume of data being brought in, data storage space and scalability are harder to manage. Data can also come from a large variety of sources and be structured or unstructured, which can create difficulties with extracting meaningful information from the data. The speed at which the data arrives and requires processing can vary. Some applications require batch processing while others may require continuous, real time analyses (Assuncao, 2014).

Relational Databases work well for structured data that has strong consistency, but Big Data doesn’t maintain these characteristics. With the explosion of Big Data, it has been necessary to create new types of databases to store this new type of information (Fowler, 2012). These databases are commonly referred to as NoSQL (Not Only SQL) databases. The advantage of these databases is they allow for more flexibility in how the data is stored and scaled, usually at the sacrifice of querying capabilities (Funck, 2011). NoSQL databases store and allow for access of data using data models, such as key-value stores, document-based stores, graph databases, and column-oriented stores (Funck, 2011).

The “hstore” table is an example of a key-value store. One advantage of a key-value store is it is schemaless (Gessert, 2017). By not having a defined schema it allows for flexibility in how the data is stored and changed. Relational databases are required to have a defined schema that the data must follow as it’s inputted. The unstructured nature of Big Data makes it difficult to adhere to a predefined schema. The key-value structure is limited in that it can only support basic operations. If there is a need to do more than basic create, read, update or delete queries the data will likely be analysed inefficiently (Gessert, 2017).

The JSON table is an example of a document-based store. While document-based stores still contain key-value pairs, the keys must be unique, and each document contain a key “ID” (Funck, 2011). In addition, more complex data can be stored in the form of nested arrays and objects. This allows the user to retrieve an entire document by its ID or create more complex queries and pick parts of the document (Gessert, 2017).

These different data models help determine how the data is stored and how it’s accessed, but another factor that is important to consider is the consistency of the data (Fowler, 2012). In 2000 Brewer (Brewer, 2000) postulated the CAP theorem which states that there is at most two of these three properties at any given time: consistency, availability, partition tolerance. For a database revolving around social media or blog posts availability is likely more necessary than consistency. Determining which two properties of the CAP theorem are most important and combining that with a data model that makes sense for the company, will produce a range of NoSQL database systems from which to choose.

# References:

Assuncao, Marcos & Calheiros, Rodrigo & Bianchi, Silvia & Netto, Marco & Buyya, Rajkumar. (2014). Big Data computing and clouds: Trends and future directions. Journal of Parallel and Distributed Computing. 75.

Brewer, E. (2000). *Towards robust distributed systems*.

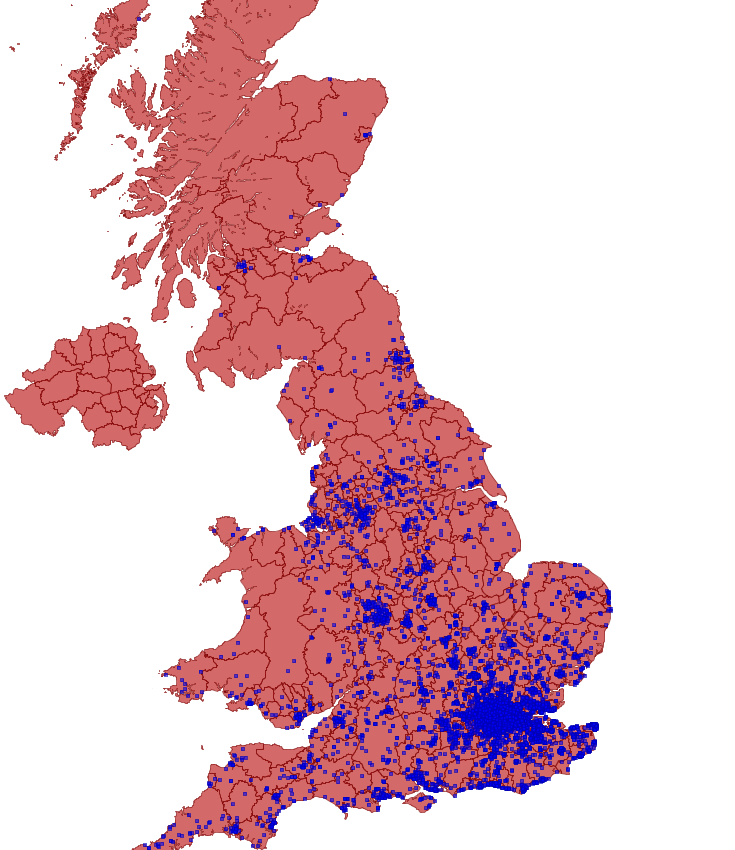
Edosio, Uyoyo. (2014). Big Data Paradigm- Analysis, Application, and Challenges.

