Analiza popularnosti video igara pomoću podataka o prenosima sa *Twitch*-a

OPIS PROJEKTA IZ PREDMETA SISTEMI BAZA PODATAKA

Agregacije

1. Pronaći 10 igrica koje su imale najviše gledalaca u jednom trenutku (u okviru svih prenosa koji su u tom trenutku bili aktivni).
2. Za svakog autora odrediti koju je igru igrao najveći broj minuta.
3. Za svaki sat u danu pronaći koja je igra imala u prosjeku najviše prenosa.
4. Koje igre je prenosio autor koji je ostvario najveći rast broja pratilaca u posmatranom vremenskom periodu?
5. Koje žanrove je prenosio najveći broj autora?
6. Za svaku igru odrediti koliko različitih autora je igralo u bar jednom prenosu, i pronaći 3 autora koja su je igrala u najviše prenosa.
7. Pronaći 5 igara koje su prenošene najveći broj minuta.
8. Koliki je broj prenosa, i prosjek, minimum, i maksimum prosječnog broja gledalaca po prenosu za svaku igru?
9. Za svaki žanr odrediti prosječno i maksimalno trajanje prenosa, i prosječni broj gledalaca po prenosu.
10. Za sve autore koji su najveći broj minuta igrali igru The Evil Within, a pored toga su igrali bar još jednu igru, pronaći koje su još igre igrali, u koliko prenosa, koliko ukupno minuta, i koliko minuta u prosjeku po prenosu.

Inicijalna logička šema

- Sastoji se iz 2 kolekcije: *games* i *streams*
- Slična strukturi skupa podataka
- Za kreiranje i popunjavanje kolekcija korištene *Python* skripte, upotrebom *PyMongo* i *csv* paketa

Inicijalna logička šema

KOLEKCIJA *GAMES*

- Dokument predstavlja zapis o jednoj video igri
- Odgovara jednom redu iz datoteke sa osnovnim podacima o video igrama

Inicijalna logička šema

SKRIPTA ZA POPUNJAVANJE KOLEKCIJE *GAMES*

```
def add_games_to_db(self, url, db_name):
    client = pymongo.MongoClient(url)
    db = client[db_name]
    games = []
    self._games = {}
    with open(self._file, 'r', encoding = 'cp850') as csv_file:
        reader = csv.DictReader(csv_file)
        for row in reader:
            game = get_game(row)
            games.append(game)
            self._games[game['name']] = get_mini_game(game)

    db['games'].insert_many(games)
```

