

תרגיל בית מספר 1 – DDL, SQL

הנחיות להגשת התרגיל:

1. תאריך הגשה - 27/11/22 בשעה 23:55.
2. מתרגלת אחראית על התרגיל - קורל שרף.
3. יש להגיש את תרגיל הבית בזוגות בלבד. הגשה ביחידים שלא תחת אישור מיוחד תתחיל מניקוד נמוך יותר.
4. יש להגיש תיקיית zip הכוללת שני קבצים בלבד: פתרון בקובץ sql לשאלה 1 (בנושא DDL) ופתרון לשאלות 2-4 בקובץ python (ראו הנחיות בסוף מסמך זה). שם תיקיית ה-zip צריך להיות מספרי זהות של המגישים בפורמט: ID1_ID2 כאשר ID1, ID2 הם מספרי הזהות של הסטודנטים. שם קובץ ה-sql (שאלה 1) צריך להיות ID1_ID2_q1. למשל, עבור זוג סטודנטים שמספרי הזהות שלהם הם 123456789 ו-987654321, התשובות לשאלה הראשונה יוגשו בקובץ sql בשם 123456789_987654321_q1, ושם תיקית הזיפ אותה הם יגישו יהיה 123456789_987654321.
5. את הקובץ 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 או סיבה בריאותית). במקרים אלה יש לפנות למתרגל האחראי על 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 המופיע בעמודים 5-6 של מסמך זה).

חלק א' - DDL (40 נקודות)

הקדמה:

"עבודה לכל דורש" היא חברת השמה להייטק שמתוכננת בין מעסיקים (חברות הייטק) לעובדים. מנכ"ל החברה מינה אתכם לאחראים על יצירת מסד הנתונים בהתאם לדרישות המתוארות כדלקמן.

"עבודה לכל דורש" מציעה משרות במספר חברות הייטק איתן היא עובדת. עבור כל חברת הייטק כזו נרצה לשמור פרטים אודותיה. כל חברה מזוהה על ידי מספר ייחודי ובנוסף נרצה לשמור את שמה, השנה בה נוסדה, כתובת האינטרנט שלה ואת תחום עיסוקה (למשל, "תוכנה"). לא ייתכנו 2 חברות בעלות כתובת אינטרנט זהה. בנוסף, נרצה לוודא שכל כתובת אינטרנט של חברה מסתיימת ב-"com".

כל חברת הייטק מחולקת למחלקות (למשל, מחלקת פיתוח, הנדסה, שיווק). מחלקה מזוהה על ידי שם ייחודי בחברה בה היא נמצאת ונרצה לתעד תיאור קצר אודות עיסוקי המחלקה. בהתאם לדרישות החברות, נרצה לוודא שבכל חברת הייטק לא קיימות 2 מחלקות בעלות תיאור קצר זהה.

"עבודה לכל דורש" מציעה מגוון של תפקידים במחלקות של חברות הייטק. כדי לשמור על אחידות בין העובדים שיגויסו לחברות ההייטק השונות, החברה מעוניינת לתעד במסד טבלת שכר עבור התפקידים השונים. כל תפקיד מזוהה על ידי שם התפקיד (למשל, "Data scientist") וניתן להעסיק בתפקיד עובדים ברמות ניסיון שונות. עבור כל תפקיד ורמת ניסיון נרצה לשמור את השכר עבור צירוף זה. יש לוודא שרמת הניסיון היא ערך מבין סט הערכים "Executive"/ "Director"/ "Associate"/ "Entry Level" ושכל תפקיד ורמת ניסיון שמתועדים בטבלה השכר קיים ערך לשכר (מספר גדול ממש מ-0).