Fowler, M. and Sadalage, P. (2012). *NoSQL distilled*. Boston, Mass.: Addison-Wesley.

Funck, Robin & Jablonski, Stefan. (2011). NoSQL evaluation: A use case oriented survey. Proc 2011 Int Conf Cloud Serv Computing. 10.1109/CSC.2011.6138544.

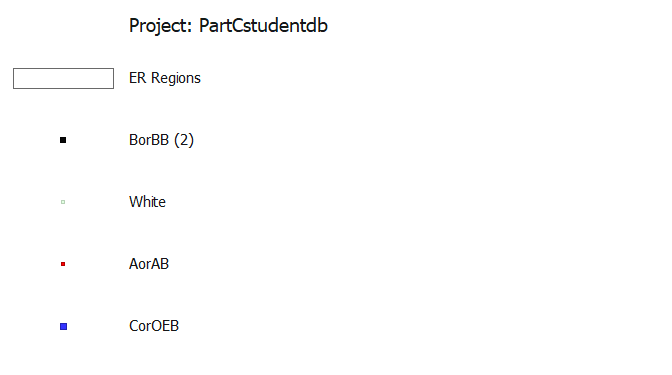
Gessert, F., Wingerath, W., Friedrich, S. *et al.* NoSQL database systems: a survey and decision guidance. *Comput Sci Res Dev* **32,**353–365 (2017) doi:10.1007/s00450-016-0334-3

Schultz, J. (2020). How Much Data is Created on the Internet Each Day? | Micro Focus Blog. [online] Blog.microfocus.com. Available at: https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/ [Accessed 15 Jan. 2020].

**FIGURES:**

**Distribution of Male Applicants**

Figure 1

****

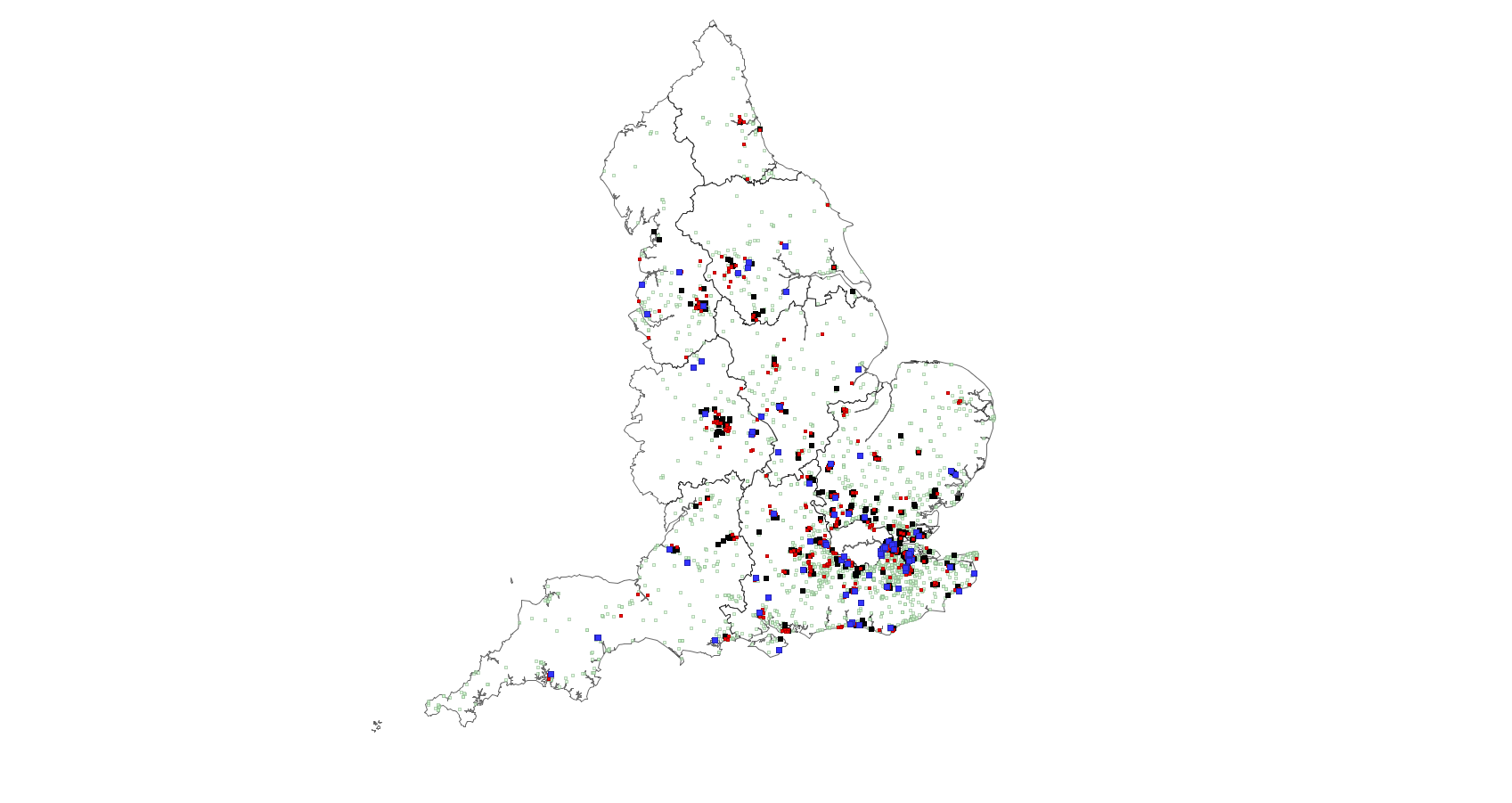
**Applicants Distribution Outside London**

