

החוק היבש...

מרצה: גלה ידגר

שעת קבלה: ימי ב' 12:30-13:30

חישוב הציון:

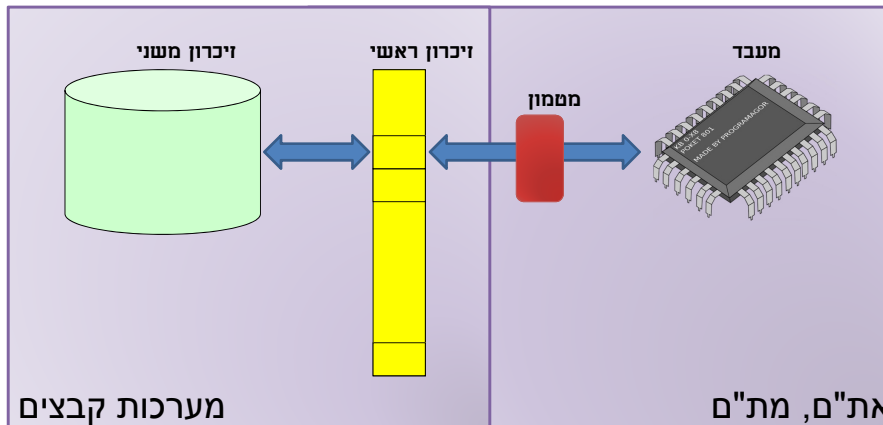
- הציון התקף מורכב מ: 3% * 4 תרגילים יבשים - מגן 88% בחינה סופית
- אם הציון בבחינה הסופית נמוך מ 45, הציון בקורס זהה לציון בבחינה הסופית

דרישות קדם: מבני נתונים 1, הסתברות, רצוי מערכות הפעלה

מערכות קבצים

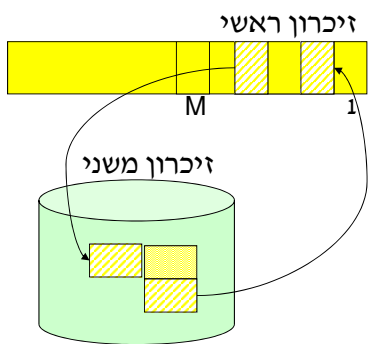
234322

ארגון המחשב



מבוא

ניהול הזיכרון המשני



- הנתונים נמצאים בזיכרון המשני, מחולקים לבלוקים
- לא מבצעים חישובים ישירות בזיכרון המשני, אלא מביאים בלוק של נתונים לזיכרון הראשי ושם מעבדים אותם
- הזיכרון הראשי מחולק למסגרות, כ"א יכולה להכיל כל בלוק מהזיכרון המשני M מתוכן מוקצות לביצוע שאילתות
- הזיכרון הראשי קטן יחסית לזיכרון המשני, ולפעמים צריך לפנות בלוקים (בלוקים מעודכנים נכתבים לזיכרון המשני)



כלל 1: קריאה וכתובה לזיכרון המשני מתבצעות רק בבלוקים

זיכרון ראשי לעומת זיכרון משני

זמן גישה אקראית	מהירות גישה/עדכון	מחיר ל-1GB	מחיר	נפח	סוג זיכרון
9 nsec	1333 MHz	7.5 \$	30\$	4 GB	ראשי (SRAM)
4.16 msec	7200 rpm	8 cents	80\$	1 TB	משני (hard disk)

זיכרון משני
 ✓ מחיר
 ✓ נפח
 ✗ מהירות

יחס בין זמני גישה אקראית:
 ראשי : משני ≈ 1 : 460,000

זיכרון משני:

- שמירת כמויות גדולות של מידע, במחיר זול יחסית
- שמירת מידע באופן אמין וממושך (גם כאשר המערכת כבויה)

מערכת קבצים



מערכת תוכנה המטפלת בניהול, ארגון, שליטה ומתן גישה ושירותים למשתמשים בקבצים.



