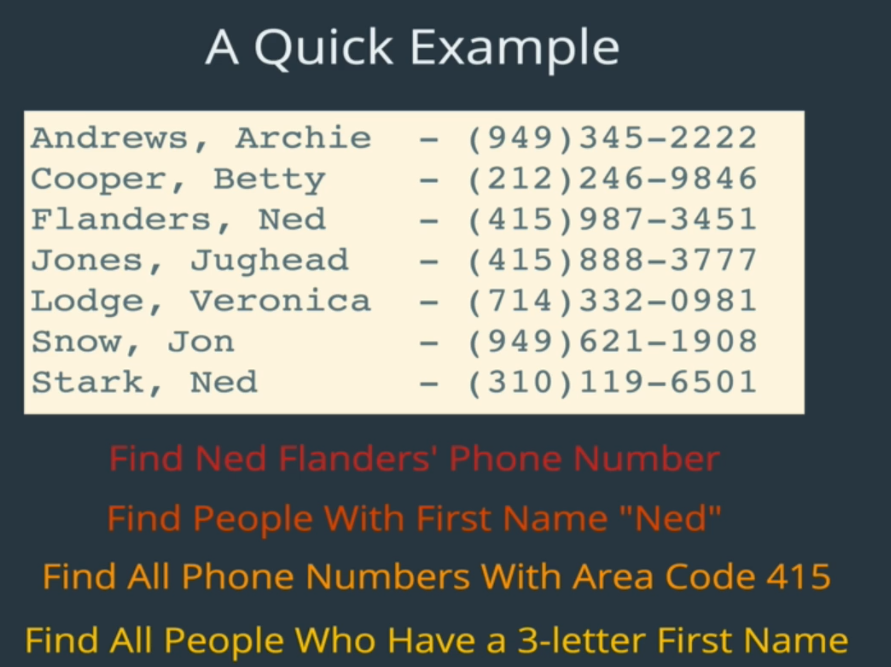
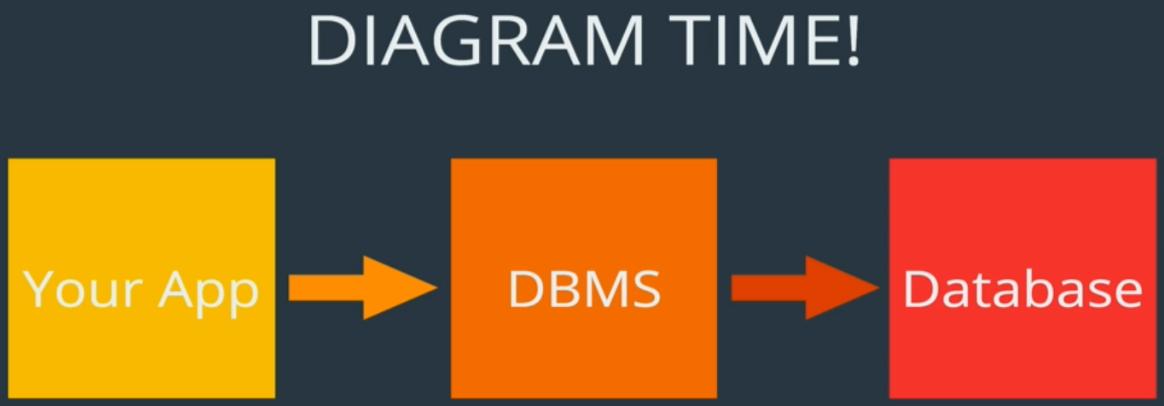
# What is a Database?

