

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«ІНФОРМАЦІЙНА СИСТЕМА ОРГАНІЗАЦІЇ НАВЧАЛЬНОГО
ПРОЦЕСУ ДЛЯ ОНЛАЙН ШКОЛИ»**

на здобуття освітнього ступеня бакалавр

за спеціальності 126 Інформаційні системи та технології

(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Віталій ЯРИГІН

(ім'я, ПРІЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група ІСД-42

Віталій ЯРИГІН

(ім'я, ПРІЗВИЩЕ)

Керівник
к. т. н. доц.

Ольга ПОЛОНЕВИЧ

(ім'я, ПРІЗВИЩЕ)

Рецензент:

(ім'я, ПРІЗВИЩЕ)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедри ІСТ

_____ Каміла СТОРЧАК

“ _____ ” _____ 2025 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Яригіну Віталію Геннадійовичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Інформаційна система організації навчального процесу для онлайн-школи

керівник кваліфікаційної роботи Ольга ПОЛОНЕВИЧ, к.т.н. доцент

(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій

від « 24 » лютого 2025 р. № 56

2. Строк подання кваліфікаційної роботи «30» травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи:

1. Принципи роботи бази даних та збереження інформації;
2. Принципи побудови моделей та діаграм;
3. Науково-технічні джерела та книги.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Дослідження понять інформаційна система та база даних.
2. Побудова моделей за допомогою мов ER-діаграми та UML.
3. Створення бази даних для онлайн-школи, використовуючи мову MySQL.

5. Перелік ілюстративного матеріалу: *презентація*.

6. Дата видачі завдання « 24 » лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів Кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Пошук книг та матеріалу.	01.03-09.03	
2	Проведення аналізу інформаційної системи.	12.03-16.03	
3	Ознайомлення з базами даних та системами керування базами даних.	05.04-14.04	
4	Побудова моделей життєвого циклу.	15.04-21.04	
5	Ознайомлення з уніфікованою мовою UML та побудова діаграм.	21.04-29.04	
6	Створення бази даних на мові MySQL для інформаційної системи.	01.05-10.05	
7	Оформлення дипломної роботи.	10.05-11.05	
8	Підготовка до доповіді та створення презентації.	13.05-15.05	

Здобувач вищої освіти

(підпис)

Віталій ЯРИГІН

(Ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи

(підпис)

Ольга ПОЛОНЕВИЧ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра (магістра): 61 стор., 49 рис., 11 табл., 10 джерел.

Мета роботи - розробка інформаційної системи, що забезпечує ефективну організацію, управління та моніторинг навчального процесу в онлайн школі.

Об'єкт дослідження - процес організації та управління навчальним процесом в онлайн освітньому середовищі.

Предмет дослідження - методи та програмні засоби створення інформаційної системи для автоматизації організації навчального процесу в онлайн школі.

У першому розділі аналізується інформаційна система платформи онлайн-школи, визначаються поняття інформація, система, інформаційна система, база даних. Також, будується інфологічна модель на основі мови ER-діаграм. У другому розділі описуються та будуються основні моделі життєвого циклу, такі як: послідовна, ітеративна/інкрементально-еволюційна та спіральна. Далі аналізується уніфікована мова UML та створюються відповідні діаграми, які пов'язані з інформаційною системою онлайн-школи та допоможуть при розробці бази даних. У третьому розділі представлена сама база даних, вбудовані таблиці, які містять основну інформацію школи. Код для вказаних маніпуляцій був написаний за допомогою мови MySQL в програмі DB Browser for SQLite.

КЛЮЧОВІСЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, MYSQL, БАЗА ДАНИХ, ОНЛАЙН-ШКОЛА, ЖИТТЄВИЙ ЦИКЛ, UML.

ЗМІСТ

ВСТУП.....	9
1 Аналітика та особливості онлайн-школи.....	10
1.1 Принцип роботи інформаційної системи.....	10
1.2 Принципи зберігання інформації.....	13
1.2.1 База даних.....	13
1.2.2 Системи керування базами даних.....	16
1.2.3 Моделі даних.....	17
1.3 Інфологічна модель.....	18
1.3.1 Основні визначення та поняття.....	18
1.3.2 Опис зв'язків.....	19
2 Моделі життєвого циклу розробки онлайн-платформи.....	23
2.1 Життєвий цикл.....	23
2.1.1 Водоспадна, ітеративна та спіральна модель.....	23
2.2 Основи моделювання UML.....	29
2.3 Проєктування моделей бази даних.....	30
2.3.1 Діаграма прецедентів.....	31
2.3.2 Діаграма діяльності.....	33
2.3.3 Діаграма послідовності.....	34
2.3.4 Діаграма класів.....	34
2.3.5 Діаграма розгортання.....	35
3 Реалізація та використання інформаційної системи.....	37
3.1 Створення таблиць для бази даних онлайн-школи.....	37
3.2 Заповнення інформацією таблиць.....	48
ВИСНОВКИ.....	56
ПЕРЕЛІК ПОСИЛАНЬ.....	57
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ).....	58

ВСТУП

Актуальність теми. З кожним днем вчені здійснюють нові відкриття та роблять винаходи, які полегшують життя. У 20-му столітті люди навіть гадки не мали, що маленький технічний збій може збити темп життя та призупинити роботу будь-якої станції, що виконує важливу роль для цілого населення країни. Суспільство розвивається та цифровий світ проник у кожен аспект людського життя.

У 2019 році Землю охопила інфекційна хвороба, що переросла у пандемію. Контактвання людей між собою зменшилася і це значно вплинуло на деякі сфери людського життя. Для прикладу взяти навчальні заклади. Студенти, школярі та викладачі перейшли на дистанційне навчання. І на допомогу прийшли інформаційні технології, які стали основним ключем до забезпечення безперервного навчання.

Більшість почали замислюватися над розвитком своєї особистості та пошуку себе. З'являється все більше відео на цю тему, випускаються нові книги, створюються цілі канали та акаунти, де люди діляться інформацією про самовдосконалення. А саме головне, це з'являються нові онлайн-шкіл, які є корисними для користувачів будь-якого віку та суспільного прошарку. Такі платформи мають, як і переваги, так і недоліки. З позитивних сторін можна визначити: навчання в зручному форматі та часі; заощадження часу та доступність. А ось з негативних сторін зазвичай визначаються такі: відсутність самодисципліни; труднощі та технічні складності у використанні платформ. Але, не зважаючи на ці сторони, питання навчання людей було і залишається важливим на сьогоднішній день.

Мета роботи - розробка інформаційної системи, що забезпечує ефективну організацію, управління та моніторинг навчального процесу в онлайн школі.

Об'єкт дослідження - процес організації та управління навчальним процесом в онлайн освітньому середовищі.

Предмет дослідження - методи та програмні засоби створення

інформаційної системи для автоматизації організації навчального процесу в онлайн школі.

Практичне значення роботи полягає у створенні інформаційної системи, яка дозволяє автоматизувати основні процеси управління навчанням в онлайн школі.

1 АНАЛІТИКА ТА ОСОБЛИВОСТІ ОНЛАЙН-ШКОЛИ

1.1 Принцип роботи інформаційної системи

За останні десятиліття інформація є основним ресурсом для розвитку суспільства. Слово "*інформація*" (від латинського *informatio*), що визначається як поняття про що-небудь. В середині 20 століття цей термін став загальнонауковим, що означає сукупність відомостей про об'єкти, події, процеси, їх якості та взаємини. Серед основних властивостей інформації можна визначити:

1. Релевантність - здатність інформації відповідати запитам клієнта.
2. Повнота - властивість, що характеризує об'єкт або/та процес, який відображається для споживача.
3. Своєчасність - здатність інформації відповідати запитам в потрібний момент.
4. Достовірність - ознака інформації мати будь-яких непотрібних помилок.
5. Доступність - ознака інформації, що описує можливість її отримання.
6. Захищеність - ознака, що дає неможливість непотрібного та незаконного користування, коригування та інше.
7. Ергономічність - властивість, що описує зручність об'єму або обсягу інформації зі сторони користувача.
8. Адекватність - також одна з ознак інформації ідентично відповідати об'єкту, що відображається. Для споживача - внутрішня характеристика інформації, що проявляється через достовірність та релевантність.
9. Живучість - здібність інформації берегти свою особливість протягом довгого часу.
10. Унікальною інформацією називають ту, яка зберігається тільки в одному екземплярі.

Унікальність та живучість є ознаками інформації, які пов'язують з процесом її збереження.

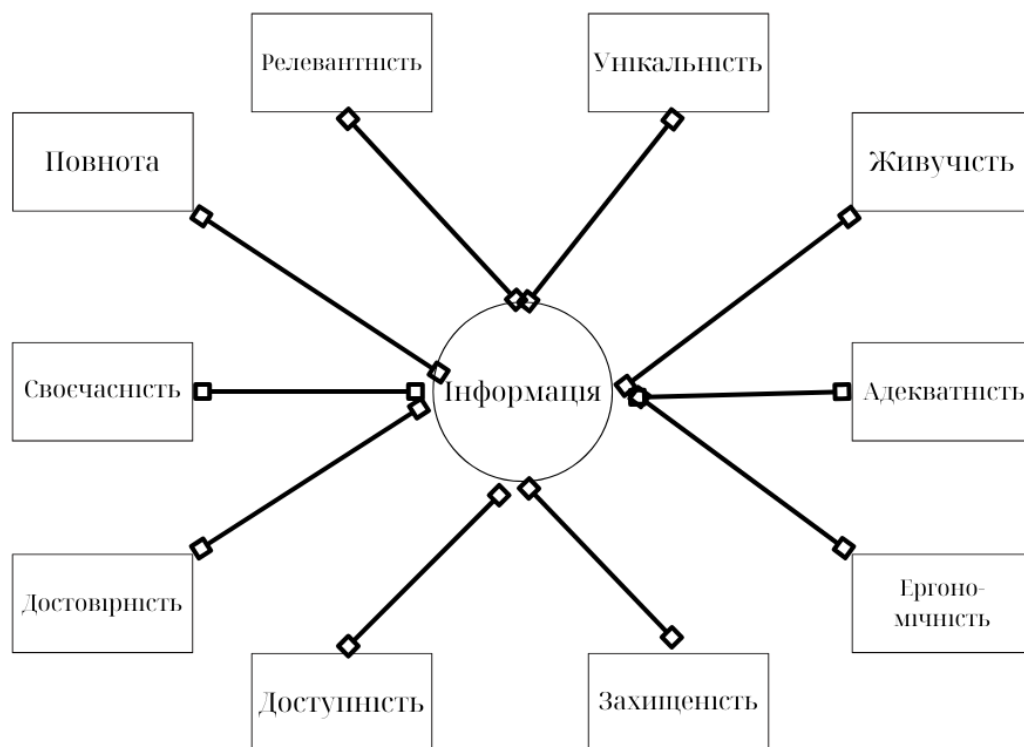


Рис.1.1 Основні властивості інформації

В житті цивілізації інформація отримала настільки широкого поширення, що вона має не тільки одне визначення. Так само можна сказати про інше поняття, як "система". В різних галузях цей термін розповсюджений і має безліч значень. *Система* (від грецького слова *systema*) означає щось складене з частин, ціле, елементи, що взаємодіють і мають зв'язки один із одним, що створюють певну єдність. Додавання до слова "інформаційна" термін "система" втрачає її головну мету функціонування та функціонування. Інформаційна система допомагає аналізувати проблеми та створювати нові продукти, завдяки зберігання, обробку, збору, пошуку, видачі інформації, яка необхідна у прийнятті рішення із будь-якої області.

Інформаційна система - це комплекс технічних, програмних та організаційних засобів, які накопичують, зберігають, коригують, здійснюють пошук, оброблюють та видають інформацію. Серед властивостей інформаційної системи можна виділити:

- ІС може аналізуватися, управлятися та споруджуватися на базі основних принципів створення систем;

- ІС може розвиватися та бути динамічною;

- при створенні ІС потрібно використовувати системний підхід.

Кожна інформаційна система має кілька ключових компонентів:

- *апаратне забезпечення* - це фізичні елементи інформаційної системи;

- *програмне забезпечення* - це програми, які керують роботою системи;

- *телекомунікації* виконують роль передачі інформації в межах системи;

- *люди* володіють системою та взаємодіють із нею;

- *дані* - це інформація, яка зберігається в системі та обробляється нею.

В інформаційних системах виконуються наступні головні процеси: збір, передача, зберігання та прийом інформації. *Збором* називають процес перетворення інформації в двійковий код; потім цей двійковий код *обмінюється* між модулями інформаційної системи, де вони кодуються - декодуються, канал зв'язку моделює - демоделює сигнали, пізніше їх узгоджують та обов'язково посилюють; далі відбувається *зберігання* інформації, вибирають спосіб збереження, актуалізують та вилучають з бази даних; останній *процес прийому* інформації - оброблені дані отримують для їх подальшого аналізу. *Теорія інформаційних процесів* - комплекс моделей та методів, які призначені (або використовуються) для проведення аналізу інформаційних процесів, що виконуються в різних системах: соціальних, біологічних, економічних та інших.

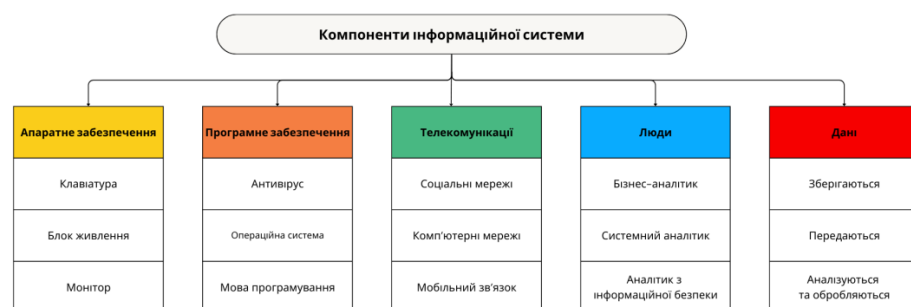


Рис.1.2 П'ять головних компонентів ІС та їх приклади

1.2 Принципи зберігання інформації

Виділяють два методи для організації даних на носіях інформацій - це файлова організація та організація, що представлена у вигляді баз даних.

Для прикладу, файлова організація спеціалізує та зберігає дані у файлах зазвичай для однієї прикладної задачі, і програмується самим прикладним програмістом. Це дозволяє отримати високу швидкість обробки даних, але має значні недоліки:

- значна надмірність даних - необхідність зберігати однакові елементи даних у відмінних системах вузьких спеціалізацій програм і файлів;
- відсутність виявлення даних, які суперечать одні одній - для виконання однакових дій з різними файлами потрібні різні програми, а керуються вони різними людьми (групами людей);
- незадоволення інформаційних запитів користувачів - організація файлів даних не забезпечує виконання більшості інформаційних запитів навіть тоді, коли для цього є всі потрібні елементи, які перебувають у наявних файлах.

Натомість, виникає питання відокремлення даних від їхнього опису, вибрати потрібну організацію збереження даних з урахуванням наявних зв'язків між ними, щоб дозволити використання цих даних для більшості додатків. Вирішення цього питання призвело до створення баз даних і систем керування базами даних, що дають змогу забезпечення успішних процесів введення та доступу до різних даних.

1.2.1 База даних

Поняття "база даних" почали вживати в 60-х роках 20-го століття. Відокремлюються десятки різних означень цього терміну. Але саме в інформаційних системах, база даних (БД) - це сукупність структурованих даних, які показують стан об'єктів конкретної предметної області та зв'язків між ними.

Предметна область (ПО) - це сфера, де застосовується конкретна база даних. Ця база містить інформацію лише про ту частину предметної області, яка необхідна для користувачів. База даних предметної області платформи онлайн-школи розробляється для потреб студентів, в ній буде міститися інформація про основну інформацію студента, навчальні дані, фінансова інформація та інше. Для викладачів база даних міститися інформацію про основну інформацію викладача, професійна інформація, навчальна інформація, звіти та оцінювання, інше.

Об'єктами ПО платформи онлайн-школи будуть студенти, викладачі, курси, модулі уроків, тестування, розклад, оплати, комунікація, сертифікати та аналітика. Також для об'єктів є характерними їх параметри та їх значення:

1. Студент: повне ім'я, дата народження, стать, контактна інформація, фото (за бажанням);
2. Викладач: повне ім'я, дата народження, стать, фото, контактні дані, освіта, сертифікати, досвід роботи;
3. Курс: назва курсу, опис, тривалість, напрямок, викладач, вартість;
4. Модуль уроку: структура курсу, відео/текстові матеріали, завдання, тести/звіти, матеріали для завантаження;
5. Тестування: тести, домашні завдання, критерії оцінювання, відгуки викладача;
6. Розклад: дата та час уроків, записи вебінарів, календар подій;
7. Оплата: стан рахунку, історія транзакцій, плани підписки, акції/знижки;
8. Комунікація: чати, форуми, повідомлення, коментарі до уроків;
9. Сертифікати: умови отримання, дата видачі, ідентифікатор;
10. Аналітика: прогрес студента, активність викладача, відвідуваність, результати навчання.

