

סיכום ונוסחאון במערכות בסיסי נתונים

קורס 20277 · עודכן ב-2026

(נכתב על ידי דניאל וונג ומנחם מנדל קליש – www.wongi.io)

כל הכללים, האלגוריתמים והנוסחאות לפי מדריך הלמידה של הקורס, עם מספרי העמודים שבו.
האלגוריתמים כתובים כצדדים, ולצידם דוגמה פתורה ושרטוט.

תוכן העניינים

<u>נושא</u>	<u>עמוד</u>	<u>נושא</u>	<u>עמוד</u>
אלגברה של יחסים	1	טיפוסי נתונים מורכבים	9
SQL	1	אינדקסים	11
מודל ישויות-קשרים	3	עיבוד שאילתות	15
תלויות פונקציונליות ותיכון	5		

ביבליוגרפיה

ארד, י' (עורכת). מערכות בסיסי נתונים – מדריך למידה (20277). רעננה: האוניברסיטה הפתוחה.
Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill.

1 אלגברה של יחסים

1.1 פעולות יסודיות

ניסוח המשפט	כתיב מתמטי
יחסים תואמים – אותה דרגה (אותו מספר עמודות) ואותם תחומי תכונות, בהתאמה. רק על יחסים תואמים מותר להפעיל איחוד, חיתוך והפרש. (עמ' 29)	wish, qualified – תואמים applicant, wish – לא (דרגה 3 מול 2)
איחוד / חיתוך / הפרש – כמו בקבוצות: שורות שנמצאות באחד מהם / בשניהם / בראשון ולא בשני. חיתוך אפשר לבטא בעזרת שני הפרשים. (עמ' 26–28)	$r \cup s \quad r \cap s \quad r - s$ $r \cap s = r - (r - s)$
בחירה σ – משאירה את השורות שמקיימות תנאי (אפשר לחבר תנאים ב- $\neg, \vee, \wedge$). פעולה אונרית. (עמ' 29–33)	(schedule) 'שלישי' $\sigma_{\text{day}=\text{'שלישי'}}$
הטלה π – משאירה רק את העמודות המבוקשות, ומוחקת כפילויות (התוצאה קבוצה). (עמ' 29–33)	$\pi_{\text{class, hour}}(\text{schedule})$
מכפלה קרטזית $\times$ – כל צירוף של שורה מ- r_1 עם שורה מ- r_2 ; התבנית היא שרשרת שתי התבניות (תכונה עם אותו שם מסמנים בשם היחס: $r.A$). (עמ' 34)	$ r_1 \times r_2 = n_{r_1} \cdot n_{r_2}$
צירוף לפי תנאי $\bowtie_\theta$ – מכפלה ואחריה בחירה לפי התנאי θ . צירוף טבעי $\bowtie$ – התנאי הוא שוויון בכל העמודות בעלות אותו שם, וכל עמודה משותפת מופיעה פעם אחת. (עמ' 38)	$r_1 \bowtie_\theta r_2 = \sigma_\theta(r_1 \times r_2)$ $r \bowtie s = \pi_{R \cup S}(\sigma_{r.A=s.A}(r \times s))$
שכפול ρ – אותו יחס בשם חדש (ואפשר גם שמות חדשים לעמודות). משמש כדי להשוות בין שתי שורות של אותו יחס: מכפלה של היחס עם עצמו. (עמ' 41–42)	$\rho_s(r) \quad \rho_{s(A_1, \dots, A_n)}(r)$
השמה $\leftarrow$ – נותנת שם לתוצאה, כדי לפצל שאילתה לשלבים. (עמ' 43)	$T \leftarrow \pi_{\text{id}}(\sigma_{\dots}(r))$
חילוק $r_1 \div r_2$ – התבנית של המחלק r_2 חלקית ממש לזו של המחולק r_1 ; התוצאה: צירופי ערכים בשאר העמודות של r_1 , שמופיעים ב- r_1 עם כל שורה של r_2 . מתאים ל"כל ה...". (עמ' 44–45)	wish $\div$ $\pi_{\text{j-no}}(\text{job})$ "מועמדים שמעוניינים בכל המשרות"

1.2 תבניות לשאלות נפוצות

ניסוח המשפט	כתיב מתמטי
"גם X וגם Y" (בשורות שונות) – שני ביטויים עם אותן עמודות, וחיתוך. בתנאי אחד 'ים' $\wedge$ 'אוור' = mode = ריקה – בכל שורה ערך אחד.	$\pi_{\text{id}}(\sigma_X(\dots)) \cap \pi_{\text{id}}(\sigma_Y(\dots))$
"לא" / "אף פעם" – כל המועמדים, פחות אלה שכן. (עמ' 41)	$\pi_{\text{class}}(r) - \pi_{\text{class}}(\sigma_{\text{subject}=\text{'Biology'}}(r))$
"לפחות שני ... שונים" – מכפלה של היחס עם עצמו, אותו מפתח וערכים שונים. "זוגות, כל זוג פעם אחת" – $A.\text{id} < B.\text{id}$. (עמ' 42)	$\sigma_{t.\text{tch}=s.\text{tch} \wedge t.\text{sub} \neq s.\text{sub}}(\rho_t(r) \times \rho_s(r))$
מקסימום – מוצאים את כל הערכים שקטנים מערך אחר, ומורידים אותם מכל הערכים. (עמ' 43–44)	$T \leftarrow \sigma_{a.h < b.h}(\rho_a(r) \times \rho_b(r))$ $\pi_h(r) - \pi_{a.h}(T)$
"לכל" – חילוק, כשהמחלק זהה לכולם. כשלכל אחד קבוצה משלו (למשל "לכל הבקשות שלי"): מוצאים את מי שחסר לו משהו, ומורידים אותו.	$M \leftarrow \pi_{\text{id}}(H - D)$ $\pi_{\text{id}}(H) - M$

2 SQL

2.1 מבנה שאילתה

ניסוח המשפט	כתיב מתמטי
סדר החישוב – from (מכפלה/צירוף) $\leftarrow$ where (שורות) $\leftarrow$ group by (קבוצות) $\leftarrow$ having (קבוצות) $\leftarrow$ select (עמודות) $\leftarrow$ order by. בתנאי ב-where אי אפשר להשתמש בפונקציית הקבוצה. (פרק 3)	SELECT a, COUNT(*) FROM r JOIN s ON r.k = s.k WHERE ... GROUP BY a HAVING COUNT(*) > 50 ORDER BY a;

ניסוח המשפט	כתיב מתמטי
כפילויות – select לא מוחק כפילויות; distinct מוחק. like – % כל מחרוזת, _ תו אחד. (עמ' 56–59)	SELECT DISTINCT ... WHERE content LIKE '%בריאות%'
פונקציות הקבצה – count, sum, avg, min, max; מתעלמות מ-null (חוץ מ-count(*)). count(distinct x) – ערכים שונים. בדיקת null: is null. (עמ' 61–62)	COUNT(DISTINCT d.id) > 50 AVG(capacity)
פעולות על קבוצות – union, intersect, except (מוחקות כפילויות; עם all – לא). (עמ' 60)	(SELECT ...) EXCEPT (SELECT ...)

