

בסיסי נתונים

מודל ישויות הקשרים

Entity Relation Model

תוכן עניינים

ER

דוגמא

הגדרות

Big data is like teenage sex:
everyone talks about it,
nobody really knows how to do it,
everyone thinks everyone else is
doing it, so everyone claims they
are doing it...

(Dan Ariely)

מודל ישויות-קשרים – (ER) Entity Relationship Model

מאפשר למדל מערכת מושגים ולבטא מספר מצומצם של אילוצים.
משמש בעיקר ככלי תכנון.

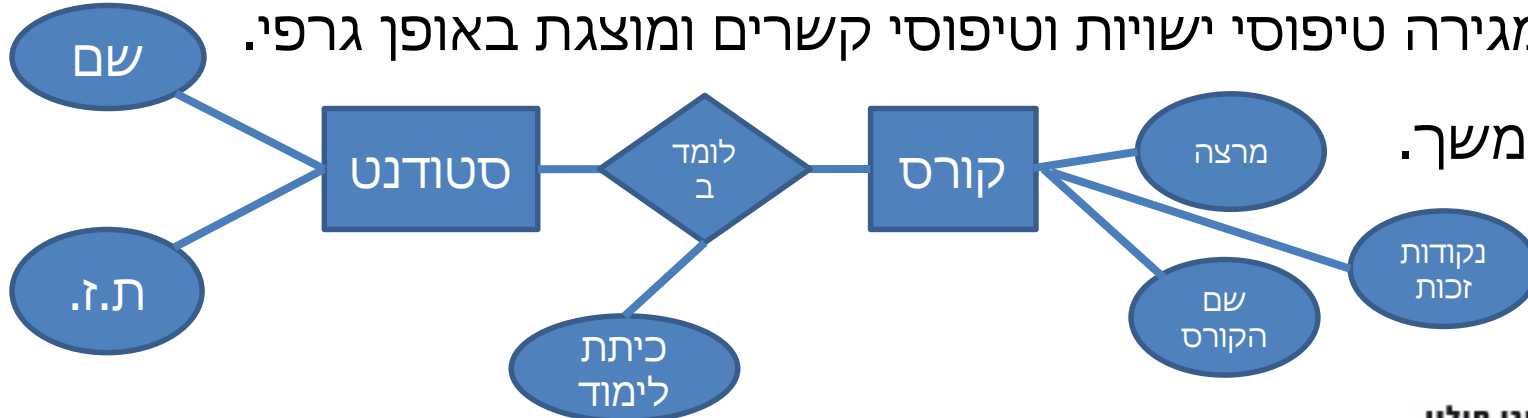
המודל מבחין בין:

ישויות וקשרים ביניהם.

לישויות ולקשרים יש **תכונות** שונות עם **ערכים** למופעים רגעיים.

התבנית מגירה טיפוס ישויות וטיפוסי קשרים ומוצגת באופן גרפי.

פירוט בהמשך.



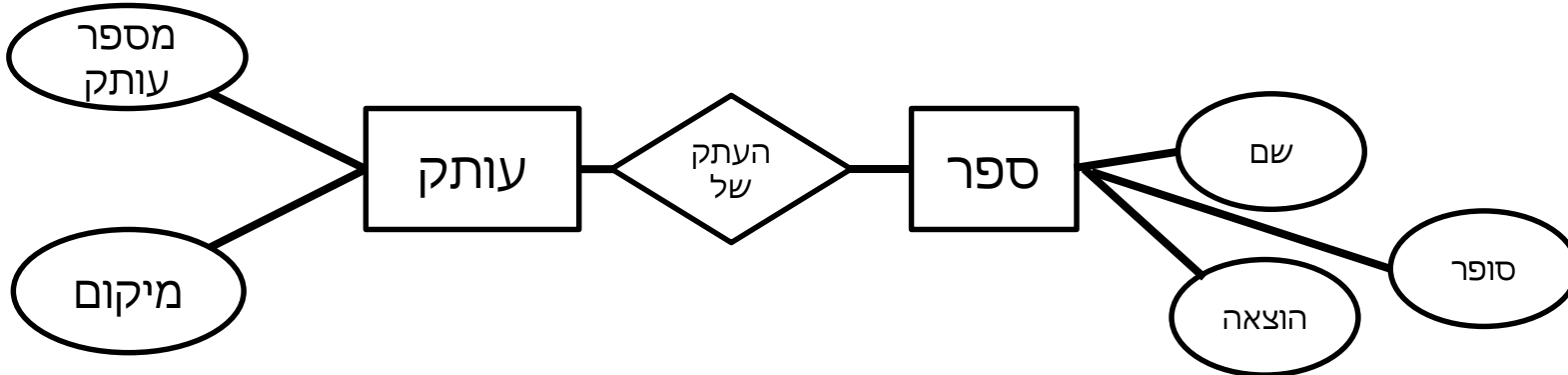
דוגמא – מערכת לניהול ספריה

טיפוס ישות – ספר, הוצאה, מספר עותק, לקוח, קטגוריה, מנוי, שפה.

דוגמא לטיפוס ישות - ספר.

דוגמא לטיפוס ישות - לקוח.

דוגמא לישות – כל ספר השייך לקבוצת הספרים הוא ישות.



הגדרות ER

ישות (Entity) – אובייקט שיש עליו מידע וניתן לאבחנה מאובייקטים אחרים. האבחנה נעשית באמצעות תכונות המתארות כל ישות. לדוגמא: ישראל ישראלי – הסטודנט, יש לו שם, טלפון ותכונות נוספות.

טיפוס ישויות (Entity Type/set) – קבוצה של ישויות מאותו סוג. לדוגמא: סטודנט, זהו טיפוס ☺ ויש כל מיני ישויות שלו.

תכונה של טיפוס ישויות (Attribute) – תכונה המתארת את הישויות השונות מטיפוס כלשהו. לדוגמא: שם פרטי או ת.ז. עבור הטיפוס סטודנט.

קבוצת התכונות של טיפוס ישויות – אוסף כל התכונות המתארות טיפוס ישויות מסוים. לדוגמא: לטיפוס ישות סטודנט: שם, ת.ז., טלפון, כתובת, תאריך לידה.

דוגמא – ספרייה – תכונות לכל ישות בסוגריים

לקוח – client (ת.ז., שם משפחה, שם פרטי, כתובת, טלפון, ת. לידה)

מנוי – subscriber (קוד מנוי, ת. תחילת מנוי, ת. סיום מנוי, מספר עותקים מקס. להשאלה)

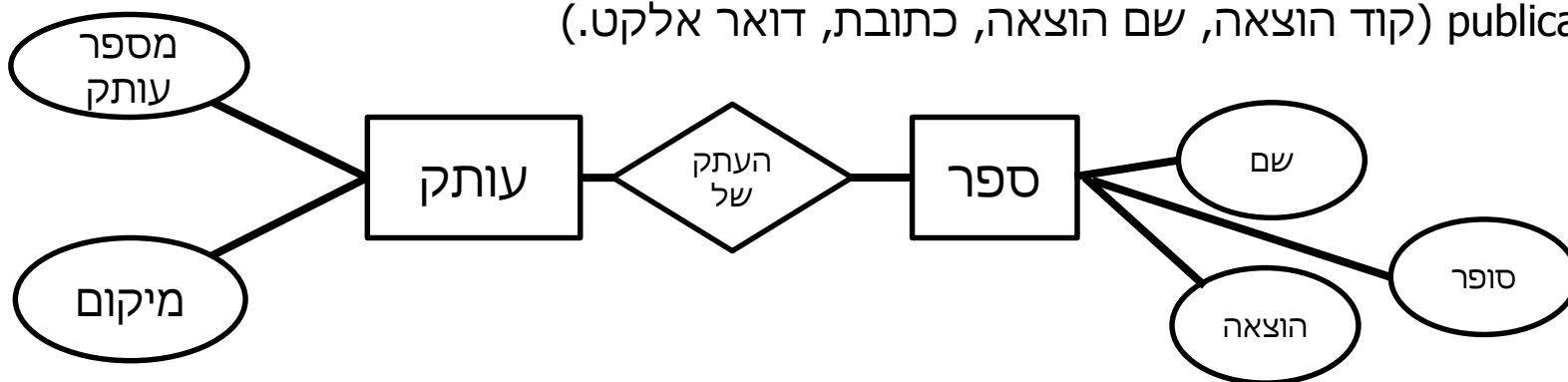
ספר – book (שם ספר, קוד ספר, שם סופר, שם מתרגם, ת. הוצאה, מספר עותקים)

עותק copy (מספר עותק, מיקום, מושאל)

קטגוריה category (קוד קטגוריה, שם קטגוריה).

שפה language (קוד שפה, שם שפה).

הוצאה publication (קוד הוצאה, שם הוצאה, כתובת, דואר אלקט.).



ER – מודל ישויות הקשרים

מה אומרים הציורים?

Entity

מלבנים – מייצגים טיפוסי ישות Entity

תכונה

אליפסות – מייצגות תכונות

לומד
ב

מעוינים – טיפוסי קשרים

קוים – מקשרים בין תכונה לישות שלה, או בין טיפוס קשרים לטיפוס ישות

שלו ובין טיפוסי ישות קשורים למשולש הפרדה

קווים עבים – קשר בין משולש ISA לטיפוס ישות

מלבנים כפולים – טיפוסי ישות חלשים נשלטים

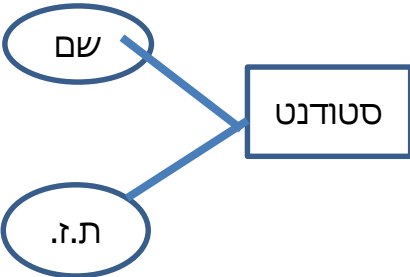
מעוינים כפולים – טיפוסי קשרים בין ישות נשלטת לשולטת או חלש וחזק

Entity

משולש ISA – ציון קשר הכללה או הפרדה

לומד
ב

ISI



הגדרות ER

ערך של תכונה (Attribute Value) – נתון המייחס ערך לתכונה של ישות מסויימת מטיפוס ישויות. לדוגמא הערך **אלון** לתכונה **שם** עבור טיפוס **סטודנט**.

תחום של תכונה (Attribute Domain) – קבוצת הערכים שתכונה יכולה לקבל. לדוגמא: גיל: מספר בין 0 ל 120.

תכונה של טיפוס קשרים relationship attribute – תכונה המאפיינת טיפוס קשרים, לא טיפוס ישות. לדוגמא: הקשר של סטודנט הלומד בקורס והתכונה "שעה".

תפקיד של טיפוס ישויות (Role) – טיפוס ישויות המשתתף בקשר ממלא בו תפקיד

דוגמא – ספריה – ערכים ותחומי תכונות

ערכים:

שם שפה – ערכים אפשריים: עברית, אנגלית, ספרדית, ...

שם משפחה – ערכים אפשריים: ישראלי, לוי, כהן, ...

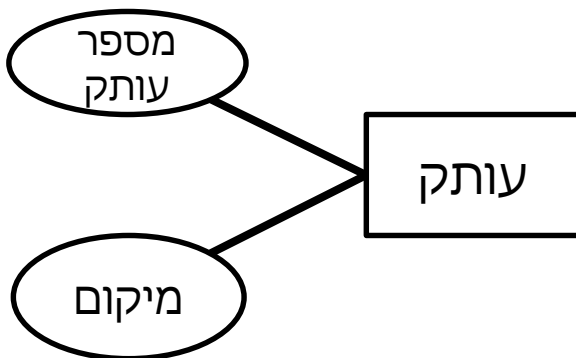
קוד ספר – 2304, 2400, ...

תחומי תכונות:

קוד שפה – קבוצת המספרים הטבעיים

ת. תחילת מנוי – תאריך

עותק – קבוצת המספרים השלמים החיוביים.



דוגמא – ספריה לטיפול קשרים

BookCopy – הקשר בין ספר והעותק שלו.

BookPublication – הקשר בין ספר וההוצאה שלו.

