

dff9-W4111-Fall-2025-002-HW1.ipynb

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.

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.

Clone Course Project

Follow the [online instructions](#) for [cloning](#) a GitHub project in PyCharm.

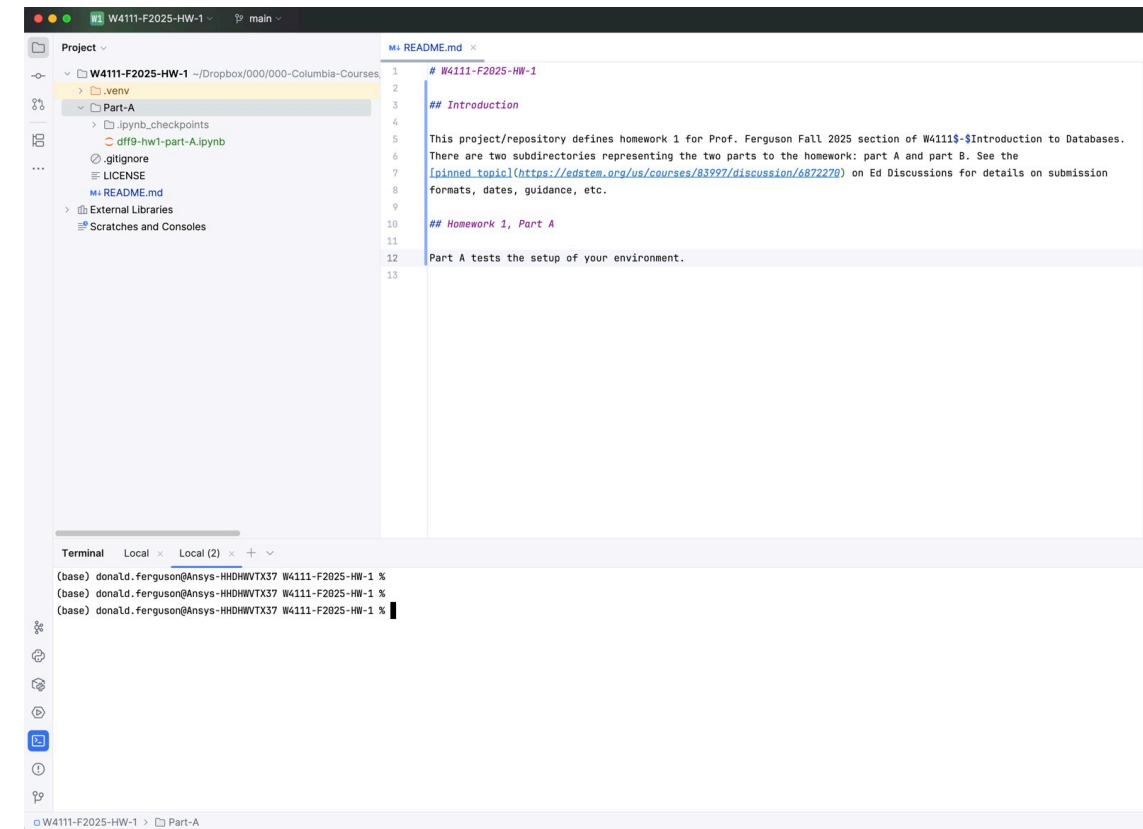
You will choose a local directory where you want to store your clone of the project.

The HTTPS URL for cloning the repository is <https://github.com/donald-ferguson/W4111-Introduction-to-Databases-New.git>.

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.