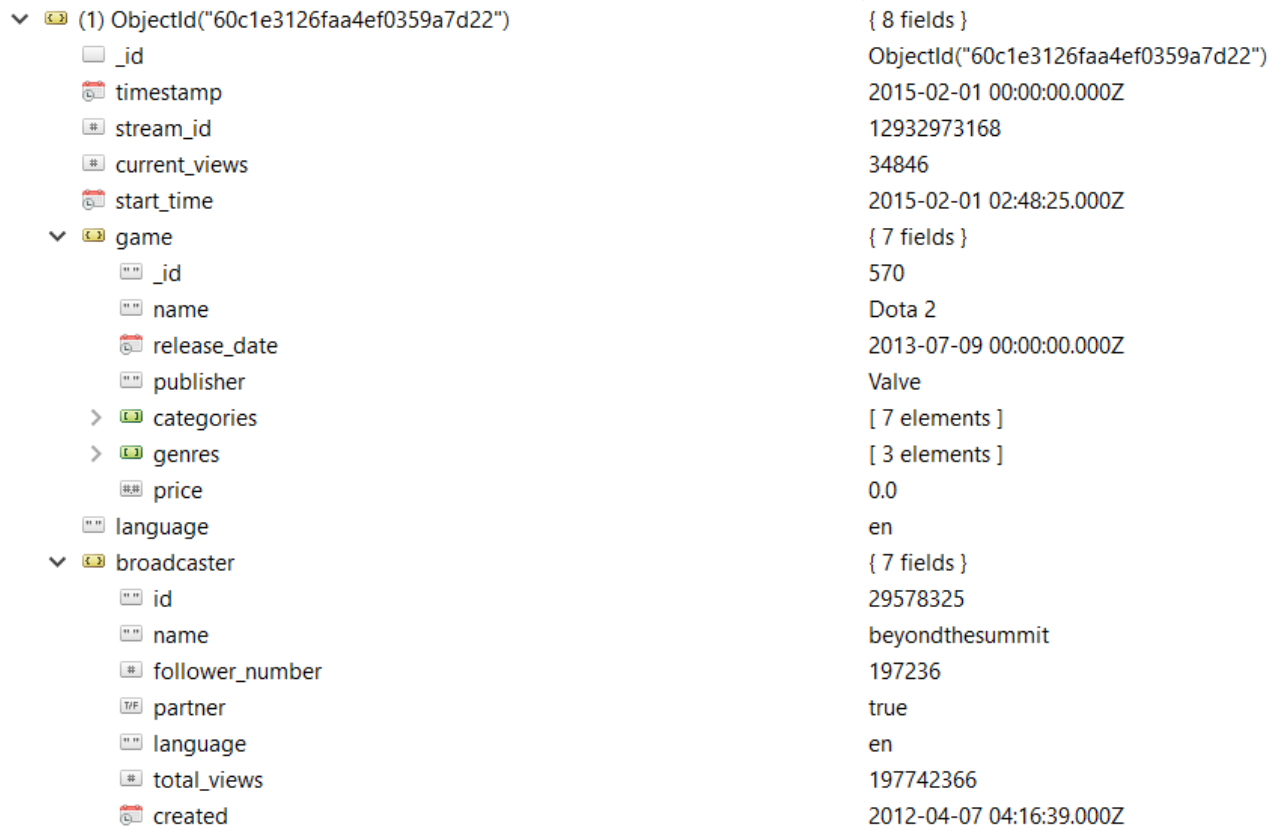
```
def get_game(row) -> dict:
    eng = False
    if (row['english'] == 1):
        eng = True

    return {
        '_id': row['appid'],
        'name': row['name'],
        'release_date': parser.parse(row['release_date']),
        'english': eng,
        'developer': row['developer'],
        'publisher': row['publisher'],
        'platforms': row['platforms'].split(';'),
        'required_age': int(row['required_age']),
        'categories': row['categories'].split(';'),
        'genres': row['genres'].split(';'),
        'steamspy_tags': row['steamspy_tags'].split(';'),
        'achievements': int(row['achievements']),
        'positive_ratings': int(row['positive_ratings']),
        'negative_ratings': int(row['negative_ratings']),
        'average_playtime': int(row['average_playtime']),
        'median_playtime': int(row['median_playtime']),
        'owners': row['owners'],
        'price': float(row['price'])
    }
```

Inicijalna logička šema

KOLEKCIJA *STREAMS*

- Dokument odgovara jednom redu iz datoteka skupa podataka o prenosima, odnosno predstavlja zapis o jednom prenosu u jednom vremenskom trenutku
- Dodat je podatak o vremenu iz naziva datoteke
- Dodata je proširena referenca na video igru



Inicijalna logička šema

SKRIPTA ZA POPUNJAVANJE KOLEKCIJE *STREAMS*

```
def add_games_to_db(self, url, db_name, games):
    client = pymongo.MongoClient(url)
    db = client[db_name]

    for file in os.listdir(self._dir):
        print('\t' + file)
        timestamp = get_datetime(file)
        streams = []
        with open(os.path.join(self._dir, file), 'r', encoding = 'cp850') as csv_file:
            reader = csv.DictReader(csv_file, fieldnames=['stream_id', 'current_views',
            for row in reader:
                if row['game'] in games:
                    try:
                        streams.append(get_stream(timestamp, row, games[row['game']]))
                    except:
                        print("Bad row")

        if (len(streams) > 0):
            db['streams'].insert_many(streams)
```

```
def get_stream(timestamp, row, game):
    partner = True
    if (row['partner'] == '-1'):
        partner = False

    broadcaster = {
        'id': row['broadcaster_id'],
        'name': row['broadcaster_name'],
        'follower_number': int(row['follower_number']),
        'partner': partner,
        'language': row['broadcaster_language'],
        'total_views': int(row['total_views']),
        'created': parser.parse(row['broadcaster_created'])
    }

    if broadcaster['follower_number'] == -1:
        broadcaster['follower_number'] = 0
    if broadcaster['total_views'] == -1:
        broadcaster['total_views'] = 0

    return {
        'timestamp': timestamp,
        'stream_id': int(row['stream_id']),
        'current_views': int(row['current_views']),
        'start_time': parser.parse(row['start_time']),
        'game': game,
        'language': row['language'],
        'broadcaster': broadcaster
    }
```

