

## **קובץ תיעוד תוכנה MUSIC MASTER**

**תוכן עניינים:**

**עמוד 2-5:** מבנה הטבלאות (שדות, מפתחות ראשיים, זרים ואופן יצירתם)

**עמוד 6:** הקשרים בין הטבלאות

**עמוד 7-8:** מבנה הפרויקט ופירוט מחלקות

**עמוד 9-12:** פירוט השאילתות

## מבנה הטבלאות:

Albums: (מבוסס על קובץ albums.csv)

(מתבסס על פרמטרים שמתקבלים מסקריפט מס' 1)

שדות הטבלה:

- קוד אלבום (AlbumID) – הקוד של האלבום שמורכב ממספרים.
- שם אלבום (AlbumName) - השם של האלבום.

Artists: (מבוסס על קובץ artists.csv)

(מתבסס על פרמטרים שמתקבלים מסקריפט מס' 3)

שדות הטבלה:

- קוד אומן (artistID) – מפתח ראשי של הטבלה אשר הערך ממוספר בצורה נוחה מ 1 עד כמות האומנים שיש (בניגוד לסקריפט מס' 1 שם הקוד היה מורכב מאותיות), בוצע ע"י סקריפט מס' 3.
- שם אומן (artistName) – מציג את שם האומן.

SimilarArtists: (מבוסס על קובץ similarArtists.csv)

(מתבסס על פרמטרים שמתקבלים מסקריפט מס' 3)

שדות הטבלה:

- קוד אומן (ArtistID) – הקוד של אומן א' – מפתח זר לArtistID בטבלת Artists.
  - קוד אומן ב' (SimilarToArtistID) – הקוד של אומן ב' – מפתח זר לArtistID בטבלת Artists.
- שני השדות מתפקדים כמפתח ראשי בטבלה הזאת על מנת למנוע כפילויות מידע של אומנים דומים.

SongInfo: (מבוסס על קובץ songInfo.csv)

(מתבסס על פרמטרים שמתקבלים מסקריפט מס' 3 ומסקריפט מס' 2)

שדות הטבלה:

- קוד שיר (SongID) - מפתח ראשי של הטבלה אשר מכיל את הקוד של השיר.
- שם השיר (Title) – שם השיר.
- קוד אלבום (AlbumID) – קוד אלבום של השיר, מפתח זר לAlbumID בטבלת Albums.
- קוד אומן (ArtistID) – קוד אומן של השיר, מפתח זר לArtistID בטבלת Artists.
- שנה (Year) – השנה שבה יצא השיר.
- רגע השיא של השיר (EndOfFadeIn) – מס' מסוג float אשר הספרה הראשונה אחרי הנקודה מציינת את הדקה והספרות הבאות את השניות שבהם מתחיל השיר להתנגן בפועל (אחרי מנגינות הרקע).

Genre:

אופן יצירת הטבלה:

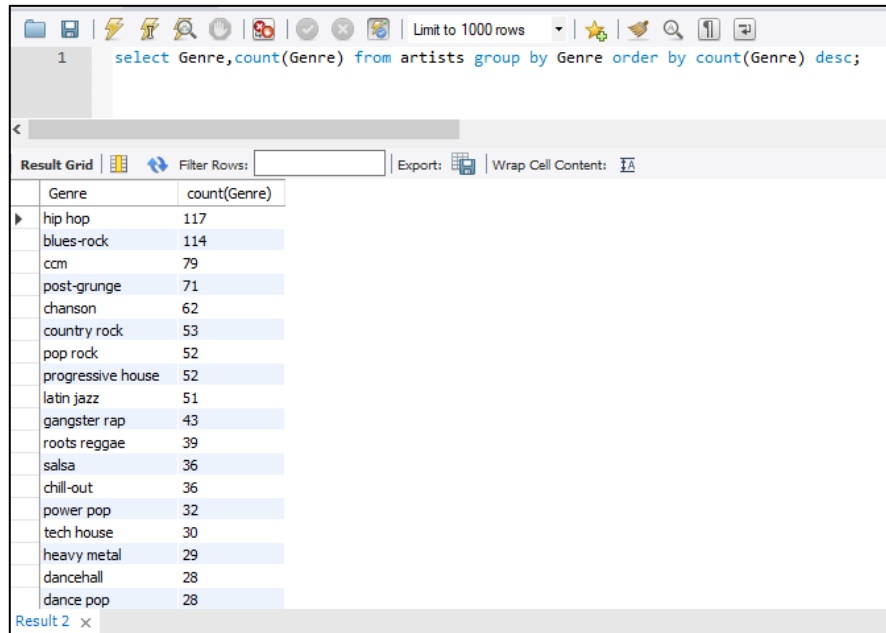
השתמשו במקביל בשני סקריפטים שקיבלנו , סקריפט מס' 1 יצר את הטבלה הבאה:

כמו שניתן לראות יש כאן הרבה מאוד ז'אנרים ע"י הסקריפט הנ"ל, חלק מן הז'אנרים מאוד דומים לז'אנרים אחרים (לדוגמא רוק אמריקאי / בריטי / ישן כולם סוג של רוק).

