# **Данные для задания**

Оглавление

[Данные для задания 1](#_Toc194335690)

[1. Задача 1](#_Toc194335691)

[Разработать поисковой модуль на базе LLM для интеграции его в систему VK Tek. 1](#_Toc194335692)

[Модуль должен уметь самостоятельного изучать и анализировать источников информации в корпоративном портале и представления полноценного ответа на запрос пользователя, с указанием источников. В качестве источника информации мы предоставляем дамп базы данных сервисов, в которых будет храниться нужная для обработки информация. 1](#_Toc194335693)

[2. Описание размещенной информации на портале 1](#_Toc194335694)

[3. Описание баз данных для задания 4](#_Toc194335695)

[**3.1.** **Как восстановить базу данных** 6](#_Toc194335696)

[**3.2.** **Как забрать файлы из хранилища** 7](#_Toc194335697)

## **Задача**

## Разработать поисковой модуль на базе LLM для интеграции его в систему VK Tek.

## Модуль должен уметь самостоятельного изучать и анализировать источников информации в корпоративном портале и представления полноценного ответа на запрос пользователя, с указанием источников. В качестве источника информации мы предоставляем дамп базы данных сервисов, в которых будет храниться нужная для обработки информация (страницы, списки с контентов, а также файлы различных форматов).

## **Описание размещенной информации на портале**

В портале с помощью сервисе Конструктор сайтов (описание работы сервиса см [тут](https://people-hub.ru/documentation/guides/sitector/sities/)), были созданы сайты в которых представлены некие базы знаний, см. Таблица 1.

Таблица 1 Список сайтов

| **Название сайта** | **Содержит** | **Описание** |
| --- | --- | --- |
| People hub инструкции | * Информационные страницы | Страницы содержат контент в виде текста, таблиц и изображений. |
| People hub архитектура | * Информационные страницы | Страницы содержат контент в виде текста, таблиц и изображений. |
| Информация о Хакатоне | * Информационные страница * Список | Страницы содержат контент в виде текста, таблиц и изображений.  Ячейки в списках содержат данные с типом текст. |
| Работы Фролова | * Информационные страница * Файловое хранилище | Страница содержат контент в виде текста.  Файловое хранилище содержит следующие файлы:   * Фролов\_К\_В\_Горные\_машины\_МЭ,\_том\_IV\_24\_2010.pdf * Фролов\_К\_В\_Двигатели\_внутреннего\_сгорания\_МЭ,\_том\_IV\_14\_2013.pdf * Фролов\_К\_В\_Авиационные\_двигатели\_МЭ,\_том\_IV\_21,\_книга\_3\_2010.pdf * Фролов\_К\_В\_Динамика\_и\_прочность\_машин\_МЭ,\_том\_I\_3,\_книга\_1\_1994.pdf |
| Музеи | * Информационные страница * Файловое хранилище | Страница содержат контент в виде текста.  Файловое хранилище содержит следующие файлы:   * 05\_Великие\_музеи\_мира\_Метрополитен\_2011.pdf * 06\_Великие\_музеи\_мира\_Эрмитаж\_Часть\_1\_2011.pdf * 04\_Великие\_музеи\_мира\_Египетский\_музей\_2011.pdf * 02\_Великие\_музеи\_мира\_Прадо\_Мадрид\_2011.pdf * 03\_Великие\_музеи\_мира\_Лувр\_Париж\_2011.pdf |
| Литература | * Информационные страница * Файловое хранилище | Страница содержат контент в виде текста.  Файловое хранилище содержит следующие файлы:   * Хладнокровное-убийство.doc * Завтрак-у-Тиффани.doc * Приваловские-миллионы.doc * Улисс.doc |
| Таблицы | * Информационные страница * Файловое хранилище | Страница содержат контент в виде текста.  Файловое хранилище содержит следующие файлы Excel с различной инфомацией:   * MOCK\_DATA.xlsx * MOCK\_DATA-2.xlsx * MOCK\_DATA-3.xlsx * MOCK\_DATA-4.xlsx * MOCK\_DATA-5.xlsx * MOCK\_DATA-6.xlsx * MOCK\_DATA-7.xlsx * MOCK\_DATA-8.xlsx * MOCK\_DATA-9.xlsx * MOCK\_DATA-7.xlsx |

## **Описание баз данных для задания**

Ниже в Таблица 2, представлены названия баз данных в разрезе сервисов,

Таблица 2 Дампы баз данных

|  |  |
| --- | --- |
| **Сервис** | **Дамп** |
| Конструктор сайтов | cms.dump |
| Конструктор сайтов - списки | lists.dump |
| Файловое хранилище | filestorage.dump |

