**תרגול מספר 3**

אילוצים ביצירת טבלאות (עד 8 אילוצים סה"כ).

PK – לא חייב להיות מורכב ממשהו אחד, יכול להיות מצירוף של כמה דברים. במידה ויש רק PK אחד, הוא יהפוך להיות UNIQUE.

NN- מחייב להיות בעל ערך, אם לא יכניסו ערך הוא יזרוק שגיאה.

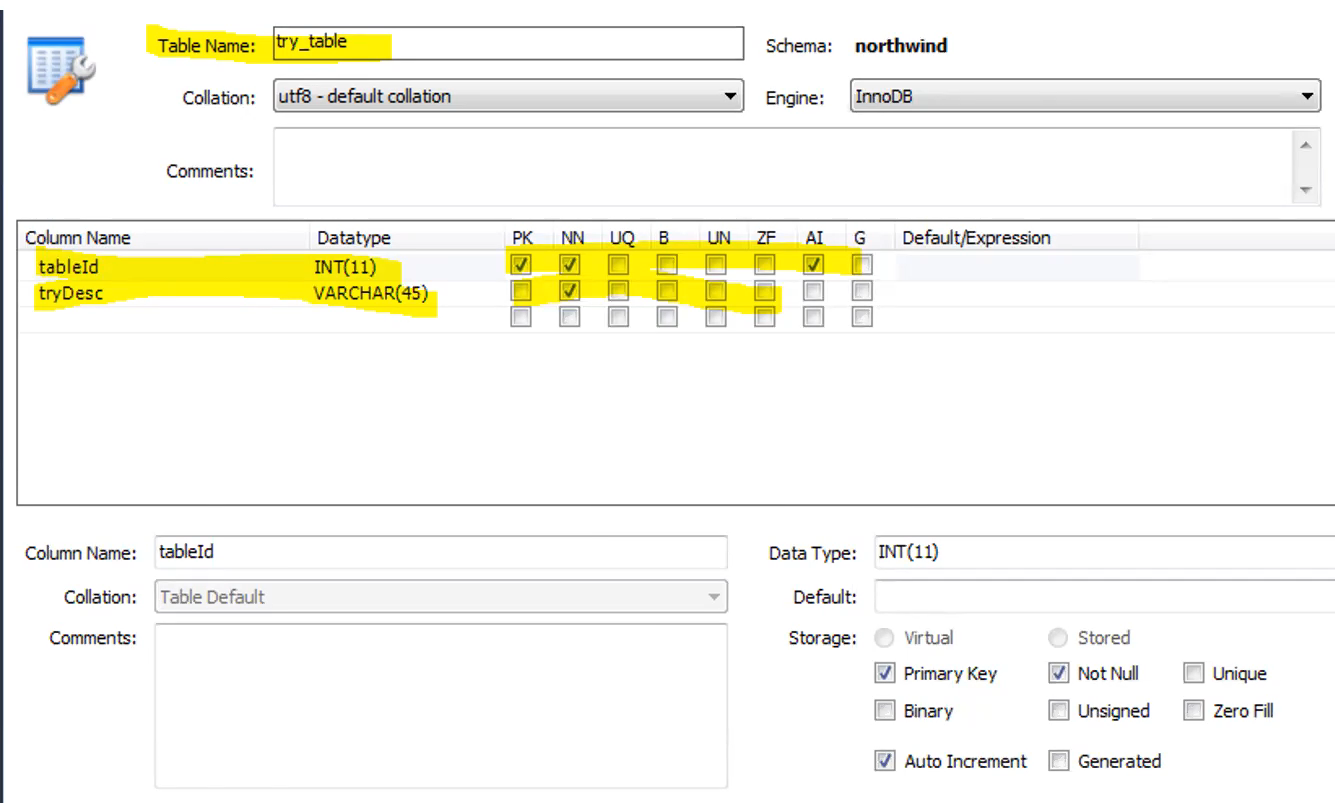
UQ – אם נרצה שיהיה בעל ערך ייחודי (כלומר ללא כפילויות)

B – מס בינארי (101 יהיה בינארי, לא עשרוני)

