

פרויקט קורס חלק א' - דיאגרמות ER ו-SQL

הנחיות להגשת התרגיל:

1. תאריך הגשה – 11.08.24 בשעה 23:55.
2. מתרגלת אחראי על המטלה: קורל שרף
3. יש להגיש את תרגיל הבית בזוגות בלבד. ציון המטלה של סטודנט שיגיש לבד ללא אישור יתחיל מ-90.
4. יש להגיש תיקיית ZIP ששמה הוא מספרי זהות של המגישים בפורמט: ID1_ID2 כאשר ID1, ID2 הם מספרי הזהות של הסטודנטים. התיקה 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 לשאלות 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 תוכן. בחלק א' (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.

תיאור המסד:

חברת שירותי תוכן חדשה הוקמה, ונקראתם לסייע לה למדל את מסד הנתונים שלה בהתאם לדרישות המצורפות.

יש לשמור במסד מידע אודות כל התכנים (Content) המתועדים באתר. כל תוכן מזוהה באמצעות שמו, ובנוסף יש לשמור את השפה המדוברת בו ואת תאריך העלאתו. האתר מציע מגוון רחב של תכנים, ביניהם סרטים (Movie) ושירים (Song). עבור כל סרט נרצה לשמור את הז'אנר שלו ועבור שיר את האורך שלו (בשניות). כמו כן, נרצה לשמור במסד מידע אודות אנשי תוכן. כל איש תוכן מזוהה באמצעות שמו והוא יכול להיות שחקן (Actor) או אמן (Artist). לא קיימים אנשי תוכן שלא שייכים ללפחות אחת מן הקבוצות, ויתכן ששחקן הוא גם אמן. נרצה לתעד לכל סרט את השחקנים ששיחקו בו ואת השכר שהם מקבלים עבור השתתפותם בסרט. יש לוודא כי לכל שחקן שהשתתף בסרט מתועד שכר שגבוה מ-10,000 דולר. בנוסף, לעתים נרצה לתעד מי האמן שביצע שיר כלשהו. לא ייתכן סרט שלא כולל אף שחקן, ולא ייתכן שמספר אמנים מבצעים שיר מסוים. כל אמן מבצע שיר אחד לפחות, ונרצה לתעד מהו השיר האהוב על האמן (לא בהכרח שיר שהוא מבצע). שחקנים עשויים לקנא בשחקנים אחרים. במקרים כאלו, נרצה לתעד את הדבר ולציין את הסיבה לקנאה. שימו לב כי אם שחקן א' מקנא בשחקן ב', אין הכרח שגם שחקן ב' מקנא בשחקן א'.

בנוסף, במסד נשמר מידע אודות כל המשתמשים במערכת. כל משתמש מזוהה על ידי תעודת זהות, ונשמר עבורו שמו, כתובת מייל והמדינה ממנה הוא מגיע. לכל משתמש יש מייל ייחודי. החברה מעוניינת לתעד את כל התכנים שמשתמש צרך (בין אם בצפייה או האזנה). משתמש יכול לצורך תוכן במספר ימים שונים, אך לא יתכן שצרך את אותו התוכן באותו היום יותר מפעם אחת. בנוסף, משתמש יכול לתת ציון (מספר שלם בין 1 ל-5) לתוכן אותו צרך שמבטא את מידת ההנאה שלו מהתוכן במועד בו צרך אותו.

לחברת התוכן מאגר של פרסומות (Advertisement) שהיא עשויה להציג למשתמש לפני הצגת התוכן אותו בחר. כדי למשוך את תשומת הלב של משתמשים, החברה מקפידה להציג בכל פרסומת אמן אחד בדיוק. לכל פרסומת יש מזוהה ייחודי ונשמר תיאור קצר של תוכן הפרסומת. במקרה שהוצגה פרסומת למשתמש בתחילת צריכת התוכן, נרצה לתעד מהי. לא יתכן שתופיע יותר מפרסומת אחת לפני הצגת התוכן.

משתמשים יכולים ליצור רשימת השמעה (Playlist) שכוללת מספר שירים. קיים לפחות שיר אחד בכל רשימת השמעה. לכל רשימת השמעה יש שם, והיא מזוהה על ידי השילוב של שמה ומזוהה המשתמש שיצר אותה. משתמשים ביקורתיים במיוחד עשויים לבדוק רשימות השמעה של משתמשים אחרים, ולסמן שירים בהם כלא מתאימים לרשימת ההשמעה. למשל, אם רשימת ההשמעה נקראת "שירים רגועים" ויש שיר קצבי, הוא עשוי להיות מסומן כלא מתאים על ידי משתמש כלשהו. נרצה לתעד כל מקרה בו משתמש סימן שיר ברשימת השמעה כשיר לא מתאים. אם שיר סומן כלא מתאים ברשימת השמעה יותר מ-5 פעמים, נרצה למחוק שיר זה מרשימת ההשמעה של המשתמש.

