ביצוע שאילתות במסדי נתונים רלציוניים

תזכורת: יחסים (relations)

רב-קבוצה (multiset) של איברים (שורות) ממכפלה הקרטזית של תחומים

students $\subseteq$ string $\times$ year $\times$ grade

:relation scheme

שם היחס והתכונות שלו, ותחומיהם.

Students(name,birth_year,grade)

שם פרטי	שנת לידה	ציון
אלה	1980	85
אלון	1963	66
נורית	1998	52
ארז	1959	73
צאלה	1980	98

שורות יכולות להופיע מספר פעמים, ולסדר יש חשיבות

תזכורת: פעולות על יחסים

פעולות שמחזירות יחס חדש

- $\cup \cap \setminus \times$ – איחוד, חיתוך, הפרש, מכפלה
- σ – בוחר חלק מהרשומות
- π – הטלה (projection) – בוחר חלק מהעמודות
- $\rho_{A\rightarrow B}$ – החלפת שם של שדה
- δ – הפוך את היחס לקבוצה (השאר עותק אחד מכל רשומה)
- $\bowtie$ – צירוף יחסים
- sort – מארגן מחדש את היחס על ידי מיון על פי תכונה או מספר תכונות

פונקציות הקבצה על עמודה המכילה מספרים

- max, min, average, sum – מחזירות מספר

פונקציות הקבצה על טבלה

- count – מחזירה מספר

תזכורת: סימונים

עבור יחס R, נסמן

T(R) = מספר הרשומות ב R.

V(R) = מספר הרשומות השונות של R: V(R)=T(δ(R)).

B(R) = מספר הבלוקים (גזרות) ש R תופס.

M = מספר הבלוקים שניתן להחזיק בזיכרון הראשי.

לא נחייב על הפעולות הדרושות לכתיבת התוצאה לדיסק.

- אם תוצאת ביניים, מועבר לשלב הבא (pipeline) ולא נכתב לדיסק
- אם זו פעולה אחרונה, גודל הפלט אינו תלוי בשיטת החישוב ולא משפיע על בחירתה

תזכורת: אלגוריתמים לביצוע פעולה בודדת

- מעבר יחיד (לא תמיד ניתן ליישום)
- לולאות מקוננות (לא תמיד ניתן ליישום)
- אלגוריתמים מבוססי מיון
- אלגוריתמים מבוססי ערבול
- אלגוריתמים המשתמשים באינדקס קיים של יחס

סיבוכיות אלגוריתמי מעבר יחיד

הפעולה	דרישות הזיכרון	גישות דיסק
σ, π, ממוצע	O(1)	B(R)
$\cup$	O(1)	B(R) + B(S)
δ במעבר אחד	δ(R)	B(R)
$\bowtie \times \cap \setminus$	δ(S)	B(R) + B(S)

השיטה היעילה ביותר מבחינת זמן, אבל דרישות זיכרון גבוהות

לולאות מקוננות nested loops

אלגוריתם **נאיבי** לביצוע פעולות בינריות בין יחסים גדולים:

```
for each tuple s in S
 for each tuple r in R
     process tuples r and s
```

למשל עבור צירוף:

```
for each block S {
 read a tuple s;
 for each block in R {
     read a tuple r;
     if s and r join to make a tuple t
            output t; }
}
```

סיבוכיות זמן: T(S)B(R)

שיטה שלרב עובדת, אבל מאוד לא יעילה מבחינת זמן

צירוף בלולאות מקוננות: שיפור

ננצל M-4 בלוקים של זיכרון לאחסון S

איטרציה ראשונה:

קרא M-4 בלוקים של S

קרא את כל R (עם חוצצים כפולים), ולכל רשומה r ב-R

– חפש בין הבלוקים של S שבזיכרון את כל הרשומות s שעבורן: r.A = s.A

– הוצא את s $\bowtie$ r לפלט

מספר גישות דיסק בכל מחזור: $M + B(R) \cong (M-4) + B(R)$

מספר המחזורים: $B(S) / M \cong \lceil B(S)/(M-4) \rceil$ אם M >> 4

סך-הכל:

$B(S)/M(M + B(R)) \cong B(S) + B(S)\ B(R) / M \cong B(S)\ B(R) / M$

אם אחד היחסים גדול בהרבה מהשני, נעדיף שהיחס הגדול יהיה ה- R.

