**Задание по части 1:**

1. Скриптом на SQL загрузить txt файл, содержащий order log единой таблицей в базу данных, и проверить поля таблицы на наличие очевидных ошибок (пропуски данных, отрицательные значение, [одну ошибку придумать и проверить самостоятельно]). Форматы полей должны соответствовать спецификациям от биржи.
2. С помощью запроса SQL (возможно вложенных запросов) разбить таблицу на три, содержащие 1) поток заявок с обыкновенными акциями; 2) поток заявок с привилегированными акциями; 3) поток заявок с облигациями. Для этого внести базу в таблицу-классификатор тикеров по указанным трем типам ценных бумаг (классификацию скачать с сайта Московской биржи).
3. SQL запросом в таблице торгов обыкновенными акциями найти тикер, по которым было совершенно больше всего сделок, т.е. исполнено больше всего рыночных заявок.

Данные: https://drive.google.com/open?id=15u2lcSKh4kmjDm8BjGcGs70YrbaYxHs4

**Задание по части 2:**

1. Соединить Python с полученной в ходе первого задания БД по ODBC.
2. В Python написать код, который:
   1. Запрашивает в базе данных таблицу-классификатор торгуемых ценных бумаг.
   2. Предлагает пользователю ввести тикер и момент времени в течение торговой сессий [можно узнать на сайте Московской биржи или определить из данных orderlog’а].
   3. По введенным данным запрашивает из соответствующей таблице БД данные о потоке заявок по указанному тикеру до указанного момента времени.
   4. По полученным данным построит «стакан» и визуализирует его в виде таблицы и графика; на графике в явном виде отмечает такие аспекты ликвидности как сжатость (бид-аск спрэд) и глубину (объем на лучших ценах покупки и продажи).
   5. Выявляет заявки типа «айзберг» и сохраняет информацию их номерах, выявленных в них скрытых объемах и времени обнаружения в виде таблицы.

**(Бонус)** отправить информацию о скрытых объемах обратно в базу данных.

**Задание по части 3:**

1. Соединить Python с полученной в ходе первого задания БД по ODBC.
2. В Python написать код, который:
   1. Предлагает пользователю ввести тикер ценной бумаги (например, SBER) и шаг (например, 1 мин, 5 мин, 30 мин, 1 час); *рекомендует также предлагать ввести пределы периода, внутри которого буду выполняться вычисления (например, с 14:15:00 до 17:45:00)*.
   2. Внутри торгового дня (или выбранного периода времени) для выбранной ценной бумаги определить временные ряды с заданным шагом для следующих показателей: средняя цена (MID=(BID+ASK)/2); сжатость (бид-аск спрэд=ASK-BID) и глубину (объем на лучших ценах покупки ASK и продажи BID).
   3. Предлагает пользователю задать торговый объем V (в шт. или лотах) и направление сделки (купить/продать); вычисляет ряды с выбранным шагом и строит соответствующие графики для показателя транзакционных издержек TC(V), равный модулю разности между ценой MID и средней ценой, по которой объем был реализован.
   4. Для полученных временных рядов построит два графика: 1) linear plot динамики показателей в координатах (t, f(t)) и на графике отметить 5% и 95% квантили, среднее и медиану ряда; 2) гистограмму значений ряда (на графике указать 99% процентный квантиль).

**В четвертой части** требуется создать отчет о проделанной работе с обязательным центрированным полем «Отчет» в начале файла. Отчет должен содержать информацию о выбранном периоде, один графический элемент и одну таблицу с пояснениями. Отчет должен выводиться автоматически запросом из SQL.

***Материалы для сдачи (по каждой части отдельно):***

1. Краткая инструкция, содержащая 1) подготовительные действия для работы с данными (где и в каком виде хранить данные, какой софт установить и т.д.); 2) краткое содержания процедур, заложенных в скриптах и запросах, + порядок их запуска; 3) описание ожидаемых результатов.
2. Коды скриптов и запросов.
3. Все (кроме данных) можно размещать на github и присылать только ссылку.

***Требование к выполнению задания:***

1. Все процедуры и результаты должны быть реплицируемы и описаны в мере, обеспечивающей использование Ваших кодов третьим лицом.