תרגיל 3 : SQL מתקדם ואינדקסים

תאריך הגשה : 6.12.20, 23:55.

הוראות הגשה:
בתרגיל זה אתם נדרשים להגיש קובץ zip בודד שיכלול את הקבצים הבאים :

- ex3.pdf עם התשובות לשאלות בחלק ב : אינדקסים.
- q1.sql
- q2.sql
- q3.sql
- q4.sql
- q5.sql
- README שמכיל שורה בודדת ובו ה-login של הסטודנט שמגיש את התרגיל. אם התרגיל מוגש בזוגות, על שורה זאת להכיל את שני ה-login מופרדים בפסיק.

שימו לב:

- נא לקרוא על הדרישות המנהליות של הקורס בלינק באתר הקורס כדי למלא אחר ההוראות להגשה של קבצים סרוקים!
- תרגיל מוקלד יזכה ב- 2 נקודות בונוס!

נתונים היחסים הבאים מתוך מסד נתונים של IMDb (זהים ליחסים בתרגיל 2) :

Movies (movieId, title, rating, year, duration, genre)

Actors (actorId, name, byear, dyear)

PlaysIn (movieId, actorId, character)

הערות :

- לכל סרט (Movie) יש מספר מזהה (movieId), כותרת (title), דירוג הסרט (rating), שנת יציאה (year), משך הסרט בדקות (duration) וז'אנר (genre).
- לכל שחקן (Actor) יש מספר מזהה (actorId), שם (name), שנת לידה (byear), ושנת פטירה (dyear). עבור שחקן חי, dyear הוא null .
- לכל משחק בסרט (PlaysIn) יש מספר מזהה של סרט ושחקן (movieId, actorId), ואת שם הדמות שהשחקן שיחק (character).

באתר הקורס יש קובץ create.sql המכיל הגדרות עבור הטבלאות וקובץ drop.sql המכיל פקודות המוחקות את הטבלאות.
כמו כן, נתונים הקבצים :

- actors.csv
- movies.csv
- playsIn.csv

הקבצים מכילים מידע חלקי אך אמיתי אודות סרטים ושחקנים מהאתר IMDb.
את המידע המלא ניתן למצוא ב https://www.imdb.com/interfaces/
הקבצים שלנו מכילים מידע על כ10,000 סרטים, שעבר "ניקוי" והתאמה לצורך התרגיל.

ניתן למצוא את הקבצים גם במערכת המחשבים במעבדה בתיקיה :

~ db/data/

ניתן להעתיק אותם לתיקיה שלכם.

על מנת לבדוק את התרגיל שלכם, יש ליצור את הטבלאות בעזרת create.sql, ולטעון לתוכן נתונים בעזרת הפקודות

```
cat movies.csv | psql -h dbcourse public -c "copy movies from STDIN DELIMITER ',' CSV HEADER"
cat actors.csv | psql -h dbcourse public -c "copy actors from STDIN DELIMITER ',' CSV HEADER"
cat playsIn.csv | psql -h dbcourse public -c "copy playsIn from STDIN DELIMITER ',' CSV HEADER"
```

חלק א: שאילתות SQL (40 נקודות): (להגשה בקבצי sql המתאימים לכל שאלה)

כתבו את השאילתות הבאות בSQL. שם הקובץ שבו צריכה להופיע התשובה לכל שאלה נמצא בתחילת השאלה.

בכל התשובות לשאלות בחלק זה:

- השתמשו ב SELECT DISTINCT כדי למנוע כפילויות בתשובות (אם כפילויות עלולות להוויצר בתשובה).
- **שימו לב:** בכל סעיף כתוב באיזה סדר למיין את התוצאות וכן את שמות העמודות בתוצאה.

1. **(q1.sql)** לכל שחקן, החזירו את actorId, את משך הסרט הארוך ביותר ששיחק בו, משך הסרט הקצר ביותר ששיחק בו, וממוצע אורך הסרטים ששיחק בהם.
   יש להחזיר טבלה עם העמודות actorId,max,min,avg ממויין לפי actorId.

2. **(q2.sql)** החזר את הכותרת ומספר מזהה של כל הסרטים שממוצע הגילאים של השחקנים בשנת יציאת הסרט היה לפחות 70.
   יש להחזיר טבלה עם העמודות movieid,title ממויין לפי movieid

3. **(q3.sql)** החזר את שם ומספר מזהה של השחקן שממוצע הדירוג (rating) של הסרטים ששיחק בהם הוא הגבוה ביותר.
   אם יש מספר שחקנים כאלו, החזר את כולם.
   יש להחזיר טבלה עם העמודות actorId, name, ממויין לפי actorId.

4. **(q4.sql)** החזר את מספר השחקים ששיחקו אך ורק בסרטים עם לפחות 6 שחקנים (כולל עצמם).
   יש להחזיר טבלה עם עמודה בודדת הנקראת num.

