

מסדי נתונים – תרגיל בית 2

Basic Database Functions

CreateTables

יצרנו את היחסים הבסיסיים שהם Photo, Disk, Ram, תוך כדי שאנו שומרים על ההכנסות והמגבלות של הערכים.

הוספנו שני יחסים כדי לתמוך בשאר הפעולות באופן יעיל, המפתחות שלהם זרים מהיחסים הבסיסיים.

- DiskPhoto(Photo ID, Disk ID) - טבלה שנותנת גישה למציאת תמונות בדיסק מסוים.
- DiskRam(Disk ID, RAM ID) - טבלה שמקשרת בין דיסק ל-Ram.

ClearTables

שמירה הטבלאות תוך כדי מחיקת כל הרשומות הקיימות.

DropTables

מחיקת כל היחסים תוך שימוש בניהול טרנזקציות.

CRUD API

AddPhoto

הכנסת תמונה ליחס Photo תוך כדי בדיקת הערכים.

GetPhotoByID

גישה ליחס Photo והחזרת התמונה המתאימה למזהה הנתון. מתבצע תוך כדי בדיקת תקינות הערכים.

DeletePhotoByID

נבצע את הפעולות הבאות:

- ניגש ליחס Photo ונמחק את הרשומה המתאימה לנתונים שקיבלנו, נשמור את גודל התמונה.
- ניגש ליחס DiskPhoto ונמחק את הרשומה המתאימה לתמונה, במידה וקיימת (יכול להיות שהתמונה לא מקושרת לדיסק מסוים)
- ניגש ליחס Disk ונעדכן את המקום הפנוי בו.

AddDisk

הכנסת דיסק ליחס Disk תוך כדי בדיקת הערכים.

GetDiskByID

גישה ליחס Disk והחזרת הדיסק המתאים למזהה הנתון. מתבצע תוך כדי בדיקת תקינות הערכים.

DeleteDisk

גישה לטבלה Disk ומחיקת הרשומה המתאימה למזהה שהתקבל, מכיוון שב-DiskPhoto ו-DiskRAM יש מפתחות זרים, הם יימחקו גם משם.

AddRAM

הכנסת RAM ליחס Ram תוך כדי בדיקת הערכים.

GetRAMByID

גישה אל הטבלה RAM והחזרת ה RAM המתאים תוך כדי בדיקת ערכים.

DeleteRAM

גישה לטבלה RAM ומחיקת הרשומה המתאימה למזהה שהתקבל, מכיוון שב-DiskRAM הוא מפתח זר, הוא יימחק גם משם.

AddDiskAndPhoto

גישה אל היחסים Photo ו-Disk, והוספת האובייקטים תוך כדי בדיקת תקינות הערכים.

Basic API

AddPhotoToDisk

נבצע את הפעולות הבאות:

- נכניס את התמונה והדיסק אל היחס DiskPhoto על מנת לבדוק שהרשומה הזו לא חזרה על עצמה.
- במידה והייתה הכנסה, כלומר הרשמה לא הייתה קיימת, נמחק את הרשומה.
- כעת נכניס שוב את הרשומה תוך כדי בדיקת תקינות הערכים.
- נעדכן את המקום הפנוי בדיסק הרלוונטי על ידי גישה ליחס Photo, מציאת גודל התמונה ועדכון של גודל המקום הפנוי בדיסק על ידי גישה ליחס Disk.

RemovePhotoFromDisk

נבצע את הפעולות הבאות:

- בדיקה כי התמונה אכן קיימת בדיסק על ידי גישה ליחס DiskPhoto.
- מחיקה של הרשומה הרלוונטית מהיחס DiskPhoto.
- עדכון המקום הפנוי בדיסק הרלוונטי על ידי גישה ליחס Photo, מציאת גודל התמונה ועדכון של גודל המקום הפנוי בדיסק על ידי גישה ליחס Disk.

AddRAMToDisk

נבצע את הפעולות הבאות:

- נכניס את ה-RAM והדיסק אל היחס DiskRAM על מנת לבדוק שהרשומה הזו לא חזרה על עצמה.
- במידה והייתה הכנסה, כלומר הרשמה לא הייתה קיימת, נמחק את הרשומה.
- כעת נכניס שוב את הרשומה תוך כדי בדיקת תקינות הערכים.

RemoveRAMFromDisk

גישה אל היחס DiskRAM ומחיקה של הרשומה הרלוונטית.

AveragePhotosSizeOnDisk

נבצע את הפעולות הבאות:

- ניגש לטבלה DiskPhoto ובבחר רק את הרשומות בעלות המזהה המתאים לדיסק.
- ניגש ליחס Photo ונשתמש באגרציה AVG על גודל התמונות של התמונות המתאימות.