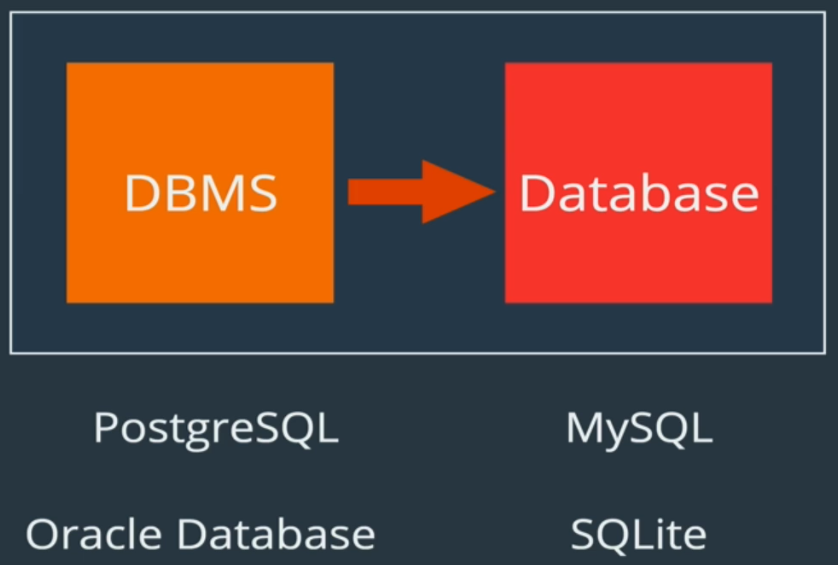
* Lecture slides: <http://webdev.slides.com/coltsteele/mysql>
* What is a **Database**?
  + A database is **collection of data**
    - Examples include to do lists, medical records at your doctor’s office, digital rolodexes
    - Consider this fictional example of phone numbers and what we can do with it. Some of these questions could be easily answered by flipping through a traditional phone book. But other questions are considerably more complicated.



* + A database also contains a **method for accessing and manipulating that data**, or an **interface**
    - The difficult questions above are much more easily tackled when using the methods that a database allows
* Databases vs Database Management Systems (DBMS)
  + This refers to the fact that data and interfaces for data are two different things
  + On its own, a database is just a giant collection of data. It can’t really do much on its own
  + A database management system is needed to interface with the database, allowing you to organize it, edit it, make changes to it, delete some or all of it, etc.
  + Consider this diagram



* + Be aware that sometimes people refer to both the DBMS and Database together as “the database”
  + This will be helpful later one to understand the practice of giving commands to the DBMS, which in turn leads to giving commands to the data itself
* Examples of database management systems



* To summarize, a database is *a structured set of computerized data with an accessible interface*

# MySQL vs SQL

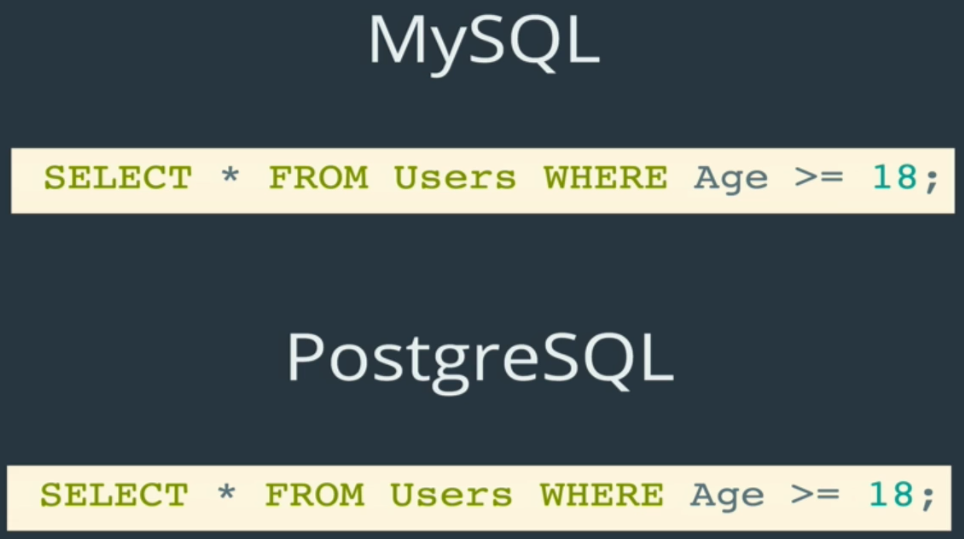
* What is the difference between SQL and MySQL?
* **SQL** stands for **S**tructured **Q**uery **L**anguage
  + It is the language we use when we “talk” to our databases
  + It’s how we access and manipulate data in the databases
  + We can do things like:
    - Find all users
    - Delete all users
    - Find all users who are 18 and older
      * Here’s a preview of SQL code for “find all users who are 18 or older”



