MEMORANDUM

TO: Manager

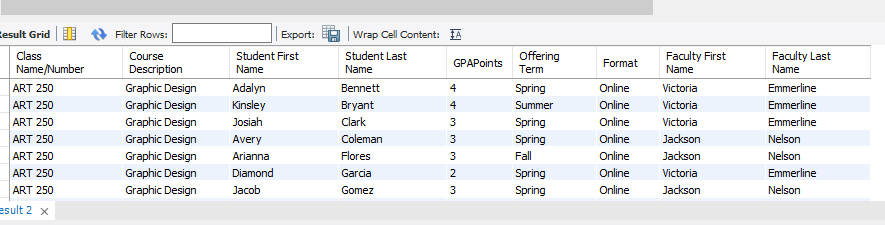
DATE: January 31, 2021

FROM: Chief Data Officer, Melissa Hunfalvay

DEPT: Research and Development

Dear Manager,

This memo is in response to your request for a flat file (Excel, CSV file) of student data. Please find this data file appended to this memo. You can also see a portion of this data below, with the first 5 rows displayed (see Figure 1).



It is my understanding that you are looking for the following:

* A simple way to do business
* A streamlined way to process data
* An understanding of data in “near” real time, that is, we need to obtain, understand, and interpret meaningful outcomes from the data quickly and accurately. This will inform business decisions now, not months from now.

Assuming I have clearly understood your requirements, I would like to provide some information about ways we store, retrieve, analyze, and understand the data. However, before discussing these options I want to provide a summary of our data, the database, the type of data and structure of data. By understanding our data, we can create the correct structure and database design to meet your requirements.

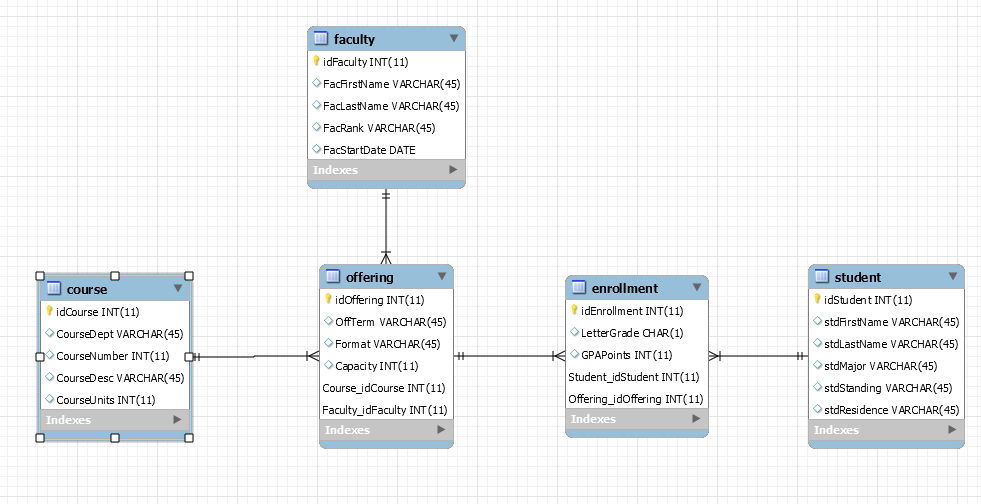
**The Data**

The data in our university database is quite complex. Figure 2 shows you the database schema. Database schema is a visual way to view the database like a blueprint of how the database is constructed.

There are four tables (or entities) for example, the faculty table, the offering table. There are many attributes which form columns within these entities, for example, within the faculty entity are the attributes of the faculty members name (FacFirstName, FacLastName).

Between each of these entities are relationships. For instance, the relationship between faculty and offering shows there is a relationship between these two tables. This means each offering is assigned to one and only one faculty member and each faculty member is assigned to one or more offering.

For the most part our data is structured, meaning that the data is predictable with a regularly occurring format. Some examples of this are the course number and course department.



Relationships

Attributes

Entities

Figure 2: University Database Schema

Now I have explained our current database I would like to address your specific request for a streamlined way to process data. You had requested a flat file, which is one option. Another option is a relational database, which is our current structure.

**Flat files**

*Description:* What is a flat file? A flat file is one that has no internal hierarchy. In other words, the schema you see in figure 1, does not exist for flat files. The most common types of flat files are csv files and excel files.

