# CASE

### נתונה הטבלה הבאה:

**מפקד היחידה ביקש לדעת את מחזור הגיוס וסטטוס השירות של קבוצת חיילים.**

**כיצד נעשה זאת?**

## ביטוי CASE

### מהו ביטוי ה- CASE

עד כה השתמשנו בתנאים אך ורק בפסוקית הWHERE -, כעת נכיר פעולה חדשה, ביטוי הCASE -שנותן לנו **מבנה לוגי** (בסגנון IF/THEN שאנו מכירים מפייתון) המאפשר להחזיר ערכים שונים כתלות בערכי עמודה אחרת.

**הגדרה**: דרך ליישם תנאי לוגי (IF/THEN) בו נוכל להגדיר ערכים שונים לעמודה על פי תנאי או תנאים המתבססים על ערכיה של עמודה ו-או עמודות אחרות.

בדרך זו נוכל לפרמט נתונים, לבצע aggregations , pivot tables, ליצור טווחים (bins) וליישם שליפות יותר מורכבות.

### מבנה של פקודת שליפה עם פסוקית CASE בסיסית.

SELECT col1,

col2,

col3,

CASE

WHEN col1 < 10 THEN 'Below 10'

WHEN col1 = 10 THEN 'Equal 10'

WHEN col1 > 10 THEN 'Above 10'

ELSE [optional] 'No Value'

END AS col1\_name

FROM table1

אחרי ה- **SELECT** יבואו שמות העמודות שנרצה לקבל מהטבלאות + ביטוי **CASE.**

**חוקים לביטוי CASE**

* בתוך ביטוי ה- CASE יגיעו התנאים והערך שיוחזר בצורה הבאה:

WHEN [condition] THEN [result value]

* כל הפונקציות LIKE, BETWEEN) וכו') והאופרטורים (AND, OR) שניתנים לשימוש בתנאי WHERE ניתנים לשימוש בתנאי CASE.
* לאחר כתיבת התנאים ניתן להוסיף פקודת **ELSE** ואחריה ערך אשר יוחזר (במידה ואף תנאי לא מומש), פקודת ה- ELSE הינה אופציונלית ובמידה ותושמט יוחזר NULL במקום.
* ביטוי CASE חייב להיגמר בפקודת **END** אשר מכריזה על סוף הביטוי. לאחר END נותנים שם לעמודה בעזרת AS.

**שימו לב:** סדר כתיבת התנאים הוא קריטי, בריצה, מנוע הSQL - עובר שורה שורה ובודק אם הערך מהשורה עומד באחד התנאים. בדיקת התנאים תתבצע לפי סדר הופעתם כך שברגע שתנאי הוא TRUE בדיקת התנאים תיעצר, התנאים שלאחריו לא יבדקו והמנוע יעבור לשורה הבאה.

אחרי ה – **FROM**  יהיה שם הטבלה הראשונה (מרכזית).

אחריה – **WHERE** ישנה אפשרות להוסיף תנאים נוספים לסינון הטבלה.

**דוגמה**

SELECT   תיאור מקצוע,

שם חייל,

CASE

WHEN 1 = סטטוס שרות THEN 'סדיר'

WHEN 2 = סטטוס שרות THEN 'קבע'

WHEN 3 = סטטוס שרות THEN 'משוחרר'

END AS 'סטטוס שירות'

FROM טבלת החיילים

### שליפת CASE מ- DVD RENTAL

**כמה סרטים המתאימים לקבוצות הגיל 12-13, 14-17, 18+ יש במלאי של החנות בקנדה? (G/PG, PG-13/R, NC-17)**

SELECT CASE

WHEN film.rating IN ('G', 'PG') THEN '12-13'

WHEN film.rating IN ('PG-13', 'R') THEN '14 - 17'

WHEN film.rating = 'NC-17' THEN '18+'

ELSE 'no rating'

END AS suitable\_age,

COUNT(inv.inventory\_id) AS movies

FROM inventory inv

INNER JOIN store str ON

str.store\_id = inv.store\_id

INNER JOIN address adrs ON

adrs.address\_id = str.address\_id

INNER JOIN city ON

city.city\_id = adrs.city\_id

INNER JOIN country cnt ON

cnt.country\_id = city.country\_id

INNER JOIN film ON

film.film\_id = inv.film\_id

WHERE cnt.country = 'Canada'

GROUP BY suitable\_age;

כמו שניתן לראות בדוגמה, אפשר להשתמש בJOIN בשליפה עם CASE, השאילתה תתבצע לפי הסדר המוכר לנו כבר.

FROM ß JOIN/WHERE ß SELECT(בשלב זה רץ ביטוי הCASE)

הסר את המלבן הלבן בשביל לחשוף את התשובה

תיאור השליפה:

1. שלב ראשון: שולפים מטבלת המלאי, מחברים אליה את טבלאות חנות ß כתובות ß ערים ß מדינות בשביל לדעת את המדינה בה נמצאת החנות, ובנוסף מחברים את טבלת הסרטים לטבלת המלאי בשביל לקבל את המידע על כל הסרטים במלאי.

לבסוף יתבצע סינון של כל השורות של חנויות שלא בקנדה (סדר הביצוע יכול להשתנות בהתאם לתוכנית הביצוע של ה - DB אך התוצאה תהיה זהה(.

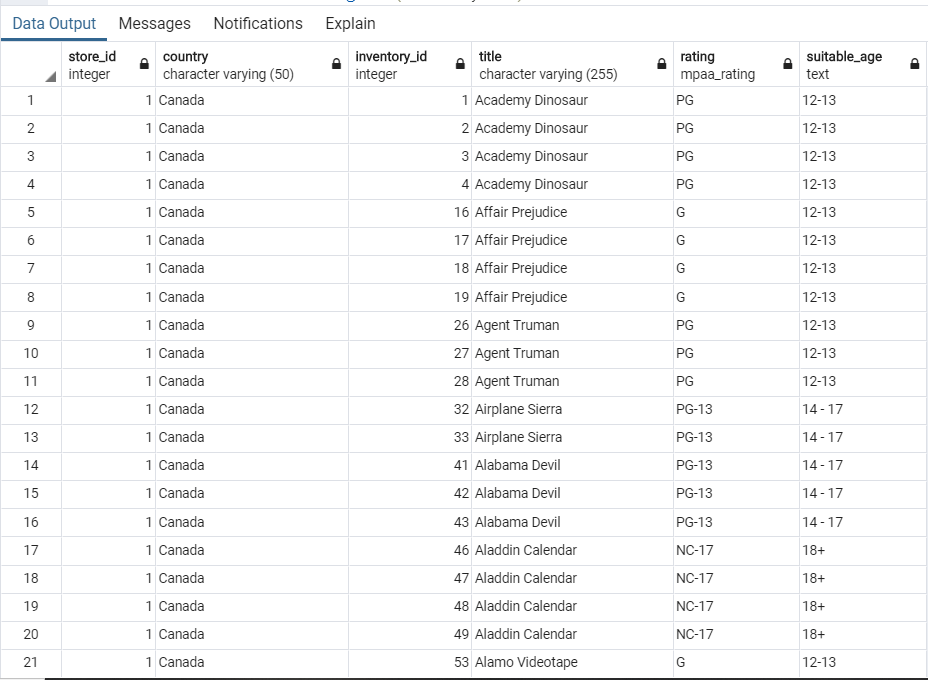
1. שלב שני: ביצוע ביטוי הCASE - בשלב זה. מנוע ה - SQL יעבור שורה שורה ויבדוק אם ה - rating של הסרט שמצוין במלאי נמצא ב (G,PG) אם כן תגדיר את הערך '12-13' עבור השורה בעמודה החדשה suitable\_age, אם לא תעבור לתנאי הבא וכך הלאה.

