

Темы для ИДЗ

Из чего складывается сложность:

- 1. Доступность датасета
- 2. Необходимость использовать несколько датасетов
- 3. Необходимость использовать СУБД не совсем по назначению
- 4. Алгоритмы

Темы в порядке возрастания субъективной сложности. Поле “Комментарий” содержит отправную точку для формулировки окончательного задания.

№	Название	Предлагаемая СУБД	Набор данных/API	Комментарий
24	Сервис для дизайна мозаек	Neo4j	Синтетические данные	Идея - вам очень захотелось превратить любимую картинку в мозаику из керамической плитки у себя на даче, для этого вам необходимо как-то прикинуть из каких частей оно будет состоять, сколько и какой плитки вам нужно закупить (по цвету и площади), как эту плитку раскроить и как ее наклеить (в каком порядке). Задача в рамках ИДЗ - сделать систему для автоматического (можно придумать свое, можно уже готовое https://github.com/nuno-faria/tiler) построения керамических мозаек из изображений, а также для генерации инструкций по сборке и расчета количества цветных плиток. Необходимые (но не достаточные) фичи - список готовых мозаик, загрузка и обработка нового изображения, визуализация в виде графа, анализ и сводка по мозаике (сколько и каких плиток нужно, сколько метров реза вам нужно проделать), инструкция по нарезке, инструкция по сборке.
24	Построение графа ссылок на веб-страницах	Neo4j	Синтетические данные	Задача в рамках ИДЗ - сделать систему обхода всех веб-страниц и ссылок на сайте для составления графа и его анализа. Необходимые (но не достаточные) фичи - список обработанных сайтов, страница визуализации графа ссылок, анализ графа (изолированные компоненты, длина пути от главной страницы).
24	Карта транспортной доступности водоемов севера Ленинградской области	Neo4j	OSM	Задача в рамках ИДЗ - сделать систему для построения и анализа графа транспортной доступности водоемов (реки, озера, водохранилища) севера ЛО. Необходимые (но не достаточные) фичи - список водоемов, страница визуализации графа , анализ графа (сортировка по доступности, изолированные узлы).
24	сервис для планирования кабельных сетей	Neo4j	Синтетические данные	Задача в рамках ИДЗ - сделать систему для проектирования и анализа кабельной инфраструктуры зданий (локальные сети). Необходимые (но не достаточные) фичи - список проектов, сводка по проекту, поэтажные планы, анализ проекта (сколько нужно кабелей, оборудования, расходников для монтажа), визуализация проекта в виде графа.
24	Каталог учебных планов и программ	Neo4j	https://etu.ru/sveden/education/eduop/	Задача - сделать сервис для управления и визуализацией (в виде графа) учебных программ и планов в рамках ВУЗа. Пользователи - преподаватели, студенты. Необходимые (но не достаточные фичи) - страница учебной программы, учебной дисциплины, статистика (для админов, по отдельным серверам транскрипции, по пользователям, по задачам), интерфейсы оплаты на заглушках. Саму транскрипцию делать не обязательно, но если очень хочется - можно использовать whisper.ai
24	Сервис транскрипции видео и аудио	MongoDB	Синтетические данные	Задача - сделать сервис для организации транскрипции видео и аудиоматериалов с помощью некоторой распределенной инфраструктуры (это означает, что ресурсы для транскрипции определяются динамически (администраторами) и система должна уметь планировать обработку входных данных исходя из текущих мощностей). Пользователи - администраторы, пользователи. Необходимые (но не достаточные фичи) - страница отдельного задания, профили пользователей, статистика (для админов, по отдельным серверам транскрипции, по пользователям, по задачам), интерфейсы оплаты на заглушках. Саму транскрипцию делать не обязательно, но если очень хочется - можно использовать whisper.ai
24	Хаб данных умной фермы	InfluxDB + Mongo	Синтетические данные	Задача - сделать приложение для автоматизированного мониторинга тепловой фермы овощей. Пользователи - рабочие, бригады, владельцы бизнеса. Необходимые (но не достаточные фичи) - страницы пользователей, страница для работы с нарядами рабочих и бригадиров, склад (собранный продукция, расходники), инциденты, график дежурств, инфраструктура
24	Информационная система сети прачечных	Neo4j	Синтетические данные	Задача - сделать систему для управления сетью максимально автоматизированных прачечных. Пользователи - администратор (reception), директор филиала, владелец бизнеса. Необходимые (но не достаточные) фичи - график работы, страница филиала, склад, расчет ЗП, расчет прибыли по филиалам
24	Информационная система стоматологической клиники	MongoDB	Синтетические данные	Задача - сделать сервис для управления стоматологической клиникой. Пользователи - администраторы (те, которые сидят на ресепшн), врачи, пациенты. Необходимые (но не достаточные фичи) - карты пациентов, статистика для бухгалтерии, статистика и динамика больных, хранение снимков, склад и обслуживание оборудования
24	Составление маршрутов интересных пеших прогулок по СПб	Neo4j	Wikimapia + OSM	Задача - сделать сервис для составления пеших маршрутов по Санкт-Петербургу. Пользователи - администраторы, пользователи. Необходимые (но не достаточные фичи) - личные страницы, страницы маршрутов, страница генерации маршрута (задаем настройки - интересы, протяженность - учитываем время года, ширину тротуаров)
24	Визуализатор и анализатор электронной почты в виде графа	Neo4j	Синтетические данные	Задача - сделать сервис, который позволяет визуализировать и анализировать электронную почту с помощью графов. Вершины - письма и авторы, ребра - хронология (цепочки писем), связи между авторами (работа).