*Pros of a Flat File*

Flat files are:

1. Easy to set up
2. Ideal for smaller databases
3. Great if the data is less complex (that is, no relationships between the data)
4. Have relatively easy sorting and searching tools

*Cons of a Flat File*

The disadvantage of flat files that relate to our University data are:

1. There is a lack of security, meaning that anyone can open and see what is in the file. This is especially problematic for our University data as we have Health Insurance Portability and Accountability Act (HIPAA) data within our database e.g., names of faculty and students. For more information on this see Chickowski, (2010).
2. Flat files are designed purely to manage single file databases. With such a product you cannot build a multi-table database. You can create more than one table. However, you cannot link the tables, or more specifically, you cannot link the information within the various tables. This is important to our University database because it means our flat file needs to contain everything, making it very large and hard to read. It also means that as data is added this problem grows exponentially.
3. To add more data, you need to pull it into the datafile again. This can be done by manually adding it. The primary problems with manually adding data are a) input errors, 2. Reduced speed and efficiency. With large amounts of data this is time consuming and inefficient to do by hand.
4. Data cleaning and organizing becomes a problem that grows exponentially with the use of flat files. Cleaning or organizing must be done again with new data. This is where most of the time and effort in understanding the data occurs, therefore data analysts would spend most of their days updating, cleaning, and organizing data rather than understanding the data. This will limit the time and value we could spend using the data to help inform business decisions.
5. By delaying the insights into our data, we further reduce the usefulness of data that is time sensitive. For example, when Professor Emmerline retired right before the Spring session, we needed to immediately contact all the students from her class to inform them of a different professor. This task would have taken hours, if not days, to search through a flat file and find the classes that Professor Emmerline taught, the students within them and then to make sure we only called the same student one time.

In summary, flat files are advantageous if the data is less complex. However, the reason flat files are not a good option for us is because we have a) we need to keep our data secure, b) we have a large volume of data, c) there are lots of different elements/variables and inputs into our data with multiple different tables (entities) that need to be connected to fully understand our data. Furthermore, spending time on cleaning and organizing delays insights causing a time gap between what we should do with the outcome of our data.

**Relational Databases**

*Description:* What is a relational database? A relational database is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables (<https://www.oracle.com/database/what-is-a-relational-database/> ). As a point of interest, in the development of data, the relational database was a huge step forward in how we architect our data. Prior to this development data history had been focused on storage, rather than architecture (Inmon & Linatedt, 2015).

*Cons of a Relational Database*

The disadvantages of a relational database are:

1. In some cases, it is overkill. In other words, for data that is small then a flat file is more appropriate. For instance, home finances, such as mortgage, electrical bills, are easily tracked on an Excel spreadsheet and a relationship database is not needed.
2. Relational databases take longer to set up than a flat file
3. If the data does not have many relationships, that is if the data is not complex across many associations, then a relational database is not needed.

*Pros of a Relational Database*

The advantages of relational databases for our University data are:

1. Better security with access levels that password protected and are controlled by the database manager. The database manager can also track access and changes to the database.
2. Data consistency: all the data is in one place, therefore, changes made get updated to the database at one time across the entire platform. Unlike flat files where there may be multiple versions.
3. More efficient data updates, as the data is only needed to be updated in one column not multiple places.
4. Quick insights into complex questions because programming code can be added, stored, repeated, and even automated to reveal insights at any time.
5. Multiple scalable queries at one time, this means that multiple people can be working on the database at any one time and they are all working on the ‘same version of the truth.’
6. Scalability of the database. As our data grows and changes, we can adapt with a relational database.
7. Data validation: queries can be made to validate input into the database to entire the data is accurate.
8. We can easily sort and manipulate data in a relational database by asking questions (queries).
9. We can provide management flexible reports both on paper and on screen. When coupled with programming we can even update these reports automatically as new data is added giving management real time insights and providing informative, easy-to-understand reporting.
10. On a final note, the Research and Development Department includes experienced data analysts. I have worked in past companies where we did not have the correct database infrastructure and the analyst were spending most of their time cleaning and organizing the same data time and again. They became very dissatisfied with their day-to-day jobs. Our employees are the life of our company. And although we cannot always please their needs, this is an opportunity where I believe we can satisfy the data analyst while also improving company output. We can also reduce cost by creating efficiencies and providing better insights from our data.

If interested you can read more about relational databases, including their benefits here: <https://www.oracle.com/database/what-is-a-relational-database/>

**Recommendation**

*“Data is fairly worthless unless it can be analyzed”* (Inmon & Linstedt, 2015).

To summarize, because we have complex data that has relationship between the data, a large volume of data that needs continual modification (additions and deletions), and we want to use our data to provide valuable insights I recommend we continue to use our relational database. This will provide the streamlined and simple way you require us to do business.

If interested, you can find more information and resources <https://www.ibm.com/cloud/learn/relational-databases>

Should you have further questions or concerns please do not hesitate to ask.

Thank you for your time and consideration of our data needs.

Melissa Hunfalvay

**References**

Chickowski, E. (2010). *Flat-File Databases Often Overlooked in Security Schemes.* Retrieved from: <https://www.darkreading.com/risk/flat-file-databases-often-overlooked-in-security-schemes/d/d-id/1133363>

IBM Cloud Education (2019). *Relational Databases.* Retrieved from: <https://www.ibm.com/cloud/learn/relational-databases>

Inmon, W.H. & Linstedt, D. (2015). *Data Architecture: A Primer for the Data Scientist.* Elsevier, Boston: MA.

Oracle (2021). *What a Relational Database Is.* Retrieved from: <https://www.oracle.com/database/what-is-a-relational-database/>