סקריפט מספר 3 יצר את הטבלה הבאה:

ניתן לראות כי סקריפט זה התבסס על כך שלכל אומן יש ז'אנר אחד (הז'אנר הפופולרי ביותר), לדעתנו הנחה זו אינה נכונה כיוון שלכל אומן יש מבחר ז'אנרים בהם הוא שר.

לכן, הפעלנו שאילתה(בהתבסס על הטבלה האחרונה) אשר מחזירה את כל הז'אנרים והכמות שלהם בטבלה הנוכחית.



The screenshot shows a database query interface. At the top, there is a toolbar with various icons and a text input field containing the following SQL query:

```
1 select Genre, count(Genre) from artists group by Genre order by count(Genre) desc;
```

Below the query, there is a section labeled "Result Grid". It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The results are displayed in a table with two columns: "Genre" and "count(Genre)".

Genre	count(Genre)
hip hop	117
blues-rock	114
ccm	79
post-grunge	71
chanson	62
country rock	53
pop rock	52
progressive house	52
latin jazz	51
gangster rap	43
roots reggae	39
salsa	36
chill-out	36
power pop	32
tech house	30
heavy metal	29
dancehall	28
dance pop	28

על רשימה זו עברנו באופן ידני ובחרנו 30 ז'אנרים שונים שעליהם ביססנו את יצירת הטבלה. שדות הטבלה:

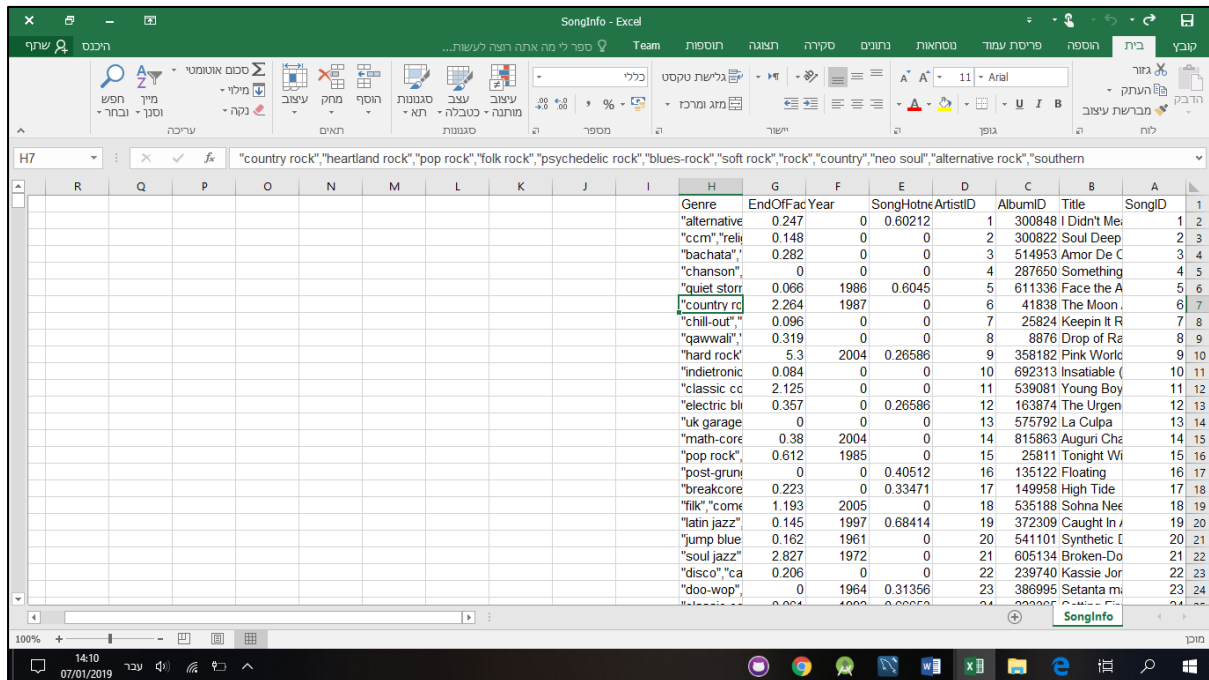
- קוד ז'אנר (GenreID) – הקוד של הז'אנר , ממוספר מ 1 עד 30 לפי כמות הז'אנרים שהוצאו מהשאלתה.
- שם ז'אנר (GenreName) – השם של הז'אנר.