Figure 2

**Regions with Lowest Number of Applicants**

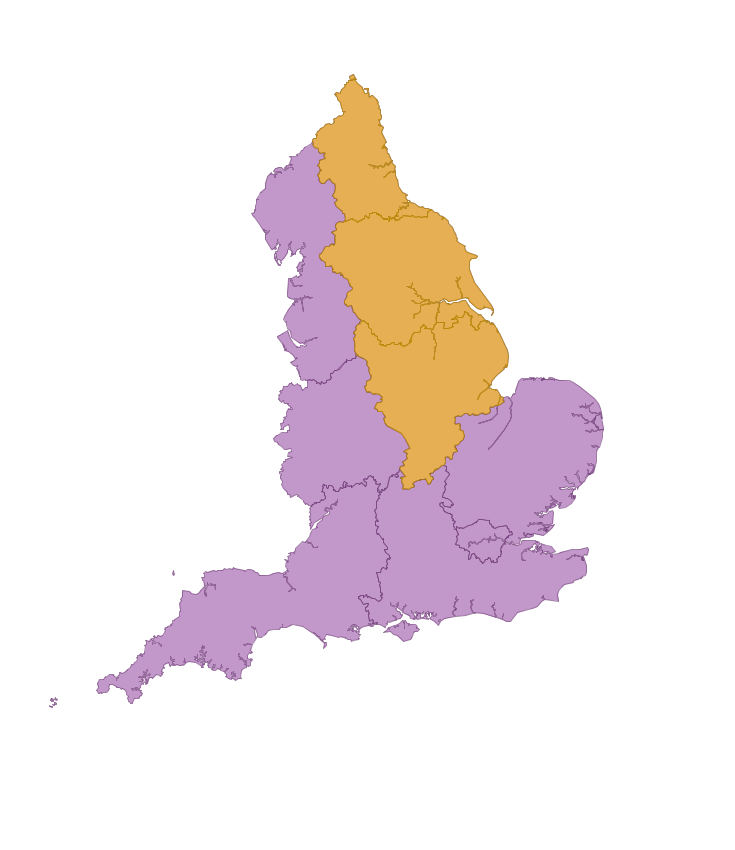
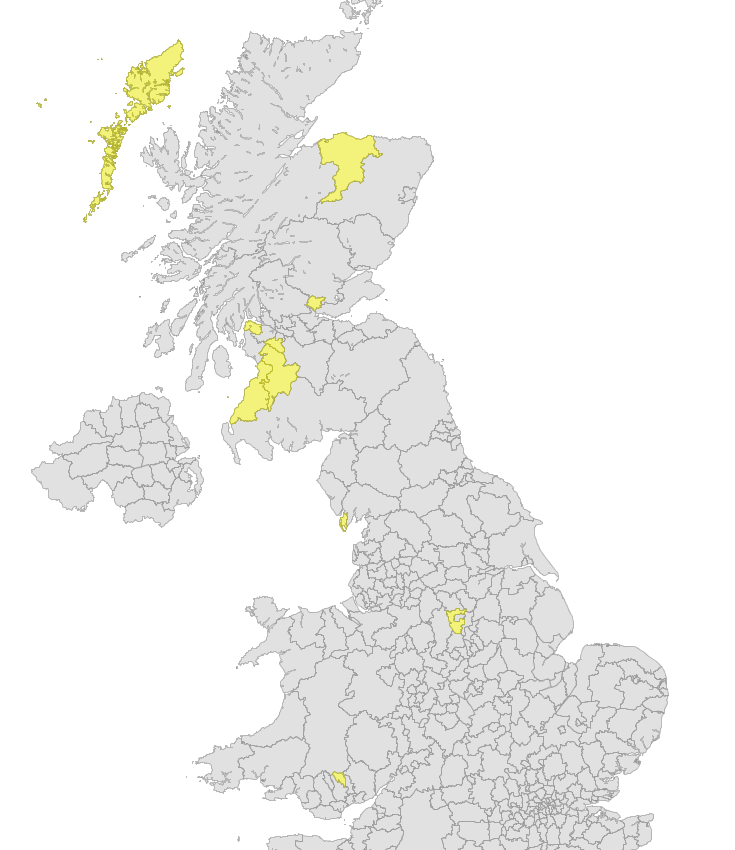
** (highlighted in orange)**

