

פרויקט קורס חלק א' - דיאגרמות ER ו-SQL

הנחיות להגשת התרגיל:

1. תאריך הגשה – 18.12.22 בשעה 23:55.
2. מתרגל אחראי על התרגיל- בר גנוסר
3. 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 תכנת 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).
4. את הקובץ יש להגיש דרך אתר הקורס במקום המתאים ב-moodle על ידי **אחד** מבני הזוג.
5. איחור בהגשת התרגיל יגרור קנס בגובה 20% מהציון עבור כל יום איחור (פרט למקרים חריגים כגון מילואים. במקרים אלה יש לפנות טרם הגשת התרגיל).
6. 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:

חברת שירותי הדיגיטל "אל-נתק" מעוניינת למדל את מסד הנתונים שלה בצורה נוחה יותר על מנת לתעד מידע אודות עובדיה, לקוחותיה, הרגלי השימוש שלהם, התכנים שהיא מציעה ועוד. אתם מתבקשים לסייע להם בהתאם לדרישות המצורפות.

"אל נתק" שומרת מידע אודות העובדים השונים בה. כל עובד מזוהה באמצעות מספר הזהות שלו, ובנוסף נשמר שמו ושם המחלקה בה הוא עובד. כל מחלקה מזוהה באמצעות שם ייחודי ובנוסף מתועד תיאור מילולי קצר המכיל לכל היותר 100 תווים אודות תפקידה של המחלקה. חלק מהעובדים בחברה מוגדרים כמומחים טכניים. עבור כל עובד שכזה יש לשמור את תחום המומחיות שלו. כמו כן, חלק מהעובדים בחברה הם מנהלים, אשר יש לתעד את שכרם. ייתכן כי במסד מתועדים מנהלים שהם גם מומחים טכניים. עבור כל מחלקה יש לשמור את זהות המנהל שעומד בראשה.

"אל נתק" מעוניינת לתעד מידע אודות כל המשפחות המשתמשות בשירותיה. כל משפחה מזוהה באמצעות מזוהה ייחודי, ובנוסף נשמר גם שם המשפחה ורמת העושר של משק הבית (בציון הנע בין 1 ל-9, כך ש-1 הוא רמת העושר הנמוכה ביותר ו-9 היא הגבוהה ביותר). עבור כל משפחה המתועדת במסד נשמר מידע אודות לפחות אדם אחד מבני המשפחה. כל אדם המתועד במסד מזוהה חד ערכית באמצעות השילוב בין שמו הפרטי לבין מזוהה המשפחה שלו, ובנוסף נשמרים מספר הטלפון ותאריך הלידה שלו. "אל נתק" מתעדת במסד הנתונים את המשפחות לפי רמת העושר שלהם כך שמשפחות שציון רמת העושר שלהן הוא 7 ומעלה מוגדרות כמשפחות פרימיום, בעוד שמשפחות עם ציון רמת עושר נמוך יותר מוגדרות כמשפחות רגילות. "אל נתק" דואגת לשביעות רצונם של משפחות הפרימיום ועל כן לכל משפחה שכזו מוקצה מומחה טכני יחיד שמטפל בתקלות של המשפחה.

כאשר משפחה רגילה מעוניינת להתנתק משירותי החברה (ניתן להניח כי משפחות פרימיום לא רוצות להתנתק אף פעם), עליה להגיש בקשה למנהל כלשהו ולציין את הסיבה לבקשה. יש לתעד את כל הבקשות הללו, את תאריך הגשתן ואת ההחלטה הסופית בנוגע לבקשה. משפחה יכולה להגיש בקשה להתנתקות משירותי החברה לאותו המנהל בתאריכים שונים. לעיתים המנהל שאליו הוגשה בקשת ההתנתקות לא יודע כיצד לפעול ולכן ההחלטה מועברת לטיפול על ידי מנהל אחר שיקבל את ההחלטה. יש לתעד את מקרים אלו, ולציין את הסיבה להעברת הטיפול לידי מנהל אחר.

