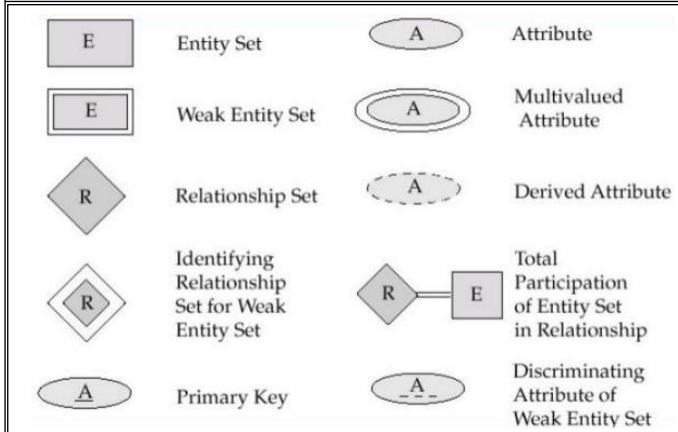


ER diagram: בדיוק 1 נסמן בחץ רגיל, לפחות 1 ולא יותר מ1 נסמן בחץ מעוגל מיקום תכונות 1:1 נשים בצד אחד שיותר מתאים, 1: N נשים בצד של הרבים, M: N שמים את התכונה על הקשר עצמו השתתפות מלאה נסמן ב2 קווים, בישות חלשה תמיד יש השתתפות מלאה כי אחרת היא לא קיימת isa המפתח של השורש הוא גם של התתי מחלקות disjoint השתייכות רק לתת מחלקה אחת נסמן עם d overlapping אפשר להשתייך לתת מחלקה 1 או יותר(אין סימון מיוחד-ברירת מחדל)



מפתחות מפתח על sk מכיל מפתח קביל מפתח על מינימלי כזה שלא ניתן להוריד ממנו תכונה אחרת לא יהיה מפתח, מפתח קביל מפתח על מינימלי מפתח ראשי מפתח על מינימלי שמייצג את הרלציה pk מפתח זר מפתח שמתייחס למפתח של יחס אחר ויכול להיות חלק ממפתח של טבלה, אינו חלק ממפתח של טבלה ובעצמו להיות מפתח של טבלה **בעיות שלמות בעת ביטול שינוי** כאשר משנים משהו אצל האב וישלו רשומות תואמות בבן restricted קודם נשנה/נבטל במפתח הזר אצל הבן ואז אצל האב בהתאם nullify נאפס את ערכי המפתח הזר אצל הבן רק במידה וזה לא חלק מקא וככה כן כאשר אצל האב, cascade נשנה/נבטל אצל הבן בהתאם(שרשור) set-default נאשר ונכניס ערך ברירת מחדל למפתח הזר אצל הבן **אילווצי entity** שום תכונה של מפתח ראשי לא יכולה להיות null **אילווצי referential** כל שורה שמתייחסת לטבלה אחרת תמיד מתייחסת למופע קיים

המרה למודל הרלציוני 1 לכל ישות חזקה נבנה טבלה, תכונה מורכבת תהיה כמו סט תכונות 2 לכל ישות חלשה נייצר טבלה ונוסיף גם כמפתח זר את המפתח הראשי של הישות החזקה שעליה היא נשענת 3 מיפוי יחסים בינאריים 1:1 שמים בצד אחד של היחס ולוקחים את המפתח של הישות שאליה לא שייכנו כמפתח זר 4 מיפוי יחסים בינאריים 1:N שמים בצד הרבים ולוקחים את המפתח של הצד ה-1 ומכניסים אותו כמפתח זר ברבים 5 מיפוי יחסים בינאריים N:M נייצר טבלה שתכיל את המפתחות של ה-2 רלציות ותכונות של היחס עצמו 6 כל תכונה מרובת ערכים תקבל טבלה 7 מיפוי יחסים N-ary כמו ב-m:n 8 מיפוי הכללה והפרדה 8A= מגדירים טבלה לכולם 8B= מגדירים טבלה רק לתתי מחלקות שייכלו את השדות של האב והמפתח טוב עבור מקרים של השתתפות מלאה 8C= עושים טבלה אחת ומוגדר שם טייפ שמאפשר להגדיר תת מחלקות, הרבה null יותר טוב עבור disjoint 8D= טבלה עם דגלים שאומרים באיזה תת מחלקה מדובר מתאים ל-overlapping