Figure 3

**Top and Bottom 10 LAD’s (number of applicants)**

** Bottom 10 Top 10**

*Figure 4 Figure 5*

**# of Non-White Female Applicants Outside London**

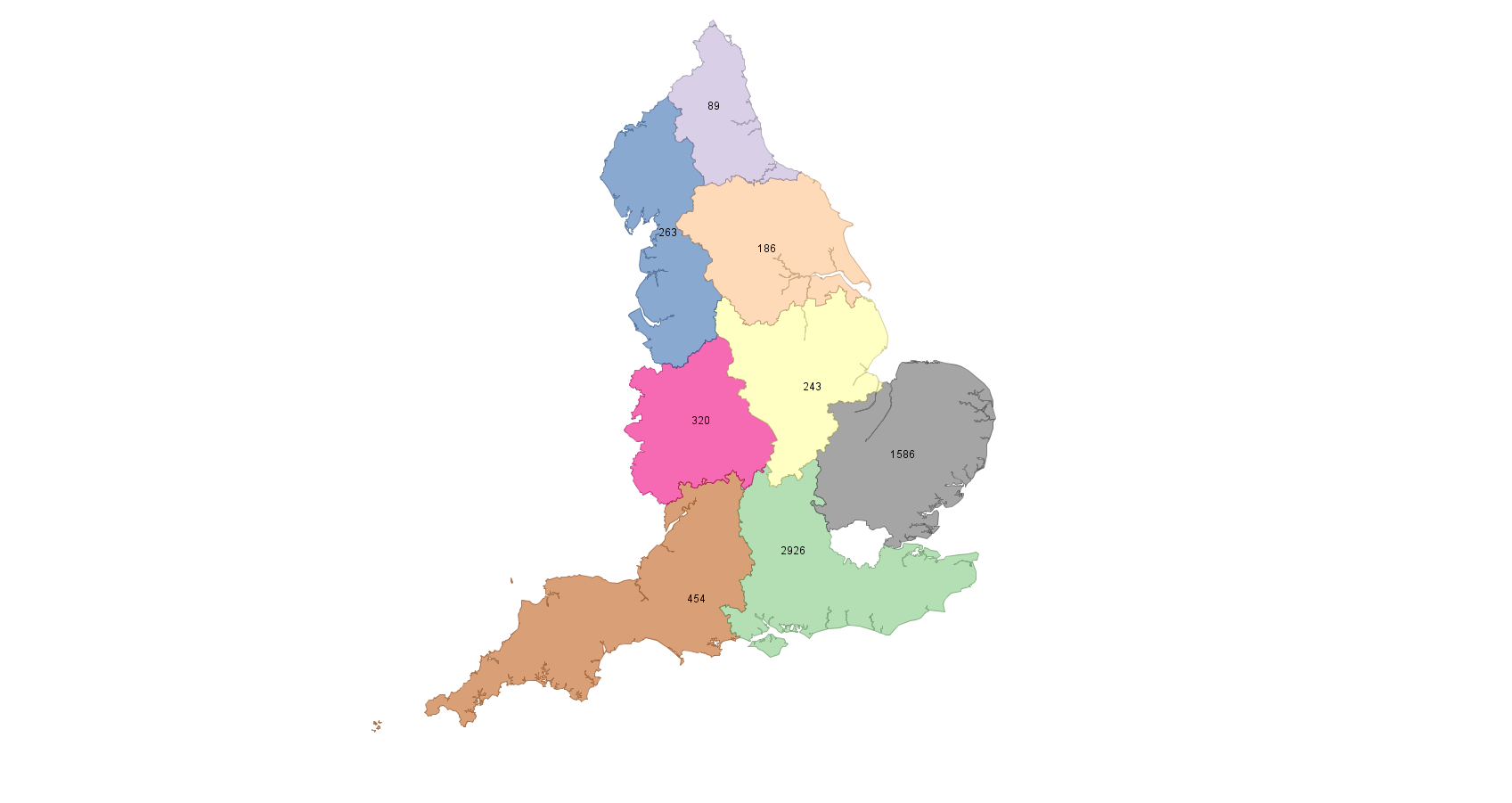
 (by region)

