**משימה – MySQL**

**מטרה**

במשימה זו נתרגל כתיבת אפליקציה המבוססת JDBD API  
על מנת שנוכל לייבא בקלות יחסית את MySQL Driver נעבוד עם פרויקט מסוג Maven ולא Java

**דרישות**

* התקנה של MySQL & Workbench
* הכרות עם פרויקט Maven
* הכרות עם חנות mvenrepository.com

**שלב א – יצירת אפליקציית חדשה**

1. פתחו פרויקט Maven חדש
2. בדקו את גרסת MySQL המתוקנת במחשב
3. חפשו את ה- Driver המתאים לגרסה הספציפית שלכם עבור MySQL בסביבת JAVA
4. עדכנו את קובץ pom.xml והוסיפו את תגית <dependencies></dependencies>
5. בתוך התגית שהוספת (ממש באמצע) הוסיפו את ה-dependency עבור MySQL Driver
6. עדכנו את Maven
7. בדקו תחת External Libraries ש- MySQL Driver הותקן כנדרש
8. בדקו תחת user windows תיקיית .m2 שההתקנה הותקנה משרתי Maven Central העולמים למחשב הפרטי שלכם!

**שלב ב – הגדרת שכבת Java Beans**

הגדירו את מחלקת Member באופן הבא:

* Attributes
  + - id: int
  + - name: String
  + - city: String
  + - age: double
* Constructors
  + + Member(id: int, name: String, city: String, age: double)
  + + Member(name: String, city: String, age: double)
* Methods
  + + Getters & Setters
  + + toString(): String

**שלב ג – הגדרת DatabaseManager**

תמכו באפשרויות הבאות:

1. בניית סכמה java822
2. מחיקת סכמה java822
3. יצירת את טבלת gym\_members המוגדרת באופן הבא:

table **gym\_members**

* **id**: **int – PK AI NN**
* **name**: **varchar(30) NN**
* **city**: **varchar(30) NN**
* **phone\_number**: **varchar(11) NN**
* **age**: **double NN**

צרו אסטרטגיה בשם dropAndCreate המוחקת את הסכמה, בונה את הסכמה ובונה את הטבלה

**שלב ג – הגדרת שכבת DAO**

הגדירו את הממשק MemberDAO באופן הבא:

* Methods
  + addMember(member: Member): void
  + updateMember(id: int, member: Member): void
  + deleteMember(id: int): void
  + findAll(): List<Member>
  + findById(id: int): Member
  + isExistById(id: int): boolean
  + findAllByCity(city: string): List<Member>
  + findByAgeOver(age: double): List<Member>
  + findByAgeUnder(age: double): List<Member>
  + findTop1ByAge: Member
  + count(): int
  + avgAge(): double

צרו את המחלקה MemberDAOImpl המממשת את הממשק MemberDAO .

**שלב ד – עבודה נכונה וחיננית ע"י יצירת Infrastructure משלנו**

1. צרו Infrastructure הממחזרת מקסימום קוד באמצעות static methods
2. הוסיפו תמיכה ב-Connection Pool עם מבנה נתונים Thread Safe
3. הקפידו להסתיר מתודות פנימיות עם private access modifier
4. חשפו החוצה מתודות רלוונטיות כ public access modifier

**שלב ה – שכבת הבדיקות**

צרו 10 מתאמנים והכניסו אותם לבסיס הנתונים.

הפעילו את כלל המתודות ובדקו שהן עובדות, הקפידו על הדפסות

**בהצלחה!**