**Подоксенова К.М**

**Группа Java**

**Задание по SQL. Тема : сеть LIKE\_IT**

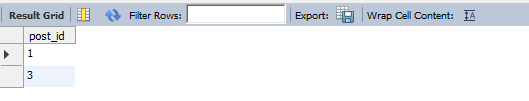
1. **2 запроса с различными WHERE условиями**
2. Запрос выбирает id прочитанных постов для определенного юзера по id. В данном случае – id = 1;

SELECT post\_id

FROM like\_it.readed\_posts

WHERE user\_id = 1;

Результат:



1. Выбрать все опубликованные категории, в заголовке которых участвует определенное слово. В данном случае – “все”;

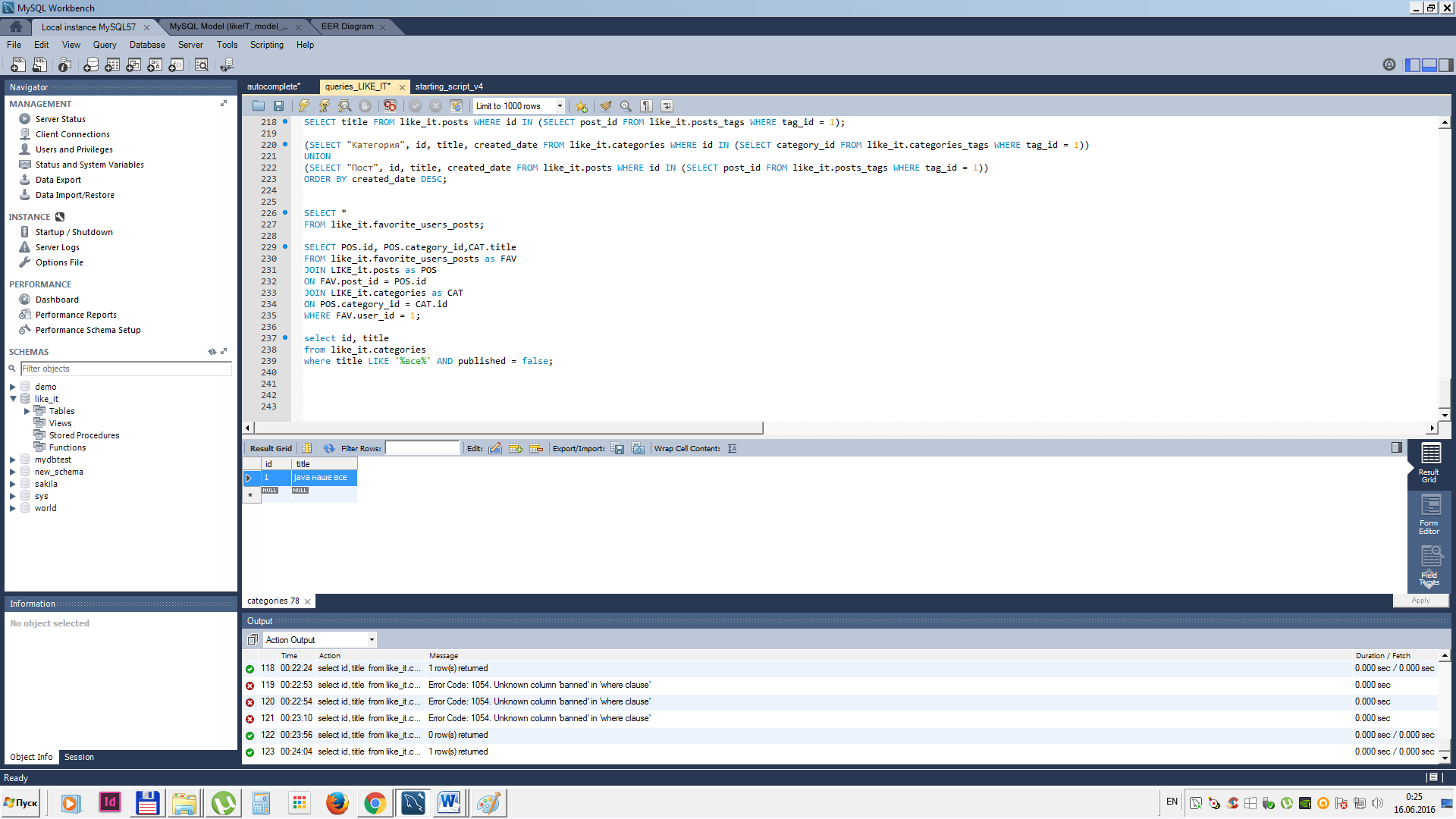
Применение: поиск по запросу пользователя.

select id, title

from like\_it.categories

where title LIKE '%все%' AND published = true;

Результат:



1. **2 запроса с применением встроенных функций**
2. Посчитать количество количество избранных постов в по каждой категории постов и вывести на информацию вместе с заголовком категории.

Применение: отображение на странице пользователя.

SELECT count(POS.id), POS.category\_id,CAT.title

FROM like\_it.favorite\_users\_posts as FAV

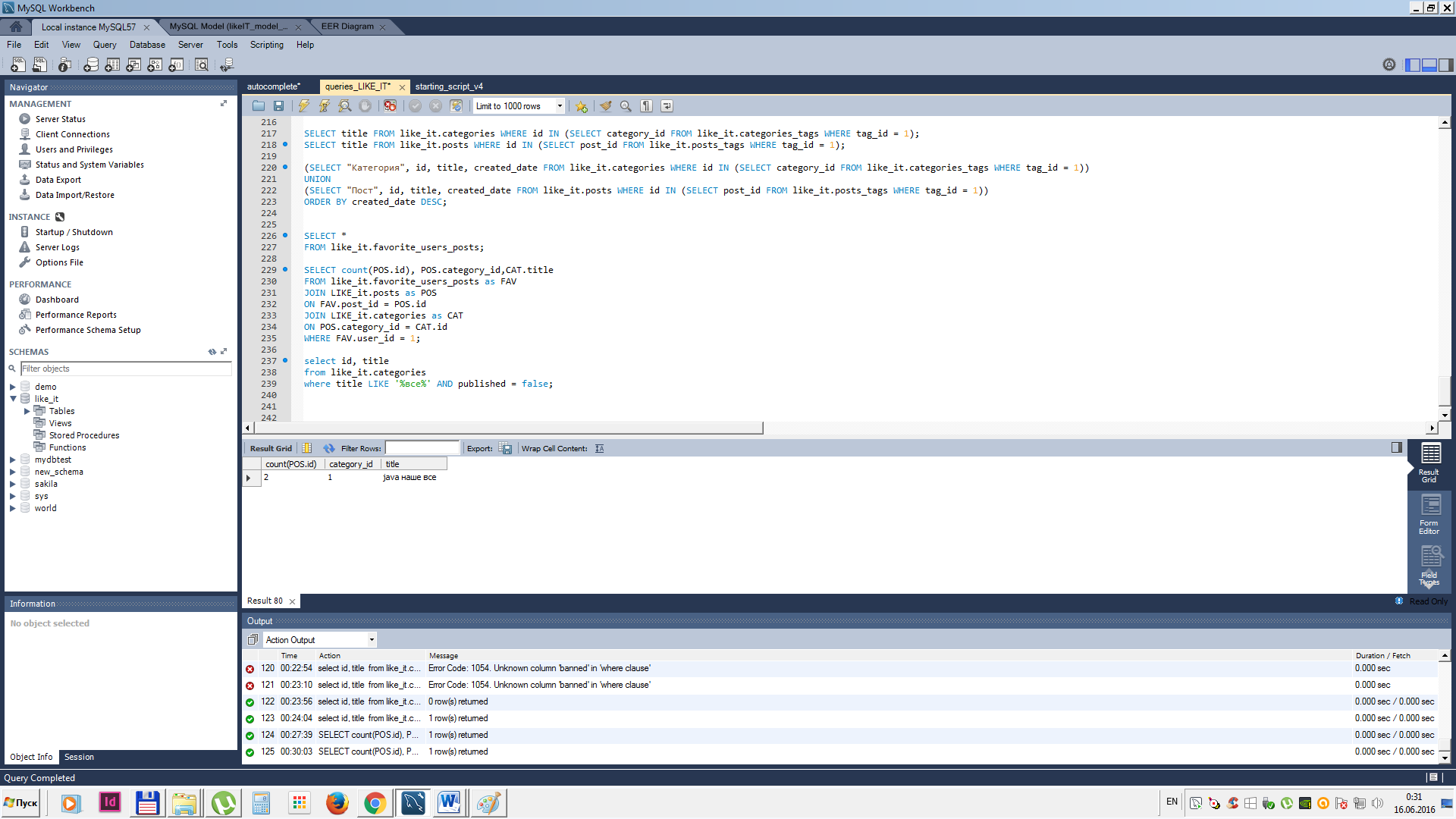
JOIN LIKE\_it.posts as POS

ON FAV.post\_id = POS.id

JOIN LIKE\_it.categories as CAT

ON POS.category\_id = CAT.id

WHERE FAV.user\_id = 1;



1. Посчитать количество просмотра определенного поста

Применение: информация отображается на странице поста

SELECT count(post\_id)

FROM like\_it.readed\_posts

WHERE post\_id = 3;

1. **Запросы на соединения таблиц(JOIN): INNER, LEFT, RIGHT, CROSS.**
2. Запрос выбирает всех юзеров и их почтовые адреса , добавивших один из данных тегов в избранное. В данном случае - "java" и "sql"

Потенциальная польза: создание рекомендаций для пользователей

SELECT US.id, US.email

FROM LIKE\_IT.users AS US

JOIN (SELECT TAG.name, TAG.id, FAV.user\_id

FROM LIKE\_IT.favorite\_user\_tags AS FAV

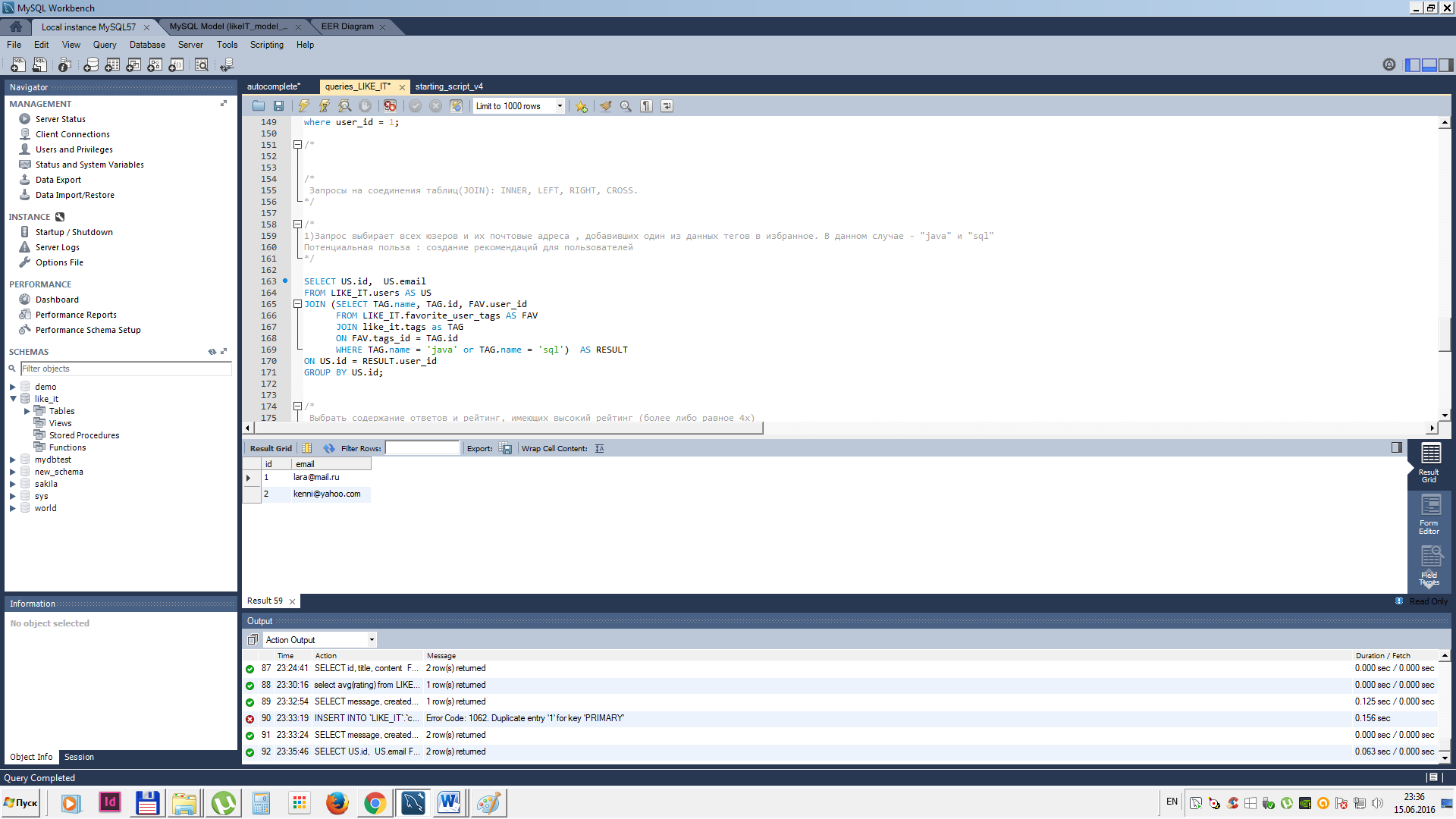
JOIN like\_it.tags as TAG

ON FAV.tags\_id = TAG.id

WHERE TAG.name = 'java' or TAG.name = 'sql') AS RESULT

ON US.id = RESULT.user\_id

GROUP BY US.id;



1. Запрос c GROUP BY и HAVING и агрегирующими функциями.
2. Отобразить id, тело сообщения и среднюю оценку за сообщение всех сообщений для определенного поста (в данном примере с id = 3) , у которых средняя оценка больше 3х баллов.

SELECT MSG.message, AVG(RATE.rating) as avRate

FROM like\_it.answers AS MSG

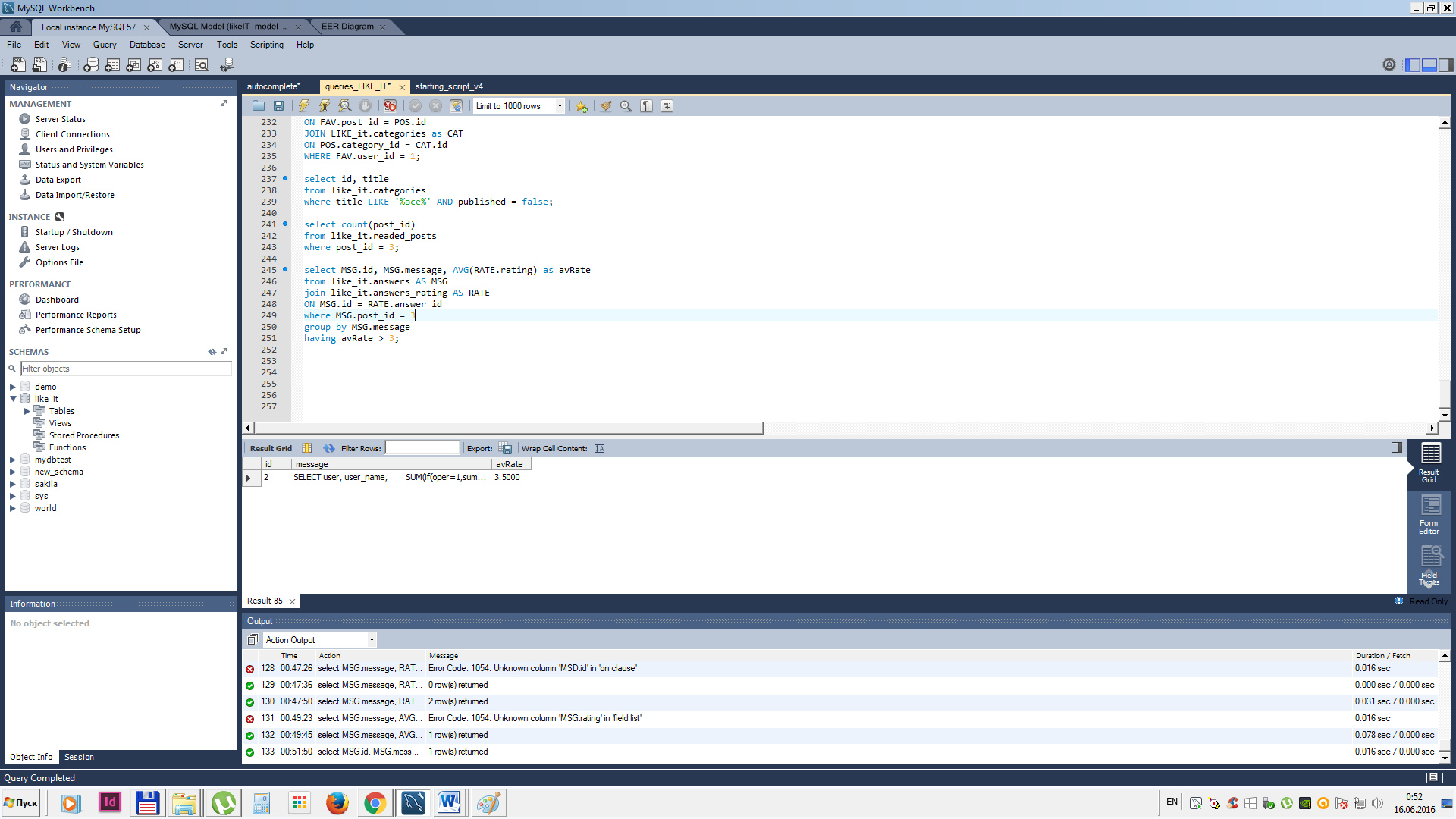
JOIN like\_it.answers\_rating AS RATE

ON MSG.id = RATE.answer\_id

WHERE MSG.post\_id = 3

GROUP BY MSG.message

HAVING avRate > 3;



1. **Запрос c UNION**
2. Выбрать все посты и категории, в котые связаны с определенным тегом (в данном случае – с id=1) . Отобразить их тип, id, заголовок и дату создания. Упорядочить по дате создания.

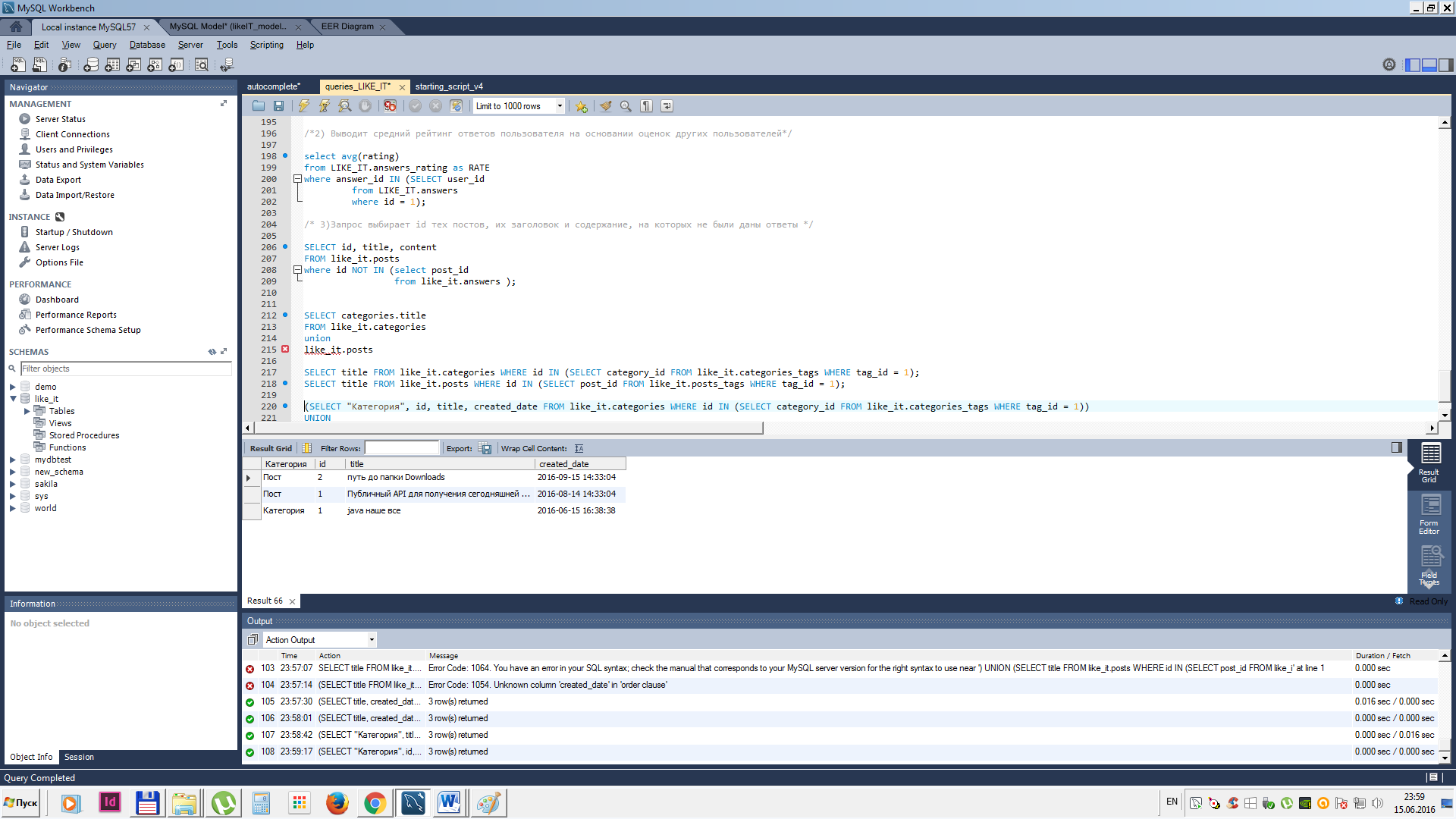
Применение: использование для рекомендаций.

(SELECT "Категория", id, title, created\_date FROM like\_it.categories WHERE id IN (SELECT category\_id FROM like\_it.categories\_tags WHERE tag\_id = 1))

UNION

(SELECT "Пост", id, title, created\_date FROM like\_it.posts WHERE id IN (SELECT post\_id FROM like\_it.posts\_tags WHERE tag\_id = 1))

ORDER BY created\_date DESC;



1. **3 запроса с подзапросами, в т.ч. коррелирующие(взаимосвязанные).**
2. Запрос выбирает id тех постов, их заголовок и содержание, на которых не были даны ответы.

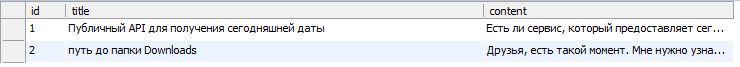
Применение : предложение юзерам ответить на данные посты.

SELECT id, title, content

FROM like\_it.posts

WHERE id NOT IN (select post\_id

from like\_it.answers );



1. Выводит средний рейтинг определенного пользователя (в данном случае – с id = 1) на основании оценок других пользователей за его ответы.

SELECT avg(rating)

FROM LIKE\_IT.answers\_rating as RATE

WHERE answer\_id IN

(SELECT user\_id

FROM LIKE\_IT.answers

WHERE id = 1);

C:\Users\user\IdeaProjects\by\epam\6.SQL\Безымянный2.png

1. Выводит комментарии и даты создания комментов, опубликованных за последний месяц, на ответы, опубликованные определенным Id пользователем. (В данном случае: пользователем с id = 2;

SELECT message, created\_date

FROM LIKE\_IT.comments

WHERE created\_date > DATE\_SUB(NOW(), INTERVAL 1 Month)

AND answers\_id IN

(SELECT user\_id

FROM LIKE\_IT.answers

WHERE user\_id = 2);

