**Этап 1. Установка ПО**

Установить MySQL по данной ссылке: <https://dev.mysql.com/downloads/installer/>

Данный скрипт был создан и протестирован на версии MySQL Installer 5.7.24

Загрузить MySQL Installer, и через него загрузить необходимые пакеты.

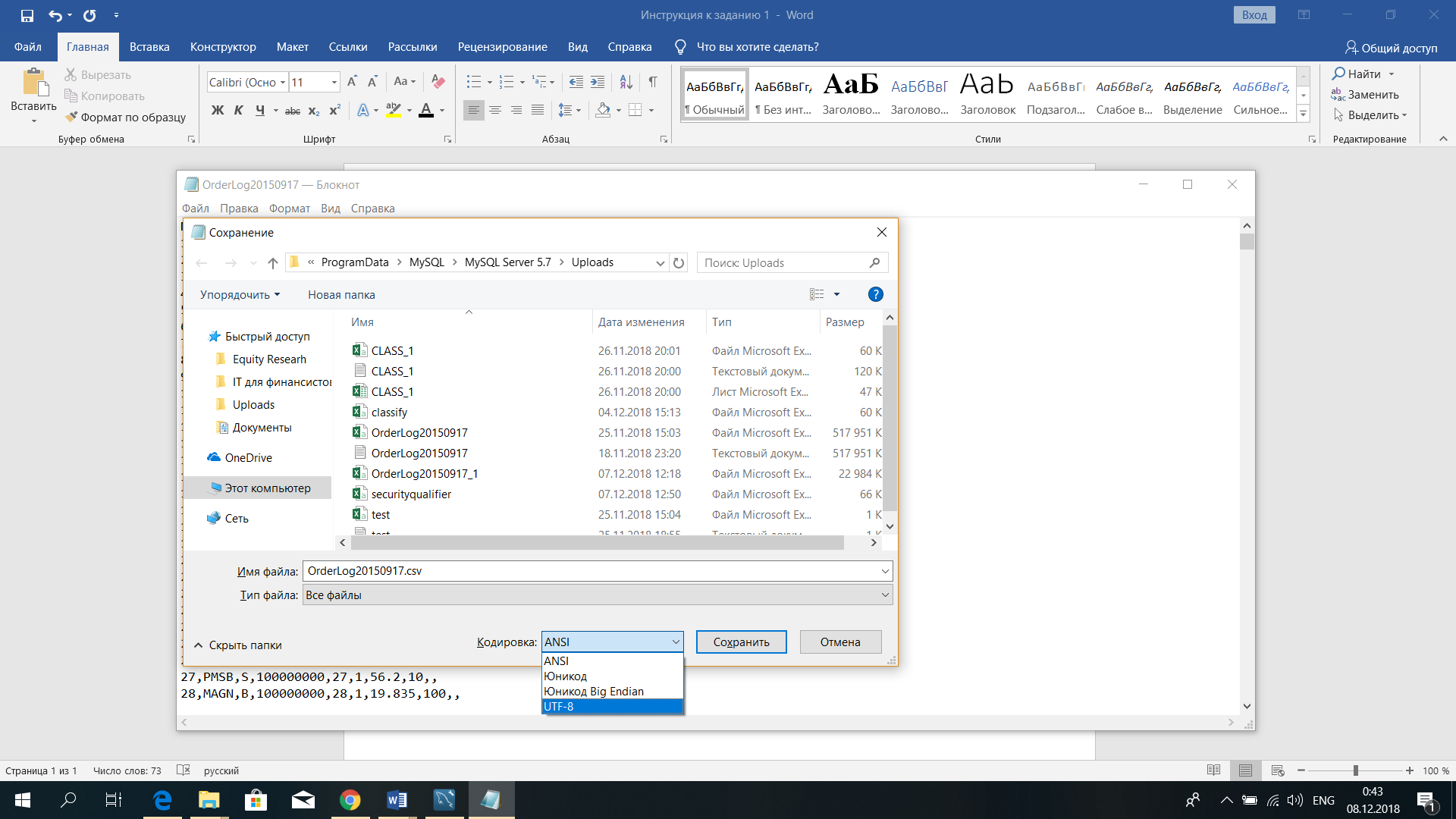
**Этап 2. Подготовительные действия с подгружаемыми файлами**

Orderlog

Необходимо преобразовать текстовый файл OrderLog20150917.txt в формат csv.

Для этого открываем файл и через «сохранить как…»

Вручную прописываем формат csv, предварительно выбрав тип файла: все файлы и кодировку UTF-8.



Классификатор

Скачиваем классификатор ценных бумаг с сайта Московской биржи: <https://www.moex.com/ru/listing/securities-list.aspx>

В формате csv (разделители - запятые), открываем, преобразовываем через:

Данные –> Получение внешних данных –> Из текста

Получаем эксель фал с колонками. Удаляем все колонки, кроме:

INSTRUMENT\_ID

INSTRUMENT\_TYPE

TRADE\_CODE

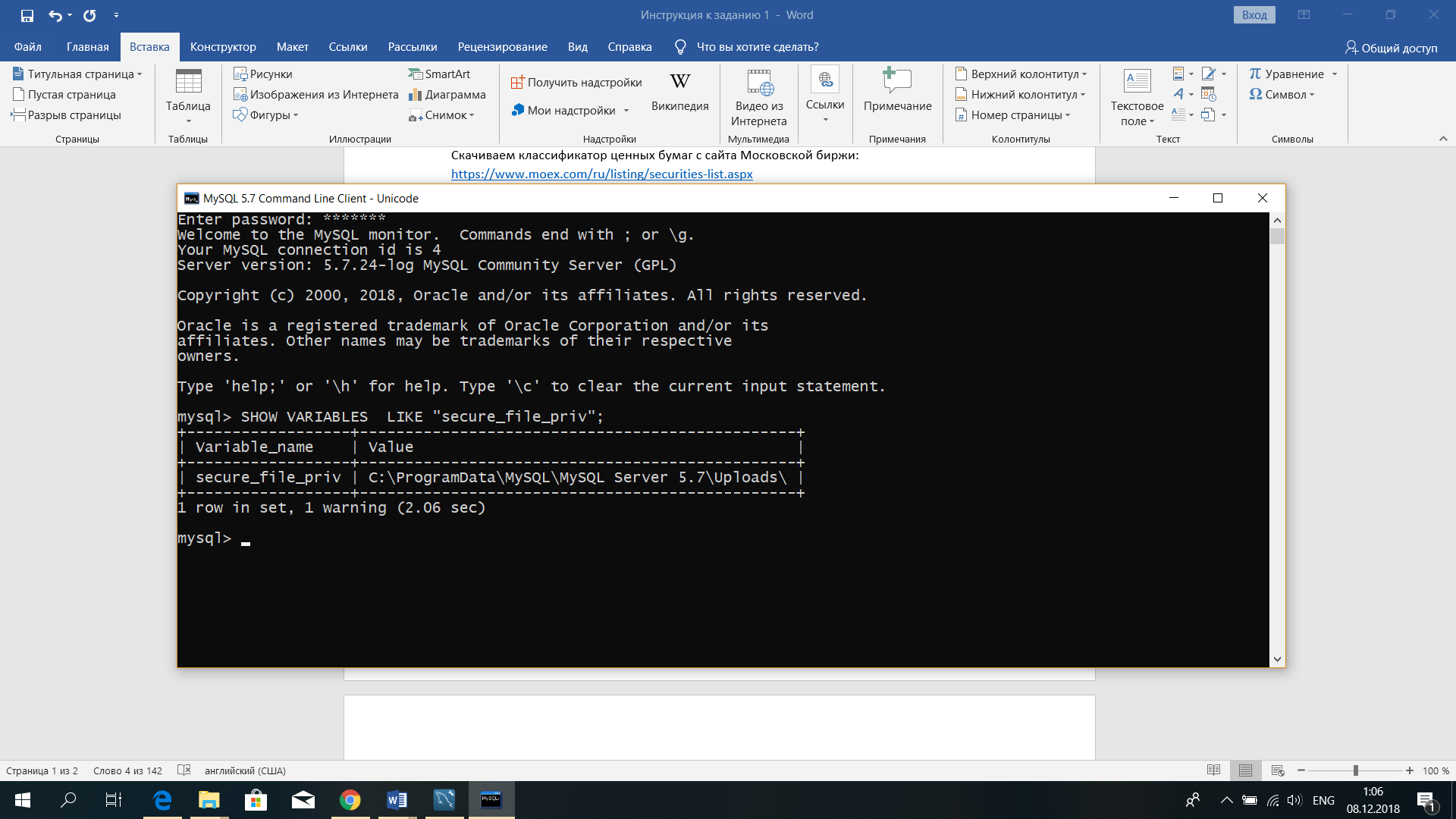
Затем снова преобразовываем файл в формат csv с разделителями – запятая и кодировкой UTF-8

