**Data Base – Design**

בחרנו בdata base רלציוני שנממש ע"י קובץ shelveשהשליפה ממנו מהירה ויש אליו גישה ישירה כמו למילון, קובץ shelve הוא בינארי ולכן יעיל בזיכרון.

* כל קובץ מייצג טבלה.
* שימוש במונחי Row – oriented.
* סכימה קבועה.
* הגדרת טבלה – ניצור קובץ חדש בשם הטבלה, כאשר נגדיר:

Key: מפתח הטבלה.

Value: dictionary המכיל כkeys- – את שמות השדות וכ-values - את ערכי כל שדה.

* מחיקת טבלה – מחיקת הקובץ התואם.
* הכנסת רשומה לטבלה – נוסיף את הרשומה לקובץ שמייצג את הטבלה, כך שכל זוג של מפתח וערך יוכנסו לקובץ המתאים ע"פ שם הטבלה.
* מחיקת רשומה מטבלה – נמחק את הרשומה, המפתח והערך מהקובץ המתאים ע"פ שם הטבלה.
* מחיקת מספר רשומות לפי תנאי/ים מסוימים מטבלה – נחפש בקובץ את הרשומות המקיימות את התנאי/ים ונמחק אותם, מפתח וערך.
* עדכון רשומה בטבלה – ניגש לקובץ המתאים ע"פ שם הטבלה ונשנה את הערך של המפתח הנתון ברשומה לערך העדכני.
* חיפוש רשומות העונות על תנאי/ים מסוימים בטבלה – ניגש לקובץ המתאים ע"פ שם הטבלה ונחפש את המפתחות שעומדים בתנא/ים ונחזיר את ערכיהם.

נתייחס ל-4 היבטים חשובים:

* Flexibility – גמישות: הdata base שלנו הוא רלציוני ולכן יש בו פחות גמישות מאשר db רלציוני, אבל שומר על שלמות המידע ונכונותו.
* Complexity – סיבוכיות: שאילתות פר שורה: O(1)

שאילתות פר עמודה: O(n)

* Performance – ביצועים: מימוש בקובץ shelve מביא לנו גישה מהירה לנתונים כמו לdictionary, כלומר גישה מהירה לביצוע פעולות כמו הכנסה, שליפה, מחיקה עדכון.
* Space utilization – ניצול השטח: קובץ shelve הוא קובץ בינארי ולכן לא תופס הרבה מקום בזיכרון.

אופי הdata שנשמור יכיל מחרוזות, מספר או תאריך בהתאם לנתונים.