### **Описание базы данных**

|  |
| --- |
| 2. **# 📦 Сервис "Конструктор сайтов"** 3. **## Основные таблицы** 4. **### 🔹 Таблица `sites\_site`** 5. Представляет собой структуру для хранения информации о сайте внутри системы. Может быть как самостоятельным сайтом, так и вложенным (дочерним сайтом или разделом). 6. **#### Поля:** 7. - `id` (`UUID`, primary key) – Уникальный идентификатор сайта, генерируется автоматически. Строковый формат. 8. - `name` (`String`, unique) – Уникальное название сайта. Обязательное. 9. - `description` (`String`, default="") – Описание сайта. Необязательное текстовое поле. 10. - `slug` (`String`, unique) – Уникальный slug (человеко-читаемый идентификатор), используется, например, в URL. 11. - `status` (`String`, default=SiteStatus.draft, indexed) – Статус сайта (enum): 12. ```python 13. class SiteStatus(str, Enum): 14. draft = "draft" 15. published = "published" 16. archived = "archived" 17. ``` 18. - `is\_navigation\_visible` (`Boolean`, default=True) – Показывается ли сайт в навигации. 19. - `filestorage\_root\_folder\_id` (`UUID`, nullable) – Идентификатор корневой папки файлов (может быть NULL). 20. - `parent\_id` (`UUID`, nullable) – FK на родительский сайт (ON DELETE SET NULL). 21. - `created\_by\_id` (`UUID`, nullable) – FK на пользователя, создавшего сайт. 22. - `updated\_by\_id` (`UUID`, nullable) – FK на пользователя, который последним редактировал сайт. 24. **#### Особенности:** 25. - *\*\****\*\*Иерархия:\*\***\*\* сайты могут быть вложенными, строя древовидные структуры. 26. - *\*\****\*\*Гибкие связи:\*\***\*\* через ServiceObject и Page можно связать с контентом, меню и т.п. 27. --- 28. **### 🔹 Таблица `pages\_page`** 29. Представляет сущность страницы, используемую для хранения контента, метаданных, связей и полнотекстового поиска. 30. **#### ВАЖНО:** 31. Принадлежность страницы к сайту определяется через таблицу `sites\_serviceobject`, где: 32. - `ServiceObject.external\_id` = `id страницы` 33. - `ServiceObject.site\_id` = `id сайта` 35. **##### Пример SQL-запроса: найти страницы сайта с `id = 45f1ce96-405a-439a-ad12-92e381e34835`** 36. ```sql 37. SELECT \* 38. FROM pages\_page AS pp 39. LEFT JOIN sites\_serviceobject AS sso ON sso.external\_id = pp.id::TEXT 40. WHERE sso.site\_id = '45f1ce96-405a-439a-ad12-92e381e34835'; 41. ``` 42. **#### Поля:** 43. - `id` (`UUID`) – Первичный ключ 44. - `name` (`Text`) – Название страницы 45. - `slug` (`Text`, unique) – Слаг URL 46. - `body` (`JSON`) – Контент страницы 47. - `views\_count` (`Integer`) – Кол-во просмотров 48. - `status` (`PageStatus`, enum) – Статус публикации 49. - `is\_main` (`Boolean`) – Основная страница сайта 50. - `settings` (`JSONB`) – Настройки отображения страницы 51. **#### Метаданные:** 52. - `created\_at` (`DateTime`) – Дата создания 53. - `created\_by\_id` (`UUID`) – Пользователь-создатель 54. - `updated\_at` (`DateTime`) – Дата обновления 55. - `updated\_by\_id` (`UUID`) – Последний редактор 56. **#### Иерархия и связи:** 57. - `parent\_id` – FK на родительскую страницу 58. --- 59. **# 🗂 Сервис "Файловое хранилище"** 60. **# Таблица `StorageObject`** 61. `StorageObject` — это основная модель файлового хранилища, представляющая как файлы, так и папки, с поддержкой версионирования, иерархии, прав доступа, контекста, избранного и т.д. 62. **## 📌 Общие поля** 63. - `id (UUIDField)` — Уникальный идентификатор объекта, используется UUID v4. 64. - `type (PositiveSmallIntegerField)` — Тип объекта (`Файл` или `Папка`), определяется через enum `StorageObjectType`. 65. - `name (TextField)` — Название объекта. 66. - `description (CharField, max\_length=255)` — Описание (необязательное поле). 67. - `deprecated\_tags (ArrayField)` — Массив строк-тегов, устаревшая функциональность. 68. - `categories (ManyToManyField)` — Категории, к которым относится объект. 69. **## 📁 Структура и контекст** 70. - `parent (ForeignKey)` — Родительский объект (`StorageObject`), используется для построения иерархии. 71. - `context\_folder (ForeignKey)` — Папка, определяющая контекст (например, `Blog\_{id}`). 72. - `file\_storage\_display\_type (CharField)` — Тип отображения (`tile`, `list`, и т.д.). 73. **## ⚙️ Метаданные** 74. - `size (PositiveIntegerField)` — Размер файла (в байтах), если это файл. 75. - `additional (JSONField)` — Произвольные дополнительные данные, включая `context`. 