

תרגיל 1 : מידול דיאגרמות ישויות קשרים

תאריך הגשה : 23: 55 , 27/03/2022

הוראות הגשה:

בתרגיל זה אתם נדרשים להגיש קובץ zip בודד שיכלול את הקבצים הבאים :

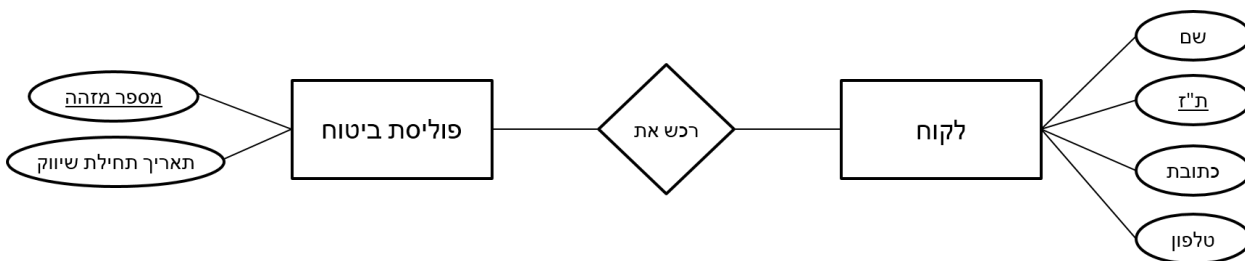
- ex1.pdf עם התשובות לשאלות להלן.
- create.sql
- drop.sql
- ex1.py
- README שמכיל שורה בודדת ובה שני ה-login של הסטודנטים שמגישים את התרגיל מופרדים בפסיק. במקרה שאושרה הגשה ביחיד יש לכתוב את login הסטודנט המגיש ללא סימנים נוספים.

שימו לב:

- נא לקרוא על הדרישות המנהליות של הקורס בלינק באתר הקורס כדי למלא אחר ההוראות להגשה של קבצים סרוקים!
- תרגיל מוקלד יזכה ב- 2 נקודות בונים!

שאלה 1:

נתונה דיאגרמה בסיסית של מסד נתונים שמכיל מידע אודות הלקוחות והפוליסות של חברת ביטוח. לכל לקוח יש שם, מספר ת"ז, כתובת וטלפון. לכל פוליסת ביטוח יש מספר מזהה, והתאריך בו חברת הביטוח החלה לשווק את הפוליסה. במסד נשמר גם המידע על איזה פוליסה רכש איזה לקוח.



בכל סעיף יש לצייר בדיאגרמה רק את המידע הנדרש באותו סעיף.

- (א) איך היית משנה את הדיאגרמה הבסיסית אם ידוע שלכל לקוח פוליסה אחת לכל היותר? (בכל סעיף יש לצייר מחדש את הדיאגרמה)
- (ב) איך היית משנה את הדיאגרמה הבסיסית אם ידוע ש:
- לקוח רוכש פוליסה עד לתאריך סיום מוגדר שבו הרכישה פוקעת, שאותו יש לשמור.
 - בנוסף, לכל לקוח יש סוכן ביטוח אחד בדיוק שדרכו הוא רוכש את כל פוליסות הביטוח שלו. לסוכן בחברת ביטוח ישנו שם, דרגת שכר ומספר עובד.
- (ג) יש מספר חברות ביטוח. לכל חברת ביטוח ישנו שם חברה ומספר זיהוי. במאגר שומרים, בנוסף למידע על רכישות של פוליסות ביטוח, גם את המידע אלו פוליסות משווקות על ידי כל חברת ביטוח. יצוין כי מספרים מזהים של פוליסות בחברות שונות עשויים להיות זהים, אך כל פוליסה שייכת לחברת ביטוח אחת בלבד. כיצד היית משנה את הדיאגרמה כעת?