במידה ו rating - השורה לא עומד באף תנאי (אנו מניחים שאין דירוג) יוחזר .'no rating'

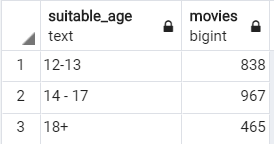
בסוף שלב זה צריכה להיות לנו טבלה זהה אך עם עמודה נוספת בשם suitable\_age שמכילה את הערכים שציינו ב – CASE עבור כל שורה.

1. שלב שלישי: חלוקה לקבוצות, פסוקית ה - GROUP BY מחלקת את השורות לקבוצות לפי הערכים בsuitable\_age ופונקציית COUNT סופרת את ה inventory\_id - שבכל קבוצה.

**\*שלבים 2 ו - 3 מוצגים בסדר לוגי לשם הבנת התהליך אך יכולים להתבצע בסדר שונה או בו זמנית לפי תוכנית הביצוע של ה - .DB הטבלה לאחר שלב 2:**

****

**הטבלה שתחזור לאחר שלב 3 תראה כך:**

****

קיבלנו טבלה יפה ונוחה שמסכמת את כמות הסרטים בחנות שבקנדה שמתאימים לכל קבוצת גיל.

**אך מה אם היינו משנים את השאלה ורוצים לשלוף את אותו הדבר עבור החניות בכל הארצות?**

כיצד נעשה זאת?

**כמה סרטים המתאימים לקבוצות הגיל 12-13, 14-17, 18+ יש במלאי של החניות בכל הארצות? (G/PG, PG-13/R, NC-17)**

SELECT cnt.country,

CASE

WHEN film.rating IN ('G', 'PG') THEN '12-13'

WHEN film.rating IN ('PG-13', 'R') THEN '14 - 17'

WHEN film.rating = 'NC-17' THEN '18+'

ELSE 'no rating'

END AS suitable\_age,

COUNT(inv.inventory\_id) AS movies

FROM inventory inv

INNER JOIN store str ON

str.store\_id = inv.store\_id

INNER JOIN address adrs ON

adrs.address\_id = str.address\_id

INNER JOIN city ON

city.city\_id = adrs.city\_id

INNER JOIN country cnt ON

cnt.country\_id = city.country\_id

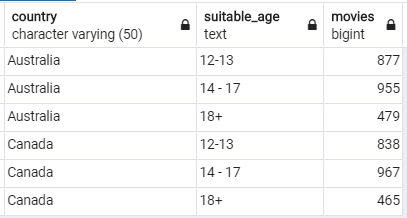
INNER JOIN film ON

film.film\_id = inv.film\_id

GROUP BY cnt.country, suitable\_age

ORDER BY cnt.country, suitable\_age;

הסר את המלבן הלבן בשביל לחשוף את התשובה

**טבלת התוצאה**

כמו שניתן לראות קיבלנו טבלה דומה לטבלה הקודמת שגם היא מסכמת את המידע בצורה נוחה.

אך האם יש דרך נוספת להציג את המידע הנ"ל?

האם ניתן לתמצת את הטבלה לכדי טבלה נוחה יותר?

**PIVOT TABLE**

**PIVOTING**

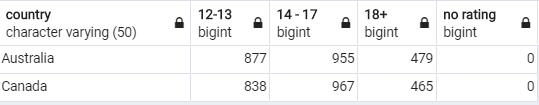
פיבוטינג היא פעולה שמשמעותה לסובב את הטבלה על ציר, משמע, **עמודות הופכות לשורות או לחילופין שורות הופכות לעמודות**.