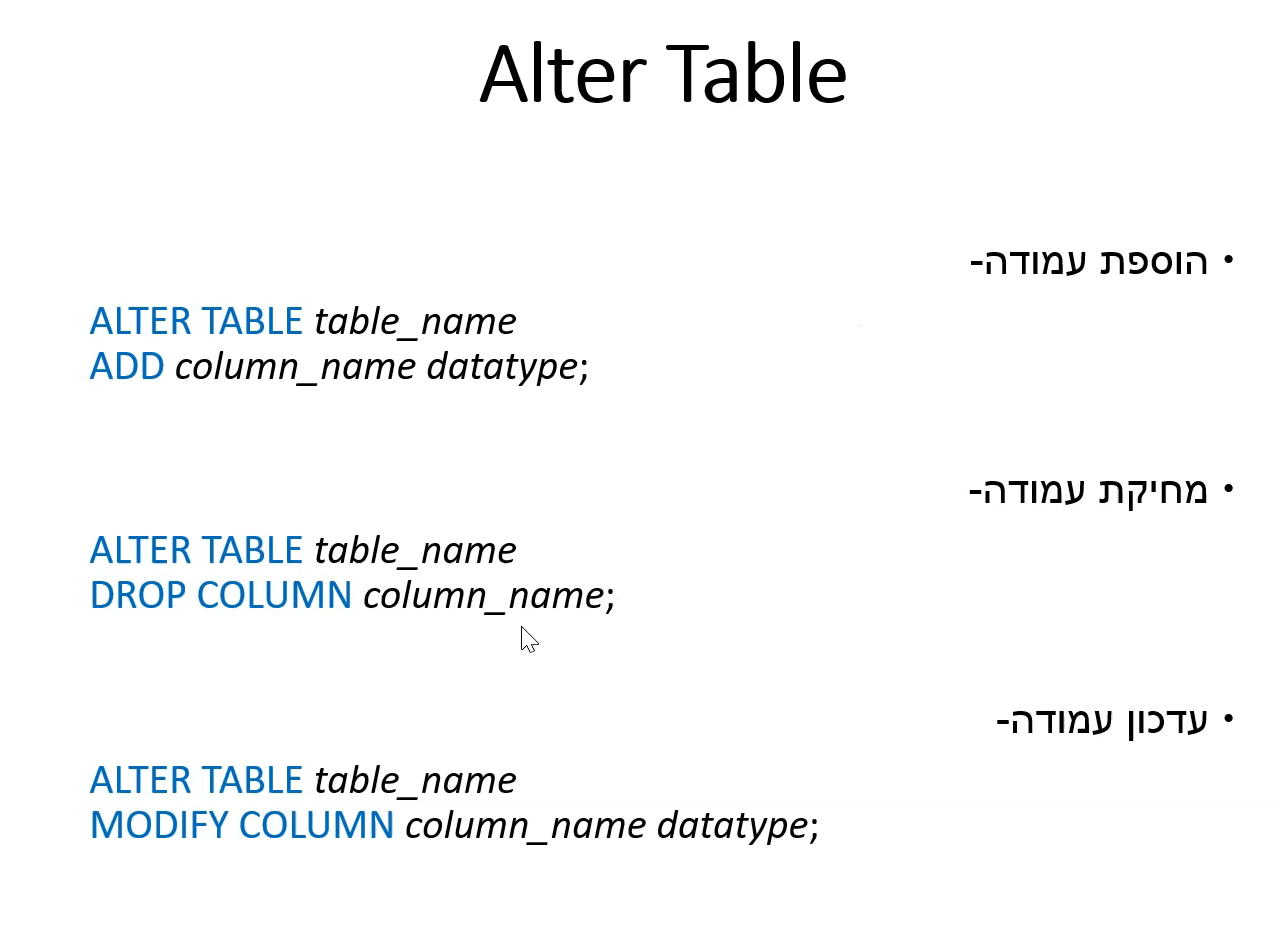
UN – UNSIGNED כלומר ערך חיובי בלבד.

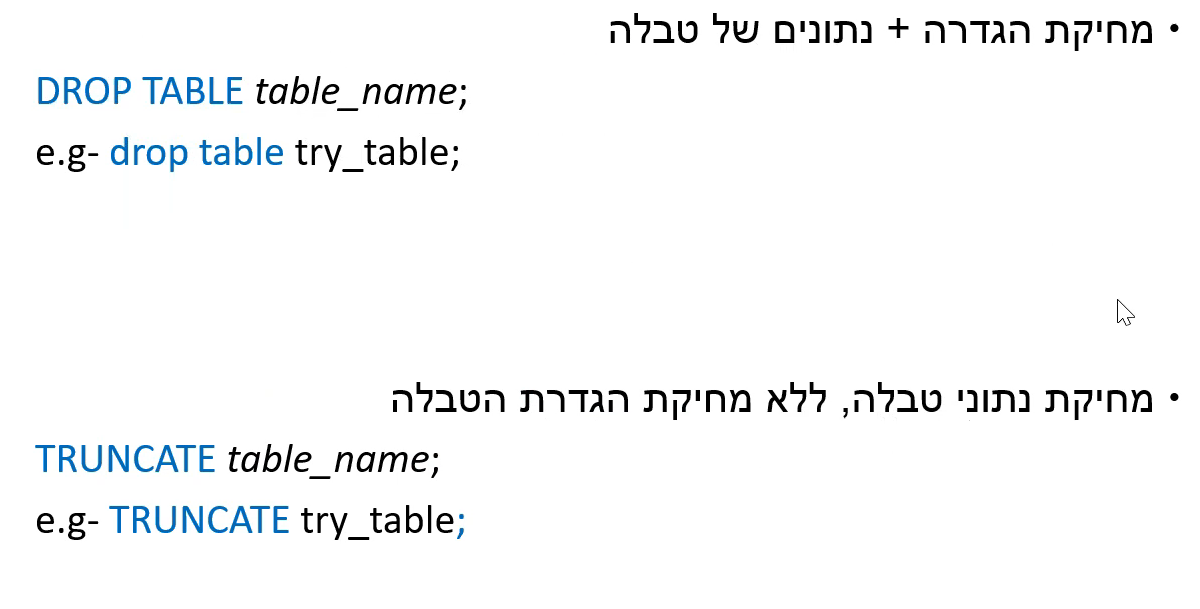
ZF – אוטומטית יאתחל את העמודה עם 0

AI – בכל פעם שנכניס רשומה, היא תקבל ערך שיגדל בצורה סדרתית (1,2,3... ) נוצר בצורה אוטומטית.

