

פרויקט קורס חלק א' - דיאגרמות ER ו-SQL

הנחיות להגשת התרגיל:

1. **תאריך הגשה – 07.03.24 בשעה 23:55.**
2. **מתרגל אחראי על המטלה: בר גנוסר**
3. יש להגיש את תרגיל הבית בזוגות בלבד. ציון המטלה של סטודנט שיגיש לבד ללא אישור יתחיל מ-90. עבור סטודנטים שקיבלו אישור חריג להגיש לבד שימו לב כי בכל המקומות בהנחיות בהם רשום ID1_ID2 עליכם להשתמש בפורמט ID1_ID1 (כלומר לכתוב פעמיים את מספר הזהות שלכם).
4. יש להגיש תיקיית ZIP ששמה הוא מספרי זהות של המגישים בפורמט: ID1_ID2 כאשר ID1, ID2 הם מספרי הזהות של הסטודנטים. התיקיה תכלול את הקבצים הבאים:
 - (i) קובץ pdf ובו פתרון לשאלה 1. קובץ ה-pdf צריך להכיל עמוד שער עם מספרי תעודות הזהות של המגישים ושמותיהם. שם הקובץ צריך להיות מספרי זהות של המגישים בפורמט ID1_ID2_ERD.pdf. אין להגיש את הפתרון לחלק זה בכתב יד, אלא באמצעות תוכנת סרטוט (המלצה בהמשך).
 - (ii) קובץ SQL עבור שאלה 2 (בנושא DDL). שם הקובץ צריך להיות בפורמט ID1_ID2_Q2.
 - (iii) קובץ פייתון עבור שאילתות 3-4 בפורמט ID1_ID2_Queries וקובץ פייתון עבור ה-views של שאילתות 3-4 בפורמט ID1_ID2_Views (ראו הנחיות בסוף המסמך).
5. את תיקיית ה-ZIP יש להגיש דרך אתר הקורס במקום המתאים ב-moodle על ידי **אחד** מבני הזוג.
6. איחור בהגשת התרגיל יגרור קנס בגובה 20% מהציון עבור כל יום איחור (פרט למקרים חריגים כגון מילואים. במקרים אלה יש לפנות למתרגל האחראי טרם הגשת התרגיל).
7. פתרונות לשאילתות שלא יעברו את הבדיקה האוטומטית יקבלו את הציון 0. ניתן להגיש ערעור, אך בהנחה שיתברר כי הסיבה לכישלון בבדיקה היא אי מעקב אחרי הוראות ההגשה, ציון הסטודנטים יתחיל מ-90 והתרגיל ייבדק מחדש (ראו פירוט בסוף מסמך זה).

במסגרת הפרויקט בקורס תשלימו שני חלקים. מטרת הפרויקט היא לבנות מערכת מידע אודות חברת שירותי דיגיטל ולקוחותיה. בחלק א' (כלומר חלק זה) של הפרויקט תידרשו:

- א. ליצור דיאגרמת ER על סמך הדרישות המפורטות.
- ב. לכתוב את פקודות ה-CREATE TABLE הנחוצות ליצירת מסד הנתונים על סמך דיאגרמת ה-ER שבניתם.
- ג. ליצור views (יוסבר בהמשך) ולהשתמש בהם בתוך שאילתות אותן תתבקשו ליצור.

תאור המסד:

בית ההשקעות "יותר תשואה" מעוניין למדל מחדש את מסד הנתונים המיושן בו נשמר המידע הנחוץ לצרכי החברה. "יותר תשואה" מספקת ללקוחותיה פלטפורמה [למסחר בבורסה](#) ולצורך כך עליה לשמור מידע אודות לקוחותיה (המשקיעים בבורסה), עובדיה, החברות הנשכרות וערכי המניות שלהן.