## GenreArtists: (מבוסס על הקובץ genreartists.csv)

יצירת הטבלה התבססה על כך שיש קשר רב רב ערכי בין אומנים לז'אנרים, כלומר כל אומן יכול לשיר במגוון ז'אנרים, ובכל ז'אנר יכולים לשיר כמה אומנים. לכן, יצרנו טבלה מקשרת בין האומנים לז'אנרים.

אופן יצירת הטבלה:

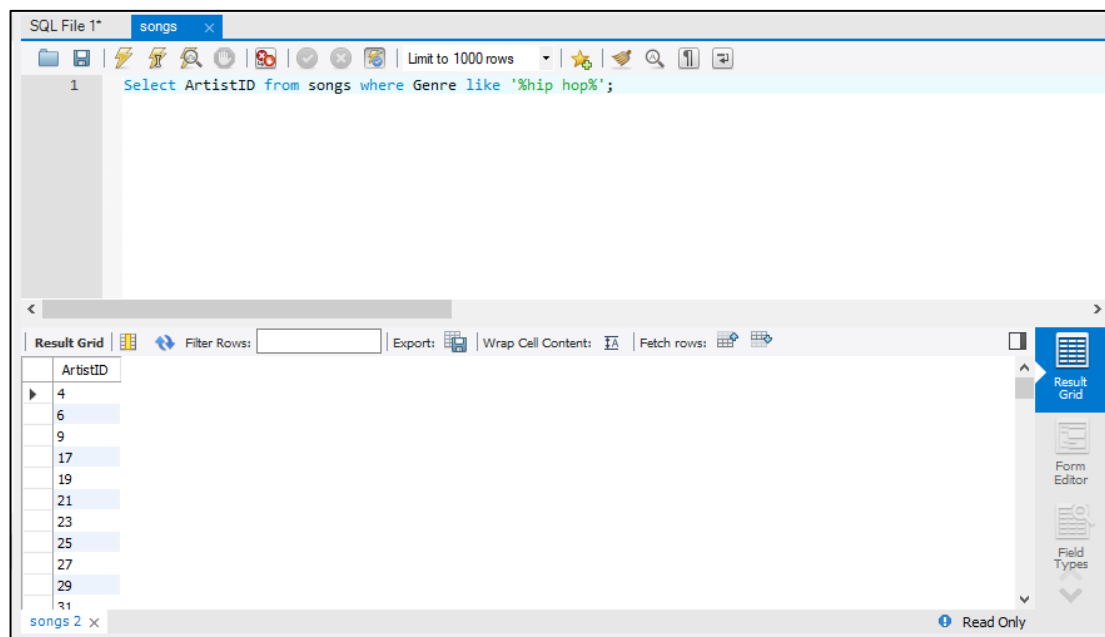
בהתבסס על 30 הז'אנרים מהטבלה Genre, השתמשנו גם בטבלה הבאה:



Genre	EndOfFacYear	SongHotneArtistID	AlbumID	Title	SongID
"alternative"	0.247	0	0.60212	1	300848 I Didn't Me
"ccm", "trel"	0.148	0	0	2	300822 Soul Deep
"bachata", "	0.282	0	0	3	514953 Amor De C
"chanson", "	0	0	0	4	287650 Something
"quiet stor"	0.066	1986	0.6045	5	611336 Face the A
"country rd"	2.264	1987	0	6	41838 The Moon
"chill-out", "	0.096	0	0	7	25824 Keepin It R
"qawwali", "	0.319	0	0	8	8876 Drop of Ra
"hard rock"	5.3	2004	0.26586	9	358182 Pink Worl
"indietronic	0.084	0	0	10	692313 Insatiable
"classic cc"	2.125	0	0	11	539081 Young Boy
"electric bli	0.357	0	0.26586	12	163874 The Urgan
"uk garage"	0	0	0	13	575792 La Culpa
"math-core"	0.38	2004	0	14	815863 Auguri Cha
"pop rock", "	0.612	1985	0	15	25811 Tonight Wi
"post-grun"	0	0	0.40512	16	135122 Floating
"breakcore"	0.223	0	0.33471	17	149958 High Tide
"folk", "come	1.193	2005	0	18	535188 Sohna Nee
"latin jazz"	0.145	1997	0.68414	19	372309 Caught In
"jump blue"	0.162	1961	0	20	541101 Synthetic
"soul jazz"	2.827	1972	0	21	605134 Broken-Do
"disco", "ca	0.206	0	0	22	239740 Kassie Jor
"doo-wop", "	0	1964	0.31356	23	386995 Setanta m

ניתן לראות בה מיפוי מסוים בין אומן לז'אנר.

