Rajesh Ayyappanpillai

[Module 12.2 Assignment](https://cyberactive.bellevue.edu/webapps/assignment/uploadAssignment?content_id=_15490810_1&course_id=_526136_1&group_id=&mode=view)

Assignment: Milestone #5

5/15/2024

Recommendations: From Prof.

Assignment: Module 10.1 Assignment

**Assignment: Milestone #2**

The data looks good! Hard to tell if you have everything you need without having access to a similar database and running some queries.

One potential problem I see is with the Orders table; you generally want to track the Order separate from the Products. The way your schema is set up, it would be one item per order.

I deducted 8 points for the deliverables not meeting the assignment instructions. The CREATE/INSERT scripts were to be done via a SQL script in MySQL, and the SELECT statements for each table were to be done in the Python script.

You have them opposite, SQL in Python and SELECT in MySQL.

In order to adhere to the suggestions the code aligns with the assignment guidelines, I have modify the CREATE and INSERT statements to a MySQL script, while managing the SELECT statements within the Python script.

This script will handle the creation of the tables and the insertion of sample data.

-- setup\_bacchus\_db.sql

CREATE TABLE Employees (

    employee\_id INT AUTO\_INCREMENT PRIMARY KEY,

    name VARCHAR(255),

    department VARCHAR(255),

    title VARCHAR(255)

);

CREATE TABLE Suppliers (

    supplier\_id INT AUTO\_INCREMENT PRIMARY KEY,

    name VARCHAR(255),

    product\_type VARCHAR(255),

    delivery\_frequency VARCHAR(255)

);

CREATE TABLE Products (

    product\_id INT AUTO\_INCREMENT PRIMARY KEY,

    name VARCHAR(255),

    type VARCHAR(255)

);

CREATE TABLE Orders (

    order\_id INT AUTO\_INCREMENT PRIMARY KEY,

    product\_id INT,

    supplier\_id INT,

    quantity INT,

    order\_date DATE,

    FOREIGN KEY (product\_id) REFERENCES Products(product\_id),

    FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id)

);

CREATE TABLE Shipments (

    shipment\_id INT AUTO\_INCREMENT PRIMARY KEY,

    supplier\_id INT,

    expected\_delivery DATE,

    actual\_delivery DATE,

    FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id)

);

CREATE TABLE Distributors (

    distributor\_id INT AUTO\_INCREMENT PRIMARY KEY,

    name VARCHAR(255),

    product\_id INT,

    FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

CREATE TABLE EmployeeHours (

    employee\_id INT,

    quarter INT,

    hours\_worked INT,

    FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id)

);

-- Insert data

INSERT INTO Employees (name, department, title) VALUES

('Jane Vu', 'Finance', 'Financial Analyst'),

('Margaret Murphy', 'Marketing', 'Marketing Head'),

('Krish Bob', 'Marketing', 'Assistant'),

('David Doyle', 'Production', 'Production Manager'),

('John Sexton', 'Distribution', 'Distribution Manager'),

('Charles Watson', 'Supply', 'Supply Manager');

INSERT INTO Suppliers (name, product\_type, delivery\_frequency) VALUES

('Supplier A', 'Blue and Red', 'Monthly'),

('Supplier B', 'Sky and Moon', 'Monthly'),

('Supplier C', 'Yellow and Boxes', 'Monthly');

INSERT INTO Products (name, type) VALUES

('Pinot Noir', 'Red Wine'),

('Syrah', 'Red Wine'),

('Riesling', 'White Wine'),

('Chardonnay', 'White Wine');

INSERT INTO Distributors (name, product\_id) VALUES

('Distributor 1', 1),

('Distributor 2', 2),

('Distributor 3', 3),

('Distributor 4', 4);

This script will handle connecting to the database and performing SELECT queries.

import mysql.connector

from mysql.connector import errorcode

def fetch\_data(query):

    try:

        db = mysql.connector.connect(

            user="root",

            password="popcorn",

            host="127.0.0.1",

            database="bacchus",

            raise\_on\_warnings=True

        )

        cursor = db.cursor()

        cursor.execute(query)

        results = cursor.fetchall()

        for row in results:

            print(row)

        cursor.close()

        db.close()

    except mysql.connector.Error as err:

        if err.errno == errorcode.ER\_ACCESS\_DENIED\_ERROR:

            print("Error: Access denied. Please check your username and password.")

        elif err.errno == errorcode.ER\_BAD\_DB\_ERROR:

            print("Error: Database does not exist.")

        else:

            print(err)

# SELECT queries

queries = {

    "Employees": "SELECT \* FROM Employees",

    "Suppliers": "SELECT \* FROM Suppliers",

    "Products": "SELECT \* FROM Products",

    "Orders": "SELECT \* FROM Orders",

    "Shipments": "SELECT \* FROM Shipments",

    "Distributors": "SELECT \* FROM Distributors",

    "EmployeeHours": "SELECT \* FROM EmployeeHours"

}

# Fetch and print data from each table

for table, query in queries.items():

    print(f"\nData from {table}:")

    fetch\_data(query)