**Отзыв**
**о прохождении технологической (производственной) практики**

**студенткой 3 курса очной формы обучения Шевцовой Варварой Антоновной,**
**обучающейся в ФГБОУ ВО «СГУ имени Н.Г. Чернышевского» по направлению подготовки 09.03.04 – Программная инженерия.**

Студентка Шевцова Варвара Антоновна проходила производственную практику в ООО «ИНТЕЛЛЕКТУАЛЬНЫЕ РЕШЕНИЯ» с 22.06.2023г. по 19.07.2023г.

Производственная практика направлена на закрепление и углубление студентами полученных теоретических знаний и практических навыков и получение навыков решения научно-практических задач с использованием современных программно-аппаратных средств в составе научно-производственного коллектива.

В процессе прохождения практики студентка выполняла следующие обязанности:

- своевременное и стабильное взаимодействие с командой проекта;
- анализ рынка валют;
- разработка запросов к базе данных;
- анализ полученной в ходе работы проекта информации;
- корректировка имеющихся решений в связи с меняющимися данными;
- разработка оценочного материала для усвоения материала.

Индивидуальное задание, содержание и планируемые результаты практики Шевцовой Варвары Антоновны были согласованы с руководителем практики от университета старшим преподавателем кафедры математической кибернетики и компьютерных наук Сафрончик М. И.

В рамках своего участия Шевцова Варвара Антоновна проявила себя как активный, вовлеченный, стремящийся к изучению новых технологий работник.

Шевцова Варвара Антоновна успешно применяла полученные в университете теоретические знания в области информатики и разработки программно-информационных систем.

За время работы студентка освоила и закрепила следующие практические навыки:

- работы в научно-производственном коллективе;
- к самообразованию;
- постановки, обоснования и проверки проектных решений;
- самостоятельного выполнения научно-исследовательской работы.
- объяснения типовых ошибок и способов их исправления;
- разработки оценочного материала для усвоения материала.

Таким образом, производственная практика способствовала формированию следующих компетенций:

- Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде (УК-3);
- Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни (УК-6);
- Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций (УК-8);
- Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно- коммуникационных технологий и с учетом основных требований информационной безопасности (ОПК-3);
- Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-8);
- Готов к разработке, созданию, сопровождению требований, технических заданий на разработку, техническую поддержку, сопровождение информационных систем, информационных ресурсов, компонентов программных продуктов (ПК-1);
- Способен к формулированию требований безопасности информационных систем, способен анализировать риски при разработке, к созданию и сопровождению программных продуктов (ПК - 2).

За прохождение производственной практики студентка Шевцова Варвара Антоновна заслуживает оценки отлично.

Заместитель генерального директора ООО «ИНТЕЛЛЕКТУАЛЬНЫЕ РЕШЕНИЯ»:


Е.А. Лащ

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

УТВЕРЖДАЮ
Зав.кафедрой,
доцент, к. ф.-м. н.
_____ С. В. Миронов

# ОТЧЕТ О ПРАКТИКЕ

студентки 3 курса 351 группы факультета КНиИТ
Шевцовой Варвары Антоновны

вид практики: производственная
кафедра: математической кибернетики и компьютерных наук
курс: 3
семестр: 6
продолжительность: 4 нед., с 22.06.2023 г. по 19.07.2023 г.

Руководитель практики от университета,
ст. преп., к. ф.-м. н. _____ М. И. Сафрончик

Руководитель практики от организации (учреждения, предприятия),
заместитель ген. директора _____ Е. А. Лащ

Тема практики:«Анализ арбитражных ситуаций»

**СОДЕРЖАНИЕ**

# ВВЕДЕНИЕ

Практика проходила на базе предприятия ООО «Интеллектуальные решения» в должности младшего аналитика. Работа велась над проектом «Arbitoring» — стартапом, направленным на создание уникального инструмента, который будет обеспечивать сканирование и выявление арбитражных ситуаций на финансовых рынках.

В процессе разработки использовались язык программирования Python и объектно-реляционная система управления базами данных PostgreSQL.

Целью практики был анализ текущей ситуации на рынке криптовалют и сбор информации. В рамках производственной практики должны были быть решены следующие задачи:

1. идентификация потенциальных арбитражных ситуаций;
2. мониторинг комиссии по переводам криптовалют;
3. загрузка информации в базу данных;
4. анализ информации из базы данных.

## 1 Описание проекта

В наши дни арбитраж криптовалют актуален из-за высокой волатильности рынка, разнообразия торговых площадок и ограниченной эффективности рынка. Криптовалюты торгуются круглосуточно, что позволяет оперативно реагировать на изменения цен. Рост рынка деривативов также создает новые возможности для арбитража. С развитием технологии блокчейн и увеличением участников рынка растет интерес к арбитражу в этой сфере, несмотря на связанные риски.

Проект «Arbitoring» представляет собой стартап, направленный на создание уникального инструмента, который будет обеспечивать сканирование и выявление арбитражных ситуаций на финансовых рынках [1]. Эти арбитражные ситуации включают в себя различные типы операций, такие как P2P-сделки, торговля на спотовых и фьючерсных рынках, а также международные переводы с использованием протокола SWIFT и переводы через криптовалютные обменники.

