תרגיל DB

ליאור שפירא וכרמל גרוס

חלק א: מצורפים קבצי SQL

חלק ב:

1. הרצנו את השאילתה SELECT actorId FROM playsin WHERE character = ‘Sheriff’;

```
public=> select actorID from playsin where character like 'Sheriff';
public=> explain analyse select actorID from playsin where character like 'Sheri
ff';
                                                QUERY PLAN
-----------------------------------------------------------------------------------------------------
 Seq Scan on playsin  (cost=0.00..615.15 rows=50 width=4) (actual time=0.658..3.815 rows=50 loops=1)
   Filter: (("character")::text ~~ 'Sheriff'::text)
   Rows Removed by Filter: 32602
 Planning Time: 0.288 ms
 Execution Time: 3.861 ms
(5 rows)
```

ניתן לראות כי זמן חישוב השאילתה הינו $0.288 + 3.861 = 4.149$ שכן ללא אינדקס נעבוד על כל השורות של הטבלה ונבדוק האם ה- character זהה

לאחר מכן נוסיף את האינדקס CREATE INDEX char_name ON playsin(character)

```
ff';ic=> explain analyse select actorID from playsin where character like 'Sherif
                                                QUERY PLAN
-----------------------------------------------------------------------------------------------------
 Bitmap Heap Scan on playsin  (cost=4.67..122.36 rows=50 width=4) (actual time=0.124..0.254 rows=50 loops=1)
   Filter: (("character")::text ~~ 'Sheriff'::text)
   Heap Blocks: exact=37
   ->  Bitmap Index Scan on char_new_name  (cost=0.00..4.66 rows=50 width=0) (actual time=0.106..0.107 rows=50 loops=1)
         Index Cond: (("character")::text = 'Sheriff'::text)
 Planning Time: 0.440 ms
 Execution Time: 0.333 ms
(7 rows)
```

כעת ניתן לראות כי זמן חישוב השאילתה הוא $0.440 + 0.333 = 0.773$ שכן הוספת אינדקס מקצרת את זמן הריצה שכן נצטרך להגיע לעלים ולחפש בעלים המתאימים בלבד.

2. חישוב שאילתות:

הנחות:

- גודל בלוק הוא 1,000 בייטים.
- בטבלה Movies יש 10,000 שורות,
- כל שורה תופסת 150 בייטים.
- התכונה movieId תופסת 8 בייט.
- התכונה duration תופסת 8 בייט.
- התכונה genre תופסת 10 בייט.
- מצביע תופס 8 בייט.
- הערכים בduration בטבלה Movies מתפלגים אחיד בטווח [1,200]
- הערכים בgenre בטבלה מחולקים ל4 קטגוריות באופן אחיד.

ראשית נחשב את מספר הבלוקים שיש ב-movies: $\left\lceil \frac{10000}{6} \right\rceil = 1667 \rightarrow \left\lfloor \frac{1000}{150} \right\rfloor = 6$

א.

```
SELECT DISTINCT “exists”
FROM Movies
WHERE duration > 100
```

```
CREATE index on movies(duration)
```

1) עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה היא $\boldsymbol{I/O}$ 1667 שכן נצטרך במקרה הגרוע ביותר לעבור על כל השורות בטבלה.

2) כעת נחשב מהו דרגת הפיצול האופטימלית של האינדקס-

$$8 \cdot d + 8 \cdot (d-1) \leq 1000 \rightarrow \quad \boldsymbol{d = 63}$$

3) עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם.

גובה העץ: $\log_{\lceil \frac{d}{2} \rceil} 10{,}000 \rightarrow \lceil \log_{32} 10{,}000 \rceil = 3$

חישוב עם אינדקס: אין צורך לחפש על העלים ולכן העלות על הטיול העלים היא 1 שכן רק נצטרך לבקר בעלה אחד כלומר סה"כ העלות תהיה $3 + 1 = \boldsymbol{4\ I/O}$

ב.

```
SELECT avg (duration)
FROM Movies
WHERE duration > 100
CREATE index on movies(duration)
```

1) עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה היא $\boldsymbol{I/O}$ 1667 כבסעיף הקודם.