על הטבלה הרצנו שאלות ב-sql אשר ימפו בין שם הז'אנר בטבלת הז'אנרים לבין האומנים אשר שרים בז'אנר הזה:



```
Select ArtistID from songs where Genre like '%hip hop%';
```

ArtistID
4
6
9
17
19
21
23
25
27
29
31

כך נוצרה טבלה המקשרת בין כל האומנים ששרים באותו ז'אנר לבין הז'אנר (הוספנו עמודה של קוד הז'אנר הרלוונטי) חזרנו על התהליך עבור כל 30 הז'אנרים, ויצרנו את הטבלה המקשרת ביניהם.

שדות הטבלה:

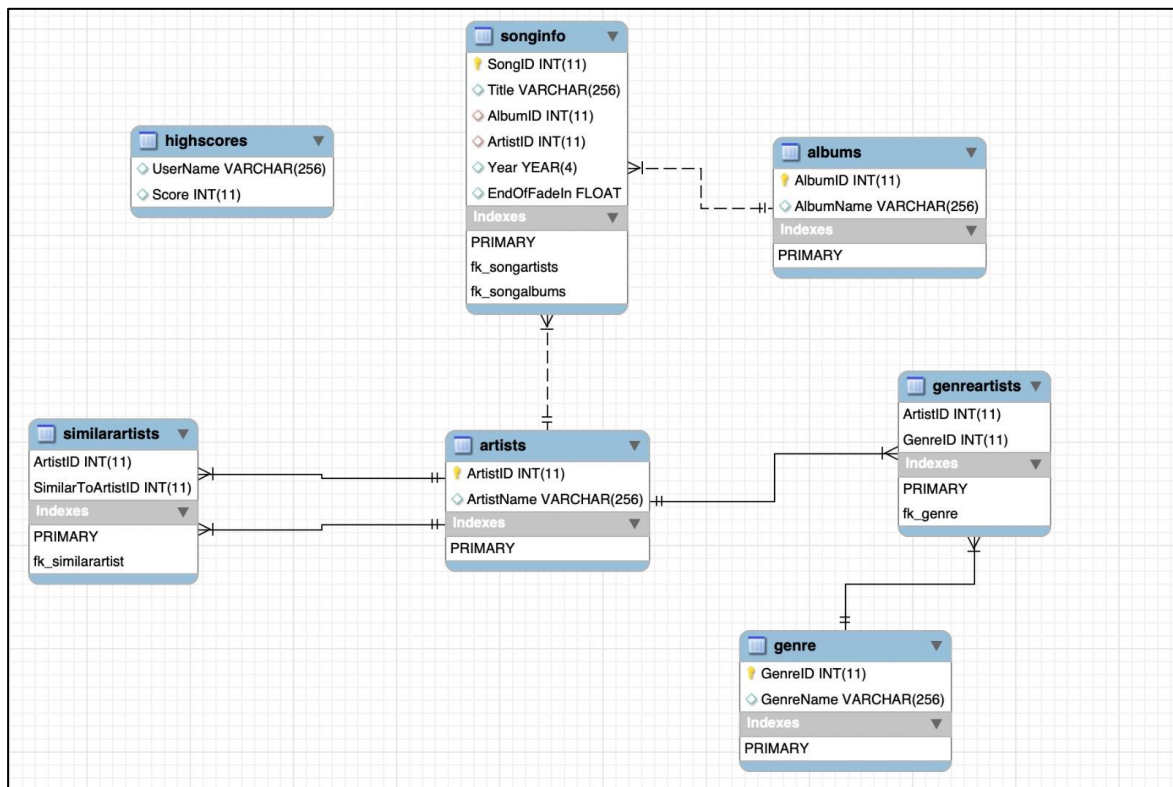
- קוד ז'אנר (GenreID) – קוד הז'אנר , מפתח זר לGenreID בטבלת Genre.
- קוד אומן (ArtistID) - קוד אומן , מפתח זר לArtistID בטבלת Artists.

## הקשרים בין הטבלאות:

שירים לאומנים – שיר יכול לשיר אומן אחד , אומן יכול לשיר מס' שירים ← קשר חד רב ערכי ולכן הוספנו שדה של קוד זמר בטבלת השירים.

