

דוח סיכום פרויקט חישוב נוסחאות דינמיות

1. מטרת הפרויקט

הפרויקט נועד לבנות מערכת שמבצעת חישוב של נוסחאות מתמטיות ודינמיות על מאגר נתונים גדול (1,000,000 רשומות), ומשווה בין שתי שיטות חישוב:

- ## (sp_CalculateFormulas_Fast פרצדורה SQL Server .1 NumPy (python_cal.py)-ו Pandas מ Python .2

2. מבנה המערכת

2.1 מסד הנתונים

- – טבלת הנתונים: מכילה עמודות **d**, **a**, **b**, **c**, **data_t** עם ערכים אקראיים.
 - – טבלת נסחאות: מכילה נוסחאות, תנאים (**tnai**) ותוצאות חלופיות (**false targil**).
 - – טבלת תוצאות: מ אחסנת את התוצאות מכל נוסחה ומהשיטה שבה חישבו את התוצאה.
 - – **results_t**
 - – טבלת לוג: מ אחסנת את זמני הריצה של כל נוסחה לפי שיטה.
 - – **log_t**

Python: python_calc.py קובץ 2.2

- מתחבר למסד הנתונים דרך **.SQLAlchemy + ODBC**
 - שולף את הנתונים (`data_t`) ואת הנוסחאות (`targil_t`)
 - ממיר את הפקודות SQL לפונקציות SQL וכוכו.
 - ממיר תנאים לבתיחה בפייתון: `==` → `=`, `|` → `AND` → `&`, `OR`
 - מחשב את כל הנוסחאות על כל הרשומות, כולל תנאים (`np.where`).
 - מטפל בערכים חריגים (`NaN`, `Inf`) וממיר אותם למספרים תקינים.
 - שומר את התוצאות (`results`) ואת זמני הריצה (`t`)

SQL: sp_CalculateFormulas Fast 2.3 פרוצדורת

- מבצעת את אותו חישוב נוסחאות באמצעות SQL בלבד.
 - משתמשת ב-**CASE WHEN** עבור תנאים.
 - מטפלת בשגיאות מתמטיות: אם חישוב נכשל, מחזירה ס' לכל הרשומות של הנוסחה.
 - רושמת את זמני הritable בלוג_t.(log).

3. אופו העבודה

- .1 יצירה מסד הנתונים וטבלאות **(data_t, targil_t, results_t, log_t)**.
 - .2 מיילוי טבלאות הנתונים והנוסחאות בדעתה אקראי ודוגמאות נסחאות.
 - .3 הרצת הפרויקטורה ב-**SQL Server** ובבדיקה התוצאות **(SQL-Fast)**.
 - .4 הרצת סקריפט **Python** לחישוב עם **Numpy** ו-**Pandas** **(Python_Pandas_Eval_Fixed_Chunks_Safe_v2)**
 - .5 השוואת תוצאות בין **SQL** ל-**Python** לוודא זהות **(ISNULL(sql.result,o) <> ISNULL(py.result,o))**
 - .6 ניתוח זמני הריצה והשוואה ביצועים.

4. התמודדות עם אתגרים

אתגר	פתרון
חישוב על מיליון רשומות עלול לקרוס בזיכרונו	שימוש בחישוב ב- <code>chunks</code> קטנים (500 רשומות) בפייתו
ערכימ חיריגים (<code>NaN</code> , <code>Inf</code> , חלוקה באפס)	שימוש ב- <code>np.clip(np.nan_to_num(x))</code> לפונקציות <code>LN</code> ו- <code>EXP</code>
תנאים לוגיים בנוסחאות	המרה של תנאים מ- <code>SQL</code> ל- <code>&</code> , <code> -></code> ו- <code>OR</code>
שמירה מהירה במאסד הנתונים	שימוש ב- <code>insert</code> ו- <code>fast_executemany</code> ב- <code>chunks</code>

5. תוצאות

- כל הנוסחאות חושבו בהצלחה עבור כל הרשומות.
- בדיקות השוואת בין `SQL` ל-`Python` ע"י פרוצדורה: לא הבדלים בתוצאות.
- זמני הריצה של `Python` מעט ארוכים יותר מאשר חלק מהנוסחאות, אך השיטה בטוחה ומטפלת בכל החיריגות המתמטיות.
- ניתן להרחיב את המערכת לנוסחאות חדשות בקלות על ידי הוספה ל-`targil_t.chunks`.

