

Databases for Analytics

Course Introduction
Syllabus, Software, Tutorials, etc.

Topics

- Introductions
- Why this course?
- Developers vs Analysts
- Course Syllabus, etc.
- Software Tools

Welcome

- Dr. Christopher L. Huntley
 - PhD in Systems Engineering (UVa, 1995)
 - At Fairfield U since 1997, before that (mostly) in industry
 - Mastered over a dozen programming languages so far
- Questions for you:
 - Who are you? (name, nickname, and hometown)
 - Background? (degrees and professional experience)
 - Something ***distinctive*** about yourself that we can't tell by looking at you?

The Big Picture

Databases and Business Analytics

Why Learn SQL? Isn't Python Enough?

Python with NumPy+Pandas+Matplotlib is a great toolset for crafting data-driven analyses. It does just about everything we need to analyze datasets (i.e, files) from a variety of sources.

However, sometimes data is found in *databases* instead of *files*. This is especially true of live **transaction data** like that found in just about any corporate information system. For that, we use SQL.

Transaction Data vs Analytical Data

| | Transaction Processing | Analytical Processing |
|------------------|---|--------------------------|
| Example | Bank Accounts | Quarterly Financials |
| Age | Online/Live | Historical |
| Focus | Data Integrity & Controls | Informed Decision Making |
| Access | Multiple concurrent users Read and Write | Single user Read-only |
| Lang/Tech | SQL Database | Python, R, Excel |

But can't we just use APIs?

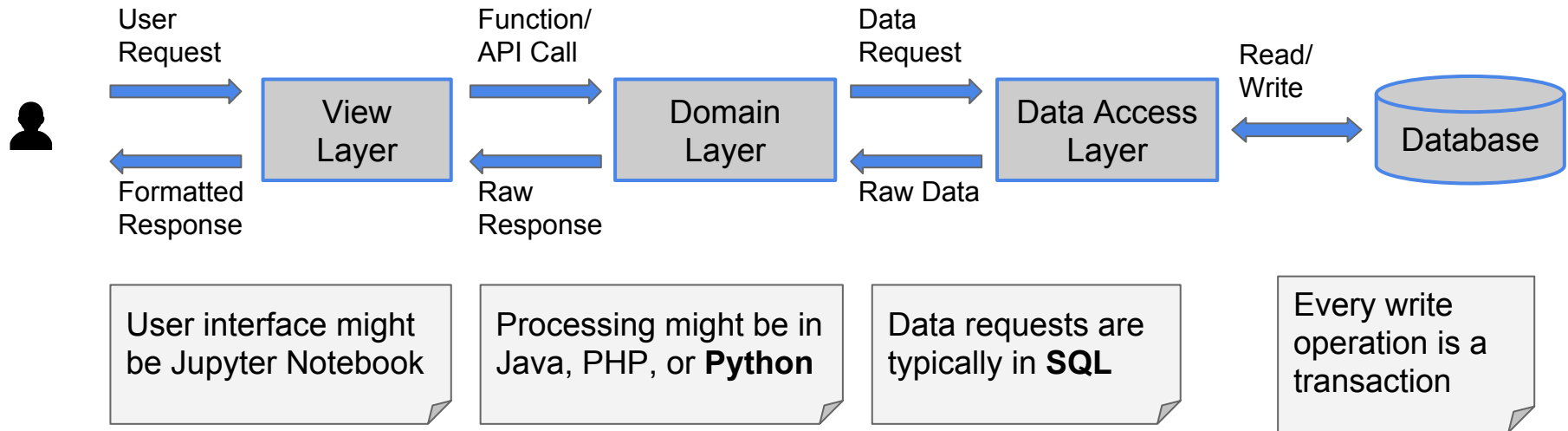
Many big corporate systems have **Application Programmer Interfaces (APIs)** that can be used to access data in real time.

- The programmers provide a function for every conceivable task one might want to ask the system to perform.
- Many of these functions are for data Creation, Retrieval, Updating, or Deletion (CRUD).

However, 'every conceivable task' does not mean full access to the data. You'll need SQL for that.

Three-Tiered Architecture

Virtually all modern information systems are organized into layers between the users and data.



Goal: Unfettered Access to Data

So, to **ensure that you always get the most current and complete view** of transactional data (not just the functions in the APIs), you will need to **know and use SQL**.

Fortunately, SQL works great with Python!

Course Expectations

What does success look like?

Knowing Our Limits

Knowing SQL is not the same thing as being a Database Engineer.

We only need to know enough SQL to ...

- Get the data we need out of the system
- Manage (add/update/delete) the data in the system
- Perhaps suggest design changes to the system that would improve/simplify our analytical results

Course Objectives

- **Develop new skills**
 - Structured Query Language
 - Basic DB administration
- **Learn fundamentals of relational database systems**
 - Entity-Relationship Modeling
 - Relational model and table normalization
- **Apply knowledge and skills to business analytics**
 - Database-backed analytics project
 - Build on Python/Pandas skills from IS505

Course Plans and Policies

Assignments, Grading, etc.

Coursework

- Tutorials (ungraded but required)
 - Cover specific theory and practice needed for the graded assignments. ***Progress is tracked online.***
- Quizzes (50% of course grade)
 - 5 Quizzes, with lowest grade dropped from Quiz Avg
- Team Project (40% of grade)
 - 2-3 students per team
 - Assigned in the fifth week of the course
- Professionalism (10% of grade)
 - Participation and timely completion of assigned work

Grading System: Curve *Everything*

Every graded assignment will be **scored** and then **normalized** using the following formula:

$$QP = 3.5 + \frac{1}{2} (x - \mu) / \sigma,$$

← The average QP is 3.5,
which is an A-

where

- x is the student's raw score for the assignment
- μ and σ are the class average and standard deviation for the assignment

Letter grades are then 3.67+ → A, 3.34-3.66 → A-, ...

