**Проект «Ликвидность»**

**Блок 1. Автоматический импорт в БД неструктурированных данных.**

Данный блок реализуется в среде MySQL посредством исполнения файла script.sql. Для этого в используемой среде создается пустая база данных «exercise1» с 3-мя пустыми таблицами: для данных по заявкам с привилегированными акциями, для данных по заявкам с обыкновенными акциями и для данных по заявкам с облигациями.

База данных была создана с помощью следующего запроса:

create database exercise1;

Классификатор ценных бумаг по типам строится по классификации с сайта Московской Биржи *(файл* *ListingSecurityList.csv)* с помощью следующего запроса:

*use exercise1;*

*create table list(*

*Instrument\_id bigint,*

*Instrument\_type varchar(100),*

*TRADE\_CODE varchar(30) not NULL);*

*LOAD DATA Infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/ListingSecurityList.csv'*

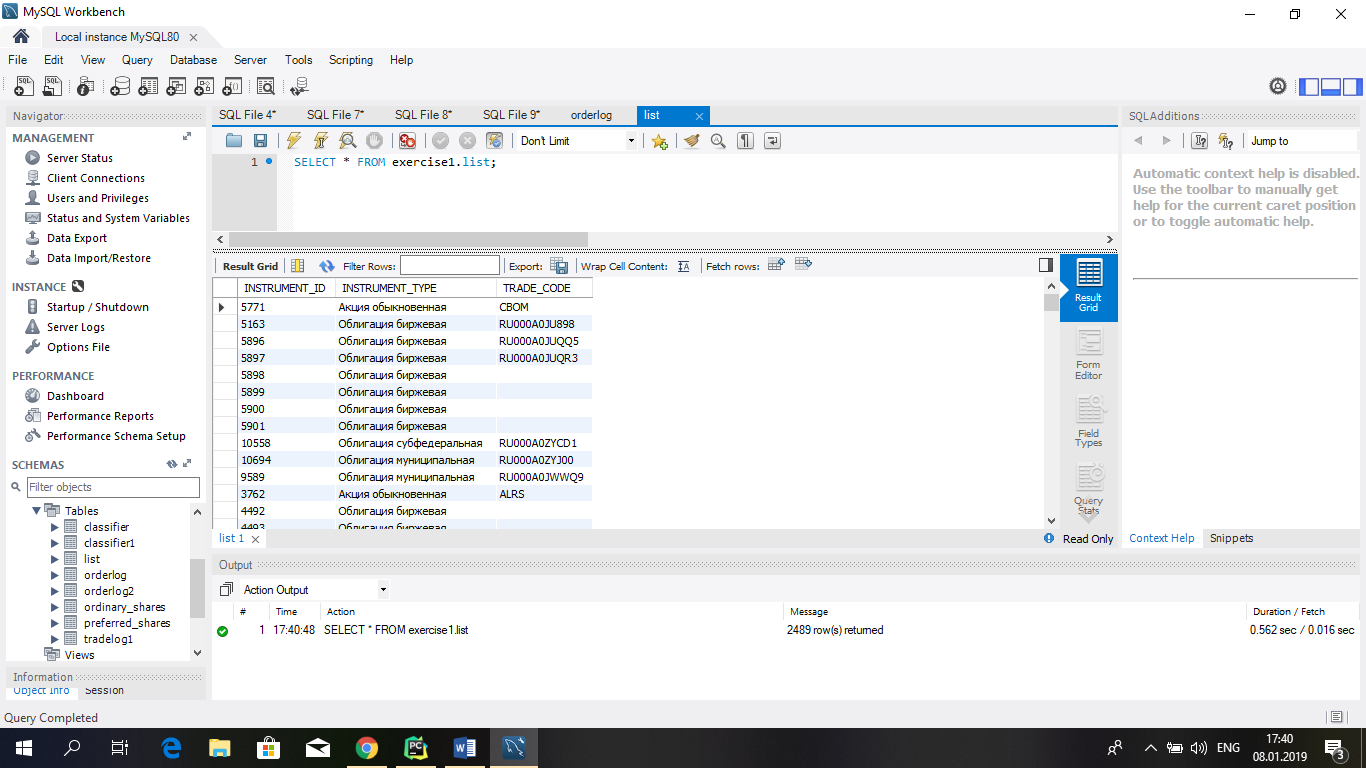
*INTO TABLE list*

*FIELDS TERMINATED BY ';'*

*LINES TERMINATED BY '\r\n'*

*IGNORE 1 LINES;*

*Рисунок 1: Таблица-классификатор.*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Производится следующая работа с исходными данными: выкачиваются два файла OrderLog20151207 и OrderLog20151222, соответствующие 7-му и 22-му числам декабря 2015 года, и форматируются из .txt к формату .csv. Далее выбранные данные по двум дням объединяются в единую таблицу с указанием даты происходивших заявок. Так как выполнение запросов для объединенной таблицы занимает много времени, запросы выполнялись для отдельных таблиц обыкновенных акций, привилегированных акций и облигаций.

В процессе описанных выше действий создадется таблица Orderlog2 на 2015.12.07 с пустыми значениями даты с помощью запроса:

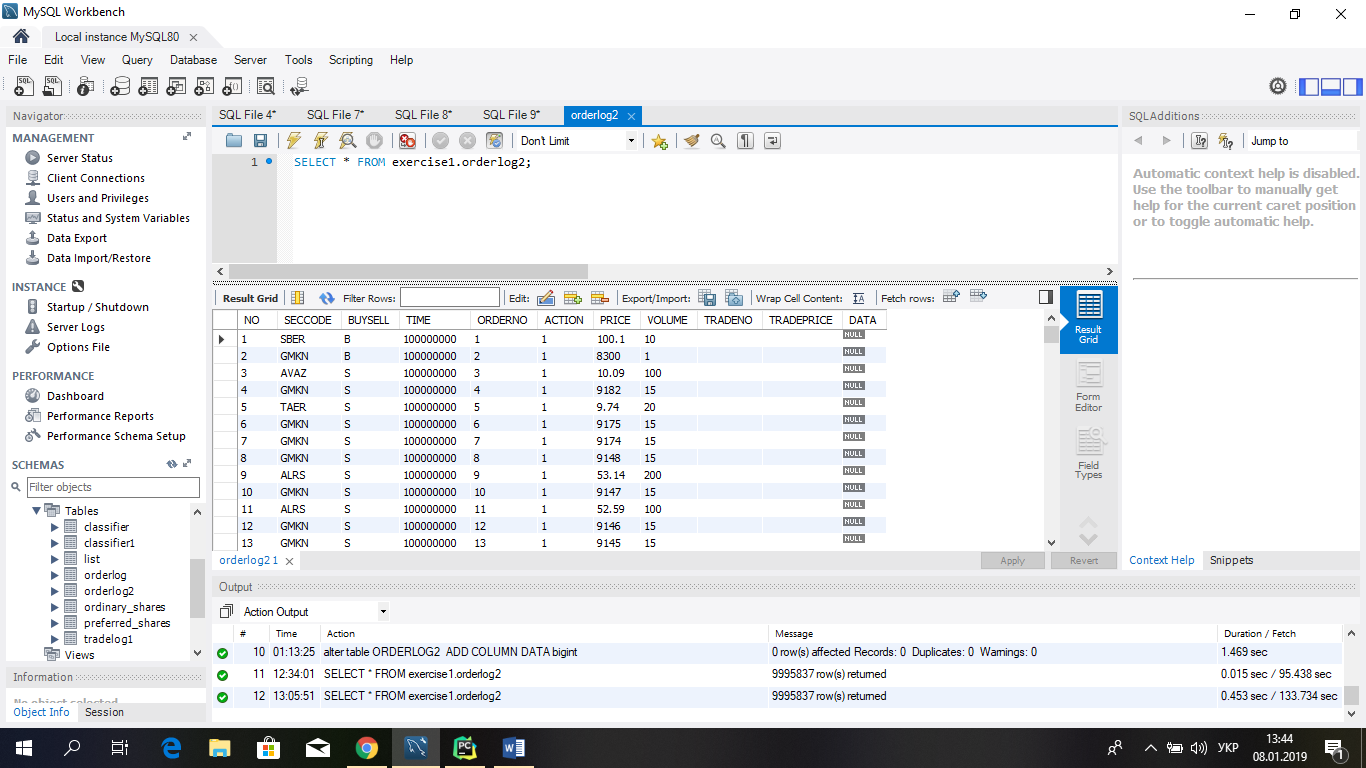
*use EXERCISE1;*

*alter table ORDERLOG2*

*ADD COLUMN DATA bigint;*

Результатом исполнения данного запроса является таблица, представленная на рисунке 2.

*Рисунок 2: Таблица «Orderlog2» для данных на 07.12.2019.*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

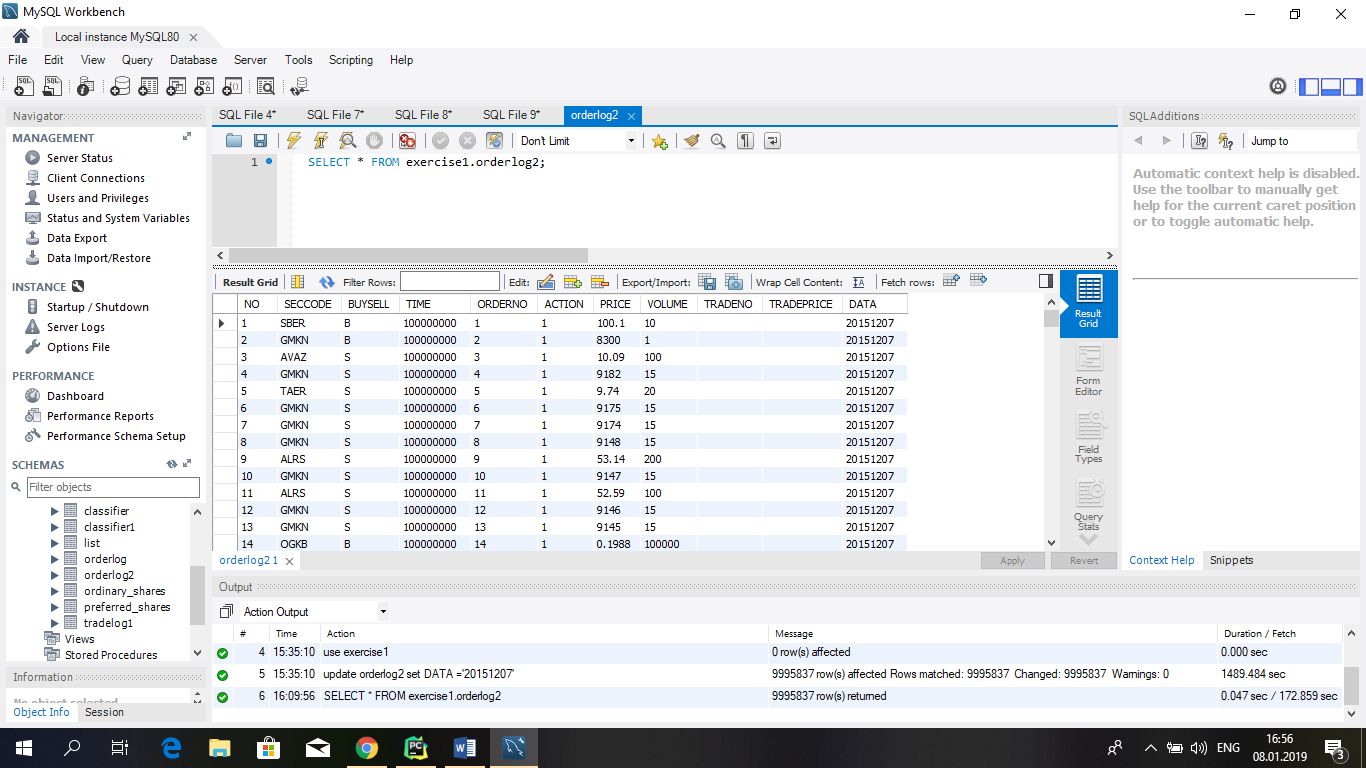
Далее происходит заполнение последней колонки представленной выше таблицы датой 20151207 с помощью следующего запроса:

*SET SQL\_SAFE\_UPDATES = 0;*

*use exercise1;*

*update orderlog set DATA ='20151207';*

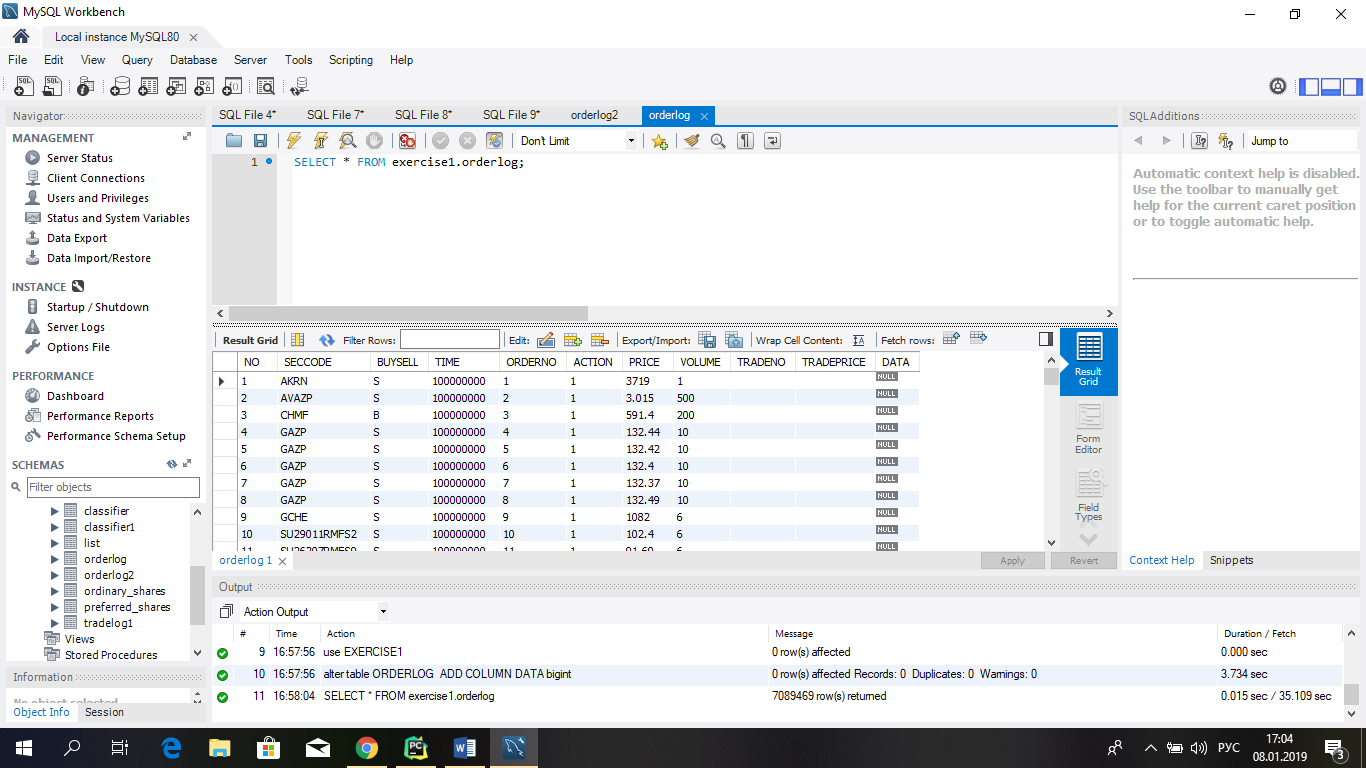
*Рисунок 3: Таблица «Orderlog2» для данных на 07.12.2019 конечная.*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Далее осуществляется создание колонки DATA в таблице «Orderlog» для данных по заявкам с 22.12.2015.

