Домашнее задание Джоины и агрегации

- Создание структуры:

```
CREATE DATABASE imdb;


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.genres
(
    movie_id UInt32,
    genre    String
) ENGINE = MergeTree ORDER BY (movie_id, genre);


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


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);

- Наполнение данными:

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.genres

SELECT *

FROM s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/imdb/imdb_ijs_movies_genres.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');

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');

- Запросы:

--Найти жанры для каждого фильма

SELECT

m.id **AS** *movie_id*,

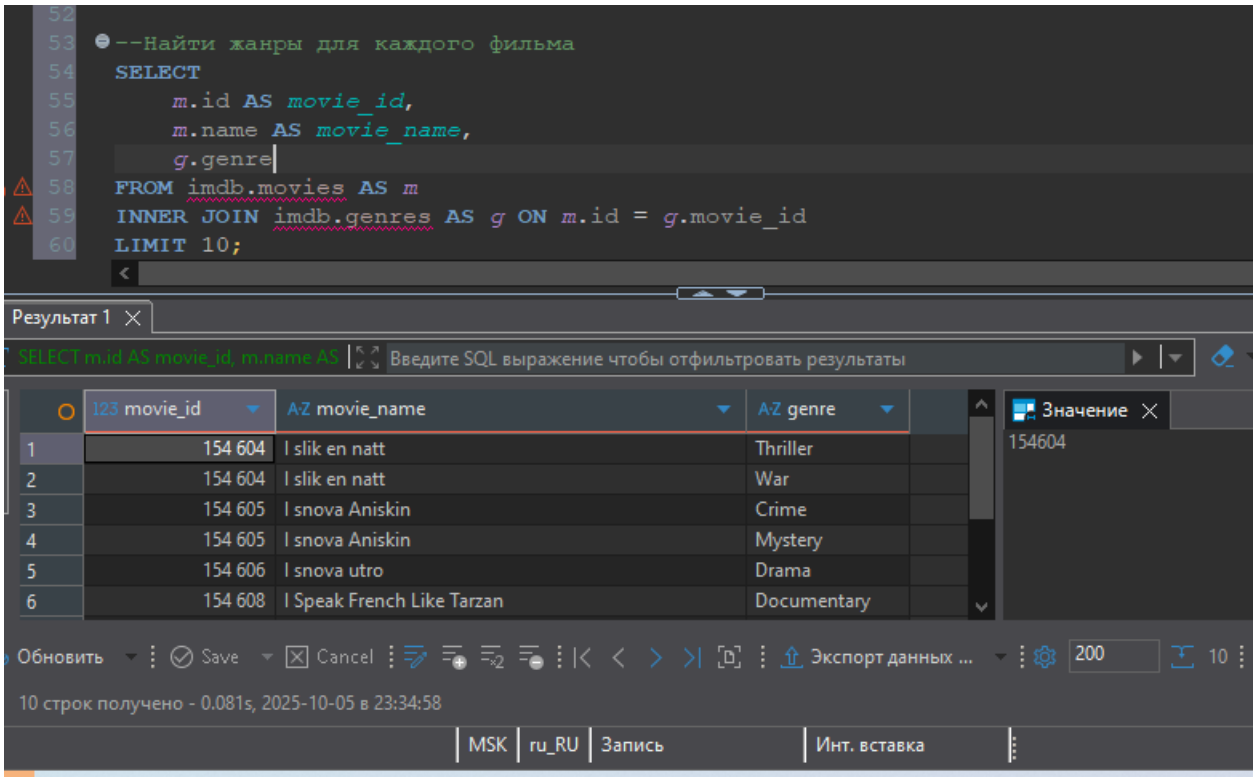
m.name **AS** *movie_name*,

g.genre

FROM imdb.movies **AS** *m*

INNER JOIN imdb.genres **AS** *g* **ON** *m*.id = *g*.movie_id

LIMIT 10;



--Запросить все фильмы, у которых нет жанра

SELECT

m.id,

m.name,

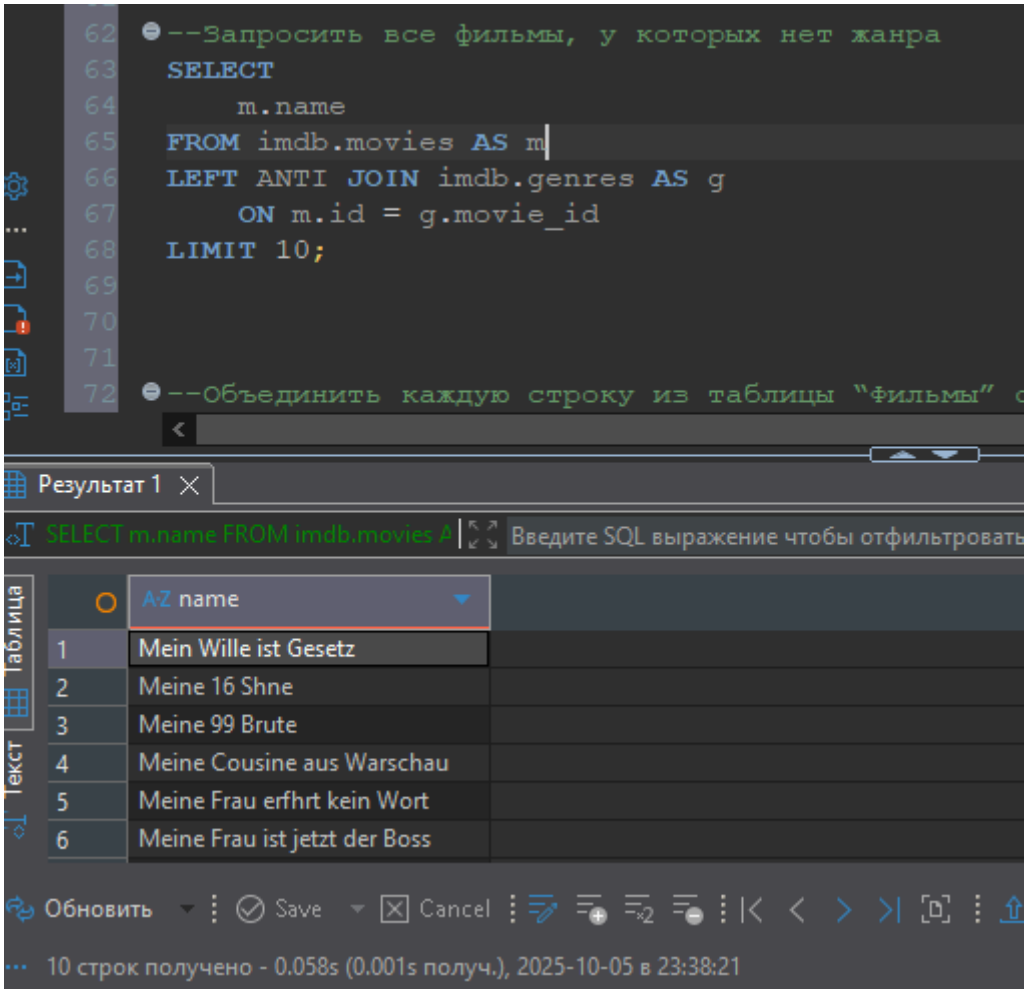
m.**year**

FROM imdb.movies **AS** *m*

LEFT JOIN imdb.genres **AS** *g* **ON** *m*.id = *g*.movie_id

WHERE *g*.movie_id **IS NULL**

LIMIT 10;



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

SELECT

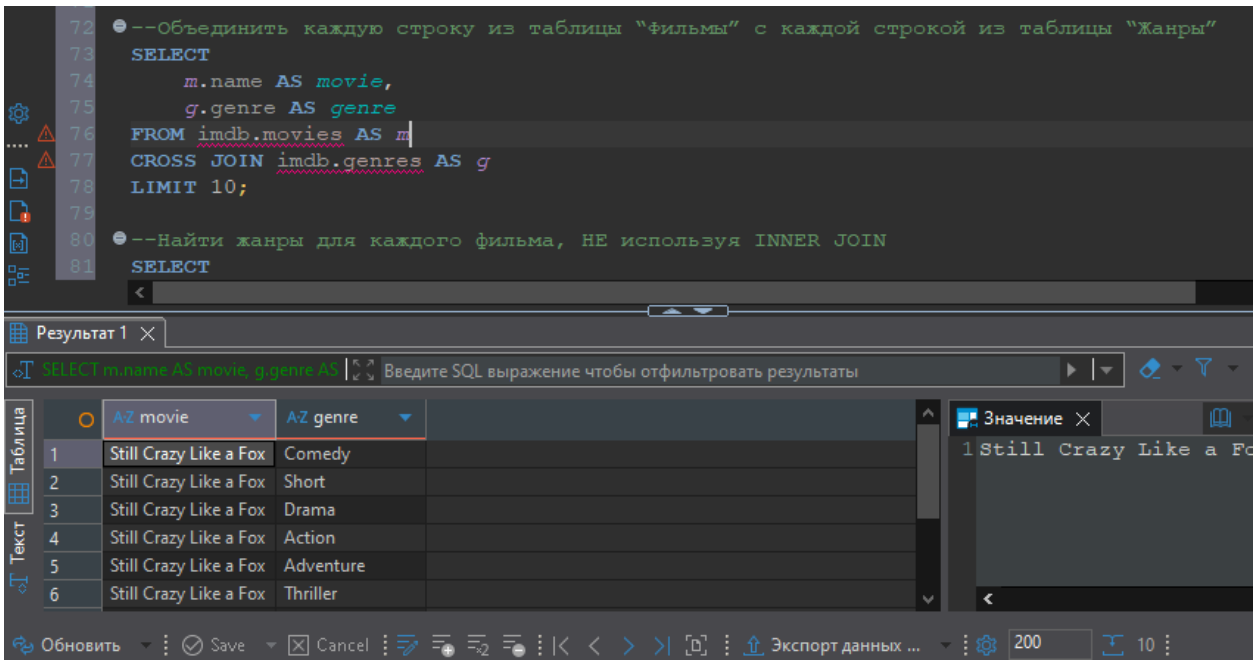
m.name **AS** *movie*,

g.genre **AS** *genre*

FROM imdb.movies **AS** *m*

CROSS JOIN imdb.genres **AS** *g*

LIMIT 10;



--Найти жанры для каждого фильма, НЕ используя INNER JOIN

SELECT

m.id,

m.name,

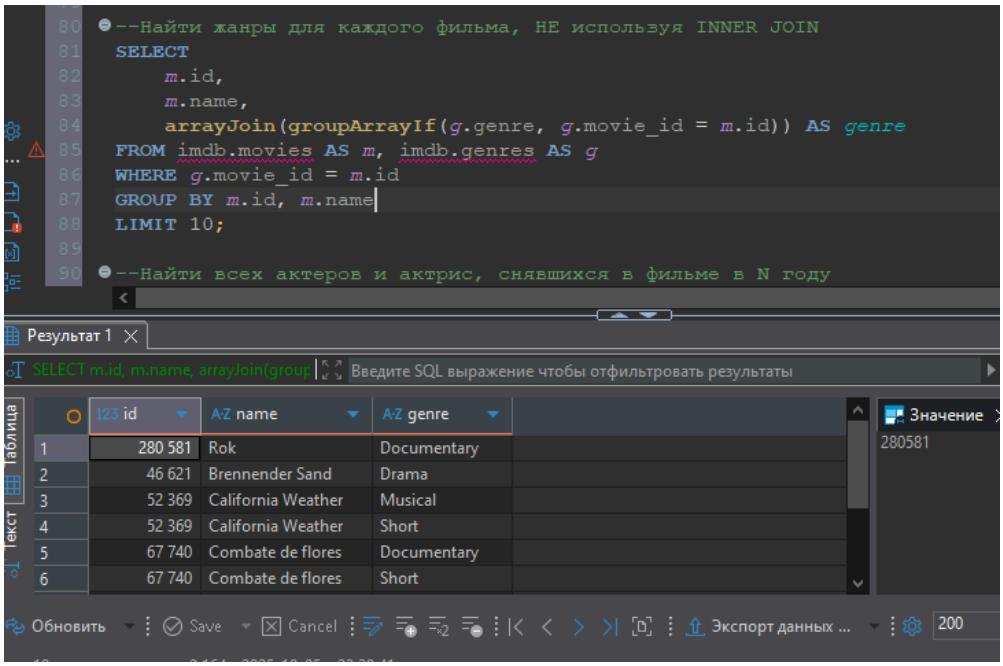
arrayJoin(**groupArrayIf**(*g*.genre, *g*.movie_id = *m*.id)) **AS** *genre*

FROM imdb.movies **AS** *m*, imdb.genres **AS** *g*

WHERE *g*.movie_id = *m*.id

GROUP BY *m*.id, *m*.name

LIMIT 10;



--Найти всех актеров и актрис, снявшихся в фильме в N году

SELECT

DISTINCT *a*.first_name,

a.last_name,

a.gender,

m.name **AS** *movie_name*,

m.**year**

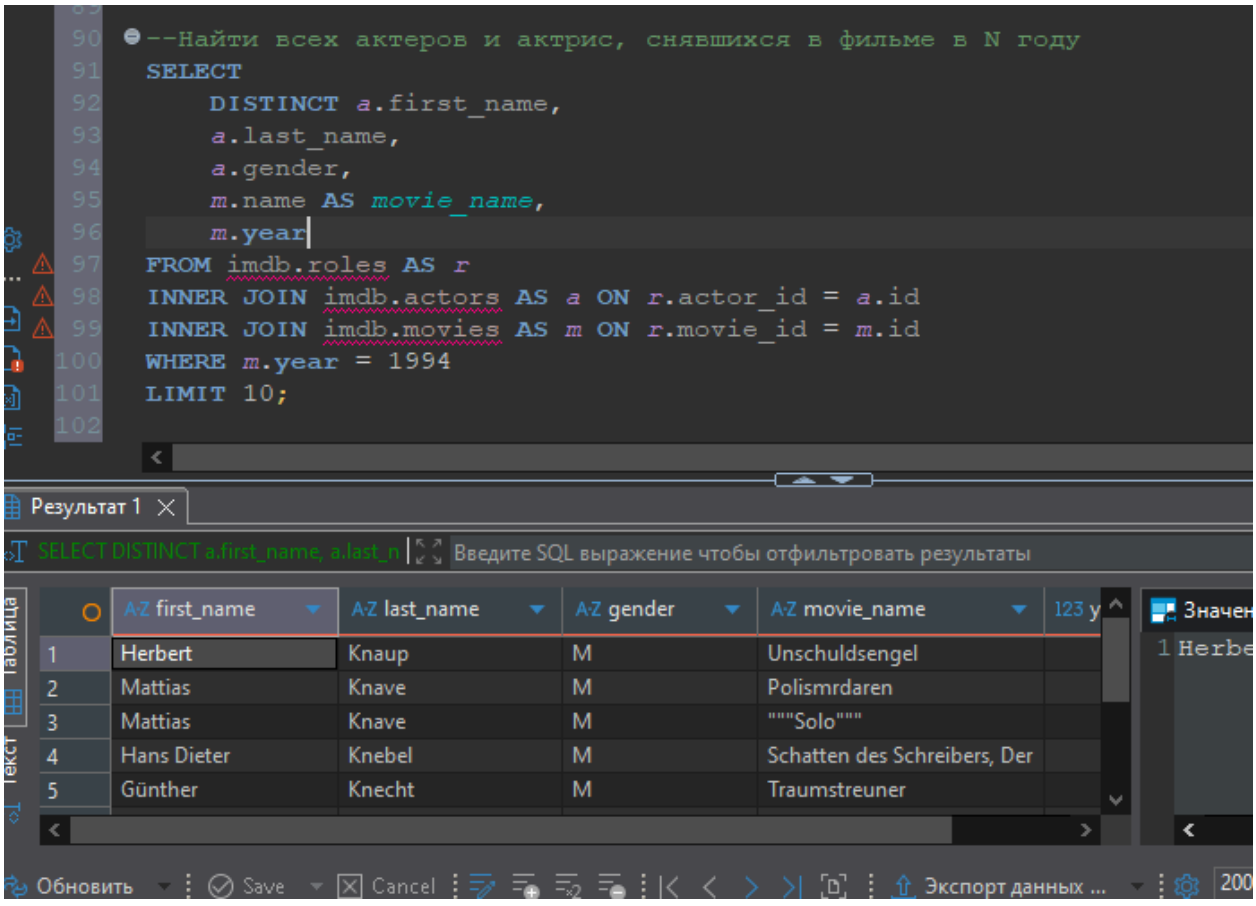
FROM imdb.roles **AS** *r*

INNER JOIN imdb.actors **AS** *a* **ON** *r*.actor_id = *a*.id

INNER JOIN imdb.movies **AS** *m* **ON** *r*.movie_id = *m*.id

WHERE *m*.**year** = 1994

LIMIT 10;



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

```
SELECT
    m.id,
    m.name,
    m.year
FROM imdb.movies AS m
LEFT ANTI JOIN imdb.genres AS g ON m.id = g.movie_id
LIMIT 10;
```