Data vs Information in Banner

As I work with ORACLE's Banner every day as a user, I thought why not use that for this essay. It is, however huge and used by every department here at Marist in some capacity. Even the portion that is used in my department is too large to list all the elements of "data". You did mention Banner and grading in class, so, again why not. Banner is relational and is composed of tables of rows and columns.

The elements used in grading are from different tables. I have only listed some of the tables and some of their elements as I don't know all of them. From a student information table are each student's first, middle, and last name and CWID; from another table is the semester; from a course table are the department, course number, and section of each class; from an instructor table are each instructor's first, middle and last name; and from a grade table is each grade a student earned in each class.

This and other data is organized into information in many ways. I will only name a few, as this data is used by more than my department and could become extremely detailed. Each class a student takes is shown on their transcript, on their schedule, and on their degreeworks. When an instructor is ready to grade a class, they are required to login to access it. Once the class is graded, the grade is calculated into the student's GPA, the grade is shown on the transcript and in degreeworks, and the class is shown as completed. That is a very simple explanation of grading, as there is much more data that is actually entailed in the process. Grades and GPAs are used by other departments to ascertain if a student needs to be academically dismissed, placed on the Dean's list, graduate summa cum laude, etc.

Without other data related to the CWID, whether it's an address, a name, a grade, an amount of money, a class name, or any other data, the CWID is completely meaningless. Just giving a list of grades doesn't mean much. If a student's transcript only listed their grades without the class, how would an employer know what types of classes the student took? Giving

"context" to any given data allows us to better understand that data. Data organized into information in a transcript allows us to see that a student took business courses, which would be a plus for an employer in a financial field.   Any of the data elements that I listed above left alone in a list is also meaningless without more information about it.

Data Models

Hierarchical Data Models are one to many or one to one relationships that can be visualized as a family-tree structure with one root and many branches and subdivisions based on parent child relationships. They are efficient to navigate, **if** you know the structure. They can be self-documenting, **if** you know the structure. Without programming, it is difficult to enforce the rules. There is possible inconsistency due to duplication, they are not physically independent, and there is no standalone query interface.

Network Data Models allow for many to many relationships which eliminate the duplication inconsistencies. They are still largely programming based, however. They are physically independent and use CODASYL, a data-manipulation language. There are no inconsistencies in inserting, updating or deleting data. For both the Hierarchical and Network models, if an available "item" is not in use, it has no place in the structure of either model and this breaks the Law of Least Astonishment.

Relational Data Models have one to one, one to many, and many to many relationships. They use English like query languages such as SQL to manipulate data. These models also have no inconsistencies in inserting, updating or deleting data. More flexible than the other two models, as they can: provide for queries not previously programmed, can combine data from different sources, and can add new data and records without disturbing existing programs and applications.

As for XML (eXtensible Markup Language) as a model for data storage, it is Hierarchical and would have the same problems. It also is sequential, slower to search as it is not indexed in any way, and is not secure because it is easy to change data. It is capable of using large collections of small documents as a database.   It has data independence, as it is not tied to specific software or operating systems. XML would be good for certain applications such as

files used for static html pages, caches, archives and logs as it appends data at the end of the file.

pgAdmin 4 | ← → | pgAdmin 4 | ✕

🐘 **pgAdmin 4** File ▾ Object ▾ Tools ▾ Help ▾

🔍 Browser

⊞ 🗄 Servers

🚲 Dashboard ⚙ Properties 📄 SQL 📈 Statistics 🖱 Dependencies ependents

Preferences

- ⊟ Browser
  - Display
  - Nodes
- ⊟ Dashboards
  - Graphs
- ⊟ Paths
  - Binary paths
  - Help
- ⊟ SQL Editor
  - Display
  - Explain
  - Options
- ⊟ Storage
  - Options

Items per page in grid

Kathy Coomes

The number of rows to display per page in the results grid. Value should be between 50 and 2000.

Query info notifier timeout

5

The length of time to display the query info notifier after execution has completed. A value of -1 disables the notifier and a value of 0 displays it until clicked. Values greater than 1 display the notifier for the number of seconds specified.

💾 OK   ✕ Cancel

Add New Server

Configure pgAdmin

Getting Started

PostgreSQL Documentation   pgAdmin Website   Planet PostgreSQL