Analiza IMDb-a

Opis projekta iz predmeta sistemi baza podataka

Agregacije

1. 10 najpopularnijih žanrova u odnosu na njihovu prosečnu ocenu i broj pojavljivanja
2. Koliko epizoda imaju 10 serija sa najvećom prosečnom ocenom?
3. Koja sezona je najbolje ocenjena u okviru serije?
4. Kolika je razlika prosečne ocene serije u odnosu na prosečnu ocenu njenih epizoda?
5. Koji reditelj je snimio najviše filmova?
6. 5 najboljih glumaca u odnosu na broj snimljenih filmova i prosečne ocene
7. Osobe koje istovremeno obavljaju funkciju reditelja i glumca
8. Koja kombinacija žanra je najpopularnija?
9. Koliki je broj umrlih osoba u 21. veku?
10. Koliki je ukupan broj minuta koje su proveli glumci na sceni sa najvišom prosečnom ocenom i bar 10 snimljenih filmova?

Inicijalna logička šema

- Sastoji se iz jedne kolekcije: imdb
- Kreirana korišćenjem Python skripte, uvezivanjem četiri različitih csv datoteka (~6GB)

Inicijalna logička šema

- Dokument predstavlja zapis o jednom naslovu iz IMDb baze podataka
- Odgovara jednom redu iz datoteke sa osnovnim podacima o nasovu


```
{
  "tconst":"tt0651840",
  "primaryTitle":"Blind Date",
  "titleType":"tvShort",
  "isAdult":false,
  "startYear":1993,
  "endYear":null,
  "runtimeMinutes":16,
  "genres":[
    "Comedy",
    "Short"
  ],
```

```
"rating":{
  "avgRating":8.0,
  "numVotes":1074
},
"episodes":{
  "tconst":"1",
  "primaryTitle":"episode_title",
  "titleType":"tvEpisode",
  "startYear":1993,
  "endYear":1995,
  "runtimeMinutes":20
}
```

```
"cast":[
  {
    "nconst":"nm0000100",
    "primaryName":"Rowan Atkinson",
    "birthYear":1955,
    "deathYear":null,
    "category":[
      "actor"
    ]
  }
],
```

Inicijalna logička šema – python skripta

```python
@measure_time
def initial_dictionary(file_path=None) -> dict:
    if not file_path:
        raise("Require file_path, got 'None'")

    df = load_data_into_dataframe(file_path=file_path)

    df["_id"] = df['tconst']
    convert_to_numeric(['isAdult', 'startYear', 'endYear', 'runtimeMinutes'], df, 'integer')
    df['isAdult'] = df['isAdult'].astype(bool)
    # df['startYear'] = df['startYear'].astype(int)
    # df['endYear'] = df['endYear'].astype(int)

    df['genres'] = df['genres'].str.split(",")
    df.set_index(df['tconst'], inplace=True)
    recnik = df.to_dict('index')

    for key, value in recnik.items():
        if recnik[key]['titleType'] in ['tvSeries', 'tvMiniSeries']:
            recnik[key]['episodes'] = []

    del df
    return recnik
```

```python
@measure_time
def insert_ratings(file_path, recnik):
    rating_df = load_data_into_dataframe(file_path=file_path)
    convert_to_numeric('averageRating', rating_df, 'float')
    convert_to_numeric('numVotes', rating_df, 'integer')

    for row in rating_df.itertuples():
        if not recnik.get(row.tconst):
            continue
        recnik[row.tconst]['rating'] = {
            "avgRating": row.averageRating,
            "numVotes": row.numVotes
        }

    del rating_df
```

```python
@measure_time
def insert_episodes(file_path, recnik):
    episode_df = load_data_into_dataframe(file_path=file_path)
    convert_to_numeric(['episodeNumber', 'seasonNumber'], episode_df, 'integer')

    for row in episode_df.itertuples():
        if not recnik.get(row.parentTconst) or not recnik.get(row.tconst):
            continue
        episode_info = recnik[row.tconst]
        episode_info['seasonNumber'] = row.seasonNumber
        episode_info['episodeNumber'] = row.episodeNumber
        recnik[row.parentTconst]['episodes'].append(episode_info)
        recnik.pop(row.tconst)

    del episode_df
```