חברת "אל-נתק" מספקת ללקוחותיה ממירים מיוחדים, כך שלכל משפחה המתועדת במסד יש לפחות ממיר אחד. כל ממיר מקבל מספר סידורי בהתאם לסדר התקנתו בבית המשפחה בו נמצא (כלומר אין שני ממירים בעלי אותו מספר סידורי באותו הבית). ממיר כלשהו עשוי להתקלקל פעם אחת לכל היותר, ובמקרה זה הוא נשלח לתיקון אצל מומחה טכני במעבדה (אם מדובר במשפחת פרימיום אז הוא יישלח למומחה הטכני שאחראי על הטיפול בבעיות של משפחה זו). אם ממיר אכן נשלח מתישהו לתיקון יש לתעד את זהות המומחה שתיקן אותו ואת עלות התיקון.

בממירים של חברת "אל-נתק" מופיע מגוון רחב של ערוצים. כל ערוץ מזוהה חד ערכית לפי מספרו ובנוסף נשמר שמו. עבור כל ממיר יש לתעד את כל הערוצים הזמינים בו (בכל ממיר יש לפחות ערוץ זמין אחד). בערוצים, כפי שניתן להניח כמובן, משודרות תכניות. כל תכנית מזוהה באמצעות שמה, ובנוסף נשמר הז'אנר שלה ואורכה. במסד מתועד לוח השידורים של כל התכניות בכל הערוצים. בלוח זה נשמרים כל מועדי ההקרנות של תכניות בערוצים (ייתכן כי אותה התכנית מוקרנת בערוץ כלשהו יותר מפעם אחת). "אל נתק" מעוניינת לעקוב אחר הפעילות בממירים שברשותה, ולכן כל פעולה של מעבר ערוץ המבוצעת בממיר כלשהו מתועדת במסד. עבור פעולה שכזו יש לתעד את הממיר, מספר הערוץ אליו התבצע המעבר והמועד בו בוצעה הפעולה.

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 המגבלה 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-100 תווים. 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 בארבע הרלציות המוגדרות כדלקמן (שימו לב כי הרלציות הללו לא בהכרח תואמות בדיוק לאלו שמוגדרות בסיפור שניתן עבור שאלות 1 ו-2):

Households (hID, size, netWorth)

Devices (dID, hID)

Programs (pCode, title, genre, duration)

Viewing (dID, eTime, pCode)

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

תיאור הרלציות:

Households – רלציה שכוללת מידע אודות משפחות.

- hID – מזהה המשפחה.
- size – מספר הנפשות בבית המשפחה.
- netWorth – רמת העושר של המשפחה.

Devices – רלציה שכוללת מידע אודות ממירים

- dID – מזהה ממיר.

Programs – רלציה שכוללת מידע אודות תכניות

- pCode – מזהה תכנית.
- title – שם התכנית.
- genre – הז'אנר של התכנית
- duration – משך התכנית (בדק')

Viewing – רלציה שכוללת מידע אודות צפיה בתכניות

- eTime – הזמן (בפורמט DATETIME) בו התרחש המעבר לתכנית.

(3) (15 נק'):

משפחה עם קלאס היא משפחה בת 3 נפשות **לפחות** שרמת העושר שלה **גבוהה מ-5** ושבתקופה המתועדת במסד לא נצפתה תכנית מז'אנר Reality באף אחד מהממירים שבביתה. עבור כל משפחה עם קלאס החזירו את המזהה שלה ואת מספר הממירים בביתה.

(4) (15 נק'):

תכנית פופלארית היא תכנית בה צפו לפחות 3 משפחות שונות ושמשכה (duration) ארוך יותר ממשך כל התכניות האחרות מאותו הז'אנר (תכנית שהז'אנר שלה הוא NULL לא יכולה להיחשב כתכנית פופלארית). משפחה מודרנית היא משפחה בה צפו בתקופה המתועדת במסד בלפחות 3 תכניות פופלאריות. עבור כל משפחה מודרנית החזירו את מזהה המשפחה, את **שם** התכנית הפופלארית הראשונה בה צפו בבית בתקופה המתועדת במסד (כלומר שמועד המעבר אליה הוא מועד המעבר הראשון לתכנית פופלארית שהתרחש באותו בית) ואת מועד המעבר לערוץ בה שודרה תכנית זו. יש למיין את התוצאה בסדר עולה לפי מועד המעבר (מהמוקדם למאוחר), ובמקרה שהתרחש מעבר אצל שתי משפחות שונות באותו הזמן בדיוק להחזיר קודם את הרשומה שבה מזהה המשפחה קטן יותר. ניתן להניח לשם פשטות כי לא ייתכן שבאותו הרגע בדיוק התבצע מעבר משני ממירים שונים באותו הבית לתכנית פופלארית.