כל משקיע (investor) מזוהה באמצעות מספר תעודת הזהות שלו (מספר בן 9 ספרות שלא יכול להתחיל בספרה 0), ובנוסף נשמרים שמו, תאריך לידתו, כתובת המייל שלו ותאריך הרשמתו למערכת. "יותר תשואה" לא מאפשר למשקיעים שנולדו בשנת 2006 ואילך להירשם למערכת. כמו כן, לא ייתכנו שני משקיעים שונים המשתמשים באותה כתובת מייל. משקיע חדש המתחיל להשתמש בשירותי החברה מוגדר בשלושת החודשים הראשונים שלאחר הרשמתו כמשקיע מתחיל (beginner), ולאחר מכן הוא הופך להיות משקיע פרימיום (premium). עבור כל משקיע פרימיום יש לשמור את יעדיו הפיננסיים (שדה מילולי). במסד מתועד מידע אודות עובדי החברה. כל אחד מהעובדים הוא בהכרח גם משקיע פרימיום המשתמש בעצמו בשירותי החברה. כל משקיע מתחיל מודרך על ידי (guided by) עובד אחד בדיוק, ולכל עובד יש לפחות משקיע מתחיל אחד אותו הוא מדריך.

במסד נשמר מידע אודות החברות (company) שבמניותיהן ניתן לשכור. חברה מזוהה באמצעות קוד זיהוי (symbol, שדה טקסטואלי) ובנוסף נשמר התחום (sector) אליו היא משתייכת (נדל"ן, אנרגיה וכד'), שנת הקמתה (founded) והמדינה בה יושב המטה הראשי שלה (location). שתי חברות שונות עשויות להיות ביריבות אחת עם השנייה. במקרים אלו יש לתעד את הסיבה ליריבות. עובד בחברה עשוי לעקוב אחר יריבות בין שתי חברות ולכתוב דוח סיכום אודות היריבות ביניהן. לא ייתכן כי שני עובדים שונים או יותר עוקבים אחר אותה היריבות, ולא ייתכן כי עובד יחיד עוקב אחר יותר מיריבות אחת.

לכל אחד מהמשקיעים יש לפחות חשבון מסחר אחד. חשבון מסחר מזוהה באמצעות מחרוזת בת 10 תווים בדיוק, ושייך למשקיע אחד בדיוק. משקיע יכול להעביר כסף מחשבון הבנק הפרטי שלו לחשבון מסחר לצורך רכישת מניות. כל העברה (transaction) מזוהה באמצעות תאריך ומזהה חשבון המסחר אליו בוצעה ההעברה, ובנוסף נשמר סכום הכסף שהועבר. לא ניתן להעביר סכום כסף הקטן מ-1,000 דולר. לעיתים העברה כלשהי נראית למערכת חשודה, ובמקרים אלו היא מועברת לבדיקה של עובד החברה. יש לתעד את החלטת עובד החברה בנוגע לחוקיות ההעברה (משתנה בוליאני).

מניה של חברה (stock) מזוהה באמצעות תאריך והחברה אליה היא מתייחסת. כמו כן, יש לשמור את ערך המניה באותו יום (ניתן להניח לשם פשטות כי ערך המניה של חברה כלשהי לא משתנה במהלך היום). עבור כל אחת מהחברות המופיעות במסד קיים תיעוד של לפחות מניה אחת ביום כלשהו. פלטפורמת המסחר של "יותר תשואה" מאפשרת למשקיעים לבצע קניית מניות (buying) דרך חשבון המסחר שלהם. קניית מניות מוגדרת כרכישת כמות כלשהי של מניות שבוצעה מחשבון מסחר לפי מחיר המניה ביום הקניה. עבור כל קניה יש לתעד את פרטי המניה וחשבון המסחר, בנוסף לכמות המניות שנקנו. לא ניתן לבצע יותר מקניה אחת של מניות השייכות לחברה כלשהי דרך אותו חשבון מסחר ביום יחיד (כן ניתן לבצע קניות מרובות באותו היום דרך אותו החשבון אם מדובר במניות של חברות שונות). עבור כל חשבון מסחר יש לתעד את סכום הכסף הזמין בו לרכישת מניות (סכום זה הוא סך הסכומים שהועברו לחשבון פחות הסכום שהושקע בקניית מניות).