2) כעת נחשב מהו דרגת הפיצול האופטימלית של האינדקס-

$$8 \cdot d + 8 \cdot (d-1) \leq 1000 \rightarrow \quad \boldsymbol{d = 63}$$

3) עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם.

גובה העץ: $\log_{\frac{d}{2}} 10{,}000 \rightarrow \lceil \log_{32} 10{,}000 \rceil = 3$

חישוב עם אינדקס: $\left\lceil \frac{200-100}{200} \right\rceil = \frac{1}{2} \rightarrow \left\lceil \frac{\frac{1}{2} \cdot 10000}{32-1} \right\rceil = 162$

אזי סה"כ העלות תהיה $162 + 3 = \boldsymbol{165\ I/O}$

ג.

```
SELECT name
FROM Movies
WHERE movieid=200
CREATE index on movies(movieid)
```

1) עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה היא $\boldsymbol{I/O}$ 1667.

2) כעת נחשב מהו דרגת הפיצול האופטימלית של האינדקס-

$$8 \cdot d + 8 \cdot (d-1) \leq 1000 \rightarrow \quad \boldsymbol{d = 63}$$

3) עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם.

גובה העץ: $\log_{\frac{d}{2}} 10{,}000 \rightarrow \lceil \log_{32} 10{,}000 \rceil = 3$

מכיוון ש-*movieID* הוא מפתח הוא יופיע בעלה אחד בלבד וכן ניגש לטבלה פעם אחת אזי סה"כ העלות

$$3 + 1 + 1 = \mathbf{5}\ \boldsymbol{I/O}$$

ד. נתונה השאילתה

```
SELECT avg(duration)
FROM Movies
WHERE genre = 'Drama'
```

```
create index on movies(genre)
```

1) עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה היא $\boldsymbol{I/O}$ 1667.

2) כעת נחשב מהו דרגת הפיצול האופטימלית של האינדקס-

$$10 \cdot d + 8 \cdot (d-1) \leq 1000 \rightarrow \quad \boldsymbol{d = 56}$$

3) עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם.

גובה העץ: $\log_{\frac{d}{2}} 10{,}000 \rightarrow \lceil \log_{28} 10{,}000 \rceil = 3$

הערכים מתפלגים אחיד על 4 קטגוריות ולכן יש סה"כ $\frac{10000}{4} = 2500$ שורות עם ג'אנר דרמה. אזי עבור חישוב עם אינדקס נקבל כי יש $\frac{2500}{\frac{d}{2}-1} = 93$ בלוקים.

כעת הגישה לטבלה היא גישה למינימום מבין מספר השורות המתאימות למול מספר הבלוקים ולכן $\min(1667, 2500) = 1667$

אזי סה"כ העלות תהיה $1667 + 93 + 3 = \mathbf{1763}\ \boldsymbol{I/O}$

ה.

```
SELECT avg(duration)
FROM Movies
WHERE genre = 'Drama'
```

```
create index on movies(genre,duration)
```

1) עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה היא $\boldsymbol{I/O}$ 1667.

2) כעת נחשב מהו דרגת הפיצול האופטימלית של האינדקס-

$$10 \cdot (d-1) + 8 \cdot (d-1) + 8d \leq 1000 \rightarrow \quad \boldsymbol{d = 39}$$

3) עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם.

גובה העץ: $\log_{\frac{d}{2}} 10{,}000 \rightarrow \lceil \log_{20} 10{,}000 \rceil = 4$

יש סה"כ $\frac{10000}{4} = 2500$ שורות המתאימות לג'אנר דרמה.

עבור חישוב עם אינדקס: $\left\lceil \frac{2500}{\frac{d}{2}-1} \right\rceil = 132$, בנוסף הפעם יש אינדקס ולכן לא יעלה לנו לגשת לטבלה. אזי סה"כ העלות תהיה $132 + 3 = \boldsymbol{136\ I/O}$