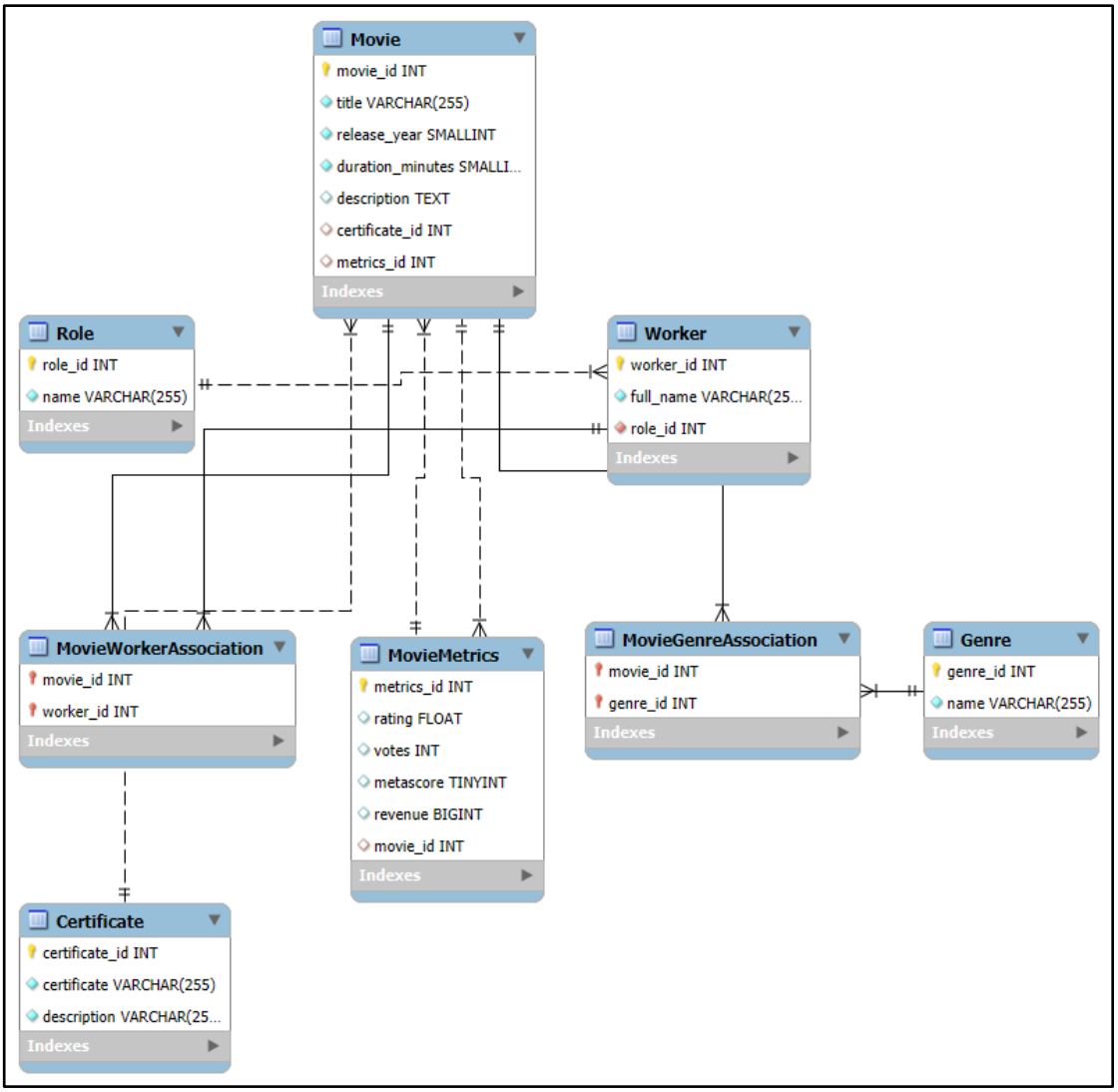
System Documentation

תיאור ה- Data Scheme Structure

- תיאור גרפי

-



פירוט

מדובר על Database של סרטים, ולכן ה- Movie Table מהווה את הטבלה המרכזית

(כפי שמעידים מספר הקשרים לתוכה ומיקומה באיור).