שירים לאלבומים – אלבום יכול להכיל מספר שירים , שיר יכול להופיע באלבום אחד (מתבסס על הנתונים מהאתר של millionsongdataset) ← קשר חד רב ערכי ולכן הוספנו שדה של קוד אלבום בטבלת השירים.

אומנים לז'אנרים – ז'אנר יכול להיות של מס' אומנים , אומן יכול לשיר בכמה ז'אנרים ← לכן קשר רב רב ערכי ולכן יצרנו טבלה חדשה המקשרת בין הז'אנרים לאומנים.



## מבנה הפרויקט:

הפרויקט מחולק ל3 שכבות שונות(3 פאקג'ים):

DataBase – כל המחלקות שמדברות עם DB. כל השאילות נמצאות כאן.

Logic – כל המחלקות שקשורות ללוגיקה של המשחק והData classes. זוהי בעצם השכבה המקשרת בין UI לDB.

UIInterface – כל המחלקות שקשורות לתצוגה של המשחק, קבצי css,xaml ותמונות הרקע.

## פירוט המחלקות:

### :DataBase

DBConnection – מחלקה זו אחראית לחיבור לDB. המחלקה היא סינגלטון כיוון שאנו רוצים חיבור יחיד. החיבור נוצר פעם אחת בתחילת המשחק ע"י קריאת פרטי ההתחברות מקובץ קונפיגורציה.

DBArtists – מחלקה זו אחראית לכל השאילות אשר קשורות לאומנים (Artist):

מציאת כל הזמרים השרים בז'אנר מסוים, מציאת תשובות מבלבלות לשאלה "מי הזמר ששר" - ע"י מציאת זמרים דומים לפי טבלה similar artists ומציאת זמרים דומים לפי ז'אנר.

DBGenre – מחלקה זו אחראית לשאילות אשר קשורות לז'אנרים (Genre):

לקיחת כל הז'אנרים הקיימים בDB.

DBSongs – מחלקה זו אחראית לשאילות אשר קשורות לשירים (Song):

מציאת שירים של זמר מסוים, מציאת תשובות מבלבלות לשאלה "מה השיר המושמע" - ע"י מציאת שירים של זמרים השרים באותו הז'אנר של הזמר של הנבחר בחיתוך עם הז'אנרים שהשחקן בחר או ע"י הוספה של תנאי שהשיר יצא באותה השנה כמו השיר הנבחר.

DBAlbums – מחלקה זו אחראית לשאילות אשר קשורות לאלבומים (Album):

קבלת כל האלבומים בהם יש יותר מ3 שירים.

DBHighScore - מחלקה זו אחראית לשאילות אשר קשורות לטבלה השיאים (HighScores):

לקיחת 10 השיאים הגבוהים ביותר בטבלה, הכנסת שיא חדש לטבלה ועדכונה.

Search – מחלקה זו אחראית לחיפוש השיר ביוטיוב, מבוססת על הYouTube API של גוגל. מקבלת את שם השיר + שם הזמר ומחזירה את הID של השיר ביוטיוב.

### :Logic

מחלקות data class:

Artist – כל אובייקט מסוג זה מייצג אומן מהDB ושומר את מאפייניו.

Song - כל אובייקט מסוג זה מייצג שיר מהDB ושומר את מאפייניו.

Genre - כל אובייקט מסוג זה מייצג ז'אנר מהDB ושומר את מאפייניו.

Album - כל אובייקט מסוג זה מייצג אלבום מהDB ושומר את מאפייניו.

HighScores - כל אובייקט מסוג זה מייצג רשומה מטבלת השיאים בDB ושומר את מאפייניו.

Question - כל אובייקט מסוג זה מייצג שאלה במשחק ושומר את מאפייניה.

## מחלקות לוגיקה:

GameLogic – אחראית ללוגיקה של מהלך המשחק.

יצירת השאלות למשחק, חיפוש השיר הנבחר ביוטיוב, בדיקה אם התשובה של השחקן נכונה או לא, בדיקה אם המשחק נגמר, מתן ניקוד למשתמש.

GenreLogic – אחראית ללוגיקה הקשורה לג'אנרים.

לוקחת מהDB את הז'אנרים כדי להציג למשתמש, לוקחת את כל האומנים השרים בז'אנרים מהרשימה שהמשתמש בחר.

HighScoreLogic – אחראית ללוגיקה של מסך טבלת השיאים.

קבלת 10 התוצאות הגבוהות ביותר, שמירת שיא חדש בטבלה.

## :UIInterface

GameController – מחלקת הקונטרולר של מסך המשחק. אחראית לתצוגה של המשחק, עדכון התשובות לשאלה, השמעת השיר בכל שלב, עדכון ניקוד, חיים, זמן וכו'. מעבר למסך סיום המשחק בסיומו.

GenreController – מחלקת הקונטרולר של מסך בחירת הז'אנרים. מבקשת מהלוגיקה את שמות הז'אנרים ויוצרת את הcheckboxים, בודקת בלחיצה על Ok האם נבחרה כמות נכונה של ז'אנרים (עד 3) ומעבר למסך המשחק.

GameOverController – מחלקת הקונטרולר של מסך סיום המשחק. למשתמש מוצג הניקוד הסופי שלו ותיבה להכנסת שמו על מנת לשמור את הניקוד שלו לטבלה.

HighScoreController – מחלקת הקונטרולר של מסך טבלת השיאים. אחראי לתצוגה של הטבלה, קבלת המידע מהלוגיקה של עשרת השיאים הגבוהים ביותר.

MenuController – מחלקת הקונטרולר של המסך הראשי. אחראי לתצוגה של המסך הראשי, מעבר למסך הרצוי לפי הלחיצה של המשתמש, בדיקה האם יש חיבור לאינטרנט.

Main – המחלקה שמפעילה את התוכנית, מגדירה את החלון ועוברת למסך של התפריט. מגדירה פעולה ביציאה מהחלון.

כמו כן קיימים כאן קבצים של fxml עבור כל חלון וקבצי css המגדירים את עיצוב המסכים.



## פירוט השאלות:

### מחלקת DBGenre:

#### פונקציה GenreList:

"select GenreID, GenreName from genre"

שאלתה זו מחלצת את כל הז'אנרים הקיימים בDB בטבלה genre. אנו משתמשים בשאלתה זו בתחילת המשחק כדי להציג למשתמש את הז'אנרים הקיימים על מנת שיבחר עם איזה ז'אנרים ברצונו לשחק.

### מחלקת DBArtist:

#### פונקציה FilterArtistByGenre:

בפונקציה זו אנו מקבלים את רשימת הז'אנרים שהמשתמש בחר ומחזירים רשימה של אומנים השרים בז'אנרים האלו. אנו משתמשים בפונקציה לשם בניית השאלות, למעשה בכל פעם נגריל אומן מרשימה זו ולאחר מכן נגריל שיר מבין שיריו.

כיוון שהשאלתה מקבלת פרמטרים נראה דוגמה להרצה אפשרית:

נניח שהמשתמש בחר בז'אנרים rock, pop.

"select distinct artists.ArtistID, ArtistName from artists,genre,genreartists where artists.ArtistID = genreArtists.ArtistID and genreartists.GenreID = genre.GenreID and (GenreName = **"rock"** or GenreName = **"pop"** or GenreName=**"rap"**) limit 50;"

כל מה שמודגש בשאלתה נלקח מהפרמטרים לפונקציה.

#### פונקציה CreateConfusionAns:

בפונקציה זו אנו מקבלים את האומן שהוא התשובה הנכונה בשאלה הנוכחית ומחזירים רשימה של אומנים הדומים לאומן זה בהם נשתמש כמסיחים לשאלה.

אנו מוצאים את האומנים הדומים לפי טבלת similarArtists, יש לנו צורך לבדוק בשתי העמודות ולכן נבצע שתי שאלות.

בשתי השאלות נשתמש לדוגמה באומן עם ID 30.

- "select distinct artists.ArtistID, ArtistName from similarartists,artists where artists.ArtistID = similarArtists.similarToArtistID and similarArtists.ArtistID = **30** limit 10"
- "select distinct artists.ArtistID, ArtistName from similarartists,artists where artists.ArtistID = similarArtists.ArtistID and similarToArtistID = **30** limit 10"

כל מה שמודגש בשאלות נלקח מהפרמטרים לפונקציה.

#### פונקציה FilterArtistDifferent:

בפונקציה זו אנו מקבלים את האומן שהוא התשובה הנכונה בשאלה הנוכחית, רשימת אומנים דומים (שנלקחו מטבלת similarArtists בפונקציה CreateConfusionAns) ואת רשימת הז'אנרים שהשחקן בחר.

אנו משתמשים בשאלתה זו במקרים ובהם יש פחות מ-3 אומנים דומים לאומן שנבחר בטבלה של similarArtists ולכן בעצם יש לנו צורך בלמצוא עוד מסיחים לשאלה.