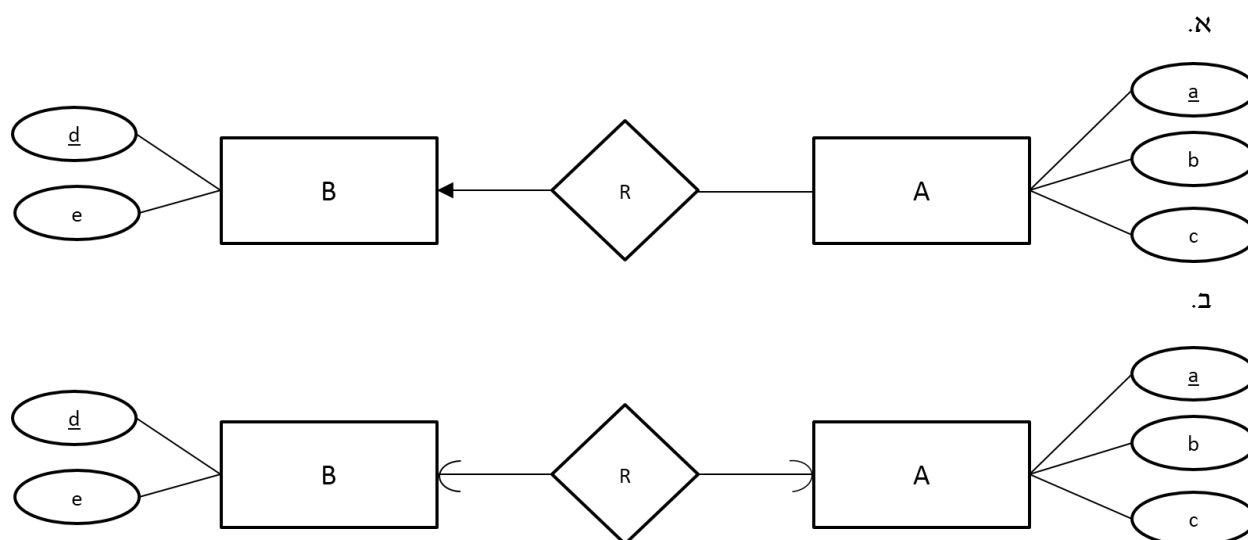
- ד) כיצד היית משנה את הדיאגרמה אם ידוע כי חלק מהלקוחות הם VIP. ללקוח כזה, ישנו מספר בין 1 ל-5 המגדיר את רמת השירות לה הוא זוכה. בנוסף, בעוד כל לקוח זכאי לרכוש פוליסה אחת לכל היותר, לקוח VIP רשאי לרכוש פוליסות נוספות ככל שירצה.
- ה) על מנת להתאים את עצמם לדרישות בשוק, בחברת הביטוח החלו לשווק אך ורק פוליסות ביטוח מותאמות אישית. פוליסה כזו מוגדרת באמצעות פוליסת הביטוח הרגילה של החברה, והלקוח עבורו הותאמה (בלעדיהן אין לה משמעות). לפוליסה מותאמת אישית מוגדר אחוז ההנחה ביחס לפוליסה שממנה הוגדרה. כך למשל, ייתכן שללקוח בשם יוסי הותאמה פוליסת שארים ייחודית עבורו הכוללת 5% הנחה ביחס לפוליסת ביטוח השארים המקורית שהוגדרה על ידי החברה. יצוין כי לכל פוליסה מותאמת אישית נדרש אישור של בדיוק מנהל אחד בחברת הביטוח, אשר לו שם ומספר מנהל ייחודי. כיצד היית משנה את הדיאגרמה הבסיסית כעת?