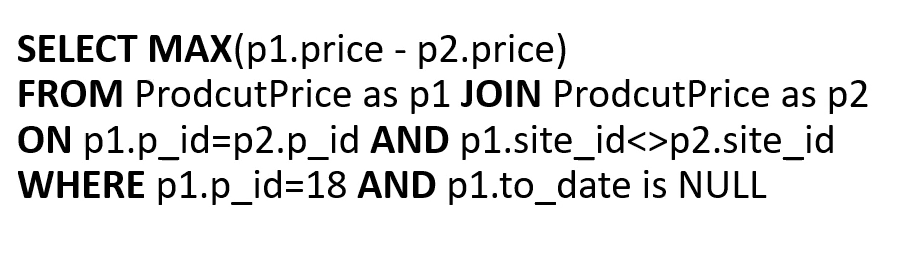
G – מתחולל אוטומטית, אך לא בצורה סדרתית

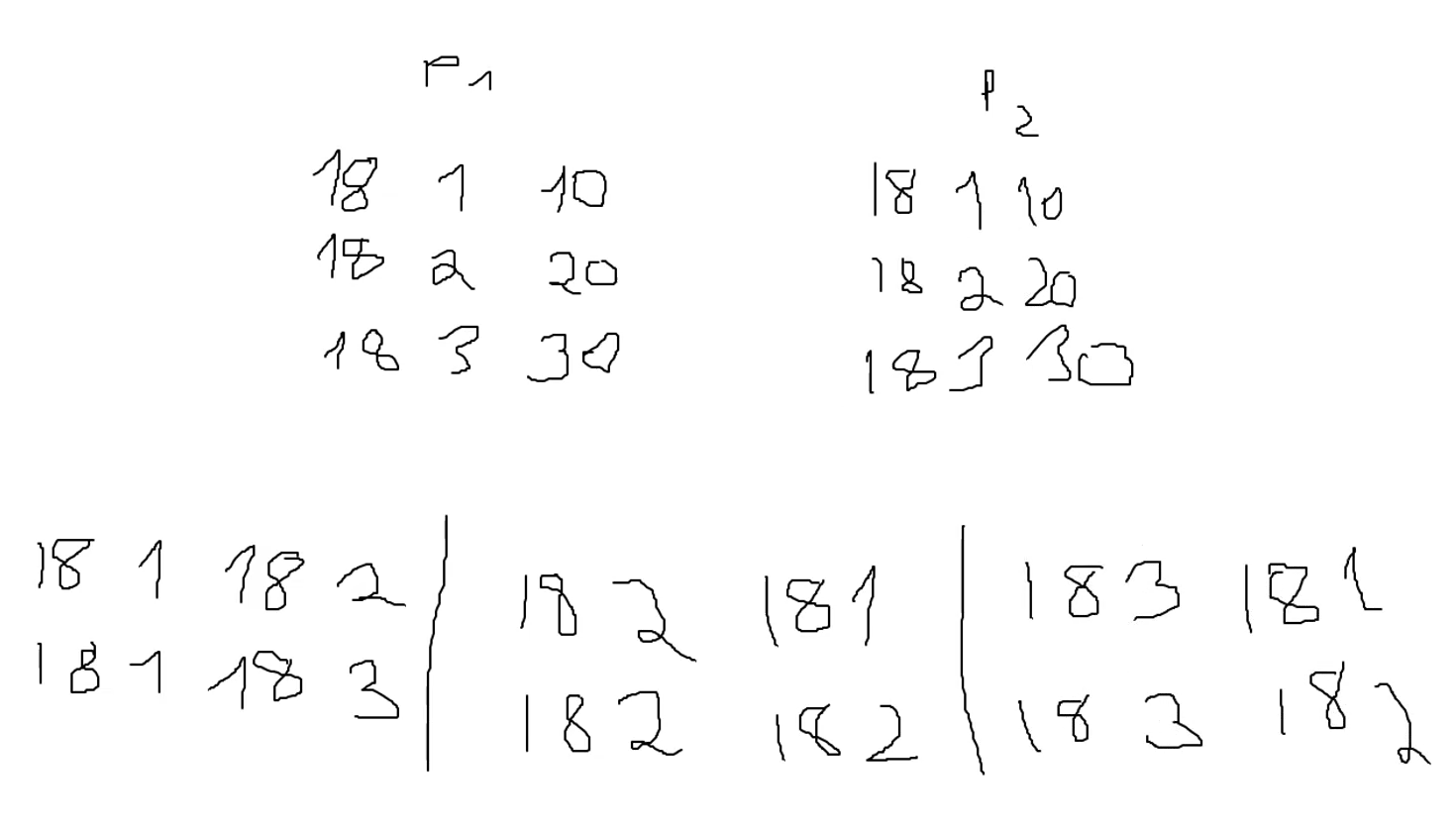
ALTER TABLE





מה תחזיר השאילתא הבאה?

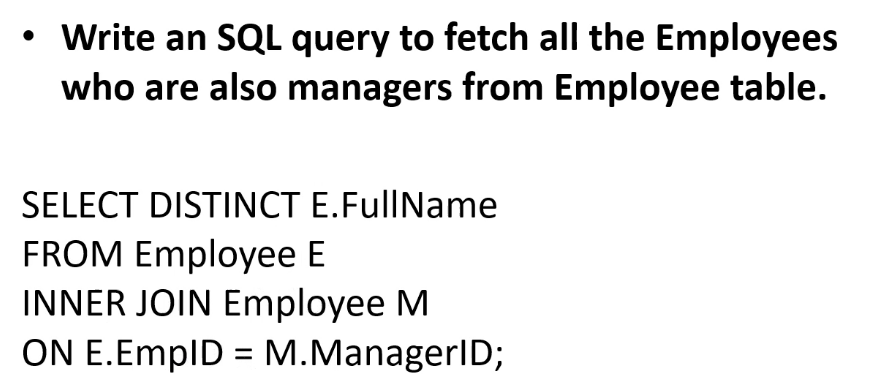
קודם נסתכל על ה-FROM, כדי להבין מאיפה אנחנו שולפים, איזה סינונים יש על המידע הזה. JOIN מרמז על 2 טבלאות, לפעמים טבלה יכולה לעשות JOIN עם עצמה. מה שחשוב זה לספק ALIAS. JOIN הוא למעשה חיתוך בין 2 טבלאות (שזה INNER JOIN בעצם). ניתן לעשות JOIN על 2 שדות. נרצה שה-ID יהיה אותו דבר, אבל SITE\_ID יהיה שונה. ה-JOIN למעשה מבצע השוואה בין 2 הטבלאות ומחפש את הדברים הבאים : ID זהה, אך SITE\_ID שונה.

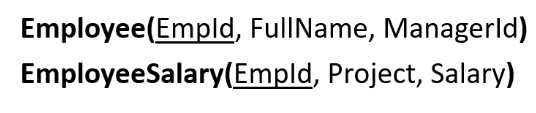


בעצם נראה בכמה נמכר כל מוצר בכל אתר. השורה האחרונה של הקוד דורשת שID יהיה 18 ושהמוצר יהיה בתוקף. לבסוף נבצע הפרשים בין כל מוצר מכל טבלה ונחזיר את המקסימום.

הפרש המחירים של מוצר מס 18 שלא פג תוקף באתרים השונים. בוחר את ההפרש הגדול ביותר.

