

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет _____ Магістерської та
_____ аспірантської підготовки

Кафедра інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Моделювання та розробка підсистеми «Викладач» інформаційної системи «Навчальний процес університету»»

Виконала студентка 2 року групи МК-61
спеціальності 122 Комп'ютерні науки

Тичінский Артур Валерійович

Керівник к.фіз.-мат.н., доц.

Козловська Валентина Петрівна

Консультант _____

Рецензент к.т.н., доц.

Гнатовська Ганна Арнольдівна

АНОТАЦІЯ

Магістерська робота на тему «Моделювання та розробка підсистеми «Викладач» інформаційної системи «Навчальний процес університету»» обумовлена задачею автоматизації основних робіт, які забезпечують навчальний процес у вищому навчальному закладі. Це визначає актуальність даної роботи.

Наукова новизна роботи полягає в тому, що у даній роботі розглядається весь навчальний процес повністю, з урахуванням взаємодії різних об'єктів та суб'єктів цього процесу. Навчальний процес університету має багато складових частин та багато суб'єктів, що забезпечують його роботу. Дана робота розглядає складову частину навчального процесу, яка відноситься до роботи викладача закладу вищої освіти.

Метою і задачею дослідження є вивчення методів та моделей автоматизації робіт, що забезпечують проведення навчального процесу у закладі вищої освіти.

Об'єктом дослідження є навчальний процес у закладі вищої освіти.

Предметом дослідження є засоби автоматизації даного процесу.

Вихідні дані. Використовуються дані, необхідні для проведення навчального процесу у вищій: списки викладачів закладу вищої освіти, академічних груп, навчальних планів, навчальних дисциплін, аудиторного фонду закладу вищої освіти, тощо.

Результати даної роботи можуть використовуватись для автоматизації роботи викладача при складанні робочої програми навчальної дисципліни та проведенні навчального процесу у будь-якому закладі вищої освіти, що використовує кредитно-модульну систему навчання.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, КОНТЕКСТНА ДІАГРАМА IDEF0, КОНТЕКСТНА ДІАГРАМА DFD, ER-ДІАГРАМА, РЕЛЯЦІЙНА БАЗА ДАНИХ, СЕРВЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, АВТОМАТИЗОВАНЕ РОБОЧЕ МІСЦЕ.

Обсяг роботи 61 сторінка. Вона містить 46 рисунків та 15 посилань.

SUMMARY

Master's work on the topic “Simulation and Development of the ‘Teacher’ Subsystem for the Information System of ‘Educational Process at the University’” is due to the task of automating the basic work that provides the educational process at the university. This determines the relevance of this work.

The scientific novelty of the work lies in the fact that in the present work the whole educational process is considered in full, taking into account the interaction of various objects and subjects of this process. The educational process of the university has many components and many subjects ensuring its work. This work considers an integral part of the educational process, relates to the work of a teacher at a higher education institution.

The purpose and objective of the study is to study the methods and models of work automation that ensure the conduct of the educational process in a higher education institution.

The object of study is the educational process in institutions of higher education.

The subject of research is the automation of this process.

Initial data. The data required for the educational process in higher education are used: lists of teachers of higher education institutions, academic groups, curricula, academic disciplines, the class fund of higher education institutions, and the like.

The results of this work can be used to automate the work of the teacher in drawing up the work program of the academic discipline and conducting the educational process in any institution of higher education, uses a credit-modular system of training.

Keywords: INFORMATION SYSTEM, IDEF0 CONTEXT DIAGRAM, DFD CONTEXT DIAGRAM, ER-DIAGRAM, RELATIONAL DATABASE, SERVER SOFTWARE, AUTOMATED WORKPLACE.

The amount of work is 61 pages. It contains 46 figures and 15 references.

ЗМІСТ

| | |
|--|--|
| ЗМІСТ | 6 |
| СПИСОК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ..... | 7 |
| ВСТУП | 8 |
| 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ | 9 |
| 1.1 Об'єкти та суб'єкти навчального процесу університету | 9 |
| 2 МЕТОДОЛОГІЯ ФУНКЦІОНАЛЬНОГО МОДЕЛЮВАННЯ (IDEF0) | 11 |
| 2.1 Розробка контекстної діаграми IDEF0 | 11 |
| 2.1 Декомпозиція контекстної діаграми IDEF0 | 12 |
| 3 МЕТОДОЛОГІЯ МОДЕЛЮВАННЯ ПОТОКІВ ДАНИХ (DFD) | 14 |
| 4 СИСТЕМНИЙ АНАЛІЗ І КОНЦЕПТУАЛЬНЕ ПРОЕКТУВАННЯ БАЗИ ДАНИХ ІНФОРМАЦІЙНОЇ СИСТЕМИ..... | 16 |
| 4.1 Етапи проектування бази даних | 16 |
| 4.2 Концептуальне проектування бази даних | 17 |
| 4.2.1 Визначення основних типів сутностей та типів зв'язків | 17 |
| 4.2.2 Визначення атрибутів і зв'язування їх з типами сутностей і зв'язків | 27 |
| 4.2.3 Визначення атрибутів, що є потенційними і первинними ключами .. | 28 |
| 4.2.4 Визначення відповідності концептуальної моделі транзакціям користувачів..... | 28 |
| 5 ВИБІР СУБД ТА СЕРЕДОВИЩА РОЗРОБКИ ЗАСТОСУВАННЯ..... | 30 |
| 5.1 Вибір СУБД | 30 |
| 5.2 Вибір середовища розробки застосування та мови програмування | 30 |
| 6 СЕРВЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ..... | 32 |
| 7 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ | 36 |
| 7.1 Загальні відомості | 36 |
| 7.2 Функціональне призначення | 36 |
| 7.3 Керівництво програміста | 36 |
| 7.4 Посібник користувача | 40 |
| ВИСНОВКИ..... | 59 |
| ПЕРЕЛІК ПОСИЛАНЬ | 61 |
| Д О Д А Т К И..... | Ошибка! Закладка не определена. |
| ДОДАТОК А ДІАГРАМА БД..... | Ошибка! Закладка не определена. |
| ДОДАТОК Б ФІЗИЧНА СХЕМА БД ПІДСИСТЕМИ «ВИКЛАДАЧ» | Ошибка! Закладка не определена. |
| ДОДАТОК В СЕРВЕРНЕ ПЗ ПІДСИСТЕМИ «ВИКЛАДАЧ» ІС | Ошибка! Закладка не определена. |

ДОДАТОК Г ВИХІДНИЙ КОД КЛІЄНТСЬКОГО ЗАСТОСУВАННЯ

.....**Ошибка! Закладка не определена.**

СПИСОК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення

| | |
|---------|--|
| АРМ | – Автоматизоване робоче місце |
| БД | – База даних |
| ЗВО | – Заклад вищої освіти |
| ІС | – Інформаційна система |
| МОН | – Міністерство освіти та науки |
| ОНП | – Освітньо-наукова програма підготовки фахівців |
| ОПП | – Освітньо-професійна програма підготовки фахівців |
| СУБД | – Система управління базами даних |
| ASP.NET | – Платформа розробки веб-додатків, до складу якої входить: веб-сервіси, програмна інфраструктура, модель програмування |
| DFD | – Діаграми потоків даних |
| DLR | – Динамічна виконуюча середа |
| ER | – Модель «сутність – зв'язок» |
| IDEF0 | – Методологія функціонального моделювання і графічна нотація, призначена для формалізації і опису бізнес-процесів. |
| MVC | – Модель-Представлення-Контролер |
| T-SQL | – Процедурне розширення мови SQL |

ВСТУП

В Одеському державному екологічному університеті на протязі останніх років при виконанні дипломних проектів та магістерських робіт розроблялись інформаційні системи, що описують різні аспекти навчального процесу в університеті. Прикладом подібних інформаційних систем, або прикладних баз даних, можуть слугувати такі системи: «Електронний деканат», «Тестувальна система», «Інтегральні відомості», «Навантаження викладача», «Розклад занять в університеті».

Всі ці системи розроблялись як окремі незалежні системи баз даних. Подібний підхід є помилковим, оскільки у перерахованих базах даних міститься дуже багато спільної інформації, а саме: перелік підрозділів університету, список викладацького складу університету, перелік навчальних дисциплін, навчальні плани факультетів, тощо. Для можливості зручної роботи користувачів з перерахованими системами потрібно об'єднати їх у єдину інформаційну систему (ІС) «Навчальний процес університету».

При проектуванні вказаної єдиної інформаційної системи виявляється, що вона повинна містити багато складових частин, також виявляється декілька груп користувачів, які будуть працювати з цією системою. Тому після виявлення основних змістовних блоків діаграми декомпозиції IDEF0 та основних груп користувачів потрібно розділити процес проектування ІС на моделювання та проектування декількох підсистем у складі єдиної ІС. Всі підсистеми повинні мати спільні таблиці-довідники, що містять основні дані по навчальному процесу університету.

Задачею даної магістерської роботи є виявлення складових частин інформаційної системи «Навчальний процес університету», основних груп користувачів. Головною задачею магістерської роботи є моделювання та проектування підсистеми «Викладач» у складі ІС «Навчальний процес університету», та розробка застосування «АРМ викладача» у якості додатку до підсистеми «Викладач».

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Головною задачею вищого навчального закладу є надання студентам можливості отримання вищої освіти рівня бакалавр або магістр. Навчальний процес є основним у вищі, і всі інші види діяльності не можливі без існування цього процесу. При створенні інформаційної системи університету у першу чергу потрібно включити до неї саме навчальний процес.

При проектуванні ІС «Навчальний процес університету» необхідно розглянути всі підсистеми, які беруть участь у даному процесі. Після розгляду навчального процесу взагалі та виявлення всіх можливих учасників цього процесу, складових частин процесу, всіх груп користувачів, можна розділити ІС «Навчальний процес університету» на складові підсистеми та виконати моделювання та проектування кожної окремої підсистеми.

1.1 Об'єкти та суб'єкти навчального процесу університету

Виявити складові навчального процесу у вищому навчального закладі простіше, якщо розглянути цей процес у зворотному порядку – від кінцевої мети процесу до початку, через всі проміжні системи та механізми.

Метою існування вищих навчальних закладів є надання студентам можливості отримання вищого рівня освіти. Таким чином, основним суб'єктом навчального процесу є студент, який має бажання отримати цю освіту. Можливість здобуття освіти надається студенту за допомогою спілкування з викладачем, який є другим основним суб'єктом навчального процесу.

Освіту студент набуває за допомогою аудиторних занять та самостійної роботи. Якість отриманої освіти перевіряється за допомогою різних контролюючих заходів.

Тематика аудиторних занять, вид практичних занять, кількість і вид контролюючих заходів визначаються в робочій програмі з дисципліни, яку складає провідний викладач. Робоча програма з дисципліни розробляється на основі навчального плану, за яким навчаються студенти відповідної спеціальності, а також нормативних документів ЗВО.

Для можливості проведення аудиторних занять навчальний відділ ЗВО складає розклад занять на кожний поточний семестр. Основою для складання розкладу занять є навчальні плани за спеціальностями, контингент студентів, та заплановане аудиторне навантаження викладачів.

Навчальні плани за спеціальностями є складовою частиною освітньо-професійних програм (ОПП) та освітньо-наукових програм (ОНП) зі спеціальності, що розробляються уповноваженими комісіями. Навчальні плани ОПП та ОНП є основою робочих навчальних планів, які розробляються деканатами факультетів на наступний навчальний рік, та затверджуються методичним відділом університету.

Навчальні плани містять перелік дисциплін, які повинні вивчати студенти, що навчаються за даним навчальним планом. Методичний відділ університету складає перелік всіх дисциплін, що будуть вивчатись у наступному навчальному році, та розподіляє ці дисципліни між кафедрами університету.

На кафедрах за кожною дисципліною закріплюється провідний викладач, який відповідає за результати навчання студентів з даної дисципліни, розробку робочої програми та іншої методичної літератури. Провідний викладач читає лекції, якщо вони передбачені навчальним планом з дисципліни, проводить консультації. Також провідний викладач визначає контролюючі заходи, які оцінюють знання та вміння, засвоєні студентами при вивченні даної дисципліни. Також провідний викладач сам або за допомогою асистентів проводить практичні заняття та контролюючі заходи з дисципліни. На кафедрі кожен навчальний рік відбувається розподіл навчального навантаження між викладачами. Навчальне навантаження викладача складається з аудиторних занять, керівництва курсовим та дипломним проектуванням студентів, проведення та перевірки результатів контролюючих заходів.

Частина навчального навантаження викладача – аудиторне навантаження – є одною з складових частин для складання розкладу занять у ЗВО. Іншими складовими частинами є робочі навчальні плани, контингент студентів, аудиторний фонд ЗВО, вимоги викладачів до власного розкладу занять та нормативи ЗВО до якісного розкладу занять студентів.

2 МЕТОДОЛОГІЯ ФУНКЦІОНАЛЬНОГО МОДЕЛЮВАННЯ (IDEF0)

2.1 Розробка контекстної діаграми IDEF0

IDEF0 – це діаграма, розташована на вершині деревовидної структури діаграм, що представляє собою саме загальний опис системи та її взаємодію з зовнішнім середовищем. Контекстна діаграма складається з одного блоку, що описує функцію верхнього рівня, її входи, виходи, управління, і механізми, разом з формулюваннями мети моделі і точки зору, з якої будується модель [1 – 3].

Головною функцією системи, що моделюється, є забезпечення навчального процесу в університеті.

Список даних: ОПП, ОНП, нормативні документи МОН та ЗВО, студент, викладач, навчальний план, робоча програма з дисципліни, розклад занять, контролюючі заходи, успішність студента.

Список функцій: розробка навчальних планів за спеціальностями, розробка робочих програм з дисциплін, розподіл занять та іншого навчального навантаження між викладачами, складання розкладу занять, проведення аудиторних занять, проведення контролюючих заходів поточного контролю в семестрі, проведення заходів заключного контролю – іспит або залік, отримання оцінки за успішність студента з дисципліни

Мета створення інформаційної системи: Організувати проведення навчального процесу в університеті.

Перелік запитань:

- Які навчальні плани повинні бути у наступному навчальному році?
- Які академічні групи навчаються за яким навчальним планом?
- Які лекції читаються для потоку академічних груп?
- Які викладачі проводять заняття по дисциплінах навчального плану?
- В яких аудиторіях повинні проводитись конкретні заняття академічних груп?
- Які заходи поточного контролю передбачені з дисципліни?
- Які заходи підсумкового контролю передбачені з дисципліни?
- З яких складових частин складається оцінка студента за вивчення дисципліни?
- Які оцінки отримали студенти академічної групи при проведенні контролюючого заходу з дисципліни?

Контекстна діаграма IDEF0 навчального процесу ЗВО зображена на рис. 2.1.