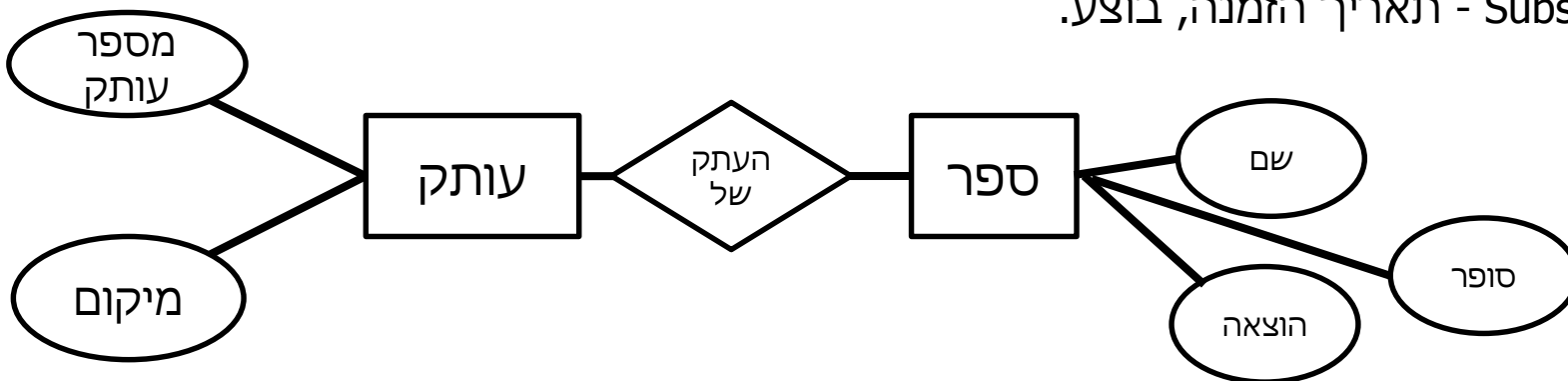
BookCategory – הקשר בין הספר לקטגוריה.

ClientSubscriber – הקשר בין מנוי ללקוח.

נוספים: SubscriberBook, SubscriberCopy, BookLanguage

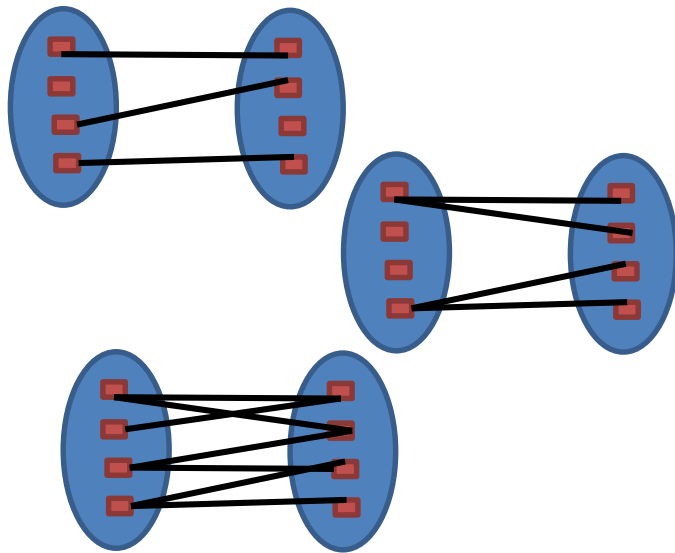
לטיפול הקשרים יתכנו תכונות: SubscriberCopy – תאריך השאלה, תאריך החזרה.

SubscriberBook – תאריך הזמנה, בוצע.



דרגת הקשר/רמת הקשר

מידת ריבוי mapping cardinality – מספר הישויות המשתתפות בקשר. הגבלה על מספר הישויות מטיפוס קישור אחד הקשורות למספר הישויות מטיפוס קישור אחר.



- אחד לאחד 1:1 one-to-one
- אחד לרבים 1:N one-to-many
- רבים לרבים M:N many-to-many

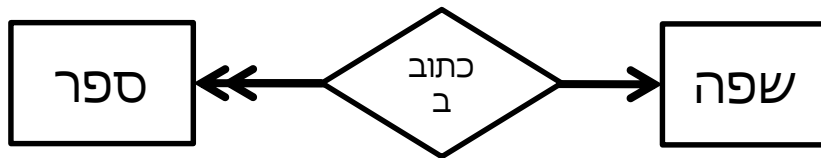
דוגמא – ספריה מידת הריבוי

BookLanguage – כל ספר כתוב בשפה אחת, אך בשפה מסויימת יש הרבה ספרים. כלומר מידת הריבוי של טיפוס הקשרים מספר לשפה הוא רבים-ליחיד משפה לספר הוא יחיד-לרבים.

BookCopy – הקשר בין ספר והעותק שלו, אחד-לרבים.

BookPublication – הקשר בין ספר וההוצאה שלו, אחד לרבים. הוצאה יכולה לפרסם הרבה ספרים, אבל לספר יש רק הוצאה אחת.

ClientSubscriber – הקשר בין מנוי ללקוח.



בהנחה שלמנוי יכולים להיות מספר לקוחות, למשל מנוי משפחתי. וללקוח מותר לעשות מספר מנויים. הקשר הוא רבים לרבים.

דוגמא – ספריה מידת הריבוי

BookCategory – לספר מספר קטגוריות, ולקטגוריה מספר ספרים. רבים-לרבים.

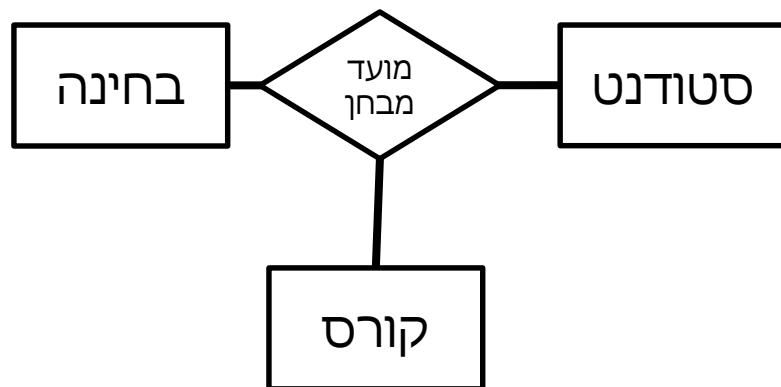
SubscriberCopy – מנוי יכול להשאיל מספר ספרים, אך עותק מסויים יכול להיות מושאל לאדם אחד. רבים-לאחד.

SubscriberBook – מנוי יכול להזמין מספר ספרים, וספר יכול להיות מזומן ע"י מספר מנויים. רבים-לרבים.

דרגת הקשר/רמת הקשר

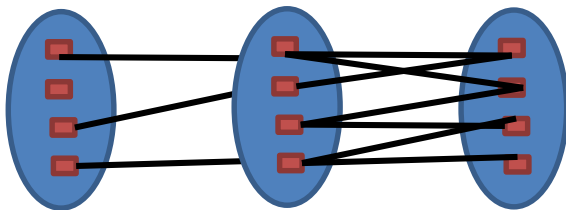
מידת ריבוי לטיפוס קשרים משולש - טרינרי

- אפשר לשלב את כל האפשרויות P:M:N כשכל אחד מהם יכול להיות רבים או יחיד.



- מה מספר האפשרויות? 2^3

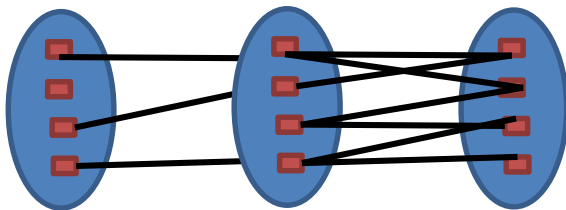
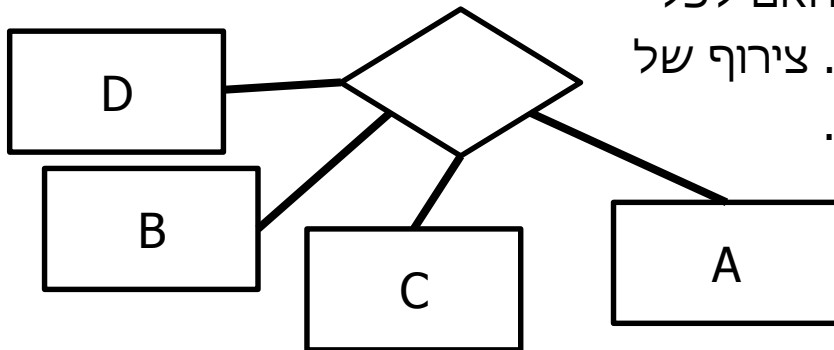
- דוגמא: קשר סטודנט-קורס-בחינה



דרגת הקשר/רמת הקשר

מידת ריבוי לטיפוס קשרים – כמה שתרצו

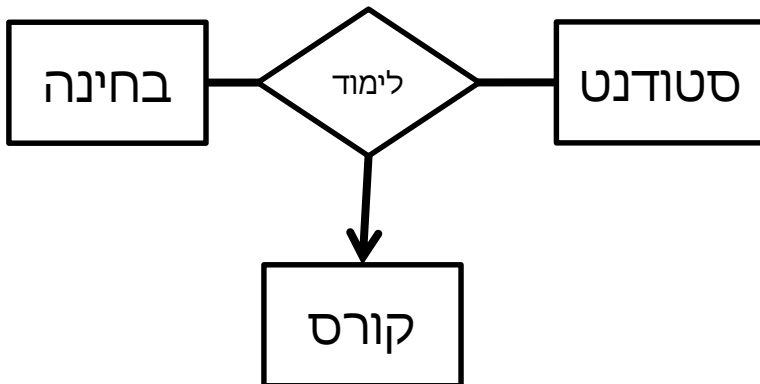
- אפשר לשלב יותר משלוש אפשרויות. נניח N ישויות.
- מסתכלים על קבוצה, ללא הישות המדוברת, וחושבים על הריבוי בין הקבוצה ללא הישות, לישות. האם לכל מופע בקבוצה יש מופע אחד בישות, או יותר. צירוף של $n-1$ ישויות יכול להיות קשור לטיפוס n אחד.



- סימון:
- חץ יחיד – מופע אחד.
- חץ כפול – רבים (גם ללא חץ בחלק מהשיטות)
- דוגמא: קשר סטודנט-קורס-בחינה-חדר

דרגת הקשר/רמת הקשר

מידת ריבוי לטיפול קשרים – כמה שתרצו



- נניח טיפוס לימוד - סטודנט יכול להיבחן באותו קורס כמה פעמים. בבחינה יכולים להיות כמה סטודנטים. סטודנט לא יכול להיבחן בבחינה אחת בקורסים שונים. קורס הוא אחד, הקשר לימוד בין קורס-סטודנט-בחינה הוא אחד-רבים-רבים.

טיפוסי קשרים

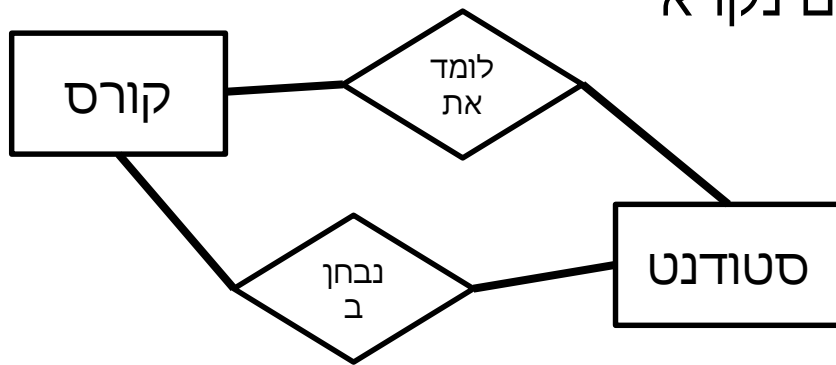
יתכנו שני טיפוסים קשרים שונים בין אותם
טיפוסי ישויות.

מספר הישויות המרכיבות טיפוס קשרים נקרא
הדרגה.