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.
לא לשכוח לסיים כל שאילתה ופקודת יצירת view ב-;.
עבור כל אחת משתי השאילתות מותר להשתמש ב-5 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. 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_Checker.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 5. 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"
```

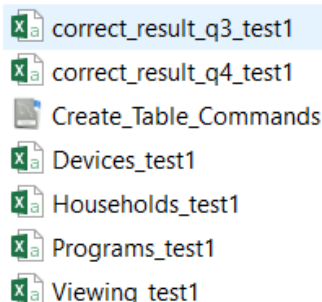
(2) שנו את שם את הקובץ `ID1_ID2_VIEWS.py` בהתאם למספרי הזהות שלכם. למשל, עבור זוג סטודנטים שמספרי הזהות שלהם הם 123456789 ו-987654321 יש לקרוא לקובץ `123456789_987654321_VIEWS.py`. בקובץ זה תכתבו את ה-`Views` שלכם ואותו תגישו.

(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) אין צורך להגיש גם את הקובץ `Automatic_Checker.py`.

(5) הזינו את שם המשתמש הטכניוני של אחד מבני הזוג תחת השדות `DATABASE_NAME` ו-`NAME_USER` בשורות 8 ו-9 (לא לשכוח להחליף " " ב-0 במקרה הצורך).

ודאו כי הקבצים המצורפים מופיעים בתיקיה בשם `data`, אשר יושב בתיקיה המרכזית בה ממוקם הקובץ `Automatic_Checker.py`. תוכן התיקיה הוא:



- correct_result_q3_test1
- correct_result_q4_test1
- Create_Table_Commands
- Devices_test1
- Households_test1
- Programs_test1
- Viewing_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- ל-". מוזכרים שוב כי ניתן להשתמש ב-VIEWS 5 לכל היותר. אמנם גם תחרגו ממגבלה זו הטסט עשוי לעבור, אך על המסך הודעה המתריאה על החריגה (ובבדיקה שלנו הטסט ייכשל). מבנה הזנת המידע הוא:

```
VIEWS_DICT = {
    "Q3": [
        """CREATE VIEW V1
        AS
        SELECT...;"""
        """CREATE VIEW V2
        AS
        SELECT...;"""
    ],
    "Q4": [
        """CREATE VIEW V1
        AS
        SELECT...;"""
        """CREATE VIEW V2
        AS
        SELECT...;"""
    ]
}
```

(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) 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_Checker.py.

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