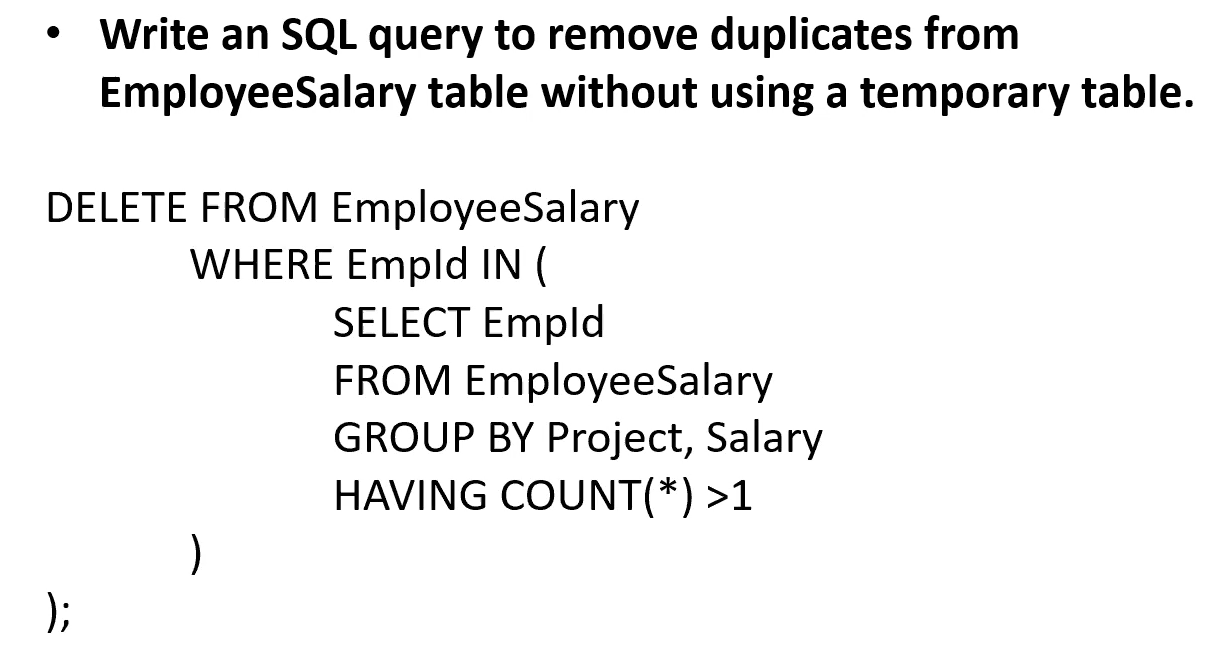
**שאלה נוספת**

בהינתן הטבלאות הבאות:

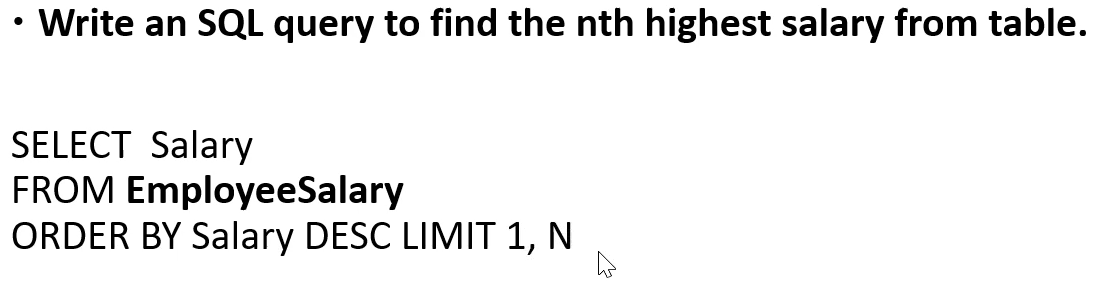


נשלוף את השמות המלאים של כל העובדים מטבלת ה-JOIN כאשר EMPLOYE\_ID=MANAGAR\_ID . ונציג מתוכם את השם המלא .\

\**לא חייבים להכניס AS כדי ליצור ALIAS (E)*

*****שאלה נוספת עם אותם טבלאות***

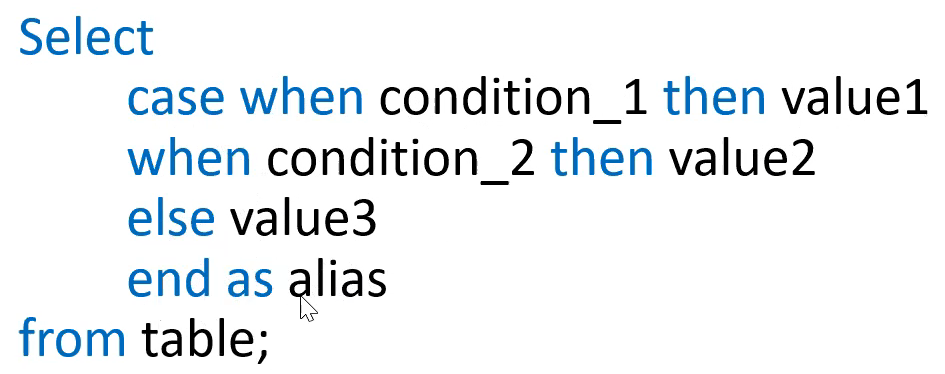
*איך נאתר כפילויות? נשלוף את כל הנתונים מטבלת EMPLOYEE\_SALARY, נקבץ לקבוצות ונספור כמה יש לי בכל קבוצה. אם בכל קבוצה יש יותר מאחד, אז יש כפילות. נעבור על כל שורה ונמחק.*

