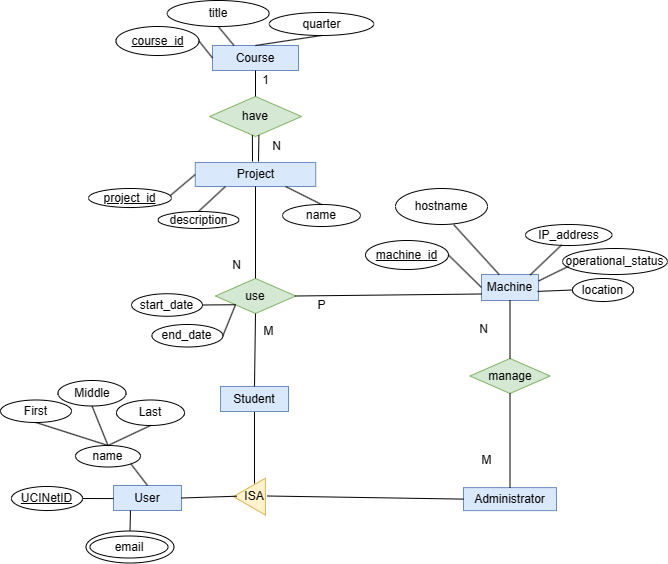
Project - Implement a Python program with MySQL

**Introduction**

In this project, you will implement a command-line program to manage the computing servers in ICS (a simplified version of HW2). You are required to use Python mysql connector to access and manipulate the database. The Python program should 1) accept command-line arguments as the inputs, 2) parse the inputs into SQL statements, 3) execute the statements in the MySQL server, 4) handle and print the results.

The ER diagram is shown as follows, refer to HW2 solution for the DDLs.

Changelog:

1. 02-17: Added Python and package requirements in the regulation section point 9
2. 02-20: Added test dataset link

**Dataset**

There will be one CSV file for each table. Each line represents one record, and the columns are separated by a comma ‘,’’. The order of columns follows the order of attributes in the DDL.

The [provided dataset](https://drive.google.com/file/d/1t_Y9daoi7riAxUB8PX2tnePLNyhlImEA/view?usp=sharing) is for testing only so you can make any changes to cover more cases. A hidden dataset will be used during the grading.

**Regulations and Assumptions**

1. You can have any number of Python files, but the entry file must be named “project.py”
2. The command to run the program will be

python3 project.py <function name> [param1] [param2] …

The list of function names and their parameters is in the function requirements section.

1. You can assume that the command-line input is always correct IN FORMAT ONLY. There won’t be a non-existing function name as input, and the parameters will be given in the correct order and format. So you don’t need to handle unexpected input. However, input content can be faulty - e.g. given a duplicate netID for insertion.
2. You can assume that the dataset files are always correct IN FORMAT AND CONTENT. So there won’t be errors when parsing the file, or when inserting the records to DB.
3. Every date has the format YYYY-MM-DD, e.g. 2024-02-29
4. If the input is NULL, treat it as NULL type, not a string called “NULL”
5. If the output is boolean, print “Success” or “Fail”
6. If the output is a result table, print each record in one line and separate columns with ‘,’ - just like the format of the dataset file.
7. You must use Python 3. The standard Python libraries and mysql-connector-python will be installed in the autograder — other third-party packages are not allowed.

**Function Requirements**

1. **Import data**

Delete existing tables, and create new tables. Then read the csv files in the given folder and import data into database. You can assume that the folder always contains all the necessary CSV files and the files are correct.

Input:

python3 project.py import [folderName:str]

python3 project.py import test\_data

Output:

Table - (Number of users,Number of machine, Number of Course)

1. **Insert student**

Insert a new student into the related tables.

Input:

python3 project.py insertStudent [UCINetID:str] [email:str] [First:str] [Middle:str] [Last:str]

python3 project.py insertStudent testID test@uci.edu Alice NULL Wong

Output:

Bool

1. **Add email**

Add email to a user

Input:

python3 project.py addEmail[UCINetID:str] [email:str]

python3 project.py addEmail testID test@gmail.com

Output:

Bool

1. **Delete student**

Delete the student in both the User and Student table.

Input:

python3 project.py deleteStudent [UCINetID:str]

python3 project.py deleteStudent testID

Output:

Bool

1. **Insert machine**

Insert a new machine.

Input:

python3 project.py insertMachine [MachineID:int] [hostname:str] [IPAddr:str] [status:str] [location:str]

python3 project.py insertMachine 102 test.com 192.168.10.5 Active “DBH 1011”

Output:

Bool

1. **Insert use record**

Insert a new use record.

Input:

python3 project.py insertUse [ProjId:int] [UCINetID:str] [MachineID:int] [start:date] [end:date]

python3 project.py insertUse 2005 testID 102 2023-01-09 2023-03-10

Output:

Bool

1. **Update course**

Update the title of a course

Input:

python3 project.py updateCourse [CourseId:int] [title:str]

python3 project.py updateCourse 1287 “Intro to db managment”

Output:

Bool

1. **Course attended**

Given a student ID, list all unique courses the student attended. Ordered by courseId ascending.

Input:

python3 project.py listCourse [UCINetID:int]

python3 project.py listCourse testID

Output:

Table - CourseId,title,quarter

1. **Popular course**

List the top N course that has the most students attended. Ordered by studentCount, courseID descending.

Input:

python3 project.py popularCourse [N: int]

python3 project.py popularCourse 10

Output:

Table - CourseId,title,quarter,studentCount

1. **Emails of administrators**

Given a machine ID, find all administrators of that machine. List the emails of those administrators. Ordered by netid ascending.

List of emails should be concatenated with ‘;’ e.g. test@uci.edu;test@gmail.com

Input:

python3 project.py adminEmails [machineId: int]

python3 project.py adminEmails 102

Output:

Table - UCINETId,first name,middle name,last name,list of email

1. **Active Students**

Given a machine Id, find all active students that used it more than N times in a specific time range. Ordered by netid ascending. N will be at least 1.

Input:

python3 project.py activeStudent [machineId: int] [N:int] [start:date] [end:date]

python3 project.py activeStudent 102 1 2023-01-09 2023-03-10

Output:

Table - UCINETId,first name,middle name,last name

1. **Number of machine usage**

Given a course id, count the number of usage of each machine in that course. Each unique record in the MachineUse table counts as one usage. Machines that are not used in the course should **have a count of 0** instead of NULL. Ordered by machineId descending.

Input:

python3 project.py machineUsage [courseId: int]

python3 project.py machineUsage 102

Output:

Table - machineID,hostname,ipAddr,count

**Submission**

Submit a zip file (one submission each group) to gradescope. The zip file should contain all the python source file, including the entry point **project.py**.

In gradescope autograder, you need to connect the MySQL server with:

mysql.connector.connect(user='test', password='password', database=’cs122a’)

Make sure to change your code before submission if you are using other accounts in your local database.