Між конкретними об'єктами також існують зв'язки:

1. Студент: *записується на* → курс, *проходить* → модуль уроку, *має* → розклад, *проходить* → тестування, *бере участь у* → комунікації, *здійснює* → оплату, *отримує* → сертифікат, *має* → аналітику;

2. Викладач: *викладає*→ курс, *створює*→ модуль уроку, *проводить*→ тестування, *створює* → розклад, отримує → оплату,*бере участь у*→ комунікації, *генерує* → аналітику, *затверджує*→ сертифікати;

3. Курс: *складається з* → модулів, *має* → студентів, *має*→ викладача, *має*→ розклад, *має* → тестування*пов'язаний з*→ оплатою, *видає* → сертифікати, *має*→ аналітику, *має*→ комунікацію;

4. Модуль уроку: *належить до*→ курсу, *може містити*→ тестування, *має*→ розклад, *має*→ комунікацію;

5. Тестування: *прив'язане до*→ курсу, *проходить*→ студент, *створює*→ викладач, *результат входить у*→ аналітику;

6. Розклад: *пов'язаний з* → модулем уроку, *створюється* → викладачем, *використовується*→ студентом;

7. Оплата: *здійснюється*→ студентом, *прив'язана до*→ курсу;

8. Комунікація: *пов'язана з*→ курсом; *між* → студентом та викладачем;

9. Сертифікати: *отримуються*→ студентом; *за проходженням*→ курсу; *підтверджуються*→ викладачем;

10. Аналітика: *створюється для*→ студента/курсу/викладача, *базується на*→ тестуваннях/модулем уроку.

Тож, на сьогоднішній день бази даних можна порівняти з електронними картотеками, де кожна картка містить інформацію про конкретний об'єкт, а також має доступ на другі картки, що представляють собою зв'язані об'єкти. *Структуру (схему)* бази даних відрізняють, орієнтуючись на її наповнення. Структура встановлює, які **параметри** мають мати об'єкти, що містяться в базі, а наповнення - це **значення** зміст параметрів певних об'єктів, які зберігаються в базі даних на даний момент.

1.2.2 Системи керування базами даних

Бази даних - це не окремий вид програмного забезпечення (ПЗ), а лиш документи, з якими взаємодіють певні прикладні програми - системи керування базами даних. Система керування базами даних (СКБД) - це програма, що створює БД та виконує різні операції з даними, які зберігаються в самій системі.

Із системою керування базами даних взаємодіють не лише люди, а й другі прикладні програми. Ця система, так би мовити, створює оболонку навколо баз даних і створює правила роботи з ними для внутрішніх та зовнішніх користувачів. Ці правила розповсюджуються, як і для людей, так і для програм одномірно.

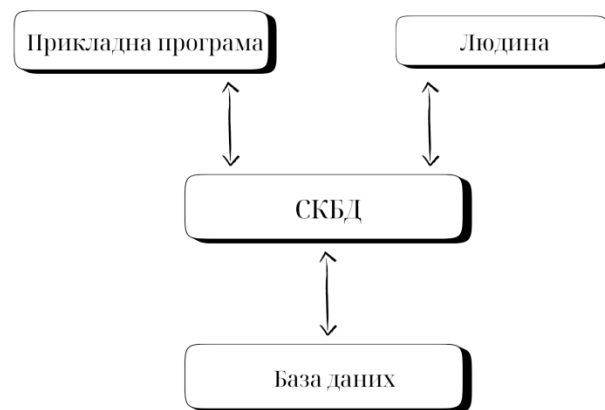


Рис.1.3 Структура роботи СКБД

Саме ці правила роблять інформацію в базі даних більш захищеною, на відміну документів інших типів, до яких прикладні програми можуть звертатися протягом будь-якого часу. Для прикладу, можна вводити будь-що у текстовий документ або будь-які дані у таблицю, але база даних такого не допустить, адже введення інформації, що не відносяться до структури бази або іншим умовам, буде відхилено самою СКБД. Таку властивість даних, що містяться в базах, називають цілісністю. Цілісність даних - це сукупність умов та правил відповідної структури БД, що забезпечують послідовність і несуперечливість даних.

Бази даних та системи керування базами даних не є найдосконалішими

засобами зберігання інформації. Для прикладу, інформацію, що людина запам'ятовує, адекватно відтворити в БД майже неможливо, бо відомості, які люди запам'ятовують - нечіткі, неповні та суперечливі. Здебільшого, СКБД не роблять логічні висновки зі збереженою інформацією у вигляді фактів в самій базі. Для вирішення таких питань користуються *базами знань* та *експертними системами*.

1.2.3 Моделі даних

Як вже згадувалося раніше, структура об'єктів та зв'язків між ними, інформація, що зберігається в базі даних, визначає схема БД. А ось самі правила будови схеми та всі операції, які ми здатні робити над даними, визначає модель даних.

Модель даних - це порядок правил, за якими створюють структуру даних, що дають доступ до змін над даними. Більшість сучасних систем керування базами даних підтримують *реляційну модель даних*, предками якої були *мережева* та *ієрархічна* моделі. В основу реляційної моделі даних покладено поняття відношення (англійською - relation), яка була визначена англійським кібернетиком Едгаром Франком "Теддом" Коддом у 1970 році. В реляційній моделі у формі відношень подаються у вигляді таблиць, кожна з яких має набір однотипних об'єктів. Серед них найбільш перспективною вважають *об'єктно-орієнтовану модель даних*. Проте ця модель зображає ідеологію ієрархічної моделі і розвивається повільно, тому ще довгий час на ринку СКДБ будуть на першому місці реляційні системи.

Маніпулювання даними відбуваються спеціальними мовами, якими клієнти описують запити до СКДБ - це *мови маніпулювання даними*, або, як їх ще називають, *мови запитів*. Також, до складу моделі даних, відносять ще одну мову, яка описує структуру баз даних - *мова визначення даних*.

У реляційній моделі відомою і часто використовуваною є SQL (від англійського *StructuredQueryLanguage* - структурована мова запитів), яка в

містить в собі засоби мови визначення та мови маніпулювання даними. На сьогоднішній день саме цією мовою можна описувати запити до всіх відомих реляційних СКБД. Але SQL не є мовою програмування, вона декларативна, тобто користувач може описати, що хоче отримати, не описуючи, як саме комп'ютер повинен розрахувати необхідний результат. Ця мова, хоч і є популярною, але вона не зовсім точно відповідає теорії реляційної моделі даних, за що її розкритикував сам Е. Кодд. Він запропонував інші дві мови маніпулювання даними - реляційне числення та реляційну алгебру, ще до появи SQL та саме засоби цих двох мов було включено до складу SQL, але з незначними неточностями.

1.3 Інфологічна модель

Інфологічна модель - це відображення інформації про предметну область на логічному рівні абстрагування та опис об'єктів, їх взаємозв'язків та властивостей. Метою такого моделювання є забезпечити природний спосіб відображення та збору інформації для людини, щоб відтворити в створюваній базі даних. Тому таку модель даних будують спільно з природною мовою, адже використати її в автентичному вигляді не можливо через труднощі обробки комп'ютерних текстів та різноманітність будь-якої природної мови).

1.3.1 Основні визначення та поняття

Серед основних конструктивних елементів інфологічної моделі вирізняють сутності, зв'язки, які створюються між ними та їхні атрибути (властивості), які згадувалися вище.

Сутність - будь-який об'єкт (за умови, якщо його можна вирізнити від іншого об'єкта), його інформацію потрібно тримати в базі даних.

Атрибут - іменована характеристика сутності. Найменування має бути особливим для певного типу сутності, але, також, може бути ідентичним для іншого типу сутностей. Використовується для того, щоб зібрати всю можливу

інформацію про сутність.

Ключ - набір атрибутів, який дає можливість відшукати потрібний екземпляр сутності. Такий набір має бути мінімальний, тобто, при видаленні з набору якогось атрибуту не можна буде знайти сутність по атрибутам, які залишилися.

Зв'язок - зв'язування двох або більше сутностей. Одне з головних вимог до структурування бази даних - це гарантування можливості відшукати одні сутності за значеннями інших, тому головне правильне встановлення зв'язків між ними.

1.3.2 Опис зв'язків

При побудуванні інфологічної моделі можна користуватися мовою ER-діаграм (*EntityRelationship*, або *ж* *сутність-зв'язок*). В цій мові сутності позначаються прямокутниками, асоціації - шестикутниками або ромбами, атрибути відображають овалами, а зв'язки - ненаправленими ребрами, над якими проставляють ступінь зв'язку (буква М, тобто *many*, або 1) і обов'язково пояснення.

Для прикладу візьмемо сутності з платформи онлайн-школи та побудуємо від них важливі зв'язки:

1. **Зв'язок "Один до одного" (1:1).** Одна сутність "Викладач" має доступ до однієї сутності "Курс" (рис. 1.4).



Рис.1.4 Зв'язок "Один до одного"

2. (1:M). Одна сутність "Курс" може складатися з жодного, одного або декількох екземплярів сутності "Модуль уроку" (рис. 1.4).



Рис.1.4 Зв'язок "Один до багатьох"

3. Зв'язок "Багато до одного"(1:M). Жоден, один або декілька екземплярів сутностей "Модуль уроку" може містити одне "Тестування" (рис. 1.5).



Рис.1.5 Зв'язок "Багато до одного"

4. Зв'язок "Багато до багатьох"(M:M). Жоден, один або кілька екземплярів сутності "Студент" може здійснювати жоден, один або багато екземплярів сутності "Оплата" (рис. 1.6).



Рис.1.6 Зв'язок "Багато до багатьох"

На основі даної інформації побудуємо таблицю тільки важливих зв'язків

між об'єктами предметної області онлайн-школи, у форматі, який показує тип зв'язку та напрямок взаємодії (таблиця 1.1).

Таблиця 1.1

Таблиця важливих зв'язків обраної ПО

СутністьА	Зв'язок	СутністьВ	Тип зв'язку	Коментар
<i>Студент</i>	здійснює	<i>Оплату</i>	багато до багатьох	Різні студенти можуть здійснювати декількаоплат
<i>Викладач</i>	має доступ до	<i>Курсу</i>	один до одного	Один конкретний викладач має доступ до одного курсу
<i>Оплата</i>	пов'язана з	<i>Курсом</i>	багато до одного	Багато оплат від різних студентів можуть бути пов'язані з одним курсом
<i>Оплата</i>	надходить до	<i>Викладача</i>	багато до одного	Багато оплат надходять до одного викладача
<i>Курс</i>	складається з	<i>Модулів уроку</i>	один до багатьох	Один конкретний курс може складатися з декількох модулів уроків
<i>Курс</i>	має	<i>Сертифікат</i>	один до одного	Один курс може мати тільки один сертифікат
<i>Курс</i>	містить	<i>Аналітику</i>	один до одного	Один курс містить тільки одну аналітику, пов'язана з цим курсом
<i>Модуль уроку</i>	містить	<i>Тестування</i>	багато до одного	Модулі уроків містять тільки одне певне тестування
<i>Модуль уроку</i>	включає	<i>Комунікацію</i>	багато до одного	Модулі уроків включають тільки одну комунікацію

<i>Модуль уроку</i>	має	<i>Розклад</i>	багато до одного	Модулі уроків мають тільки один розклад
---------------------	-----	----------------	---------------------	---

На основі цієї таблиці можна побудувати інфологічну модель майбутньої платформи онлайн-школи користуючись мовою ER-діаграм (рис. 1.7). На даному малюнку розглядаються основні та необхідні зв'язки між сутностями для майбутньої бази даних.

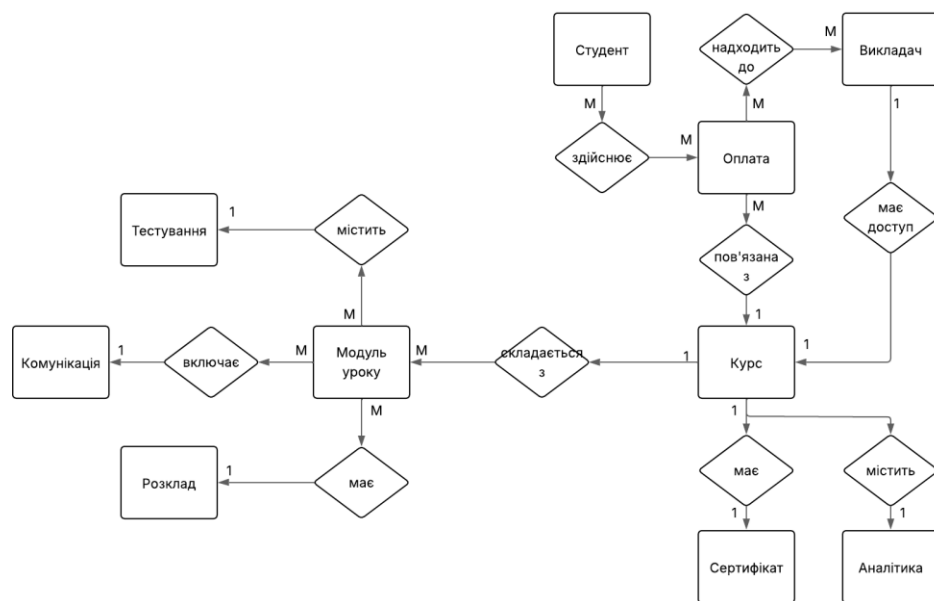


Рис.1.7 Інфологічна модель платформи онлайн-школи

2 МОДЕЛІ ЖИТТЄВОГО ЦИКЛУ РОЗРОБКИ ОНЛАЙН-ПЛАТФОРМИ

2.1 Життєвий цикл

Поняття "життєвий цикл" для інформаційних систем у програмній інженерії означає спосіб застосування до штучних систем програмного забезпечення та визначають зміни, що відбуваються всередині програмного продукту.

Існують стадії від початку існування виробу та його можливого завершення, найбільш вживане поняття як стадії життєвого циклу. Virізняють 3 головні моделі життєвого циклу:

- водоспадна (каскадна) або послідовна модель;
- інкрементально-еволюційна (або ж змішана, гібридна) і ітеративна моделі;
- модель Боема, або ж ще відома як спіральна.

2.1.1 Водоспадна, ітеративна та спіральна моделі

Водоспадна (каскадна) модель вимагає чіткого, послідовного і однократного виконання всіх етапів проєкту з строгим і, заодно, детальним попереднім планування конкретних умов до програмної системи. До позитивних сторін такої моделі відносять:

- незмінність умов, які будуть відбуватися під час розробки життєвого циклу програмного забезпечення;
- чітке розуміння вимог замовника;
- спроможність послідовного позбавлення складнощів, які можуть виникнути при розробці;
- зрозуміле та визначене виконання етапів моделі і легкість її застосування.

Серед недоліків каскадної моделі virізняють:

- на початкових етапах складно чітко сформулювати вимоги та неможливо їх змінити на протязі розробки життєвого циклу;
- недоступний попередній відгук користувача про якість системи;
- проблеми з фінансуванням проєкту, адже виникають труднощі одноразового розподілу коштів, які були виділені для проєктування.

Не зважаючи на ці якості, можна побудувати послідовну модель, орієнтуючись на майбутню інформаційну систему онлайн-школи (рис. 2.1).



Рис.2.1 Послідовна модель ІС онлайн-школи

Звертаючи увагу на ітеративну модель або ж інкрементально-еволюційну (змішану, гібридну), можна виділити розділ життєвого циклу проєкту на послідовні вітки ітерацій. Кожна гілка створює "міні-задачі" , враховуючи кожну фазу циклу у виконанні до створення найменших фрагментів функціональності (порівнюючи проєкт в цілому).

Завданням кожної ітерації є здобуття функціональної версії системи, що виконує працездатність, яка визначається інтегрованою сутністю всіх та даної ітерації. Остання ітерація має вміщувати всю важливу функціональність системи, тобто, завершуючи кожне завдання ітерації, інформаційна система формується інкрементально. Ітеративною називається модель, яка розглядається зі сторони

структури життєвого циклу, а інкрементальною - зі сторони розвитку системи. Аналізувати окремо кожен модель життєвого циклу неможливо. Таку гібридну (змішану), еволюційну модель характеризують просто ітеративною (дивлячись на процес) або/та інкрементальною (дивлячись на функціональну сторону системи).

