

מסדי נתונים מסוג Document

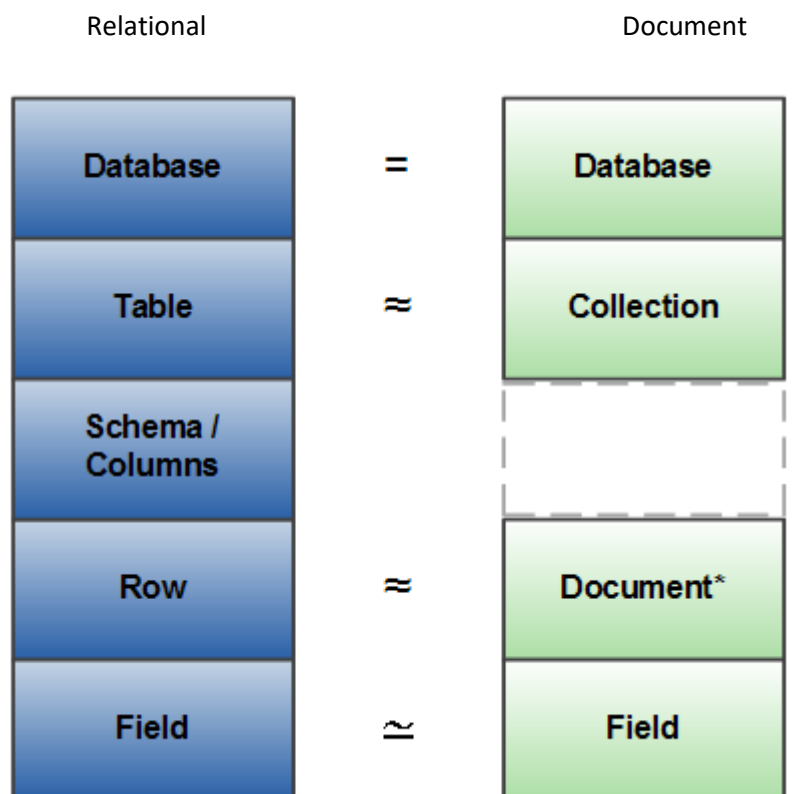
עקרונות

1. ה DB מורכב מאוסף של Collections (ב RDB קוראים להם טבלאות)
2. Collection מורכב מאוסף של מסמכים (ב RDB קוראים להם שורות)
3. כל מסמך הוא אוסף של מאפיינים/שדות שמוגדרים בפורמט ייחודי שמכונה Json (Java Script Object Notation)
4. כל מאפיין מוגדר על ידי שם וערכים אפשריים
5. כל מאפיין יכול להיות בעל ערך אחד , מספר ערכים רצוי או תת מסמך
6. למעשה בניגוד ל DB טבלאי בו ביצענו נירמול והצגנו בנפרד את כל תת הישויות של ישות כאן אנחנו מכניסים את כל ההיררכיה למסמך אחד
7. אין שום הכרח שכל המסמכים יהיו בעלי מבנה זהה

יתרונות

1. גמישות גבוהה בהוספת מאפיינים ותת ישויות
2. אין צורך בפעולות Join כל נתוני הישות נמצאים במסמך אחד

השוואת מושגים לעולם הרלציוני



דוגמה לייצוג של אובייקט המתאר אדם בפורמט Json

```
{
  "firstName": "יעקב",
  "lastName": "ישראלי",
  "address": {
    "streetAddress": "רחוב המשעול, 13",
    "city": "ירושלים",
    "state": "ישראל",
    "postalCode": 10021
  },
  "phoneNumbers": [
    "212 555-1234",
    "646 555-4567"
  ]
}
```

מימושים עיקריים

1. MongoDB
2. Couchbase
3. Amazon DynamoDB
4. Elastic Search

סיכום

- השימוש במשפחה זו של בסיסי נתונים נפוץ בעיקר עבור אפליקציות שדורשות גמישות רבה במבנה הנתונים שלהם וסכמה דינמית בד בבד עם ביצועים קבועים ויכולות תשאול גבוהות
- מערכות לדוגמא: Gaming, real time, Product catalogs, mobile apps, content management
- שמירת הנתונים מתבצעת בפורמט של מסמך : Json או XML
- הגדרת הסכמה היא דינמית, כל מסמך מכיל כמות שונה של שדות והמבנה יכול להתעדכן בהתאם לדרישות האפליקציה. כל מסמך מכיל את כל המידע ששייך לו
- ניתן להגדיר סוגים שונים של אינדקסים לשיפור ביצועי השליפות, למשל (Mongo) compound, geospatial, text
- Document Databases ברוב המקרים תומכים ב ACID ברמת Document