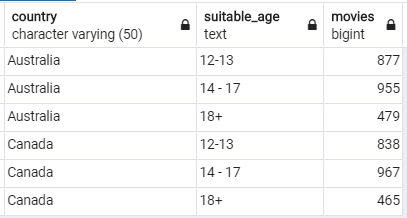
בעזרת שימוש בCASE - בתוך אגרגציות נוכל להשיג את הטבלה הרצויה:

**דוגמה**

**מצוי**

**רצוי**



****

**פתרון**

SELECT cnt.country,

COUNT(CASE

WHEN film.rating IN ('G', 'PG') THEN 1

ELSE NULL

END) AS "12-13",

COUNT(CASE

WHEN film.rating IN ('PG-13', 'R') THEN 1

ELSE NULL

END) AS "14 - 17",

COUNT(CASE

WHEN film.rating = 'NC-17' THEN 1

ELSE NULL

END) AS "18+",

COUNT(CASE

WHEN film.rating NOT IN ('G', 'PG', 'PG-13', 'R', 'NC-17')

THEN 1

ELSE NULL

END) AS "no rating"

FROM inventory inv

INNER JOIN store str ON

str.store\_id = inv.store\_id

INNER JOIN address adrs ON

adrs.address\_id = str.address\_id

INNER JOIN city ON

city.city\_id = adrs.city\_id

INNER JOIN country cnt ON

cnt.country\_id = city.country\_id

INNER JOIN film ON

film.film\_id = inv.film\_id

GROUP BY cnt.country

ORDER BY cnt.country;

למעשה רוב השליפה נשארת זהה אך במקום לספור את inventory\_id ניתן להוסיף עמודה אחת של CASE ולחלק לקבוצות על פיה. אנו מנצלים את העובדה ש - COUNT לא סופר NULL ויוצרים ארבע עמודות COUNT שסופרות תנאי CASE, אם הוא TRUE מחזיר 1, משמע סופר, אם הוא FALSE מחזיר NULL משמע לא סופר.

למעשה אנחנו סופרים עבור כל תנאי כמה פעמים הוא יוצא TRUE כשכל תנאי הוא עמודה בפני עצמו, לאחר מכן אנחנו מחלקים לקבוצות על פי מדינות.

כעת עשינו PIVOT לטבלה, לקחנו את העמודה sutiable\_age ואת הערכים בעמודה (שורות) ויצרנו מספר עמודות (עבור כל ערך ייחודי).

**האם ניתן להשתמש ב - CASE רק בפסוקית ה- SELECT?**

**התשובה הקצרה:**

לא, אך לרוב השימוש יהיה בSELECT - .

**התשובה הארוכה:**

ניתן להשתמש ב - CASE בפסוקיות SELECT, WHERE, ORDER BY, GROUP BY, JOIN, אך לרוב השימוש יהיה ב SELECT/GROUP BY לשליפה של ערכים על בסיס תנאים ויצירת קבוצות בהתאם.

בפסוקית ה - ORDER BY נוכל להשתמש בCASE - לשם סידור מותנה של השורות.

דוגמה – שליפת כל העובדים בחברה לפי הא-ב אך לשים את המנכ"ל למעלה בדרך כלל יהיה שימושים להגשת דוחות טבלאיים שיוגשו ישר מן הDB - .

בפסוקית ה - WHERE וה - JOIN לרוב משתמשים ב - CASE לשם יצירת תנאים דינמיים (תחום שלא נעמיק בו).

**סיכום**

לביטוי הCASE - פונקציונליות רחבה ושילובו בפקודות הSELECT - יעזור לנו לשכלל את השליפות, בין אם זה פרמוט המידע, יצירת קבוצות ויצירה של PIVOT TABLE -.

**זכרו!**

* CASE משומש לרוב בSELECT - .
* כל ביטוי CASE חייב לכלול WHEN, THEN, END אך ELSE הוא אופציונלי.
* ניתן לכתוב כל תנאי ולהשתמש בכל אופרטור לוגי (כמו ב - (WHERE בין WHEN ל - THEN.
* ניתן לכתוב מספר מקרים WHEN בתוך ביטוי CASE.