* + As you can see, there’s a special syntax that is used for the instructions, when then goes off and interacts with the database in some way
* It is important to understand that *working with MySQL is primarily writing SQL code*.
  + There are multitudes of database management systems that use SQL; it is not unique to MySQL



* Here’s an example of a line of SQL code in MySQL vs. PostgreSQL



* + That’s right – since both of these DBMS’s use SQL as their language, the code is exactly identical between them
  + There are *slight* differences in syntax, but altogether they are very similar
* There is also a **SQL Standard**, which is a standardized way of how SQL should work
* Takeaways:
  + Once you learn SQL, it’s pretty easy to switch to another DB that uses SQL
  + What makes databases or DGMS’s unique are the *features that they offer*, NOT the language of SQL itself
    - Features could include speed, security, download sizes, user permissions, etc.

# Installing MySQL

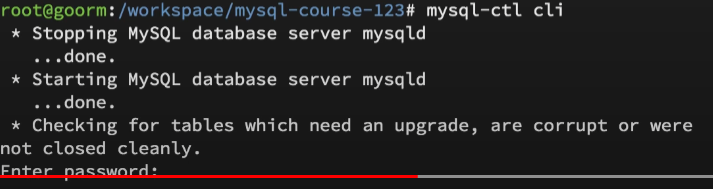
* Installation slides: <https://webdev.slides.com/coltsteele/mysql-97>
* Goorm.io installation and navigations instructions: <https://gist.github.com/nax3t/2773378c4d1bada8d66d12f4d5210248>
  + We now use goorm.io instead of Cloud9, which was changed after Amazon Web Services acquired it
  + Goorm is an IDE that is very similar to the legacy version of Cloud9

# Goorm.io

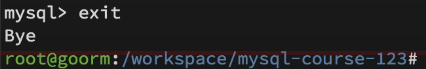
* Navigating Goorm and creating/deleting files and folders: <https://youtu.be/T3h0JivOir8?t=44>
* Cheat sheet for Goorm MySQL: <https://gist.github.com/nax3t/767e06f6af0bafc70b4c4cba0c8d38e7>
* Cheat sheet for Goorm MySQL: <https://gist.github.com/nax3t/767e06f6af0bafc70b4c4cba0c8d38e7>
  + How to launch the mysql shell, list databases, and open databases
* Mysql-ctl commands/shell: <https://www.youtube.com/watch?v=JMZtWvJxfuM>
  + The following commands are used to **start** and **stop** the MySQL shell/server



* + - –ctl stands for “command tool”
  + The command mysql-ctl cli stops any existing servers and starts a brand new one each time we run it, and then starts the MySQL shell. This is important because it runs all commands we learn in this course



* + - “cli” stands for “command line interface”
    - There is no password – leave it blank and press Enter
    - You know the shell is active when you see **mysql>** in the terminal
    - This will be the main command that we’ll be running for this course
  + To get back to the bash terminal from the shell, just type “exit”



* Sourcing SQL files inside of the goorm IDE container: <https://www.youtube.com/watch?v=DjhthBTXvXg>
  + It is super helpful to write queries in a SQL file and then paste them into the MySQL shell instead of manually typing them into the shell directly. This helps tremendously when you, for example, make typo errors.
  + You can also run SQL files using the source command in the shell. However, you must *open the command line from the same directory that the file is located in*. You can check this by checking our current directory and its contents
    - The command is **source <filename>.sql**
  + By updating a “query.sql” file, we can simply replace our existing query with a new one! All you have to do is write your query into the **query.sql** file, save it, and execute it.
    - Make sure you’re using the correct database (“use”) when running the query.sql file
* Uploading a file to goormide
  + <https://www.loom.com/share/17d9b6d23cd7457ca7d5a8e585fe6733>