חברות ההייטק פונות אל "עבודה לכל דורש" בבקשה לפרסום משרה בתפקיד באחת ממחלקות החברה. יש לתעד מידע אודות משרות אלו במסד. בעת פרסום משרה, היא מקבלת מספר ייחודי שבאמצעותו היא מזוהה. עבור כל משרה נרצה לשמור את המחלקה והחברה בה המשרה המוצעת, התפקיד המוצע והניסיון שנדרש עבור המשרה. ניתן להניח שהמשרה דורשת עובד בעל הניסיון המצויין בדרישות בלבד. עבור כל צירוף של שם התפקיד והניסיון הדרוש מתועד שכר בטבלת השכר. יש לשמור עבור כל משרה מהו היקף המשרה (מספר השעות החודשיות שנדרשות מעובד שיועסק). במקרה שמשרה מדווחת ללא ציון היקף המשרה, יש להניח כי מדובר במשרה מלאה המוגדרת כ-182 שעות חודשיות. לא תיתכן משרה המתועדת במסד מבלי שכל הפרטים בה מוזנים.

החברה כוללת מגייסים שתפקידם לגייס עובדים למשרות פנויות. כל מגייס מזוהה על ידי תעודת זהות (מספר בן 9 ספרות) ובנוסף נשמר גם שמו. מגייס שמאייש משרה שפורסמה מקבל בונוס לשכר שלו. על כן, במקרה שמשרה אוישה נרצה לתעד עבורה מיהו המגייס שגייס עובד עבורה. לא ייתכן ש-2 מגייסים שונים או יותר גייסו עובד לאותה משרה.

שאלה 1 (40 נקודות):

עליכם לכתוב סקריפט DDL אשר מייצר את הטבלאות בהתאם לתיאור הנתון. אם קיימת דרישה אשר לא ניתן לממש באמצעות DDL אנא ציינו זאת במפורש והסבירו מדוע לא ניתן לעשות זאת. כתבו את ההסבר בקובץ ה-sql (כהערה (באמצעות '--')). ניתן להניח ששדה מסוג מחרוזת הוא באורך של לכל היותר 50 תווים.

חלק ב' – SQL (60 נקודות):

בבעלות החברה תיאור **חלקי** של מסד נתונים המתעד מידע אודות עובדים בחברות הייטק, פרויקטים בחברות אלו והקצאת עובדים לפרויקטים בחברות לפי חודשים. המסד מכיל את הרלציות הבאות:

Employee (eID, compID, depName, hourlyWage)

Project (compID, depName, pID, budget)

WorkOnProject (compID, depName, pID, wDate, eID, hours)

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

תיאור הרלציות:

Employee – רלציה אשר מתעדת את העובדים בחברות ההייטק.

- eID – מזהה העובד. מספר טבעי ייחודי.
- compID – מזהה חברת ההייטק בה העובד מועסק. מספר טבעי ייחודי בקרב החברות השונות.
- depName – שם המחלקה בחברת ההייטק בה העובד מועסק. משתנה מטיפוס מחרוזת באורך של לכל היותר 50 תווים. כל עובד מועסק בדיוק במחלקה אחת בחברת ההייטק יחידה.
- hourlyWage – שכר שעתי של העובד. מספר טבעי. לא קיים עובד שלא מוגדר לו שכר שעתי.

Project – רלציה שכוללת מידע על פרויקטים שמתבצעים במחלקות בחברות ההייטק.

- compID – מזהה חברת ההייטק.
- depName – שם מחלקה בחברת ההייטק.
- pID – מזהה פרויקט. מספר טבעי. ייתכנו 2 פרויקטים בעלי מזהה פרויקט זהה שמתבצעים במחלקה שונה באותה חברת ההייטק או בחברות הייטק שונות.
- budget – התקציב שמוקצה לפרויקט. מספר טבעי. לא קיים פרויקט שלא מוגדר לו תקציב.

WorkOnProject – רלציה אשר מתעדת את עבודת העובדים על פרויקטים לפי חודשים.

- compID – מזהה חברת ההייטק.
- depName – שם המחלקה בחברת ההייטק.
- pID – מזהה פרויקט.
- wDate – החודש בשנה בו עובד עובד על הפרויקט. משתנה מטיפוס תאריך בפורמט 'YYYY-MM-DD'. לצורך פשטות, נניח שהיום נשמר עבור התאריך הוא תמיד 01. למשל, התאריך שיתועד עבור עבודה על פרויקט בחודש מאי 2019 הוא 2019-05-01.
- eID – מזהה העובד. מפתח זר לרלציה Employee.
- hours – מספר שעות העבודה שעובד עבד על הפרויקט בחודש מסוים. מספר טבעי. לא קיים תיעוד של עבודת עובד על פרויקט בחודש ללא ציון מספר שעות העבודה.
- קיים אילוץ מפתח זר על שלוש השדות compID, depName, pID לרלציה Project. ניתן להניח כי עובד יכול לעבוד רק בפרויקטים המתקיימים במסגרת המחלקה שלו.

עליכם לענות על שאילתות ה-SQL הבאות תוך שימוש בטבלאות הנתונות בחלק ב' בלבד.

שאלה 2 (20 נקודות):

כתבו שאילתת SQL אשר מוצאת עבור כל מחלקה המתועדת במסד את מספר העובדים ששייכים למחלקה זו שעבדו על פרויקט כלשהו במשך לפחות שישה חודשים (לא בהכרח רצופים).

שאלה 3 (20 נקודות):

השכר שעובד מקבל על פרויקט מוגדר כמספר השעות הכולל שהעובד עבד על הפרויקט כפול השכר השעתי שלו. "פרויקט חורג" מוגדר כפרויקט שהתקציב שמוגדר עבורו קטן ממש מהשכר הכולל שקיבלו כל העובדים שעבדו על הפרויקט.

עליכם לכתוב שאילתת SQL שמוצאת את החברות בהן אין אף פרויקט חורג. עבור החברות הללו, החזירו את התקציב הממוצע של הפרויקטים שהתקיימו בה. החזירו את התוצאה בצורה ממוינת בסדר יורד לפי התקציב הממוצע. ניתן להניח שלא קיימות 2 חברות בעלות תקציב ממוצע זהה.

שאלה 4 (20 נקודות):

"עובד מוכשר" הוא עובד שלקח חלק בכל הפרויקטים שהתבצעו במחלקה אליה הוא שייך, ובנוסף הוא העובד בעל השכר השעתי הגבוה ביותר בה. כתבו שאילתת SQL המחזירה את מזהי החברות שבהן התקיים לפחות פרויקט אחד שבו השתתף לפחות עובד מוכשר אחד.

הערה חשובה - אסור להשתמש בשלב זה של הקורס בכלים כמו Views או With לצורך יצירת טבלאות ביניים. רק בהמשך הקורס נלמד כיצד לעשות זאת.

בהצלחה!

הערות חשובות

באחריותכם לוודא שמבנה הרציה (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_PartB.sql` שכולל פקודות DDL ליצירת הטבלאות הנתונות. בנוסף, מצורפים קבצי קלט ופלט לדוגמה. תיאור הקלטים, אופן העלאת הנתונים לטבלאות והפלטים הנכונים מצורפים בנספח 2 שבסוף המסמך.

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.

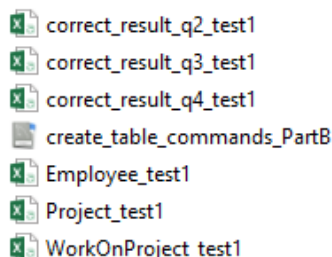
(1) הזינו את מספרי הזהות של שני השותפים בפורמט `ID1_ID2` תחת המשתנה `IDS` שבשורה 5. למשל, עבור זוג סטודנטים שמספרי הזהות שלהם הם 123456789 ו-987654321, יש לכתוב בשורה 5 (שימו לב כי קו תחתון מפריד בין המספרים):

```
IDS = "123456789 987654321"
```

(2) שנו את שם את הקובץ `ID1_ID2.py` בהתאם למספרי הזהות שלכם. למשל, עבור זוג סטודנטים שמספרי הזהות שלהם הם 123456789 ו-987654321 יש לקרוא לקובץ `123456789_987654321.py`. בקובץ זה תכתבו את השאילתות שלכם ואותו תגישו. **אין צורך להגיש גם את הקובץ `Automatic_Checker.py`.**

(3) 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_q2_test1`
- `correct_result_q3_test1`
- `correct_result_q4_test1`
- `create_table_commands_PartB`
- `Employee_test1`
- `Project_test1`
- `WorkOnProject_test1`

(4) בקובץ ששמו **(לפני שתשנו אותו בהתאם למספרי הזהות שלכם)** `ID1_ID2.py` מוגדר מילון בשם `QUERIES_ANSWER`. למילון זה שלושה מפתחות – `Q2`, `Q3` ו-`Q4`. הערך אותו יש להזין עבור כל אחד ממפתחות אלו הוא השאילתה המתאימה. 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`.

לדוגמה, נניח (וזהו כמובן לא המצב באמת) שהפתרון לכל אחת מהשאלות הוא החזרת כל הרשומות מהרלציה Project. במקרה זה המילון ייראה כך:

```
QUERIES ANSWER = {
  "Q2":
    """
    SELECT *
    FROM Project;
    """

  "Q3":
    """
    SELECT *
    FROM Project;
    """

  "Q4":
    """
    SELECT *
    FROM Project;
    """
}
```

מדגישים שוב – זהו הקובץ שעליכם להגיש (לצד הפתרון של שאלה 1 בנושא DDL).

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

```
Test number 1
Correct result for query 2
Correct result for query 3
Correct result for query 4
Test number 1 finished
*****
```

הערה חשובה – פתרונכם ייבדק בצורה אוטומטית. באחריותכם לוודא כי הקפדתם על הנחיות ההגשה

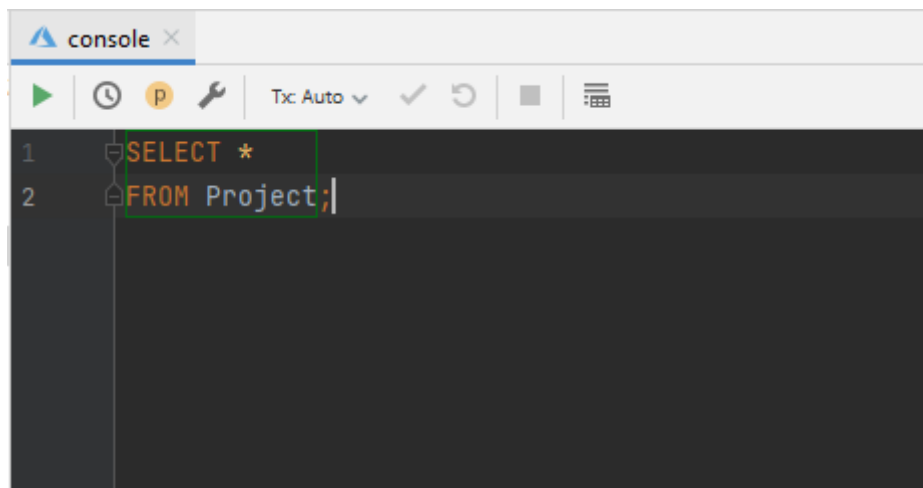
המפורטות בקובץ זה. אם הבדיקה האוטומטית נכשלה ציונכם בתרגיל יהיה 0. כמובן שתוכלו להגיש ערעור והתרגיל ייבדק מחדש אך ציונכם יתחיל מ-90 (אלא אם יתברר שלא חרגתם מאף אחת מההנחיות). אנא ודאו כי ההגשה שלכם מקיימת את כל סעיפי ה-checklist הבא:

- הגשת קובץ SQL שפורמט שמו ID1_ID2_q1 (יש להחליף את ID1 ו-ID2 במספרי הזהות של הסטודנטים המגישים) עבור פקודות ה-DDL המבוקשות בשאלה 1.
- 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 הקובץ שפורמט שמו הוא ID1_ID2.py (יש להחליף את ID1 ו-ID2 במספרי הזהות של הסטודנטים המגישים) כאשר במילון המוגדר בקובץ זה יש להזין את פתרונותיכם לשאלות 2-4.
- 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 at the top says 'console' with a close button. Below the title bar is a toolbar with icons for running, debugging, and other functions. 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 rectangular 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_PartB.sql מתוך ה-console.

