

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ

КУРСОВА РОБОТА
з навчальної дисципліни “Бази даних та інформаційні системи-2”
на тему
ОБЛІК ВІДВІДУВАННОСТІ СТУДЕНТАМИ ЗАНЯТЬ

Виконав студент групи КМ-62
СЕДІНІН Є.А.

Керівник
ТЕРЕЩЕНКО І.О.

Оцінка:
Кількість балів:

ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Створити інформаційну систему, що дозволить вести облік та досліджувати відвідуваність занять з різних дисциплін студентами різних груп. Для цього виконати передпроектне дослідження (проаналізувати бізнес-процеси, бізнес-правила та елементарні події), визначити сутності, що міститимуться у базі даних, проаналізувати операції, що можуть виконуватись над кожною із сутностей, забезпечити цілісність системи. Додатково виконати кореляційний та кластерний аналіз. Розгорнути завдання на віддаленому сервері.

АНОТАЦІЯ

В пояснювальній записці до курсової роботи описано основні етапи розробки інформаційної системи. Перший розділ містить інформацію про виконання перед проектного дослідження. У другому розділі надано детальну постановку задачі. В третьому розділі виконано моделювання бізнес-процесів. У четвертому розділі визначено модель даних, описано сутності інформаційної системи, а в п'ятому розділі надано графічне представлення моделі.

РЕФЕРАТ

В ході виконання курсового проекту було створено інформаційну систему на базі веб-фреймворку Flask. База даних, на якій виконано завдання – PostgreSQL 11.6. Власне модель створюється на базі Python, SQLAlchemy, а зі сторони PostgreSQL налаштовано обмеження на унікальність, які виникли після оптимізації бази даних та введення штучних ключів.

Веб-застосунок дозволяє переглядати дані, наявні в інформаційній системі, додавати нові записи, змінювати та видаляти дані, де це допустимо та не порушує обмежень. Додавання даних валідується на WTForms, на яких також, за допомогою налаштування випадаючих списків та обмеженого вибору підтримано цілісність даних. З Python до HTML-сторінок передається інформація та відображається з допомогою шаблонізатора Jinja, а веб-сторінки зверстані з використанням фреймворку Bootstrap-4.

Також можна виконувати пошук за групою або студентом для перегляду детальної статистики відвідувань – кластерний аналіз. Машинне навчання та кореляційний аналіз (залежність між дисципліною, що викладається та частотою відвідувань) виконані з використанням Random Forest Regressor, а віддалений сервер для розгортання виконаного завдання міститься на Heroku.

ЗМІСТ

ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ.....	2
АНОТАЦІЯ.....	3
РЕФЕРАТ.....	4
ЗМІСТ.....	5
ВСТУП.....	6
1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ.....	7
2 ПОСТАНОВКА ЗАДАЧІ.....	8
2.1 Категорії користувачів.....	8
2.2 Класи даних.....	9
2.3 Бізнес-правила.....	11
2.4 Матриця елементарних подій (сценарії).....	11
3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ.....	14
4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ.....	15
5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ.....	16
ВИСНОВКИ.....	18
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	20

ВСТУП

Станом на сьогодні сервіс, що відслідковує будь-які події (в нашому випадку — відвідування заняття студентом) не чимось новим чи унікальним. Проте цінність будь-яки даних полягає в тому, що після накопичення деякої їх кількості можна використати ці дані для проведення досліджень та оптимізації деяких процесів.

Метою курсової роботи є створити підґрунтя для мікросервісу, що зможе допомогти збирати ці дані та надати інструмент для їх обробки та аналізу. Згодом на базі виконаних досліджень можна буде прогнозувати відвідуваність, оптимізувати розклад занять та зміст курсу навчання. Недоліком такого рішення є те, що не можна отримати достовірні дані про відвідуваність, що буде погіршувати точність то коректність роботи сервісу.

Об'єктом дослідження у курсовій роботі є функціонал наявних інформаційних систем, існуючі рішення, які піддаються інтеграції та облік відвідуваності, його співставлення із групами, студентами, або дисциплінами навчання.