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 הישויות 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-50 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 שאלתת 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 השאלתת 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 שאלתת 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):

Movies (title, language, releaseDate)

ActorsInMovies (aName, title, salary)

Jealous (aName1, aName2)

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

תיאור הרלציות:

Movies – רלציה המתעדת את הסרטים.

- title – שם הסרט.
- language – השפה המדוברת בסרט.
- releaseDate – תאריך העלאת הסרט.

ActorsInMovies – רלציה המתעדת השתתפות של שחקן בסרט.

- aName – שם השחקן.
- title – שם הסרט.
- salary – השכר על השתתפות השחקן בסרט.

Jealous – רלציה המתעדת רגשי קנאות של שחקן אחד באחר.

- aName1 – שם השחקן שמפגין קנאות לשחקן 2.
- aName2 – שם השחקן ששחקן 1 מקנא בו.

(3) (15 נק') :

"שחקן על זמני" הוא שחקן שהשתתף בסרט כלשהו **בכל שנה**, מבין השנים בהם הועלו סרטים. "סרט משתלם" הוא סרט שהשכר **שכל** שחקן קיבל בו גדול או שווה מהשכר הממוצע של אותו שחקן שמחושב על פני כל הסרטים בהם השתתף.

מצאו את הסרטים המשתלמים ששפתם היא אנגלית (English) וכוללים **לפחות** שני שחקנים על זמניים. עבור סרטים אלו, החזירו את שם הסרט, מספר השחקנים בסרט והשכר הממוצע שניתן לשחקן בסרט (מעוגל כ-2 ספרות אחרי הנקודה). החזירו את התוצאה ממוינת בסדר עולה לפי השכר הממוצע. במקרה של שוויון בשכר הממוצע בין סרטים שונים, יש להדפיסם בסדר עולה לפי שם הסרט.

(4) (15 נק') :

"סרט נטול קנאה" הוא סרט **שאף** שחקן בו לא מקנא בשחקן אחר, אך **לכל** שחקן קיים שחקן אחר שמקנא בו. מצאו את השחקנים שמקיימים את שלוש התנאים הבאים :

- השחקן שיחק בסרט נטול קנאה אחד לפחות.
 - השחקן שיחק בלפחות 2 סרטים בעלי שפה שונה.
 - **לא קיים** סרט בו השחקן שיחק וקיבל את השכר הכי גבוה מבין השחקנים שמשחקים בסרט.
- עבור שחקנים אלו, החזירו את שם הסרט האחרון ששיחקו בו (הסרט שתאריך העלאתו הוא המאוחר ביותר) ותאריך העלאת הסרט. אם יש כמה סרטים שתאריך העלאתם הוא המאוחר ביותר, יש להחזיר את כולם.

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-4 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 ותבצע עבור מספר קלטים שונים. לתרגיל מצורף קובץ בשם `create_table_commands.sql` שכולל פקודות DDL ליצירת הטבלאות הנתונות. 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.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 242 (שימו לב כי קו תחתון מפריד בין המספרים):

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

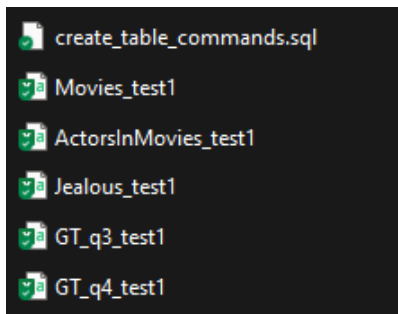
(2) שנו את שם את הקובץ `ID1_ID2_VIEWS.py` בהתאם למספרי הזהות שלכם. למשל, עבור זוג סטודנטים שמספרי הזהות שלהם הם 123456789 ו-987654321 יש לקרוא לקובץ `123456789_987654321_VIEWS.py`. בקובץ זה תכתבו את ה-`Views` שלכם ואותו תגישו.

