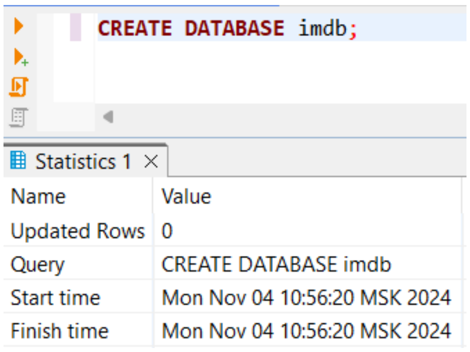
1. Создать БД и таблицы

```
CREATE DATABASE imdb;
```

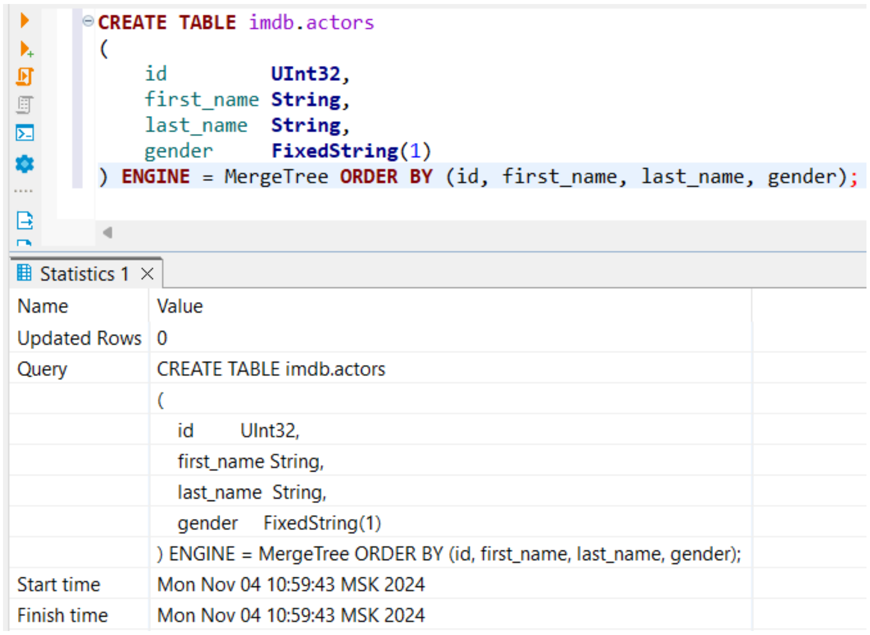


```
CREATE DATABASE imdb;
```

Statistics 1

Name	Value
Updated Rows	0
Query	CREATE DATABASE imdb
Start time	Mon Nov 04 10:56:20 MSK 2024
Finish time	Mon Nov 04 10:56:20 MSK 2024

```
CREATE TABLE imdb.actors
(
    id         UInt32,
    first_name String,
    last_name  String,
    gender     FixedString(1)
) ENGINE = MergeTree ORDER BY (id, first_name, last_name, gender);
```

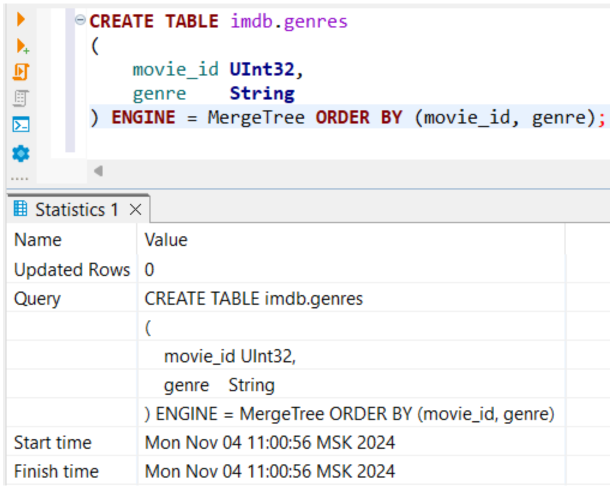


```
CREATE TABLE imdb.actors
(
    id         UInt32,
    first_name String,
    last_name  String,
    gender     FixedString(1)
) ENGINE = MergeTree ORDER BY (id, first_name, last_name, gender);
```

Statistics 1

Name	Value
Updated Rows	0
Query	CREATE TABLE imdb.actors
	(
	id UInt32,
	first_name String,
	last_name String,
	gender FixedString(1)
	) ENGINE = MergeTree ORDER BY (id, first_name, last_name, gender);
Start time	Mon Nov 04 10:59:43 MSK 2024
Finish time	Mon Nov 04 10:59:43 MSK 2024

```
CREATE TABLE imdb.genres
(
    movie_id UInt32,
    genre    String
) ENGINE = MergeTree ORDER BY (movie_id, genre);
```

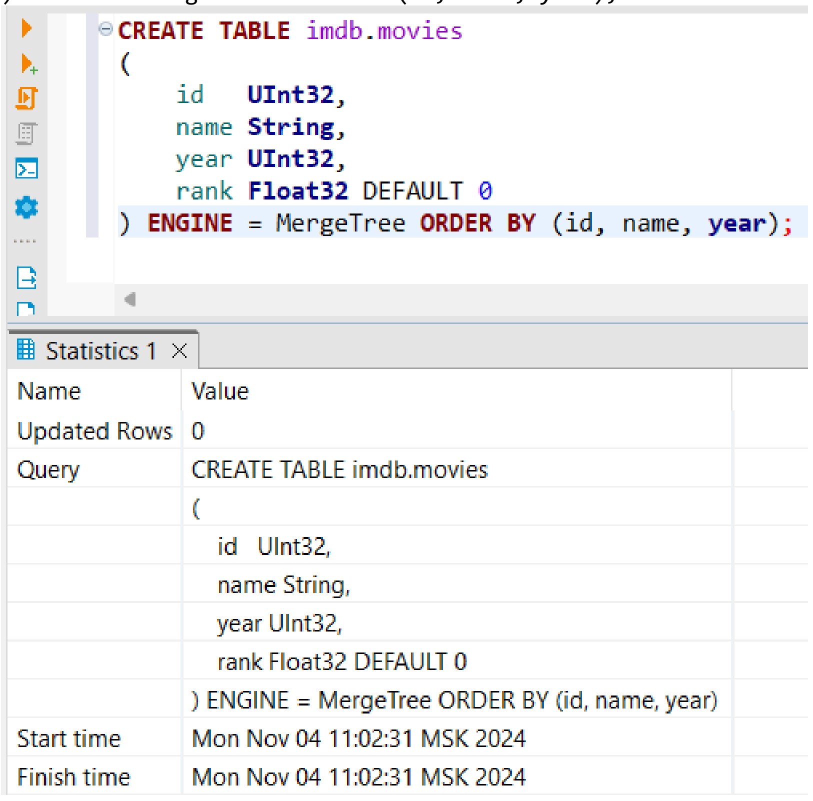


```
CREATE TABLE imdb.genres
(
    movie_id UInt32,
    genre    String
) ENGINE = MergeTree ORDER BY (movie_id, genre);
```

Statistics 1

Name	Value
Updated Rows	0
Query	CREATE TABLE imdb.genres
	(
	movie_id UInt32,
	genre String
	) ENGINE = MergeTree ORDER BY (movie_id, genre)
Start time	Mon Nov 04 11:00:56 MSK 2024
Finish time	Mon Nov 04 11:00:56 MSK 2024

```
CREATE TABLE imdb.movies
(
    id   UInt32,
    name String,
    year UInt32,
    rank Float32 DEFAULT 0
) ENGINE = MergeTree ORDER BY (id, name, year);
```

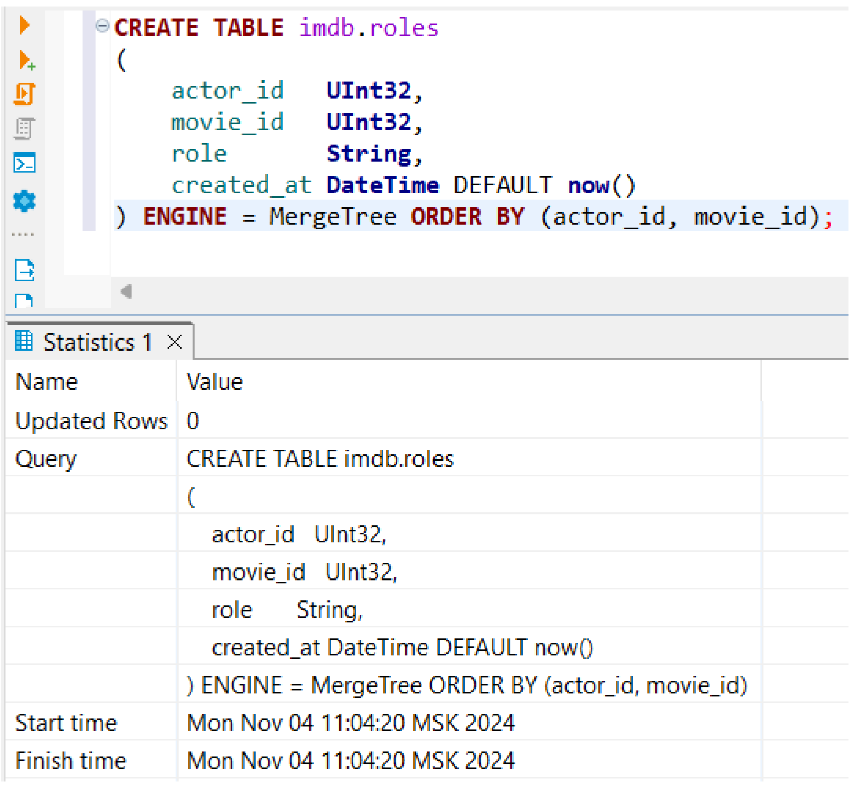


```
CREATE TABLE imdb.movies
(
    id   UInt32,
    name String,
    year UInt32,
    rank Float32 DEFAULT 0
) ENGINE = MergeTree ORDER BY (id, name, year);
```

Statistics 1

Name	Value
Updated Rows	0
Query	CREATE TABLE imdb.movies
	(
	id UInt32,
	name String,
	year UInt32,
	rank Float32 DEFAULT 0
	) ENGINE = MergeTree ORDER BY (id, name, year)
Start time	Mon Nov 04 11:02:31 MSK 2024
Finish time	Mon Nov 04 11:02:31 MSK 2024

```
CREATE TABLE imdb.roles
(
    actor_id   UInt32,
    movie_id   UInt32,
    role       String,
    created_at DateTime DEFAULT now()
) ENGINE = MergeTree ORDER BY (actor_id, movie_id);
```



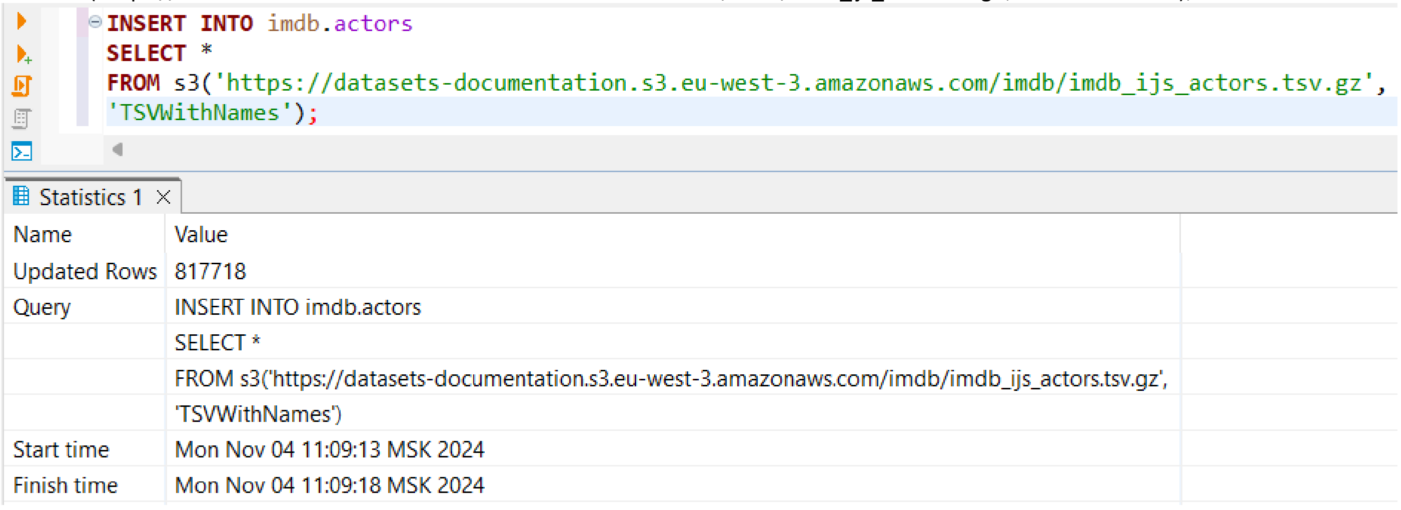
```
CREATE TABLE imdb.roles
(
    actor_id   UInt32,
    movie_id   UInt32,
    role       String,
    created_at DateTime DEFAULT now()
) ENGINE = MergeTree ORDER BY (actor_id, movie_id);
```

Statistics 1

Name	Value
Updated Rows	0
Query	CREATE TABLE imdb.roles
	(
	actor_id UInt32,
	movie_id UInt32,
	role String,
	created_at DateTime DEFAULT now()
	) ENGINE = MergeTree ORDER BY (actor_id, movie_id)
Start time	Mon Nov 04 11:04:20 MSK 2024
Finish time	Mon Nov 04 11:04:20 MSK 2024

2. <u>Вставить тестовые данные, используя функцию S3</u>

```
INSERT INTO imdb.actors
SELECT *
FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_actors.tsv.gz', 'TSVWithNames');
```

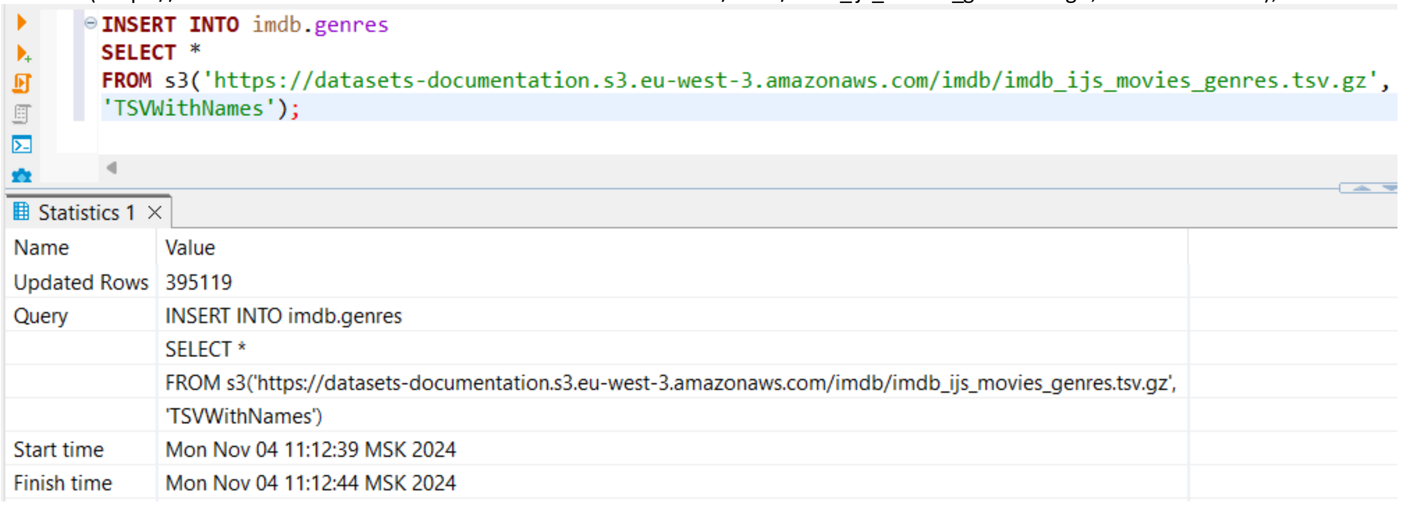


```
INSERT INTO imdb.actors
SELECT *
FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_actors.tsv.gz',
'TSVWithNames');
```

Statistics 1

Name	Value
Updated Rows	817718
Query	INSERT INTO imdb.actors
	SELECT *
	FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_actors.tsv.gz',
	'TSVWithNames')
Start time	Mon Nov 04 11:09:13 MSK 2024
Finish time	Mon Nov 04 11:09:18 MSK 2024

```
INSERT INTO imdb.genres
SELECT *
FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_movies_genres.tsv.gz', 'TSVWithNames');
```

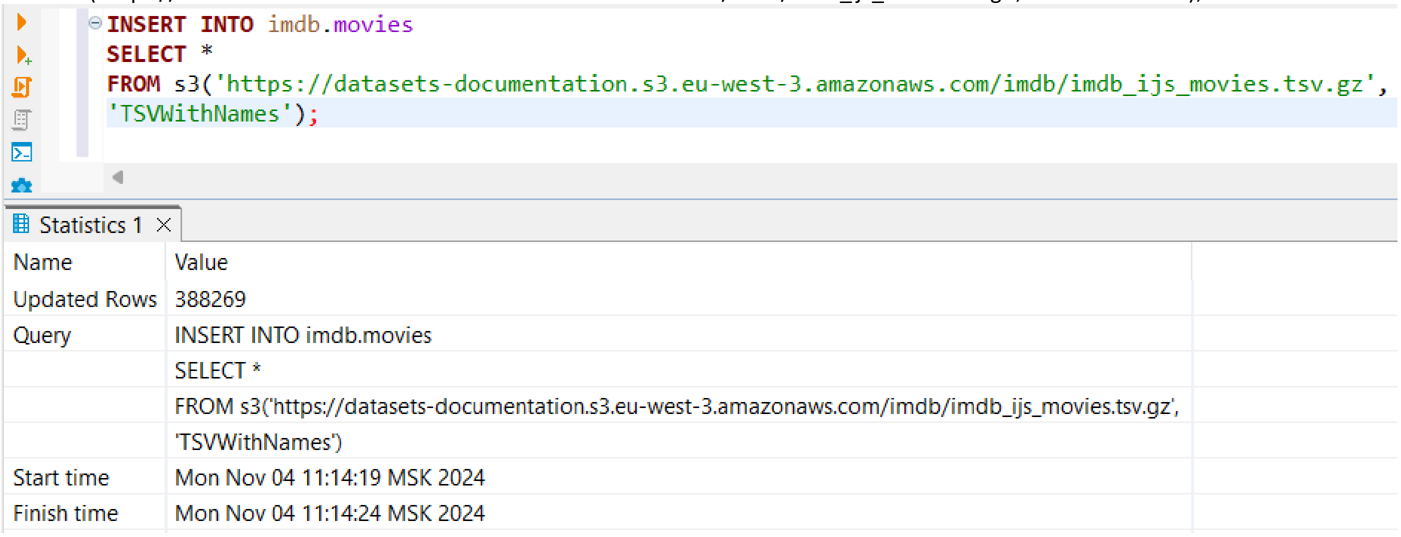


```
INSERT INTO imdb.genres
SELECT *
FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_movies_genres.tsv.gz',
'TSVWithNames');
```

Statistics 1

Name	Value
Updated Rows	395119
Query	INSERT INTO imdb.genres
	SELECT *
	FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_movies_genres.tsv.gz',
	'TSVWithNames')
Start time	Mon Nov 04 11:12:39 MSK 2024
Finish time	Mon Nov 04 11:12:44 MSK 2024

```
INSERT INTO imdb.movies
SELECT *
FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_movies.tsv.gz', 'TSVWithNames');
```

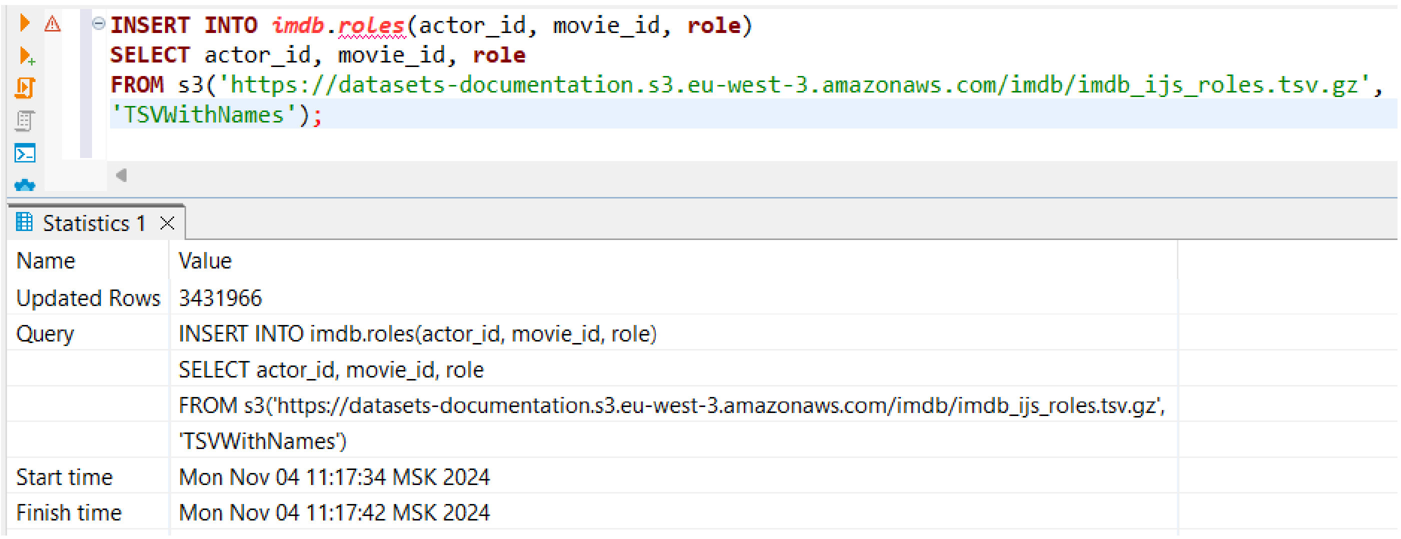


```
INSERT INTO imdb.movies
SELECT *
FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_movies.tsv.gz',
'TSVWithNames');
```

Statistics 1

Name	Value
Updated Rows	388269
Query	INSERT INTO imdb.movies
	SELECT *
	FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_movies.tsv.gz',
	'TSVWithNames')
Start time	Mon Nov 04 11:14:19 MSK 2024
Finish time	Mon Nov 04 11:14:24 MSK 2024

```
INSERT INTO imdb.roles(actor_id, movie_id, role)
SELECT actor_id, movie_id, role
FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_roles.tsv.gz',
'TSVWithNames');
```



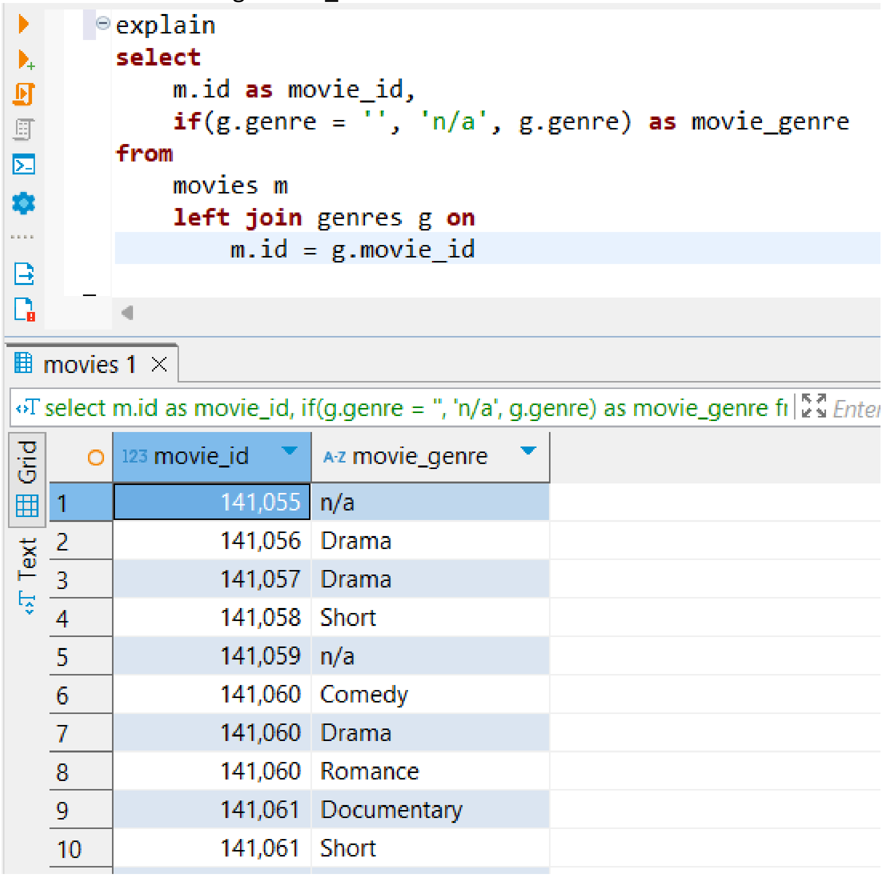
```
INSERT INTO imdb.roles(actor_id, movie_id, role)
SELECT actor_id, movie_id, role
FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_roles.tsv.gz',
'TSVWithNames');
```

Statistics 1

Name	Value
Updated Rows	3431966
Query	INSERT INTO imdb.roles(actor_id, movie_id, role)
	SELECT actor_id, movie_id, role
	FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_roles.tsv.gz',
	'TSVWithNames')
Start time	Mon Nov 04 11:17:34 MSK 2024
Finish time	Mon Nov 04 11:17:42 MSK 2024

3. **Используя изученные материалы, построить запросы, отвечающие на следующие задачи:**
   - **Найти жанры для каждого фильма (не только для которых жанры определены)**

```
select
    m.id as movie_id,
    if(g.genre = '', 'n/a', g.genre) as movie_genre
from
    movies m
    left join genres g on
           m.id = g.movie_id
```



```
explain
select
    m.id as movie_id,
    if(g.genre = '', 'n/a', g.genre) as movie_genre
from
    movies m
    left join genres g on
        m.id = g.movie_id
```

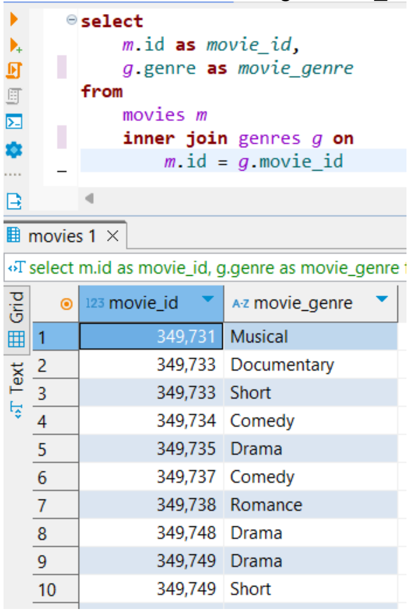
movies 1

select m.id as movie_id, if(g.genre = '', 'n/a', g.genre) as movie_genre fi

	movie_id	movie_genre
1	141,055	n/a
2	141,056	Drama
3	141,057	Drama
4	141,058	Short
5	141,059	n/a
6	141,060	Comedy
7	141,060	Drama
8	141,060	Romance
9	141,061	Documentary
10	141,061	Short

- **Найти жанры для каждого фильма (только для которых жанры определены)**

```
select
	m.id as movie_id,
	g.genre as movie_genre
from
	movies m
	inner join genres g on
		m.id = g.movie_id
```



```
select
    m.id as movie_id,
    g.genre as movie_genre
from
    movies m
    inner join genres g on
        m.id = g.movie_id
```

movies 1

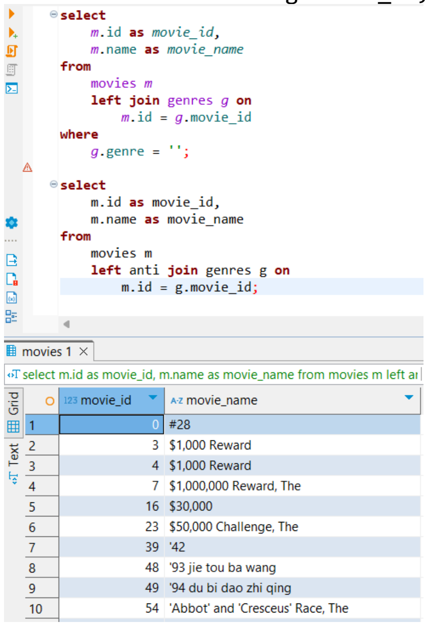
select m.id as movie_id, g.genre as movie_genre

```
   movie_id  movie_genre
1   349,731  Musical
2   349,733  Documentary
3   349,733  Short
4   349,734  Comedy
5   349,735  Drama
6   349,737  Comedy
7   349,738  Romance
8   349,748  Drama
9   349,749  Drama
10  349,749  Short
```

- **Запросить все фильмы, у которых нет жанра**

```
select
	m.id as movie_id,
	m.name as movie_name
from
	movies m
	left join genres g on
		m.id = g.movie_id
where
	g.genre = '';

select
	m.id as movie_id,
	m.name as movie_name
from
	movies m
	left anti join genres g on
		m.id = g.movie_id;
```



```
select
    m.id as movie_id,
    m.name as movie_name
from
    movies m
    left join genres g on
        m.id = g.movie_id
where
    g.genre = '';

select
    m.id as movie_id,
    m.name as movie_name
from
    movies m
    left anti join genres g on
        m.id = g.movie_id;
```

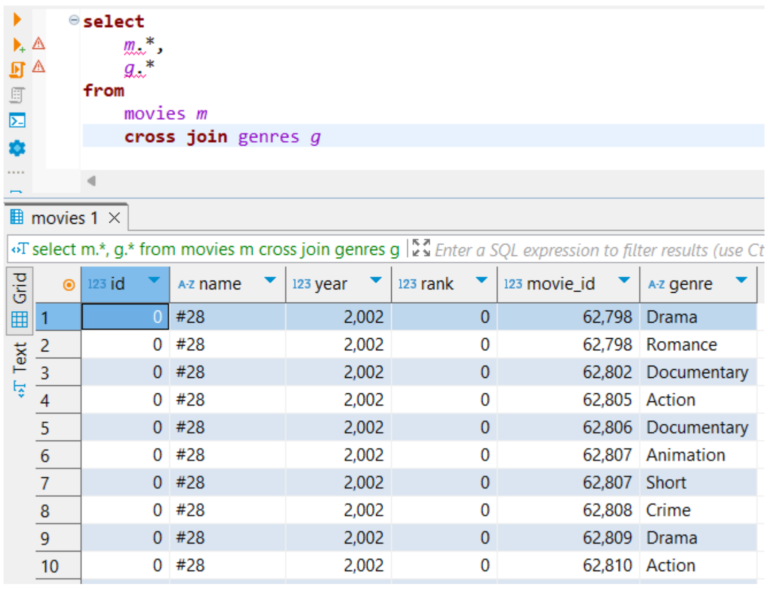
movies 1

select m.id as movie_id, m.name as movie_name from movies m left ar

```
   movie_id  movie_name
1         0  #28
2         3  $1,000 Reward
3         4  $1,000 Reward
4         7  $1,000,000 Reward, The
5        16  $30,000
6        23  $50,000 Challenge, The
7        39  '42
8        48  '93 jie tou ba wang
9        49  '94 du bi dao zhi qing
10       54  'Abbot' and 'Cresceus' Race, The
```

- **Объединить каждую строку из таблицы “Фильмы” с каждой строкой из таблицы “Жанры”**

```
select
        m.*,
        g.*
from
        movies m
        cross join genres g
```



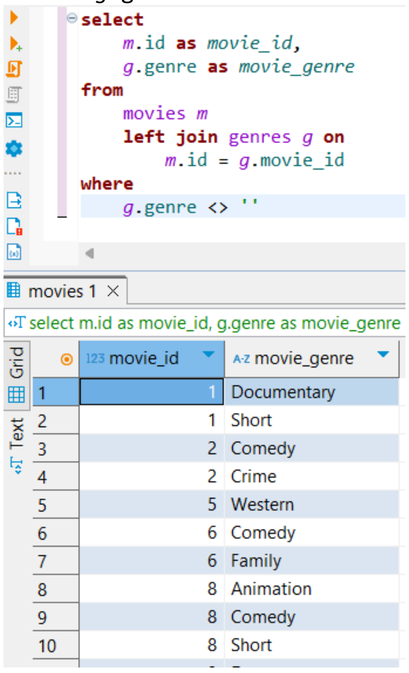
```
select
    m.*,
    g.*
from
    movies m
    cross join genres g
```

movies 1
select m.*, g.* from movies m cross join genres g
Enter a SQL expression to filter results (use Ct
Grid
Text

	id	name	year	rank	movie_id	genre
1	0	#28	2,002	0	62,798	Drama
2	0	#28	2,002	0	62,798	Romance
3	0	#28	2,002	0	62,802	Documentary
4	0	#28	2,002	0	62,805	Action
5	0	#28	2,002	0	62,806	Documentary
6	0	#28	2,002	0	62,807	Animation
7	0	#28	2,002	0	62,807	Short
8	0	#28	2,002	0	62,808	Crime
9	0	#28	2,002	0	62,809	Drama
10	0	#28	2,002	0	62,810	Action

- **Найти жанры для каждого фильма, НЕ используя INNER JOIN**
не только для которых жанры определены – уже написан
только для которых жанры определены

```
select
        m.id as movie_id,
        g.genre as movie_genre
from
        movies m
        left join genres g on
                m.id = g.movie_id
where
        g.genre <> ''
```



```
select
    m.id as movie_id,
    g.genre as movie_genre
from
    movies m
    left join genres g on
        m.id = g.movie_id
where
    g.genre <> ''
```

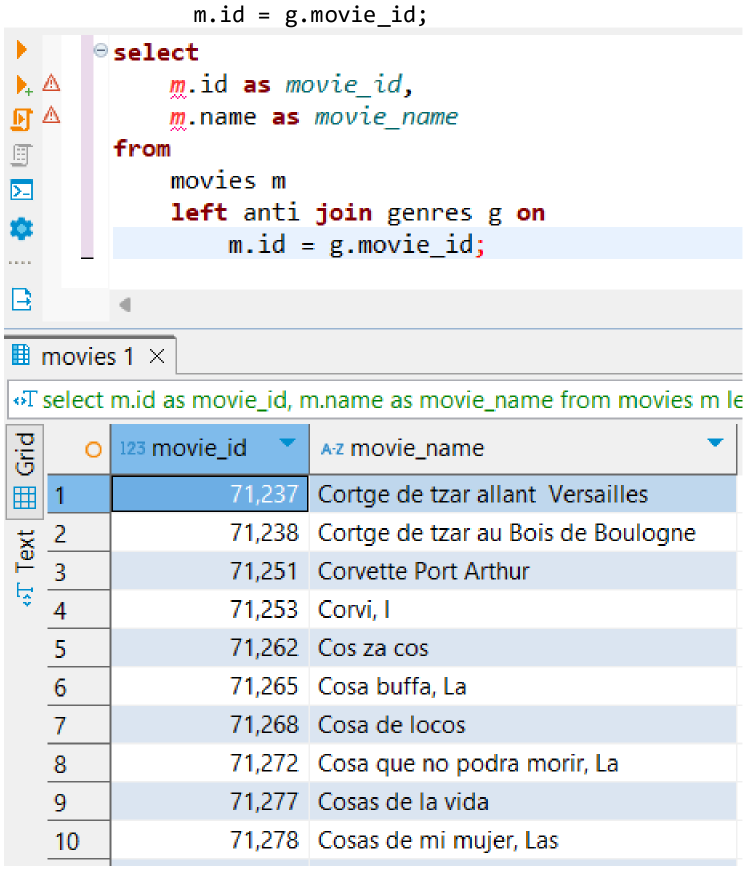
	movie_id	movie_genre
1	1	Documentary
2	1	Short
3	2	Comedy
4	2	Crime
5	5	Western
6	6	Comedy
7	6	Family
8	8	Animation
9	8	Comedy
10	8	Short

- **Найти всех актеров и актрис, снявшихся в фильме в 2023 году**

```
SELECT
    a.id,
    a.first_name,
    a.last_name,
    a.gender
FROM
    imdb.actors as a
    left semi join (
       select
              actor_id
       from
              imdb.roles as r
       where
              movie_id in (select id from imdb.movies as m where m.year = 2023)
    ) as b on b.actor_id = a.id
```

- **Запросить все фильмы, у которых нет жанра, через ANTI JOIN**

```
select
       m.id as movie_id,
       m.name as movie_name
from
       movies m
       left anti join genres g on
              m.id = g.movie_id;
```



```
select
    m.id as movie_id,
    m.name as movie_name
from
    movies m
    left anti join genres g on
        m.id = g.movie_id;
```

movies 1

select m.id as movie_id, m.name as movie_name from movies m le

	movie_id	movie_name
1	71,237	Cortge de tzar allant Versailles
2	71,238	Cortge de tzar au Bois de Boulogne
3	71,251	Corvette Port Arthur
4	71,253	Corvi, I
5	71,262	Cos za cos
6	71,265	Cosa buffa, La
7	71,268	Cosa de locos
8	71,272	Cosa que no podra morir, La
9	71,277	Cosas de la vida
10	71,278	Cosas de mi mujer, Las