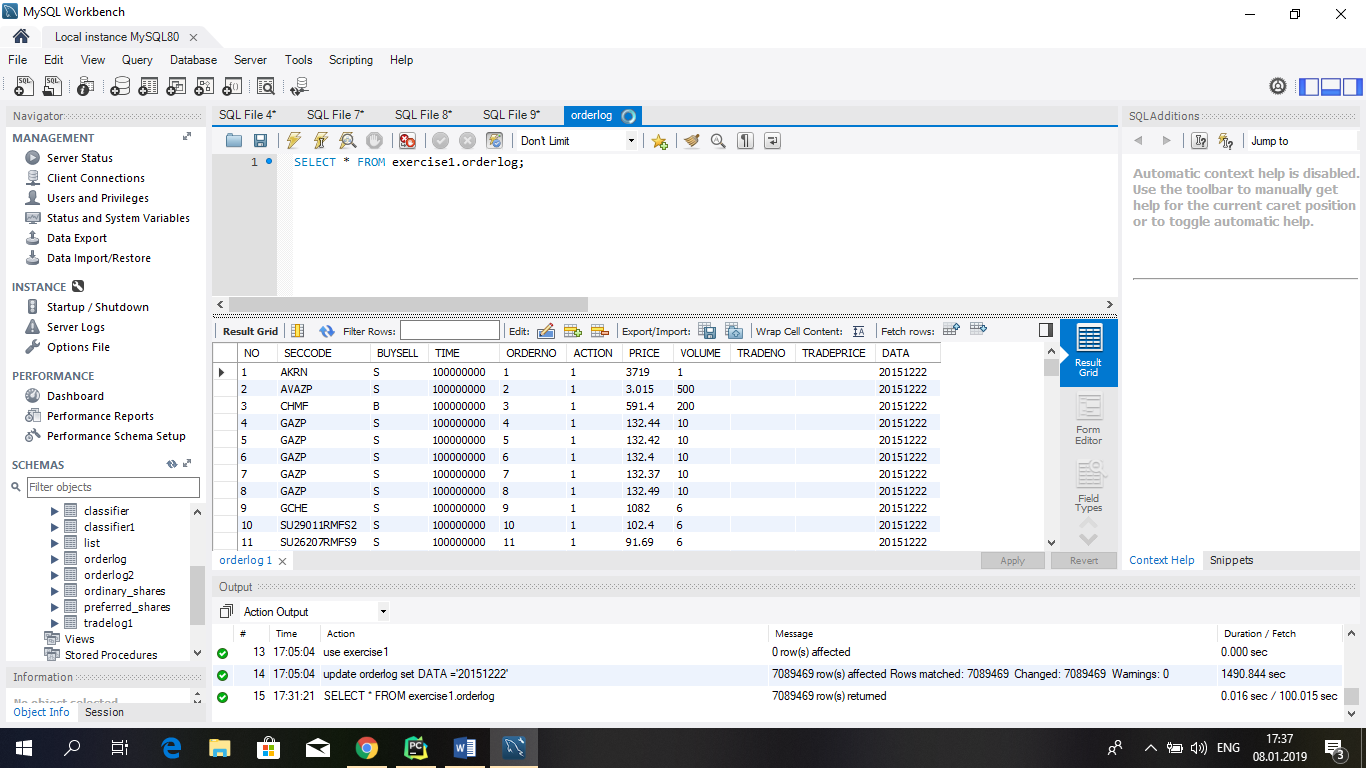
*Рисунок 4: Таблица «Orderlog» для данных на 22.12.2019.*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Далее путем осуществления части запроса производится заполнение последней колонки таблицы «Orderlog» датой «20151222» для получения следующего результата:

*Рисунок 5: Таблица «Orderlog» для данных на 22.12.2019 конечная.*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

После вышеописанных процедур с помощью таблицы-классификатора происходит деление таблиц на таблицы с данными по заявкам с обыкновенными акциями (Рисунок 6), с привилегированными акциями (Рисунок 7) и с облигациями (Рисунок 8).

Для создания таблицы с заявками по обыкновенным акциям за 22.12.2015 используем следующий код:

*use exercise1;*

*create table if not exists Ordinary\_shares1*

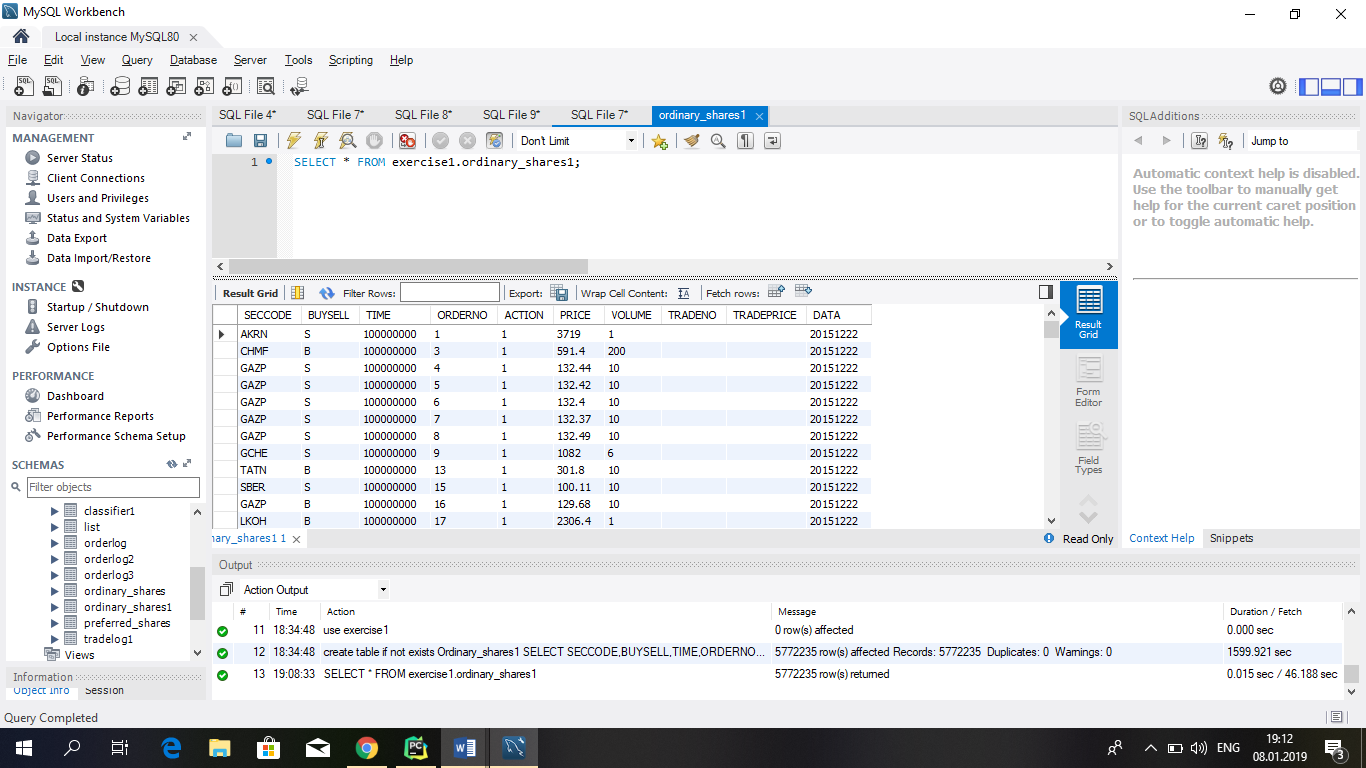
*SELECT SECCODE, BUYSELL,TIME,ORDERNO,ACTION,PRICE,VOLUME,TRADENO,TRADEPRICE,DATA*

*FROM Orderlog JOIN list*

*ON SECCODE=TRADE\_CODE*

*WHERE INSTRUMENT\_TYPE = 'Акция обыкновенная'*

*Рисунок 6. Таблица обыкновенных акций за 22.12.2015*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Для создания таблицы с заявками по привилегированным акциям за 22.12.2015 используем следующий код:

*use exercise1;*

*create table if not exists Preferred\_shares1*

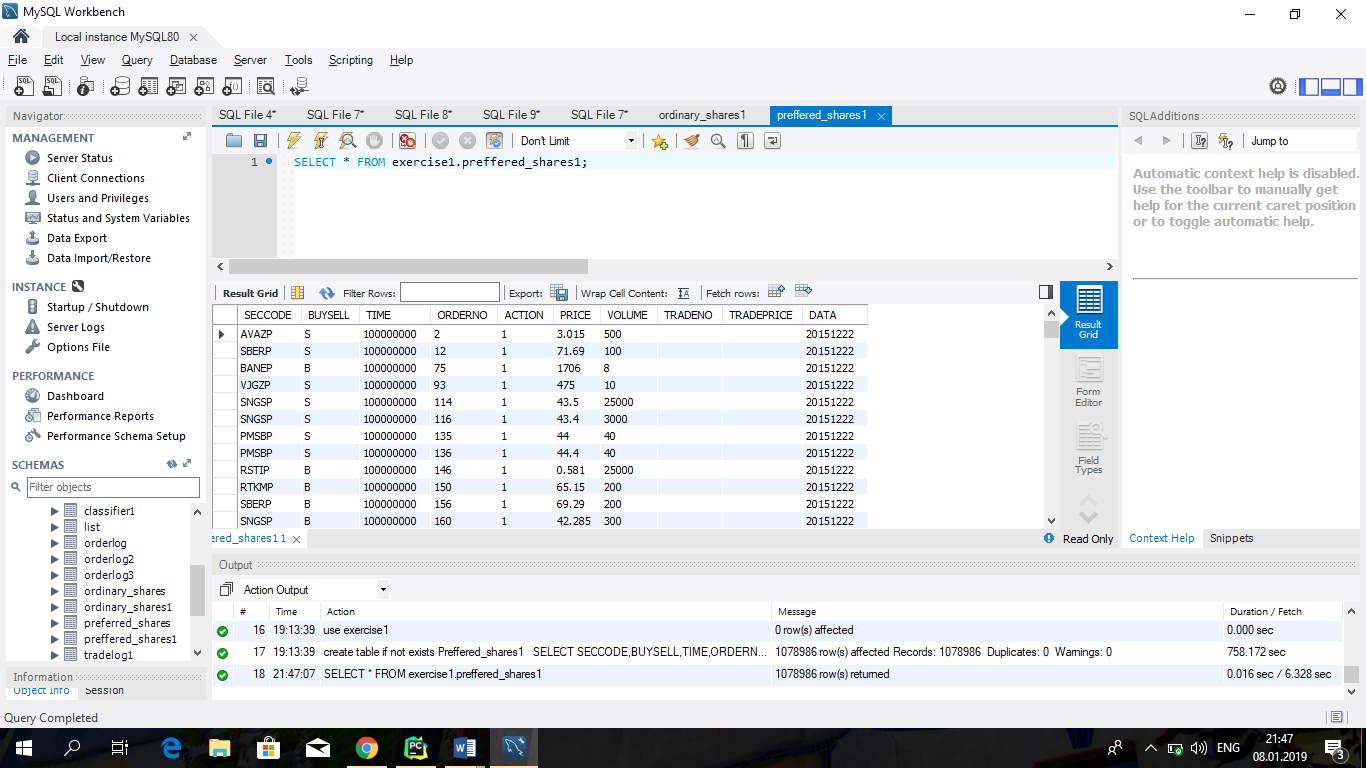
*SELECT SECCODE, BUYSELL,TIME,ORDERNO,ACTION,PRICE,VOLUME,TRADENO,TRADEPRICE,DATA*

*FROM Orderlog JOIN list*

*ON SECCODE=TRADE\_CODE*

*WHERE INSTRUMENT\_TYPE = 'Акция привилегированная'*

*Рисунок 7. Таблица привилегированных акций за 22.12.2015*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Для создания таблицы с заявками по обыкновенным акциям за 07.12.2015 используем следующий код:

*use exercise1;*

*create table if not exists Ordinary\_shares2*

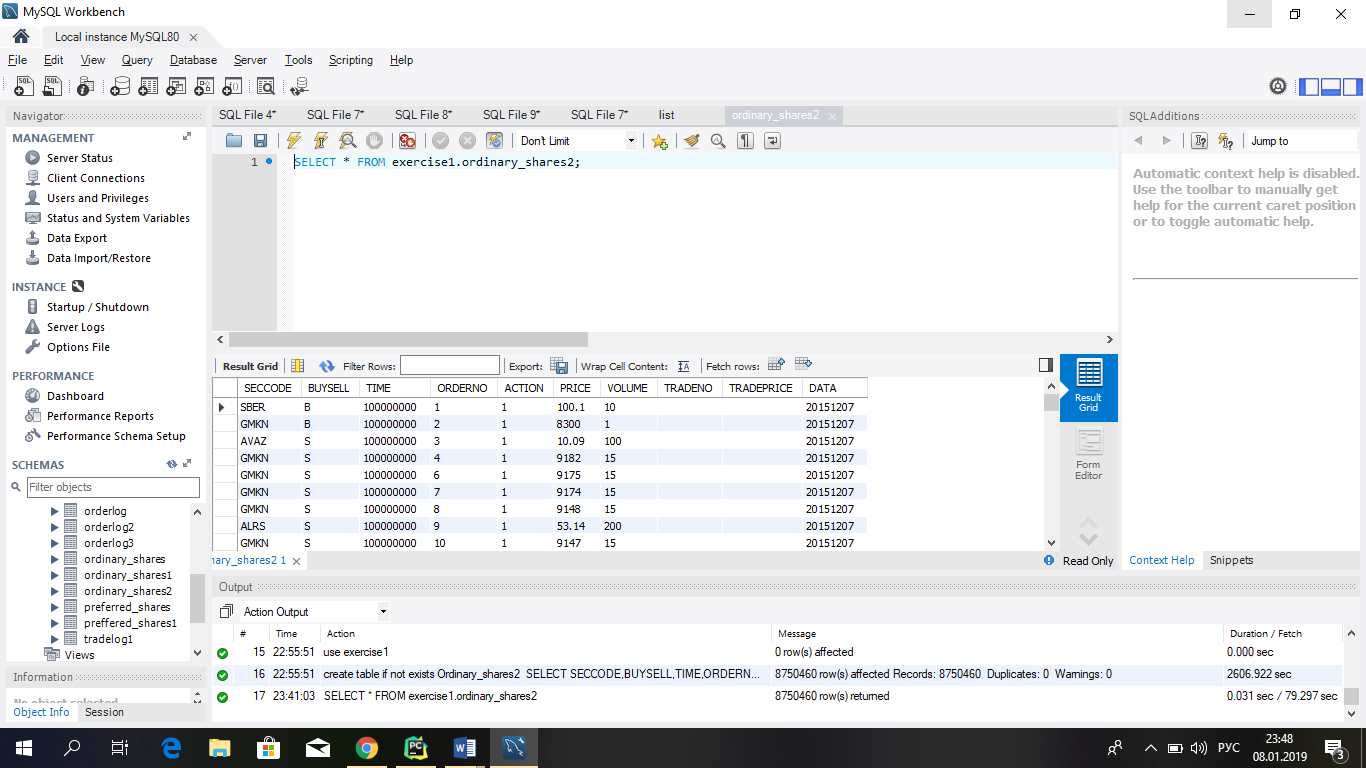
*SELECT SECCODE, BUYSELL,TIME,ORDERNO,ACTION,PRICE,VOLUME,TRADENO,TRADEPRICE,DATA*

*FROM Orderlog2 JOIN list*

*ON SECCODE=TRADE\_CODE*

*WHERE INSTRUMENT\_TYPE = 'Акция обыкновенная'*

*Рисунок 8. Таблица обыкновенных акций за 07.12.2015*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Для создания таблицы с заявками по привилегированным акциям за 07.12.2015 используем следующий код:

*use exercise1;*

*create table if not exists Preferred\_shares1*

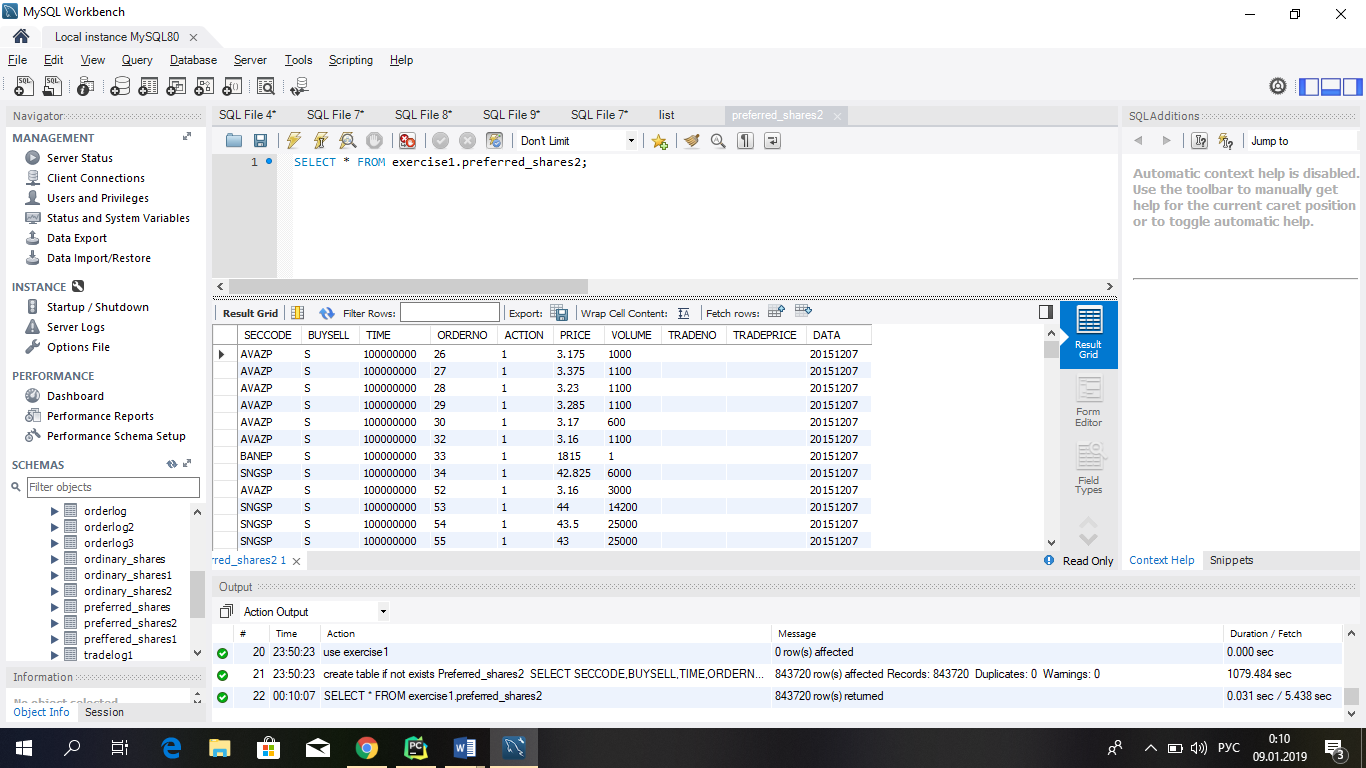
*SELECT SECCODE, BUYSELL,TIME,ORDERNO,ACTION,PRICE,VOLUME,TRADENO,TRADEPRICE,DATA*

*FROM Orderlog2 JOIN list*

*ON SECCODE=TRADE\_CODE*

*WHERE INSTRUMENT\_TYPE = 'Акция привилегированная'*

*Рисунок 9. Таблица обыкновенных акций за 07.12.2015*



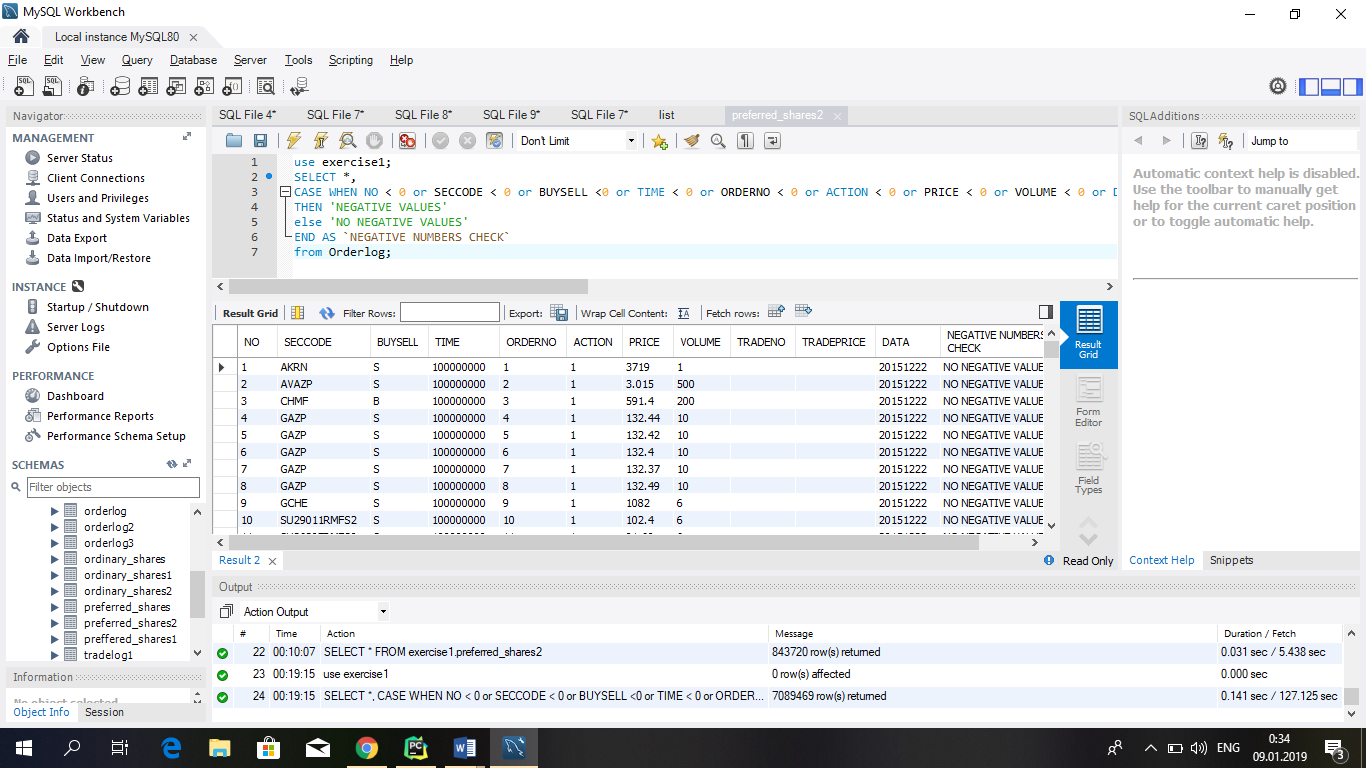
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

**Блок 2. Анализ и предобработка данных.**

В данном блоке посредством дальнейшей реализации файла script.sql в среде MySQL осуществляется выявление пропусков данных и ошибок, а также приводится описательная статистика используемых данных.

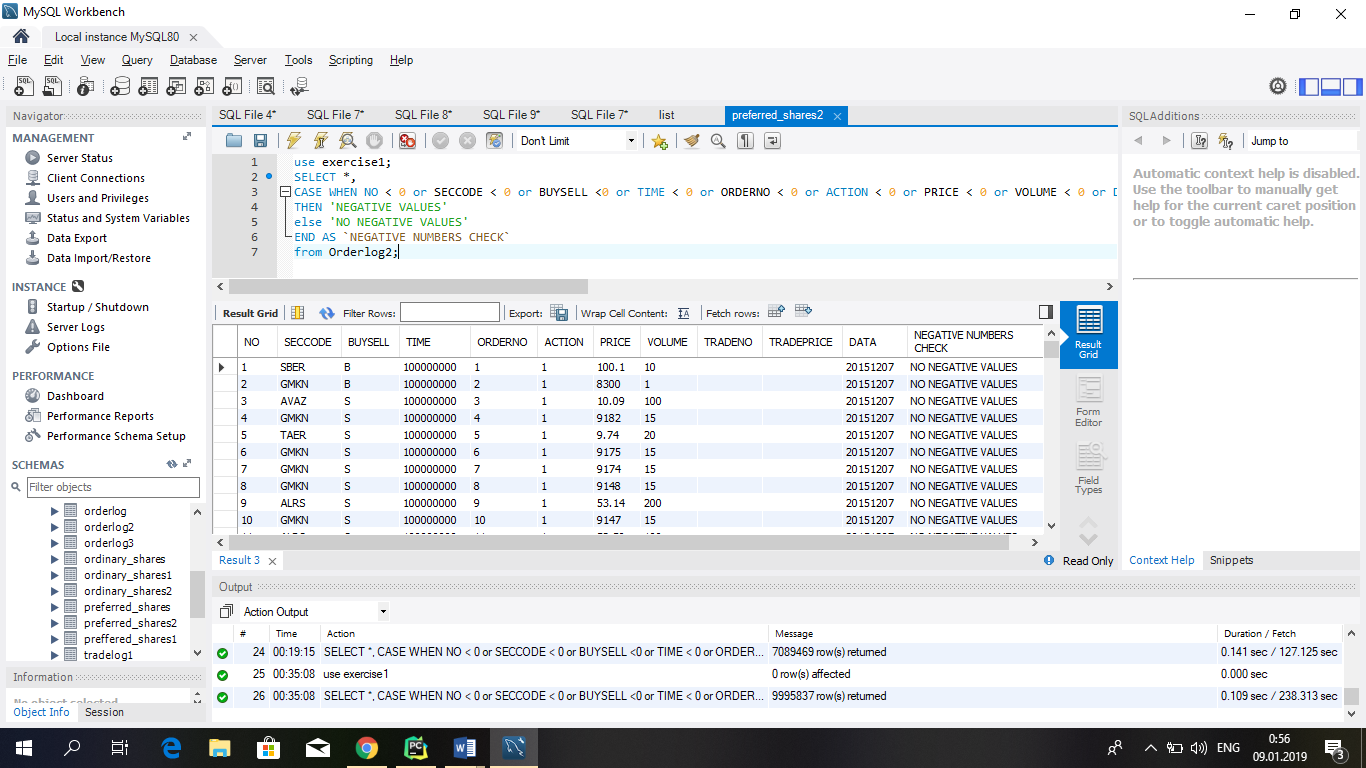
Для начала производится проверка данных на отрицательные значения в таблицах Orderlog (Рисунок 10) и Orderlog2 (Рисунок 11).

*Рисунок 10. Проверка данных на отрицательные значения в таблице «Orderlog».*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

*Рисунок 11. Проверка данных на отрицательные значения в таблице «Orderlog2».*

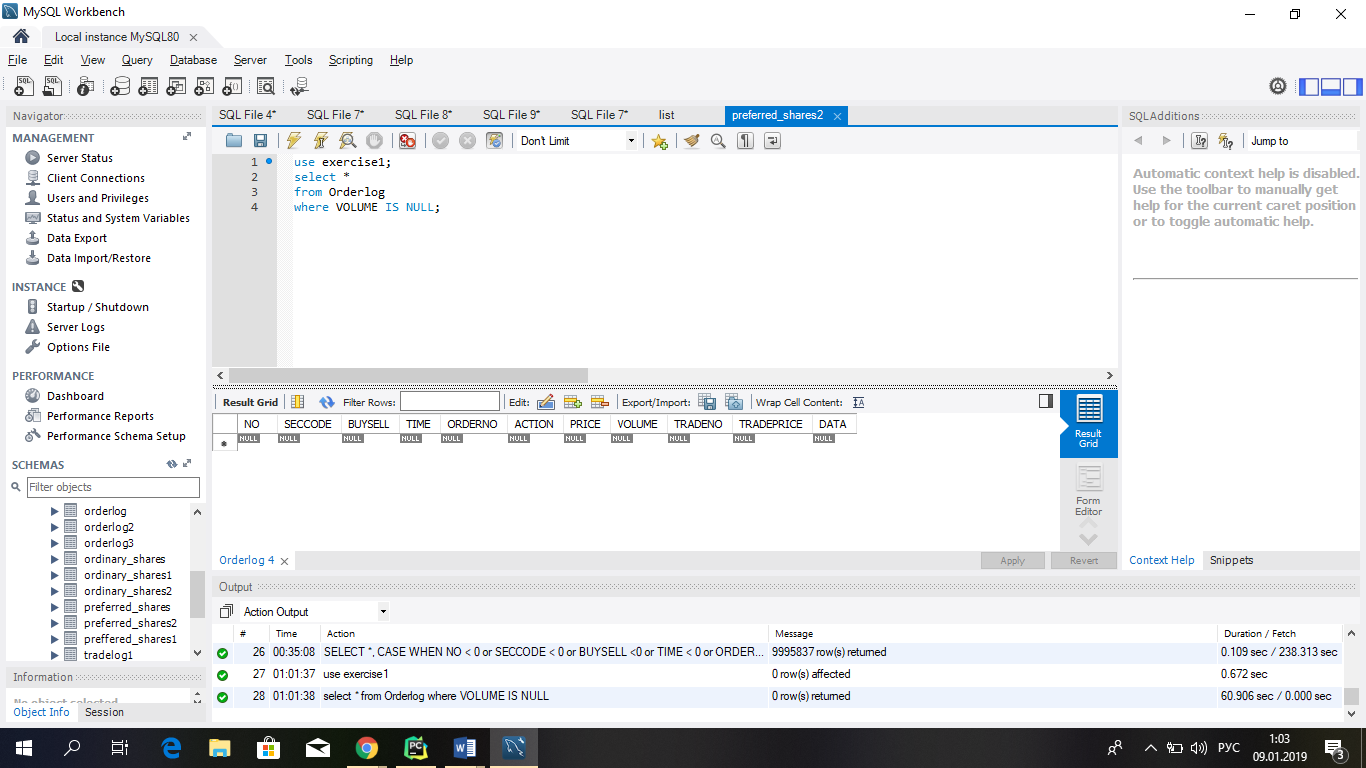


*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

После чего производится проверка данных из таблиц «Orderlog» (Рисунок 12) и «Orderlog2» (Рисунок 13) на наличие нулевых значений объёма посредством реализации кодов:

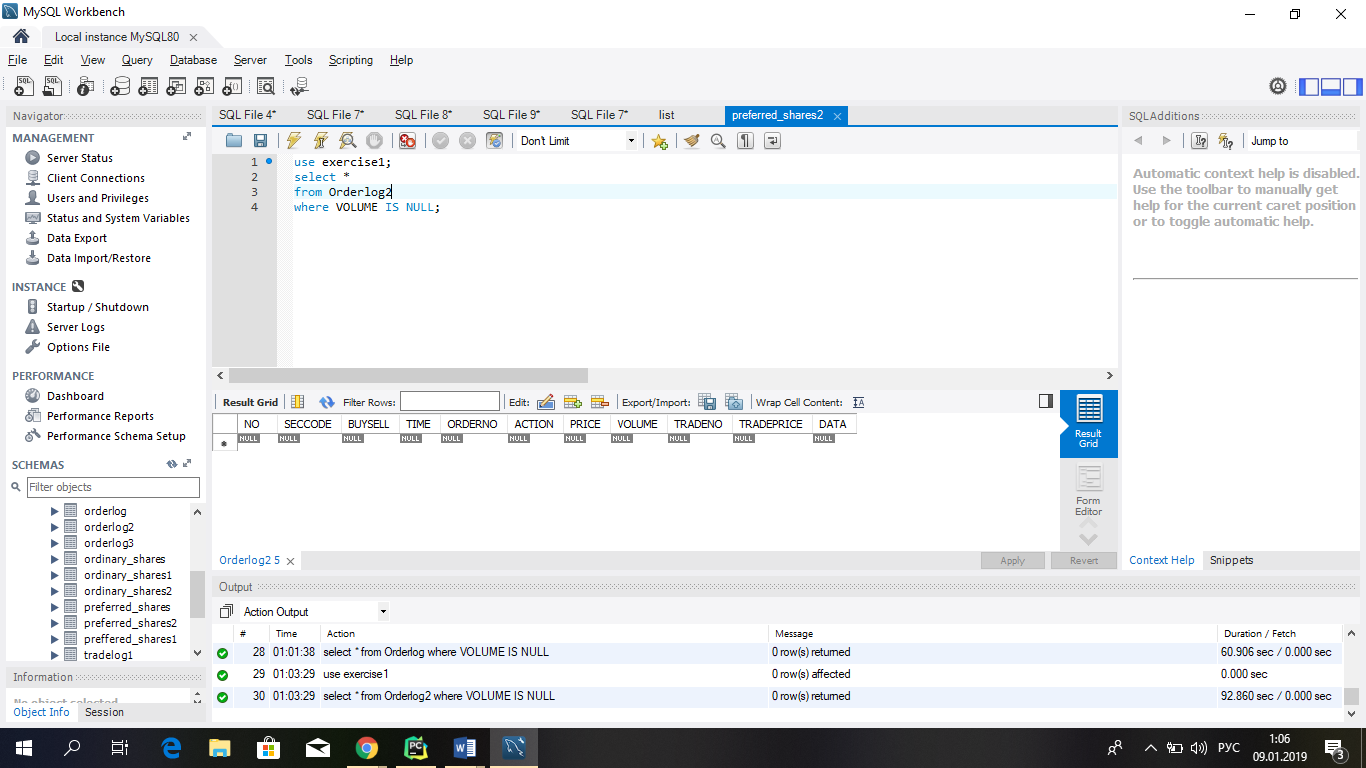
|  |  |
| --- | --- |
| *use exercise1;*  *select \**  *from Orderlog*  *where VOLUME IS NULL;* | *use exercise1;*  *select \**  *from Orderlog2*  *where VOLUME IS NULL;* |

*Рисунок 12. Проверка объемов на нулевые значения в таблице «Orderlog».*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

*Рисунок 13. Проверка объемов на нулевые значения в таблице «Orderlog2».*



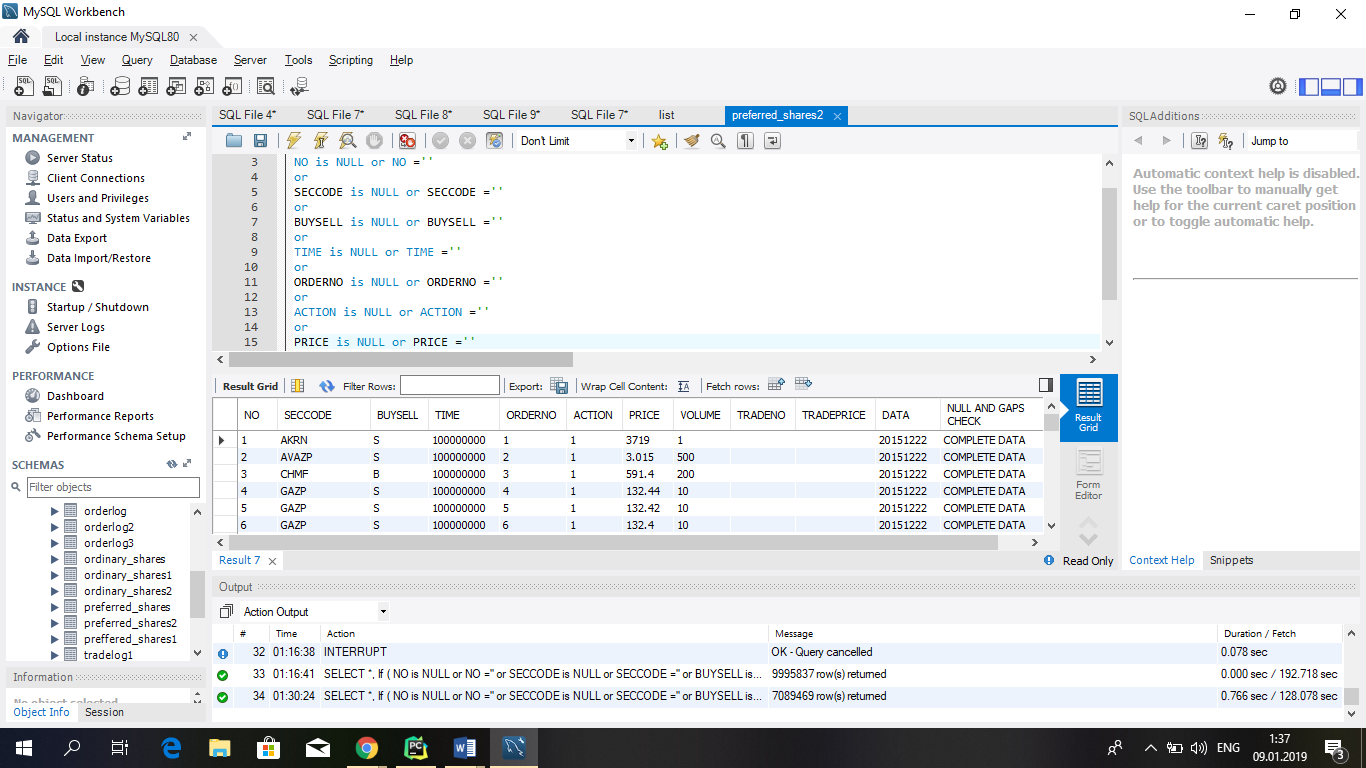
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Таким образом, можно заметить, что в используемых данных нулевых значений объема не было обнаружено.

Далее посредством реализации следующих кодов для таблиц «Orderlog» (Рисунок 14) и «Orderlog2» (Рисунок 15) производится проверка данных на полноту:

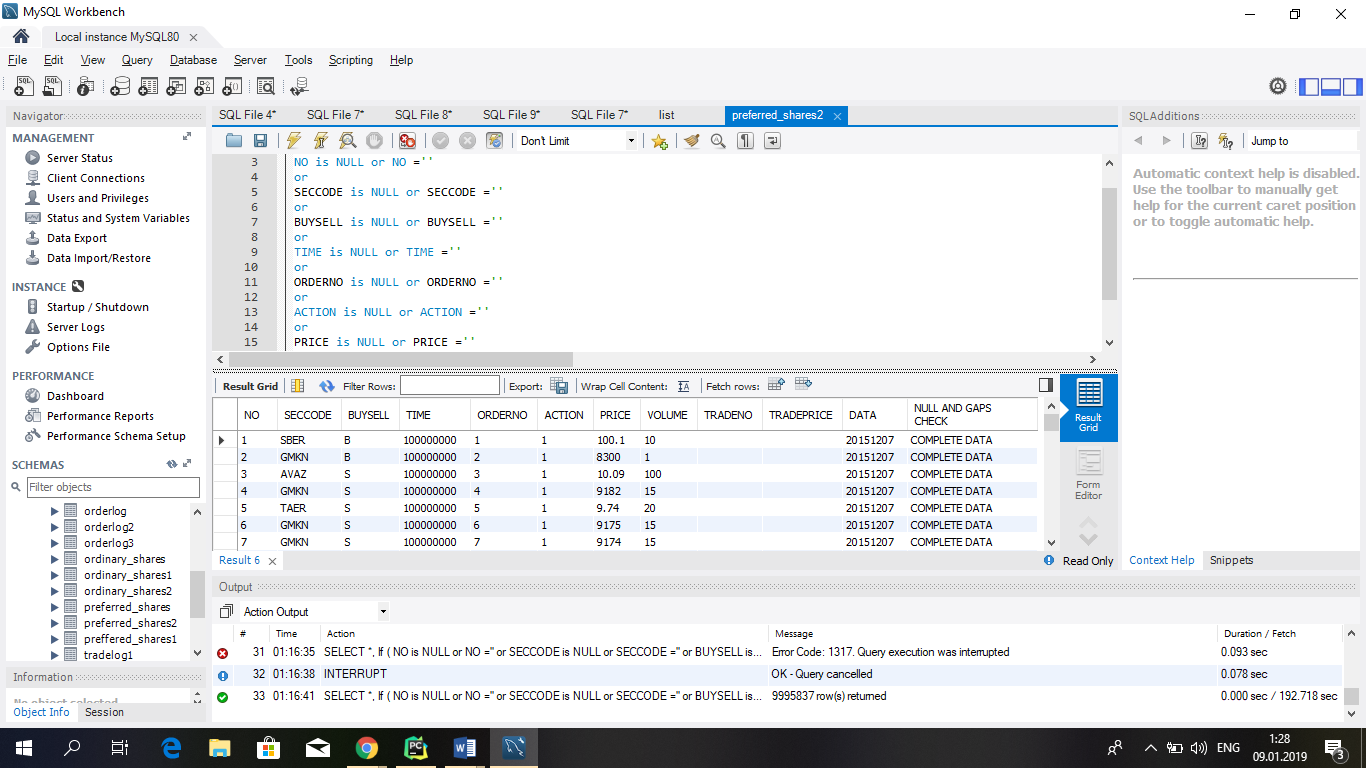
|  |  |
| --- | --- |
| *SELECT \*,*  *If (*  *NO is NULL or NO =''*  *or*  *SECCODE is NULL or SECCODE =''*  *or*  *BUYSELL is NULL or BUYSELL =''*  *or*  *TIME is NULL or TIME =''*  *or*  *ORDERNO is NULL or ORDERNO =''*  *or*  *ACTION is NULL or ACTION =''*  *or*  *PRICE is NULL or PRICE =''*  *or*  *VOLUME is NULL or VOLUME = '', 'DATA IS REQUIRED','COMPLETE DATA'*  *) as `NULL AND GAPS CHECK` from exercise1.Orderlog;* | *SELECT \*,*  *If (*  *NO is NULL or NO =''*  *or*  *SECCODE is NULL or SECCODE =''*  *or*  *BUYSELL is NULL or BUYSELL =''*  *or*  *TIME is NULL or TIME =''*  *or*  *ORDERNO is NULL or ORDERNO =''*  *or*  *ACTION is NULL or ACTION =''*  *or*  *PRICE is NULL or PRICE =''*  *or*  *VOLUME is NULL or VOLUME = '', 'DATA IS REQUIRED','COMPLETE DATA'*  *) as `NULL AND GAPS CHECK` from exercise1.Orderlog2;* |

*Рисунок 14. Проверка данных на полноту в таблице «Orderlog».*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

*Рисунок 15. Проверка данных на полноту в таблице «Orderlog2».*



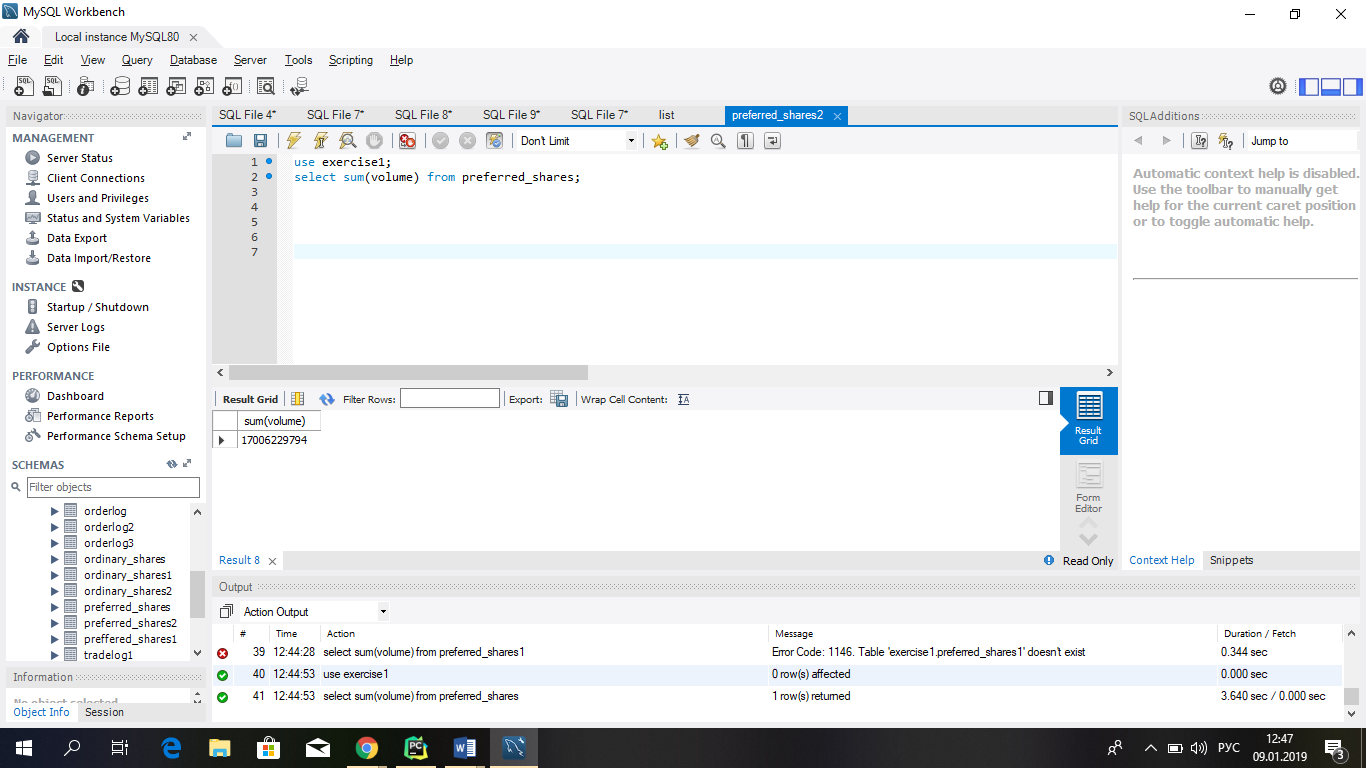
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Далее производится вывод описательных статистик. Как пример, выведем для таблицы по заявкам с облигациями за 22.12.2015 общий объем торгов с помощью следующего кода:

*use exercise1;*

*select sum(volume) from preferred\_shares;*

*Рисунок 16. Общий объем торгов по облигациям.*



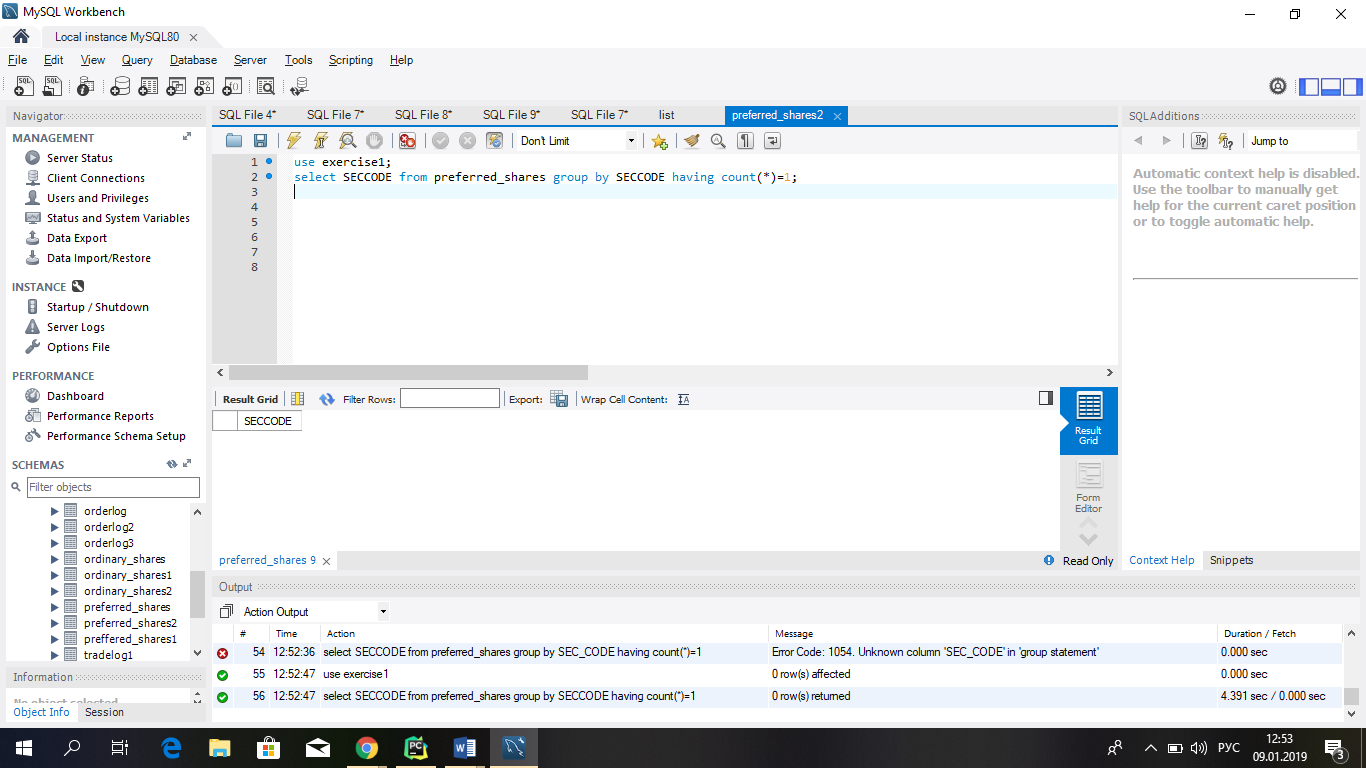
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Далее найдем количество уникальных облигаций за 22.12.2015 с помощью запроса:

*use exercise1;*

*select SECCODE from preferred\_shares group by SECCODE having count(\*)=1;*

*Рисунок 17. Количество уникальных облигаций.*



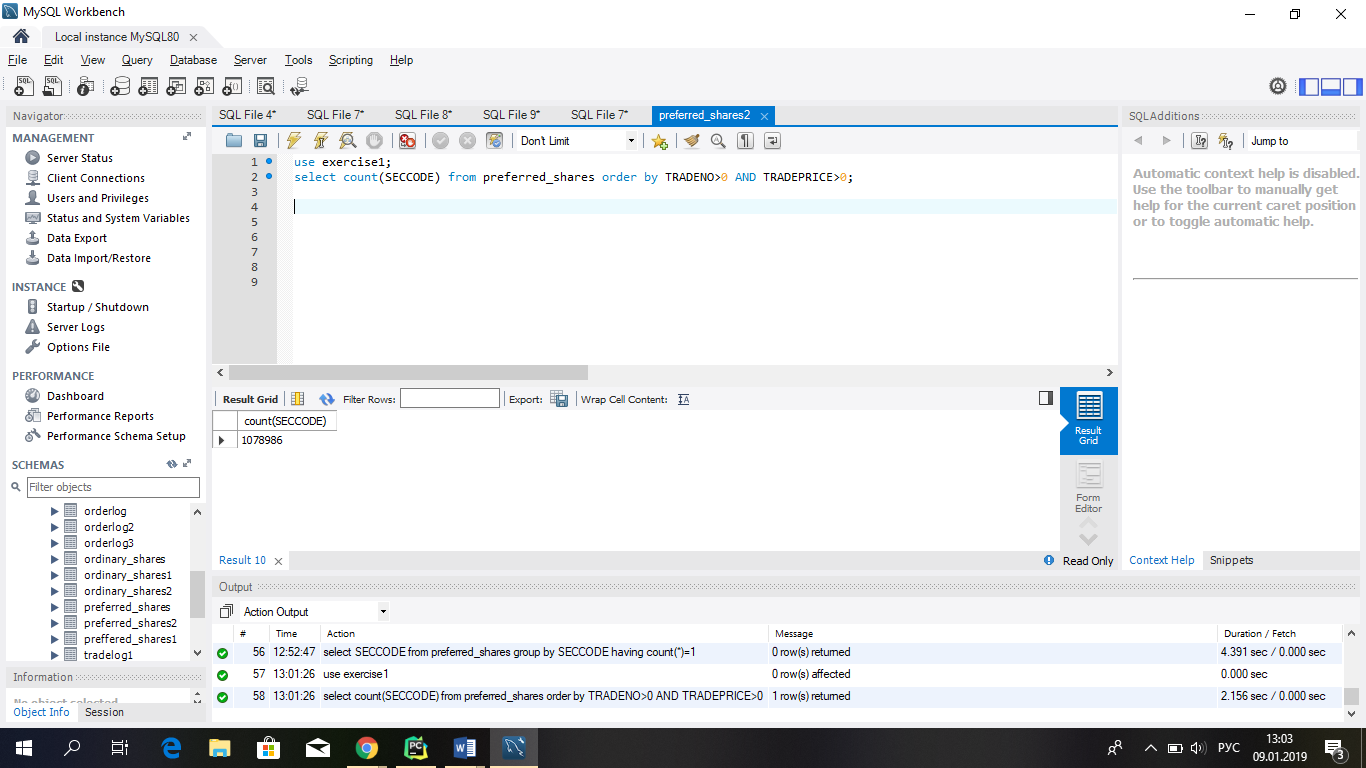
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Посчитаем общее количество сделок по привилегированным акциям за 22.12.2015 с помощью запроса:

*use exercise1;*

*select count(SECCODE) from preferred\_shares order by TRADENO>0 AND TRADEPRICE>0;*

*Рисунок 18. Общее количество сделок по привилегированным акциям.*



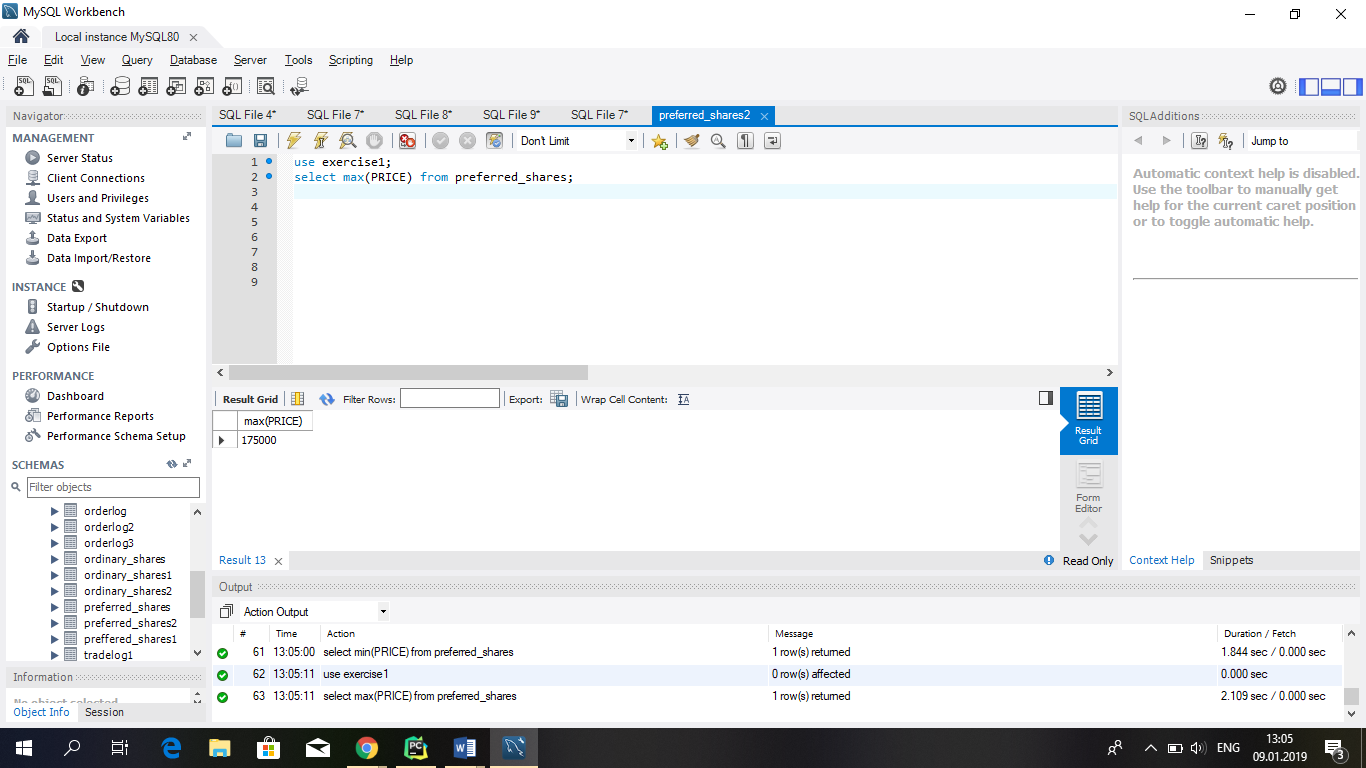
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Выберем максимальную цену по привилегированным акциям за день 22.12.2015 с помощью запроса:

*use exercise1;*

*select max(PRICE) from preferred\_shares;*

*Рисунок 19. Максимальная цена по привилегированным акциям.*



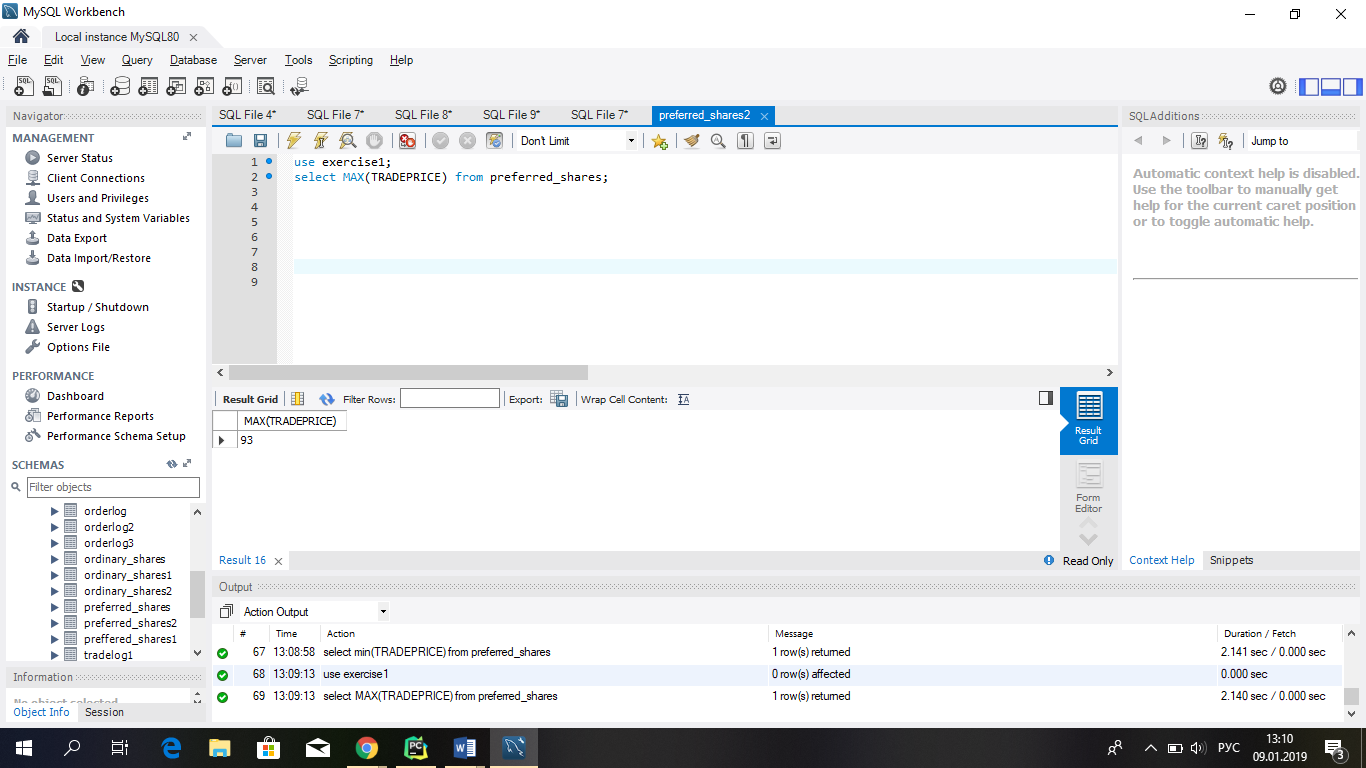
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Найдем, к примеру максимальную цену по сделкам с привилегированными акциями за 22.12.2015 с помощью запроса:

*use exercise1;*

*select MAX(TRADEPRICE) from preferred\_shares;*

*Рисунок 20. Максимальная цена по сделкам с привилегированными акциями.*



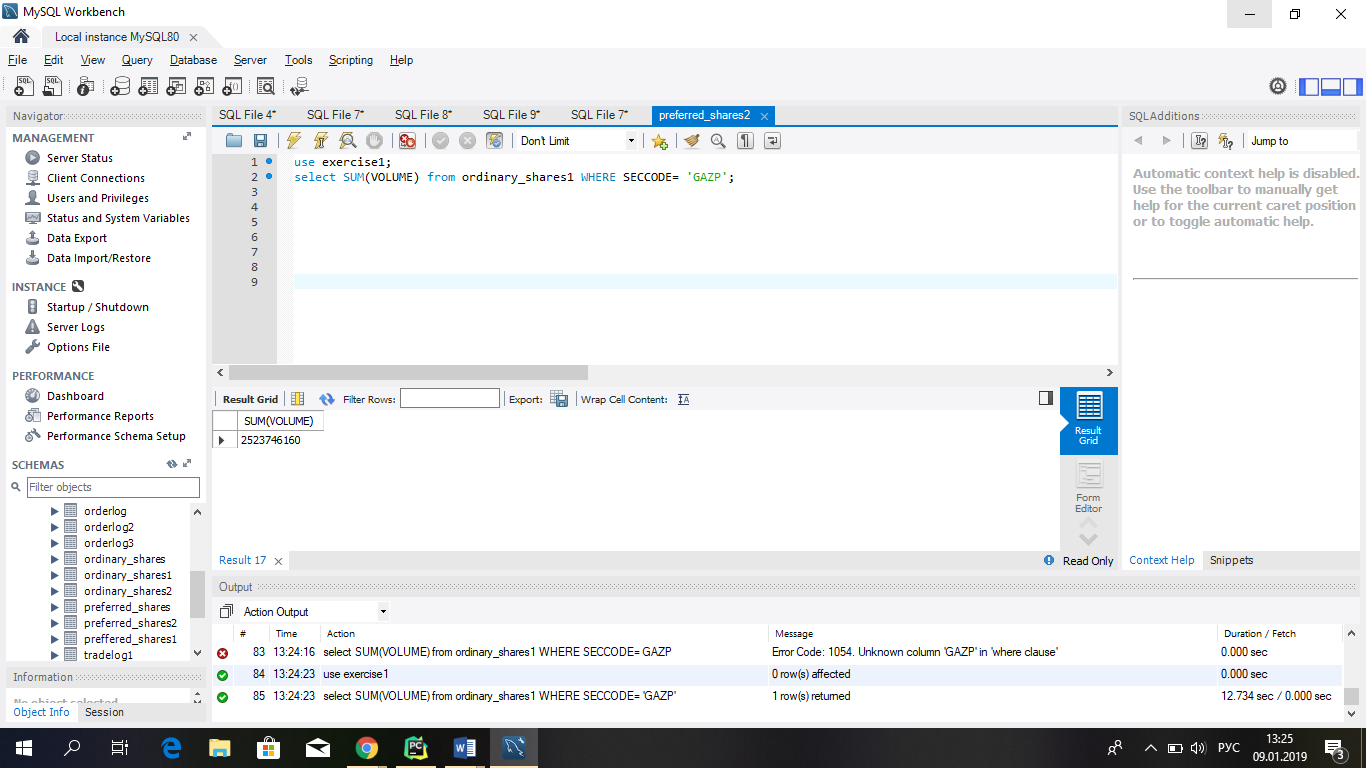
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Выведем описательные статистики по отдельно взятой бумаге (к примеру акциям Газпрома) за 22.12.2015. Найдем общий объем торгов за выбранный день по акциям Газпрома с помощью запроса:

*use exercise1;*

*select SUM(VOLUME) from ordinary\_shares1 WHERE SECCODE= 'GAZP';*

*Рисунок 21. Общий объем торгов за день по акции GAZP.*



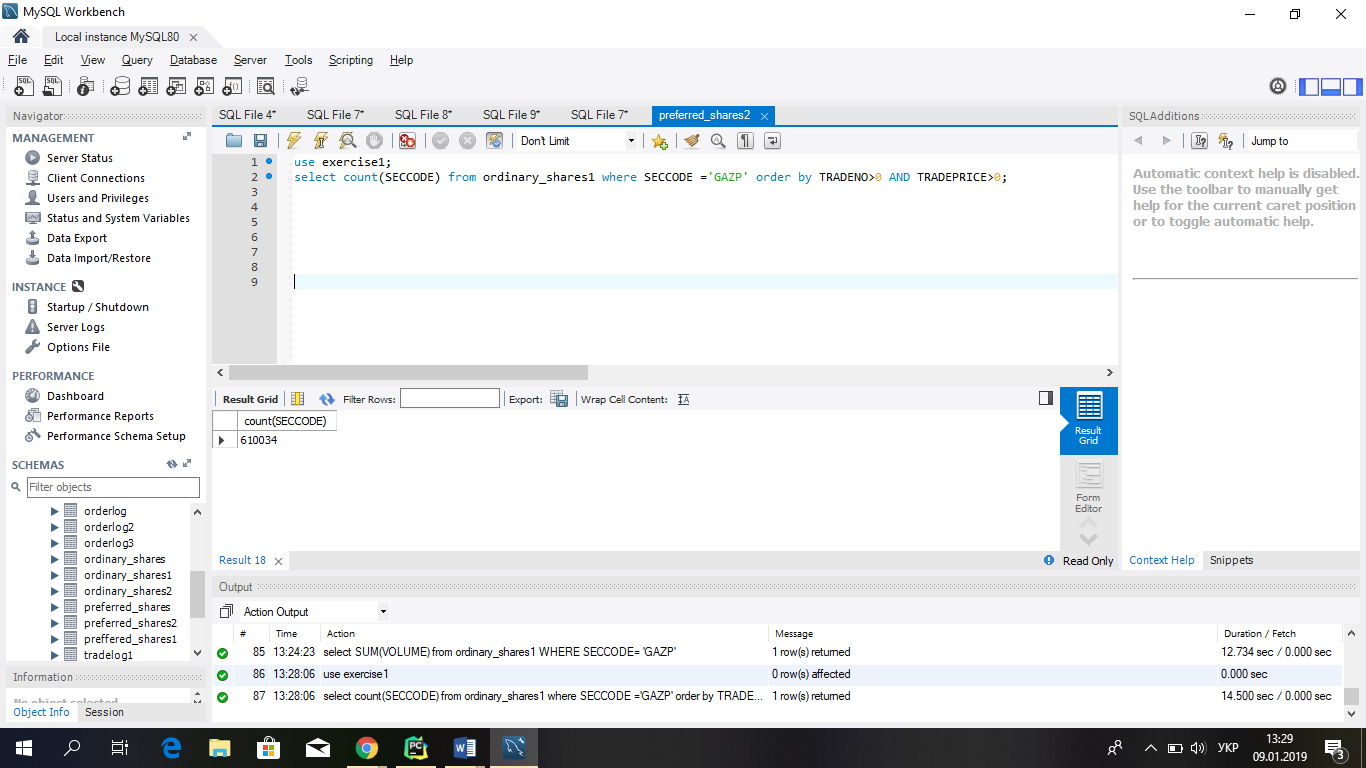
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Найдем общее количество сделок по акциям Газпрома за 22.12.2015 с помощью следующего запроса:

*use exercise1;*

*select count(SECCODE) from ordinary\_shares1 where SECCODE ='GAZP' order by TRADENO>0 AND TRADEPRICE>0;*

*Рисунок 22. Общее количество сделок за день по акции GAZP.*



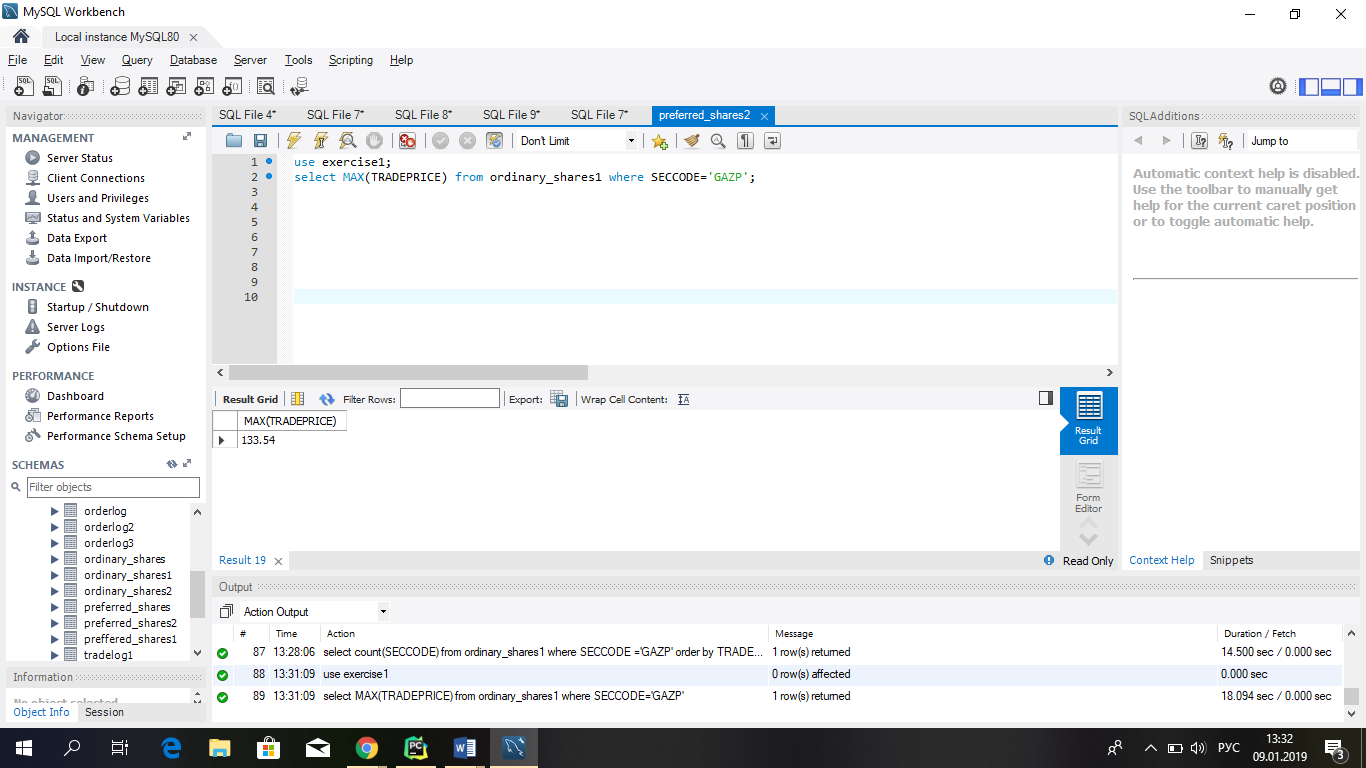
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Найдем максимальную цену сделки по акциям Газпрома за 20151222 с помощью запроса:

*use exercise1;*

*select MAX(TRADEPRICE) from ordinary\_shares1 where SECCODE='GAZP';*

*Рисунок 23. Максимальная цена сделки по акции GAZP.*



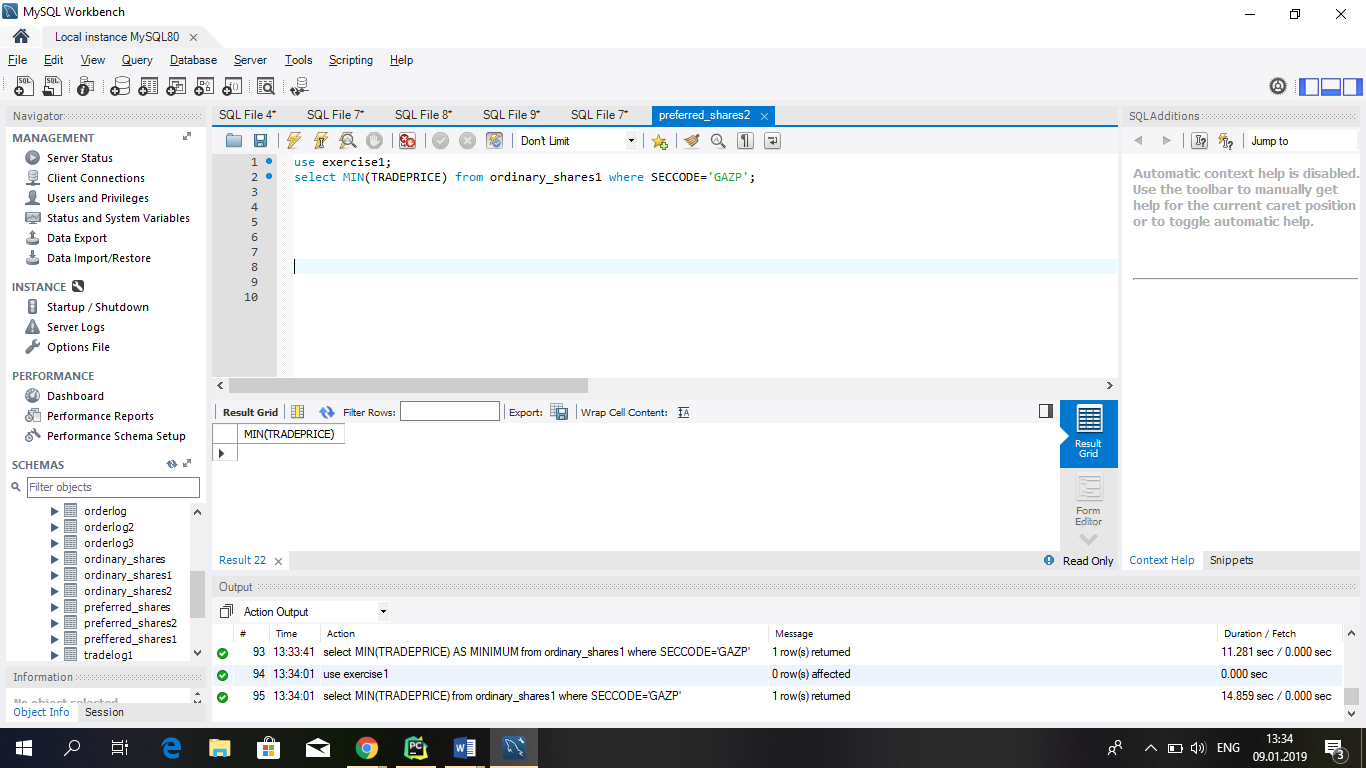
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Найдем минимальную цену сделки по акциям Газпрома за 20151222 с помощью запроса:

*use exercise1;*

*select MIN(TRADEPRICE) from ordinary\_shares1 where SECCODE='GAZP';*

*Рисунок 24. Минимальная цена сделки по акции GAZP.*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Найдем минимальный объем исполненной рыночной заявки за 20151222. Данные заявки имеют цену равную 0, поэтому запрос выглядит следующим образом:

*use exercise1;*

*select MIN(VOLUME) from ordinary\_shares1 where SECCODE='GAZP' ORDER BY PRICE=0;*

*Рисунок 25. Минимальный объем исполненной рыночной заявки по акции GAZP.*



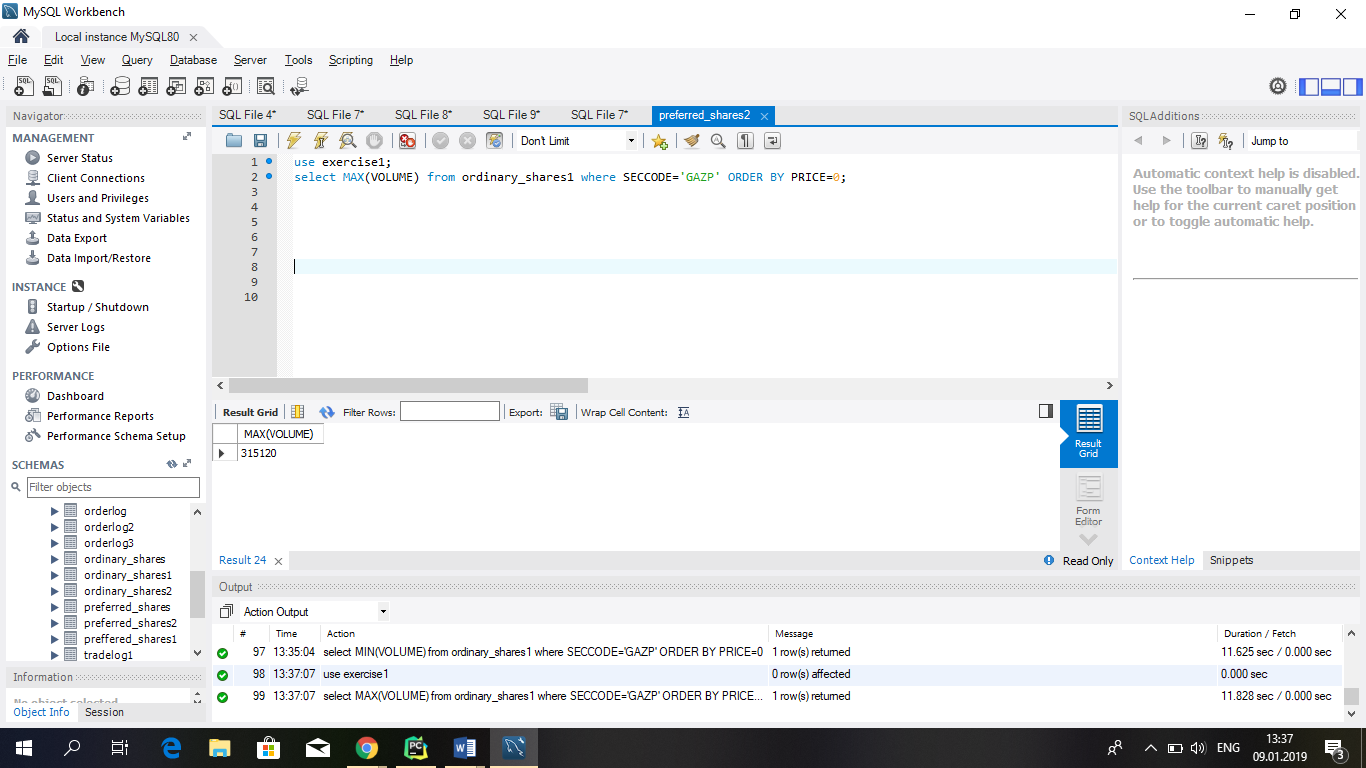
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Найдем максимальный объем исполненной рыночной заявки за 20151222. Данные заявки имеют цену равную 0, поэтому запрос выглядит следующим образом:

*use exercise1;*

*select MAX(VOLUME) from ordinary\_shares1 where SECCODE='GAZP' ORDER BY PRICE=0;*

*Рисунок 26. Максимальный объем исполненной рыночной заявки по акции GAZP.*



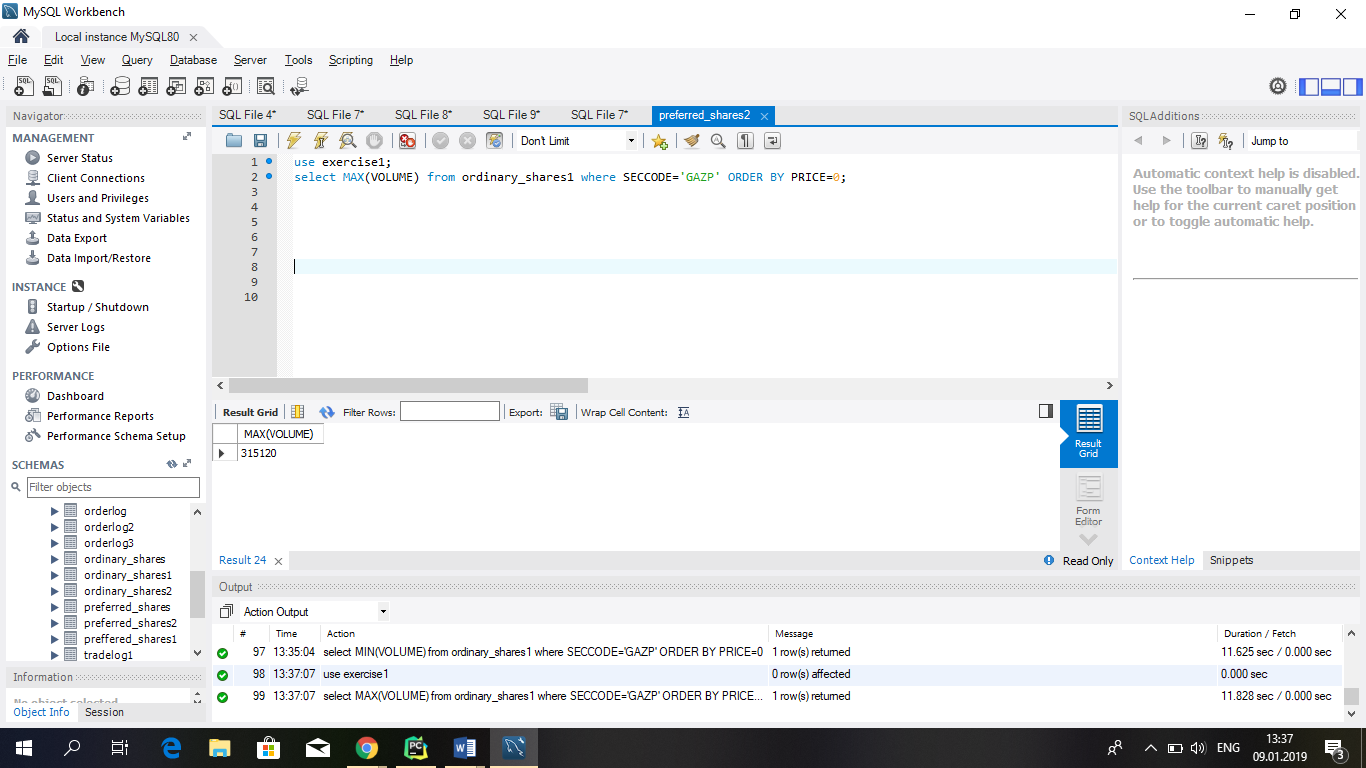
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Найдем максимальный объем лимитированной заявки по Газпрому за 20151222. Данные заявки имеют цену больше 0, поэтому запрос выглядит следующим образом:

*use exercise1;*

*select MAX(VOLUME) from ordinary\_shares1 where SECCODE='GAZP' ORDER BY PRICE>0 AND TRADEPRICE>0 AND TRADENO>0;*

*Рисунок 27. Максимальный объем лимитированной заявки по акции GAZP.*



*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

Найдем минимальный объем исполненной лимитированной заявки по Газпрому за 20151222. Данные заявки имеют цену больше 0, поэтому запрос выглядит следующим образом:

use exercise1;

select min(VOLUME) from ordinary\_shares1 where SECCODE='GAZP' ORDER BY PRICE>0 AND TRADEPRICE>0 AND TRADENO>0;

*Рисунок 28. Минимальный объем лимитированной заявки по акции GAZP.*



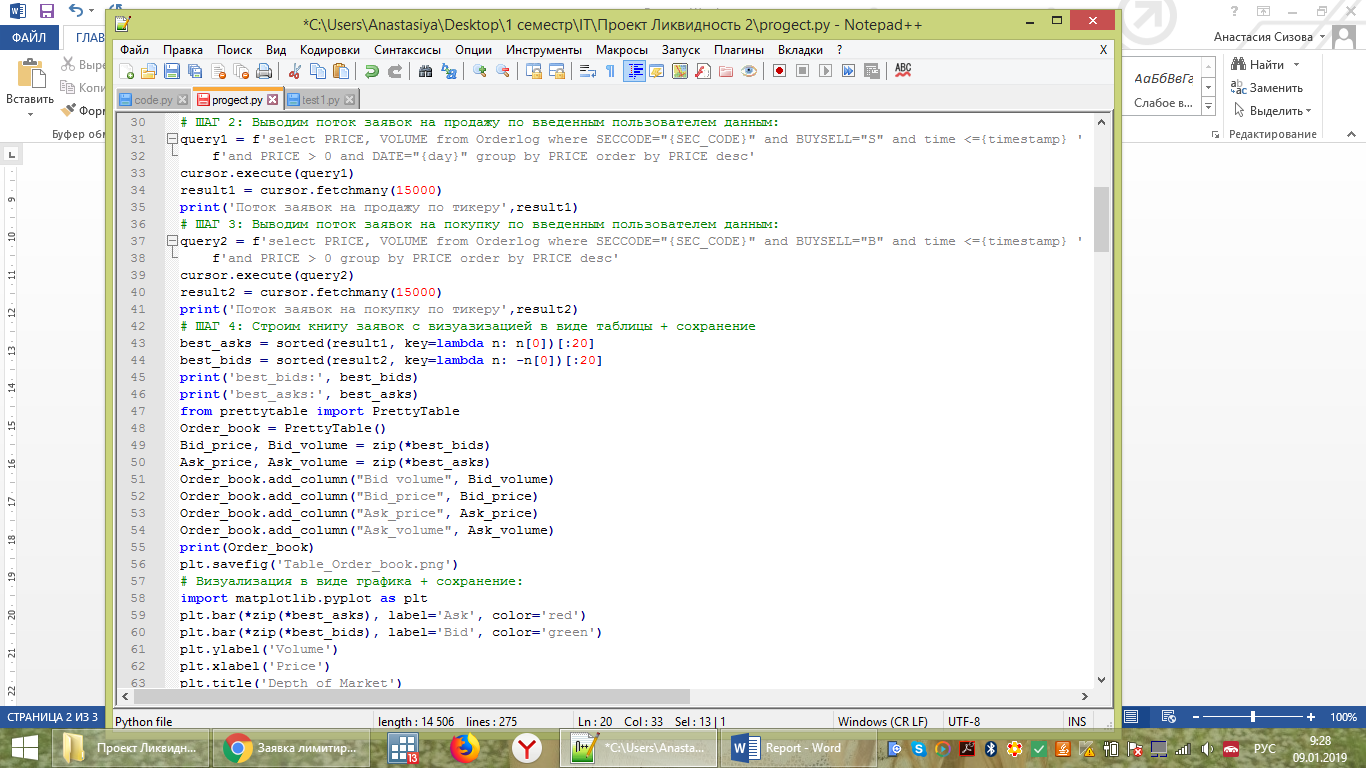
*Рисунок составлен авторами проекта при помощи MySQL Workbench 8.0.13.*

**Блок 3. Построение книги лимитированных заявок**.

В данном блоке посредством исполнения файла project.py осуществляется построение книги лимитированных заявок для отдельной ценной бумаги в определённый момент времени.

Код дает возможность пользователю установить свои параметры: дату (на выбор из представленных), тикер ценной бумаги, момент времени (в пределах торгового дня). По результату исполнения указанной части кода выводится итоговая таблица и график с указанием цены MID (среднее лучших цен покупки и продажи), бид-аск спрэда и суммарного объема на 5 лучших ценах покупки и продажи.

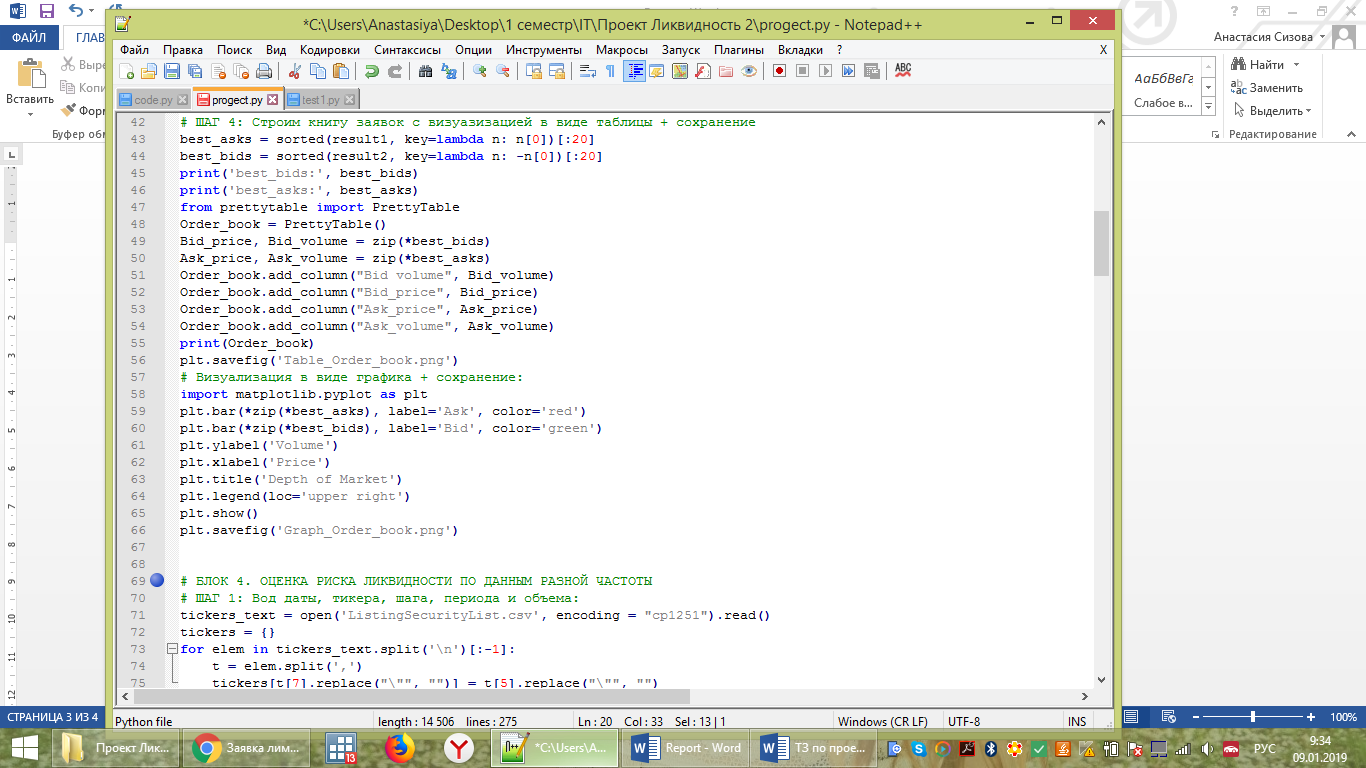
*Рисунок 29. Код, отвечающий за выполнение блока 3.*



*Рисунок составлен авторами проекта при помощи Notepad++.*

Далее предоставляется возможность визуализировать полученную книгу в виде графика с указанием цены MID (среднее лучших цен покупки и продажи), бид-аск спрэда и суммарного объема на 5 лучших ценах покупки и продажи. Также пользователю предлагается выгрузить таблицу и график в формате .png.

*Рисунок 30. Код, отвечающий за графическую визуализацию.*



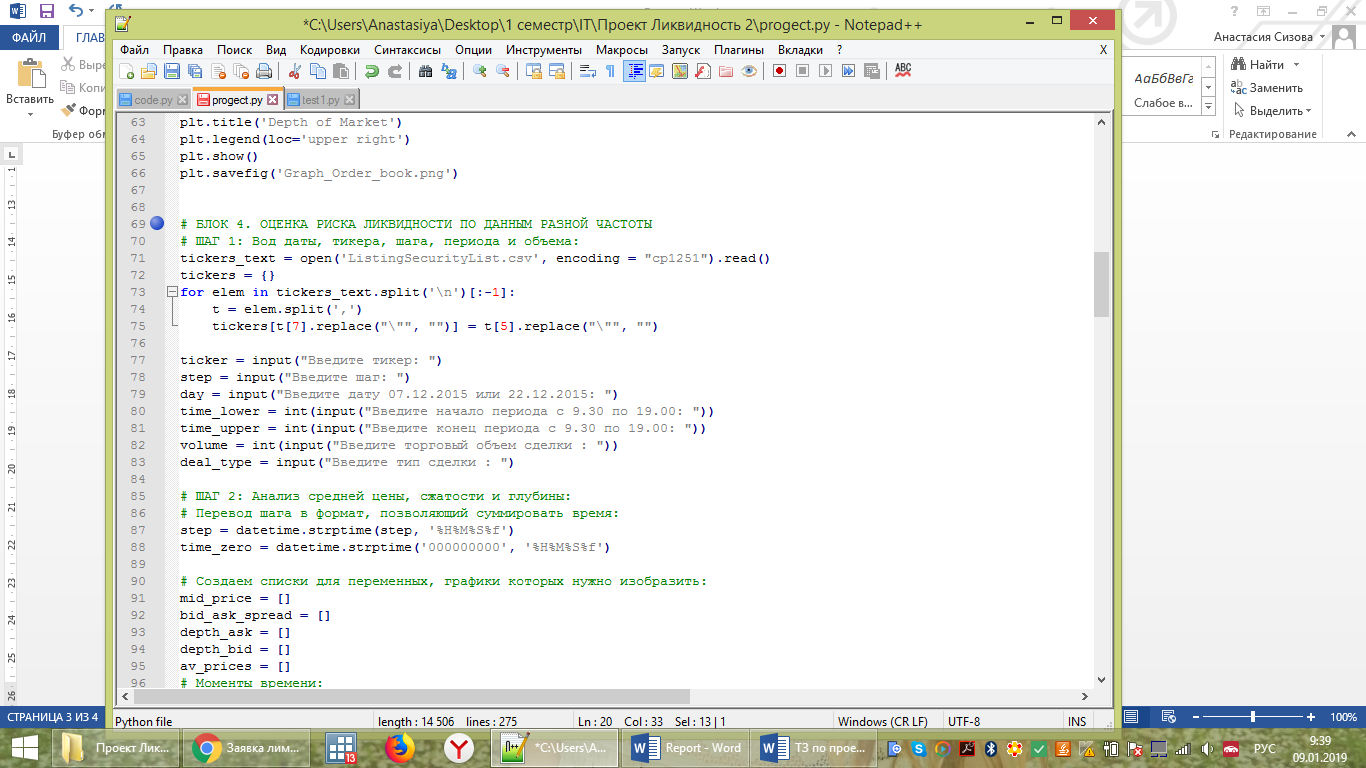
*Рисунок составлен авторами проекта при помощи Notepad++.*

**Блок 4. Оценка риска ликвидности по данным разной частоты.**

В данном блоке посредством исполнения файла project.py происходит оценка риска ликвидности через построение и анализ временных рядов показателей ликвидности и измерение риска для задаваемого пользователем объема.

