Отчет к ДЗ по теме «Работа с джоинами и агрегациями»

Используя изученные материалы, построить запросы, отвечающие на следующие задачи:

Примечание: LIMIT 15 выбран просто в целях что бы в скриншот влез результат.

1. Найти жанры для каждого фильма.

   Примечание: Смущает немного для «КАЖДОГО». Получается что те, которые НЕ имеют жанра тоже должны быть в выборке (поэтому LEFT)

```
NB-R911LJCT.o3.ru :) select m.name, groupArray(g.genre) from imdb.movies m left join imdb.genres g on m.id = g.movie_id  group by m.name limit 15;

SELECT
    m.name,
    groupArray(g.genre)
FROM imdb.movies AS m
LEFT JOIN imdb.genres AS g ON m.id = g.movie_id
GROUP BY m.name
LIMIT 15

Query id: 7eae2d36-8067-4598-a594-50a9359e27f6

    ┌─name──────┬─groupArray(genre)─────────────────────────┐
 1. │ Ogostos   │ ['Drama']                                 │
 2. │ Gumiho    │ ['Fantasy','Romance']                     │
 3. │ Aphone    │ ['Short']                                 │
 4. │ Cascadia  │ ['Crime']                                 │
 5. │ Mit Herz  │ ['']                                      │
 6. │ Gengas    │ ['Short']                                 │
 7. │ Nehir     │ ['Adventure','Crime','Thriller']          │
 8. │ Killer    │ ['Short','Drama','Thriller']              │
 9. │ 23 Hours  │ ['Action','Thriller']                     │
10. │ Serrana   │ ['Musical','Short']                       │
11. │ Galate    │ ['Short']                                 │
12. │ Exchange  │ ['Short']                                 │
13. │ Chitthi   │ ['','']                                   │
14. │ Ola, La   │ ['']                                      │
15. │ Idhayam   │ ['']                                      │
    └───────────┴───────────────────────────────────────────┘

15 rows in set. Elapsed: 0.082 sec. Processed 783.39 thousand rows, 19.95 MB (9.55 million rows/s., 243.11 MB/s.)
Peak memory usage: 147.38 MiB.
```

2. Запросить все фильмы, у которых нет жанра

```
NB-R911LJCT.o3.ru :) select name from imdb.movies m left anti join imdb.genres g on m.id = g.movie_id limit 15;

SELECT name
FROM imdb.movies AS m
ANTI LEFT JOIN imdb.genres AS g ON m.id = g.movie_id
LIMIT 15

Query id: 184e6ad8-f4f0-4e06-9cb0-d86787745f42

    ┌─name──────────────────────────┐
 1. │ Botschaft                     │
 2. │ Botte i farten                │
 3. │ Bottom Land                   │
 4. │ Botn de fuego, El             │
 5. │ Botts sajt krsre              │
 6. │ Bouffon, Le                   │
 7. │ Bought & Sold                 │
 8. │ Bouka                         │
 9. │ Bouncer, The                  │
10. │ Bound and Determined          │
11. │ Bound for Home                │
12. │ Bouquetire des innocents, La  │
13. │ Bourrier, Le                  │
14. │ Boustaguy, al-                │
15. │ Bout d'essai                  │
    └───────────────────────────────┘

15 rows in set. Elapsed: 0.028 sec. Processed 783.39 thousand rows, 13.87 MB (27.79 million rows/s., 491.95 MB/s.)
Peak memory usage: 17.89 MiB.

NB-R911LJCT.o3.ru :)
```

3. Объединить каждую строку из таблицы “Фильмы” с каждой строкой из таблицы “Жанры”

```
NB-R911LJCT.o3.ru :) select * from imdb.movies m full join imdb.genres g on m.id = g.movie_id limit 15;

SELECT *
FROM imdb.movies AS m
FULL OUTER JOIN imdb.genres AS g ON m.id = g.movie_id
LIMIT 15

Query id: 7c173ab1-8d06-43fa-93b1-5c56a2d98e1a

    ┌─────id─┬─name──────────────────┬─year─┬─rank─┬─movie_id─┬─genre─────┐
 1. │ 250742 │ Patriot, The          │ 1917 │    0 │   250742 │ War       │
 2. │ 250743 │ Patriot, The          │ 1928 │  6.6 │   250743 │ Drama     │
 3. │ 250744 │ Patriot, The          │ 1986 │  3.1 │   250744 │ Action    │
 4. │ 250745 │ Patriot, The (1998/I) │ 1998 │  4.1 │   250745 │ Action    │
 5. │ 250745 │ Patriot, The (1998/I) │ 1998 │  4.1 │   250745 │ Thriller  │
 6. │ 250746 │ Patriot, The          │ 2000 │  6.8 │   250746 │ Action    │
 7. │ 250746 │ Patriot, The          │ 2000 │  6.8 │   250746 │ Drama     │
 8. │ 250746 │ Patriot, The          │ 2000 │  6.8 │   250746 │ War       │
 9. │ 250748 │ Patriote, Le          │ 1938 │    0 │   250748 │ Drama     │
10. │ 250750 │ Patrioten             │ 2005 │    0 │   250750 │ Drama     │
11. │ 250751 │ Patrioterne           │ 1997 │    0 │   250751 │ Documentary │
12. │ 250752 │ Patriotes, Les        │ 1994 │  6.8 │   250752 │ Thriller  │
13. │ 250753 │ Patriotic             │ 2000 │    0 │   250753 │ Short     │
14. │ 250756 │ Patriotic Pooches     │ 1942 │    0 │   250756 │ Animation │
15. │ 250756 │ Patriotic Pooches     │ 1942 │    0 │   250756 │ Short     │
    └────────┴───────────────────────┴──────┴──────┴──────────┴───────────┘

15 rows in set. Elapsed: 0.080 sec. Processed 783.39 thousand rows, 23.06 MB (9.76 million rows/s., 287.31 MB/s.)
Peak memory usage: 92.76 MiB.
```

4. Найти жанры для каждого фильма, НЕ используя INNER JOIN

Примечание: Сомнения, так как запрос с LEFT уже использовался. Но если по «букве» задания это же не INNER?

```
NB-R911LJCT.o3.ru :) select m.name, groupArray(g.genre) from imdb.movies m left join imdb.genres g on m.id = g.movie_id  group by m.name limit 15;

SELECT
    m.name,
    groupArray(g.genre)
FROM imdb.movies AS m
LEFT JOIN imdb.genres AS g ON m.id = g.movie_id
GROUP BY m.name
LIMIT 15

Query id: 7eae2d36-8067-4598-a594-50a9359e27f6

    ┌─name─────────┬─groupArray(genre)─────────────────────┐
 1. │ Ogostos      │ ['Drama']                             │
 2. │ Gumiho       │ ['Fantasy','Romance']                 │
 3. │ Aphone       │ ['Short']                             │
 4. │ Cascadia     │ ['Crime']                             │
 5. │ Mit Herz     │ ['']                                  │
 6. │ Gengas       │ ['Short']                             │
 7. │ Nehir        │ ['Adventure','Crime','Thriller']      │
 8. │ Killer       │ ['Short','Drama','Thriller']          │
 9. │ 23 Hours     │ ['Action','Thriller']                 │
10. │ Serrana      │ ['Musical','Short']                   │
11. │ Galate       │ ['Short']                             │
12. │ Exchange     │ ['Short']                             │
13. │ Chitthi      │ ['','']                               │
14. │ Ola, La      │ ['']                                  │
15. │ Idhayam      │ ['']                                  │
    └──────────────┴───────────────────────────────────────┘

15 rows in set. Elapsed: 0.082 sec. Processed 783.39 thousand rows, 19.95 MB (9.55 million rows/s., 243.11 MB/s.)
Peak memory usage: 147.38 MiB.
```

5. Найти всех актеров и актрис, снявшихся в фильме в 2023 году

Примечание: Нет ни одного фильма, который вышел в 2023 году. Поэтому для примера взял 2002

```
NB-R911LJCT.o3.ru :) select first_name, last_name from imdb.actors a join  (select actor_id from imdb.roles where movie_id in (select id from imdb.movies where year
 = 2023)) m_id on a.id = m_id.actor_id limit 15;

SELECT
    first_name,
    last_name
FROM imdb.actors AS a
INNER JOIN
(
    SELECT actor_id
    FROM imdb.roles
    WHERE movie_id IN (
        SELECT id
        FROM imdb.movies
        WHERE year = 2023
    )
) AS m_id ON a.id = m_id.actor_id
LIMIT 15

Query id: 8e262a8c-3ed9-46a3-be96-484984d1d330

Ok.

0 rows in set. Elapsed: 0.014 sec. Processed 806.06 thousand rows, 16.24 MB (58.12 million rows/s., 1.17 GB/s.)
Peak memory usage: 12.36 MiB.
```

```
NB-R911LJCT.o3.ru :) select first_name, last_name from imdb.actors a join  (select actor_id from imdb.roles where movie_id in (select id from imdb.movies where year
 = 2002)) m_id on a.id = m_id.actor_id limit 15;

SELECT
    first_name,
    WHERE movie_id IN (
        SELECT id
        FROM imdb.movies
        WHERE year = 2002
    )
) AS m_id ON a.id = m_id.actor_id
LIMIT 15

Query id: 3cdbd1e5-7442-47e1-aced-80b6f52d3c39

    ┌─first_name──┬─last_name─┐
 1. │ Oswald      │ 'Ô'       │
 2. │ Oswald      │ 'Ô'       │
 3. │ Oswald      │ 'Ô'       │
 4. │ Rappin'     │ 4-Tay     │
 5. │ Mick        │ 5         │
 6. │ Ott         │ Aardam    │
 7. │ Hank        │ Aaron     │
 8. │ Travis      │ Aaron     │
 9. │ Sjors       │ Aarts     │
10. │ Gali        │ Abajdulov │
11. │ Paul Vito   │ Abato     │
12. │ Alex        │ Abbad     │
13. │ David (VII) │ Abbott    │
14. │ Lousnak     │ Abdalian  │
15. │ Tarik       │ Abdel-Monem │
    └─────────────┴───────────┘
```

6. Запросить все фильмы, у которых нет жанра, через ANTI JOIN

```
NB-R911LJCT.o3.ru :) select name from imdb.movies m left anti join imdb.genres g on m.id = g.movie_id limit 15;

SELECT name
FROM imdb.movies AS m
ANTI LEFT JOIN imdb.genres AS g ON m.id = g.movie_id
LIMIT 15

Query id: 184e6ad8-f4f0-4e06-9cb0-d86787745f42

    ┌─name──────────────────────────┐
 1. │ Botschaft                     │
 2. │ Botte i farten                │
 3. │ Bottom Land                   │
 4. │ Botn de fuego, El             │
 5. │ Botts sajt krsre              │
 6. │ Bouffon, Le                   │
 7. │ Bought & Sold                 │
 8. │ Bouka                         │
 9. │ Bouncer, The                  │
10. │ Bound and Determined          │
11. │ Bound for Home                │
12. │ Bouquetire des innocents, La  │
13. │ Bourrier, Le                  │
14. │ Boustaguy, al-                │
15. │ Bout d'essai                  │
    └───────────────────────────────┘

15 rows in set. Elapsed: 0.028 sec. Processed 783.39 thousand rows, 13.87 MB (27.79 million rows/s., 491.95 MB/s.)
Peak memory usage: 17.89 MiB.

NB-R911LJCT.o3.ru :)
```