|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ | | | | | | | | |
|  | |  |  | | |  |  | |
| Пермский государственный национальный  исследовательский университет | | | | | | | | |
|  |  | |  | | |  | |  |
|  | ОТЧЕТ  по лабораторной работе №2  «Создание клиентского приложения по выполнению CRUD операций на популярном информационном ресурсе»  по дисциплине  «Технологии разработки распределенных приложений» | | | | | | |  |
|  |  | | |  |  | | |  |
|  | Работу выполнил студент гр. КМБ-16  Тарарков Андрей Викторович  16.10.2020 | | |  | Проверил  доцент кафедры прикладной математики и информатики  Деменев Алексей Геннадьевич | | |  |
|  |  | | |  |  | | |  |
| Пермь, 2020 г. | | | | | | | | |

**Введение**

Цель: изучение возможностей технологии передачи сообщений и сокетов для создания распределенных приложений. Сравнение методов реализации взаимодействия компонент распределенной системы.

Формируемые компетенции: способность применять на практике теоретические основы и общие принципы разработки распределенных систем; способность использовать на практике стандарты сетевого взаимодействия компонент распределенной системы.

**Требования к выполнению работы**

|  |  |
| --- | --- |
| **Требование к заданию** | **Максимальное количество баллов** |
| Приложение позволяет выполнять прием и передачу данных из ненормализованной БД в нормализованную без модификации данных. | 2 |
| Приложение позволяет передавать информацию с помощью очередей сообщений. | 3 |
| Приложение позволяет передавать информацию с помощью сокетов. | 3 |
| При передаче данных они шифруются с помощью симметричного ключа. | 2 |
| При передаче симметричного ключа шифрования данные шифруются с помощью асимметричного ключа. | 2 |
| Приложение, написанное студентом, работает в сети без сбоев. | 2 |

**Выбор предметной области**

Предметная область: продажа электроники в оффлайн магазине

**Ход выполнения работы**

Реализован импорт данных из ненормализованной настольной СУБД sqllite в корпоративную СУБД MySQL в нормализованной форме (3 НФ)

В качестве очереди сообщений для выполнения лабораторной работы была выбрана система Apache Kafka

Приложение написано на языке Python 3.8[3]. Среда разработки приложения: PyCharm[1]

Исходный код приложения сохранён в GitHub репозиторий <https://github.com/Dasem/trrp2> .

**Сценарии использования приложения**

1. Передача данных с использованием сокетов. Предусловия: у пользователей должны быть установлены соответствующие настройки в файле «default.properties» (для использования сокетов, используется параметр mode=SOCKET)
   1. Принимающий пользователь запускает приложение
   2. Передающий пользователь запускает приложение
   3. Происходит автоматическая передача данных с уведомлением об окончании
2. Передача данных с использованием очереди сообщений. Предусловия: у пользователей должны быть установлены соответствующие настройки в файле «default.properties» (для использования очереди сообщений, используется параметр mode=KAFKA)
   1. Принимающий пользователь запускает приложение
   2. Передающий пользователь запускает приложение
   3. Происходит автоматическая передача данных с уведомлением об окончании

**Описание приложения**

С помощью приложения есть возможность конвертации базы данных без потери информации из первой НФ в третью НФ. Так же приложение позволяет передавать подобным образом данные на другие хосты с помощью очереди сообщений, либо с использованием сокетов. Оба варианты предусмотрено для передачи всей информации в целях повышения отказоустойчивости приложения.

**Приёмочное тестирование**

Изначально имеется заполненная база данных sqlite, а также пустая MySql база данных sellers

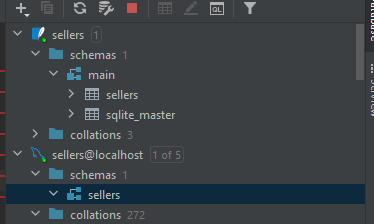


Рисунок 1

Установим передачу данных с помощью сокетов:

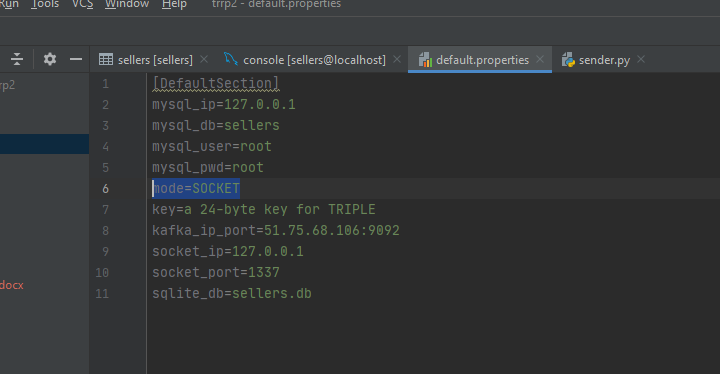


Рисунок 2

После последовательного запуска reciever.py, sender.py в консоли sender.py появляется информация об отправке данных, а так же их зашифрованное представление