Метод дослідження – пошук та систематизація наявних рішень для створення інформаційних систем та мікросервісів.

1 АНАЛІЗ ПІДПРИЄМСТВА АВТОМАТИЗАЦІЇ

Відповідно до work-flow із постановки задачі, маємо процеси збору інформації про студентів, групи, дисципліни та їх розклад. У короткому представленні ієрархія процесів наведена на рисунку 1.1.

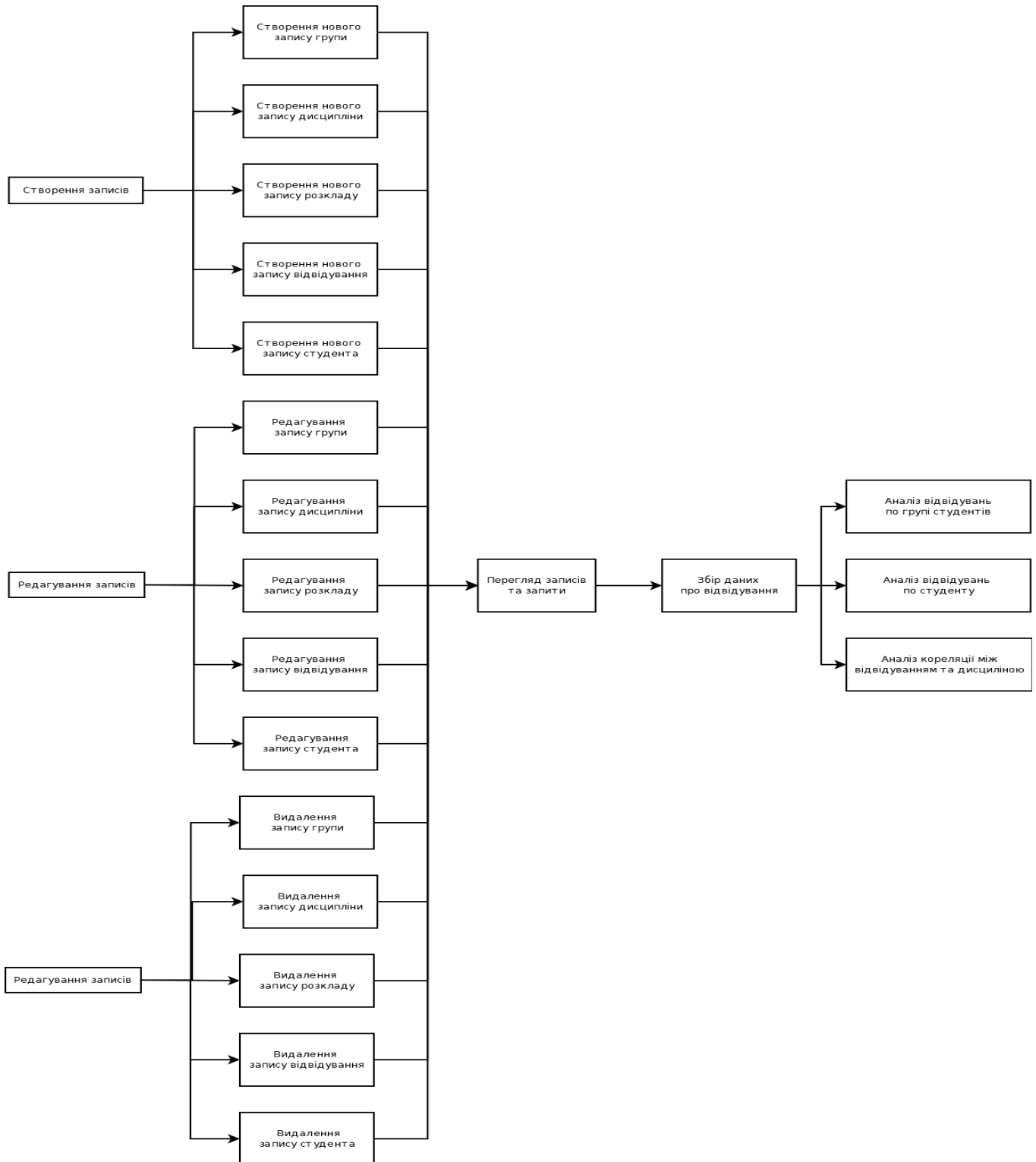


Рисунок 1.1 – Ієрархія процесів

2 ПОСТАНОВКА ЗАДАЧІ

Основна відмінність простої автоматизації робочого процесу від пропонованого мікросервісу в тому, що будемо здобувати корисну інформацію та робити дослідження. Наприклад, інтересу до дисципліни від ріння її відвідуваності студентами.

Основна функція мікросервісу: збирати та зберігати дані про відвідування та надавати їх аналіз.

Вхідні дані мікросервісу: статистика відвідувань (група, студент, дисципліна розклад, відвідування).

Приклад структури вхідних даних мікросервісу: меню або текстове повідомлення в інтерфейсі. Словник: студент – заняття – відвідування.

Результат роботи мікросервісу: дата – файл.

Задача кластеризації: залежність між дисциплінами, групами та рівнем відвідування.

Мета задачі кластеризації для поставленої задачі: для того, щоб будувати правила залежності між дисциплінами та відвідуваністю.

Задача кореляційного аналізу: пов'язувати відвідуваність занять дисципліни та саму дисципліну/групу студентів.

Визначити мету кореляційного аналізу та для яких атрибутів: на основі дисципліни та правил залежності прогнозувати відвідуваність.

Далі розглянемо основні категорії користувачів сервісу, класи даних та елементарні події, що відбуваються у ході користування системою.

2.1 Категорії користувачів

У сервісі, що реалізовано в курсовому проекті є тільки одна категорія користувачів. Звичайно, сервіс можна розширити з додаванням авторизації та різних ролей. Тоді до різних сутностей та дій будуть різні права доступу.