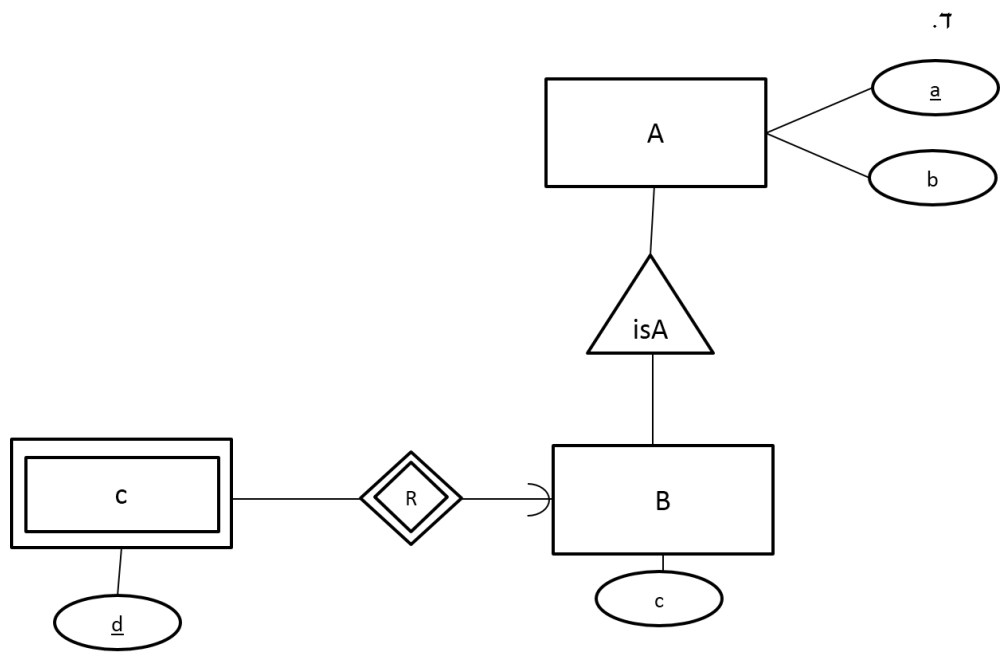
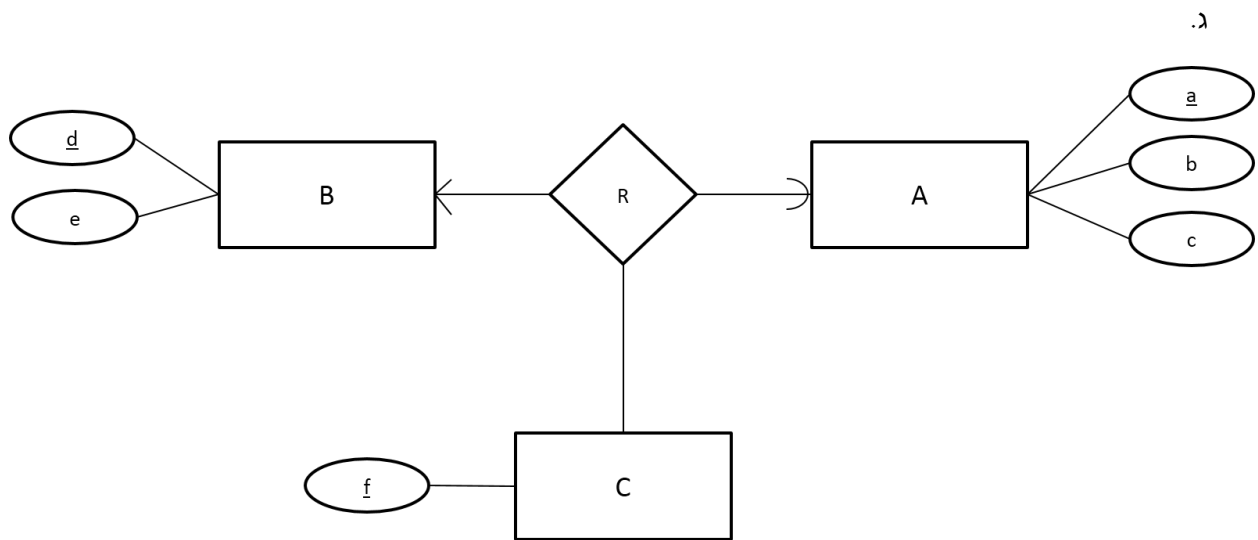
שאלה 2:

בכל סעיף:

(i) יש לתרגם את הדיאגרמה ליחסים ולציין את השדות של כל יחס, ואת המפתחות. אם יש כמה אפשרויות למפתח, ציינו את כולן. אם יש ירושה (isA), תרגמו בשיטת E/R style.

(ii) נסמן ב-|A| את מספר הישויות בקבוצת הישויות A. מה ניתן לומר על מספר הישויות בקבוצה A לעומת מספר הישויות בקבוצה B. יש להתייחס לשתי קבוצות אלו בלבד ולהשתמש בסימנים: <, >, <=, >=, =. במקרה שלא ניתן לקבוע יש לציין "לא ניתן לקבוע".