Змішана модель складає не лише працюючі версії продукту, а й розгортається в дійсних операційних умовах з урахуванням коментарів користувачів для з'ясування змісту та покроковим плануванням подальшої ітерації. "Прозора" інкрементальна модель не розгортається між проміжними збірками (релізами) продукту, тому клієнти отримують тільки результат кінцевої ітерації як основну версію системи, адже ітерації визначаються заздалегідь по плану зростання функціональності.

Завдяки цінності еволюційного підходу на основі планування, знижується неоднозначність кожної ітерації. У висновку чого, можна уникнути ризиків при розробці продукту. Ітеративному розбиттю підлягає не лише життєвий цикл загалом, що, наприклад, містить в собі етапи: визначення вимог, проєктування, розробка і подібне; а й кожен етап може конкретизуватися уточнюючими ітераціями для роз'яснення деталей архітектури декомпозиції проєкту.

Серед основних позитивних сторін ітеративної моделі визначають:

- якнайшвидша реалізація системи, внаслідок чого набагато раніше виставляється продукт на ринок та експлуатація;
- зменшення проблемних місць та ризиків, тобто виявлення суперечностей між моделями, запитами та реалізацією проєкту. Завдяки цьому розробники краще фокусуються на важливих вимогах, динамічно формують та керують завданнями;
- планування високопродуктивного зворотного зв'язку з клієнтом, що призводить до розробки продукту, що дійсно відповідає його запитам.

Серед негативних сторін ітеративної моделі виділяють:

- відсутність фіксованого терміну та бюджету, внаслідок чого виникають проблеми з накладними витратами. Якщо при розробці виникають нові

вимоги і без дослідження повного плану, архітектура системи може страждати, тож для виправлення ситуації можуть бути необхідні додатковий час та ресурси;

- необхідність залучення замовника в процеси розробки ітерацій. Не для всіх клієнтів ці умови співпраці підходять, тому в таких випадках потрібно використовувати водоспадну модель.

На основі цієї інформації можна побудувати ітераційну модель майбутньої системи онлайн-школи (рис. 2.2). Кожна з ітерацій виконує свою функціональність та допомагає отримати кінцевий результат архітектуру системи:

1. Ітерація 0 (встановлення вимог). На цьому етапі розробники займаються плануванням, визначають основні запити користувачів та, насамперед, вимоги до MVP (мінімально життєздатний продукт, англ. Minimumviableproduct);

2. Ітерація 1 (базова функціональність). Цей етап передбачає створення основних функцій продукту, таких як: реєстрація/авторизація, створення курсів, запис студентів на курс, відображення модулів та уроків, інше;

3. Ітерація 2 (додаткові функції). Завдяки цій ітерації розробники додають до базових завдань додаткові: тестування і домашні завдання, побудова розкладу, створюють сторінки для комунікації студентів та викладачів, інше;

4. Ітерація 3(монетизація та аналітика). Для фінального етапу додають платіжні системи, видачу сертифікатів, проведення аналітики роботи студентів та викладачів, тощо.

Остання модель, яка найбільш популярна та поширена в життєвому циклі - спіральна модель. Саме вона стала практично незалежною та має різноманітні сценарії деталізації і прогресу. Вперше спіральну модель сформулював інженер-програміст БарріБоем ще в 1988 році. Найголовнішою властивістю є детальна увага ризикам, які можуть сильно впливати на організування архітектури життєвого циклу.

Така модель ідеально підійде для розробки унікальних (новаторських) систем. На етапі планування проєкту в розробника та замовника може не бути

чіткого бачення фінального продукту або високих шансів в успішному проєктуванні. Тому чудовим рішенням цих проблем є розробка "по частинам" з перспективою змін у вимогах чи відмовою від розвитку системи, тобто завершення не лише після етапу впровадження, але й після етапу аналізу ризиків.

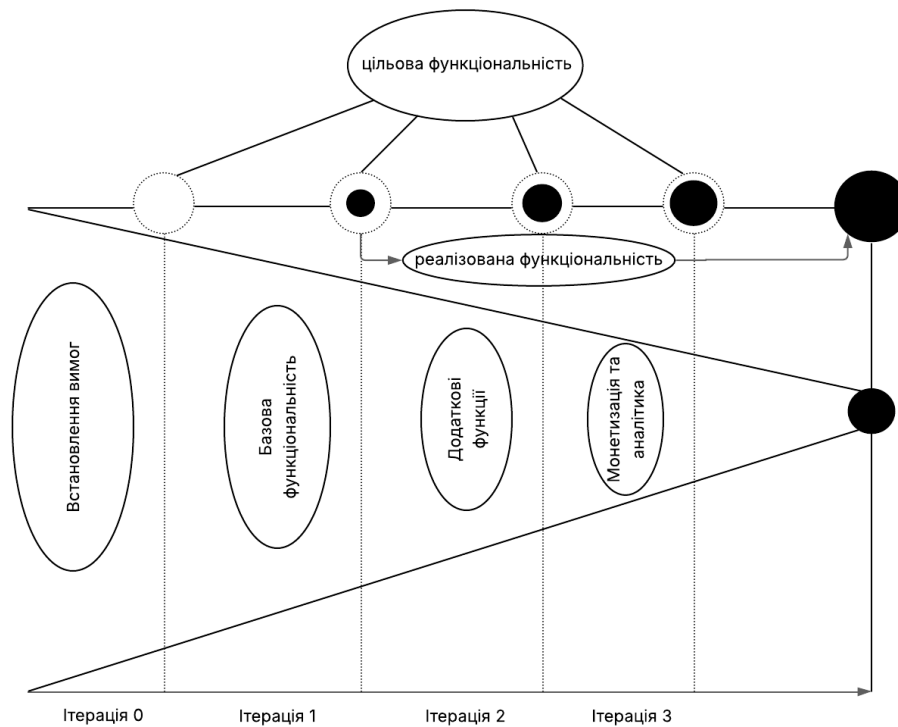


Рис.2.2 Ітеративна модель життєвого циклу ІС при інкрементальному розширенню функціональності

Аналізуючи цю модель Боема, можна виділити великий ряд *переваг*:

- удосконалення процесу розробки завдяки аналізу, який проводиться в кожній ітерації та оцінювати саму організацію розробки для її поліпшення на наступному етапі;
- допускання змін вимог при проєктуванні інформаційної системи;
- зменшення фінансових витрат замовника при завершенні розвитку можливого неперспективного продукту;
- виявлення слабких місць та виправлення помилок по мірі проєктування системи, що забезпечує отримання надійнішої та стійкішої системи.

Недоліків в такій моделі небагато, але серед основних виділяють:

- збільшення невизначеності перспектив проектування в розробника.

Ця негативна сторона впливає з однієї із переваг моделі;

- ускладнення потенційного та тимчасового планування проекту в загалом. Щоб вирішити таку проблему потрібно тимчасово обмежити кожен етап життєвого циклу. Перехід виконується згідно до основного плану навіть тоді, коли вся задумана робота не виконана. Такий план формується завдяки статичних даних, які були отримані в раніше виконуваних проєктах та, безпосередньо, на власному досвіді розробників.

Дану модель можна побудувати і для майбутньої платформи онлайн-школи, де розглядається ризик кожної ітерації та прийняттям рішення: продовжити, змінити чи призупинити проєкт (рис. 2.3).



Рис.2.3 Спиральна модель платформи онлайн-школи

2.2 Основи моделювання UML

Існує величезна чисельність інструментальних засобів, якими користуються для відтворення інформаційної системи від етапу дослідження до відтворення програмного коду. Окремо вирізняють CASE-засоби верхнього рівня (upperCASEtools) і Case-засоби нижнього рівня (lowerCASEtools). CASE (англійською Computer-Aided Software Engineering) - комплекс методів та інструментів програмної інженерії для проєктування програмного забезпечення, що сприяє високій якості програм, простоту в обслуговуванні та відсутність помилок в програмних продуктах.

Використовуючи CASE-засоби верхнього рівня вирізняють основну проблему при їх використанні, а саме їх пристосування під певні проєкти, оскільки вони примушують дотримуватися правила при процесі розробки і не дозволяють організовувати роботу на рівні інших елементів проєкту. Заміною їм можуть стати CASE-засоби нижнього рівня, але вони створюють інші труднощі - ускладнення взаємодії та взаєморозуміння між командами певної організації, що займаються над проєктуванням інших елементів проєкту.

Саме уніфікована мова об'єктно-орієнтованого моделювання (Unified Modeling Language - UML), дозволяє поєднати ці інструментальні засоби. Серед позитивних сторін мови UML можна виділити різні підходи, які піддержують життєвий цикл інформаційної системи та дають можливість відтворити та налаштувати своєрідність діяльності розробників інших елементів проєкту. Головними характеристиками уніфікованої мови моделювання UML є:

- побудова взаємодії розробника (або групи розробників) і замовника інформаційної системи шляхом організації візуальних моделей;
- спеціалізація фундаментальних позначень для конкретної ПО.

Для моделювання та підтримки різноманітних етапів життєвого циклу інформаційної системи об'єктно-орієнтована мова UML має цілу сукупність діаграм, а саме такі визначені діаграми як:

- варіантів використання (use case diagram);
- кооперацій (collaboration diagram);
- станів (statechart diagram);
- компонентів (component diagram);
- класів (class diagram);
- послідовності (sequence diagram);
- діяльності (activity diagram);
- розгортки (deployment diagram).

Ці діаграми і їх назви представляють собою невід'ємний фрагмент графічного відтворення мовою UML. Саме цей процес об'єктно-орієнтоване проєктування неподільно пов'язують із процесом відтворення цих діаграм. Побудовані за таким способом діаграми є самостійними, тому що вони містять всю необхідну інформацію для реалізації проєктування складної системи.

Кожна з представлених діаграм уточнює та деталізує різні уявлення про модель складної системи в рамках мови UML. Наприклад, діаграма варіантів використання представляє собою відому концептуальну модель складної системи, що є початковою для відтворення інших діаграм. Діаграма класів - логічна модель, адже показує статичні зв'язки структурної побудови складної системи. Діаграми послідовностей та кооперації - також різноманітність логічної моделі, що показують динамічні зв'язки роботи складної системи. Діаграми діяльності та станів представляють фізичні компоненти складної системи і тому належать до її фізичної моделі. Окрім графічних аспектів, що призначені для кожної діаграми, на них зображають текстову інформацію, що посилюють зміст фундаментальних елементів.

2.3 Проєктування моделей бази даних

Використовуючи методику об'єктно-орієнтованого моделювання слід пам'ятати про два етапи та їх послідовність у виконанні: об'єктно-орієнтований аналіз та об'єктно-орієнтоване проєктування. Ці два терміни рівнозначні один

одному та, аналізуючи їх зміст, можна детально описати поняття аналіз системних вимог, як встановлення вимог до майбутньої розробки програмної системи, та об'єктний аналіз, як оцінювання ефективності об'єктів ПО).

Якщо говорити про дослідження потреб предметної області, то вона має мати лише ті об'єкти та взаємозв'язки, що підходять до відтворення вимог і вирішення конкретної задачі. Аналізуючи ці вимоги, потрібно прояснити головні потреби клієнта, використовувати інформацію від керівника підприємства та особливу увагу приділяти майбутнім користувачам. А ось при аналізі компонентів та об'єктів, потрібно консультиватися та співпрацювати з фахівцями, які краще знають про ПО. Складність полягає в тому, що правила та процеси використовувати неформально - заборонено.

Методика проектування реляційних баз даних структурними методами, започатковані на ER-діаграмах, та об'єктно-орієнтовані способи, використовуються мовою UML. Тому ER-модель абстрактно легше UML, адже відрізняється тільки термінологією (містить менше термінів та варіантів використання).

2.3.1 Діаграма прецедентів

Діаграма прецедентів, або ж варіантів використання (usecase) використовується для того, щоб подивитися як система буде функціонувати в довколишньому середовищі. Для побудови такої діаграми виділяють такі компоненти:

- актор (actor) - компонент, відображає користувача, який функціонує з конкретною сутністю (рис. 2.4);

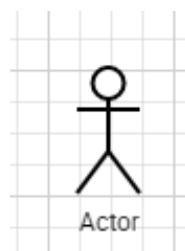


Рис. 2.4 Позначення компонента "Актор"

- прецедент - компонент, який показує дії, що відтворюється самою системою, або ж певними акторами (рис.2.5).

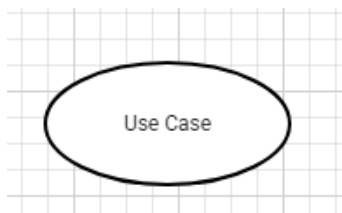


Рис.2.5 Позначення компонента "Прецедент"

Також в такій діаграмі між прецедентами можуть встановлюватися зв'язки:

- узагальнення (generalization) - показує єдність акторів;
- включення (include) - показує контактування кількох варіантів використання, головний з яких постійно функціонально взаємодіє з пов'язаними з ним елементами;
- розширення (extend) показує контактування головного варіанта використання і варіантів, які можуть бути його особливими випадками.

На основі цієї інформації будемо діаграму прецедентів для онлайн-школи (рис. 2.6).

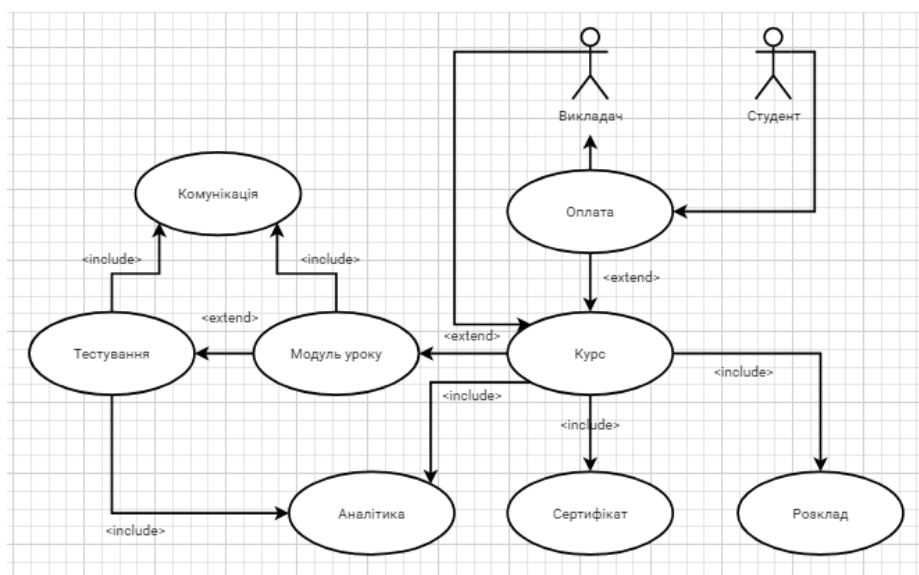


Рис.2.6 Діаграма варіантів використання онлайн-школи

2.3.2 Діаграма діяльності

- діаграма, що моделюється для бізнес-процесів. На ній показано розділення на фрагменти певної діяльності, а саме направленої реалізування дій і вкладення різновидів діяльності, які поєднуються між один одним від виходу одного вузла до входу іншого, за умови зазначення акторів. Такі діаграми можуть змінюватися, якщо будуть виявлені нові обставини в описуванні бізнес-процесами предмета автоматизації під час аналізування та проектування.

На рис. 2.7 представлено діаграму діяльності для інформаційної системи онлайн-школи, а саме вибір студентом курсу, який він хоче вивчати та проведення оплати з умовними переходами.

