

# Movie Data Management System (TMDB)

Database Management Systems

Master's of Data Science in Business

Student: Izabela Madoyan

Այս նախագծի նպատակն է ստեղծել ֆիլմերի տվյալների բազա՝ օգտագործելով իրական տվյալներ TMDB (The Movie Database) API-ից: Համակարգը նախատեսված է ֆիլմերի, դերասանների, ժամրերի և դրանց միջև առկա փոխկապակցվածությունների վերլուծության համար:

Բազան բաղկացած է հետևյալ table-ներից՝

Table	Key Columns	Relationships
genres	genre_id (PK), genre_name	-
people	person_id (PK), name, birth_date, biography	-
movies	movie_id (PK), release_date (PK), title, runtime, budget, revenue, popularity	-
movie_genres	movie_id (FK), release_date (FK), genre_id (FK)	→ movies, genres
movie_cast	cast_id (PK), movie_id (FK), release_date (FK), person_id (FK), character_name, role	→ movies, people

Ֆիլմերը **Partitioning**-ի միջոցով բաժանված են երկու սեղմենտի՝ movies\_classic (մինչև 2000թ.) և movies\_modern (2000-2030թթ.):

Նախագծում ներկայացված են 10 SQL վերլուծական հարցումներ:

- Նկարագրություն:** Գտնել բոլոր «Drama» ժամրի ֆիլմերը և դրանց գլխավոր դերասաններին՝ ներառելով ֆիլմի վերնագիրը, ժամրը, դերասանի անունը և կերպարի անունը:

**Concepts:** JOINs, Filtering (WHERE), Limiting (LIMIT)

- Նկարագրություն:** Հաշվել յուրաքանչյուր ժամրի ֆիլմերի ընդհանուր քանակը և դրանց թերած ընդհանուր եկամուտը:

**Concepts:** Aggregations (COUNT, SUM), Grouping (GROUP BY), Sorting (ORDER BY)

3. **Նկարագրություն:** Գտնել այն Փիլմերը, որոնց եկամուտը գերազանցում է բյուջեն, և հաշվարկել զուտ շահույթը:  
**Concepts:** Filtering (WHERE), Grouping (GROUP BY)
4. **Նկարագրություն:** Գտնել այն դերասաններին, ովքեր տվյալների բազայում ունեն գոնե 2 և ավելի գրանցված դերեր:  
**Concepts:** Grouping (GROUP BY), Advanced Filtering (HAVING), Aggregations (COUNT)
5. **Նկարագրություն:** Յուրաքանչյուր ժանրի ներսում առանձին դասակարգել (rank) Փիլմերն ըստ դրանց հանրաճանաչության:  
**Concepts:** Window Functions (RANK()), Partitioning (OVER PARTITION BY), Sorting inside Windows (ORDER BY)
6. **Նկարագրություն:** Յուրաքանչյուր ժանրի համար առանձնացնել միայն այն մեկ Փիլմը, որն ունի ամենաբարձր վարկանիշը:  
**Concepts:** Subqueries, Partitioning (OVER PARTITION BY), Sorting inside Windows (ORDER BY), Filtering on Window results (WHERE)
7. **Նկարագրություն:** Համեմատել յուրաքանչյուր Փիլմի եկամուտը նախորդի հետ և հաշվել եկամուտների աճողական հանրագումարը:  
**Concepts:** Window Functions (LAG(), SUM() OVER)
8. **Նկարագրություն:** Գտնել այն դերասաններին, ովքեր խաղացել են այնպիսի Փիլմերում, որոնց բյուջեն բարձր է բազայում առկա բոլոր Փիլմերի միջին բյուջեից:  
**Concepts:** CTE (Common Table Expressions), Aggregations in CTE (AVG), Joins
9. **Նկարագրություն:** Առանձնացնել Փիլմերի թողարկման տարեթիվը և դասակարգել դրանք ըստ տևողության (Կարճամետրաժ, Լիամետրաժ, Եպիկ):  
**Concepts:** Date Functions (EXTRACT), Conditional Logic (CASE WHEN), Advanced Filtering (BETWEEN)
10. **Նկարագրություն:** Ստուգել տվյալների բազայի արդյունավետությունը ժամանակակից Փիլմերի (Modern Partition) հարցման ժամանակ:  
**Concepts:** EXPLAIN ANALYZE, Filtering on Partition Key

Վերջում ներկայացնենք ER դիագրամը՝

