בס"ד

**DB Build**

**מוטיבציה**: בנית DataBase לתמיכה בחיפוש מהיר של טקסטים, עם קבצי מידע גדולים.

**אופן מימוש**:DB רלציוני. מימוש באמצעות קבצי csv בצורת row oriented מכיון שאנו מתמקדים בחיפוש מהיר של רשומה לפי מפתח ולא בניתוח הנתונים בכללותם.

**הפעולות הנתמכות:**

* **הגדרת טבלה:**

כל טבלה בקובץ נפרד. קיים קובץ שמכיל נתונים על קבצי הטבלאות.

* **מחיקת טבלה:**

מחיקת הקובץ.

* **הכנסת רשומה לטבלה (ברשומה יכולים להיות מחרוזות, מספרים ותאריכים)**

על בסיס מקום פנוי או בסוף הקובץ.

* **מחיקת רשומה מטבלה:**

נשמור ערך בוליאני עבור כל רשומה האם היא זמינה או לא. במקביל נשמור עבור כל קובץ מעין free-list. אם מספר הרשומות הבלתי זמינות גדול (כמו 40%), נכווץ את הקובץ.

* **מחיקת מספר רשומות לפי תנאי/ים מסוימים מטבלה:**

כנ"ל

* **עדכון רשומה בטבלה:**

דריסת השורה המקורית - אם אפשרי. אחרת, מחיקת נשורה והוספת השורה החדשה.

* **חיפוש רשומות העונות על תנאי/ים מסוימים בטבלה:**

סריקת הטבלה והחזרת הערכים המתאימים.

**performance:** קובץ csv הוא קובץ בפורמט פשוט, ומאפשר טעינה ועיבוד עם כמות גדולה של נתונים בצורה יעילה במיוחד.

**space utilization:** הפורמט הטבלאי הינו חסכוני מבחינת ניצול השטח שלו, עם צפי להמרת קבצי הcsv לפורמט בינארי שימקסם את אפשרות דחיסת הנתונים.

**complexity:** פעולות הCRUD:

* CREATE: O(1) - יצירת קובץ חדש והכנסת שמות העמודות.
* READ: O(n) בגודל הטבלה - חיפוש החזרת הרשומה המבוקשת.
* UPDATE: O(n) - חיפוש הרשומה, ועדכונה בO(1)
* DELETE: O(n) - חיפוש הרשומה, ומחיקתה בO(1)

**flexibility:** צורת שמירת הטבלאות קבועה ואינה ניתנת לשינוי עבור ערכים מסוגים שונים.

**יתרונות**: חסכון במקום, עיבוד מהיר, כדי לאפשר עבודה יעילה עם כמות גדולה של נתונים. לאחר הוספת אינדקס מהירות החיפוש עשויה להיות גבוהה מאד.

**חסרונות**: חיפוש איטי ללא אינדקס, הגדרת טבלאות שאינה ניתנת לשינוי.