בינארי – דרגה 2

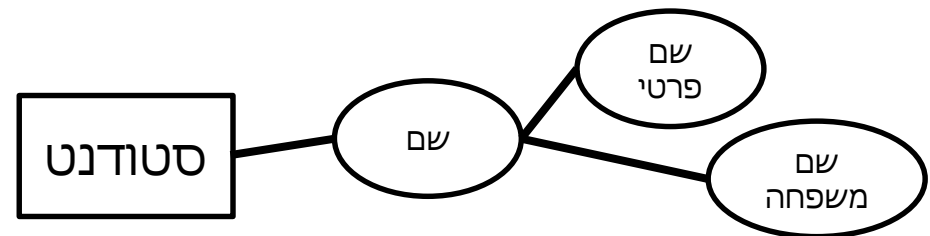
טרינרי – דרגה 3

וכן הלאה.



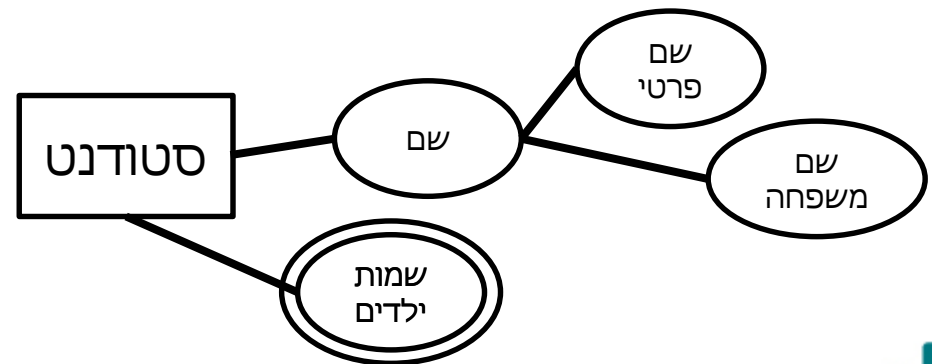
תכונה מורכבת

Composite attribute – תכונה שכל ערך אפשרי שלה הוא צירוף של שני ערכים, אך לפעמים נרצה להגיע לאחד הערכים.
למשל: שם – מורכב משם פרטי ושם משפחה. תאריך – מורכב מיום, חודש, שנה.



תכונה מרובת ערכים

Multi-valued attribute – תכונה שכל ערך אפשרי שלה עבור ישות מסויימת יכול להיות קבוצה שלמה של ערכים.
מסומן ע"י אליפסה עם קו כפול.
למשל: שמות ילדים – יתכן אחד או יותר.



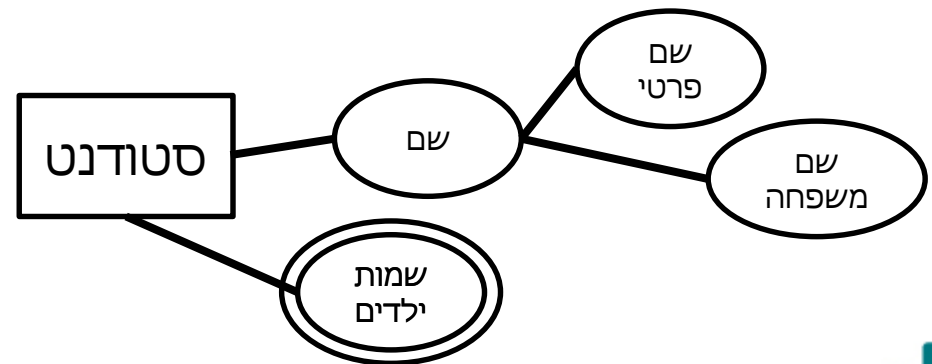
תכונה ריקה

Null attribute – אם לא לכל הישויות יש ערך לתכונה מסויימת או היא אינה רלוונטית לחלקם, נסמן אותה כריקה.

למשל: שמות ילדים – יתכן גם שהערך ריק לחלק מהסטודנטים.

ערך ריק נקרא Null value, ותתכן ישות עבודה התכונה ריקה.

אין ביטוי בדיאגרמה, אך חשוב לתכנון.

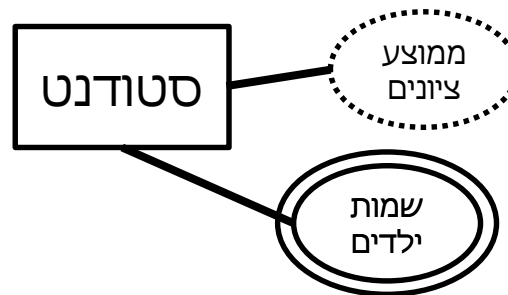


תכונה מחושבת

derived attribute – תכונה שניתן להסיק את ערכה לפי ערכיהן של תכונות אחרות.

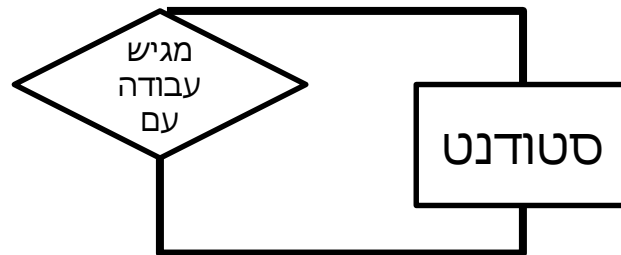
מסומנת בקו מקווקו.

למשל: ממוצע ציונים.



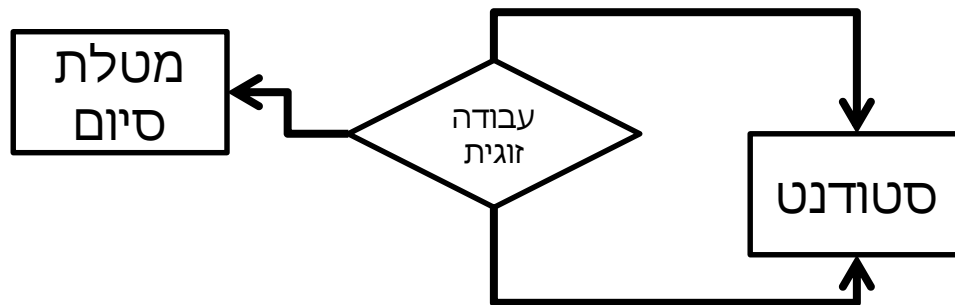
טיפוס קשרים רקורסיבי

recursive relationship – בין טיפוס קשרים לבין עצמו.

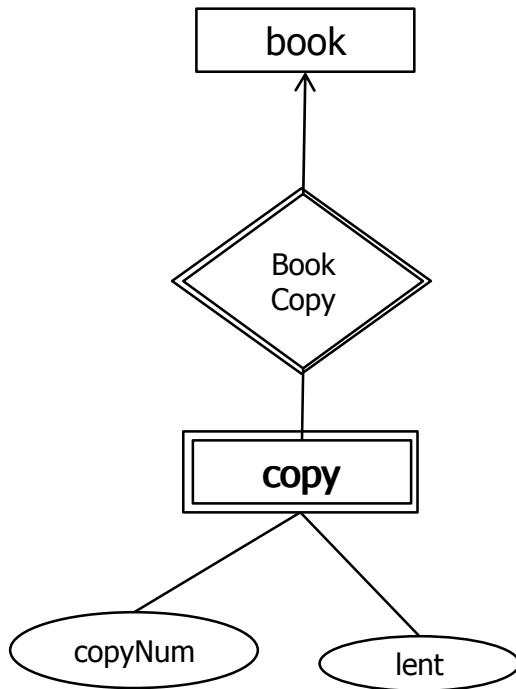


טיפוס קשרים רקורסיבי

recursive relationship – בין טיפוס קשרים לבין עצמו. אפשר גם:



תלות קיום בן טיפוסים ישויות



טיפוס ישויות X תלוי **תלות קיום** (existence dependency) בטיפוס ישויות Y, אם כל ישות מטיפוס ישויות X תלויה בקיומן של ישויות מטיפוס Y.

X נקרא **טיפוס ישויות נשלט** (subordinate).

Y נקרא **טיפוס ישויות שולט** (dominant).

הטיפוס **הנשלט** מוצג בדיאגרמה ע"י מלבן כפול. הקשר ע"י מעויין כפול.

אם מוחקים את Y, אז גם X נמחק.

האילוץ לא נראה בדיאגרמה. וכן יש קשר למידת הריבוי.

ספריה דוגמא - תלות קיום בן טיפוס ישויות

BookCopy – תלות קיום מריבוי רבים-לאחד מהטיפוס הנשלט (עותק) לטיפוס השולט (ספר). העותק תלוי בספר, אם מבטלים ספר אז מבטלים גם את העותקים.

ClientSubscriber – תלות קיום מריבוי רבים-לרבים מהמנוי (הנשלט) ללקוח (השולט). בהנחה שמנוי (משפחתי) יכול להיות שייך למספר לקוחות, וכן יתכן לקוח עם מספר מנויים. אם מבטלים את כל הלקוחות (השולט) בעלי מנוי מסוים, יש לבטל את המנוי, אך אם מבטלים חלק מהלקוחות, אין לבטל המנוי.

תלות זיהוי

כאשר יש תכונות של ישות אחרת שנדרשות לצורך זיהוי.

לדוגמא: BookCopy - עותק תלוי תלות זיהוי בספר. כי כדי להבחין בין עותקים יש להשתמש בתכונות של ספר. פרטים בנושא מפתחות. נבחין בין תלות קיום ותלות זיהוי. אם יש לנשלט מפתח ראשי משלו, אין תלות זיהוי, רק תלות קיום. למשל subscriber יש תלות קיום ב client, אבל יש לו מזהה משל עצמו subscriberId, כך שאין תלות זיהוי.

קביעת מפתחות

מפתח-על superkey - צירוף תכונות המאפשר להבחין בין ישויות שונות של אותו טיפוס ישויות או מופעים שונים של אותו טיפוס קשרים.

מפתח קביל (מפתח מינימלי) candidate key - מפתח על שאין בו תכונות מיותרות. אין קבוצת תכונות חלקית שהיא בעצמה מפתח-על.

מפתח ראשי primary key – מפתח קביל כלשהו שנבחר להיות המפתח העיקרי.

***מפתחות ראשיים מסמנים עם קו תחתי.**

ספריה דוגמא - קביעת מפתחות

לקוח client (ת.ז., שם משפחה, שם פרטי, כתובת, טלפון, שנת לידה)

מנוי subscriber (קוד מנוי, ת. תחילת מנוי, ת. סיום מנוי, מספר עותקים מקסימלי להשאלה)

ספר book (קוד ספר, שם ספר, שם מתרגם, ת. הוצאה, מספר עותקים במלאי)

קטגוריה category (קוד קטגוריה, שם קטגוריה)

שפה language (קוד שפה, שם שפה)

הוצאה publication (קוד הוצאה, שם הוצאה, כתובת, טלפון, דוא"ל)

ספריה דוגמא – בחירת מפתח ראשי

לקוח client (ת.ז., שם משפחה, שם פרטי, כתובת, טלפון, שנת לידה)

ת.ז. הוא מפתח קביל

אבל גם

{שם משפחה, שם פרטי, כתובת} יחד הם מפתח קביל.

לעומת זאת

{שם משפחה, כתובת} – לא קביל

{שם פרטי, שם משפחה} – לא קביל

{ת.ז., שם משפחה} – מפתח על, אבל לא קביל / מינימלי

טיפוסי ישויות חזק-חלש

טיפוס ישויות חזק strong entity type – כזה שניתן להגדיר לו מפתח ראשי.

טיפוס ישויות חלש weak entity type – כזה שלא ניתן להגדיר לו מפתח ראשי. הוא יהיה גם תמיד טיפוס ישויות נשלט.

מזהה/מבדיל discriminator - קבוצת תכונות **מינימלית** של טיפוס ישויות חלש המבחינה בין הישויות התלויות באותו טיפוס ישויות חזק. למשל עותק של ספר, נדרש מזהה משלו.

מפתח ראשי של טיפוס ישויות חלש – את המפתח הראשי יוצרים מצירוף המפתח הראשי של טיפוס-הישויות החזק השולט בו, אל המזהה של טיפוס הישויות החלש.

ספריה דוגמא - טיפוס ישויות חלש

עותק copy (קוד ספר, מספר עותק, מיקום, מושאל)

עותק מציין עותק מספר בספריה.