GetTotalRAMOnDisk

נבצע את הפעולות הבאות:

- ניגש לטבלה DiskRAM ובבחר רק את הרשומות בעלות המזהה המתאים לדיסק.
- ניגש ליחס RAM ונשתמש באגרציה SUM על גודל ה-RAM של הרשומות המתאימות.

GetCostForDescription

נבצע את הפעולות הבאות:

- נמחק את ה-Views הרלוונטים כדי להמנע מגישה לערכים לא נכונים (נפרט בסעיף הבא על Views)
- נוסיף את ה-Views הבאים על מנת לפשט:
 - Photos ID Size - שומר את כל התמונות בעלות התיאור שקיבלנו ואת הגודל שלהן.
 - Disk ID CPB - שומר את מזהה הדיסק ואת עלות פר בייט של הדיסק המתאים.
 - Disk IDS2 -

GetPhotosCanBeAddedToDisk

ניגש אל הטבלה Photo ובבחר את 5 התמונות שהגודל שלהם קטן או שווה לגודל הפנוי בדיסק שאת מספר ה-disk_id שלו קיבלנו כקלט. לאחר מכן נחזיר רשימה ובה 5 התמונות בסדר יורד לפי ה-photo_id שלהן.

GetPhotosCanBeAddedToDiskAndRAM

תחילה ניצור view (בשם RAM_PART_OF_DISK) של גודל הראם בדיסק המבוקש בעזרת הטבלה DiskRam ששומרת כל דיסק ואת הראם המשווין לו. נשמור מבט נוסף (בשם RAM_SIZED) שיעשה join בין היחס RAM לבין RAM_PART_OF_DISK וכך נקבל את ה-RAM של הדיסק המבוקש. לבסוף ניצור view שיכיל את כל התמונות שגודלן קטן מסכום ה-RAM ב-RAM_SIZED וגם קטן מ-disk_free_space של הדיסק הרצוי. מתוך ה-view האחרון נחזיר את 5 התמונות בלעות המזהה נמוך ביותר.

IsCompanyExclusive

נייצר VIEW שמכיל את כל החברות שייצרו את ה-RAM בדיסק (בשם RAM_IN_DISK_COMPANY) המבוקש ובנוסף ניצור VIEW שישמור את שם החברה שייצרה את הדיסק (בשם DISK_COMPANY_NAME). לבסוף נגדי VIEW שישמור את כל החברות שנמצאות ב RAM_IN_DISK_COMPANY אך לא ב DISK_COMPANY_NAME. נפעיל COUNT על התוצאה ונחזיר אמת רק במידה ו-COUNT=0.

IsDiskContaininAtLeastNumExists

תחילה, נייצר VIEW (בשם MATCH_PHOTOS) שמכיל את כל התמונות שלהן תיאור דומה לתיאור שקיבלנו בקלט, לאחר מכן נגדיר VIEW נוסף (בשם MATCH_DISKS) שיכיל INNER JOIN בין DiskPhoto לMATCH_PHOTO. לבסוף נבצע GROUP BY disk_id על MATCH_DISKS ונחזיר רק את הדיסקים שהופיע מספר Num של פעמים ומעלה.

GetDisksContainingTheMostData

תחילה נבצע LEFT JOIN בין DiskPhoto ל-Disk ישמרו ונקבל רשימה של כל הדיסקים עם התמונות המשוייכות להם. על התוצאה נפעיל שוב LEFT JOIN עם Photo כל שכל ערכי ה-JOIN הראשוני ישמרו ונקבל רשימה של כל הדיסקים עם התמונות המשוייכות אליהם וגודלן. לבסוף נעשה GROUP BY disk_id ונחזיר את 5 הדיסקים בהם תפוס המקום הגדול ביותר (בעזרת SUM על photo_size) בסדר יורד, ומיון משני לפי disk_id בסדר עולה.

Advanced API

getConflictingDisks

תחילה ניצור VIEW (בשם PHOTO_COUNT_OF_DISKS) שיכיל את מספר הדיסקים השונים בהם כל תמונה נמצאת, לאחר מכן ה VIEW בשם PHOTO_COUNT_GREATER_THAN_ONE ישמור את כל התמונות שנמצאות ביותר מדיסק אחד. בVIEW בשם RESULT נשמור את כל הדיסקים שמכילים את התמונות שנמצאות ב PHOTO_COUNT_GREATER_THAN_ONE ללא כפילויות ממויינים בדרך עולה לפי disk_id, את החיפוש נעשה בעזרת יחס DiskPhoto.

