

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные базы данных»
Тема: Криптовалюты Neo4j

Студент гр. 8304	_____	Холковский К.В.
Студент гр. 8304	_____	Птухов Д.А.
Студент гр. 8304	_____	Воропаев А.О.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург

2021

ЗАДАНИЕ

Студенты:

Холковский К.В.

Птухов Д.А.

Воропаев А.О.

Группа 8304

Тема проекта: криптовалюта Neo4j

Исходные данные:

Необходимо реализовать информационную систему, выполняющую задачи хранения истории курсов (и динамической подгрузки в БД) криптовалют, анализа истории (примитивные критерии), оценки правильности решений по покупке продаже в определенные моменты прошлого.

Содержание пояснительной записки:

Содержание

Введение

Качественные требования к решению

Сценарий использования

Модель данных

Разработка приложения

Вывод

Приложение

Предполагаемый объем пояснительной записки:

Не менее 10 страниц.

Дата выдачи задания:

Дата сдачи реферата:

Дата защиты реферата:

Студент	_____	Холковский К.В.
---------	-------	-----------------

Студент	_____	Птухов Д.А.
---------	-------	-------------

Студент	_____	Воропаев А.О.
---------	-------	---------------

Преподаватель	_____	Заславский М.М.
---------------	-------	-----------------

АННОТАЦИЯ

В рамках данного курса предполагалось разработать WEB приложение в команде на одну из поставленных тем. Была выбрана тема создания приложения для симуляции покупок криптовалют, данные хранятся при помощи Neo4j. Найти исходный код и всю дополнительную информацию можно по [ссылке](#).

ANNOTATION

Within the framework of this course, it was supposed to develop a WEB application in a team on one of the topics. The topic of creating an application for simulating purchases of cryptocurrencies was chosen, the data is stored using Neo4j. You can find the source code and all additional information [here](#).

СОДЕРЖАНИЕ

	Введение	6
1.	Качественные требования	6
2.	Сценарий использования	6
3.	Макет	9
4.	Модель данных	11
5.	Разработка приложения	15
	Заключение	17
	Список использованных источников	18

ВВЕДЕНИЕ

Цель работы – создать приложения для тренировки навыков трейдерства.

Было решено разработать веб-приложение, которое позволит совершать покупки криптовалют и наблюдать за их ростом.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

Требуется разработать приложение с использованием Neo4j – графической СУБД.

2. СЦЕНАРИЙ ИСПОЛЬЗОВАНИЯ

Сценарий использования - «Просмотр истории операций»

Действующее лицо: Пользователь

Основной сценарий:

- Пользователь заходит на сайт и попадает на главную страницу
- Пользователь нажимает на кнопку портфель и попадает на соответствующую страницу
- Пользователь нажимает на кнопку «История операций» и попадает на соответствующую страницу
- Пользователь видит историю покупок и продаж криптовалют

Альтернативный сценарий:

- Пользователь переходит на главную страницу с помощью кнопки «Trader»
- Пользователь видит уведомление о том, что история покупок пустая

Сценарий использования - «Покупка/продажа криптовалют»

Действующее лицо: Пользователь

Основной сценарий:

- Пользователь заходит на сайт и попадает на главную страницу
- Пользователь нажимает на клиптовалюту, которую желает купить/продать

- Пользователь попадает на страницу покупки/продажи криптовалюты, где видит график динамики изменения курса валюты
- Пользователь вводит кол-во криптовалюты и нажимает на кнопку купить/продать

Альтернативный сценарий:

- При отсутствии криптовалюты в кошельке пользователя при продаже, он видит соответствующее уведомление

Сценарий использования - «Экспорт криптовалют»

Действующее лицо: Пользователь

Основной сценарий:

- Пользователь заходит на сайт и попадает на главную страницу
- Пользователь нажимает на кнопку портфель и попадает на соответствующую страницу
- Пользователь нажимает кнопку «Экспорт»
- У пользователя начинается скачивание базы данных в формате .xml
-

Альтернативный сценарий:

- Пользователь получает уведомление, что база данных пустая и скачать файл невозможно

Сценарий использования - «Импорт криптовалют»

Действующее лицо: Пользователь

Основной сценарий:

- Пользователь заходит на сайт и попадает на главную страницу
- Пользователь нажимает на кнопку портфель и попадает на соответствующую страницу
- Пользователь нажимает кнопку «Импорт»
- У пользователя открывается окно в интерфейсе ОС для выбора файла формата .xml
- После импорта в кошельке пользователя появляются данные из файла

Альтернативный сценарий:

- Пользователь получает уведомление, что файл пустой/некорректный

3. МАКЕТ

1. Портфель криптовалют см. рис 1.

Balance

Trader История операций

Текущий баланс:
300 USD

Валюты в наличии

Импорт Экспорт

Валюта	Количество	Состояние

Рис 1 - Портфель

2. История операций см. рис 2.

History

Trader Портфель

История операций

Введите название валюты 🔍

Вид операции ▾	Валюта	Количество	Дата ▾	Курс

Рис 2 – История операций

3. Покупка криптовалюты см. рис 3.

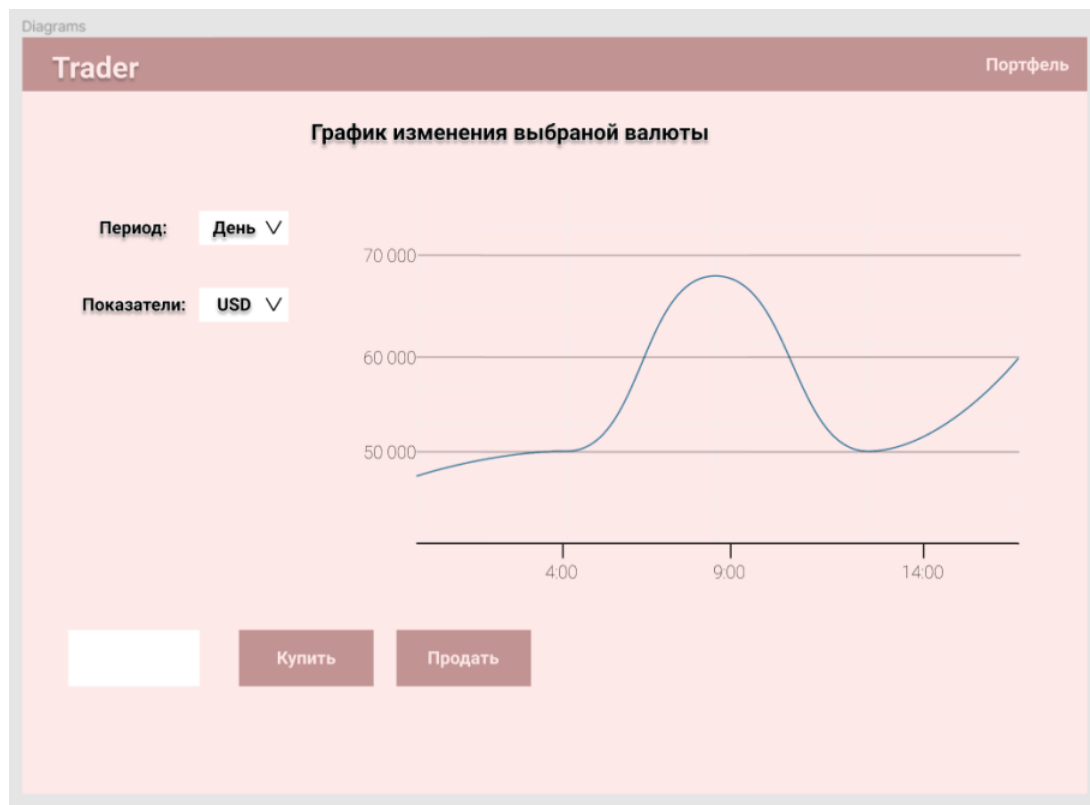


Рис 3 – Покупка валют

4. Рынок на текущий момент см. рис 4.

The screenshot shows a web application titled "Trader" with a "Портфель" (Portfolio) link in the top right. The main heading is "Рынок на текущий момент" (Market at the current moment). Below the heading is a search bar with the placeholder text "Введите название валюты" (Enter the name of the currency) and a magnifying glass icon. Below the search bar is a table with the following columns: "Валюта" (Currency), "Обозначение" (Symbol), "Цена(USD)" (Price(USD)), "Изм.(24ч)" (Change(24h)), and "Изм.(7д)" (Change(7d)). The table is currently empty, showing only the column headers.

Валюта	Обозначение	Цена(USD)	Изм.(24ч)	Изм.(7д)

Рис 4 – Рынок на текущий момент

5. Макет переходов по страницам см. рис 5.

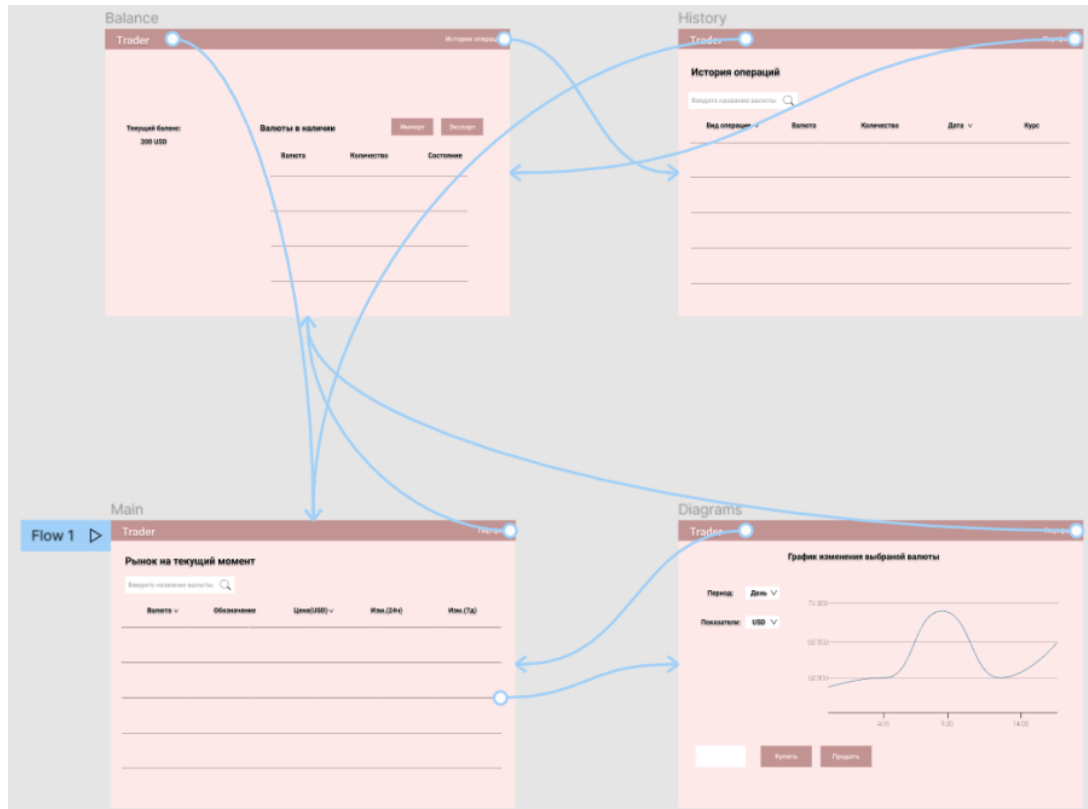


Рис 5 – Макет переходов

4. МОДЕЛЬ ДАННЫХ

Нереляционная модель

Графическое представление



Описание сущностей и типов данных

Сущность Client. Атрибуты:

- name String 20 - Имя клиента
- balance Float 4 - Запас средств на счету клиента в USD

Сущность Crypto. Атрибуты:

- name String 20 - Название криптовалюты
- symbol String 10 - Обозначение криптовалюты

Связь Operation. Атрибуты:

- type Boolean 1 - Тип операции true - покупка, false - продажа
- count Float 4 - Сколько валюты участвует в операции
- price Float 4 - Цена покупки/продажи USD
- date Integer 4 - TimeStamp выполнения операции

Связь Have. Атрибуты:

- count Float 4 - Сколько валюты имеем на данный момент
- spending Float 4 - Сумма затраченная на ее приобретение USD

Избыточность модели

Пусть N,M,K,L это количество записей для Client, Crypto, Operation и Have, соответственно. Пусть id будет весить 4 байта, тогда получим:

$$(4 + 20 + 4)N + (4 + 20 + 10)M + (4*3 + 1 + 4 + 4 + 4)K + (4*3 + 4 + 4)L = 28N + 34M + 25K + 20L$$

Пусть $M = 2N$, $K = 2M$, $L=M$; Получим: $236N$

Чистые данные не содержат id, тогда: $152N$

Вычислим отношение "фактического" и чистого объемов данных:

$$236 / 152 = 1.55$$

Т.к. в данной модели преобладают связи, а в них содержится сразу 3 id, то модель сильно увеличивает в объеме.

Направление роста

По клиентам: 24b

По криптовалютам: 30b

По связи "Операция": 13b

По связи "Имеет": 8b

Запросы

Просмотр истории операций:

```
MATCH (n:Client)-[r:Operation]->(m:Crypto) RETURN r.type, m.name, r.count, r.date, r.price
```

Продажа криптовалют:

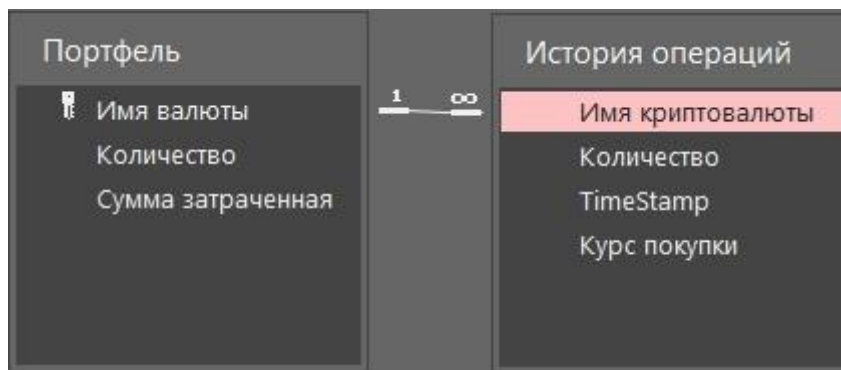
```
MATCH (n:Client)-[r:Have]->(m:Crypto { name:"BitCoin" }) CREATE (n)-[:Operation { count:0.001, price:140, date:1624523131, type:false}]->(m) SET r.count = r.count - 0.001, r.spending = r.spending - 140, n.balance = n.balance + 140
```

Просмотр портфеля криптовалют:

```
MATCH (n:Client)-[r:Have]->(m:Crypto) RETURN m.name, r.count, r.spending
```

Реляционная модель

Схема данных



Описание таблиц и типов данных

Таблица Портфель. Поля:

- Имя валюты - строка, содержащая в себе название валюты, тип SHORT TEXT. Максимальное кол-во символов - 20.
- Количество - вещественное число, содержащее в себе кол-во валюты доступной на данный момент, тип NUMBER.
- Затраченная сумма - вещественное число, содержащее в себе сумму, затраченную на покупку валюты, тип NUMBER.

Таблица История операций. Поля:

- Имя криптовалюты - строка, содержащая в себе название валюты, тип SHORT TEXT. Максимальное кол-во символов - 20.

- Количество - вещественное число, содержащее в себе кол-во купленной/проданной криптовалюты, тип NUMBER.
- TimeStamp - число, содержащее в себе информацию о времени покупки, тип NUMBER.
- Курс покупки - вещественное число, содержащее в себе информацию о курсе на момент покупки, тип NUMBER.

Избыточность модели

Пусть L, K - кол-во записей для таблиц История операций и Портфель, соответственно. Тогда общий вес: $(20 + 4 + 4)K + (20 + 4 + 4 + 4)L = 28K + 32L$
 Пусть $L = 2K$, тогда общий вес будет равен $92K$ Так как поле Имя криптовалюты дублируется в обеих таблицах (для обеспечения связи между ними) одно из этих полей является излишним, тогда чистый вес: $52K$
 Отношение фактического и чистого объема данных: $92/52 = 1.76$

Направление роста

По таблице Портфель - 28b

По таблице История операций - 12b

Запросы

Просмотр Истории операций:

```
SELECT [История операций].[Имя криптовалюты], [История
операций].Количество, [История операций].TimeStamp, [История
операций].[Курс покупки] FROM [История операций];
```

Просмотр Портфеля:

```
SELECT Портфель.[Имя валюты], Портфель.Количество, Портфель.[Сумма
затраченная] FROM Портфель;
```

Продажа криптовалют (2 последовательных запроса):

```
INSERT INTO [История операций] ( [Имя криптовалюты], Количество,
[TimeStamp], [Курс покупки] ) SELECT [Bitcoin], [0.001], [1687236872], [140];
UPDATE Портфель SET Портфель.Количество = [Портфель].[Количество]-
0.001, Портфель.[Сумма затраченная] = [Портфель].[Сумма затраченная]-0.001
WHERE (((Портфель.[Имя валюты])="Bitcoin"));
```

Сравнение моделей

Можно заметить, что модели хранения SQL и NOSQL схожи. Только в модели SQL необходимо дублирование данных.

Пример хранения данных

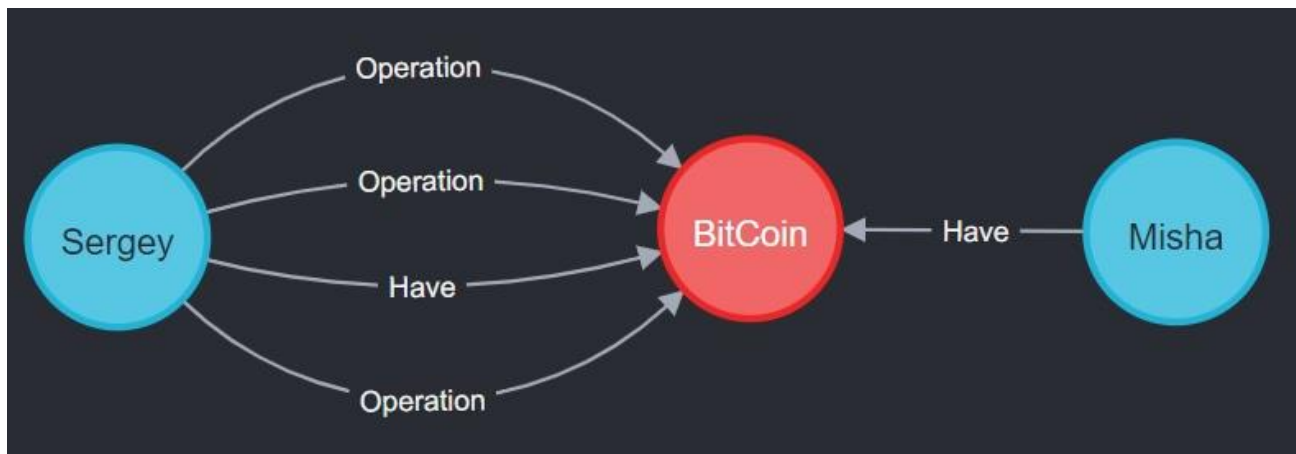


Рис 6 – Пример хранения данных

5. РАЗРАБОТКА ПРИЛОЖЕНИЯ

Для создания приложения был использован веб-фреймворк Flask и публичное апи криптовалют CoinCap.io В бд хранятся операции выполненные пользователем.

Trader					Портфель
Рынок на текущий момент					
Type currency					
Валюта	Обозначение	Цена(USD)	Изм.(24ч)	Ранг	
Bitcoin	BTC	50873.3002111192390190	0.1362720847231236	1	
Ethereum	ETH	4076.2223170445418660	-1.0066903793857187	2	
Binance Coin	BNB	548.1738577197368459	-0.5294287848460818	3	
Tether	USDT	1.0023928433727961	-0.1695338129498313	4	
Solana	SOL	198.3864356567131411	2.1025959417913966	5	
Cardano	ADA	1.4573443435121362	-0.1156219426878817	6	
USD Coin	USDC	1.0012832008843909	-0.2059914719860804	7	
XRP	XRP	0.9218876707171185	-1.0330389913675650	8	
Terra	LUNA	99.8614977016954438	0.9067089563779458	9	
Polkadot	DOT	31.4410420127472922	8.1260439202470450	10	
Avalanche	AVAX	115.5128740423580382	-0.1563010675980208	11	
Dogecoin	DOGE	0.1905551645315928	-0.9414478116324800	12	

Рис 7 – Рынок на текущий момент

Trader			История операций	
Текущий баланс 300 USD			Валюты в наличии	
			Валюта	Количество
			bitcoin	13.0
			ethereum	0.002
			Состояние	
			656638.3041841636	
			8.183816938300875	
			Export	Import
			Download	Choose file
			Submit	

Рис 8 - Портфель

Trader

Портфель

История операций

Type currency

Вид операции	Валюта	Количество	Дата	Цена
Buy	bitcoin	4.0	1640549845554	201954.28462985624
Sell	bitcoin	3.0	1640549820542	151481.03252763685
Buy	ethereum	0.002	1640549712776	8.183816938300875
Buy	bitcoin	2.0	1640549691998	101027.50868032404
Sell	bitcoin	2.0	1640549690455	101027.50868032404
Buy	bitcoin	12.0	1640549674978	606165.0520819442

Рис 9 – История операций

Trader		История операций					
Ethereum 4077.2927725007795443							
<div> <div>Форма ввода</div> <div> <div>Выбор дня</div> <div>Type in amount to buy/sell</div> <div> <div>Купить</div> <div>Продать</div> </div> </div> </div> <div> <div>THERE WILL</div> <div>BE GRAPH!</div> </div>							

Рис 10 – Покупка валют

ЗАКЛЮЧЕНИЕ

Исходя из сравнения, проведенного выше, можно сделать вывод, что нереляционные СУБД в данной задаче имеют преимущество в количестве занимаемой памяти и количестве запросов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

<https://docs.coincap.io/>

<https://neo4j.com/>

<https://flask.palletsprojects.com/en/2.0.x/>