```
Test number 1
Correct result for query 3
Correct result for query 4
Test number 1 finished
*****
```


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

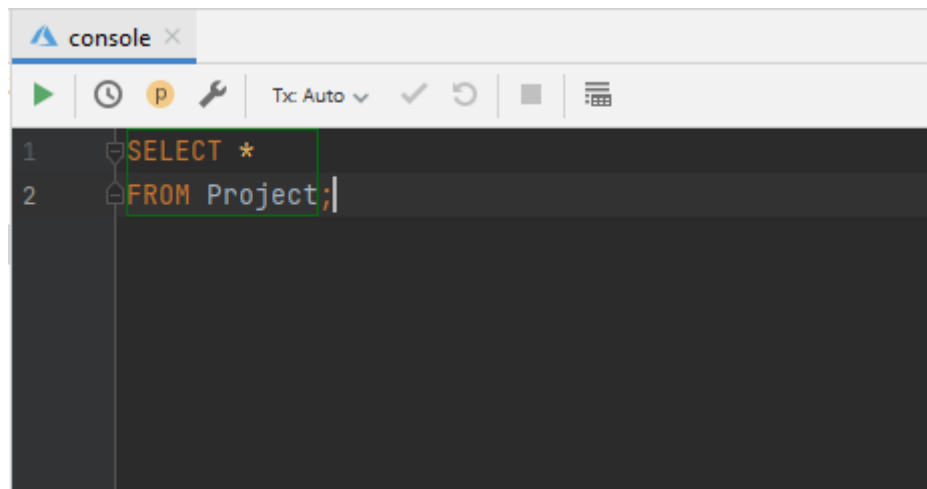
הנחיות ההגשה המפורטות בקובץ זה. אם הבדיקה האוטומטית נכשלה ציונכם בתרגיל יהיה 0. כמוכן שתוכלו להגיש ערעור והתרגיל ייבדק מחדש אך ציונכם יתחיל מ-90 (אלא אם יתברר שלא חרגתם מאף אחת מההנחיות). אנא ודאו כי ההגשה שלכם מקיימת את כל סעיפי ה-checklist הבא :

- 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 5 בקובץ Automatic_Checker.py.
- לצורך הבדיקה – הזנת שם המשתמש של אחד מבני הזוג בשורות 8 ו-9 בקובץ Automatic_Checker.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

איך לשמור את התוכן הנכתב 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 shows 'console' with a close button. Below the title bar is a toolbar with icons for running, debugging, and other actions. The main area of the console shows two lines of SQL code: '1 SELECT *' and '2 FROM Project;'. The text is highlighted with a green selection box.

(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

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

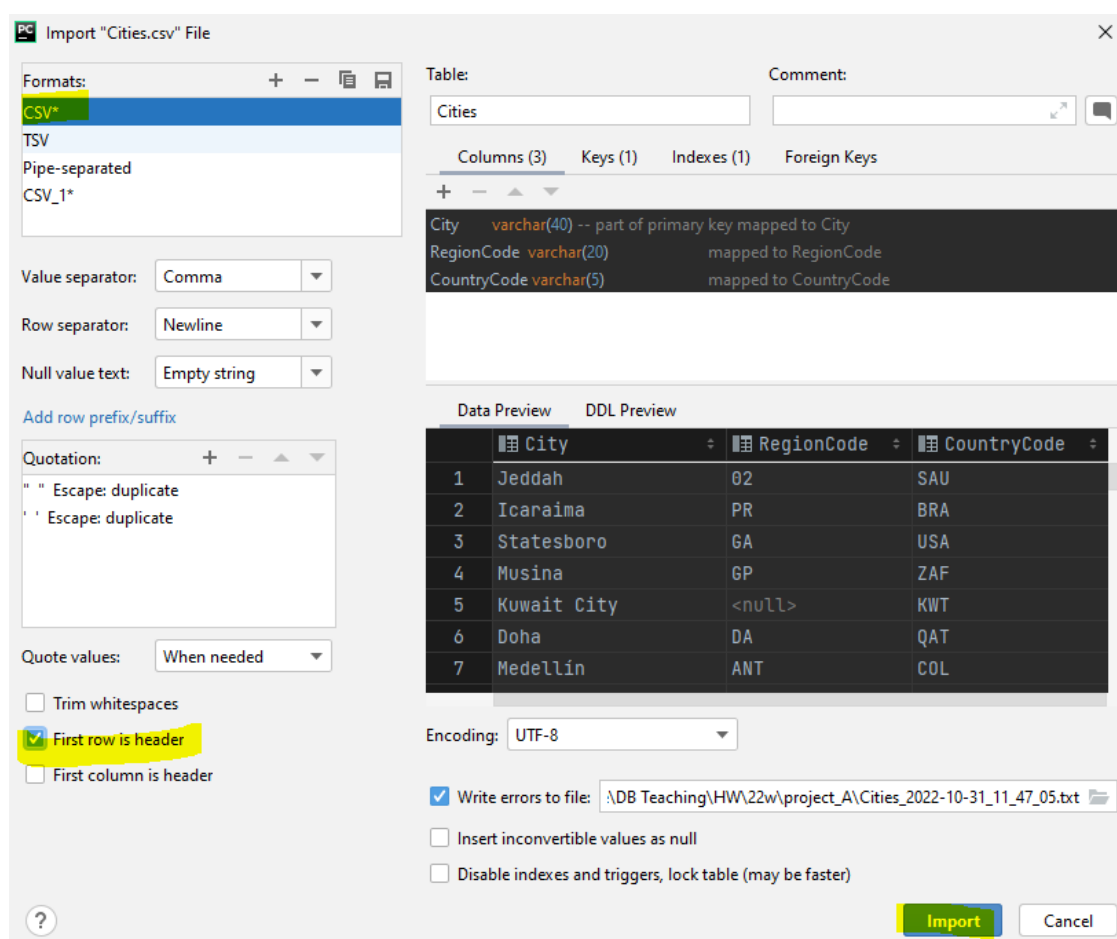
2. הזינו את הערכים המופיעים בקבצי ה-CSV המצורפים לתרגיל באמצעות טעינה ישירה שלהם למסד הנתונים. עליכם לעשות זאת בהתאם לסדר יצירת הטבלאות בקובץ יצירת הטבלאות.

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 4 הרלציות ניתן להתחיל בעבודה.