

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Компьютерные науки и прикладная математика»
Кафедра №806 «Вычислительная математика и программирование»**

**Курсовая работа по курсу
«Базы данных»**

**Разработка клиентского приложения
«Сервис генеративных сетей»**

**Студент: Головенко А.В.
Группа: М8О–312Б–22
Преподаватель: А.В. Малахов**

Москва, 2024.

1. Схема базы данных

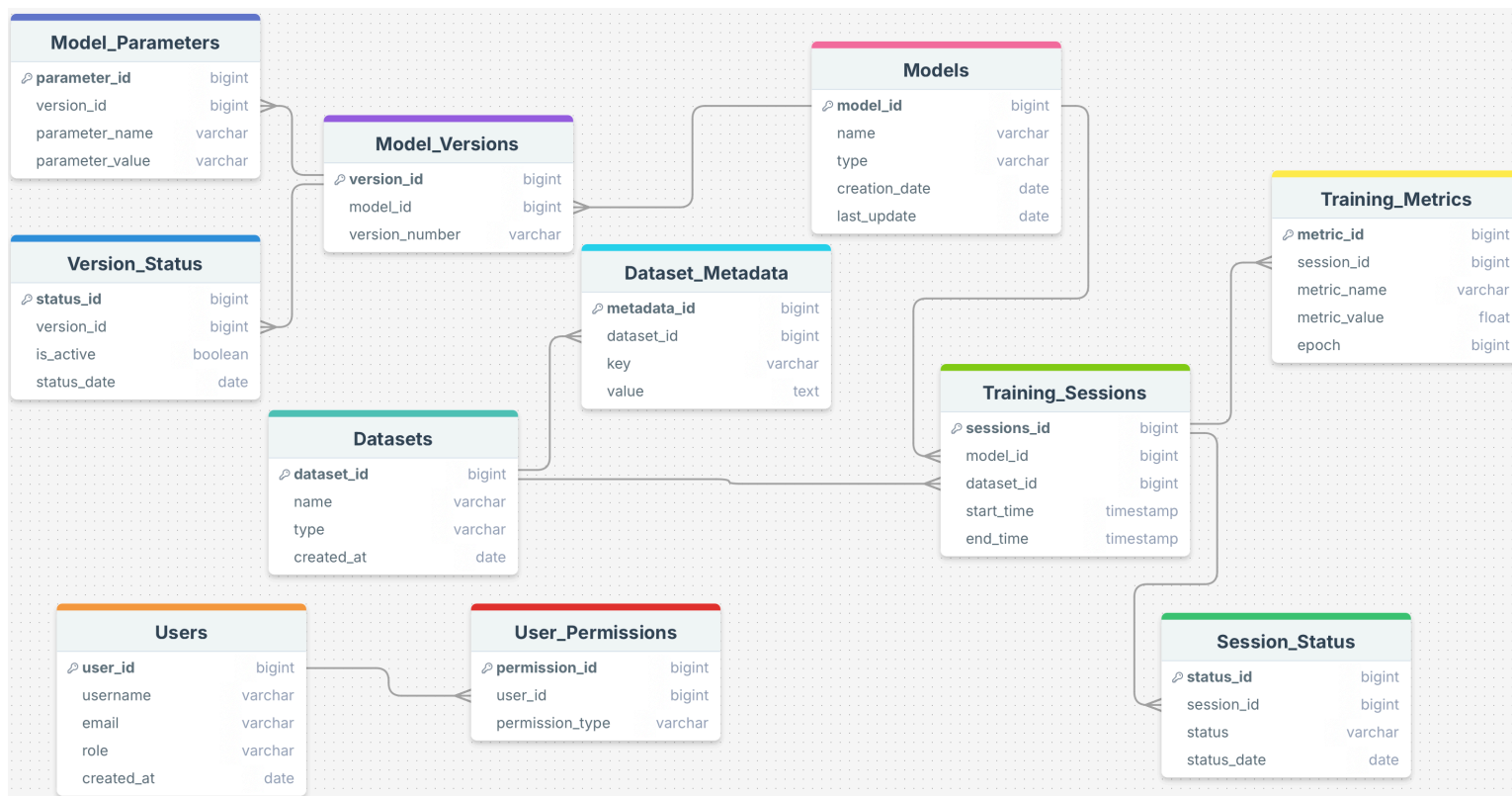


Рисунок 1 – Схема Базы данных

Схема состоит из 7 таблиц (Рисунок 1):