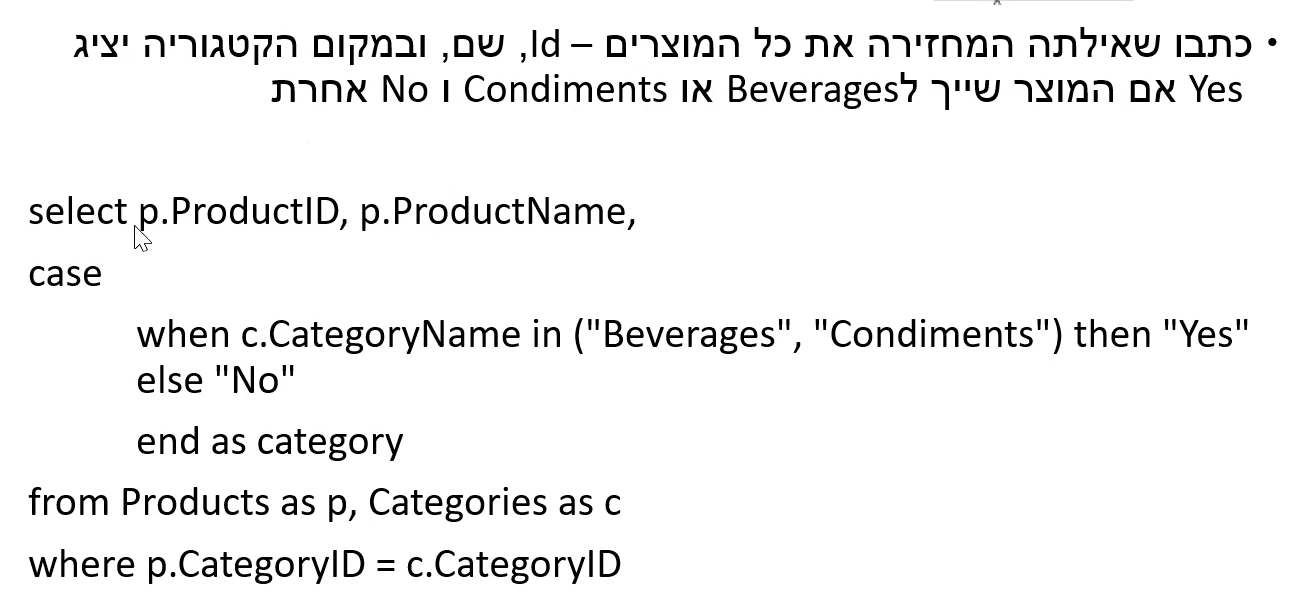
*****שאלה נוספת***

נרצה למצוא את העובד עם המשכורת ה-Nית הכי גבוהה.

מתוך טבלת משכורת, נמיין אותם לפי סדר יורד, נגביל את כמות התוצאות החוזרות ל-1 ונשלוף את האופציה במקום ה-N.



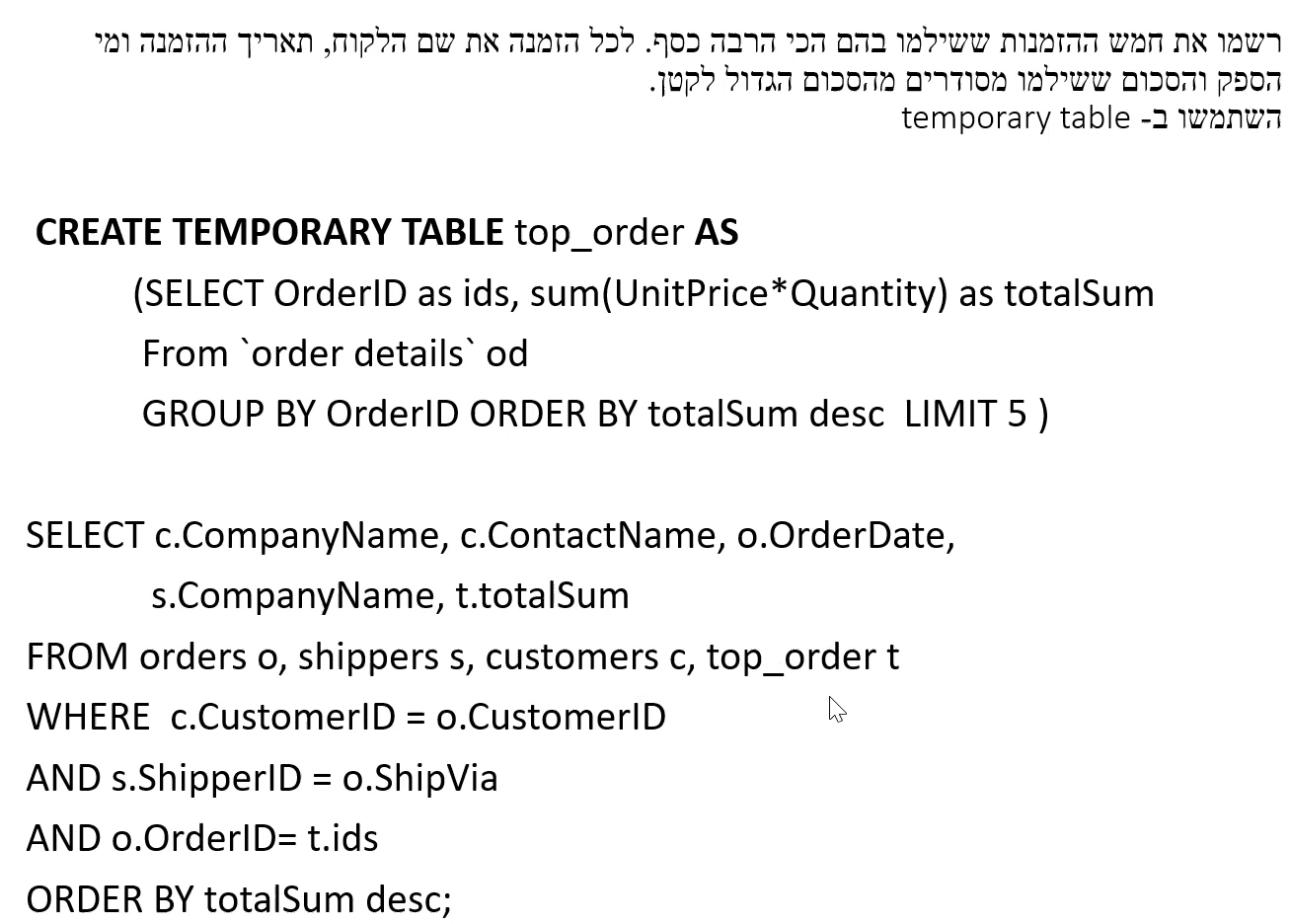
 מאפשר לשלט על התצוגה של הנתונים. במקום לשלוף רק עמודה, נוכל לבצע SELECT על העמודה. המבנה של השימוש :