Наприклад, адміністратор зможе робити будь-які CRUD операції прямо на сервісі, не дістаючись при цьому до інтерфейсу PostgreSQL, все буде повністю автоматизовано. Також можна розробити інтерфейс для спілкування із базою.

Наприклад, при натисканні на певну кнопку буде проходити аналіз таблиці «Відвідування», вибирати серед них ті дисципліни які є найменш відвідуваними та відправити звіт.

Для користувача тоді ж буде логічно реєструватися, дивитися свій розклад, тощо.

2.2 Класи даних

Опишемо наявні у системі класи даних у таблицях 2.1 –2.6.

Таблиця 2.1 – Клас даних «Група»

Сутність	Група	
Опис сутності	Група студентів університету	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	Студент, Дисципліна
Назва цієї групи	Назва групи в університеті, наприклад “КМ-62”	—

Таблиця 2.2 – Клас даних «Студент»

Сутність	Студент	
Опис сутності	Основна сутність системи, студент університету	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	Група, Відвідування
Ім'я	Ім'я студента	—
Університет	Університет, в якому навчається студент	—

Факультет	Факультет, на якому навчається студент	—
Група	Група, в якій навчається студент	Група

Таблиця 2.3 – Клас даних «Дисципліна»

Сутність	Виконавець	
Опис сутності	Дисципліна, що викладається	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	Розклад, Група
Назва	Назва дисципліни	—
Група	Група, якій викладається дисципліна	—

Таблиця 2.4 – Клас даних «Розклад»

Сутність	Розклад	
Опис сутності	Розклад занять для груп	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	Відвідування
Дисципліна	Назва дисципліни	Дисципліна
Аудиторія	Аудиторія, де проводиться заняття	—
Дата	Дата проведення лекції	

Таблиця 2.5 – Клас даних «Відвідування»

Сутність	Відвідування	
Опис сутності	Таблиця з обліком відвідувань	
Атрибути сутності	Опис атрибуту	Пов’язана сутність із атрибутом
ID	Універсальний ідентифікатор	—
Дисципліна	Дисципліна, що	Дисципліна