Academic Honesty

- Cheating will be dealt with swiftly in accordance with Fairfield University policy
 - Unless given **explicit permission** to collaborate, do not share your work with others
 - *Avoid even the appearance of cheating!*
- Each graded assignment will be accompanied by the following (signed) pledge:
 - *On my honor as a Fairfield University student, I have neither given nor received any unauthorized aid on this assignment/quiz/project.*

Class Docs / Website

All lectures, programming assignments, etc. are available here:

<https://christopherhuntley.github.io/is510-docs>

The class syllabus is linked from the home page:

<https://christopherhuntley.github.io/is510-docs/Syllabus.html>

Setup

Accounts & Software Installation

If you took IS505 then you can skip to [here](#).

Sign Up for DataCamp

- Data Camp is an online school for data analytics in Python, R, and SQL. We have a “class group” for IS510 where your progress can be tracked.
- Invitation emails will be sent to your `@student.fairfield.edu` address.
- Follow the instructions to confirm your enrollment on the class roster.

GitHub / GitHub Classroom

All class documents, assignments, and projects will be managed online using GitHub.

- Syllabus, lectures, etc. are in the is510-docs repo:
 - <https://github.com/christopherhuntley/is510-docs>
- GitHub Classroom will be used to post and grade programming assignments
 - Invitations for each assignment will be sent by email
- We will more about GitHub as we go along, starting with installation in class tonight

Sign Up for GitHub

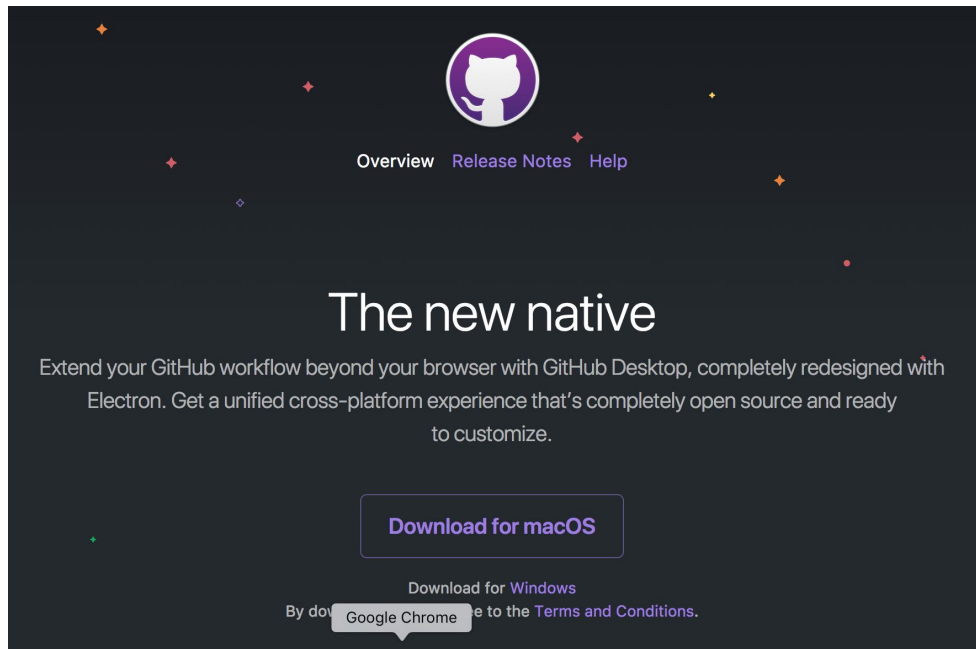
1. Go to GitHub.com
2. Sign up for a new account using your @student.fairfield.edu account.
3. Send an email from your student email to chuntley@fairfield.edu with your GitHub account username. The email subject is “GitHub account”.

Skip steps 1 and 2 if you already have a GitHub account **linked to your student email address.**

Install GitHub Desktop

Download from
desktop.github.com.

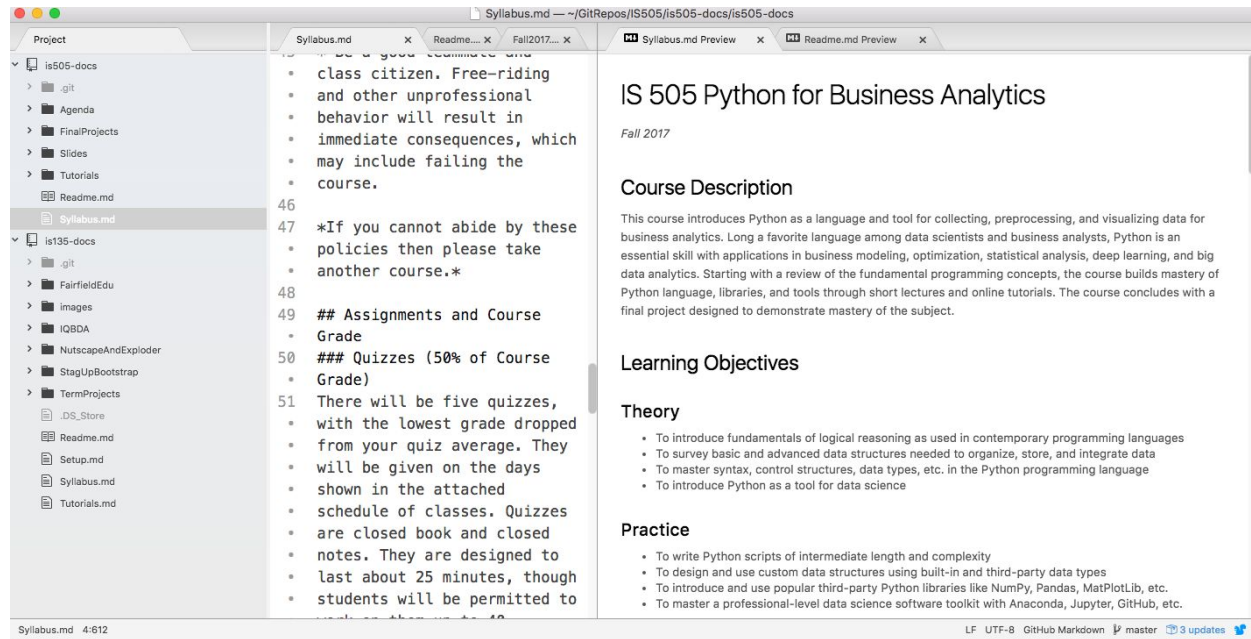
Then install as usual.