Inicijalna logička šema

PRIMJERI AGREGACIJA

8. Koliki je broj prenosa, i prosjek, minimum, i maksimum prosječnog broja gledalaca po prenosu za svaku igru? – 176 sekundi

```
db.getCollection('streams').aggregate([
    {$group: {_id: {game: "$game.name", stream_id: "$stream_id"}, average_views: {$avg: "$current_views"}}},
    {$group: {_id: "$_id.game",
              average_views: {$avg: "$average_views"},
              max_views: {$max: "$average_views"},
              min_views: {$min: "$average_views"},
              count: {$sum: 1}}},
    {$sort: {"average_views": -1}},
    {$project: {_id: 0,
                game: "$_id",
                max_views_per_stream: "$max_views",
                average_views_per_stream: "$average_views",
                min_views_per_stream: "$min_views",
                number_of_streams: "$count"}}
], {allowDiskUse: true})
```

Inicijalna logička šema

PRIMJERI AGREGACIJA

4. Koje igre je prenosio autor koji je ostvario najveći rast broja pratilaca u posmatranom vremenskom periodu? – 467 sekundi

```
db.getCollection('streams').aggregate([
    {$sort: {"timestamp": 1}},
    {$group: {_id: "$broadcaster.name",
              follower_counts: {$push: "$broadcaster.follower_number"},
              played_games: {$addToSet: "$game.name"}}},
    {$project: {_id: 0,
                broadcaster: "$_id",
                played_games: 1,
                start_follower_count: {$arrayElemAt: [ "$follower_counts", 0 ]},
                end_follower_count: {$arrayElemAt: [ "$follower_counts", -1 ]}}},
    {$project: {broadcaster: 1, played_games: 1, follower_change: {$subtract: ["$end_follower_count", "$start_follower_count"]}}},
    {$sort: {"follower_change": -1}},
    {$limit: 1}
], {allowDiskUse: true})
```

Inicijalna logička šema

PRIMJERI AGREGACIJA

1. Pronaći 10 igrica koje su imale najviše gledalaca u jednom trenutku (u okviru svih prenosa koji su u tom trenutku bili aktivni). – 154 sekunde

```
db.getCollection('streams').aggregate([
    {$group: {_id: {game: "$game.name", timestamp: "$timestamp"}, total_views: {$sum: "$current_views"}}},
    {$group: {_id: "$_id.game", max_views: {$max: "$total_views"}}},
    {$sort: {"max_views": -1}},
    {$limit: 10},
    {$project: {_id: 0, game: "$_id", max_views: 1}}
], {allowDiskUse: true})
```

Inicijalna logička šema

PRIMJERI AGREGACIJA

2. Za svakog autora odrediti koju je igru igrao najveći broj minuta. – 192 sekunde

```
db.getCollection('streams').aggregate([
    {$group: {_id: {game: "$game.name", stream_id: "$stream_id", broadcaster: "$broadcaster.name"},
              start: {$min: "$timestamp"}, end: {$max: "$timestamp"}}},
    {$project: {game: "$_id.game",
                broadcaster: "$_id.broadcaster",
                duration_in_minutes: {$divide: [{$subtract: ["$end", "$start"]}, 60000]}}},
    {$group: {_id: {game: "$game", broadcaster: "$broadcaster"}, total_minutes: {$sum: "$duration_in_minutes"}}},
    {$project: {game: "$_id.game", broadcaster: "$_id.broadcaster", total_minutes: 1}},
    {$sort: {"total_minutes": -1}},
    {$group: {_id: "$broadcaster", games: {$push: {name: "$game", total_minutes: "$total_minutes"}}}},
    {$project: {_id: 0, broadcaster: "$_id", most_played_game: {$arrayElemAt: [ "$games", 0 ]}}}
], {allowDiskUse: true})
```

