Assignment 8.2 - PySports: Setup

For this module’s assignment you will be learning how to create database tables, run SQL scripts from the terminal window, and how to connect a Python program to MySQL. There is a lot of information to absorb in this assignment. Make sure you take your time to understand what you are doing and why you are doing it.

Connecting VS Code to MySQL

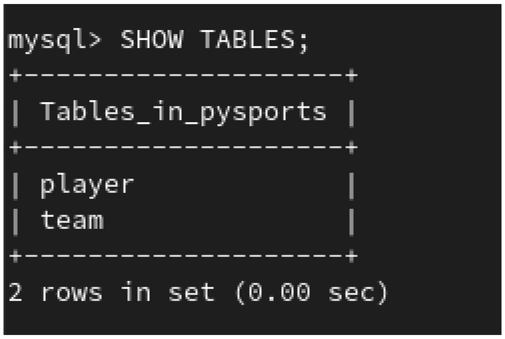
- In the left-hand menu pane expand “MySQL” and click on the + sign.
    - If you do not see MySQL, press Ctrl + Shift + P and enter “MySQL.”
    - There should be an option for “Explorer: Focus on MySQL View”
- For host enter: **localhost**
- For user enter: **root**
- For password enter the root password you selected for MySQL
- Press “Enter” through the remaining options
- If successful, you should see a new entry named **localhost**
- Expanding localhost should show the pysports database

Note: if you receive a connection error you will need to run this command in the MySQL terminal (make sure MySQL is running before you execute this command).

- **ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'YourRootPassword';**

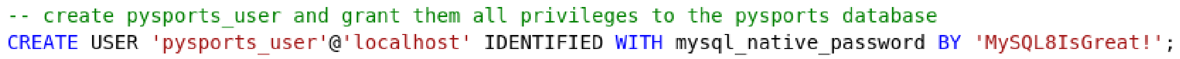
MySQL: CLI Scripts

- Start MySQL
- Activate the desired database
    - use <database name>;
    - Example: use pysports;
- Run the SQL script
    - source <path_to_the_sql_script>.sql
    - Example: **source /home/rkrasso/csd/csd-310/module_7/db_init.sql**
    - Note: the db_init.sql script is located in the courses GitHub repository under the module_7 directory.
- Show a list of database tables (this assumes you have activated the database
    - Activate database: use <database name>;
        - **USE pysports;**
    - Show tables
        - **SHOW TABLES;**



```
mysql> SHOW TABLES;
+--------------------+
| Tables_in_pysports |
+--------------------+
| player             |
| team               |
+--------------------+
2 rows in set (0.00 sec)
```

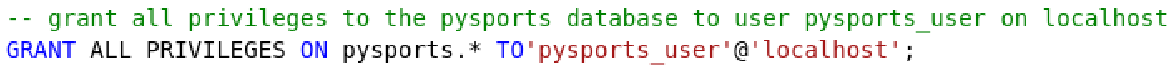
MySQL: Create User
CREATE USER <user_name>@<host> IDENTIFIED WITH mysql_native_password BY <password>;
Example



```
-- create pysports_user and grant them all privileges to the pysports database
CREATE USER 'pysports_user'@'localhost' IDENTIFIED WITH mysql_native_password BY 'MySQL8IsGreat!';
```

MySQL: Grant User Privileges
GRANT ALL PRIVILEGES ON <database_name>.* TO <database_user>@<database_host>

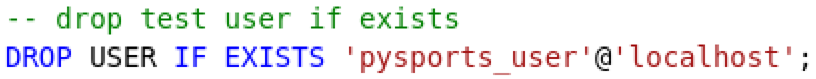
Example



```
-- grant all privileges to the pysports database to user pysports_user on localhost
GRANT ALL PRIVILEGES ON pysports.* TO'pysports_user'@'localhost';
```

MySQL: Drop User
DROP USER IF EXISTS <user_name>@<database_host>

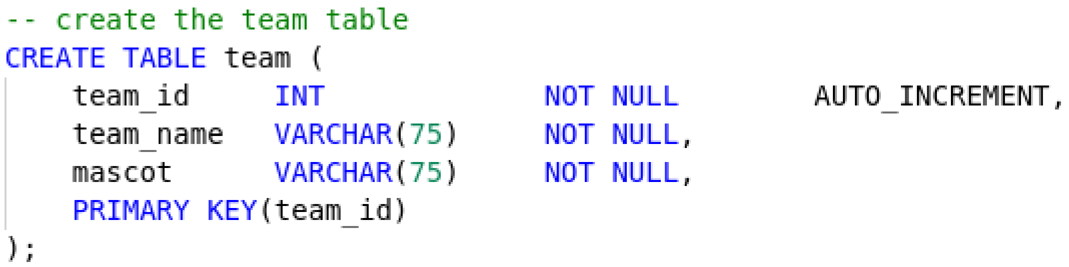
Example



```
-- drop test user if exists
DROP USER IF EXISTS 'pysports_user'@'localhost';
```

MySQL: Create Table
CREATE TABLE <table_name> (<field_values>);

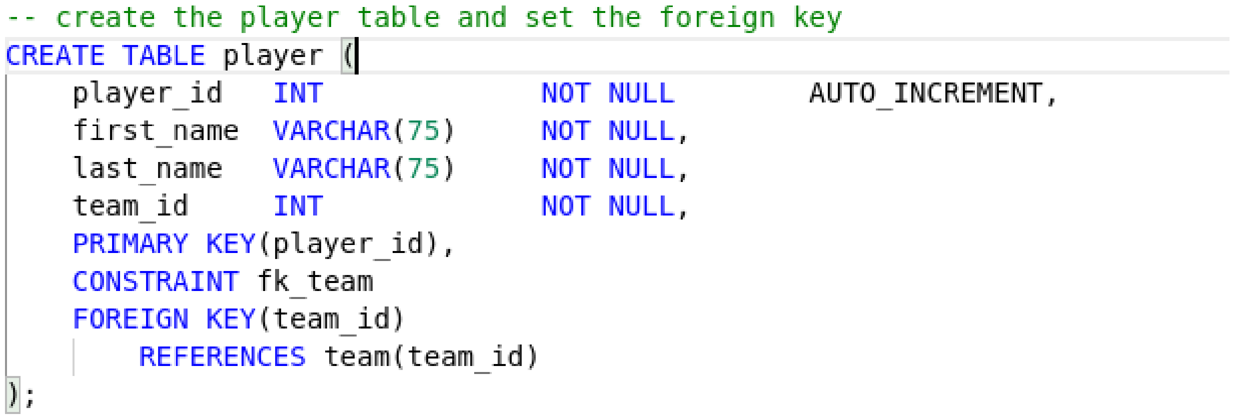
Example:



```
-- create the team table
CREATE TABLE team (
    team_id     INT             NOT NULL        AUTO_INCREMENT,
    team_name   VARCHAR(75)     NOT NULL,
    mascot      VARCHAR(75)     NOT NULL,
    PRIMARY KEY(team_id)
);
```

In this example, **team** is the name of the table and **team_id, team_name, and mascot** are the column values. **INT** is the data type for team_id and **VARCHAR(75)** are the data types for team_name and mascot.

VARCHAR(75) represents the number of allowable characters in the string value. **NOT NULL** tells MySQL to not allow the insertion of null values. **AUTO_INCREMENT** tells MySQL to increment the team_id by 1 each time a new record is inserted. **PRIMARY KEY(team_id)** tells MySQL that team_id is the table's primary key.
MySQL: Foreign Key Constraints

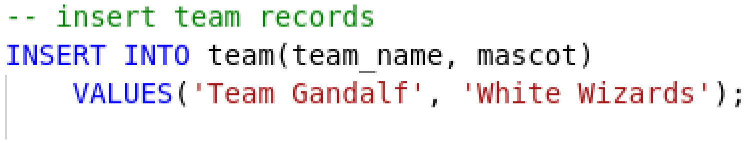


```
-- create the player table and set the foreign key
CREATE TABLE player (
    player_id   INT             NOT NULL        AUTO_INCREMENT,
    first_name  VARCHAR(75)     NOT NULL,
    last_name   VARCHAR(75)     NOT NULL,
    team_id     INT             NOT NULL,
    PRIMARY KEY(player_id),
    CONSTRAINT fk_team
    FOREIGN KEY(team_id)
        REFERENCES team(team_id)
);
```

Foreign keys are how we represent one-to-many relationships. They are in fact, the mechanism for connecting related data sets. In this example, we add a new **CONSTRAINT** named **fk_team**, which contains a **FOREIGN KEY** of **team_id**. **REFERENCES** is used to tell MySQL that this foreign key connects to the **team** table by **teamp_id**.

MySQL: Insert
INSERT INTO <table_name>(<fields_to_insert>) VALUES (<values_to_insert>);

Example

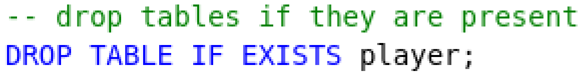


```
-- insert team records
INSERT INTO team(team_name, mascot)
    VALUES('Team Gandalf', 'White Wizards');
```

In this example, we are inserting a new record in the **team** table and mapping the values **Team Gandalf** and **White Wizards** to the fields **team_name** and **mascot**. The order you place the items in the **VALUES** section must match the order you have for the **COLUMN VALUES**.

MySQL: Drop Table
DROP TABLE IF EXISTS <table_name>;

Example

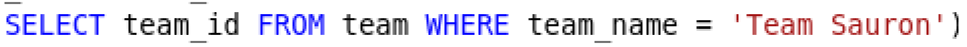


```
-- drop tables if they are present
DROP TABLE IF EXISTS player;
```

In this example, we are telling MySQL to delete the table **player** if it already exists in the database. If the table **player** does not exist, no action is invoked.

