W4111 Spring 2026 Homework 1a

Introduction

Homework Overview

There are three parts to the homework.

- Part 1 walks you through the setup of your personal computer that is necessary for this course. You demonstrate completion by inserting screenshots or running code cells that demonstrate you successfully completed setup.

- Part 2 is a set of written questions that demonstrate you studied and understand lecture 1 material from the course and from the lecture 1 slides for the recommended textbook.

- Part 3 is a set of practical questions demonstrating basic knowledge of relational algebra and SQL.

This notebook defines only part 1. We will release the remainder of homework 1 after lecture 2 because the homework requires knowing material from the lecture

Part 1 — Setup

PyCharm

Install and Start PyCharm

Install the most recent version of PyCharm. There is online installation documentation and are several tutorials.

Students can signup for/register for a free, one year license to all JetBrains development tools. Please apply for/register for a license using your Columbia University email. Once your registration is confirmed, please activate PyCharm. The documentation explains how to register your product.

Start PyCharm.

Note: You may use a different integrated development environment (IDE) for Python if you prefer. For the HW, simply put the screenshots in the places that request PyCharm screenshots.

Clone Course Project

Follow the online instructions for cloning a GitHub project in PyCharm.

You will choose a local directory where you want to store your clone of the project.

The HTTPS URL for cloning the repository is https://github.com/donald-f-ferguson/W4111-Intro-to-Databases-Spring-2026.

Initialize the Virtual Environment and Jupyter Notebook Environment

Follow the online instructions to create a new Python environment for the project. Once you have setup the Python environment, open a terminal window inside PyCharm. You can open a terminal window by clicking on the icon in the left sidebar. The icon is highlighted in blue in the image below.

You will see a command prompt. The specific prompt you see will vary based on the operating system you are using.

PyCharm Terminal Window

In the terminal window, type the command `pip install jupyter`

When the installation completes, type the command `jupyter notebook` This will start a Jupyter Notebook environment on your PC. A browser window will open that should something like ... ...

Jupyter Notebook

Navigate to the folder `Homework/HW1` and open the Jupyter Notebook `W4111-Spring-2026-002-HW1a.ipynb`.

Note: If you do not see the directories. 1) Stop jupyter notebook. 2) In the terminal window, navigate to your home directory. 3) Start jupyter notebook. 4) In the browser windown, navigate to the directory where you cloned the project. 5) Navigate to the folder containing the notebook and open it.

Using Jupyter Notebook

Getting Started

There are several good, online introductions to using Jupyter notebooks. For example, How to Use Jupyter Notebook: A Beginner's Tutorial. There are many, many other ones. If you are unfamiliar with Jupyter notebooks, please follow one of the tutorials.

You will complete the homework in the notebook you opened.

Complete Personal Information

Rename this notebook file to "xyz1234-W4111-Fall-2025-002-HW1.ipynb" where `xyz1234` is your UNI. This replaces `dff9` in the implementation template. We suggest that you accomplish this by making a copy of the notebook and renaming the copy. This allows you to retain the original for reference.

In the table below, replace Professor Ferguson's personal information with your personal information by editing the markdown cell and table.

Field	Value
UNI	on2207
Last name	Nardin
First name	Oliviero

Demonstrate PyCharm

Take a screen capture of your PyCharm window. Copy the file into the same directory as this notebook. Edit the markdown cell below to display your image. **Make sure that the terminal window shows that you are in a directory on your laptop.**

My PyCharm

MySQL

Install MySQL Server Community Edition. There are several ways to install and several online tutorials. Follow one of the online tutorials or instructions that are appropriate for your operating system. *REMEMBER THE USER NAME AND PASSWORD FOR THE ROOT ACCOUNT.*

Note: If you have an old version of MacOS, Linux or Windows, *make sure you choose a version of MySQL that is compatible with your operating system version.* You may have

to get an older version from the archive or update your OS version.