В основе сканера «Arbitoring» лежит комплексный математический алгоритм. Этот алгоритм представляет собой набор формул и логических операций, разработанных для анализа и выявления цепочек арбитражных событий. Цепочка арбитража включает в себя последовательность шагов, выполняемых на различных финансовых платформах, таких как криптобиржи, банки и криптовалютные обменники.

В контексте проекта используемые термины можно определить следующим образом:

**Арбитражные ситуации** — разница в ценах или условиях на разных финансовых платформах, которая позволяет осуществить выгодные сделки или операции.

**P2P (Peer-to-Peer)** — прямые сделки между участниками без посредничества центральной инстанции.

**Спотовый рынок** — рынок, на котором активы продаются или покупаются для мгновенной поставки.

**Фьючерсный рынок** — контракты на будущие поставки активов по заранее установленной цене.

**SWIFT (Society for Worldwide Interbank Financial Telecommunication)** — международный стандарт для обмена информацией и инструкциями о финансовых транзакциях между банками.

**Криптобиржи и криптообменники** — электронные платформы для торговли криптовалютами и обмена ими.

## 2 Выполненные в рамках проекта задачи

### 2.1 Идентификация потенциальных арбитражных ситуаций

#### 2.1.1 Постановка задачи

Провести анализ данных на бирже Binance с целью выявления потенциальных арбитражных ситуаций между различными рынками (например, криптобиржи, банковский рынок). Определить ключевые параметры, влияющие на возможность арбитража.

#### 2.1.2 Решение

1. Начнем с сбора данных с криптобиржи Binance и банковского рынка. Для криптобиржи используется Binance API для получения актуальных данных о ценах на различные криптовалюты. Для банковского рынка потребуется обращение к сторонним источникам данных или использование финансовых API.
2. Обрабатываем и фильтруем данные, чтобы получить только необходимую информацию. Это включает в себя цены на активы, объемы торгов, временные метки и другие ключевые параметры.
3. Сопоставляем данные между криптобиржей и банковским рынком. Использованы общие инструменты (криптовалютные пары и соответствующие валюты на банковском рынке) для установления соответствия данных.
4. Определяем потенциальные арбитражные ситуации. Проводим анализ цен и объемов для выявления различий между ценами активов на криптобирже и банковском рынке. Потенциальные арбитражные ситуации могут возникнуть, если есть разрывы в ценах между рынками.
5. Определяем ключевые параметры, которые могут влиять на возможность арбитража. Это могут быть факторы, такие как задержки в передаче данных, комиссии, ликвидность и другие.

После эмпирического анализа и изучения информации на разных биржах с целью ознакомления был написан код на языке программирования Python для автоматизации процесса и сбора конкретных данных в одно место [2]:

```
import requests
import time

# установка ключа API и секрета Binance
# (скрыты в целях безопасности)
api_key = ''
api_secret = ''

def get_bank_data(crypto_symbol, fiat_symbol):
    params = {
        'crypto_symbol': crypto_symbol,
        'fiat_symbol': fiat_symbol,
    }

    try:
        # выполняем GET-запрос к API банковского рынка
        response = requests.get(bank_api_url, params=params)
        response.raise_for_status()

        # возвращаем цену с банковского рынка
        bank_data = response.json()
        return bank_data['price']

    except requests.exceptions.RequestException as e:
        print(f"Ошибка при получении данных с банковского рынка: {e}")
        return None

# функция для получения данных с Binance
def get_binance_data(symbol):
    endpoint = 'https://api.binance.com/api/v3/ticker/price'
    params = {'symbol': symbol}
    response = requests.get(endpoint, params=params)
    data = response.json()
    return float(data['price'])

# функция для выполнения арбитражного анализа
def perform_arbitrage_analysis(crypto_symbol, fiat_symbol):
    # получаем цену криптовалюты на Binance
    binance_crypto_price = get_binance_data(crypto_symbol + fiat_symbol)

    # получаем цену с банковского рынка
```

```
bank_price = get_bank_data(crypto_symbol, fiat_symbol)

# арбитражный анализ
price_difference = binance_crypto_price - bank_price

if price_difference > 0:
    print(f"Обнаружена потенциальная арбитражная возможность")
    print(f"Цена на Binance: {binance_crypto_price} {fiat_symbol}")
    print(f"Банковская цена: {bank_price} {fiat_symbol}")
    print(f"Разница в ценах: {price_difference} {fiat_symbol}")
else:
    print("Арбитражная возможность не обнаружена.")
```

API (Application Programming Interface) в данном контексте представляет собой интерфейс, который позволяет взаимодействовать с программным обеспечением или службой. Это набор правил и определений, который позволяет различным программам обмениваться данными и выполнять определенные действия.

В криптовалютном контексте, API используется для взаимодействия с биржами. Криптобиржевые API предоставляют программный интерфейс, с помощью которого разработчики могут отправлять запросы к бирже для получения данных о рынке, выполнения торговых операций, проверки баланса счета и других операций [3].

