



## תרגול מס' 1 – בסיסי נתונים

### זרישות קדם

יסודות מבנה המחשב  
מבוא לתכנות

### מטרת הקורס

מטרת קורס זה הינה להקנות לסטודנטים היכרות ראשונית בעבודה למול בסיסי נתונים.  
בקורס נחקור את אלגברת היחסים עליה מושתת השפה, נבין את הצורך בבסיסי נתונים בחיי  
היומיום, נלמד את רכיבי השפה והכלים הקיימים בכדי לבנות שאילתות יעילות לשליפת נתונים.

### תכני הקורס

התכנים יועלו לאתר MOODLE, מפתח הצטרפות: 89281  
במקביל ניתן יהיה להוריד את התכנים גם באתר <http://www.korneto.com/courses.html>

### מנהלות

אימייל לשאלות ובעיות: [roei27@gmail.com](mailto:roei27@gmail.com) או דרך "צור קשר" באתר האינטרנט

### מופעי שיעורים

3 התרגולים זהים מבחינת התוכן אך חשוב להתמיד באותה קבוצת תרגול כדי לשמור על רצף  
התכנים, כמו כן יש להגיש את התרגילים לפי קבוצות הלימוד בהם אתם רשומים.

16:00-18:00	יום א'	-	קבוצה 01
15:00-17:00	יום ה'	-	קבוצה 02
17:00-19:00	יום ה'	-	קבוצה 03



הגדרות בסיסיות:

**(1) שדה Field:** אוסף תווים בעלי קשר לוגי, יחידת הנתונים הקטנה ביותר (שם, ת.ז.), כאשר נוכל להגביל את גודלו של שדה לכמות תווים מסוימת, לדוגמא ת.ז. תוגבל ל 9 ספרות.

**(2) רשומה Record:** אוסף של שדות בעלי קשר לוגי (רשומת סטודנט, פרטי חשבון בנק).

לדוגמא (פרטי אדם):



**(3) קובץ:** אוסף של רשומות [ הבעיה היא שכל שדה יכול להיות בגודל מסויים מה שיצור מבנה לא אחיד, ואם נרצה לשמור על מבנה אחיד נצטרך בעצם להשחיל רווחים].



**בעיות שיכולות להיגרם לנו בעבודה עם מערכת קבצים ?**

- כל תוכנית שתצא לעבוד עם הקובץ בעצם תדרוש לדעת את המבנה הפיזי של הקובץ.
- **תמיכה בשינויים** במבנה הקובץ (כל שינוי יצטרך להיות מעודכן כדי לתמוך בכל הניגשים לקובץ).
- כדי למצוא רשומה צריך היה לסרוק את כל הקובץ, בתקווה שהרשומה הדרושה איננה האחרונה.
- בעיה של **אבטחת מידע** (לא נרצה להציג את כל השדות, כמו טלפון של המתרגל בבית).
- **כפילויות ואי אמינות** (אם יש קובץ אב ושני קבצי בן אזי שינוי נתון בקובץ בן 1 לא יגרור שינוי בקובץ בן 2, לצייר ראש עם שני עלים ולהסביר).
- **תקינות הנתונים** (שינוי שם מ-דן ל-דני שהתבצע רק בחלק מהקבצים ולא בכלם).
- בעיה לחבר בין מספר קבצים ולשתף מידע ביניהם.
- **ריבוי משתמשים** (ניתן ע"י SHARE וניהול).
- **בעיה למיין לפי שדות פנימיים.**

← זו בעצם השיטה הישנה לאחסון נתונים, כי אנו בעצם שומרים את המידע בצורה סדרתית, לדוגמא את הרשומה למעלה היינו רושמים בקובץ בצורה הבאה:

Roei#####Zerahia###01234567807712345672000

← הפתרון לכל בעיות אלו: שימוש בבסיס נתונים !, המבנה הפיזי של הנתונים הופך לשקוף עבור המשתמש ומאפשר שיתוף בין התוכניות השונות.



4) **בסיס נתונים** או בשמו המוכר יותר DataBase הוא בעצם אוסף של נתונים מסוגים שונים המקושרים ביניהם בקשר לוגי כלשהו.

5) **מערכת ניהול בסיסי נתונים** DBMS – DataBase Management system, שזו בעצם מערכת (תוכנה/אפליקציה) לניהול בסיס נתונים המאפשרת יצירה, תחזוקה, שימוש וגישה לנתונים עם אפשרות לעבודה של ריבוי משתמשים במקביל.

#### הדרישות מבסיס הנתונים:

- 1) אחסון נתונים ושליפה נוחה
- 2) אבטחת מידע (רק בעל הרשאה יכול לגשת לנתונים).
- 3) ריבוי משתמשים בו זמנית.
- 4) פעולות אוטומטיות (גיבוי, שיחזור, העתקה, כיווץ וכד'').