המזהה שלו הוא מספר עותק, כי לפיו מבחינים בין הישויות השונות, העותקים השונים.
הוא טיפוס ישויות חלש, לא קיים צירוף תכונות שיכול לשמש בו מפתח ראשי.
יתכנו שני ספרים עם אותו מספר עותק (לדוגמא 2), אך כל אחד שייך לספר אחר.
לכן יוצרים מפתח ראשי מצירוף של טיפוס הישויות החזק השולט בו – ספר, אל
המזהה שלו – מספר עותק.

קביעת מפתחות

מפתח ראשי של טיפוס-קשרים – יהיה איחוד המפתחות הראשיים של טיפוס הישויות המגדירים אותו.

*בין טיפוס חלש וטיפוס חזק, המפתח הראשי של טיפוס הקשרים יהיה זהה למפתח הראשי של הטיפוס החלש.

פורמלית

לקשר בינארי R, בין טיפוס E1 ו E2. צירוף המפתחות הראשיים של E1-E2 הוא מפתח-על. אם מידת הריבוי היא רבים-לרבים, הצירוף יהיה גם מפתח קביל.

אבל אם מידת הריבוי היא רבים-לאחד, הרי כל ישות מ E1 קשורה לכל היותר לישות אחת מ E2, כך שהמפתח של E1 הוא מפתח קביל של טיפוס הקשרים R.

* טיפוס בריבוי אחד, המפתח שלו לא רלוונטי למפתח טיפוס הקשרים.

ספריה דוגמא – מפתחות של טיפוסים קשרים

BookLanguage (קוד שפה, קוד ספר)

מפתח ראשי מורכב מאיחוד המפתחות הראשיים של טיפוסים הישות
המגדירים את הקשר. קוד שפה – מפתח ראשי של שפה, קוד ספר –
מפתח ראשי של ספר.

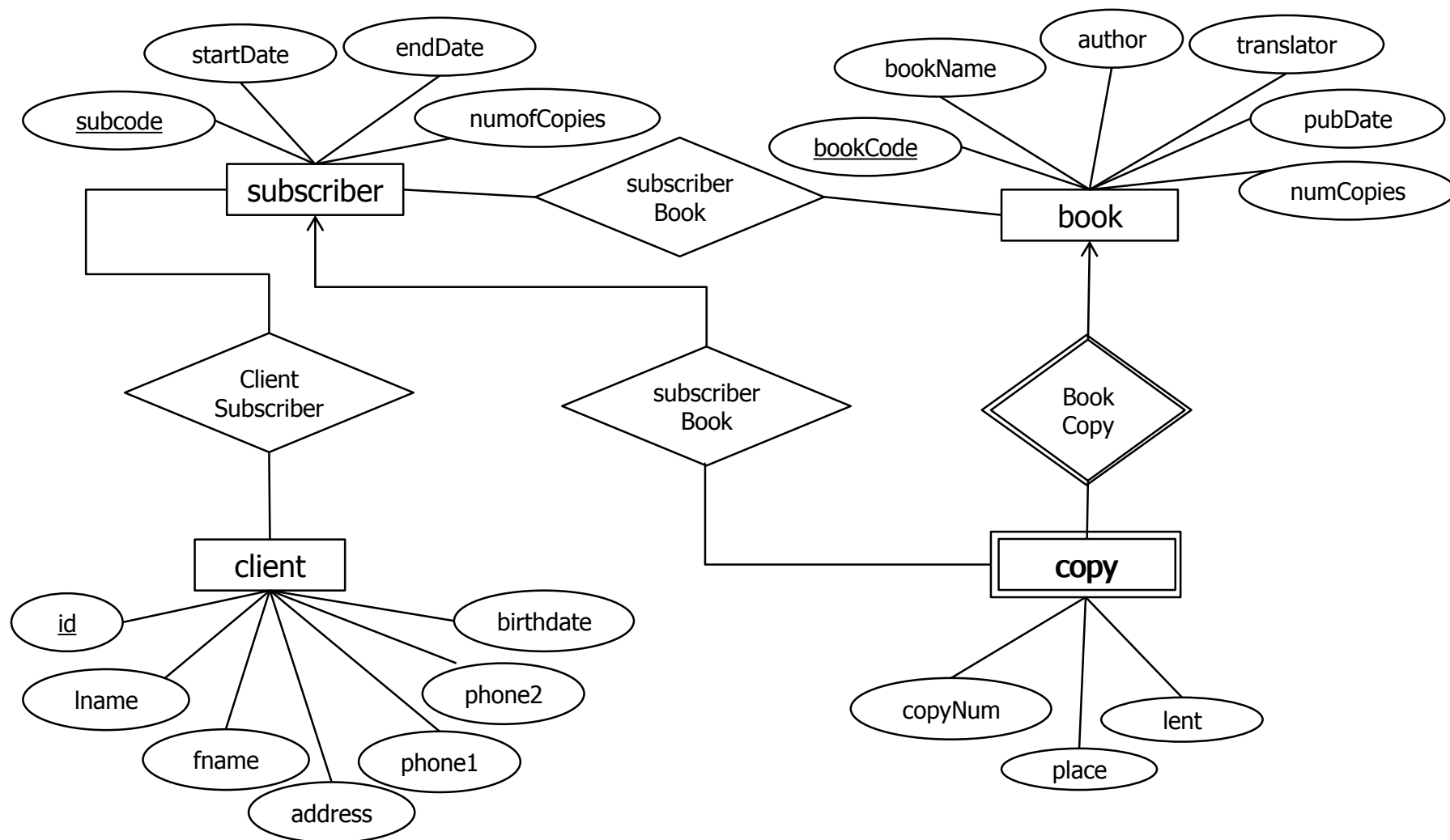
BookCopy (קוד ספר, קוד עותק)

זהה למפתח הראשי של טיפוס הישויות החלש – .copy

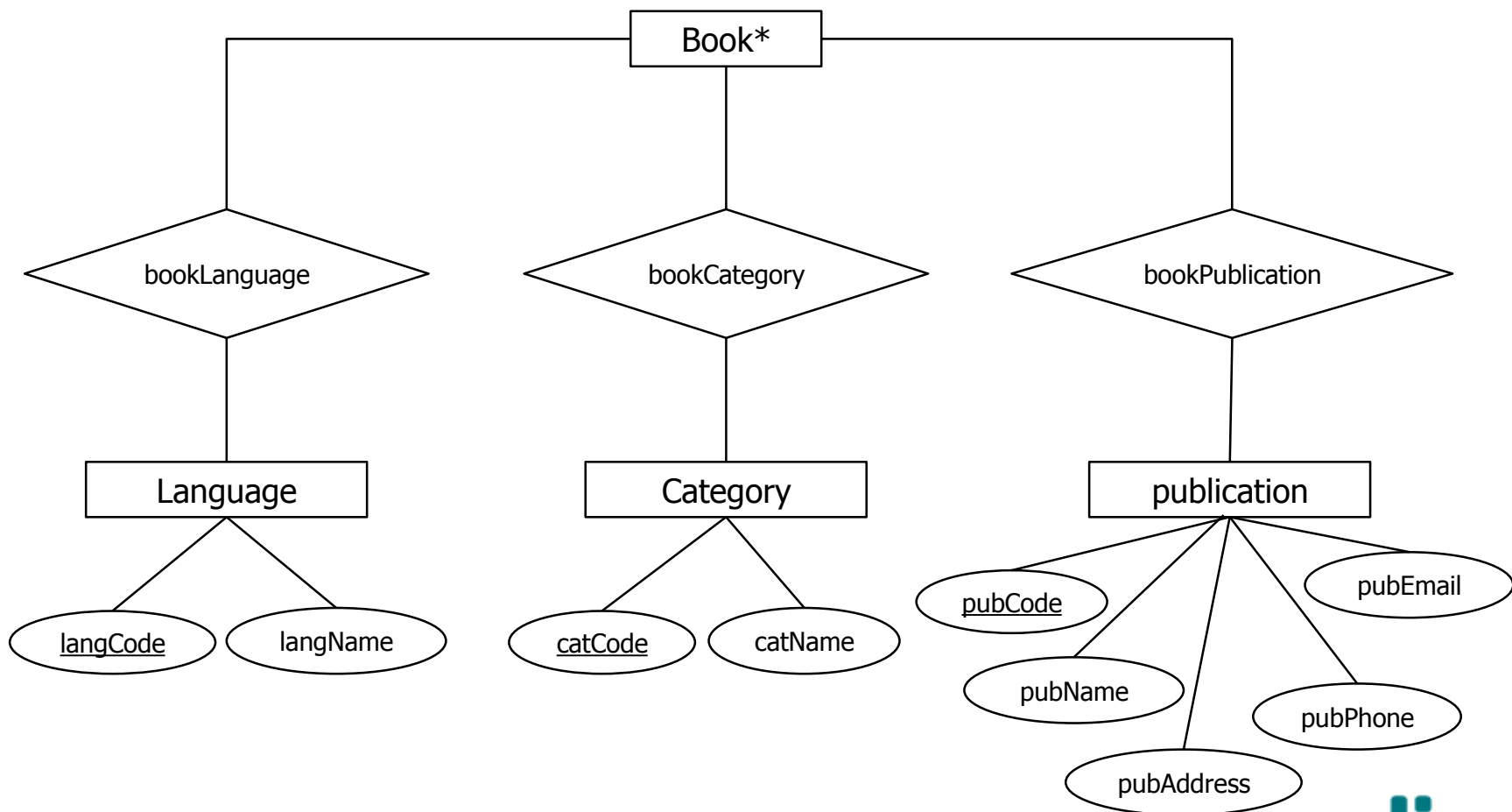
BookPublication (קוד ספר, קוד הוצאה)

BookCategory (קוד ספר, קוד קטגוריה)