**Этап 3. Подготовительные действия с MySQL**

**А.** Теперь необходимо проверить директорию сохранения файлов (ордерлога и классификатора)

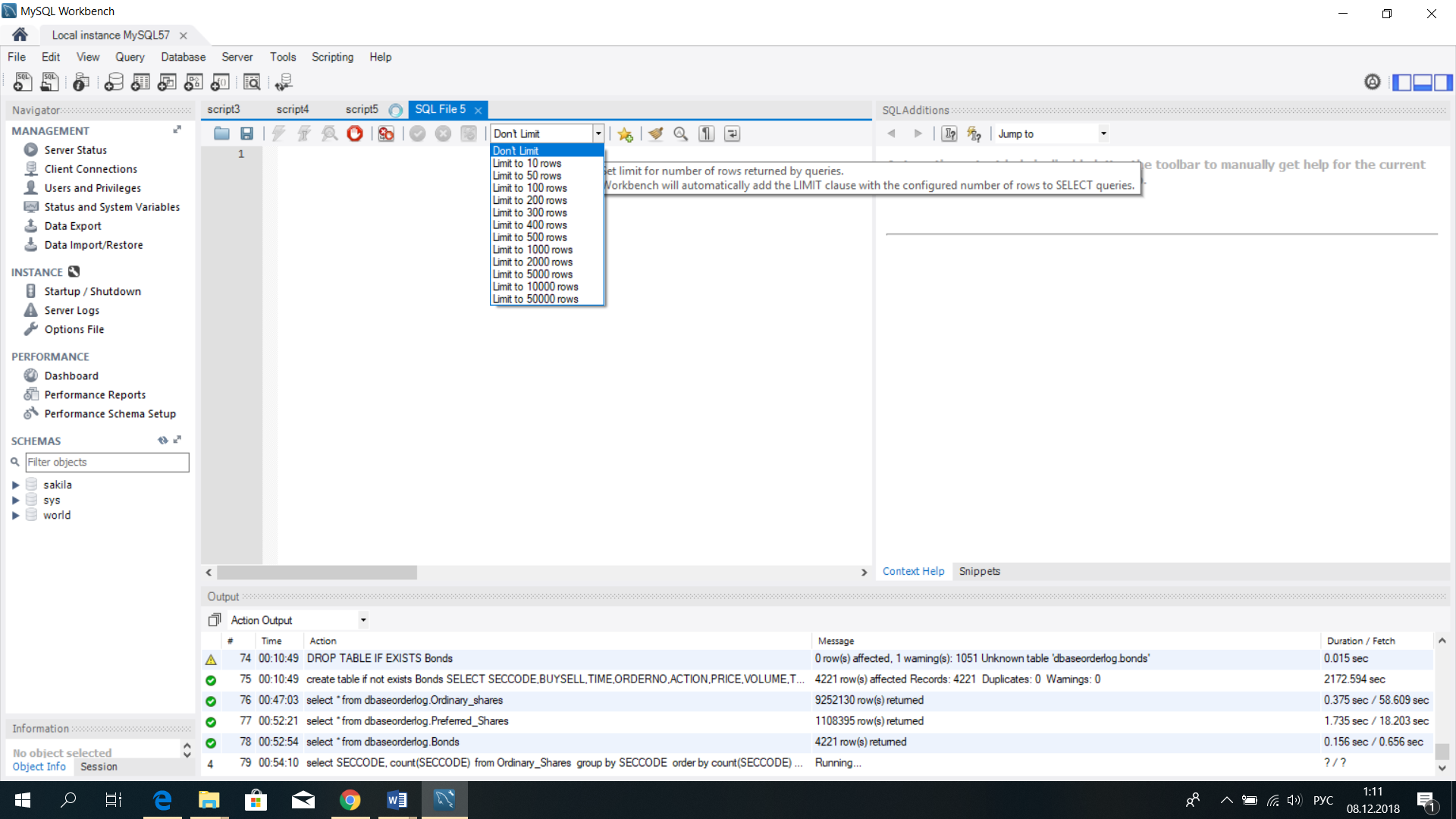
Проверка через консоль

mysql> SHOW VARIABLES LIKE "secure\_file\_priv";



Таким образом, файлы должны быть перемещены в данную директорию.

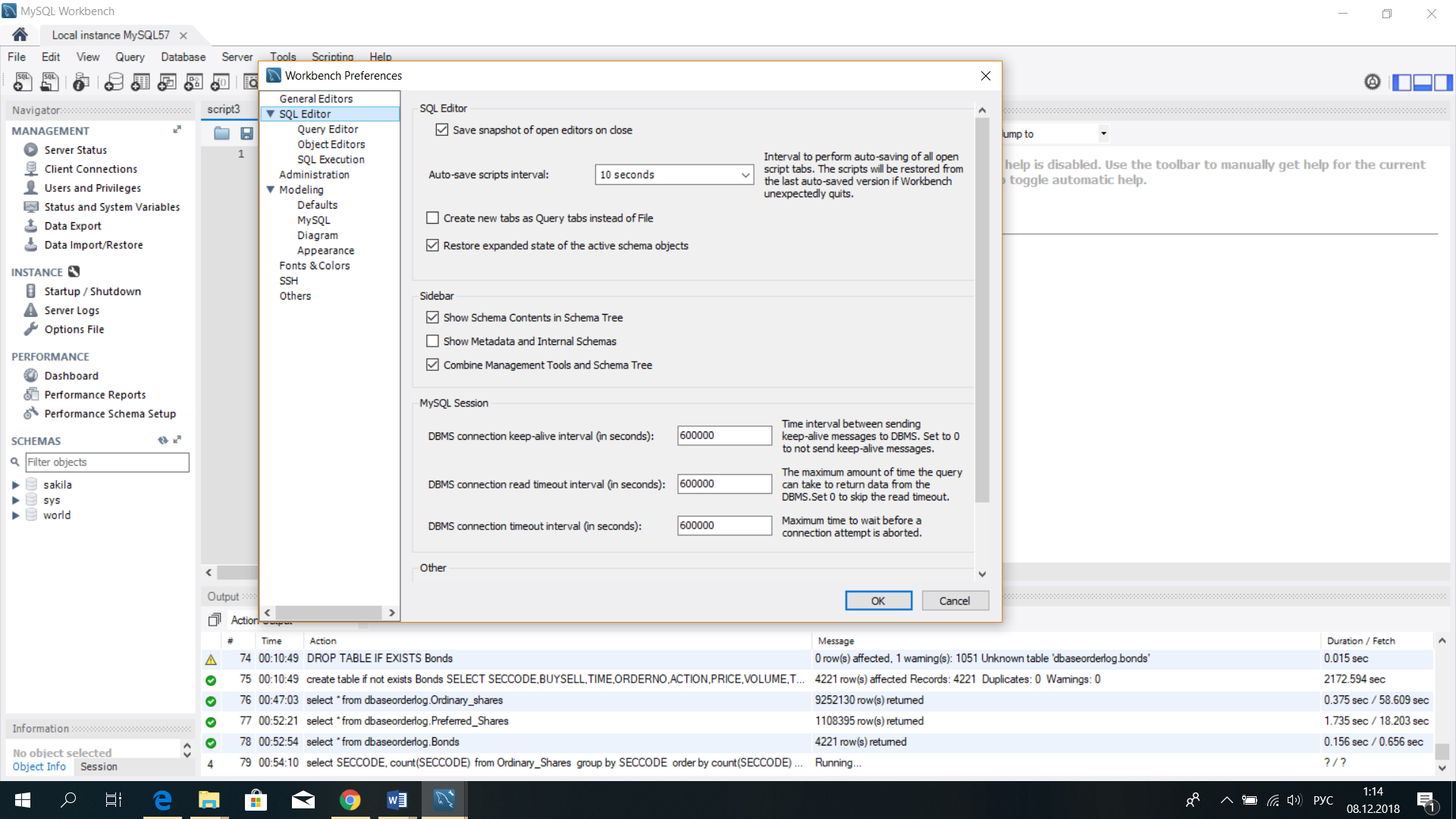
**Б.** убираем лимиты по количеству загружаемых строк «Don’t limit»



**В.** Идем в настройки

Edit - Preferences - SQL Editor - MySQL session

Меняем временные интервалы ожидания и пр.



В случае загрузки моего файла хватило 600 000. Если выдает ошибку lost connection with server, Вам следует увеличить временной интервал загрузки.