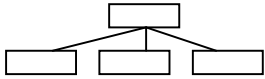
#### מודלים של בסיסי נתונים:

- 1) מודל היררכי
- 2) מודל רישתי
- 3) מודל יחסי/טבלאי

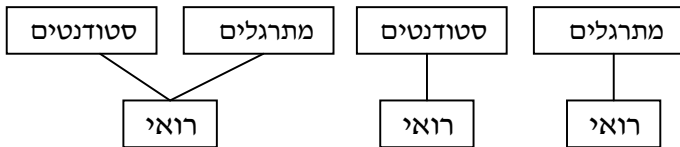
← נעבור על המודלים הנ"ל ונראה את היתרונות והחסרונות של כל אחד מהם, כאשר ניתן לשים לב שהסיבה להתפתחות מודל מתקדם בעל יתרונות מסויימים הינה בעקבות חסרונות שהועלו מהמודל שקדם לו.



1) המודל ההיררכי:



זהו מודל שבו הנתונים מסודרים במבנה של עץ, כאשר קודקודי העץ מציינים את הרשומות והחצים את הקשרים בין הרשומות.



כללים:

a. לכל (רשומה) בן יש אבא אחד

-----מותר----- אסור-----

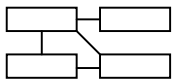
על פי מודל זה כאשר נוצר מצב שיש לנו מתרגל שהוא גם סטודנט אזי נצטרך כפילות (בזבוז).

- b. לכל אבא יש מספר כל שהוא (0..n) של בנים (מבנה של עץ רגיל) [שנקר ← מתרגלים, מרצים, סטודנטים]
- c. קשר אחד לרבים 1: N בין האב לבן.
- d. אסור שיהיו קשרים בין אחים.
- e. השאילתה היא בעצם מסלול עד לרשומה מסוימת בעץ.

בעיות במודל זה:

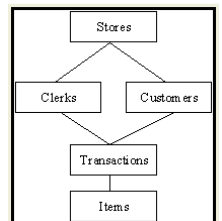
**כפילויות!** כמו כן, אי אפשר להוסיף רשומה לטבלת בן עד שזו לא הוכלה בטבלת האב, ז"א שאם לדוגמה רוצים להוסיף סטודנט שעדיין לא נרשם לאף קורס כמו כן מודל זה לא יודע לתאר קשרים מסוג רבים לרבים.

2) המודל הרשתי:



רשת של קודקודים וחצים ללא הגבלה, ז"א קשרים רב-רב-ערכיים M:N.

המודל הרשתי דומה מאוד למודל ההיררכי (תת קבוצה שלו) במקום שימוש בעץ שבו לכל קודקוד רק הורה אחד, המודל הרשתי מציע קבוצה המאוגדת במבנה דמוי עץ, עם שינוי אחד: לבן מותרים כמה אבות ← זאת אומרת העץ מוחלף בגרף. שינוי זה מאפשר למודל הרשתי לתמוך בקשרים מסוג רבים-לרבים.



בעיות במודל זה:

לאור העובדה שניתן לבצע קישור בין כולם לכולם, נוצר מצב של **מורכבות גבוהה ואי גמישות לשינויים** (מחיקת רשומה אחת תגרור המון פעולות), ולאור כך התחזוקה הינה קשה ומסובכת.



3) המודל היחסי / טבלאי (רלציוני מלשון relation):

בסיס הנתונים בנוי מטבלאות, כאשר הקשר בין הטבלאות יתבצע ע"י עמודה/ות משותפות.

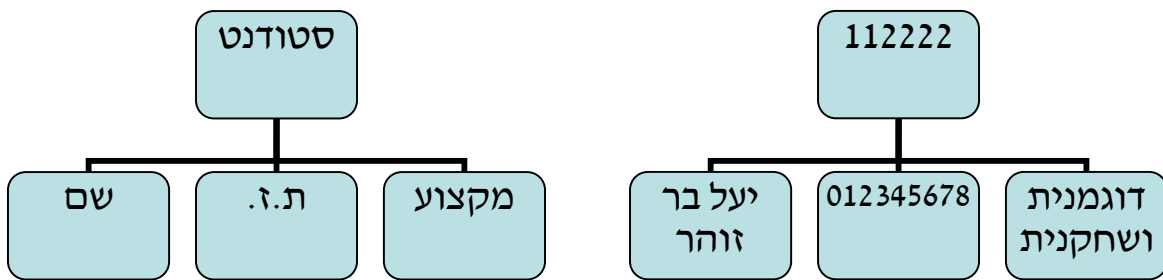
עקרונות:

- אפשר לתאר קשרים רב ערכיים.
- מספר הרשומות הינו בלתי מוגבל.

הגדרות:

- טבלה = יחס
- עמודה = תכונה
- **סכמה**: מבנה בסיס הנתונים, סדרה של תכונות המתארות את הפורמט / המבנה הלוגי של הנתונים.