מידול:

1 (40 נק') צרו דיאגרמת ER של המערכת. ציינו את כל ההנחות עליהן אתם מתבססים. עבור דרישות שלא ניתנות למידול באמצעות הדיאגרמה, יש לציין מילולית מהי הדרישה ולהציע דרך התמודדות עימה (למשל ברמת ה-DDL). ניתן להיעזר לצורך יצירת התרשים באתר <https://www.draw.io/>. יש לבנות את הדיאגרמה באנגלית כך ששמות הישיות והתכונות יתאמו לאלו שבהם תשתמשו בשאלה 2.

2 (30 נק') צרו סקריפט DDL עם פקודות Create Table מתאימות ליצירת מסד הנתונים. השתמשו בטיפוסים מתאימים לפי הערכים הנשמרים בכל שדה. בשדות בהם לדעתכם יש להשתמש בשדה מסוג מחרוזת אתם יכולים להניח כי לא ייתכן ערך שאורכו גדול מ-40 תווים, וכי בכל פעם שיש בסיפור התייחסות לתאריך הכוונה היא לפורמט DATE (YYYY-MM-DD). זכרו – יש חשיבות לסדר יצירת הטבלאות! הקפידו עליו. כמו כן, יש לוודא כי שמות הטבלאות והשדות תואמים לאלו שבהם השתמשתם בתרשים (שאלה 1).

טיפ: במקביל ליצירת סקריפט המקיים את הטבלאות, צרו סקריפט המוחק אותן (שייכתב בסדר הפוך לסדר בו הן נוצרו במסד), כך במקרה שתהיה לכם טעות ביצירת הטבלאות תוכלו למחוק בצורה מהירה את המסד ולהתחיל מחדש.

שאלות (Views):

בעולם מסדי הנתונים view הינו טבלה וירטואלית הנוצרת כתוצאה מהפעלת שאלתה על טבלאות רגילות במסד הנתונים. מכיוון ש-view דומה לטבלאות הרגילות במסד הנתונים בכך שגם הוא מורכב משורות ועמודות, ניתן לשלוף ממנו מידע ולעדכן אותו בדיוק כמו טבלה רגילה. במסד הנתונים, view מוגדר על ידי שאלת SQL. כשהמידע בטבלאות עליהן ה-view בנוי משתנה, המידע ב-view משתנה אף הוא בהתאם. נדגים את יתרונות ה-view ואופן הגדרתו בעזרת הדוגמא הבאה :

נניח ובמסד הנתונים שלנו קיימת טבלה בשם Order Details המוגדרת כדלקמן :

OrderDetails: (OrderNumber, ProductNum, QuantityOrdered, PriceEach)

כאשר :

– OrderNumber : מספר הזמנה

– ProductNum : מק"ט המוצר שהוזמן

– QuantityOrdered : כמות שהוזמנה מאותו מוצר

– PriceEach : מחיר ליחידה

כעת, נוכל לבנות view מעל טבלה זו בו יוצג לכל מוצר סכום תשלומי כלל ההזמנות עבורו :

```
CREATE VIEW SalesPerProduct
```

```
AS
```

```
SELECT ProductNum, SUM (QuantityOrdered * PriceEach) as TotalPrice
```

```
FROM OrderDetails
```

```
GROUP by ProductNum
```

יצרנו טבלה וירטואלית בשם SalesPerProduct, ובכל פעם שנרצה לדעת מהו סך ההכנסות ממוצר מסוים נוכל להריץ את השאלתה הבאה (עבור מזהה המוצר המתאים, למשל 102 בדוגמה המופיעה כאן) :

```
SELECT TotalPrice
```

```
FROM SalesPerProduct
```

```
WHERE ProductNum = 102
```

כדי להגדיר view נשתמש בפקודה CREATE VIEW ואחריה נרשום את שם טבלת ה-view (הטבלה הוירטואלית).

לאחר מכן נקבע איך view זה צריך להיראות באמצעות שאלת SQL רגילה המופיעה לאחר האופרטור AS. * מחיקת view מתבצעת בדומה למחיקת טבלה :

```
DROP VIEW viewname;
```

כאשר viewname הוא שם ה-view שברצוננו למחוק.

