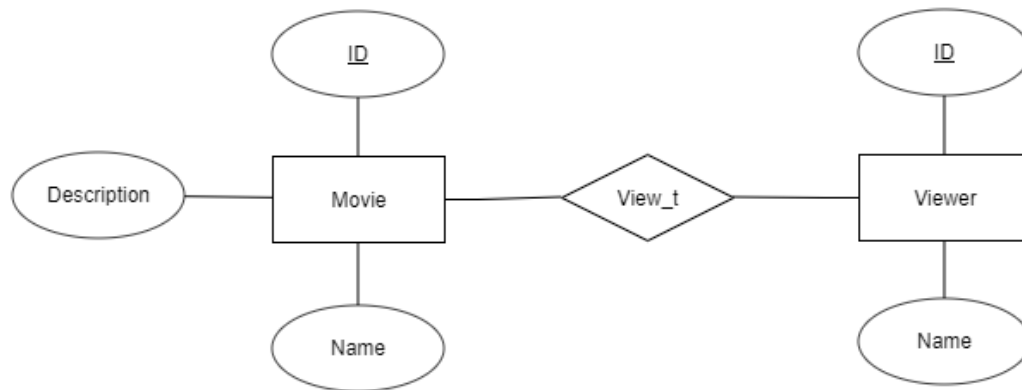


תיאור עיצוב המבנה

על מנת לממש את דרישות התרגיל החזקנו 3 טבלאות; 2 טבלאות Entities וטבלת Relationship:

- Entity - Viewer
- Entity - Movie
- Relationship – View_t



טבלת View_t-

Viewer_ID	Movie_ID	Rating
-----------	----------	--------

כאשר אדם צופה בסרט מסוים, נוספת צפייה בטבלת View_t עם Viewer_ID של הצופה והMovie_ID של הסרט בו צפה. השדה Rating מאוחל לNULL, וכאשר אותו צופה מעוניין לדרג את הסרט בעל Movie_ID השדה Rating מתעדכן בהתאם, גם אם אותחל כבר בעבר.

המערכת תומכת בפונקציות בסיסיות של Set, Get, Add, Update, Remove לטבלאות, ובנוסף ממומשות 4 הפונקציות מהAPI Advanced כדלקמן:

ArrayList getSimilarViewers(Integer viewerId)

תחילה נרצה לייצר טבלה שהעמודות בה הן viewerId ומספר הסרטים המשותפים שצפו בהם המשתמש בעל viewerId וכל אחד מן הצופים האחרים במערכת. אחר כך נמשך מאותה טבלה את ה-Viewer_ID שצפו בלפחות 75% מהסרטים שצפה בהם viewerId ובנוסף נדרוש שה-ID שלהם אינו viewerId וגם שהם צפו באיזשהו סרט (על מנת לענות על הדרישה כי אנשים שלא צפו באף סרט לא יחשבו "צופים דומים") לאחר מכן נמיין את הטבלה על פי ה-Viewer_ID בסדר עולה ונמשוך מן הטבלה הזמנית שיצרנו את העמודה Viewer_ID

ArrayList mostInfluencingViewers()

ניצור טבלה שהעמודות בה הן Viewer_ID ומספר הצפיות של כל אחד מהם בסרטים השונים. אחר כך ניצור טבלה חדשה שהיא איחוד של הטבלה שיצרנו מקודם וטבלה חדשה, שהעמודות בה הן Viewer_ID ומספר הדירוגים של כל אחד מהם על הסרטים השונים (נבחר את השורות שבהן הערך בעמודת Rating הוא 'LIKE' או 'DISLIKE' ולא NULL). האיחוד יתבצע לפי העמודה השמאלית ביותר כלומר לפי ה-Viewer_ID הזהים. העמודות בטבלה החדשה שיצרנו הן: Viewer_ID, cnt_views, cnt_ratings. נמיין את הטבלה מיון ראשוני לפי cnt_views בסדר יורד, מיון משני לפי cnt_ratings בסדר יורד, ומיון שלישוני לפי Viewer_ID בסדר עולה.

ArrayList getMoviesRecommendations(Integer viewerId)

נרצה לייצר את טבלת ה"צופים הדומים" (טבלה 1) בעזרת שאילתא, ובנוסף ניצור טבלה 2 של כל הסרטים שהצופה בעל viewerId לא צפה בהם. כעת, ניצור טבלה חדשה (3) בעלת העמודות movie_id ומספר הצופים הדומים שצפו בסרט ודירגו אותו בדירוג 'LIKE'. נעשה זאת על ידי מציאה של הסרטים בטבלה 2 שצפו בהם צופים אשר מופיעים בטבלה 1 ובנוסף הם דורגו בדירוג 'LIKE'. נמיין את טבלה 3 במיון ראשוני לפי כמות הלייקים שיש להם בסדר יורד, ובמיון משני לפי movie_id בסדר עולה.

לבסוף נגביל את גודלה של טבלה 3 ל-10 שורות, וניצור טבלה רביעית ואחרונה שמכילה רק את עמודת movie_id שכאמור ממיינת בסדר הנכון. בנוסף, נציין כי אם אין בכלל/אין מספיק סרטים עם דירוג, אז נוסיף סרטים שנצפו על ידי הצופים הדומים אבל ללא דירוג/דירוג שלילי.

ArrayList getConditionalRecommendations(Integer viewerId, Integer MovieId)

ניצור טבלת "מדרגים דומים" לסרט הספציפי בעל הזהה MovieID, נעשה זאת בצורה זהה ליצירת טבלת "צופים דומים" אבל נקשה את התנאים כך שיכנסו רק צופים דומים שאהבו את הסרט בעל המזהה MovieId. בנוסף ניצור טבלה 2 של כל הסרטים שהצופה בעל viewerId לא צפה בהם. כעת, ניצור טבלה חדשה (3) בעלת העמודות movie_id ומספר המדרגים הדומים שצפו בסרט בעל המזהה movie_id. נעשה זאת על ידי מציאה של הסרטים בטבלה 2 שצפו בהם צופים אשר מופיעים בטבלה 1.

נמייין את טבלה 3 במיון ראשוני לפי כמות הלייקים שיש להם בסדר יורד, ובמיון משני לפי movie_id בסדר עולה.

לבסוף נגביל את גודלה של טבלה 3 ל10 שורות, וניצור טבלה רביעית ואחרונה שמכילה רק את עמודת movie_id שכאמור ממויינת בסדר הנכון. , נציין כי אם אין בכלל/אין מספיק סרטים עם דירוג, אז נוסיף סרטים שנצפו על ידי הצופים הדומים אבל עם דירוג שלילי.