

דוח מסכם – מערכת חישוב נוסחאות דינמיות/ רבקה טהר אפל

מימשי מערכת שמריצה נוסחאות דינמיות על מיליון רשומות בשלוש שיטות שונות (SQL, C#, Python), שומרת תוצאות וזמני ריצה, ומשווה בין השיטות כדי לבחור את היעילה יותר.

• **SQL_SP – חישוב דינמי בצד מסד הנתונים**

השיטה מחשבת את הנוסחה ישירות ב־SQL Server בעזרת SQL דינמי. החישוב מתבצע כולו בתוך ה־DB על מיליון הרשומות, ללא העברת נתונים לאפליקציה, ולכן מתקבלות תוצאות יציבות ומהירות.

יתרון: ביצועים גבוהים וניצול מלא של מנוע ה־SQL.

חסרון: גמישות מוגבלת ושינוי נוסחאות דורש זהירות בעבודה עם SQL דינמי.

• **C_SHARP -NCalc – חישוב בצד האפליקציה**

האפליקציה טוענת את הנתונים מה־DB פעם אחת, מריצה את הנוסחה באמצעות מנוע הביטויים NCalc, ומבצעת את כל החישובים בזיכרון.

השיטה תומכת בפונקציות מתמטיות מורכבות ובעלת יכולת הרחבה גבוהה.

יתרון: גמישות מלאה, קוד נקי, מהירות גבוהה מאוד בזכות חישוב בזיכרון.

חסרון: דורש טיפול נכון בזיכרון ובחיבורי DB.

• **PYTHON – חישוב דינמי בלולאה**

שיטה זו נטענים הנתונים ומתבצע חישוב של כל נוסחה בעזרת eval על כל רשומה בנפרד.

זה מאפשר גמישות גבוהה ופשטות בקוד, אך דורש פירוש מחדש בכל איטרציה.

יתרון: קל להרחבה, גמיש מאוד, מתאים לניסויים.

חסרון: איטי משמעותית, בעיקר בגלל חישוב שורה-אחר-שורה ללא אופטימיזציות פנימיות.

מדידת ביצועים

להלן זמני הריצה שנמדדו עבור כל שיטה על כלל רשומות הנתונים:

- **#C** – זמן ריצה של כ־1–3 שניות
- **SQL_SP** – זמן ריצה של כ־7–8 שניות
- **Python** – זמן ריצה של כ־30–33 שניות

	method	avg_run_time
1	C_SHARP	2.25
2	PYTHON	32.86
3	SQL_SP	7.89

- C# הוא המהיר ביותר בזכות חישוב בזיכרון (In-Memory) ומנוע ביטויים יעיל.
- SQL_SP ביצועי, אך איטי מעט יותר בשל אופטימיזציות SQL פנימיות.
- Python איטי משמעותית בגלל פירוש בכל איטרציה.

בדיקת התאמת תוצאות בין שיטות

כתבתי פרוצדורה שמוודאת שכל שלוש השיטות הניבו את אותה התוצאה עבור כל נוסחה וכל רשומה:

- בודקת פער בין MIN ל-MAX של כל (targil_id, data_id)
- בודקת שכל שלוש השיטות החזירו תוצאה (השלמות)
- מחזירה את מספר הסטיות לכל תרגיל

תוצאה שהתקבלה: **0 סטיות** → כל השיטות הפיקו תוצאות זהות. (הוחזרה טבלה ריקה- 0 סטיות)

targil_id	mismatches

לסיכום:

C# - הפתרון המהיר והמדויק ביותר

- חישוב מהיר מאוד בזיכרון
- מנוע ביטויים יעיל
- גמישות גבוהה להוספת פונקציות ותנאים
- תוצאות זהות ובטוחות כמו SQL

ולכן זו השיטה המתאימה ביותר למערכות שמבצעות חישובים דינמיים רבים, לאורך זמן.

SQL_SP - מצוין כשכל החישובים נשארים בתוך מסד הנתונים. מהיר ויציב, אך פחות גמיש לשינויים בנוסחאות.

Python - גמיש מאוד ומתאים לניסויים ופיתוח מהיר. חישוב טורי של מיליון רשומות איטי יחסית, אך הביצועים יכולים להשתפר משמעותית בעזרת כלים כמו Pandas.