מערכת ניהול מסדי נתונים

מערכת תוכנה המאפשרת יצירת וניהול מסדי נתונים מאפשרת ארגון, שליטה ומתן גישה ושירותים למאגרי מידע מובנים

שאלות ונושאים בהם עוסק הקורס



- ✓ מהו המבנה של מערכת אחסון מידע ומה השפעתו על הביצועים שלהן?
- ✓ איך לתכנן מבני נתונים גדולים מאוד שלא נכנסים לזיכרון הראשי?
- ✓ איך מחשבים את העלות (המקורבת) של ביצוע פעולות?
- ✓ איך לאפשר גישה במקביל והתאוששות?

מסדי נתונים רלציונים (טבלאיים)

רמת תצפית

התצפית של כל משתמש על פי היישום

רמה לוגית

ייצוג הנתונים על פי המודל כטבלאות

רמה פיזית

שמירה בפועל על הדיסק

דרך מקובלת לארגון מאגרי נתונים גדולים

- המידע נמצא בטבלאות (structured)
- המערכת שומרת את המידע בקבצים מתאימים ומנהלת את הקבצים
- יצירת אינדקס לטבלה
- מתן הרשאות גישה למשתמשים
- שונים לפרטי מידע שונים

סוגי קבצים

לא מובנים (unstructured): המבנה אינו ידוע למערכת הקבצים
 • ניתן לגשת רק לקובץ השלם (לקרוא, לכתוב או למחוק)
 • למשל: קבצי אודיו, וידאו

מובנים (structured): סדרה ארוכה של רשומות בעלות מבנה זהה או דומה.
 – בדרך כלל, מאורגנות על-פי שדה מפתח (אחד או יותר).
 במיוחד: קבצים הנוצרים על ידי מערכת לניהול מסדי נתונים



- ניתן לגשת לכל רשומה (ולרוב גם לעדכנה) בנפרד
- פעולות:
 - חיפוש רשומה, עדכון רשומה, הכנסת רשומה, מחיקת רשומה
 - מעבר סדרתי על כל הרשומות
 - חיפוש תת-תחום – חיפוש רשומות הממלאות תנאי לוגי נתון
 - פעולות הקבצה (aggregation): חישוב סכום, ממוצע וכדומה

יחס (רלציה, טבלה)

$Customer \subseteq string \times no \times city$

שורה בטבלה היא רשומה

שם פרטי	מספר חשבון	עיר מגורים
משה	23412	ת"א
שרה	56234	חיפה
דוד	90567	גדרה
אליהו	4591	עפולה
מינה	62902	חיפה

תחום (domain): קבוצה של ערכים אפשריים עבור שדה ברשומה
 • כל המחרוזות באורך 20
 • מיקוד (כל המספרים בין 10000 ל-99999)
 • תאריך (1.1.1900 - 31.12.2999)
 • מספר שלם

יחס (relation): רב-קבוצה (multiset) של איברים מהמכפלה הקרטזית של התחומים

רשומות ושדות

רשומה (record) - יחידת מידע בעלת מבנה זהה לאחרות (או דומה)
 – קבצים בעלי רשומות שוות גודל (fixed size records)
 – קבצים בעלי רשומות עם גודל משתנה (variable size records)
 – בקורס זה נניח שהרשומות הן שוות גודל

שדה (field) - כל רשומה מתחלקת לשדות

שדה מפתח (key field) - מזהה את הרשומה לצרכי מיון או חיפוש

שדה מפתח ראשי (primary key field) - משמש לארגון הפנימי של הקובץ

שם השדה	מספר סטודנט	שם סטודנט	ציון בת"ב 1	ציון בת"ב 2	ציון בת"ב 3	ציון במבחן מועד ב'	ציון במבחן מועד א'
גודל השדה בביתם	9	20	1	1	1	1	1

סדר וכפילויות

בפועל, סדר שמירת הרשומות משפיע על המימוש של חלק מהפעולות.