מידע נוסף (כולל syntax והסברים מפורטים) ניתן למצוא בכתובת הבאה :

http://www.w3schools.com/sql/sql_view.asp

בשתי השאלות הבאות עליכם להשתמש אך ורק בשלוש הרלציות המוגדרות כדלקמן (שימו לב כי הרלציות הללו לא בהכרח תואמות בדיוק לאלו שמוגדרות בסיפור שניתן עבור שאלות 1 ו-2):

Company (Symbol, Sector, Founded, Location)

Stock (tDate, Symbol, Price)

Buying (ID, tDate, Symbol, BQuantity)

עבור כל אחת מהרלציות, ה- primary key מסומן באמצעות קו תחתון.

Company – רלציה הכוללת מידע אודות חברות.

- Symbol – מזהה החברה.
- Sector – הסקטור אליו משתייכת החברה.
- Founded – השנה בה הוקמה החברה.
- Location – המדינה בה יושב המטה הראשי של החברה.

Stock – רלציה הכוללת מידע אודות מניות של חברות.

- tDate – תאריך.
- Symbol – מזהה החברה.
- Price – ערך המניה של החברה בתאריך הנתון.

Buying – רלציה הכוללת תיעודי קניית מניות על ידי משקיע.

- ID – מספר זהות של משקיע.
- tDate – תאריך.
- Symbol – מזהה החברה.
- BQuantity – כמות המניות של החברה שרכש המשקיע בקניה זו.

(3) (15 נק') :

"משקיע פעיל" הוא משקיע שרכש בכל אחד מימי המסחר המתועדים במסד מניות של **לפחות** שתי חברות שונות. עבור כל משקיע פעיל החזירו את הפרטים הבאים :

- מספר פעולות הקניה שביצע המשקיע.
- הסכום הכולל שהשקיע המשקיע בקניית מניות (עם דיוק של 3 ספרות אחרי הנקודה).
- הסקטור שמספר פעולות קניית המניות של חברות המשתייכות אליו הוא הגבוה ביותר עבור אותו משקיע. אם קיים שוויון בין שני סקטורים או יותר יש להחזיר את הסקטור הקטן יותר בהשוואה לקסיקוגרפית (מוקדם יותר ב-ABC).
- יש להחזיר את התוצאה ממוינת בסדר יורד לפי מספר פעולות הקניה שביצע המשקיע (שבירת שוויון לפי שם הסקטור בהשוואה לקסיקוגרפית בסדר עולה, כפי שמפורט לעיל).

הערה : האופרטור $\text{ROUND}(x, 3)$ מעגל את הערך של x לכדי דיוק של 3 ספרות אחרי הנקודה.

(4) (15 נק') :

"א גרוייסע מציאה" מוגדרת כחברה המקיימת את שני התנאים הבאים :

- במסד מתועדת רק אירוע קניית מניות יחיד של אותה החברה.
- ביום המסחר העוקב (השדה $tDate$ ברלציה $Stock$) לאירוע קניית מניות החברה ערכה של מניית החברה עלה **ביותר** מ-2%. על כן, התנאי לא יכול להתרחש עבור קניה שהתרחשה ביום המסחר האחרון המתועד במסד.
- שימו לב**- ייתכן כי קיים הפרש של יותר מיום אחד בין יום מסחר ליום המסחר העוקב שלו. למשל, יום המסחר העוקב ל-23.12.21 הוא 27.12.21.

"משקיע חד" הוא משקיע שקנה מניות של חברה המוגדרת כ"א גרוייסע מציאה".

עבור כל משקיע חד יש להחזיר את מספר פעולות קניית המניות שביצע המשקיע עבור חברות שנוסדו לפני שנת 2000 ומיקומן הוא ב-California. התוצאה לא צריכה להיות ממוינת.

ניתן ואף מומלץ להשתמש ב-views לצורך פתרון שתי השאלות.

את כל פקודות יצירת ה-views (של שתי השאלות) יש להכניס למקום המתאים בקובץ שפורמט שמו הוא `ID1_ID2_Views.py`. את השאילתה האחרונה שמחזירה את התוצאה המבוקשת אין לכתוב בתור `view` אלא בתור שאילתה רגילה ואותה להכניס לקובץ שפורמט שמו הוא `ID1_ID2_Queries.py`. ההנחיות מפורטות בהמשך מסמך זה.

