

# מבחן ב-SQL פייטון

חלק א' – בחר תשובה אחת נכונה (50 נקודות):

1. מה ההבדל בין HAVING ל-WHERE?

- 1 WHERE משמש לסינון שורות לפני קיבוץ הנתונים, ואילו HAVING מסנן לאחר ביצוע GROUP BY.
- 2 HAVING משמש לסינון לפני הקיבוץ, ואילו WHERE מסנן לאחר GROUP BY.
- 3 אין הבדל ביניהם – ניתן להשתמש בשניהם בכל שלב של השאילתה.
- 4 HAVING משמש רק במסדי נתונים של PostgreSQL, בעוד WHERE קיים בכל מערכות ה-SQL.

2. מה ההבדל בין RIGHT JOIN ל-LEFT JOIN?

- 1 LEFT JOIN מחזיר את כל הרשומות מהטבלה השמאלית ואת ההתאמות מהימנית, בעוד RIGHT JOIN מחזיר את כל הרשומות מהטבלה הימנית ואת ההתאמות מהשמאלית.
- 2 RIGHT JOIN מבצע סינון על הנתונים ומחזיר רק רשומות עם התאמה מלאה, בעוד LEFT JOIN מחזיר את כל הנתונים משתי הטבלאות.
- 3 LEFT JOIN מחזיר רק רשומות מהטבלה השמאלית שיש להן התאמה בטבלה הימנית, בעוד RIGHT JOIN מחזיר רשומות מהטבלה הימנית גם אם אין להן התאמה בטבלה השמאלית.
- 4 LEFT JOIN ו-RIGHT JOIN הם זהים לחלוטין ואין הבדל ביניהם.

3. מה מטרת DISTINCT ב-SQL?

- 1 הוא מחליף את GROUP BY כאשר רוצים למנוע כפילויות.
- 2 הוא מסנן את כל הנתונים ומחזיר רק רשומה אחת מכל טבלה.
- 3 הוא מחזיר רשומות ייחודיות בעמודות שנבחרו ומסיר כפילויות.
- 4 הוא גורם לשאילתא לרוץ מהר יותר מאשר SELECT \*.

4. כיצד ניתן להגביל את מספר השורות המוחזרות ב-SQLite?

- 1 שימוש ב-TOP
- 2 שימוש ב-LIMIT
- 3 שימוש ב-MAXROWS
- 4 שימוש ב-FIRST

5. מהו FOREIGN KEY ומה מטרתו?

- 1 הוא יוצר קשר בין שתי טבלאות כדי להבטיח שלמות נתונים.
- 2 הוא מזהה ייחודית רשומה בתוך הטבלה.
- 3 הוא משמש כתחליף ל-PRIMARY KEY.
- 4 FOREIGN KEY קיים רק ב-PostgreSQL.

6. מה קורה כאשר ON DELETE CASCADE מופעל על FOREIGN KEY?

- 1 הרשומות בטבלה הראשית נמחקות והנתונים בטבלת הבן נשמרים.
- 2 הרשומות בטבלת הבן נמחקות כאשר הרשומה בטבלה הראשית נמחקת.
- 3 הרשומות בטבלה הראשית מוגנות מפני מחיקה.
- 4 אין השפעה, יש צורך למחוק ידנית את הרשומות בטבלת הבן.

7. מה עושה COUNT(name) בשאילתת SQL? (בהנחה שקיימת עמודת name בטבלה)

- 1 סופר את מספר הרשומות בטבלה.
- 2 מחזיר את הערך הגבוה ביותר בעמודה.
- 3 מחזיר את כל הנתונים בטבלה.
- 4 סופר רק את הערכים שאינם NULL.

8. מהי מטרת DEFAULT בעת יצירת טבלה?

- 1 קובעת ערך ברירת מחדל עבור עמודה במקרה שלא מוזן ערך.
- 2 קובעת שהערך חייב להיות ייחודי.
- 3 מבטיחה שהערך בטור חייב להיות מספרי.
- 4 הופכת את העמודה למפתח ראשי.

9. איזו שאילתת JOIN תחזיר את כל הנתונים משתי הטבלאות, גם אם אין התאמה ביניהן?

- 1 INNER JOIN
- 2 LEFT JOIN
- 3 RIGHT JOIN
- 4 FULL OUTER JOIN

10. מהו CHECK ב-SQL?

- 1 מגביל אילו ערכים יכולים להיכנס לעמודה מסוימת.
- 2 מאפשר להכניס ערכים כפולים לטבלה.
- 3 מגדיר מפתח ראשי לעמודה.
- 4 משמש להמרת נתונים בין מספרים לטקסט.

## חלק ב' – כתיבת שאילתות ב-SQL (40 נקודות):

### 1. יצירה ואכלוס של מסד נתונים:

רוצים למחשב חנות למכירת טלוויזיות

- יש להקים מסד נתונים עם המידע הבא:

- id – מזהה ייחודי לכל טלוויזיה (מספר רץ, לצורכי החנות)
- catalog\_number – מספר קטלוגי עולמי
- brand – שם המותג: Samsung, LG, Sony
- model – דגם הטלוויזיה
- screen\_size – גודל המסך באינצ'ים
- resolution – רזולוציית המסך: 4K, 8K, Full HD
- price – מחיר הטלוויזיה
- stock\_quantity – כמות במלאי
- release\_year – שנת יציאה לשוק
- smart\_tv – האם הטלוויזיה חכמה (כן/לא)
- Os – מערכת הפעלה רק במידה והטלוויזיה חכמה, אחרת יופיע NULL
- panel\_type – סוג הפאנל: OLED, QLED, LED

א. כתוב שאילתא ליצירת טבלה עם כל המידע המופיע לעיל. בחר בסוגי שדות מתאימים השתמש היכן שצריך ב- default, check, unique וכו' (ניתן לפצל ליותר מטבלה אחת אם רוצים)

ב. כתוב 3 שאילתות של INSERT להוספת טלוויזיות

### 2. שאילתות GROUP BY

יש ליצור את מסד הנתונים -סרטים- מהלינק הבא:

[https://github.com/pythonai250824/05.02.2025\\_exam/blob/main/movies.sql](https://github.com/pythonai250824/05.02.2025_exam/blob/main/movies.sql)

movies	
123	id
A-Z	movie_name
A-Z	genre
A-Z	country
A-Z	language
123	year
123	revenue

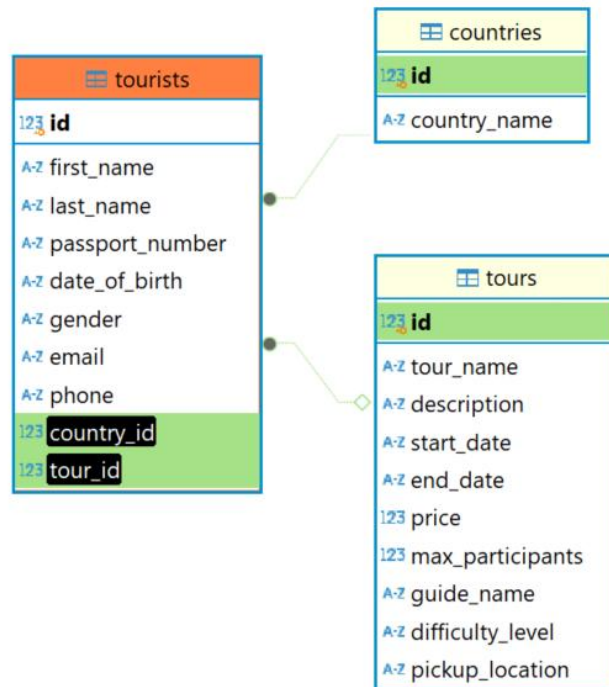
כתוב שאילתות כדי למצוא:

- ספור כמה סרטים יש מכל GENRE?
- כמה רווח revenue היה בכל שנה בתעשיית הסרטים?
- מה ממוצע הרווחים לכל ז'אנר?
- מה ממוצע הרווחים לכל ז'אנר עבור כל שפה בנפרד?
- באיזו שפה יש הכי פחות סרטים? רמז: מניין את הקבוצות בסדר עולה ובחר את הראשון
- לאיזו מדינה יש הכי הרבה סרטים? רמז: מניין את הקבוצות בסדר יורד ובחר את הראשון
- הצג את הז'אנרים בהם יש יותר מ- 2 סרטים
- הצג שנים (years) בהם סך הרווחים היה גדול מ- 1,000
- הצג שפות בהן יש לפחות 3 סרטים