**Этап 4. Программирование**

Рекомендуется использовать 3 скрипта для выполенения каждого этапа задания.

*Скрипт 1.*

Сначала создаем базу данных:

**drop database if exists dbaseorderlog;**

**create database if not exists dbaseorderlog;**

**use dbaseorderlog;**

Затем создаем таблицу Orderlog:

**DROP TABLE IF EXISTS Orderlog;**

**create table Orderlog (**

**NO INT PRIMARY KEY not null,**

**SECCODE text,**

**BUYSELL CHAR(1),**

**TIME INT,**

**ORDERNO INT (1),**

**ACTION INT,**

**PRICE Float,**

**VOLUME INT,**

**TRADENO text,**

**TRADEPRICE text**

**);**

В созданную таблицу загружаем данные из csv, команды SET… использованы для оптимизации времени загрузки:

**SET autocommit =0;**

**SET unique\_checks = 0;**

**SET foreign\_key\_checks = 0;**

**LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 5.7/Uploads/OrderLog20150917.csv'**

**INTO TABLE Orderlog**

**FIELDS TERMINATED BY ','**

**ENCLOSED BY '"'**

**LINES TERMINATED BY '\r\n'**

**IGNORE 1 LINES**

**(NO,SECCODE,BUYSELL,TIME,ORDERNO,ACTION,PRICE,VOLUME,TRADENO,TRADEPRICE);**

**COMMIT;**

**SET unique\_checks = 1;**

**SET foreign\_key\_checks = 1;**

P.S. Если при загрузке файла произошла ошибка 1290, но при этом директория файла указана верно, то возможно решить поможет изменение слэшей с «\» на «/».

Далее проверяем данные на нулевые значения:

**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 dbaseorderlog.Orderlog;**

Проверяем данные на отрицательные значения:

**SELECT \*,**

**If (**

**NO < 0**

**or**

**SECCODE < 0**

**or**

**BUYSELL <0**

**or**

**TIME < 0**

**or**

**ORDERNO < 0**

**or**

**ACTION < 0**

**or**

**PRICE < 0**

**or**

**VOLUME is NULL or VOLUME < 0, 'NEGATIVE NUMBERS ARE NOT PERMITTED','NO NEGATIVE NUMBERS WERE FOUND'**

**) as `NEGATIVE NUMBERS CHECK` from dbaseorderlog.Orderlog;**

Проверяем корректность тикеров, если длина тикера не соотвествует длине тикера обыкновенной акции, привелегированной акции или облигации, программа выдаст комментарий о том, что данные неидефицированны:

**SELECT \*,**

**If (**

**length(SECCODE) = 5 or length(SECCODE) = 4 or length(SECCODE) = 12, 'data correct','data unidentified'**

**) as `DATA CHECK` from dbaseorderlog.Orderlog;**

Создаем таблицу классификатор и подгружаем данные из csv – файла:

**DROP TABLE IF EXISTS Classifier;**

**create table if not exists Classifier (**

**INSTRUMENT\_ID int,**

**INSTRUMENT\_TYPE text,**

**TRADE\_CODE text**

**);**

**SET autocommit =0;**

**SET unique\_checks = 0;**

**SET foreign\_key\_checks = 0;**

**LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 5.7/Uploads/securityqualifier.csv'**

**INTO TABLE Classifier**

**fields terminated by ','**

**lines terminated by '\r\n'**

**ignore 1 lines**

**(INSTRUMENT\_ID,INSTRUMENT\_TYPE,TRADE\_CODE);**

**COMMIT;**

**SET unique\_checks = 1;**

**SET foreign\_key\_checks = 1;**

*Скрипт 2.*

Создаем таблицу с обыкновенными акциями:

**DROP TABLE IF EXISTS Ordinary\_Shares;**

**create table if not exists Ordinary\_Shares**

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

**FROM orderlog JOIN classifier**

**on SECCODE=TRADE\_CODE**

**where INSTRUMENT\_TYPE='Ordinary share';**

Создаем таблицу с привелегированными акциями:

**DROP TABLE IF EXISTS Preferred\_Shares;**

**create table if not exists Preferred\_Shares**

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

**FROM orderlog JOIN classifier**

**on SECCODE=TRADE\_CODE**

**where INSTRUMENT\_TYPE='preferred shares';**

Создаем таблицу с облигациями:

**DROP TABLE IF EXISTS Bonds;**

**create table if not exists Bonds**

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

**FROM orderlog JOIN classifier**

**on SECCODE=TRADE\_CODE**

**where INSTRUMENT\_TYPE='Bond'**

**or INSTRUMENT\_TYPE='Corporate bond'**

**or INSTRUMENT\_TYPE='Subfederal bond'**

**or INSTRUMENT\_TYPE='Bonds of a foreign issuer'**

**or INSTRUMENT\_TYPE='Eurobonds'**

**or INSTRUMENT\_TYPE='Federal loan bond';**

Смотрим на конечные таблицы:

**select \* from dbaseorderlog.Ordinary\_shares;**

**select \* from dbaseorderlog.Preferred\_Shares;**

**select \* from dbaseorderlog.Bonds;**

*Скрипт 3.*

Выводим значение тикера в таблице с обыкновенными акциями, с которым было совершенно больше всего сделок:

**select SECCODE, count(SECCODE)**

**from Ordinary\_Shares**

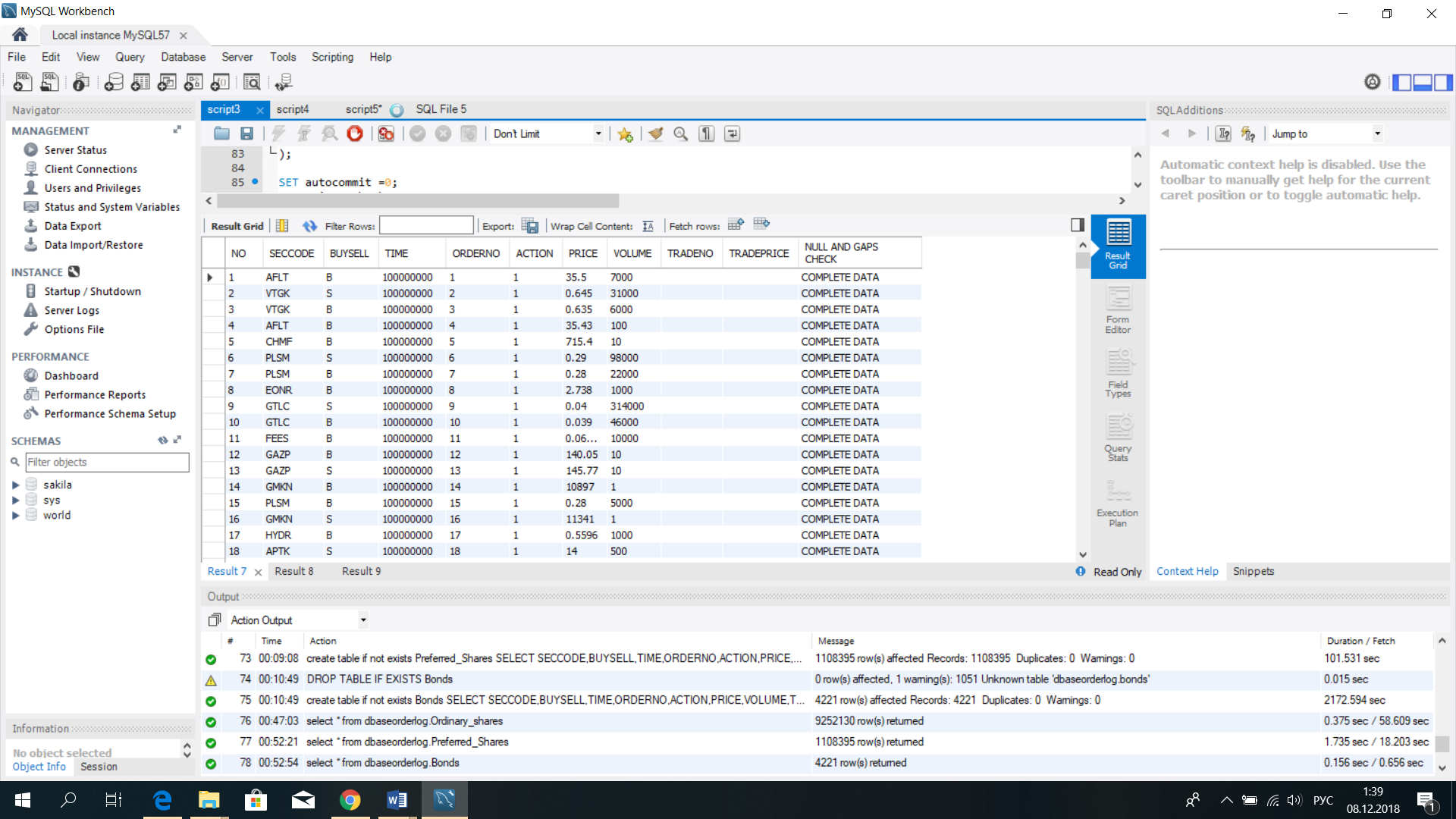
**group by SECCODE**

**order by count(SECCODE) desc**