לא לשכוח לסיים כל שאילתה ופקודת יצירת `view` ב- ;.

עבור כל אחת משתי השאילתות **מותר להשתמש ב-6 שאילתות VIEWS לכל היותר** (בנוסף לשאילתה המרכזית המחזירה את התשובה הסופית). כמובן שניתן להשתמש גם בפחות.

בהצלחה,

צוות הקורס

הערות חשובות

באחריותכם לוודא שמבנה הרלציה (כולל סדר השדות) בשאילתות שתגישו תואם את המבנה הנדרש כפי שמופיע בפלטים לדוגמה (יש משמעות לסדר התכונות בתוצאה).

הבדיקה בחלק זה תורץ באופן אוטומטי ותתבצע עבור מספר קלטים שונים. לתרגיל מצורף קובץ בשם `Create_Table_Commands.sql` שכולל פקודות DDL ליצירת הטבלאות הנתונות (המפתחות והאילוצים השונים של הטבלאות לא מופיעים בהגדרה כדי לא לחשוף תשובות לחלק הראשון). בנוסף, מצורפים קבצי קלט ופלט לדוגמה. תיאור הקלטים, אופן העלאת הנתונים לטבלאות והפלטים הנכונים מפורטים בהמשך. מומלץ ליצור את הטבלאות במסד הנתונים שלכם ולבדוק את נכונות הפתרון מול הקלטים והפלטים הנתונים בעצמכם ורק אז לוודא את תקינות ההגשה במנגנון הבדיקה האוטומטית (מוסבר עוד מעט). שימו לב כי בבדיקה האוטומטית שנריץ אצלנו נשתמש בקלטים ופלטים נוספים ולכן אנו ממליצים לכם לחשוב על מקרים נוספים ולבחון את עצמכם עליהם, שכן, ייתכן כי הפתרון שלכם יחזיר תוצאה נכונה עבור קלט מסוים אך לא עבור קלטים אחרים.

לתרגיל מצורף קובץ בשם `Automatic_Grader(Students).py` אשר מריץ בדיקה אוטומטית על קבצי הקלט והפלט. עליכם לבצע מספר פעולות כדי להריץ את הבדיקה.

(1) הזינו את מספרי הזהות של שני השותפים בפורמט `ID1_ID2` תחת המשתנה `IDS` שבשורה 242. למשל, עבור זוג סטודנטים שמספרי הזהות שלהם הם 123456789 ו-987654321, יש לכתוב בשורה 5 (שימו לב כי קו תחתון מפריד בין המספרים):

```
IDS = "123456789_987654321"
```

(2) שנו את שם את הקובץ `ID1_ID2_VIEWS.py` בהתאם למספרי הזהות שלכם. למשל, עבור זוג סטודנטים שמספרי הזהות שלהם הם 123456789 ו-987654321 יש לקרוא לקובץ `123456789_987654321_VIEWS.py`. בקובץ זה תכתבו את ה-`Views` שלכם ואותו תגישו.

(3) שנו את שם את הקובץ `ID1_ID2_QUERIES.py` בהתאם למספרי הזהות שלכם. למשל, עבור זוג סטודנטים שמספרי הזהות שלהם הם 123456789 ו-987654321 יש לקרוא לקובץ `123456789_987654321_QUERIES.py`. בקובץ זה תכתבו את השאילתות שלכם ואותו תגישו.

(4) הזינו את שם המשתמש הטכניוני של אחד מבני הזוג תחת השדות `DATABASE_NAME` ו-`NAME_USER` בשורות 244 ו-245 (לא לשכוח להחליף " " ב-0 במקרה הצורך).

```
SERVER_NAME = "techniondbcourse01.database.windows.net"
DATABASE_NAME = "user"
USER_NAME = "user"
PASSWORD = "Qwerty12!"
```

(5) ודאו כי הקבצים המצורפים מופיעים בתיקיה בשם data, אשר יושב בתיקיה המרכזית בה ממוקם הקובץ Automatic_Grader(Students).py. יש לעשות לתיקיה unzip ולשים אותה כתיקיה רגילה עם כל הקבצים (ראו תמונה בנקודה 8 שבעמוד הבא) תוכן התיקיה הוא :