כיוון שהשאלתה מקבלת פרמטרים נראה דוגמה להרצה אפשרית:

כאשר הID של האומן הנבחר הוא 67, האומנים הדומים לו הם 1393 ו773 ואילו הז'אנרים שהמשתמש בחר הם 6,10,11 ( מתוכם 6 הוא ז'אנר השייך לאומן הנוכחי)

```
"select distinct artistid,ArtistName from artists,(select distinct artistid as newar from genreartists as a, (select distinct genreID as newgen from genreartists where ArtistID=67 and genreID in(6,10,11)) as b where b.newgen=a.genreid) as c where artistid != 67 and artistid not in(1393,773) limit 50"
```

כל מה שמודגש בשאילתה נלקח מהפרמטרים לפונקציה.

השאילתה למעשה קודם כל בודקת מהם הז'אנרים של האומן הנתון, לאחר מכן לוקחת את החיתוך בינם לבין הז'אנרים שהמשתמש בחר, לאחר מכן עבור הז'אנרים שהתקבלו ניקח את כל הזמרים אשר שרים בז'אנרים האלו ונסנן את הזמר הנוכחי ואת הזמרים הדומים לו שכבר מצאנו על מנת למנוע כפילויות במסיחים. ( הפקודה `and artistid not in(1393,773)` רק במקרה יש לפחות זמר אחד דומה לו ברשימה הנתונה)

מחלקת DBSongs:

פונקציה FilterSong:

פונקציה זו מקבלת את האומן שנבחר לשאלה ומחזירה רשימה של שירים של האומן הזה.

נראה את השאילתה לדוגמה עבור אומן עם ID 10.

```
" select distinct SongID,year,Title,EndOfFadeln from songinfo where ArtistID = 10 limit 10;"
```

כל מה שמודגש בשאילתה נלקח מהפרמטרים לפונקציה.

מכל שורה שחזרה ניצור אובייקט מסוג song. אנו זקוקים לפרטים של year,title,endOfFadeln לשאילתות עתידיות/למהלך המשחק.

אנו משתמשים בשאילתה זו על מנת לבחור את השיר שיתנגן בשאלה(למעשה נגריל שיר מתוך רשימה זו).

פונקציה getListOfSimilarSongs:

פונקציה זו מקבלת את השיר שהוא נושא השאלה בשאלות מסוג "זהה את השיר" ואת רשימת הז'אנרים שהמשתמש בחר. הפונקציה תחזיר לנו רשימה של שירים דומים לשיר הנבחר.

אנו משתמשים בפונקציה זו על מנת ליצור מסיחים לשאלה.

כיוון שהשאילתה מקבלת פרמטרים נראה דוגמה להרצה אפשרית:

נניח השיר הנבחר הוא עם ID 40 נקבל שהID של האומן הוא 39 ונניח שהז'אנרים שנבחרו הם 9,17,21 (מתוכם 9 הוא ז'אנר השייך לאומן הנוכחי), השנה של שיר 40 היא 1995.

```
"select distinct songid, title,year,artistid from songinfo ,(select distinct artistid as newar from genreartists as a,(select distinct genreID as newgen from genreartists where ArtistID= 39 and genreID in(9,17,21)) as b where b.newgen=a.genreid) as n where n.newar=songinfo.artistid and year= 1995 and songid != 40 order by songid limit 50; "
```

כל מה שמודגש בשאילתה נלקח מהפרמטרים לפונקציה.

השאילתה למעשה קודם כל בודקת מהם הז'אנרים של האומן של השיר הנתון, לאחר מכן לוקחת את החיתוך בינם לבין הז'אנרים שהמשתמש בחר, לאחר מכן עבור הז'אנרים שהתקבלו ניקח את כל הזמרים אשר שרים בז'אנרים האלו ולבסוף נבחר את כל השירים של האומנים הללו.

על מנת לצמצם את השירים וליצור מסיחים מבלבלים יותר הוספנו תנאי שהשיר יהיה מאותה השנה של השיר הנבחר בשאלה וכמובן בסוף נסנן מהתוצאה את השיר הנוכחי על מנת למנוע כפילות.

פונקציה `getDifferentSongsByGenre`:

בפונקציה זו אנו מקבלים את השיר שהוא התשובה הנכונה בשאלה הנוכחית, רשימת שירים דומים (שנלקחו מהפונקציה `getListOfSimilarSongs`) ואת רשימת הז'אנרים שהשחקן בחר.

אנו משתמשים בשאלתה זו במקרים ובהם יש פחות מ-3 שירים דומים לשיר שנבחר לפי התנאים בפונקציה הקודמת ולכן בעצם יש לנו צורך בלמצוא עוד מסיחים לשאלה.