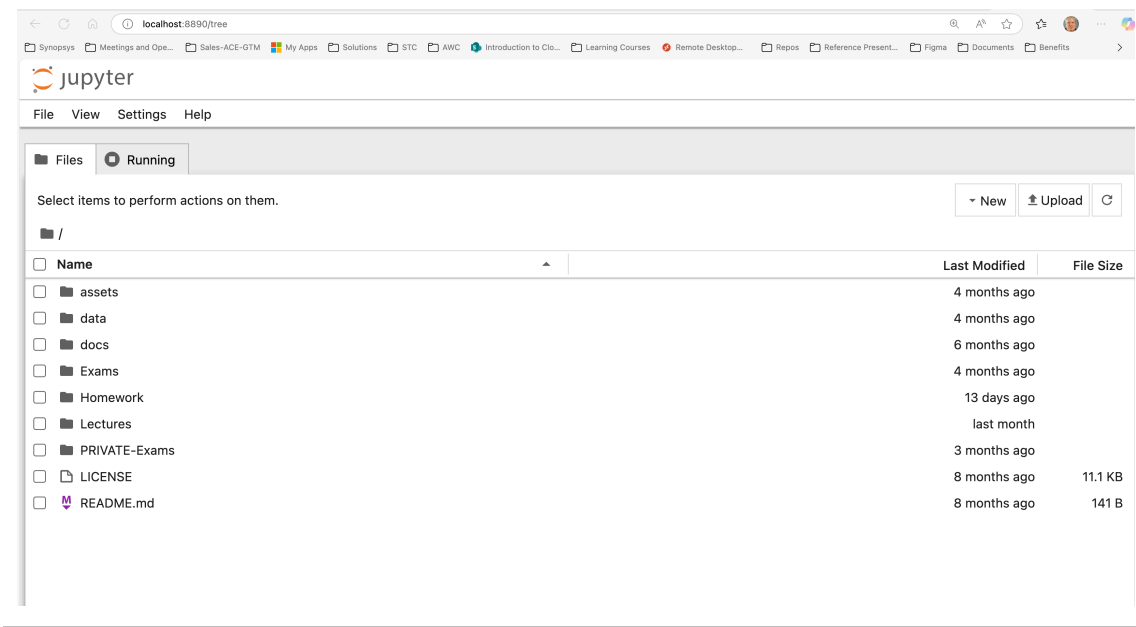
You will see a command prompt. The specific prompt you see will vary based on the operating system you are using.



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/F2025/HW1 and open this Jupyter Notebook.

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 window, 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

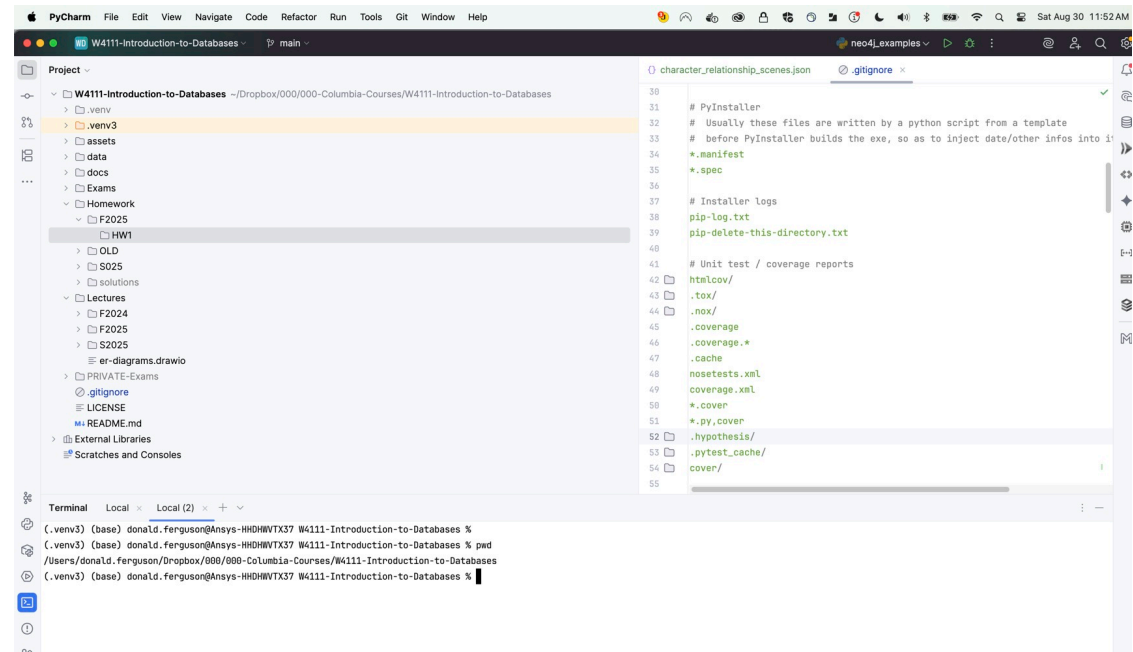
Rename this notebook file to "dff9-W4111-Fall-2025-002-HW1.ipynb" where xyz1234 is your UNI. This replaces dff9 in the implementation template. We suggest that you accomplish this by making a copy of the notebook and renaming the copy. This allows you to retain the original for reference.