Buying_test1
Company_test1
create_table_commands
GT_q3_test1
GT_q4_test1
Stock_test1

(6) בקובץ ששמו (לפני שתשנו אותו בהתאם למספרי הזהות שלכם) ID1_ID2_VIEWS.py מוגדר מילון בשם VIEWS_DICT. למילון זה שני מפתחות – Q3 ו-Q4. הערך עבור כל אחד משני מפתחות אלו הוא רשימה שבה תזינו את ה-VIEWS שלכם. כל פקודת יצירת VIEW היא איבר ברשימה (סדר הופעת האיברים ברשימה חייב להיות כסדר יצירת ה-VIEWS הרצוי) המופיע בין "" ל-"". מזכירים שוב כי ניתן להשתמש ב-6 VIEWS לכל היותר. אמנם גם אם תחרגו ממגבלה זו הטסט עשוי לעבור, אך על המסך תופיעה הודעה המודיעה על החריגה (ובבדיקה שלנו הטסט ייכשל). מבנה הזנת המידע הוא :

```
VIEWS_DICT = {  
    "Q3":  
        [ ""  
  
          ""  
  
          ""  
  
          ""  
  
        ]  
    "Q4":  
        [ ""  
  
          ""  
  
          ""  
  
          ""  
  
        ]  
}
```

(7) בקובץ ששמו (לפני שתשנו אותו בהתאם למספרי הזהות שלכם) ID1_ID1_Queries.py מוגדר מילון בשם QUERY_ANSWERS. למילון זה שני מפתחות – Q3 ו-Q4. הערך אותו יש להזין עבור כל אחד ממפתחות אלו הוא השאילתה המתאימה. שימו לב כי עבור שתי השאלות, יש להחזיר את התשובה הסופית דרך שאילתה (זהו התוכן שמוזן בקובץ זה), ולא דרך VIEW. מבנה הזנת המידע הוא :

```
QUERY_ANSWERS = {  
    "Q3":  
        ""  
  
        ""  
  
    "Q4":  
        ""  
  
        ""  
  
}
```


(8) התיקיה שבה יושב קובץ ההרצאה האוטומטית צריכה להכיל את הקבצים הבאים (עם החלפת השם ID1_ID2 במספרי הזהות שלכם כמובן):

data
Automatic_Grader(Students)
ID1_ID2

כאשר תיקית הזיפ היא התיקיה אותה אתם מגישים שמכילה את הקבצים הבאים (עם החלפת השם ID1_ID2 במספרי הזהות שלכם כמובן):

ID1_ID2_ERD
ID1_ID2_Q2
ID1_ID2_Queries
ID1_ID2_Views

(9) אנו ממליצים לכם לעבוד קודם כל דרך ה-console, לוודא שאתם בטוחים לגבי תשובתכם ורק לאחר מכן להעתיק את השאילתה למקום המתאים ולהריץ את Automatic_Grader(Students).py.
(10) כעת, הריצו את הקובץ. אם אכן פתרוכם נכון עבור הקלט הנתון, הפלט אותו תראו יהיה:

```
*****  
query 3, test 1: Good  
*****  
query 4, test 1: Good
```

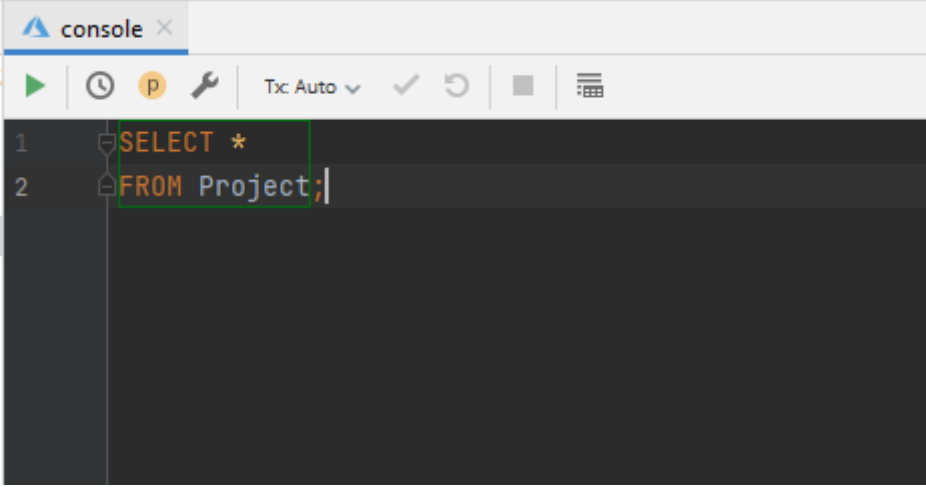
הערה חשובה - פתרוכם בשאלות 3 ו-4 ייבדק בצורה אוטומטית. באחריותכם לוודא כי הקפדתם על

הנחיות ההגשה המפורטות בקובץ זה. אם הבדיקה האוטומטית נכשלה ציונכם בתרגיל יהיה 0. כמובן שתוכלו להגיש ערעור והתרגיל ייבדק מחדש אך ציונכם יתחיל מ-90 (אלא אם יתברר שלא חרגתם מאף אחת מההנחיות). אנא ודאו כי ההגשה שלכם מקיימת את כל סעיפי ה-checklist הבא:

- הגשת קובץ PDF שפורמט שמו ID1_ID2_ERD (יש להחליף את ID1 ו-ID2 במספרי הזהות של הסטודנטים המגישים) עבור סרטוט התרשים ופירוט ההנחות
- הגשת קובץ SQL שפורמט שמו ID1_ID2_q2 (יש להחליף את ID1 ו-ID2 במספרי הזהות של הסטודנטים המגישים) עבור פקודות ה-DDL המבוקשות בשאלה 2.
- לצורך הבדיקה – הזנת מספרי הזהות של המגישים בשורה 202 בקובץ Automatic_Grader(Students).py.
- לצורך הבדיקה – הזנת שם המשתמש של אחד מבני הזוג בשורות 245 ו-246 בקובץ Automatic_Grader(Students).py.
- הגשת שני קבצי הפיתוח שפורמט שמם הוא ID1_ID2_Views.py ו-ID1_ID2_Queries.py (יש להחליף את ID1 ו-ID2 במספרי הזהות של הסטודנטים המגישים) כאשר את שני הקבצים הללו יש למלא בהתאם להנחיות הניתנות בנקודות 6 ו-7 שבעמוד הקודם.
- יש לשמור על הסדר הפנימי בין ID1 ל-ID2 בכל המקומות בהם אתם נדרשים לכתוב את מספר הזהות. למשל אין לכתוב בשורה 5 בקובץ 987654321_123456789 ולהגיש את התרגיל כולו תחת השם 123456789_987654321.

נספח- שמירת תוכן הקונסול כקובץ SQL

נניח כי ברצוננו לשמור כקובץ sql את השאילתה הבאה אותה כתבנו ב-console :

A screenshot of a console window with a dark background. The title bar says 'console'. The toolbar includes a play button, a clock, a 'p' icon, a wrench icon, a dropdown menu set to 'Tx: Auto', a checkmark, a refresh icon, a close button, and a menu icon. The code is entered in two lines: '1 SELECT *' and '2 FROM Project;'. A green rectangular box highlights the text 'SELECT * FROM Project;'.

(1) לחיצה עם הכפתור הימני של העכבר על כותרת הלשונית (ליד הלוגו הכחול).

(2) ריחוף על האפשרות של Open in.

(3) בחירה באפשרות של Explorer (במחשבי מאק הפקודה היא Finder)

(4) כעת תיפתח התיקיה בה נשמר הקובץ. שנו את שמו בהתאם לצרכיכם.

נספח 2- טעינת קבצי CSV לרלציות

1. צרו את הטבלאות במסד הנתונים על ידי הרצת הפקודות בקובץ Create_Table_Commands.sql מתוך ה-console.