1. **Models** - хранит информацию о модели (название, тип, дату создания, дату последнего обновления)
 - 1.1. model_id (PK) - уникальный id модели
 - 1.2. name - имя модели, название
 - 1.3. type - тип модели (регрессия, классификация и т.д.)
 - 1.4. creation_date - дата создания
 - 1.5. last_update - дата последнего обновления
2. **Model_Versions** - версии моделей
 - 2.1. version_id (PK) - id для просмотра версий
 - 2.2. model_id (FK) - id модели
 - 2.3. version_number - версия модели
3. **Model_Parameters** - параметры для моделей
 - 3.1. parameter_id (PK) - id параметра модели
 - 3.2. version_id (FK) - id версии модели, в которой смотрятся параметры
 - 3.3. parameter_name - название параметра модели
 - 3.4. parameter_value - значение параметра в тренированной модели
4. **Version_Status** - информация о версии (активна ли, и т.д.)
 - 4.1. status_id (PK) - id статуса версии
 - 4.2. version_id (FK) - id для просмотра версий
 - 4.3. is_active - активна ли
 - 4.4. status_date - день присвоения статуса

5. Datasets - таблица для информации о датасетах, на которых тренировали модели
 - 5.1. dataset_id (PK) - id датасета
 - 5.2. name - название
 - 5.3. type - тип датасета
 - 5.4. created_at - дата создания
6. Dataset_Metadata - скрытая/дополнительная информация о датасете
 - 6.1. metadata_id (PK) - id информации
 - 6.2. dataset_id (FK) - id датасета
 - 6.3. key - ключ
 - 6.4. value - значение
7. Users – хранит информацию о пользователях.
 - 7.1. (PK) user_id – Уникальный идентификатор пользователя
 - 7.2. username – Логин пользователя
 - 7.3. email- Почта пользователя
 - 7.4. role – Роль пользователя
 - 7.5. created_at – Дата регистрации пользователя
 - 7.6. hash_password – Хэшированный пароль пользователя
8. User_permissions – хранит уровень доступа пользователя.
 - 8.1. user_id (FK) – Уникальный идентификатор пользователя.
 - 8.2. permission_type – Тип доступа пользователя
9. Training_Sessions – хранит тренировочные сессии
 - 9.1. (PK) session_id – Уникальный идентификатор сессии
 - 9.2. model_id (FK) – Уникальный идентификатор модели
 - 9.3. dataset_id (FK) – Уникальный идентификатор датасета
 - 9.4. start_time – Время начала обучения
 - 9.5. end_time – Время конца обучения
10. Training_Metrics – хранит метрики для тренировок
 - 10.1. (PK) metric_id – Уникальный идентификатор метрики
 - 10.2. session_id (FK) – Уникальный идентификатор сессии
 - 10.3. metric_name – Название метрики
 - 10.4. metric_value – Значение метрики
 - 10.5. epoch – Количество итераций обучения
11. Session_Status - итог статуса сессии
 - 11.1. status_id (PK) - id статуса
 - 11.2. session_id (FK) - id сессии
 - 11.3. status - статус
 - 11.4. status_date - день присвоения статуса

2. Схема приложения

Структура приложения выглядит следующим образом (Рисунок 2):

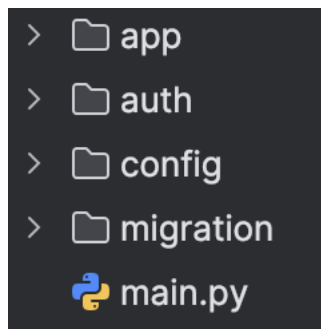


Рисунок 2 – Структура приложения

- 1) В папке `app` хранятся реализации класса базы данных, вспомогательные методы и папка `pages`, в которой хранятся страницы фронта для разных методов (вставки, удаления, добавления, загрузки из `.csv`, просмотра)
- 2) В `auth` реализован класс авторизации, который для каждого пользователя проверяет его права при попытке вызвать тот или иной метод
- 3) В `config` хранится общая информация о бд, в то числе и файл `env.env`
- 4) В `migration` находится файл для создания бд в докере
- 5) `main.py` - основной файл, который в себе объединяет остальные

3. Код приложения

Весь код приложения хранится на Github репозитории по ссылке:

https://github.com/TheAnatolii/BD_kp

Также код приложения можно найти по qr-коду:



4. Вывод

В ходе выполнения курсовой работы на тему "Разработка клиентского приложения «Сервис генеративных сетей»" была успешно разработана схема базы данных и создано клиентское приложение на Python с использованием SQL для взаимодействия с базой данных.

Созданная схема базы данных позволяет структурировать данные, обеспечивая их целостность и эффективность обработки.

Нормализация базы данных помогла минимизировать избыточность и улучшить производительность запросов.

В заключение, данная курсовая работа не только углубила понимание принципов работы с базами данных, но и позволила развить практические навыки разработки программных приложений.