צירוף מבוסס מיון

איפה משתמשים בהנחה הזו?

נניח ש-R ו-S ממויינים לפי A, קבוצת התכונות המשותפות ל-R ול-S
ונניח שלכל ערך a של A מתקיים $|\sigma_{A=a}(S)| \le M$.

עבור על S ועל R.
הכנס לזיכרון את הרשומות הראשונה s של S ו- r של R.
אם $\pi_A(r) > \pi_A(s)$ החלף את s ברשומה הבאה של S
אחרת אם $\pi_A(r) < \pi_A(s)$ החלף את r ברשומה הבאה של R
אחרת קרא לזיכרון את כל הרשומות של S עבורן $\pi_A(s') = \pi_A(s)$
לכל רשומה r' של R עבורה $\pi_A(r') = \pi_A(s)$:
לכל s' בזיכרון הוצא לפלט את הרשומה $r' \bowtie s'$

סיבוכיות: B(S) + B(R)
בלי לקחת בחשבון את הזמן הדרוש לכתיבת הפלט, שהוא $B(S \bowtie R)$

דוגמה לביצוע צירוף באמצעות מיון

R

A	B	C
a	b	c
a1	b1	c
a1	b1	c1
a1	d	c2

S

C	D
c	d1
c2	d2
c2	d3

$R \bowtie S$

A	B	C	D
a	b	c	d1
a1	b1	c	d1
a1	d	c2	d2
a1	d	c2	d3

סיבוכיות אלגוריתמים מבוססי מיון

הפעולה	מספר גישות לדיסק לקריאת קלט	מספר הבלוקים בזיכרון הראשי
δ	B(R)	1
Min, Max	1	1
$\cap, \cup, /$ (שומר מיון)	B(S)+B(R)	*2
$\bowtie$	B(S)+B(R)	*2

* בהנחה שבכל יחס, כל הרשומות הזהות בתכונות הצירוף נכנסות בבלוק אחד

סיכום פעולות בעזרת ערבול

עבור $B(R) \le M^2 / 2$ מספר גישות הדיסק הוא:

δ 3B(R)

חיתוך, צירוף, וכו' 3B(S) + 3B(R)

אם B(R) גדול יותר, נחליף את המקדם 3 ב-5 (מדוע?)

סיכום: אלגוריתמים לביצוע פעולה בודדת

- מעבר יחיד (לא תמיד ניתן ליישום)
- לולאות מקוננות (לא תמיד ניתן ליישום)
- אלגוריתמים מבוססי מיון
- אלגוריתמים מבוססי ערבול
- אלגוריתמים המשתמשים באינדקס קיים של יחס

אלגברה רלציונית

אפשר לכתוב כל מיני ביטויים אלגבריים עם הפעולות שהכרנו, למשל:

הביטוי:

$\pi_{name,addr}(\sigma_{title=\text{“harry potter”} \text{ and } gender = F}(MovieStar \bowtie StarsIn))$

עם הסכמות:

MovieStar(name, addr, gender), StarsIn(title, year, name)

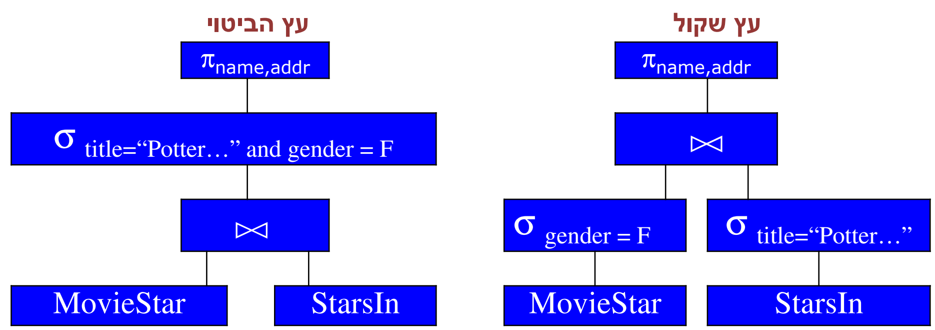
מחזיר את שמות כל השחקניות של "הארי פוטר" ואת הכתובות שלהן.

