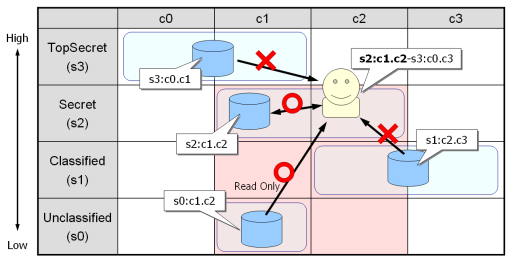
* Open source object-relational database
* Uses and extends the SQL language
* Includes features that safely store and scale most complicated data workloads
* Runs on all major operating systems
* ACID compliant - guarantees validity even in the event of errors, power failures, etc
* Database in the form of a table
* Relational database: when one column refers to another set of data

**Using Postgres with SELinux labels**

* <https://wiki.postgresql.org/wiki/SEPostgreSQL_SELinux_Overview>
* Security Context
  + Database entries in Postgres/SELinux have a ‘security\_context’ (or ‘security\_label’) column in the following format:
  + <selinux\_user>:<role>:<type>:<range>
  + selinux\_user
    - Corresponds to the user id of the operating system. There are a few predefined users on an selinux system: <https://wiki.gentoo.org/wiki/SELinux/Users_and_logins>
  + role
    - The role of a user. This is much like user except a linux account has only one user type. A user type can have multiple roles.
  + type
    - This allows for the creation of rules or ‘security policies’ in SELinux (e.g. A user with type ‘good\_boy’ cannot access data with type ‘bad\_data’
  + range
    - This is the range of labels/categories in the security context
    - 
    - Security levels (s0, s1, etc.) are considered as hierarchical
    - Which data you can read or write based on your security level is configurable from inside SELinux. The above picture uses MLS rules, detailed below.
    - You cannot read from files with a higher security level than yours
    - (For our database, we can replace ‘files’ with ‘database items’
  + More on ranges
    - Format
      * s0-s3:c0.c3,c5
        + Security levels 0 through 3
        + Categories 0 through 3, and also category 5
    - MLS (Multi Level Security)
      * Comprised of 2 ranges separated by a hyphen
      * lowerRange-higherRange
        + (e.g.) s2:c0-s4:c0.c5
      * This gives a process two security settings for 2 different kinds of domains
        + Certain domains may allow the user a higher level of security (the higher-range)
      * Read access is granted if the file’s range is dominated by the range of the process
        + If a file has a security level less than the process range and has a category range contained within the category range of the process
      * Write access is only granted if the file’s range is *identical* to that of the process
  + pg\_security system catalog
    - The system maintains its own tables for keeping track of security contexts in the database. This is because many database objects could have identical security contexts, so it’s easier for them to just have an id which links to the proper context in the pg\_securtity catalog
* Security Label
  + It seems like the system column was called security\_context during development but was later renamed to security\_label
  + <https://selinuxproject.org/page/NB_MLS#Security_Levels>