5. **(q5.sql)** כיתבו שאילתה **רקורסיבית** אשר מחזירה את שמות כל השחקנים שמספר Frank Bacon שלהם הוא לכל היותר 5. הסבר על מספר Bacon אפשר למצוא כאן.
   שימו לב שבנתונים שלנו לא מופיע השחקן Kevin Bacon, ולכן אנחנו נשתמש בFrank Bacon במקומו.
   יש להחזיר טבלה עם 2 עמודות actorid,name ממויינת לפי actorid.

חלק ב: אינדקסים (60 נקודות): (להגשה בכתב בקובץ ex3.pdf)

בחלק זה של התרגיל אנחנו עדיין נשתמש בסכמה המובאת פה שוב לייתר נוחות:

Movies (movieId, title, rating, year, duration, genre)

Actors (actorId, name, byear, dyear)

PlaysIn (movieId, actorId, character)

שאלה 1:

א. כתבו שאילתה בSQL המחזירה את המספר המזהה של כל השחקנים ששיחקו בסרט כלשוה דמות ששמה 'Sheriff'.

ב. הריצו את השאילתה עם פקודת explain analyse, שמראה את הquery plan של השאילתה, צרפי אותה לתשובות.
כיתבו כמה זמן לקח להריץ את השאילתה, והסבירו את אופן חישוב השאילתה.

ג. כיתבו פקודה אשר תייצר אינדקס **על שדה בודד** שישפר את זמן הריצה של השאילתה.

ד. הריצו את פקודת הבנייה של האינדקס ואת השאילתה עם פקודת explain analyse, שמראה את הquery plan של השאילתה, צרפו אותה לתשובות.
כיתבו כמה זמן לקח להריץ את השאילתה, והסבירו את אופן חישוב השאילתה.
אם אתם לא מבינים לגמרי את ה query plan חפשו באינטרנט דוקומנטציה שתעזור לכם להסביר.

שאלה 2:

בסעיפים הבאים, יש לכתוב הסבר לדרך הפתרון, ולהדגיש את התוצאה הסופית של כל חישוב!

הנחות:

- גודל בלוק הוא 1,000 בייטים.
- בטבלה Movies יש 10,000 שורות,
- כל שורה תופסת 150 בייטים.
- התכונה movieId תופסת 8 בייט.
- התכונה duration תופסת 8 בייט.
- התכונה genre תופסת 10 בייט.
- מצביע תופס 8 בייט.
- הערכים בduration בטבלה Movies מתפלגים אחיד בטווח [1,200]
- הערכים בgenre בטבלה מחולקים ל4 קטגוריות באופן אחיד.

א. נתונה השאילתה הבאה:

```
SELECT DISTINCT "exists"
FROM Movies
WHERE duration > 100
```

1. מה עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה?

כעת, נתון האינדקס הבא על הטבלה:

```
CREATE index on movies(duration)
```

2. מה תהיה דרגת הפיצול האופטימלית של האינדקס?

3. מה תהיה עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם?

ב. נתונה השאילתה הבאה:

```
SELECT avg (duration)
FROM Movies
WHERE duration > 100
```

1. מה עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה?

כעת, נתון האינדקס הבא על הטבלה:

```
CREATE index on movies(duration)
```

2. מה תהיה דרגת הפיצול האופטימלית של האינדקס?

3. מה תהיה עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם?

ג. נתונה השאילתה הבאה:

```
SELECT name
FROM Movies
WHERE movieid=200
```

1. מה עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה?

כעת, נתון האינדקס הבא על הטבלה:

```
CREATE index on movies(movieid)
```

2. מה תהיה דרגת הפיצול האופטימלית של האינדקס?

3. מה תהיה עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם?

ד. נתונה השאילתה הבאה:

```
SELECT avg(duration)
FROM Movies
WHERE genre = ‘Drama’
```

1. מה עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה?

כעת, נתון האינדקס הבא על הטבלה:

```
create index on movies(genre)
```

2. מה תהיה דרגת הפיצול האופטימלית של האינדקס?

3. מה תהיה עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם?

ה. נתונה השאילתה הבאה:

```
SELECT avg(duration)
FROM Movies
WHERE genre = 'Drama'
```

1. מה עלות חישוב השאילתה בהנחה שאין אינדקסים על הטבלה?

כעת, נתון האינדקס הבא על הטבלה:

```
create index on movies(genre,duration)
```

2. מה תהיה דרגת הפיצול האופטימלית של האינדקס?

3. מה תהיה עלות חישוב השאילתה באמצעות האינדקס, בהנחה שדרגת הפיצול היא זו שחושבה בסעיף הקודם?

בהצלחה!