### 3. שאלות JOIN

יש ליצור את מסד הנתונים -תיירים- מהלינק הבא:

[https://github.com/pythonai250824/05.02.2025\\_exam/blob/main/tourist.sql](https://github.com/pythonai250824/05.02.2025_exam/blob/main/tourist.sql)



- א. הצג את רשימת התיירים ביחד עם המדינות מהם הגיעו. רמז: INNER JOIN
- ב. הצג את רשימת התיירים ואת פרטי הטיול אליו הם משובצים, רק לתיירים משובצים
- ג. הצג את רשימת כל התיירים ואת פרטי הטיול אליו הם משובצים, היכן שהתייר איננו משובץ יופיע NULL בעמודות של פרטי הטיול
- ד. הצג את רשימת כל התיירים ואת פרטי הטיול אליו הם משובצים, היכן שהתייר איננו משובץ יופיע NULL בעמודות של פרטי הטיול + וגם את פרטי כל הטיולים, היכן שאין תיירים משובצים לאותו הטיול יופיע NULL בעמודות של פרטי התייר. רמז: FULL JOIN
- ה. הצג את רשימת התיירים אשר אינם משובצים לשום טיול בשאלתא נפרדת, כתוב שאלתא המוחקת את אחד התיירים שלא נרשם לשום טיול
- ו. הצג את רשימת הטיולים אשר אין שום תייר משובץ אליהם בכדי שיהיה מספיק זמן להירשם, שנה את תאריך הטיולים (שאין להם בכלל נרשמים), שנה אחת קדימה. רמז: שאלתא UPDATE
- ז. ספור לכמה טיולים אין שום תייר משובץ (אפילו לא אחד)
- ח. הצג את כל הצירופים האפשריים של כל התיירים מול כל הטיולים

חלק ג' – הרצת שאלות ב-SQL דרך הפייטון (10 נקודות):

4. השתמש במסד הנתונים -סרטים- משאלה 2
  - א. כתוב קוד בפייטון המציג את כל הסרטים
  - ב. כתוב קוד בפייטון הקולט מהמשתמש שם או חלק מ- שם של סרט ומחזיר את כל הסרטים אשר מכילים את אותו השם, רמז: LIKE
  - לדוגמא אם נקלט השם "batman" יוצג הסרט – "the batman"
  - ג. כתוב קוד בפייטון הקולט מהמשתמש פרטי סרט ואז מבצע שאלתא INSERT להוספת הסרט לטבלת הסרטים

**\*חלק ד' – בונוסים/רשות (15 נקודות לשאלה):**

5. מימוש many : many

בשאלת התיירים, שנה את ה- CREATE בטבלת התיירים- והסר את ה- FK של הטיול

צור טבלה נוספת של מס' טיול ומס' תייר

הוסף באמצעות INSERT מספר טיולים ומספר תיירים משובצים

6. צור את טבלת הסרטים ב- PostgreSQL

כתוב SP אשר מקבל שנה ומחזיר את רוח כל הסרטים באותה השנה

הזמן את ה- SP באמצעות SELECT



הגשת המבחן:

- יש להגיש את המבחן לאימייל: [pythonai250824+sqlexam@gmail.com](mailto:pythonai250824+sqlexam@gmail.com)
- יש לציין שם פרטי + משפחה + את ימי הקורס בהגשה
- פתרון לשאלות האמריקאיות נא לכתוב בקובץ TXT כחלק מה- GIT
- יש לייצר REPO ב- GIT **נפרד** להגשה ביום המבחן
- יש לייצר REPO ב- GIT **נפרד** להגשה של תשובות שלא ביום המבחן
- נא להגיש כל תשובה לשאלה בקובץ נפרד
- בשאלות מרובות סעיפים- נא לציין מעל לכל תשובה לאיזה סעיף היא שייכת
- ציון עובר: 60%

בהצלחה!