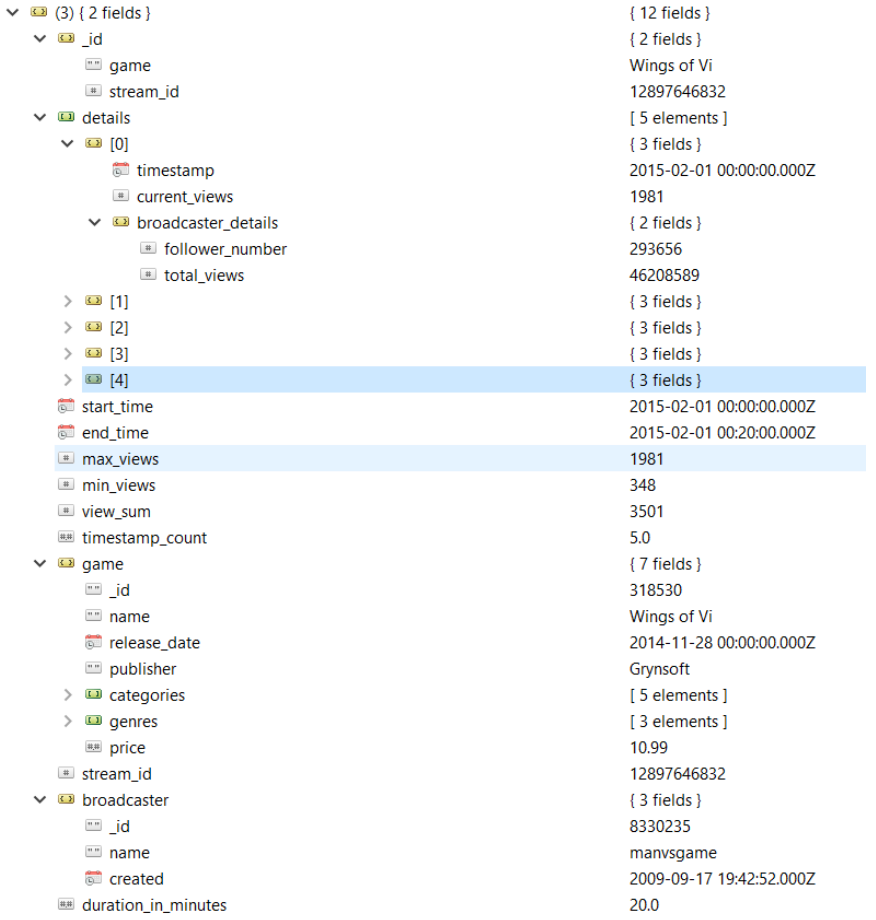
Logička šema prilagođena agregacijama

- Sastoji se iz 5 kolekcija: *games*, *streams*, *broadcasters*, *games-per-time*, i *games-broadcasters*
- Kolekcija *games* je ista kao u inicijalnoj šemi
- Ostale kolekcije su dobijene primjenom šablona baketiranja i proračunavanja na postojeće kolekcije
- Grupisanje pri baketiranju je vršeno po onim atributima koji su se često koristili za grupisanje u agregacijama
- Za kreiranje i popunjavanje novih kolekcija korišteni su Mongo upiti nad starim kolekcijama

Logička šema prilagođena agregacijama

KOLEKCIJA *STREAMS*

- Dobijena primjenom šablona baketiranja i proračunavanja na kolekciju *streams* iz inicijalne šeme, grupisanjem dokumenata po id-ju prenosa i video igri
- Jedan dokument nove kolekcije se odnosi na sekciju jednog prenosa u toku koje je igrana jedna video igra
- Podaci specifični za vremenski trenutak grupisani su u listu
- Dodati su atributi dobjieni proračunavanjem koji su često korišteni u upitima