– לעיתים, משתמשים מעוניינים בתוצאה ממוינת או רק ברשומות הראשונות של יחס (על פי סדר כלשהו)

בפועל, במערכות לניהול מסדי נתונים טבלאיים, יחס הוא רב קבוצה (רשומה יכולה להופיע יותר מפעם אחת).

• אנחנו נניח כי כל הפעולות הן פעולות על רב-קבוצות.

סכמת היחס

relation scheme: שם היחס והתכונות שלו, ותחומיהם.

שם היחס: בדוגמא הקודמת - Customer

התכונות (attributes): בדוגמא הקודמת - {שם פרטי, מספר חשבון, עיר}

דוגמאות (קצת שונה):

```
Branch(b_name, assets, b_city)
Customer(c_name, street, c_city)
Deposit(b_name, account_number, customer_name, balance)
Borrow(b_name, loan_number, c_name, amount)
```

לפעמים נצרף את שם היחס S לשם התכונה a ונכתוב S.a, למשל Branch.b_name או Deposit.b_name

פעולות על יחסים

פעולות אונריות שמחזירות יחס חדש

- π – הטלה (projection): בוחר חלק מהעמודות
- σ בחירה (selection): בוחר חלק מהרשומות
- δ : הפוך את היחס לקבוצה (השאר עותק אחד מכל רשומה)
- $\rho_{A \rightarrow B}$: החלפת שם של שדה
- Sort: מארגן מחדש את היחס על ידי מיון על פי תכונה או מספר תכונות

פעולות בינריות שמחזירות יחס חדש

- \cup , \cap , \setminus : איחוד, חיתוך, הפרש, מכפלה
- Join: צירוף

פונקציות הקבצה על עמודה המכילה מספרים: max, min, average

פונקציית הקבצה על טבלה: count

שאלות במסד נתונים רלציוני

פעולות אבסטרקטיות על הנתונים באופן שאינו תלוי במימוש.

שאלות נכתבות בשפה שמיועדת לבני-אדם, על-מנת לפשט אותן. למשל, הצג את שמות כל הסטודנטים בני פחות מ-20, ב-SQL:

```
SELECT name
FROM Students
WHERE birth_year > 1992;
```

השאלות מתורגמות לפעולות על יחסים (כמו תרגום Java לשפת מכונה), אשר מגדירות אלגברה של יחסים (רלציונית)

בהמשך, נראה איך מבצעים את הפעולות על היחסים (בנפרד, או כחלק מביטוי מורכב)

הטלה π projection

בוחרת חלק מעמודות היחס

Customer

שם פרטי	מספר חשבון	עיר מגורים
אלון	23412	ת"א
אלה	56234	חיפה
אורן	90567	גדרה
הדס	62902	חיפה

$\pi_{\text{שם, מספר}}$ (Customer)

שם פרטי	מספר חשבון
אלון	23412
אלה	56234
אורן	90567
הדס	62902

$\pi_{\text{עיר}}$ (Customer)

עיר מגורים
ת"א
חיפה
גדרה
חיפה

עשוי ליצור רשומות כפולות

$$\pi_{\ell_1, \ell_2, \dots, \ell_n}(R) = \{(r_{\ell_1}, r_{\ell_2}, \dots, r_{\ell_n}) : (r_1, \dots, r_m) \in R\}$$

בחירה σ select

בוחר את הרשומות המקיימות תנאי נתון C

Customer

שם פרטי	מספר חשבון	עיר מגורים
אלון	23412	ת"א
אלה	56234	חיפה
אורן	90567	גדרה
הדס	62902	חיפה

$\sigma_{\text{city} = \text{חיפה}}$ (Customer)