לסרט יש:

- רשומה ב- Movie Table שרלוונטית עבורו
- מזהים של העובדים שהשתתפו בו ניתנים להשגה דרך MovieWorkerAssociation ומשם דרך Worker Table לפרטיהם המדויקים ואף דרך Role Table לתפקידם

- רשומה ב- Certificate שרלוונטית עבורו
- רשומות Genre שרלוונטיות עבור דרך MovieGenreAssociation ומשם דרך Genre לפרטיהם המדויקים

הסבר לבחירת העיצוב הנוכחי של הטבלאות

בנינו לפי העקרונות שנלמדו:

- התחשבות בקשרים מובנים: one-to-one, one-to-many עם forgien key לחיסכון, many-to-many בטבלה (בשביל Normalization)
- התחשבות ב- Normalization
- שמות משמעותיים
- אינדקסים בשביל השאילתות

על הבחירה ב- MovieMetrics
בחרנו ליצור MovieMetrics ולא לשים את המידע הזה גם בטבלה של Movies כי מדובר על מטריקות יותר מספריות וסטטיסטיות פחות מהותיות כחלק מהסרט עצמו.

חסרון קיים: עוד טבלה שצריך לתחזק, ולהתחשב בה ב- Queries.
יתרון קיים: החלוקה לעוד טבלה עושה סדר לוגי, ההפרדה בין מטריקות למידע החיוני.

בצורה דומה גם על הבחירה בטבלה Certificate
(אם כי בה הייתה פחות התלבטות, וזה ממילא היה יותר מתבקש להפריד)

בנוסף, נציין תיקון שעשינו והוא שיש one-to-one בין metrics ל- movie ולכן בחרנו לשים foreign key בשניהם במקום טבלה נוספת MovieMetricsAssociation,
מה שמתאים אל many-to-many.

על הבחירה ב- Worker וכן Role
בחרנו לא לעשות את ה- Director וה- Actors בירושה מה- Worker.
אלא לעשות טבלה נוספת של Role שאומרת על התפקיד.

חסרון קיים: אם היו מאפיינים מפרידים של Director ו- Actor (מעבר להיותם מה שהם), לעשות את זה בירושה היה נוח מאוד.

יתרון קיים: המצב האמיתי של ה- Database הוא שאין מאפיינים מפרידים.
ולא ייאסף עוד מידע (בהתאם ל- user manual).
לכן העיצוב של טבלת Role ולא טבלאות בירושה מ- Worker עם Foreign Key ובלי שדות נוספים, הוא פחות בזבזני בטבלאות ויותר הגיוני לנו עיצובית.

על הבחירה ב- Association Tables
הן משקפות יחסים של many-to-many ואלטרנטיבות אחרות לא היו עומדות ב-

.Normalization
כאשר היה אפשר לעשות את היחס דרך forgien key בטבלה
(one-to-one, one-to-many) כך נעשה ממקום של חסכון בטבלאות.

אופטימיזציות של ה- Database

אופטימיזציה של Queries

- ביצוע של Select רק על העמודות הרלוונטיות
- הימנעות מטבלאות זמניות שלא לצורך קריאות
- שכתוב שאילתות לצורה יעילה יותר

אופטימיזציה של Scheme

- שימוש ב- Data Types קטנים ככל האפשר, שימוש ב- UNSIGNED כשניתן
- עקרונות Normalization שעוזרות ליעילות
- הוספת בדיקות CHECK בתוך ה- Scheme כדי להימנע מבדיקות כאלה ב- Queries שכתבנו (שאינן DML)
- **שימוש באינדקסים** (מעבר ל- defaults, גם custom ones)

שימוש באינדקסים Custom
הוספנו אינדקסים מותאמים אישית לשאילתות שכתבנו.

- idx_movie_release_year הוא B-Tree כי release_year משמש ל- Order By וגם GROUP BY ב- query_1, וכן WHERE לפי =< ב- query_2 וגם ORDER BY בה.

- idx_metascore הוא B-Tree כי metascore משמש ל- Order By ב- query_4 וכן ל- ORDER BY ב- query_3

- idx_genre_name הוא Hash כי name של Genre מושווה בתנאי WHERE של query_2 (כדי להציג רק את ה- genre שבו המשתמש בחר)

- idx_role_name הוא Hash כי name של Role מושווה בתנאי WHERE של query_3 (להבדיל 'director' לעומת 'actor')

```
"CREATE INDEX idx_movie_release_year ON Movie(release_year)",
"CREATE INDEX idx_metascore ON MovieMetrics(metascore)",
"CREATE INDEX idx_genre_name ON Genre(name) USING HASH",
"CREATE INDEX idx_role_name ON Role(name) USING HASH",
```

אינדקס Full-Text עבור query_4 וכן query_5

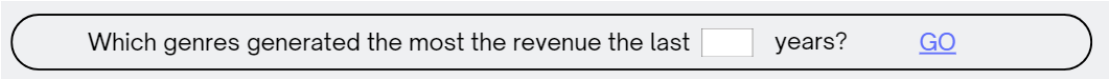
```
ALTER TABLE Movie
ADD FULLTEXT(description)
```

מעבר ל- custom indices:

- אינדקסים של Primary Key נוצרים ב- default.
- בנוסף, עשינו 2 טבלאות של Association Tables בשביל many-to-many ועשינו 2 Composite Indices כי לכל טבלה יצרנו Primary Key משתי ה- attributes
(על מנת לייעל את השאילתות JOIN עם הטבלה)

פירוט על השאילתות

Query 1:



המטרה: לאפשר למשתמש לראות בצורה נוחה את הז'אנרים הכי מוצלחים מבחינת הכנסה כלכלית, בכל אחת מהשנים לאחור שבחר.

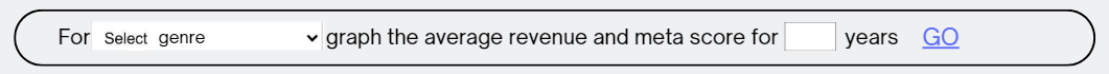
מחזיר: לכל שנה לפי מה שבחר המשתמש מחזיר את השנה, הז'אנר הכי מצליח מבחינה כלכלית, ההכנסה הכלכלית שלו.

נתמך בעיצוב ה- Database:
הטבלאות שבהם משתמשים הן: Movie, MovieMetrics, Genre.
הן כוללות את כל המידע הנדרש לשאילתא.
בנוסף, כדי להשתמש בהם עם JOINs ולאפשר יחס many-to-many
עשינו 2 טבלאות Association.
מעבר לזה, עשינו אינדקס על release_year בתור אופטימיזציה על השאילתא.

תיאור מילולי כללי של השאילתא:

- מחשבת את ההכנסות הכוללות לכל סרט ומסווגת את ההכנסות לפי שנה וז'אנר
- בוחרת את הז'אנר עם ההכנסה המקסימלית לכל שנה, משאירה רק אותם
- ומוסיפה את השדות הרלוונטיים, ומתקבלת טבלת הפלט

Query 2:



המטרה: לאפשר למשתמש לראות בצורה נוחה את ממוצע ההכנסה וה- metascore של הז'אנר שבחר, בטווח השנים שבחר.

מחזיר: לכל שנה לפי מה שבחר המשתמש מחזיר את השנה, ההכנסה, דירוג metascore.

נתמך בעיצוב ה- Database:
הטבלאות שבהם משתמשים הן: Movie, MovieMetrics, Genre.
הן כוללות את כל המידע הנדרש לשאילתא.
בנוסף, כדי להשתמש בהם עם JOINs ולאפשר יחס many-to-many
עשינו 2 טבלאות Association.
מעבר לזה, עשינו אינדקס על release_year וכן אינדקס על name של Genre
בתור אופטימיזציות על השאילתא

תיאור מילולי כללי של השאילתא:

- מפלטרת לפי ז'אנר ושנה שהמשתמש בחר את המידע הרלוונטי (שנה, רווחים, דירוג)
- בוחרת את הז'אנר עם ההכנסה המקסימלית לכל שנה, משאירה רק אותם

Query 3:

Show the directors ordered by the average meta score of the movies they produced. GO

המטרה: לאפשר למשתמש להבין איזה במאים מוצלחים ולאיזה שחקנים שיתוף פעולה מוצלח עם במאי. מדד ההצלחה הוא ממוצע על ה- metascore.

מחזיר: מחזירה טבלה המדורגת לפי ה- metascore של הבמאים
לכל במאי, בשורה שלו השחקנים מסודרים לפי ההצלחה של כל אחד מהם עם הבמאי עצמו.
מדד ההצלחה של שיתוף פעולה של במאי ושחקן הוא ממוצע metascore של הסרטים שעבדו עליהם ביחד.

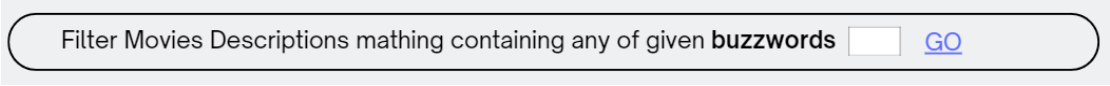
נתמך בעיצוב ה- Database:
הטבלאות שבהם משתמשים הן: Movie, MovieMetrics, Role, Worker
הן כוללות את כל המידע הנדרש לשאילתא.
בנוסף, כדי להשתמש בהם עם JOINs ולאפשר יחס many-to-many
עשינו טבלת MovieWorkerAssociation
מעבר לזה, עשינו אינדקס name של Role בתור אופטימיזציה על השאילתא.

תיאור מילולי כללי של השאילתא:

- מוצאת את שם הבמאי לכל סרט ואת רשימת השחקנים שהיו בסרטים שלו
- לכל במאי, מסדרת אותם לפי ממוצע metascore של הסרטים
- רשימת הבמאים מסודרת לפי metascore של ממוצע הסרטים שלו

נעשה שימוש ב- nested-query בשביל לעשות average metascore לכל שחקן עם במאי

Query 4:



המטרה: לאפשר למשתמש לראות תיאור של הסרטים
שכוללים בתיאור שלהם את אחת מה- buzzwords שהוא בחר
(למשל, כדי לשאוב השראה מסרטים אחרים שעוסקים בנושאים דומים)

מחזיר: לכל סרט שכולל בתיאור שלו אחד ה- buzzwords הוא מחזיר
שם סרט, תיאור, metascore

נתמך בעיצוב ה- Database:
הטבלאות שבהם משתמשים הן: Movie, MovieMetrics.
הן כוללות את כל המידע הנדרש לשאילתא.
מעבר לזה, עשינו אינדקס Full Text על description של Movie כדי לאפשר חיפוש בפנים הטקסט.

תיאור מילולי כללי של השאילתא:

- עושה פלטור לפי הסרטים שכוללים בתיאור לפחות buzzword אחד
- עושה JOINs בשביל metascore לכל סרט
- מיון לפי metascore, עד 20 סרטים

Query 5:

המטרה: לאפשר למשתמש לראות את רשימת הבמאים של הסרטים המוצלחים (מעל הממוצע revenue) שבהם התיאור כולל את מילת ה- buzzword שלו
(למשל, כדי שהמשתתף יבין עם איזה במאי לעבוד, מבחינת ניסיון שלו בתחום.
או ממי לשאוב השראה)

מחזיר: לכל סרט שכולל בתיאור שלו אחד ה- buzzwords הוא מחזיר
שם סרט, במאים, הכנסה, ממוצע הכנסה

נתמך בעיצוב ה- Database:
הטבלאות שבהם משתמשים הן:
Movie, MovieMetrics, Worker, MovieWorkerAssociation
הן כוללות את כל המידע הנדרש לשאילתא.
בנוסף, כדי להשתמש בהם עם JOINs ולאפשר יחס many-to-many
עשינו טבלה MovieWorkerAssociation.
מעבר לזה, עשינו אינדקס Full Text על description של Movie כדי לאפשר חיפוש בפנים הטקסט.