76. **## 👤 Аудит** 77. - `created\_by`, `updated\_by`, `deleted\_by (ForeignKey → User)` — Пользователи, выполнявшие действия. 78. - `created\_at`, `updated\_at`, `deleted\_at (DateTimeField)` — Соответствующие временные метки. 79. **## 🔄 Версионность** 80. - `version (OneToOneField)` — Связь с последней версией файла (`Version`), если применимо. 81. **\*\*ДОПОЛНИТЕЛЬНО\*\*** 82. Основные таблицы, которые нужны — `storage\_storageobject`, `storage\_version` содержат информацию о файлах и папках. версиях файлов. 83. `storage\_storageobject` - это обьект файлового хранилища. Это может быть как папка, так и файл. Что это за объект можно определить по полю `type`, которое может быть `file`, `folder` (1, 0). 84. ```python 85. class StorageObjectType(models.IntegerChoices): 86. FOLDER\_TYPE = 0, 'Папка' 87. FILE\_TYPE = 1, 'Файл' 88. ``` 89. В таблице `storage\_version` поле `link` содержит ссылку на сам файл. Учитывайте, что может быть несколько версий — используйте последнюю. 90. **\*\*ВАЖНО\*\*** 91. Как определить файл в папке? `storage\_storageobject` с типом `file`(1) у которого в parent\_id лежит обьект с типом `folder`(0). Это значит что данный файл находится в папке, указанной в parent\_id. 92. Как определить принадлежность папки к сайту/странице? У сайта в базе данных есть поле - `filestorage\_root\_folder\_id`. Это рутовая папка данного сайта. К этой папке крепятся все файловые хранилища сайта. Также у обьекта файлового хранилища есть поле site\_id по которому можно определить принадлежность обьекта в сайта. 93. Например, вот пример запроса для получения всех файловых хранилищ сайта c `id='45f1ce96-405a-439a-ad12-92e381e34835'`: 94. ```sql 95. SELECT child.\* 96. FROM storage\_storageobject AS child 97. INNER JOIN storage\_storageobject AS parent ON child.parent\_id = parent.id 98. WHERE parent.site\_id = '45f1ce96-405a-439a-ad12-92e381e34835' 99. AND parent.type = 0 100. AND child.type = 0; 101. ``` 102. Получение всех файлов cайта с `id='45f1ce96-405a-439a-ad12-92e381e34835'`: 103. ```sql 104. SELECT \* 105. FROM storage\_storageobject AS so 106. WHERE so.site\_id = '45f1ce96-405a-439a-ad12-92e381e34835' 107. AND so.type = 1; 108. ``` 109. **\*\*Важно\*\*** 110. `storage\_storageobject` - представляет собой иерархию, поэтому нужно проходится рекурсивно вниз/вверх чтобы получить все объекты(подпапки/файлы в подпапках) 111. Чтобы получить ссылку на файл, нужно приджойнить таблицу `storage\_version` и получить поле `link` 113. --- 114. **# 📋 Сервис "Списки"** 115. **### 🔹 Таблица `lists\_list\_row`** 116. Хранит строки списков и данные о создателе и редакторе. 117. - `created\_by\_id` (`UUID`) – FK на `users\_user.keycloak\_id`, может быть NULL 118. - `updated\_by\_id` (`UUID`) – FK на `users\_user.keycloak\_id`, может быть NULL 119. - `created\_at` – Дата создания 120. - `updated\_at` – Дата обновления 122. **### 🔹 Таблица `lists\_list\_column`** 123. Содержит столбцы списка, включая историю состояний. 124. - `is\_visible` (`Boolean`) – Флаг видимости (default = True) 125. - `state\_history` (`JSONB`) – История состояний (может быть NULL) 126. --- 127. **### 🔹 Таблица `lists\_files\_file`** 128. Хранит информацию о загруженных файлах. 130. - `id` – Primary key (автоинкремент) 131. - `name` – Название файла (обязательное) 132. - `storage\_object\_id` (`UUID`) – Уникальный ID объекта хранения (обязателен) 133. - `file\_data` (`JSONB`) – Метаданные (опционально) 134. - `created\_at` – Дата создания (UTC, default = now) 135. - `updated\_at` – Дата обновления (авто) 136. > Таблица поддерживает расширение структуры благодаря JSONB. |

### **Как восстановить базу данных**

Чтобы восстановить базы данных нужно использовать команду ниже, вместо PG\_\* необходимо подставить значения для своего Postgres. Postgres нужен 17й версии.

|  |
| --- |
| pg\_restore --create --dbname "postgresql://PG\_USER:PG\_PASSWORD@PG\_HOST:PG\_PORT/PG\_DB" --no-owner --format c --no-privileges filestorage.dump |

### **Как забрать файлы из хранилища**

Чтобы, забрать файлы, нужно к link из БД filestorage добавить id файла.

Link где хранятся файлы - https://hackaton.hb.ru-msk.vkcloud-storage.ru/media/

Пример id - /attachments/68594c06-0bc6-11f0-b3ed-1e065ec9454b/MOCK\_DATA-6.xlsx

Файл будет доступен по следующей ссылке:

https://hackaton.hb.ru-msk.vkcloud-storage.ru/media/attachments/68594c06-0bc6-11f0-b3ed-1e065ec9454b/MOCK\_DATA-6.xlsx