	розглядається	
Розклад	Дані про розклад цього заняття	Розклад
Студент	Студент, що розглядається	Студент
Відвідування	Чи було відвідано заняття	—

2.3 Бізнес-правила

Далі слід визначити бізнес-правила для інформаційної системи:

а) студент повинен мати факультет, групу, де він навчається;

б) студент повинен мати ім'я;

в) дисципліна повинна мати групу, в якій викладається;

г) розклад повинен мати дисципліну до якої відноситься;

д) відвідування може бути заповнено лише на розклад, дисципліну та студента, що пов'язані один з одним;

е) можна оперувати та дивитися статистику лише за екземплярами сутностей, що наявні у базі даних.

2.4 Матриця елементарних події (сценарії)

Перший рівень містить багато однотипних процесів, тому в їх рамках будемо описувати лише один, а інші – аналогічні.

Таблиця 2.6 – Процес створення нового запису в сутності

Назва процесу	Додавання запису до сутності
Сутності	Відповідно кожна із сутностей, зазначена у таблицях 2.1–2.5 цього розділу
Вхідні атрибути сутності	В процесі додавання вхідні атрибути пусті, бо запис ще не вставлено
Опис функціоналу	Дані з форми на веб-сторінці записуються у поля класу, що відповідають сутності та відправляються на сервер

Змінені атрибути сутності	Рядок вставлених даних
---------------------------	------------------------

Таблиця 2.7 – Процес редагування запису в сутності

Назва процесу	Редагування запису у сутності
Сутності	Відповідно кожна із сутностей, зазначена у таблицях 2.1–2.5 цього розділу
Вхідні атрибути сутності	Рядок, отриманий за ID поля, яке збираємося відредагувати
Опис функціоналу	Дані приходять із серверу та пишуться у поля форми. Після цього у формі користувач змінює значення атрибутів. Дані зчитуються та відправляються на сервер
Змінені атрибути сутності	Рядок відредагованих даних

Таблиця 2.8 – Процес видалення запису із сутності

Назва процесу	Видалення запису із сутності
Сутності	Відповідно кожна із сутностей, зазначена у таблицях 2.1–2.5 цього розділу
Вхідні атрибути сутності	Унікальний ідентифікатор
Опис функціоналу	Отримуємо запис сутності за ID, що надійшов після натискання на відповідну кнопку, та видаляємо запис за цим ID
Змінені атрибути сутності	Тепер рядок за цим ідентифікатором відсутній

Таблиця 2.9 – Процес виставлення відвідування

Назва процесу	Виставлення відвідування
Сутності	Відвідування
Вхідні атрибути сутності	Всі атрибути сутності відвідування

Опис функціоналу	Обробка відвідувань, які з них відповідають дійсності (це заняття було, з цієї дисципліни, та на неї мав прийти цей студент)
Змінені атрибути сутності	Виставлено логічне значення — відвідано заняття або ні

Таблиця 2.10 – Процес аналізу відвідувань групи

Назва процесу	Аналіз відвідувань групи
Сутності	Студент, Група, Розклад, Дисципліна, Відвідування
Вхідні атрибути сутності	Група
Опис функціоналу	Користувач обирає групу з випадального списку та натискає кнопку пошуку. Відбувається операція SELECT із об'єднання сутностей по ключам. У вигляді графіку відображаються кількості відвідувань занять по дисциплінам.
Змінені атрибути сутності	Сутність не змінює атрибутів

Таблиця 2.11 – Процес аналізу відвідувань студента

Назва процесу	Аналізу відвідувань студента
Сутності	Студент, Розклад, Дисципліна, Відвідування
Вхідні атрибути сутності	Студент
Опис функціоналу	Користувач обирає студента з випадального списку та натискає кнопку пошуку. Відбувається операція SELECT із об'єднання сутностей по ключам. У вигляді графіку відображаються кількості відвідувань занять по дисциплінам.
Змінені атрибути сутності	Сутність не змінює атрибутів

3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

Змодельований Use-Case для процесів, що можна виконувати наразі у веб-застосунку, зображено на рисунку 3.1.