After completing the installation, open a new terminal window. You can use a native window or open a new window in PyCharm.

Type the command `mysql -u root -p`. If the command is not found, you can navigate to the installation directory for MySQL and reenter the command.

- The installation directory on a Mac is usually `/usr/local/mysql/bin.` If you used Homebrew to install, the directory may be different.
- On Windows, the directory is usually `C:\Program Files\MySQL\MySQL Server X.Y\bin` where `X.Y` is the version of MySQL you installed.

Enter the password you set for `root.` This should open a command prompt for MySQL. Enter the command `show databases;` Your list of databases will be much shorter than mine.

Take a screenshot and replace the image below with your screenshot. Please make sure that your screenshot shows a directory indicating that the screenshot was taken on your PC. You can do this by navigating to your home directory and printing the working directory.

 **No description has been provided for this image**

My PyCharm

ipython-sql

Execute the Python cell below to install ipython-sql. Your status and progress messages will be different from mine. As long as they complete without an error, you should be fine.

There are several online documentation and tutorial pages that explain the installation process.

In [2]:
```
%pip install pymysql
%pip install sqlalchemy
%pip install ipython-sql
```

```
Collecting pymysql
  Downloading pymysql-1.1.2-py3-none-any.whl.metadata (4.3 kB)
Downloading pymysql-1.1.2-py3-none-any.whl (45 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.2
Note: you may need to restart the kernel to use updated packages.
Collecting sqlalchemy
  Downloading sqlalchemy-2.0.46-cp39-cp39-macosx_11_0_arm64.whl.metadata (9.
5 kB)
Requirement already satisfied: typing-extensions>=4.6.0 in /Users/olivierona
rdin/Desktop/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages
(from sqlalchemy) (4.15.0)
Downloading sqlalchemy-2.0.46-cp39-cp39-macosx_11_0_arm64.whl (2.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.2/2.2 MB 5.8 MB/s  0:00:00 eta
0:00:01
Installing collected packages: sqlalchemy
Successfully installed sqlalchemy-2.0.46
Note: you may need to restart the kernel to use updated packages.
Collecting ipython-sql
  Downloading ipython_sql-0.5.0-py3-none-any.whl.metadata (17 kB)
Collecting prettytable (from ipython-sql)
  Downloading prettytable-3.16.0-py3-none-any.whl.metadata (33 kB)
Requirement already satisfied: ipython in /Users/olivieronardin/Desktop/Colu
mbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ipython-sq
l) (8.18.1)
Requirement already satisfied: sqlalchemy>=2.0 in /Users/olivieronardin/Desk
top/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ip
ython-sql) (2.0.46)
Collecting sqlparse (from ipython-sql)
  Downloading sqlparse-0.5.5-py3-none-any.whl.metadata (4.7 kB)
Requirement already satisfied: six in /Users/olivieronardin/Desktop/Columbi
a/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ipython-sql)
(1.17.0)
Collecting ipython-genutils (from ipython-sql)
  Downloading ipython_genutils-0.2.0-py2.py3-none-any.whl.metadata (755 byte
s)
Requirement already satisfied: typing-extensions>=4.6.0 in /Users/olivierona
rdin/Desktop/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages
(from sqlalchemy>=2.0->ipython-sql) (4.15.0)
Requirement already satisfied: decorator in /Users/olivieronardin/Desktop/Co
lumbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ipython-
>ipython-sql) (5.2.1)
Requirement already satisfied: jedi>=0.16 in /Users/olivieronardin/Desktop/C
olumbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ipython
->ipython-sql) (0.19.2)
Requirement already satisfied: matplotlib-inline in /Users/olivieronardin/De
sktop/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from
ipython->ipython-sql) (0.2.1)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in /Users/olivi
eronardin/Desktop/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-pac
kages (from ipython->ipython-sql) (3.0.52)
Requirement already satisfied: pygments>=2.4.0 in /Users/olivieronardin/Desk
top/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ip
ython->ipython-sql) (2.19.2)
Requirement already satisfied: stack-data in /Users/olivieronardin/Desktop/C
olumbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ipython
```

```
->ipython-sql) (0.6.3)
Requirement already satisfied: traitlets>=5 in /Users/olivieronardin/Deskto
p/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ipyt
hon->ipython-sql) (5.14.3)
Requirement already satisfied: exceptiongroup in /Users/olivieronardin/Deskt
op/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ipy
thon->ipython-sql) (1.3.1)
Requirement already satisfied: pexpect>4.3 in /Users/olivieronardin/Desktop/
Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from ipytho
n->ipython-sql) (4.9.0)
Requirement already satisfied: wcwidth in /Users/olivieronardin/Desktop/Colu
mbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from prompt-too
lkit<3.1.0,>=3.0.41->ipython->ipython-sql) (0.4.0)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /Users/olivieronardin/
Desktop/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (fro
m jedi>=0.16->ipython->ipython-sql) (0.8.5)
Requirement already satisfied: ptyprocess>=0.5 in /Users/olivieronardin/Desk
top/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from pe
xpect>4.3->ipython->ipython-sql) (0.7.0)
Requirement already satisfied: executing>=1.2.0 in /Users/olivieronardin/Des
ktop/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from s
tack-data->ipython->ipython-sql) (2.2.1)
Requirement already satisfied: asttokens>=2.1.0 in /Users/olivieronardin/Des
ktop/Columbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from s
tack-data->ipython->ipython-sql) (3.0.1)
Requirement already satisfied: pure-eval in /Users/olivieronardin/Desktop/Co
lumbia/Spring 2026/COMSW4111/venv/lib/python3.9/site-packages (from stack-da
ta->ipython->ipython-sql) (0.2.3)
Downloading ipython_sql-0.5.0-py3-none-any.whl (20 kB)
Downloading ipython_genutils-0.2.0-py2.py3-none-any.whl (26 kB)
Downloading prettytable-3.16.0-py3-none-any.whl (33 kB)
Downloading sqlparse-0.5.5-py3-none-any.whl (46 kB)
Installing collected packages: ipython-genutils, sqlparse, prettytable, ipyt
hon-sql
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4/4 [ipython-sql]
Successfully installed ipython-genutils-0.2.0 ipython-sql-0.5.0 prettytable-
3.16.0 sqlparse-0.5.5
Note: you may need to restart the kernel to use updated packages.
```