**שאלה**

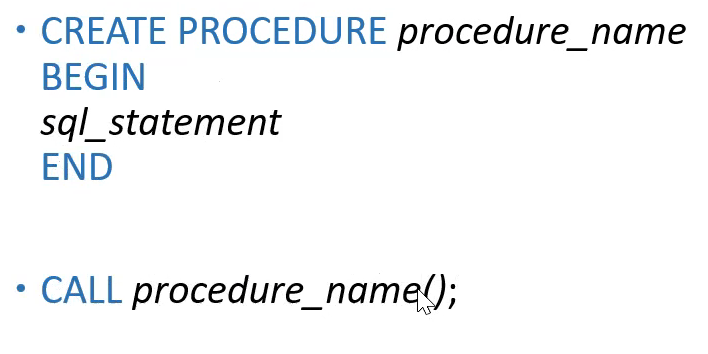
אם לא נשים ALIAS בסוף, הטבלה שלנו תצא "מכוערת".



ליצור טבלה זמנית, עד שנסגור את WHERE, או שנבצע רענון למבנה נתונים שלנו.





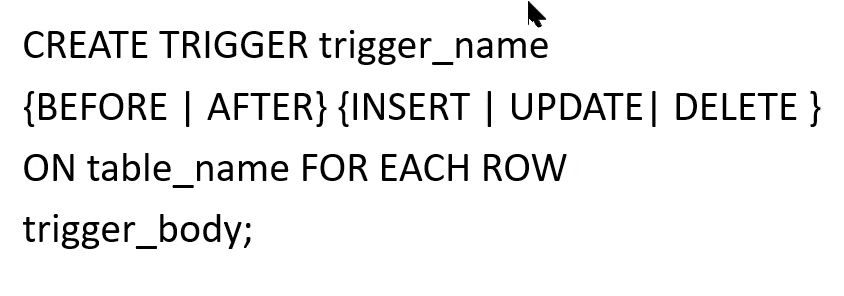
מה זה? פונקציה. לפעמים נרשום SELECT מאוד מורכב, ולא נרצה כל פעם מחדש לכתוב אותו. במקום זה נשתמש בפונקציה הנ"ל, ובכל פעם שנרצה לקרוא לאותו SELECT פשוט נשתמש בשם שלה. המבנה של הפונקציה:

למה נבצע drop procedure? כי נרצה לייעל את הפונקציה.

טריגר הוא מאוד דומה לסטורד-פרוסידור. אממה, סטורד-פרוסידור, כדי לקרוא לו אני צריך לקרוא לו בצורה ישירה. טריגרים הם סוג מיוחד של סטורד-פרוסידור, אך הוא מתעורר בצורה אוטומטית בתגובה לאירועים מסויימים בתגובה לאירועים בדאטה-בייס. קיים גם הבדל בסינטקס. טריגר מיוחס לטבלה מסוימת, תוך 3 אירועים ספציפיים שיעירו את הטריגר.

האירועים הם: insert,update,delete. נגדיר טריגרים לשלושת האירועים לטבלה מסוימת ואז הם יקראו בצורה אוטומטית. קצת מסוכן, מאחר והם קורים מאחורי הקלעים.

ב-MY SQL קיים רק סוג אחד של טריגרים : RAW LEVEL. אם נבצע פעולה שגורמת לטריגר להיקרא, אזי כל שורה המושפעת מכך תשונה.



או לפני הכנסה, לפני עדכון, לפני מחיקה

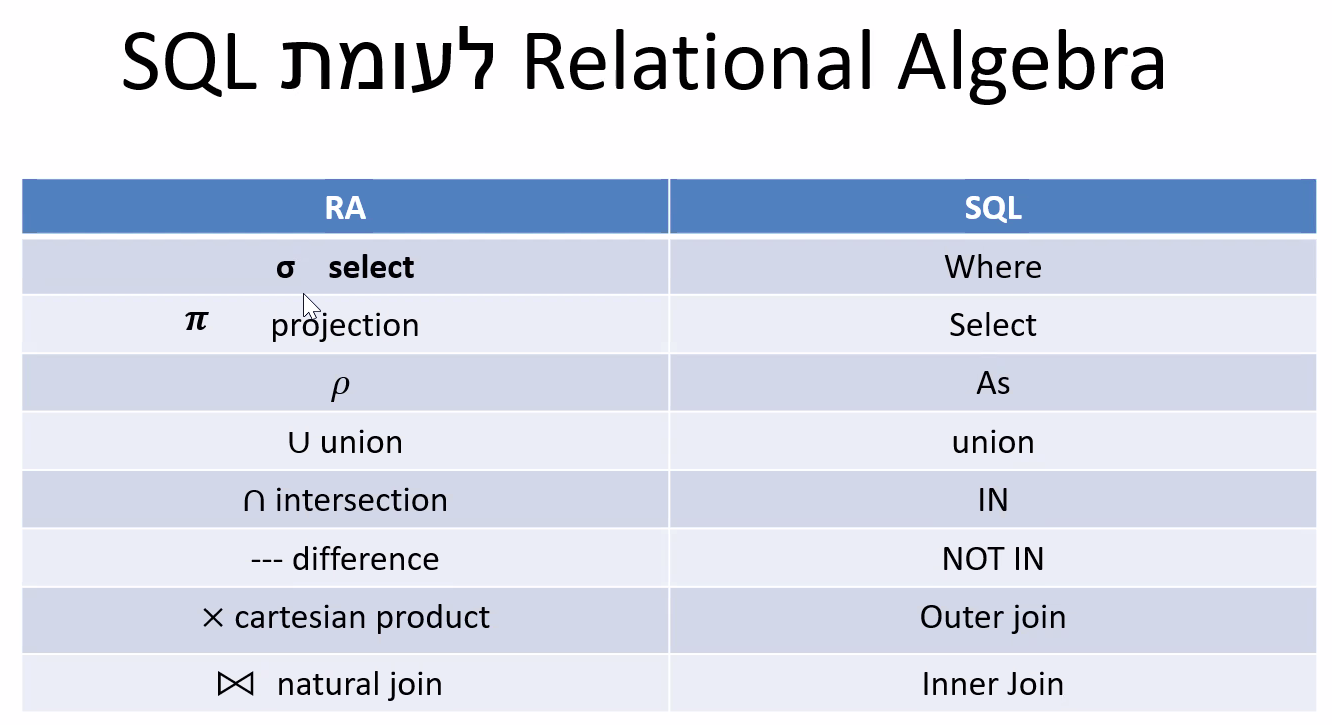
או אחרי הכנסה, אחרי עדכון, אחרי מחיקה. 6 מצבים. כלומר 6 אפשרויות עבור כל טבלה.