דיאגרמה מלאה - ספרייה



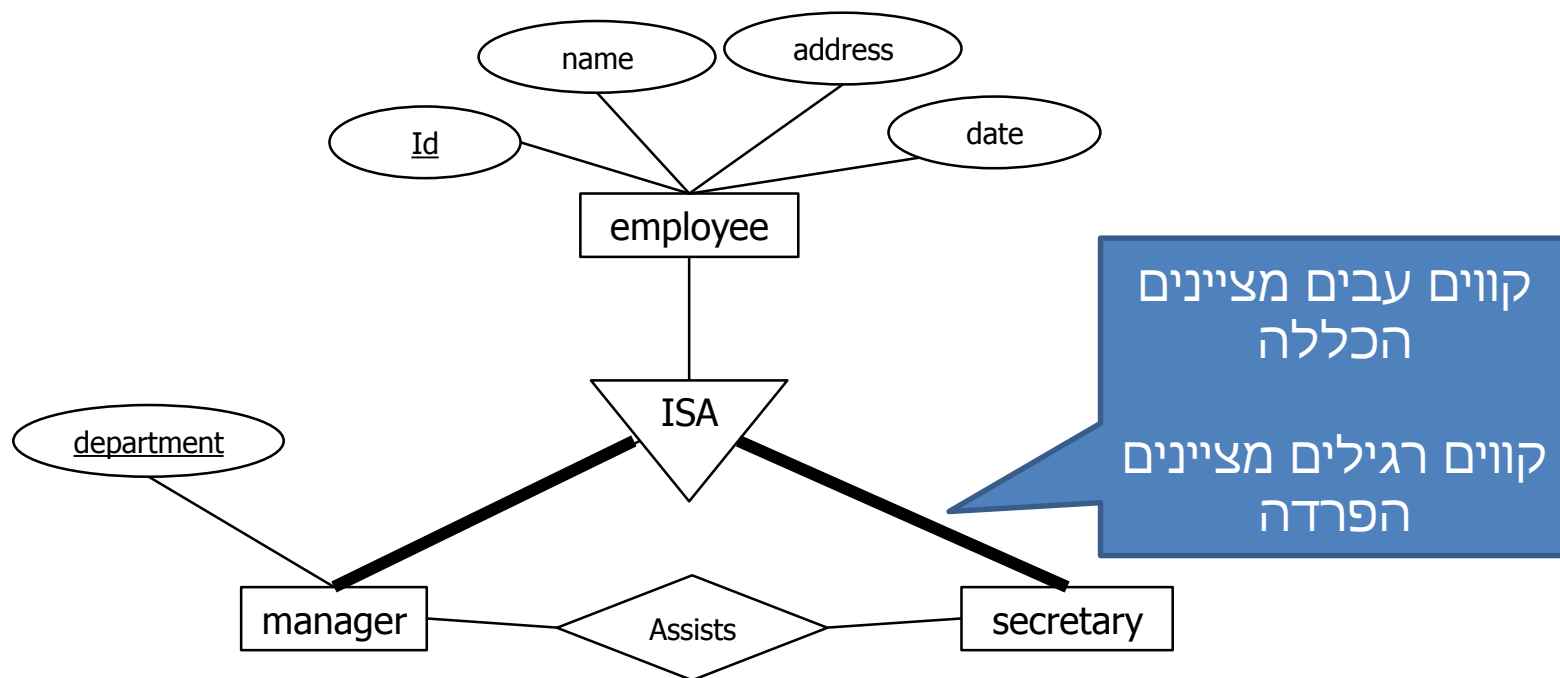
דיאגרמה מלאה - ספרייה



יחס ISA בין ישויות

- יחס ISA – "משהו הוא" מבטא קשר בין ישויות. ניתן להבחין בין שני מקרים:
 - **הכללה Generalization** - איחוד מספר טיפוסים ישויות לטיפוס ישות מרמה גבוהה יותר. הדגש הוא על קווי הדמיון בין הישויות. לדוגמא: מנהל ומזכירה הם עובדים.
 - **צמצום, הפרדה Specialization** - הפיכת תת-קבוצה של ישויות מרמה גבוהה לקבוצת ישויות מרמה נמוכה. הדגש הוא על ההבדלים. לדוגמא: מנהל הוא עובד.
 - לאיזה צורך נציין קשר זה:
בהכללה, הורשת התכניות לישות מהרמה הנמוכה. מדגישה את קווי הדמיון (בצמצום זה לא הכרחי, ויתכן שיש עוד ישויות).

דוגמא - הכללה



הכללה – כל עובד הוא מנהל או מזכירה. הפרדה - יתכנו עוד סוגי עובדים.

אילוצי שלמות completeness

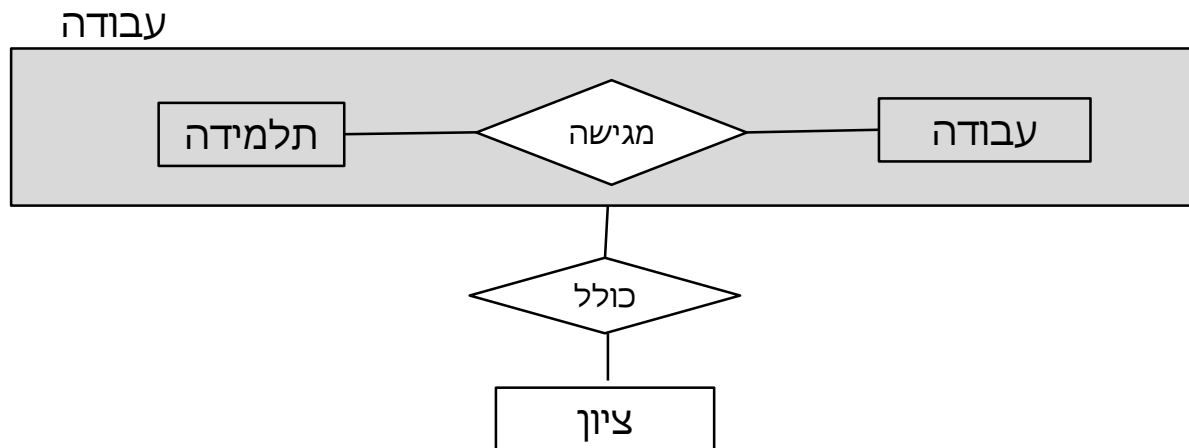
אילוץ שלמות **מלא** – כל ישות ברמה הגבוהה, היא מופע של אחד הישויות ברמה הנמוכה (הכללה).

אילוץ שלמות **חלקי** – יש ישויות ברמה הגבוהה, שאין להן ביטוי ברמה הנמוכה (הפרדה).

הכללה יכולה לאפשר קבוצות זרות disjoint, או קבוצות חופפות overlapping.

הקבצה - Aggregation

הפשטה שמאפשרת היררכיה, ושימוש בביטוי קשרים בתוך קשר.
לדוגמא:



עקרונות תיכון

לייצג את כל המידע.

לא לייצג עודף מידע.

שיהיה פשוט.

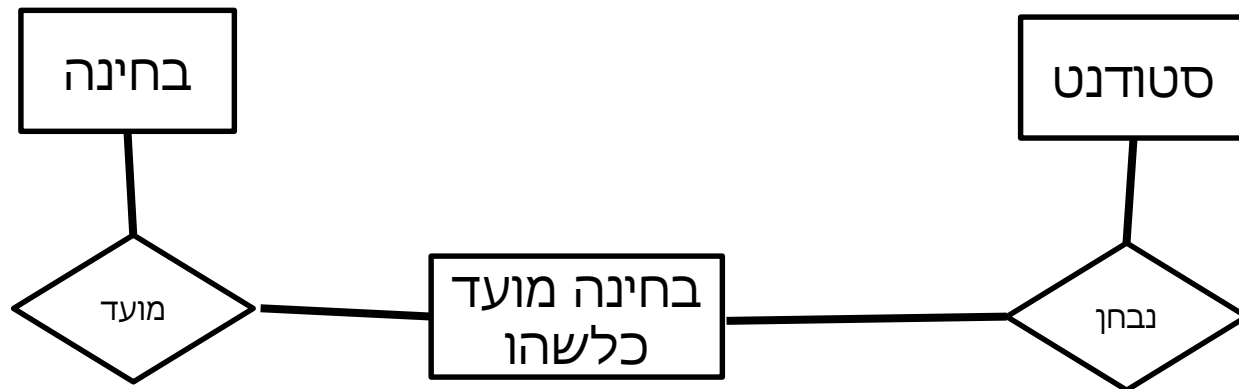
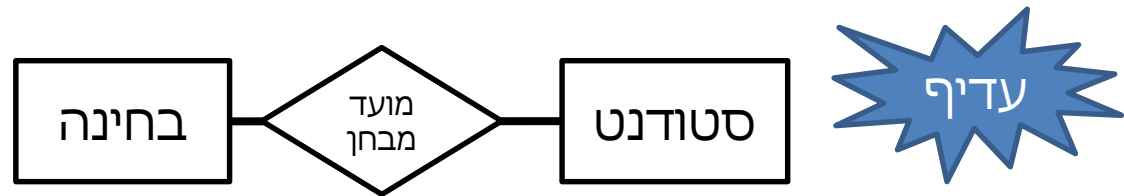
- נאמנות למציאות

- רלבנטיות למערכת.

- מניעת כפילויות. לא לשכפל, אם צריך להוסיף קשרים. הבעיות:

מקום, אבל יותר חשוב מידע לא עקבי שצריך לעדכן בכמה מקומות.

פשוט זה עדיף



אילוצים

כללים החלים על טיפוסים ישויות וקשרים. בין היתר אילו צירופים אינם אפשריים.

שאלות כמו: האם סטודנט יכול לקחת כמה קורסים?

האם ניתן פעמיים את אותו קורס?

האם יתכן קורס ללא סטודנטים?

לא ניתן לבטא כל אילוץ ה ER. מה ניתן?

ריבוי, השתתפות, תלות קיום

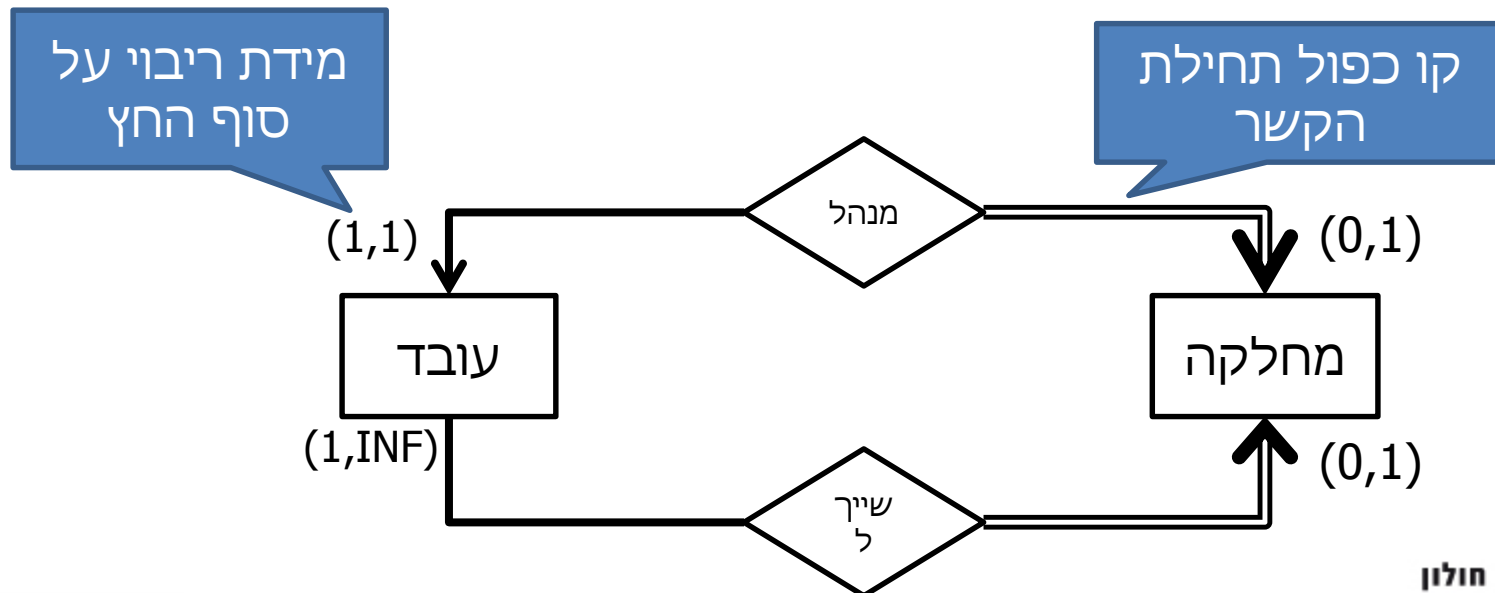
אילוצי השתתפות

אילוצי השתתפות participation constraints – ניתן לבטא ישירות בדיאגרמת הקשרים.

אם כל ישות חייבת להשתתף בקשר – **השתתפות מלאה** (קו כפול בתחילת הקשר)

אם אין חובה כזאת – **השתתפות חלקית** (קו רגיל בתחילת הקשר).

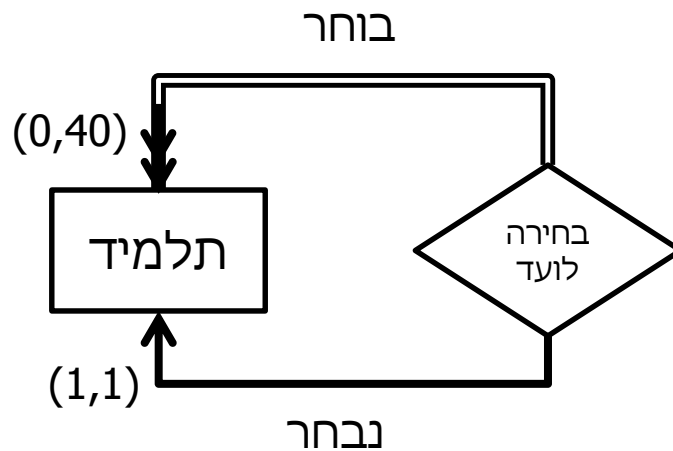
אפשר לרשום ליד סוף החץ של הקשר את טווח ההגבלות על מידת הריבוי, ואז אם יש במינימום 1 או יותר זאת השתתפות מלאה, 0 זאת השתתפות חלקית. במקסימום לכל היותר אחד, זה ריבוי יחיד.