В представленном коде API Binance используется для отправки запросов на получение данных о комиссиях и другой информации о счете. API ключи (API key) и API секреты (API secret) нужны для аутентификации и обеспечения безопасности взаимодействия между клиентом и сервером биржи.

Когда программа отправляет запрос с использованием API, она обращается к серверу биржи, который обрабатывает запрос и отправляет обратно необходимую информацию в формате, который можно легко интерпретировать и использовать в коде программы [4].

В ходе выполнения задачи были выявлены следующие ключевые параметры, влияющие на возможность арбитража:

— различия в ценах: арбитраж возможен, если есть различия в ценах на один и тот же актив между криптобиржей и банковским рынком; это может быть вызвано различиями в ликвидности, спросе и предложении на обоих

рынках;

— комиссии: нужно учитывать комиссии на криптобирже и банковском рынке; некоторые операции могут стоить больше из-за комиссий, что важно для расчета потенциальной прибыли от арбитража;
— ликвидность: банковский рынок и криптобиржа могут иметь разную ликвидность; в более ликвидных рынках арбитраж может быть более эффективным, так как они обычно имеют более узкие спреды.

## 2.2 Мониторинг комиссии по переводам криптовалют

### 2.2.1 Постановка задачи

Собрать информацию о комиссиях по переводу различных монет (список предоставлен) на биржах Binance, Bybit, OKX и Huobi.

### 2.2.2 Решение

Перейдём сразу к написанию кода и соберём имеющиеся данные в одно место (приведён пример для биржи Binance) [5]:

```
import requests
import hashlib
import hmac
import time

# установка API ключа и секрета Binance
# (скрыты в целях безопасности)
binance_api_key = ''
binance_api_secret = ''

# функция для создания подписи HMAC-SHA256 для Binance API
def generate_binance_signature(api_secret, params):
    query_string = '&'.join(f'{key}={value}' for key, value in params.items())
    signature = hmac.new(api_secret.encode('utf-8')
                , query_string.encode('utf-8'), hashlib.sha256).hexdigest()
    return signature

# функция для получения данных о комиссиях на спотовой торговле с Binance
def get_binance_spot_commissions():
    # устанавливаем endpoint для получения информации о комиссиях
    endpoint = 'https://api.binance.com/api/v3/account'
```

```
# устанавливаем необходимые параметры запроса
params = {
    'timestamp': int(time.time() * 1000),  # временная метка в миллисекундах
    'recvWindow': 5000  # Окно времени для запроса (5 секунд)
}

# создаем подпись HMAC-SHA256
params['signature'] = generate_binance_signature(binance_api_secret, params)

try:
    # выполняем GET-запрос к Binance API
    response = requests.get(endpoint, params=params
                            , headers={'X-MBX-APIKEY': binance_api_key})
    response.raise_for_status()

    # возвращаем данные о комиссиях
    data = response.json()
    commissions = {}

    # извлекаем комиссии для каждого актива
    for asset in data['balances']:
        symbol = asset['asset']
        free = float(asset['free'])
        locked = float(asset['locked'])

        if free + locked > 0:
            commissions[symbol] = {
                'free': free,
                'locked': locked,
                'total': free + locked,
            }

    return commissions

except requests.exceptions.RequestException as e:
    print(f"Ошибка при получении данных с Binance: {e}")
    return None
```

Подпись HMAC (Hash-based Message Authentication Code) используется для обеспечения целостности и подлинности данных при обмене информацией между клиентом и сервером. В контексте API запросов, подпись HMAC обычно используется для создания электронной подписи, которая прикрепляется к запросу и позволяет серверу проверить, что запрос действительно был создан от имени определенного клиента и не был изменен в процессе передачи.

Далее необходимо перенести собранные данные в таблицу Excel. Она выглядит следующим образом (на рис. 1 представлена часть):