MySQL: Select Statements
SELECT <column_names> FROM <table_name> WHERE <column_name> = <value_name>;

Example



```
SELECT team_id FROM team WHERE team_name = 'Team Sauron')
```

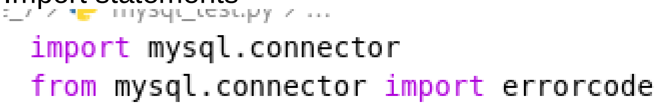
In this example, we are telling MySQL to return the **team_id** for the record where **team_name** equals **Team Sauron**.

MySQL: Python Driver (pip)

- pip install mysql-connector-python

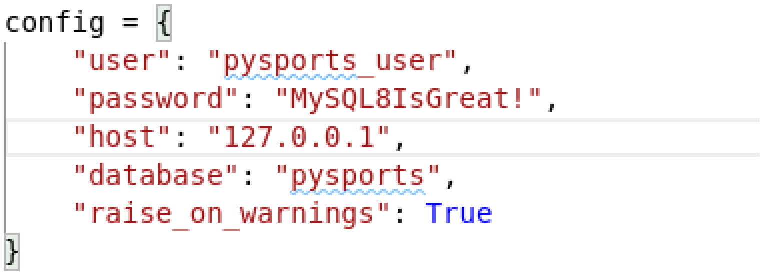
MySQL: Python Connector Test

Import statements



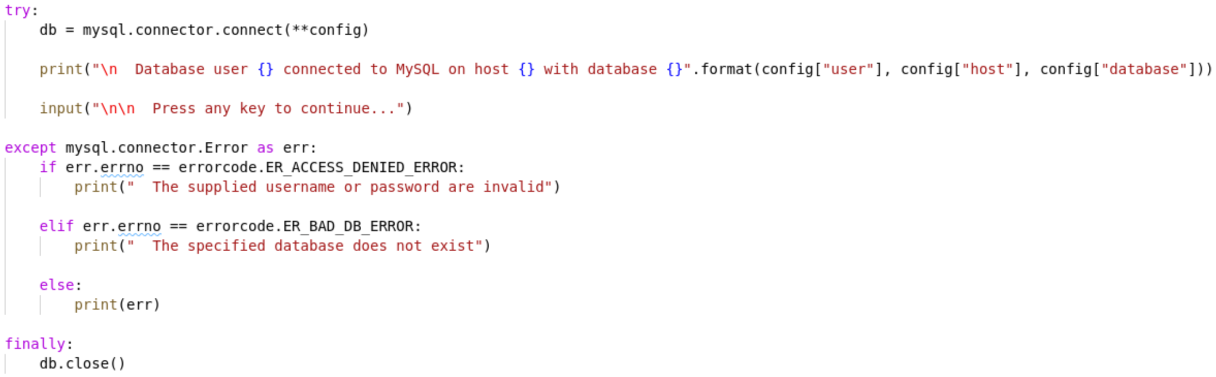
```
import mysql.connector
from mysql.connector import errorcode
```

Database config object



```
config = {
    "user": "pysports_user",
    "password": "MySQL8IsGreat!",
    "host": "127.0.0.1",
    "database": "pysports",
    "raise_on_warnings": True
}
```

Connection test code



```
try:
    db = mysql.connector.connect(**config)

    print("\n  Database user {} connected to MySQL on host {} with database {}".format(config["user"], config["host"], config["database"]))

    input("\n\n  Press any key to continue...")

except mysql.connector.Error as err:
    if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("  The supplied username or password are invalid")

    elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("  The specified database does not exist")

    else:
        print(err)

finally:
    db.close()
```

Checklist

- Completed the MySQL Installation Guide
- Created a new database named **pysports**
- Reviewed the guide on how to activate a database
- VS Code installed with the recommended extensions.
- Connecting VS Code to MySQL
- MySQL: Python Driver (pip)
- MySQL: Python Connector Test
- MySQL: CLI Scripts
- MySQL: Create User
- MySQL: Grant User Privileges
- MySQL: Drop User

- MySQL: Create Table
- MySQL: Foreign Key Constraints
- MySQL: Drop Table
- MySQL: Select Statements

Instructions

Before proceeding any further, make sure you have completed the checklist.

- Start MySQL and activate the **pysports** database.
- Run the db_init.sql file
  - Take a screenshot of the output and of the show tables command.
- Create a new directory under csd-310 and name it **module_8**
- Under the module_7 directory create a new file and name it **mysql_test.py**
- Using the coding example I provided, create a new Python program that connects to the pytest database.
  - Take a screenshot of the output and make sure it matches what I have provided.
- Combine all images in a single word document and include your name, date, and assignment number.

GitHub

- Stage, commit, and push your work to GitHub.

Deliverables

- Link to your GitHub repository.
- <your-last-name>-<assignment-name>.docx
- mysql_test.py