(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 המסד של אחד מבני הזוג תחת השדות `DATABASE_NAME` ו-`NAME_USER` בשורות 244 ו-245.

```
SERVER_NAME = "sqlserverdds094241course.database.windows.net"
DATABASE_NAME = "Fname_Lname"
USER_NAME = "Fname_Lname"
PASSWORD = "Qwerty12!"
```

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



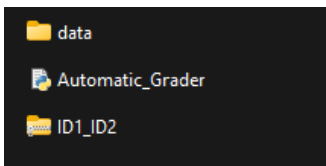
(6) בקובץ ששמו (לפני שתשנו אותו בהתאם למספרי הזהות שלכם) ID1_ID2_Views.py מוגדר מילון בשם VIEWS_DICT. למילון זה שני מפתחות – Q3 ו-Q4. הערך עבור כל אחד משני מפתחות אלו הוא רשימה שבה תזינו את ה-VIEWS שלכם. כל פקודת יצירת VIEW היא איבר ברשימה (סדר הופעת האיברים ברשימה חייב להיות כסדר יצירת ה-VIEWS הרצוי) המופיע בין "" ל-"". מזכירים שוב כי ניתן להשתמש ב-4 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": [
        ""
        ""
        ""
    ]
}
```

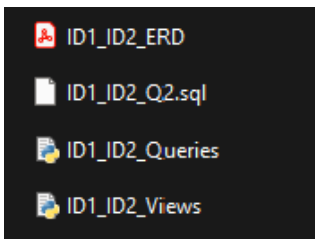
(7) בקובץ ששמו (לפני שתשנו אותו בהתאם למספרי הזהות שלכם) ID1_ID1_Queries.py מוגדר מילון בשם QUERY_ANSWERS. למילון זה שני מפתחות – Q3 ו-Q4. הערך אותו יש להזין עבור כל אחד ממפתחות אלו הוא השאילתה המתאימה. שימו לב כי עבור שתי השאלות, יש להחזיר את התשובה הסופית דרך שאילתה (זהו התוכן שמוזן בקובץ זה), ולא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



כאשר תיקיית ה-zip היא התיקיה אותה אתם מגישים שמכילה את הקבצים הבאים (עם החלפת השם ID1_ID2 במספרי הזהות שלכם):



(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.py.

(10) כעת, הריצו את הקובץ. אם אכן פתרונכם נכון עבור הקלט הנתון, הפלט אותו תראו יהיה:

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

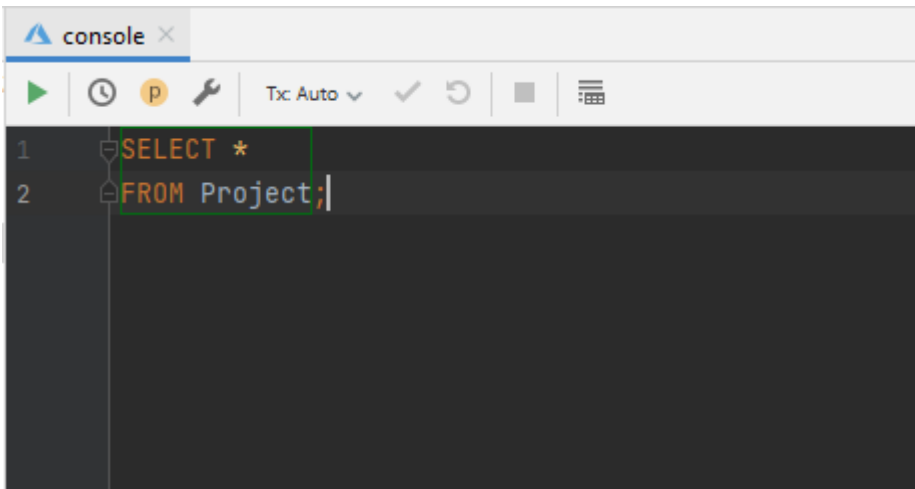
הערה חשובה- פתרונכם בשאלות 3 ו-4 ייבדק בצורה אוטומטית. באחריותכם לוודא כי הקפדתם על

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.py.
- לצורך הבדיקה – הזנת שם המסד של אחד מבני הזוג בשורות 245 ו-246 בקובץ Automatic_Grader.py.
- הגשת שני קבצי הפייתון 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 1 – שמירת תוכן שנכתב ב-Console

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, 'Tx: Auto', a checkmark, a refresh icon, a square, and a list icon. The code area shows two lines: '1 SELECT *' and '2 FROM Project;'. A green box highlights the text '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 – העלאת רשומות לרלציה מקובץ CSV

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 של הטסטים באמצעות טעינה ישירה שלהם למסד הנתונים. עליכם לעשות זאת בהתאם לסדר יצירת הטבלאות בקובץ יצירת הטבלאות.

ניתן לעשות זאת באמצעות באופן הבא :

א) לחיצה (עם הלחצן הימני של העכבר) על שם הרלציה, בחירה באפשרות של Import/Export ובה באפשרות Import Data from file.

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

ג) וידוא המיפוי של השדות המופיעים בקובץ לשדות ברלציה.

ד) לחיצה על OK ווידוא קבלת הודעת הצלחה על העלאת כל הרשומות.

3. לאחר שטענתם את כל הרלציות ניתן להתחיל בעבודה.