Database Homework 1

1. What is a database management system and how do relational databases organize data?

   A Database Management System or DBMS, for short, is a set of programs or software that enables the creation, access, maintenance, management, and use of a computerized database. It is an environment that is convenient and efficient to use for a systematic way to store, retrieve, and manage data A DBMS also ensures data consistency, integrity, and security by controlling access to the database.

   A general purpose DBMS supports the definition, creation, querying, update, and administration of databases in accordance with some data model.

   Relational databases on the other hand, have sets of relations where a relation is an unordered set that contains the relationship of attributes that represent entities. Entities are made up of two parts, an instance which is a table with rows (records) and columns (fields), where each row represents a record, and each column represents a particular attribute of the record. There is also a schema which describes the data, such as specifying the name of the relation, and the name and type of each column. These tables are related to each other through keys, typically primary and foreign keys, enabling relationships between different tables. This structure helps in avoiding data redundancy and ensures the efficient retrieval of information.

2. What are four different types of DBMS table relationships? Give a brief explanation for each.

   The four types of DBMS table relationships are one-to-one, one-to-many, many-to-one, and many-to-many.

   A one-to-one relationship is a type of relationship, where a record in one table corresponds to exactly one record in another table. This is commonly used to split a large table into smaller ones for optimization or security purposes. An example of this can be one person and their unique passport information.

   A one-to-many relationship is a relationship where a record in one table can be related to multiple records in another table. This is the most common type of relationship. An example of this can be a customer can own multiple cars, but each car is linked to one customer.

   A many-to-one relationship is similar to the one-to-many relationship, however from the perspective of the other table. An example of this can be that many employees can belong to one department, but each department can have multiple employees.

   Lastly, a many-to-many relationship is a relationship in which the records in one table can relate to multiple records in another table, and vice versa. This is implemented using a junction table that contains foreign keys from both related tables. An example of this can be students and classes. A student can take multiple classes, and a class can have multiple students.

3. What is the primary key and why is it important?

   A primary key is a unique identifier for each record in a database table. It ensures that each record can be uniquely identified and helps maintain data integrity. The primary key cannot contain NULL values and must contain unique values for each row. It is important to have a primary key in each record of a database table, because it ensures uniqueness, meaning that each record in the table is distinct. It enforces data integrity because it helps avoid duplicate records in a table. It allows efficient access, facilitating quick searching, sorting, and indexing of data. It can also help establish relationships in which other tables can refer to the primary key using a foreign key.

4. Some of the Fordham University databases/applications represent the year/semester attribute of a section in the form “2023_2”. The first four characters are the academic year, and the last character is the semester (1, 2, or 3). The data type for this attribute might be char(6). Using this example, explain the concepts of domain and atomic domain. How is the domain different from type?

   A domain is the set of allowable values for a particular attribute. A domain has a logical definition, and has a data-type of format defined for it. In the example “2023_2”, the domain for the “year/semester” attribute might be constrained to a specific format, such as “YYYY_S”, where “YYYY” is a four-digit year and “S” is a one-digit semester (1, 2 or 3 representing the Fall, Spring and Summer). The domain specifies that only these types of values can be assigned to the attribute.

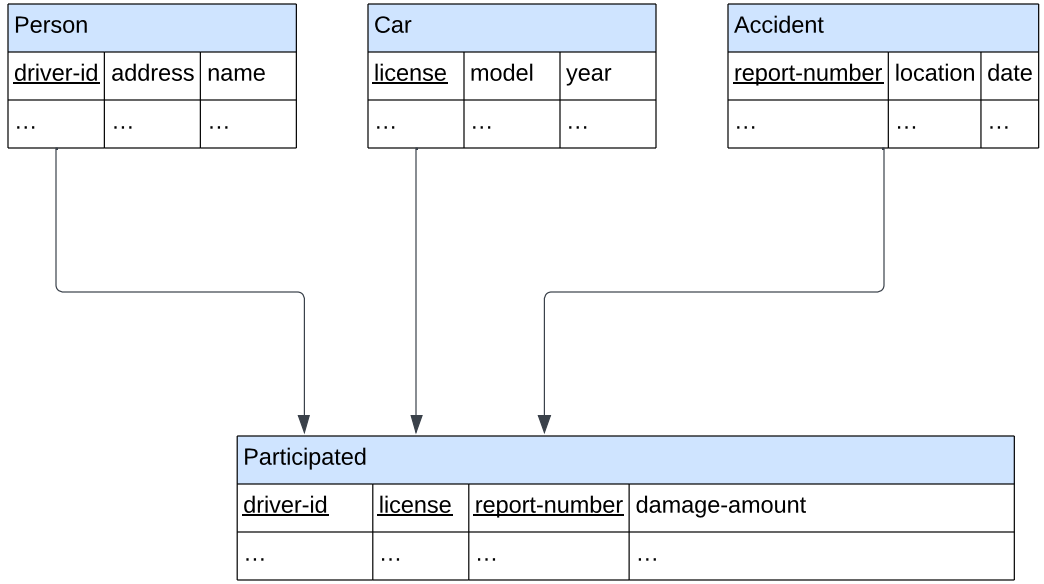
   An atomic domain refers to the idea that values in a domain are indivisible. An atomic domain ensures that each attribute contains only a single, indivisible piece of data. In the example “2023_2”, the atomicity of the domain would mean that the value “2023_2” is treated as a single, indivisible unit, even though it is composed of the year and the semester.

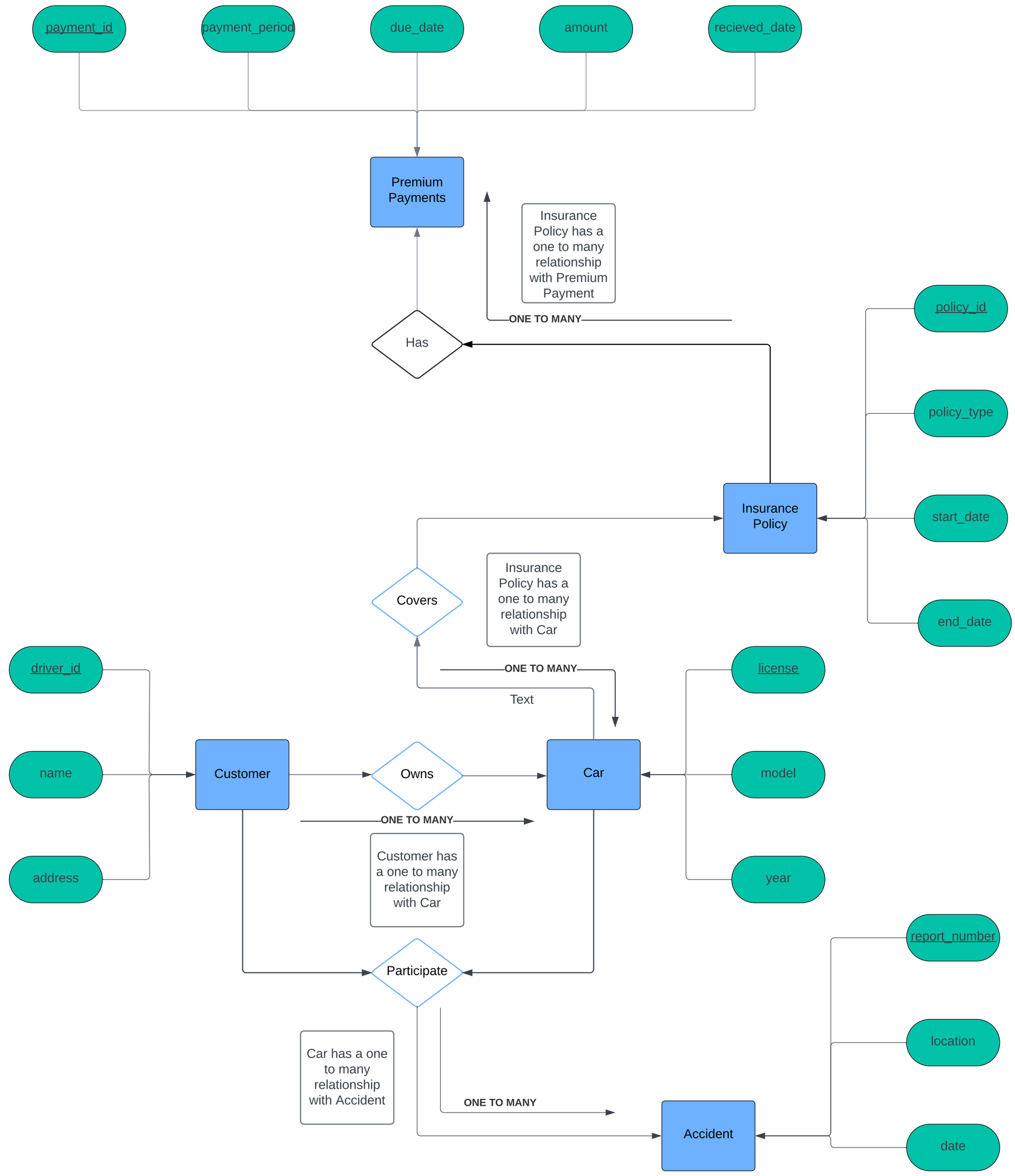
   The domain is different from the type by how the domain refers to the specific set of values an attribute can take, whereas the type defines the kind of data an attribute can store (i.e. an integer, char, varchar). A domain is often more restrictive than a type, as it imposes additional constraints on what specific values are valid within the broader data type.

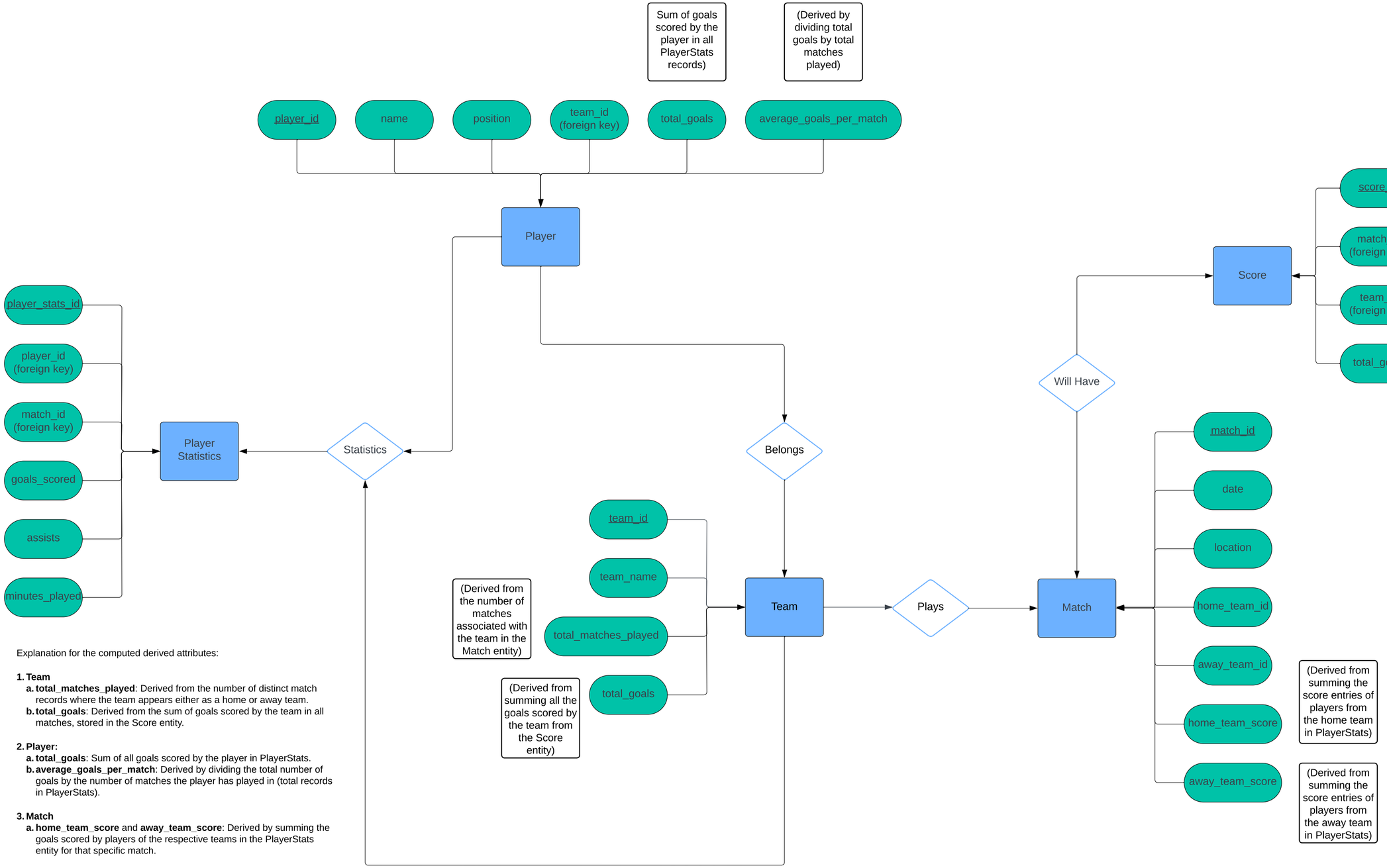
5. Briefly explain the difference between a schema and instance.

   A schema refers to the logical structure of the database, defining how data is organized, describing the data. It is the blueprint of the database, outlining the tables, relationships, and constraints. The schema is defined at the design stage and remains relatively unchanged over time. It specifies the name of the relation, plus the name and type of each column. There are two types of schema, a physical schema which is the database design at the physical level, and a logical scheme which is the database design at the logical level.

   An instance refers to the actual data in the database at any given point in time. It is a table with rows and columns where the number of rows is known as the cardinality and the number of fields is known as the degree or arity. A field is a variable of any type that is declared directly in a class or struct. Fields are members of their containing type. While the schema remains static, the instance can change frequently as new data is added, updated, or deleted.


Person
driver-id
address
name
…
…
…
Car
license
model
year
…
…
…
Accident
report-number
location
date
…
…
…
Participated
driver-id
license
report-number
damage-amount
…
…
…
…


payment_id
payment_period
due_date
amount
recieved_date
Premium Payments
Insurance Policy has a one to many relationship with Premium Payment
ONE TO MANY
Has
policy_id
policy_type
start_date
end_date
Insurance Policy
Covers
Insurance Policy has a one to many relationship with Car
ONE TO MANY
Text
driver_id
name
address
Customer
Owns
Car
license
model
year
ONE TO MANY
Customer has a one to many relationship with Car
Participate
Car has a one to many relationship with Accident
ONE TO MANY
Accident
report_number
location
date



Explanation for the computed derived attributes:

1. **Team**
   a. **total_matches_played**: Derived from the number of distinct match records where the team appears either as a home or away team.
   b. **total_goals**: Derived from the sum of goals scored by the team in all matches, stored in the Score entity.

2. **Player:**
   a. **total_goals**: Sum of all goals scored by the player in PlayerStats.
   b. **average_goals_per_match**: Derived by dividing the total number of goals by the number of matches the player has played in (total records in PlayerStats).

3. **Match**
   a. **home_team_score** and **away_team_score**: Derived by summing the goals scored by players of the respective teams in the PlayerStats entity for that specific match.

CISC 5325: Databases

Homework 1: Environment Setup and Test

Submission Instructions

Complete all the tests in this notebook and submit only this notebook as a PDF to Blackboard. To convert the jupyter notebook into a pdf you can use either of the following methods:

- File --> Print Preview --> Print --> Save to PDF
- File --> Download As HTML --> Print --> Save to PDF

It is recommended that you put the screenshots into the same folder as this notebook so you do not have to alter the path to include your images.

Please read all the instructions thoroughly!

Add Student Information

1. Replace my name with your full name.
2. Replace my UNI with your UNI.

In [6]:
```
# Print your name, fdin, and track below

name = "Jan C. Bierowiec"
uni = "A18424173"
track = "Cool Track"

print(name)
print(uni)
```

```
Jan C. Bierowiec
A18424173
```

Testing Anaconda and Python

Run the following cells to ensure that you have the correct version of Python and all necessary packages installed.

Python Version

The test below should return the path to the Python interpreter for your Anaconda environment. The exact path may be differ from Mac to Windows, or based on installation choices you made. As long as the path has "anacondaq3" in it, you should be OK.

The code cells below have the results of my execution. You must execute on your computer and show your results.

Your results will be similar but different in details because you are on a different computer.

In [8]:
```
#
# Run this cell to print your current working directory.
%pwd
```

Out[8]:
```
'/Users/janbierowiec/Downloads/assignment1_coding_settled fall 2024'
```

In [10]:
```
#
# Display your node information.
import socket
hostname = socket.gethostname()
print("Your values")
print("Host name = ", hostname)
IPAddr = socket.gethostbyname(hostname)
print("IPAddr = ", IPAddr)
```

```
Your values
Host name =  Jasios-MBP-2
IPAddr =  192.168.1.156
```

In [12]:
```
import sys
```

In [14]:
```
ex = sys.executable
ex
```

Out[14]:
```
'/opt/anaconda3/bin/python'
```

In [16]:
```
# Checking that anacoda3 is in the path.
#
if 'anaconda3' in ex:
```

```
    print("Test seems OK.")
else:
    print("Not cool.")
```

```
Test seems OK.
```

The following tests that you have a sufficiently up to date version of Python.

In [18]:
```
print("Python version information:", sys.version_info, "\n")
if sys.version_info.major != 3 or \
    ((sys.version_info.major == 3) and (sys.version_info.minor < 8)):
    print("You have an invalid version of Python.")
else:
    print("Your Python version is OK.")
```

```
Python version information: sys.version_info(major=3, minor=12, micro=4, rel
easelevel='final', serial=0)

Your Python version is OK.
```

If the test fails, you have to install Anaconda properly.

Install ipython-sql

The actual message below will vary based on what you do/do not already have installed. You are fine as long as there is not a major error.

In [20]:
```
%pip install ipython-sql
```

```
Collecting ipython-sql
  Downloading ipython_sql-0.5.0-py3-none-any.whl.metadata (17 kB)
Collecting prettytable (from ipython-sql)
  Downloading prettytable-3.11.0-py3-none-any.whl.metadata (30 kB)
Requirement already satisfied: ipython in /opt/anaconda3/lib/python3.12/site
-packages (from ipython-sql) (8.25.0)
Requirement already satisfied: sqlalchemy>=2.0 in /opt/anaconda3/lib/python3
.12/site-packages (from ipython-sql) (2.0.30)
Collecting sqlparse (from ipython-sql)
  Downloading sqlparse-0.5.1-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: six in /opt/anaconda3/lib/python3.12/site-pac
kages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils in /opt/anaconda3/lib/python
3.12/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/anaconda3/li
b/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /opt/anaconda3/lib/python
3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (3.0.1)
Requirement already satisfied: decorator in /opt/anaconda3/lib/python3.12/si
```

```
te-packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /opt/anaconda3/lib/python3.12/s
ite-packages (from ipython->ipython-sql) (0.18.1)
Requirement already satisfied: matplotlib-inline in /opt/anaconda3/lib/pytho
n3.12/site-packages (from ipython->ipython-sql) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in /opt/anacond
a3/lib/python3.12/site-packages (from ipython->ipython-sql) (3.0.43)
Requirement already satisfied: pygments>=2.4.0 in /opt/anaconda3/lib/python3
.12/site-packages (from ipython->ipython-sql) (2.15.1)
Requirement already satisfied: stack-data in /opt/anaconda3/lib/python3.12/s
ite-packages (from ipython->ipython-sql) (0.2.0)
Requirement already satisfied: traitlets>=5.13.0 in /opt/anaconda3/lib/pytho
n3.12/site-packages (from ipython->ipython-sql) (5.14.3)
Requirement already satisfied: pexpect>4.3 in /opt/anaconda3/lib/python3.12/
site-packages (from ipython->ipython-sql) (4.8.0)
Requirement already satisfied: wcwidth in /opt/anaconda3/lib/python3.12/site
-packages (from prettytable->ipython-sql) (0.2.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/anaconda3/lib/pyt
hon3.12/site-packages (from jedi>=0.16->ipython->ipython-sql) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /opt/anaconda3/lib/python3
.12/site-packages (from pexpect>4.3->ipython->ipython-sql) (0.7.0)
Requirement already satisfied: executing in /opt/anaconda3/lib/python3.12/si
te-packages (from stack-data->ipython->ipython-sql) (0.8.3)
Requirement already satisfied: asttokens in /opt/anaconda3/lib/python3.12/si
te-packages (from stack-data->ipython->ipython-sql) (2.0.5)
Requirement already satisfied: pure-eval in /opt/anaconda3/lib/python3.12/si
te-packages (from stack-data->ipython->ipython-sql) (0.2.2)
Downloading ipython_sql-0.5.0-py3-none-any.whl (20 kB)
Downloading prettytable-3.11.0-py3-none-any.whl (28 kB)
Downloading sqlparse-0.5.1-py3-none-any.whl (44 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.2/44.2 kB 1.4 MB/s eta 0:00:0
0
Installing collected packages: sqlparse, prettytable, ipython-sql
Successfully installed ipython-sql-0.5.0 prettytable-3.11.0 sqlparse-0.5.1
Note: you may need to restart the kernel to use updated packages.
```

- If you got errors, please follow the instructions in the ipython-sql site to install the magic.
- **NOTE:** Running the cell above may produce multiple notifications about installing requirements or requirement already satisfied. That is normal.
- Once you get the install to work without errors, run the following cell.

In [22]:
```
%load_ext sql
```

- If you did not get an error response, your test passed.

- If you run the cell twice, your answer should be:

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

SQLAlchemy/PyMySQL

Install `sqlalchemy` and `pymysql.` These are Python language packages for interacting with SQL and MySQL databases. Your actual response message may be different. Your environment is OK if you do not get a major error.

In [24]:
```
%pip install sqlalchemy
%pip install pymysql
```

```
Requirement already satisfied: sqlalchemy in /opt/anaconda3/lib/python3.12/s
ite-packages (2.0.30)
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/anaconda3/li
b/python3.12/site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /opt/anaconda3/lib/python
3.12/site-packages (from sqlalchemy) (3.0.1)
Note: you may need to restart the kernel to use updated packages.
Collecting pymysql
  Downloading PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
Downloading PyMySQL-1.1.1-py3-none-any.whl (44 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.0/45.0 kB 549.4 kB/s eta 0:00
:000:00:01
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
Note: you may need to restart the kernel to use updated packages.
```

MySQL Connectivity

You installed MySQL Community Edition. You have to choose a userID and password during the installation.

Please set the values in the cell below.

In [26]:
```
#
# Replace root with the user ID for MySQL and dbuserdbuser with the passwor
#
%sql mysql+pymysql://root:dbuserdbuser@localhost
```

In [28]:
```
#
```

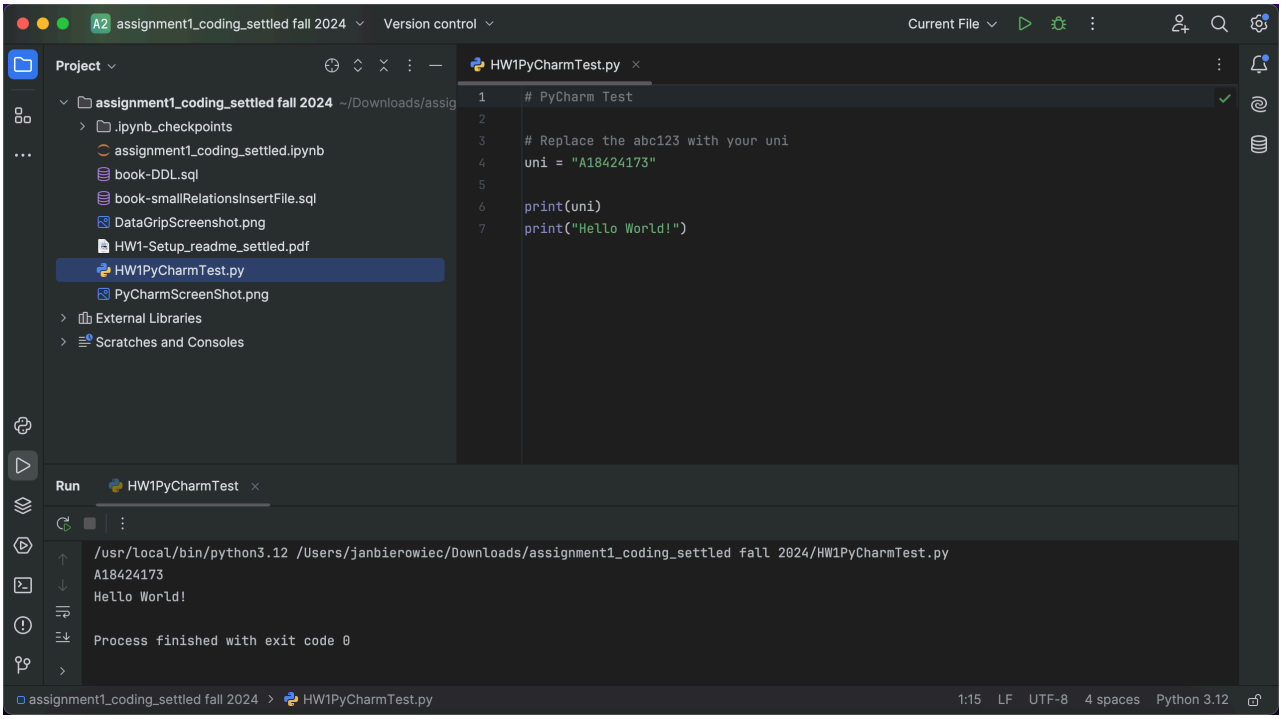
PyCharm

Required for Programming Track only, but recommended for all. Follow the instructions to setup PyCharm and launch. Take a screenshot and insert it into the notebook using the cell below. You may have to change the path to the name and/or location of your image.

In [30]:
```
from IPython.display import Image

Image("/Users/janbierowiec/Desktop/PyCharmScreenshot.png")
```

Out[30]:

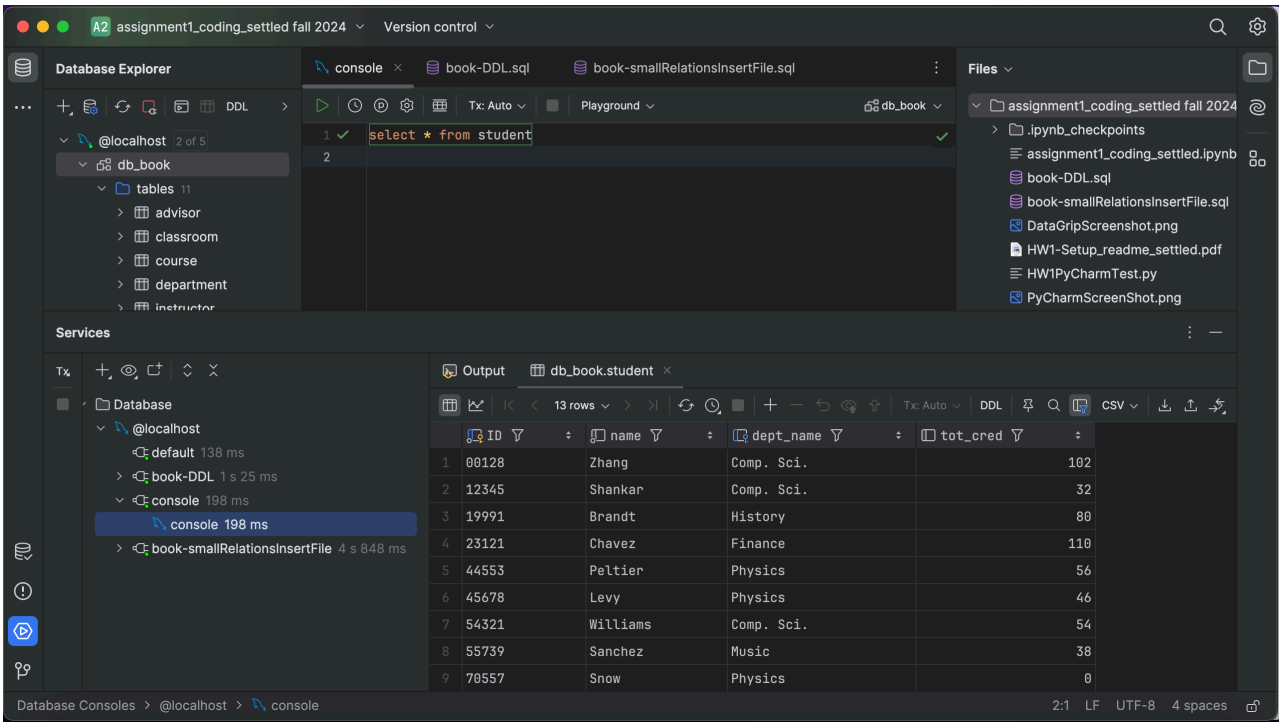


DataGrip

Follow the instructions in the homework definition to setup DataGrip and connect DataGrip to MySQL. Insert your screenshot of the successful query on the sample database below. You may have to change the path to the name and/or location of your image.

In [32]:
```
Image("/Users/janbierowiec/Desktop/DataGripScreenshot.png")
```

Out[32]:



Sample Database

The recitation showed how to install the first database/dataset we will use in the course.

Please follow the recitation and load the data, then run the query below.

In [36]:
```
%sql select * from db_book.student
```

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

Out[36]:

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

In [ ]:

CISC 5325: Databases

Homework 1: Environment Setup and Test

Submission Instructions

Complete all the tests in this notebook and submit only this notebook as a PDF to Blackboard. To convert the jupyter notebook into a pdf you can use either of the following methods:

- File --> Print Preview --> Print --> Save to PDF
- File --> Download As HTML --> Print --> Save to PDF

It is recommended that you put the screenshots into the same folder as this notebook so you do not have to alter the path to include your images.

Please read all the instructions thoroughly!

Add Student Information

1. Replace my name with your full name.
2. Replace my UNI with your UNI.

In [3]:
```
# Print your name, fdin, and track below

name = "Jan C. Bierowiec"
uni = "A10424173"
track = "Cool Track"

print(name)
print(uni)
```

```
Jan C. Bierowiec
A10424173
```

Testing Anaconda and Python

Run the following cells to ensure that you have the correct version of Python and all necessary packages installed.

Python Version

The test below should return the path to the Python interpreter for your Anaconda environment. The exact path may be differ from Mac to Windows, or based on installation choices you made. As long as the path has "anaconda3" in it, you should be OK.

The code cells below have the results of my execution. You must execute on your computer and show your results.

Your results will be similar but different in details because you are on a different computer.

In [5]:
```
#
# Run this cell to print your current working directory.
%pwd
```

Out[5]:
```
'/Users/janbierowiec/Desktop/assignment1_coding_settled fall 2024'
```

In [7]:
```
#
# Display your node information.
import socket
hostname = socket.gethostname()
print("Your values:")
print("Host name = ", hostname)
IPAddr = socket.gethostbyname(hostname)
print("IPAddr = ", IPAddr)
```

```
Your values
Host name =  Jasios-MacBook-Pro-2.local
IPAddr =  127.0.0.1
```

In [9]:
```
import sys
```

In [11]:
```
ex = sys.executable
ex
```

Out[11]:
```
'/opt/anaconda3/bin/python'
```

In [13]:
```
# Checking that anaconda3 is in the path.
#
if 'anaconda3' in ex:
    print("Test seems OK.")
else:
    print("Not cool.")
```

```
Test seems OK.
```

The following tests that you have a sufficiently up to date version of Python.

In [15]:
```
print("Python version information:", sys.version_info, "\n")
if sys.version_info.major != 3 or \
    ((sys.version_info.major == 3) and (sys.version_info.minor < 8)):
    print("You have an invalid version of Python.")
else:
    print("Your Python version is OK.")
```

```
Python version information: sys.version_info(major=3, minor=12, micro=4, releaselevel='final', serial=0)

Your Python version is OK.
```

If the test fails, you have to install Anaconda properly.

Install ipython-sql

The actual message below will vary based on what you do/do not already have installed. You are fine as long as there is not a major error.

In [17]:
```
%pip install ipython-sql
```

```
Requirement already satisfied: ipython-sql in /opt/anaconda3/lib/python3.12/site-packages (0.5.0)
Requirement already satisfied: prettytable in /opt/anaconda3/lib/python3.12/site-packages (from ipython-sql) (3.11.0)
Requirement already satisfied: ipython in /opt/anaconda3/lib/python3.12/site-packages (from ipython-sql) (8.25.0)
Requirement already satisfied: sqlalchemy>=2.0 in /opt/anaconda3/lib/python3.12/site-packages (from ipython-sql) (2.0.30)
Requirement already satisfied: sqlparse in /opt/anaconda3/lib/python3.12/site-packages (from ipython-sql) (0.5.1)
Requirement already satisfied: six in /opt/anaconda3/lib/python3.12/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils in /opt/anaconda3/lib/python3.12/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/anaconda3/lib/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /opt/anaconda3/lib/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (3.0.1)
Requirement already satisfied: decorator in /opt/anaconda3/lib/python3.12/site-packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /opt/anaconda3/lib/python3.12/site-packages (from ipython->ipython-sql) (0.18.1)
Requirement already satisfied: matplotlib-inline in /opt/anaconda3/lib/python3.12/site-packages (from ipython->ipython-sql) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in /opt/anaconda3/lib/python3.12/site-packages (from ipython->ipython-sql) (3.0.43)
Requirement already satisfied: pygments>=2.4.0 in /opt/anaconda3/lib/python3.12/site-packages (from ipython->ipython-sql) (2.15.1)
Requirement already satisfied: stack-data in /opt/anaconda3/lib/python3.12/site-packages (from ipython->ipython-sql) (0.2.0)
Requirement already satisfied: traitlets>=5.13.0 in /opt/anaconda3/lib/python3.12/site-packages (from ipython->ipython-sql) (5.14.3)
Requirement already satisfied: pexpect>4.3 in /opt/anaconda3/lib/python3.12/site-packages (from ipython->ipython-sql) (4.8.0)
Requirement already satisfied: wcwidth in /opt/anaconda3/lib/python3.12/site-packages (from prettytable->ipython-sql) (0.2.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/anaconda3/lib/python3.12/site-packages (from jedi>=0.16->ipython->ipython-sql) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /opt/anaconda3/lib/python3.12/site-packages (from pexpect>4.3->ipython->ipython-sql) (0.7.0)
Requirement already satisfied: executing in /opt/anaconda3/lib/python3.12/site-packages (from stack-data->ipython->ipython-sql) (0.8.3)
Requirement already satisfied: asttokens in /opt/anaconda3/lib/python3.12/site-packages (from stack-data->ipython->ipython-sql) (2.0.5)
Requirement already satisfied: pure-eval in /opt/anaconda3/lib/python3.12/site-packages (from stack-data->ipython->ipython-sql) (0.2.2)
Note: you may need to restart the kernel to use updated packages.
```

- If you get errors, please follow the instructions in the ipython-sql site to install the magic.
- **NOTE:** Running the cell above may produce multiple notifications about installing requirements or requirement already satisfied. That is normal.
- Once you get the install to work without errors, run the following cell.

In [18]:
```
%load_ext sql
```

- If you did not get an error response, your test passed.
- If you run the cell twice, your answer should be:

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

SQLAlchemy/PyMySQL

Install `sqlalchemy` and `pymysql`. These are Python language packages for interacting with SQL and MySQL databases. Your actual response message may be different. Your environment is OK if you do not get a major error.

In [21]:
```
%pip install sqlalchemy
%pip install pymysql
```

```
Requirement already satisfied: sqlalchemy in /opt/anaconda3/lib/python3.12/site-packages (2.0.30)
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/anaconda3/lib/python3.12/site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /opt/anaconda3/lib/python3.12/site-packages (from sqlalchemy) (3.0.1)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: pymysql in /opt/anaconda3/lib/python3.12/site-packages (1.1.1)
Note: you may need to restart the kernel to use updated packages.
```

MySQL Connectivity

You installed MySQL Community Edition. You have to choose a userID and password during the installation.

Please set the values in the cell below.

In [23]:
```
#
# Replace root with the user ID for MySQL and dbuserdbuser with the password you set.
#
%sql mysql+pymysql://root:dbuserdbuser@localhost
```

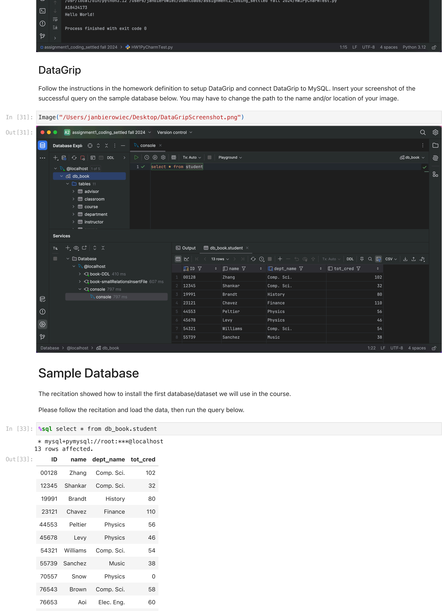
In [25]:
```
#
```

PyCharm

Required for Programming Track only, but recommended for all. Follow the instructions to setup PyCharm and launch. Take a screenshot and insert it into the notebook using the cell below. You may have to change the path to the name and/or location of your image.

In [29]:
```
from IPython.display import Image

Image("/Users/janbierowiec/Desktop/PyCharmScreenshot.png")
```

Out[29]:


DataGrip

Follow the instructions in the homework definition to setup DataGrip and connect DataGrip to MySQL. Insert your screenshot of the successful query on the sample database below. You may have to change the path to the name and/or location of your image.

In [31]:
```
Image("/Users/janbierowiec/Desktop/DataGripScreenshot.png")
```

Out[31]:

Sample Database

The recitation showed how to install the first database/dataset we will use in the course.

Please follow the recitation and load the data, then run the query below.

In [33]:
```
%sql select * from db_book.student
```

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

Out[33]:

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

In [ ]:
```

```