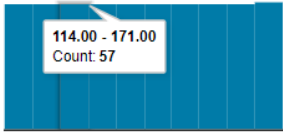




שאלה 3:

אתם נדרשים לתכנן ולבנות מסד נתונים עבור מאגר מידע אמיתי של נתוני מועמדות וזכייה בטקסי האוסקר. כדי להשיג את המידע תשתמשו באתר <http://www.kaggle.com>. האתר Kaggle מהווה קהילה למדעני דאטה ולמידה חישובית. היא מאפשרת למשתמשים למצוא ולפרסם מאגרי מידע, לבנות מודלים חישוביים ולעבוד עם מומחים אחרים כדי לגלות תובנות. מכיוון שכך, זה אתר שחשוב להכיר.

תרפרפו קצת באתר של Kaggle כדי לראות אלו סוגים של מידע אפשר למצוא. אנחנו נשתמש במידע של oscar-movies שנמצא בכתובת <https://www.kaggle.com/martinmraz07/oscar-movies>. כאשר תסתכלו בדף תראו רק חלק מהשורות ומהעמודות. על מנת לראות את כל 30 העמודות, יש לבחור select all בחיצה על מספר העמודות:

Detail Compact Column				10 of 30 columns
About this file				
This data contains the Oscar Best Picture winners and nominees. Additionally, the data contains IMDB and Rotten Tomato ratings. The inspiration for this dataset is to eventually develop a classifier to identify winners of future Best Picture awards.				
#	Film	Oscar Year	Film Studio/Produ..	
Index	Title of Film	Year of award show. Earlier years were grouped together. Source: Wiki	Film Studio/Producer Film Source: Wiki	
				
564 unique values				
1934 2% 1935 2% Other (547) 96%				Metro-Goldwyn-Ma... Warner Bros. Other (511)
0	Wings	1927/28	Famous Players-La	

כדי לראות את כל השורות, יש צורך להוריד את הקובץ archive.zip שבתוכו oscar_df.csv. הורדת הקובץ דורשת לבצע הרשמה לאתר. מומלץ להירשם. מי שאינו מעוניין בכך יכול למצוא את הקובץ באתר של הקורס וגם במערכת המחשבים במעבדה בתיקה:

~ db/data/ex1/archive.zip

ניתן להעתיק אותו לתיקה שלכם. הקובץ הזה מכיל טבלה אחת ענקית עם כל המידע על אירועי האוסקר, מועמדויות וזכיות. אך, במסד נתונים לא כדאי לשמור את המידע בצורה כזאת, כי יש בו הרבה מאוד כפילות מידע. בתרגיל זה אנו נשתמש רק בעמודות הבאות על מנת לפשט את הפתרון (בסוגריים מופיע מספר העמודה החל מ-0):

- Index (0)
- Film (1)
- Oscar year (2)
- Film studio/ producer (3)

- Award (4)
- Year of release (5)
- Movie time (6)
- Movie genre (7)
- IMDB rating (8)
- IMDB votes (9)
- Content rating (13)
- Directors (14)
- Authors (15)
- Actors (16)
- Film ID (29)

(א) ציירו דיאגרמת ישויות קשרים מתאימה הממדלת את המידע בעמודות של הקובץ `oscars_df.csv` (רק אלו שהופיעו בתיאור לעיל). השתמשו בידע הכללי שלכם על אופן ההתנהלות של אירועי האוסקר. אפשר להשתמש גם בהנחות טבעיות. שימו לב שההנחות צריכות להיות עקביות עם הנתונים, למשל, לא ניתן להניח שבכל טקס אוסקר היה סרט אחד בלבד. מומלץ להוסיף תיאור מילולי של הדיאגרמה המכיל את כל הידע. חובה לציין את ההנחות עליהם הסתמכתם. ייתכן שבדיאגרמה לא תצליחו למדל את כל ההנחות שמתקיימות בנתונים. במקרה כזה, ציינו אילו הנחות הדיאגרמה שלכם איננה ממדלת.