2.2 שאילתות מקוננות

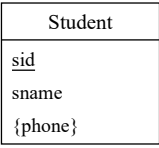
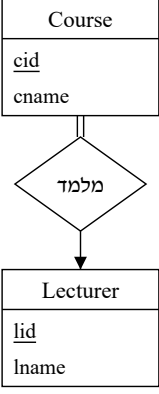
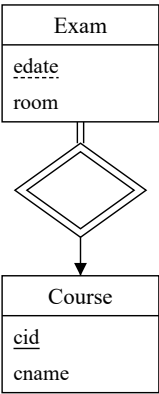
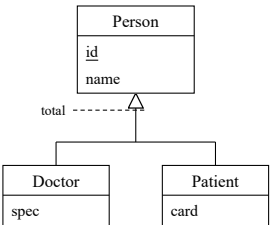
ניסוח המשפט	כתיב מתמטי
ערך מול קבוצה – in, not in, some, >, all, >=. תת-שאילתה סקלרית – מחזירה ערך אחד, ואפשר להשוות אליו. (עמ' 66–68)	WHERE t.price >= ALL (SELECT t2.price FROM trade t2 WHERE t2.fid = t.fid)
exists / not exists – האם תת-השאילתה מחזירה שורה. "לכל": אין אף אחד ש... אין לו – שני not exists. (עמ' 69–70)	WHERE NOT EXISTS (SELECT * FROM follow f WHERE f.fuid = u.uid AND NOT EXISTS (SELECT * FROM likes l WHERE ...))
with / תת-שאילתה ב-from – יחס זמני עם שם, לחישוב בשלבים (למשל סכום לכל מדינה ואחר כך המקסימום). (עמ' 73–74)	WITH p AS (SELECT cid, SUM(q * pr) AS total FROM ... GROUP BY cid) SELECT ... FROM p WHERE total = (SELECT MAX(total) FROM p);

2.3 הדקים (טריגרים) ב-PostgreSQL

ניסוח המשפט	כתיב מתמטי
הדק – אירוע (insert / update / delete על יחס), תנאי ופעולה. new – השורה החדשה (insert, update); old – השורה הקודמת (delete, update). (עמ' 132–133)	create trigger name before after instead of insert update delete on table for each row execute procedure f();
הפונקציה – מחזירה trigger, ולכן חייבת להחזיר new, old או null. בהדק before: החזרת null מבטלת את הפעולה לשורה הזו בלבד, החזרת new מבצעת אותה. (עמ' 134–135)	create function f() returns trigger as \$\$ begin if ... then raise notice 'הודעה'; return null; end if; return new; end; \$\$ language plpgsql;
הודעות – raise notice מוציא הודעה בלי לעצור; raise exception מוציא הודעת שגיאה ועוצר את כל הביצוע. TG_OP – איזו פעולה הפעילה את ההדק. (עמ' 135–136)	+ raise notice ← "רשומה זו בלבד לא תתווסף" return null

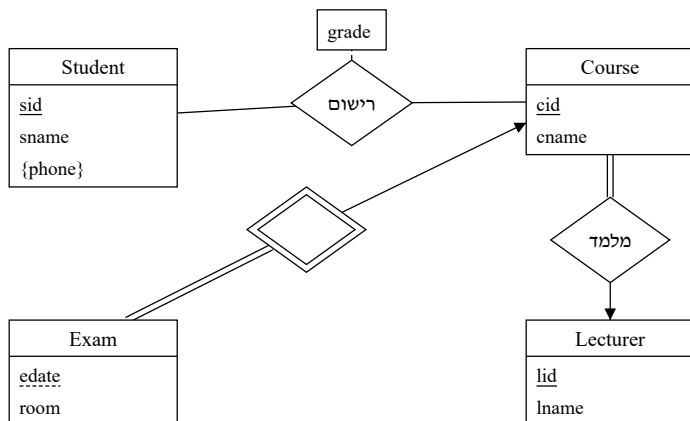
3 מודל ישויות-קשרים

3.1 סימונים

כתיב מתמטי	שרטוט	ניסוח המשפט
		טיפוס ישויות – מלבן עם התכונות; המפתח הראשי מסומן בקו תחתון. טיפוס קשרים – מעוין; תכונות של הקשר – במלבן מחובר בקו מקווקו. (עמ' 139–154) תכונה מרובת ערכים – בסוגריים מסולסלים { }. (עמ' 145)
	 "לכל קורס בדיוק מרצה אחד"	מידות ריבוי – רבים-לרבים, אחד-לרבים, רבים-לאחד, אחד-לאחד. חץ מהמעוין אל ישות שמשתתפת כ"אחד" (לכל היותר אחת). בקשר n-ארי: בודקים כל צירוף של $n - 1$ ישויות מול הנותרת. (עמ' 150–152) השתתפות מלאה – כל ישות חייבת להשתתף: קו כפול. אחרת – חלקית, קו יחיד. (עמ' 152–153)
		טיפוס ישויות חלש – אין לו מפתח משלו; המפתח שלו הוא המפתח של הישות החזקה ועוד מזהה פנימי (קו תחתון מרוסק). מחובר לישות החזקה בקשר מזהה – מעוין כפול – עם קו כפול ממנו וחץ אל החזקה. (עמ' 155–157)
$\Rightarrow$ רבים-לרבים E_1-E_2 קשר (K_1, K_2) מפתח		מפתחות – מפתח-על: מבחין בין כל שתי ישויות; מפתח קביל: מפתח-על בלי תכונות מיותרות; מפתח ראשי: אחד הקבילים שנבחר. מפתח של קשר רבים-לרבים: צירוף המפתחות של המשתתפים. (עמ' 153–154)
	 הכללה: $E_1 \cup \dots \cup E_k = E$ פירוט: $E_1 \cup \dots \cup E_k \subseteq E$	ISA (פירוט / הכללה) – חץ עם ראש חלול מהישויות ברמה הנמוכה אל הרמה הגבוהה; הנמוכות יורשות את התכונות. זרות – קו אחד שמחבר את הישויות היורשות וחץ אחד; חופפות – חץ נפרד מכל אחת. אילוץ שלמות מלא (כל ישות אב היא לפחות אחת מהיורשות) – קו מקווקו עם המילה total. קיבוץ – קשר עם הישויות שלו כישות אחת. (עמ' 165–169)

3.2 דוגמה – דיאגרמה והיחסים שמתקבלים ממנה

לכל יחס – הכלל שיצר אותו מתוך כללי ההמרה א-1 בסעיף 3.4 (מדריך הלמידה, עמ' 158–162).



יחס	כלל ההמרה (3.4, עמ' 158–162)
Student (<u>sid</u> , sname)	כלל א – ישות חזקה
Student_phone (<u>sid</u> , <u>phone</u>)	כלל ג – תכונה מרובת ערכים {phone}
Course (<u>cid</u> , cname, lid)	כלל ו – הקשר מלמד (אחד-לרבים) נכנס ל-Course
Lecturer (<u>lid</u> , lname)	כלל א – ישות חזקה
Exam (<u>cid</u> , <u>edate</u> , room)	כלל ב – ישות חלשה של Course
(<u>sid</u> , <u>cid</u> , grade)	כלל ה – הקשר רשום (רבים-לרבים)

3.3 מיחסים לדיאגרמה – טבלת עזר

אלגוריתם לשרטוט דיאגרמת ישויות-קשרים מתוך היחסים:

עבור כל יחס בשאלון:

1. בדוק אם יש בו **מפתח זר** – שדה שמזהה רשומה ביחס אחר:
 - אם כן – יש **קשר** בין היחס לבין היחס שהמפתח מפנה אליו.
 - 2. בדוק את המפתח הראשי של היחס:
 - אם יש לו מזהה **משלו** – הוא **ישות**.
 - אם המפתח הוא **צירוף מזהים של ישויות אחרות** – הוא **קשר** (רבים-לרבים), ושאר התכונות הן תכונות הקשר.
 - אם המפתח הוא **מזהה חלקי משלו ועוד מזהה של ישות אחרת** – הוא **ישות חלשה**, עם מזהה פנימי וקשר מזהה (מעוין כפול).
 - אם כמה יחסים חולקים מזהה ונבדלים בתכונות – **ISA**.
3. הוסף את האילוצים מהטקסט: "לכל ... יש ..." – קו כפול; "לכל היותר אחד" – חץ.
4. **בדיקה**: המר את הדיאגרמה בחזרה ליחסים (3.4) – צריך לקבל בדיוק את היחסים שבשאלון, כולל המפתחות.

לדוגמה נתון:

Request(rid, id, sid, ...) Shelter(id, ...) Supplies(sid, ...)