אלגברה רלציונית: פעולות אונאריות σ select בחירת שורות לפי תנאי, קומוטטיבי, לא מסיר כפילויות project π בחירת עמודות, מסיר כפילויות, לא קומוטטיבי ρ rename שינוי שם רלציה/שמות התכונות **תורת הקבוצות** תאימות טיפוסים-מספר זהה של שדות וכל זוג שדות אותו טיפוס איחוד מכיל את כל מה שיש ב2 היחסים, מסיר כפילויות, קומוטטיבי, דורש תאימות טיפוסים חיתוך מה שזהה אצל 2 הרלציות, קומוטטיבי, דורש תאימות טיפוסים הפרש R-S כל מה שנמצא בR ובהכרח לא קיים בS, לא קומוטטיבי, דורש תאימות מכפלה קרטזית X כל שורה ברלציה 1 עם כל שורה ברלציה 2 עמודות זה כל העמודות ב2 הרלציות ושורות זה השורות ב $r1*r2$ **פעולות בינאריות** join שילוב של מכפלה קרטזית וסלקט (כאשר התנאי הוא = מדובר בעquijoin לא מוריד כפילויות) natural join כותבים * מוריד כפילויות והוא מתבצע על עמודות משותפות חלוקה התוצאה כוללת את כל העמודות שלא בתוך המחלק left outer join כל מה שיחס שמאל ומקסימום מכניסים null עבור יחס ימין בחלק מהשדות, אותו דבר לגבי right outer join רק עם יחס ימין וouter רגיל שלוקחים הכל משני הצדדים וזה מתבצע כמו join רגיל על העמודה המשותפת aggregate אפשר לבצע count,sum,average,max,min שלא מסיר כפילויות, משמאל לאופרטור תיכתב התכונה שלפיה נרצה למיין את שאר התכונות

תחשיב יחסים התוצאה תהיה כל השורות שמקיימות את $t \mid \text{cond}(t)$ לכל $(\forall t)(F)$ כל שורה ביחס חייבת לקיים את F לפי t
אם קיים $(\exists t)(F)$ לפחות שורה אחת ביחס תקיים את F לפי t

SQL כדי לבחור את כל התכונות נשתמש ב *select, כדי למנוע כפילויות נרשום distinct לאחר select, איחוד חיתוך והפרש גם מסיר כפילויות IN\NOT IN בודק האם הערך שאני מחפש נמצא או לא לפני not יגיע מה שעליו בודקים EXISTS בודק שלא חוזרת רלציה ריקה some לפחות משהו אחד שמקיים את התנאי all כל הערכים יקיימו את התנאי join אפשר להגדיר כבר from ואז התנאי where יהיה על הטבלאות המאוחדות: select a.name from actor as a join role as r on a.id=r.id aggregate functions גם פה יש את כל ה sum avg count max min על ידי group by נגדיר את העמודות שנקבץ לפיהן ועל ידי having נגדיר מה עושים(התנאי)

FD מפתח הוא מקרי פרטי של **FD** חוקים רפלקסיביות $y \rightarrow x$ הרחבה $x \rightarrow y$ אם $x \rightarrow yz$ טרנזיטיביות $yz \rightarrow x$ $x \rightarrow y$ $y \rightarrow z$ $x \rightarrow z$ הפרדה $x \rightarrow yz$ $x \rightarrow y$ $x \rightarrow z$ איחוד $x \rightarrow yz$ $x \rightarrow y$ ועוד אחד $x \rightarrow y$ $wy \rightarrow z$ $wx \rightarrow z$

שקילות של FD אם יש $G \rightarrow F$ נגיד שהן שקולות כאשר מגיעים לאותו סגור (והסגור נבדק על מה שמשותף) נגיד F מכסה את G אם כל תלות $G \rightarrow F$ ניתן להסיק F כלומר $G \rightarrow F$ + תת קבוצה של F ואותו דבר ההפך **FD מינימלי** אם נוריד תלות זה כבר לא יהיה שקול ל F ואי אפשר להחליף תלויות **סגור של F** כל התלויות שניתן להסיק מתוך התלות F prime attribute תכונה ששייכת למפתח קביל