שם פרטי	מספר חשבון	עיר מגורים
אלה	56234	חיפה
הדס	62902	חיפה

$$\sigma_C(R) = \{r \in R : r \text{ satisfies } C\}$$

שאלות טווח [a,b] מוגדרות באמצעות $\sigma_{a \leq A_2 \leq b}(R)$

חיתוך \cap

R1

שם פרטי	רחוב מגורים	עיר מגורים
אלון	הרצל	ת"א
אלה	חושי	חיפה
אורן	הרצל	גדרה
אלון	הרצל	ת"א

$R1 \cap R2$

שם פרטי	רחוב מגורים	עיר מגורים
אלון	הרצל	ת"א

מספר המופעים של רשומה הוא המינימום בין מספר המופעים בכל אחד מהיחסים

R2

שם פרטי	רחוב מגורים	עיר מגורים
אלון	הרצל	ת"א
כרמל	חושי	חיפה

$$R \cap S = \{r : r \in R \text{ and } r \in S\}$$

דורש סכמות זהות

איחוד \cup

R1

שם פרטי	רחוב מגורים	עיר מגורים
אלון	הרצל	ת"א
אלה	חושי	חיפה
אורן	הרצל	גדרה
אלון	הרצל	ת"א

$R1 \cup R2$

שם פרטי	רחוב מגורים	עיר מגורים
אלון	הרצל	ת"א
אלה	חושי	חיפה
אורן	הרצל	גדרה
אלון	הרצל	ת"א
אלון	הרצל	ת"א
כרמל	חושי	חיפה

מספר המופעים של רשומה הוא סכום מספר המופעים בשני היחסים

R2

שם פרטי	רחוב מגורים	עיר מגורים
אלון	הרצל	ת"א
כרמל	חושי	חיפה

$$R \cup S = \{r : r \in R \text{ or } r \in S\}$$

דורש סכמות זהות

מכפלה קרטזית ×

R	A1	A2	A3			
	a	b	c			
	a	b	c			
	c	b	d			

S	B1	B2			
	a	b			
	c	b			

R × S		A1	A2	A3	B1	B2
		a	b	c	a	b
		a	b	c	a	b
		c	b	d	a	b
		a	b	c	c	b
		a	b	c	c	b
		c	b	d	c	b

מספר הרשומות בתוצאה: (מספר רשומות S) × (מספר רשומות R)

$$R \times S = \{(r_1, r_2, \dots, r_n, s_1, \dots, s_m) : r \in R \text{ and } s \in S\}$$

הפרש \

R1	שם פרטי	רחוב מגורים	עיר מגורים
	אלון	הרצל	ת"א
	אלה	חושי	חיפה
	אורן	הרצל	גדרה
	אלון	הרצל	ת"א

R2	שם פרטי	רחוב מגורים	עיר מגורים
	אלון	הרצל	ת"א
	כרמל	חושי	חיפה

R1 \ R2	שם פרטי	רחוב מגורים	עיר מגורים
	אלון	הרצל	ת"א
	אלה	חושי	חיפה
	אורן	הרצל	גדרה

מספר המופעים של רשומה הוא ההפרש בין מספר המופעים ביחס הראשון ומספר המופעים ביחס השני (אבל לא פחות מאפס)

$$R \setminus S = \{r : r \in R \text{ and } r \notin S\}$$

דורש סכמות זהות

עוד דוגמה ל join

customers מכיל לקוחות: מס' לקוח, שם וכתובת.
bills מכיל חיובים: מס' לקוח וסכום שיש לחייב.

customers			bills		=			
Name	Add	No	No	Sum	Name	Add	No	Sum
Jim	1 st st	1	17	23.1	Jim	1 st st	1	12.0
Mary	Main	4	1	12.0	Jim	1 st st	1	100.5
Joe	5 th Ave.	17	1	100.5	Joe	5 th Ave.	17	23.1
Alice	Main	23						

ב-SQL

SELECT Name, Add, No, Sum
FROM customers, bills
WHERE customers.No = bills.No

צירוף join:

יחס חדש שמתקבל משני יחסים שיש להם תכונות עם שמות זהים.