תרגום של

```
SELECT name, addr
FROM MovieStar, StarsIn
WHERE title="harry potter" and
      gender=F and
      MovieStar.name=StarsIn.name;
```


עץ של ביטוי באלגברה רלציונית

$\pi_{name,addr}(\sigma_{title=\text{“Potter…”}, gender = F}(MovieStar \bowtie StarsIn))$



$\pi_{name,addr}(\sigma_{gender = F}(MovieStar) \bowtie \sigma_{title=\text{“Potter…”}}(StarsIn))$

שקילויות באלגברה רלציונית

אסוציאטיביות של צירוף:

$(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$

קומוטטיביות של צירוף:

$R \bowtie S = S \bowtie R$

הקדמת ביצוע של בחירה (ל S ול R אותן תכונות):

$\sigma_C(R \cap S) = \sigma_C(R) \cap S = R \cap \sigma_C(S) = \sigma_C(R) \cap \sigma_C(S)$

אם תנאי הבחירה C מתייחס לתכונות משותפות של R ושל S:

$\sigma_C(R \bowtie S) = \sigma_C(R) \bowtie S = R \bowtie \sigma_C(S) = \sigma_C(R) \bowtie \sigma_C(S)$

הקדמת ביצוע של פעולת בחירה

ביצוע של פעולת בחירה מקטין את מספר הרשומות ביחס,
כדי להיפטר מרשומות כמה שיותר מוקדם, נוריד את הפעולה בעץ הביטוי

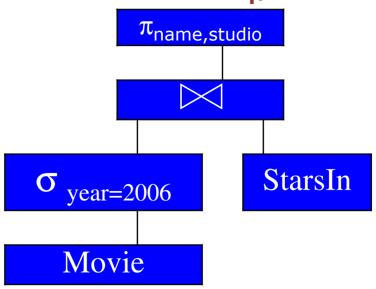
$\sigma_{\text{title} = \text{"Potter..."},\ \text{gender} = F}(\text{MovieStar} \bowtie \text{StarsIn}) =$
$\sigma_{\text{gender} = F}(\text{MovieStar}) \bowtie \sigma_{\text{title}=\text{"Potter..."}}(\text{StarsIn})$

$\sigma_{\text{gender} = F}$ יקטין (במחצית?) את גודל היחס
$\sigma_{\text{title}=\text{"Potter..."}}$ יקטין מאוד את היחס (ייכנס כולו לזיכרון)

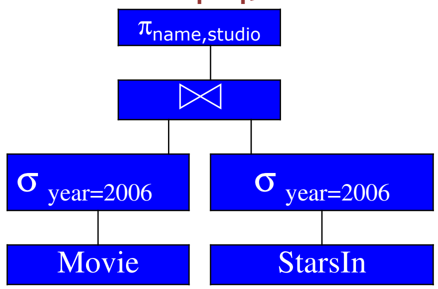
יותר מתוחכם: העלאה לצורך הורדה

Movie(title, year, studio) , StarsIn(title, year, name)
$\pi_{\text{name,studio}}(\sigma_{\text{year} = 2006}(\text{Movie}) \bowtie \text{StarsIn})$

עץ הביטוי



עץ שקול



$\pi_{\text{name,studio}}(\sigma_{\text{year} = 2006}(\text{Movie}) \bowtie \sigma_{\text{year} = 2006}(\text{StarsIn}))$

הקדמת הטלות

<u>המוטיבציה:</u>
הטלה מקטינה את גודל הרשומות,
בדרך-כלל, כדאי להקטין את הרשומות כמה שאפשר יותר מוקדם.

מתי לא כדאי להוריד את ה- π בעץ הביטוי?

כאשר ליחס יש אינדקס,
ביצוע ההטלה ייצור יחס חדש חסר אינדקס.
עבור יחס ממוין, ההטלה אינה בהכרח ממוינת.

הקדמת δ

פעולת δ מקטינה את מספר הרשומות,
בדרך כלל, כדאי לבצעה מוקדם ככל האפשר.

$$\begin{aligned}\delta(R \bowtie S) &= \delta(\delta(R) \bowtie S)\\ &= \delta(R \bowtie \delta(S))\\ &= \delta(R) \bowtie \delta(S)\end{aligned}$$

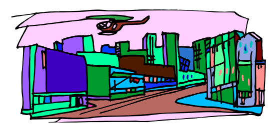
ואפילו יותר מפעם אחת

כמו הטלה, δ עלולה לגרום לאובדן של אינדקס,
כך שלפעמים הקדמת הפעולה אינה כדאית.

הערכת גודל התוצאה

מספר גישות הדיסק לביצוע פעולה תלוי בגודל הקלט, אבל ...

- כאשר מחשבים זמן ביצוע לעץ, זמן הביצוע של צומת פנימי תלוי בגודל הפלט של בניו, ועל כן יש לחשב את גודל התוצאה של ביצוע הפעולות הרלציוניות המתאימות להם.



- בד"כ, אי-אפשר לחשב מראש את הגודל במדויק, ולכן מטרתנו היא לקבל הערכה.
- דוגמה להערכת גודל תוצאה בהטלה π: אם ליחס R יש n רשומות, ובשדות שנבחרו יש f בתים, אזי הגודל של $\pi(R)$ הוא $n \cdot f$ בתים

הערכת גודל תוצאת בחירה σ

מספר הערכים השונים שיש לתכונה A

$V(R,A) = T(\delta(\pi_A(R)))$

תוחלת מספר הרשומות עם ערך ספציפי הוא

$E(T(\sigma_{A=x}(R))) = T(R) / V(R,A)$

חישוב זה אינו תלוי בהתפלגות הערכים בתוך R, אלא רק בהנחה שהערך x בפעולת ה- σ נבחר באופן אחיד מתוך הערכים שב R

לדוגמה, אם לתכונה A שני ערכים a, b ($V(R,A) = 2$) ומתקיים:

$P(\pi_A(R) = a)= 0.9,\ P(\pi_A(R) = b)= 0.1$

ונניח ש-x נבחר בהסתברות שווה בין a,b, אז

$$E(T(\sigma_{A=x}(R))) = \tfrac{1}{2} T(\sigma_{A=a}(R)) + \tfrac{1}{2} T(\sigma_{A=b}(R)) = \tfrac{1}{2}(0.9\, T(R)) + \tfrac{1}{2}(0.1\, T(R)) = T(R)/2 = T(R) / V(R,A)$$

הערכת גודל שאילתות טווח $T(\sigma_{A\le x}(R))$

אם ההתפלגות על התחום ידועה:

$E(T(\sigma_{A\le x}(R))) = P(A \le x)\, T(R)$

אחרת נניח שההתפלגות אחידה:

- עבור תחום רציף $A = [a,b]$:

$E(T(\sigma_{A\le x}(R))) = (x-a)/(b-a)\, T(R)$

- עבור תחום סופי $A= \{a_1,\dots,a_n\}$ כאשר $a_1< a_2 < \cdots <a_n$:

$E(T(\sigma_{A\le a_i}(R))) = (i / n)\, T(R)$

- עבור תחום בגודל לא-ידוע, נניח באופן שרירותי שמתקיים:

$T(\sigma_{A\le x}(R)) = \tfrac{1}{3}T(R) \Rightarrow T(\sigma_{A>x}(R)) = \tfrac{2}{3}T(R)$

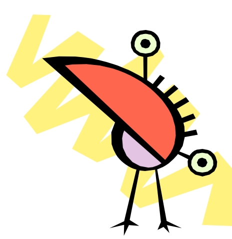
הערכת גודל שאילתות טווח 2

מה עם $T(\sigma_{x\le A\le y}(R))$?

$\sigma_{x\le A\le y}(R) = \sigma_{x\le A}(\sigma_{A\le y}(R))$

עבור תחום לא ידוע נניח ש-2/9 מהרשומות מקיימות את התנאי לפי השיקול הבא:

$T(\sigma_{x\le A\le y}(R)) = \tfrac{2}{3} T(\sigma_{A\le y}(R)) = \tfrac{2}{3}\,\tfrac{1}{3} T(R) = (2/9)T(R)$



הערכת גודל תוצאת δ

נביט ביחס בעל סכמה $R(A_1,\dots,A_n)$

בהגדרה $T(\delta(R)) = V(R)$, אבל בדרך-כלל $V(R)$ אינו ידוע.

נסמן $V(R,A_i) = V(\pi_{A_i}(R))$

מתקיים $1 \le V(R) \le \prod_{1\le i\le n} V(R,A_i)$

כלל אצבע: $V(R) = \min\left\{\prod_{1\le i\le n} V(R,A_i), \frac{1}{2}T(R)\right\}$

הערכת גודל צירוף $R(X,Y) \bowtie S(Y,Z)$

הנחות:

- Y היא תכונה בודדת
- ערכי Y ממוינים כך ש- $p(y_1)\ge\dots\ge p(y_n)$
- ערכי Y הם רישא של הסדרה $y_1,\dots,y_n$

כלומר, $y_i\in\pi_Y(R) \Leftarrow y_{i+1}\in\pi_Y(R)$

דוגמה:

נסתכל על ההתפלגות של אותיות באנגלית e, t, a, o, …, q, z
לפי סדר שכיחות (e היא האות הנפוצה ביותר, ו-z הנדירה ביותר)
נניח שהשדה Y מכיל אות בודדת בהתפלגות הנ"ל.
אז ההנחה אומרת שאם $o \in V(Y,R)$ אז גם $e, t, a \in V(Y,R)$

הערכת גודל צירוף $R(X,Y) \bowtie S(Y,Z)$

הנחות:

- Y היא תכונה בודדת
- ערכי Y ממוינים כך ש- $p(y_1)\ge\dots\ge p(y_n)$
- ערכי Y הם רישא של הסדרה $y_1,\dots,y_n$

כלומר, $y_i\in\pi_Y(R) \Leftarrow y_{i+1}\in\pi_Y(R)$

⇦ אם $V(S,Y) \le V(R,Y)$ אז $\pi_Y(S)\subseteq\pi_Y(R)$

- שמירת ערכים: $\pi_X(R\bowtie S) = \pi_X(R)$
  - לכל התכונות של R
  - באופן סימטרי, לכל התכונות של S

הערכת גודל צירוף $R(X,Y) \bowtie S(Y,Z)$

אם $V(S,Y) \le V(R,Y)$ אז נעריך כי:
ניתן לשלב זוג רשומות $r\in R$, $s\in S$ בהסתברות $1/V(R,Y)$
⇦ בממוצע s ישתלב עם $T(R)/V(R,Y)$ רשומות מ-R
⇦ $E(T(R\bowtie S)) = T(S)\cdot T(R)/V(R,Y)$

אם $V(S,Y)>V(R,Y)$,
אז $E(T(R\bowtie S)) = T(S)\cdot T(R)/V(S,Y)$

ובאופן כללי:
$E(T(R\bowtie S)) = T(S)\cdot T(R)/\max\{V(S,Y),V(R,Y)\}$

השוואת אלטרנטיבות לחישוב

דוגמה: חישוב $R(X,Y) \bowtie S(Y,Z) \bowtie U(Z,W)$

R(X,Y)	S(Y,Z)	U(Z,W)
T(R)=1000	T(S)=2000	T(U)=5000
V(R,Y) = 20	V(S,Y) = 50	
	V(S,Z) = 100	V(U,Z) = 500

ראשית, $R(X,Y) \bowtie S(Y,Z)$:

$$E(T(R \bowtie S)) = T(S) \cdot T(R)/\max\{V(S,Y),V(R,Y)\}$$
$$= 2{,}000 \times 1{,}000 / 50 = 40{,}000$$

הערכת $T(R \bowtie S \bowtie U)$

$$T((R(X,Y) \bowtie S(Y,Z)) \bowtie U(Z,W)) =$$
$$= T(R \bowtie S)\, T(U) / \max\{V(R \bowtie S, Z), V(U,Z)\}$$

R(X,Y)	S(Y,Z)	U(Z,W)
T(R)=1000	T(S)=2000	T(U)=5000
V(R,Y) = 20	V(S,Y) = 50	
	V(S,Z) = 100	V(U,Z) = 500

בגלל ההנחה על שמירת הערכים

$$V(R \bowtie S, Z) = V(S,Z) = 100$$

לכן:

$$T((R \bowtie S) \bowtie U) = 40{,}000 \times 5{,}000 / \max\{100, 500\}$$
$$= 400{,}000$$

הערכת $T(R \bowtie S \bowtie U)$

$$T(S \bowtie U) = T(S) \times T(U) / \max\{V(S, Z),V(U,Z)\} = 2{,}000 \times 5{,}000 / 500 = 20{,}000$$

R(X,Y)	S(Y,Z)	U(Z,W)
T(R)=1000	T(S)=2000	T(U)=5000
V(R,Y) = 20	V(S,Y) = 50	
	V(S,Z) = 100	V(U,Z) = 500

בגלל ההנחה על שמירת ערכים

$$V(S \bowtie U, Y) = V(S,Y) = 50$$

$$T(R \bowtie (S \bowtie U)) = T(R) \times T(S \bowtie U)/\max\{V(R,Y),V(S \bowtie U,Y)\}$$
$$= 1{,}000 \times 20{,}000/\max\{20,50\} = 400{,}000$$

כיוון ש $(R \bowtie S) \bowtie U) = R \bowtie (S \bowtie U)$, קיבלנו אותו ערך בשתי הדרכים

הערכת זמן החישוב לפי גודל תוצאות הביניים

כדי לקבל הערכה גסה לזמן החישוב,
נסכם את גודלי תוצאות הביניים (בלי העלים והשורש).

דוגמה קודמת: חישוב $R(X,Y) \bowtie S(Y,Z) \bowtie U(Z,W)$

א. $(R \bowtie S) \bowtie U$

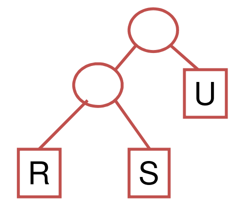
תוצאת הביניים: $E(T(R \bowtie S)) = 40{,}000$

ב. $R \bowtie (S \bowtie U)$

תוצאת הביניים: $E(T(S \bowtie U)) = 20{,}000$

Pipelining

ניתן לחשב את $(R \bowtie S) \bowtie U$ בשני אופנים:



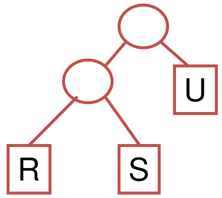
א. לכתוב את $R \bowtie S$ לדיסק ולקרוא אותו
שוב לצורך חישוב הצירוף שבשורש (עם U).

ב. לבצע pipelining:
להעביר כל שורה מיד עם היווצרותה
לתכנית שתבצע את הצירוף שבשורש.

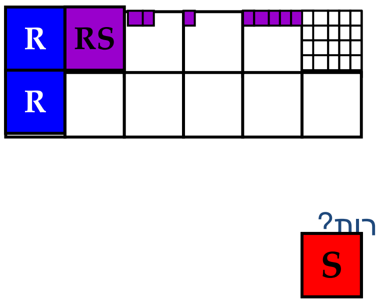
ביצוע pipeline דורש יותר זיכרון.
אם אין מספיק זיכרון, ייתכן שנאלץ לבצע אלגוריתם פחות יעיל.

Pipelining- דוגמא

$(R \bowtie S) \bowtie U$



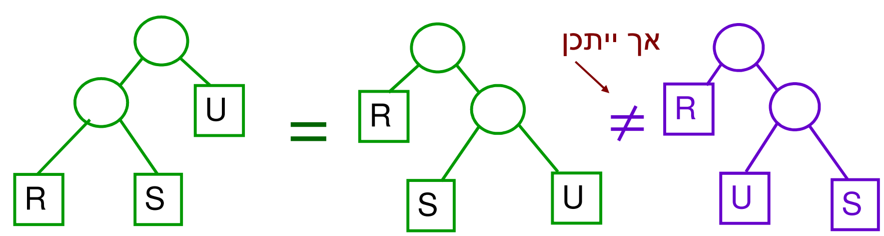
- הזיכרון בגודל 12 חוצצים, B(R)=2, B(S)=30, B(U)=40
- נבצע $RS=(R \bowtie S)$ במעבר יחיד כאשר R בזיכרון:
  - 2 חוצצים לשמירת R בזיכרון
  - 2 חוצצים לקריאה יעילה של S
  - נותרו 8 חוצצים: נשתמש בהם לחלוקה לדליים של התוצאה
- נחלק בהמשך את U לדליים (כמה?)
  לצורך חישוב התוצאה הסופית
- החיסכון: $2{*}B(R \bowtie S)$ גישות לדיסק



- האם אפשר להשתמש גם בשיטות ביצוע אחרות?