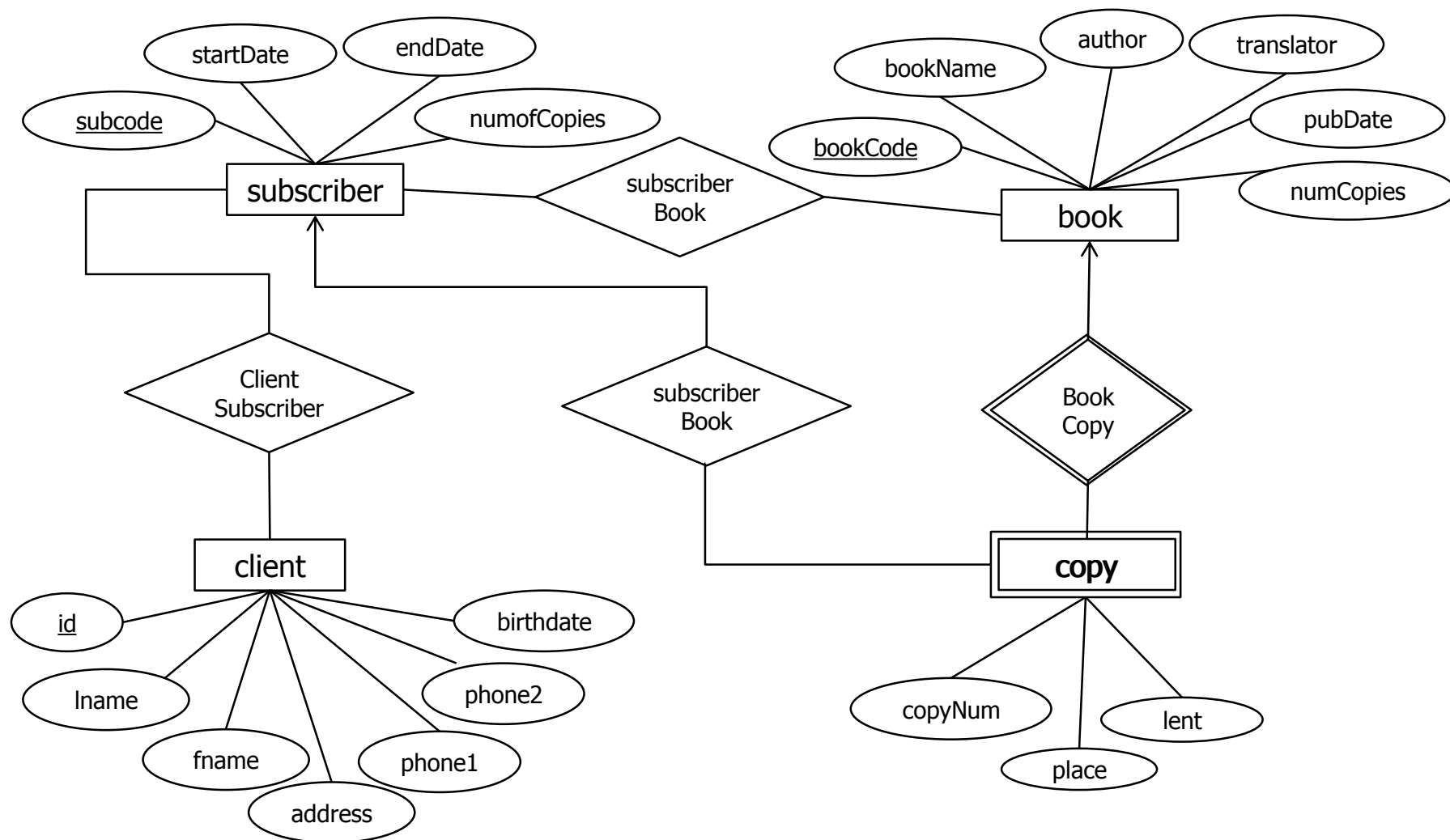
אילוצי השתתפות

אם מציינים את התפקיד של ישות בקשר להבהרה, למשל בקשר רקורסיבי, התפקיד בקשר מצויין על תחילת החץ, כמו גם אילוץ ההשתתפות.

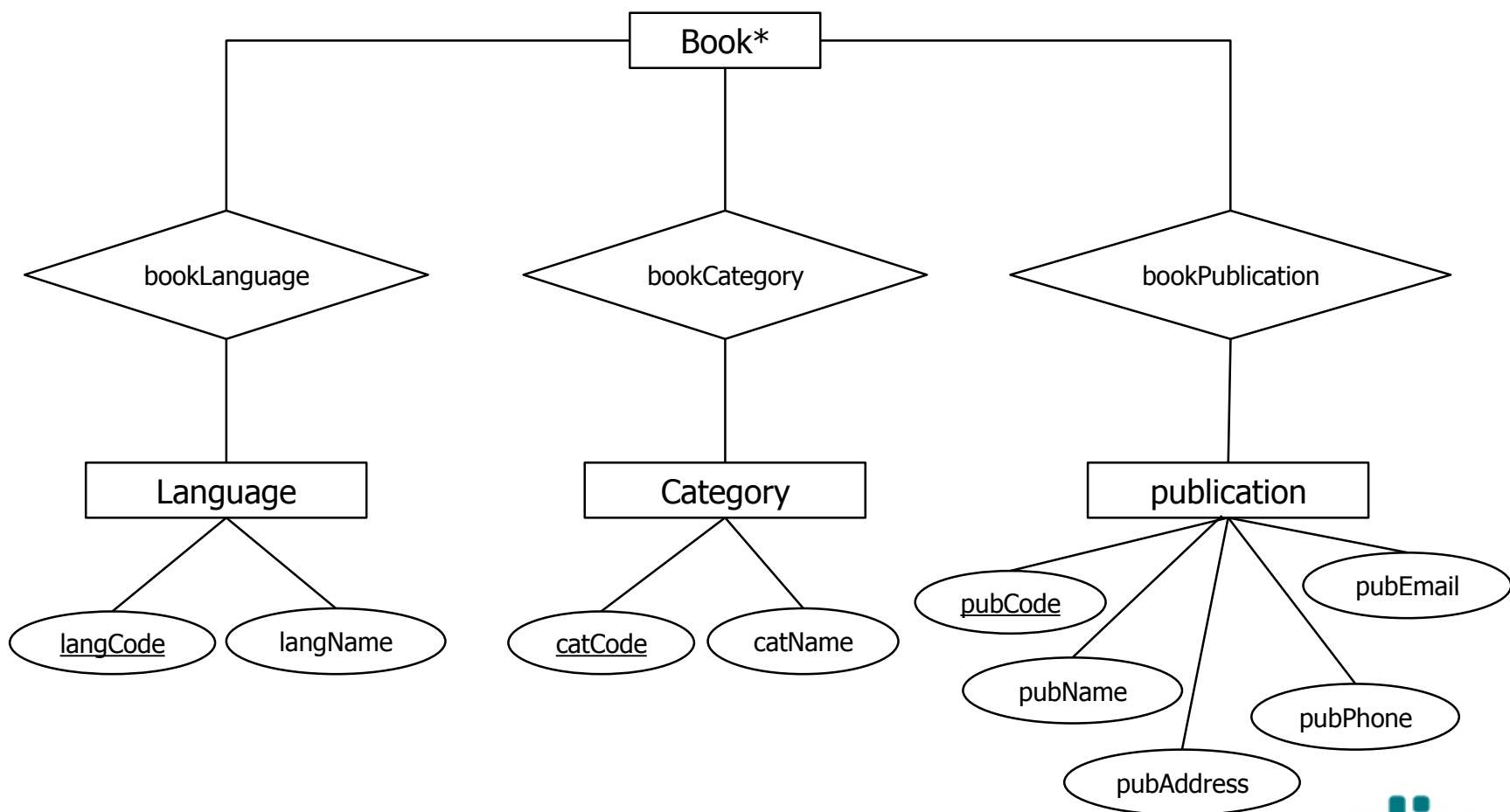
לדוגמא: תלמיד חייב לבחור בתלמיד אחד לכל היותר. כל תלמיד יכול להיבחר (או שלא) ע"י כל אחד מ 40 התלמידים בכיתה.



דיאגרמה מלאה - ספריה



תרגום לטבלאות



תרגום דיאגרמות לטבלאות

Language(LangCode, LangName)

BookLanguage(BookCode, LangCode)

Publication(PubCode, pubName, pubAddress, pubPhone, pubEmail)

BookPublication(PubCode, BookCode)

Copy(BookCopy, CopyNum, place, lent)

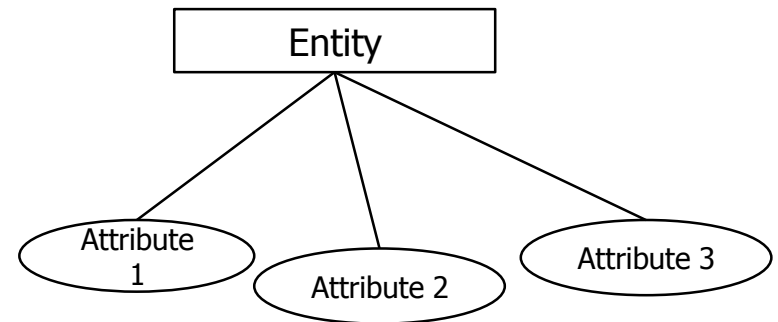
....

נדגים המרה לפי כללים

כלל א': לכל טיפוס ישויות חזק נגדיר טבלה נפרדת עם עמודה נפרדת לכל אחת מתכונות טיפוס הישויות.

Entity

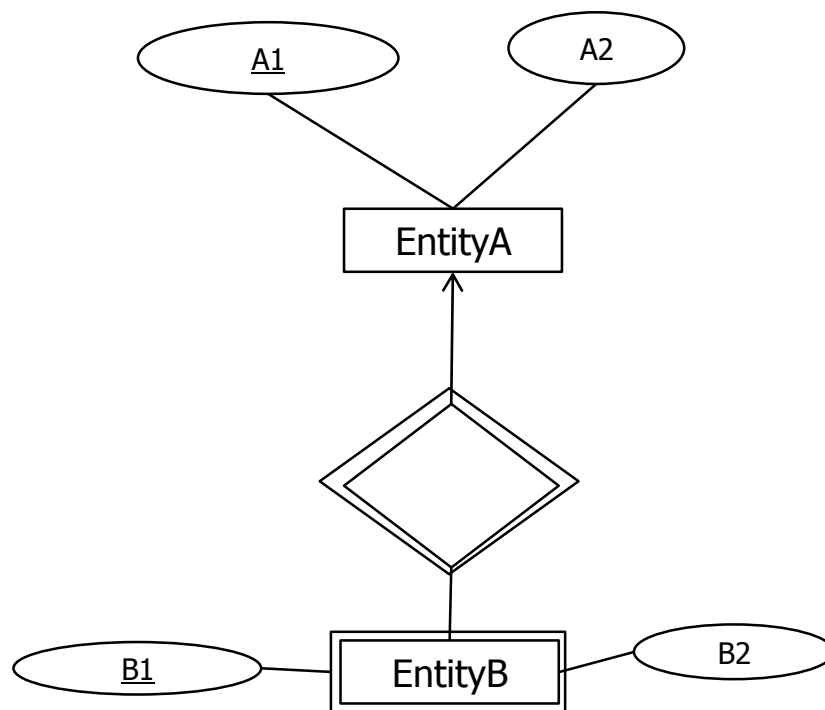
Attribute 3	Attribute 2	Attribute 1



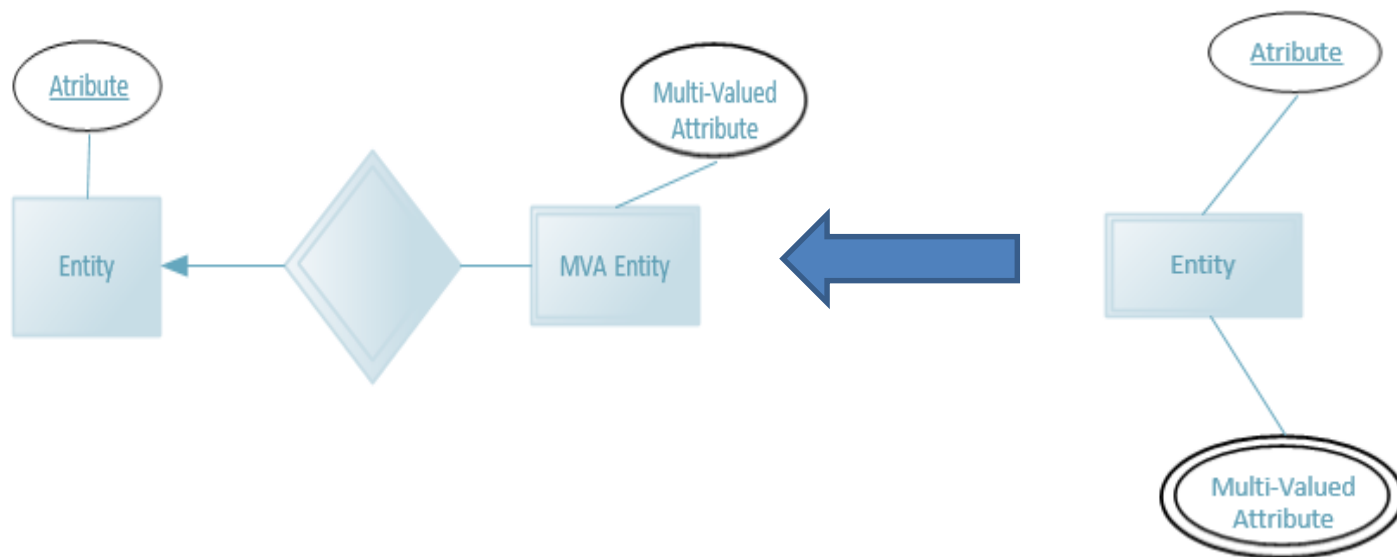
כלל ב': לכל טיפוס ישויות חלש נבנה טבלה עם עמודה לכל תכונה בתוספת עמודה עם תכונת המפתח של הטיפוס החזק.