תיאור מילולי כללי של השאילתא:

- עושה פלטור לפי הסרטים שכוללים בתיאור את ה- buzzword
- מחשבת את ה- average revenue ומשאירה את שהכניסו מעל הממוצע
- מחזירה ביחד עם שמות הסרטים האלה, את שמות הבמאים, את הכנסותיהם, ואת ההכנסה הממוצעת

דוגמות הרצה

1. ליצירת ה- DB עם Tables ו- indexes לפי ה- schema יש להריץ
תוך תיקיית /src בצורה הבאה: **python create_db_script.py**

2. למילוי את ה- Tables במידע מתוך ה- Dataset יש להריץ
מתוך תיקיית /src בצורה הבאה: **python api_data_retrieve.py**

3. על מנת להריץ את ה- UI הטקסטואלי של ה- Queries יש להריץ
מתוך תיקיית /src בצורה הבאה: **python queries_execution.py**

מומלץ לעשות **pip install -r requirements.txt** לפני תחילת ההרצות.

נרחיב הדגמה של ה- queries_execution.

כך נראה הממשק, לאחר ההרצה:

```
---------------------------------------------
Select an option:
1 - Show table of top genres by year by revenue
2 - Graph revenue and rating by year according to genre
3 - Display directors ordered by Average meta score of their movies
4 - Display the TOP 20 movies containing one of the buzzwords, and their descriptions
5 - Show metrics on movie that contains the buzzword and has more than average revenue, shows the revenue and director
exit - exits from the program
help - shows the options menu
---------------------------------------------

Enter your choice (1, 2, 3, 4, 5, exit, help):
```

- בחירה של help מראה שוב את צג זה
- בחירה של exit מסיימת את ריצת הסקריפט
- בחירה באפשרויות שתואמות כל query, מודגמות להלן

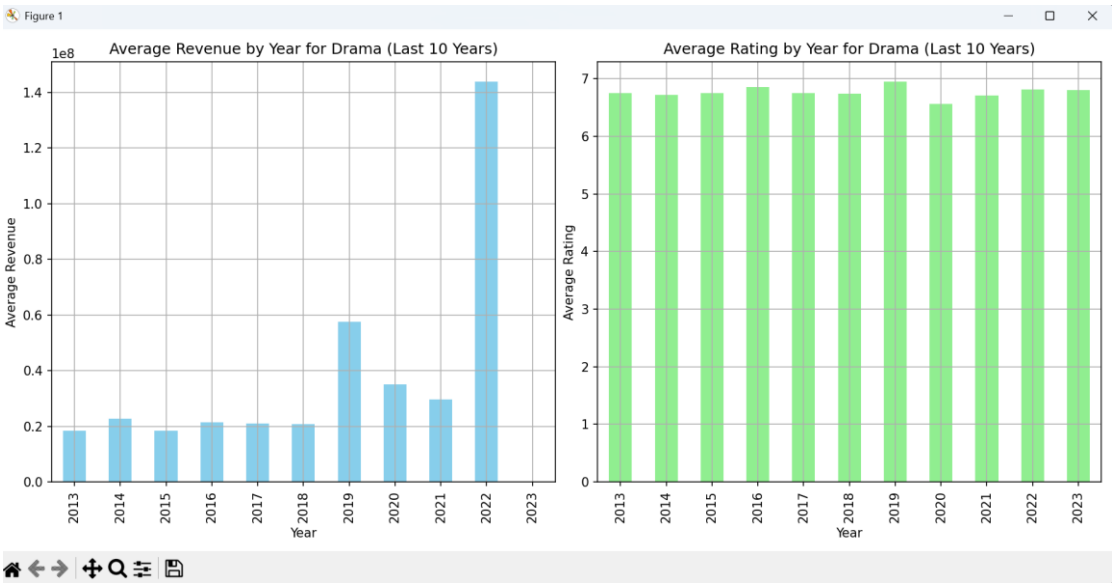
דוגמה להרצה של **query1** – Show table of top genres by year by revenue

```
Enter your choice (1, 2, 3, 4, 5, exit, help): 1
Please enter how many years of data you want to see : 4
    Year  Top Genre Max Revenue
0   2022     Action   718730000
1   2022      Drama   718730000
2   2021     Action   804750000
3   2021  Adventure   804750000
4   2021    Fantasy   804750000
5   2020     Action   206310000
6   2020     Comedy   206310000
7   2020      Crime   206310000
8   2019     Action   858370000
9   2019  Adventure   858370000
10  2019      Drama   858370000
Enter your choice (1, 2, 3, 4, 5, exit, help):
```