היחס מתקבל מצירוף הרשומות שמסכימות על התכונות עם השמות הזהים.
R(name,address) ⋈ S(name,tel#) = Q(name, address,tel#) למשל

name	address	name	tel#	=	name	address	tel#
Nick	2 nd st	Nick	234		Nick	2 nd st	234
Jane	Maine	Jane	892		Jane	Maine	892
Jane	NYC				Jane	NTC	892

ביצוע פעולות במעבר יחיד

קל לבצע פעולה בחירה σ במעבר יחיד

- נקרא את רשומות R בבולקים
- לכל רשומה r ב R: אם r מקיימת את התנאי, נוציא אותה

באופן דומה, אפשר לבצע הטלה π או פעולת הקבצה, למשל (min, max, sum, average)

מספר גישות לדיסק: מספר הבולקים ב R

גם איחוד אפשר לבצע במעבר יחיד

הקלט והפלט הן רב-קבוצות לא ממוינות.

מספר גישות לדיסק: מספר הבולקים ב S + מספר הבולקים ב R

דוגמה לביצוע $S \setminus R$

R		S	
A	B	A	B
11	c2	20	c1
40	c2	11	c2
20	c1	30	c3
11	c2	20	c1
20	c1	11	c2
20	c1	11	c2
50	c5		
40	c2		

(בדיסק)

(בדיסק)

באופן יותר פורמלי

קבוצת התכונות המשותפות ליחסים R ו-S היא $A = attr(R) \cap attr(S)$

נגדיר join בין שורות:

- יהיו $r \in R, s \in S$ כך ש- $r.A = s.A$
- $s \bowtie r$ היא השורה שמתקבלת משרשר s ו- r
- תוך שמירת עותק אחד בלבד של תכונות A.



$s = (\text{Homer Simpson}, 555-2121)$

$r = (\text{Homer Simpson}, 123 \text{ Fake St. Springfield})$

$r \bowtie s = (\text{Homer Simpson}, 123 \text{ Fake St. Springfield}, 555-2121)$

עבור היחסים

$$R \bowtie S = \{r \bowtie s : r \in R, s \in S, \pi_A(r) = \pi_A(s)\}$$

מה עם פעולות בינריות אחרות?

ניתן לבצע במעבר יחיד אם היחס הקטן מבין השניים נכנס כולו לזיכרון

דוגמה: $S \setminus R$, כאשר S נכנס לזיכרון הראשי

- נשמור עותק מכל רשומה של S עם מונה של מספר המופעים שלה.
- נעבור על כל רשומה r ב R,
- אם היא מופיעה ב-S, נקטין את המונה ב 1.
- בסוף נחזיר כל איבר של S שהמונה שלו נותר חיובי, בריבוי המתאים.

ביצוע צירוף במעבר יחיד

אפשרי אם היחס S נכנס לזיכרון

- נכניס את כל רשומות S לזיכרון הראשי (בבלוקים)
- נקרא את רשומות R (בבלוקים)
- לכל רשומה r ב R (כמידת ריבוייה):
 - נחפש בין רשומות S שנמצאות בזיכרון הראשי
 - לכל רשומה s עבורה $r.A = s.A$, נוציא $r \bowtie s$ לפלט



מספר גישות לדיסק: מספר הבלוקים ב S + מספר הבלוקים ב R

דוגמה לביצוע S \ R

פלט

(11,c2)
(30,c3)

R

A	B
11	c2
40	c2
20	c1
11	c2
20	c1
20	c1
50	c5
40	c2

(בדיסק)

S

A	B	#
20	c1	0
11	c2	1
30	c3	1

(בזיכרון הראשי)

ואם שני היחסים גדולים?