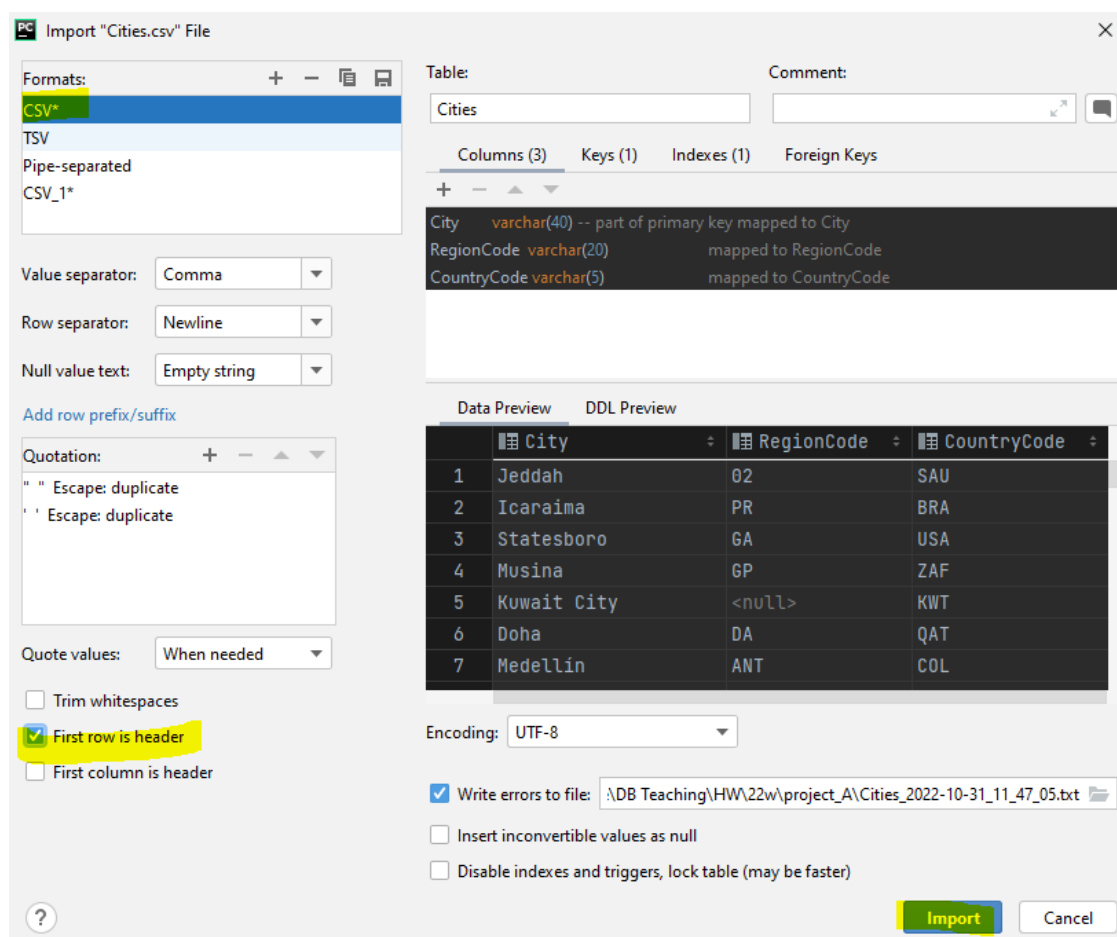
2. הזינו את הערכים המופיעים בקבצי ה-CSV המצורפים לתרגיל (Employee_test1, Project_test1 ו-WorkOnProject_test1) באמצעות טעינה ישירה שלהם למסד הנתונים. עליכם לעשות זאת בהתאם לסדר יצירת הטבלאות בקובץ יצירת הטבלאות.

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.