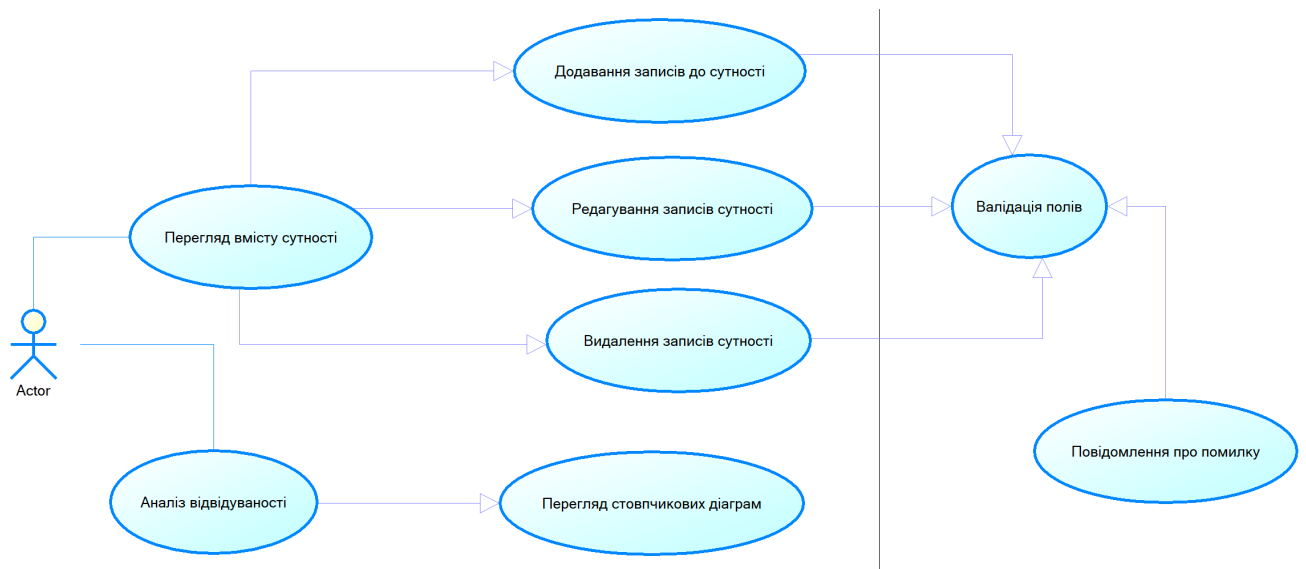


Рисунок 3.1 – Use-case diagram для веб-застосунку

4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

Змоделюємо сутності та зв'язки звичною мовою.

а) Відмічається відвідування студентом заняття з дисциплін.

б) Заняття має розклад, дисципліну, час та місце проведення. Заняття проходить у рамках однієї дисципліни та в одн час. В той час дисципліна має багато занять.

в) Дисципліна викладається для однієї групи та має багато занять.

г) Група має багато студентів

д) Студент належить до однієї групи, факультету, ніверситету та має ім'я.

Таким чином, студента ідентифікують ім'я, група, факультет та університет. Групу та дисципліну можна визначити за назвою та групою, для якої викладається. Розклад визначається за дисципліною. Відвідування визначається заняттям (розклад), дисципліною та студентом.

Концептуальна діаграма зображена на рисунку 4.1.