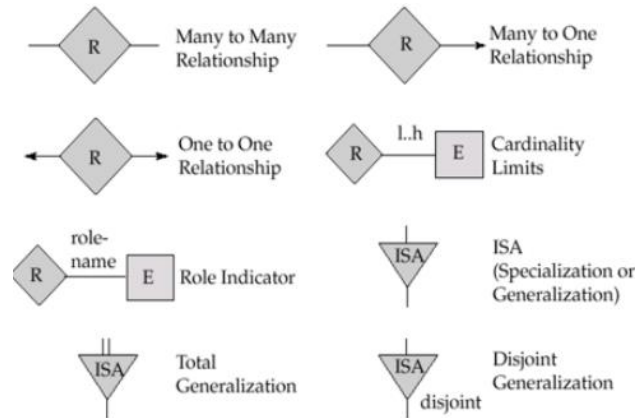
צורות נורמליות 1NF בלי תכונות מורכבות/מרוכות ערכים/מקוננות 2NF כל התכונות הלא ראשיות תלויות תלות מלאה במפתח הראשי 3NF כל התכונות שהן לא תכונות ראשיות תלויות במפתח ראשיות/קביל אם $x \rightarrow a$ אז x הוא חלק ממפתח ראשי או a תכונה ראשית **BCNF** כל תלות היא במפתח על וכל סגור של תלות יביא בסוף את כל R (כיסוי קנוני- הכי מצומצם שיש)

כיסוי קנוני בהתחלה $f(c)=f$ בכל שלב מחפשים מה לאחד אם קיים ואז עוברים לבדיקת תכונות עודפות מצד ימין ומצד שמאל תכונה עודפת מצד ימין $x \rightarrow yz$ אז כדי לבדוק אם y עודפת נוריד אותה מ f נחשב $f(c)'$ וזה עם התלות בלי y ונעשה סגור על x ואם נקבל את y אז אפשר להוריד את y תכונה עודפת מצד שמאל $xy \rightarrow z$ אז כדי לבדוק אם x תכונה עודפת נחשב את הסגור על y ואז נקבל את z אז אפשר להוריד את x **שימור תלויות** איחוד של היחסין המפורקים צריך להיות שווה לסגור של F ואם לא רואים תלות כמו שהיא נעשה עליה סגור תחת האיחוד של היחסים כלומר תחת $F_1 \cup F_2 \dots$ **שימור מידע** פירוק R ל R_1 R_2 אז החיתוך של R_1 ו R_2 יביא מפתח של אחד מהם וכשיש כמה יחסים אז נתחיל עם R_1 ונבדוק חיתוך שלו עם R_2 ואז עושים $R_1 \Join R_2$ כלומר אלה שבדקנו ואז על זה עושים חיתוך עם היחס השלישי וכך הלאה **פירוק לBCNF** משמר מידע אך לא תמיד תלויות, נמצא תלות שמפרה $x \rightarrow y$ ונפרק ל2 יחסים $R_1=(x,y)$ $R_2=(R-y)$ (כל הקבוצה חוץ מ y) נחשב F_1 ו F_2 ושוב נבדוק לגבי עמידה בתנאי **BCNF** ואם לא עומד בתנאי אז ממשיך לפרק את מי שמפר)

פירוק ל-3NF משמר מידע ומשמר תלויות, נחשב כיסוי קנוני ולכל תלות בכיסוי נעשה יחס שמכיל את כל התכונות שלה ואם לא קיבלנו יחס שמכיל את כל התכונות של אחד המפתחות הקבילים סימן שזה משמר מידע ותלויות ולבסוף נוסף גם יחס שיכיל רק אותם **מציאת מפתחות קבילים** אלו שלא נמצאים בצד ימין (לא תלויה באף אחד) חייבים להיות חלק מהמפתח הקביל

```
SELECT Department, sum(Salary) as Salary
FROM employee
GROUP BY department
HAVING SUM(Salary) >= 50000;
```

כשנשתמש בcount אז נעשה
having count(distinct e.d)>3
העמודה בselect חייבת להיות
תואמת לערך בby group
נעשה בלי ה as (בסלקט)



```
SELECT SSN, NAME
FROM EMPLOYEE AS E
WHERE BDATE > ALL
(SELECT BDATE
FROM EMPLOYEE
WHERE SUPERSSN = E.SSN)
```

דוגמה לשימוש
select SSN, Fname
from EMPLOYEE AS E
where Bdate > Some
(select Bdate
From EMPLOYEE
where superSSN = E.SSN)

BCNF מספיק שנפרק ליחסים יחס אחד שמפר ונלך איתו עד הסוף

```
SELECT column-names
FROM table-name
WHERE EXISTS
(SELECT column-name
FROM table-name
WHERE condition)
```

Q1: Retrieve the name and address of all employees who work for the 'Research' department.