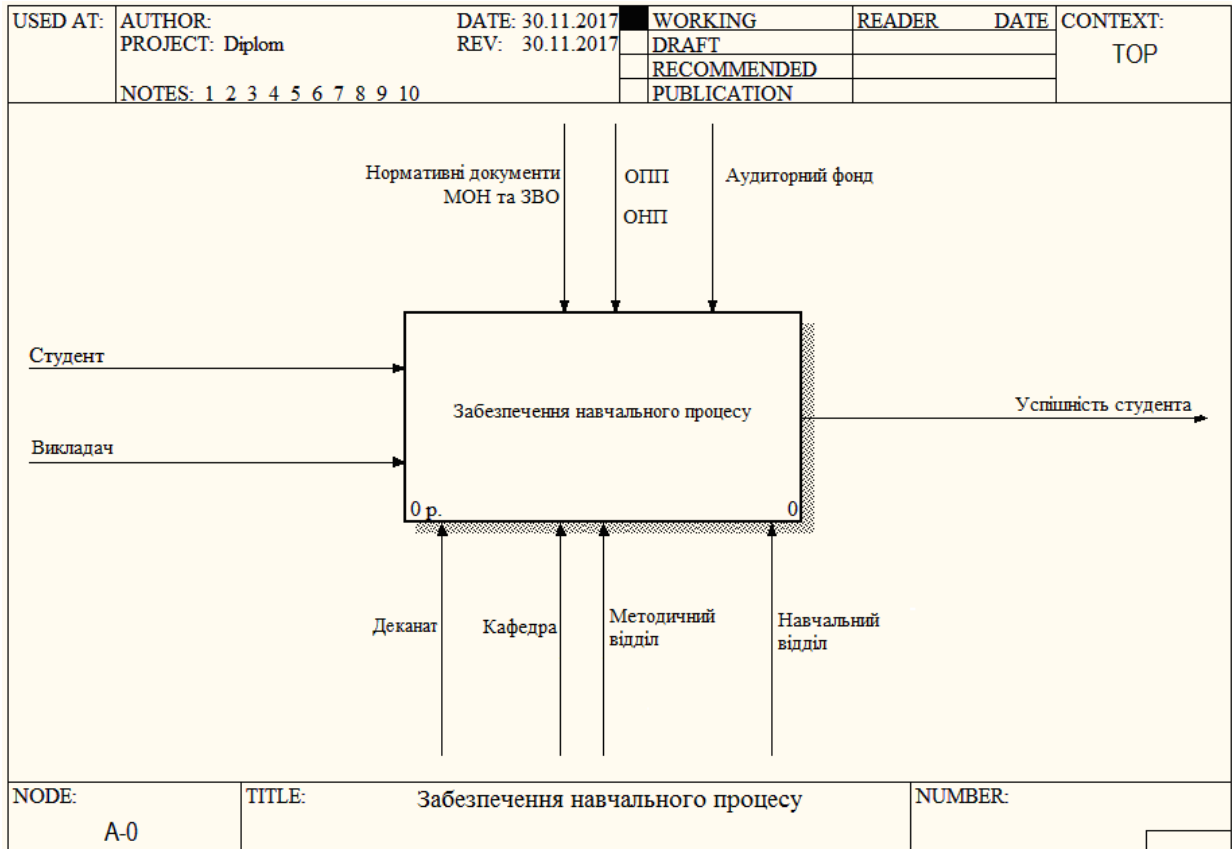


Рисунок 2.1 – Контекстна діаграма IDEF0

У контекстній діаграмі присутні такі граничні стрілки:

- 1) Входи: студент, викладач.
- 2) Управління: нормативні документи МОН та ЗВО, ООП, ОНП, аудиторний фонд.
- 3) Механізми: деканат, методичний відділ, кафедра, навчальний відділ.
- 4) Виходи: успішність студента.

2.1 Декомпозиція контекстної діаграми IDEF0

Відповідно до стандарту, на кожному рівні декомпозиції IDEF0 повинен використовуватися принцип обмеження об'єкта, тому відповідно до цього принципу вважається, що єдиний блок і кілька стрілок на самому верхньому (контекстному) рівні використовуються для визначення кордону всієї системи. Відповідно, стрілки, що стосуються цього блоку, описують головні управління, входи, виходи і механізми цієї системи [4].

З опису навчального процесу ЗВО зрозуміло, що декомпозиція контекстної діаграми IDEF0 повинна включати блоки розробки робочих навчальних планів, розподілу навчального навантаження між викладачами, складання розкладу занять, проведення поточного навчального процесу.

На рис. 2.2 зображено, як в загалі працює система «Навчальний процес університету».

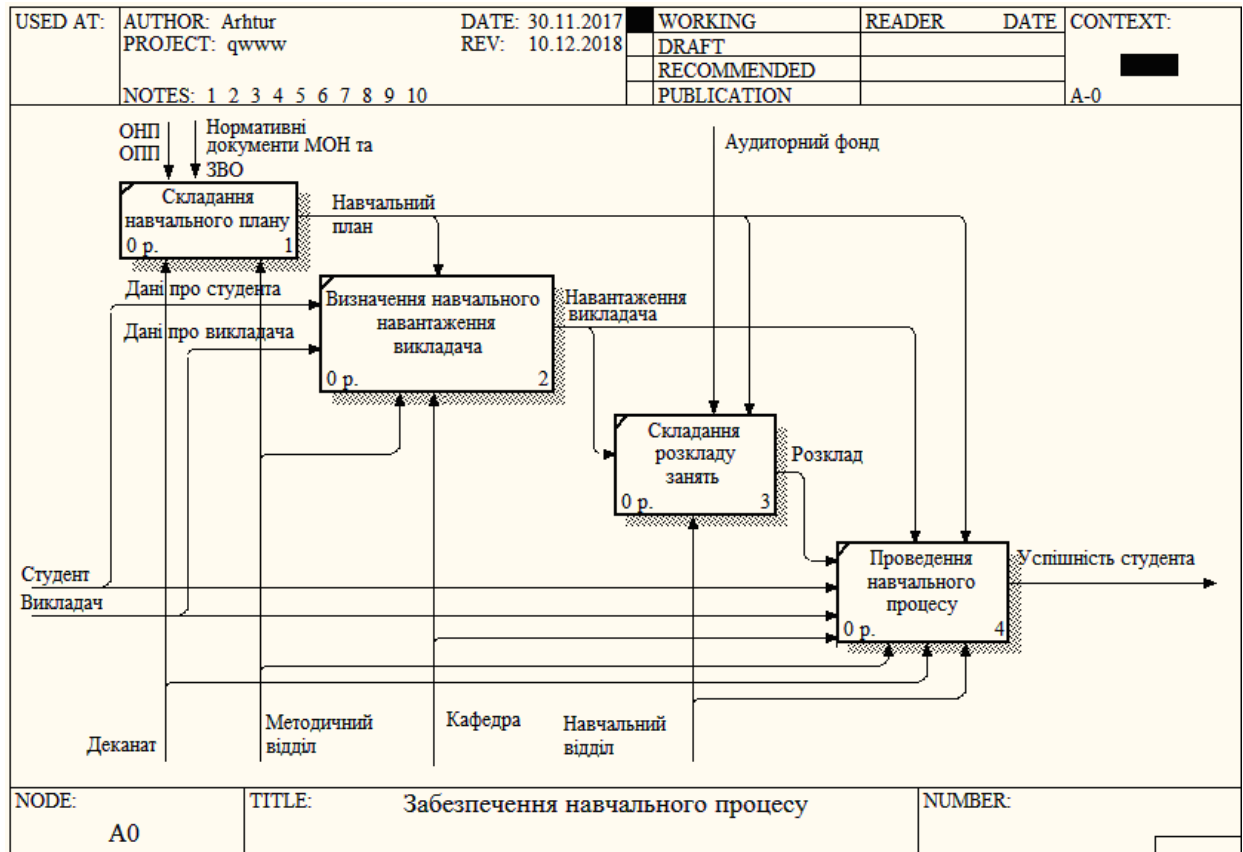


Рисунок 2.2 – Діаграма декомпозиції IDEF0

Для деталізації процесів, що відбуваються у середині блоків діаграми декомпозиції IDEF0 використовується методологія послідовного виконання процесу.

3 МЕТОДОЛОГІЯ МОДЕЛЮВАННЯ ПОТОКІВ ДАНИХ (DFD)

Діаграми потоків даних є ієрархію функціональних процесів, пов'язаних потоками даних. Мета такого подання – продемонструвати, як кожен процес перетворює свої вхідні дані у вихідні, а також виявити відносини між цими процесами.

Основними компонентами діаграм потоків даних є:

- зовнішні сутності;
- системи і підсистеми;
- процеси;
- накопичувачі даних;
- потоки даних.

З початку створюється початкова контекстна діаграма потоків даних, яка містить нульові процеси з іменами, що відображають діяльність організації, зовнішні сутності, з'єднані з нульовим процесом за допомогою потоків даних. Потоки даних відповідають документам, запитам або повідомленням, якими зовнішні сутності обмінюються з організацією (рис. 3.1).

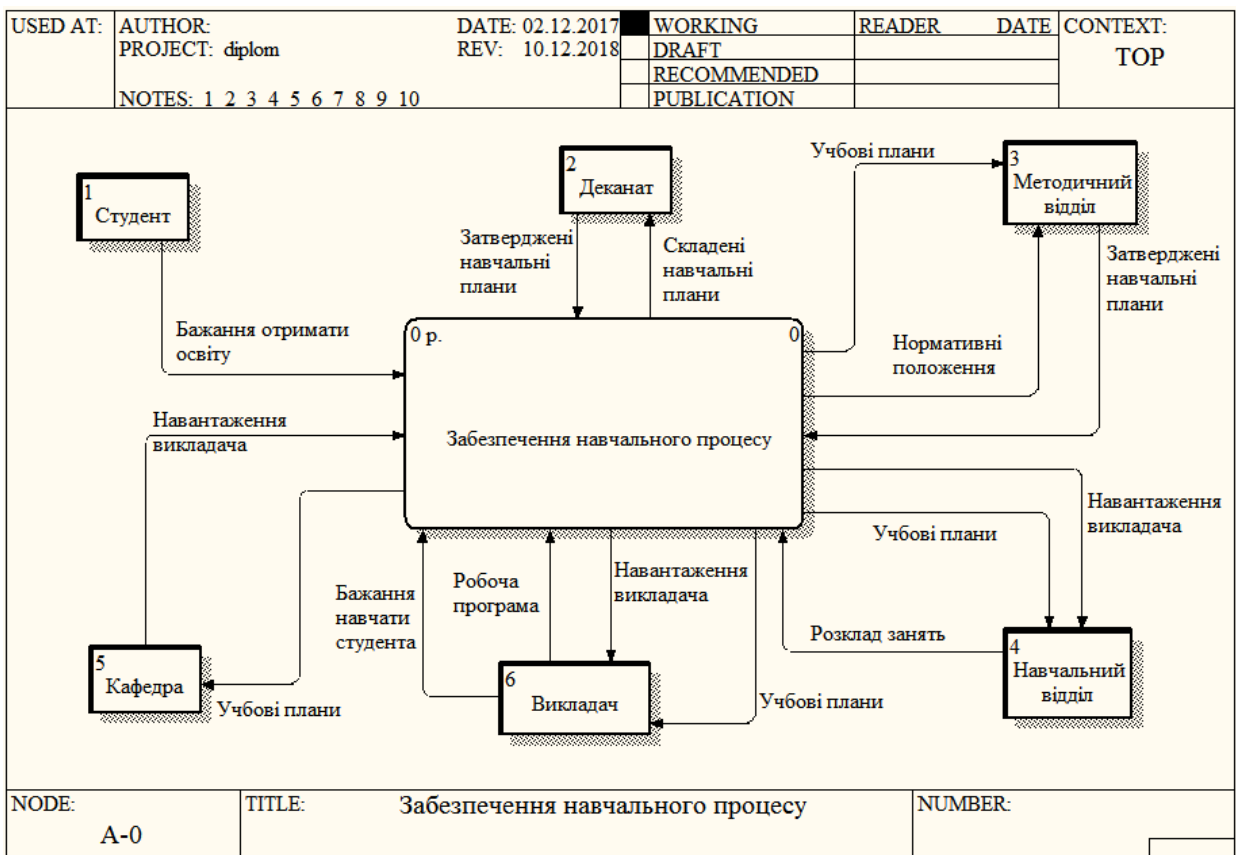


Рисунок 3.1 – Контекстна діаграма DFD навчального процесу ЗВО

Далі виконується декомпозиція контекстної діаграми DFD. Нульовий процес розбивається на складові системи і підпроцеси, які звичайно відповідають блокам декомпозиції діаграми IDEF0. Уточнюються потоки даних, які існують між зовнішніми сутностями та процесами. На рис. 3.2 зображена діаграма декомпозиції DFD навчального процесу ЗВО

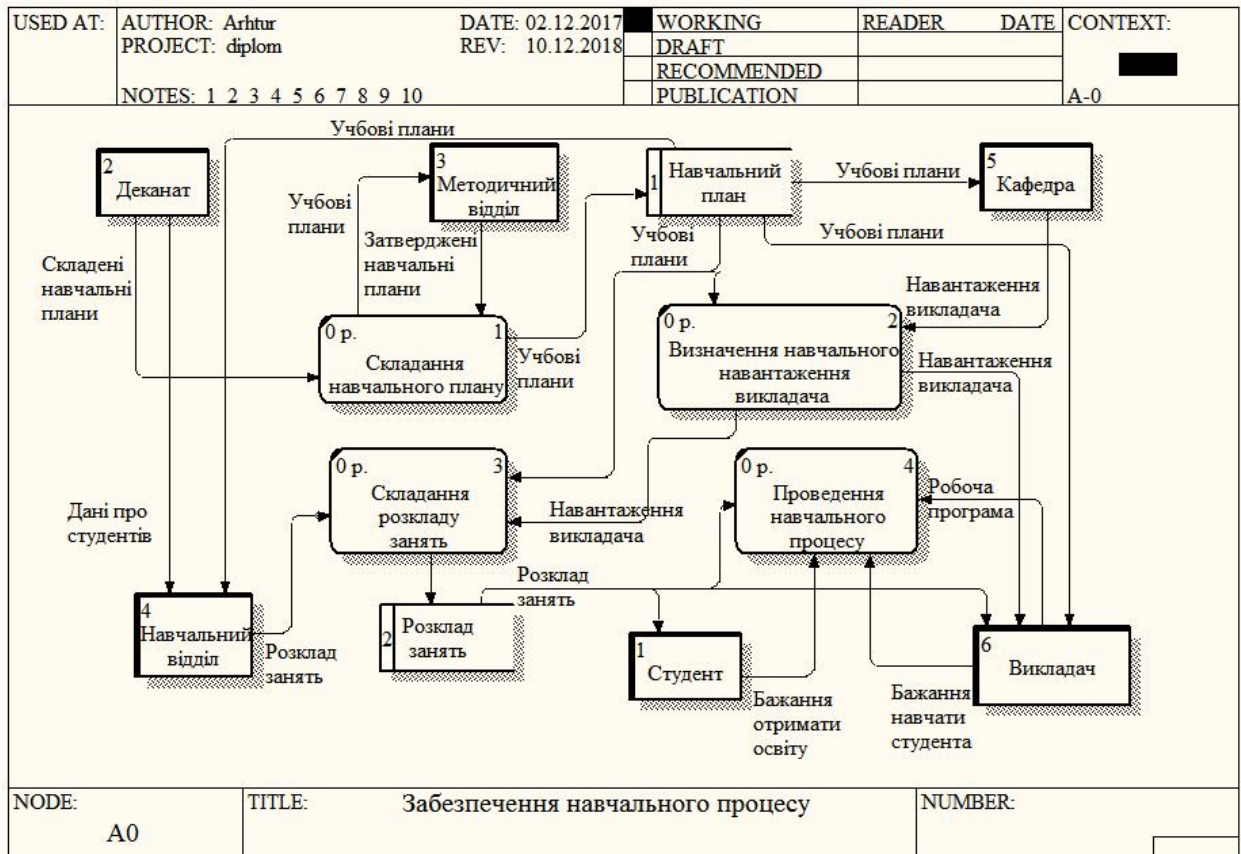


Рисунок 3.2 – Діаграма декомпозиції DFD навчального процесу ЗВО

Накопичувачі даних D1 та D2 можуть бути складовими однієї бази даних, або відноситись до різних баз даних.

4 СИСТЕМНИЙ АНАЛІЗ І КОНЦЕПТУАЛЬНЕ ПРОЕКТУВАННЯ БАЗИ ДАНИХ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Інформаційна система повинна ґрунтуватися на базі даних з кількох причин. По-перше, в навчальному процесі ЗВО задіяні великі обсяги даних. По-друге, ці дані повинні використовувати різні групи користувачів системи. По-третє, ці користувачі повинні мати можливість працювати з системою одночасно. Тому необхідно спроектувати базу даних інформаційної системи, яка буде працювати під управлінням СУБД, що забезпечить можливість працювати одночасно різним групам користувачів та дозволить надати цим групам користувачів права та привілеї, які необхідні їм для роботи.

Проектування бази даних починається зі збору зовнішніх уявлень на базу даних всіх груп користувачів. Групами користувачів бази даних інформаційної системи, що проектується, будуть усі зовнішні сутності, що присутні на діаграмі декомпозиції DFD навчального процесу ЗВО (рис. 3.2).