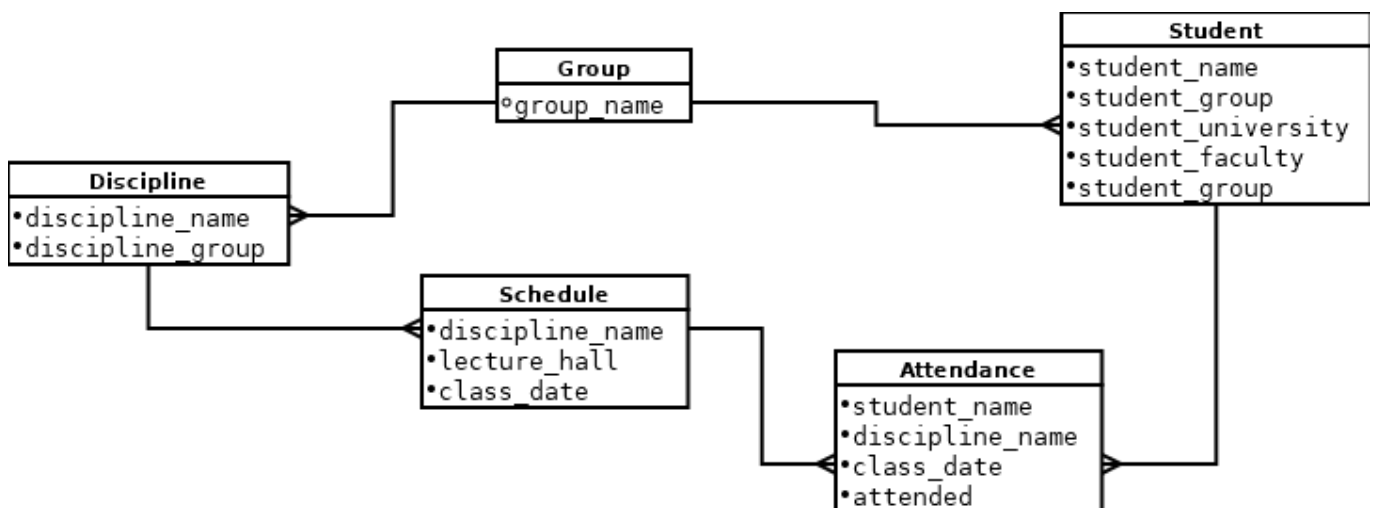


Рисунок 4.1 – Концептуальна діаграма бази даних

5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

Наступний крок – перетворити концептуальну модель на логічну та оптимізувати її. При такому перетворенні вводять штучні ключі, а на атрибути, що були ключовими у концептуальній моделі накладається обмеження унікальності. Оптимізована логічна модель зображена на рисунку 5.1.

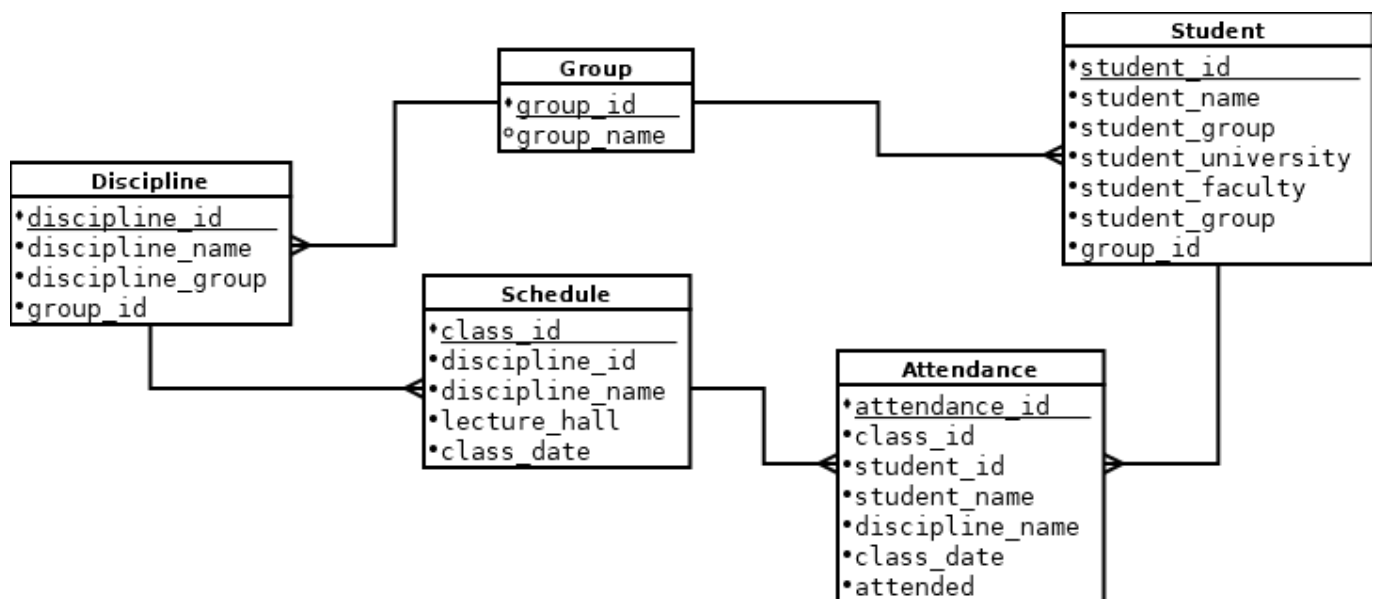


Рисунок 5.1 – Оптимізована логічна модель даних

Таким чином, отримали модель даних, де кожна сутність містить штучні ключі. Штучні ключі в отриманому сервісі редагуватись користувачем не будуть: відбуватиметься автоінкремент у влаштованих процедурах. Тепер завдяки тому, що маємо штучні ключі, WTF-форми на сайті працюватимуть коректно, будуть наявні HiddenField().

Кожну таблицю моделі можна буде пов'язати з будь-якою іншою, виконавши перехресні записи із JOIN. Цілісність за посиланнями підтримуватиметься перехресними запитами з кожної форми, якщо у відповідній формі сутності є

батьківська. Якщо батьківських сутностей немає – не обмежуємо користувача у вводиті інформації, окрім валідації полів.

ВИСНОВКИ

В ході виконання курсової роботи було визначено наступні етапи впровадження проекту: побудова ієрархії процесів, обґрунтування постановки задачі, визначення сценаріїв перебігу елементарних процесів, визначення класів даних, сутностей та атрибутів, побудова моделей даних, інтеграція відомих рішень та їх покращення новим функціоналом.

Як результат виконання проекту досягнуто: інформаційна система, що дозволяє переглядати, додавати, іноді змінювати та видаляти зміст сутностей. База даних, що взята за основу – PostgreSQL 11, а на Python – SQLAlchemy. Веб-застосунок збудований на базі фреймворків Flask та Bootstrap-4. Ресурс розгорнутий на хостингу Heroku.

Окремі вкладка, зроблена на сайті, дозволяють переглядати статистику відвідувань по групі або по студенту.

У частині, що відповідає за машинне навчання застосовано Random Forest Regressor для визначення кореляції між дисципліною та її відвідуваністю – дані взято з власної бази даних. Точність моделі – 97%. В залежності від кількості дерев прийняття рішень варіюється. При точності моделі 97% (200 дерев, 80% даних навчальні, 20% – тестові) середньоквадратичне відхилення становить 1.7. При зменшенні кількості навчальних даних точність може збільшуватись. Це обумовлене тим, що кореляція нерівномірна в рамках датасету.

Розроблена система безумовно потребує допрацювання. Для того, щоб далі з нею могло працювати більше людей, необхідно розробити авторизацію та різні ролі, а веб-інтерфейс для користувачів системи перенести у систему таку, як Telegram або Google Calendar, зробити зручні кнопки та повну інтерактивність. А для режиму розробника залишити можливість виконання усіх CRUD-операції із веб-інтерфейсу. До збудованої бази даних можна робити запити складнішої структури, щоб

діставати коректні дані для їх аналізу та побудови статистики/звітв. Поки що надана інформація не може дати можливість зробити конкретні висновки або аналіз. Також можна навчити модель змінити розклад занять в залежності від її відвідуваності, часу та місця проведення.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Create An API To Deploy Machine Learning Models Using Flask and Heroku [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/create-an-api-to-deploy-machine-learning-models-using-flask-and-heroku-67a011800c50>.
2. PostgreSQL 10.11 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/docs/10/index.html>.
3. SQLAlchemy 1.3 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.sqlalchemy.org/en/13/>.
4. WTForms Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://wtforms.readthedocs.io/en/stable/>.