(ב) תרגמו את כל הדיאגרמה ליחסים רלציוניים. לכל יחס ציינו את האטריביוטים שהם המפתח. אם יש מספר אפשרויות למפתח מספיק לבחור מפתח אחד.

את סעיפים א' וב' יש להגיש בקובץ `ex1.pdf` ביחד עם התשובות לשאלות 1 ו-2.

בחלק הבא תשתמשו במסד הנתונים Postgres ובקוד `python` כדי לבנות טבלאות ולטעון את הנתונים לתוך הטבלאות. הסבר על הגישה לחשבון משתמש שלכם במערכת Postgres מצורפת בסוף התרגיל.

שימו לב! יש לוודא שהקבצים שלכם רצים על מחשבי המעבדה. לא יינתנו נקודות לתשובות שנכשלות בטעינה לתוך מסד הנתונים.

(ג) בסעיף זה, תתנסו ביצירת טבלאות, טעינת נתונים ומחיקת טבלאות בעזרת קבצי עזר. **שימו לב: הסעיף הזה להתנסות בלבד. אין תוצר להגשה מסעיף זה.**

הורידו מאתר הקורס את הקבצים: `ex1.py`, `create.sql`, `drop.sql`:

- `create.sql` מכיל פקודה אשר יוצרת במערכת ה Postgres טבלה אחת בשם Oscars הוזה בצורתה לטבלה המקורים של המידע.
- `drop.sql` מכיל פקודה המוחקת את הטבלה הנ"ל.
- `ex1.py` מכיל קוד השולף מתוך קובץ המידע המכוון (תחת השם `archive.zip`) את שורות המידע, וכותב אותן לתוך קובץ חדש, `oscars.csv` ע"י שימוש בפונקציה `process_row`. שימו לב – קובץ זה רץ באמצעות python3 ומעלה בלבד (במחשבי המעבדה השתמשו בפקודה `python3` כדי להריצו).

כעת:

- הריצו את הקוד בקובץ `ex1.py` וודאו שנוצר לכם הקובץ `oscars.csv`.
- התחברו למערכת `postgres` מתוך התיקיה שבו שמרתם את כל הקבצים על ידי הפקודה: (ההוראות המצורפות בסוף התרגיל, אבל גם רשומות כאן באופן חלקי לנוחיותכם).

```
psql -h dbcourse public
```

- הריצו את הקובץ `create.sql` ליצירת הטבלה Oscars בעזרת הפקודה

`\i create.sql`

- התנתקו מהמערכת בעזרת הפקודה

`\q`

- טענו את הנתונים לתוך הטבלה שייצרתם בעזרת הפקודה

`cat oscar.csv | psql -h dbcourse public -c "copy oscar from STDIN DELIMITER ',' CSV HEADER"`

- התחברו שוב למערכת postgres והריצו את השאילתה הבא המחזירה את כל השורות בטבלה שייצרתם:

`SELECT * FROM Oscars;`

תוודאו שאכן הנתונים נטענו לטבלה כראוי.

- הריצו את הקובץ `drop.sql` כדי למחוק את הטבלה
`\i drop.sql`

(ד) כעת אתם נדרשים לעדכן את הקבצים `create.sql`, `drop.sql` כך שייצרו את הטבלאות המתאימות ליחסים שהגדרתם בסעיף ב. ניתן לשנות מעט את הגדרות הטבלאות על מנת לנצל את תכונות מסד הנתונים (למשל, המסד מאפשר ערכי null).

- כתבו פקודות `create table` בתוך הקובץ `"create.sql"` היוצרות את הטבלאות שלכם. בפתרון וודאו שכללתם את כל התנאים והמגבלות (`key`, `foreign key`, `check`, etc). שיכולות להיות מוגדרות על הטבלאות. אתם יכולים להניח שכל נתון טקסטואלי הוא באורך מקסימלי 100.
- כתבו פקודות `drop table` בקובץ `"drop.sql"` שמוחקות את כל הטבלאות שייצרתם.

התחברו למערכת postgres וודאו שהפקודות שלכם רצות ללא הודעות שגיאה.

(ה) לבסוף, בסעיף זה אתם נדרשים לשנות את הקוד בקובץ `ex1.py` כך שיפצל את המידע לקבצים שונים, בהתאם להגדרות הטבלאות שלכם.