לולאה כפולה

נקרא בלוק של S ולכל רשומה s
נעבור על כל הבלוקים של R, ולכל רשומה r
נבדוק אם אפשר לצרף את r עם s

גישות לדיסק: מספר הרשומות ב S × מספר הבלוקים ב R

שיפור: ממלאים את הזיכרון בבלוקים של S (M בלוקים)
קוראים את R בבלוקים ומנסים לצרף כל רשומה של R עם כל הרשומות של S שנמצאות בזיכרון

גישות לדיסק (בערך): (מספר הבלוקים ב S × מספר הבלוקים ב R) / M

כלל 2: גודל הזיכרון הראשי קובע את מספר הגישות לדיסק

דוגמה להרצה

פלט

(a,b1,c2)
(a,b1,c3)
(a1,b1,c2)
(a1,b1,c3)
(a6,b3,c4)

R

A	B
a	b1
a2	b2
a1	b1
a6	b3
a4	b3
a5	b3

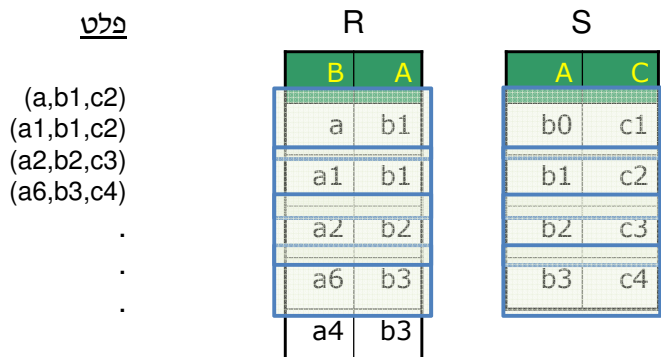
(בדיסק)

S

B	C
b	c1
b1	c2
b1	c3
b3	c4

(בזיכרון הראשי)

דוגמה להרצה



כלל 3: כדאי לגשת למידע באופן סידרתי

שיפור אם היחסים ממוינים

נניח שהצירוף לפי תכונה אחת A, שהיא **מפתח** של S ו- R ממוינים לפי A

עוברים על R ו-S בבלוקים,
 וכל פעם בוחנים רשומה s של S ורשומה r של R
 אם $s.A < r.A$, עבור לרשומה הבאה ב-S
 אם $s.A > r.A$, עבור לרשומה הבאה ב-R
 אחרת, $s.A = r.A$, הוצא לפלט את הרשומה $r \bowtie s$
 ועבור לרשומה הבאה ב-R

גישות לדיסק: מספר הבלוקים ב S + מספר הבלוקים ב R

מאגר מפתח-ערך (Key-Value Store)

- לכל אובייקט נתונים (ערך) מוצמד **מפתח** חד-ערכי לזיהוי האובייקט
- מבנה הנתונים ומשמעותם לא ידועים למערכת (unstructured)
- מפתחות: מחרוזת טקסט, מספר סידורי, תוצאת ערבול ועוד
- ערכים: קבצים מכל הסוגים, מצביעים וכתובות, URL, מחרוזת טקסט ועוד

ערך	מפתח
Front.jpg	"תמונת שער"
0x112b210	"p_song"
www.cs.technion.ac.il/courses	"1.1"
"123abc??"	"סיסמא"

מסקנות

- 1: חייבים לגשת ליחסים (קבצים) בבלוקים
- 2: הגודל של הזיכרון הראשי קובע
- 3: מאוד כדאי לגשת לנתונים (קבצים / יחסים) באופן סידרתי (ולכן טוב אם הם ממוינים)

MapReduce

- ביצוע שאילתות מורכבות במאגר מפתח-ערך
- נתון מאגר התחלתי (בדרך כלל מבוזר)
- השאילתא מוגדרת בשני שלבים:
 1. $\text{map}(k1, v1) \rightarrow \text{list}(k2, v2)$
 2. $\text{reduce}(k2, \text{list}(v2)) \rightarrow \text{list}(v3)$
- המתכנת מגדיר את הסמנטיקה של הפונקציות map , reduce
- המערכת אחראית על
 - חלוקת תפקידים בין שרתים
 - Shuffle - העברת תוצאות שלב ה- map לשרת המתאים
 - פלט של התוצאות ממויינות על פי $k2$