Inicijalna logička šema - python skripta

```python
@measure_time
def insert_cast(title_info_file, cast_info_file, principal_info_path, recnik):
    basic_df = load_data_into_dataframe(title_info_file)
    principal_df = load_data_into_dataframe(principal_info_path)

    principal_df = filter_dataframe(principal_df, 'tconst', basic_df)

    del basic_df

    cast_df = load_data_into_dataframe(cast_info_file)

    principal_df = filter_dataframe(principal_df, 'nconst', cast_df)

    principal_df = principal_df.groupby(['tconst', 'nconst']).agg({'category': lambda x: ', '.join(set(x))}).reset_index()
    principal_df['category'] = principal_df['category'].str.split(',')

    joined_df = pd.merge(cast_df, principal_df, on='nconst')

    del cast_df, principal_df

    joined_df.drop(['primaryProfession', 'knownForTitles'], axis=1, inplace=True)
    convert_to_numeric(['birthYear', 'deathYear'], joined_df, 'integer')

    cast = (
        joined_df.groupby('tconst')
        .apply(lambda x: x[['nconst', 'primaryName', 'birthYear', 'deathYear', 'category']].to_dict('records'))
        .reset_index()
        .set_index('tconst')
        .rename(columns={0: 'cast'})
        .to_dict('index')
    )

    for key, value in cast.items():
        if not recnik.get(key):
            continue
        # value = {k: v for k, v in value.items() if v is not None}
        recnik[key]['cast'] = value['cast']
```

```python
@measure_time
def export_json(recnik, output_file_name):
    # json_output = json.dumps(recnik)
    i = 0
    with open(output_file_name, "w") as output_file:
        for _, value in recnik.items():
            value = {k: v for k, v in value.items() if k != 'tconst'}
            value['_id'] = i
            i+=1
            json_object = json.dumps(value)
            output_file.write(json_object + "\n")
```

Inicijalna logička šema – primeri agregacija

- Koliki je ukupan broj minuta koje su proveli glumci na sceni sa najvišom prosečnom ocenom i bar 10 snimljenih filmova?
- Vreme izvršavanja: 187 sekundi

```
db.imdb.aggregate([
    { $unwind: "$cast" },
    { $group: { _id: { "nconst": "$cast.nconst", "primaryName": "$cast.primaryName", "birthYear": "$cast.birthYear",
                       "deathYear": "$cast.deathYear" },
                "rating": {$avg: "$rating.avgRating"},
                "totalDuration": {$sum: "$runtimeMinutes"},
                "count": { $sum: 1 } } },
    { $match: { "count": {$gte: 10} } },
    { $sort: { "rating": -1 } },
    { $project: { _id: 0, "person": "$_id.primaryName", "rating": 1, "totalDuration": 1, "count": 1 } }
])
```

Inicijalna logička šema – primeri agregacija

- Osobe koje istovremeno obavljaju funkciju reditelja i glumca
- Vreme izvršavanja: 33 sekunde

```
db.imdb.aggregate([
    { $unwind: "$cast" },
    { $match: { $or: [ { "cast.category": {$all: ["actor", "director"]}},
                       { "cast.category": {$all: ["actress", " director"]}}]} },
    { $group: { _id: { "nconst": "$nconst", "person": "$cast.primaryName", "roles": "$cast.category" },
                "titles": {$addToSet: "$primaryTitle" } } },
    { $project: { _id: 0, "person": "$_id.person", "titles": "$titles" } },
])
```

Inicijalna logička šema – primeri agregacija

- 5 najboljih glumaca u odnosu na broj snimljenih filmova i prosečne ocene
- Vreme izvršavanja: 55 sekunde

```
db.imdb.aggregate([
    { $match: { $and: [ { "rating": { $exists: true }},
                        { "rating.avgRating": { $ne: null }} ],
                "cast.category": { $in: ["actor", "actress"] } } },
    { $unwind: "$cast" },
    { $group: { _id: { "nconst": "$cast.nconst", "primaryName": "$cast.primaryName" },
                "avgOcena": { $avg: "$rating.avgRating" },
                "occurrences": { $sum: 1 } } },
    { $sort: { "avgOcena": 1, "occurrences": -1 } },
    { $project: { _id: 0, "title": "$_id.primaryName", "rating": "$avgOcena", "occurrences": "$occurrences" } }
])
```

Inicijalna logička šema – primeri agregacija

- Kolika je razlika prosečne ocene serije u odnosu na prosečnu ocenu njenih epizoda?
- Vreme izvršavanja: 17 sekundi

```
db.imdb.aggregate([
    { $match: { "titleType": { $in:["tvSeries", "tvMiniSeries"] },
                $expr: { $gt: [{$size:"$episodes"}, 0] } } },
    { $unwind:"$episodes" },
    { $group: { _id: { idSerije:"$tconst", title:"$primaryTitle" },
                rating_episode: { $avg:"$episodes.rating.avgRating"},
                rating_series: { $first: "$rating.avgRating" } } },
    { $project: { _id:1, title:"$_id.title", rating_episode:1, rating_series:1,
                  difference:{ $abs: { $subtract:["$rating_episode", "$rating_series"] } } } },
    { $match: { difference: { $ne:null } } },
  ])
```