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

Field	Value
UNI	dff9
Last name	Ferguson
First name	Donald

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 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.

The screenshot shows a PyCharm IDE window with a terminal open. The terminal output shows the user navigating to the MySQL bin directory and running the command to show databases. The output lists several databases, including 'columbiamodel' and 'course_management'.

```

PyCharm File Edit View Navigate Code Refactor Run Tools Git Window Help
W4111-F2025-HW-1 ~ /Dropbox/000/000
Project W4111-F2025-HW-1
  .ipynb_checkpoints
  .venv
  README.md
  W4111-F2025-HW-1
  # W4111-F2025-HW-1
  ## Introduction
Terminal Local Local (2)
(.venv) (base) donald.ferguson@Ansys-RH0HWITX37 bin % cd ~
(.venv) (base) donald.ferguson@Ansys-RH0HWITX37 ~ %
(.venv) (base) donald.ferguson@Ansys-RH0HWITX37 ~ % pwd
/Users/donald.ferguson
(.venv) (base) donald.ferguson@Ansys-RH0HWITX37 ~ % cd /usr/local/mysql/bin
(.venv) (base) donald.ferguson@Ansys-RH0HWITX37 bin % ./mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| $output_schema |
| cannonical_project |
| classic_models_oLap_simple |
| classic_models_oLap_v2 |
| classicmodels |
| classicmodels_final |
| columbiamodel |
| course_management |
+-----+

```

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

Requirement already satisfied: pymysql in ./venv/lib/python3.12/site-packages (1.1.2)

[notice] A new release of pip is available: 25.0.1 -> 25.2

[notice] To update, run: `pip install --upgrade pip`

Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: sqlalchemy in ./venv/lib/python3.12/site-packages (2.0.43)

Requirement already satisfied: typing-extensions>=4.6.0 in ./venv/lib/python3.12/site-packages (from sqlalchemy) (4.14.1)

[notice] A new release of pip is available: 25.0.1 -> 25.2

[notice] To update, run: `pip install --upgrade pip`

Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: ipython-sql in ./venv/lib/python3.12/site-packages (0.5.0)

Requirement already satisfied: prettytable in ./venv/lib/python3.12/site-packages (from ipython-sql) (3.16.0)

Requirement already satisfied: ipython in ./venv/lib/python3.12/site-packages (from ipython-sql) (9.4.0)

Requirement already satisfied: sqlalchemy>=2.0 in ./venv/lib/python3.12/site-packages (from ipython-sql) (2.0.43)

Requirement already satisfied: sqlparse in ./venv/lib/python3.12/site-packages (from ipython-sql) (0.5.3)

Requirement already satisfied: six in ./venv/lib/python3.12/site-packages (from ipython-sql) (1.17.0)

Requirement already satisfied: ipython-genutils in ./venv/lib/python3.12/site-packages (from ipython-sql) (0.2.0)