Execute the following Python cell. Your result may be slightly different. You are fine as long as you do not get an error message.

In [4]:
```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

Set the proper root user ID and password for MySQL in the python cell below.

In [5]:
```
mysql_root_user = 'root'
mysql_root_password = 'Nova2014!'

mysql_url = f"mysql+pymysql://{mysql_root_user}:{mysql_root_password}@localh
```

In [6]:
```
mysql_url
```

Out[6]: 'mysql+pymysql://root:Nova2014!@localhost'

Run the following cell.

In [7]:
```
%sql $mysql_url
```

Run the following cell. Your list of databases should be different.

In [8]:
```
%config SqlMagic.style = '_DEPRECATED_DEFAULT'
%sql show databases;
```

```
 * mysql+pymysql://root:***@localhost
4 rows affected.
```

Out[8]:

Database
information_schema
mysql
performance_schema
sys

If you get here, `ipython-sql` is installed and working.

DataGrip

Install DataGrip. There is online documentation and are several tutorials.

Follow the online instructions to register your installation using the ID you set up when getting a free license.

Start DataGrip.

Follow the online instructions to create a connection to your local instance of MySQL. You may have to choose the option to install the database drivers.

Open a query console on your local database connection.

In the query console, enter the command. `SHOW VARIABLES WHERE Variable_name = 'hostname';` Take a screenshot and replace the image below with your screenshot. Make sure the image shows the query execution result.


No description has been provided for this image

DataGrip

Sample Database

Execute the following ipython-sql cell. You must be successfully connect to MySQL using the previous `%sql` command.

As long as you do not get any error messages, you have successfully create the sample database.

In [9]:
```
%%sql