נפעיל את האלגוריתם כדלקמן:

sid מפתח זר ב-Request $\Rightarrow$ קשר בין Request ל-Supplies

המפתח (rid, id), ו-id מזהה של Request $\Rightarrow$ Shelter ישות חלשה של Shelter

נקבל אפוא את המבוקש:

Request, Shelter, Supplies – ישויות; Request – ישות חלשה (מזהה פנימי rid)

ניסוח המשפט	כתיב מתמטי
בדיקה – ממירים את הדיאגרמה בחזרה ליחסים (3.4), וצריך לקבל בדיוק את היחסים שבשאלון, כולל המפתחות.	היחסים בשאלון $\xrightarrow{3.4}$ דיאגרמה

3.4 מדיאגרמה לתבניות יחסים

ניסוח המשפט	כתיב מתמטי
כלל א – ישות חזקה $\leftarrow$ יחס עם כל התכונות שלה. כלל ב – ישות חלשה $\leftarrow$ יחס עם התכונות שלה ועם המפתח של הישות החזקה. (עמ' 158–159)	$E_1(A_1, A_2)$ $E_2(A_1, B_1, B_2)$
כלל ג – תכונה מרובת ערכים $\leftarrow$ יחס נפרד: המפתח של הישות ועמודה לתכונה (שניהם המפתח). כלל ד – תכונה מורכבת $\leftarrow$ עמודה לכל מרכיב. (עמ' 159–160)	$E(K, \dots)$ $E_MV(K, MV)$

ניסוח המשפט	כתיב מתמטי
כלל ה – קשר רבים-לרבים $\leftarrow$ יחס נפרד : המפתחות של המשתתפים ותכונות הקשר. כלל ו – קשר אחד-לרבים : אפשר בלי יחס נפרד – מוסיפים את המפתח של הצד "אחד" ואת תכונות הקשר ליחס של הצד "רבים". קשר מזהה (מעוין כפול) – אין לו יחס. (עמ' 160–162)	$R(\underline{A_1}, \underline{B_1}, C)$ $E_2(\underline{B_1}, \underline{B_2}, A_1, C)$
ISA לתבניות – (1) שתי הרמות : יחס לכל טיפוס, והנמוכים מקבלים את המפתח של הגבוה (הדרך הכללית); (2) רק הרמה הנמוכה : כל התכונות בכל יחס נמוך (כשהפירוט מלא וזר); (3) רק הרמה הגבוהה : עם תכונה שממיינת לסוגים. (עמ' 169–170)	$\text{doctor}(\underline{\text{id}}, \text{name}, \dots, \text{spec})$

4
 תלויות פונקציונליות ותיכון

דוגמה אחת לכל הפרק: כל האלגוריתמים בפרק מופעלים על אותם נתונים, וכל אחד ממשיך מהתוצאה של הקודם:

$$R = (A, B, C, D, E, G), \quad F = \{AG \rightarrow BD, DE \rightarrow AC, B \rightarrow D, B \rightarrow E, G \rightarrow D\}$$

4.1
 תלויות וסגור

ניסוח המשפט	כתיב מתמטי
תלות פונקציונלית $X \rightarrow Y$ – אין שתי שורות שזהות ב-X ושונות ב-Y. טריוויאלית – כש-$Y \subseteq X$. (עמ' 174–179) כללי היסק – רפלקסיביות, הכפלה, טרנזיטיביות (ארמסטרונג), ונוספים : איחוד ופירוק. (עמ' 181)	$X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$ $X \rightarrow Y, X \rightarrow Z \Leftrightarrow X \rightarrow YZ$
סגור של קבוצת תכונות X^+ – הקבוצה הגדולה ביותר של תכונות שתלויות ב-X. בדיקה אם $X \rightarrow Y$ נובעת מ- F : אם $Y \subseteq X^+$. (עמ' 181–185)	$X \rightarrow Y \in F^+ \iff Y \subseteq X^+$

אלגוריתם לחישוב הסגור X^+ תחת F (עמ' 183):

- אתחל $X^+ := X$.
- עבור על התלויות ב- F : לכל תלות $Y \rightarrow Z$ שבה $Y \subseteq X^+$ אך $Z \not\subseteq X^+$ – הוסף את Z ל- X^+ .
- חזור על צעד 2 כל עוד נוספות תכונות.
- X^+ הוא התוצאה.

לדוגמה נתון (חישוב B^+):

$$R = (A, B, C, D, E, G), \quad F = \{AG \rightarrow BD, DE \rightarrow AC, B \rightarrow D, B \rightarrow E, G \rightarrow D\}$$

נפעיל את האלגוריתם כדלקמן:

$$B \xrightarrow{B \rightarrow D} BD \xrightarrow{B \rightarrow E} BDE \xrightarrow{DE \rightarrow AC} ABCDE$$

$AG \rightarrow D$ ו- $G \rightarrow D$ לא מוסיפות דבר: $G \notin ABCDE$ – וסיימנו

נקבל אפוא את המבוקש:

$$B^+ = ABCDE$$

4.2
 כיסוי קנוני

ניסוח המשפט	כתיב מתמטי
כיסוי קנוני $F_C^+ = F^+ - F_C$, אף תלות לא מכילה תכונה עודפת, ואין שתי תלויות עם אותו צד שמאל. (עמ' 186) תכונה עודפת בצד ימין – ב-$X \rightarrow Y$ מורידים אותה, ובודקים אם $B \in X^+$ (לפי שאר התלויות). בצד שמאל – A: אם $Y \subseteq (X - A)^+$. (עמ' 186–187)	$X \rightarrow Y, B \in Y$ $B \in X_{F'}^+ \Rightarrow B \text{ עודפת}$

אלגוריתם לחישוב כיסוי קנוני F_C (עמ' 188):

1. אתחל $F_C := F$.

2. אחד תלויות עם אותו צד שמאל: $X \rightarrow Y_1, X \rightarrow Y_2 \Rightarrow X \rightarrow Y_1 Y_2$.

3. בדוק אם יש ב- F_C תלות עם תכונה עודפת:

• **בצד ימין** – $B \rightarrow Y$ ב- $X \rightarrow Y$: החלף את התלות ב- $X \rightarrow (Y - B)$ וחשב X^+ ; אם $B - B \in X^+$ עודפת.

• **בצד שמאל** – $A \rightarrow Y$ ב- $X \rightarrow Y$: חשב $(X - A)^+$ לפי F_C ; אם $A - Y \subseteq (X - A)^+$ עודפת.

• אם נמצאה – מחק אותה מהתלות.

4. חזור על צעדים 2–3 כל עוד יש שינוי.

לדוגמה נתון:

$$F = \{AG \rightarrow BD, DE \rightarrow AC, B \rightarrow D, B \rightarrow E, G \rightarrow D\}$$

נפעיל את האלגוריתם כדלקמן:

$$B \rightarrow D, B \rightarrow E \Rightarrow B \rightarrow DE$$

$$D \text{ ב-} AG \rightarrow BD: \text{נחליף ב-} AG \rightarrow B, \text{ואז } AG^+ = ABCDEG \ni D$$

$$\text{שאר צד ימין: בלי } A - A \text{ ב-} DE^+ = CDE \not\subseteq A - A; \text{בלי } C - C \text{ ב-} DE^+ = ADE \not\subseteq C - C; \text{בלי } BE - D \text{ ב-} B^+ = BE - D; \text{בלי } BD - E \text{ ב-} B^+ = BD - E$$

$$\text{צד שמאל: } D^+ = D, E^+ = E \not\subseteq AC; A^+ = A, G^+ = DG \not\subseteq B$$

נקבל אפוא את המבוקש:

$$F_C = \{AG \rightarrow B, DE \rightarrow AC, B \rightarrow DE, G \rightarrow D\}$$

4.3 מפתחות קבילים – בשלושה שלבים

אלגוריתם למציאת כל המפתחות הקבילים של R (עמ' 184–185):

1. מצא את התכונות שלא מופיעות באף צד ימין של תלות – הן חלק מכל מפתח.

2. מצא את התכונות שמופיעות בצד ימין ולא באף צד שמאל – הן לא חלק משום מפתח.

3. חשב סגורים: התכונות משלב 1, ועוד 0, 1, 2, ... מהתכונות שנשארו (מהקבוצות הקטנות לגדולות):

• אם הסגור הוא R – הקבוצה **מפתח קביל**.

• אל תבדוק קבוצה שמכילה מפתח שכבר נמצא.

לדוגמה נתון:

$$R = (A, B, C, D, E, G), \quad F_C = \{AG \rightarrow B, DE \rightarrow AC, B \rightarrow DE, G \rightarrow D\}$$

נפעיל את האלגוריתם כדלקמן:

$$\text{שלב 1: } G \text{ (בכל מפתח)} \cdot \text{שלב 2: } C \text{ (באף מפתח)} \cdot \text{שלב 3: } DG - G^+ = DG \text{ לא מפתח}$$

$$AG^+ = ABCDEG = R \quad BG^+ = ABCDEG = R \quad DG^+ = DG \quad EG^+ = ABCDEG = R$$

כל קבוצה של 3 תכונות עם G מכילה את AG, BG או EG – וסיימו

נקבל אפוא את המבוקש:

$$\text{המפתחות הקבילים: } AG, BG, EG$$

4.4 פירוקים

אלגוריתם למציאת F_i מתוך F_C עבור יחס R_i :

עבור כל תלות $X \rightarrow Y$ ב- F_C :

1. חשב את החיתוך של $X \cup Y$ עם R_i .