6. מסקנות

- השיטה `Python` מאפשרת חישוב גמיש של נוסחאות מתמטיות עם טיפול מובנה בשגיאות.
- השיטה `SQL` מהירה יותר ברוב המקרים אך דורשת בתיבה מורכבת יותר של הנוסחאות והטיפול בתנאים.
- הפרויקט מדגים את החשיבות של השוואת תוצאות בין שתי שיטות ומדידות ביצועים.

צילומי מסך:

שליפת רשומות אחרונות

```

SELECT targil_id, COUNT(*) AS RecordsPerTargil
FROM results_t
GROUP BY targil_id;

```

118 %

Results Messages

	results_id	data_id	targil_id	method	result
1	32076546	1536696	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
2	32076545	1536695	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
3	32076544	1000030	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
4	32076543	1000029	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
5	32076542	1000028	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
6	32076541	1000027	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
7	32076540	1000026	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
8	32076539	1000025	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
9	32076538	1000024	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
10	32076537	1000023	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
11	32076536	1000022	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
12	32076535	1000021	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
13	32076534	1000020	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
14	32076533	1000019	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
15	32076532	1000018	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
16	32076531	1000017	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
17	32076530	1000016	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
18	32076529	1000015	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
19	32076528	1000014	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435
20	32076527	1000013	22	Python_Pandas_Eval_Fixed_Chunks_Safe_v2	-8.89482908192435

שליפה התוצאות לפי תרגיל מס' 18:

```

SELECT *
FROM results_t
WHERE targil_id = 18

```

18 %

Results Messages

	results_id	data_id	targil_id	method	result
1	5063587	1	18	SQL	-2.40178307887134
2	5063588	2	18	SQL	53.5964273800181
3	5063589	3	18	SQL	71.4820441960832
4	5063590	4	18	SQL	1.18105845609408
5	5063591	5	18	SQL	94.3188715577482
6	5063592	6	18	SQL	45.2090229368166
7	5063593	7	18	SQL	100.719607455064
8	5063594	8	18	SQL	61.6986800258644
9	5063595	9	18	SQL	85.3819119978226
10	5063596	10	18	SQL	90.556634568232
11	5063597	11	18	SQL	59.7070936334702
12	5063598	12	18	SQL	12.4653620562959
13	5063599	13	18	SQL	65.1602602874259
14	5063600	14	18	SQL	13.7337923408795
15	5063601	15	18	SQL	23.6891982328411
16	5063602	16	18	SQL	41.4518050285757
17	5063603	17	18	SQL	42.5651949570143
18	5063604	18	18	SQL	12.2471637416274
19	5063605	19	18	SQL	75.0229936297813
20	5063606	20	18	SQL	58.43911933399

Executing query...

השווות זמני ביצוע:

SQL:

The screenshot shows a SQL query window with the following code:

```
SELECT TOP 10*
FROM log_t
WHERE method = 'SQL';
```

Below the code is a results grid:

	log_id	targil_id	method	time_run
1	1	2	SQL	12.257
2	2	6	SQL	0.018
3	3	7	SQL	11.856
4	4	12	SQL	12.644
5	5	16	SQL	12.098
6	6	18	SQL	11.426
7	9	22	SQL	13.26

Python:

The screenshot shows a SQL query window with the following code:

```
SELECT TOP 10*
FROM log_t
WHERE method = 'Python_Pandas_Eval_Fixed';
```

Below the code is a results grid:

	log_id	targil_id	method	time_run
1	246	2	Python_Pandas_Eval_Fixed	26.0901420116425
2	247	6	Python_Pandas_Eval_Fixed	80.7969741821289
3	248	7	Python_Pandas_Eval_Fixed	59.5884108543396
4	249	12	Python_Pandas_Eval_Fixed	108.036586761475
5	250	16	Python_Pandas_Eval_Fixed	79.4303312301636
6	251	18	Python_Pandas_Eval_Fixed	119.058474302292

SQL ↔ Python בדיקת זהות תוצאות – השוואת

```
--ן&gt;&gt;--  
SELECT sql.data_id, sql.targil_id,  
       sql.result AS sql_result,  
       py.result AS py_result  
  FROM results_t sql  
 JOIN results_t py  
    ON sql.data_id = py.data_id  
   AND sql.targil_id = py.targil_id  
 WHERE sql.method = 'SQL-Fast'  
   AND py.method = 'Python'  
   AND ISNULL(sql.result, 0) <> ISNULL(py.result, 0);
```

results Messages

data_id	targil_id	sql_result	py_result
---------	-----------	------------	-----------

הפעלה



The query executed successfully.

| DESKTOP-D01DQH2\