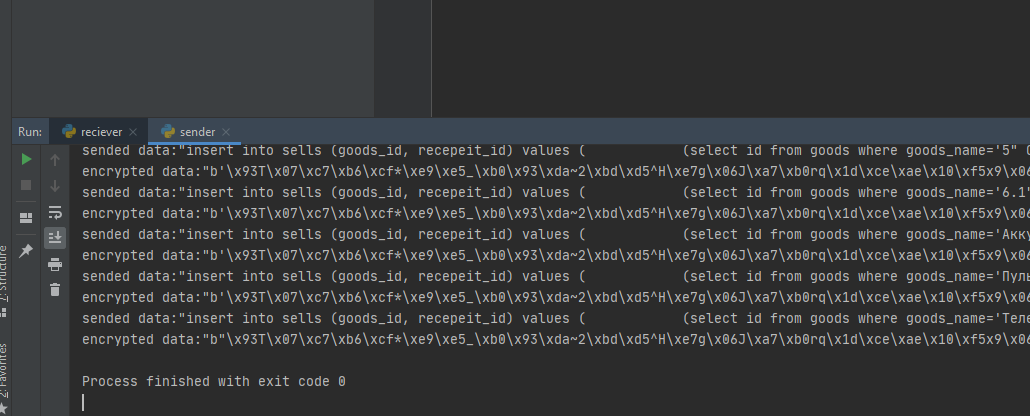


Рисунок 3

После отправки данных можно наблюдать созданные таблицы

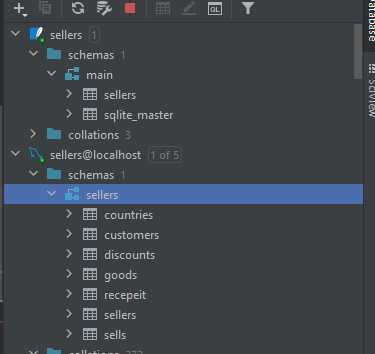


Рисунок 4

Проверим корректность передачи данных с помощью сайта <http://www.jsondiff.com/>

Скопируем данные из нормализованной базы по запросу

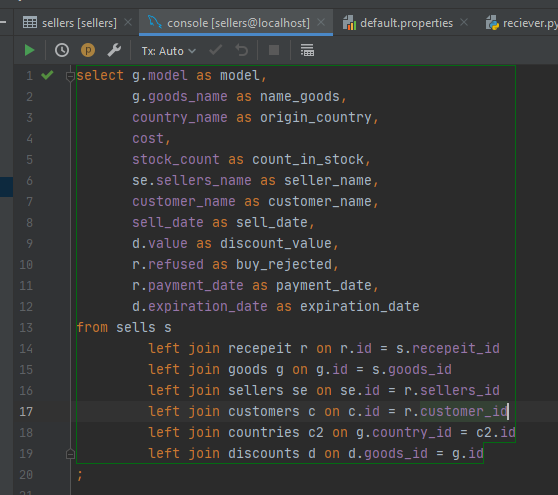


Рисунок 5

И данные из ненормализованной базы в формате JSON



Рисунок 6

Результирующие выборки идентичны, следовательно, передача данных происходит без изменений, а также передача с помощью сокетов корректна

Проверка передачи данных через сокет: до включения программы сокет не активен, во время активен, после завершения в ожидании

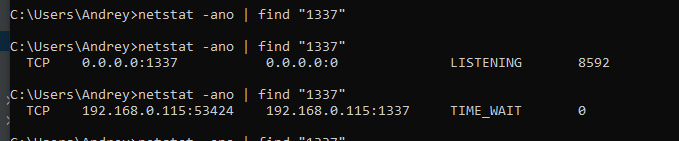


Рисунок 7

Повторим те же действия с очередью сообщений Apache Kafka, предварительно изменив конфигурацию программы на mode=KAFKA. Проверим передачу данных подключившись параллельно к топику сообщений.