drop schema if exists db_book;
create schema db_book;
use db_book;

drop table if exists prereq;
drop table if exists time_slot;
drop table if exists advisor;
drop table if exists takes;
drop table if exists student;
drop table if exists teaches;
drop table if exists section;
drop table if exists instructor;
drop table if exists course;
drop table if exists department;
drop table if exists classroom;



create table classroom
        (building               varchar(15),
         room_number            varchar(7),
         capacity               numeric(4,0),
         primary key (building, room_number)
        );

create table department
        (dept_name              varchar(20),
         building               varchar(15),
         budget                 numeric(12,2) check (budget > 0),
         primary key (dept_name)
        );

create table course
        (course_id              varchar(8),
         title                  varchar(50),
         dept_name              varchar(20),
         credits                numeric(2,0) check (credits > 0),
         primary key (course_id),
         foreign key (dept_name) references department (dept_name)
                on delete set null
        );

create table instructor
        (ID                     varchar(5),
         name                   varchar(20) not null,
         dept_name              varchar(20),
```

```
        salary                  numeric(8,2) check (salary > 29000),
        primary key (ID),
        foreign key (dept_name) references department (dept_name)
               on delete set null
       );

create table section
       (course_id               varchar(8),
        sec_id                  varchar(8),
        semester                varchar(6)
               check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
        year                    numeric(4,0) check (year > 1701 and year < 2
        building                varchar(15),
        room_number             varchar(7),
        time_slot_id            varchar(4),
        primary key (course_id, sec_id, semester, year),
        foreign key (course_id) references course (course_id)
               on delete cascade,
        foreign key (building, room_number) references classroom (building,
               on delete set null
       );

create table teaches
       (ID                      varchar(5),
        course_id               varchar(8),
        sec_id                  varchar(8),
        semester                varchar(6),
        year                    numeric(4,0),
        primary key (ID, course_id, sec_id, semester, year),
        foreign key (course_id, sec_id, semester, year) references section
               on delete cascade,
        foreign key (ID) references instructor (ID)
               on delete cascade
       );

create table student
       (ID                      varchar(5),
        name                    varchar(20) not null,
        dept_name               varchar(20),
        tot_cred                numeric(3,0) check (tot_cred >= 0),
        primary key (ID),
        foreign key (dept_name) references department (dept_name)
               on delete set null
       );

create table takes
       (ID                      varchar(5),
        course_id               varchar(8),
        sec_id                  varchar(8),
        semester                varchar(6),
        year                    numeric(4,0),
        grade                   varchar(2),
        primary key (ID, course_id, sec_id, semester, year),
        foreign key (course_id, sec_id, semester, year) references section
               on delete cascade,
        foreign key (ID) references student (ID)
```

```
                on delete cascade
        );

create table advisor
        (s_ID                   varchar(5),
         i_ID                   varchar(5),
         primary key (s_ID),
         foreign key (i_ID) references instructor (ID)
                on delete set null,
         foreign key (s_ID) references student (ID)
                on delete cascade
        );

create table time_slot
        (time_slot_id           varchar(4),
         day                    varchar(1),
         start_hr               numeric(2) check (start_hr >= 0 and start_hr
         start_min              numeric(2) check (start_min >= 0 and start_m
         end_hr                 numeric(2) check (end_hr >= 0 and end_hr < 2
         end_min                numeric(2) check (end_min >= 0 and end_min <
         primary key (time_slot_id, day, start_hr, start_min)
        );

create table prereq
        (course_id              varchar(8),
         prereq_id              varchar(8),
         primary key (course_id, prereq_id),
         foreign key (course_id) references course (course_id)
                on delete cascade,
         foreign key (prereq_id) references course (course_id)
        );

delete from prereq;
delete from time_slot;
delete from advisor;
delete from takes;
delete from student;
delete from teaches;
delete from section;
delete from instructor;
delete from course;
delete from department;
delete from classroom;
insert into classroom values ('Packard', '101', '500');
insert into classroom values ('Painter', '514', '10');
insert into classroom values ('Taylor', '3128', '70');
insert into classroom values ('Watson', '100', '30');
insert into classroom values ('Watson', '120', '50');
insert into department values ('Biology', 'Watson', '90000');
insert into department values ('Comp. Sci.', 'Taylor', '100000');
insert into department values ('Elec. Eng.', 'Taylor', '85000');
insert into department values ('Finance', 'Painter', '120000');
insert into department values ('History', 'Painter', '50000');
insert into department values ('Music', 'Packard', '80000');
insert into department values ('Physics', 'Watson', '70000');
insert into course values ('BIO-101', 'Intro. to Biology', 'Biology', '4');
```

```
insert into course values ('BIO-301', 'Genetics', 'Biology', '4');
insert into course values ('BIO-399', 'Computational Biology', 'Biology', '3
insert into course values ('CS-101', 'Intro. to Computer Science', 'Comp. Sc
insert into course values ('CS-190', 'Game Design', 'Comp. Sci.', '4');
insert into course values ('CS-315', 'Robotics', 'Comp. Sci.', '3');
insert into course values ('CS-319', 'Image Processing', 'Comp. Sci.', '3');
insert into course values ('CS-347', 'Database System Concepts', 'Comp. Sci.
insert into course values ('EE-181', 'Intro. to Digital Systems', 'Elec. Eng
insert into course values ('FIN-201', 'Investment Banking', 'Finance', '3');
insert into course values ('HIS-351', 'World History', 'History', '3');
insert into course values ('MU-199', 'Music Video Production', 'Music', '3')
insert into course values ('PHY-101', 'Physical Principles', 'Physics', '4')
insert into instructor values ('10101', 'Srinivasan', 'Comp. Sci.', '65000')
insert into instructor values ('12121', 'Wu', 'Finance', '90000');
insert into instructor values ('15151', 'Mozart', 'Music', '40000');
insert into instructor values ('22222', 'Einstein', 'Physics', '95000');
insert into instructor values ('32343', 'El Said', 'History', '60000');
insert into instructor values ('33456', 'Gold', 'Physics', '87000');
insert into instructor values ('45565', 'Katz', 'Comp. Sci.', '75000');
insert into instructor values ('58583', 'Califieri', 'History', '62000');
insert into instructor values ('76543', 'Singh', 'Finance', '80000');
insert into instructor values ('76766', 'Crick', 'Biology', '72000');
insert into instructor values ('83821', 'Brandt', 'Comp. Sci.', '92000');
insert into instructor values ('98345', 'Kim', 'Elec. Eng.', '80000');
insert into section values ('BIO-101', '1', 'Summer', '2017', 'Painter', '51
insert into section values ('BIO-301', '1', 'Summer', '2018', 'Painter', '51
insert into section values ('CS-101', '1', 'Fall', '2017', 'Packard', '101',
insert into section values ('CS-101', '1', 'Spring', '2018', 'Packard', '101
insert into section values ('CS-190', '1', 'Spring', '2017', 'Taylor', '3128
insert into section values ('CS-190', '2', 'Spring', '2017', 'Taylor', '3128
insert into section values ('CS-315', '1', 'Spring', '2018', 'Watson', '120'
insert into section values ('CS-319', '1', 'Spring', '2018', 'Watson', '100'
insert into section values ('CS-319', '2', 'Spring', '2018', 'Taylor', '3128
insert into section values ('CS-347', '1', 'Fall', '2017', 'Taylor', '3128',
insert into section values ('EE-181', '1', 'Spring', '2017', 'Taylor', '3128
insert into section values ('FIN-201', '1', 'Spring', '2018', 'Packard', '10
insert into section values ('HIS-351', '1', 'Spring', '2018', 'Painter', '51
insert into section values ('MU-199', '1', 'Spring', '2018', 'Packard', '101
insert into section values ('PHY-101', '1', 'Fall', '2017', 'Watson', '100',
insert into teaches values ('10101', 'CS-101', '1', 'Fall', '2017');
insert into teaches values ('10101', 'CS-315', '1', 'Spring', '2018');
insert into teaches values ('10101', 'CS-347', '1', 'Fall', '2017');
insert into teaches values ('12121', 'FIN-201', '1', 'Spring', '2018');
insert into teaches values ('15151', 'MU-199', '1', 'Spring', '2018');
insert into teaches values ('22222', 'PHY-101', '1', 'Fall', '2017');
insert into teaches values ('32343', 'HIS-351', '1', 'Spring', '2018');
insert into teaches values ('45565', 'CS-101', '1', 'Spring', '2018');
insert into teaches values ('45565', 'CS-319', '1', 'Spring', '2018');
insert into teaches values ('76766', 'BIO-101', '1', 'Summer', '2017');
insert into teaches values ('76766', 'BIO-301', '1', 'Summer', '2018');
insert into teaches values ('83821', 'CS-190', '1', 'Spring', '2017');
insert into teaches values ('83821', 'CS-190', '2', 'Spring', '2017');
insert into teaches values ('83821', 'CS-319', '2', 'Spring', '2018');
insert into teaches values ('98345', 'EE-181', '1', 'Spring', '2017');
insert into student values ('00128', 'Zhang', 'Comp. Sci.', '102');
insert into student values ('12345', 'Shankar', 'Comp. Sci.', '32');
```

```
insert into student values ('19991', 'Brandt', 'History', '80');
insert into student values ('23121', 'Chavez', 'Finance', '110');
insert into student values ('44553', 'Peltier', 'Physics', '56');
insert into student values ('45678', 'Levy', 'Physics', '46');
insert into student values ('54321', 'Williams', 'Comp. Sci.', '54');
insert into student values ('55739', 'Sanchez', 'Music', '38');
insert into student values ('70557', 'Snow', 'Physics', '0');
insert into student values ('76543', 'Brown', 'Comp. Sci.', '58');
insert into student values ('76653', 'Aoi', 'Elec. Eng.', '60');
insert into student values ('98765', 'Bourikas', 'Elec. Eng.', '98');
insert into student values ('98988', 'Tanaka', 'Biology', '120');
insert into takes values ('00128', 'CS-101', '1', 'Fall', '2017', 'A');
insert into takes values ('00128', 'CS-347', '1', 'Fall', '2017', 'A-');
insert into takes values ('12345', 'CS-101', '1', 'Fall', '2017', 'C');
insert into takes values ('12345', 'CS-190', '2', 'Spring', '2017', 'A');
insert into takes values ('12345', 'CS-315', '1', 'Spring', '2018', 'A');
insert into takes values ('12345', 'CS-347', '1', 'Fall', '2017', 'A');
insert into takes values ('19991', 'HIS-351', '1', 'Spring', '2018', 'B');
insert into takes values ('23121', 'FIN-201', '1', 'Spring', '2018', 'C+');
insert into takes values ('44553', 'PHY-101', '1', 'Fall', '2017', 'B-');
insert into takes values ('45678', 'CS-101', '1', 'Fall', '2017', 'F');
insert into takes values ('45678', 'CS-101', '1', 'Spring', '2018', 'B+');
insert into takes values ('45678', 'CS-319', '1', 'Spring', '2018', 'B');
insert into takes values ('54321', 'CS-101', '1', 'Fall', '2017', 'A-');
insert into takes values ('54321', 'CS-190', '2', 'Spring', '2017', 'B+');
insert into takes values ('55739', 'MU-199', '1', 'Spring', '2018', 'A-');
insert into takes values ('76543', 'CS-101', '1', 'Fall', '2017', 'A');
insert into takes values ('76543', 'CS-319', '2', 'Spring', '2018', 'A');
insert into takes values ('76653', 'EE-181', '1', 'Spring', '2017', 'C');
insert into takes values ('98765', 'CS-101', '1', 'Fall', '2017', 'C-');
insert into takes values ('98765', 'CS-315', '1', 'Spring', '2018', 'B');
insert into takes values ('98988', 'BIO-101', '1', 'Summer', '2017', 'A');
insert into takes values ('98988', 'BIO-301', '1', 'Summer', '2018', null);
insert into advisor values ('00128', '45565');
insert into advisor values ('12345', '10101');
insert into advisor values ('23121', '76543');
insert into advisor values ('44553', '22222');
insert into advisor values ('45678', '22222');
insert into advisor values ('76543', '45565');
insert into advisor values ('76653', '98345');
insert into advisor values ('98765', '98345');
insert into advisor values ('98988', '76766');
insert into time_slot values ('A', 'M', '8', '0', '8', '50');
insert into time_slot values ('A', 'W', '8', '0', '8', '50');
insert into time_slot values ('A', 'F', '8', '0', '8', '50');
insert into time_slot values ('B', 'M', '9', '0', '9', '50');
insert into time_slot values ('B', 'W', '9', '0', '9', '50');
insert into time_slot values ('B', 'F', '9', '0', '9', '50');
insert into time_slot values ('C', 'M', '11', '0', '11', '50');
insert into time_slot values ('C', 'W', '11', '0', '11', '50');
insert into time_slot values ('C', 'F', '11', '0', '11', '50');
insert into time_slot values ('D', 'M', '13', '0', '13', '50');
insert into time_slot values ('D', 'W', '13', '0', '13', '50');
insert into time_slot values ('D', 'F', '13', '0', '13', '50');
insert into time_slot values ('E', 'T', '10', '30', '11', '45 ');
insert into time_slot values ('E', 'R', '10', '30', '11', '45 ');
```

```
insert into time_slot values ('F', 'T', '14', '30', '15', '45 ');
insert into time_slot values ('F', 'R', '14', '30', '15', '45 ');
insert into time_slot values ('G', 'M', '16', '0', '16', '50');
insert into time_slot values ('G', 'W', '16', '0', '16', '50');
insert into time_slot values ('G', 'F', '16', '0', '16', '50');
insert into time_slot values ('H', 'W', '10', '0', '12', '30');
insert into prereq values ('BIO-301', 'BIO-101');
insert into prereq values ('BIO-399', 'BIO-101');
insert into prereq values ('CS-190', 'CS-101');
insert into prereq values ('CS-315', 'CS-101');
insert into prereq values ('CS-319', 'CS-101');
insert into prereq values ('CS-347', 'CS-101');
insert into prereq values ('EE-181', 'PHY-101');
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
1 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
```

```
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
```

```
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
```

```
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
1 rows affected.
```

Out[9]: []

Execute the following cell to verify that you have correctly created and loaded the sample database.

In [10]:
```
%sql select * from db_book.student where dept_name='Comp. Sci.'
```

```
 * mysql+pymysql://root:***@localhost
4 rows affected.
```

Out[10]:

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
54321	Williams	Comp. Sci.	54
76543	Brown	Comp. Sci.	58

Summary

If you were able to accomplish all of the tasks above, you successfully completed part 1 of homework 1.

Create PDF

1. Use the `File -> Save and Export Notebook as -> HTML` option to save your notebook as an HTML file.
2. Copy the created HTML file to the same directory as the notebook and images you created.
3. Open the HTML file in a browser.
4. Using the *browser's* `File -> Print` option print the HTML file to a PDF. This is your submission format.
5. Follow the submission instructions on Ed Discussion to submit your file to GradeScope.

In [ ]:
```
```