

תרגיל: בניית מערכת ניהול עובדים עם MySQL

מטרת התרגיל

בתרגיל זה תבנו מערכת מלאה לניהול עובדים מבוססת נתונים MySQL-OOP-טור שימוש ב-operations.

דרישות מוקדמות

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

:הגדרות חיבור מבוססת נתונים

- Host: localhost
- User: root
- Password: (סיסמה שלכם)
- Database: company_db

חלק 1: יצירת המחלקה הבסיסית (10 נקודות)

:עם EmployeeDB צרו מחלקה בשם

1.1 מетод __init__

:מזהה שמקבלת את פרמטרי החיבור מבוססת נתונים

- host
- user
- password
- database

:השמרו את הפרמטרים כattribute self.connection = None

:רמז

```
def __init__(self, host, user, password, database):  
    self.host = ...  
    self.user = ...  
    ... המשיכו #
```

חלק 2: התחברות מבוססת נתונים (15 נקודות)

2.1 מетод connect()

:כתבו מזהה שמתחברת מבוססת נתונים

- לטיפול בשגיאות try-except-השתמשו בmysql.connector.connect()
- צרו חיבור עם אם החיבור הצליח, הדפסו "Connected to MySQL"
- אם יש שגיאה, הדפסו אותה

:דוגמת פלט מצופה

✓ Connected to MySQL

רמז:

- לבודיקה (self.connection.is_connected())
- אם MySQL תיפסו את Error mysql.connector.Error

חלק 3: ייצירת טבלה (15 נקודות)

3.1 create_table()

עם employees כתבו מתודה שיזירת טבלה:

- id - מفتח ראשי - AUTO_INCREMENT
- name - VARCHAR(100), NOT NULL
- department - VARCHAR(50)
- salary - DECIMAL(10, 2)

CREATE TABLE IF NOT EXISTS employees (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, department VARCHAR(50), salary DECIMAL(10, 2))

✓ Table ready

דוגמת פלט:

חלק 4: הוספת עובד (20 נקודות)

4.1 add_employee()

שימושיפה עובד חדש add_employee(name, department, salary) עם %s:

- prepared statements
- לשמירה השינויים (commit()) בוצעו
- של העובד החדש ID-החזירו את ה
- ID-הדיויסו הודעת הצלחה עם השם והוא

דוגמת פלט:

✓ Added David Chen (ID: 1)

רמז:

- cursor.lastrowid מוחזר את ה
- cursor.execute(sql, (name, department, salary))

חלק 5: שילוף מידע (20 נקודות)

5.1 get_all_employees()

כתבו מתודה שמחזירה את כל העובדים:

- dictionaries רשימה של
- טפלו בשגיאות

5.2 get_employee_by_id()

כתבו מתודה שמחזירה עובד ספציפי:

- dictionary אחד החזירו
- אם לא נמצא None החזירו

רמז:

```
cursor = self.connection.cursor(dictionary=True) # מוחזר dictionaries!
```

חלק 6: עדכון ומחיקה (15 נקודות)

6.1 update_salary() מתודה

שמעדכנת שכר (update_salary(emp_id, new_salary)) כתבו מתודה:

- UPDATE statement
- בוצעו commit
- הדפיסו הודעת הצלחה

6.2 delete_employee() מתודה

שמוחקת עובד (delete_employee(emp_id)) כתבו מתודה:

- DELETE statement
- בוצעו commit
- הדפיסו הודעת הצלחה

דוגמת פלט:

- ✓ Updated salary for employee 1
- ✓ Deleted employee 2

חלק 7: חיפוש לפי מחלקה (10 נקודות)

7.1 get_department_employees() מתודה

שמחזירה את כל העובדים במחלקה (get_department_employees(department)) כתבו מתודה מסויימת:

- WHERE clause
- החזירו רשימת dictionaries

חלק 8: סגירת חיבור (5 נקודות)

8.1 close() מתודה

כתבו מתודה שסגורת את החיבור למסד הנתונים:

- בדקו אם החיבור מחובר לפני הסגירה
- הדפיסו הודעת הצלחה

בדיקות התוכנית

לאחר שיימתם, בדקו את הקוד שלכם:

```
'צירת מופע #  
db = EmployeeDB('localhost', 'root', 'your_password', 'company_db')  
  
התחברות #  
db.connect()  
  
'צירת טבלה #  
db.create_table()
```

```

הוסף עובדים #
db.add_employee("David Cohen", "Development", 15000)
db.add_employee("Sarah Levi", "Marketing", 12000)
db.add_employee("Michael Ben-David", "Development", 16000)

שליפת כל העובדים #
employees = db.get_all_employees()
for emp in employees:
    print(emp)

עדכן שכר #
db.update_salary(1, 17000)

חישוב לפי מחלוקת #
dev_employees = db.get_department_employees("Development")
print(f"\nDevelopment employees: {len(dev_employees)}")

סגירת חיבור #
db.close()

```

שאלות בonus (20 נקודות נוספת)

- B1.** שמחזירה את השכר הממוצע לכל () `get_average_salary_by_department()` מתחודה מחלוקת
- B2.** שמחזירה עובדים עם שכר מעל הסכום שהוזן `get_high_earners(min_salary)` מתחודה
- B3.** שמאפשרת לעדכן כל שדה (args) `update_employee(emp_id, **kwargs)` מתחודה (name, department, salary) או
- B4.** שאפשר להשתמש ב(`__enter__` `__exit__`) למחלוקת context manager מתחודה `with statement`

טיפים כלליים

- 防范 SQL injection** -prepared statements (עם %s)
- 事务** (commit()) אחורי INSERT/UPDATE/DELETE
- 异常处理** (try-except)
- 游标** (cursors) -או השתמשו ב (context manager)
- בדקו את הקוד שלכם עם נtones שונים

הערכתה

מחלקה	נקודות
מחלקה בסיסית	10
התחברות	15

מחלקה	נקודות
יצירת טבלה	15
הוספת עובד	20
שליפת מידע	20
עדכן ומחיקה	15
חיפוש לפי מחלוקת	10
סגירת חיבור	5
סה"כ	110
בונוס	20

! בהצלחה 