Install Atom (Recommended)

A code editor
that works
great with
GitHub.

Install from
atom.io



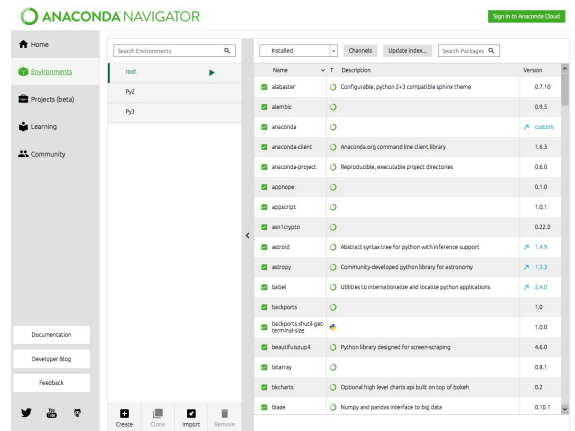
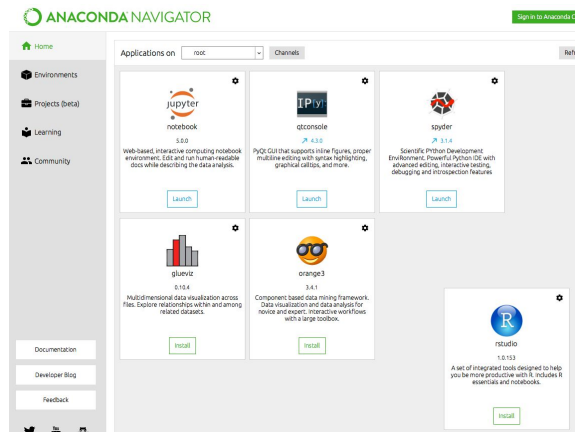
Create a Folder for your work

1. Create a new folder called **IS510** in your **documents** folder or desktop.
2. All your Git repositories and other work will be in this new **IS510** folder.
3. Take note of where you created the folder. You will need it later.

Anaconda

Anaconda is a desktop Python environment that bundles lots of tools and packages:

- Python (Installation)
- Apps: Jupyter Notebooks, Spyder IDE, etc.
- Libraries: NumPy, Matplotlib, etc.
- Conda: command line tools



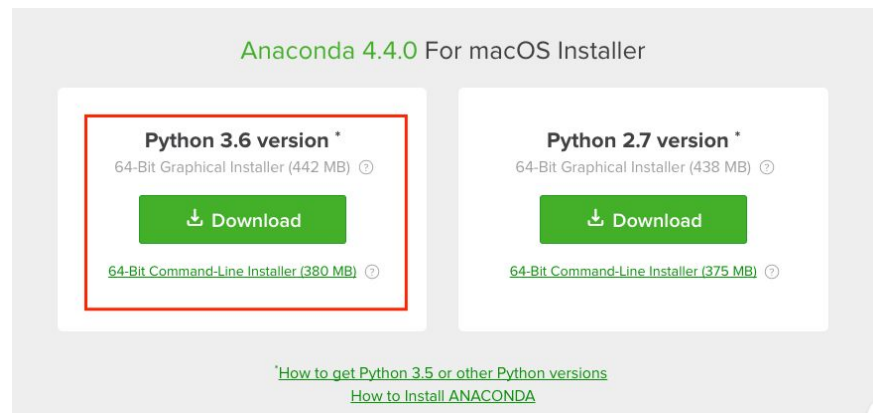
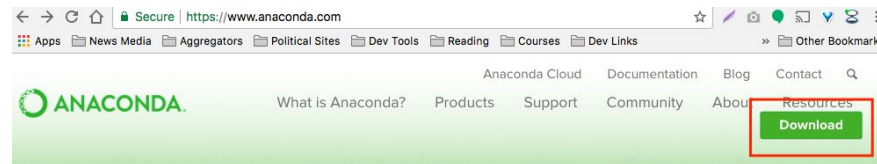
Install Anaconda

Go to anaconda.com and click the download button.

Choose the Python 3.* version for your OS.

The download may take a while. Be patient.

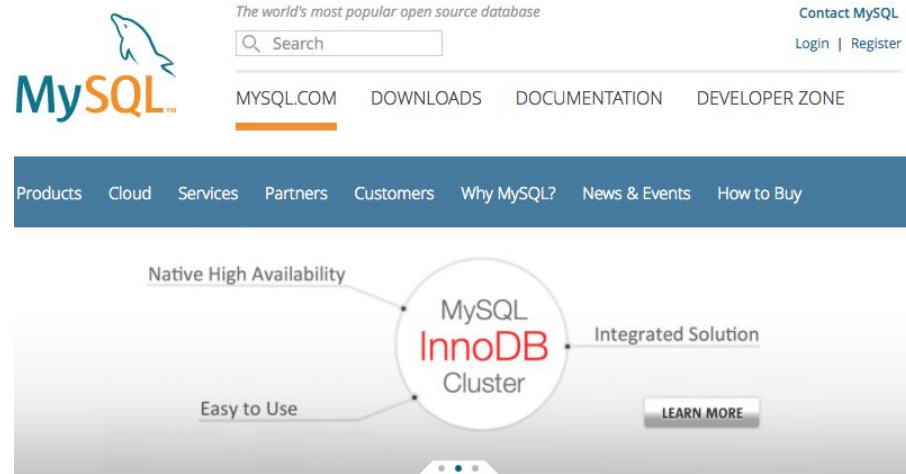
Install as usual.



Install MySQL

MySQL is Oracle's open source DBMS. It is widely used for web apps.

We need both **MySQL Server** and **MySQL Workbench**.



MySQL Enterprise Edition

The most comprehensive set of advanced features, management tools and technical support to achieve the highest levels of MySQL scalability, security, reliability, and uptime.