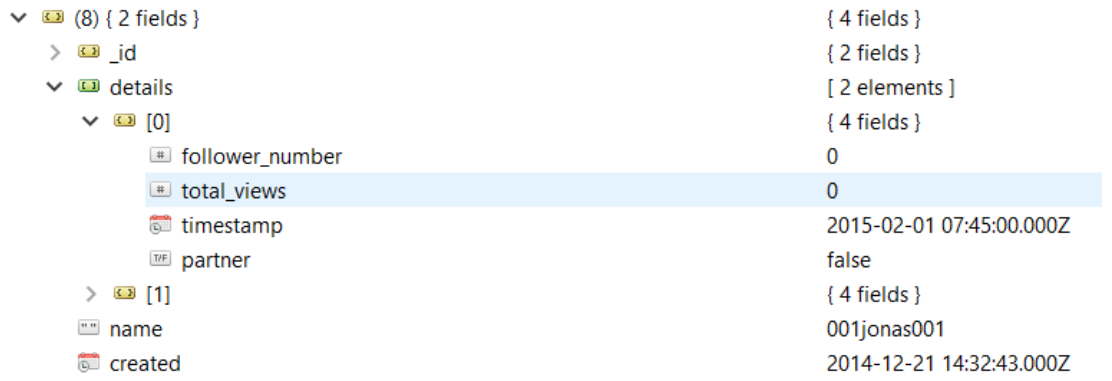
Logička šema prilagođena agregacijama

SKRIPZA ZA KREIRANJE KOLEKCIJE *STREAMS*

```
db.getCollection('streams').aggregate([
    {$group: {_id: {stream_id: "$stream_id",
                    game: "$game",
                    broadcaster: {_id: "$broadcaster.id", name: "$broadcaster.name", created: "$broadcaster.created"}},
              details: {$push: {timestamp: "$timestamp",
                                current_views: "$current_views",
                                broadcaster_details: {
                                        follower_number: "$broadcaster.follower_number",
                                        total_views: "$broadcaster.total_views"}}},
              start_time: {$min: "$timestamp"},
              end_time: {$max: "$timestamp"},
              max_views: {$max: "$current_views"},
              min_views: {$min: "$current_views"},
              view_sum: {$sum: "$current_views"},
              timestamp_count: {$sum: 1}
              }},
    {$project: {_id: {game: "$_id.game.name", stream_id: "$_id.stream_id"},
                game: "$_id.game",
                stream_id: "$_id.stream_id",
                broadcaster: "$_id.broadcaster",
                details: 1,
                start_time: 1,
                end_time: 1,
                duration_in_minutes: {$divide: [{$subtract: ["$end_time", "$start_time"]}, 60000]},
                max_views: 1,
                min_views: 1,
                view_sum: 1,
                timestamp_count: 1}},
    {$out: {db: "sbp-v2", coll: "streams"}}
], {allowDiskUse: true});
```

Logička šema prilagođena agregacijama
KOLEKCIJA *BROADCASTERS*

- Dobijena primjenom šablona baketiranja na kolekciju *streams* iz inicijalne šeme, grupisanjem po autoru
- Izbačeni su atributi koji se ne odnose direktno na autora
- Podaci specifični za vremenski trenutak grupisani su u listu



Logička šema prilagođena agregacijama

SKRIPZA ZA KREIRANJE KOLEKCIJE *BROADCASTERS*

```
db.getCollection('streams').aggregate([
    {$sort: {"timestamp": 1}},
    {$group: {_id: {name: "$broadcaster.name", id: "$broadcaster.id", created: "$broadcaster.created"},
              details: {$push: {
                        follower_number: "$broadcaster.follower_number",
                        total_views: "$broadcaster.total_views",
                        timestamp: "$timestamp",
                        partner: "$broadcaster.partner"}}
              }},
    {$project: {_id: {id: "$_id.id", name: "$_id.name"}, name: "$_id.name", created: "$_id.created", details: 1}},
    {$out: {db: "sbp-v2", coll: "broadcasters"}}
], {allowDiskUse: true})
```

Logička šema prilagođena agregacijama

KOLEKCIJA *GAMES-PER-TIME*

- Dobijena grupisanjem kolekcije *streams* iz inicijalne šeme po video igri i vremenskom trenutku
- Proračunati su podaci o gledanosti i broju prenosa za igru u posmatranom trenutku koji su bili potrebni za upite