בחירת עץ החישוב

עבור פעולות אסוציאטיביות:

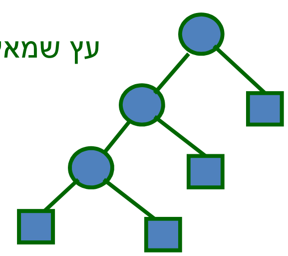


עבור פעולות קומוטטיביות ואסוציאטיביות שלושת העצים שקולים.

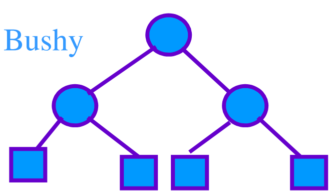
שימוש בעץ חישוב שמאלי

עץ שמאלי מאפשר ביצוע pipeline: בכל צומת, היחס השמאלי מתקבל מחישוב והימני נקרא מהדיסק

עץ שמאלי



Bushy



<u>היוריסטיקה:</u> לסדר את היחסים לפי גודלם
- היחס הקטן ביותר יתאים לעלה השמאלי ביותר,
- כך שתוצאות הביניים תהיינה קטנות ככל הניתן.

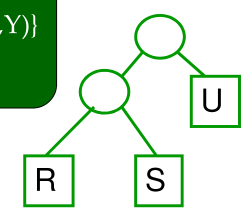
בחזרה לדוגמה הקודמת

R(X,Y)	S(Y,Z)	U(Z,W)
T(R)=1000	T(S)=2000	T(U)=5000

העץ המתקבל מתוך ההיוריסטיקה

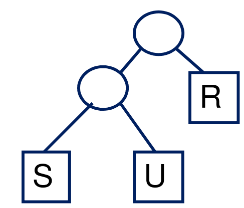
$E(T(R \bowtie S))$
$= T(S)T(R)/\max\{V(S,Y),V(R,Y)\}$
$= 2{,}000 \times 1{,}000 / 50$
$= 40{,}000$

אינו אופטימלי



Estimated cost = 40,000

עץ אופטימלי



Estimated cost = 20,000

דוגמה מסכמת

	W(A,B)	X(B,C)	Y(C,D)	Z(D,E)
T	100	200	300	400
V	A: 20	B: 50	C: 50	D: 40
V	B: 60	C: 100	D: 50	E: 100

רוצים לבצע $W \bowtie X \bowtie Y \bowtie Z$

נשתמש בעץ שמאלי: $((W \bowtie X) \bowtie Y) \bowtie Z$

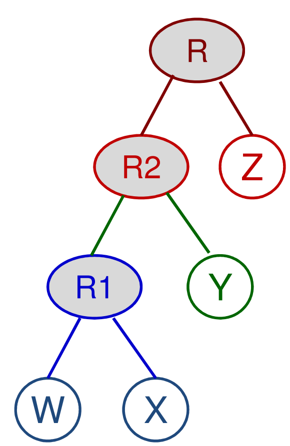
נעריך תחילה את זמן הצירוף ע"י חישוב גדלי תוצאות הביניים

גודל $R1 = W \bowtie X$: 333 רשומות

גודל $R2 = (W \bowtie X) \bowtie Y$: 1000 רשומות

מחיר: $333 + 1000 = 1333$

דוגמה מסכמת: תכנון מפורט (1)



נבצע את $R1=W \bowtie X$ במעבר יחיד.

את רשומות R1 נעביר ישירות לפעולה הבאה.

את $R2 = R1 \bowtie Y$ נבצע ע"י ערבול:

נחלק את R1 לדליים שייכתבו על הדיסק

נחלק את רשומות Y לדליים

נבצע צירוף בין הדליים המתאימים של Y ושל R1;

ונעביר לפעולה הבאה.

את $R2 \bowtie Z$ נבצע גם ע"י ערבול:

נחלק את רשומות R2 לדליים

נחלק את רשומות Z לדליים

נבצע צירוף בין הדליים המתאימים של Z ושל R2

את התוצאה נעביר לפלט.