דוגמה להרצה של **query2** – Graph revenue and rating by year according to genre

```
Enter your choice (1, 2, 3, 4, 5, exit, help): 2
Available Genres:
Action
Adventure
Animation
Biography
Comedy
Crime
Drama
Family
Fantasy
Film-Noir
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Sport
Thriller
War
```

```
Please enter a genre from the list above: Drama
Please enter how many years of data you want to see : 10
```



דוגמה להרצה של **query3** –
Display directors ordered by Average meta score of their moves

```
Enter your choice (1, 2, 3, 4, 5, exit, help): 3
0            Jacques Tati
1           Steve McQueen
2       Ronnie Del Carmen
3            Orson Welles
4            Jafar Panahi
               ...
304         Steffen Haars
305         Til Schweiger
306       Victor Sjöström
307    Vondie Curtis-Hall
308        Yilmaz Erdogan
Name: Director, Length: 309, dtype: object

Please enter a director's name: Steve McQueen
Director Steve McQueen is suitable to work with these actors in this order:
 (according to Average meta score of the movies they worked together on)
Chiwetel Ejiofor
```

דוגמה להרצה של **query4 -**

Display the TOP 20 movies containing one of the buzzwords, and their descriptions

```
Enter your choice (1, 2, 3, 4, 5, exit, help): 4
Please enter a buzzword. Type 'N' to stop: israel
Please enter a buzzword. Type 'N' to stop: tennis
Please enter a buzzword. Type 'N' to stop: N
-----------------------------------------

TOP 1 best-match by metascore
Movie Title: Strangers on a Train
Description: A psychopathic man tries to forcibly persuade a tennis star to agree to his theory that two strangers can get away with murder by submitt
ing to his plan to kill the other's most-hated person.
Metascore: 88

TOP 2 best-match by metascore
Movie Title: Can You Ever Forgive Me?
Description: When Lee Israel falls out of step with current tastes, she turns her art form to deception.
Metascore: 87

TOP 3 best-match by metascore
Movie Title: The Band's Visit
Description: A band comprised of members of the Egyptian police force head to Israel to play at the inaugural ceremony of an Arab arts center, only to
 find themselves lost in the wrong town.
Metascore: 80

TOP 4 best-match by metascore
Movie Title: King Richard
Description: A look at how tennis superstars Venus and Serena Williams became who they are after the coaching from their father Richard Williams.
Metascore: 76
```

דוגמה להרצה של **query5** -

Show metrics on movie that contains the buzzword and has more than average revenue, shows the revenue and director

```
Enter your choice (1, 2, 3, 4, 5, exit, help): 5
Please enter the buzzword: wedding
_____________________________________________

TOP 1 best-match by metascore
title: The Hangover Part II
directors: Bradley Cooper
revenue: 254460000
average_revenue: 40746071.4286

TOP 2 best-match by metascore
title: The Hangover
directors: Bradley Cooper
revenue: 277320000
average_revenue: 40746071.4286

TOP 3 best-match by metascore
title: Corpse Bride
directors: Johnny Depp
revenue: 53360000
average_revenue: 40746071.4286
```