mostAvailableDisks

ראשית ניצור VIEW (בשם DISK_ID_DISK_SPEED_NUMBER_OF_PHOTOS) שישמור את תוצאת ה LEFT JOIN בין Disk לPhoto לפי התנאי שגודל התמונה קטן או שווה מהשטח הפנוי בדיסק, נעשה GROUP BY disk_id ונשמור לכל דיסק את מספר הקבצים שיכולים להיות מאוחסנים בו. לאחר מכן נגדיר VIEW נוסף (בשם RESULT) שבו נסדר את ערכי DISK_ID_DISK_SPEED_NUMBER_OF_PHOTOS לפי הסדר שנדרש בשאלה ונחזיר את 5 הדיסקים הראשונים.

getClosePhotos

ניצור VIEW שיכיל את כל הדיסקים בהם התמונה שקיבלנו בקלט נמצאת. לאחר מכן ניצור VIEW (בשם allPhotos) שישמור את כל התמונות ששונות מתמונת הקלט, בשלב הבא ניצור VIEW (בשם CountCommon) שייספור עבור כל תמונה את מספר הדיסקים המשותפים לה ולתמונת הקלט, ואת הVIEW Count2 שישמור את מספר הדיסקים המשותפים לכל תמונה מCountCommon. VIEW נוסף בשם comb יחזיר התמונות שמצאנו ב Count2 ללא כפילויות בעזרת שאילתה מקוננת המשתפת את allPhotos. הVIEW הסופי יחזיר את התמונות מcomb שמקיימות את התנאי וחולקות לפחות 50 אחוז מהדיסקים עם תמונת הקלט.

חלק יבש

1. השאילתה מחזירה זוגות של סטודנטים ($n1, n2$) ששניהם לקחו לפחות פעם אחת את אותו הקורס באותו סמסטר, כך שהשם של $n1$ קטן לקסיקוגרפית מ $n2$ ושהציון הכי נמוך של $n2$ בקורסים המשותפים גדול מהציון הכי נמוך של $n1$ בקורסים המשותפים.

2.

StudentName	CourseName	Grade	Semester
Eitan	DB	88	A
Bar	DB	95	A
Bar	CS	70	B
Eitan	CS	80	B
Bar	OS	82	C

השאילתה תחילה תבצע Join עם התנאים הבאים:

FROM Learns L1, Learns L2

WHERE l1.StudentName < l2.StudentName

AND l1.CourseName = l2.CourseName

AND l1.Semester = l2.Semester

ולכן נקבל את הטבלה הבאה:

l1.StudentName	l1.CourseName	l1.Grade	l2.Semester	l2.StudentName	l2.CourseName	l2.Grade	l2.Semester
Bar	DB	95	A	Eitan	DB	88	A
Bar	CS	70	B	Eitan	CS	80	B

לאחר מכן נבצע GROUP BY עם התנאים הבאים:

GROUP BY I1.StudentName, I2.StudentName

HAVING COUNT(DISTINCT I1.CourseName) = COUNT(Distinct I2.CourseName)

AND MIN(I1.Grade) < MIN(I2.Grade)

ונקבל:

I1.StudentName	MIN(I1.Grade)	I2.StudentName	MIN(I2.Grade)
Bar	70	Eitan	80

לאחר ה select נקבל את הזוג Bar וEitan למרות שהשניים לא לקחו את אותם הקורסים באותם סמסטרים (לדוגמה הקורס CS) וכמו כן Eitan לא קיבל ציונים יותר טובים בכל הקורסים שהשניים לקחו במשותף (Eitan קיבל 88 בDB בעוד בר קיבל 95)

```

SELECT DISTINCT l1.StudentName AS n1, l2.StudentName AS n2
FROM Learns l1 INNER JOIN Learns l2
ON (l1.CourseName = l2.CourseName AND l1.Semester = l2.Semester)
WHERE l1.StudentName <> l2.StudentName
GROUP BY l1.StudentName, l2.StudentName
HAVING NOT EXISTS
(SELECT * FROM Learns t1 INNER JOIN Learns t2
ON (t1.CourseName = t2.CourseName AND t1.Semester = t2.semester)
WHERE (t1.StudentName = l1.StudentName AND t2.StudentName =
l2.StudentName AND t1.Grade >= t2.Grade))
AND NOT EXISTS
(SELECT * FROM (Learns t3 WHERE t3.StudentName = l1.StudentName) tab1
LEFT OUTER JOIN (SELECT * FROM Learns t4 WHERE t4.StudentName =
l2.StudentName) tab2
ON (tab1.CourseName = tab2.CourseName AND tab1.Semester =
tab2.Semester)
Where tab2.StudetName = NULL)

```