

Implementation DB

נממש DB לא רלציוני כאשר שמירת הנתונים תהיה באמצעות קבצי Bson אשר שומרים טבלה לפי שורות כאובייקט Dict כאשר הסכימה היא משתנה.

הגדרת טבלה:

ניצור קובץ Bson עם שם הטבלה $O(1)$

מחיקת טבלה:

נמחק את קובץ ה Bson השייך לטבלה $O(1)$

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

נוסיף שורה לקובץ Bson של הטבלה $O(1)$

מחיקת רשומה מטבלה:

נקרא את הקובץ של הטבלה, נעבור עליו לחיפוש הרשומה המיועדת למחיקה, נמחק אותה

ונחזיר את הנתונים המעודכנים לקובץ $O(n)$

מחיקת מספר רשומות לפי תנאים מסוימים מטבלה:

נקרא את הקובץ, נעבור על הנתונים ונבדוק אילו רשומות מקיימות את התנאי ונמחק אותן $O(n)$

עדכון רשומה בטבלה:

נקרא את הטבלה, נעבור עליה ונעדכן את הרשומה $O(n)$

חיפוש רשומות העונות על תנאים מסוימים בטבלה:

נקרא את הטבלה מהקובץ ונשוף את המתאימים $O(n)$

יתרונות:

- יכולות big data
- אין נקודת כישלון יחידה
- מספק ביצועים מהירים
- תכנות מונחה עצמים – קלה לשימוש וגמישה מאד
- מטפל בנתונים גדולים שמנהלים את מהירות הנתונים, המגוון, הנפח והמורכבות
- מציע עיצוב סכימה גמיש שניתן לשנות בקלות ללא השבתה או הפרעה בשירות חסרונות
- ניתן לשנות נתונים ללא עלות על הנתונים הקיימים.

חסרונות:

- אין כללי תקינה
- הוא אינו מציע יכולות מסד נתונים מסורתיות, (כמו עקביות) כאשר מספר עסקאות מבוצעות בו זמנית
- לא עובד טוב גם עם נתונים יחסיים

שיקולים:

- שליפת נתונים מDB לא רלציוני מהירה מאד
- קל לשימוש שיתמוך בטרות של מידע
- כמות רבה של אנשים יכולים להשתמש בו זמנית תוך מתן שירות הוגן ומקסימלי
- DB לא רלציוני מאפשר שינוי בקלות בעל גמישות רבה
- שמירה בקובץ Bson מאפשרת טעינה בקלות של מידע
- קובץ Bson כתוב במספרים בינאריים מה שמאפשר דחיסת נתונים ותופש מעט מקום בזיכרון, מה שגורם לניצול בשטח הזיכרון