HW1 due on GitLab on Wednesday, July 3 at 11:00 P.M.

Make sure to check out the submission instructions!

Sign up for Piazza!
Virtual Machine
CSE Virtual Machine (VM)

We will be developing on Linux, CentOS
https://www.cs.washington.edu/lab/software/linuxhomevm

The VM will help!
A lot of stuff is pre-installed
Fewer setup bugs (a huge problem previous quarters)
VM Demo!
Git/Gitlab
Instructions
Install and Configure Git

Linux (Debian/Ubuntu):

```
sudo apt-get update
sudo apt-get install git
```

Mac:

http://git-scm.com/download/mac

Windows:

http://git-scm.com/download/win

Verify git installation (do this first to check to see if git is already installed)

```
git --version
```

Configure username:

```
git config --global user.name
"John Doe"
```

Configure user email:

```
git config --global user.email
"netid@uw.edu"
```
Register your Computer on Gitlab (SSH)

To generate a SSH key pair (if you don’t have one):

Linux/Mac: `ssh-keygen -t rsa -C "netid@uw.edu" -b 4096`

Windows: use [PuttyGen](https://www.putty.org/) to generate and save keys

Instructions

Use the suggested save path (`~/.ssh/id_rsa.pub`)  
Password for keys is optional

On Gitlab > User Settings (under your profile icon) > SSH Keys:

Paste in your **public** key (copy from typing `cat ~/.ssh/id_rsa.pub`)  
Mac users can type `pbcopy < ~/.ssh/id_rsa.pub`

Name your key anything  
Hit “Add Key”
Gitlab

Login with UW NetID

https://gitlab.cs.washington.edu/users/sign_in

Make sure you have access to your CSE 344 project repository (cse344-netid)

Try to clone your repo

```
git clone [repoURL]
```
Add Upstream

```bash
git remote -v
```

Should have your local repo with fetch and push
We want to add the class repo as an upstream

```bash
git remote add upstream
```

Now check again

```bash
git remote -v
```
Git Basics

Git commands you should be familiar with:

- `git status`
- `git clone [repo path]`
- `git pull`
  - (for HWs, “git pull upstream master”)
- `git add [files]`
- `git commit -m “a useful message”`
- `git push`
- `git diff`

Dealing with conflicts:

You must manually edit files that have conflicts
- (git doesn’t know which version is “right”)

More resources for learning git:

https://help.github.com/articles/git-and-github-learning-resources/
https://try.github.io/levels/1/challenges/1
https://courses.cs.washington.edu/courses/cse391/17sp/lectures/9/391Lecture09-Git-17sp.pdf
Git Demo!
DB Review & Using SQLite
Review: Database and DBMS

- What is a database?
- What is a DBMS?
Review: Database and DBMS

- What is a database?
  - Collection of organized files containing related data persisting over a long period of time

- What is a DBMS?
  - Program that allows for efficient management of large databases
SQL (Structural Query Language)

- Language designed for managing data held in a relational database management system (RDBMS)
- Declarative query language
- What can it do?
  - Data insert, delete, query, schema creation, etc.
SQLite: What is it?

- C library that implements a relational data management system (DBMS)

- sqlite3: a standalone program that can run programs and manage an SQLite database
SQLite Installation

Linux - Open a terminal, then run the command:

```
sudo apt-get install sqlite3
```

Mac -

1) Download Homebrew: instructions @ https://brew.sh/
2) Open a terminal, then run the command:

```
brew install sqlite3
```
SQLite Installation (con’t)

Windows -

1) Go to [https://www.sqlite.org/download.html](https://www.sqlite.org/download.html) and download the third option down (sqlite-tools-win32-x86-3200100.zip) under “Precompiled Binaries for Windows”
2) Extract files into directory of your choice
3) Add that directory to the environment variable “path”
Running SQLite

Linux/Mac - Open a terminal, then run the command:

```
sqlite3 [database]
```

where “database” is the name of the database

Windows -

1) In cmd, go to directory where you extracted sqlite3.exe files
2) Run the command: `sqlite3 [database]`
(not supported but) helpful tool: Datagrip

Download URL: https://www.jetbrains.com/datagrip/

Students can use it for free with .edu email addresses.
Questions on installation or running SQLite? Post on Piazza or visit us during OH!
SQLite: Special Commands

**.help** - list other . commands

**.header on/off** - show/hide column headers in query results

**.mode [mode type]** - change how to separate the columns in each row/tuple (for better formatting)

  Mode type examples: csv, tabs, line
SQLite: Basic SQL Statements

`CREATE TABLE`: creates a new table

[ex] `CREATE TABLE tableName (columnName int, ... );`
**INSERT INTO**: inserts new data into table

[ex] `INSERT INTO tableName VALUES (value1, ...);`
SELECT: gets existing data from table

[ex] SELECT columnName FROM tableName;
**UPDATE**: updates data in table

[ex] UPDATE tableName
    SET ....
    WHERE [condition];
DELETE: deletes data in table

[ex] DELETE FROM tableName
    WHERE [condition];
SQLite: Special Operators

DATE operator: lets you work with dates and times; declare as varchar (see hw1 documentation)

[ex] SELECT * FROM tableName WHERE dateColumn = 'YYYY-MM-DD';

SELECT * FROM tableName WHERE dateColumn < DATE('now', '-1 month');

Other operators: LIKE, LENGTH(string), SUBSTR(string, start index, end index), etc.
● WHERE clause - filter records
● AND, OR operator - filter records based on more than one condition
● LIKE operator - used in a WHERE clause to search for a specified pattern in a column
● AS - give an alias name to a table or a column
● Relational operators: =, >, >=, <, <=
SQL Demo!
SQL Programming Style

Make sure to alias all of your tables in the FROM clause

Although it is not necessary when trying out on sqlite3, it is considered bad style in this course to not do so

Ex: SELECT * FROM Students AS S WHERE S.name = ‘Johnny’

This will be especially important when writing queries with self joins (more than one instance of the same table) in future homework assignments

You can check to see if your hw queries follow these guidelines using the cosette tool

Cosette query checker: https://demo.cosette.cs.washington.edu/
Didn’t understand everything? That’s okay! This was just a preview.

SQL basics will be explained further in lecture before your homework is due.

*Post on Piazza or come to OH if you have any questions!