№	Название	Предлагаемая СУБД	Набор данных/API	Комментарий
24	Сервис климат-контроля	InfluxDB	Синтетические данные	Задача - сделать сервис для сбора данных о климате в протяженном помещении, управлении климатической техникой, мониторинге ее состояния. Необходимые (но не достаточные фичи) - визуализация состояния помещений, страницы приборов, настройки стратегий работы приборов, уведомления о расходах
24	Сервис поддержки ЖКХ	Neo4j	OSM	Задача - сделать сервис фиксации проблем с ЖКХ инфраструктурой (стрижка газонов, неполадки в трубах, покраска ...), планирования регулярных и внеплановых работ, формирования ответов на обращения собственников. Необходимые (но не достаточные фичи) - аккаунты собственников и ЖКХ-администраторов, визуализация на карте, подача и дискуссии по обращениям
24	Онлайн-магазин цветов и комнатных растений	Neo4j	Синтетические данные	Задача - сделать сервис для продажи цветов и комнатных растений, который бы учитывал малый срок жизни цветов. Пользователи - продавцы, администраторы, покупатели. Необходимые (но не достаточные фичи) - аккаунты пользователей, страницы товаров, страница "Склад", статистика
24	Умный склад на стройплощадке	MongoDB	Синтетические данные	Задача - сделать сервис для учета, отпуска и поступления материалов, нужных для строительства дома. Пользователи - рабочие, прорабы, кладовщики. Необходимые (но не достаточные фичи) - аккаунты пользователей, страницы позиций, страница "Склад", статистика, страница для формирования накладных, страница для анализа потребления и прогноза позиций.
24	Сервис составления генеалогических деревьев	Neo4j	Синтетические данные	Задача - сервис, позволяющий различным пользователям вести свои родословные деревья, анализировать их, находить похожие, предлагать мержи деревьев. Необходимые (но не достаточные фичи) - аккаунты пользователей, страница родословного дерева, страница рекомендаций, предоставления доступа к дереву, страница мержа
24	Каталог объявлений о продаже автомобилей	Mongo	Синтетические данные	Задача - сервис, позволяющий размещать объявления о покупке или продаже автомобилей, связывать продавцов и покупателей. Необходимые (но не достаточные фичи) - аккаунты продавцов и покупателей, рейтинги и отзывы, страница подробных данных об автомобиле (пробег, номера двигателя ... , марка машины, год, фото). Пользователи - администраторы, клиенты.
24	Сервис редактирования и автоматического составления расписания в ВУЗе	Mongo	Расписание ЛЭТИ	Задача - сервис, позволяющий принимать пожелания от преподавателей, принимать данные об аудиториях и их вместимости, данные учебных планов (предметы / курсы / группы), данные о студентах (сколько в какой группе человек, какие группы к каким направлениям относятся. Необходимые (но не достаточные фичи) - автоматическое составление расписания (исходя из предоставленных ограничений), визуализация и редактирование расписания, логика согласования расписания (препод высказал пожелание, админ поправил, препод одобрил), анализ составленного расписания (насколько равномерно используются аудитории, насколько студентам / преподам нужно между парами бегать в новую аудиторию). Пользователи - преподаватели, администраторы, студенты.
24	Сервис оценки земельных участков	Neo4j	Avito + OSM	Задача - подготовить веб-сервис, который реализует каталог участков, а также механику их оценки. Смысл сервиса - позволить добавлять в систему полные данные об участках координаты всех узловых точек и проводить на базе формы, окружающего ландшафта и соседей оценку, например: сколько соседских домов, как далеко водоем / остановка / магазин, насколько участок ниже / выше соседей. Необходимые (но не достаточные) фичи - личные страницы пользователей, рейтинги и отзывы пользователей и участков, визуализация и поиск на карте, страницы отдельных участков с подробными данными.
24	Сервис "Crowd-редакция"	Neo4j	Синтетические данные + wikibooks	Задача - подготовить веб-сервис, который реализует вычитку и редактирование книг силами сообщества (crowdsourcing). Одни и те же пользователи могут быть как авторами, так и редакторами. Необходимые (но не достаточные) фичи - личные страницы пользователей, рейтинги и отзывы пользователей, статистика рецензий, списки рецензий, страницы отдельных рецензий, страницы книг.
24	Сервис поиска врачей частной практики	MongoDB	Синтетические данные	Задача - подготовить веб-сервис, которые сводит пациентов и врачей, ведущих частную практику. Категории пользователей - врачи, пациенты, админы. Необходимые (но не достаточные) фичи - личные страницы врачей и пациентов, рейтинги и отзывы, статистика, страницы для финансовой стороны вопроса (реальные платежи подключать не нужно) - оплаты и статистика, карточки и анализы пациентов.
24	Telegram-бот для вывода данных из таблиц	MongoDB	Синтетические данные	Задача - сделать простого бота, который сможет решать задачу "покажи мне строку по запросу А из таблицы Б". Для определенности можно считать, что такой бот нужен для курсов МОЭВМ, чтобы скрывать от студентов всю таблицу успеваемости и показывать студенту только его собственную строчку. Таблиц / студентов / преподав может быть много.
24	Telegram-бот для напоминаний преподавателям о новых записях в таблице	MongoDB	Синтетические данные	Задача - сделать простого бота, который сможет решать задачу "пришли пользователю уведомление, если его _неким_ образом упомянули в таблице". Для определенности можно считать, что такой бот нужен для преподавателей МОЭВМ, которые ждут появления студенческих отправок в определенной таблице. Преподав и таблиц может быть много.
24	Сервис хранения данных биржевой торговли forex	MongoDB	Синтетические данные	Задача - веб приложение, которое будет хранить / обеспечивать поиск / анализ и визуализацию данных о рынке по указанным периодам / запросам / валютным парам.
24	Сервис сбора и визуализации логов Apache2	InfluxDB	Синтетические данные	Задача - создать приложение, которое агрегирует логи Apache2 в influx. https://github.com/influxdata/telegraf/blob/master/plugins/inputs/tail/README.md , https://github.com/influxdata/telegraf/blob/master/plugins/parsers/grok/README.md . Необходимо поддерживать одновременно все файлы логов apache - access.log, error.log, other_vhosts_access.log, а также время загрузки страницы как один из элементов данных.
24	Сервис локального (на базе нескольких гитхаб-репо) антиплагиата для текстов на естественном языке	MongoDB	Синтетические данные	Задача - веб приложение, которое будет выгружать, версионировать (Используя данные гита) данные из набора гитхаб-репо (кастомизируются пользователями), хранит эти данные для удобства синтаксического анализа и формирует отчеты по возможному плагиату между пользователями репо.
24	Сервис хранения экспериментов инструмента Sumo	Neo4j	https://sumo.dlr.de/docs/index.html	Задача - сделать простое веб-приложение для импорта / хранения / поиска / визуализации результатов экспериментов в эмуляторе.