$RESEARCH_DEPT \leftarrow \sigma_{DNAME='Research'}(DEPARTMENT)$
 $RESEARCH_EMPS \leftarrow (RESEARCH_DEPT \bowtie_{DNUMBER=DNOEMPLOYEE} EMPLOYEE)$
 $RESULT \leftarrow \pi_{FNAME, LNAME, ADDRESS}(RESEARCH_EMPS)$

TRC Q1: {t.FNAME, t.LNAME, t.ADDRESS | EMPLOYEE(t) and ($\exists d$)
(DEPARTMENT (d) and d.DNAME = 'Research' and
d.DNUMBER=t.DNO) }

רשום את השאלתא הבאה באמצעות אלגברה של יחסים:

הבא את כל מספרי הזיהוי של הסטודנטים שממוצע הציונים שלהם גבוה מ 80 וכן קבלו בקורס אחד לפחות ציון נמוך מ 60 אך מעולם לא קיבלו ציון גבוה מ 90.

$lowerThan60 \leftarrow \pi_{SID}(\sigma_{grade < 60}(Enrolls))$

$maxGrade \leftarrow \pi_{SID, grade}(\sigma_{SID \neq fmax_grade}(Enrolls))$

$maxLowerThan90 \leftarrow \sigma_{grade < 90}(maxGrade)$

$result1 \leftarrow lowerThan60 * maxLowerThan90 * student$

$result2 \leftarrow \pi_{SID}(\sigma_{Average > 80})$

הבא את שמות כל זוגות הסרטים (film1, film2) שהיו מועמדים לאותו פרס באותה שנה כך שהראשון (film1) זכה אבל לשני (film2) יש דרוג צופים גבוה יותר

Film(title, year, director, country, rating, genre, producer, original-language)

FilmActors(title, year, actorName, roleName)

Awards(title, year, award, status)

Actor(actorName, birthDate, country, gender)

Translated(title, year, language)

$AllFilmAwards(title, year, award, status, rating) \leftarrow \pi_{title, year, award, status, rating}(Film * Awards)$

$AllFilmAwards2(title, year, award, status, rating) \leftarrow \pi_{title, year, award, status, rating}(Film * Awards)$

$CompareFilms(film1, film2) \leftarrow \pi_{title, title2}((\sigma_{status1='won' \wedge rating1 > rating2}(AllFilmAwards1 * AllFilmAwards2)))$

דוג' לחלוקה

R	S	T
A	A	B
a1	a1	b1
a2	a2	b4
a3		
a1		
a3		
a2		
a1		
a2		

• b2 לא מופיע ברציה T כי אין לו שורה עם a2
• b3 לא מופיע ב-T כי אין לו שורה עם a1

• אין כפילויות

• אם ב-R הייתה גם עמודה C, אז נוסף ל-T את עמודה C ונצטרך שבכל שורה A, C תהיה תואמת עם T.
- דוגמה לשורות שק יכנסו ל-T:

R	B	C
a1	b1	c1
a2	b1	c1

• אם בשורה 2 היה c2 אז זה לא היה נכנס ל-T.

```
SELECT <attribute list>
FROM <table list>
[WHERE <condition>]
[GROUP BY <grouping attribute(s)>]
[HAVING <group condition>]
[ORDER BY <attribute list>]
```

Order יכול להתבצע לפי עמודות, לפי סדר יורד
DESC לפי סדר עולה

אפשר גם לעשות כמה פונקציות צבירה, לעשות על ידי AND או שאפשר להשוות
1. count(distinct x)= count(distinct y)
2. count(distinct x)>3 and count(distinct y)=5

אם משתמשים בdistinct בפונקציות צבירה אז נציין אותו גם בשורת הסלקט
select count(distinct something) from employee
group by e.id
having count(distinct something) ...