כיוון שהשאלתה מקבלת פרמטרים נראה דוגמה להרצה אפשרית:

כאשר ה-ID של השיר הנבחר הוא 98 וה-ID של האומן שלו הוא 102, השירים הדומים לו הם 2239 ו-3756 ואילו הז'אנרים שהמשתמש בחר הם 16, 27, 26 (מתוכם 26 הוא ז'אנר השייך לאומן של השיר הנוכחי).

```
"select distinct songid, title, year, artistid from songinfo, (select distinct artistid as newar from genreartists as a, (select distinct genreid as newgen from genreartists where ArtistID=102 and genreid in(26,27,16)) as b where b.newgen=a.genreid) as n where n.newar=songinfo.artistid and songid != 98 and songid not in(2239,3756) order by songid limit 50;"
```

כל מה שמודגש בשאלתה נלקח מהפרמטרים לפונקציה.

השאלתה למעשה קודם כל בודקת מהם הז'אנרים של האומן של השיר הנתון, לאחר מכן לוקחת את החיתוך בינם לבין הז'אנרים שהמשתמש בחר, לאחר מכן עבור הז'אנרים ניקח את כל הזמרים אשר שרים בז'אנרים האלו ולבסוף נבחר את כל השירים של האומנים הללו ונסנן את השיר הנוכחי ואת השירים הדומים לו שכבר מצאנו על מנת למנוע כפילויות במסיחים.

(הפקודה `and songid not in(2239,3756)` מופיעה בשאלתה רק במקרה ויש לפחות שיר אחד דומה לו ברשימה הנתונה)

פונקציה `getListSongsByAlbum`:

בפונקציה זו אנו מקבלים אלבום מסוים ומחזירים את כל השירים השייכים לאלבום זה (האלבום נבחר בצורה רנדומלית מתוך כל אוסף האלבומים שמכילים יותר מ-3 שירים אותם נקבל מהפונקציה `getListOfAlbumsWithMoreThan3Songs`). (DBAlbums)

אנו משתמשים בשאלתה זו בשאלות מהסוג "מה שם השיר" על מנת למצוא מסיחים שיהיו שייכים לאותו האלבום שאליו שייך השיר שמושמע ברקע (והוא התשובה הנכונה).

כיוון שהשאלתה מקבלת פרמטרים נראה דוגמה להרצה אפשרית:

כאשר ה-ID של האלבום שנבחר רנדומלית הוא 57714 נרצה למצוא את כל השירים באלבום זה. אלבום זה מכיל 4 שירים ולכן נקבל אותם מהשאלתה.

```
"SELECT Songid, Title, Year, ArtistID, EndOfFadeIn FROM songinfo where songinfo.AlbumID = 57714 "
```

כל מה שמודגש בשאלתה נלקח מהפרמטרים לפונקציה.

## מחלקת DBAlbums:

### פונקציה getListOfAlbumsWithMoreThen3Songs:

פונקציה זו מחזירה רשימת אלבומים בהם יש יותר מ-3 שירים. מרשימה זו נבחר לאחר מכן אלבום רנדומלי, מאלבום זה נבחר שיר רנדומלי שיהיה התשובה הנכונה בשאלה מסוג "מה שם השיר" ושאר השירים ישמשו כמסיחים.

```
"SELECT distinct songinfo.AlbumID,albums.AlbumName FROM songinfo,albums,(SELECT  
albumId,count(SongID) FROM songinfo group by albumId HAVING count(SongID) > 3 order by  
count(SongID) asc) as b where b.albumId = songinfo.AlbumID and songinfo.AlbumID =  
albums.AlbumID order by songinfo.AlbumID LIMIT 100 "
```

## מחלקת DBHighScores:

### פונקציה TheBest10Score:

הפונקציה מחזירה את עשרת התוצאות הטובות ביותר בטבלת השיאים השמורה ב-DB.

אנו משתמשים בשאילתה במסך טבלת השיאים על מנת להציג למשתמש את עשרת התוצאות הטובות ביותר.

```
" select * from highscores order by Score desc limit 10"
```

### פונקציה UpdateHighScoresTable:

פונקציה זו מקבלת תוצאה (HighScore) ומכניסה אותה לטבלת השיאים ב-DB.

אנו משתמשים בשאילתה זו כאשר שחקן נפסל, מועבר למסך הסיום ומכניס את שמו. אנו שומרים את התוצאה שלו ב-DB.

כיוון שהשאילתה מקבלת פרמטרים נראה דוגמה להרצה אפשרית:

עבור שם השחקן: Maria וניקוד 1000

```
" INSERT INTO highscores(Username, Score) VALUES("Maria", 1000)"
```

כל מה שמודגש בשאילתה נלקח מהפרמטרים לפונקציה.