Figure 6

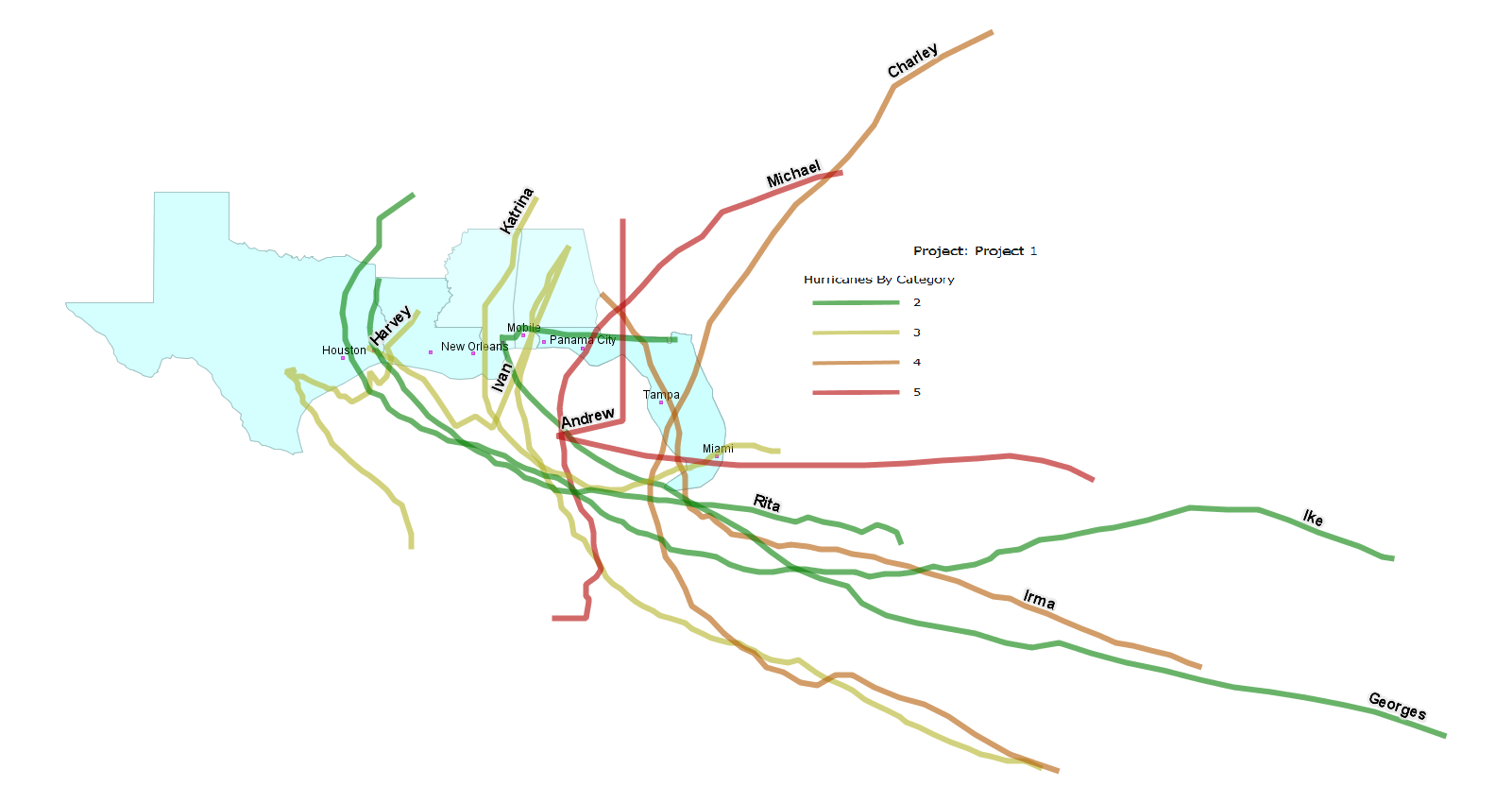
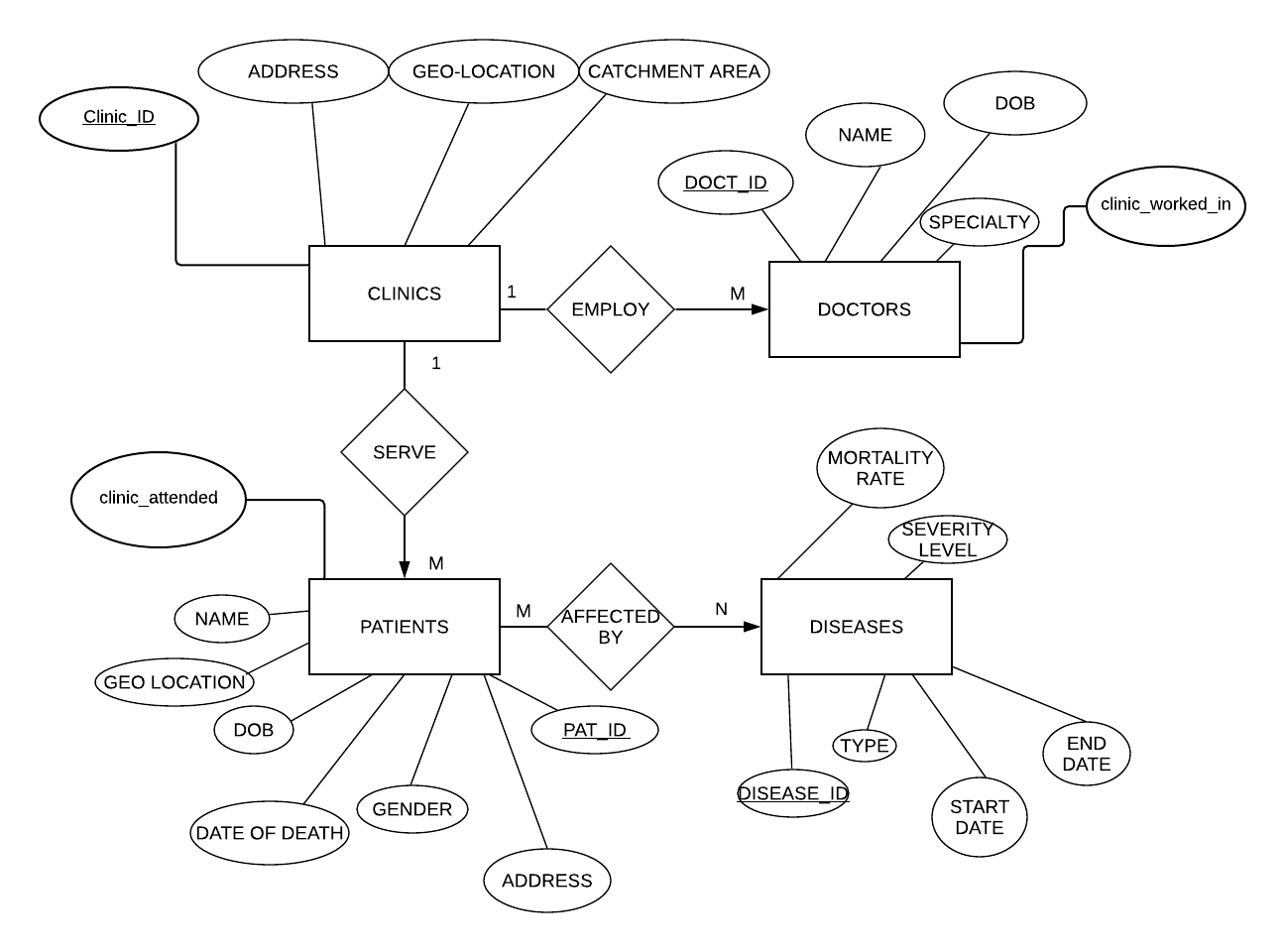