4.1 Етапи проектування бази даних

Процес проектування бази даних складається з трьох основних етапів: концептуальне, логічне і фізичне проектування [5 – 6].

Завданням концептуального проектування є отримання концептуальної моделі інформаційної системи, не залежної від будь-яких фізичних аспектів представлення інформації. Створена концептуальна модель є джерелом інформації для етапу логічного проектування бази даних.

Основні етапи концептуального проектування [5]:

- інтеграція зовнішніх представлень користувачів;
- визначення типів сутностей;
- визначення типів зв'язків;
- визначення атрибутів і зв'язування їх з типами сутностей і зв'язків;
- визначення доменів атрибутів;
- визначення атрибутів, що є потенційними і первинними ключами;
- перевірка моделі на відсутність надмірності;
- перевірка відповідності локальної концептуальної моделі конкретним транзакціям користувачів;
- обговорення локальних концептуальних моделей даних з кінцевими користувачами.

4.2 Концептуальне проектування бази даних

4.2.1 Визначення основних типів сутностей та типів зв'язків

З опису предметної області та з діаграмі декомпозиції DFD можна виділити основні групи користувачів ІС «Навчальний процес університету» і їх транзакції.

Група користувачів «Працівник деканату»:

- 1) Відновити навчальні плани на наступний навчальний рік.
- 2) Відновити контингент студентів на наступний навчальний рік.
- 3) Отримати затверджені методичним відділом навчальні плани факультету на наступний навчальний рік
- 4) Отримати затверджений методичним відділом список навчальних дисциплін на наступний навчальний рік з розподілом них по кафедрах.
- 5) Відновити прикріплення академічних груп до навчальних планів.
- 6) Отримати готовий розклад занять.
- 7) Зарахувати нових студентів на факультет.
- 8) Перевести студентів на наступний курс.
- 9) Відрахувати студентів, що закінчили навчання.
- 10) Відрахувати студентів, що не здали сесію.
- 11) Отримати від кафедр графік контролюючих заходів, які повинні виконувати у поточному семестрі студенти факультету.
- 12) Отримати від викладачів оцінки за виконання студентами контролюючих заходів.

Група користувачів «Працівник методичного відділу»:

- 1) Відновити нормативні документи щодо навчального процесу на наступний навчальний рік.
- 2) Отримати від факультетів робочі навчальні плани на наступний навчальний рік.
- 3) Затвердити робочі навчальні плани факультетів на наступний навчальний рік.
- 4) Затвердити список навчальних дисциплін на наступний навчальний рік та розподілити дисципліни по кафедрам.
- 5) Відновити список можливих контролюючих заходів для оцінки успішності навчання студентів.

- 6) Відновити перелік методичних документів, що забезпечують проведення навчального процесу в університеті.

Група користувачів «Працівник кафедри»:

- 1) Отримати дані за робочими навчальними планами, контингентом студентів і прикріпленням груп до навчальних планів.
- 2) Отримати список дисциплін кафедри на наступний навчальний рік.
- 3) Відновити списки дисциплін, для яких лекції читаються на потоках і назви потоків.
- 4) Відновити списки дисциплін і груп, для яких лабораторні заняття проводяться з розбивкою на підгрупи.
- 5) Прикріпити викладача до кожного заняття з дисциплін кафедри.
- 6) Оновити дані про допустимі аудиторії для занять.
- 7) Відновити дані про розподіл не аудиторного навчального навантаження викладачів.

Група користувачів «Диспетчер навчального відділу»:

- 1) Отримати дані, необхідні для складання розкладу занять, від деканатів факультетів і кафедр.
- 2) Скласти розклад занять.

Група користувачів «Викладач»:

- 1) Отримати навчальне навантаження на наступний навчальний рік.
- 2) Отримати розклад занять на наступний семестр.
- 3) Розробити робочу програму своїх дисциплін – для провідних викладачів.
- 4) Скласти графік контролюючих заходів для своєї дисципліни – для провідних викладачів.
- 5) Розробити контролюючі заходи результатів навчання з дисциплін.
- 6) Провести контролюючі заходи та поставити оцінки студентам.

Група користувачів «Студент»

- 1) Отримати розклад занять.
- 2) Отримати графік контролюючих заходів для всіх дисциплін, що вивчаються у поточному семестрі.
- 3) Отримати графік роботи викладачів на поточний семестр.
- 4) Виконати контролюючий захід.
- 5) Отримати оцінку за виконання контролюючого заходу.

Таким чином, є такі основні базові типи сутностей в базі даних:

- «Факультет»;

- «Кафедра»;
- «Навчальний План»;
- «Група»;
- «Викладач»;
- «Предмет» (навчальна дисципліна);
- «Вид Занять»;
- «Заняття».

Факультет має декілька академічних груп та декілька навчальних планів – між типами сутностей Факультет та Група існує тип зв'язку «один до багатьох»; так саме, як і між типами сутностей Факультет та Навчальний План.

Протягом одного навчального року кожна академічна група навчається за деяким одним навчальним планом, але декілька груп можуть іноді навчатись за одним навчальним планом – особливо це стосується перших курсів, коли вивчається небагато спеціалізованих дисциплін. Таким чином між типом сутностей Навчальний_план і типом сутностей Група існує тип зв'язку «один до багатьох».

Деякі навчальні дисципліни можуть зустрічатись у декількох навчальних планах, наприклад, загальноосвітні дисципліни. У різних навчальних планах для таких дисциплін може передбачатись різне аудиторне навантаження, тобто різна кількість занять різного виду. Тому необхідно у схему бази даних додати похідний тип сутності ПунктПлану (пункт навчального плану, дисципліна навчального плану), відповідно до якого і потрібно визначати заняття з деякої дисципліни для деякої групи.

Кожен навчальний план містить декілька пунктів, кожен пункт плану відноситься лише до однієї дисципліни, але будь-яка дисципліна (предмет) може зустрічатись у декількох пунктах навчальних планів, як у різних навчальних планах, так і у тому самому навчальному плані у обох семестрах. Таким чином між типами сутностей НавчальнийПлан та ПунктПлану існує тип зв'язку «один до багатьох»; так саме, як і між типами сутностей Предмет та ПунктПлану.

У однієї дисципліни навчального плану зазвичай заплановано декілька видів навчального навантаження: лекції, практичні заняття (семінари), лабораторні роботи. Усі три види занять зустрічаються для однієї дисципліни досить рідко, зазвичай планується проведення лекцій та одного з видів практичних занять: семінари або лабораторні роботи. Інколи для дисципліни можуть бути призначені тільки лекції, або тільки практичні заняття. Таким чи-

ном, потрібен додатковий похідний тип сутностей НавантаженняПунктуПлану, який буде містити дані про те, якого виду заняття передбачені для кожної дисципліни, та кількість годин цих аудиторних занять на семестр. Між типами сутностей ПунктПлану та НавантаженняПунктуПлану існує тип зв'язку «один до багатьох», оскільки у загальному випадку для однієї дисципліни протягом семестру проводяться заняття декількох видів.

Для одного виду занять одного пункту навчального плану може бути призначено більше однієї академічної пари занять на тиждень, якщо кількість годин занять на семестр більше $2 * \text{weeks}$, де weeks – кількість тижнів у семестрі. Для складання розкладу занять потрібен похідний тип сутності ЗаняттяГрупи, у якому екземпляр сутності містить дані про те, для якої групи з якої дисципліни призначена академічна пара занять, якого виду заняття, та проводяться вони щотижня, або раз на два тижні. Між типами сутностей НавантаженняПунктуПлану та ЗаняттяГрупи у загальному випадку існує тип зв'язку «один до багатьох», оскільки до одного екземпляру сутності НавантаженняПунктуПлану може бути прив'язане декілька екземплярів сутності ЗаняттяГрупи.

На кожній кафедрі працює декілька викладачів, та кожен викладач, як правило, проводить декілька занять. Між типами сутностей Кафедра та Викладач існує тип зв'язку «один до багатьох». Перед розподілом занять між викладачами кафедри спочатку визначається, для яких дисциплін лекції будуть проводитись як потокові, так склад цих академічних потоків. Також визначається, по яких дисциплінах лабораторні заняття будуть проводитись з розподілом академічних груп на підгрупи. Потім для кожного отриманого заняття призначається викладач. Скільки для з одного лабораторного заняття групи може бути отримано два заняття підгруп, потрібен новий похідний тип сутності Заняття. Між типом сутностей ЗаняттяГрупи та Заняття існує тип зв'язку «один до багатьох», так саме, як і між типами сутностей Викладач та Заняття.

Визначивши типи зв'язків між виявленими типами сутностей, отримуємо основну ER-діаграму бази даних для підсхеми «Розклад занять» (рис. 4.1).

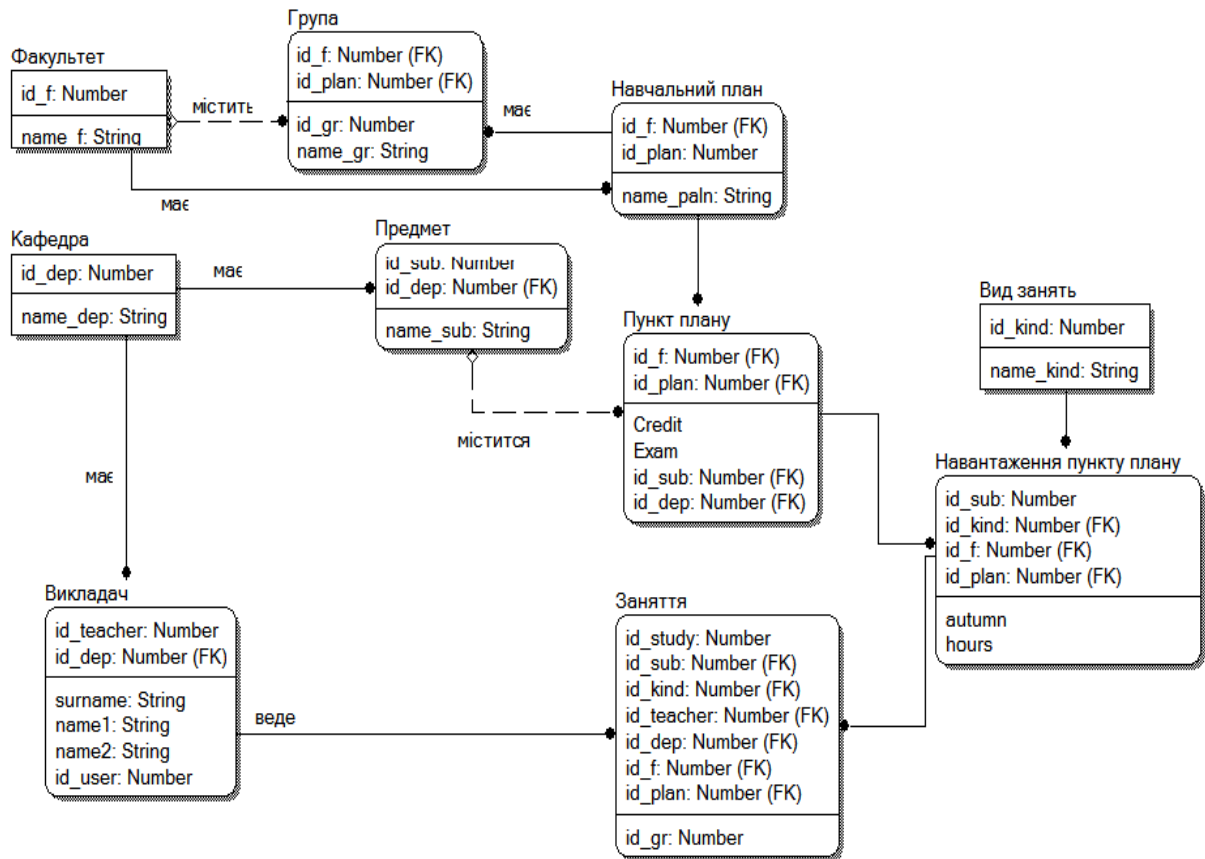


Рисунок 4.1 – Основна ER-діаграма підсхеми «Розклад занять» бази даних ІС «Навчальний процес університету»

Деякі лекції читаються на потоці з декількох груп. Для складання розкладу занять ця інформація також повинна зберігатися в базі даних. Отримаємо новий тип сутності Потік.

Визначивши типи зв'язків між виявленими типами сутностей, отримаємо доповнену ER-діаграму бази даних для підсхеми «Розклад занять» (рис. 4.2).

На доповненій ER-діаграмі на рис. 6 відображена підсхема «Розклад занять» у контексті всіх груп користувачів, крім групи «Працівник навчального відділу». Для вказаної групи користувачів у схему «Розклад занять» повинні ще входити типи сутностей, які задіяні у процесі складання розкладу занять.

Оскільки в даній роботі розглядається моделювання та проектування підсистеми «Викладач», діаграма на рис. 4.2 є остаточною для підсхеми «Розклад занять» бази даних ІС «Навчальний процес університету» у контексті вибраної групи користувачів.

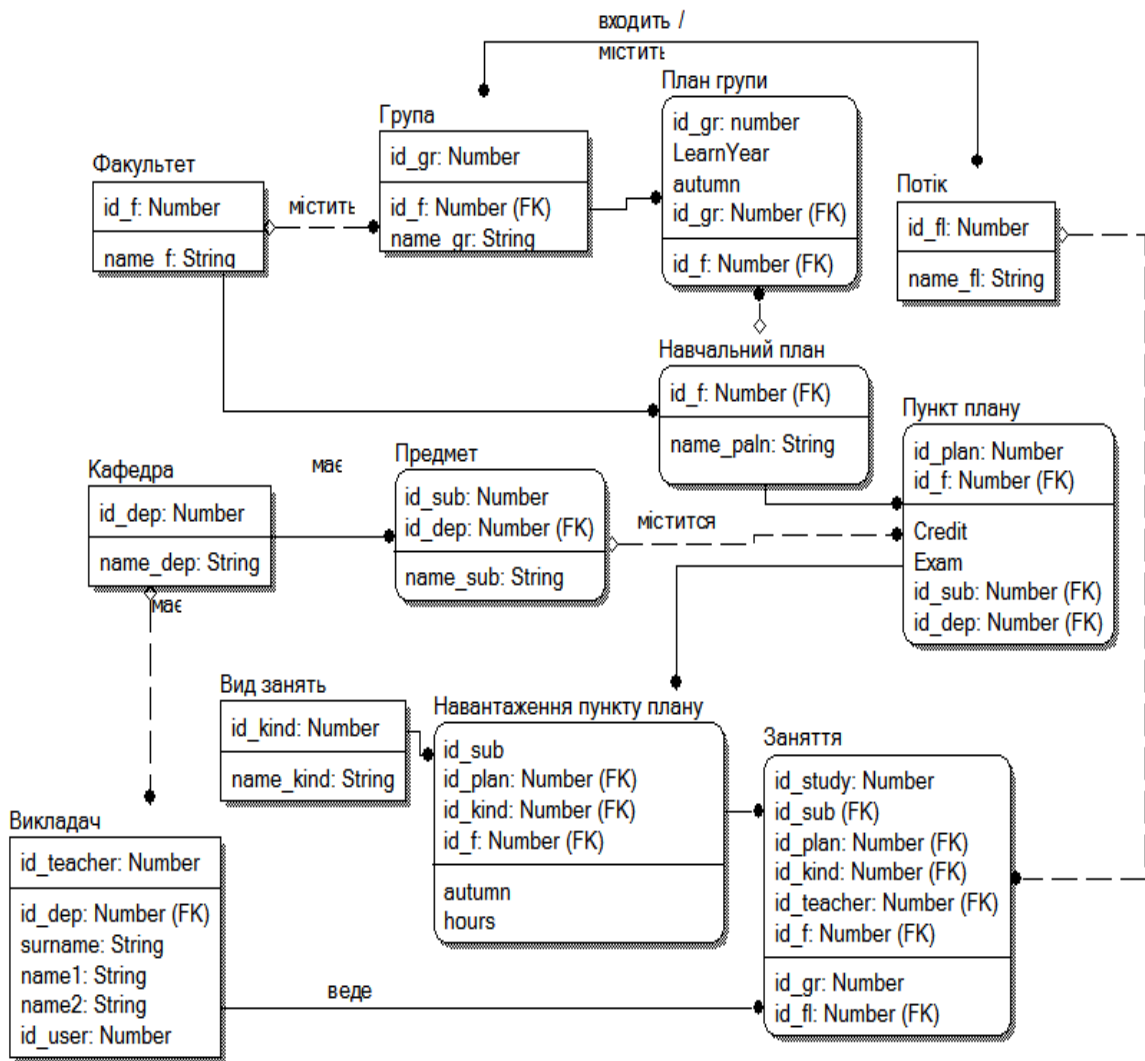


Рисунок 4.2 – Доповнена ER-діаграма підсхеми «Розклад занять» бази даних ІС «Навчальний процес університету»

Крім підсхеми «Розклад занять» потрібно розглянути підсхему, що описує проведення викладачем навчального процесу.