[Learn More »](#)

Windows vs MacOS

How to install MySQL depends on your operating system.

Windows: Use the [all-in-one installer](#).

MacOS: Install **MySQL Server (v5.7)** and then **MySQL Workbench (v6.1)** to work around a bug in Mac OS 10.13 (High Sierra).

MacOS: MySQL Server Community Edition

Download and install the latest release in the v5.7 series.

After installing, you will need to reboot to get the MySQL launcher in your preferences panel.

The screenshot shows the MySQL Community Server 5.7.19 download page. The 'Generally Available (GA) Releases' tab is selected. The 'Select the OS' dropdown menu is set to 'Mac OS X'. A red box highlights the 'Mac OS X 10.12 (x86, 64-bit), DMG Archive' download option, which is 338.8M in size. Below it, there are links for 'Mac OS X 10.12 (x86, 64-bit), Compressed TAR Archive' and 'Mac OS X 10.12 (x86, 64-bit), Compressed TAR Archive Test Suite'. A red box also highlights the 'Download the installer' link. At the bottom, a note suggests using MD5 checksums and GnuPG signatures to verify the integrity of the packages.

MySQL Community Server 5.7.19 **Select the OS**

Select Operating System: **Mac OS X** [Looking for previous GA versions?](#)


Download the installer

| Package Name | Version | Size | Action |
|--|---------|--------|--------------------------|
| Mac OS X 10.12 (x86, 64-bit), DMG Archive (mysql-5.7.19-macos10.12-x86_64.dmg) | 5.7.19 | 338.8M | Download |
| Mac OS X 10.12 (x86, 64-bit), Compressed TAR Archive (mysql-5.7.19-macos10.12-x86_64.tar.gz) | 5.7.19 | 314.2M | Download |
| Mac OS X 10.12 (x86, 64-bit), Compressed TAR Archive Test Suite (mysql-test-5.7.19-macos10.12-x86_64.tar.gz) | 5.7.19 | 24.3M | Download |

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

MacOS: Check for MySQL launcher

MySQL should show up in your **System Preferences** panel. Click to start/stop the server.



MySQL Server Status


The MySQL Database Server is started and ready for client connections. To shut the Server down, use the "Stop MySQL Server" button.

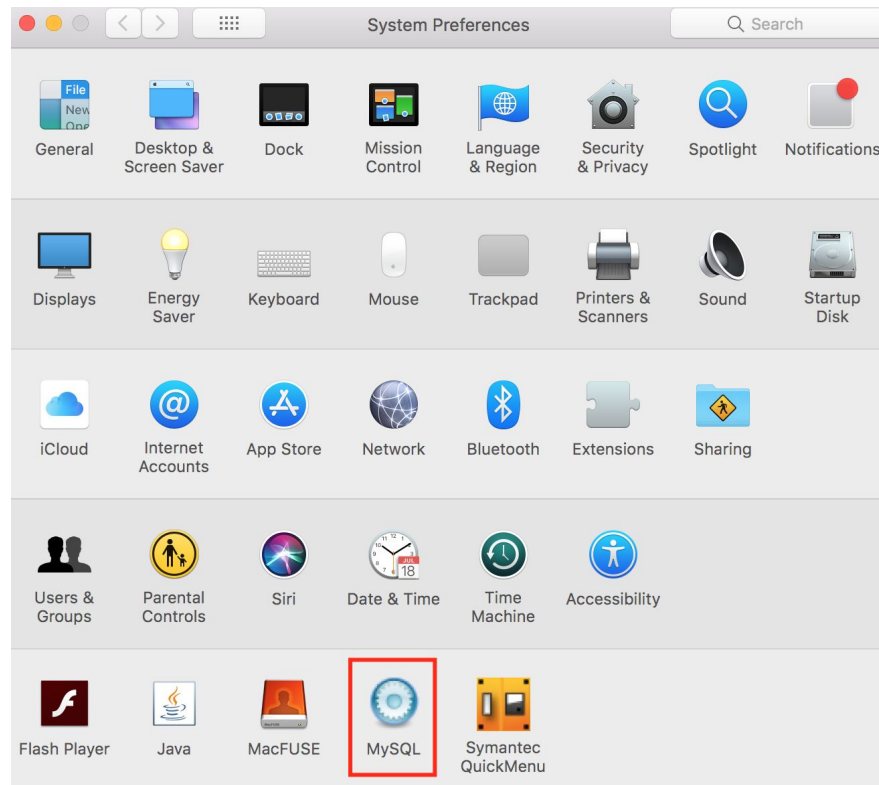
The MySQL Server Instance is **running** Stop MySQL Server

If you stop the server, you and your applications will not be able to use MySQL and all current connections will be closed.

☒ Automatically Start MySQL Server on Startup

You may select to have the MySQL server start automatically whenever your computer starts up.

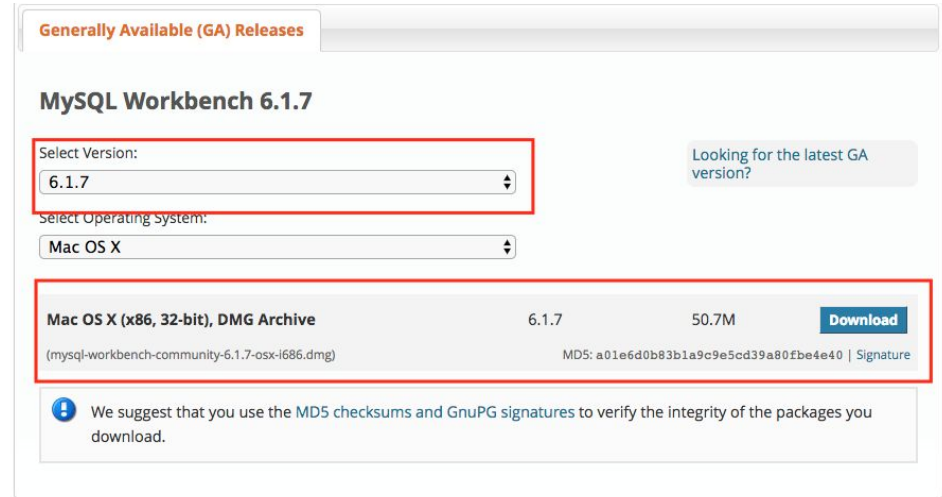




MacOS: Install MySQL Workbench

MySQL Workbench is an app for managing and querying MySQL Databases.

Install version 6.1.7. Later version are not yet compatible with MacOS 10.13



Anaconda Add-ons

Anaconda is missing a few things we'll want in order to connect Jupyter to our databases. We'll need to ...

1. Make sure **sqlalchemy** is installed/enabled
2. Install the **pymysql** bridge library
3. Install the **ipython-sql** magic for Jupyter

We will use a combination of Anaconda Navigator and the command line.

Anaconda Environment/Packages

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Projects (beta)

Learning

Community

Documentation

Developer Blog

Feedback

Create Clone Import Remove

Search Environments

Installed Channels Update index... sql

| Name | T | Description | Version |
|--------------------------|---|---|---------|
| ✓ ipython-sql | | | 0.3.8 |
| ✓ mysql-connector-python | | | 2.0.4 |
| ✓ pymysql | | | 0.7.9 |
| ✓ sqlalchemy | | Python sql toolkit and object relational mapper | 1.1.11 |
| ✓ sqlite | | Sql database engine | 3.13.0 |
| ✓ sqlparse | | Python-sqlparse | 0.2.4 |

6 packages available matching "sql"

A complete Installation looks like this.

We'll do it one step at a time.

SQLAlchemy

SQLAlchemy provides a bunch of useful Python utilities.

1. Check for SQLAlchemy in your Installed packages for the root environment.
2. If it is not installed then install it from the Not Installed packages list.

Not installed Channels Update index... sql X

| Name | T | Description | Version |
|--|---|---|---------|
| <input type="checkbox"/> pandasql | ○ | SqlDf for pandas | 0.7.3 |
| <input type="checkbox"/> postgresql | ○ | A powerful, open source object-relational database system | 9.5.4 |
| <input type="checkbox"/> psycopg2 | ○ | Postgresql database adapter for python | 2.7.1 |
| <input type="checkbox"/> r-rsqlite | ○ | | 1.1_2 |
| <input checked="" type="checkbox"/> sqlalchemy-utils | ○ | | 0.32.14 |
| <input type="checkbox"/> zope.sqlalchemy | ○ | | 0.7.7 |

This is just an example showing how to install a new package in Anaconda Navigator.

6 packages available matching "sql" 1 package selected Apply Clear

PyMySQL Package

PyMySQL is a Python driver for connecting to MySQL databases.

1. Open the Command Prompt (Windows)/ Terminal (MacOS).
2. Use the conda package manager to find and install the package.

```
conda install -c anaconda pymysql
```

chuntley — conda install -c anaconda pymysql — 80×24

Last login: Wed Oct 11 22:08:21 on ttys003

DSB1122-C4148M:~ chuntley\$ conda install -c anaconda pymysql

Fetching package metadata

Solving package specifications: .

Package plan for installation in environment /Users/chuntley/anaconda:

The following packages will be UPDATED:

| | | | | |
|----------|-----------------------|-----|-----------------------|----------|
| conda: | 4.3.27-py36hb556a21_0 | --> | 4.3.30-py36h173c244_0 | anaconda |
| pymysql: | 0.7.9-py36_0 | --> | 0.7.11-py36h75d80ff_0 | anaconda |

The following packages will be SUPERSEDED by a higher-priority channel:

conda-env: 2.6.0-0

Proceed ([y]/n)? █

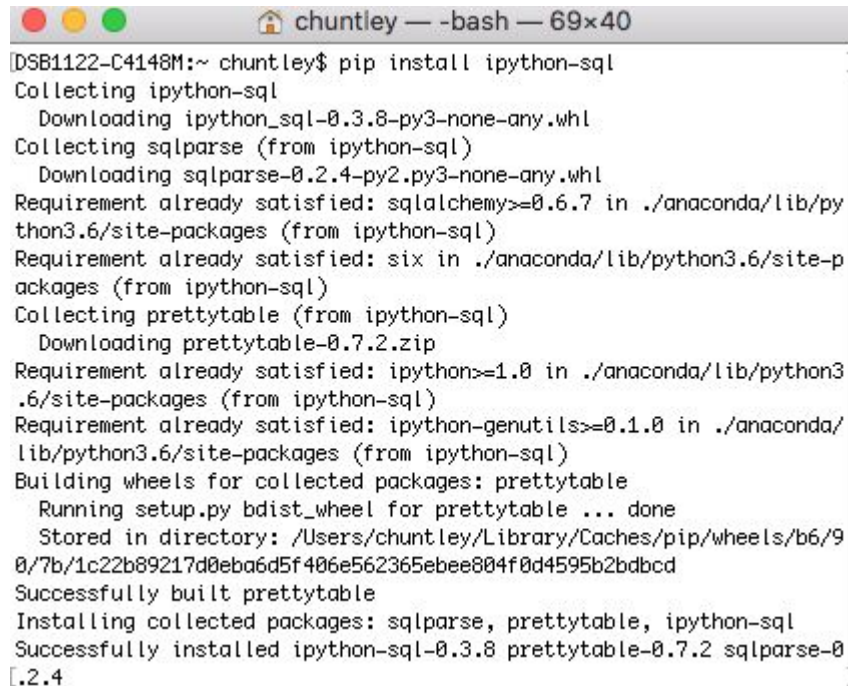
This is the MacOS Terminal, but it looks similar in the Windows Command Prompt

ipython-sql Package

This adds special sql "magic" for Jupyter Notebooks.


1. Install from the command line.
2. Use pip as the package manager.

```
pip install ipython-sql
```

A terminal window titled 'chuntley — -bash — 69x40' showing the command 'pip install ipython-sql' and its output. The output details the collection and downloading of ipython-sql, sqlparse, and prettytable, and confirms their successful installation.

```
DSB1122-C4148M:~ chuntley$ pip install ipython-sql
Collecting ipython-sql
  Downloading ipython_sql-0.3.8-py3-none-any.whl
Collecting sqlparse (from ipython-sql)
  Downloading sqlparse-0.2.4-py2.py3-none-any.whl
Requirement already satisfied: sqlalchemy>=0.6.7 in ./anaconda/lib/python3.6/site-packages (from ipython-sql)
Requirement already satisfied: six in ./anaconda/lib/python3.6/site-packages (from ipython-sql)
Collecting prettytable (from ipython-sql)
  Downloading prettytable-0.7.2.zip
Requirement already satisfied: python>=1.0 in ./anaconda/lib/python3.6/site-packages (from ipython-sql)
Requirement already satisfied: ipython-genutils>=0.1.0 in ./anaconda/lib/python3.6/site-packages (from ipython-sql)
Building wheels for collected packages: prettytable
  Running setup.py bdist_wheel for prettytable ... done
  Stored in directory: /Users/chuntley/Library/Caches/pip/wheels/b6/90/7b/1c22b89217d0eba6d5f406e562365ebee804f0d4595b2bdbcd
Successfully built prettytable
Installing collected packages: sqlparse, prettytable, ipython-sql
Successfully installed ipython-sql-0.3.8 prettytable-0.7.2 sqlparse-0.2.4
```

Anaconda Navigator Again

 ANACONDA NAVIGATOR Sign in to Anaconda Cloud

Home

Environments

Projects (beta)

Learning

Community

Documentation

Developer Blog

Feedback

Twitter YouTube GitHub

Create Clone Import Remove







Search Environments

root

Py2

Py3

Installed Channels Update index... sql

| Name | T | Description | Version |
|--------------------------|---|---|------------------------|
| ✓ ipython-sql |  | | 0.3.8 |
| ✓ mysql-connector-python |  | | 2.0.4 |
| ✓ pymysql |  | | 0.7.9 |
| ✓ sqlalchemy |  | Python sql toolkit and object relational mapper | 1.1.11 |
| ✓ sqlite |  | Sql database engine | 3.13.0 |
| ✓ sqlparse |  | Python-sqlparse | 0.2.4 |

6 packages available matching "sql"

A Dry Run ...

A system check to make sure everything is working properly

GitHub

As in IS505, we will again be using GitHub for managing coding assignments.

Our first assignment is [Deals Database Part 1](#), which has further instructions.

The slides that follow walk through the assignment, step by step.

GitHub Classroom Check

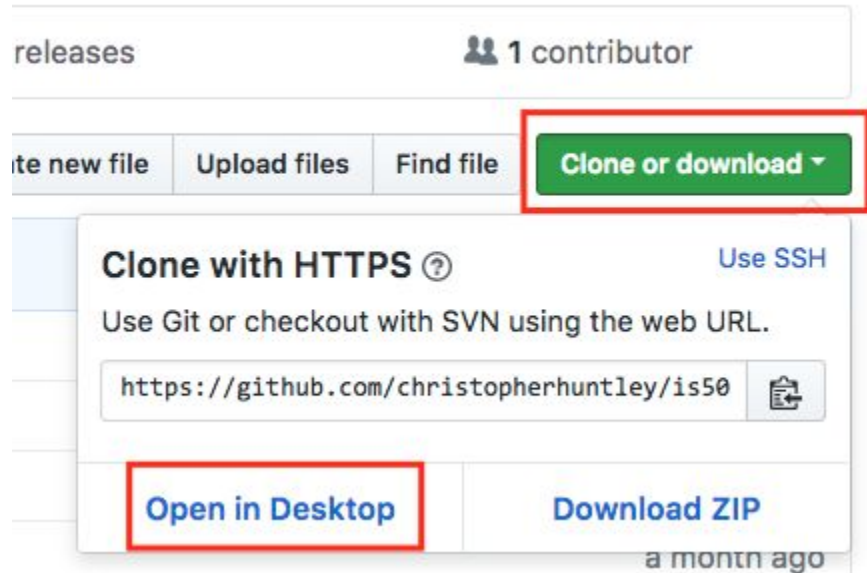
Assignments will be listed in [GitHub Classroom](#):

- Once you are listed on the roster, you should get an email invitation for each assignment. So, ... check your email.
- If no email arrives then follow [this link](#).

GitHub Desktop Check

Clone your forked copy of the repository to your desktop.

Save the repository in your new **IS510** folder.

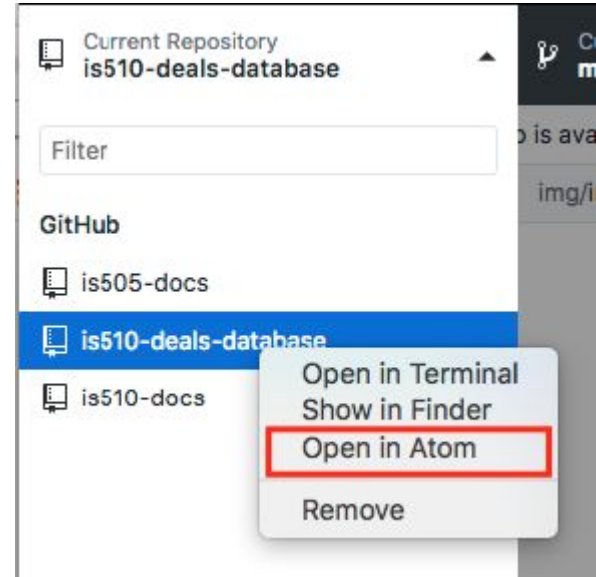


Atom Editor Check

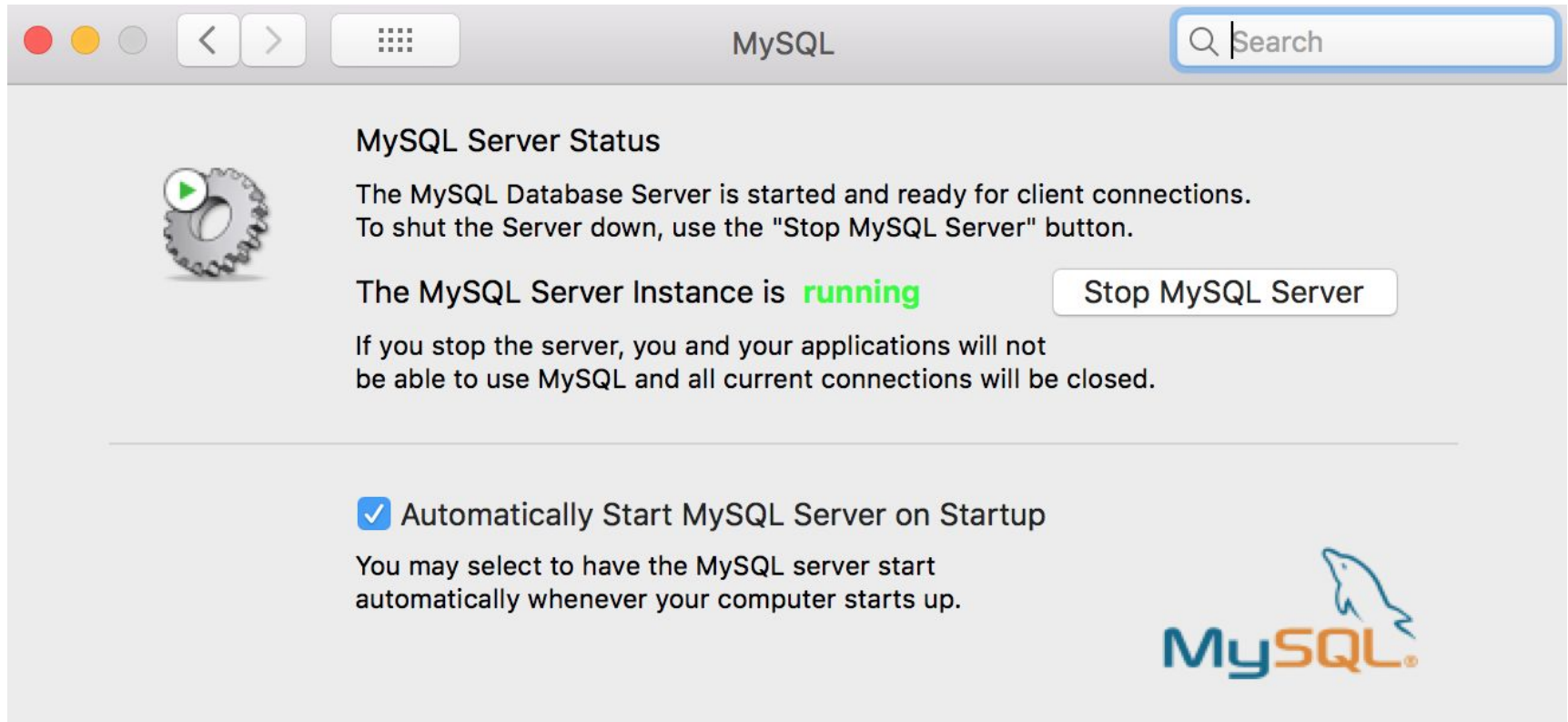
From the repository pane, right click on the repository and select Open in Atom.

Atom should appear with the repository contents listed on the left.

Open the **deals.sql** file.

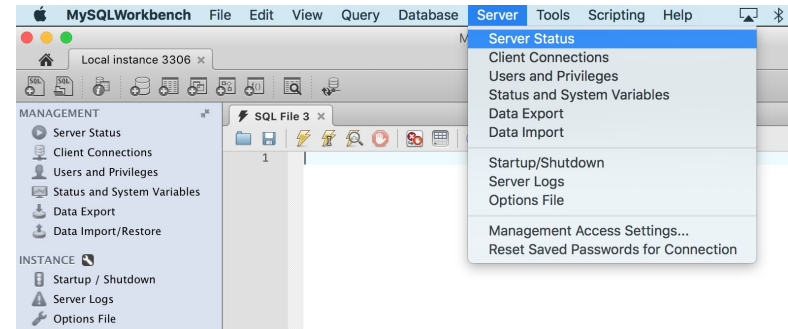
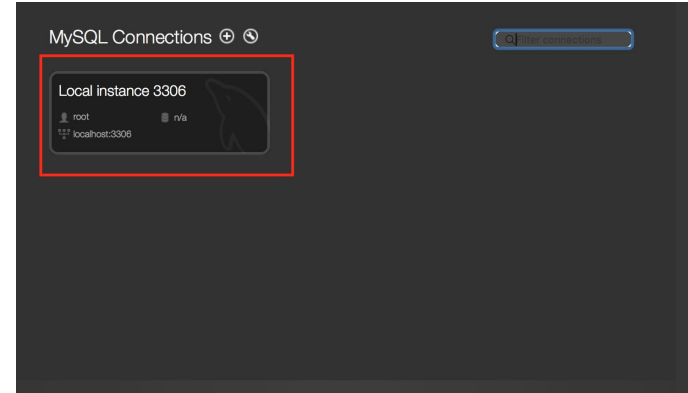