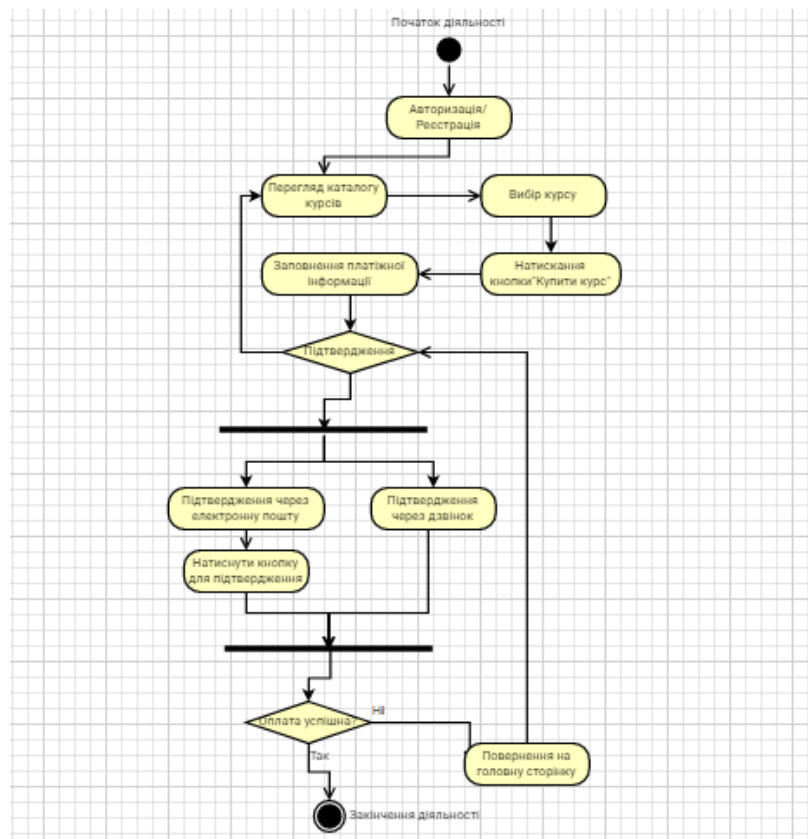


Рис.2.7 Діаграма діяльності основних дій покупки курсу

2.3.3 Діаграма послідовності

Діаграма послідовності, або ж sequencediagram - це діаграма, яка відтворює систематичні за часом взаємодіяння об'єктів. На таких діаграмах відображаються самі об'єкти; вертикальні лінії (англ. lifeline), які змінюються на протязі певного часу, якщо з об'єктом виконують якісь дії; стрілки, які показують маніпуляції, що здійснюються об'єктом. Під час встановлення вимог до ІС, використовують модель системних прецедентів, де зовнішнім та внутрішнім користувачам назначаються певні функції при використанні інформаційної системи. Самі моделі системних прецедентів організовуються завдяки бізнес-моделям, спроектованих на ранній стадії, але також потрібно якомога точніше охарактеризувати прецеденти з поняттями відтворених вже раніше даних і обов'язково притримуватися послідовності їх використання. На рис. 2.8 зображено діаграму послідовності, де показано, що буде відбуватися після сплати курсу зі сторони викладача.

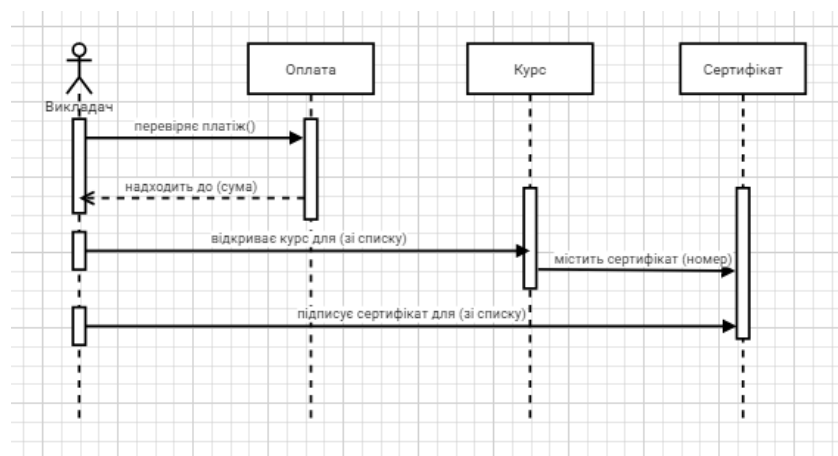


Рис.2.8 Діаграма послідовності взаємодії між об'єктами

2.3.4 Діаграма класів

Після того, як описано системні прецеденти (тобто, як користувачі взаємодіють із системою), створюються *діаграми класів*.

Ці діаграми показують, з чого складається система всередині - її структуру

та архітектуру, використовуючи терміни з об'єктно-орієнтованого програмування. На діаграмі класів можна побачити, які саме об'єкти та підсистеми є у системі, як вони між собою пов'язані, і яка між ними взаємодія.

Клас уUML - це група об'єктів, які мають спільні властивості та поведуться подібним чином, тобто мають однакову структуру та зв'язки з іншими класами. У діаграмі класів показуємо об'єкти, які вже були описані в прецедентах, додаючи до них їхні характеристики та зв'язки між ними.

2.3.5 Діаграма розгортання

На наступному етапі розробки системи додаємо до моделі бази даних і додатків ще й діаграму, яка показує, на якому обладнанні все буде працювати.

Тут використовуються такі поняття з UML:

- компонент - це частина системи, яка виконує певні функції та має свої інтерфейси;
- залежність - зміна одного елемента в системі впливає на інший;
- процесор і пристрій - це вузли, де процесор обробляє дані, а пристрій - ні;
- з'єднання - засоби, які пов'язують між собою процесори та пристрої.

Використовуючи діаграму розгортання можна змодельовати фізичну сторону окремих складових системи по технічним засобам. Основний акцент робиться на тому, щоб відобразити, яке обладнання використовується в інформаційній системі, тобто саме пристрої (вузли), наприклад сервер баз даних, веб-сервер чи сервер додатків.

Коли моделюється складна ІС, її необхідно розділити на окремі частини й розглядати кожну окремо. Для цього існують два підходи:

1. Структурне (або функціональне) розділення - коли система ділиться за її основними функціями;
2. Об'єктно-орієнтоване розділення - коли система складається з окремих частин, які виконують певні завдання та взаємодіють між собою.

Особливість об'єктного підходу - чітке розділення система на незалежні компоненти, кожен із яких виконує свою роль. Для об'єктного підходу часто використовують мову моделювання UML, щоб візуально зобразити структуру системи. А от при функціональному розділенні UML зазвичай не використовується, бо він більше підходить саме для об'єктного дизайну.

На рис. 2.9 зображена приклад діаграми розгортання для інформаційної системи онлайн-школи.

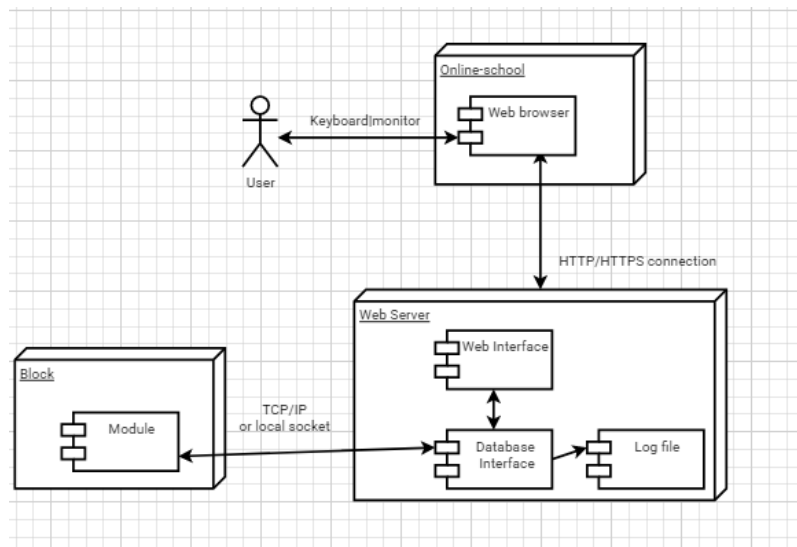


Рис.2.9 Діаграма розгортання для ІС онлайн-школи

3 РЕАЛІЗАЦІЯ ТА ВИКОРИСТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Створення таблиць для бази даних онлайн-школи

Використовуючи всю необхідну інформацію, таблиці, моделі та діаграми, приступаємо до створення бази даних та необхідних компонентів за допомогою декларативної мови SQL. Для реалізації буде використана програма "BDBrowserforSQLite". На рис. 3.1 представлена база даних під назвою "OnlineSchool" та створені в ній таблиці, необхідні для її функціональності.

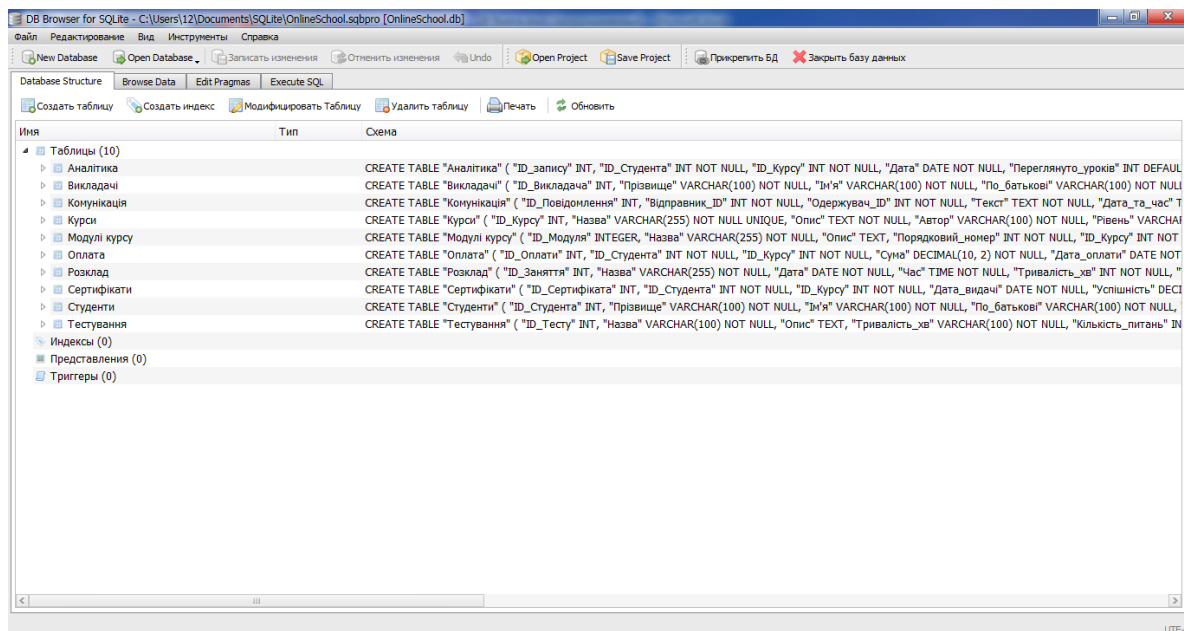


Рис.3.1 Створена база даних для ІС онлайн-школи

Проаналізуємо кожну таблицю, створену для бази даних та проаналізуємо код, за допомогою якого була створена таблиця.

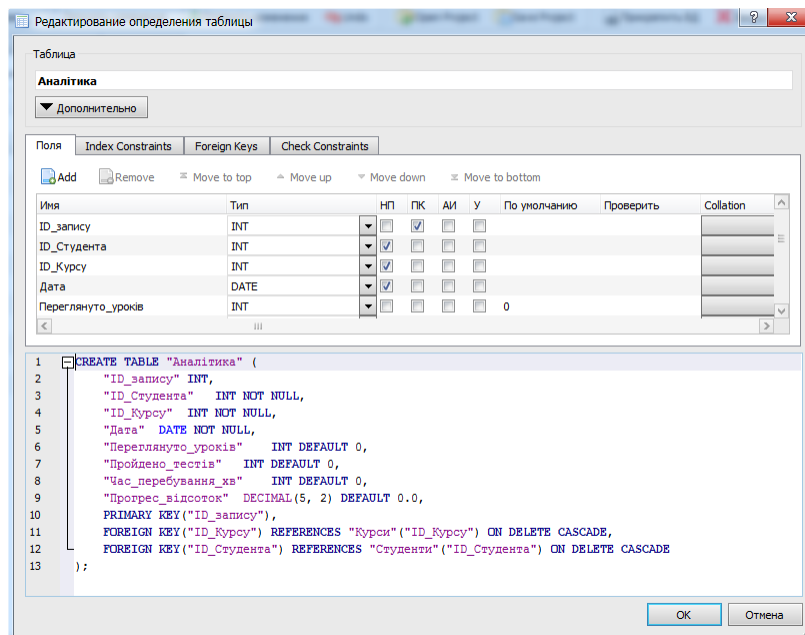


Рис.3.2 Таблица "Аналітика"

Таблиця 3.1

Аналіз написання коду для таблиці "Аналітика"

Назва	Тип	Опис
<i>ID_запису</i>	<i>INT</i>	Унікальний ідентифікатор запису, <i>первинний ключ</i> .
<i>ID_Студента</i>	<i>INT NOT NULL</i>	Ідентифікатор студента, зовнішній ключ до таблиці "Студенти".
<i>ID_Курсу</i>	<i>INT NOT NULL</i>	Ідентифікатор курсу, зовнішній ключ до таблиці "Курси".
<i>Дата</i>	<i>DATE NOT NULL</i>	Дата запису аналітики.
<i>Переглянуто_уроків</i>	<i>INT DEFAULT 0</i>	Кількість переглянутих уроків.
<i>Пройдено_тестів</i>	<i>INT DEFAULT 0</i>	Кількість пройдених тестів.
<i>Час_перебування_хв</i>	<i>INT DEFAULT 0</i>	Час перебування на платформі (в хвилинах).
<i>Прогрес_відсоток</i>	<i>DECIMAL(5,2)</i> <i>DEFAULT 0</i>	Прогрес у відсотках з точністю до 2 знаків після коми.

Також були використані деякі обмеження та зв'язки:

- PRIMARYKEY (*ID_запису*) - забезпечує унікальність кожного запису.
- FOREIGNKEY(*ID_Курсу*) - посиляється на *Курси(ID_Курсу)* з каскадним видаленням, тобто при видаленні курсу, пов'язані аналітичні записи також видаляються;
- FOREIGNKEY(*ID_Студента*) - посиляється на *Студенти(ID_Студента)* з каскадним видаленням.

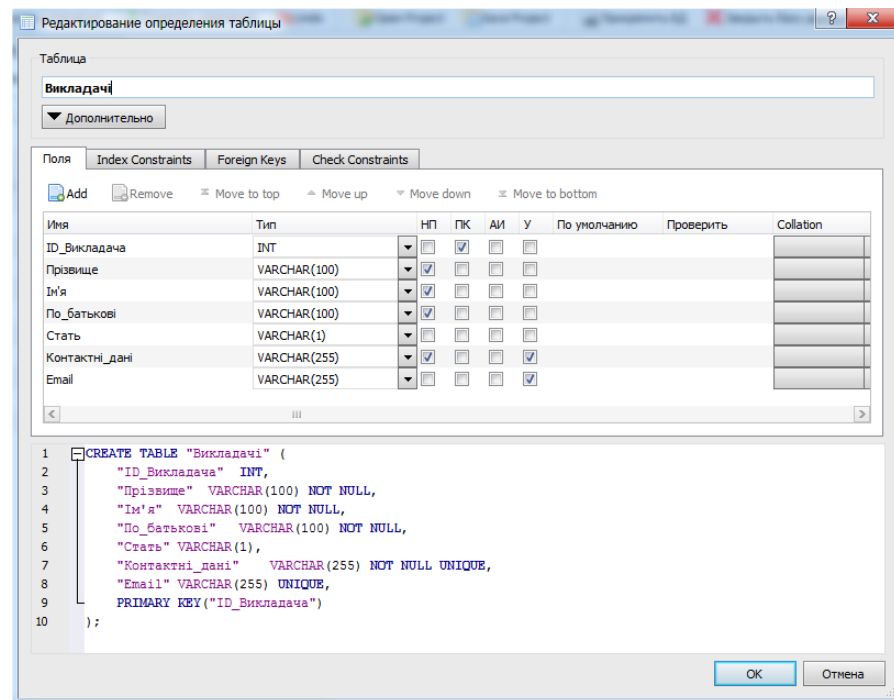


Рис. 3.3 Таблица "Викладачі"

Таблиця 3.2

Аналіз написання коду для таблиці "Викладачі"

Назва	Тип	Опис
<i>ID_Викладача</i>	<i>INT</i>	Унікальний ідентифікатор викладача, <i>первинний ключ</i> .
<i>Прізвище</i>	<i>VARCHAR(100) NOT NULL</i>	Прізвище викладача (обов'язкове поле).
<i>Ім'я</i>	<i>VARCHAR(100) NOT NULL</i>	Ім'я викладача.
<i>По_батькові</i>	<i>VARCHAR(100) NOT NULL</i>	По батькові викладача.

<i>Стать</i>	<i>VARCHAR(1)</i>	Стать (наприклад, "Ч", "Ж").
<i>Контактні_дані</i>	<i>VARCHAR(255) NOT NULL UNIQUE</i>	Унікальні контактні дані.
<i>Email</i>	<i>VARCHAR(255) UNIQUE</i>	Електронна адреса, також має бути унікальною.

Серед обмежень виділяємо такі:

- PRIMARYKEY(*ID_Викладача*) - забезпечує унікальність кожного викладача;
- UNIQUE(*Контактні_дані*) - не допускає дублювання контактної інформації;
- UNIQUE(*Email*) - забороняє використання однієї пошти для кількох викладачів.