Провідні викладачі з дисципліни складають робочу програму. У цій робочій програмі вказуються теми, які розглядаються в дисципліні, а також вказуються контролюючі заходи і оцінки за них у вигляді балів. З оцінок за виконання контролюючих засобів розраховується накопичувальна оцінка поточної успішності студента з дисципліни.

У робочій програмі вказуються змістовні модулі, на які розподіляється вивчення дисципліни. Один модуль містить одну або декілька тем, та може включати в себе один або декілька контролюючих заходів. Оцінки за окремі контролюючі заходи модулю підсумовуються, і сумарна оцінка за виконання

кожного змістовного модулю виставляється у інтегральну відомість поточної успішності студента з дисципліни.

Контролюючі заходи можуть бути у вигляді комп'ютерних тестів і некомп'ютерних завдань: контрольних робіт, рефератів, колоквиумів і т.д. Оцінка за виконання комп'ютерних тестів виставляється автоматично по завершенню студентом тесту, некомп'ютерні контролюючі заходи викладач повинен перевіряти вручну і виставляти оцінку сам.

Комп'ютерні тести можуть виконуватись за допомогою тестувальної програми, яка є частиною ІС «Навчальний процес університету». У цьому випадку тестувальна програма буде використовувати вихідні дані інформаційної системи та записувати оцінки за виконані контролюючі заходи у відповідні таблиці загальної бази даних «Навчальний процес університету». Але у загальному випадку тести можуть виконуватись за допомогою окремої програми, яка записує результати виконання тесту студентами у текстовий файл.

Таким чином, можна виділити нові типи сутностей в схемі бази даних:

- «Студент»;
- «Тема дисципліни»;
- «Модуль»;
- «Контроль» (контролюючий захід);
- «Оцінка» (студента по контролюючому заходу).

У групі навчається декілька студентів, кожен студент відноситься лише до однієї групи – між типами сутностей Група та Студент існує тип зв'язку «один до багатьох».

Теми дисципліни (предмету) повинні асоціюватись саме з дисципліною, а не з пунктом навчального плану, оскільки одні й ті теми можуть вивчатись при вивченні дисципліни за різними навчальними планами, хоча кількість годин, що відводиться на вивчення теми, може бути різною у різних навчальних планах. Між типом сутностей Предмет та типом сутностей ТемаПредмету існує тип зв'язку «один до багатьох».

Для кожного пункту навчального плану провідний викладач складає робочу програму, яка містить декілька змістовних модулів – між типами сутностей ПунктПлану та Модуль існує тип зв'язку «один до багатьох». Один змістовний модуль може включати в себе одну або декілька тем з дисципліни, отже між типами сутностей Модуль та ТемаПредмету існує тип зв'язку «один до багатьох» (рис. 4.3).

Програма лекційних модулів

| Змістовні модулі | Назва змістовного модуля | Назви тем | Денна форма | | | |
|------------------|-----------------------------------|---|--------------------------|-------------------|----------------------|------------------------------|
| | | | Кіл-сть аудиторних годин | Кіл-сть годин СРС | Форми завдань на СРС | Форми поточного контролю СРС |
| ЗМ-Л1 | Моделювання даних | Тема 1. Системи баз даних. Основні поняття та архітектура | 2 | 1 | ПЛЗ ПМКР | КР-1 |
| | | Тема 2. Моделі даних | 2 | 1 | ПЛЗ ПМКР | |
| | | Тема 3. Реляційна модель даних | 4 | 2 | ПЛЗ ПМКР | |
| | | Тема 4. Методи аналізу та проектування баз даних | 2 | 1+5 | ПЛЗ ПМКР | |
| | | Тема 5. Теорія нормалізації реляційної моделі | 2 | 1 | ПЛЗ ПМКР | |
| | Разом за модуль | | 12 | 11 | | |
| ЗМ-Л2 | Робота з реляційними базами даних | Тема 6. Мова SQL | 6 | 3 | ПЛЗ ПМКР | КР-2 |
| | | Тема 7. Мова T-SQL | 6 | 2 | ПЛЗ ПМКР | |
| | | Тема 8. Розробка застосувань до баз даних | 8 | 2+5 | ПЛЗ ПМКР | |
| | Разом за модуль | | 20 | 12 | | |

Рисунок 4.3 – Опис лекційних змістовних модулів у робочій програмі

Один модуль може включати в себе один або декілька контролюючих заходів – між типами сутностей Модуль та Контроль існує тип зв'язку «один до багатьох». Для лекційних модулів можуть призначатись контрольні роботи та усне опитування. Практичний модуль може складатись з декілька лабораторних робіт (рис. 4.4).

По кожному контролюючому заходу викладач ставить декілька оцінок (різним студентам), та у кожного студента є декілька оцінок (по різним контролюючим заходам) – між типами сутностей Контроль та Оцінка існує тип зв'язку «один до багатьох»; так саме, як і між типами сутностей Студент та Оцінка.

Виявивши типи зв'язків між виявленими типами сутностей, отримуємо ER-діаграму підсхеми «Успішність студентів» БД «Університет» (рис. 4.5).

ПРОГРАМА ПРАКТИЧНИХ МОДУЛІВ

| Змістовні модулі | Форма занять (назва) | Теми робіт (занять) | Денна форма | | | |
|------------------|----------------------|---|--------------------------|-------------------|----------------------|------------------------------|
| | | | Кіл-сть аудиторних годин | Кіл-сть годин СРС | Форми завдань на СРС | Форми поточного контролю СРС |
| ЗМ-П11 | Лабораторні | 1. Створення бази даних у реляційній СКБД MS SQL Server 2000. | 2 | 2+1 | ПУОП ПЛР | УО ЛР |
| | | 2. Прості запити до БД. | 4 | 2+1 | | |
| | | 3. Складні запити до БД: вкладені запити та групування | 4 | 2+1 | | |
| | | 4. Маніпулювання даними в БД командами мови Transact-SQL | 4 | 2+1 | | |
| Разом за модуль | | | 14 | 12 | | |
| ЗМ-П11 | Лабораторні | 5. Проектування та розробка БД | 4 | 2+1 | ПУОП ПЛР | УО ЛР |
| | | 6. Розробка серверного ПЗ | 6 | 2+1 | | |
| | | 7. Створення клієнтського додатку до БД | 4 | 2+1 | | |
| Разом за модуль | | | 14 | 9 | | |

Рисунок 4.4 – Опис практичних змістовних модулів у робочій програмі

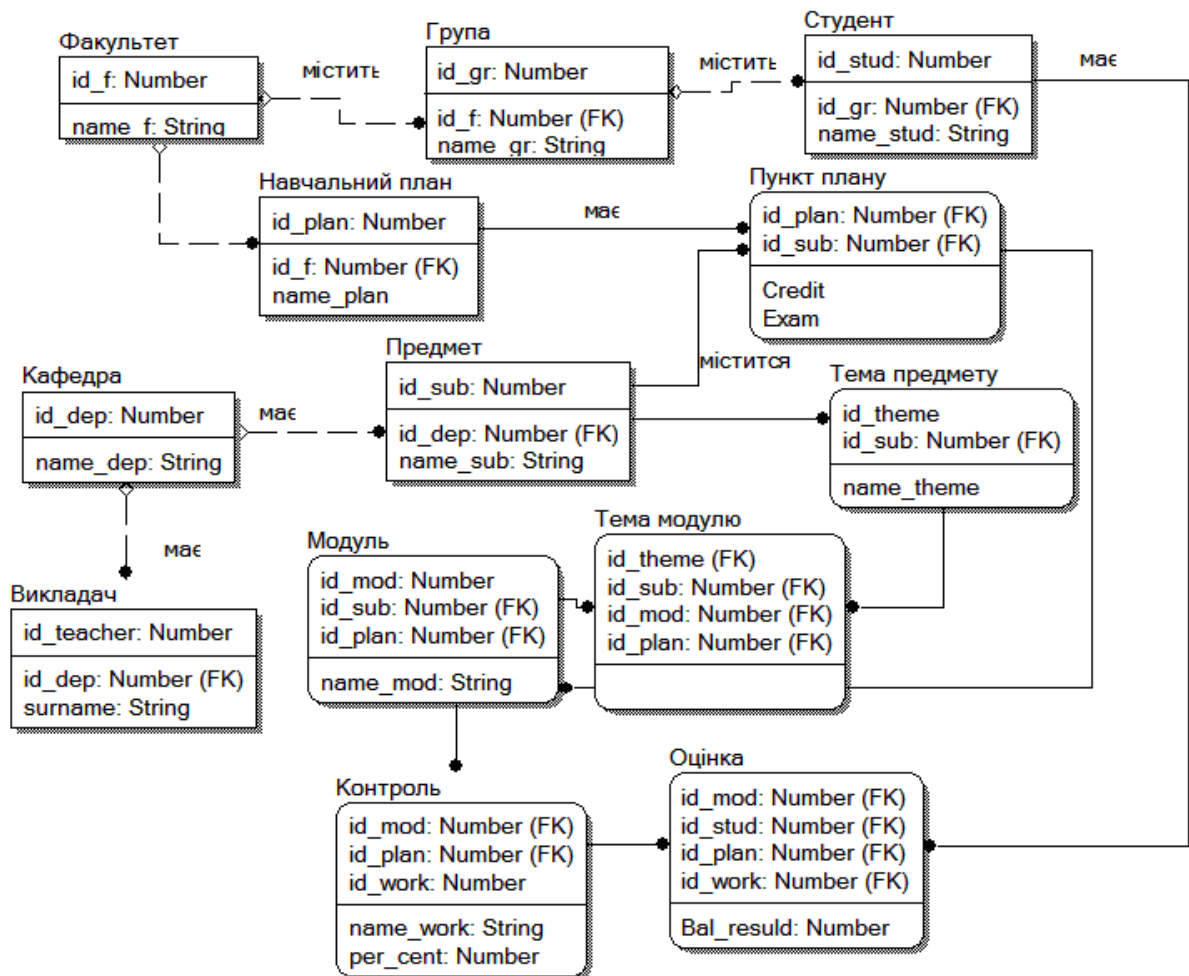


Рисунок 4.5 – Остаточна ER-діаграма підсхеми «Успішність студентів»

Для отримання схеми БД для підсистеми «Викладач» потрібно поєднати дві схеми – «Розклад занять» та «Успішність студентів» (рис. 4.6).

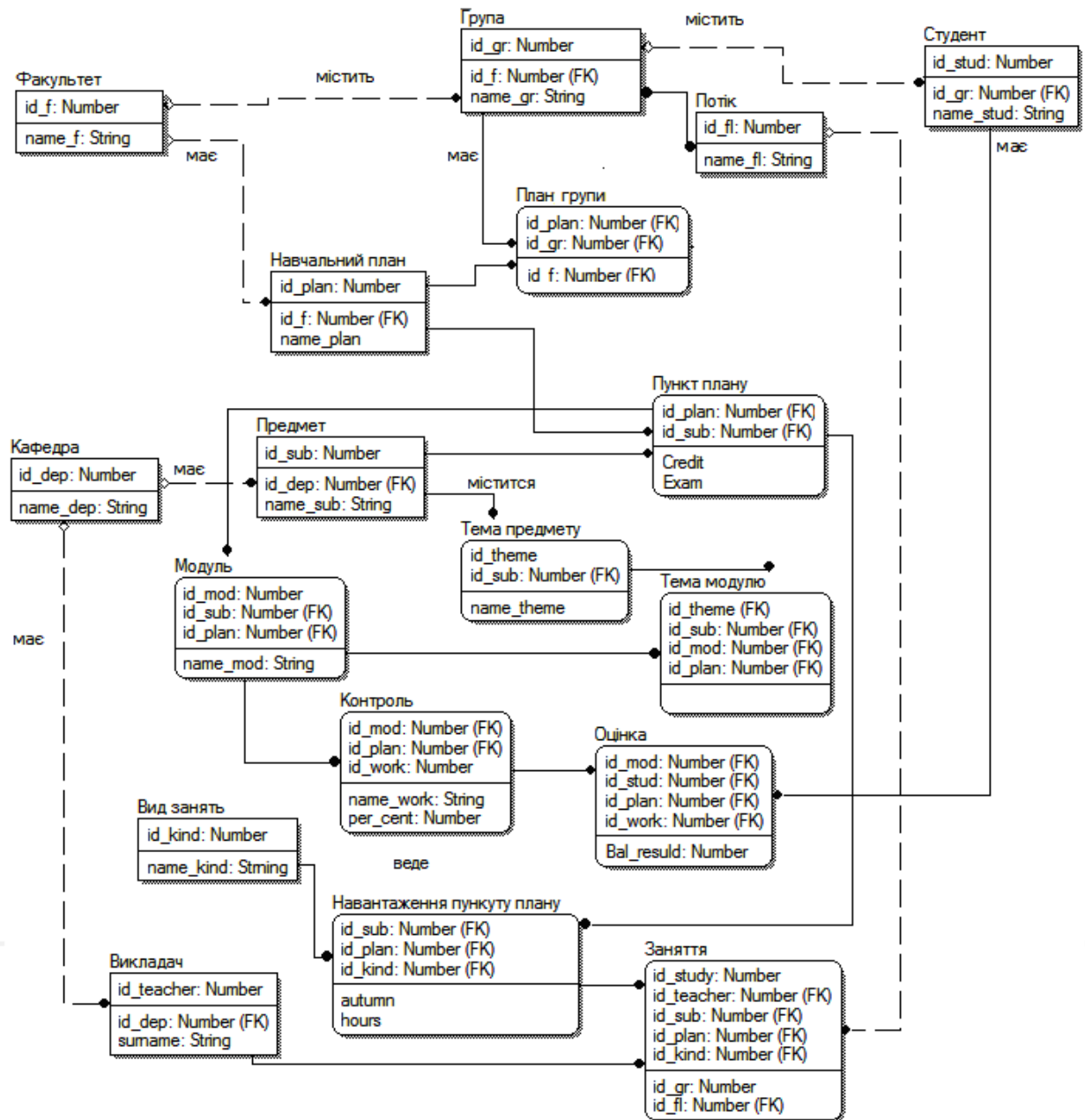


Рисунок 4.6 – ER-діаграма підсхеми «Викладач» бази даних ІС «Навчальний процес університету»

Діяльність викладача не обмежена аудиторними заняттями та заходами контролю. Він повинен вести наукову, методичну, організаційну та виховну роботу. У подальших версіях ІС бажано відобразити всі види діяльності викладачів. Але дана версія ІС відображає саме навчальний процес в універси-

теті. Тому підсистему «Викладач» цієї системи можна вважати достатньо повною.

4.2.2 Визначення атрибутів і зв'язування їх з типами сутностей і зв'язків

Наступним кроком концептуального моделювання є визначення атрибутів і зв'язування їх з типами сутностей і зв'язків.

У типу сутності Факультет будуть атрибути: назва, код факультету (ідентифікатор).

У типу сутності Група будуть атрибути: назва, код групи (ідентифікатор), код факультету, кількість студентів.

У типу сутності Навчальний_план будуть атрибути: назва, код навчального плану (ідентифікатор), код факультету, навчальний рік, семестр, кількість тижнів навчання.

У типу сутності Кафедра будуть атрибути: назва, код кафедри (ідентифікатор).

У типу сутності Посада_викладача будуть атрибути: назва, код посади (ідентифікатор), чи може читати лекції, чи може керувати дипломним проектуванням, чи може керувати науковою роботою магістрів.

У типу сутності Предмет будуть атрибути: код предмета (ідентифікатор), код кафедри, назва, коротка назва.

У типу сутності Викладач будуть атрибути: код викладача (ідентифікатор), код кафедри, прізвище, ім'я, по батькові, код посади, розмір ставки.

У типу сутності ПунктПлану будуть атрибути: код навчального плану, код предмета, кількість кредитів, вид семестрового контролю (іспит, залік), наявність курсового проекту (роботи), вид курсового проектування (проект, робота), кількість годин навчальної практики.

У типу сутності Вид_занять будуть атрибути: назва, код виду занять (ідентифікатор).

У типу сутності НавантаженняПунктуПлану будуть атрибути: код навчального плану, код предмета, код виду занять, кількість годин занять (за семестр).

У типу сутності Заняття будуть атрибути: код заняття (ідентифікатор), код предмету, код виду занять, код групи, код викладача, код потоку (для потокових лекцій), номер підгрупи (для занять з розподілом на підгрупи).

У типу сутності Модуль будуть атрибути: назва, номер модулю, код навчального плану, код дисципліни, максимальний бал (що може отримати студент у якості оцінки за виконання модулю).

У типу сутності ТемаПредмету будуть атрибути: назва, номер теми, код дисципліни.

У типу сутності ТемаМодулю будуть атрибути: назва, номер модулю, код навчального плану, код дисципліни, кількість годин аудиторних занять, кількість годин самостійної підготовки.

Більшість виявлених атрибутів відносяться або до домену імен (назв): прізвище, ім'я, по батькові, назва предмета, назва факультету, назва кафедри, назва дисципліни, – або до доменна цілих чисел: номер (код) предмета, код кафедри. Атрибути «чи може ...» відносяться до логічного типу. Атрибути «розмір ставки» та «розмір навантаження на одну групу (студента)» відносяться до не цілочисельного типу даних. Вказані типи даних допустимі для СУБД, заснованих на будь-якої з сучасних моделей даних.

4.2.3 Визначення атрибутів, що є потенційними і первинними ключами

У всіх базових типів сутностей первинним ключем буде атрибут з позначкою (ідентифікатор). У типів сутностей, що за змістом є таблицями-довідниками: Факультет, Кафедра, Предмет, Посада_викладача, Вид_занять, Вид_навантаження, – потенційним ключем (унікальним) є атрибут «назва».

Похідні та асоціативні типи сутностей повинні мати складові потенційні (унікальні) ключі, до складу яких входять атрибути, що є зовнішніми ключами.

4.2.4 Визначення відповідності концептуальної моделі транзакціям користувачів

У розділі 4.2.1 перелічені транзакції основних груп користувачів ІС «Університет», що стосуються процесів розрахунку навчального навантаження викладачів та процесу складання розкладу занять.

Розглянемо кожну транзакцію.

- 1) Отримати навчальне навантаження викладача на наступний навчальний рік. Для цієї транзакції потрібно зробити вибірку даних з зв'язаних таблиць: Викладач – Заняття – Навантаження пункту плану – Вид занять – Пункт плану – Предмет.

- 2) Отримати розклад занять на наступний семестр. Для цієї транзакції потрібно зробити вибірку даних з зв'язаних таблиць Викладач – Заняття розкладу – Заняття – Навантаження пункту плану – Пункт плану – Предмет.
- 3) Розробити робочу програму своїх дисциплін – для провідних викладачів. Для цієї транзакції потрібно зробити вибірку даних з зв'язаних таблиць Викладач – Заняття розкладу – Заняття – Навантаження пункту плану – Модуль.
- 4) Скласти графік контролюючих заходів для своєї дисципліни – для провідних викладачів. Для цієї транзакції потрібно зробити вибірку даних з зв'язаних таблиць Викладач – Заняття розкладу – Заняття – Навантаження пункту плану – Модуль – Контроль.
- 5) Розробити контролюючі заходи результатів навчання з дисциплін. Для цієї транзакції також потрібно зробити вибірку даних з зв'язаних таблиць Викладач – Заняття розкладу – Заняття – Навантаження пункту плану – Модуль – Контроль.
- 6) Провести контролюючі заходи та поставити оцінки студентам. Для цієї транзакції також потрібно зробити вибірку даних з зв'язаних таблиць Викладач – Заняття розкладу – Заняття – Навантаження пункту плану – Модуль – Контроль – Студент – Група.

Для транзакцій, призначених групі користувачів «Викладач» визначені сутності, їх атрибути і зв'язки між ними. Отже, отримана концептуальна модель коректна у контексті підсистеми «Викладач».

5 ВИБІР СУБД ТА СЕРЕДОВИЩА РОЗРОБКИ ЗАСТОСУВАННЯ

5.1 Вибір СУБД

База даних ІС «Навчальний процес університету» розробляється поступово на протязі декількох років. Спочатку БД була створена у СУБД MS SQL Server 2005, потім була перероблена для СУБД MS SQL Server 2008.

Поступово поповнюється серверне програмне забезпечення інформаційної системи. Для можливості використання попередніх розробок потрібно залишити для ІС СУБД MS SQL Server, можна лише вибрати більш нову версію. Але вибирати нову версію слід обачно, щоб не виникла потреба переробки серверного ПЗ, або зміни структури таблиць, наприклад, типів даних атрибутів таблиць.

MS SQL Server 2008 є безкоштовним випуском SQL Server і являє собою ідеальну платформу даних для навчання і створення невеликих серверних додатків, які можуть поширюватися незалежними постачальниками програмного забезпечення [7 –10].

Тому на цей час можна залишити СУБД версії MS SQL Server 2008.

5.2 Вибір середовища розробки застосування та мови програмування

Для розробки додатків до баз даних звичайно використовуються середовища швидкої розробки застосувань. Раніше лідером у розробці середовищ швидкої розробки програмних застосувань була компанія Borland, але у останні роки лідерство перейшло до компанії Microsoft з її продуктом Visual Studio.

У наш час продукти Microsoft для розробників входять до списку найбільш затребуваного програмного забезпечення для розробників додатків до баз даних [8 – 11].

Продукт компанії Microsoft – Visual Studio – є потужним середовищем розробки, що забезпечує високу якість коду на протязі всього циклу життя програмного забезпечення, від проектування до впровадження.

Microsoft Visual Studio 2010 – це потужне середовище швидкої розробки програмних застосувань різного виду, якому властиві всі переваги минулих версій цього середовища розробки, та у якому додані нові можливості. Основні зміни, які відбулися в порівнянні з попередніми версіями, можна віднести до таких груп, як ефективність, поліпшення існуючих технологій, розширюваність, вплив поточних тенденцій.

Visual Studio 2010 Ultimate оснащена всіма удосконаленими засобами тестування для забезпечення стабільної якості коду. Розробники застосувань можуть використовувати кодовані тести інтерфейсу користувача, за допомогою яких можна автоматизувати тестування інтерфейсу веб-додатків і додатків Windows, засоби ручного тестування, функцію Test Professional, тестування продуктивності веб-додатків і інші корисні функції, які недоступні в інших версіях Visual Studio.

Модульне тестування та розробка, керована тестами – середовище розробки Visual Studio 2010 включає в себе велику кількість розширень, які допомагають в реалізації цих стратегій. Так само з'явилися нові динамічні засоби і середовище DLR, а також архітектура ASP.NET MVC.

Розробка баз даних вимагає тієї ж ретельності і уваги що і розробка додатків. У Visual Studio 2010 Ultimate це враховується: користувачам надаються надійні засоби розробки та управління змінами, які дозволяють синхронізувати додаток і базу даних.

Таким чином програмне середовище розробки застосувань компанії Microsoft Visual Studio 2010 є дуже вдалим вибором у якості середовища розробки застосувань для баз даних.

6 СЕРВЕРНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

В інформаційній системі «Навчальний процес університету», підсистемою якого є ІС «Викладач», передбачається наявність декілька груп користувачів, які мають права маніпулювати даними в БД за допомогою офісних застосувань. У групі користувачів «Викладач» може бути багато користувачів з такими правами.

У цих випадках бажано, щоб офісні клієнтські застосування лише надавали користувачам необхідний інтерфейс, а обробка даних відбувалася на боці серверу. Тобто, розробляти клієнтські застосування бажано у вигляді «тонких клієнтів».

Для тонких клієнтів не бажано генерувати SQL-код команд на додавання, оновлення, усунення записів у таблицях БД. Ці застосування повинні лише передавати до серверу виклик збережених процедур, які і будуть виконувати необхідні дії по маніпулюванню даними у таблицях БД.

Тому до серверного програмного забезпечення потрібно додати збережені процедури для додавання, оновлення, усунення записів у нових таблицях БД [11].

Процедура ModulAdd додає запис у таблицю Modul:

```
CREATE procedure ModulAdd
```

```
  @id_mod smallint output,
```

```
  @id_sub smallint,
```

```
  @id_plan smallint,
```

```
  @name_mod varchar (60),
```

```
  @DateReport smalldatetime,
```

```
  @KindModul bit,
```

```
  @MaxBal tinyint,
```

```
  @k tinyint output,
```

```
  @msg varchar (50) output
```

Параметри процедури:

@id_mod – ідентифікатор модулю;

@id_plan – ідентифікатор навчального плану;

@id_sub – ідентифікатор навчальної дисципліни;

@name_mod – назва змістовного модулю;

@DateReport – дата підведення підсумків виконання модулю;

@MaxBal – максимальний бал, який може отримати студент за виконання даного модулю;

@k – код помилки, вихідний параметр;

@msg – вихідний параметр, повідомлення про помилку.

Процедура ModulAdd перевіряє, чи вже є в БД змістовний модуль з наступними атрибутами: id_sub= @id_sub and id_plan=@id_plan and name_mod=@name_mod. Якщо запис про такий модуль є, то генерується повідомлення про помилку. Якщо такого запису немає, то запис із зазначеними значеннями атрибутів додається в таблицю Modul.

Процедура ModulEdit змінює запис у таблиці Modul:

```
CREATE procedure ModulEdit
```

```
  @id_mod smallint,  
  @id_sub smallint,  
  @id_plan smallint,  
  @name_mod varchar (60),  
  @DateReport smalldatetime,  
  @KindModul bit,  
  @MaxBal tinyint,  
  @k tinyint output,  
  @msg varchar (50) output
```

Параметри процедури такі самі, як у процедури ModulAdd.

Процедура ModulEdit перевіряє, чи вже є в БД змістовний модуль з наступними атрибутами: id_sub= @id_sub and id_plan=@id_plan and name_mod=@name_mod and id_mod!= @id_mod.

Якщо запис про такий модуль є, то генерується повідомлення про помилку.

Якщо такого запису немає, то в таблиці Modul у записі з атрибутами id_sub = @id_sub and id_plan = @id_plan and id_mod = @id_mod змінюються значення атрибутів name_mod, DateReport, KindModul, MaxBal на значення параметрів процедури @name_mod, @DateReport, @KindModul, @MaxBal відповідно.

Процедура ModulDel усуває запис з таблиці Modul:

```
CREATE procedure ModulDel
```

```
  @id_mod smallint,  
  @id_sub smallint,  
  @id_plan smallint,  
  @m bit,  
  @k tinyint output,  
  @msg varchar (50) output
```

Параметри процедури такі самі, як у процедури ModulAdd.

Якщо @m=0, то процедура ModulDel перевіряє, чи є у підлеглих таблицях записи, що зв'язані з записом таблиці Modul з атрибутами id_sub=@id_sub and id_plan=@id_plan and id_mod=@id_mod. Якщо такі записи є, то генерується повідомлення про помилку. Якщо таких записів немає, то запис із зазначеними значеннями атрибутів видаляється з таблиці Modul.

Якщо @m=1, то процедура ModulDel видаляє з таблиці Modul запис з вказаними атрибутами, а також з усіх підлеглих таблиць видаляються записи, що зв'язані з цим записом таблиці Modul.

Процедура ThemeSubAdd додає запис у таблицю ThemeSubject:

```
CREATE procedure ThemeSubAdd
```

```
  @id_sub smallint,  
  @name_theme varchar (60),  
  @id_theme smallint output,  
  @k tinyint output,  
  @msg varchar (50) output
```

Параметри процедури:

@id_theme – ідентифікатор теми;
@id_sub – ідентифікатор навчальної дисципліни;
@k – код помилки, вихідний параметр;
@msg – вихідний параметр, повідомлення про помилку.

Процедура ThemeSubAdd перевіряє, чи вже є в таблиці ThemeSubject запис з атрибутами: id_sub = @id_sub and name_theme = @name_theme. Якщо запис про таку тему є, то генерується повідомлення про помилку. Якщо такого запису немає, то запис із зазначеними значеннями атрибутів додається в таблицю ThemeSubject.

Процедура ThemeSubEdit змінює запис у таблиці ThemeSubject:

```
CREATE procedure ModulEdit
```

```
  @id_theme smallint,  
  @id_sub smallint,  
  @name_mod varchar (60),  
  @k tinyint output,  
  @msg varchar (50) output
```

Параметри процедури такі самі, як у процедури ThemeSubAdd.

Процедура ThemeSubEdit перевіряє, чи вже є в таблиці ThemeSubject запис з атрибутами: id_sub= @id_sub and name_theme= @name_theme and id_theme!= @id_theme.

Якщо такий запис є, то генерується повідомлення про помилку.

Якщо такого запису немає, то в таблиці ThemeSubAdd у запису з атрибутами `id_sub = @id_sub and id_theme = @id_theme` значення атрибуту `name_theme` змінюються на `@name__theme`.

Процедура ThemeSubDel усуває запис з таблиці ThemeSubject:

```
CREATE procedure ThemeSubDel
```

```
  @id_theme smallint,
```

```
  @id_sub smallint,
```

```
  @m bit,
```

```
  @k tinyint output,
```

```
  @msg varchar (50) output
```

Параметри процедури такі самі, як у процедури ThemeSubAdd.

Якщо `@m=0`, то процедура ThemeSubDel перевіряє, чи є у підлеглих таблицях записи, що зв'язані з записом таблиці ThemeSubject з атрибутами `id_sub= @id_sub and id_theme = @id_theme`.

Якщо такі записи є, то генерується повідомлення про помилку. Якщо зв'язаних записів немає, то з таблиці ThemeSubject видаляється запис із зазначеними значеннями атрибутів.

Якщо `@m=1`, то процедура ThemeSubDel з таблиці ThemeSubject видаляє запис з вказаними атрибутами, а також з усіх підлеглих таблиць видаляються записи, що зв'язані з цим записом таблиці ThemeSubject.

7 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ

7.1 Загальні відомості

Програма Teacher.exe розроблена у середовище швидкої розробки застосувань Visual Studio 2010. Програма написана на мові програмування С# [12 – 15]. Програма є додатком до бази даних StudyingProcess, тому вона може працювати у локальній мережі, де розташований сервер баз даних з завантаженою БД StudyingProcess.

7.2 Функціональне призначення

Програма Teacher.exe розроблена у якості додатку до бази даних StudyingProcess у вигляді автоматизованого робочого місця викладача ЗВО.

Програма призначена для перегляду даних, необхідних для роботи викладачеві: розкладу занять, аудиторного навантаження викладача, тем дисциплін, що веде викладач, тощо), – в зручному вигляді. Програма дозволяє зручно змінювати ті дані, які користувач має право оновлювати.

7.3 Керівництво програміста

Програма розроблена за модульним принципом. Кожний модуль є окремою формою застосування. Кожна форма має свій клас форми.

Крім класів форм є класи параметрів форм та клас List_Item, ці класи не мають конструкторів та методів.

Клас Form1 – клас головної форми програми. Конструктор класу не має параметрів:

```
public Form1()
```

Конструктор класу Form1 ініціалізує компоненти форми, заповнює компонент derComboBox рядками тексту з запиту до БД, ініціалізує об'єкти, що є відображають збережені процедури бази даних.

Методи класу форми Form1.

Метод init_commands_DelMLCmd:

```
private void init_commands_DelMLCmd()
```

Призначений для ініціалізації команди виклику збереженої процедури ModulDel:

Метод init_commands_AddThCmd:

```
private void init_commands_AddThCmd ()
```

Призначений для ініціалізації команди виклику збереженої процедури ThemeSubAdd.

Метод init_commands_EditThCmd:

```
private void init_commands_EditThCmd ()
```

Призначений для ініціалізації команди виклику збереженої процедури ThemeSubEdit.

Метод init_commands_DelThCmd:

```
private void init_commands_DelThCmd ()
```

Призначений для ініціалізації команди виклику збереженої процедури ThemeSubDel.

Метод init_commands_AddResCmd:

```
private void init_commands_AddResCmd ()
```

Призначений для ініціалізації команди виклику збереженої процедури ResWorkAdd.

Метод init_commands_DelResCmd:

```
private void init_commands_DelResCmd ()
```

Призначений для ініціалізації команди виклику збереженої процедури ResWorkDel.

Метод init_commands_EditResCmd:

```
private void init_commands_EditResCmd ()
```

Призначений для ініціалізації команди виклику збереженої процедури ResWorkEdit.

Метод init_commands_DelThMLCmd:

```
private void init_commands_DelThMLCmd ()
```

Призначений для ініціалізації команди виклику збереженої процедури ThemeModDel.

Метод fillDepComboBox заповнює випадний список кафедр університету DepComboBox значеннями запиту до БД:

```
private void fillDepComboBox()
```

Метод fillTeachComboBox заповнює випадний список викладачів ЗВО:

```
private void fillTeachComboBox(short id_dep)
```

Параметр методу:

id_dep – ідентифікатор кафедри, де працюють викладачі.

Метод fillSubComboBox заповнює значеннями з запиту до БД випадний список з переліком дисциплін, які веде викладач:

```
private void fillSubComboBox(short id_teach)
```

Параметр методу:

id_teach– ідентифікатор викладача.

Решта методи класу Form1 являються обробниками подій натискання на кнопки або вибору пункту випадного списку.

Обробник події натискання на кнопку yearButton:

```
private void yearButton_Click(object sender, EventArgs e)
```

Метод yearButton_Click є обробником події натискання на кнопку «ОК» біля текстового поля для вводу номеру навчального року. При натисканні на цю кнопку числове значення, введене у поле yearTextBox записується у змінну класу форми Form1 LearnYear. Змінна LearnYear використовується як параметр у запитах до БД.

Обробник події вибору пункту випадного списку depComboBox:

```
private void depComboBox_SelectedIndexChanged(object sender, EventArgs e)
```

Метод depComboBox_SelectedIndexChanged є обробником події вибору пункту випадного списку «Кафедра». Цей метод зчитує ідентифікатор вибраної кафедри та записує у змінну id_dep. Змінна id_dep використовується як параметр запитів до БД, які виводять дані на компоненти типу DataGridView на формі програми. Також змінна id_dep є параметром запиту, за допомогою якого випадний список teachComboBox заповнюється прізвищами викладачів вибраної кафедри.

Обробник події вибору пункту випадного списку teachComboBox:

```
private void teachComboBox_SelectedIndexChanged(object sender, EventArgs e)
```

Метод є обробником події вибору пункту випадного списку викладачів кафедри. Метод teachComboBox зчитує в змінну id_teach ідентифікатор вибраного у списку викладача.

Обробник події натискання на кнопку addThemeButton:

```
private void addThemeButton_Click(object sender, EventArgs e)
```

Метод addThemeButton_Click є обробником події натискання на кнопку «Добавить тему» на вкладці форми «Тематика дисциплін». Цей метод робить недоступними кнопки на сторінці та робить видимими приховані елементи форми, які використовуються для введення даних при додаванні запису у таблицю тем навчальної дисципліни.

Обробник події натискання на кнопку editThemeButton:

```
private void editThemeButton_Click(object sender, EventArgs e)
```

Метод editThemeButton_Click є обробником події натискання на кнопку «Изменить название или номер темы» на вкладці «Тематика дисциплін».

Цей метод також робить недоступними кнопки на сторінці та робить видимими приховані елементи форми, які використовуються для оновлення запису.

Обробник події натискання на кнопку delBThemeButton:

```
private void delThemeButton_Click(object sender, EventArgs e)
```

Метод є обробником події натискання на кнопку «Удалить работу» також на вкладці «Тематика дисциплін». Цей метод виводить на форму запитання, чи дійсно користувач хоче видалити з бази даних запис про вибрану тему вибраної навчальної дисципліни. Якщо користувач вибирає відповідь «ОК», то викликається зрережена процедура бази даних ThemeSubDel для видалення запису в таблиці ThemeSubject.

Обробник події натискання на кнопку addModButton:

```
private void addModButton_Click(object sender, EventArgs e)
```

Метод addModButton_Click є обробником події натискання на кнопку «Добавить модуль» на вкладці «Модули по дисциплине» головної форми програми. Цей метод створює об'єкт класу ModulParam – класу параметрів для конструктору форми Form2. В методі ініціалізуються значення параметрів, та створюється об'єкт класу Form2.

Клас Form2 – клас форми для додавання та редагування записів у таблиці Modul бази даних:

```
public Form2(ModulParam inout_params,
System.Data.OleDb.OleDbConnection connection)
```

Параметри конструктору:

inout_params – об'єкт класу ModulParam, що містить дані, необхідні для створення або редагування запису у таблиці Modul бази даних;

connection – змінна, яка використовується для передачі зв'язку з базою даних на форму.

Клас Form2 має 2 методи, що ініціалізують збережені процедури бази даних та метод OkButton1_Click

```
private void OkButton1_Click(object sender, EventArgs e)
```

Метод OkButton1_Click є обробником події натискання на кнопку «ОК» на формі Form2. Метод считує дані з текстових полів на сторінці. Ці дані використовуються для виклику збереженої процедури:

- процедури ModulAdd для додавання запису у таблицю Modul;
- процедури ModulEdit для оновлення запису у таблиці Modul.

7.4 Посібник користувача

При запуску програми Teacher.exe на екрані з'являється головна форма програми (рис. 7.1).

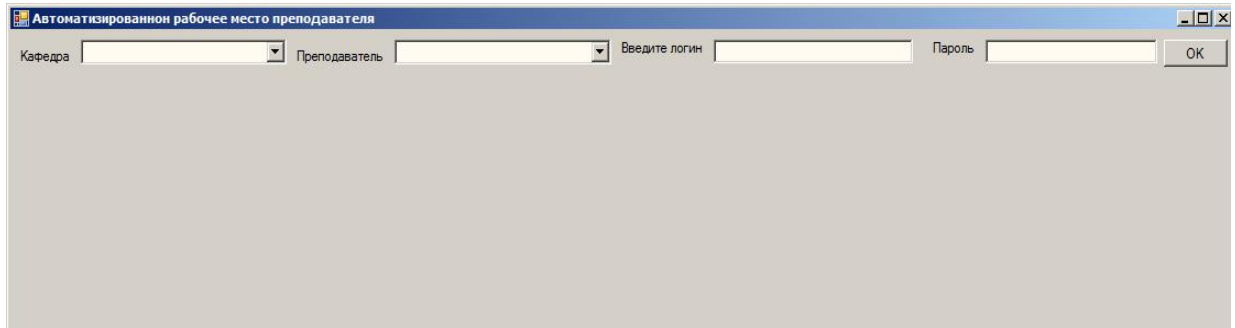


Рисунок 7.1 – Початок роботи з програмою

Користувач повинен вибрати власну кафедру зі списку кафедр університету (рис. 7.2).

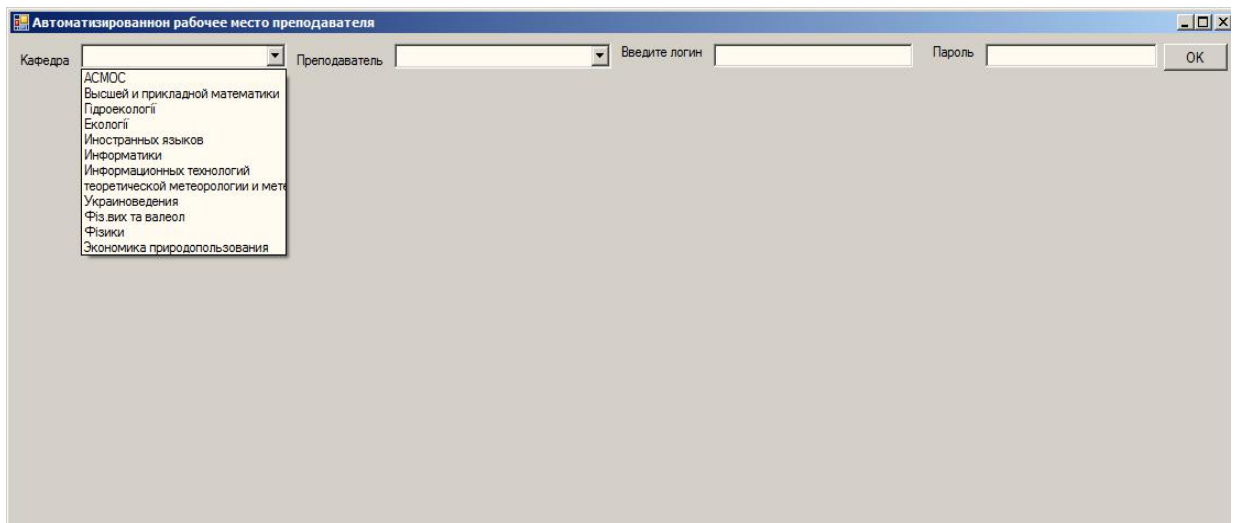


Рисунок 7.2 – Вибір кафедри користувачем

Після вибору кафедри заповнюється список викладачів цієї кафедри, і користувач повинен знайти себе у цьому списку (рис. 7.3).

Після цього користувач повинен ввести свій логін та пароль, які він призначив для роботи з застосуванням, та натиснути на кнопку «ОК» у правому верхньому куті форми (рис. 7.4).

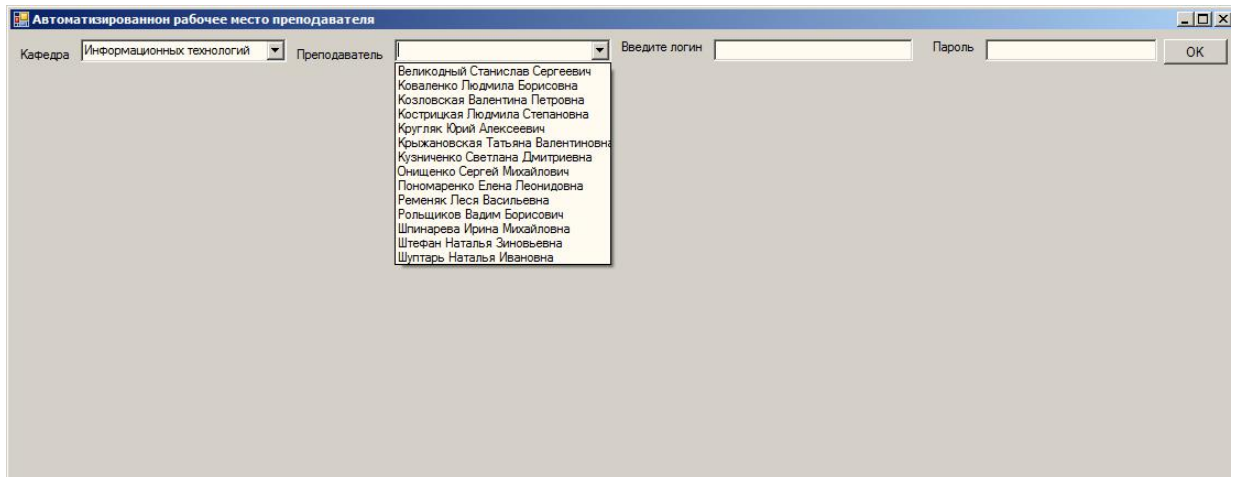


Рисунок 7.3 – Вибір викладача кафедри

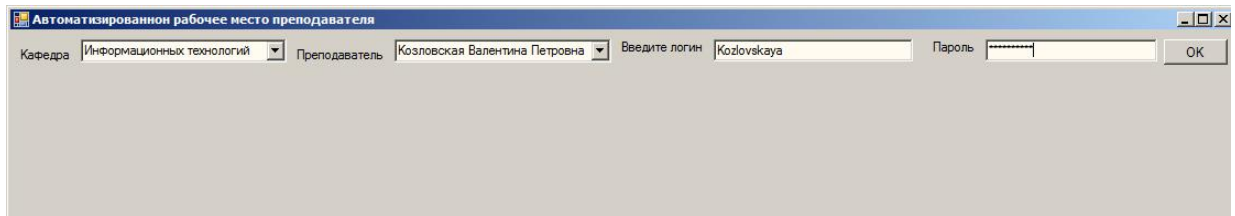


Рисунок 7.4 – Введення користувачем логіну та паролю

Після натискання користувачем кнопки «ОК» він отримує доступ для роботи з програмою, якщо він авторизований у системі. Якщо введені користувачем логін та пароль невірні, користувач не отримує можливості роботи з програмою. Для подальшої роботи користувач повинен ввести навчальний рік, дані по якому йому потрібні. Вводити треба рік осіннього семестру навчального року (рис. 7.5).

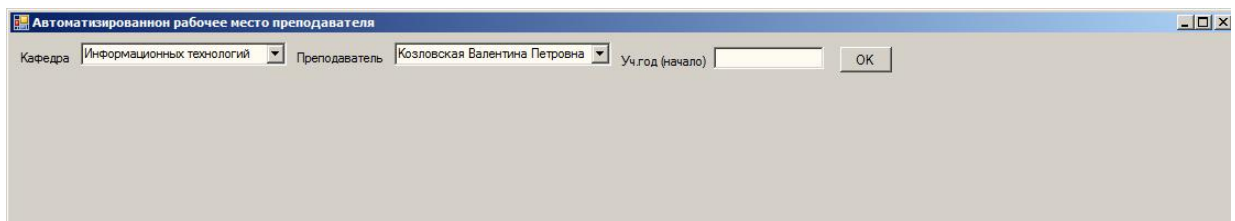


Рисунок 7.5 – Запрошення на введення навчального року

Після введення навчального року на формі програми з'являється багатосторінковий інструмент, на першій вкладці якого одразу виводиться аудиторне навантаження даного викладача у вказаному році (рис. 7.6).

| Семестр | Дисциплина | Вид занятий | Группа/Поток | Часы |
|---------|-----------------------------------|-------------|--------------|------|
| осень | Методи та засоби інтеграції даних | лекція | МИС-18 | 30 |
| осень | Методи та засоби інтеграції даних | семинар | МИС-18 | 15 |
| осень | Організація баз даних та знань | лекція | К-31-32,К-15 | 45 |
| осень | Організація баз даних та знань | лаб.робота | К-31 | 45 |
| осень | Організація баз даних та знань | лаб.робота | К-32 | 45 |
| весна | Мобільні технології | лекція | К-31 | 30 |
| весна | Мобільні технології | лаб.робота | К-31 | 30 |

Рисунок 7.6 – Вкладка «Аудиторная нагрузка»

Дані на 1-й вкладці користувач може лише переглядати.

Якщо перейти на другу вкладку, користувач має можливість побачити свій розклад занять у заданому семестрі (рис. 7.7 – 7.9).

Рисунок 7.7 – Вибір семестру для перегляду розкладу занять на вкладці «Расписание»

Як видно з рис. 7.8, розклад занять виводиться у дві таблиці: у лівій таблиці виводиться розклад занять у тижні з непарними номерами, у правій – з парними номерами. Якщо викладач вибрав семестр, для якого немає розкладу занять у БД, на вкладці виводяться дві порожні таблиці (рис. 7.9).

Автоматизированное рабочее место преподавателя

Кафедра: Информационных технологий | Преподаватель: Козловская Валентина Петровна | Уч.год (начало): 2018

Аудиторная нагрузка | Расписание | Учебные планы | Тематика дисциплин | Модули по дисциплинам | Контролирующие мероприятия | Оценки по модулям

Семестр: осень

| Нечетная неделя | | | Четная неделя | | |
|-----------------|-------------------------|-----------|---------------|-------------------------|-----------|
| Пара | Занятие | Аудитория | Пара | Занятие | Аудитория |
| Пн-1 | ОБД лекция,К-31-32,К-15 | 241(1) | Пн-1 | ОБД лекция,К-31-32,К-15 | 241(1) |
| Пн-2 | | | Пн-2 | | |
| Пн-3 | ОБД лаб. работа, К-32 | 327а(1) | Пн-3 | ОБД лаб. работа, К-32 | 327а(1) |
| Пн-4 | ОБД лаб. работа, К-31 | 327а(1) | Пн-4 | ОБД лаб. работа, К-32 | 327а(1) |
| Пн-5 | | | Пн-5 | | |
| Вт-1 | ОБД лекция,К-31-32,К-15 | 241(1) | Вт-1 | | |
| Вт-2 | | | Вт-2 | | |
| Вт-3 | | | Вт-3 | | |
| Вт-4 | | | Вт-4 | | |
| Вт-5 | | | Вт-5 | | |
| Ср-1 | ОБД лаб. работа, К-31 | 327а(1) | Ср-1 | ОБД лаб. работа, К-31 | 327а(1) |
| Ср-2 | | | Ср-2 | | |
| Ср-3 | | | Ср-3 | | |
| Ср-4 | | | Ср-4 | | |
| Ср-5 | | | Ср-5 | | |
| Чт-1 | | | Чт-1 | МЗД семинар, МИС-18 | 241(1) |
| Чт-2 | МЗД лекция, МИС-18 | 241(1) | Чт-2 | МЗД лекция, МИС-18 | 241(1) |
| Чт-3 | | | Чт-3 | | |
| Чт-4 | | | Чт-4 | | |
| Чт-5 | | | Чт-5 | | |
| Пт-1 | | | Пт-1 | | |
| Пт-2 | | | Пт-2 | | |
| Пт-3 | | | Пт-3 | | |
| Пт-4 | | | Пт-4 | | |
| Пт-5 | | | Пт-5 | | |

Рисунок 7.8 – Розклад занять викладача на вказаний семестр

Автоматизированное рабочее место преподавателя

Кафедра: Информационных технологий | Преподаватель: Козловская Валентина Петровна | Уч.год (начало): 2018

Аудиторная нагрузка | Расписание | Учебные планы | Тематика дисциплин | Модули по дисциплинам | Контролирующие мероприятия | Оценки по модулям

Семестр: весна

| Нечетная неделя | | | Четная неделя | | |
|-----------------|---------|-----------|---------------|---------|-----------|
| Пара | Занятие | Аудитория | Пара | Занятие | Аудитория |
| Пн-1 | | | Пн-1 | | |
| Пн-2 | | | Пн-2 | | |
| Пн-3 | | | Пн-3 | | |
| Пн-4 | | | Пн-4 | | |
| Пн-5 | | | Пн-5 | | |
| Вт-1 | | | Вт-1 | | |
| Вт-2 | | | Вт-2 | | |
| Вт-3 | | | Вт-3 | | |
| Вт-4 | | | Вт-4 | | |
| Вт-5 | | | Вт-5 | | |
| Ср-1 | | | Ср-1 | | |
| Ср-2 | | | Ср-2 | | |
| Ср-3 | | | Ср-3 | | |
| Ср-4 | | | Ср-4 | | |
| Ср-5 | | | Ср-5 | | |
| Чт-1 | | | Чт-1 | | |
| Чт-2 | | | Чт-2 | | |
| Чт-3 | | | Чт-3 | | |
| Чт-4 | | | Чт-4 | | |
| Чт-5 | | | Чт-5 | | |
| Пт-1 | | | Пт-1 | | |
| Пт-2 | | | Пт-2 | | |
| Пт-3 | | | Пт-3 | | |
| Пт-4 | | | Пт-4 | | |
| Пт-5 | | | Пт-5 | | |

Рисунок 7.9 – Відсутність розкладу занять на вибраний семестр

Наступна вкладка призначена для перегляду навчальних планів. На форму виводяться окремі пункти навчальних планів, які стосуються дисциплін, що веде викладач (рис. 7.10).

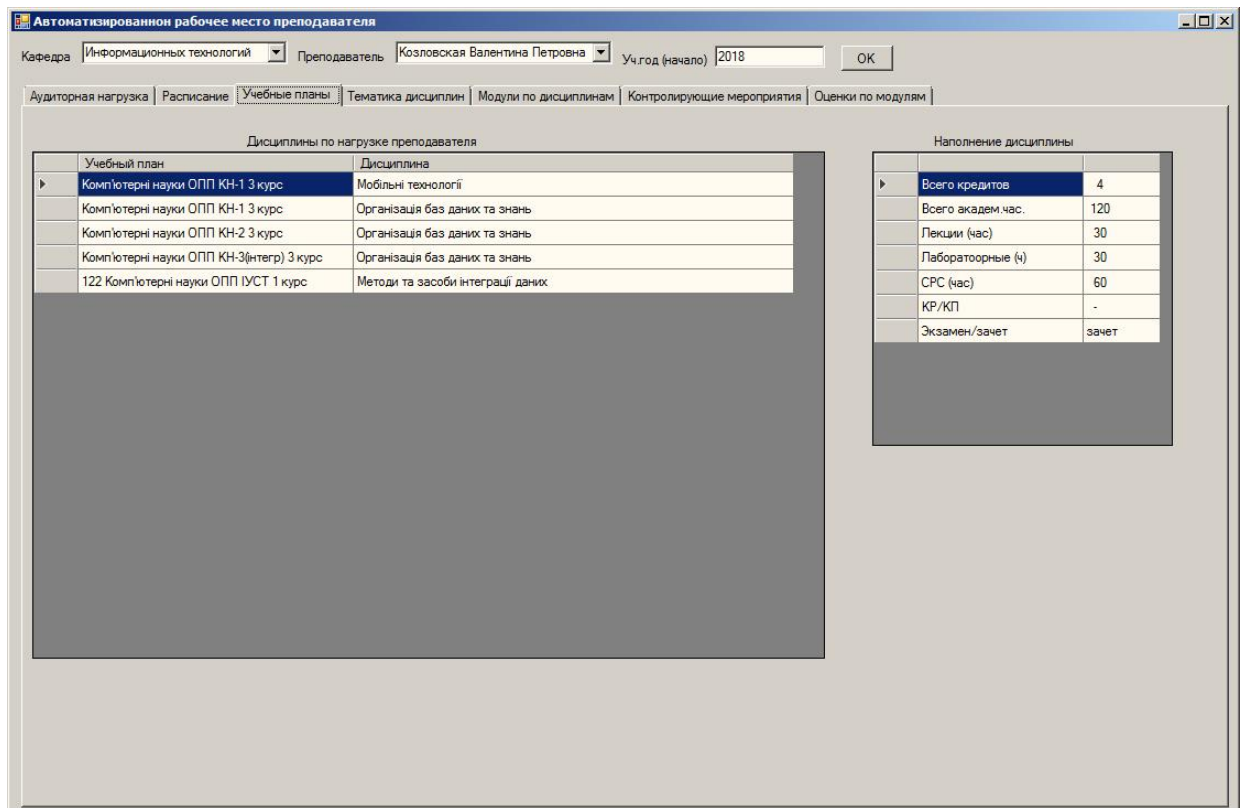


Рисунок 7.10 – Вкладка «Учебные планы»

На вкладці ліворуч виводиться таблиця з двома стовпчиками. У першому виводиться назва навчального плану, в другому – назва навчальної дисципліни цього плану, яку веде викладач-користувач.

Праворуч для поточної дисципліни лівої таблиці виводиться її навантаження згідно навчального плану: кількість кредитів, відведених на вивчення даної дисципліни, кількість годин, відведених на вивчення даної дисципліни, скільки з них відведено на лекції, практичні заняття (семінари), лабораторні роботи, самостійну підготовку студентів, наявність курсового проекту або курсової роботи та вид підсумкового контролю (іспит або залік).

Як видно з рис. 7.10 – 7.12, якщо у навчальних планах не передбачені які-небудь види занять – лекції, практичні або лабораторні, – то програма не виводить відповідного рядка з навантаженням для цього виду занять. Завжди виводиться кількість годин на самостійну підготовку студентів.

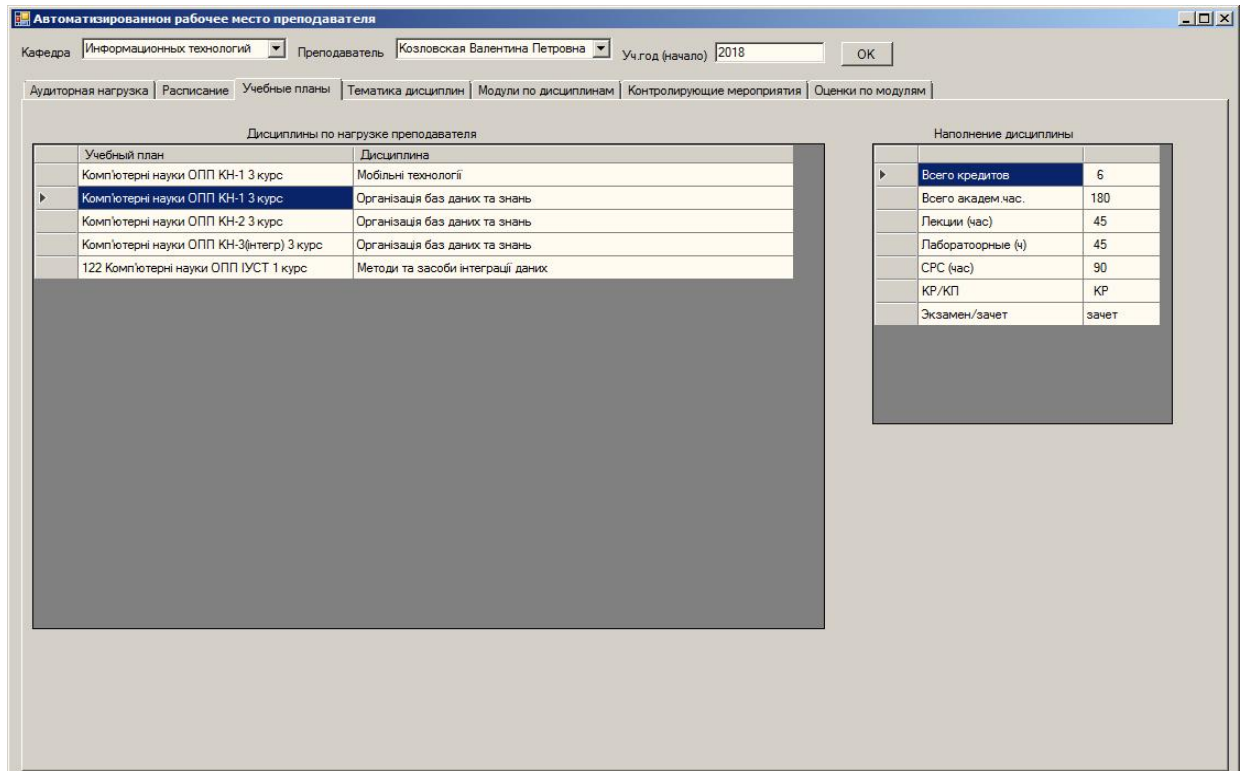


Рисунок 7.11 – Вкладка «Учебные планы»: наявність у плані дисципліни курсової роботи та лабораторних робіт

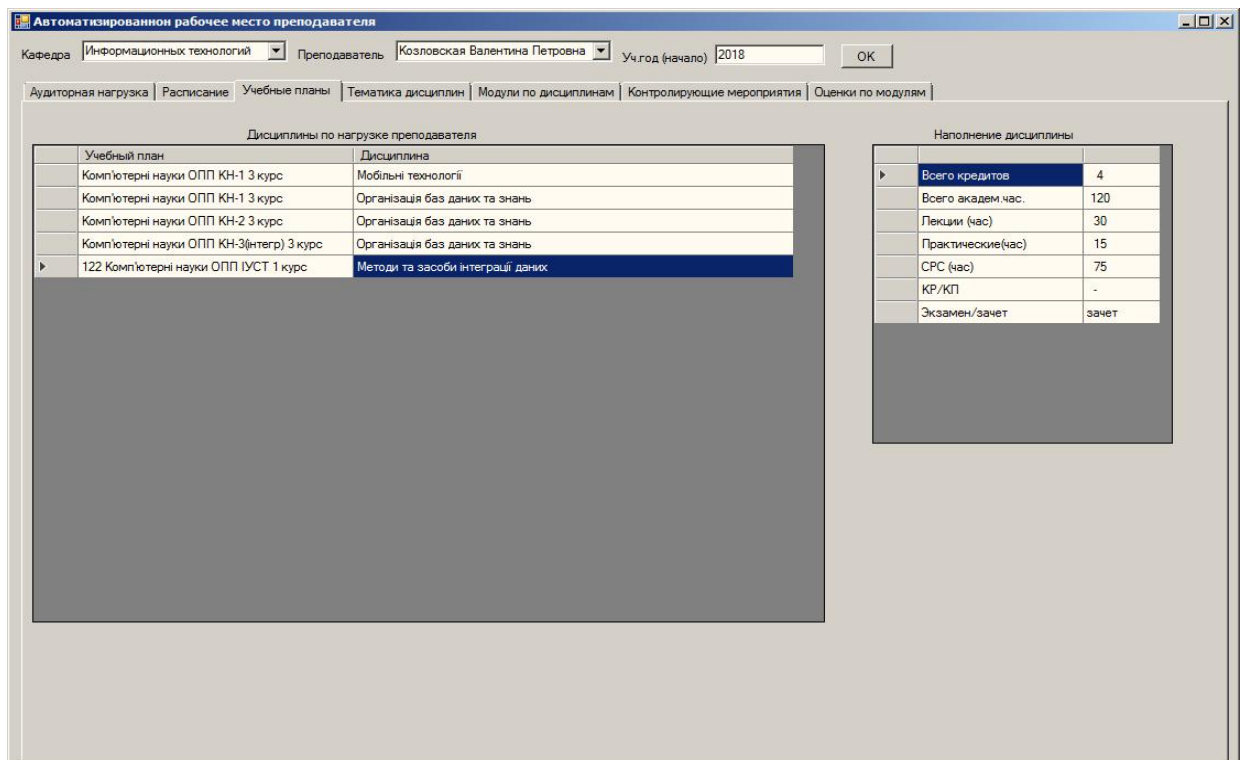


Рисунок 7.12 – Вкладка «Учебные планы»: наявність у плані дисципліни практичних занять та відсутність лабораторних робіт

Четверта вкладка «Тематика дисциплин» дозволяє переглянути теми дисциплін, які веде викладач. Спочатку користувач має вибрати дисципліни, тематику якої він хоче переглянути або змінити (рис. 7.13). Якщо для дисципліни не призначені теми в БД, вибірка буде порожньою, і користувач може лише додати тему дисципліни (рис. 7.14).