דוגמה מסכמת: תכנון מפורט (2)

נניח שהזיכרון יכול להכיל 20 חוצצים בני 10 רשומות כ"א.

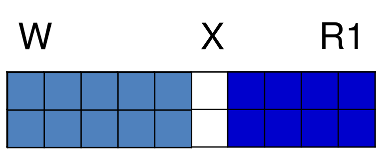
במקרה זה ניתן לבצע את $R1=W \bowtie X$ במעבר יחיד:

W דורש 100/10 חוצצים.

נכניס את W לזיכרון ונקרא את X סדרתית (שני חוצצים).

יישאר מקום לעוד 8 = 2 - 10 – 20 חוצצים.

התוצאה תועבר ישירות ל- $R1 \bowtie Y$ מבלי לכתוב אותה על הדיסק.



המחיר: קריאת W: 10 = 100/10 גישות

קריאת X: 20 = 200/10 גישות

סה"כ: 30 גישות

דוגמה מסכמת: תכנון מפורט (3)

בחישוב $R2 = R1 \bowtie Y$, היחסים לא נכנסים לזיכרון, ולכן נשתמש בערבול

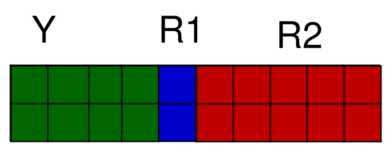
לערבול של R1 אין צורך בחוצצים עבור הקלט. למה?
מהפעולה הקודמת נותרו 8 חוצצים, מספר הדליים $\le 8/2 = 4$

מחיר חלוקת R1 הוא $333/10 = 34$ גישות דיסק.
לחלוקת Y נזדקק ל- $2\times300/10 = 60$ גישות דיסק נוספות.
הצירוף מחייב את קריאת כל דליי R1 ו-Y, $34 + 30 = 64$ גישות דיסק.
סה"כ: 158 גישות.

לביצוע הצירוף נקרא דלי שלם של Y לזיכרון, $\lceil(300/10)/4\rceil = 8$ חוצצים.

צריך שני חוצצים נוספים כדי לקרוא את הדליים של R1.

— נותרו 10 חוצצים.



דוגמה מסכמת: תכנון מפורט (4)

לביצוע $R2 \bowtie Z$, שוב נשתמש בערבול.
נחלק את R2 ל-5 דליים, כי נותרו 10 חוצצים.
הכתיבה תדרוש עוד $1000/10 = 100$ גישות.

אחר-כך, נחלק גם את Z ל-5 דליים. דורש $2\times400/10 = 80$ גישות.

נעבור על כל דליי Z:
נקרא כל דלי לזיכרון, ונבצע צירוף עם הדליים המתאימים של R2.
כל דלי של Z מכיל $\lceil 40/5 \rceil = 8$ חוצצים, ונכנס כולו לזיכרון.
נבצע צירוף במעבר יחיד עם הדלי המתאים של R2.

פעולה זו תדרוש את קריאת כל R2 וקריאת Z.
סה"כ $40 + 100 = 140$ גישות.

דוגמה מסכמת: תכנון מפורט (5)

סה"כ

חישוב R1	30
חישוב R2	158
חלוקת R2	100
חלוקת Z	80
ביצוע הצירוף	140
	508 גישות דיסק.

ניתן לבצע הערכה דומה לדרכים אחרות לחישוב הביטוי, ולבחור בניהן.

לתרגול עצמי: בצעו את החישוב עם לולאות מקוננות, ועם גודל זיכרון כפול.

סיכום

- חלק ניכר מהפעולות על קבצים כיום נעשות באמצעות מסדי נתונים רלציוניים (טבלאיים)
- הגדרת הפעולה במסד הנתונים מנותקת מאלגוריתם המימוש
- המימוש נבחר (אוטומטית) ע"י תכנית query optimizer על סמך הפעולות הדרושות, והערכות של גודלי תוצאות הביניים
- לכל מימוש מעריכים את הזמן הדרוש, ובוחרים את המימוש עם תוחלת הזמן המינימאלית
- אפשר לשנות את המימוש אם מתברר שההנחות לא היו נכונות