בס"ד

תרגיל כתיבת מסד נתונים

# הקדמה

במסגרת תרגיל זה תבנו מסד נתונים משלכם. בשלב זה של התרגיל לא יהיה ממשקים חיצוניים כך שאת הפעולות השונות תצטרכו לממש בעצמכם.

מסד נותנים מאפשר למשתמש שלו מגוון רחב של אפשרויות, אך לא פחות חשוב מזה, הוא דואג לביצועים גבוהים ומבטיח אמינות המאפשרים למשתמש במסד הנתונים שימוש מבצעי.

# הנחיות:

## שלב ראשון:

בשלב הראשון תאפיינו את מסד הנתונים ש לכם. כלומר, עליכם לתכנן את הארכיטקטורה שלו.  
מסד הנתונים שלכם צריך לתמוך בפעולות הבאות לכל הפחות:

* הגדרת טבלה
* מחיקת טבלה
* הכנסת רשומה לטבלה (ברשומה יכולים להיות מחרוזות, מספרים ותאריכים)
* מחיקת רשומה מטבלה
* מחיקת מספר רשומות לפי תנאי/ים מסוימים מטבלה
* עדכון רשומה בטבלה
* חיפוש רשומות העונות על תנאי/ים מסוימים בטבלה

הציגו את מודל הנתונים של הDB, כמו את האופן שבו אתם שומרים את המידע על הדיסק, מה האובייקטים המרכזיים במערכת שלכם ומה יחסי הגומלין ביניהם.

פרטו את היתרונות והחסרונות של הDB שלכם. מה השיקולים שלכם לבחירה זו.

## שלב שני:

שלב זה הוא שלב העיצוב. עליכם להכין מסמך המתאר את הDesign של הקוד שלכם. כאשר אתם מסיימים את כתיבת המסמך ואתם עומדים מאחורי העיצוב שלכם – הציגו אותו למדריכים/ות.

## שלב שלישי:

הגדרת אובייקט

1. עליכם לאפשר יצירה של אובייקט עם התכונות שלו
2. התכונות יכולות להיות מהסוגים הבאים
3. מחרוזת
4. מספר
5. תאריך
6. בהגדרת האובייקט יש לציין את תכונת המפתח שלו

* אפשר לצאת מנקודת הנחה שהמפתח יהיה פשוט (תכונה אחת)

1. count

## שלב רביעי:

עבודה עם נתונים

1. עליכם לאפשר עבודה עם נתונים באובייקט. להלן הפונקציות שיש לממש:
2. insert\_record - הכנסת רשומה
3. delete\_record - מחיקת רשומה
4. update\_record - עדכון רשומה

-לרשום את שמות הפונקציות

## שלב חמישי- א:

שליפות

1. query\_table - עליכם לאפשר שליפה של האובייקט
2. השליפה צריכה לתמוך בהצגה של תכונות נבחרות ובהתניות
3. אין צורך לממש SQL

## שלב חמישי- ב:

יצירת האינדקס(create\_index)

1. יש ליצור פונקציה באמצעותה תגדירו אינדקס על עמודה ספציפית
2. ניתן להגדיר אינדקס מהסוגים הבאים
3. B-Tree
4. Hash
5. Bitmap
6. לאינדקס יש חשיבות מכרעת בביצועים של שליפה. יש להציג מדדים המראים שיפור בביצועי שליפה עם וללא אינדקס.

## שלב חמישי- ג:

אפשרות לJoin (query\_multiple\_tables)

1. יש ליצור פונקציה המאפשרת לבצע איחוד בין אובייקטים על פי מזהה של האובייקט.
2. יש לתת חשיבות לאינדקסים. אם ישנו אינדקס ספציפי לעמודה עליה נעשית פעולת הJoin הביצועים יהיו טובים יותר.

## שלב שישי:

ייעול עבודה מול קבצים

1. שליפה של אובייקט צריכה לקרוא מהדיסק אל הזיכרון מינימום מידע כדי לא להעמיס על המכונה שמריצה את מסד הנתונים. שמרו מידע על הקבצים שלכם כך שתוכלו לקרוא חלקים מהקובץ בהתאם לאובייקט אותו רוצה המשתמש לקרוא.

## 

## דרישות נוספות

1. יש לחשוף פונקציות מתוך הDB שיצרתם, הפונקציות שאתם מייצר בקובץ db.py שלכם צריכות לרשת מהקובץ db\_api.py שהורדתם מהgit בקישור(<https://github.com/scaleup-suvelocity/database-exercise/>).
2. שימו לב!
   1. כדי שסקריפט הבדיקות יוכל לבדוק את הביצועים שלכם אנא שימרו את כל הקבצים שאתם יוצרים בשביל הdb שלכם תחת תיקיה בשם db\_files.
   2. יש לשמור את הקובץ פייתון שלכם בשם db.py בכדי שהקובץ בדיקות ידע לבדוק אותו
3. כדי לעבוד עם קבצים, תוכלו להיעזר בספריה האהובה pathlib.

## הערכת המסד נתונים שלכם:

ביצוע ההערכה

1. הערכה מתבצעת באמצעות הסקריפט test\_db.py
2. **חשוב**! קובץ הבדיקות יעבוד רק אחרי שמימשתם את הפונקציות השונות בפתרון שלכם.
   1. יש להוריד ולהתקין את הספריה pytest, ניתן לעשות זאת בדרך הנוחה להורדת ספריות פייתון בשבילכם.
   2. לפני הרצת הקוד צריך לפתוח python, ולהריץ את הקוד create\_db\_backup
   3. כעת הריצו את הפקודה pytest test\_db.py מcmd שלכם בכך הרצתם את הקובץ בדיקות על הקוד שכתבתם.
3. הערכה של מסד הנתונים שלכם תעזור לכם באיתור שגיאות והבנת היעלות שבניתם את הDB והאינדקס שלו. ניתן ואף רצוי להוסיף בדיקות משלכן כדי לבדוק מקרים קצה נוספים.

להריץ טסט מסוים:py.test -k test\_nameלהריץ עם מדידת זמן:pytest --durations=8

להריץ את כל הטסטים:py.test