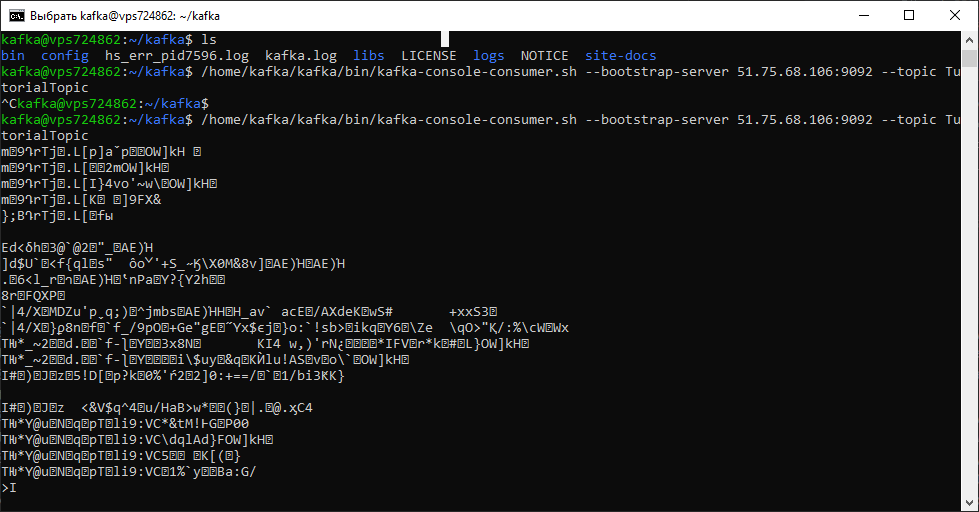


Рисунок 8

На скрине видно, что данные передаются в зашифрованном виде

Изменим IP в конфигурации на IP локальной сети

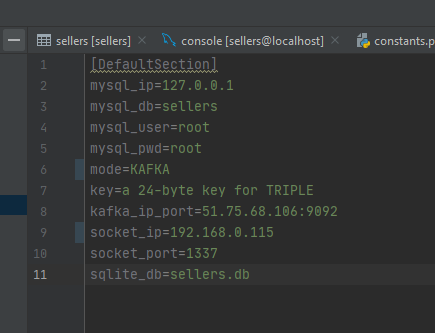


Рисунок 9

Выполнение программы так же корректно завершено

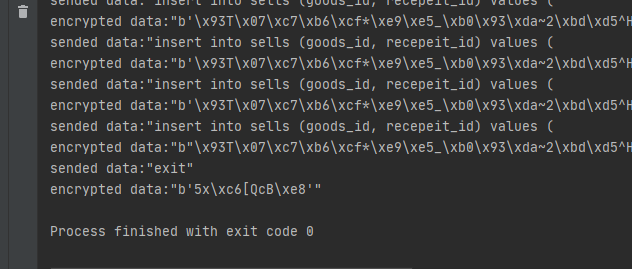


Рисунок 10

**Заключение**

В работы достигнута цель и выполнены следующие требования:

Приложение позволяет выполнять прием и передачу данных из ненормализованной БД в нормализованную без модификации данных. 2 балла

Приложение позволяет передавать информацию с помощью очередей сообщений. 3 балла

Приложение позволяет передавать информацию с помощью сокетов. 3 балла

При передаче данных они шифруются с помощью симметричного ключа (3DES). 2 балла

При передаче симметричного ключа шифрования данные шифруются с помощью асимметричного ключа. 0 баллов. Ассиметричный ключ для передачи симметричного не спользуется

Приложение, написанное студентом, работает в сети без сбоев. 2 балла. Приложение корректно работает при смене конфигурации

Требования выполнены не полностью (12/14 баллов), корректность выполненных операций доказана приёмочными тестами.

**Программное обеспечение, использовавшееся при выполнении работы, а также лицензии**

1. PyCharm, ссылка для получения бесплатной лицензии студентами и преподавателями: <https://www.jetbrains.com/shop/eform/students>, лицензия: <https://www.jetbrains.com/company/privacy.html>
2. Ресурс JSONdiff находится в свободном доступе на хостинге github : <https://github.com/zgrossbart/jdd>
3. Apache Kafka лицензия: <https://www.apache.org/licenses/LICENSE-2.0.html>. Скачать: <https://kafka.apache.org>
4. Python 3.8, скачать: <https://www.python.org/downloads/>, лицензия: <https://docs.python.org/3/license.html>
5. DES (python библиотека шифрования) Лицензия MIT. (Установка: pip install des)