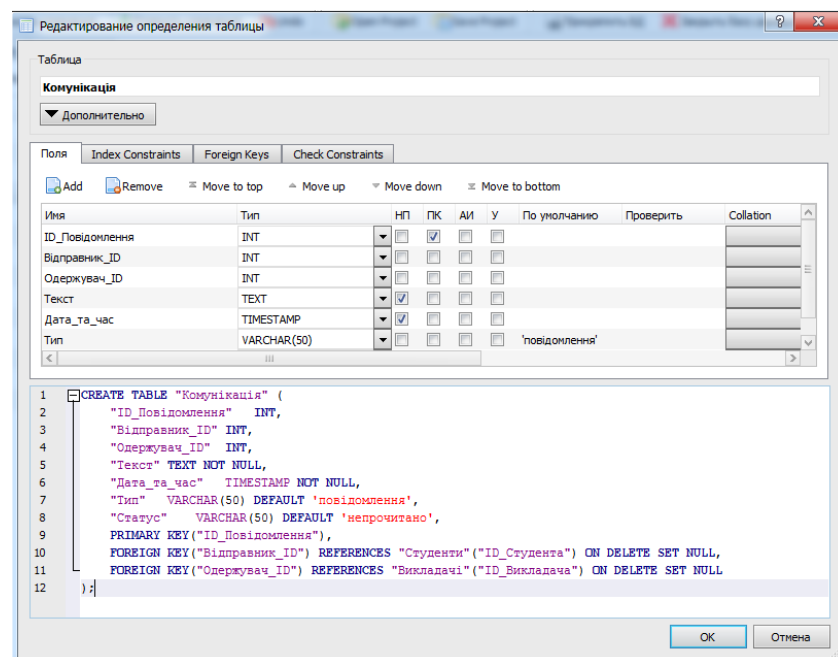


Рис.3.4 Таблица "Комунікація"

Аналіз написання коду для таблиці "Комунікація"

Назва	Тип	Опис
<i>ID_Повідомлення</i>	<i>INT</i>	Унікальний ідентифікатор повідомлення (<i>первинний ключ</i>).
<i>Відправник_ID</i>	<i>INT</i> (можливий <i>NULL</i>)	Ідентифікатор відправника.
<i>Одержувач_ID</i>	<i>INT</i> (можливий <i>NULL</i>)	Ідентифікатор одержувача.
<i>Текст</i>	<i>TEXT NOT NULL</i>	Текст повідомлення.
<i>Дата_та_час</i>	<i>TIMESTAMP NOT NULL</i>	Час та дата відправлення.
<i>Тип</i>	<i>VARCHAR(50)</i>	Тип повідомлення.
<i>Статус</i>	<i>VARCHAR(50)</i>	Статус повідомлення ("надіслано", "прочитано", інше).

Якщо звернути увагу на обмеження та зв'язки, то:

- **PRIMARYKEY**(*ID_Повідомлення*) - забезпечує унікальність;
- **FOREIGNKEY** (*Відправник_ID*) - посилається на *Студенти*(*ID_Студента*) та **ONDELETESETNULL**, тобто при видаленні студента - в полі буде *NULL*, повідомлення не зникне.
- **FOREIGNKEY** (*Одержувач_ID*) - посилається на *Викладачі*(*ID_Викладача*) та **ONDELETESETNULL**, тобто при видаленні викладача - повідомлення залишиться, без вказаного одержувача.

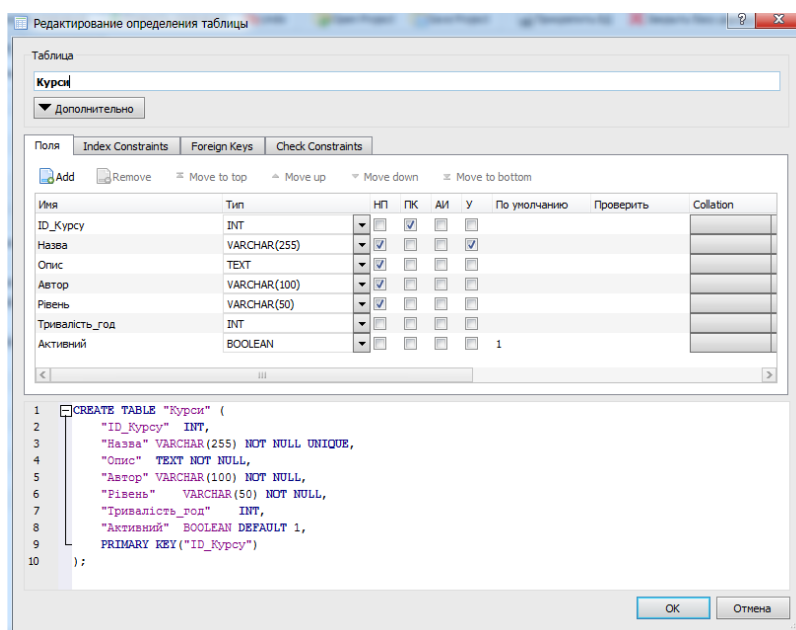


Рис.3.5 Таблиця "Курси"

Аналіз написання коду для таблиці "Курси"

Назва	Тип	Опис
<i>ID_Курсу</i>	<i>INT</i>	Унікальний ідентифікатор курсу, <i>первинний ключ</i> .
<i>Назва</i>	<i>VARCHAR(255) NOT NULL UNIQUE</i>	Назва курсу. Має бути унікальною.
<i>Опис</i>	<i>TEXT NOT NULL</i>	Детальний опис курсу.
<i>Автор</i>	<i>VARCHAR(100) NOT NULL</i>	Ім'я або ПІБ автора курсу.
<i>Рівень</i>	<i>VARCHAR(50) NOT NULL</i>	Рівень складності.
<i>Тривалість_год</i>	<i>INT</i>	Загальна тривалість курсу у годинах.
<i>Активний</i>	<i>BOOLEAN DEFAULT 1</i>	Ознака активності курсу.

Серед обмежень та властивостей можна виділити:

- PRIMARYKEY(*ID_Курсу*): гарантує унікальність курсу;
- UNIQUE(*Назва*): запобігає дублюванню курсів з однаковою назвою;
- DEFAULT(*Активний* = 1): нові курси за замовчуванням вважаються активними.

активними.

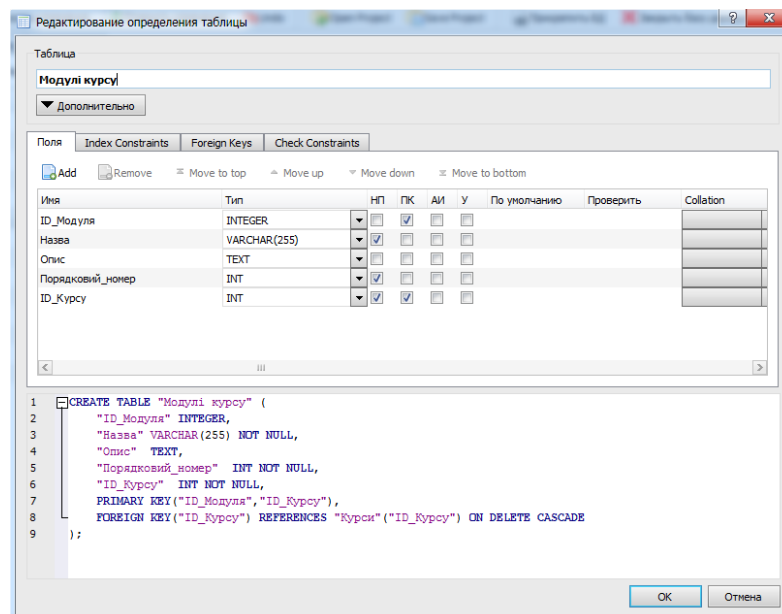


Рис.3.6 Таблиця "Модулі курсу"

Таблиця 3.5

Аналіз написання коду для таблиці "Модулі курсу"

Назва	Тип	Опис
<i>ID_Модуля</i>	<i>INTEGER</i>	Ідентифікатор модуля. <i>Унікальний</i> в межах курсу.
<i>Назва</i>	<i>VARCHAR(255) NOT NULL</i>	Назва модуля.
<i>Опис</i>	<i>TEXT</i>	Опис модуля.
<i>Порядковий_номер</i>	<i>INT NOT NULL</i>	Позиція модуля в межах курсу.
<i>ID_Курсу</i>	<i>INT NOT NULL</i>	Зовнішній ключ - до якого належить модуль.

Якщо виділяти ключі та обмеження в цьому коді, то можна зазначити:

- `PRIMARY_KEY(ID_Модуля, ID_Курсу)` - це *складений ключ*, тобто один і той самий *ID_Модуля* може повторюватися в іншому курсі, але не в межах одного й того ж курсу та гарантує унікальність модуля в рамках курсу;
- `FOREIGNKEY(ID_Курсу) REFERENCESКурси(ID_Курсу) ONDELETECASCADE` - якщо курс буде видалено, то всі пов'язані модулі також будуть автоматично видалені.

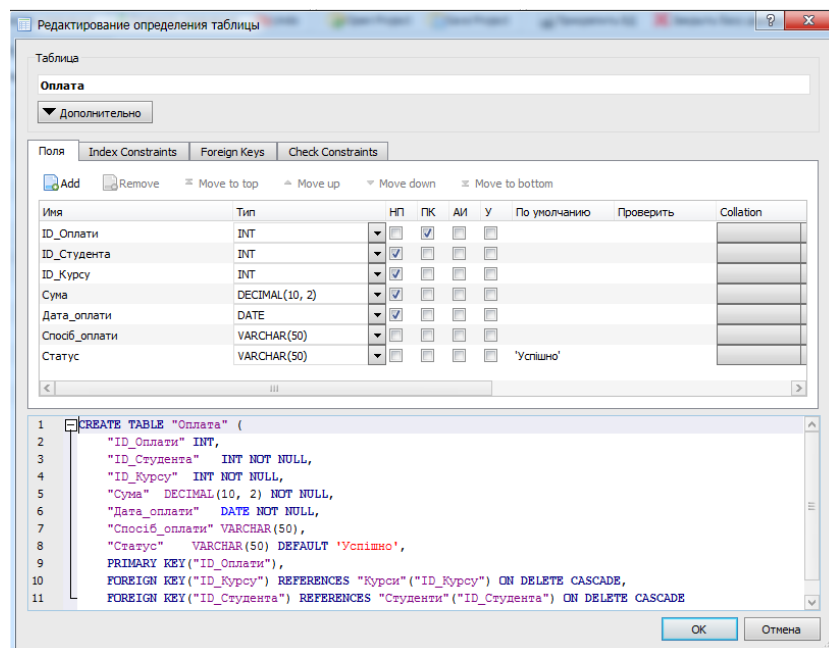


Рис.3.7 Таблица "Оплата"

Таблица 3.6

Аналіз написання коду для таблиці "Оплата"

Назва	Тип	Опис
<i>ID_Оплати</i>	<i>INT</i>	Унікальний ідентифікатор платежу.
<i>ID_Студента</i>	<i>INT NOT NULL</i>	Посилання на студента.
<i>ID_Курсу</i>	<i>INT NOT NULL</i>	Посилання на курс.
<i>Сума</i>	<i>DECIMAL(10, 2) NOT NULL</i>	Сума оплати у грошовому форматі.
<i>Дата_оплати</i>	<i>DATE NOT NULL</i>	Дата транзакції.
<i>Спосіб_оплати</i>	<i>VARCHAR(50)</i>	Наприклад: "Карта", інше.
<i>Статус</i>	<i>VARCHAR(50)</i>	Статус платежу.

Серед зв'язків та обмежень виділяємо:

- PRIMARYKEY(*ID_Оплати*) - гарантує унікальність записів;
- FOREIGNKEY(*ID_Курсу*) - зв'язок з таблицею "*Курси*".

ONDELETECASCADE означає, що якщо курс видалено, пов'язані оплати теж буде видалено;

- FOREIGNKEY(*ID_Студента*) - зв'язок з таблицею "*Студенти*" і також з ON DELETE CASCADE.

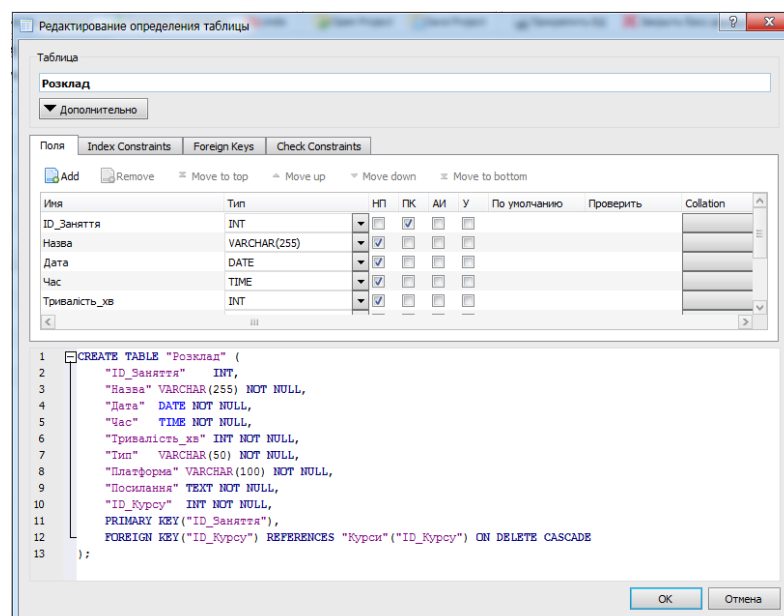


Рис.3.8 Таблиця "Розклад"

Аналіз написання коду для таблиці "Розклад"

Назва	Тип	Опис
<i>ID_заняття</i>	<i>INT</i>	Унікальний ідентифікатор заняття.
<i>Назва</i>	<i>VARCHAR(255) NOT NULL</i>	Назва заняття.
<i>Дата</i>	<i>DATE NOT NULL</i>	Дата проведення заняття.
<i>Час</i>	<i>TIME NOT NULL</i>	Час початку заняття.
<i>Тривалість_хв</i>	<i>INT NOT NULL</i>	Тривалість заняття в хвиликах.
<i>Тип</i>	<i>VARCHAR(50) NOT NULL</i>	Тип заняття.
<i>Платформа</i>	<i>VARCHAR(100) NOT NULL</i>	Назва платформи (наприклад, Zoom, Teams, Google Meet, інше).
<i>Посилання</i>	<i>TEXT NOT NULL</i>	Повне URL-посилання на заняття.
<i>ID_Курсу</i>	<i>INT NOT NULL</i>	Зовнішній ключ: до якого курсу належить заняття.

Обмеження та зв'язки, які присутні в цьому коді:

- PRIMARYKEY(*ID_Заняття*) - забезпечує унікальність заняття;
- FOREIGNKEY(*ID_Курсу*) REFERENCES*Курси*(*ID_Курсу*)

ONDELETECASCADE - якщо курс видалено, то відповідні заняття теж видаляються.

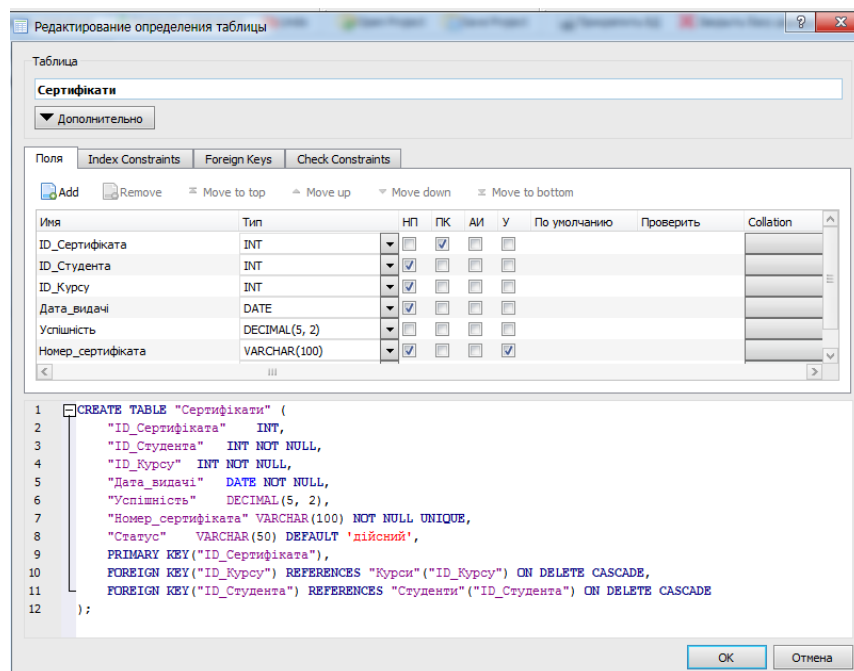


Рис.3.9 Таблица "Сертификаты"

Аналіз написання коду для таблиці "Сертифікати"

Назва	Тип	Опис
<i>ID_Сертифіката</i>	<i>INT</i>	Унікальний ідентифікатор сертифіката
<i>ID_Студента</i>	<i>INT NOT NULL</i>	Зовнішній ключ.
<i>ID_Курсу</i>	<i>INT NOT NULL</i>	Зовнішній ключ.
<i>Дата_видачі</i>	<i>DATE NOT NULL</i>	Дата, коли було видано сертифікат.
<i>Успішність</i>	<i>DECIMAL(5, 2)</i>	Оцінка, яку отримав студент.
<i>Номер_сертифіката</i>	<i>VARCHAR(100) NOT NULL UNIQUE</i>	Унікальний номер сертифіката для перевірки/ідентифікації.
<i>Статус</i>	<i>VARCHAR(50)</i>	Стан сертифіката (наприклад, "дійсний", "анульовано", тощо).

Серед основних моментів можна виділити:

- `PRIMARYKEY(ID_Сертифіката)` - обов'язкова умова для зовнішньої перевірки;
- `"Статус"` з типом `DEFAULT` - що дає можливість перевірити актуальність сертифіката.

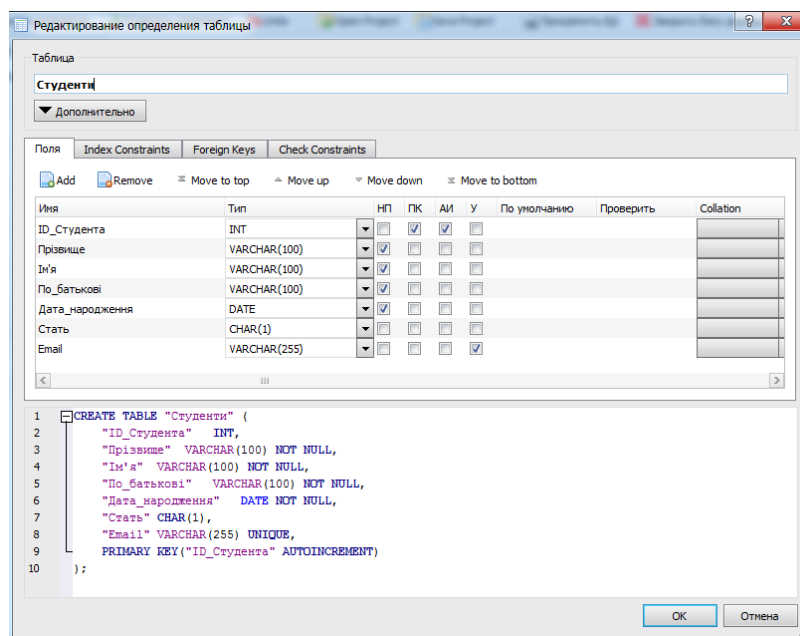


Рис. 3.10 Таблица "Студенти"

Аналіз написання коду для таблиці "Студенти"

Назва	Тип	Опис
<i>ID_Студента</i>	<i>INTEGER</i>	Унікальний ідентифікатор студента.
<i>Прізвище</i>	<i>VARCHAR(100) NOT NULL</i>	Прізвище студента.
<i>Ім'я</i>	<i>VARCHAR(100) NOT NULL</i>	Ім'я студента.
<i>По_батькові</i>	<i>VARCHAR(100) NOT NULL</i>	По батькові.
<i>Дата_народження</i>	<i>DATE NOT NULL</i>	Дата народження.
<i>Стать</i>	<i>CHAR(1)</i>	Стать: "Ч", "Ж", іноді - "І" (інше).
<i>Email</i>	<i>VARCHAR(255) UNIQUE</i>	Унікальна електронна адреса студента.

Серед обмежень та специфікацій можна виділити:

- "Email" з обмеженням UNIQUE - використане для унікальної авторизації та комунікації;
- поля з типом NOTNULLвикористані для заповнення важливої інформації та ключових персональних даних.
- використаний автоінкремент для "ID_Студента" - це дозволить автоматично генерувати значення ID.

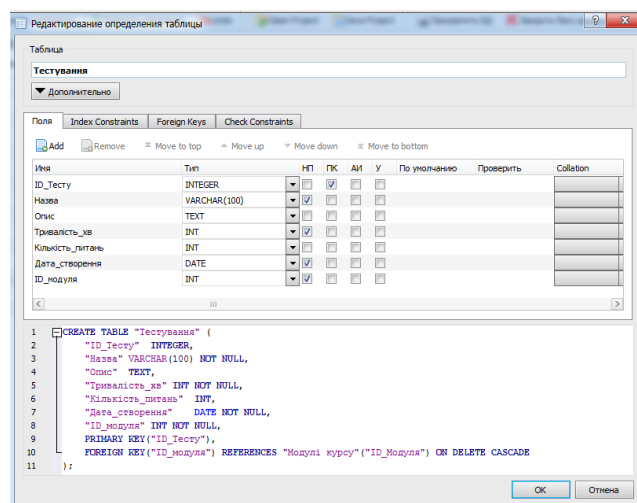


Рис.3.11 Таблица "Тестування"

Таблица 3.10

Аналіз написання коду для таблиці "Тестування"

Назва	Тип	Опис
<i>ID_Тесту</i>	<i>INTEGER</i>	Унікальний ідентифікатор тесту.
<i>Назва</i>	<i>VARCHAR(100) NOT NULL</i>	Назва тесту.
<i>Опис</i>	<i>TEXT</i>	Додаткова інформація про тест.
<i>Тривалість_хв</i>	<i>INT NOT NULL</i>	Час, відведений на виконання тесту в хвиликах.
<i>Кількість_питань</i>	<i>INT</i>	Кількість запитань у тесті.
<i>Дата_створення</i>	<i>DATE NOT NULL</i>	Дата створення тесту.
<i>ID_модуля</i>	<i>INT</i>	Посилання на модуль курсу.

Серед основного зв'язку можна виділити:зовнішній ключ "*ID_модуля*", що пов'язаний з таблицею "Модулі курсу"(*ID_модуля*) та має операцію ONDELETECASCADE означає, що пов'язані тести будуть видалені при видаленні модуля.

3.2 Заповнення інформацією таблиць

Тепер, коли створені потрібні таблиці для бази даних онлайн-школи, потрібно заповнити їх тестовою інформацією для того, щоб перевірити, як вони працюють та за необхідності в подальшому їх вдосконалювати та поновлювати дані.

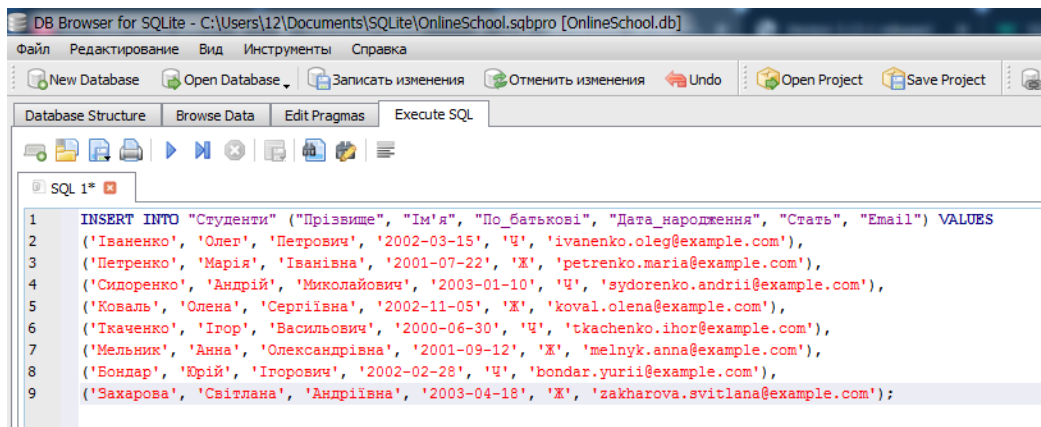


Рис.3.12 Код для заповнення таблиці "Студенти"

ID Студента	Прізвище	Ім'я	По_батькові	Дата_народження	Стать	Email
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1 Іваненко	Олег	Петрович	2002-03-15 Ч		ivanenko.oleg@exam...
2	2 Петренко	Марія	Іванівна	2001-07-22 Ж		petrenko.maria@exa...
3	3 Сидоренко	Андрій	Миколайович	2003-01-10 Ч		sydorenko.andrii@e...
4	4 Коваль	Олена	Сергіївна	2002-11-05 Ж		koval.olena@exampl...
5	5 Ткаченко	Ігор	Васильович	2000-06-30 Ч		tkachenko.igor@exa...
6	6 Мельник	Анна	Олександрівна	2001-09-12 Ж		melnyk.anna@exampl...
7	7 Бондар	Юрій	Ігорович	2002-02-28 Ч		bondar.yurii@examp...
8	8 Захарова	Світлана	Андріївна	2003-04-18 Ж		zakharova.svitlana...

Рис.3.13 Результат заповненої таблиці "Студенти"

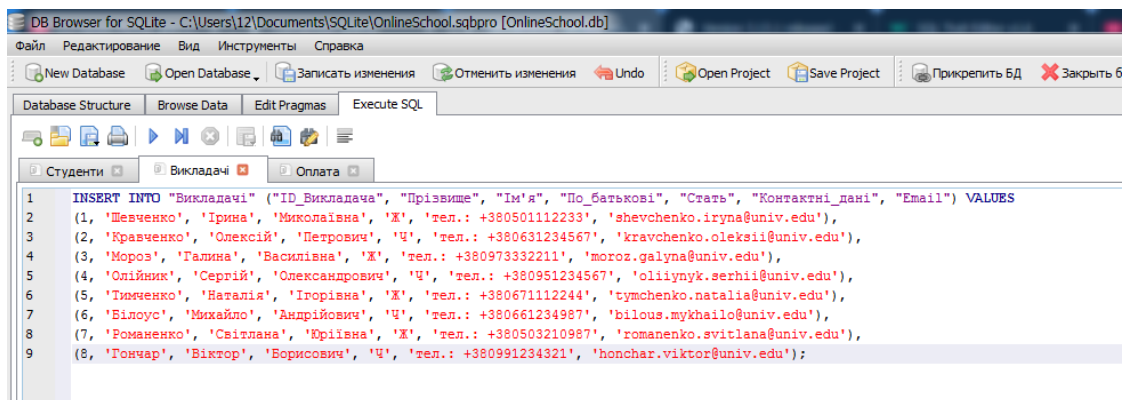


Рис.3.14 Код для заповнення таблиці "Викладачі"

ID_Викладача	Прізвище	Ім'я	По_батькові	Стать	Контактні_дані	Email
1	Шевченко	Ірина	Миколаївна	Ж	тел.: +380501112233	shvchenko.iryana@univ.edu
2	Кравченко	Олексій	Петрович	Ч	тел.: +380631234567	kravchenko.oleksii@univ.edu
3	Мороз	Галина	Василівна	Ж	тел.: +380973332211	moroz.galina@univ.edu
4	Олійник	Сергій	Олександрович	Ч	тел.: +380951234567	oliinyk.serhii@univ.edu
5	Тимченко	Наталія	Ігорівна	Ж	тел.: +380671112244	tymchenko.natalia@univ.edu
6	Білоус	Михайло	Андрійович	Ч	тел.: +380661234987	bilous.mykhailo@univ.edu
7	Романенко	Світлана	Юріївна	Ж	тел.: +380503210987	romanenko.svitlana@univ.edu
8	Гончар	Віктор	Борисович	Ч	тел.: +380991234321	honchar.viktor@univ.edu

Рис.3.15 Результат заповненої таблиці "Викладачі"

```

1 INSERT INTO "Курси" ("ID_Курсу", "Назва", "Опис", "Автор", "Рівень", "Тривалість_год", "Активний") VALUES
2 (101, 'Основи програмування', 'Курс знайомить з базовими принципами програмування на мовах C та Python.', 'Кравченко Олексій', 'Початковий', 40, 1),
3 (102, 'Бази даних', 'Розгляд основ реляційних баз даних, SQL, проектування схеми та запити.', 'Мороз Галина', 'Середній', 36, 1),
4 (103, 'Web-розробка', 'HTML, CSS, JavaScript, створення динамічних вебсайтів та вебдодатків.', 'Шевченко Ірина', 'Середній', 45, 1),
5 (104, 'Алгоритми та структури даних', 'Побудова, аналіз та оптимізація алгоритмів, робота з різними структурами даних.', 'Олійник Сергій', 'Високий', 50, 1),
6 (105, 'Машинне навчання', 'Вступ до машинного навчання, моделі, регресія, класифікація, sklearn, pandas.', 'Тимченко Наталія', 'Високий', 60, 1),
7 (106, 'Кібербезпека', 'Захист даних, принципи безпеки в IT-системах, тестування на проникнення.', 'Білоус Михайло', 'Середній', 42, 0);

```

Рис.3.16 Код для заповнення таблиці "Курси"

ID_Курсу	Назва	Опис	Автор	Рівень	Тривалість_год	Активний
101	Основи програмування	Курс знайомить з базовими принципами програмування на мовах C та Python.	Кравченко Олексій	Початковий	40	1
102	Бази даних	Розгляд основ реляційних баз даних, SQL, проектування схеми та запити.	Мороз Галина	Середній	36	1
103	Web-розробка	HTML, CSS, JavaScript, створення динамічних вебсайтів та вебдодатків.	Шевченко Ірина	Середній	45	1
104	Алгоритми та структури даних	Побудова, аналіз та оптимізація алгоритмів, робота з різними структурами даних.	Олійник Сергій	Високий	50	1
105	Машинне навчання	Вступ до машинного навчання, моделі, регресія, класифікація, sklearn, pandas.	Тимченко Наталія	Високий	60	1
106	Кібербезпека	Захист даних, принципи безпеки в IT-системах, тестування на проникнення.	Білоус Михайло	Середній	42	0

Рис.3.17 Результат заповненої таблиці "Курси"

```

1  -- Модулі для курсу "Основні програмування"
2  INSERT INTO "Модулі курсу" VALUES (1, 'Вступ до програмування', 'Основні поняття: алгоритми, мови програмування.', 1, 101);
3  INSERT INTO "Модулі курсу" VALUES (2, 'Змінні та типи даних', 'Типи даних, оголошення та ініціалізація змінних.', 2, 101);
4  INSERT INTO "Модулі курсу" VALUES (3, 'Умовні конструкції', 'Умовні оператори if-else, логічні вирази.', 3, 101);
5
6  -- Модулі для курсу "Бази даних"
7  INSERT INTO "Модулі курсу" VALUES (1, 'Вступ до баз даних', 'Що таке СУБД, поняття таблиці, атрибута, запису.', 1, 102);
8  INSERT INTO "Модулі курсу" VALUES (2, 'Мова SQL', 'Основні команди SELECT, INSERT, UPDATE, DELETE.', 2, 102);
9  INSERT INTO "Модулі курсу" VALUES (3, 'Зв'язки між таблицями', 'FOREIGN KEY, JOIN-запити.', 3, 102);
10
11 -- Модулі для курсу "Web-розробка"
12 INSERT INTO "Модулі курсу" VALUES (1, 'HTML: структура сторінки', 'Теги, структура документа, списки, таблиці.', 1, 103);
13 INSERT INTO "Модулі курсу" VALUES (2, 'CSS: стилізація', 'Селектори, кольори, шрифти, макети.', 2, 103);
14 INSERT INTO "Модулі курсу" VALUES (3, 'JavaScript: логіка на стороні клієнта', 'Основні мови, події, DOM.', 3, 103);

```

Рис.3.18 Код для заповнення таблиці "Модулі курсу"

ID Модуля	Назва	Опис	Порядковий номер	ID Курсу
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1 Вступ до програмування	Основні поняття: алгоритми, мови програмування.	1	101
2	2 Змінні та типи даних	Типи даних, оголошення та ініціалізація змінних.	2	101
3	3 Умовні конструкції	Умовні оператори if-else, логічні вирази.	3	101
4	1 Вступ до баз даних	Що таке СУБД, поняття таблиці, атрибута, запису.	1	102
5	2 Мова SQL	Основні команди SELECT, INSERT, UPDATE, DELETE.	2	102
6	3 Зв'язки між таблицями	FOREIGN KEY, JOIN-запити.	3	102
7	1 HTML: структура сторінки	Теги, структура документа, списки, таблиці.	1	103
8	2 CSS: стилізація	Селектори, кольори, шрифти, макети.	2	103
9	3 JavaScript: логіка на стороні ...	Основні мови, події, DOM.	3	103

Рис.3.19 Результат заповненої таблиці "Модулі курсу"

```

1  INSERT INTO "Розклад"
2  ("ID_Заняття", "Назва", "Дата", "Час", "Тривалість_хв", "Тип", "Платформа", "Посилання", "ID_Курсу")
3  VALUES
4  -- Основні програмування (ID_Курсу: 101)
5  (1, 'Вступ до програмування', '2025-06-01', '10:00', 90, 'Лекція', 'Zoom', 'https://zoom.us/j/101lecture', 101),
6  (2, 'Практика: перша програма', '2025-06-03', '10:00', 90, 'Практика', 'Google Meet', 'https://meet.google.com/101practice', 101),
7
8  -- Бази даних (ID_Курсу: 102)
9  (3, 'Основні SQL', '2025-06-05', '11:00', 90, 'Лекція', 'Zoom', 'https://zoom.us/j/102lecture', 102),
10 (4, 'Практика: створення таблиць', '2025-06-07', '11:00', 90, 'Практика', 'Google Meet', 'https://meet.google.com/102practice', 102),
11
12 -- Web-розробка (ID_Курсу: 103)
13 (5, 'HTML та CSS', '2025-06-09', '12:00', 90, 'Лекція', 'Zoom', 'https://zoom.us/j/103lecture', 103),
14 (6, 'Практика: створення сторінки', '2025-06-11', '12:00', 90, 'Практика', 'Google Meet', 'https://meet.google.com/103practice', 103),
15
16 -- Алгоритми та структури даних (ID_Курсу: 104)
17 (7, 'Базові алгоритми', '2025-06-13', '13:00', 90, 'Лекція', 'Zoom', 'https://zoom.us/j/104lecture', 104),
18 (8, 'Практика: сортування', '2025-06-15', '13:00', 90, 'Практика', 'Google Meet', 'https://meet.google.com/104practice', 104),
19
20 -- Машинне навчання (ID_Курсу: 105)
21 (9, 'Вступ до ML', '2025-06-17', '14:00', 90, 'Лекція', 'Zoom', 'https://zoom.us/j/105lecture', 105),
22 (10, 'Практика: класифікація', '2025-06-19', '14:00', 90, 'Практика', 'Google Meet', 'https://meet.google.com/105practice', 105);

```

Рис.3.20 Код для заповнення таблиці "Розклад"

DB Browser for SQLite - C:\Users\12\Documents\SQLite\OnlineSchool.sqbp [OnlineSchool.db]

Файл Редактирование Вид Инструменты Справка

New Database Open Database Записать изменения Отменить изменения Undo Open Project Save Project Прикрепить БД Закрывать базу данных

Database Structure Browse Data Edit Pragma Execute SQL

Таблица: Розклад

ID_Заняття	Назва	Дата	Час	Тривалість_хв	Тип	Платформа	Посилання	ID_Курсу
1	1 Вступ до програмування	2025-06-01	10:00	90	Лекція	Zoom	https://zoom.us/j/101lecture	101
2	2 Практика: перша програма	2025-06-03	10:00	90	Практика	Google Meet	https://meet.google.com/101practice	101
3	3 Основи SQL	2025-06-05	11:00	90	Лекція	Zoom	https://zoom.us/j/102lecture	102
4	4 Практика: створення таблиць	2025-06-07	11:00	90	Практика	Google Meet	https://meet.google.com/102practice	102
5	5 HTML та CSS	2025-06-09	12:00	90	Лекція	Zoom	https://zoom.us/j/103lecture	103
6	6 Практика: створення сторінки	2025-06-11	12:00	90	Практика	Google Meet	https://meet.google.com/103practice	103
7	7 Базові алгоритми	2025-06-13	13:00	90	Лекція	Zoom	https://zoom.us/j/104lecture	104
8	8 Практика: сортування	2025-06-15	13:00	90	Практика	Google Meet	https://meet.google.com/104practice	104
9	9 Вступ до ML	2025-06-17	14:00	90	Лекція	Zoom	https://zoom.us/j/105lecture	105
10	10 Практика: класифікація	2025-06-19	14:00	90	Практика	Google Meet	https://meet.google.com/105practice	105

Рис.3.21 Результат заповненої таблиці "Розклад"

DB Browser for SQLite - C:\Users\12\Documents\SQLite\OnlineSchool.sqbp [OnlineSchool.db]

Файл Редактирование Вид Инструменты Справка

New Database Open Database Записать изменения Отменить изменения Undo Open Project Save Project Прикрепить БД Закрывать базу данных

Database Structure Browse Data Edit Pragma Execute SQL

Студенти* Курси* Модулі курсу* Розклад* Тестування* Комунікація* Оплата* Сертифікати* Аналіти*

```

1 INSERT INTO "Тестування" ("ID_Тесту", "Назва", "Опис", "Тривалість_хв", "Кількість_питань", "Дата_створення", "ID_модуля")
2 VALUES
3 (1, 'Тест по вступу до програмування', 'Перевірка знань основних понять алгоритмів і мов програмування.', 30, 10, '2025-05-17', 1),
4 (2, 'Тест по змінним та типам даних', 'Тест на знання типів даних, оголошення та ініціалізацію змінних.', 25, 8, '2025-05-17', 2),
5 (3, 'Тест по умовним конструкціям', 'Перевірка знань умовних операторів if-else, логічних виразів.', 20, 7, '2025-05-17', 3),
6 (4, 'Тест по вступу до баз даних', 'Перевірка знань про СУБД, таблиці, атрибути та записи.', 30, 10, '2025-05-17', 4),
7 (5, 'Тест по мові SQL', 'Тестування знань основних SQL команд SELECT, INSERT, UPDATE, DELETE.', 35, 12, '2025-05-17', 5),
8 (6, 'Тест по зв'язкам між таблицями', 'Перевірка знань про FOREIGN KEY, JOIN-запити.', 30, 10, '2025-05-17', 6),
9 (7, 'Тест по HTML: структура сторінки', 'Перевірка знань тегів, структури документа, списків, таблиць.', 30, 10, '2025-05-17', 7),
10 (8, 'Тест по CSS: стилізація', 'Тестування знань щодо селекторів, кольорів, шрифтів, макетів.', 30, 10, '2025-05-17', 8),
11 (9, 'Тест по JavaScript: логіка на стороні клієнта', 'Перевірка знань основ мови JavaScript, подій, DOM.', 40, 12, '2025-05-17', 9);

```

Рис.3.22 Код для заповнення таблиці "Тестування"

DB Browser for SQLite - C:\Users\12\Documents\SQLite\OnlineSchool.sqbp [OnlineSchool.db]

Файл Редактирование Вид Инструменты Справка

New Database Open Database Записать изменения Отменить изменения Undo Open Project Save Project Прикрепить БД Закрывать базу данных

Database Structure Browse Data Edit Pragma Execute SQL

Таблица: Тестування

ID_Тесту	Назва	Опис	Тривалість_хв	Кількість_питань	Дата_створення	ID_модуля
1	1 Тест по вступу до програмування	Перевірка знань основних понять алгоритмів і мов програмування.	30	10	2025-05-17	1
2	2 Тест по змінним та типам даних	Тест на знання типів даних, оголошення та ініціалізацію змінних.	25	8	2025-05-17	2
3	3 Тест по умовним конструкціям	Перевірка знань умовних операторів if-else, логічних виразів.	20	7	2025-05-17	3
4	4 Тест по вступу до баз даних	Перевірка знань про СУБД, таблиці, атрибути та записи.	30	10	2025-05-17	4
5	5 Тест по мові SQL	Тестування знань основних SQL команд SELECT, INSERT, UPDATE, DELETE.	35	12	2025-05-17	5
6	6 Тест по зв'язкам між таблицями	Перевірка знань про FOREIGN KEY, JOIN-запити.	30	10	2025-05-17	6
7	7 Тест по HTML: структура сторінки	Перевірка знань тегів, структури документа, списків, таблиць.	30	10	2025-05-17	7
8	8 Тест по CSS: стилізація	Тестування знань щодо селекторів, кольорів, шрифтів, макетів.	30	10	2025-05-17	8
9	9 Тест по JavaScript: логіка на ...	Перевірка знань основ мови JavaScript, подій, DOM.	40	12	2025-05-17	9

Рис.3.23 Результат заповненої таблиці "Тестування"

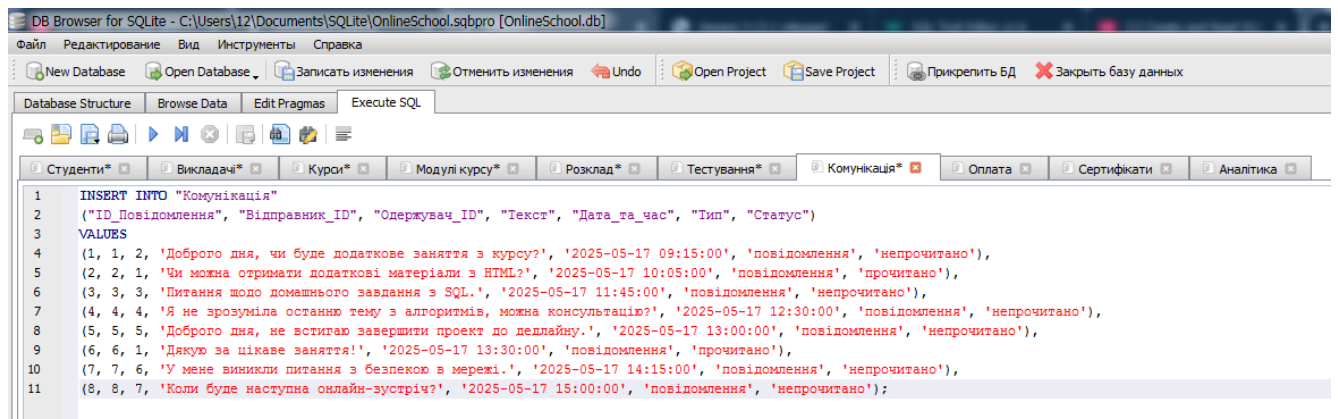


Рис.3.24 Код для заповнення таблиці "Комунікація"

ID_Повідомлення	Відправник_ID	Одержувач_ID	Текст	Дата_та_час	Тип	Статус
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1	2	Доброго дня, чи буде додаткове заняття з курсу?	2025-05-17 09:15:00	повідомлення	непрочитано
2	2	1	Чи можна отримати додаткові матеріали з HTML?	2025-05-17 10:05:00	повідомлення	прочитано
3	3	3	Питання щодо домашнього завдання з SQL.	2025-05-17 11:45:00	повідомлення	непрочитано
4	4	4	Я не зрозуміла останню тему з алгоритмів, можна консультацію?	2025-05-17 12:30:00	повідомлення	непрочитано
5	5	5	Доброго дня, не встигаю завершити проект до дедлайну.	2025-05-17 13:00:00	повідомлення	непрочитано
6	6	1	Дякую за цікаве заняття!	2025-05-17 13:30:00	повідомлення	прочитано
7	7	6	У мене виникли питання з безпекою в мережі.	2025-05-17 14:15:00	повідомлення	непрочитано
8	8	7	Коли буде наступна онлайн-зустріч?	2025-05-17 15:00:00	повідомлення	непрочитано

Рис. 3.25 Результат заповненої таблиці "Комунікація"

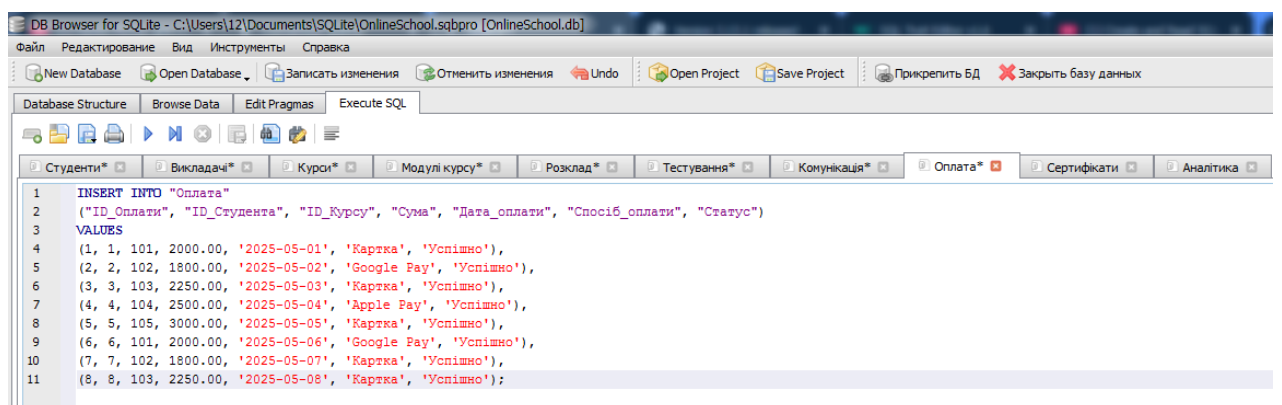


Рис.3.26 Код для заповнення таблиці "Оплата"

DB Browser for SQLite - C:\Users\12\Documents\SQLite\OnlineSchool.sq3pro [OnlineSchool.db]

Файл Редактирование Вид Инструменты Справка

New Database Open Database Записать изменения Отменить изменения Undo Open Project Save Project

Database Structure Browse Data Edit Pragmas Execute SQL

Таблица: Оплата

Filter in any column

ID_Оплаты	ID_Студента	ID_Курсу	Сума	Дата_оплаты	Способ_оплаты	Статус
Фильтр	Фильтр	Фильтр	Фил...	Фильтр	Фильтр	Фильтр
1	1	1	101 2000	2025-05-01	Картка	Успішно
2	2	2	102 1800	2025-05-02	Google Pay	Успішно
3	3	3	103 2250	2025-05-03	Картка	Успішно
4	4	4	104 2500	2025-05-04	Apple Pay	Успішно
5	5	5	105 3000	2025-05-05	Картка	Успішно
6	6	6	101 2000	2025-05-06	Google Pay	Успішно
7	7	7	102 1800	2025-05-07	Картка	Успішно
8	8	8	103 2250	2025-05-08	Картка	Успішно

Рис.3.27 Результат заповненої таблиці "Оплата"

```

1  INSERT INTO "Сертификаты"
2  ("ID_Сертификата", "ID_Студента", "ID_Курсу", "Дата_выдачи", "Успешность", "Номер_сертификата", "Статус")
3  VALUES
4  (1, 1, 101, '2025-06-01', 95.00, 'CERT-001', 'дійсний'),
5  (2, 2, 102, '2025-06-02', 85.00, 'CERT-002', 'дійсний'),
6  (3, 3, 103, '2025-06-03', 92.00, 'CERT-003', 'дійсний'),
7  (4, 4, 104, '2025-06-04', 88.00, 'CERT-004', 'дійсний'),
8  (5, 5, 105, '2025-06-05', 90.00, 'CERT-005', 'дійсний'),
9  (6, 6, 101, '2025-06-06', 93.00, 'CERT-006', 'дійсний'),
10 (7, 7, 102, '2025-06-07', 80.00, 'CERT-007', 'дійсний'),
11 (8, 8, 103, '2025-06-08', 87.00, 'CERT-008', 'дійсний');

```

Рис.3.28 Код для заповнення таблиці "Сертификаты"

DB Browser for SQLite - C:\Users\12\Documents\SQLite\OnlineSchool.sq3proj [OnlineSchool.db]

Файл Редактирование Вид Инструменты Справка

New Database Open Database Записать изменения Отменить изменения Undo Open Project Save Project

Database Structure Browse Data Edit Pragas Execute SQL

Таблица: Сертификаты

ID Сертификата	ID Студента	ID Курсу	Дата выдачі	Успішність	Номер сертифіката	Статус
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1	1	101 2025-06-01	95	CERT-001	дійсний
2	2	2	102 2025-06-02	85	CERT-002	дійсний
3	3	3	103 2025-06-03	92	CERT-003	дійсний
4	4	4	104 2025-06-04	88	CERT-004	дійсний
5	5	5	105 2025-06-05	90	CERT-005	дійсний
6	6	6	101 2025-06-06	93	CERT-006	дійсний
7	7	7	102 2025-06-07	80	CERT-007	дійсний
8	8	8	103 2025-06-08	87	CERT-008	дійсний

Рис.3.29 Результат заповненої таблиці "Сертификаты"

```

1 INSERT INTO "Аналітика"
2 ("ID_запису", "ID_Студента", "ID_Курсу", "Дата", "Переглянуто_уроків", "Пройдено_тестів", "Час_перебування_хв", "Прогрес_відсоток")
3 VALUES
4 (1, 1, 101, '2025-06-01', 5, 3, 120, 80.00),
5 (2, 1, 102, '2025-06-01', 4, 2, 100, 75.00),
6 (3, 2, 103, '2025-06-02', 7, 5, 150, 85.00),
7 (4, 2, 104, '2025-06-02', 6, 4, 130, 80.00),
8 (5, 3, 105, '2025-06-03', 8, 6, 180, 90.00),
9 (6, 3, 101, '2025-06-03', 6, 4, 140, 85.00),
10 (7, 4, 102, '2025-06-04', 5, 3, 120, 78.00),
11 (8, 4, 103, '2025-06-04', 4, 3, 110, 70.00),
12 (9, 5, 104, '2025-06-05', 7, 5, 150, 88.00),
13 (10, 5, 105, '2025-06-05', 6, 4, 140, 83.00),
14 (11, 6, 101, '2025-06-06', 5, 2, 100, 70.00),
15 (12, 6, 102, '2025-06-06', 6, 3, 120, 80.00),
16 (13, 7, 103, '2025-06-07', 8, 5, 160, 90.00),
17 (14, 7, 104, '2025-06-07', 7, 4, 140, 85.00),
18 (15, 8, 105, '2025-06-08', 6, 4, 130, 80.00),
19 (16, 8, 101, '2025-06-08', 5, 3, 110, 75.00);

```

Рис.3.30 Код для заповнення таблиці "Аналітика"

DB Browser for SQLite - C:\Users\12\Documents\SQLite\OnlineSchool.sqbp [OnlineSchool.db]

Файл Редактирование Вид Инструменты Справка

New Database Open Database Записать изменения Отменить изменения Undo Open Project Save Project Прикрепить БД Закреть базу данных

Database Structure Browse Data Edit Pragma Execute SQL

Таблица: Аналітика Filter in any column

ID_запису	ID_Студента	ID_Курсу	Дата	Переглянуто_уроків	Пройдено_тестів	Час_перебування_хв	Прогрес_відсоток	
Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	
1	1	1	101	2025-06-01	5	3	120	80
2	2	1	102	2025-06-01	4	2	100	75
3	3	2	103	2025-06-02	7	5	150	85
4	4	2	104	2025-06-02	6	4	130	80
5	5	3	105	2025-06-03	8	6	180	90
6	6	3	101	2025-06-03	6	4	140	85
7	7	4	102	2025-06-04	5	3	120	78
8	8	4	103	2025-06-04	4	3	110	70
9	9	5	104	2025-06-05	7	5	150	88
10	10	5	105	2025-06-05	6	4	140	83
11	11	6	101	2025-06-06	5	2	100	70
12	12	6	102	2025-06-06	6	3	120	80
13	13	7	103	2025-06-07	8	5	160	90
14	14	7	104	2025-06-07	7	4	140	85
15	15	8	105	2025-06-08	6	4	130	80
16	16	8	101	2025-06-08	5	3	110	75

Рис.3.31- Результат заповненої таблиці "Аналітика"

ВИСНОВКИ

Завдяки сучасним технологіям можна навчатися в будь-якому куточку світу і в будь-який час. В майбутньому нас очікують ще більше нових відкриттів, які, з однією сторони, будуть покращувати якість людського життя, а з протилежної сторони - де-не-де погіршувати. Не зважаючи на це, онлайн-школи будуть набирати ще більше нових студентів та їх популярність може поставити під питання, щодо продовження роботи звичайних шкіл.

Тож, роблячи висновки з усієї отриманої інформації, можна сказати, що ефективність баз даних залежить від коректного вибору моделі даних, системи керування базами даних (СКБД), а також від покращення процесів збереження та обробки інформації. Саме реляційні СКБД поки що залишаються найпопулярнішими та найпоширенішими, однак не слід забувати і про нереляційні підходи, які краще використовувати в роботі з великими обсягами неструктурованих даних.

У дипломній роботі було розроблено інформаційну систему для організації навчального процесу в онлайн-школі. У результаті дослідження:

- проведено аналіз вимог до систем дистанційного навчання;
- визначено та побудовано основні функціональні модулі та діаграми;
- спроектовано та реалізовано головні компоненти дослідженої бази даних.

Тому, досліджена ІС демонструє цінність і може значно підвищити ефективність організації дистанційного навчання. Сучасні вимоги потребують дієвих цифрових рішень, які покращать автоматизацію основних процесів. Результати роботи підтверджують актуальність теми та доцільність застосування інформаційних технологій у сфері освіти. Подальший розвиток системи може передбачити інтеграцію з хмарними сервісами, системами штучного інтелекту та адаптивного навчання для ще більш індивідуалізованого підходу до кожного учня.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Режим доступу: https://eprints.kname.edu.ua/20889/1/Gritsunov_2.pdf
2. Інформація - Вікіпедія. Режим доступу:
<https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%8F>
3. Теорія інформаційних процесів і систем :: Державний університет інформаційно-комунікаційних технологій. Режим доступу:
https://duikt.edu.ua/uploads/l_1141_19815844.pdf
4. Режим доступу: https://www.svpu-profi.lg.ua/pdf/library/osn_baz_danih2011.pdf
5. Проектування інформаційних систем :: Державний університет інформаційно-комунікаційних технологій. Режим доступу:
https://duikt.edu.ua/uploads/l_144_42481385.pdf
6. Режим доступу:
https://repo.btu.kharkov.ua/bitstream/123456789/55880/1/MV_BD_163_24.pdf
7. What is an Information System? Режим доступу:
<https://www.mastersindatascience.org/learning/what-is-an-information-system/>
8. Едгар Кодд — Вікіпедія. Режим доступу:
<https://uk.wikipedia.org/wiki/%D0%95%D0%B4%D0%B3%D0%B0%D1%80%D0%9A%D0%BE%D0%B4%D0%B4>
9. Лекція 3. Інфологічна модель даних “Сутність-зв’язок” План. Режим доступу: <https://studfile.net/preview/7363846/>
10. What is Spiral Model in Software Engineering? - GeeksforGeeks. Режим доступу: <https://www.geeksforgeeks.org/software-engineering-spiral-model/>

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ)

2

МЕТА ТА ЗАВДАННЯ

Мета роботи – розробка інформаційної системи, що забезпечує ефективну організацію, управління та моніторинг навчального процесу в онлайн-школі.

Об'єкт дослідження – процес організації та управління навчальним процесом в онлайн освітньому середовищі.

Предмет дослідження – методи та програмні засоби створення інформаційної системи для автоматизації організації навчального процесу в онлайн школі.

Для досягнення мети, у бакалаврській роботі успішно виконано наступні завдання:

1. Аналіз інформаційної системи, поняття бази даних та СКБД.
2. Побудовані основні моделі та діаграми для створення майбутньої бази даних.
3. Створено базу даних та таблиць, які зберігаються основну інформацію об'єктів ПО.

АНАЛІЗ ПОНЯТТЯ ІНФОРМАЦІЙНА СИСТЕМА ТА БАЗА ДАНИХ

3

Інформаційна система - це комплекс технічних, програмних та організаційних засобів, які накопичують, зберігають, коригують, здійснюють пошук, оброблюють та видають інформацію.

В інформаційних системах, **база даних (БД)** - це сукупність структурованих даних, які показують стан об'єктів конкретної предметної області та зв'язків між ними.



Рис.1 – Основні компоненти ІС

Об'єктами предметної області платформи онлайн-школи будуть:

- студенти,
- викладачі,
- курси,
- модулі уроків,
- тестування,
- розклад,
- оплата,
- комунікація,
- сертифікати та аналітика.

Інфологічна модель - це відображення інформації про предметну область на логічному рівні абстрагування та опис об'єктів, їх взаємозв'язків та властивостей.

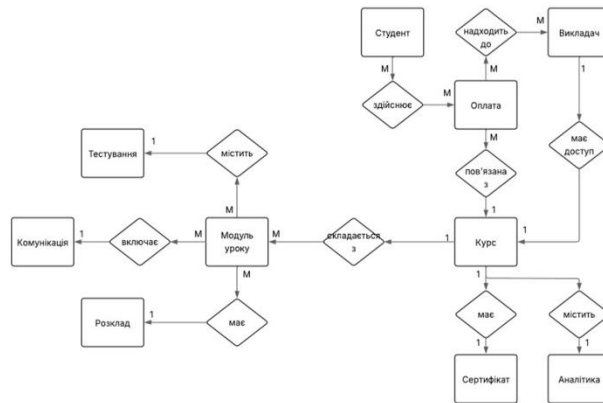


Рис.2 Інфологічна модель ПО

Метою такого моделювання є забезпечити природний спосіб відображення та збору інформації для людини, щоб відтворити в створюваній базі даних.

При побудованні інфологічної моделі можна користуватися мовою ER-діаграм (Entity Relationship, або ж сутність-зв'язок).

МОДЕЛЮВАННЯ МОВОЮ UML

Діаграма прецедентів, або ж варіантів використання (use case) використовується для того, щоб подивитися як система буде функціонувати в довколишньому середовищі.

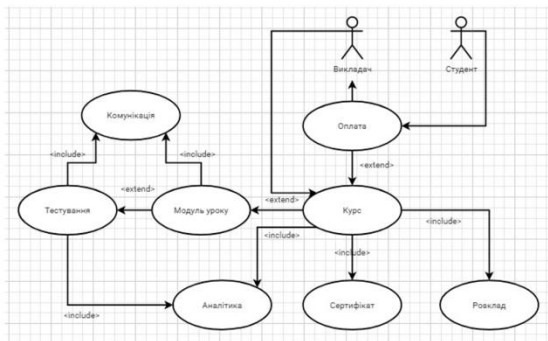


Рис. 6 Діаграма варіантів використання

Діаграма послідовності, або ж sequence diagram - це діаграма, яка відтворює систематичні за часом взаємодіяння об'єктів. На таких діаграмах відображаються самі об'єкти; вертикальні лінії (англ. *lifeline*), які змінюються на протязі певного часу, якщо з об'єктом виконують якісь дії; стрілки, які показують маніпуляції, що здійснюються об'єктом.

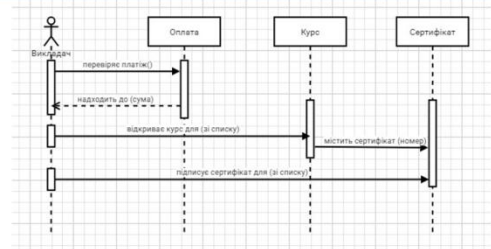


Рис. 7 Діаграма послідовності

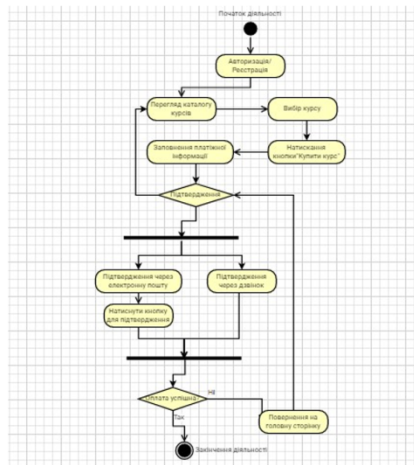


Рис. 8 Діаграма діяльності

Діаграма діяльності, або ж activity diagram - діаграма, що моделюється для бізнес-процесів. На ній показано розділення на фрагменти певної діяльності, а саме направлено реалізування дій і вкладення різновидів діяльності, які поєднуються між одним від виходу одного вузла до входу іншого, за умови зазначення акторів.

Використовуючи **діаграму розгортання** можна змоделювати фізичну сторону окремих складових системи по технічним засобам. Основний акцент робиться на тому, щоб відобразити, яке обладнання використовується в інформаційній системі, тобто саме пристрої (вузли), наприклад сервер баз даних, веб-сервер чи сервер додатків.

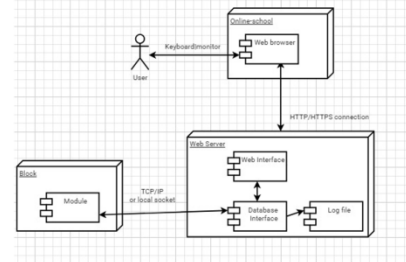


Рис. 9 Діаграма розгортання

Діаграма класів показує, з чого складається система всередині - її структуру та архітектуру, використовуючи терміни з об'єктно-орієнтованого програмування. На діаграмі класів можна побачити, які саме об'єкти та підсистеми є у системі, як вони між собою пов'язані, і яка між ними взаємодія.

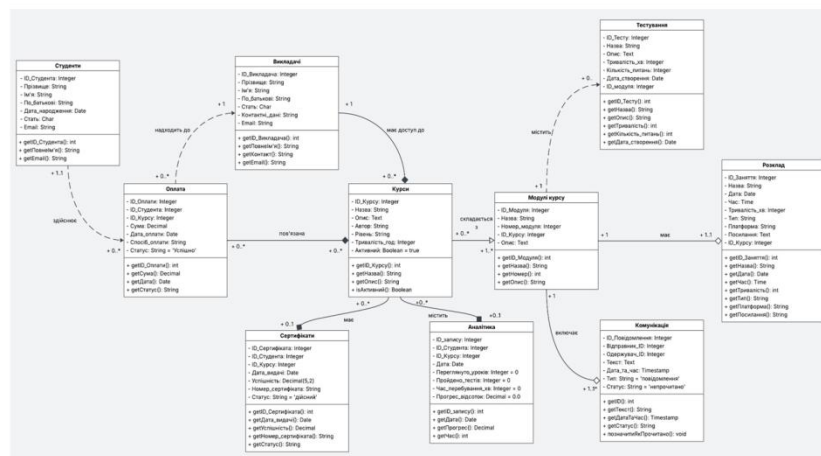


Рис. 10 Діаграма класів

Маніпулювання даними відбуваються спеціальними мовами, якими клієнти описують запити до СКДБ - це **мови маніпулювання даними**, або, як їх ще називають, **мови запитів**. Також, до складу моделі даних, відносять ще одну мову, яка описує структуру баз даних - мова визначення даних.

У реляційній моделі відомою і часто використовуваною є **SQL** (від англійського **Structured Query Language** - структурована мова запитів), яка в містить в собі засоби мови визначення та мови маніпулювання даними.

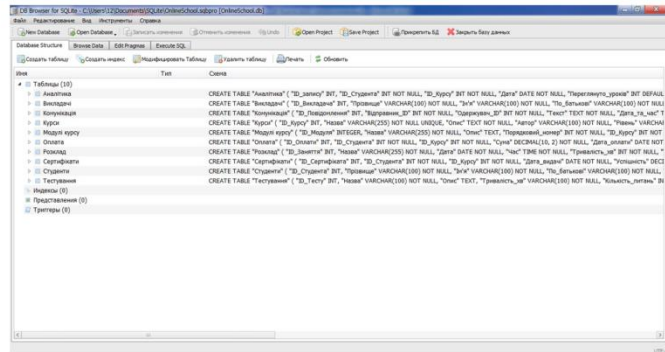


Рис. 11 Створена база даних

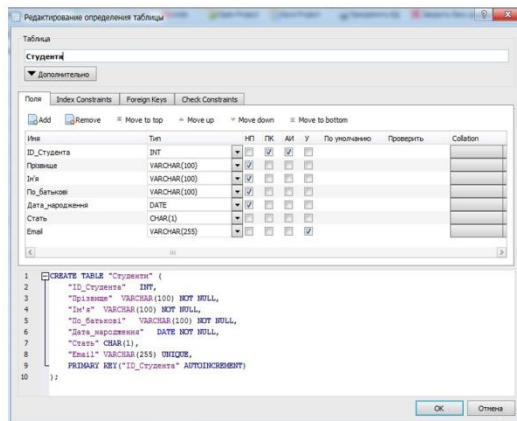


Рис. 12 Код для створення таблиці "Студенти"

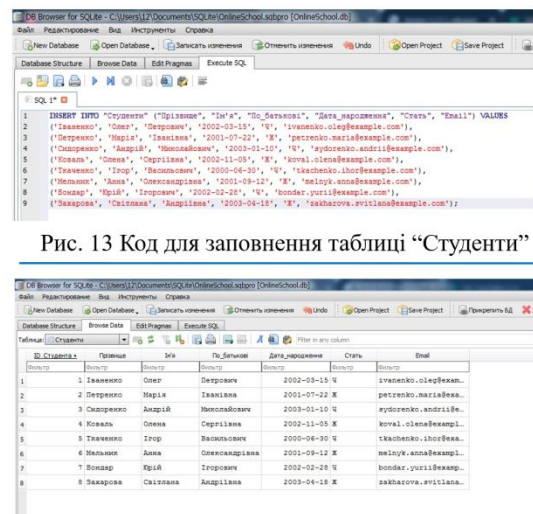


Рис. 13 Код для заповнення таблиці "Студенти"

Рис. 14 Результат заповненої таблиці "Студенти"