№	Название	Предлагаемая СУБД	Набор данных/API	Комментарий
24	Сервис хранения экспериментов инструмента pybullet-drones	MongoDB	https://github.com/utiasDSL/gym-pybullet-drones	Задача - сделать простое веб-приложение для импорта / хранения / поиска / визуализации результатов экспериментов в эмуляторе.
24	Сервис хранения экспериментов инструмента gipes	MongoDB	https://github.com/mortbopet/Ripes	Задача - сделать простое веб-приложение для импорта / хранения / поиска / визуализации результатов экспериментов в эмуляторе.
24	Построение маршрутов по рекам и каналам	Neo4j	OpenStreetMaps	Задача - максимально смоделировать данные о реках / каналах СПб, а также объектов инфраструктуры (причалы, пирсы, знаки для водного транспорта, мосты), обеспечить построение маршрутов и вычисление статистики
25	Юридические документы	Neo4j	Синтетические данные	Смысл проекта - предоставить интерфейс для хранения, версионирования и анализа (грубого) юридических документов на русском языке. Варианты анализа: разбиение на предложения, абзацы, идентификация пунктов по ключевым словам, поиск формулировок-обязательств / требований
27	ИС Справочник Музеев	Memcached	https://opendata.mkrf.ru/opendata/7705851331-museums	Тема аналогична "(Сложно) ИС Справочник образовательных организаций Санкт-Петербурга"
28	ИС Справочник образовательных организаций Санкт-Петербурга	Memcached	https://petersburgedu.ru/institution/	Цель - создать каталог, хранящий максимум информации обо всех организациях. Задачи - хранение всех доступных полей, организация поиска, агрегации, изменения данных каталога. Модель данных нужно построить так, чтобы на клиенте требовалась минимальная обработка (скорее всего, вам предстоит сделать очень много дублирования данных)
30	Построение маршрутов с учетом плохой парковки	Neo4j	Открытые данные OpenStreetMap	Необходимо взять в качестве демо данных квадрат карты, где находится Петроградский район и для него создать приложение, которое позволит строить маршруты из точки А в точку Б так, чтобы 1) при подъезде к точке Б нужно вести водителя не наискратчайшим путем, а кругами вдоль перспективных для парковки мест 2) при всем этом поездка по маршруту не была супердолгой

Своя тема

1. **Интересный, содержательный и нужный** пользователям сценарий использования
2. Должен быть понятный, достаточный и доступный **источник данных** - публичный датасет, публичное API, доступ к хранилищу данных
3. Должно присутствовать решение задач **хранения/анализа/представления/агрегации/импорта и экспорта** данных
4. Дополнительный плюс, если есть **связь с существующим проектом** и/или бакалаврской работой
5. Информация об **аналогах**

Источники данных

Помимо ранее упомянутых датасетов, можно также использовать для своей темы:

1. [Список SPARQL-endpoint](#)
2. <https://github.com/blog/2298-github-data-ready-for-you-to-explore>
3. <http://www.foodpedia.tk/about?lang=ru>
4. <http://www.imdb.com/interfaces>
5. https://www.wikidata.org/wiki/Wikidata:Main_Page
6. Выгрузка данных OSM
 1. <https://www.openstreetmap.org/search?query=saint-petersburg#map=10/59.9471/30.2495>
7. [Kaggle](#) - каталог наборов данных
8. [Datahub](#) - каталог наборов данных

From:
<http://se.moevm.info/> - **se.moevm.info**

Permanent link:
http://se.moevm.info/doku.php/staff:courses:no_sql_introduction:course_work:topics?rev=1693470339

Last update: **2023/08/31 09:25**