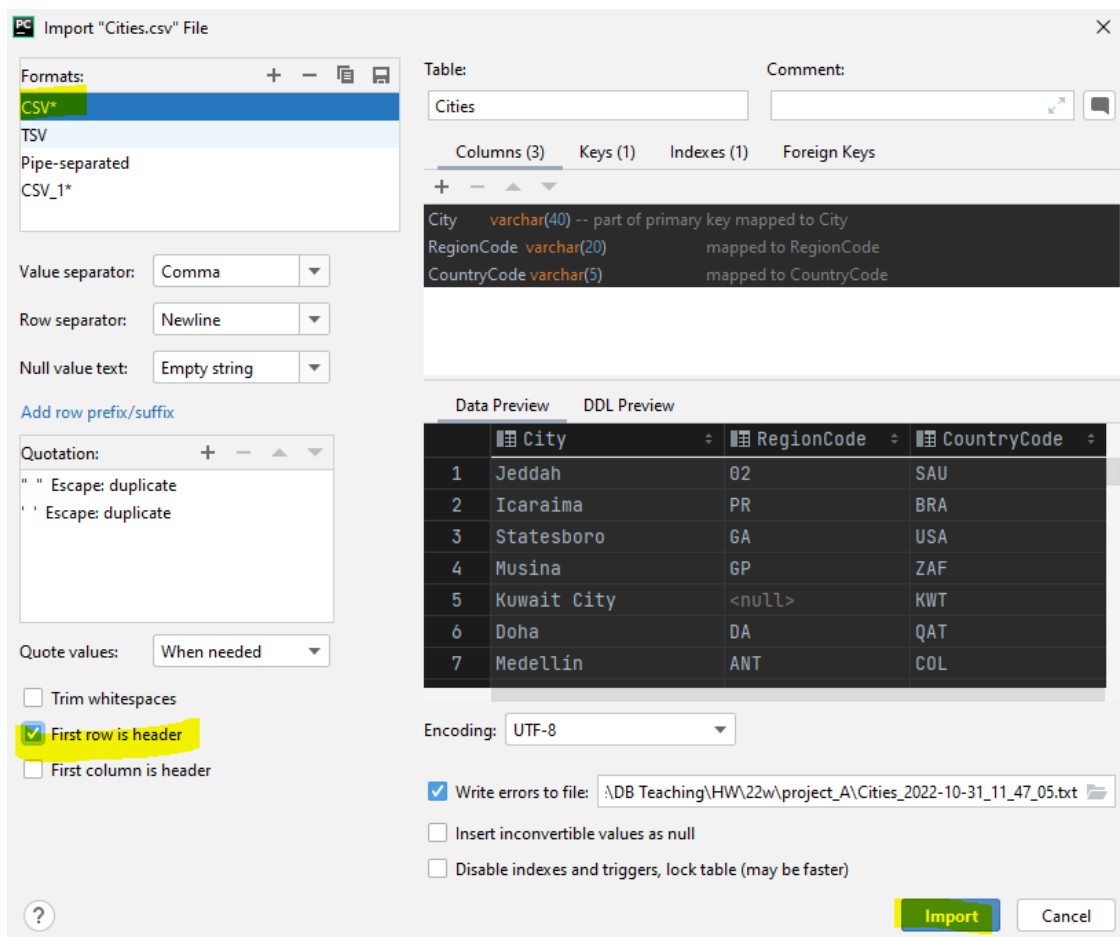
2. הזינו את הערכים המופיעים בקבצי ה-CSV המצורפים לתרגיל באמצעות טעינה ישירה שלהם למסד הנתונים. עליכם לעשות זאת בהתאם לסדר יצירת הטבלאות בקובץ יצירת הטבלאות.

3. ניתן לעשות זאת באמצעות באופן הבא :

א) לחיצה (עם הלחצן הימני של העכבר) על שם הרלציה, בחירה באפשרות של Import Data from File.

ב) בחירת קובץ ה-csv הרלוונטי מהתיקיה בה הוא יושב על מחשבכם.

ג) בתפריט שנבחר בחרו באפשרות של פורמט *CSV, סמנו את האפשרות של First row is header ולחצו על Import (כפי שמודגש בתמונה הבאה).



4. לאחר שטענתם את כל הרלציות ניתן להתחיל בעבודה.