| | | Binance | | | | | | | | Bybit | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID в БД | Криптовалюта | Сеть | Комиссия out | Сеть | Комиссия out | Сеть | Комиссия out | Сеть | Комиссия out | Сеть | Комиссия out | Сеть | Комиссия out | Сеть | Комиссия out | Сеть | Комиссия out |
| 6644 | CTXC | | | | | | | CTXC | 0,1 | | | | | | | | |
| 83 | DASH | | | | | | | DASH | 0,0016 | | | | | | | | |
| 6645 | DEP | | | | | | | | | | | | | | | | |
| 9 | DOT | BEP20 | 0,014 | | | | | DOT | 0,08 | | | | | | | DOT | 0,1 |
| 6056 | EGLD | BEP20 | 0,0019 | | | | | EGLD | 0,0008 | | | | | | | EGLD | 0,001 |
| 5894 | ENJ | | | | | ERC20 | 10 | | | | | | | ERC20 | 12 | | |
| 6565 | EURT | | | | | | | | | | | | | | | | |
| 67 | FIL | BEP20 | 0,013 | | | ERC20 | 0,79 | FIL | 0,001 | | | | | | | Filecoin | 0,001 |
| 6635 | FLR | | | | | | | FLR | 0,1 | | | | | | | FLR | 0,01 |
| 6647 | FTM | BEP20 | 0,2 | | | ERC20 | 10 | FTM | 0,3 | | | | | ERC20 | 12 | FTM | 0,01 |
| 6633 | GALA | | | | | ERC20 | 128 | | | | | | | ERC20 | 142 | | |
| 5956 | ICP | | | | | | | ICP | 0,0003 | | | | | | | ICP | 0,006 |
| 6648 | JOE | | | | | AVAXC | 0,1 | ARBITRUM | 1 | | | | | | | | |
| 6649 | KLAY | | | | | | | KLAY | 0,005 | | | | | | | KLAY | 0,01 |
| 6 | LINK | BEP20 | 0,013 | | | ERC20 | 0,51 | | | | | | | ERC20 | 1,12 | | |
| 6153 | LUNA | | | | | | | LUNA | 0,01 | выводы этого актива временно недоступны | | | | | | | |
| 6003 | MATIC | BEP20 | 0,074 | | | ERC20 | 9,72 | MATIC | 0,1 | BEP20 | 0,08 | | | ERC20 | 10 | MATIC | 0,1 |
| 5893 | NEAR | BEP20 | 0,039 | | | | | NEAR | 0,01 | | | | | | | NEAR | 0,01 |
| 6631 | OKT | | | | | | | | | | | | | | | | |
| 6113 | SC | | | | | | | SC | 0,1 | | | | | | | SC | 1 |
| 6062 | SHIB | BEP20 | 7306 | | | ERC20 | 413612 | | | BEP20 | 1500 | | | ERC20 | 460000 | | |
| 175 | TRX | BEP20 | 1,31 | TRC20 | 1 | | | | | | | TRC20 | 1 | | | | |
| 43 | USDC | BEP20 | 0,145 | TRC20 | 1 | ERC20 | 4,18 | AVAXC | 1 | BEP20 | 0,2 | TRC20 | 1 | ERC20 | 5 | AVAXC | 1 |
| 6040 | XTZ | BEP20 | 0,02 | | | | | XTZ | 0,1 | | | | | | | XTZ | 1 |
| 145 | USDT | BEP20 | 0,29 | TRC20 | 2 | ERC20 | 4,18 | BEP2 | 0,8 | BEP20 | 0,3 | TRC20 | 1 | ERC20 | 5 | AVAXC | 0,3 |
| 5924 | AVA | BEP20 | 0,12 | BEP2 | 0,26 | | | | | | | | | ERC20 | 8 | | |
| 5937 | OKSE | | | | | | | | | BEP20 | 1 | | | | | | |
| 5938 | BIT | | | | | | | | | | | | | ERC20 | 8 | | |
| 666 | TWT | BEP20 | 0,067 | | | | | | | BEP20 | 0,2 | | | | | | |
| 1994 | CAKE | BEP20 | 0,023 | | | | | APT | 0,01 | BEP20 | 0,016 | | | | | | |
| 2319 | BUSD | BEP20 | 0,7 | TRC20 | 1 | ERC20 | 4,18 | BEP2 | 0,7 | BEP20 | 0,5 | | | ERC20 | 3 | | |
| 6646 | DNA | | | | | | | | | | | | | | | | |

Рисунок 1 – Таблица с данными о комиссиях

## 2.3 Загрузка информации в базу данных

### 2.3.1 Постановка задачи

Загрузить собранные данные о комиссиях в базу данных.

### 2.3.2 Решение

На проекте используется СУБД PostgreSQL и ПО DBeaver.

PostgreSQL — это мощная объектно-реляционная система управления базами данных (СУБД). Она является СУБД с открытым исходным кодом и известна своей надежностью, производительностью и возможностями расширения. PostgreSQL распространяется под лицензией, которая позволяет свободно использовать, изменять и распространять код; сочетает в себе преимущества реля-

ционных баз данных с поддержкой сложных типов данных, индексов и функций, что делает его мощным средством хранения и обработки данных; имеет множество расширений и модулей, которые позволяют расширить функциональность PostgreSQL в соответствии с потребностями конкретного проекта [6].

Также PostgreSQL обладает высокой производительностью и масштабируемостью, что делает её подходящей для реализуемого проекта, ведь в нём происходит работа с огромным количеством разных типов данных и данных в целом.

DBeaver — это универсальный инструмент для работы с базами данных. Он является клиентом для множества СУБД и предоставляет множество возможностей для удобной работы с данными и структурой баз данных [7].