2. אם $X \subseteq R_i$:

• קבע את X (הצד השמאלי שנמצא ב- R_i).

• קבע את $Y' = ((X^+ \cap R_i) \setminus X)$ (כל התכונות שנקבעות על-ידי X , שנמצאות גם ב- R_i , אך לא שייכות ל- X עצמו).

• אם $Y' \neq \emptyset$, הוסף את התלות $X \rightarrow Y'$ ל- F_i .

3. איחוד תלויות עם אותו צד שמאלי אם נדרש (לדוגמה, אם קיימות $X \rightarrow Y_1, X \rightarrow Y_2$, יש לאחד אותן ל- $X \rightarrow (Y_1 \cup Y_2)$).

לדוגמה נתון:

$$R_1 = (A, B, D, G), \quad R_2 = (B, C, E), \quad F_C = \{AG \rightarrow B, DE \rightarrow AC, B \rightarrow DE, G \rightarrow D\}$$

נפעיל את האלגוריתם כדלקמן (R_1):

$$AG^+ = ABCDEG \xrightarrow{\cap R_1} AG \rightarrow BD \quad B^+ = ABCDE \xrightarrow{\cap R_1} B \rightarrow AD \quad G^+ = DG \xrightarrow{\cap R_1} G \rightarrow D$$

$$B^+ \cap R_2 = BCE \Rightarrow B \rightarrow CE, \quad B \subseteq R_2 \text{ רק } R_2 \text{ וב-} DE \not\subseteq R_1$$

נקבל אפוא את המבוקש:

$$F_1 = \{AG \rightarrow BD, B \rightarrow AD, G \rightarrow D\} \quad F_2 = \{B \rightarrow CE\}$$

אלגוריתם לבדיקת שימור מידע בפירוק ל- R_1, R_2 (עמ' 198):

1. חשב את החיתוך $R_1 \cap R_2$.

2. חשב את הסגור $(R_1 \cap R_2)^+$.

3. אם הסגור מכיל את R_1 או את R_2 (החיתוך מפתח של אחד מהם) – הפירוק **משמר מידע**; אחרת – לא.

• הבדיקה מתאימה לפירוק לשתי תבניות בלבד.

לדוגמה נתון:

$$R_1 = (A, B, D, G), \quad R_2 = (B, C, E), \quad F_C = \{AG \rightarrow B, DE \rightarrow AC, B \rightarrow DE, G \rightarrow D\}$$

נפעיל את האלגוריתם כדלקמן:

$$R_1 \cap R_2 = B \quad B^+ = ABCDE \supseteq R_2 = BCE$$

נקבל אפוא את המבוקש:

$$R_1 \cap R_2 \rightarrow R_2 \Rightarrow \text{הפירוק משמר מידע}$$

אלגוריתם לבדיקת שימור תלויות (עמ' 198–200):

1. חשב את F_i לכל יחס R_i (באלגוריתם למציאת F_i).

2. עבור כל תלות $X \rightarrow Y$ ב- F_C :

• אם כל התכונות שלה נמצאות ביחס אחד – היא נשמרת.

• אחרת חשב את X_P^+ – הסגור של X לפי $F_1 \cup \dots \cup F_k$.

• אם $Y \subseteq X_P^+$ – היא נשמרת; אחרת היא **אבדה**.

3. הפירוק משמר תלויות אם ורק אם כל התלויות נשמרות: $(F_1 \cup \dots \cup F_k)^+ = F^+$.

לדוגמה נתון:

$$F_1 = \{AG \rightarrow BD, B \rightarrow AD, G \rightarrow D\}, \quad F_2 = \{B \rightarrow CE\}, \quad F_C = \{AG \rightarrow B, DE \rightarrow AC, B \rightarrow DE, G \rightarrow D\}$$

נפעיל את האלגוריתם כדלקמן:

$$B_P^+ = ABCDE \supseteq DE : B \rightarrow DE \text{ נשמרת, } R_1 \text{ בתוך } G \rightarrow D \text{ ו-} AG \rightarrow B$$

$$DE \rightarrow AC : \quad DE_P^+ = DE \not\supseteq AC$$

נקבל אפוא את המבוקש:

$$DE \rightarrow AC \text{ אבדה } \Rightarrow \text{הפירוק אינו משמר תלויות}$$

4.5 צורות נורמליות

ניסוח המשפט	כתיב מתמטי
BCNF – לכל תלות לא טריוויאלית $X \rightarrow Y$ מפתח-על של R. מספיק לבדוק את התלויות של כיסוי קנוני. (עמ' 201–203) 3NF – לכל תלות: טריוויאלית, או X מפתח, או כל תכונה ב-$X - Y$ מוכלת במפתח קביל כלשהו. (עמ' 206) 1NF – כל ערך אטומי. יחס שאינו ב-BCNF ואינו ב-3NF – נכתוב שהוא ב-1NF. (עמ' 174)	$BCNF \subset 3NF \subset 1NF$ $B : B \rightarrow DE$ אינו מפתח $\leftarrow$ לא BCNF D באף מפתח $\leftarrow$ לא 3NF $R \leftarrow$ ב-1NF
הצורה של יחס – לפי הגרועה מבין התלויות. 2NF – אין תכונה לא ראשית שתלויה בחלק ממפתח; חלשה מ-3NF, ולפי המדריך בלי חשיבות מעשית. (עמ' 210)	$level(R) = \min \{level(\tau) \mid \tau \in F(R)\}$

אלגוריתם לפירוק ל-BCNF (עמ' 204–206):

- מצא תלות $X \rightarrow Y$ שחלה על התבנית ומפרה את BCNF (X אינו מפתח-על של התבנית).
- פרק את התבנית ל- $R_1 = (X, Y)$ ול- $R_2 = R - Y$.
 - הפירוק משמר מידע – X משותפת לשתייהן, והיא מפתח של R_1 .
- חזור על צעדים 1–2 לכל תבנית שעדיין יש בה תלות שמפרה BCNF.
 - התוצאה משמרת מידע, אבל לא תמיד משמרת תלויות.

לדוגמה נתון:

$$AG, BG, EG : \text{המפתחות}, F_C = \{AG \rightarrow B, DE \rightarrow AC, B \rightarrow DE, G \rightarrow D\}, R = (A, B, C, D, E, G)$$

נפעיל את האלגוריתם כדלקמן:

$$\Rightarrow (A, C, D, E), (B, D, E, G) \quad (DE^+ = ACDE \neq R) \text{ מפרה } DE \rightarrow AC$$

$$\Rightarrow (B, D, E), (B, G) \text{ מפרה } B \rightarrow DE; BG \text{ המפתח}, F = \{B \rightarrow DE, G \rightarrow D\} : (B, D, E, G) \text{ ב-}$$

$$\text{ב-}(A, C, D, E) \text{ המפתח } DE, \text{ב-}(B, D, E) \text{ המפתח } B, \text{ו-}(B, G) \text{ בלי תלויות – אין עוד הפרות}$$

נקבל אפוא את המבוקש:

$$(A, C, D, E), (B, D, E), (B, G) \Rightarrow BCNF$$

והתלות $G \rightarrow D$ אבדה – פירוק ל-BCNF לא תמיד משמר תלויות

אלגוריתם לפירוק ל-3NF (עמ' 208):

- חשב כיסוי קנוני F_C .
- עבור כל תלות $X \rightarrow Y$ ב- F_C :
 - צור תבנית עם התכונות $X \cup Y$, אלא אם כבר יש תבנית שמכילה אותן.
- אם אף תבנית לא מכילה מפתח קביל של R – הוסף תבנית עם התכונות של מפתח קביל.
 - התוצאה משמרת מידע ומשמרת תלויות, וכל התבניות לפחות ב-3NF.

לדוגמה נתון:

$$AG, BG, EG : \text{המפתחות}, F_C = \{AG \rightarrow B, DE \rightarrow AC, B \rightarrow DE, G \rightarrow D\}, R = (A, B, C, D, E, G)$$

נפעיל את האלגוריתם כדלקמן:

$$AG \rightarrow B \Rightarrow (A, B, G) \quad DE \rightarrow AC \Rightarrow (A, C, D, E) \quad B \rightarrow DE \Rightarrow (B, D, E) \quad G \rightarrow D \Rightarrow (D, G)$$

$$\text{המפתח } AG \subseteq (A, B, G) \text{ אין צורך בתבנית מפתח}$$

נקבל אפוא את המבוקש:

$$(A, B, G), (A, C, D, E), (B, D, E), (D, G)$$

5 טיפוסים נתונים מורכבים

5.1 דירוג מסמכים

ניסוח המשפט	כתיב מתמטי
TF – שכיחות מונח במסמך: $n(d, t)$ מספר ההופעות של t ב-d, $n(d)$ מספר המילים ב-d. IDF – $n(t)$ מספר המסמכים שבהם t מופיע. דירוג – סכום על מונחי השאלתה. משאירים ספרות עד הסוף ומעגלים רק את התוצאה. (עמ' 216–219)	$TF(d, t) = \log_{10} \left(1 + \frac{n(d, t)}{n(d)} \right)$ $IDF(t) = \frac{1}{n(t)}$ $r(d, Q) = \sum_{t \in Q} TF(d, t) \cdot IDF(t)$

אלגוריתם לחישוב הדירוג $r(d, Q)$ של מסמך (עמ' 216–219):

1. ספור את המילים במסמך – $n(d)$, וכמה פעמים מופיע כל מונח של השאלתה – $n(d, t)$.

2. חשב $TF(d, t) = \log_{10} \left(1 + \frac{n(d, t)}{n(d)} \right)$ לכל מונח.

3. חשב $IDF(t) = \frac{1}{n(t)}$ לכל מונח $n(t)$ בכמה מסמכים הוא מופיע.

4. סכום: $r(d, Q) = \sum_{t \in Q} TF(d, t) \cdot IDF(t)$.

• שמור ספרות עד הסוף, ועגל רק את התוצאה.

לדוגמה נתון:

$$d = \text{canada export apple train orange train}, \quad Q = \text{train apple}, \quad n(\text{apple}) = 2, \quad n(\text{train}) = 3$$

נפעיל את האלגוריתם כדלקמן:

$$n(d) = 6 \quad TF(\text{apple}) = \log_{10} \left(1 + \frac{1}{6} \right) = 0.0669 \quad TF(\text{train}) = \log_{10} \left(1 + \frac{2}{6} \right) = 0.1249$$

$$r(d, Q) = 0.0669 \cdot \frac{1}{2} + 0.1249 \cdot \frac{1}{3} = 0.0751$$

נקבל אפוא את המבוקש:

$$r(d, Q) \approx 0.075$$

ניסוח המשפט	כתיב מתמטי
PageRank – N מספר הדפים; $T[i, j] = \frac{1}{k_i}$ אם i מצביע ל-j כש-k_i מספר הקישורים שיוצאים מ-i (אחרת 0); δ פרמטר (למשל 0.15). מתחילים מ-$\frac{1}{N}$ לכל דף, ובכל איטרציה מציבים רק את הערכים של האיטרציה הקודמת. (עמ' 220–222)	$P[j] = \frac{\delta}{N} + (1 - \delta) \sum_{i=1}^N T[i, j] P[i]$

אלגוריתם לחישוב PageRank (עמ' 220–222):

1. בנה את T : לכל דף i עם k_i קישורים יוצאים – $T[i, j] = \frac{1}{k_i}$ לכל דף j שהוא מצביע אליו.

2. אתחל $P[j] = \frac{1}{N}$ לכל דף.

3. בכל איטרציה חשב לכל דף $P[j] = \frac{\delta}{N} + (1 - \delta) \sum_i T[i, j] \cdot P[i]$.

• הצב רק את הערכים של האיטרציה הקודמת – גם אם ערך חדש כבר חושב באותה איטרציה.

4. חזור על צעד 3 עד שאין שינוי (או לפי מספר האיטרציות שנדרש).

לדוגמה נתון:

$$A \rightarrow B, C, D \quad B \rightarrow C \quad C \rightarrow A, D \quad D \rightarrow A \quad \delta = 0.1, N = 4$$

נפעיל את האלגוריתם כדלקמן (איטרציה 1, מ- $\frac{1}{4}$):

$$P[A] = 0.025 + 0.9 \left(\frac{0.25}{2} + \frac{0.25}{1} \right) = 0.3625 \quad P[B] = 0.025 + 0.9 \cdot \frac{0.25}{3} = 0.1$$

$$P[C] = 0.025 + 0.9 \left(\frac{0.25}{3} + \frac{0.25}{1} \right) = 0.325 \quad P[D] = 0.025 + 0.9 \left(\frac{0.25}{3} + \frac{0.25}{2} \right) = 0.2125$$

נקבל אפוא את המבוקש:

$$P = (0.3625, 0.1, 0.325, 0.2125)$$

5.2 שפות לייצוג מידע

ניסוח המשפט	כתיב מתמטי
JSON – אובייקט: זוגות שם: ערך בתוך { }; מערך: [.] . ערך יכול להיות אטומי, אובייקט (קינון) או מערך. שמות התכונות במירכאות. (עמ' 229–230) XML – תגים שהשתמש מגדיר: <tag> ... </tag> ; לכל תג פותח – תג סוגר, וקינון לפי המבנה. (עמ' 230–232)	<pre>{ "uid": 1, "posts": [{ ... }, { ... }] }</pre> <pre><user> <posts>...</posts> </user></pre>
RDF – אוסף שלשות <נושא, תכונה, ערך>: מזהה של ישות, שם של תכונה או של קשר, וערך או מזהה של ישות אחרת. אפשר לצייר אותו כגרף: ישות – צומת, שלשה – קשת מהנושא לערך. (עמ' 232–234) SPARQL – שלשות עם משתנים (?x); התוצאה: כל ההשמות למשתנים שהופכות את כל השלשות לשלשות שבנתונים. (עמ' 234–235)	<pre>u1 name "alice" .</pre> <pre>?u name ?n .</pre>

דוגמה – אותם נתונים בשלוש השפות: המשתמשת alice (מזהה 1, מישראל) ושני הפוסטים שלה. ב-XML וב-JSON הפוסטים מקוננים בתוך המשתמשת; ב-RDF כל עובדה היא שלשה, והפוסט מצביע על הכותבת.

JSON

```
{
  "users": [
    {
      "uid": 1,
      "name": "alice",
      "country": "israel",
      "posts": [
        { "pid": 101, "content": "sunset" },
        { "pid": 105, "content": "coffee" }
      ]
    }
  ]
}
```

XML

```
<users>
  <user>
    <uid>1</uid>
    <name>alice</name>
    <country>israel</country>
    <posts>
      <post>
        <pid>101</pid>
        <content>sunset</content>
      </post>
      <post>
        <pid>105</pid>
        <content>coffee</content>
      </post>
    </posts>
  </user>
</users>
```

RDF – גרף

```
graph TD
    u1((u1)) -- name --> alice[alice]
    u1 -- country --> israel[israel]
    u1 -- author --> p101((p101))
    u1 -- author --> p105((p105))
    p101 -- content --> sunset[sunset]
    p105 -- content --> coffee[coffee]
```

RDF – שלשות

```
u1 instanceOf user .
u1 name "alice" .
u1 country "israel" .
p101 instanceOf post .
p101 author u1 .
p101 content "sunset" .
p105 instanceOf post .
p105 author u1 .
p105 content "coffee" .
u7 instanceOf user .
u7 name "grace" .
u7 country "uk" .
```

אלגוריתם לכתיבת שאילתת SPARQL ולמציאת התוצאה שלה (עמ' 234–235):

1. לכל ישות שהשאלה עוסקת בה – משתנה ($?u$, $?p$); לכל ערך שמבקשים להציג – משתנה, ואותו כותבים ב-select.
 2. לכל תנאי ולכל קשר – שלשה אחת נושא תכונה ערך:
 - ערך קבוע (למשל "israel") – תנאי בחירה.
 - משתנה שמופיע בשתי שלשות – מחבר אותן, כמו צירוף.
 3. התוצאה: עוברים על הנתונים ומחפשים את כל ההשמות שבהן כל השלשות נמצאות בנתונים:
 - מתחילים מהשלשה עם הערך הקבוע – היא מצמצמת הכי הרבה.
 - כל השמה כזו היא שורה בתוצאה, עם משתני ה-select בלבד.
- לדוגמה נתון (השלשות שלמעלה): השמות של המשתמשים מישראל, והתוכן של כל פוסט שלהם. נפעיל את האלגוריתם כדלקמן:

```
select ?n ?c where {
  ?u country "israel" .
  ?u name ?n .
  ?p author ?u .
  ?p content ?c
}
```

השלשה עם הערך הקבוע: $?u = u1$ (אצל $u7$ המדינה היא uk)

מכאן $?n = \text{alice}$; הפוסטים ש- $u1$ כתבה: $?p = p101$ או $?p = p105$; והתוכן שלהם: sunset ו-coffee

נקבל אפוא את המבוקש:

?n	?c
alice	sunset
alice	coffee

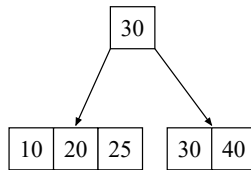
6 אינדקסים

6.1 עץ B+

ניסוח המשפט	שרטוט	כתיב מתמטי
הצמתים – בכל צומת $\lceil n/2 \rceil$ עד n מצביעים, כלומר $\lceil n/2 \rceil - 1$ עד $n - 1$ ערכים; במדריך אותו מינימום גם לעלים. רק בשורש מותר פחות. העץ מאוזן, וכל הערכים מופיעים בעלים. (עמ' 251–252)		$n = 4$: 2–4 מצביעים, 1–3 ערכים $n = 5$: 3–5 מצביעים, 2–4 ערכים
ניווט – בצומת פנימי, המצביע שמשמאל ל-V מוביל לערכים קטנים מ-V, ושמימין – לערכים גדולים או שווים. (עמ' 253)		

אלגוריתם להכנסת ערך לעץ B^+ (עמ' 257–262):

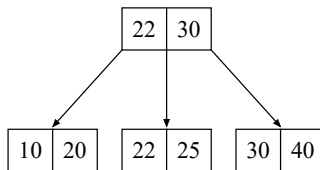
- מצא את העלה המתאים: בכל צומת פנימי המושך במצביע שמשמאל לערך הראשון שגדול מהערך החדש.
 - אם יש בעלה מקום (פחות מ- $n-1$ ערכים) – הכנס את הערך במקומו הממוין, וסיים.
 - אחרת **פצל את העלה**: יש בו עכשיו n ערכים:
 - חלק אותם לשני עלים (כש- n אי-זוגי: $\lceil n/2 \rceil$ ערכים לצד אחד).
 - העתק** להורה את הערך הקטן ביותר בעלה הימני (הוא נשאר גם בעלה).
 - אם ההורה התמלא מעבר ל- $n-1$ ערכים – **פצל צומת פנימי**:
 - הערך האמצעי **עובר** להורה ולא נשאר למטה.
 - $\lfloor (n-1)/2 \rfloor$ הערכים הקטנים לצומת השמאלי, $\lceil (n-1)/2 \rceil$ הגדולים לימני.
 - אם השורש התפצל – צור שורש חדש עם ערך אחד; העץ גדל ברמה.
- לדוגמה נתון ($n=4$, הוספת 22):



נפעיל את האלגוריתם כדלקמן:

$\Rightarrow 22 < 30 \Rightarrow$ העלה [10 20 25] – מלא [10 20 22 25]
פיצול: [10 20], [22 25], ו-22 מועתק להורה

נקבל אפוא את המבוקש:



אלגוריתם למציאת מספר העלים והרמות המקסימלי בעץ B^+ עם x ערכים:

- בכל צומת חוץ מהשורש יש לפחות $\lceil n/2 \rceil$ מצביעים, ולכן לפחות $\lceil n/2 \rceil - 1$ ערכים – גם בעלים (עמ' 251–252).
- מספר העלים המקסימלי** – בכל עלה המינימום: $L_{max} = \left\lfloor \frac{x}{\lceil n/2 \rceil - 1} \right\rfloor$
- מספר הרמות המקסימלי** – עלה מהעלים רמה אחרי רמה; ברמה עם I צמתים:
 - אם $I=1$ – זה השורש, וסיים.
 - אם $2 \leq I < \lceil n/2 \rceil$ – מעליהם רק השורש, וסיים.
 - אחרת – ברמה שמעל לכל היותר $\left\lfloor \frac{I}{\lceil n/2 \rceil} \right\rfloor$ צמתים.
- מספר הרמות הוא מספר הרמות שספרת, מהעלים ועד השורש.

לדוגמה נתון:

$$x = 600,000, \quad n = 151 \Rightarrow \lceil n/2 \rceil = 76$$

נפעיל את האלגוריתם כדלקמן:

$$L_{max} = \left\lfloor \frac{600,000}{75} \right\rfloor = 8,000 \quad \left\lfloor \frac{8,000}{76} \right\rfloor = 105 \quad \left\lfloor \frac{105}{76} \right\rfloor = 1$$

נקבל אפוא את המבוקש:

לכל היותר 8,000 עלים; עלים $\rightarrow 105 \rightarrow$ שורש – 3 רמות

אלגוריתם למציאת מספר העלים והרמות המינימלי בעץ B^+ עם x ערכים:

1. בכל צומת לכל היותר n מצביעים, ולכן לכל היותר $n - 1$ ערכים (עמ' 251–252).

2. מספר העלים המינימלי – כל עלה מלא: $L_{min} = \lceil \frac{x}{n-1} \rceil$.

3. מספר הרמות המינימלי – עלה מהעלים רמה אחרי רמה; ברמה עם I צמתים:

• אם $I = 1$ – זה השורש, וסיים.

• אחרת – לכל צומת ברמה שמעל עד n בנים, ולכן ברמה שמעל לפחות $\lceil \frac{I}{n} \rceil$ צמתים.

4. מספר הרמות הוא מספר הרמות שספרת, מהעלים ועד השורש.

לדוגמה נתון (אותו עץ):

$$x = 600,000, \quad n = 151 \Rightarrow n - 1 = 150$$

נפעיל את האלגוריתם כדלקמן:

$$L_{min} = \lceil \frac{600,000}{150} \rceil = 4,000 \quad \lceil \frac{4,000}{151} \rceil = 27 \quad \lceil \frac{27}{151} \rceil = 1$$

נקבל אפוא את המבוקש:

לפחות 4,000 עלים; עלים $\rightarrow 27 \rightarrow$ שורש – 3 רמות

ניסוח המשפט	כתיב מתמטי
הכי הרבה ערכים בעץ עם l רמות – כל צומת מלא: לשורש n בנים, בכל רמה מתחתיו פי n צמתים, עד רמת העלים ($l - 1$ פעמים), ובכל עלה $n - 1$ ערכים. (עמ' 251–252) הכי מעט ערכים – כל צומת במינימום: לשורש 2 בנים (שורש שאינו עלה), לכל צומת אחר $\lceil n/2 \rceil$ בנים ($l - 2$ פעמים), ובכל עלה $\lceil n/2 \rceil - 1$ ערכים. הפוך: עץ עם x ערכים יכול להיות בן l רמות כש- $x_{min}(l) \leq x \leq x_{max}(l)$. (עמ' 251–252, 256)	$x_{max}(l) = \underbrace{n \cdot n \cdot \dots \cdot n}_{l-1} \cdot (n - 1) = n^{l-1} (n - 1)$ $x_{min}(l) = 2 \cdot \underbrace{\left\lceil \frac{n}{2} \right\rceil \cdot \dots \cdot \left\lceil \frac{n}{2} \right\rceil}_{l-2} \cdot \left(\left\lceil \frac{n}{2} \right\rceil - 1 \right)$ $x_{min}(3) = 2 \cdot 76 \cdot 75 = 11,400$ $x_{max}(3) = 151^2 \cdot 150 = 3,420,150$ (עבור $n = 151$)

6.2 גיבוב

ניסוח המשפט	כתיב מתמטי
גיבוב סטטי – פונקציה (למשל $h(x) = x \bmod k$) ממפה כל ערך לסל. התנגשות – כמה ערכים לאותו סל; גלישה – הסל מלא, ומקצים לו סל נוסף. הפונקציה קבועה ולא מתאימה כשהקובץ גדל מאוד. (עמ' 264–267)	$56, 48, 64 \rightarrow 0 : h(x) = x \bmod 8$ סל 0 מלא $\leftarrow 64$ לסל גלישה
גיבוב מתרחב – סל נבחר לפי i הסיביות הראשונות של $h(x)$; בטבלת הסלים 2^i שורות; לכל סל רוחב רישא משלו ($\leq i$). כמה שורות בטבלה: מספר הסלים $\lceil x/סל \rceil$, וחזקת 2 הקטנה ביותר שאינה קטנה ממנו. (עמ' 267–270)	$i \leq \text{רוחב סל}, \text{ שורות } 2^i$

אלגוריתם להכנסת ערך במבנה של גיבוב מתרחב (עמ' 267–270):

1. בהתחלה: סל אחד, $i = 0$ – מכניסים ערכים בלי להתייחס לפונקציית הגיבוב.
 2. מצא את הסל לפי i הסיביות הראשונות של $h(x)$; אם יש בו מקום – הכנס, וסיים.
 3. אם הסל מלא:
 - אם רוחב הרישא שלו $i =$ – **הכפל את הטבלה**: i גדל ב-1, וכל שורה חדשה מצביעה לאותו סל כמו השורה שממנה נוצרה.
 - **פצל את הסל** לפי הסיבית הבאה: שני סלים עם רוחב רישא גדול ב-1, והערכים מתחלקים ביניהם.
 4. חזור על צעד 2 (אם כל הערכים עברו לאותו סל – מפצלים שוב).
- לדוגמה נתון (סל של 2 ערכים):

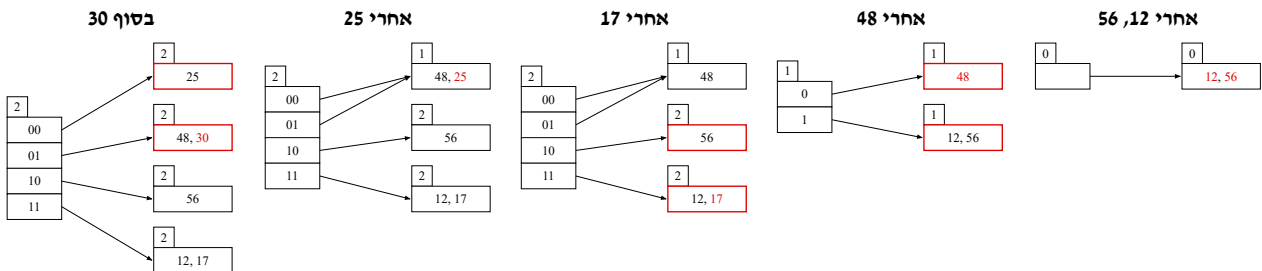
$$12 = 1100, 56 = 1010, 48 = 0111, 17 = 1111, 25 = 0010, 30 = 0100$$

נפעיל את האלגוריתם כדלקמן:

12, 56 – לסל היחיד ($i = 0$), בלי להתייחס לפונקציית הגיבוב

- 48: הסל מלא, ורוחב הרישא שלו $i = 0 \Rightarrow$ מכפילים: $i = 1$; מפצלים לפי הסיבית הראשונה: $0 \rightarrow [48]$, $1 \rightarrow [12, 56]$
- $17 = 1111$: הסל של 1 מלא, ורוחבו $i = 1 \Rightarrow$ מכפילים: $i = 2$; מפצלים לפי הסיבית השנייה: $10 \rightarrow [56]$, $11 \rightarrow [12, 17]$
- $25 = 0010$: השורה 00 מצביעה לסל של 0 (ברוחב 1), ויש בו מקום: $[48, 25]$
- $30 = 0100$: השורה 01 – לאותו סל, והוא מלא; רוחבו $i = 2 < 1 \Rightarrow$ מפצלים **בלי** להכפיל: $00 \rightarrow [25]$, $01 \rightarrow [48, 30]$

נקבל אפוא את המבוקש (המבנה אחרי כל פיצול, והאחרון – התוצאה):



אלגוריתם לחישוב מספר הסלים, מספר השורות בטבלה ורוחב הרישא – המינימליים (עמ' 267–270):

1. **מספר הסלים המינימלי a :**
 - אם נתון מספר הסלים – הוא a .
 - אם נתונים v ערכים וסל של s ערכים לכל היותר – כל סל מלא: $a = \lceil \frac{v}{s} \rceil$.
2. **רוחב הרישא המינימלי i :** לכל סל לפחות שורה אחת בטבלה, ובטבלה 2^i שורות – ולכן $2^i \geq a$: $i = \lceil \log_2 a \rceil$.
3. **מספר השורות המינימלי בטבלה** (כתובות הסלים): 2^i .

לדוגמה נתון (אותו מבנה):

$$v = 6 \text{ ערכים, } s = 2$$

נפעיל את האלגוריתם כדלקמן:

$$a = \lceil \frac{6}{2} \rceil = 3 \quad i = \lceil \log_2 3 \rceil = 2 \quad 2^2 = 4$$

נקבל אפוא את המבוקש:

לפחות 3 סלים, רוחב רישא לפחות 2 ולפחות 4 שורות בטבלה – ובמבנה שבנינו אכן $i = 2$ ו-4 שורות

ניסוח המשפט	כתיב מתמטי
מפת סיביות – סריג: שורה לכל ערך, עמודה לכל רשומה, 1 אם הרשומה מכילה את הערך. שאילתה עם כמה תנאים – and בין השורות. עצי B + טובים גם לשאילתות טווח; גיבוב – רק לשוויון. (עמ' 271–272)	... Chef: 0 0 1 0 0 0 1 ... Japan: 0 0 0 0 1 0 0

7 עיבוד שאילתות

7.1 הערכת גודל

ניסוח המשפט	כתיב מתמטי
נתונים סטטיסטיים – מספר השורות ביחס $b_r ; r$ מספר הגושים של $r ; s_r$ גודל שורה $V(A, r)$; מספר הערכים השונים של התכונה A ב- r . אם A מפתח של r – כל ערך מופיע בשורה אחת בלבד, ולכן $V(A, r) = n_r$. (עמ' 285)	$r \Rightarrow V(A, r) = n_r$ מפתח של A
בחירה $A = c$ – בהתפלגות אחידה $\frac{n_r}{V(A, r)}$ שורות ; סלקטיביות – החלק שנשאר $(\frac{1}{V(A, r)})$; כמה תנאים עם "וגם" – מכפילים. (עמ' 289–290)	$ \sigma_{A=c}(r) = \frac{n_r}{V(A, r)}$ $sel(\theta_1 \wedge \theta_2) = sel(\theta_1) \cdot sel(\theta_2)$
צירוף $r \bowtie s$ – $R \cap S = \emptyset$ (1) – מכפלה, $n_r \cdot n_s$. $R \cap S \neq \emptyset$ (2) מפתח של r : כל שורה של s מצטרפת לכל היותר לשורה אחת של r – לכל היותר n_s (כשכל שורה מפנה לשורה קיימת – בדיוק n_s). (3) אחרת : מחלקים ב- V הגדול. (עמ' 294–295)	$ r \bowtie s \approx \frac{n_r \cdot n_s}{\max(V(A, r), V(A, s))}$
מפתח – אבל לא כל ערך מוצא בן זוג – A הוא המפתח של s , אבל ב- r יש יותר ערכים שונים של A ממספר השורות של s ($V(A, r) > n_s$). רק $\frac{n_s}{V(A, r)}$ מהשורות של r מוצאות בן זוג, ולכן לא n_r אלא הנוסחה הכללית. דוגמה (באלגוריתם שלמטה) : 300,000 תגובות עם 20,000 כותבים שונים, אבל ב-user רק 10,000 משתמשים. (עמ' 294–295)	$\frac{300,000 \cdot 10,000}{\max(20,000, 10,000)} = 150,000$ ולא 300,000
כללי שקילות – (1) בחירה עם "וגם" = בחירות בזו אחר זו ; (2) בחירות מתחלפות ; (3) מכפלה ובחירה = צירוף ; (4) צירוף חילופי ; (5) צירוף קיבוצי ; (6) בחירה שנוגעת לארגומנט אחד – עוברת אליו ; (7) איחוד וחיתוך חילופיים וקיבוציים. (עמ' 296)	$\sigma_\theta(r \times s) = r \bowtie_\theta s$ $\sigma_{\theta_1}(r \bowtie s) = \sigma_{\theta_1}(r) \bowtie s$

אלגוריתם להערכת גודל התוצאה (עמ' 289–296):

- מוציאים החוצה את התנאים הבדידים – תנאי בחירה שנוגעים ליחס אחד בלבד; מפעילים אותם בסוף (כללי השקילות, עמ' 296).
- הופכים את המכפלות לצירופים – תנאי שמשווה בין שני יחסים הופך את המכפלה לצירוף (עמ' 296):
 - מצרפים יחס אחד בכל פעם – כל צירוף חדש מצרף יחס לתוצאה הקודמת.
 - אין תנאי שמקשר בין שני היחסים (ואין תכונה משותפת) – זו מכפלה: $n_r \cdot n_s$ (עמ' 291, 294).
 - צירוף לפי תכונות בשמות שונים ($A = D$) או לפי זוג תכונות – אותם כללים, עם התכונה (או הזוג) שבכל צד.
- בכל צירוף בודקים של מי התכונה המשותפת היא מפתח (עמ' 294–295):
 - אם היא המפתח של צד אחד ("צד המפתח") וגם V שלה בצד השני $\leq$ מספר השורות בצד המפתח – כל שורה בצד השני מוצאת בן זוג אחד בדיוק:
 - מספר השורות בתוצאה = כמו בצד השני.
 - המפתח של התוצאה = המפתח של הצד השני.
 - אחרת – V שלה בצד השני גדול ממספר השורות בצד המפתח (חלק מהשורות לא מוצאות בן זוג), או שהיא אינה מפתח של אף צד:
 - מספר השורות בתוצאה = הנוסחה הכללית $\cdot \frac{n_r \cdot n_s}{\max(V(A,r), V(A,s))}$.
- מחלקים בתנאים הבדידים (עמ' 289–290):
 - תנאי $A = c$ משאיר $\frac{1}{V(A,r)}$ מהשורות (ואם A מפתח – שורה אחת).
 - תנאי טווח – החלק היחסי מהטווח: שנה אחת מתוך 5 שנות נתונים משאירה $\frac{1}{5}$.
 - כמה תנאים עם "וגם" – מכפילים את החלקים.
- ההטלה משאירה מכל שורה רק את העמודות המבוקשות:
 - הטלה על תכונה אחת A של יחס $V(A, r)$ שורות, אחת לכל ערך (עמ' 285).
 - אם העמודות שנשארות לא כוללות מפתח – שורות זהות מתאחדות, והתוצאה היא לכל היותר המספר שחושב.
- פעולות על קבוצות (עמ' 291):
 - $r \cap s$ – צירוף לפי כל העמודות, באותם כללים של צעד 3.
 - $n_r - r - s$ – פחות גודל החיתוך.
 - $r \cup s$ – לכל היותר $n_r + n_s$.

לדוגמה נתון (מבחן 2025א, מועד 62, שאלה 6) – רשת חברתית:

user (uid, name, email, password, descr, country)

post (pid, uid, content, imageURL, pdate, ptime)

comment (pid, cdate, ctime, uid, content)

$$n_{\text{user}} = V(\text{uid}, \text{user}) = 10,000 \quad n_{\text{post}} = V(\text{pid}, \text{post}) = 100,000 \quad n_{\text{comment}} = 300,000$$

$$V(\text{country}, \text{user}) = 200 \quad V(\text{uid}, \text{post}) = 8,000 \quad V(\text{pid}, \text{comment}) = 90,000$$

$$V(\text{name}, \text{user}) = 9,500 \quad V(\text{uid}, \text{comment}) = 7,000$$

במערכת נתונים על 5 שנים בלבד. מה גודל התוצאה של השאילתה:

$$\pi_{\text{user.uid, name}} \left(\sigma_{\text{user.uid}=\text{comment.uid} \wedge \text{country}='Canada'} (\text{user} \times \sigma_{\text{comment.pid}=\text{post.pid}} (\text{comment} \times \sigma_{2024}(\text{post}))) \right)$$

נפעיל את האלגוריתם כדלקמן:

צעד 1 – התנאים הבדידים: $\text{country} = 'Canada'$ (על user) ו- $\text{pdate} \in 2024$ (על post) – יוצאים החוצה

$$\Rightarrow \text{comment} \bowtie_{\text{pid}} \text{post} : \text{pid המפתח של post, ו-} 100,000 \leq 90,000 - \text{"אם"} \text{ של צעד 3: כמו בצד השני } 300,000$$

$$\Rightarrow \text{user} \bowtie_{\text{uid}} (\dots) : \text{uid המפתח של user, ו-} 10,000 \leq 7,000 - \text{"אם"} \text{ של צעד 3: כמו בצד השני } 300,000$$

$$\frac{300,000}{5 \cdot 200} = 300$$

נקבל אפוא את המבוקש:

כ-300 שורות

ואם במקום $V(\text{uid}, \text{comment}) = 7,000$ נתון $V(\text{uid}, \text{comment}) = 20,000$ (שאר הנתונים כמו קודם; למשל: יש גם תגובות של משתמשים שנמחקו מ-user)

נפעיל את צעד 3 שוב:

$$\Rightarrow 300,000 \text{ "אם" } - \text{comment} \bowtie_{\text{pid}} \text{post}$$

$$\bowtie_{\text{uid}} \text{user} : (\dots) \text{ uid : המפתח של user, אבל } 10,000 > 20,000 - \text{"אחרת" של צעד 3}$$

למה? יש 20,000 כותבים שונים ורק 10,000 משתמשים – מחצית מהכותבים לא ב-user, והתגובות שלהם לא מוצאות בן זוג. לכן הנוסחה הכללית:

$$\frac{300,000 \cdot 10,000}{\max(20,000, 10,000)} = 150,000 \quad \frac{150,000}{5 \cdot 200} = 150$$

נקבל אפוא את המבוקש:

כ-150 שורות – ולא 300: רק מחצית מהתגובות כתובות על-ידי משתמש שנמצא ב-user

דוגמה לצעד 6 – ממשיכים מהתוצאה שקיבלנו: Q – השאילתה שלמעלה: משתמשים מקנדה שהגיבו על פוסט מ-2024 (300,000 שורות אחרי הצירופים, כ-300 אחרי התנאים). כמה מהם לא פרסמו בעצמם פוסט ב-2024?

$$Q - P, \quad P = \pi_{\text{user.uid, name}} (\sigma_{\text{country}='Canada'}(\text{user}) \bowtie_{\text{uid}} \sigma_{2024}(\text{post}))$$

נפעיל את האלגוריתם כדלקמן:

$$\Rightarrow 100,000 \text{ כמו בצד השני } 100,000 - \text{"אם"} : \text{uid : המפתח של user, ו-} 10,000 \leq 8,000 - \text{"אם"} : \text{כמו בצד השני } 100,000$$

$$|P| = \frac{100,000}{5 \cdot 200} = 100$$

צעד 6 – החיתוך $Q \cap P$ הוא צירוף לפי כל העמודות (uid, name) : uid : המפתח של Q (300 שורות, כל משתמש פעם אחת), ו- $100 \leq 300$

$$\Rightarrow 100 - \text{"אם"} : \text{כמו בצד השני } 100$$

$$|Q - P| = 300 - 100 = 200$$

נקבל אפוא את המבוקש:

כ-200 משתמשים. (ובאיחוד $Q \cup P$ – לכל היותר $300 + 100 = 400$).

7.2 עץ ביטוי – לפני ואחרי

עץ ביטוי – כל צומת פעולה, העלים הם היחסים, ומחשבים מלמטה למעלה (עמ' 277–278). **משמאל:** השאילתה כפי שהיא נתונה – מכפלות, ותנאים במקומות שונים. **מימין:** אחרי האלגוריתם – התנאים הבדידים למעלה, המכפלות צירופים, ובכל צומת מספר השורות. **כללי אצבע לחישוב יעיל:** בחירות והטלות מוקדם ככל האפשר, וצירופים עם תוצאה קטנה קודם (עמ' 300–301).