Logička šema prilagođena agregacijama

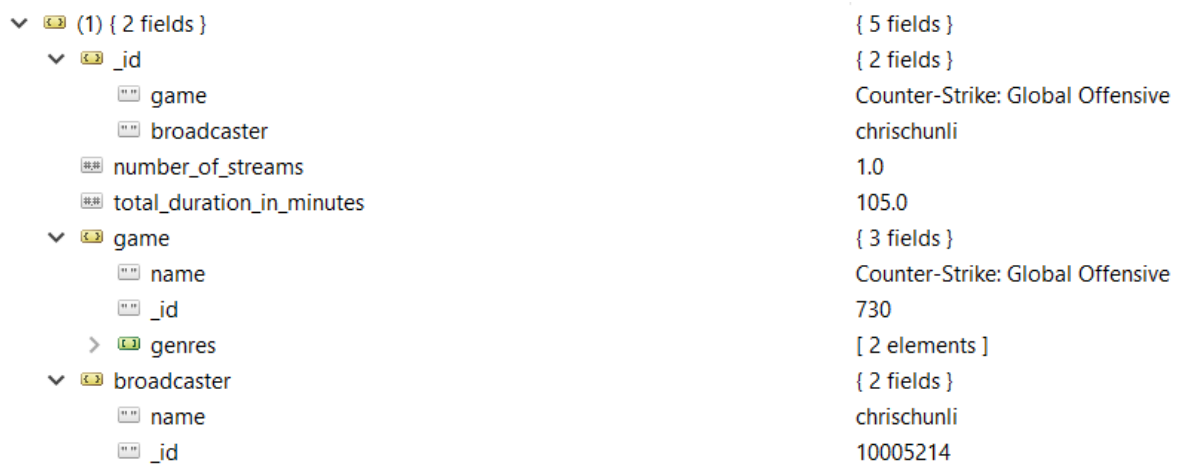
SKRIPZA ZA KREIRANJE KOLEKCIJE *GAMES-PER-TIME*

```
db.getCollection('streams').aggregate([
    {$group: {_id: {game: "$game.name", game_id: "$game._id", genres: "$game.genres", timestamp: "$timestamp"},
              total_views: {$sum: "$current_views"},
              total_streams: {$sum: 1}}},
    {$project: {_id: {game: "$_id.game", timestamp: "$_id.timestamp"},
                game: {name: "$_id.game", _id: "$_id.game_id", genres: "$_id.genres"},
                timestamp: "$_id.timestamp",
                total_views: 1,
                total_streams: 1}},
    {$out: {db: "sbp-v2", coll: "games-per-time"}}
    ], {allowDiskUse: true});
```

Logička šema prilagođena agregacijama

KOLEKCIJA *GAMES-BROADCASTERS*

- Dobijena grupisanjem kolekcije *streams* iz nove šeme po video igri i autoru
- Proračunati su podaci o broju prenosa koje je autor imao za igru, i ukupnom trajanju svih tih prenosa
- Podignuti su indeksi nad atributima *number_of_streams* i *total_duration_in_minutes*



Logička šema prilagođena agregacijama

SKRIPZA ZA KREIRANJE KOLEKCIJE *GAMES-BROADCASTERS*

```
db.getCollection('streams').aggregate([
    {$group: {_id: {broadcaster_id: "$broadcaster._id", broadcaster_name: "$broadcaster.name", game: "$game.name",
                    game_id: "$game._id", genres: "$game.genres"},
              number_of_streams: {$sum: 1},
              total_duration_in_minutes: {$sum: "$duration_in_minutes"}}},
    {$project: {_id: {game: "$_id.game", broadcaster: "$_id.broadcaster_name"},
              game: {name: "$_id.game", _id: "$_id.game_id", genres: "$_id.genres"},
              number_of_streams: 1, total_duration_in_minutes: 1,
              broadcaster: {name: "$_id.broadcaster_name", _id: "$_id.broadcaster_id"}}},
    {$out: {db: "sbp-v2", coll: "games-broadcasters"}}
], {allowDiskUse: true});
```

Logička šema prilagođena agregacijama

PRIMJERI AGREGACIJA

8. Koliki je broj prenosa, i prosjek, minimum, i maksimum prosječnog broja gledalaca po prenosu za svaku igru? – 9.61 sekundi

```
db.getCollection('streams').aggregate([
    {$project: {game: "$game.name", average_views: {$divide: ["$view_sum", "$timestamp_count"]}}},
    {$group: {_id: "$game",
              average_views: {$avg: "$average_views"},
              max_views: {$max: "$average_views"},
              min_views: {$min: "$average_views"},
              count: {$sum: 1}}},
    {$sort: {"average_views": -1}},
    {$project: {_id: 0,
                game: "$_id",
                max_views_per_stream: "$max_views",
                average_views_per_stream: "$average_views",
                min_views_per_stream: "$min_views",
                number_of_streams: "$count"}}
], {allowDiskUse: true})
```

Logička šema prilagođena agregacijama

PRIMJERI AGREGACIJA

4. Koje igre je prenosio autor koji je ostvario najveći rast broja pratilaca u posmatranom vremenskom periodu? – 5.31 sekundi

```
db.getCollection('broadcasters').aggregate([
    {$project: {_id: 0,
                broadcaster: "$name",
                start_details: {$arrayElemAt: [ "$details", 0 ]},
                end_details: {$arrayElemAt: [ "$details", -1 ]}}},
    {$project: {broadcaster: 1, follower_change: {$subtract: ["$end_details.follower_number", "$start_details.follower_number"]}}},
    {$sort: {"follower_change": -1}},
    {$limit: 1},
    {$lookup:
     {
       from: "games-broadcasters",
       localField: "broadcaster",
       foreignField: "_id.broadcaster",
       as: "games"
     }},
     {$project: {broadcaster: 1, follower_change: 1, games: "$games.game.name"}}
], {allowDiskUse: true})
```

Logička šema prilagođena agregacijama

PRIMJERI AGREGACIJA

1. Pronaći 10 igrica koje su imale najviše gledalaca u jednom trenutku (u okviru svih prenosa koji su u tom trenutku bili aktivni). – 2.53 sekundi

```
db.getCollection('games-per-time').aggregate([
    {$group: {_id: "$_id.game", max_views: {$max: "$total_views"}}},
    {$sort: {"max_views": -1}},
    {$limit: 10},
    {$project: {_id: 0, game: "$_id", max_views: 1}}
], {allowDiskUse: true})
```

Logička šema prilagođena agregacijama

PRIMJERI AGREGACIJA

2. Za svakog autora odrediti koju je igru igrao najveći broj minuta. – 5.99 sekundi sa indeksom, oko 8 sekundi bez indeksa

