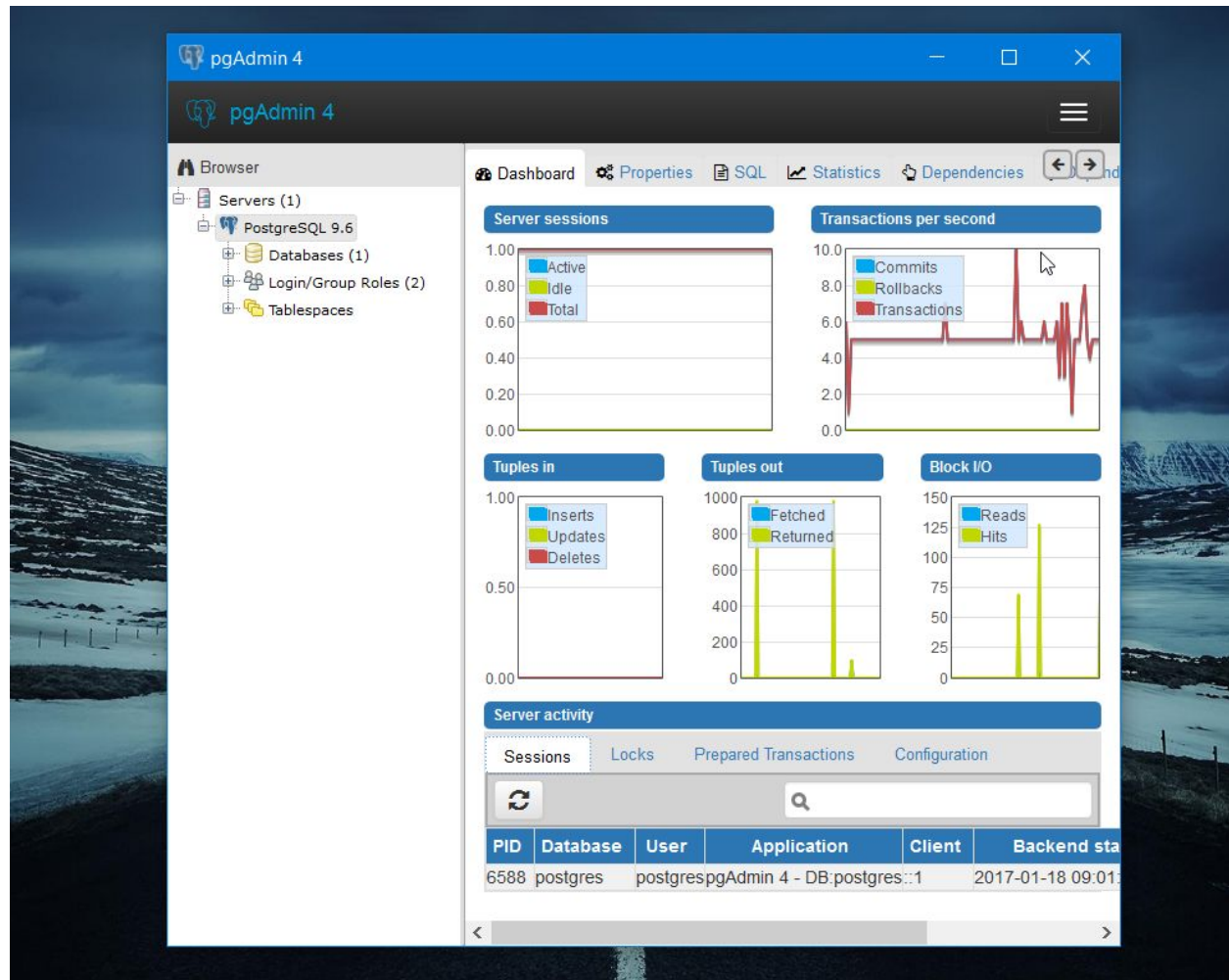


Lab 1



1. Data vs. Information

Suppose we have a database used by an online store's warehouse. In that database, there would be several elements of data, such as an item name, item ID, units in inventory, item price in USD, units sold, and date last restocked. Alone, none of these are particularly useful. However, when put together and thus into context, we now have something we can consider information. By connecting these pieces to one another, they now have meaning and can be used.

For example, let's take a piece of data, such as "number of units in inventory." Knowing a number of units in inventory can be useful, but it is not very helpful when you do not know what item this is for. If you connect it with another piece of data, namely the item name, it suddenly becomes far more useful. We can do this again if we look at the "date last restocked." This alone is not helpful, but if we give the context of number in inventory as well as the item name, we can then tell if an item has been selling quickly or has been sitting there for years. This context has created information that can be used to make meaningful decisions for the user. This information has become far more valuable once it actually can be put to use, as it now has meaning.

Lab 1

2. Data Models

A hierarchical model requires each element to be placed in a strict hierarchy. It can be the child of an element, and can have multiple children, but it can only have one parent. This means if you want to have any element in more than one spot, there must be a copy of it. The structure is essentially stuck with only one-to-many relationships.

A network model allows for elements to have multiple parents as well as children. The parents do not have to have the same parent, either; no special relation between parents is needed. This gets around the issue of the strict hierarchy in the hierarchical model. However, a problem with both of these is if something isn't linked to any of the elements, it technically doesn't exist. This is one of the things the relational model solves.

With this in mind, using XML as a model for data storage might not be the best idea if you wish to avoid the same issues. XML is not relational, and like in the previous two, would require all data to have some sort of relationship in order to exist due to its structure.