דוגמא:



מבנה נתונים / סכמה מסויימת

מופע – תאור מבנה הנתונים ברגע מסויים

- **מפתח**: שדה או אוסף שדות המייצגים באופן חח"ע ומינימלי רשומה מסויימת (כמו ת.ז.).

הצגת רשומה בבסיס הנתונים וגישה לשדה בתוך הרשומה:

$t1 = \{Roei, Zerahia, 012345678, 01053081, 1000\}$

$t1[first\_name] = 'Roei'$

$t1[Last\_name] = 'Zerahia'$



[יתרונות המודל היחסי:](#)

כל טבלה מורכבת מרשומות (שורות אופקיות), ושדות (עמודות אנכיות).

כל טבלה מזוהה חד משמעית ע"י שם ייחודי, ושם זה משמש את בסיס הנתונים כדי לאתר את הטבלה. המשתמש על מנת להשתמש בטבלה צריך לדעת את שמה בלבד. השאלה כיצד המידע מאוחסן בפועל ברמה הפיסית לא מטרידה אותו, בזה מטפל ה DBMS.

חשוב להבין זהו זהו הבדל משמעותי לעומת המודל ההיררכי והרשתי, בהם המשתמש היה צריך לדעת כיצד הנתונים מסודרים בבסיס הנתונים, על מנת לעקוב אחרי המבנה ולהגיע לנתון אותו רוצים להוסיף, למחוק או לעדכן.

גישה לנתונים בשיטה זו הופכת את המודל היחסי להרבה יותר פשוט ויעיל מהקודמים לו, בגלל שהוא הרבה יותר קל להבנה.

יתרון נוסף של המודל היחסי בכך שהוא מספק כלים יעילים מאוד לניהול בסיס הנתונים. זאת משום שטבלאות יכולות להכיל לא רק את הנתונים המאוחסנים בפועל, אלא גם מידע לניהול בסיס הנתונים:

- מידע על הטבלאות
- שמות השדות שבבסיס הנתונים
- הרשאות גישה לטבלאות
- כללי הטיפול בנתונים כו'.

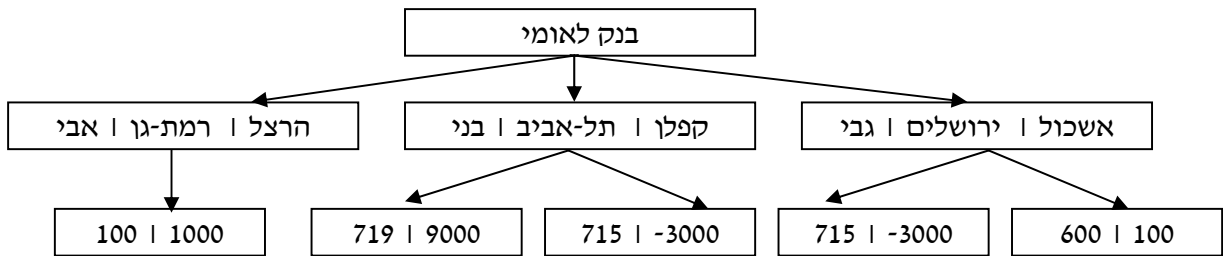
בעצם כל רכיב במודל היחסי יכול להיות מאוחסן בטבלאות.



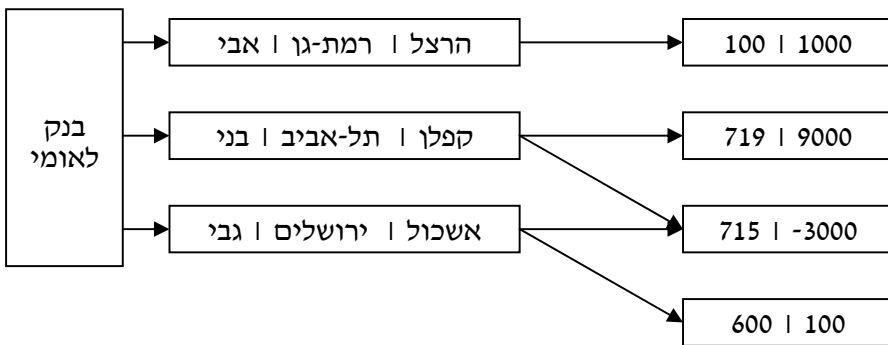
דוגמא מסכמת:

בנק שבו יש חשבונות של לקוחות כאשר עבור חשבון ידוע מספר החשבון והיתרה ועבור כל לקוח בבנק ידוע שמו והיכן הוא גר (עיר ורחוב).

מודל היררכי:



מודל רשתתי:



מודל טבלאי:

טבלת לקוחות			
שם	עיר	רחוב	מס' חשבון
אבי	רמת גן	הרצל	100
בני	תל אביב	קפלן	719
בני	תל אביב	קפלן	715
גבי	ירושלים	אשכול	715
גבי	ירושלים	אשכול	600

טבלת חשבונות	
מס' חשבון	יתרה
100	1000
719	9000
715	-3000
600	100