Перейдём к выполнению задачи. Для начала полученный ранее файл формата Excel нужно переписать в более удобную форму (рис. 2):

| ID монеты | Монета | ID промежуточной платформы | Платформа | ID биржи | Биржа | Комиссия |
|---|---|---|---|---|---|---|
| 158 | BTC | создать | BTC | 461 | Binance | 0,0001 |
| | | | | 600 | Bybit | 0,0005 |
| | | | | 460 | OKX | 0,00004 |
| | | | | 715 | Huobi | 0,00004 |
| 6022 | ETH | 723 | ERC20 | 461 | Binance | 0,000794 |
| | | | | 600 | Bybit | 0,0019 |
| | | | | 460 | OKX | 0,0006352 |
| 6630 | OKB | 723 | ERC20 | 460 | OKX | 0,4 |
| | | | | 715 | Huobi | 0,5 |
| 6632 | LTC | создать | LTC | 461 | Binance | 0,001 |
| | | | | 600 | Bybit | 0,001 |
| | | | | 460 | OKX | 0,001 |
| | | | | 715 | Huobi | 0,001 |
| 6127 | DOGE | создать | DOGE | 461 | Binance | 4 |
| | | | | 600 | Bybit | 5 |
| | | | | 460 | OKX | 4 |
| | | | | 715 | Huobi | 5 |
| 161 | SOL | создать | SOL | 461 | Binance | 0,008 |
| | | | | 600 | Bybit | 0,01 |
| | | | | 460 | OKX | 0,008 |
| | | | | 715 | Huobi | 0,01 |
| 6636 | ACA | создать | ACA | 461 | Binance | 0,2 |
| | | | | 600 | Bybit | 0,1 |
| | | | | 460 | OKX | 0,1 |
| | | | | 715 | Huobi | 0,02 |

Рисунок 2 – Альтернативный формат таблицы

Далее этот файл необходимо привести к формату, представленному ниже (рис. 3):

| id | rule_num | status | platform_ | platform_ | method | currency_ | currency_ | rate | tax | tax_const | tax_max | tax_min | tran_spee | tran_max | tran_min | update_ra | added | last_chan | bot | author | comment | liquidity |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 577352 | 577352 | N | 715 | 721 | 363 | 249 | 249 | 1 | 0 | 0,000252 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577353 | 577353 | N | 721 | 715 | 363 | 249 | 249 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577354 | 577354 | N | 715 | 721 | 363 | 145 | 145 | 1 | 0 | 0,8 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577355 | 577355 | N | 721 | 715 | 363 | 145 | 145 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577356 | 577356 | N | 460 | 722 | 363 | 145 | 145 | 1 | 0 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577357 | 577357 | N | 722 | 460 | 363 | 145 | 145 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577358 | 577358 | N | 715 | 723 | 363 | 6157 | 6157 | 1 | 0 | 0,153 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577359 | 577359 | N | 723 | 715 | 363 | 6157 | 6157 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577360 | 577360 | N | 460 | 723 | 363 | 6630 | 6630 | 1 | 0 | 0,4 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577361 | 577361 | N | 715 | 723 | 363 | 6630 | 6630 | 1 | 0 | 0,5 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577362 | 577362 | N | 723 | 460 | 363 | 6630 | 6630 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577363 | 577363 | N | 723 | 715 | 363 | 6630 | 6630 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577364 | 577364 | N | 461 | 723 | 363 | 6637 | 6637 | 1 | 0 | 30 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577365 | 577365 | N | 460 | 723 | 363 | 6637 | 6637 | 1 | 0 | 24 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577366 | 577366 | N | 723 | 461 | 363 | 6637 | 6637 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577367 | 577367 | N | 723 | 460 | 363 | 6637 | 6637 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577368 | 577368 | N | 461 | 723 | 363 | 6638 | 6638 | 1 | 0 | 5,42 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577369 | 577369 | N | 600 | 723 | 363 | 6638 | 6638 | 1 | 0 | 6,76 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577370 | 577370 | N | 460 | 723 | 363 | 6638 | 6638 | 1 | 0 | 8,974613 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577371 | 577371 | N | 715 | 723 | 363 | 6638 | 6638 | 1 | 0 | 10,28444 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577372 | 577372 | N | 723 | 461 | 363 | 6638 | 6638 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577373 | 577373 | N | 723 | 600 | 363 | 6638 | 6638 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577374 | 577374 | N | 723 | 460 | 363 | 6638 | 6638 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577375 | 577375 | N | 723 | 715 | 363 | 6638 | 6638 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577376 | 577376 | N | 461 | 723 | 363 | 5894 | 5894 | 1 | 0 | 10 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577377 | 577377 | N | 600 | 723 | 363 | 5894 | 5894 | 1 | 0 | 12 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577378 | 577378 | N | 460 | 723 | 363 | 5894 | 5894 | 1 | 0 | 28,87139 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577379 | 577379 | N | 715 | 723 | 363 | 5894 | 5894 | 1 | 0 | 29,7766 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577380 | 577380 | N | 723 | 461 | 363 | 5894 | 5894 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577381 | 577381 | N | 723 | 600 | 363 | 5894 | 5894 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577382 | 577382 | N | 723 | 460 | 363 | 5894 | 5894 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577383 | 577383 | N | 723 | 715 | 363 | 5894 | 5894 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577384 | 577384 | N | 461 | 731 | 363 | 6003 | 6003 | 1 | 0 | 0,1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577385 | 577385 | N | 600 | 731 | 363 | 6003 | 6003 | 1 | 0 | 0,1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577386 | 577386 | N | 460 | 731 | 363 | 6003 | 6003 | 1 | 0 | 0,48 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577387 | 577387 | N | 715 | 731 | 363 | 6003 | 6003 | 1 | 0 | 0,6 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577388 | 577388 | N | 731 | 461 | 363 | 6003 | 6003 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |
| 577389 | 577389 | N | 731 | 600 | 363 | 6003 | 6003 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 2023-07-0 | 2023-07-0 | false | 23 | | [NULL] |

Рисунок 3 – Файл для загрузки в базу данных

Данный шаг является необходимым для безболезненной загрузки информации в базу данных при имеющихся требованиях к содержанию таблицы.

Последний шаг — непосредственная загрузка данных в нужную таблицу базы данных через DBeaver. Итоговый результат можно увидеть на рис. 4 и 5:

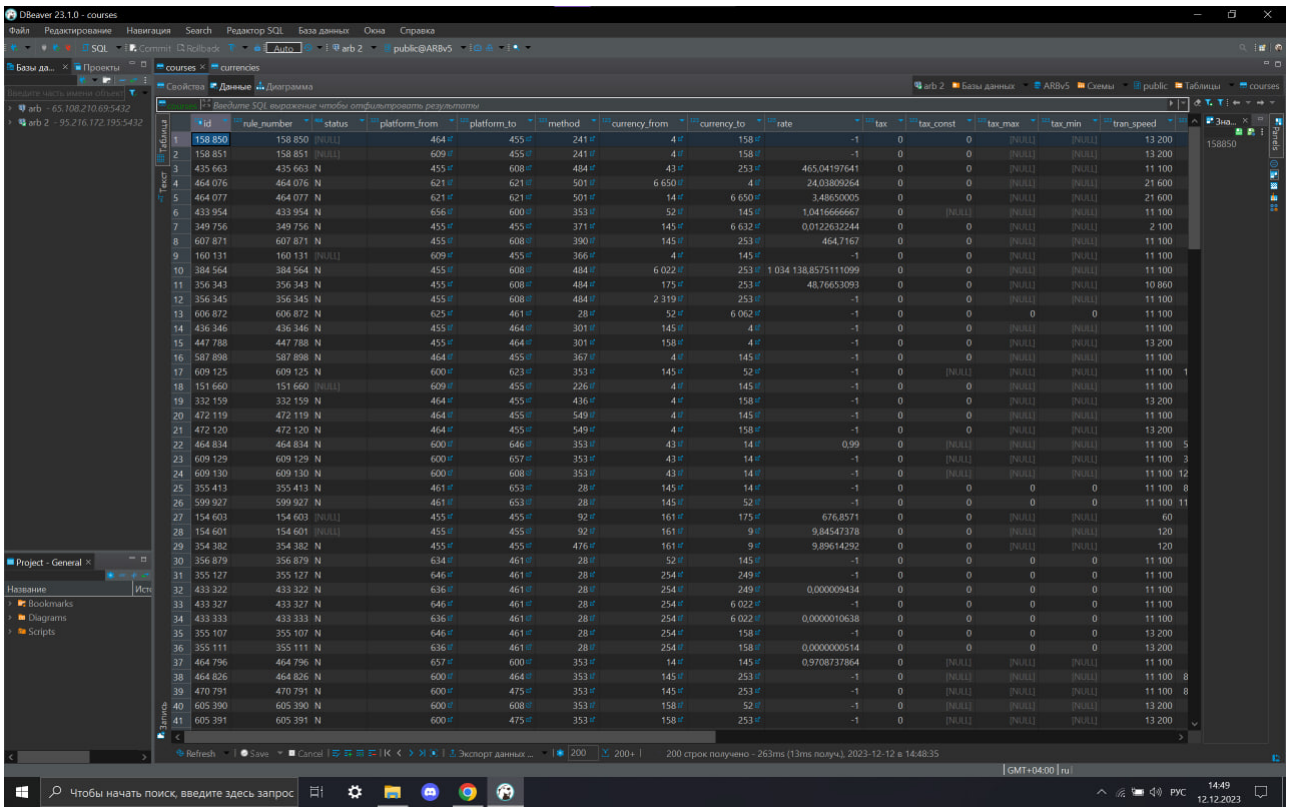

Рисунок 4 – Загруженная информация в базе данных

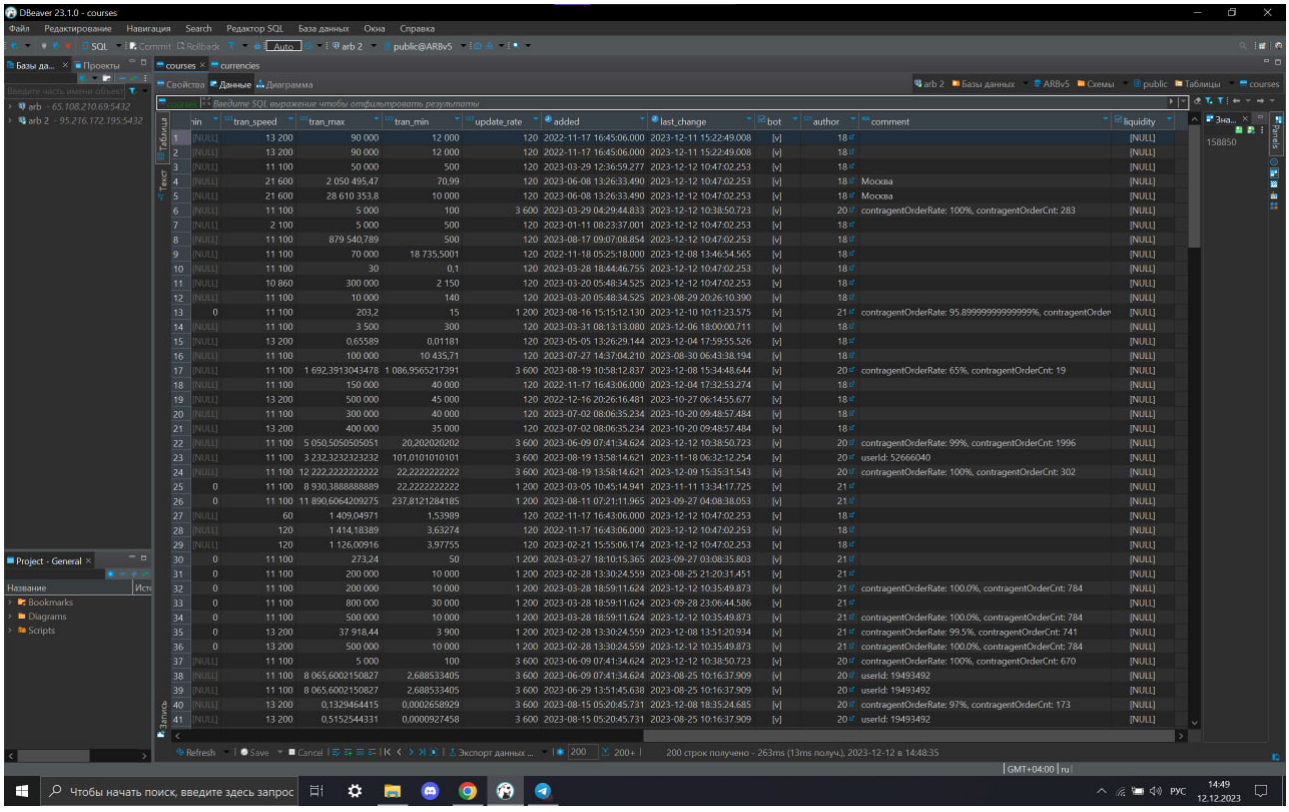

Рисунок 5 – Загруженная информация в базе данных

## 2.4 Анализ информации из базы данных

### 2.4.1 Постановка задачи

Для дальнейшей модификации работы алгоритма необходимо проанализировать имеющуюся информацию. Задача — выполнить следующие запросы к базе данных:

1. вычисление средней волатильности для каждой торговой пары;
2. определение периодов высокой волатильности;
3. расчет скользящей волатильности;
4. выявление связи между объемом и волатильностью;
5. поиск корреляции между волатильностью и другими параметрами;
6. определение топ-5 торговых пар с наибольшим потенциалом арбитража;
7. выявить тренд изменения цен на топ-3 криптовалюты (Bitcoin, Ethereum, Ripple);
8. сравнить популярность торговых пар на различных биржах [8].

### 2.4.2 Решение

Волатильность — статистический финансовый показатель, характеризующий изменчивость цены на что-либо. Волатильность является важнейшим финансовым показателем и понятием в управлении финансовыми рисками, где представляет собой меру риска использования финансового инструмента за заданный промежуток времени.

В процессе решения использовались разные таблицы имеющейся базы данных. Запросы были написаны на языке SQL [9]. Перейдём к решению задач и кодам запросов [10]:

1. вычисление средней волатильности для каждой торговой пары:

```
SELECT
    symbol,
    AVG(ABS(CLOSE - LAG(CLOSE, 1) OVER (PARTITION BY
                                        symbol ORDER BY timestamp)))
        AS average_volatility
FROM
    volat_data
GROUP BY
    symbol;
```

Запрос использует оконную функцию LAG для вычисления разницы меж-

ду текущей и предыдущей ценой для каждой торговой пары и затем находит среднее значение по абсолютным значениям этих различий.

2. определение периодов высокой волатильности;

```
SELECT
    symbol,
    timestamp,
    ABS(CLOSE - LAG(CLOSE, 1) OVER (PARTITION BY symbol ORDER BY timestamp))
        AS volatility
FROM
    volat_data
WHERE
    volatility > (SELECT PERCENTILE_CONT(0.95) WITHIN GROUP
        (ORDER BY volatility) FROM volat_data);
```

Запрос выделяет периоды с высокой волатильностью, фильтруя те записи, где разница между текущей и предыдущей ценой превышает 95-й процентиль волатильности.

3. расчет скользящей волатильности;

```
SELECT
    symbol,
    timestamp,
    STDDEV(ABS(CLOSE - LAG(CLOSE, 1)
    OVER (PARTITION BY symbol ORDER BY timestamp))
    OVER (ORDER BY timestamp ROWS BETWEEN 19 PRECEDING AND CURRENT ROW))
        AS rolling_volatility
FROM
    volat_data;
```

Запрос вычисляет скользящую волатильность с использованием окна из 20 последних записей для каждой торговой пары.

4. выявление связи между объемом и волатильностью;

```
SELECT
    symbol,
    timestamp,
    AVG(ABS(CLOSE - LAG(CLOSE, 1) OVER (PARTITION BY
                                        symbol ORDER BY timestamp)))
        AS volatility,
    AVG(volume) AS average_volume
FROM
    volat_data
```

```
GROUP BY
    symbol, timestamp;
```

Запрос агрегирует данные по торговым парам и времени, вычисляя среднюю волатильность и средний объем для каждой из них.

5. поиск корреляции между волатильностью и другими параметрами;

```
SELECT
    symbol,
    timestamp,
    AVG(ABS(CLOSE - LAG(CLOSE, 1) OVER (PARTITION BY
                                        symbol ORDER BY timestamp)))
        AS volatility,
    AVG(volume) AS average_volume,
    AVG(open) AS average_open_price
FROM
    volat_data
GROUP BY
    symbol, timestamp;
```

Запрос добавляет еще один агрегат — среднюю цену открытия — и используется для исследования корреляции между волатильностью и другими параметрами торговых пар.

6. определение топ-5 торговых пар с наибольшим потенциалом арбитража;

```
SELECT
    trading_pair,
    MAX(price_difference) AS max_price_difference
FROM (
    SELECT
        trading_pair,
        MAX(price) - MIN(price) AS price_difference
    FROM arbitrage_data
    GROUP BY trading_pair
) AS price_diff
ORDER BY max_price_difference DESC
LIMIT 5;
```

7. выявить тренд изменения цен на топ-3 криптовалюты (Bitcoin, Ethereum, Ripple);

```
SELECT
    cryptocurrency,
    timestamp,
```

```
    price
FROM price_data
WHERE cryptocurrency IN ('Bitcoin', 'Ethereum', 'Ripple')
    AND timestamp >= NOW() - INTERVAL '30 days'
ORDER BY cryptocurrency, timestamp;
```

8. сравнить популярность торговых пар на различных биржах.

```
WITH ranked_pairs AS (
    SELECT
        exchange_name,
        trading_pair,
        trade_volume,
        RANK() OVER (PARTITION BY exchange_name ORDER BY trade_volume DESC)
            AS pair_rank
    FROM trade_data
)
SELECT
    exchange_name,
    trading_pair,
    trade_volume
FROM ranked_pairs
WHERE pair_rank <= 5;
```

# ЗАКЛЮЧЕНИЕ

Таким образом, внесенные в проект изменения улучшили качество работы алгоритма и проекта в целом, цель производственной практики была достигнута, а все поставленные в ходе практики задачи решены.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1 Как устроен арбитраж криптовалюты и какие программы понадобятся инвестору [Электронный ресурс]. — URL: https://www.banki.ru/news/daytheme/?id=10991303 (Дата обращения 01.07.2023). Загл. с экр. Яз. рус.

2 Python и API: превосходное комбо для автоматизации работы с публичными данными [Электронный ресурс]. — URL: https://proglib.io/p/python-i-api-prevoshodnoe-kombo-dlya-avtomatizacii-raboty-s-publichnymi-dannymi-2021-02-26 (Дата обращения 04.07.2023). Загл. с экр. Яз. рус.

3 *Бейдер, Д.* Чистый Python. Тонкости программирования для профи / Д. Бейдер. — Питер, 2022. — С. 288.

4 Python 3.12.1 documentation [Электронный ресурс]. — URL: https://docs.python.org/3/ (Дата обращения 12.07.2023). Загл. с экр. Яз. рус.

5 Работа с файлами в формате JSON [Электронный ресурс]. — URL: https://pyneng.readthedocs.io/ru/latest/book/17_serialization/json.html (Дата обращения 11.07.2023). Загл. с экр. Яз. рус.

6 PostgreSQL: Документация [Электронный ресурс]. — URL: https://postgrespro.ru/docs/postgresql (Дата обращения 04.07.2023). Загл. с экр. Яз. рус.

7 About: DBeaver Community [Электронный ресурс]. — URL: https://dbeaver.io/about/ (Дата обращения 03.07.2023). Загл. с экр. Яз. рус.

8 Показатели волатильности. Чем они полезны для трейдеров и инвесторов [Электронный ресурс]. — URL: https://bcs-express.ru/novosti-i-analitika/pokazateli-volatil-nosti-chem-oni-polezny-dlia-treiderov-i-investorov (Дата обращения 09.07.2023). Загл. с экр. Яз. рус.

9 Запросы WITH (Общие табличные выражения) [Электронный ресурс]. — URL: https://postgrespro.ru/docs/postgresql/10/queries-with (Дата обращения 03.07.2023). Загл. с экр. Яз. рус.

10 *Уолтер, Ш.* SQL. Быстрое погружение / Ш. Уолтер. — Питер, 2022. — С. 224.