CSE 5720-60
Summer 2024
Project 2 - Database Programming

In this project, you will learn how to write a program that can interact with a database. The project is due by 11:59PM, June 25 (Tuesday).

Deliverables

1. The source code of your program that can interact with MySQL.
   - This source code should contain (preferably in one single source code file):
     - The script that creates/names the table in the DB.
     - The script that imports the rows for each table as listed below.
     - The script that runs all 10 queries requested below.
2. The output of your program (text file and screenshot).
   - Both a text file AND screenshot should be included in deliverables.

STEP 1 – Choose the programming language and install the library

In this project, we recommend you use Python to write your program. Before starting, you need to download the corresponding MySQL library. We provide sample templates for Python below.

You should use Python 3.7.0 to work with MySQL. Install mysql connector here: https://dev.mysql.com/downloads/connector/python/. A tutorial on how to install MySQL connector can be found on Canvas. Please watch carefully and follow the instructions step-by-step.

We also provide template codes for mysql interaction in Python:

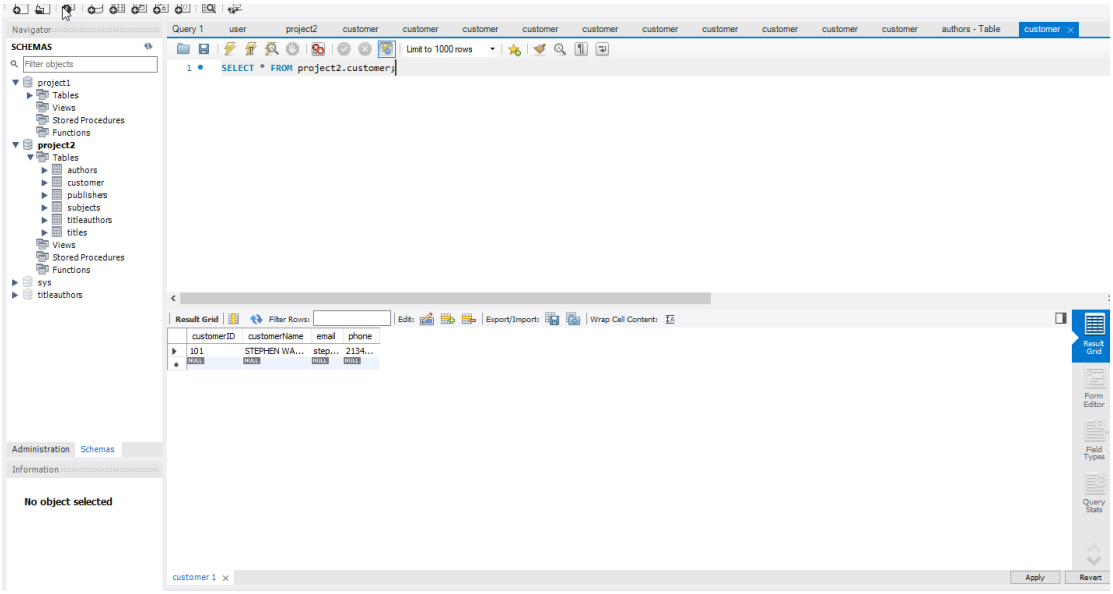
> Here is the example code, in the example codes, the program establishes the connection to the database.

```
import mysql.connector
from mysql.connector import Error

try:
    connection = mysql.connector.connect(host='localhost',
                                         database='Electronics',
                                         user='pynative',
                                         password='pynative@#29')
    if connection.is_connected():
        db_Info = connection.get_server_info()
        print("Connected to MySQL Server version ", db_Info)
        cursor = connection.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You're connected to database: ", record)

except Error as e:
    print("Error while connecting to MySQL", e)
finally:
    if (connection.is_connected()):
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

STEP 2 – Write the program

Use your localhost in MySQL as the server.

Open your MySQL. In the schemas area, right click to create a schema name it anything you prefer (here we use ‘project2’), and then double click on “project2” to make it a default DB to be used.



Remember the database is called “project2” or anything you named it. Use

username(root) and password you created to login. Modify the database name, user name and password in the code template.

NOTE: Everything below this note should be done exclusively using Python, the Python mySQL connector module, and a code editor/IDE of your choice. Any submission that suggests that mySQL was used to create tables or import their rows will be penalized, so please make sure you follow these instructions.

There are five tables in the database "project2":

publishers(pubID, pname, email, phone)
subjects(subID,sName)
authors(auID, aName, email, phone)
titles(titleID, title, pubID, subID, pubDate,cover,price)
titleauthors(titleID, auID, importance)

Create tables 'publishers', 'subjects', 'authors', 'titles' and 'titleauthors.' See below for attribute names and data types.

```
CREATE TABLE `project2`.`publishers` (
  `pubID` INT(3) NOT NULL,
  `pname` VARCHAR(30) NULL,
  `email` VARCHAR(50) NULL,
  `phone` VARCHAR(30) NULL,
  PRIMARY KEY (`pubID`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE);
CREATE TABLE `project2`.`subjects` (
 `subID` VARCHAR(5) NOT NULL,
 `sName` VARCHAR(30) NULL,
 PRIMARY KEY (`subID`));
CREATE TABLE `project2`.`authors` (
  `auID` INT(5) NOT NULL,
  `aName` VARCHAR(30) NULL,
  `email` VARCHAR(50) NULL,
  `phone` VARCHAR(30) NULL,
  PRIMARY KEY (`auID`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE);
```

```
CREATE TABLE `project2`.`titles` (
  `titleID` INT(5) NOT NULL,
  `title` VARCHAR(30) NULL,
  `pubID` INT(3) NULL,
  `subID` VARCHAR(5) NULL,
  `pubDate` DATE NULL,
  `cover` VARCHAR(10) NULL,
  `price` INT(4) NULL,
  PRIMARY KEY (`titleID`),
  INDEX `pubid_idx` (`pubID` ASC) VISIBLE,
  INDEX `subid_idx` (`subID` ASC) VISIBLE,
  CONSTRAINT `pubid`
    FOREIGN KEY (`pubID`)
    REFERENCES `project2`.`publishers` (`pubID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `subid`
    FOREIGN KEY (`subID`)
    REFERENCES `project2`.`subjects` (`subID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
CREATE TABLE `project2`.`titleauthors` (
  `titleID` INT(5) NOT NULL,
  `auID` INT(5) NOT NULL,
  `importance` INT(2) NULL,
  PRIMARY KEY (`titleID`, `auID`),
  INDEX `auID_idx` (`auID` ASC) VISIBLE,
  CONSTRAINT `titleid`
    FOREIGN KEY (`titleID`)
    REFERENCES `project2`.`titles` (`titleID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `auID`
    FOREIGN KEY (`auID`)
    REFERENCES `project2`.`authors` (`auID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

Add the following data into your tables.

```
INSERT INTO SUBJECTS VALUES ('ORA','ORACLE DATABASE');
INSERT INTO SUBJECTS VALUES ('JAVA','JAVA LANGUAGE');
INSERT INTO SUBJECTS VALUES ('JEE','JAVA ENTEPRISE EDITION');
INSERT INTO SUBJECTS VALUES ('VB','VISUAL BASIC.NET');
INSERT INTO SUBJECTS VALUES ('ASP','ASP.NET');

INSERT INTO PUBLISHERS VALUES (1,'WILLEY','WDT@VSNL.NET','9112326087');
INSERT INTO PUBLISHERS VALUES (2,'WROX','INFO@WROX.COM',NULL);
INSERT INTO PUBLISHERS VALUES (3,'TATA MCGRAW-
HILL','FEEDBACK@TATAMCGRAWHILL.COM','9133333322');
```

```
INSERT INTO PUBLISHERS VALUES
(4,'TECHMEDIA','BOOKS@TECHMEDIA.COM','9133257660');

INSERT INTO AUTHORS VALUES (101, 'HERBERT SCHILD','HERBERT@YAHOO.COM',
'2137823450');
INSERT INTO AUTHORS VALUES (102, 'JAMES GOODWILL','GOODWILL@HOTMAIL.COM',
'9095871243');
INSERT INTO AUTHORS VALUES (103, 'DAVAID HUNTER','HUNTER@HOTMAIL.COM',
'9094235581');
INSERT INTO AUTHORS VALUES (104, 'STEPHEN WALTHER','WALTHER@GMAIL.COM',
'2138773902');
INSERT INTO AUTHORS VALUES (105, 'KEVIN LONEY','LONEY@ORACLE.COM',
'9493423410');
INSERT INTO AUTHORS VALUES (106, 'ED. ROMANS', 'ROMANS@THESERVERSIDE.COM',
'9495012201');

INSERT INTO TITLES VALUES (1001,'ASP.NET UNLEASHED',4,'ASP','2002-04-02','HARD
COVER',540);
INSERT INTO TITLES VALUES (1002,'ORACLE10G COMP. REF.',3,'ORA','2005-05-01','PAPER
BACK',575);
INSERT INTO TITLES VALUES (1003,'MASTERING EJB',1,'JEE','2005-02-03','PAPER
BACK',475);
INSERT INTO TITLES VALUES (1004,'JAVA COMP. REF',3,'JAVA','2005-04-03','PAPER
BACK',499);
INSERT INTO TITLES VALUES (1005,'PRO. VB.NET',2,'VB','2005-06-15',HARD COVER,450);
INSERT INTO TITLES VALUES (1006,'INTRO. VB.NET',2,'VB','2002-12-02','PAPER BACK',425);


INSERT INTO TITLEAUTHORS VALUES (1001,104,1);
INSERT INTO TITLEAUTHORS VALUES (1002,105,1);
INSERT INTO TITLEAUTHORS VALUES (1003,106,1);
INSERT INTO TITLEAUTHORS VALUES (1004,103,1);
INSERT INTO TITLEAUTHORS VALUES (1005,103,1);
INSERT INTO TITLEAUTHORS VALUES (1005,102,2);
```

Write a program in python to run the following functions in order:

1. In table “Titles”, there is already some data: the ID for each title, their names, etc. Your program should print out all the data in this table.

Example Outputs:

Output from Titles table:

(1001, 'ASP.NET UNLEASHED', 4, 'ASP', datetime.date(2002, 4, 2), 'HARD COVER', 513)

(1002, 'ORACLE10G COMP. REF.', 3, 'ORA', datetime.date(2005, 5, 1), 'PAPER BACK', 661)

…..

….

2. Create a table customer (<u>custID</u>, custName, zip, city, state).

3. Insert 5 customers (‘ABRAHAM SILBERSCHATZ’, ' HENRY KORTH ', 'CALVIN HARRIS', 'MARTIN GARRIX' and ' JAMES GOODWILL'.) into table “customer” with the custID, custName, zip, city and state. If you want to execute your program multiple times and don’t want to see errors of trying to insert duplicate entries, you may use “INSERT IGNORE INTO” statement, which will do nothing if there is already the same entry in the table.

4. Find the publisher who has published the most titles. If 2 or more publishers are tied for the most number of titles, then your program should print all of them.

5. List all the authors and the total price of their published titles, in order of greatest to least total price. If an author has no published titles, they do not need to be listed.

6. Find the names of all titles which have more than 1 author who wrote it.

7. Find the names of all publishers who have published a book with a price below $500, with a cover type of "Paper Back".

8. Write a query to retrieve the names of all authors who have written books whose subjects contain the word "JAVA", but have not written any books on the subject "VISUAL BASIC.NET".

9. Write a query to retrieve the names of all publishers whose email addresses contains the domain ".com".

10. Form a query to decrease the price of all books published before 2003 by 5% and increase the price of all books published after 2004 by 15%.

Save your outputs to a text file using the name "output.txt", a screenshot of your output, and then archive with your source code (Use the name project2.py) to the file "project2-xxxxxxxxx.zip", xxxxxxxxx being your student id, and turn it in on Canvas. **Note: Not submitting the file under the correct folder may cause a deduction in your credit**.