EntityB

A1	B1	B2



מעבר מייצוג של תכונה מרובת-ערכים לטיפוס קשרים אחד-לרבים



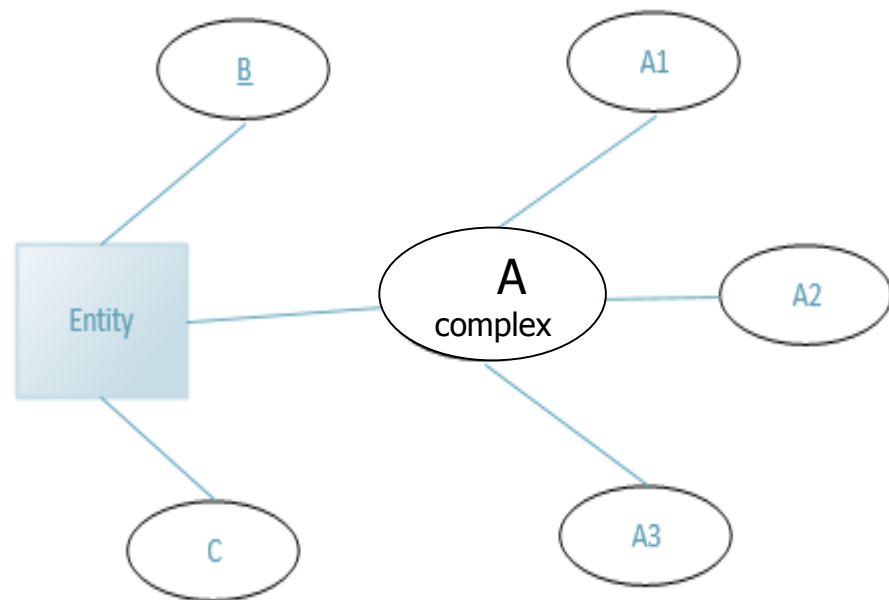
כלל ג': לכל תכונה מרובת ערכים נבנה טבלה נפרדת עם עמודה לכל אחת מתכונות המפתח של הטיפוס.

Entity		Entity MVA	
Attribute		Attribute	Multi-Valued Attribute

כלל ד': לתכונה מורכבת נגדיר עמודה לכל אחת מהתכונות.

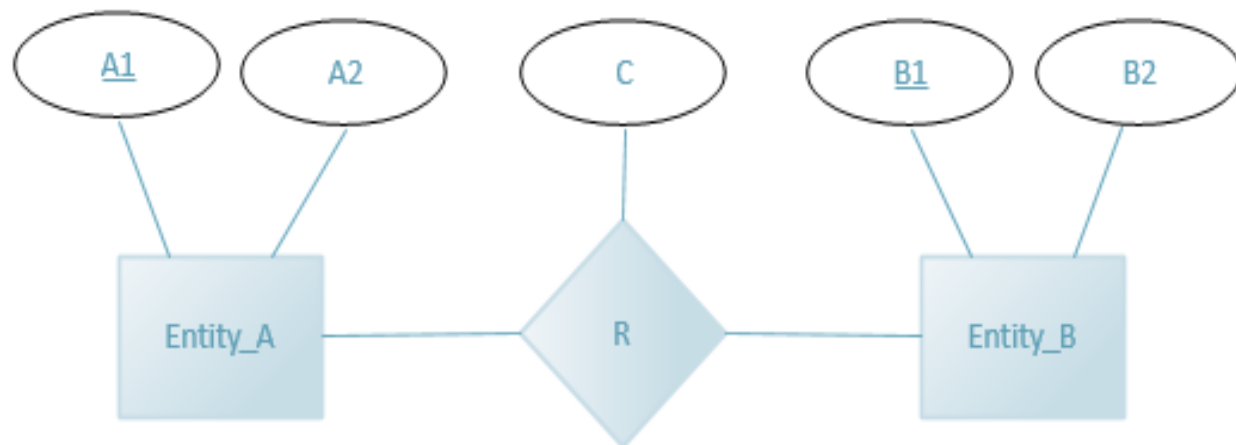
Entity

B	A1	A2	A3	C

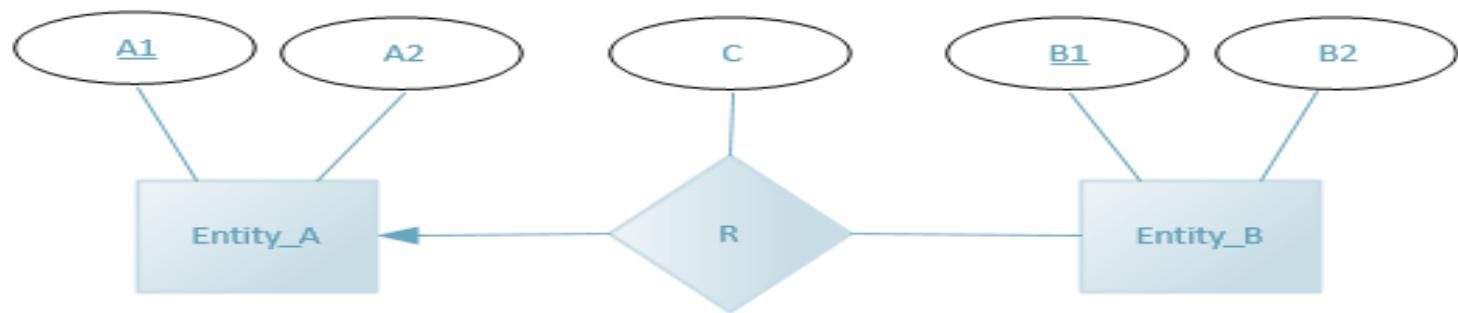


כלל ה': לכל טיפוס קשרים רבים לרבים נבנה טבלה נפרדת עם עמודה לכל תכונת מפתח של המשתתפים בקשר ועמודה לכל תכונה מהקשר.

R		
A1	B1	C



כאשר מידת הריבוי של טיפוס קשרים היא רבים לאחד, אחד לרבים או אחד לאחד, יש כמה אפשרויות להמרה.
 אם נבצע את ההמרה כמו עבור טיפוס קשרים רבים-לרבים על-פי הכלל לעיל, נקבל את הטבלאות הבאות:



EntityA

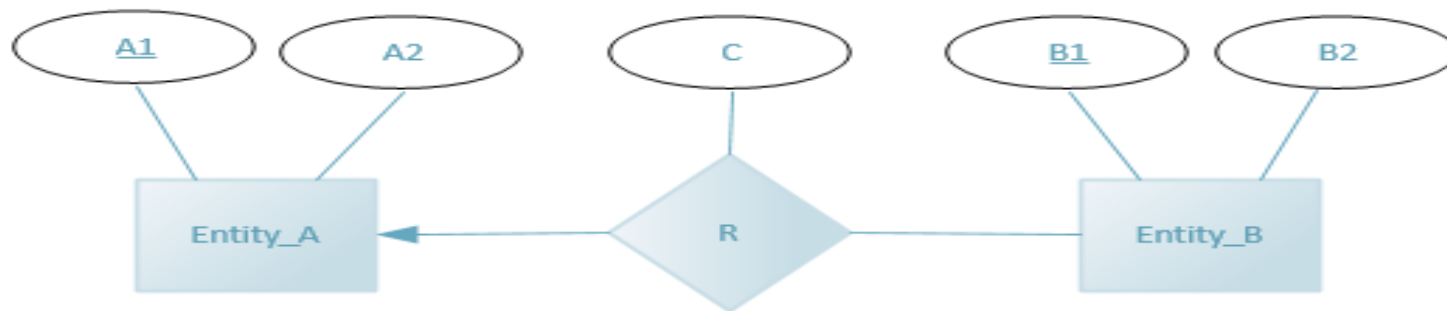
A1	A2

R

A1	B1	C

EntityB

B1	B2



EntityA

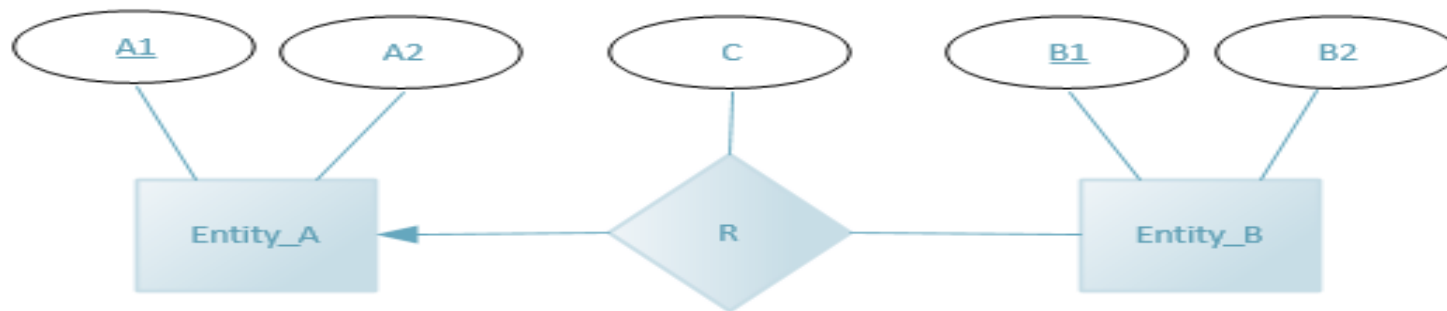
A1	A2

R

<u>A1</u>	<u>B1</u>	C
aa	bb	g
ab	bb	h
	bb	g

EntityB

B1	B2



EntityA

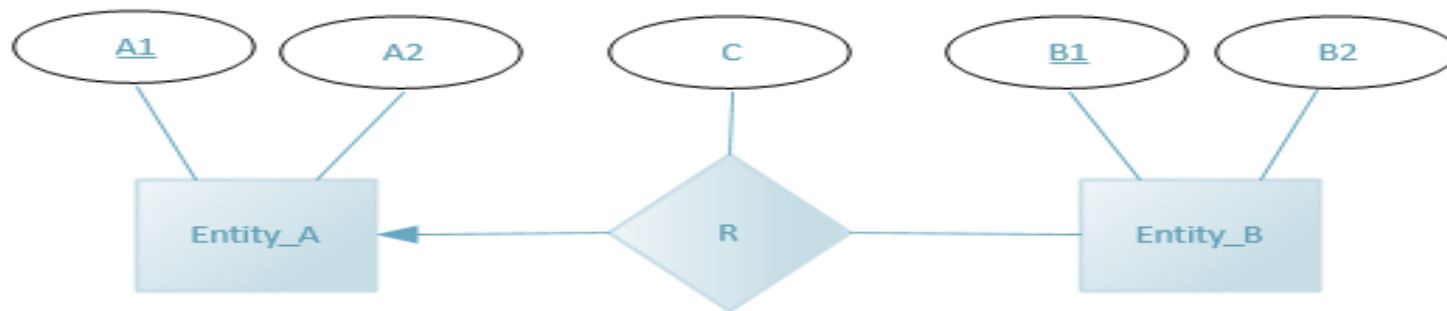
A1	A2

R

<u>A1</u>	<u>B1</u>	<u>C</u>
aa	bb	g
ab	cc	h
ac	bb	g

EntityB

B1	B2



EntityA

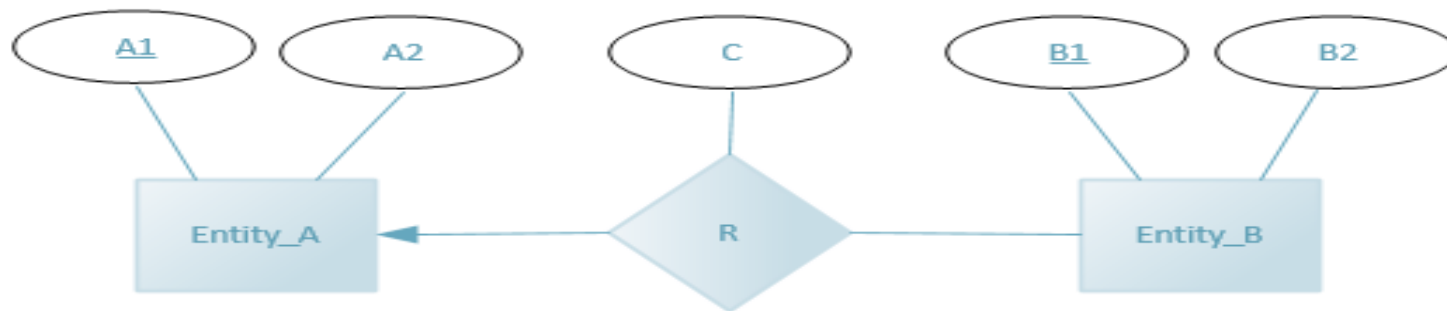
A1	A2

R

<u>A1</u>	<u>B1</u>	C
aa	bb	g
ab	cc	h
ac	bb	g

EntityB

B1	B2



EntityA

A1	A2

R

<u>A1</u>	<u>B1</u>	C	<u>B1</u>	B2
aa	bb	g	bb	
ab	cc	h	Cc	
ac	ff	g	ff	

EntityB

כיוון שמידת הריבוי של R הוא רבים-לאחד, B1 מספיק כמפתח קביל של R.

כלומר, לשתי הטבלאות, R ו- E2 יש אותו מפתח. לכן יש אפשרות לאחד את שניהם ולקבל המרה אחרת:

EntityA		EntityB			
A1	A2	<u>B1</u>	B2	A1	C

כלל ו': עבור טיפוס קשרים רבים-לאחד אפשר שלא ליצור
טבלה נפרדת, אלא להוסיף את כל תכונותיו לטבלה
המייצגת הישויות המשתתף כ'רבים', וכן את תכונות המפתח
של טיפוס הישויות המשתתף כ'יחיד'.

בהמרה של יחס ISA – יצוג של הכללה או הפרדה

אפשר:

1. ליצר רק את הרמה הגבוהה – איחוד התכונות כשחלקן

NULL

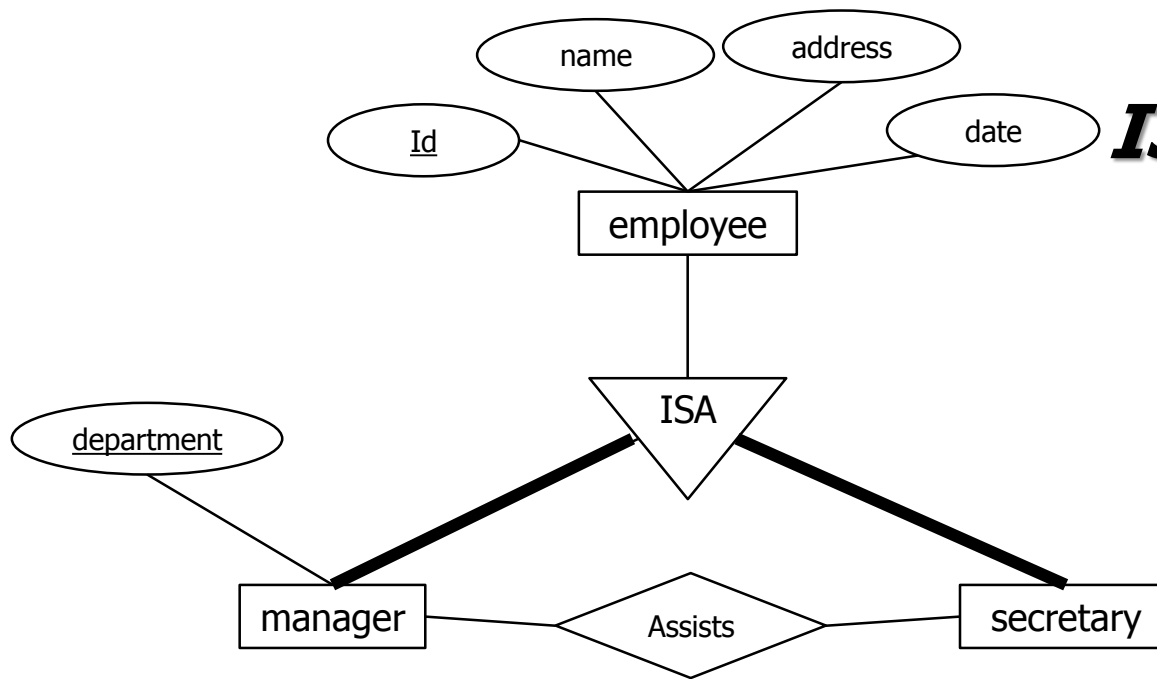
2. לייצר את הרמות הנמוכות – היורשות את הרמה

הגבוהה, כשאין בהן חפיפה

3. להמיר את שתי הרמות לחלשות, כשהן יורשות רק את

המפתח

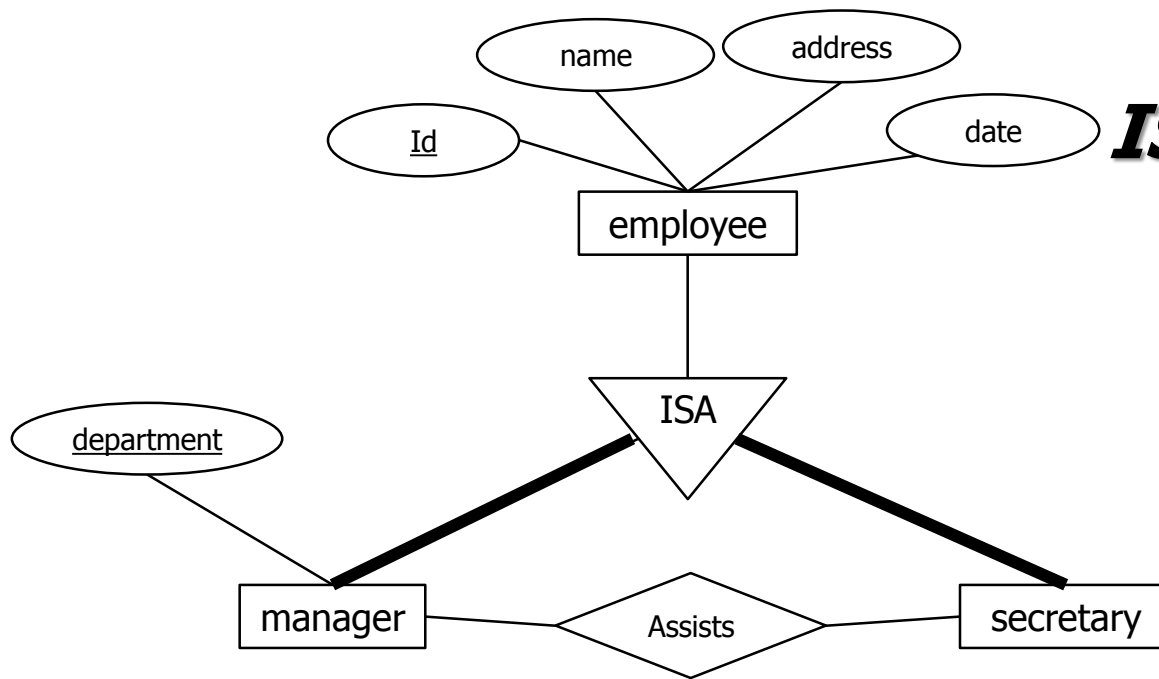
דוגמא – המרה ISA



כלל 1

Employee(Id, name, address, date, m_department, isSecretary)

דוגמא – המרה ISA

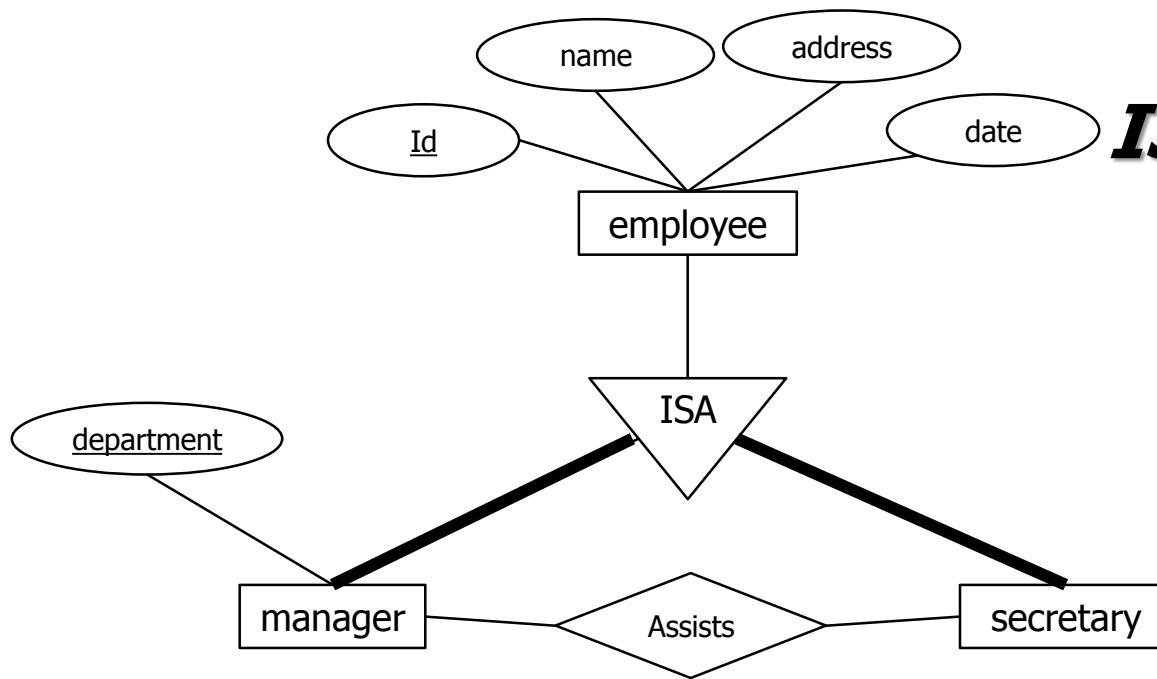


כלל 2

Secretary(Id, name, address, date)

Manager(Id, name, address, date, department)

דוגמא – המרה ISA



כלל 3

Employee(Id, name, address, date)

Secretary(Id)

Manager(Id, department)

בהמרה של קשר - סיכום:

1. רבים לרבים – נקבל טבלה עבור הקשר שמכילה מפתח של כל ישות בקשר, לפחות.
2. רבים ליחיד – ניתן לוותר על הטבלה עבור הקשר, ולהוסיף את היחיד למפתח המתאים בטבלת הרבים.
3. יחיד ליחיד – כמעט ולא קורה, ולא דורש טבלה.