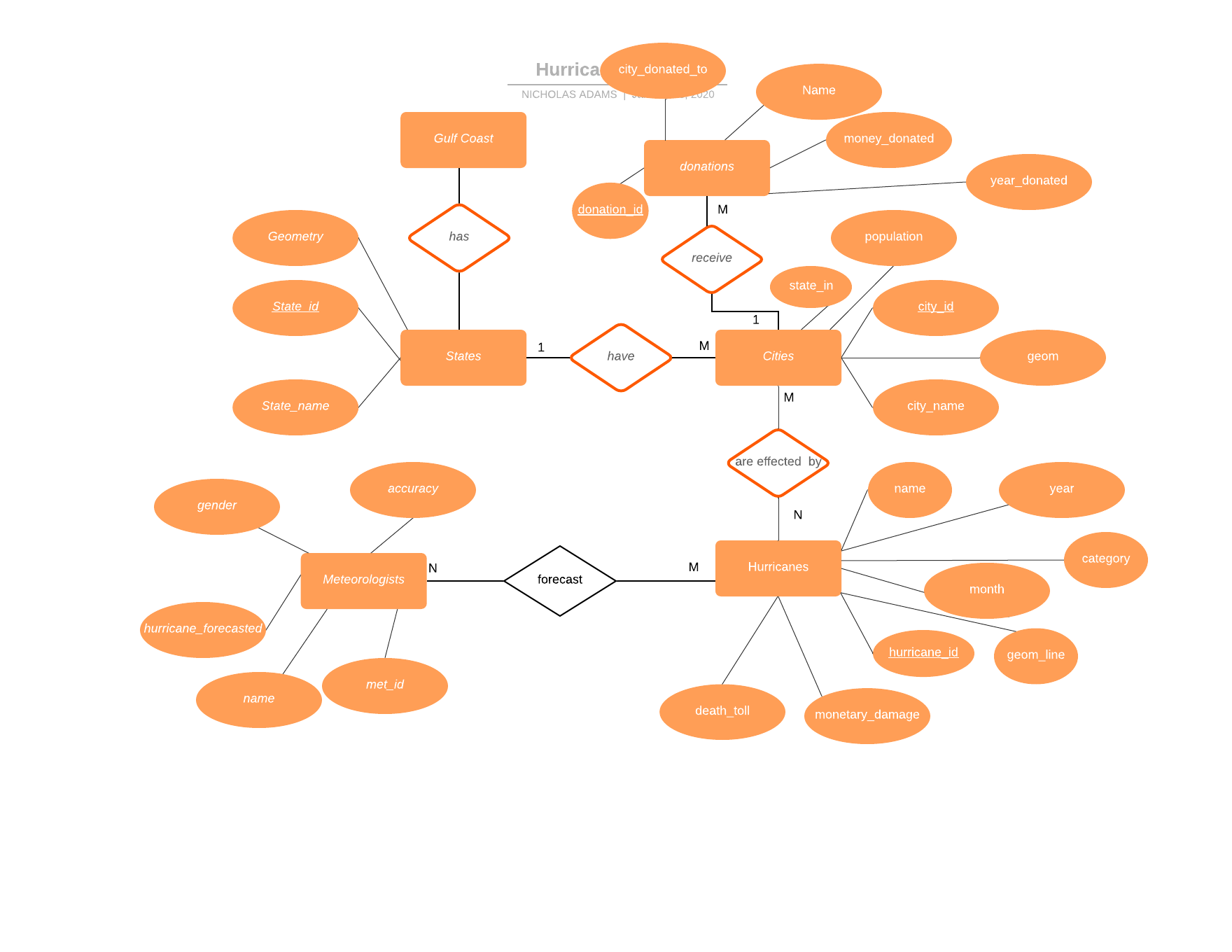
 **Hurricane Paths and Categories**

Figure 9

 **Hurricanes Impacting Mississippi**

Figure 10

**PART A ER DIAGRAM**

**PART B ER DIAGRAM**