Построение временных рядов показателей ликвидности *(средняя цена [MID]; сжатость [бид-аск спрэд]; глубину рынка [объем на лучших ценах покупки и продажи]; суммарный объем на 5 лучших ценах покупки и продажи; среднюю стоимость покупки/продажи целевого объема)* производится по задаваемым пользователем: тикеру, дате (на выбор из представленных), внутридневному периоду (в рамках торгового дня), частоте временного ряда и целевому объему (в штуках).

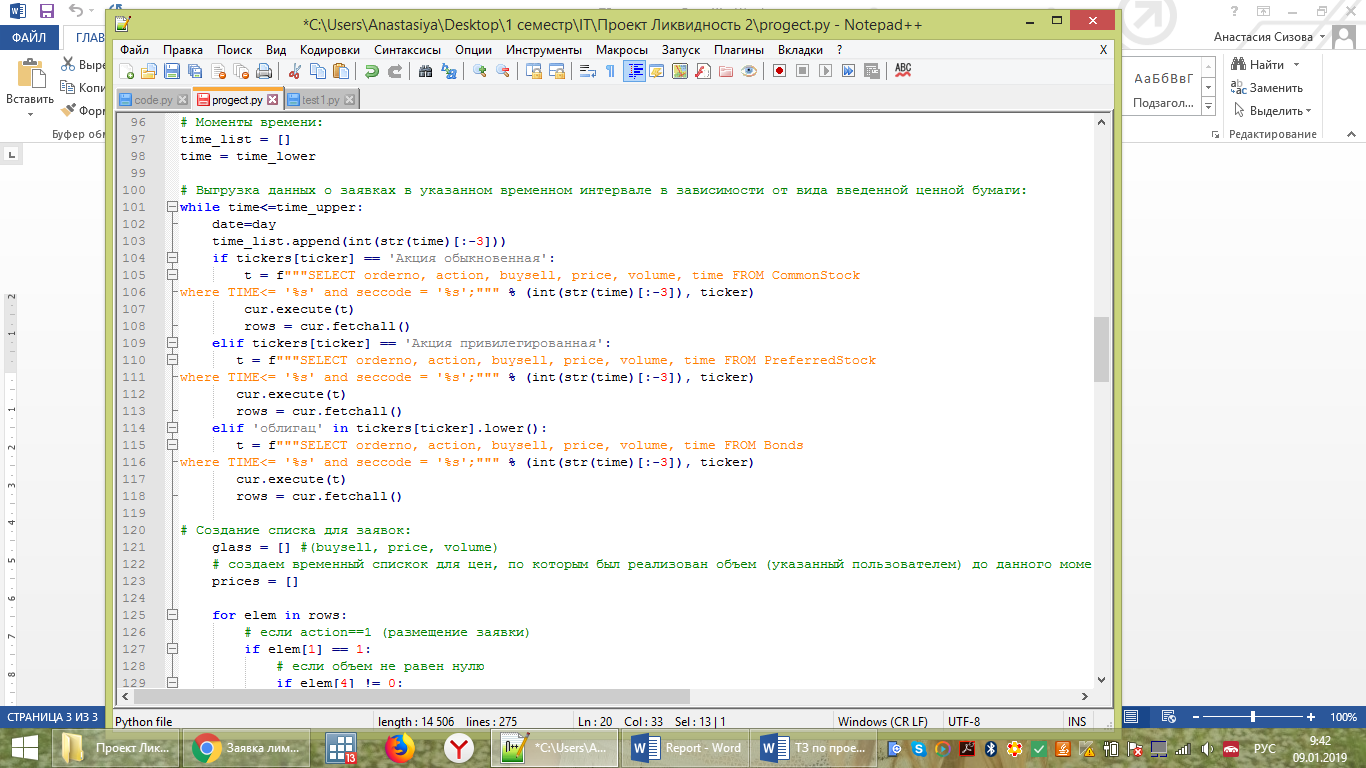
*Рисунок 31. Код, отвечающий за предложение пользователю задать параметры.*



*Рисунок составлен авторами проекта при помощи Notepad++.*

Далее производится выгрузка данных по заявкам в зависимости от типа ценной бумаги, заданной пользователем:

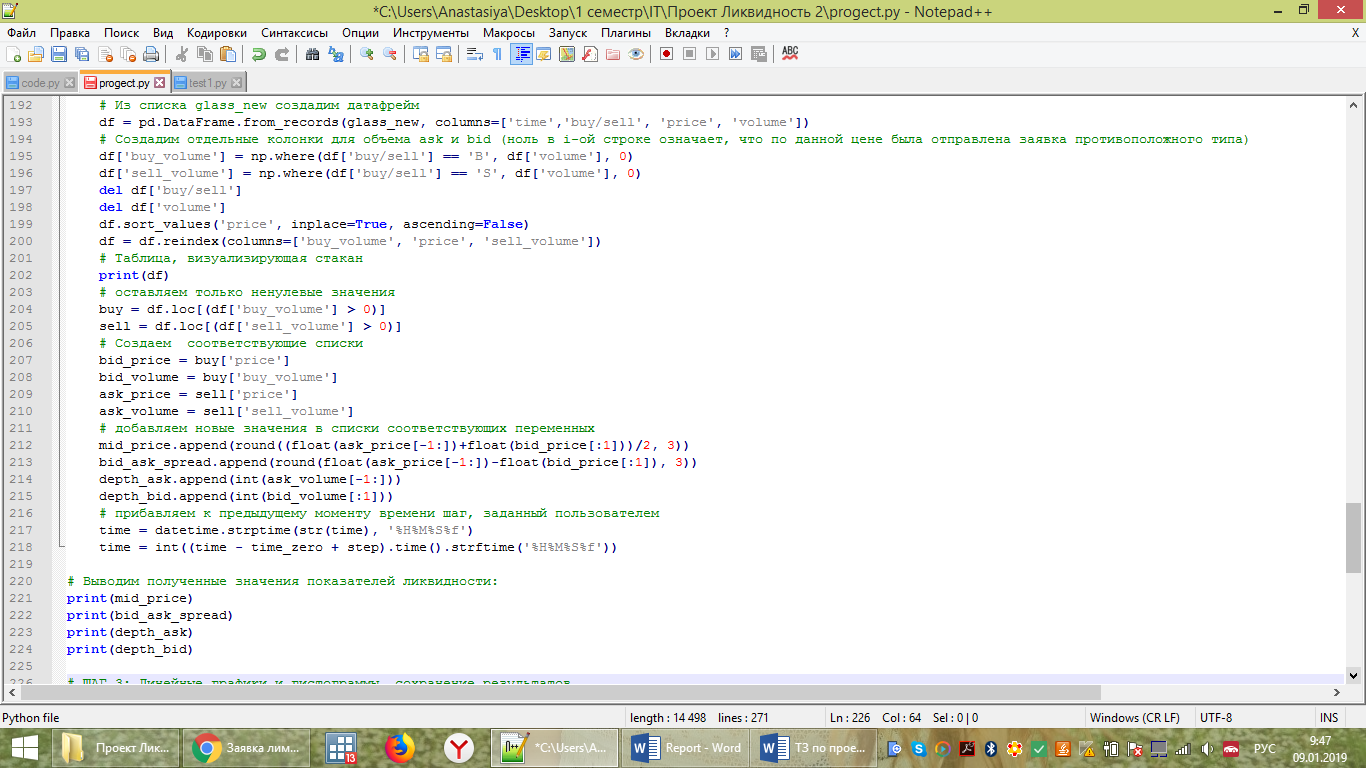
*Рисунок 32. Код, отвечающий за выгрузку данных по заявкам.*



*Рисунок составлен авторами проекта при помощи Notepad++.*

После чего происходит создание списка заявок, на основе которого рассчитываются необходимые показатели ликвидности и выводятся на экран:

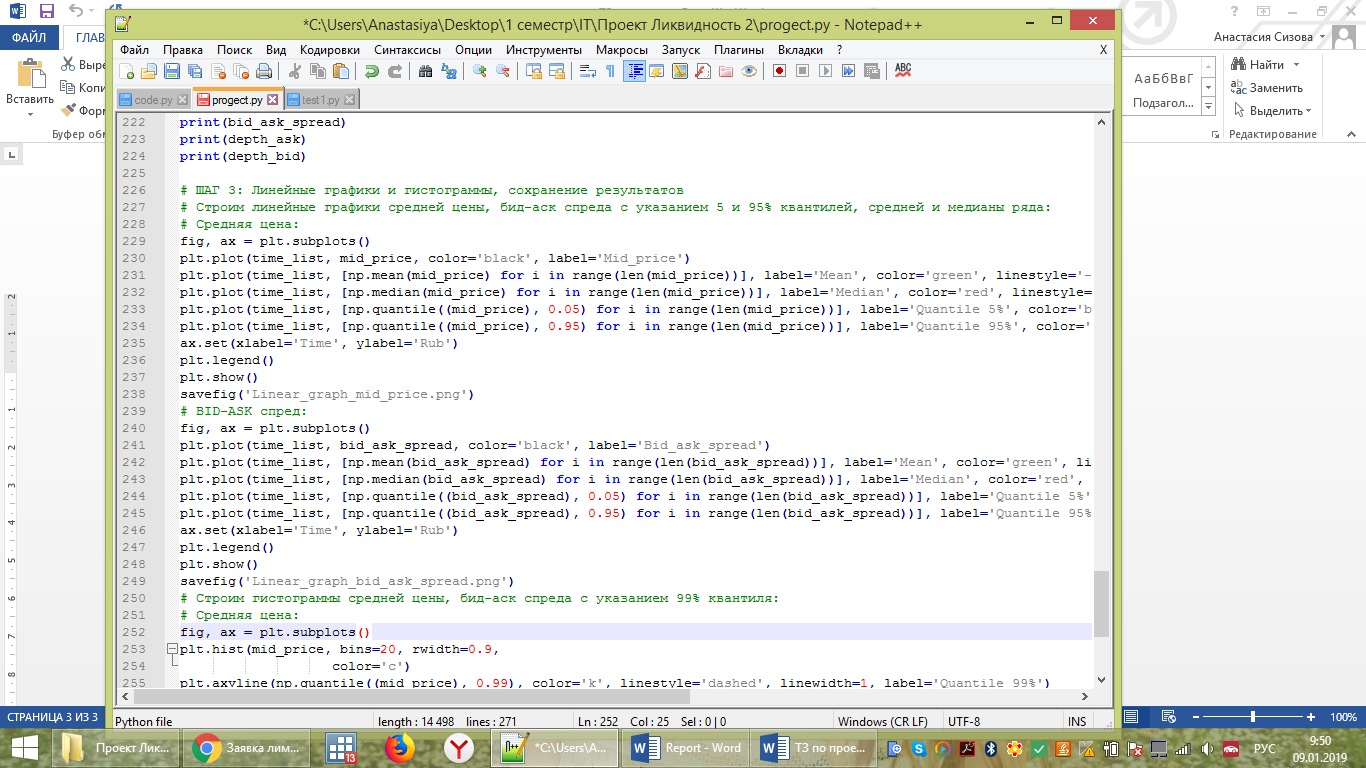
*Рисунок 33. Код, отвечающий расчет показателей ликвидности и вывод результатов.*



*Рисунок составлен авторами проекта при помощи Notepad++.*

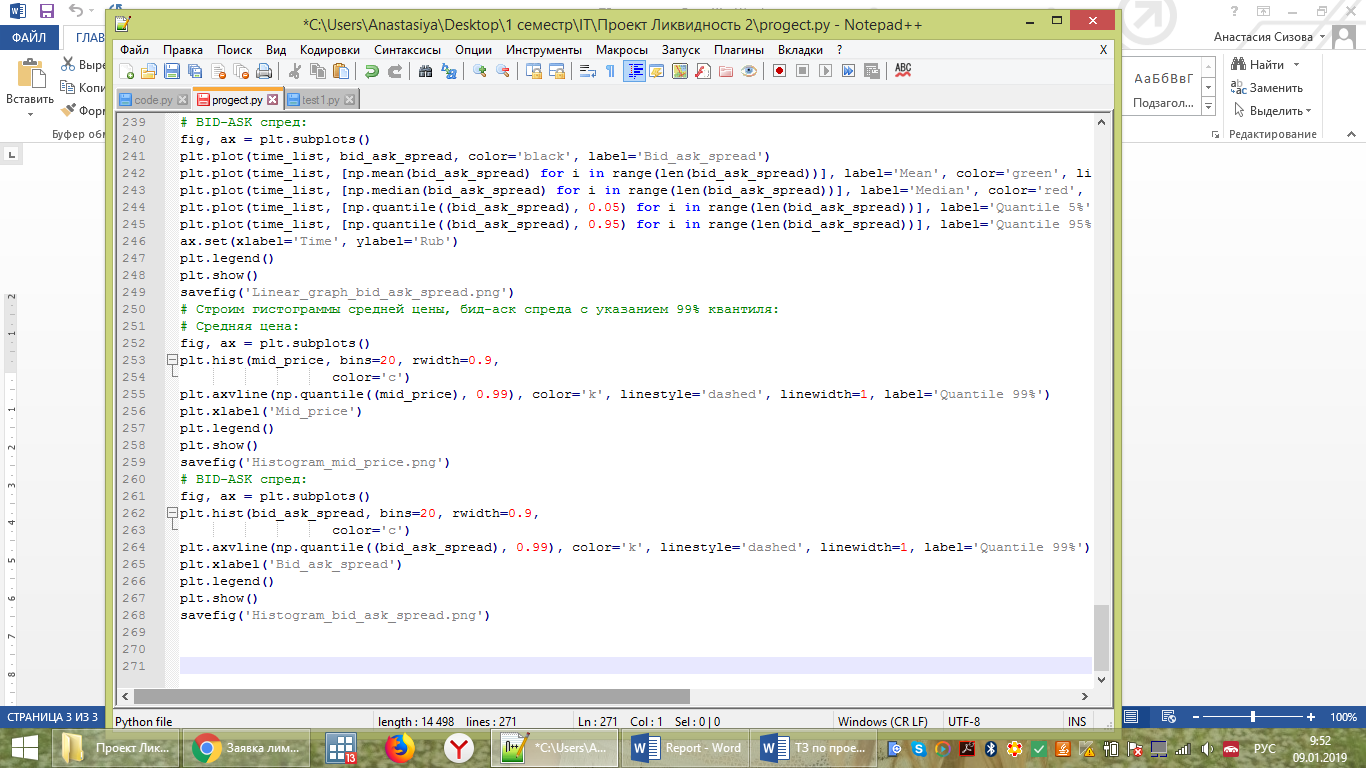
Результатами построения временных рядов по выбранным показателям являются графики (линейный – динамика показателя; гистограмма – значений ряда) с возможностью экспорта в формате .png.

*Рисунок 34. Код для построения линейных графиков временных рядов.*



*Рисунок составлен авторами проекта при помощи Notepad++.*

*Рисунок 35. Код для построения гистограммы временных рядов.*



*Рисунок составлен авторами проекта при помощи Notepad++.*

**Заключение.**

В процессе реализации проекта достигнуты основные поставленные задачи по созданию и успешному функционированию 4-х блоков задания. Проект реализуется выполнением двух файлов в двух различных средах. Первый и второй блок реализуется в среде MySQL посредством исполнения файла script.sql. В рамках данных блоков создается, редактируется и анализируется база данных «exersize1», содержащая таблицы с информацией по заявкам на две даты: 7 и 22 декабря 2015 года в зависимости от типа ценных бумаг (акции обыкновенные и привилегированные, облигации). Второй и третий блок реализуется в Python посредством выполнения скрипта project.py. Основная цель третьего блока заключается в составлении книги лимитированных заявок по заданным пользователе параметрам, а четвертого – в анализе показателей ликвидности.