בקוד שסיפקנו לכם, בפונקציה `process_file()` יש מספר שלבים של `preprocessing` שנעשים למידע:

- בחירת העמודות הרלוונטיות לתרגיל בלבד.
- בעמודות הכוללות רשימות של ערכים (`writers`, `authors`, `directors` and `genres`) הוחלפו הפסיקים בסימן `&&` על מנת לאפשר פרסור והפרדה שלהם.
- מחיקה של כל סימני הציטוט, הגרשיים והפסיקים כדי לפשט את תהליך הטעינה של הנתונים.
- החלפה של הערך NA במחרוזת ריקה כך שבתהליך הטעינה של הנתונים, זה יחשב לערך null.
- לבסוף מתבצע שינוי נקודתי בנתונים. הראשון הוא תיקון של חלק קטן משנות אירועי האוסקר משנה בצורה "1927/28" לשנה תקינה מהצורה "1927". השני הוא תיקון טעות בנתונים - בשנה 1962 סומן בטעות כי שני סרטים שונים זכו באותו הטקס, כאשר בפועל רק אחד מהם זכה. הקוד מתקן זאת כך שיהיה זוכה אחד בלבד. בנוסף, בשנת 2020 הסרט `nomadland` זכה אך מתואר בטעות רק כמועמד. הקוד מתקן זאת כך שיסומן כמנצח.

מלבד זאת, סיפקנו לכם גם פונקציה בשם `split_list_value()` המקבלת ערך בעמודה המכילה רשימות, למשל `directors`, ומחזירה `list` של כל השמות ברשימה (על בסיס הסימן `&&` שהתווסף בשלב העיבוד המקדים שסופק).

שימו לב שאין לשנות את החלק שתואר עד כה בקוד, ויש להשאירו גם בקובץ שתגישו. בקובץ `ex1.py`, אתם נדרשים לבצע את השלבים הבאים:

- עבור כל טבלה צרו קובץ עם סיומת `csv` הנקרא באותו שם כמו הטבלה.
- עדכנו את הפונקציה `process_row` כך שתורשום את המידע הרלוונטי מכל שורה לתוך קבצי ה-`csv` של הטבלאות השונות.
- עדכנו את הפונקציה `get_names` כך שתחזיר רשימה עם שמות כל הטבלאות שהגדרתם.

השמות צריכים להיות תואמים גם לשמות טבלאות שהגדרתם בסעיף ב, וגם לשמות קבצי ה-csv שהגדרתם בקוד. **שימו לב:** יש להחזיר את שמות הטבלאות לפי הסדר הנכון לטעינת נתונים. כלומר, אם יש טבלה A עם אילוף מפתח זר לטבלה אחרת B, יש להחזיר קודם את B ורק אח"כ את A ברשימה.

שימו לב! המידע בקבצי ה-csv שאתם מייצרים צריך להופיע בלי שורות שחוזרות על עצמם כדי שניתן יהיה לטעון את הנתונים באופן תקין לטבלאות מבלי להפר אילוצי מפתח.

כעת, תבדקו שניתן לטעון את הנתונים לכל אחד מהטבלאות בהצלחה. כלומר, תייצרו שוב את הטבלאות. הריצו פקודה של טעינת שורות עבור כל אחד מהטבלאות, לפי אותו סדר שהחזרתם בפונקציה `get_names`. תוודאו, על ידי שאילתות, שהנתונים נכנסו כראוי. לבסוף תמחקו את הטבלאות.

יש להגיש את הקבצים `create.sql`, `drop.sql`, `ex1.py` בתוך zip ההגשה שלכם.

Appendix: Using Postgres

You can access your database account with the command:

```
psql -h dbcourse public
```

in the computer labs. After running this command, you can enter queries and DDL commands directly into the command line prompt.

In this exercise it will be more useful for you to write your create and drop table commands in a file, and then this file can be loaded into the database for execution. To do so, use the command

```
\i a.sql
```

within the prompt of the database, assuming your commands are in the file "a.sql". Some other useful commands are:

- `\q` exit psql
- `\h [command]` help about 'command'
- `\d [name]` describe table/index/... called 'name'
- `\dt` list tables