

## המרכז האקדמי הרוב תחומי ירושלים, החוג למדעי המחשב

### אנליזה של בייג דאטה / חננאל פרל

סמסטר א', תשפ"ז

תרגיל 6 / 07.12.2025

תאריך הגשה:

יום חמישי 18.12.2025 בשעה 23:00

מטרות התרגיל:

השווואת זמני ריצה DuckDB SQLite

הוראות:

בעולמות של בייג דאטה, הרבה פעמים יש צורך להשוות זמני ריצות של כל מיני כלים ושיטות. זמני הריצה מושפעים לרוב מגודל הקלקט. בתרגיל זה נבחן שני כלים עם כמה אופציות. נרץ בכל אחד מהקונפיגורציות כמה שאלות עם כמה קלטים בגודלים שונים.

מה שנבדוק בתרגיל, יהיו 3 קונפיגורציות האלו:

(A) טבלאות שניצור בדאטاهבייס DUCKDB ועליהם נרץ השאלות.

`duckdb.connect("file.duckdb")`

(B) טבלאות בדאטاهבייס SQLITE ללא אינדקסים כלל.

`sqlite3.connect("file.sqlite")`

(C) טבלאות בדאטاهבייס SQLITE עם אינדקסים על עמודות רלוונטיות לפי שיקול דעתכם.

`sqlite3.connect("file.sqlite")`

`CREATE INDEX ..`

אנו משתמשים במדד H-TPC

<https://www.tpc.org/tpch/>

H-TPC הוא כלי מבחן ביצועים (Benchmark) סטנדרטי להערכת ביצועי מערכות כגון מחסני נתונים ובסיסי נתונים אנליטיים. המטרה העיקרית של H-TPC היא לדמות עומס עבודה אנליטי מציאותי OLAP המאפיין סביבות עסקיות. אנו נבחן את יכולת המרcura לבצע שאלות מורכבות על כמויות גדולות של נתונים. ונמדד את זמן ביצוע השאלות (Query Execution Time) ואת יכולת המרcura לעבוד נתונים במהירות.

כדי לייצר את הדטה נשתמש ברכיב ההרחבה tpch של DuckDB

[https://duckdb.org/docs/stable/core\\_extensions/tpch](https://duckdb.org/docs/stable/core_extensions/tpch)

גרסאות של חבילות ושל פיתון לפי מה שיש באנקונדה גרסה: 0-2025.06.06-Anaconda3-2025.06.06-0-Windows-x86\_64.exe  
אפשר למצוא בlienק כאן:

<https://repo.anaconda.com/archive/>

ולמשל עבור חלונות:

[https://repo.anaconda.com/archive/Anaconda3-2025.06-0-Windows-x86\\_64.exe](https://repo.anaconda.com/archive/Anaconda3-2025.06-0-Windows-x86_64.exe)

חבילות נוספות מותרכות:

1.4.2 DuckDB כiboldן, גרסה

28.1.0 Sqlglot גרסה

יש לייצר את הדטה בגודלים שונים  
נתחל מקנה מידת של 0.01 (הפרמטר  $sf = 0.01$ )  
את הנתונים ניצור על ידי DUCKDB , ואז נעביר אותם לטבלאות SQLITE.

נריץ כל שאלתה (מתוך ה 22 של TPCH), נריץ 3 פעמים ונשמר את הזמן החזוי (median).  
כל פעם נגדיל את הקנה מידת (לפי שיקול דעתכם בכמה כל פעם). נמחוק את כל הקבצים מהסבב הקודם  
(הקבצים של DUCKDB והקבצים של SQLITE). נשמר את הנתונים בטבלאות חדשות. נריץ ונשמר את הזמן  
ריצות של כל השאלות. נapis תוצאות מוגדרות של כל שאלתה.

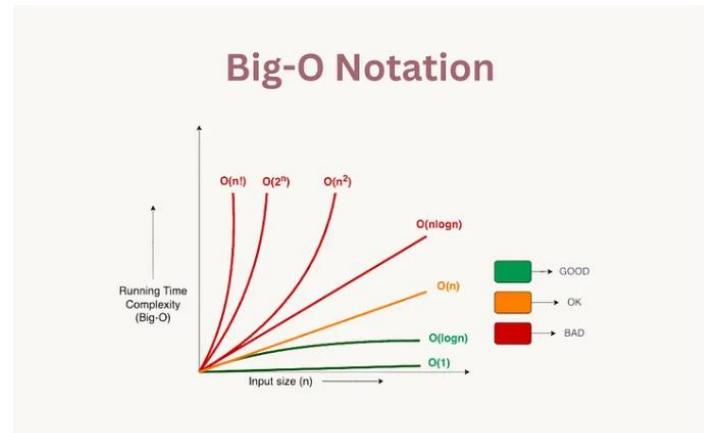
ברגע שהזמן ריצה על SQLITE נהים איטיים מדי יש להפסיק להריץ את SQLITE (אפשר לבדוק ידנית ולסמן  
בקוד בצורה קבוצה מסוימת מידת להפסיק להריץ SQLITE) ולהמשיך רק על DUCKDB. יש להמשיך להריץ  
את DUCKDB עד כמה פעמים עד שהקובץ נהיה גדול מידי, או נגמר הזמן, או זמן ריצה ארוך מדי לטעמכם (לא  
יותר מחצי שעה ריצה). כמות הסבבים תהיה סה"כ בערך 10. בהתאם לזה יש לתקן מה היה הקנה מידת בכל  
ריצה.

את הריצה של השאלות עצם יש לבצע יישור דרך הכלי הנבדק או DUCKDB בהתאם. לא  
דרך כלים אחרים (למשל אין להשתמש בPANDAS בשלב זה)  
את מדידת זמן הריצה נעשה בפייטון, למשל `time.time()`, ו��מה..

עבור כל שאלתה שהרצת יש לשמור:

- + מספר השאלתה
- + הקונפיגורציה
- + גודל הדאטאביס כולל בדיק
- + כמות זמן הריצה החזוי median
- + זמן נכון בשעון (של הריצה الأخيرة מתוך ה-3)

יש להדפיס את כל התוצאות למסך, ו גם לשמר בקובץ בטבלה CSV את כל התוצאות של זמן הרצה לפי כל הסוגים (כאן אפשר ע"י PANDAS או כל משחו דומה) כמו כן יש ליצור גרפים עבור כל שאלתיה: גרפּ קווּי שבו כל קונפיגורציה מקבלת צבע, ציר X - גודל הקטלט (לפי ה健全ה מידת = המשתנה SF), ציר Y - זמן ריצה. לדוגמה:



<https://medium.com/@clebertonfqc/big-o-notation-in-practice-ac192e23c16>

<https://www.bigoocheatsheet.com/>

שימו לב, בתוצאות שלכם, לאיזה עוקמה בתמונה לדוגמא ( $n | \log n$  |  $n \log n$  |  $n | 2^n$  |  $n | n!$  |  $n | \log n$ ), הזריםם שלכם מתאימים.

(לצורך יש להשתמש באחת מה宦官 הסטנדרטיות שמוקנות עם אנקונדה, למשל MATPLOTLIB וגדומה)

כמו כן, יש ליצור גרפּ נוסף מס' 23, סיכום שיכיל את זמן הרצה הממצאים של כל השאלות לכל קונפיגורציה.

יש לשים לב שהמחשב לא עסוק בדברים אחרים, ושאין תוכנות אחרות רצות ברקע, כדי שנוכל לקבל השוואה אמיתית.

יש לכתוב את סוג המחשב וכמות הזכרון והמעבד וכו על ידי שימוש בקוד זהה (לאו החלק של GPU)

<https://thepythontutorials.com/article/get-hardware-system-information-python>

`psutil | platform | datetime`

שימו לב, מבחינת הריצה, יש ליצור את הדטה הקטן, להכין את כל הקונפיגורציות ואז להריץ את כל השאלות, ולשמור הזמן ריצה וכו, ואז למחוק כל הקבצים. ולבסוף לגודל הבא וכו..

כל הקוד שモוגש צריך להיות שלכם מתחילה ועד סוףו, אין להגיאש קוד שמיישמו אחר כתוב (למעט הקוד של סוג המחשב וכמות הזכרון שצוין לעיל). אפשר להעזר באתר אינטראנט וצאים, אך יש להבין ולהיות מסוגל להסביר ולהצדיק את הקוד שモוגש.

זהו תרגיל תכני, ועל כן הבדיקות יהיו קפדיות גם מבחינת פיתון, וגם מבחינת מבנה הקוד וחלוקת לפונקציות וכו' וגם מבחינת צורת ההגשה וגדומה.

סך הכל צריך להגיש קובץ ZIP שמכיל:

- א. קובץ requirements.txt המכיל את כל חבילת הפיתון הנדרשת להתקנה והרצה.
- ב. קובץ פיתון שמכין את הדטה, מייצר ומכוון לבסיס הנתונים ומריץ את כל השאלות ושומר תוצאות בסוף.
- ג. קובץ טקסט המכיל את הפלט של הריצה שלכם.
- ד. קובץ CSV שמכיל את כל התוצאות של זמן הריצה לפי כל הסוגים.
- ה. קבצי התמונות של הגրפים – 1+22 קבצים, קובץ לכל שאלה והסבירם.
- ו. README עם הסברים ותוספות והחלטות לגבי דברים שלא מוגדרים בפירוש.  
(אין להגיש את הדטה שיצרתם! קובץ הפיתון אמרור לייצר את כל הדטה הנדרש לביצוע הרישות!)

**שם הקובץ ZIP צריך להיות שמות ות"ז של הזוג ש מגיש**

**בהצלחה!**