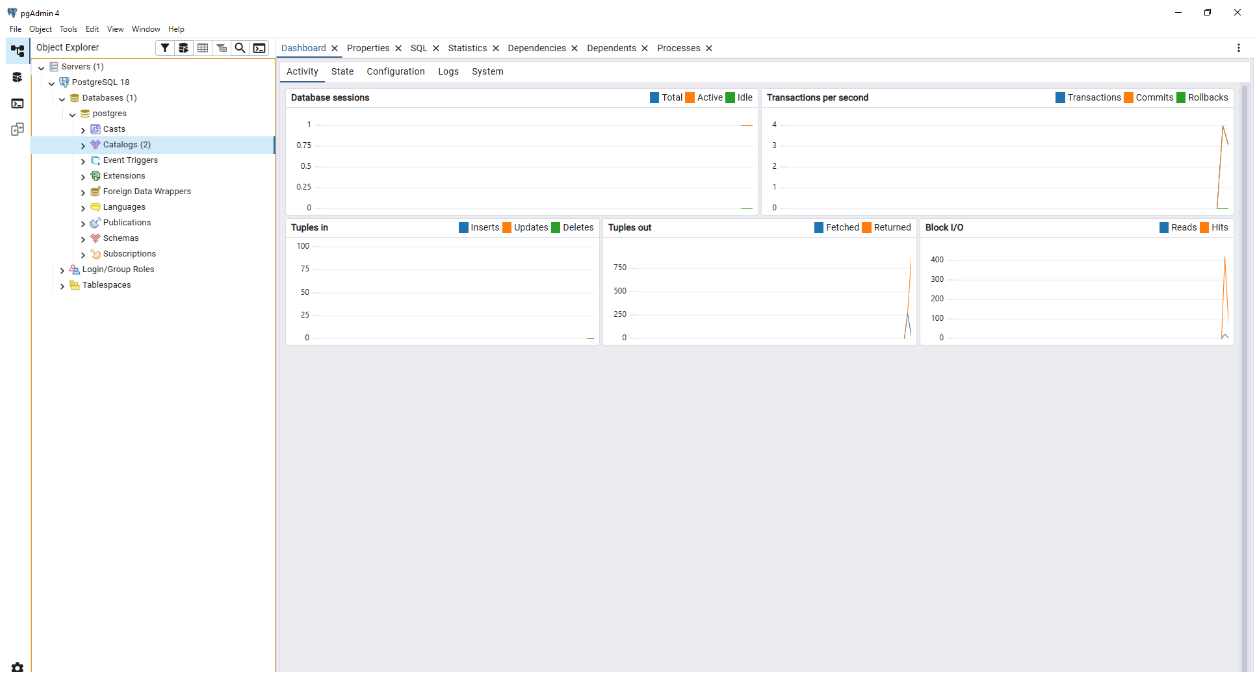

Database of Choice: MongoDB

MongoDB is a document database that stores data in flexible json documents. What this essentially means is that the MongoDB is able to directly store json data in the database. This is good because it means that not every entry has to have the same attributes or columns. You are able to put two entries in the same collection or under the same type in the database, and have one entry hold an extra attribute or one less attribute as opposed to the other entry that was put in. In MongoDB, data within each database is stored in collections. These essentially are categories for the kind of data you are storing within the database. For instance, you can have students in one collection for all the students' data that you enter. You can also have a collection for schools, for sessions, and various types of data that you store within the code. MongoDB, also allows you to make aggregation pipilenes that allow you to search through collections and your database for results that fit specific requirements. For instance, if you wanted all students that have a certain level of gpa and take a certain major, you can use an aggregation pipeline to stack that in order to get those results from your data. Now, for data, although it is valuable and it is great to have it stored somewhere, it is useless if the results and data can not be interpreted.For instance, lets say that you had compiled enrolled students from a multitude of schools, and one of the attributes you assigned and logged for each student is if they graduated or not. At first, you will see a bunch of numbers for who did and didn't graduate, but what does that tell you. Well, information would tell you that the likeliness of graduating or the graduation rate at each school when you closely surf through th data, and thus help you decide which school gives you the best chances to graduate. So information can help you understand the importance of the data and what it means, plus give you the ability to make an informed decision

Data Information:

Hierarchical and network pre-relational data models are essentially a way to divide up different tables of data within a table. For hierarchical data models, they have a parent and child-based system in which certain tables can't exist until a different table exists that has an entry that relates to that table. It essentially shows the flow and order of when tables are created and which table depends on which. Network pre-relational data models refer to tables that might have similar attributes. For instance, teachers and students both will have a CWID number at Marist, which can be a shared attribute among the tables that can at times, signify a relation between the two. Now relational tables refer to tables that show a relation or connection to one another. For example, you can have a foreign key that is found in two tables to show that the entry from one table goes to the entry of another. Foreign keys allow us to know which table connects to ours and how exactly we should match our current entry with another. The Heiarchiala nd network kind of fall short at this because they have data in the database, but it would be hard to match up what goes ot what without relation. This can cause lag and error in finding matches without any key value to look for. I feel like XML is a good model because of the fact that it does have that parent and child feature, ensuring that tables are made only if what theya re dependednt on is also available. The data is well grouped for easy readability and allows you to see what exactly falls under what.