MySQL Server Check



MySQL Workbench Check

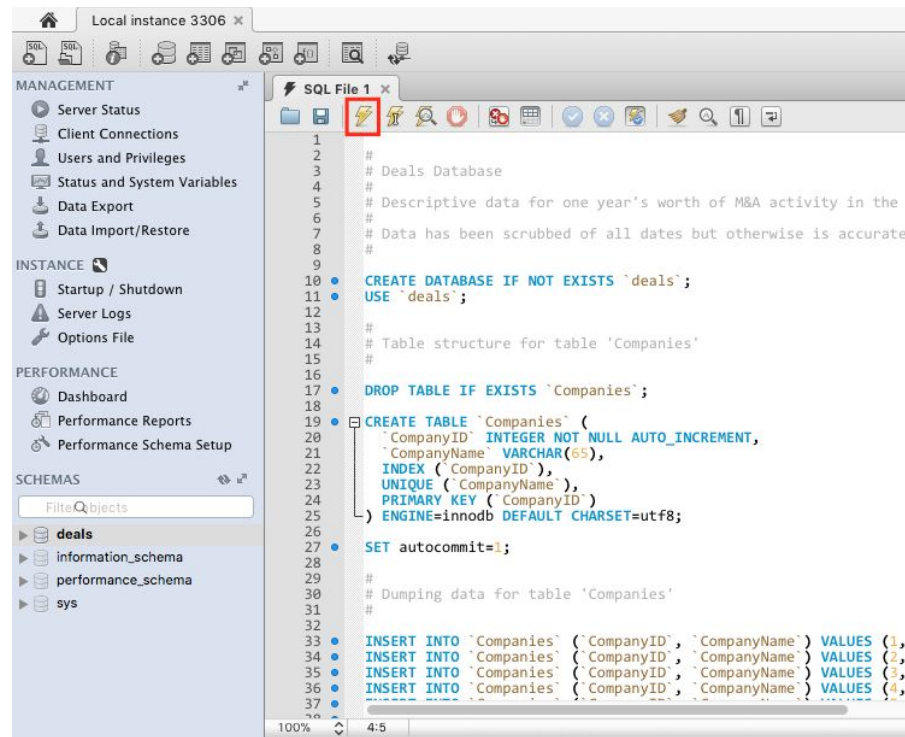
1. Open MySQL Workbench
2. Choose your running instance of MySQL Server
3. Check that MySQL Workbench can control the server.



Loading the Database

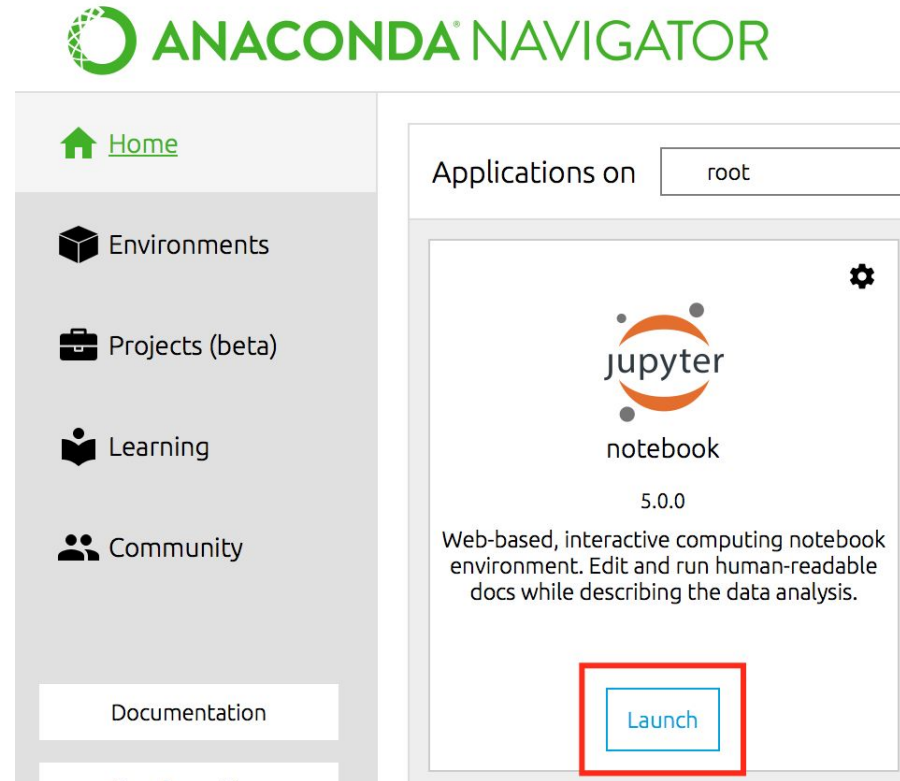
Run the **deals.sql** script:

1. File → Open SQL Script ...
2. Navigate to your repository folder.
3. Select the **deals.sql** file.
4. Click the lightning bolt icon to run the script.
5. The **deals** schema should appear in the left panel.



Jupyter Notebook Check

1. Open Anaconda Navigator (if not open).
2. Launch Jupyter Notebook.
3. Open the **Deals_Part1.ipynb** notebook in your repository folder.



SQLAlchemy, PyMySQL Check

The first part of the notebook sets up a connection to the database, much like we just did with MySQL Workbench. This is where the PyMySQL Package comes into play.

Run the first cell to check if PyMySQL is working correctly. You should get a table of company names.

`%sql` Magic Check

The next code cell uses `%sql` 'magic' to embed SQL code directly into a Python assignment statement.

Run the cell. The variable **`companies`** is a Pandas DataFrame which is displayed as a table.

If this doesn't work just like the first code cell then the `ipython-sql` package is not installed correctly.

Sign your work

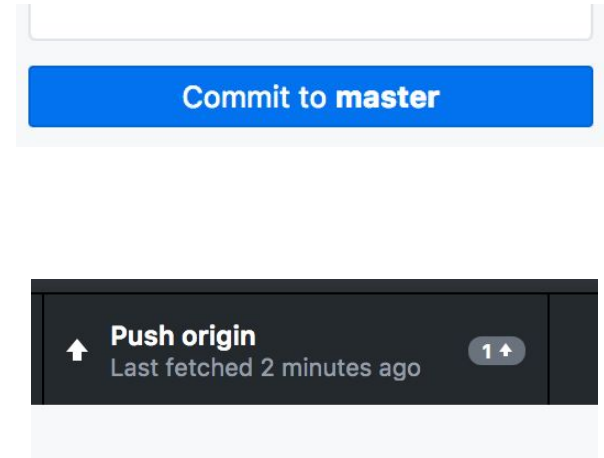
Add a new Markdown cell with your name in it to the bottom of the notebook.

Save the notebook.

Sync to GitHub

In GitHub Desktop, note that Git has detected your edit to the notebook.

- **Commit** your changes with the comment "Completed Part 1"
- **Push** your updated repository to GitHub



Python for Analytics

Course Introduction
Syllabus, Software, Tutorials, etc.