Logička šema prilagođena agregacijama

- Sastoji se iz 3 kolekcije: *imdb, cast, series*
- Kolekcija *imdb* je ista kao u inicijalnoj šemi
- Primenom šablona baketiranja na inicijalnoj šemi dobijene su ostale 2 kolekcije
- Baketiranje vršeno po onim atributima koji su se često javljali prilikom grupisanja u agregacijama
- Korišćeni su mongo upiti nad starim kolekcijama za kreiranje i popunjavanje novih

Logička šema prilagođena agregacijama

- Dobijena primenom šablona baketiranja i proračunavanja na kolekciju *cast* iz inicijalne šeme
- Jedan dokument nove kolekcije se odnosi na jednu osobu
- Dodati su atributi dobijeni proračunavanjem koji su često korišćeni u upitima


```
{
  "_id":"6487d849ce102baac9f5d974",
  "nconst":"nm0000046",
  "primaryName":"Vivien Leigh",
  "birthYear":1913,
  "deathYear":1967,
  "titles":[
    {
      "id":"64865721497807297dedb32f",
      "title":"The Prince, the Showgirl and Me"
    }
  ],
  "details":{
    "total_minutes_acting":1954.0,
    "total_times_acting":19.0,
    "total_times_directing":0.0,
    "average_rating":6.85
  }
}
```

Logička šema prilagođena agregacijama - skripta

```
db.imdb.aggregate([
    { $unwind: "$cast" },
    { $group: { _id:{ "nconst": "$cast.nconst", "primaryName": "$cast.primaryName", "birthYear": "$cast.birthYear", "deathYear": "$cast.deathYear", },
                "category": { $push: "$cast.category"},
                "titles": { $push: {"id": "$_id", "title": "$primaryTitle"}},
                "averageRating": { $avg: "$rating.avgRating"},
                "total_minutes_acting": {
                    $sum: {
                        $cond: [ { $or: [ { $in: ["actor", "$cast.category"] }, { $in: ["actress", "$cast.category"] } ]}, "$runtimeMinutes", 0 ] }},
                "total_times_acting": {
                    $sum: {
                        $cond: [ { $or: [ { $in: ["actor", "$cast.category"]}, { $in: ["actress", "$cast.category"]} ] }, 1, 0] } },
                "total_times_directing": {
                    $sum: {
                        $cond: [ { $or: [ {$in: ["director", "$cast.category"]}, {$in: [" director", "$cast.category"]}] }, 1, 0] } }, } },
    { $project: { _id: 0, "nconst": "$_id.nconst", "primaryName": "$_id.primaryName",
                  "birthYear": "$_id.birthYear", "deathYear": "$_id.deathYear", "category": "$_id.category", "titles": "$titles",
                  "details": {
                      "total_minutes_acting": "$total_minutes_acting",
                      "total_times_acting": "$total_times_acting",
                      "total_times_directing": "$total_times_directing",
                      "average_rating": "$averageRating"
                  } } },
    { $out: {db: "data", coll: "cast_new"}}
], {allowDiskUse: true});
```

Logička šema prilagođena agregacijama

- Dokument dobijen primenom šablona baketiranja nad kolekcijom imdb iz inicijalne šeme, grupisanjem po tipu naslova, konkretno serije
- Dodati su atributi dobijeni proračunavanjem koji su često korišćeni u upitima


```
{
   "_id":"64877a6d67f7238e3a8dc7a4",
   "tconst":"tt0035599",
   "primaryTitle":"Voice of Firestone Televues",
   "titleType":"tvSeries",
   "isAdult":false,
   "startYear":1943,
   "endYear":1947,
   "average_rating":{
      "episodes":3.5,
      "series":4.22
   },
   "number_of_episodes":1,
   "episodes":[
      {
         "tconst":"tt24373634",
         "primaryTitle":"Premiere Show on WNBT, New York City",
         "titleType":"tvEpisode",
         "isAdult":false,
         "startYear":1943,
         "endYear":null,
         "runtimeMinutes":12,
         "seasonNumber":1,
         "episodeNumber":1,
         "rating":{
            "avgRating":3.5,
            "numVotes":5000
         }
      }
   ]
}
```

Logička šema prilagođena agregacijama - skripta

```
db.imdb.aggregate([
    { $match: { "titleType": { $in: ["tvSeries", "tvMiniSeries"] } } },
    { $unwind: "$episodes" },
    { $group: { _id: { "tconst": "$tconst", "primaryTitle": "$primaryTitle", "titleType": "$titleType", "isAdult": "$isAdult",
                       "startYear": "$startYear", "endYear": "$endYear", "runtimeMinutes": "$runtimeMinutes", "genres": "$genres",
                        "avgRating": "$rating.avgRating", "cast": "$cast" },
            "episodes": {$addToSet: "$episodes"},
            "avg_episodes_rating": {$avg: "$episodes.rating.avgRating"} } },
    { $project: { _id: 0, "tconst": "$_id.tconst", "primaryTitle": "$_id.primaryTitle", "titleType": "$_id.titleType", "isAdult": "$_id.isAdult",
                        "startYear": "$_id.startYear", "endYear": "$_id.endYear", "runtimeMinutes": "$runtimeMinutes", "genres": "$genres",
                        "average_rating": { "series": "$_id.avgRating", "episodes": "$avg_episodes_rating" },
                        "number_of_episodes": {$size: "$episodes"}, "episodes": "$episodes", "cast": "$cast" } },
    { $out: { db: "data", coll: "series" } }
], {allowDiskUse: true})
```

Logička šema prilagođena agregacijama

primeri agregacija

- Koliki je ukupan broj minuta koje su proveli glumci na sceni sa najvišom prosečnom ocenom i bar 10 snimljenih filmova?
- Vreme izvršavanja: 8 sekundi

```
db.cast.aggregate([
    { $sort: {"details.average_rating": -1} },
    { $match: { "titles.10": {$exists: true},
                "category": { $in: ["actor", "actress"] } } },
    { $project: { "person": "$primaryName", "total_minutes_acting": "$details.total_minutes_acting", } }
])
```

Logička šema prilagođena agregacijama
primeri agregacija

- Osobe koje istovremeno obavljaju funkciju reditelja i glumca
- Vreme izvršavanja: 0.052 sekunde

```
db.cast.aggregate([
    { $match: { "category": { $in: ["actor", " director"] } } },
    { $limit: 5 },
    { $project: { "_id":0, "primaryName":1, "titles": 1, } }
])
```

Logička šema prilagođena agregacijama
primeri agregacija

- 5 najboljih glumaca u odnosu na broj snimljenih filmova i prosečne ocene
- Vreme izvršavanja: 6 sekundi

```
db.cast.aggregate([
    { $match: { "category": { $in: ["actor", "actress"] } } },
    { $sort: { "details.average_rating": -1, "details.total_times_acting":-1 } },
    { $limit:5 },
    { $project: { "primaryName":1, "details.total_times_acting":1 } }
])
```

Logička šema prilagođena agregacijama

primeri agregacija

- Kolika je razlika prosečne ocene serije u odnosu na prosečnu ocenu njenih epizoda?
- Vreme izvršavanja: 0.008 sekunde

```
db.series.aggregate([
    { $match: { $and: [{ "average_rating.series": {$ne:null}},{"average_rating.episodes": {$ne:null}}] } },
    { $project: { difference:{ $abs: { $subtract:["$average_rating.series", "$average_rating.episodes"] } },
                  _id:0, name:"$primaryTitle",
                  rating: { rating_episode:"$average_rating.episodes", rating_series:"$average_rating.series" } } }
    ])
```

Poređenje performansi – redosled agregacija

1. 10 najpopularnijih žanrova u odnosu na njihovu prosečnu ocenu i broj pojavljivanja
2. Koliko epizoda imaju 10 serija sa najvećom prosečnom ocenom?
3. Koja sezona je najbolje ocenjena u okviru serije?
4. Kolika je razlika prosečne ocene serije u odnosu na prosečnu ocenu njenih epizoda?
5. Koji reditelj je snimio najviše filmova?
6. 5 najboljih glumaca u odnosu na broj snimljenih filmova i prosečne ocene
7. Osobe koje istovremeno obavljaju funkciju reditelja i glumca
8. Koja kombinacija žanra je najpopularnija?
9. Koliki je broj umrlih osoba u 21. veku?
10. Koliki je ukupan broj minuta koje su proveli glumci na sceni sa najvišom prosečnom ocenom i bar 10 snimljenih filmova?

Poređenje performansi – redosled agregacija