MapReduce - דוגמא

Input

שם	כתובת	ה. קבע	דוא"ל
אלכס	הזמיר, חיפה	ויזה לאומי	alex@
אייל	השלום, ר"ג	בנק הפועלים	eyal@

Map()

city	name	email
חיפה	אורלי	orly@
חיפה	חנן	han@
חיפה	אלכס	alex@

שם	דוא"ל	כתובת	סכום	תאריך
אלה	ela@	ביאליק, ר"ג	300	13.8.13
אורלי	orly@	מוריה, חיפה	180	25.9.13

city	name	email
ר"ג	אייל	eyal@
ר"ג	אלה	ela@

שם	דוא"ל	טלפון	גיל	כתובת	חוג
חנן	han@	12345	7	הרצל, חיפה	טבע
נועה	noa@	23456	13	אלנבי, ת"א	סיור

city	name	email
נועה	נועה	noa@
ת"א		

מאגרי מפתח-ערך

פעולות

- **הכנסה:** של ערך חדש על פי מפתח שלא נמצא עדיין במאגר
- **הוצאה:** של ערך קיים על פי מפתח שנמצא במאגר
- **עדכון:** של ערך קיים על פי מפתח שנמצא במאגר (בפועל החלפה של האובייקט)
- **חיפוש:** של ערך על פי מפתח שנמצא/לא נמצא במאגר
- **פעולות מורכבות:** שאילתות במודל MapReduce

דוגמאות

- Apache Cassandra (Some of Facebook's features)
- Amazon Simple Storage Service (Dropbox)
- Memcached

MapReduce - דוגמא

- לחברה להגנת הטבע מספר רשימות של לקוחות ושותפים:
 - משתתפי חוגים (שם, דוא"ל, טלפון, גיל, כתובת, חוג)
 - תורמים (שם, דוא"ל, כתובת, סכום תרומה, תאריך תרומה אחרונה)
 - חברים (שם, כתובת, פרטי הוראת קבע, דוא"ל)
- סניף חיפה רוצה ליצור רשימת דיוור להפצת עלון אלקטרוני
- $\text{Map}(\text{name}, v1) \rightarrow (\text{city}, (\text{name}, \text{email}))$
- $\text{Reduce}(\text{city}, \text{list}(\text{name}, \text{email})) \rightarrow \text{list}(\text{name}, \text{email})$
- על המתכנת לזהות את שדה הכתובת ולחלץ מתוכו את העיר
- ניתן "לזרוק" את תוצאות כל הערים שאינן חיפה (בכל אחד מן השלבים או בסוף)

השוואה בין המודלים

יתרונות המודל הרלציוני

- ביטוי קשרים מורכבים בין נתונים
- מימוש יעיל (על ידי המערכת) של שפת שאילתות עשירה

יתרונות מאגר מפתח-ערך

- פשטות: יתרון במימוש ובהגנה מפני התקפות חיצוניות
- יעילות בפעולות פשוטות: כתיבה וחיפוש על פי מפתח
- קל למיקבול: מיעוט תלויות
- קל לביזור: על פני שרתי אחסון מרובים, בענן
- מקרה פרטי של NoSQL database

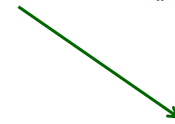
MapReduce - דוגמא

city	name	email
חיפה	אורלי	orly@
חיפה	חנן	han@
חיפה	אלכס	alex@

city	name	email
ר"ג	אייל	eyal@
ר"ג	אלה	ela@

city	name	email
ת"א	נועה	noa@

Reduce()



Output

name	email
אורלי	orly@
אלכס	alex@
חנן	han@