Requirement already satisfied: typing-extensions>=4.6.0 in ./venv/lib/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (4.14.1)

Requirement already satisfied: decorator in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (5.2.1)

Requirement already satisfied: ipython-pygments-lexers in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (1.1.1)

Requirement already satisfied: jedi>=0.16 in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (0.19.2)

Requirement already satisfied: matplotlib-inline in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (0.1.7)

Requirement already satisfied: pexpect>4.3 in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (4.9.0)

Requirement already satisfied: prompt_toolkit<3.1.0,>=3.0.41 in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (3.0.51)

Requirement already satisfied: pygments>=2.4.0 in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (2.19.2)

Requirement already satisfied: stack_data in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (0.6.3)

Requirement already satisfied: traitlets>=5.13.0 in ./venv/lib/python3.12/site-packages (from ipython->ipython-sql) (5.14.3)

Requirement already satisfied: wcwidth in ./venv/lib/python3.12/site-packages (from prettytable->ipython-sql) (0.2.13)

Requirement already satisfied: parso<0.9.0,>=0.8.4 in ./venv/lib/python3.12/site-packages (from jedi>=0.16->ipython->ipython-sql) (0.8.5)

Requirement already satisfied: ptyprocess>=0.5 in ./venv/lib/python3.12/site-packages (from pexpect>4.3->ipython->ipython-sql) (0.7.0)

Requirement already satisfied: executing>=1.2.0 in ./venv/lib/python3.12/site-packages (from stack_data->ipython->ipython-sql) (2.2.0)

Requirement already satisfied: asttokens>=2.1.0 in ./venv/lib/python3.12/site-packages (from stack_data->ipython->ipython-sql) (3.0.0)

Requirement already satisfied: pure-eval in ./venv/lib/python3.12/site-packages (from stack_data->ipython->ipython-sql) (0.2.3)

[notice] A new release of pip is available: 25.0.1 -> 25.2

[notice] To update, run: `pip install --upgrade pip`

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 [1]: `%load_ext sql`

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

In [2]: `mysql_root_user = 'root'`
`mysql_root_password = 'dbuserdbuser'`
`mysql_url = f"mysql+pymysql://{mysql_root_user}:{mysql_root_password}@localhost"`

In [3]: `mysql_url`

Out[3]: 'mysql+pymysql://root:dbuserdbuser@localhost'

Run the following cell.

In [4]: `%sql $mysql_url`

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

In [5]: `%config SqlMagic.style = '_DEPRECATED_DEFAULT'`
`%sql show databases;`

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

Out [5] :

Database

\$output_schema
cannonical_project
classic_models_olap_simple
classic_models_olap_v2
classicmodels
classicmodels_final
columbia_model
course_management
course_student_coupons
courseworks_videos
db_book
f2024_grades
F24_examples
f24_GoT
f24_imdb_clean
F24_IMDB_Raw
f24_imdb_raw_new
f24_project
f24_project_clean
fitness
hw3b4bSolution
hw4
hw5_common
import_web_data
information_schema
lahmansbaseballdb
lahmansfast
lecture_3_examples
lecture_6
lor_data
mysql
Northwind

Database

northwind_final
p1_database
performance_schema
pyspark
s2025_examples
s2025_hw3
s24_project
s25_hw3B
s25_hw3b_solution
s25_project
s25_project_fixed
s25_project_raw
sys
testdb
w4111_example_1
w4111_f24_final
w4111_f24_final_answers
w4111_f24_final_answers_imdb
w4111_final_exam_answers
w4111_project

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

DataGrip

Install [DataGrip](#). There is online documentation and are several tutorials.