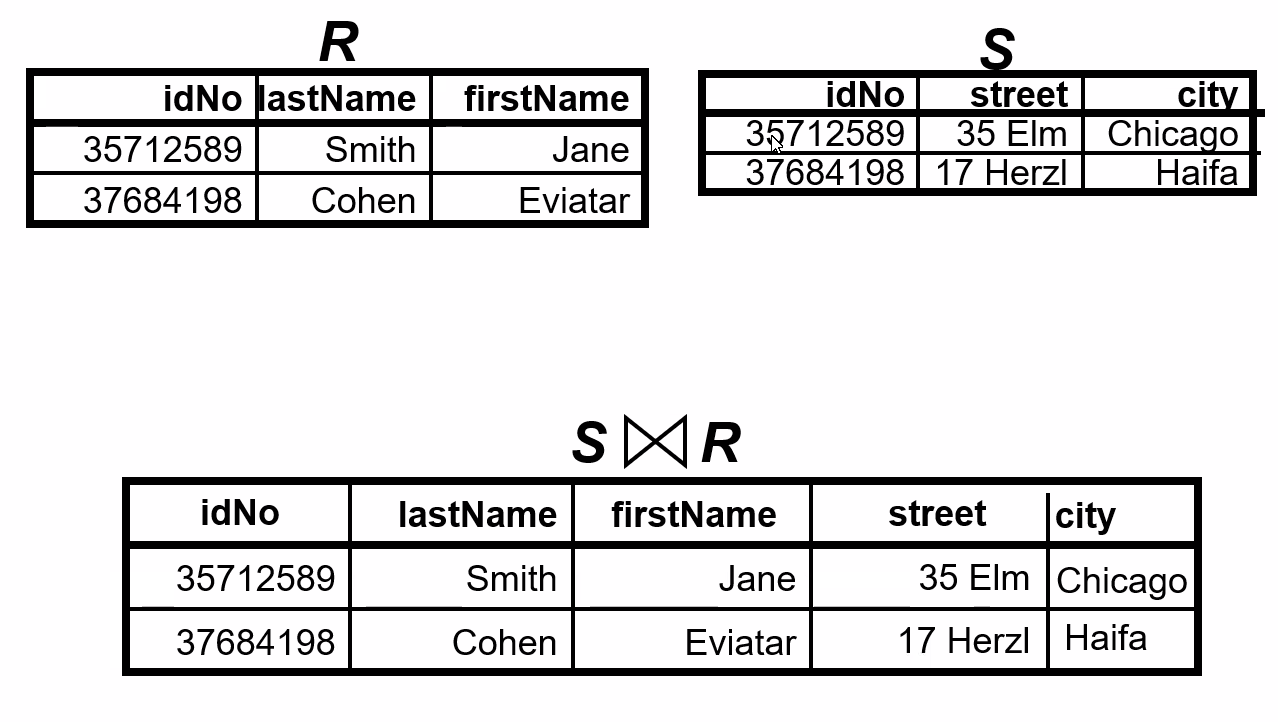
Drop triggers

Show triggers

במידה ויש טריגרים על טבלה אחת, ניתן בקידומת את השם של הטבלה.

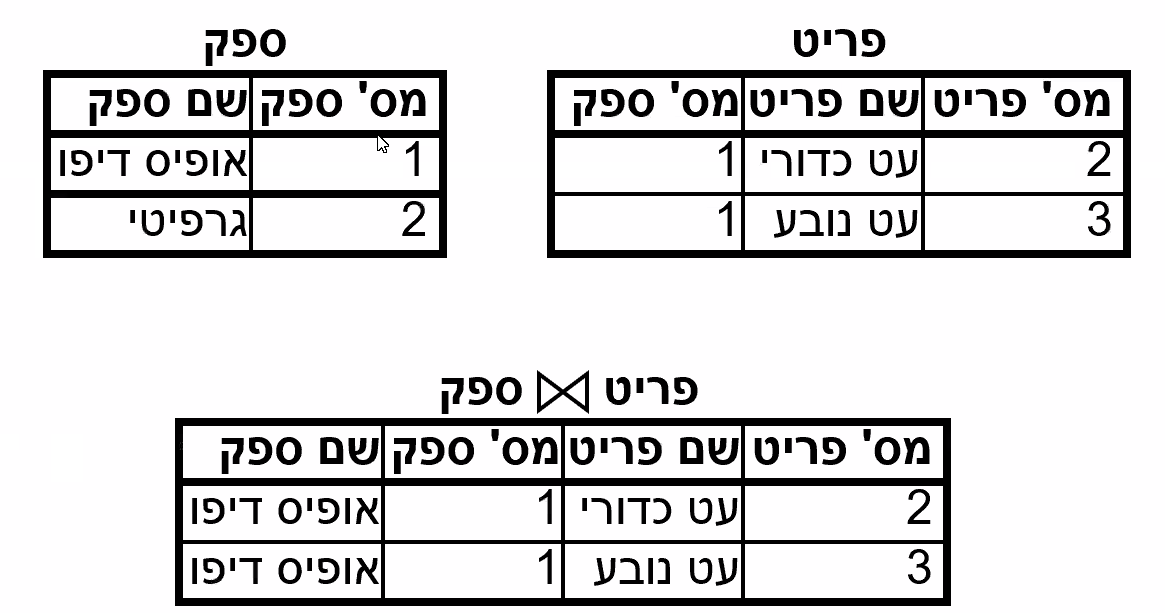
**תרגול מספר 4 : אלגברה רלציונית**

****



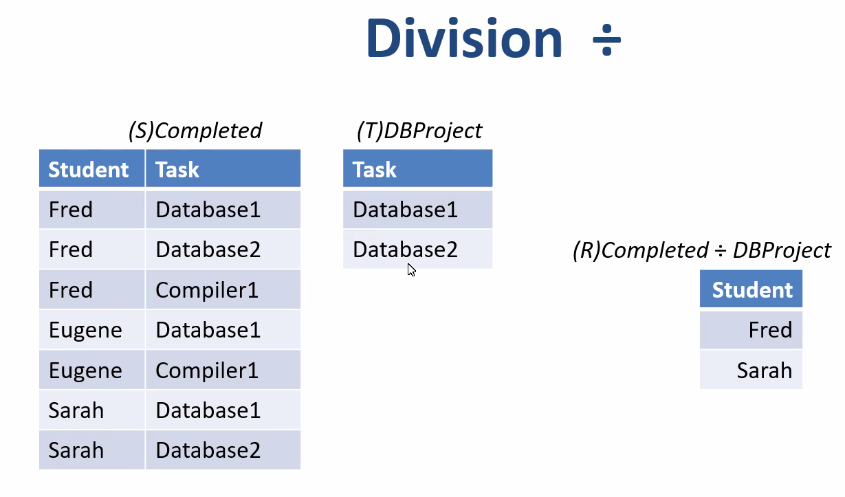
למעשה ילך לפי מה שמשותף ויחזיר אותו. מאחר ויש לו את זה הID ב2 המקומות, אז הווא מחזיר את שניהם.

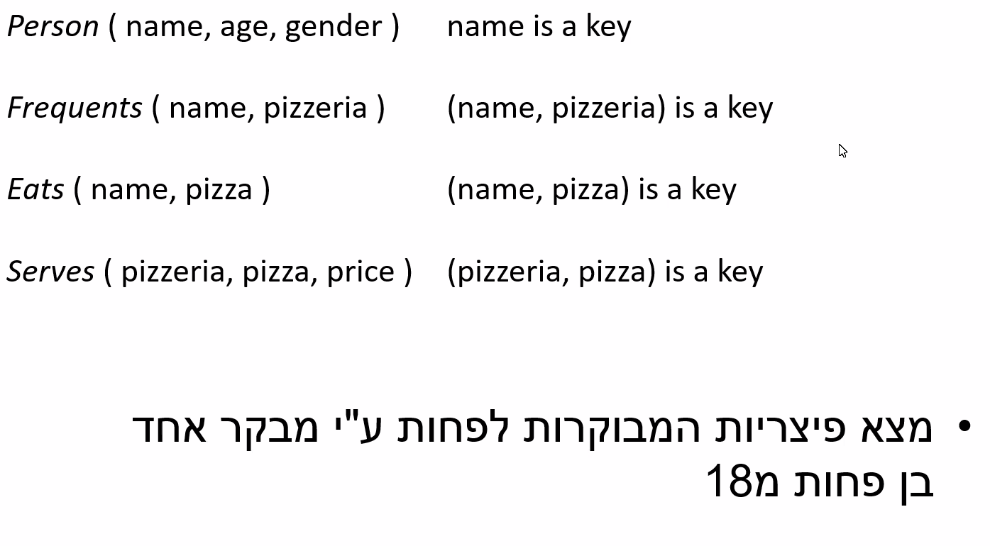
מה יקרה כשאין בדיוק אותם נתונים?

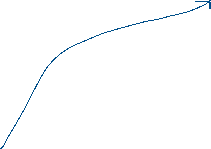
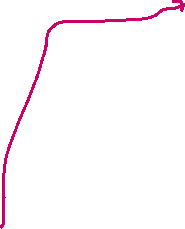
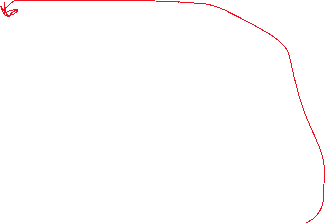
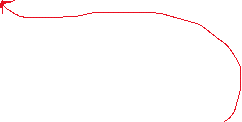


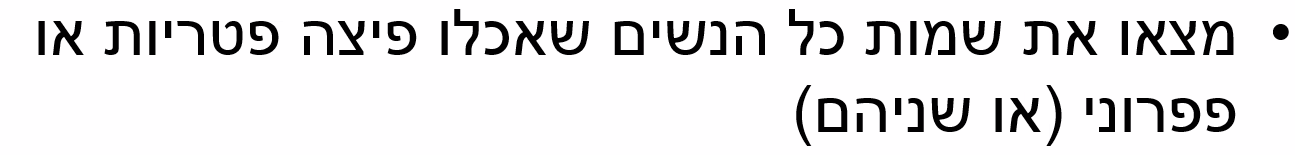


דומה מאוד לINNER JOIN באלגברה רלציונית.

מחזיר לנו רק את הסטודנטים אשר יש להם את הערכים של טבלת TASK. אם אין נקודה משותפת, נצטרך לציין בעצמנו, אך כאן כן קיימת נקודה משותפת ברורה. הוא מחזיר לנו את הקבוצה המקסימלית שעבורה המכפלה הקרטזית בין T לR מוכלת ב-S.











מה משמעות הביטוי הבא:

