**Подоксенова К.М**

**Группа Java**

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

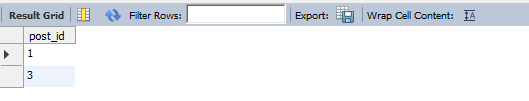
1. Запрос выбирает id прочитанных постов для определенного юзера по id. В данном случае – id = 1;

SELECT post\_id

FROM like\_it.readed\_posts

WHERE user\_id = 1;

Результат:



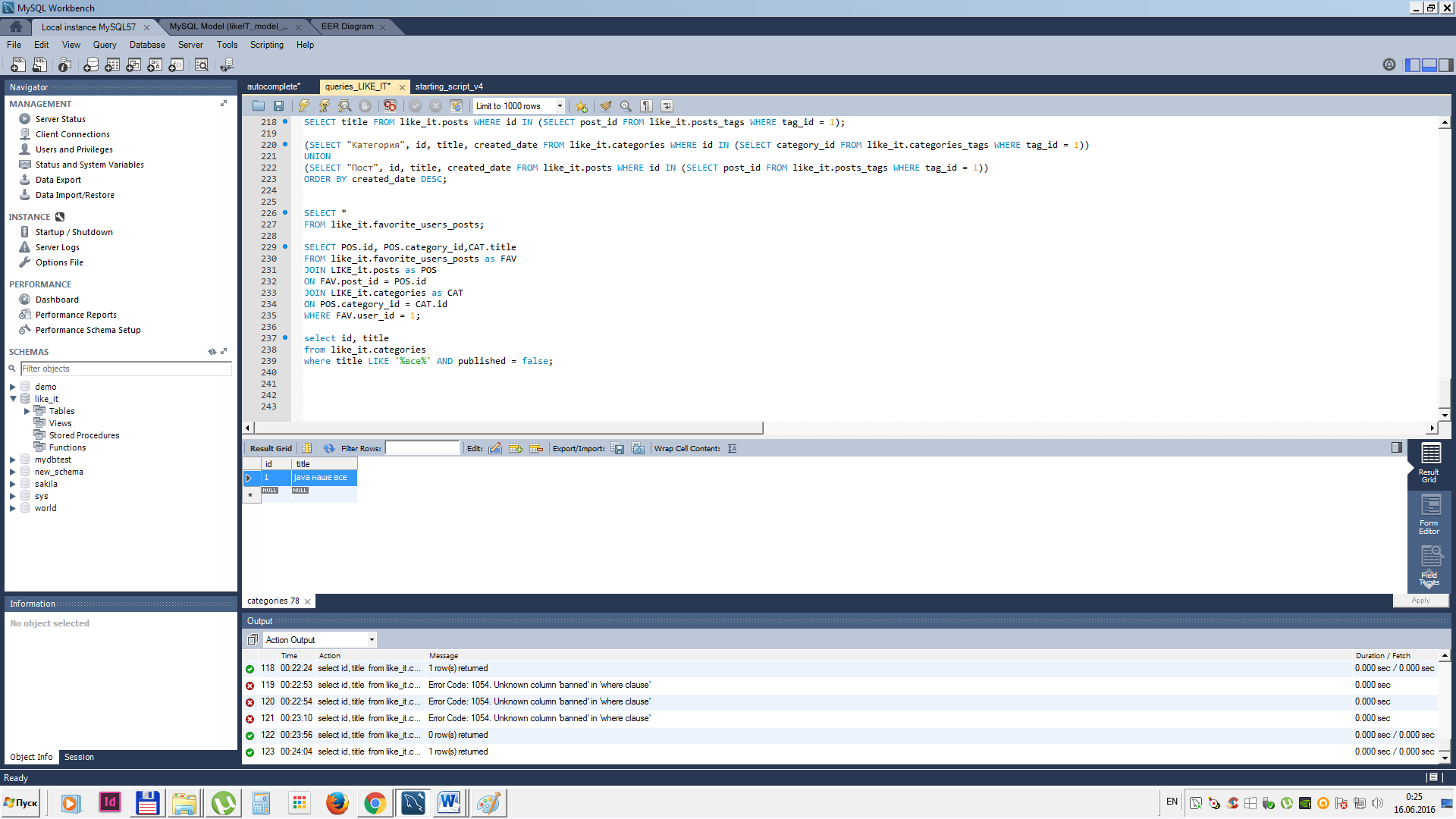
1. Выбрать все опубликованные категории, в заголовке которых участвует определенное слово. В данном случае – “все”;

Применение: поиск по запросу пользователя.

select id, title

from like\_it.categories

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



1. Выбрать все категории, соответствующие определенному id тега

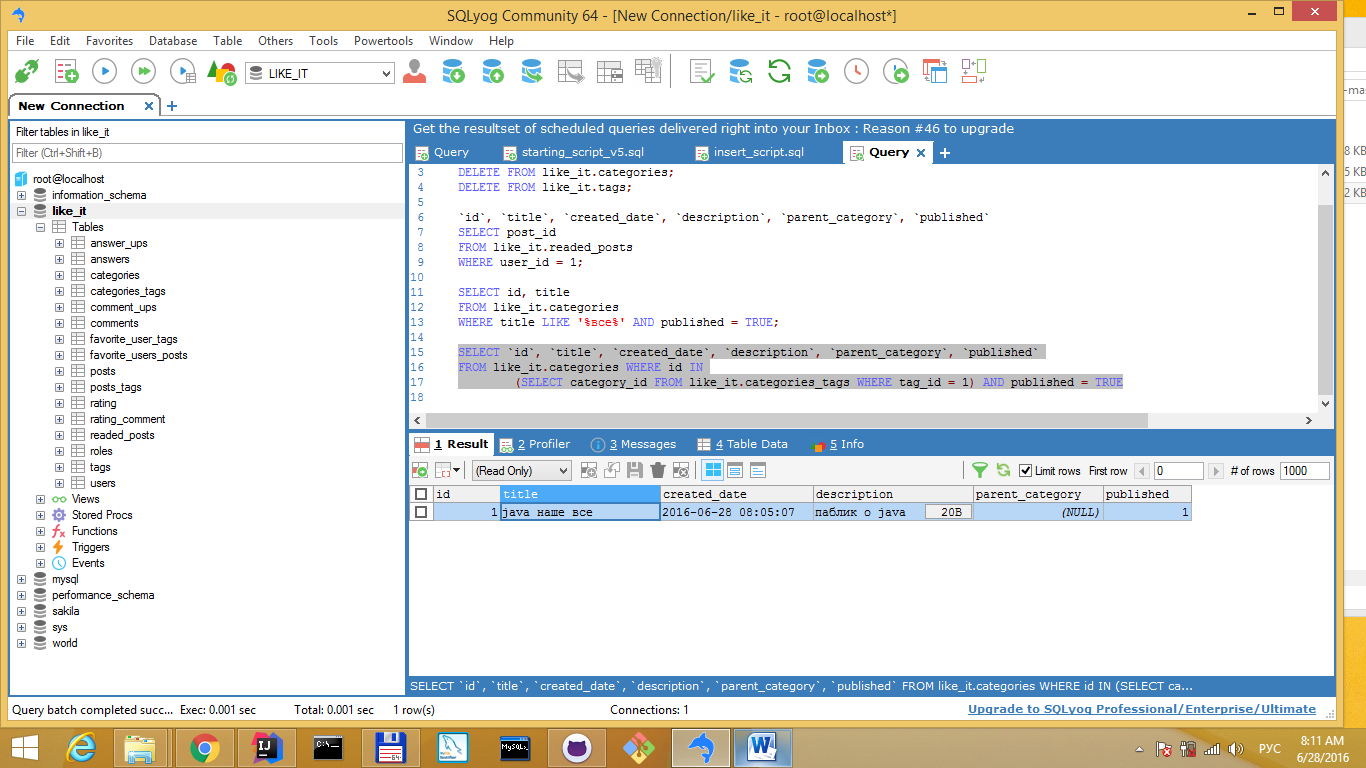
(в данном случае – 1)

Применение: поиск по запросу пользователя.

SELECT `id`, `title`, `created\_date`, `description`, `parent\_category`, `published`

FROM like\_it.categories WHERE id IN

(SELECT category\_id FROM like\_it.categories\_tags WHERE tag\_id = 1) AND published = TRUE

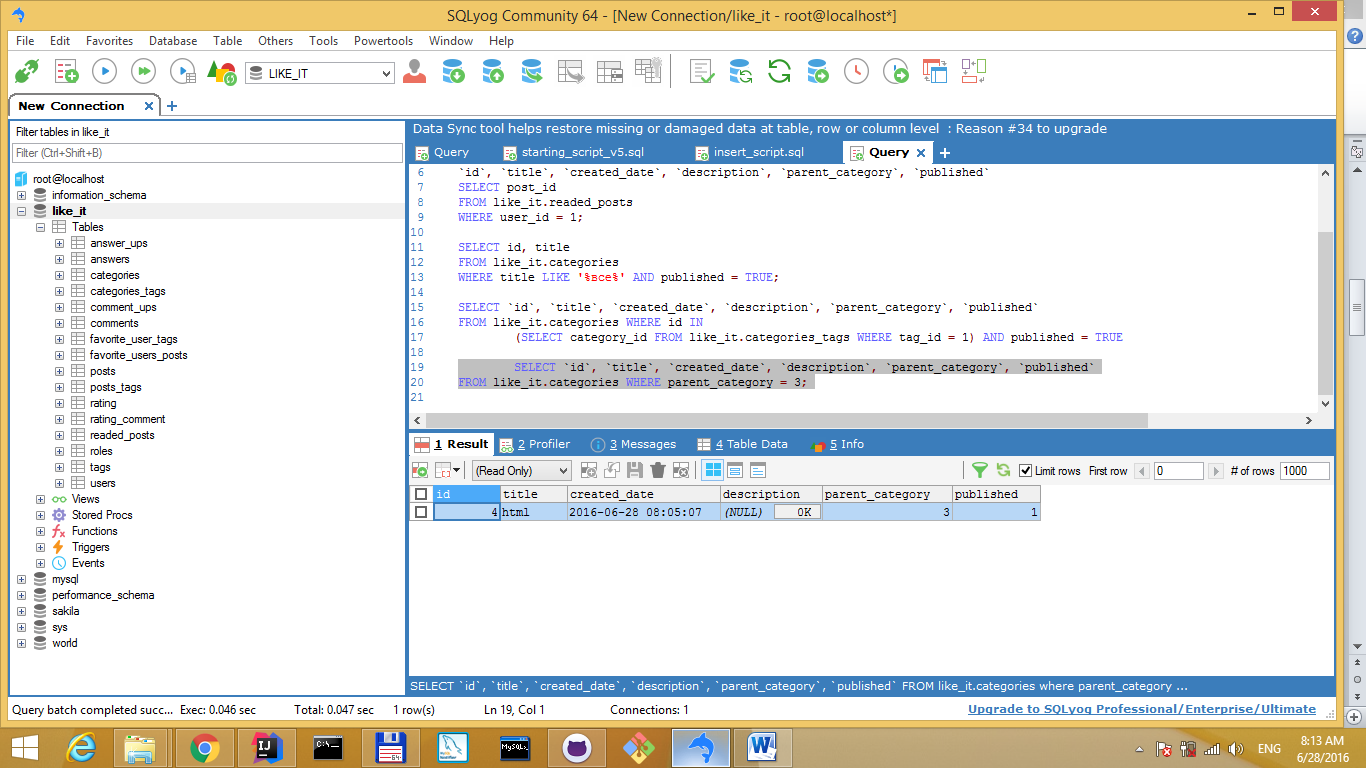


1. Выбрать все подкатегории опредленной категории, соответствующие определенному id тега

(в данном случае – 1)

SELECT `id`, `title`, `created\_date`, `description`, `parent\_category`, `published`

FROM like\_it.categories WHERE parent\_category = 3;



1. Выбрать все подкатегории опредленной категории, соответствующие определенному имени тега (в данном случае – “sql”)

SELECT `id`, `title`, `created\_date`, `description`, `parent\_category`, `published`

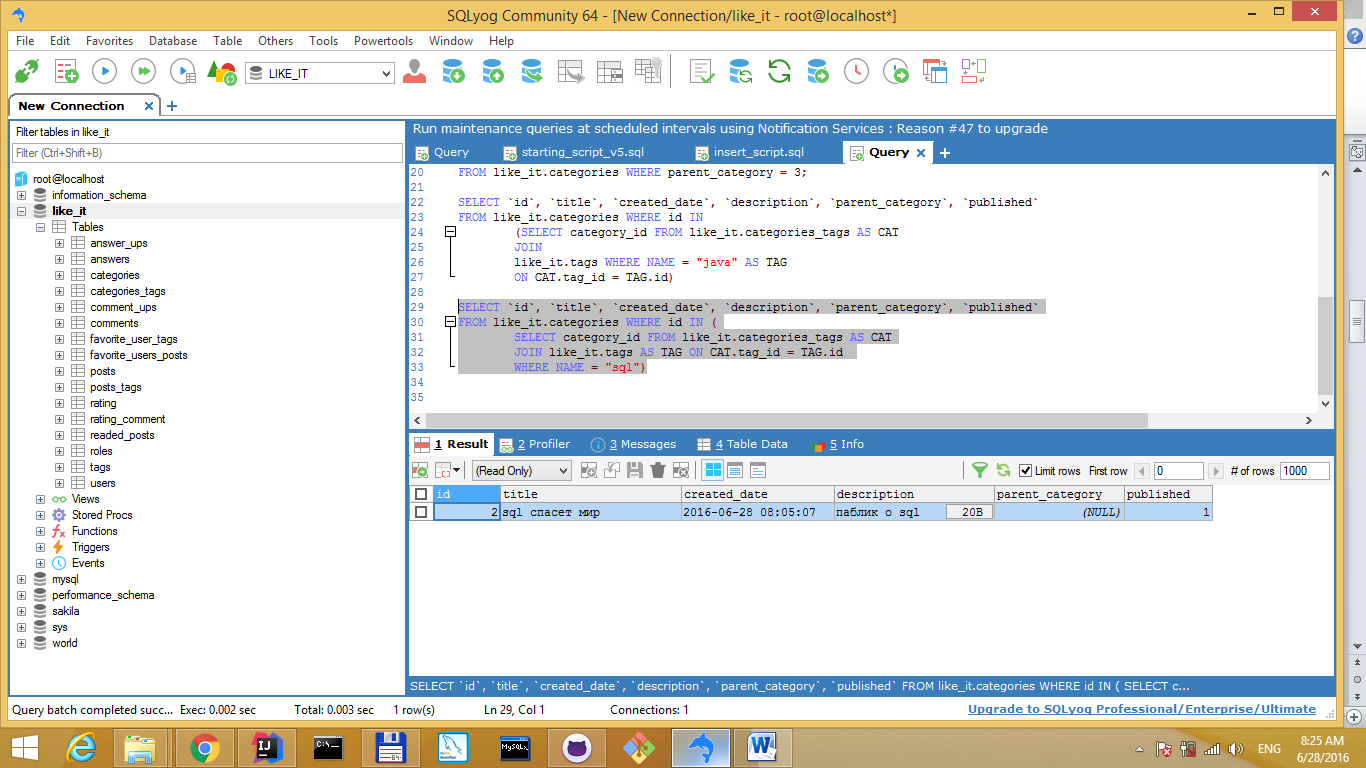
FROM like\_it.categories WHERE id IN (

SELECT category\_id FROM like\_it.categories\_tags AS CAT

JOIN like\_it.tags AS TAG ON CAT.tag\_id = TAG.id

WHERE NAME = "sql")

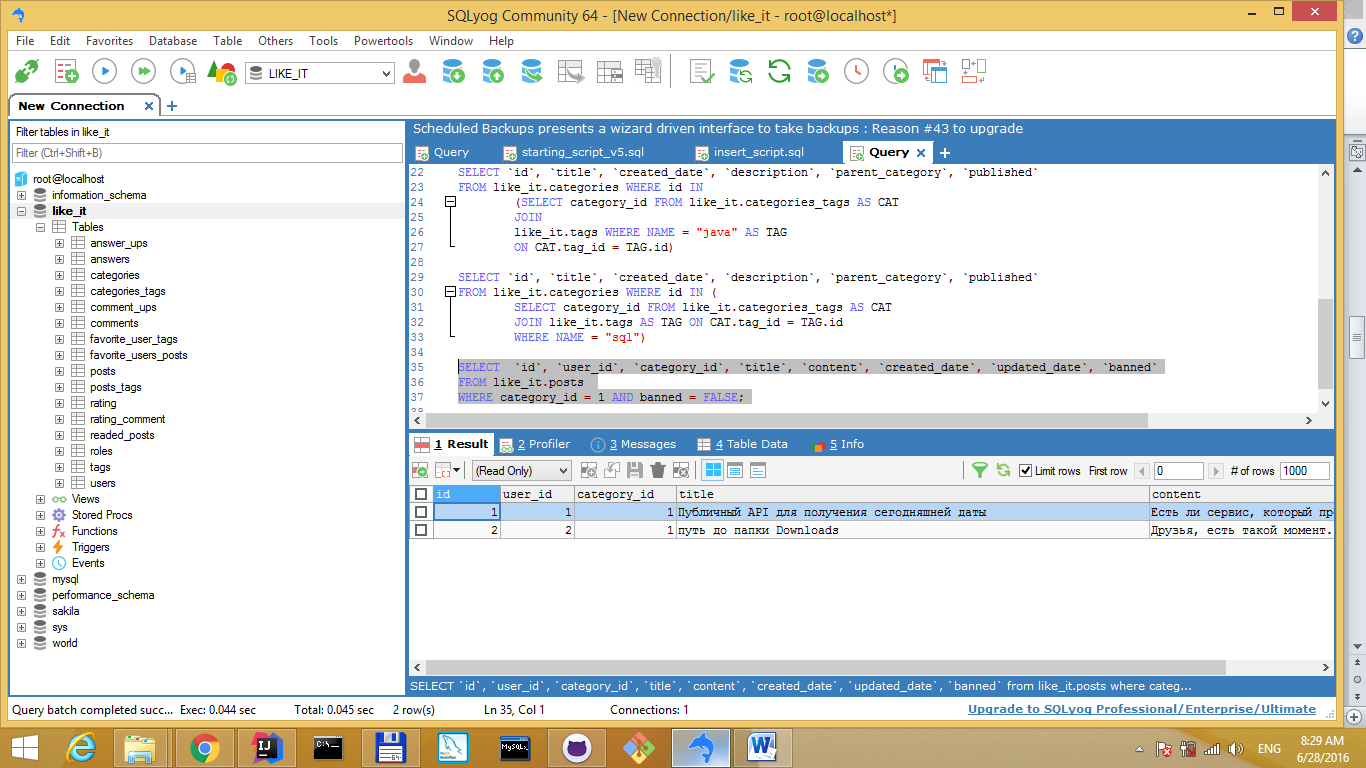
AND published = TRUE;



1. Выбрать все дочерние допустимые посты опредленной категории:

SELECT `id`, `user\_id`, `category\_id`, `title`, `content`, `created\_date`, `updated\_date`, FROM like\_it.posts

WHERE category\_id = 1 AND banned = FALSE;



1. Посчитать количество количество избранных постов в по каждой категории постов и вывести на информацию вместе с заголовком категории.

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

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

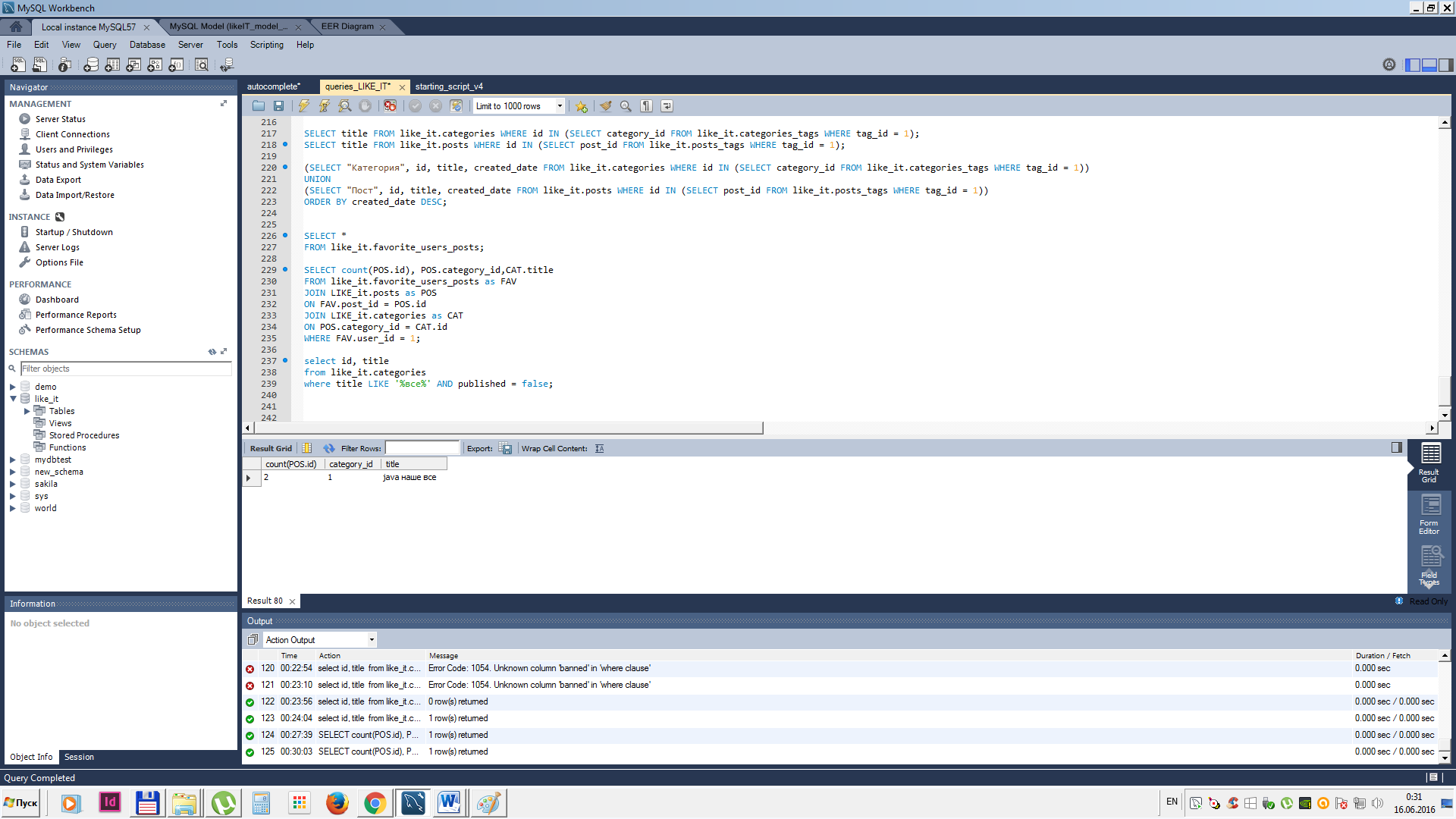
JOIN LIKE\_it.posts as POS

ON FAV.post\_id = POS.id

JOIN LIKE\_it.categories as CAT

ON POS.category\_id = CAT.id

WHERE FAV.user\_id = 1;



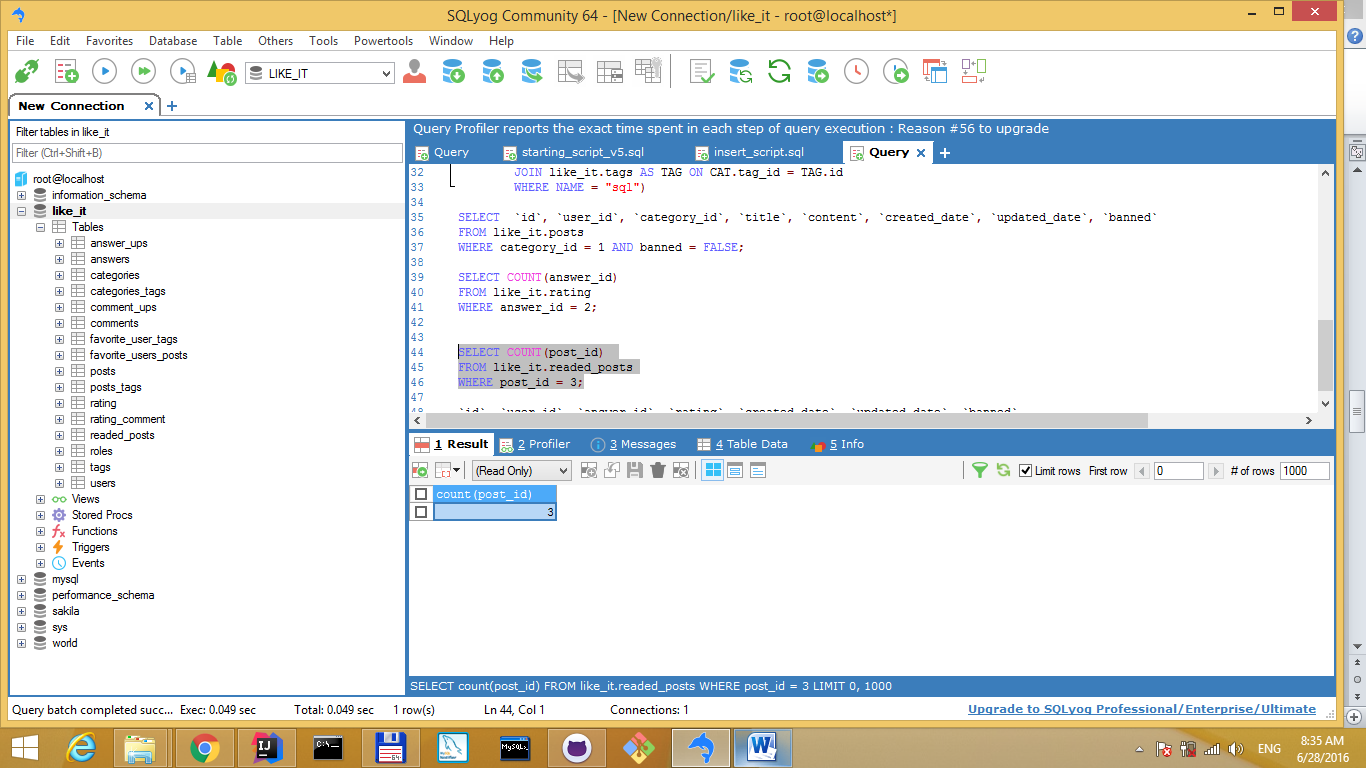
1. Посчитать количество просмотра определенного поста

Применение: информация отображается на странице поста

SELECT count(post\_id)

FROM like\_it.readed\_posts

WHERE post\_id = 3;

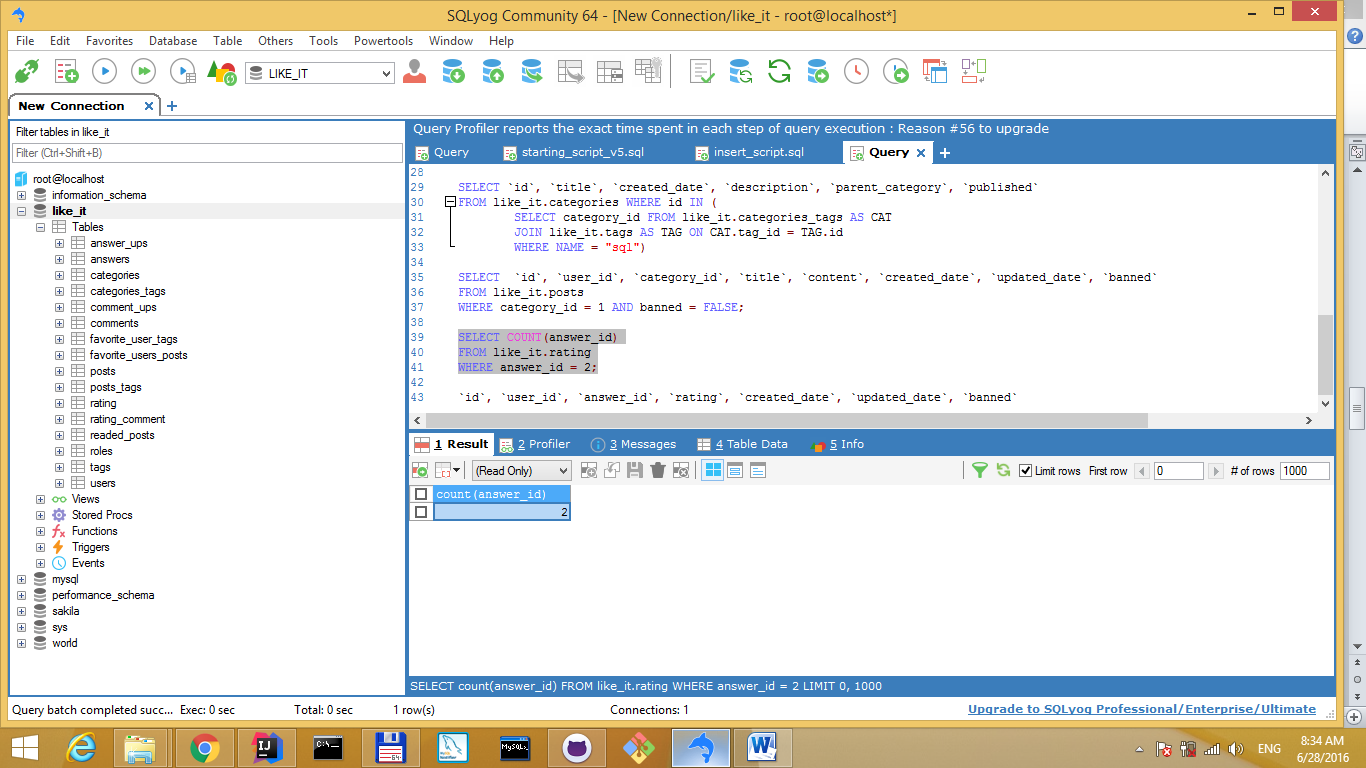


1. Посчитать количество оценок определенного ответа.

SELECT COUNT(answer\_id)

FROM like\_it.rating

WHERE answer\_id = 2;

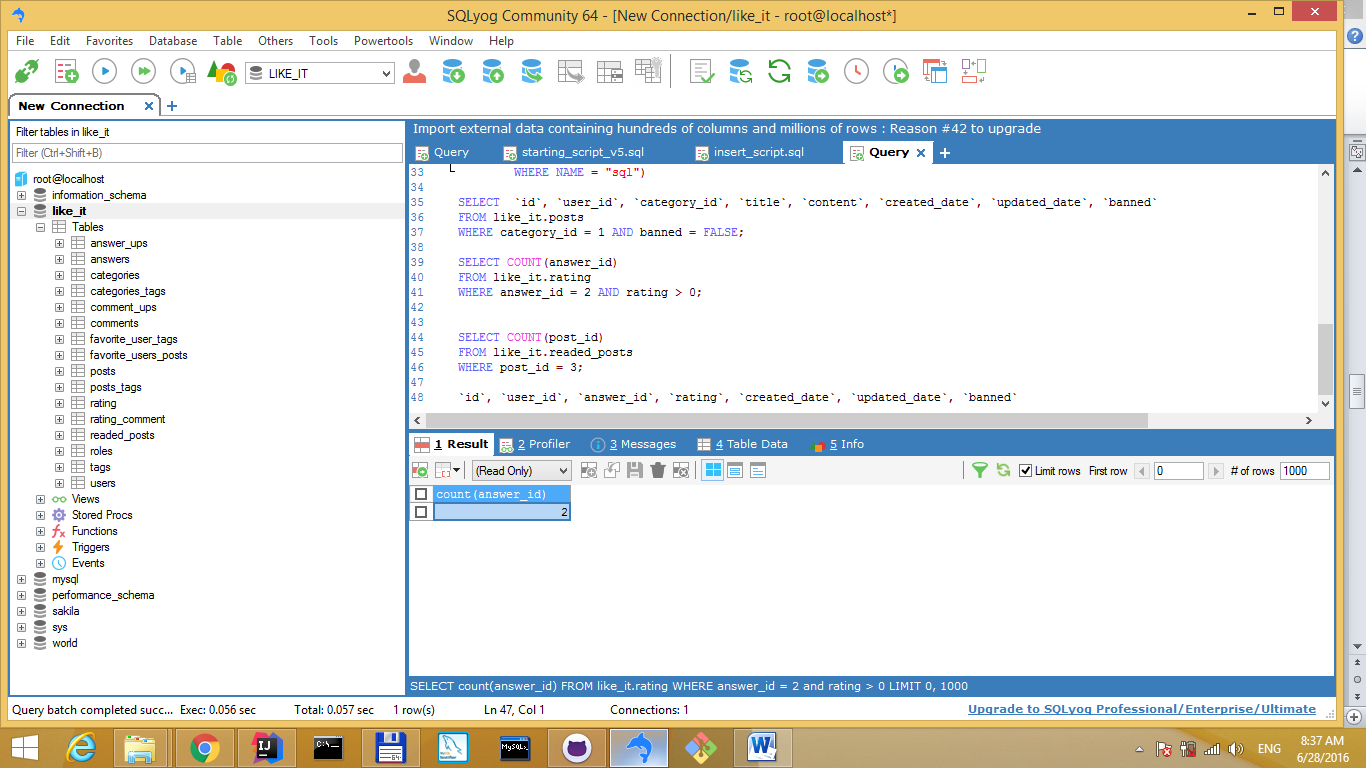


1. Посчитать количество положительных оценок определенного ответа.

SELECT COUNT(answer\_id)

FROM like\_it.rating

WHERE answer\_id = 2 AND rating > 0;

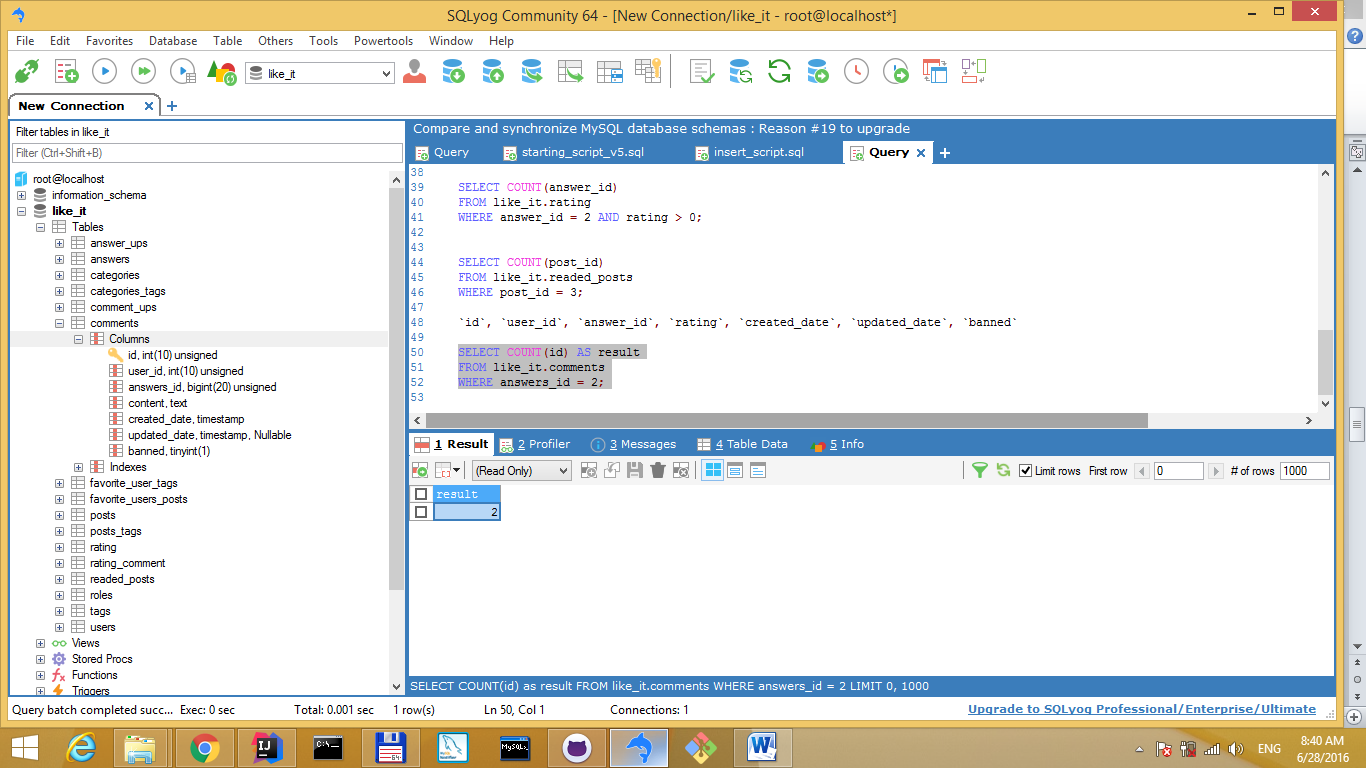


1. Посчитать количество комментов на ответ

SELECT COUNT(id) AS result

FROM like\_it.comments

WHERE answers\_id = 2;

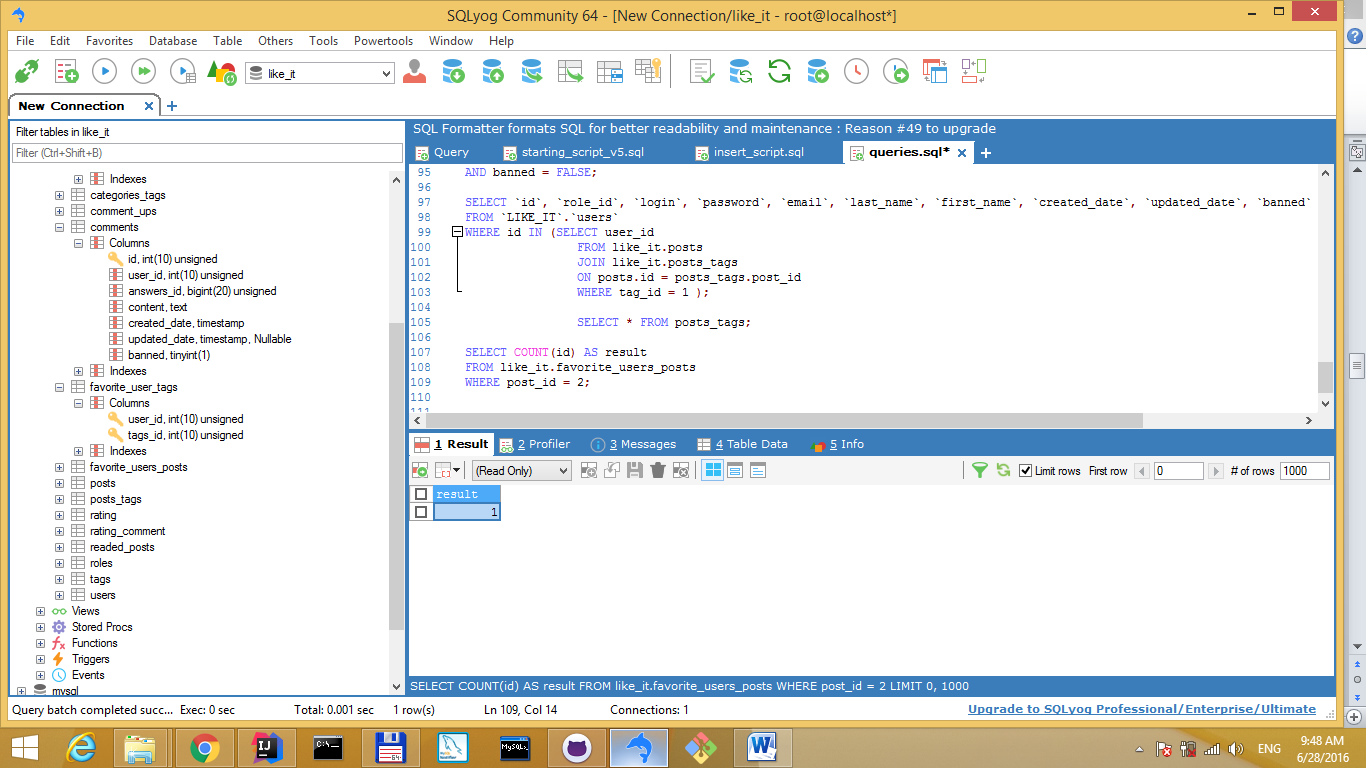


1. Посчитать количество пользователей, добавивших пост в избранное

SELECT COUNT(id) AS result

FROM like\_it.favorite\_users\_posts

WHERE post\_id = 2;



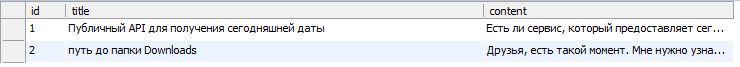
1. Вывести самые обсуждаемые посты по категории (суммарное количество ответов и комментов):
2. Запрос выбирает id тех постов, их заголовок и содержание, на которых не были даны ответы.

SELECT id, title, content

FROM like\_it.posts

WHERE id NOT IN (select post\_id

from like\_it.answers );



1. Запрос выбирает все посты определенного пользователя, на которые были размещены или изменены ответы в течение определенного периода (в данном случае – полгода)

SELECT post\_id, PO.content, AN.id, AN.content, AN.created\_date

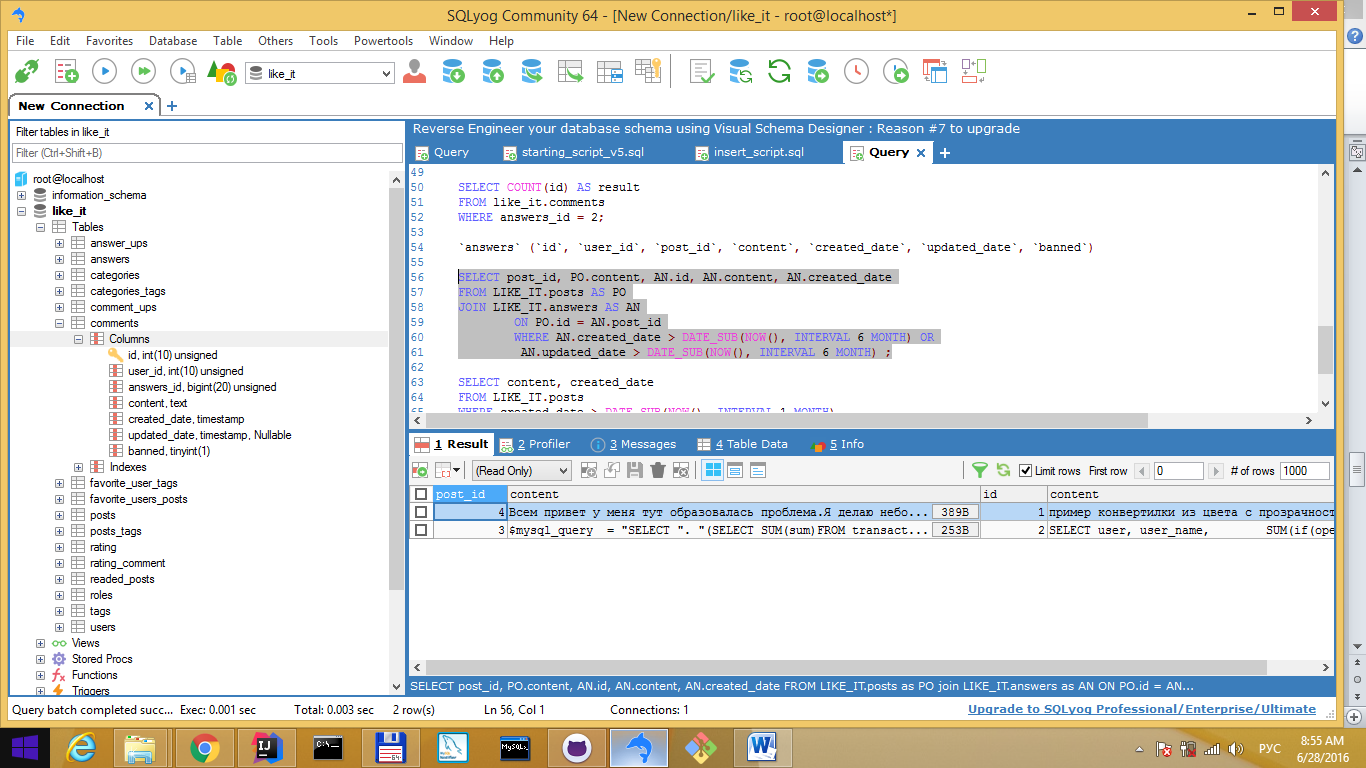
FROM LIKE\_IT.posts AS PO

JOIN LIKE\_IT.answers AS AN

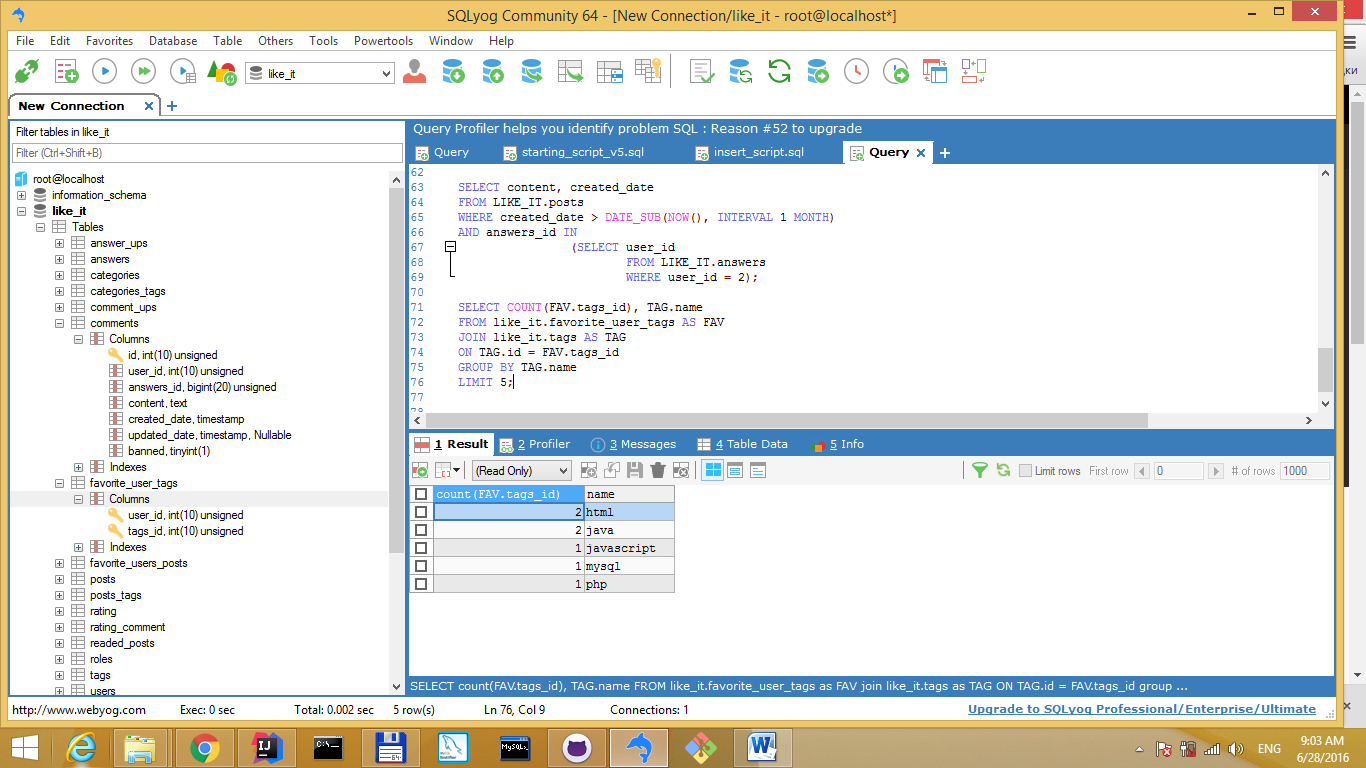
ON PO.id = AN.post\_id

WHERE AN.created\_date > DATE\_SUB(NOW(), INTERVAL 6 MONTH) OR

AN.updated\_date > DATE\_SUB(NOW(), INTERVAL 6 MONTH) ;



1. Выбрать топ 5 популярных тегов среди юзеров.



SELECT COUNT(FAV.tags\_id), TAG.name

FROM like\_it.favorite\_user\_tags AS FAV

JOIN like\_it.tags AS TAG

ON TAG.id = FAV.tags\_id

GROUP BY TAG.name

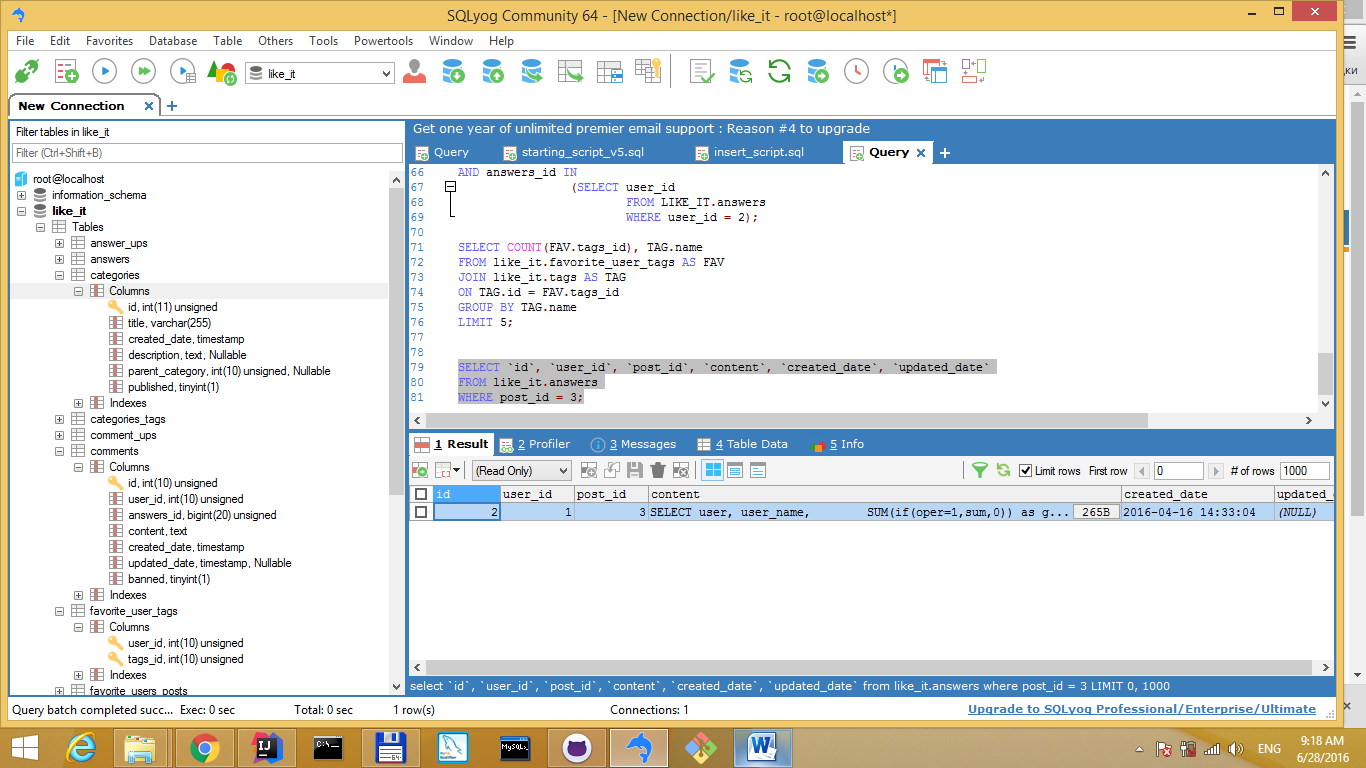
LIMIT 5;

1. Выбрать все допустимые ответы на определенный пост

SELECT `id`, `user\_id`, `post\_id`, `content`, `created\_date`, `updated\_date`

FROM like\_it.answers

WHERE post\_id = 3 and banned = false;



1. Выбрать все допустимые ответы на все посты определенного юзера

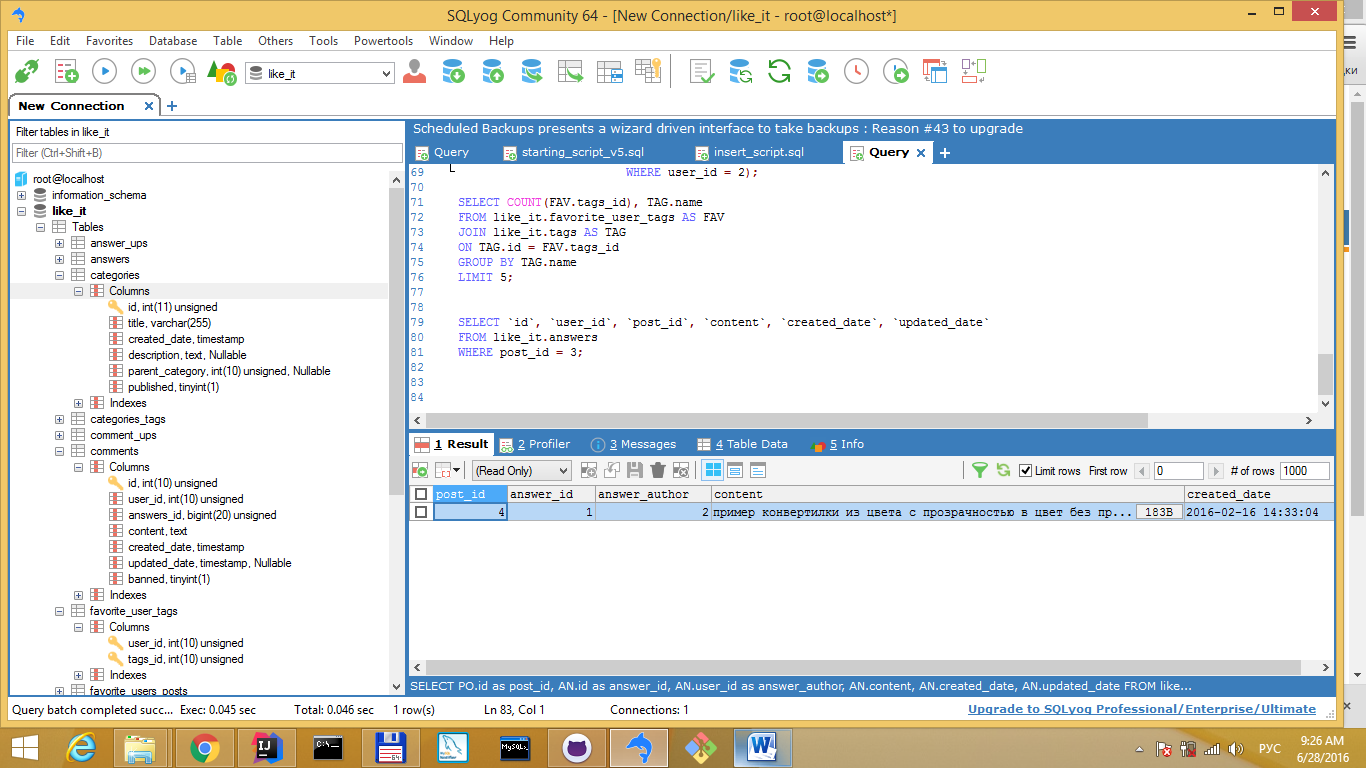
SELECT PO.id AS post\_id, AN.id AS answer\_id, AN.user\_id AS answer\_author, AN.content, AN.created\_date, AN.updated\_date

FROM like\_it.answers AS AN

JOIN like\_it.posts AS PO

ON AN.post\_id = PO.id

WHERE PO.user\_id = 3 AND PO.banned = false AND AN.banned = false;



1. Выбрать все посты, соответствующие определенному имени тега (в данном случае – “java”)

SELECT `id`, `user\_id`, `category\_id`, `content`, `created\_date`, `updated\_date`

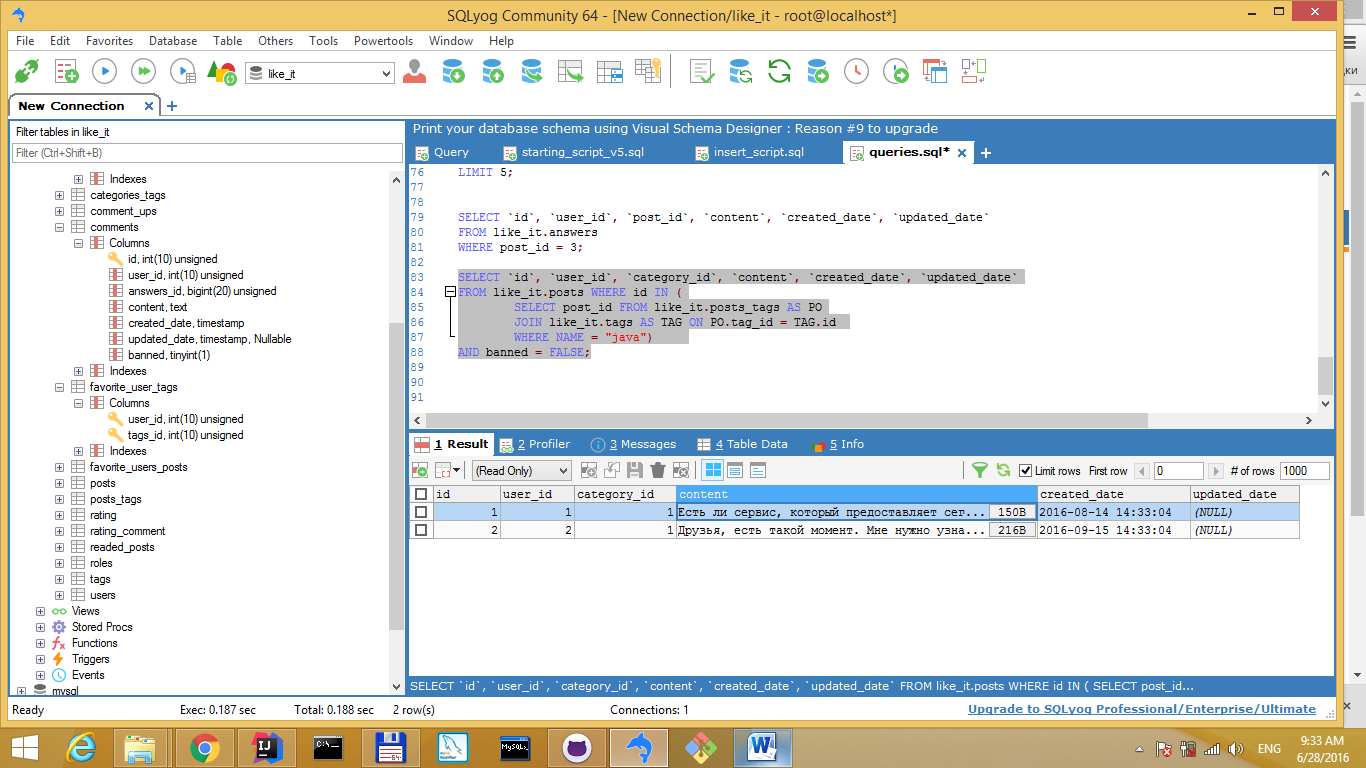
FROM like\_it.posts WHERE id IN (

SELECT post\_id FROM like\_it.posts\_tags AS PO

JOIN like\_it.tags AS TAG ON PO.tag\_id = TAG.id

WHERE NAME = "java")

AND banned = FALSE;



1. Выбрать всех юзеров, которые публиковали посты с тегом, находящимся в избранных у определенного юзера

SELECT `id`, `role\_id`, `login`, `password`, `email`, `last\_name`, `first\_name`, `created\_date`, `updated\_date`, `banned`

FROM `LIKE\_IT`.`users`

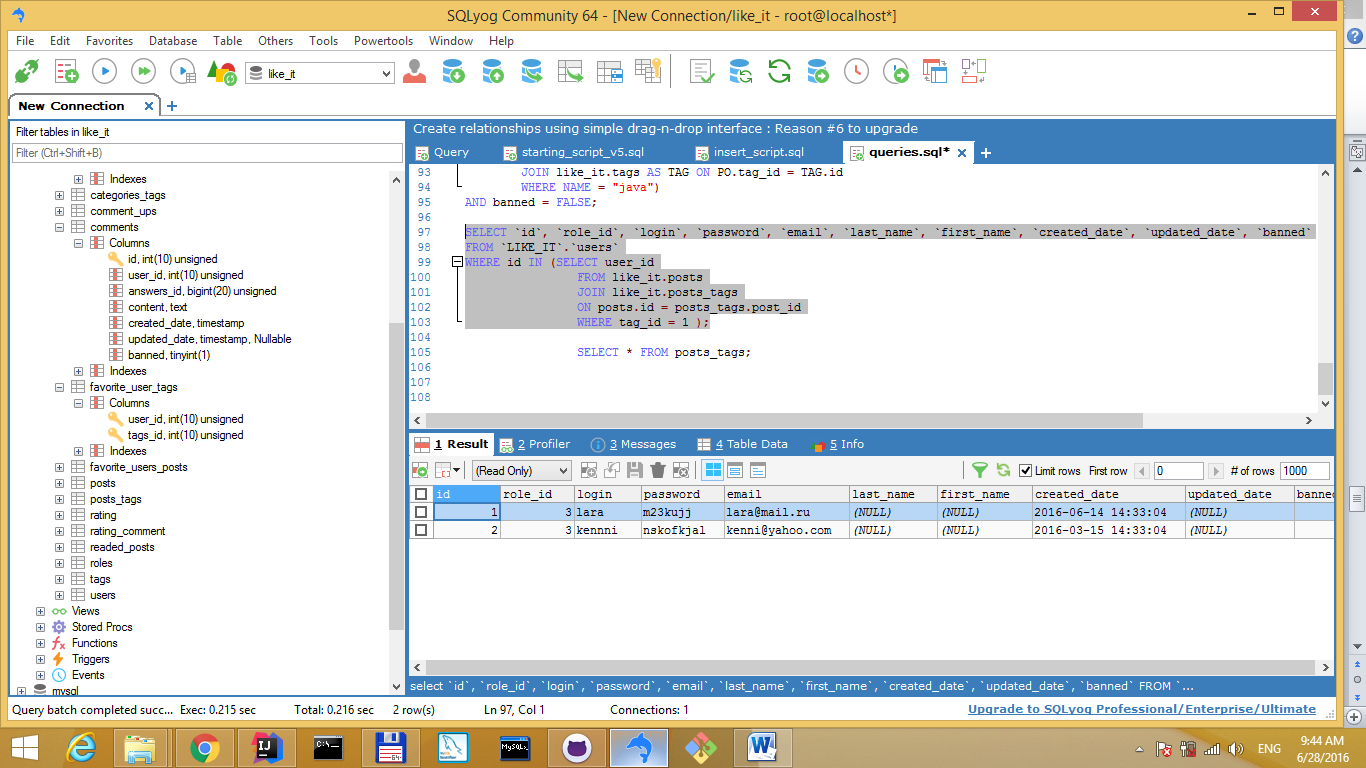
WHERE id IN (SELECT user\_id

FROM like\_it.posts

JOIN like\_it.posts\_tags

ON posts.id = posts\_tags.post\_id

WHERE tag\_id = 1 );



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

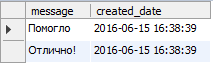
WHERE created\_date > DATE\_SUB(NOW(), INTERVAL 1 Month)

AND answers\_id IN

(SELECT user\_id

FROM LIKE\_IT.answers

WHERE user\_id = 2);



1. Запрос выбирает всех юзеров и их почтовые адреса , добавивших один из данных тегов в избранное. В данном случае - "java" и "sql"

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

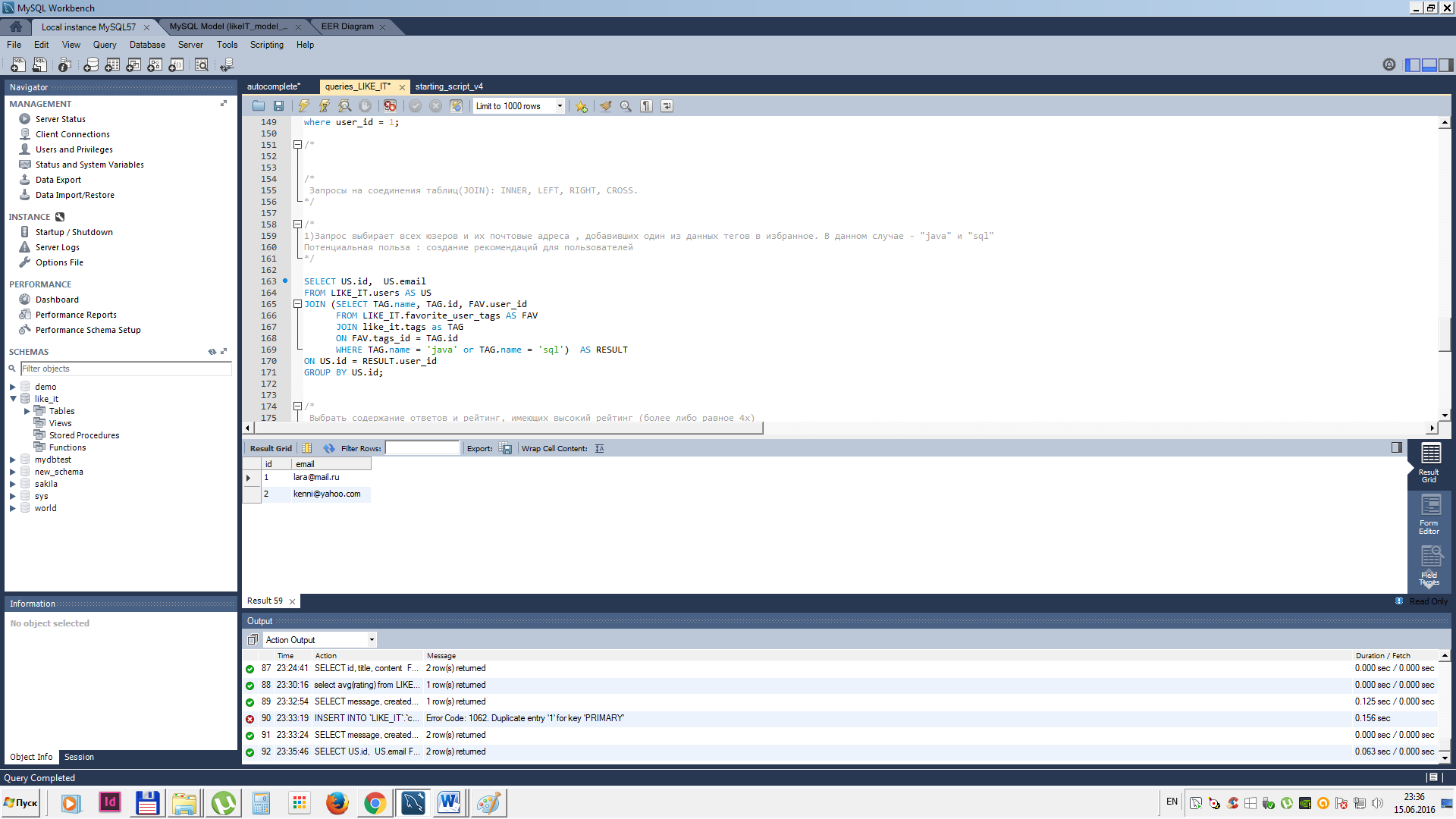
JOIN like\_it.tags as TAG

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. 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

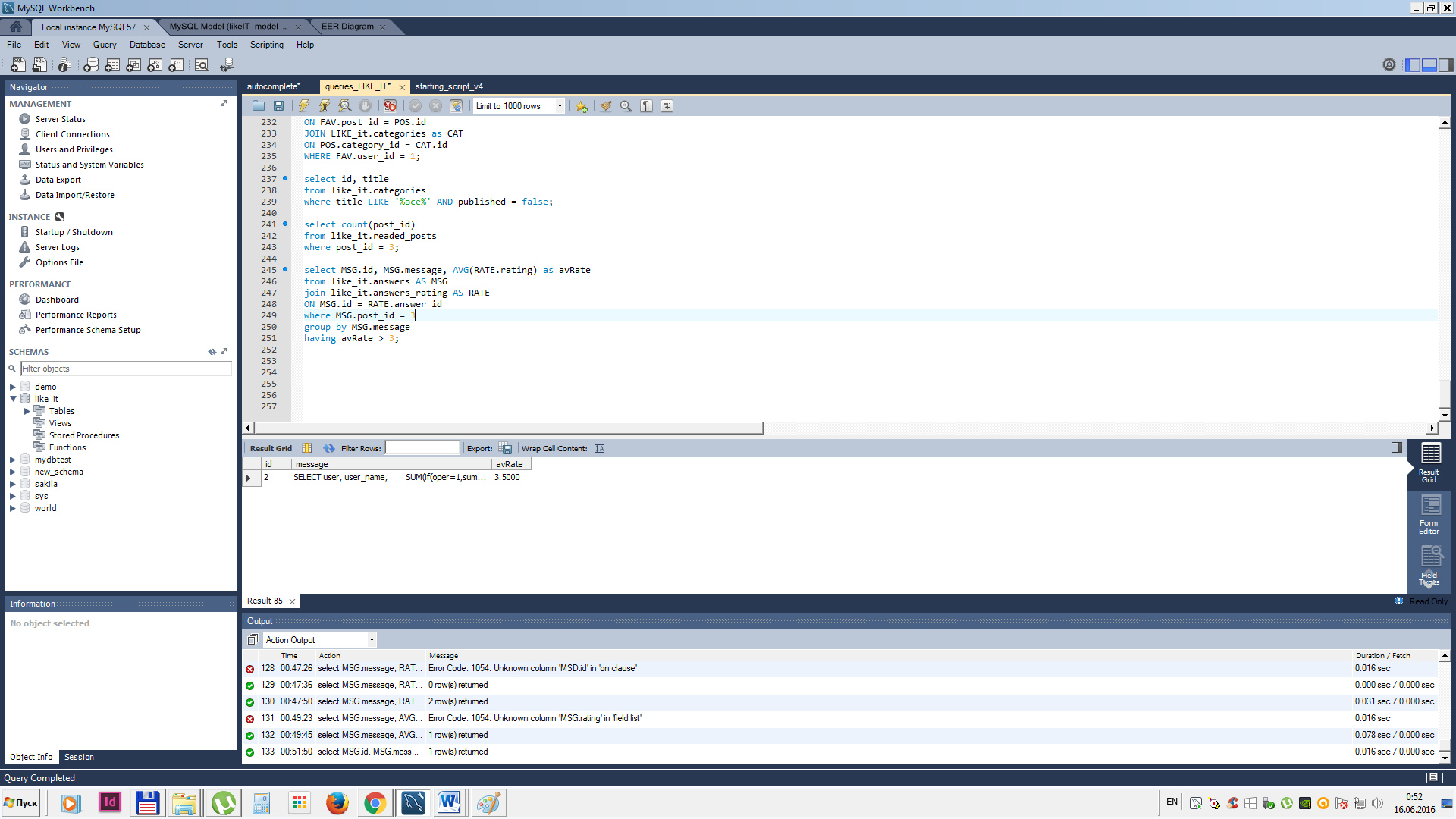
JOIN like\_it.\_rating AS RATE

ON MSG.id = RATE.answer\_id

WHERE MSG.post\_id = 3

GROUP BY MSG.message

HAVING avRate > 3;



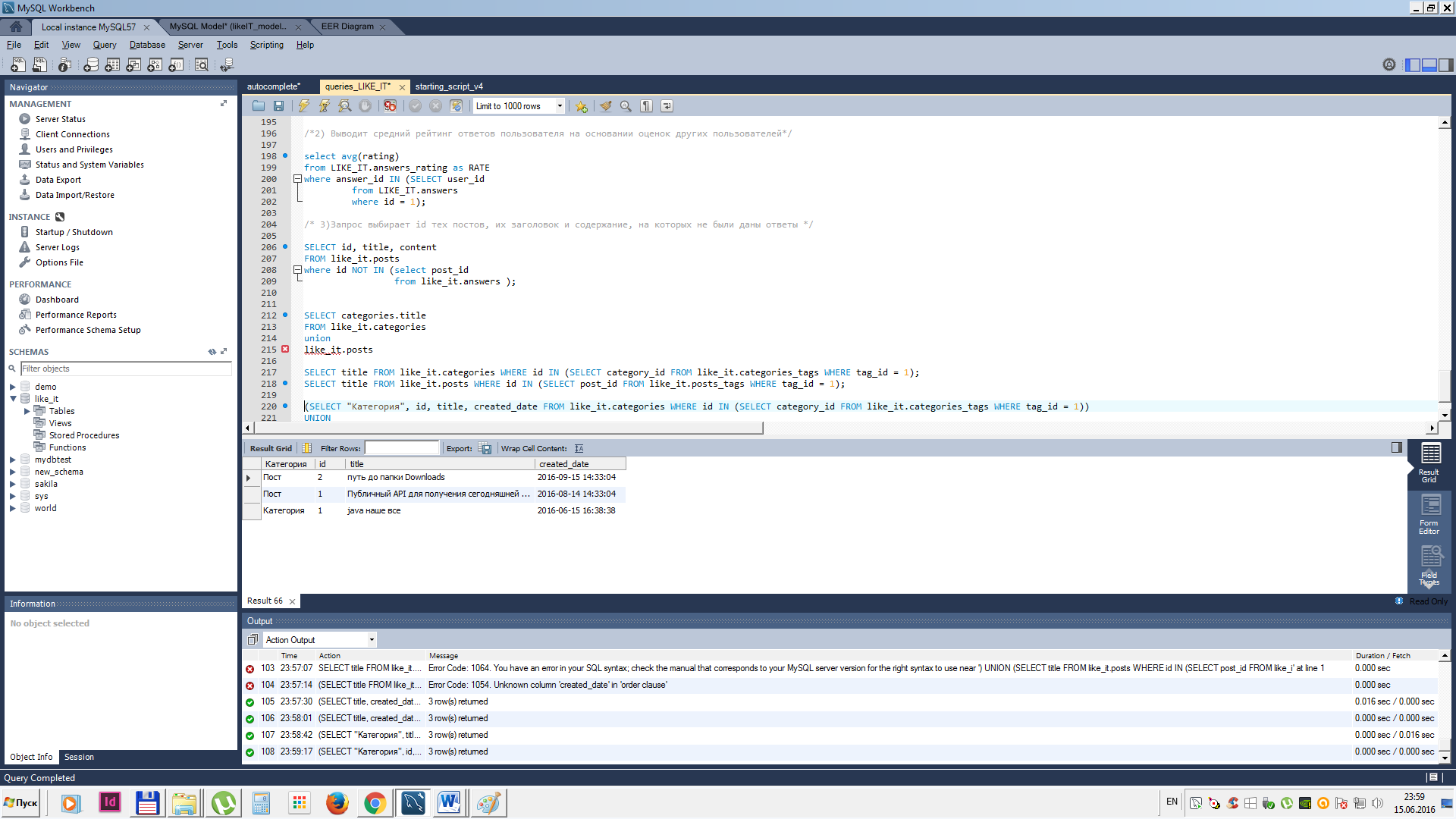
1. Выбрать все посты и категории, в котые связаны с определенным тегом (в данном случае – с id=1) . Отобразить их тип, id, заголовок и дату создания. Упорядочить по дате создания.

(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. 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. 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 количество оцененных положительно ответов пользователя

SELECT COUNT(RATE.rating)

FROM LIKE\_IT.rating AS RATE

WHERE rating > 0 AND answer\_id IN

(SELECT id

FROM LIKE\_IT.answers

WHERE user\_id = 1)