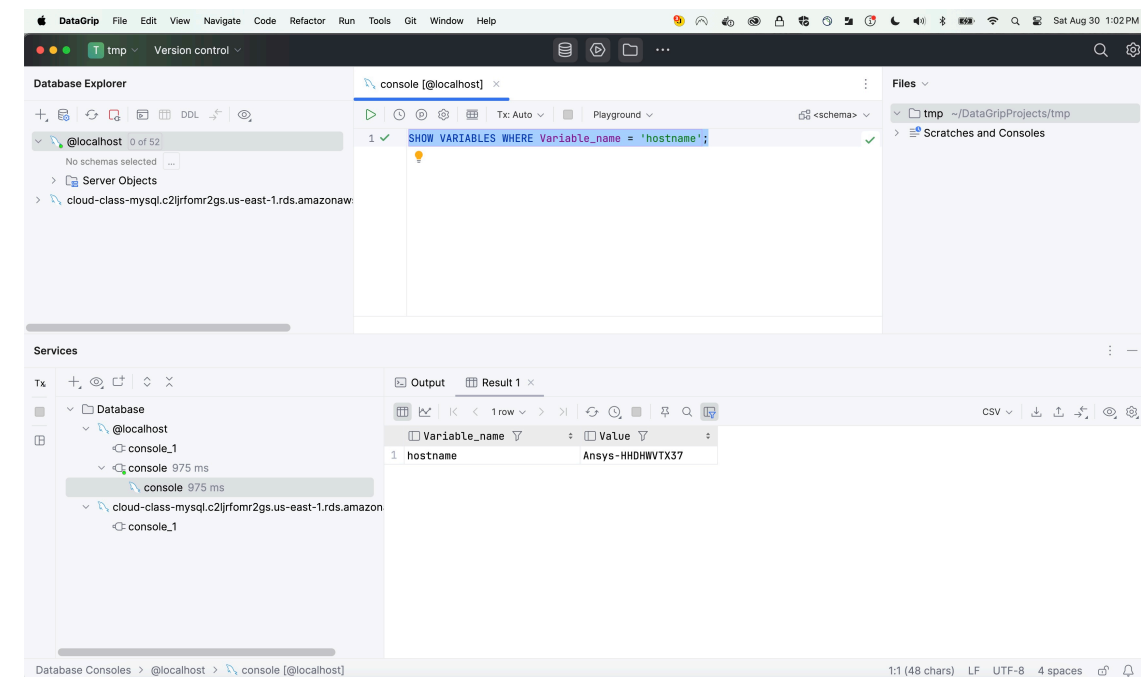
Follow the online instructions to register your installation using the ID you set up when getting a free license.

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.



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 [6]: `%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 < 2101),
    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, room_number)
        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 < 24),
    start_min    numeric(2) check (start_min >= 0 and start_min < 60),
    end_hr       numeric(2) check (end_hr >= 0 and end_hr < 24),
    end_min      numeric(2) check (end_min >= 0 and end_min < 60),
    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. Sci.', '4');
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.', '4');
insert into course values ('EE-181', 'Intro. to Digital Systems', 'Elec. Eng.', '3');
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', '101');
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');

```


17/28

18/28

19/28

```

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

```

Out [6]: []

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

In [7]: `%sql select * from db_book.student where dept_name='Comp. Sci.'`