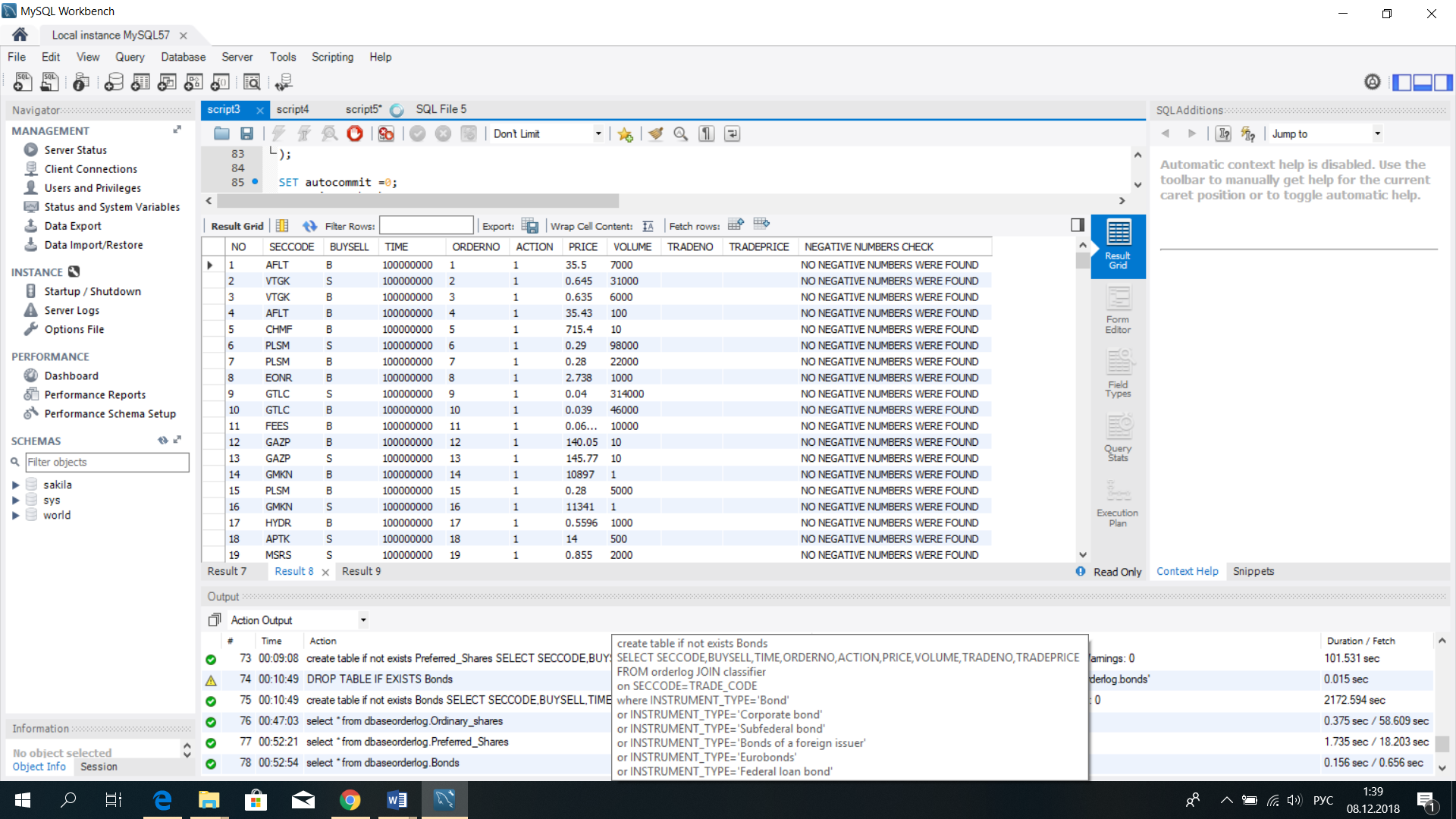
**limit 1;**

**Этап 5. Ожидаемые результаты:**

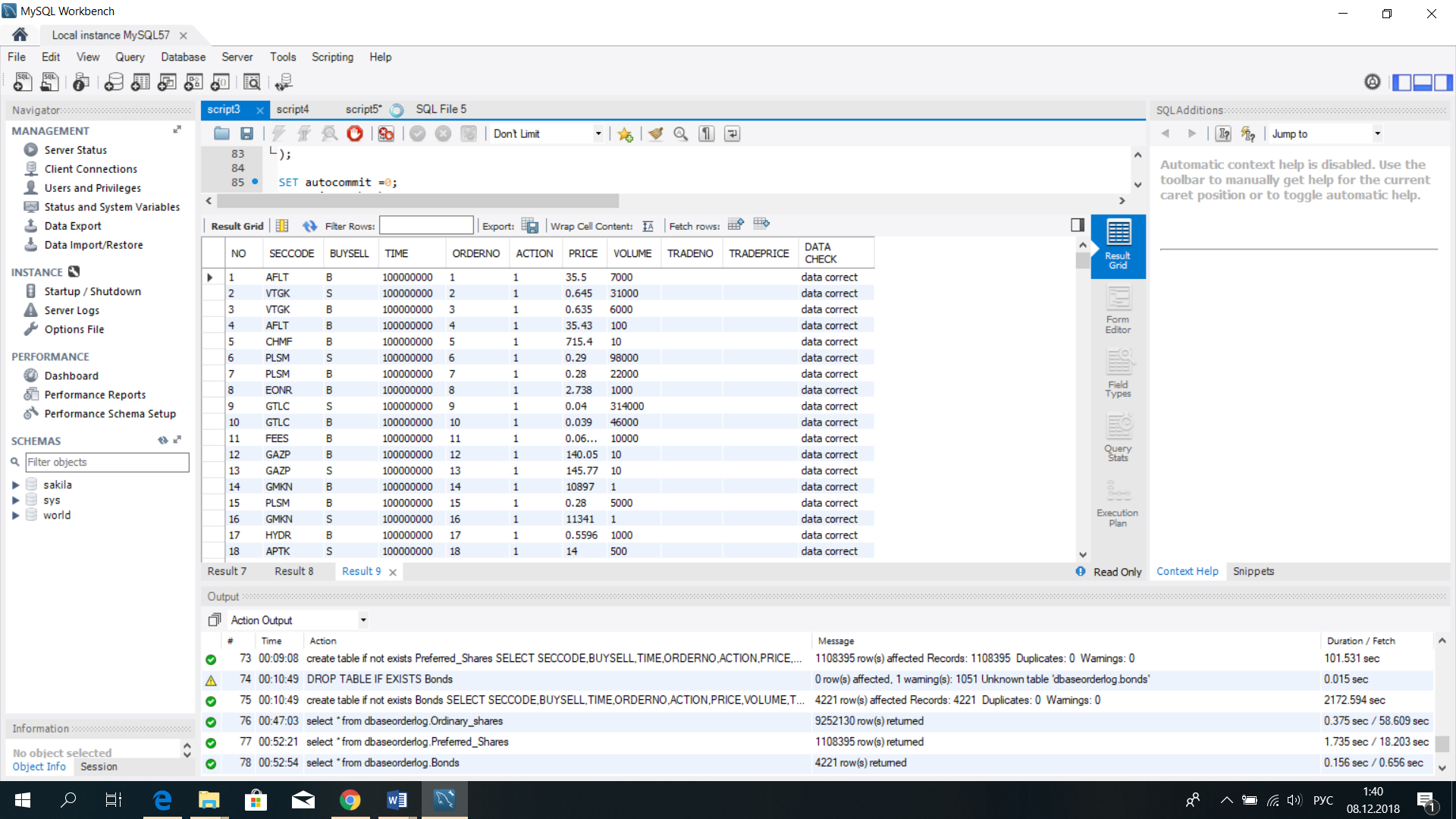
Результат проверки данных на нулевые значения:



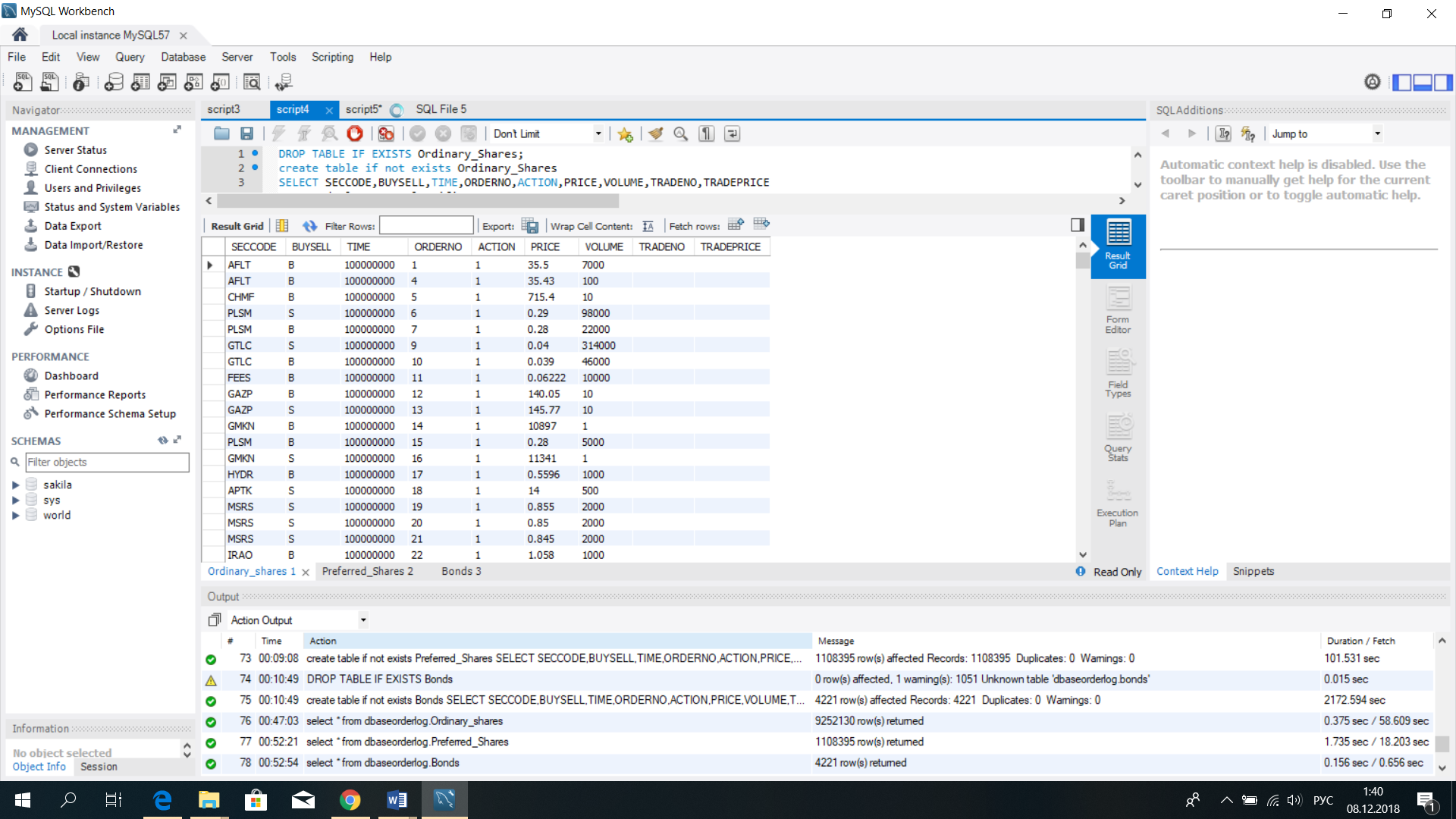
Результат проверки данных на отрицательные значения:



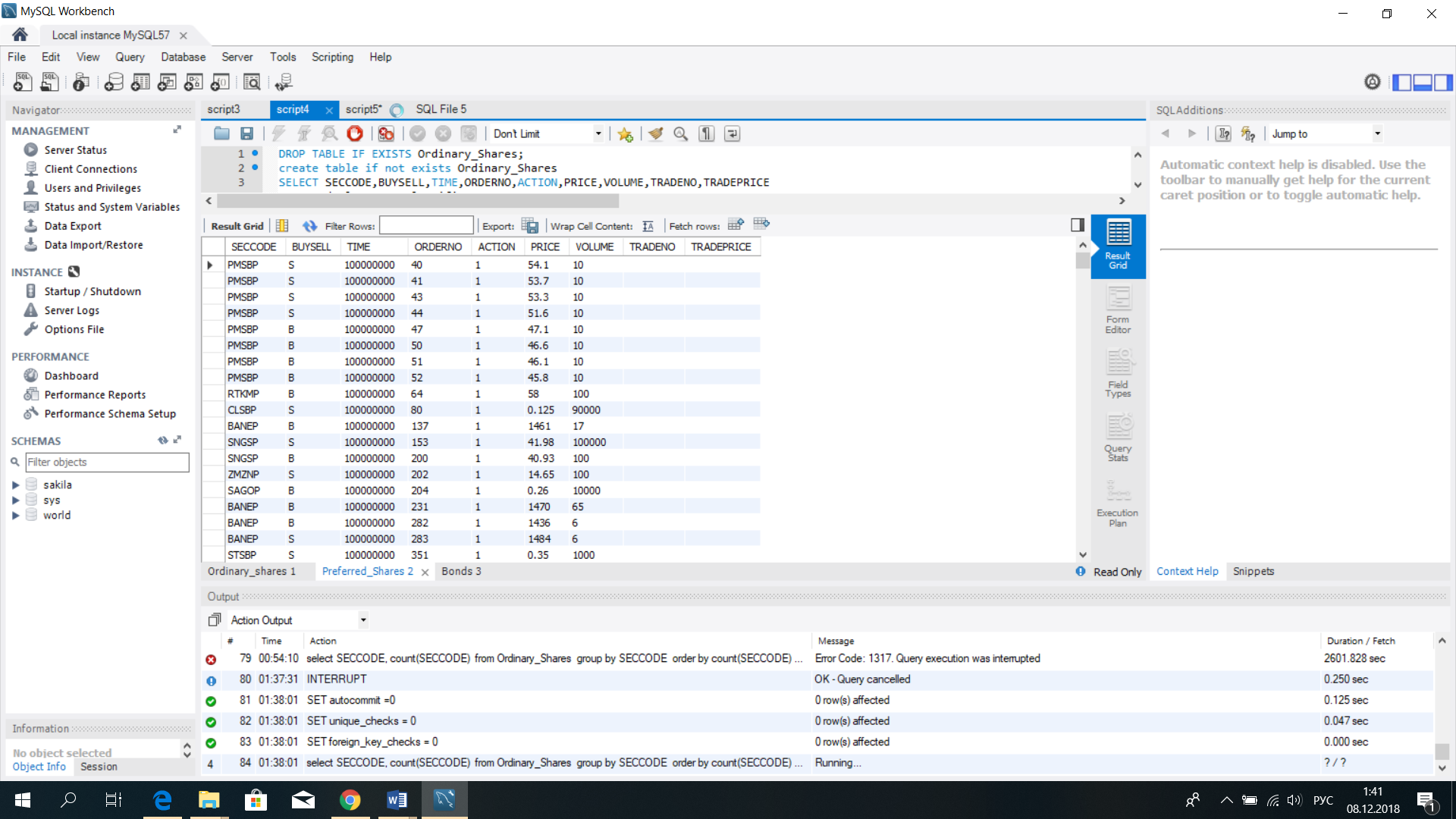
Результат проверки корректности тикеров:



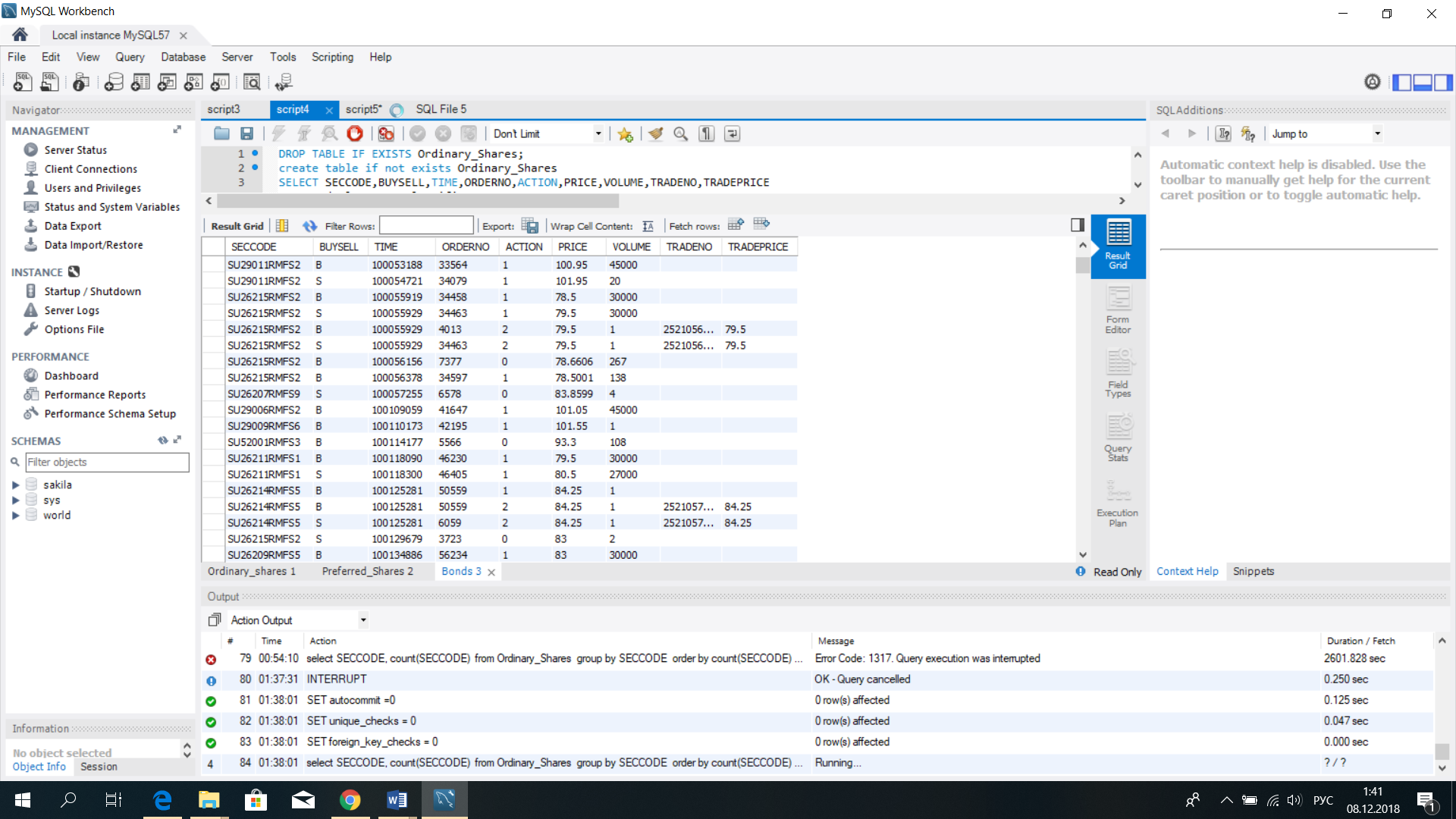
Результат создания таблицы с обыкновенными акциями:



Результат создания таблицы с привелегированными акциями:



Результат создания таблицы с облигациями:



Общий результат:

