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

КУРСОВАЯ РАБОТА

по дисциплине «Введение в нереляционные базы данных»

Тема: ИС сравнения компаний из индекса S&P 500 и NASDAQ

Студент гр. 8304 _____ Рыжиков А.В.

Студент гр. 8304 _____ Мешков М.А.

Преподаватель _____ Политова А.В.

Санкт-Петербург
2021

ВВЕДЕНИЕ

ИС должна предоставлять пользователю финансовые данные компании и позволять сравнивать компании по ряду финансовых показателей. Для сбора данных будет использован Yahoo Finance API.

Функции ИС:

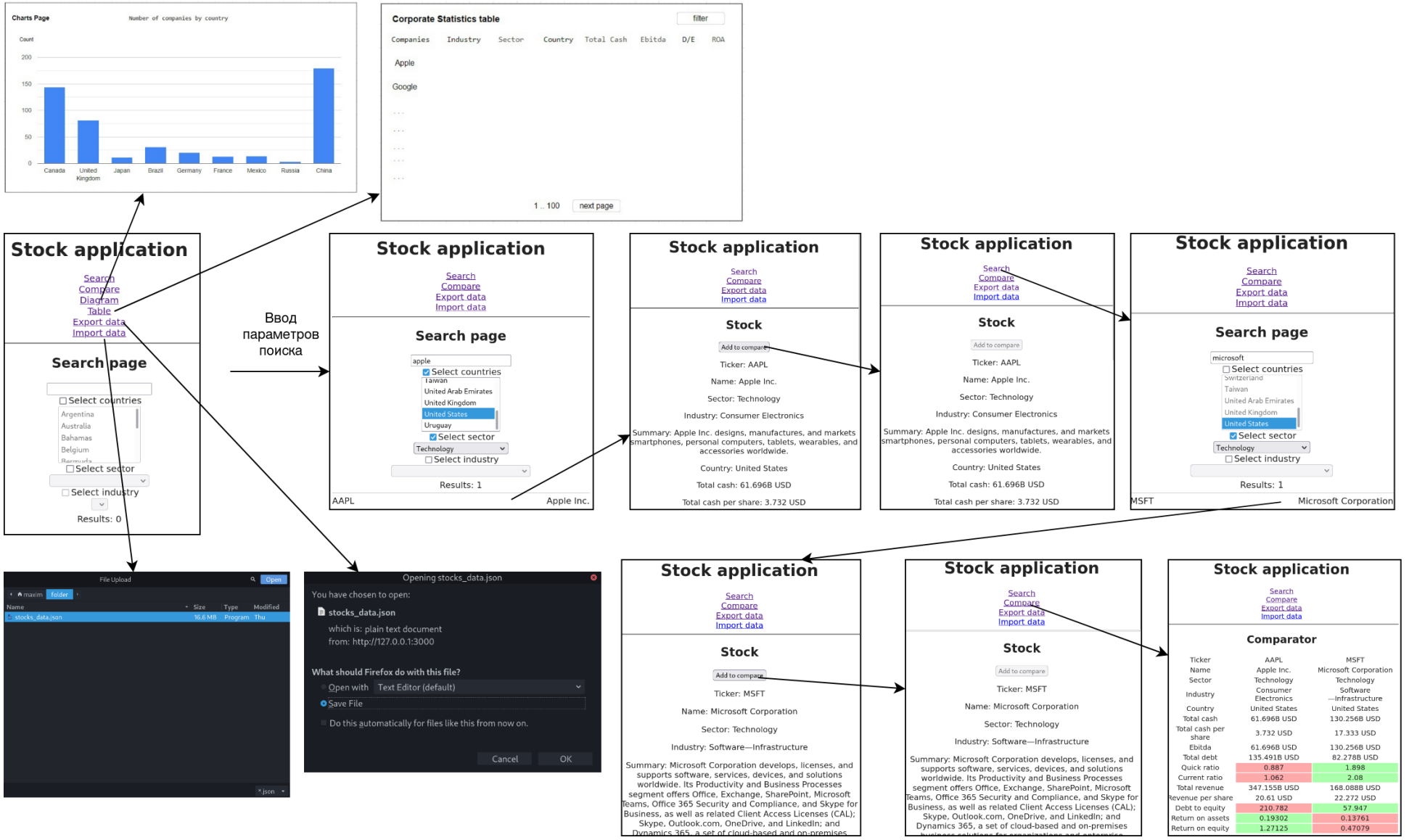
1) Предоставлять пользователю актуальные финансовые показатели компаний
2) Сравнивать компании по финансовым показателям
3) Предоставлять аналитику по компаниям

КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

Требуется реализовать веб-приложение, использующее СУБД MongoDB. Фронтенд реализуется на фреймворке Vue.js, бекенд реализуется в виде сервиса на языке Golang.

СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

UI mockup



Use cases

Получение информации о компании

Действующее лицо: Пользователь
Основной сценарий:

1. Пользователь вводит информацию о компании: фрагмент тикера или названия компании, её страну, сектор, индустрию (все параметры опциональны, чем больше информации - тем конкретнее результаты).
2. В результатах поиска пользователь нажимает на интересующую его компанию.
3. Открывается информация о компании.
4. Пользователь может выбрать показать данные по прибыли и выручке поквартально или по годам.

Получение списка компаний с заданными характеристиками

Действующее лицо: Пользователь
Основной сценарий:

1. Пользователь вводит ограничения на характеристики компании: фрагмент тикера или названия компании, её страну, сектор, индустрию (все параметры опциональны, чем больше информации - тем конкретнее результаты).
2. Пользователь видит список компаний (тикер и название), удовлетворяющих условиям поиска, а также их количество.

Сравнение двух компаний

Действующее лицо: Пользователь
Основной сценарий:

1. Пользователь вводит информацию о компании: фрагмент тикера или названия компании, её страну, сектор, индустрию (все параметры опциональны, чем больше информации - тем конкретнее результаты).
2. В результатах поиска пользователь нажимает на интересующую его компанию.
3. Открывается информация о компании.
4. Пользователь с помощью кнопки добавляет компанию в список для сравнения.
5. Пользователь переходит на экран поиска (начальный экран) и повторяет 1-4 для другой компании.
6. Пользовать переходит на экран сравнения двух компаний и видит сравнение.

Альтернативный сценарий:

- После сравнения двух компаний пользователь добавляет в сравнение еще одну компанию. В этом случае добавленная компания заменяет компанию, добавленную раньше всех, в списке компаний для сравнения.

Построения диаграмм по компаниям

Действующее лицо: Пользователь

1. Пользователь выбирает категорию.

2. Строятся столбчатые диаграммы по выбранной категории.

Табличное представление списка компаний (таблица со всеми полями + фильтр)
Действующее лицо: Пользователь

1. По умолчанию отображаются H самых важных полей, но пользователь может в интерфейсе настроить какие именно поля отображать.
2. Пользователь может выбрать страницу таблицы для просмотра. Предполагается реализовать многостраничное отображение таблицы (отображать на странице, например 10 записей, а под таблице делать ссылки на страницы - Pagination)
3. Пользователь может применить фильтр. Фильтр реализуется как набор полей для ввода в пользовательском интерфейсе (каждое поле соответствует определенному полю в БД), где указывается значение для поиска по соотв. полю.

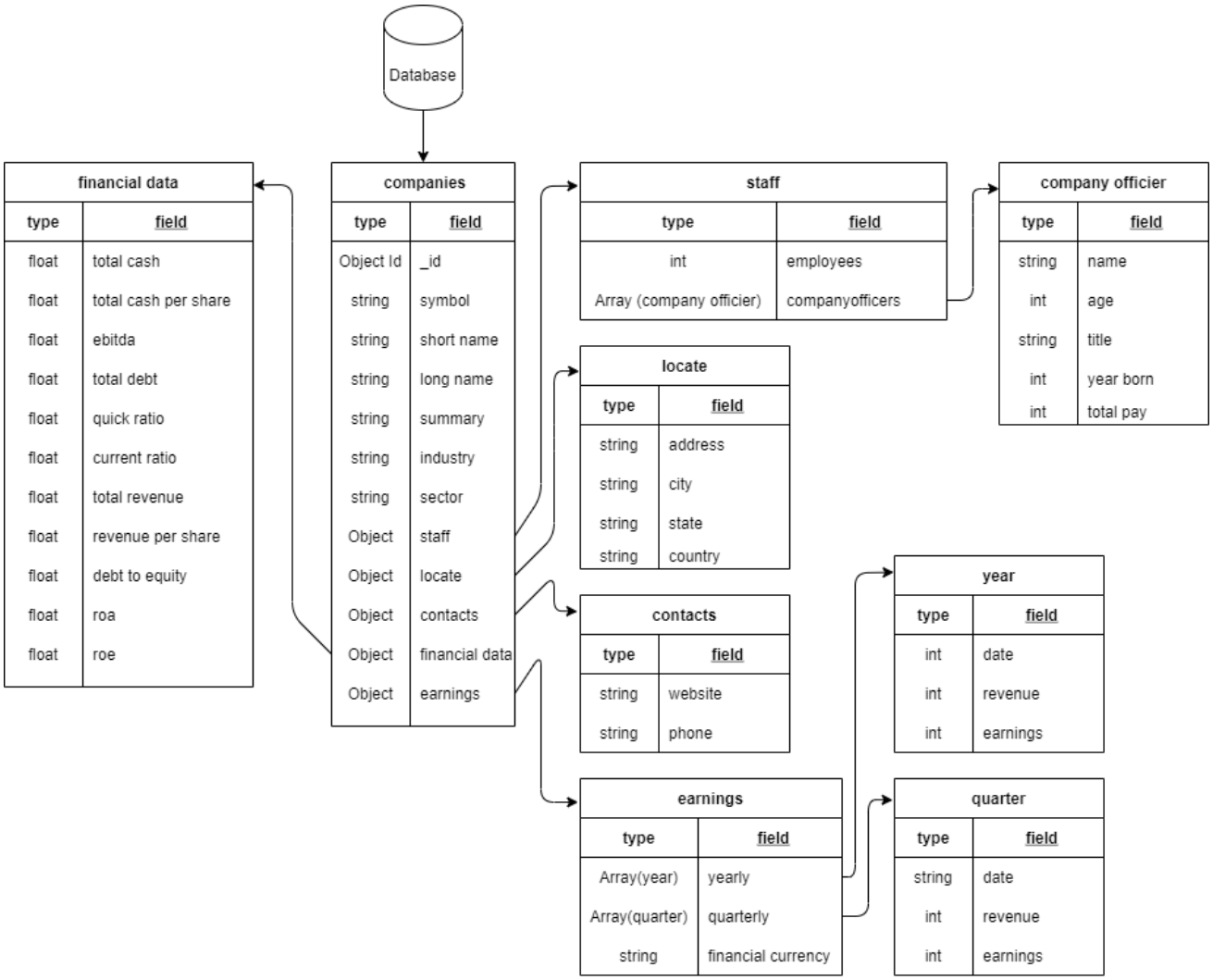
Экспорт данных
Действующее лицо: Пользователь

1. Пользователь нажимает на кнопку экспорта данных.
2. Пользователь выбирает где сохранить файл с данными.

Импорт данных
Действующее лицо: Пользователь

1. Пользователь нажимает на кнопку импорта данных.
2. Пользователь выбирает файл с данными.
3. Данные в приложении обновляются в соответствии с данными из файла.

МОДЕЛЬ ДАННЫХ



Описание назначений коллекций, типов данных и сущностей

БД содержит 1 коллекцию "companies"

Коллекция **companies** содержит объекты, которые имеют указанную ниже структуру * _id - уникальный идентификатор документа в коллекции

- symbol - тикер акции компании на бирже
- short name - краткое название компании
- long name - полное название компании
- summary - описание компании
- industry - индустрия, к которой относится компания
- sector - сектор (подкатегория для индустрии), к которой относится компания
- staff - объект хранящий информацию о сотрудниках компании
- employees - количество сотрудников в компании
- companyofficers - массив объектов companyofficer
- companyofficer - руководитель компании (топ-менеджер)
- name - имя
- age - возраст

- title - должность
- year born - год рождения
- total pay - заработная плата за год
- locate - объект хранящий информацию о месте расположении компании
- address - адрес
- city - город
- state - штат или регион
- country - страна
- contacts - объект хранящий информацию о контактах компании
- phone - телефон компании
- website - ссылка на сайт компании
- financial data - объект хранящий информацию о финансовых данных компании
- total cash - общая сумма денежных средств на счетах компании
- total cash per share - общая сумма денежных средств на акцию
- ebitda - EBITDA (Earnings before interest, taxes, depreciation and amortization)
- total debt - совокупный долг компании
- quick ratio -коэффициент быстрой ликвидности
- current ratio - коэффициент текущей ликвидности
- total revenue - общий доход
- revenue per share- общий доход на акцию
- debt to equity - мультипликатор D/E (Debt to Equity ratio)
- roa - Return on Assets, ROA - рентабельность активов
- roe - Return on Equity, ROE - рентабельность собственного капитала
- earnings
- yearly - массив данных доходов за последние 4 года
- date - указанный год
- revenue - выручка за год
- earnings - прибыль за год
- quarterly - массив данных доходов за последние 4 квартала
- date - указанный год и квартал
- revenue - выручка за квартал
- earnings - прибыль за квартал
- financial currency - валюта в которой указаны доходы

Оценка объема

Объем для одного объекта массива companyofficers

- name - string - 20 символов, 20 байт
- age - int - 4 байта
- title - string - 30 символов, 30 байт
- year born - int - 4 байта
- total pay - int - 4 байта

V(companyofficer) = 62 байта

Объем для одного объекта staff

- employees - int - 4 байта
- companyofficers - массив (приблизительно 10 объектов), 10 * V(companyofficer) = 620 байт

V(staff) = 624 байта

Объем для одного объекта locate

- address - string - 30 символов, 30 байт
- city - string - 20 символов, 20 байт
- state - string - 2 символа, 2 байт
- country - string - 10 символов, 10 байт

V(locate) = 52 байта

Объем для одного объекта contacts

- phone - string - 12 символов, 12 байт
- website - string - 25 символов, 25 байт

V(contacts) = 27 байт

Объем для одного объекта financial data

- total cash - double - 8 байта
- total cash per share - double - 8 байта
- ebitda - double - 8 байта
- total debt - double - 8 байта
- quick ratio - double - 8 байта
- current ratio - double - 8 байта
- total revenue - double - 8 байта
- revenue per share - double - 8 байта
- debt to equity - double - 8 байта
- roa - Return on Assets, ROA - double - 8 байта
- roe - Return on Equity, ROE - double - 8 байта
- financial currency - string - 3 символа, 3 байт

V(financial data) = 91 байт

Объем для одного объекта year

- date - string - 4 символа, 4 байта
- revenue - выручка - int - 4 байта
- earnings - прибыль - int - 4 байта V(year) = 12 байт

Объем для одного объекта quarter

- date - string - 6 символов, 6 байт
- revenue - выручка - int - 4 байта
- earnings - прибыль - int - 4 байта

V(quarter) = 14 байт

Объем для одного объекта earnings

- yearly - массив (4 объекта), 10 * V(year) = 120 байт
- quarterly- массив (4 объекта), 10 * V(quarter) = 140 байт
- financial currency - string - 3 символа, 3 байт

V(earnings) = 263 байта

Объем для одного объекта коллекции companies

- _id - ObjectId - 12 байт
- symbol - string - 4 символов, 4 байт
- short name - string - 10 символов, 10 байт
- long name - string - 12 символов, 12 байт
- summary - string - 1150 символов, 1150 байт
- industry - string - 10 символов, 10 байт
- sector - string - 10 символов, 10 байт
- staff - Object - V(staff) = 624 байта
- locate - Object - V(locate) = 52 байта
- contacts - Object - V(contacts) = 27 байт
- financial - Object - dataV(financial data) = 91 байт
- earnings - Object - V(earnings) = 263 байта

V(объект коллекции companies) = 2265 байт
V(объём данных в коллекции companies) = 5078 * V(объект коллекции companies) = 5078 * 2265 байт = 10,96 Мб.
При увеличении количества объектов каждой сущности рост модели будет происходить линейно. Размер данных n компаний равен n * 2265 байт.
Избыточность модели низка, повторяющиеся значительные по объему фрагменты данных найти не удается.
Объем коллекции stocks = 15.4MB. Эта величина почти в 1.5 раза больше чем объем данных, что можно объяснить тем что mongo хранить различные служебные данные в том числе ключи объектов (в BSON).

Запросы к NoSql

Информация о компании с тикером

```
db.stocks.find({"symbol": "AAPL"})
```

В данном запросе получалась информация о компании с тикером AAPL.

Получение списка стран

```
db.stocks.distinct("locate.country")
```

Получение секторов

```
db.stocks.distinct("sector")
```

Получение индустрий в секторе

```
db.stocks.distinct("industry", {"sector": "Healthcare"})
```

В данном запросе получался список индустрий в секторе Healthcare.

Подсчет по заданному критерию

Критерием может быть sector, industry или locate.country

```
db.stocks.aggregate([{$group: {_id: "$sector", amount: {$sum: 1}}}, {$sort: {amount: -1}}])
```

В данном запросе считалось количество компаний в каждом из секторов.

Поиск с фильтрацией

Возвращается краткая характеристика компании.

```
db.stocks.find(
  {
    "locate.country": {$in: ["United States", "Russia"]},
    sector: "Healthcare",
    industry: "Biotechnology",
    $or: [
      {symbol: {$regex: "pharm", $options: "i"}},
      {"long name": {$regex: "pharm", $options: "i"}},
    ],
  },
  {
    "symbol": 1,
    "short name": 1,
    "industry": 1,
    "sector": 1,
    "locate": {
      country: 1,
    },
  }
)
```

В данном запросе искались компании со следующими характеристиками:

* страна United States или Russia
* сектор Healthcare * индустрия Biotechnology
* тикер или название компании содержит "pharm"

Поиск с фильтрацией с pagination

```
db.stocks.aggregate([
  {
    $match: {
      "locate.country": {$in: ["United States", "Russia"]},
      sector: "Healthcare",
      industry: "Biotechnology",
      $or: [
        {symbol: {$regex: "pharm", $options: "i"}},
        {"long name": {$regex: "pharm", $options: "i"}},
      ],
    }
  },
  {
    "$facet": {
      "page": [
        { "$skip": 20 }, // 3 страница
        { "$limit": 10 }
      ],
      "totalStocks": [
        { "$count": "count" }
      ]
```

```
    }
  }
])
```

Похож на предыдущий запрос, однако возвращается
* вся информация о компании
* только одна страница результата

Модель данных SQL

БД содержит 4 таблицы "companies", "company_officiers", "years", "quarters"
Таблица **companies** содержит информацию о компании

- _id - уникальный идентификатор документа в коллекции
- symbol - тикер акции компании на бирже
- short name - краткое название компании
- long name - полное название компании
- summary - описание компании
- industry - индустрия, к которой относится компания
- sector - сектор (подкатегория для индустрии), к которой относится компания
- address - адрес
- city - город
- state - штат или регион
- country - страна
- phone - телефон компании
- website - ссылка на сайт компании
- total cash - общая сумма денежных средств на счетах компании
- total cash per share - общая сумма денежных средств на акцию
- ebitda - EBITDA (Earnings before interest, taxes, depreciation and amortization)
- total debt - совокупный долг компании
- quick ratio -коэффициент быстрой ликвидности
- current ratio - коэффициент текущей ликвидности
- total revenue - общий доход
- revenue per share- общий доход на акцию
- debt to equity - мультипликатор D/E (Debt to Equity ratio)
- roa - Return on Assets, ROA - рентабельность активов
- roe - Return on Equity, ROE - рентабельность собственного капитала
- employees - количество сотрудников в компании
- financial currency - валюта в которой указаны доходы

Таблица **company_officers** содержит информацию о топ-менеджерах компании

- name - имя
- age - возраст
- title – должность
- year born - год рождения
- total pay - заработная плата за год

Таблица **years** содержит по доходам компании за последние 4 года

- date - указанный год
- revenue - выручка за год
- earnings - прибыль за год

Таблица **quarters** содержит по доходам компании за последние 4 квартала

- date - указанный год
- revenue - выручка за год
- earnings - прибыль за год

Оценка объема данных

В качестве SQL базы данных был выбран Postgres:
Замечание. Для строки длины n <= 126, размер равен n+1 байт * для строки длины n > 126 , размер равен n+4 байт

Объем данных для строки таблицы companies

- symbol - varchar - 4 символов, 5 байт
- short name - varchar - 10 символов, 11 байт
- long name - varchar - 12 символов, 13 байт
- summary - varchar - 1150 символов, 1154 байт
- industry - varchar - 10 символов, 11 байт
- sector - varchar - 10 символов, 11 байт
- address - varchar - 30 символов, 31 байт
- city - varchar - 20 символов, 21 байт
- state - varchar - 2 символа, 3 байт
- country - varchar - 10 символов, 11 байт
- phone - varchar - 12 символов, 13 байт
- website - varchar - 25 символов, 26 байт
- total cash - double - 8 байта
- total cash per share - double - 8 байта
- ebitda - double - 8 байта
- total debt - double - 8 байта
- quick ratio - double - 8 байта
- current ratio - double - 8 байта
- total revenue - double - 8 байта
- revenue per share - double - 8 байта
- debt to equity - double - 8 байта
- roa - Return on Assets, ROA - double - 8 байта
- roe - Return on Equity, ROE - double - 8 байта
- financial currency - varchar - 3 символа, 4 байт
- employees - int - 4 байта

V(companies) = 1406 байта

Объем данных для строки таблицы companyofficers

- name - varchar - 20 символов, 21 байт
- age - int - 4 байта
- title - varchar - 30 символов, 31 байт
- year born - int - 4 байта
- total pay - int - 4 байта

V(companyofficers) = 64 байта

Объем данных для строки таблицы years

- date - varchar - 4 символа, 5 байта
- revenue - выручка - int - 4 байта
- earnings - прибыль - int - 4 байта

V(years) = 13 байта

Объем данных для строки таблицы quarters

- date - varchar - 6 символов, 7 байт
- revenue - выручка - int - 4 байта
- earnings - прибыль - int - 4 байта

V(quarters) = 15 байта
V(всех данных) = 5078 * (V(companies) + 10 * V(companyofficers) + 4 * V(years) + 4 * V(quarters)) = 5078 * (1406 байта + 10 * 64 байта + 4 * 13 байта + 4 * 15 байта) = 5078 * 2158 байт = 10,45 Мб
При увеличении количества объектов каждой сущности рост модели будет происходить линейно. Размер данных n компаний равен n * 2158 байт.
Избыточность модели низка, повторяющиеся значительные по объему фрагменты данных найти не удается. Однако поле `symbols` дублируется во всех четырех таблицах модели данных, т.е. 3 повторения избыточные - избыточность на одну компанию = 5 байт * 3 = 15 байт. Для всей модели избыточность по данному полю равна 15 байт * 5078 = 76170 байт (0.3%).

Запросы к Sql

Информация о компании с тикером

```
SELECT * FROM companies WHERE symbol = 'AAPL';
SELECT * FROM years WHERE symbol = 'AAPL';
SELECT * FROM quarter WHERE symbol = 'AAPL';
SELECT * FROM company_officers WHERE symbol = 'AAPL';
```

В данных 4-х запросах получалась информация о компании с тикером AAPL.

Получение списка стран

```
SELECT DISTINCT country FROM companies;
```

Получение секторов

```
SELECT DISTINCT sector FROM companies;
```

Получение индустрий в секторе

```
SELECT DISTINCT industry FROM companies WHERE sector = 'Healthcare';
```

В данном запросе получался список индустрий в секторе Healthcare.

Подсчет по заданному критерию

Критерием может быть sector, industry или country.

```
SELECT sector, COUNT(*) amount FROM companies GROUP BY sector ORDER BY amount
DESC;
```

В данном запросе считалось количество компаний в каждом из секторов.

Поиск с фильтрацией

Возвращается краткая характеристика компании.

```
SELECT symbol, "short name", industry, sector, country FROM companies WHERE
  country IN ('United States', 'Russia') AND
  sector = 'Healthcare' AND industry = 'Biotechnology' AND
  (symbol ILIKE '%' || 'pharm' || '%' OR "long name" ILIKE '%' || 'pharm' ||
'%');
```

В данном запросе искались компании со следующими характеристиками: * страна United States или Russia * сектор Healthcare * индустрия Biotechnology * тикер или название компании содержит "pharm"

Поиск с фильтрацией с pagination

```
WITH finded AS (
  SELECT * FROM companies WHERE
    country IN ('United States', 'Russia') AND
    sector = 'Healthcare' AND industry = 'Biotechnology' AND
    (symbol ILIKE '%' || 'pharm' || '%' OR "long name" ILIKE '%' || 'pharm' ||
'%')
)
SELECT *, COUNT(*) totalStocks FROM finded GROUP BY GROUPING SETS ((symbol),
())
LIMIT 10 OFFSET 20;
```

Похож на предыдущий запрос, однако возвращается * вся информация о компании * только одна страница результата

Сравнение моделей

Объём данных в нереляционной модели оказался немного больше, чем в реляционной за счет того что был удален ключ ObjectID (MongoDB) и заменен на ключ symbol в SQL (т.к. тикер для каждой компании уникален). Избыточность модели в обоих случаях оказалась низка, а также объем данных растет линейно при увеличении объектов каждой сущности. При переходе к реляционной модели, модель была немного упрощена - сделана "площе" для уменьшения количества таблиц. В случае реляционной модели приходилось делать больше запросов к базе данных, поэтому работать удобнее с нереляционной базой данных.

РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

Краткое описание

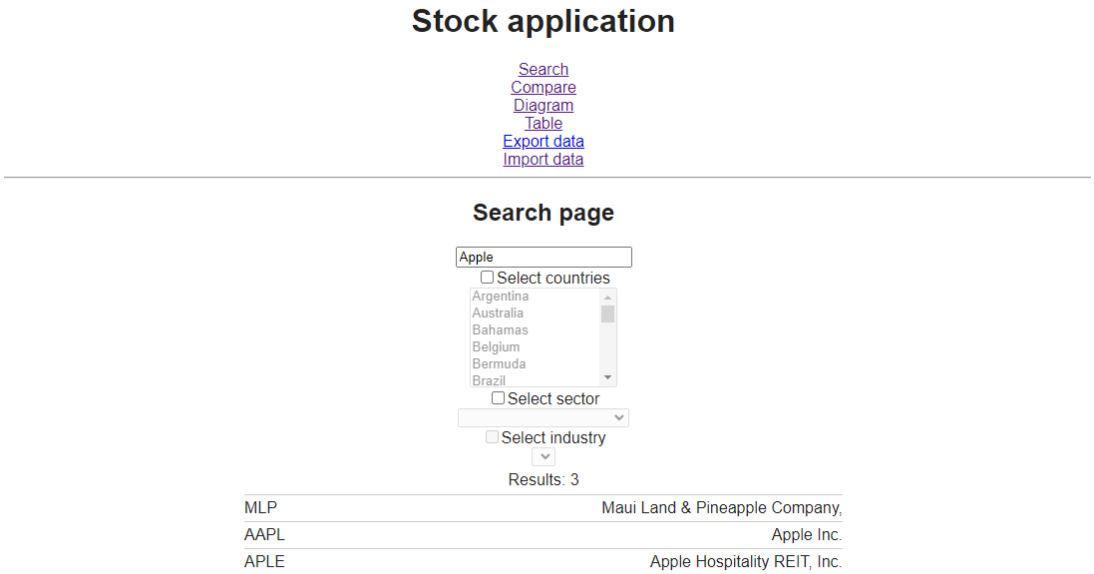
ИС должна предоставлять пользователю финансовые данные компании и позволять сравнивать компании по ряду финансовых показателей. Для сбора данных будет использован Yahoo Finance API.

Функции ИС:

1. Предоставлять пользователю актуальные финансовые показатели компаний
2. Сравнивать компании по финансовым показателям
3. Предоставлять аналитику по компаниям

Схема экранов приложения

1. Экран поиска



Stock application

Search
Compare
Diagram
Table
Export data
Import data

Search page

AAPL

Select countries

Argentina
Australia
Bahamas
Belgium
Bermuda
Brazil

Select sector

Select industry

Results: 1

AAPL	Apple Inc.

Stock application

Search
Compare
Diagram
Table
Export data
Import data

Search page

Select countries

Switzerland
Taiwan
United Arab Emirates
United Kingdom
United States
Uruguay

Select sector

Consumer Cyclical

Select industry

Auto Manufacturers

Results: 12

BLBD	Blue Bird Corporation
GM	General Motors Company
ELMS	Electric Last Mile Solutions, I
AYRO	AYRO, Inc.
GOEV	Canoo Inc.
PTRA	Proterra Inc
FSR	Fisker Inc.
WKHS	Workhorse Group, Inc.
RIDE	Lordstown Motors Corp.
TSLA	Tesla, Inc.
NKLA	Nikola Corporation
F	Ford Motor Company

2. Экран сравнения компаний

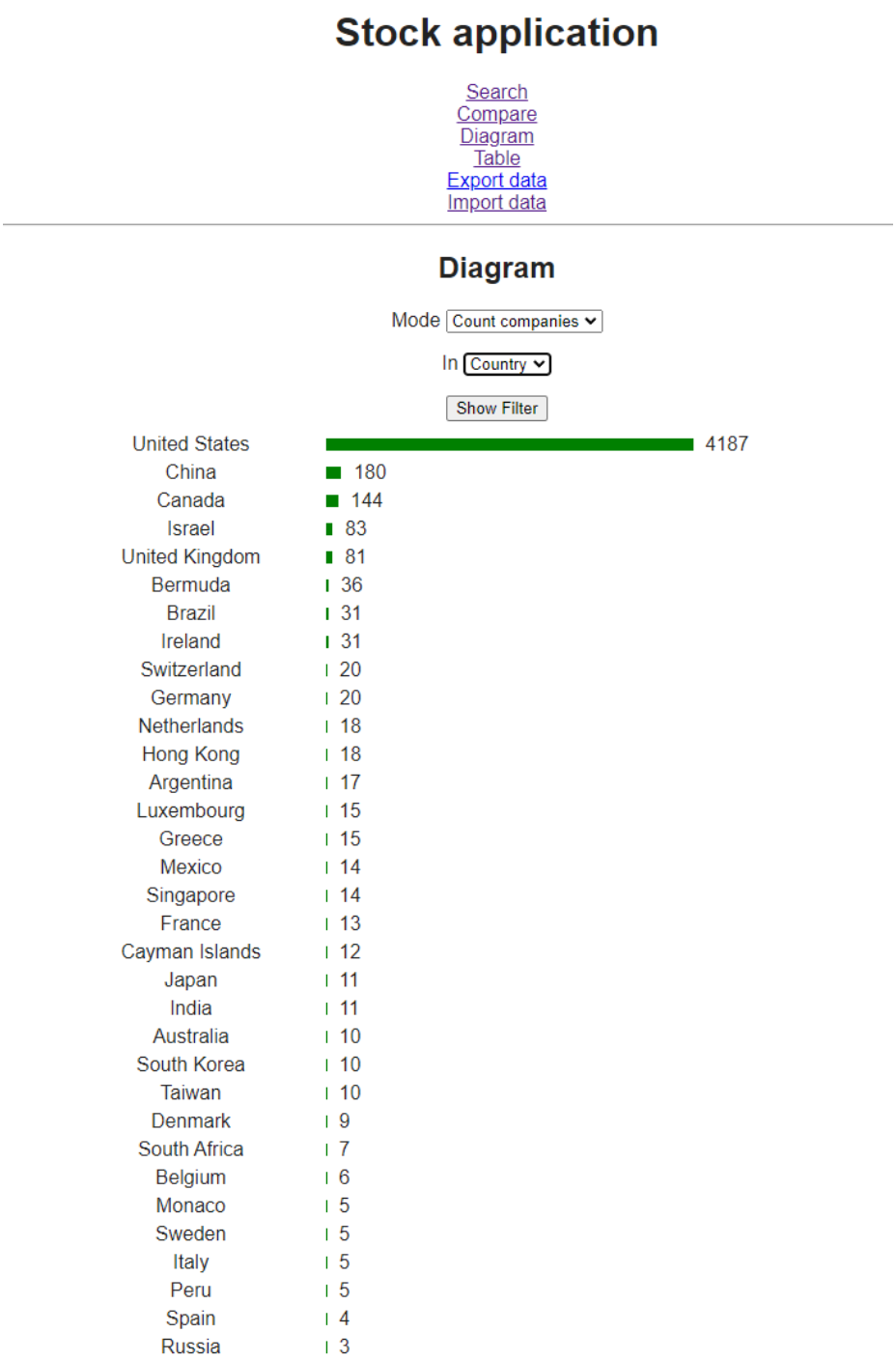
Stock application

Search
Compare
Diagram
Table
Export data
Import data

Comparator

Ticker	F	TSLA
Name	Ford Motor Company	Tesla, Inc.
Sector	Consumer Cyclical	Consumer Cyclical
Industry	Auto Manufacturers	Auto Manufacturers
Country	United States	United States
Total cash	25B USD	17.1B USD
Total cash per share	6.269 USD	17.793 USD
Ebitda	25B USD	17.1B USD
Total debt	148.2B USD	12.5B USD
Quick ratio	1.008	1.281
Current ratio	1.213	1.661
Total revenue	136.4B USD	35.9B USD
Revenue per share	34.267 USD	38.052 USD
Debt to equity	426.133	51.138
Return on assets	0.010980001	0.02994
Return on equity	0.10402001	0.0716

3. Экран для построения диаграмм по компаниям



4. Экран табличного представления данных

Stock application

Search
Compare
Diagram
Table
Export data
Import data

Show Filter Table

Table

Ticker	Name	Sector	Industry	Country
A	Agilent Technologies, Inc.	Healthcare	Diagnostics & Research	United States
AA	Alcoa Corporation	Basic Materials	Aluminum	United States
AACG	ATA Creativity Global	Consumer Defensive	Education & Training Services	China
AAIC	Arlington Asset Investment Corp	Real Estate	REIT—Mortgage	United States
AAL	American Airlines Group, Inc.	Industrials	Airlines	United States
AAME	Atlantic American Corporation	Financial Services	Insurance—Life	United States
AAN	Aarons Holdings Company, Inc.	Consumer Cyclical	Specialty Retail	United States
AAOI	Applied Optoelectronics, Inc.	Technology	Semiconductors	United States
AAON	AAON, Inc.	Industrials	Building Products & Equipment	United States
AAP	Advance Auto Parts Inc Advance	Consumer Cyclical	Specialty Retail	United States
AAPL	Apple Inc.	Technology	Consumer Electronics	United States
AAT	American Assets Trust, Inc.	Real Estate	REIT—Diversified	United States
AAWW	Atlas Air Worldwide Holdings	Industrials	Airports & Air Services	United States
ABB	ABB Ltd	Industrials	Electrical Equipment & Parts	Switzerland
ABBV	AbbVie Inc.	Healthcare	Drug Manufacturers—General	United States
ABC	AmerisourceBergen Corporation	Healthcare	Medical Distribution	United States
ABCB	Ameris Bancorp	Financial Services	Banks—Regional	United States
ABCL	AbCellera Biologics Inc.	Healthcare	Biotechnology	Canada
ABCM	Abcam plc	Healthcare	Biotechnology	United Kingdom
ABEO	Abeona Therapeutics Inc.	Healthcare	Biotechnology	United States
ABEV	Ambev S.A.	Consumer Defensive	Beverages—Brewers	Brazil
ABG	Asbury Automotive Group Inc	Consumer Cyclical	Auto & Truck Dealerships	United States
ABIO	ARCA biopharma, Inc.	Healthcare	Biotechnology	United States
ABM	ABM Industries Incorporated	Industrials	Specialty Business Services	United States
ABMD	ABIOMED, Inc.	Healthcare	Medical Devices	United States
ABNB	Airbnb, Inc.	Communication Services	Internet Content & Information	United States
ABOS	Acumen Pharmaceuticals, Inc.	Healthcare	Biotechnology	United States
ABR	Arbor Realty Trust	Real Estate	REIT—Mortgage	United States
ABSI	Absci Corporation	Healthcare	Biotechnology	United States
ABT	Abbott Laboratories	Healthcare	Medical Devices	United States
ABTX	Allegiance Bancshares, Inc.	Financial Services	Banks—Regional	United States
ABUS	Arbutus Biopharma Corporation	Healthcare	Biotechnology	United States
ABVC	ABVC Biopharma, Inc.	Healthcare	Biotechnology	United States
ACA	Arcosa, Inc.	Industrials	Infrastructure Operations	United States
ACAD	ACADIA Pharmaceuticals Inc.	Healthcare	Biotechnology	United States
ACBI	Atlantic Capital Bancshares, In	Financial Services	Banks—Regional	United States
ACC	American Campus Communities Inc	Real Estate	REIT—Residential	United States
ACCD	Accolade, Inc.	Healthcare	Health Information Services	United States
ACCO	Acco Brands Corporation	Industrials	Business Equipment & Supplies	United States
ACEL	Accel Entertainment, Inc.	Consumer Cyclical	Gambling	United States

Next Page = 1; TotalPages = 127

Использованные технологии

Vue.js, Golang, MongoDB

Go – библиотеки

github.com/caarlos0/env/v6	v6.6.2
github.com/go-chi/chi/v5	v5.0.5-0.20210830173112-df44563f0692
github.com/go-chi/cors	v1.2.0
github.com/gobeam/mongo-go-pagination	v0.0.7
go.mongodb.org/mongo-driver	v1.7.1

ВЫВОДЫ

Достигнутые результаты

Разработанная ИС предоставляет пользователю финансовые данные компании и позволяет сравнивать компании по ряду финансовых показателей, отображать данные в табличном представлении с заданием фильтром, сторить диаграммы по различным данным с заданием фильтров, экспортировать и ипортировать данные.

Недостатки и пути для улучшения полученного решения

Недостатком является отсутствие кеширования для запросов на стороне сервера. Добовления кешировния при использовании Redis снизет нагрузку на сервер базы данных MongoDB.

Будущее развитие решения

Возможными будущими решениями явлвяется добавления кеширования и добавления работы со сторонним api, возращающие актуальные курсы валют, для реализации сравннения данных в разных валютах.

ДОКУМЕНТАЦИЯ ПО СБОРКЕ И РАЗВЕРТЫВАНИЮ ПРИЛОЖЕНИЯ

Инструкция по сборке бекенда:

```
go build nosql1h21-stock-backend/backend/cmd/main.go
```

Инструкция по сборке фронтенда:

В директории frontend:

```
npm run serve
```

Инструкция для docker-compose

1) Run docker-compose up --build being in the repository root
2) Wait until the getting stocks info is completed (about 1 minute)
3) Go to http://localhost:8080

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Документация по MongoDB https://docs.mongodb.com/manual
2) Документация по Vue https://ru.vuejs.org/v2/guide
3) Mongo go driver https://github.com/mongodb/mongo-go-driver