```

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

```

Out [7]:

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.

Lecture 1 and Book Slides from Lecture 1 Questions

Q1

Question

In []:

Consider the applications used for managing courses and enrollments at Columbia, e.g. CourseWorks, Vergil. These applications use database management systems. Consider an alternate approach in which the university used a set of shared Google sheets to manage classes and enrollments. Faculty, administrators and students would edit the sheets to create classes, enroll in classes, etc. Identify five problems with the shared sheet approach and explain the features of a DBMS that solve the problem.

Answer

Q2

Question

What is data abstraction? What are the levels of data abstraction?

Answer

Q3

Question

Briefly explain the concepts of database *schema* and *instance*.

Answer

Q4

Question

Consider a scenario in which multiple users and programs access a database. For this scenario, what is an example of the benefits of *physical data independence*?

Answer

Q5

Question

What are the two types/classes of languages that a DBMS system provides.

Answer

Q6

Question

Briefly explain the concepts of *declarative* and *procedural* database manipulation languages. What is one benefit of declarative languages?

Answer

Q7

Question

What are the 3 levels of data modeling/entity relationship modeling?

Answer

Q8

Question

What is a feature of the DBMS *storage manager* that provides fast access to data in a database?

Answer

Q9

Question

Briefly explain/define the concept of a transaction.

Answer

Q10

Question

Consider Vergil. Is this a *two tier* or *three tier* database architecture?

Answer

Q11

Question

In the setup tasks from the previous section, you created a database schema and loaded data. You then wrote a simple query. Which tasks would a database administrator perform and which tasks would a database user perform?

Answer

Briefly explain the concepts of *unstructured*, *semi-structured* and *structured* data. Which type of data is a file holding an audio recording.

Q12

Question

For a file holding an audio recording, what would typical metadata be?

Answer

Q13

Question

What are "The 5 Vs" of data?

Answer

Q14

Question

Briefly explain the concepts of *entity set*, *relationship set* and *attributes*.

Answer

Q15

Question

Explain the relationship between the relational algebra and the SQL language.

Answer

Practical Questions

P1

Question

Consider the following description of a datamodel.

There are two entity sets with the following attributes:

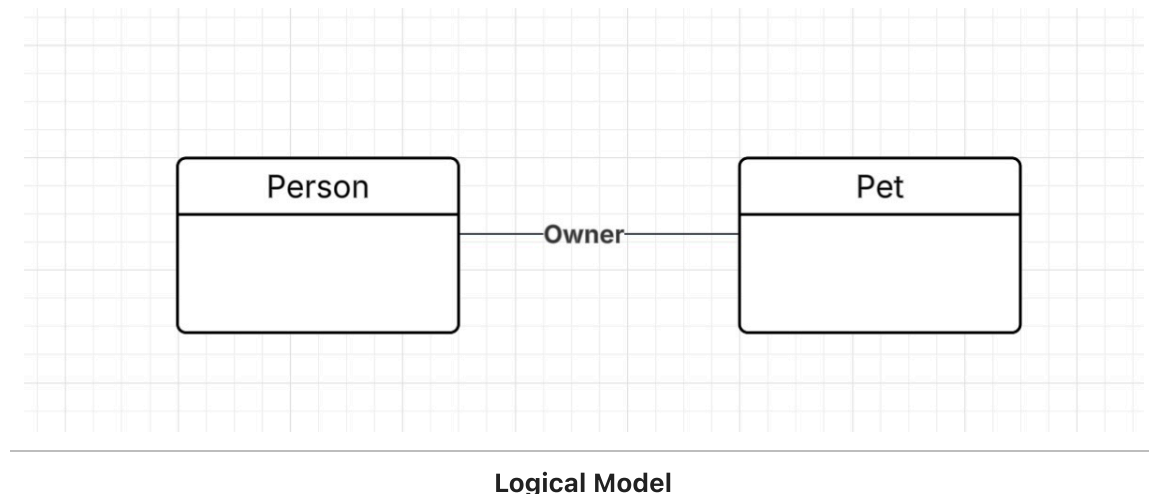
1. **Course**
 - **course_no**, which uniquely identifies a course.
 - **course_title**
 - **course_description**
2. **Section**
 - **call_no**, which uniquely identifies a section.
 - **course_no**
 - **section_no**
 - **semester**
 - **year**
 - **maximum_enrollment**

A section is a section of exactly one course. A course may have 0, 1 or many sections.

Using Crow's Foot Notation and Lucidchart or an equivalent tool, draw a logical datamodel for the scenario. Replace the image below with a screen capture of your diagram.

NOTE: The diagram below is a conceptual model. The question is asking you to define a logical model.

Answer



P2

Question

Using the [RelaX relational calculator](#) and the sample data that comes with the recommended textbook, write an algebra expression that computes the the **courses** that are in the "Comp. Sci." **department** and have **4** credits. Your answer should only contain **course_id**, **title** and **credits**.

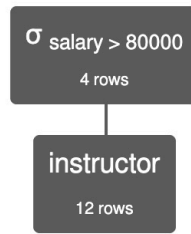
Your answer should be in the format below:

1. The text of your query.
2. A screen capture of the execution result.

The answer area below contains an example of the result format but for a different query.

Answer

σ salary > 80000 (instructor)



$\sigma_{\text{salary} > 80000}(\text{instructor})$

Execution time: 0 ms

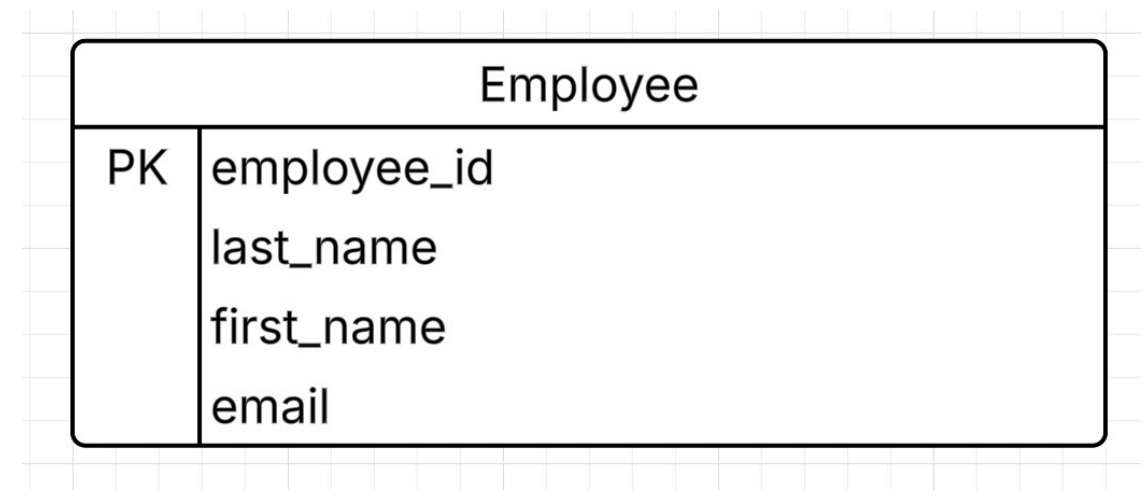
instructor.ID	instructor.name	instructor.dept_name	instructor.salary
12121	'Wu'	'Finance'	90000
22222	'Einstein'	'Physics'	95000
33456	'Gold'	'Physics'	87000
83821	'Brandt'	'Comp. Sci.'	92000

P2 Answer

P3

Question

The following is a simple Crow's Foot Diagram of a single entity set. Write and execute a `SQL CREATE TABLE` statement that defines a table implementing the diagram. You can assume that the data type for all attributes is `VARCHAR(64)`.



P3 Diagram

Answer

In [27]: `%%sql`

```
/*
    Put your create table statement below and execute this cell.
*/
```

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

Out[27]: `[]`

P4

Question

For the sample database you setup and loaded, write a `SQL SELECT` statement that returns the the courses that are in the "Comp. Sci." department and have 4 credits.

Answer

In [28]: `%%sql`

```
/*
    Write your SQL below and execute the cell.
*/
use db_book;
```

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

Out[28]: `[]`

P5

Question

For the sample database, write and execute a statement that adds the following professor to the instructor's table.

```
{
    ID: 666,
    name: "Ferguson",
    dept_name: "Comp. Sci.",
    salary: 0.00
}
```

Answer

```
In [ ]: %%sql

/*
    Write your SQL statement below and execute.
*/
```

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 [ ]:
```