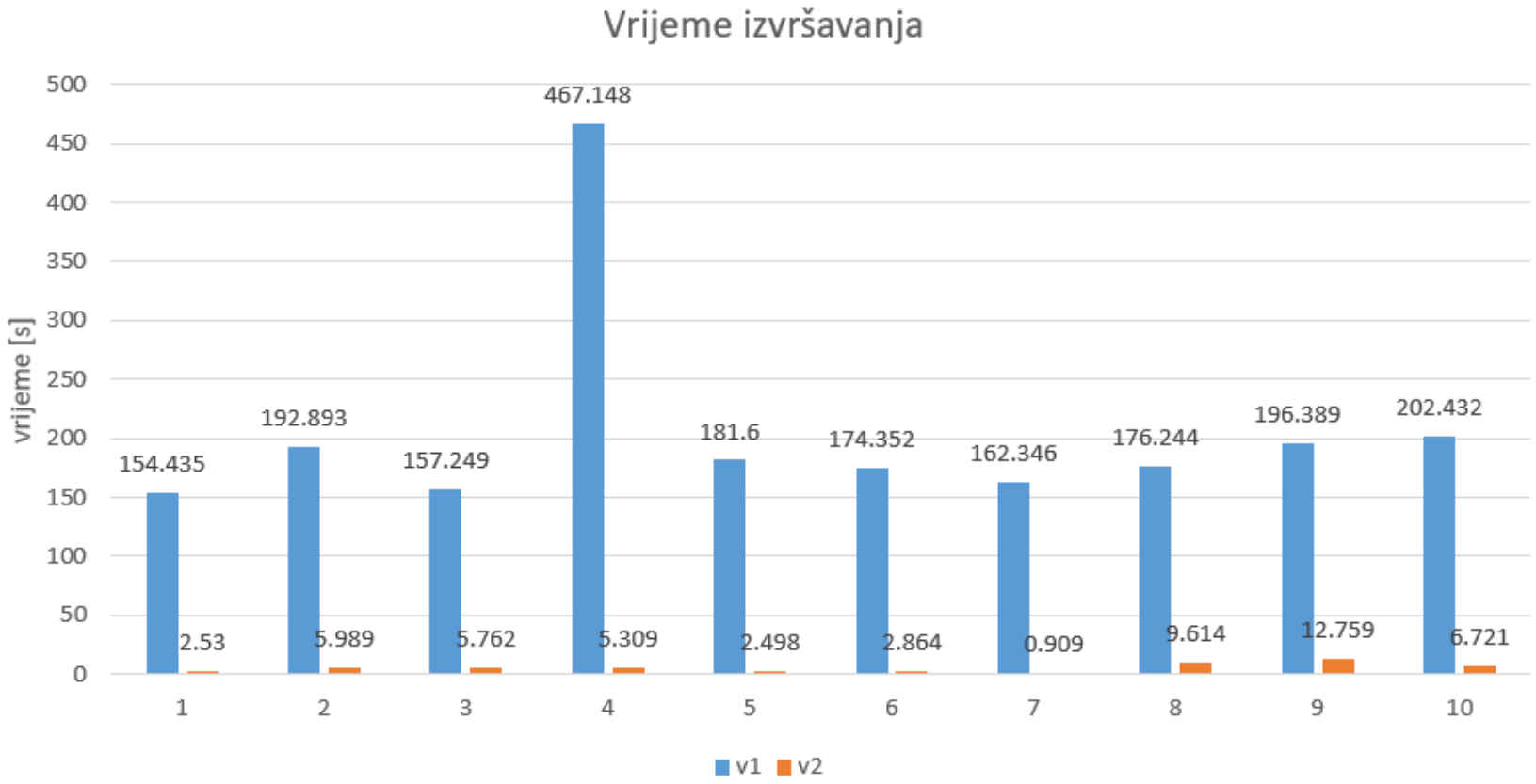
```
db.getCollection('games-broadcasters').aggregate([
    {$sort: {"total_duration_in_minutes": -1}},
    {$group: {_id: "$broadcaster.name", games: {$push: {name: "$game.name", total_minutes: "$total_duration_in_minutes"}}}},
    {$project: {_id: 0, broadcaster: "$_id", most_played_game: {$arrayElemAt: [ "$games", 0 ]}}}
], {allowDiskUse: true})
```

Poređenje performansi

REDOSLED AGREGACIJA

1. Pronaći 10 igrica koje su imale najviše gledalaca u jednom trenutku (u okviru svih prenosa koji su u tom trenutku bili aktivni).
2. Za svakog autora odrediti koju je igru igrao najveći broj minuta.
3. Za svaki sat u danu pronaći koja je igra imala u prosjeku najviše prenosa.
4. Koje igre je prenosio autor koji je ostvario najveći rast broja pratilaca u posmatranom vremenskom periodu?
5. Koje žanrove je prenosio najveći broj autora?
6. Za svaku igru odrediti koliko različitih autora je igralo u bar jednom prenosu, i pronaći 3 autora koja su je igrala u najviše prenosa.
7. Pronaći 5 igara koje su prenošene najveći broj minuta.
8. Koliki je broj prenosa, i prosjek, minimum, i maksimum prosječnog broja gledalaca po prenosu za svaku igru?
9. Za svaki žanr odrediti prosječno i maksimalno trajanje prenosa, i prosječni broj gledalaca po prenosu.
10. Za sve autore koji su najveći broj minuta igrali igru The Evil Within, a pored toga su igrali bar još jednu igru, pronaći koje su još igre igrali, u koliko prenosa, koliko ukupno minuta, i koliko minuta u prosjeku po prenosu.

Poređenje performansi



Poređenje performansi