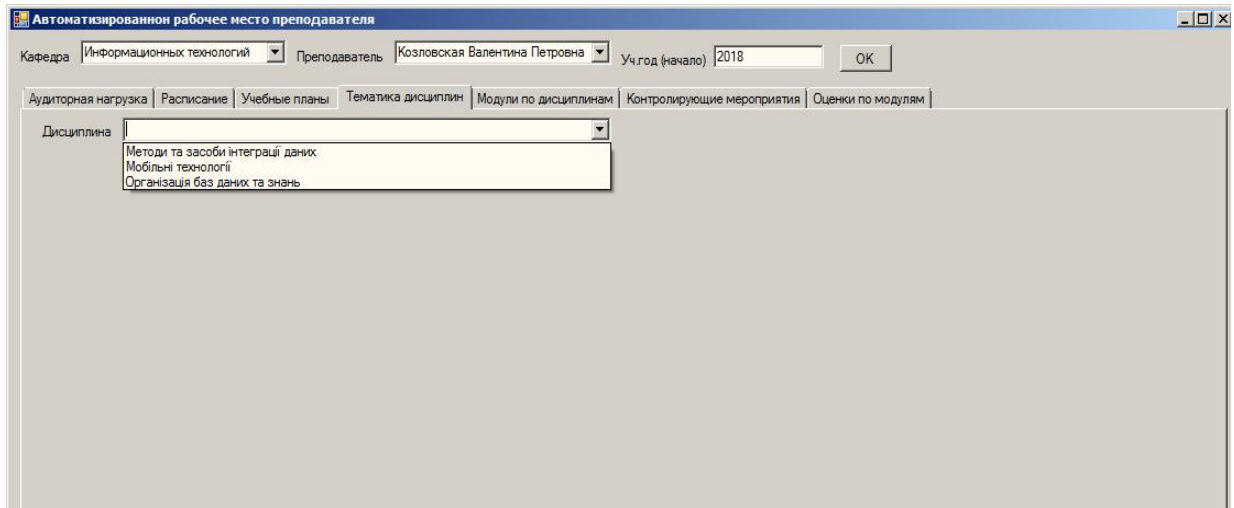


Рисунок 7.13 – Вкладка «Тематика дисциплин»: вибір дисципліни

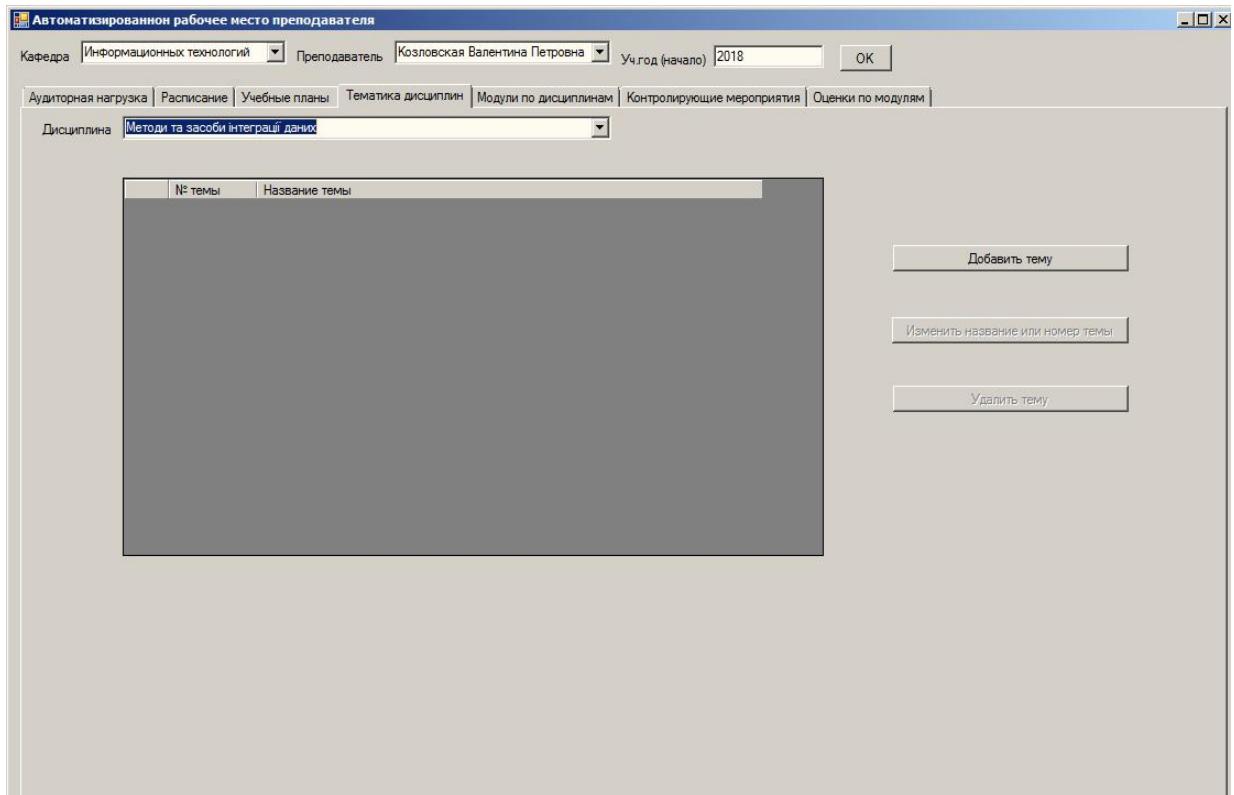


Рисунок 7.14 – Відсутність тем з дисципліни у БД

Якщо для дисципліни вже призначені теми, їх список відобразиться у таблиці на формі, та стануть доступними всі кнопки маніпулювання даними у таблиці (рис. 7.15). При натисканні кнопки «Добавить тему» під списком тем з'являються елементи, потрібні для додавання нового запису (рис. 7.16).

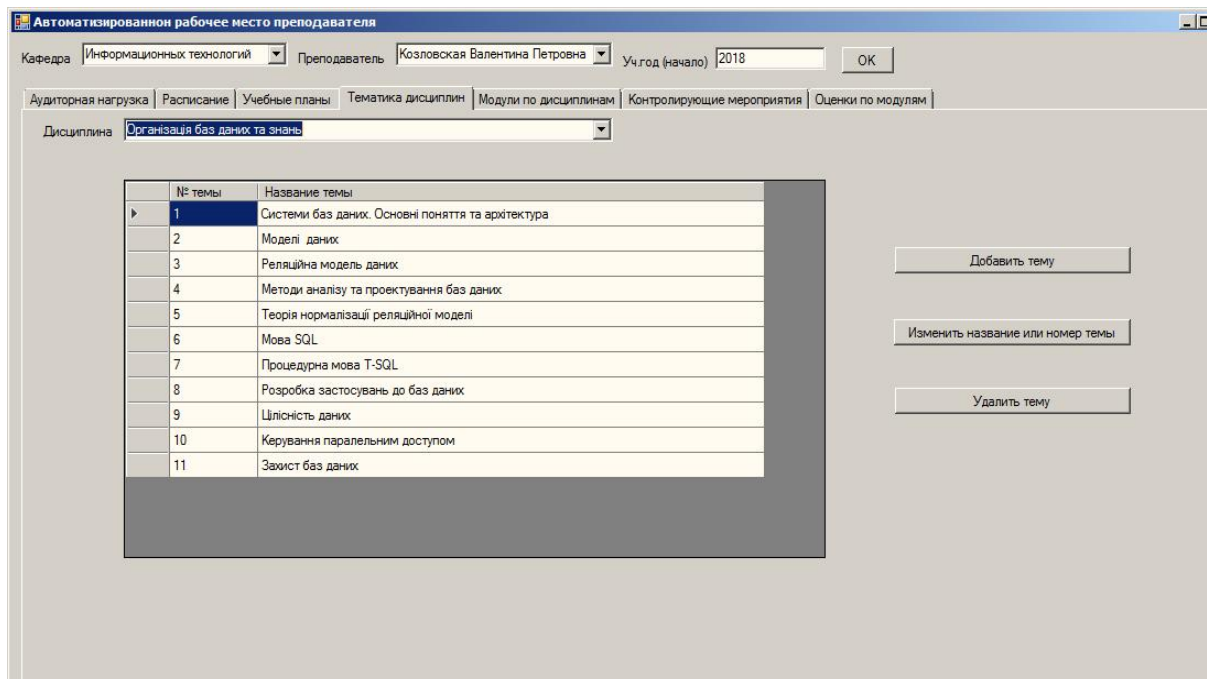


Рисунок 7.15 – Наличие тем с дисциплины у БД

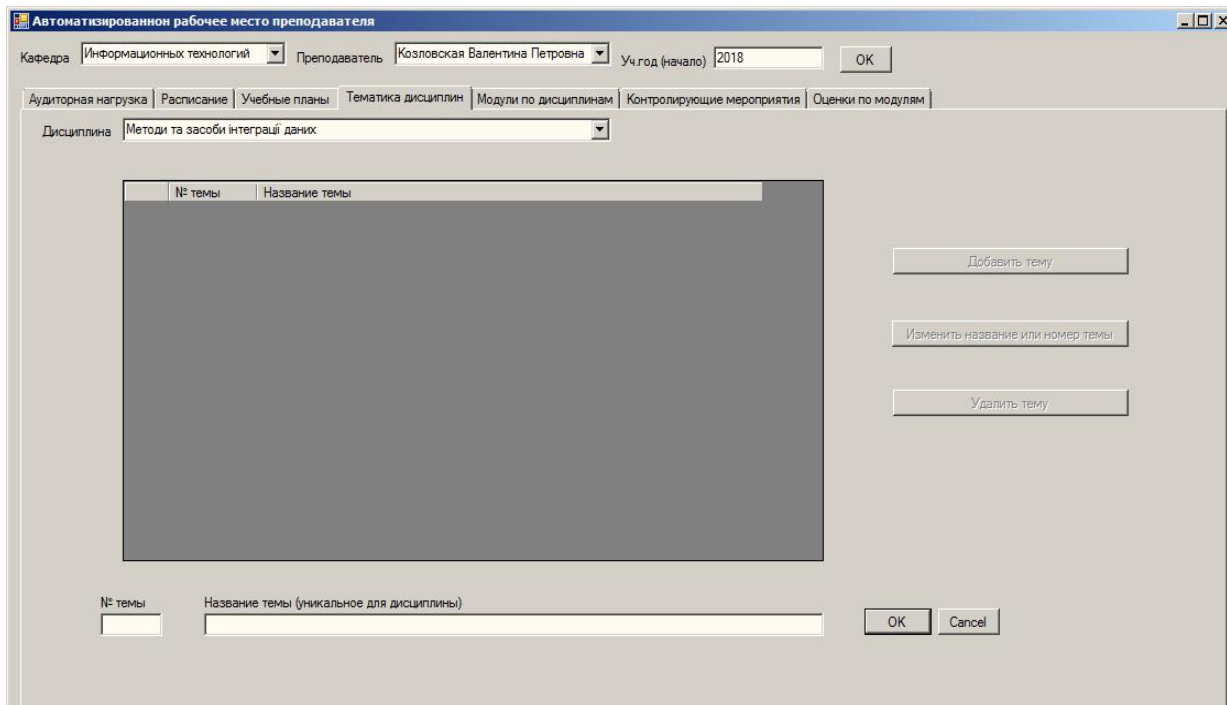


Рисунок 7.16 – Элементы формы для добавления нового запису у таблицу

При натисканні кнопки «Изменить название или номер темы» також стають видимими елементи, потрібні для вказаної дії (рис. 7.17).

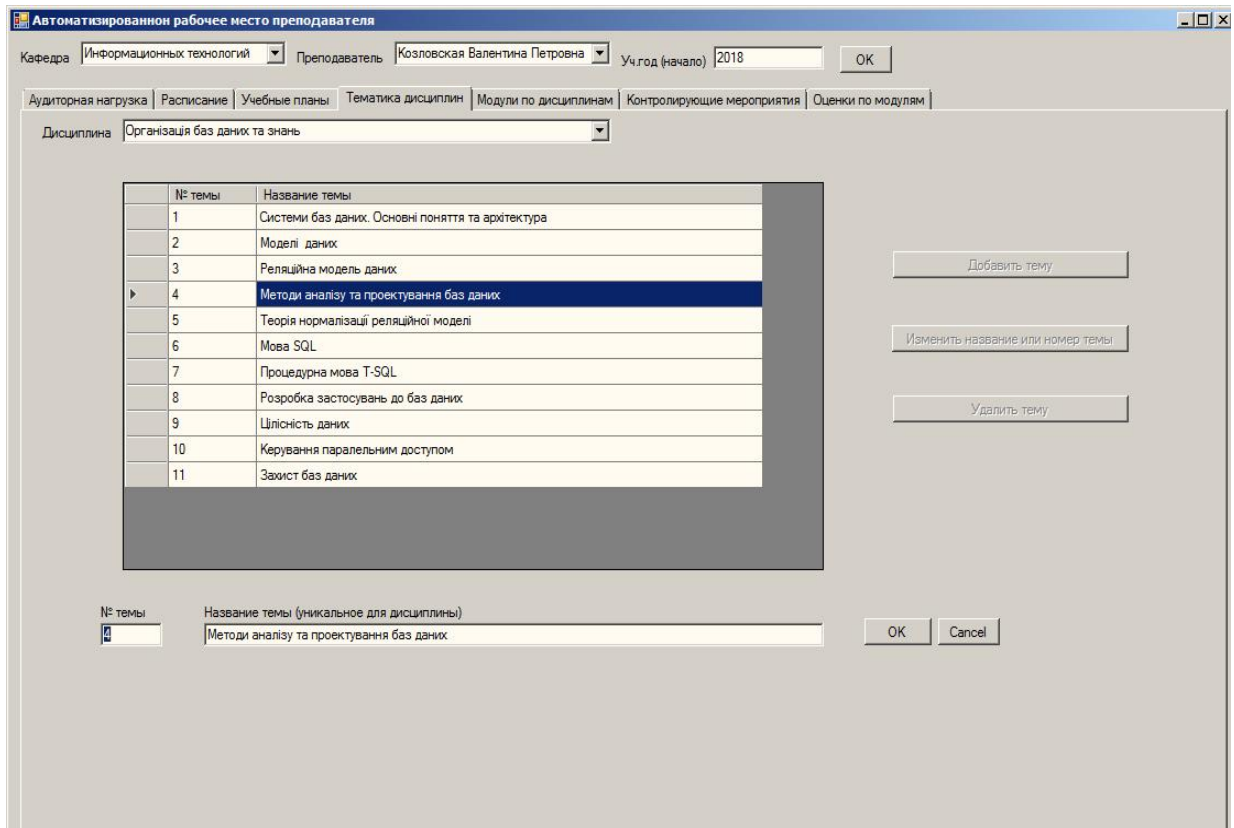


Рисунок 7.17 – Редагування теми дисципліни

При натисканні кнопки «Удалить тему» програма запитує підтвердження дії користувача (рис. 7.18). Якщо користувач натисне на «ОК», викликається збережена процедура БД для видалення записів в цій таблиці. Збережена процедура перевіряє, чи немає у підлеглих таблицях записів, що зв'язані з даними, що усуваються. Якщо такі записи є, програма знов потребує підтвердження на видалення записів з БД (рис. 7.19).

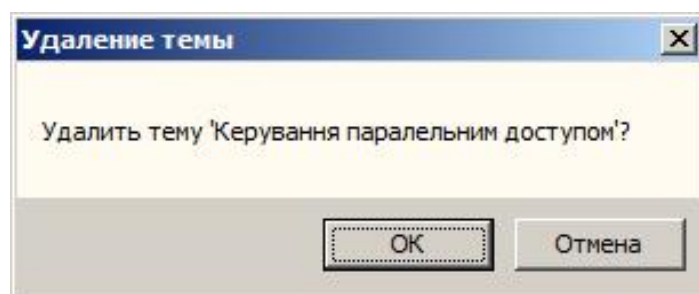


Рисунок 7.18 – Запит про необхідність усунення даних

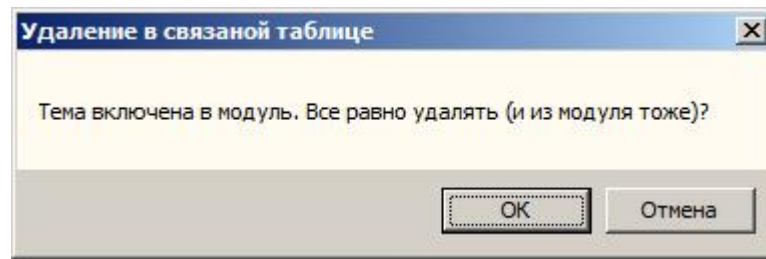


Рисунок 7.19 – Попередження про каскадне видалення даних

П'ята вкладка «Модулі по дисциплінам» містить відомості про змістовні модулі з дисциплін викладача. Спочатку користувач повинен також вибрати одну зі своїх дисциплін для перегляду модулів (рис. 7.20).

Якщо для вибраної дисципліни ще не додано модулів у БД, користувач може лише додати модуль (рис. 7.21).

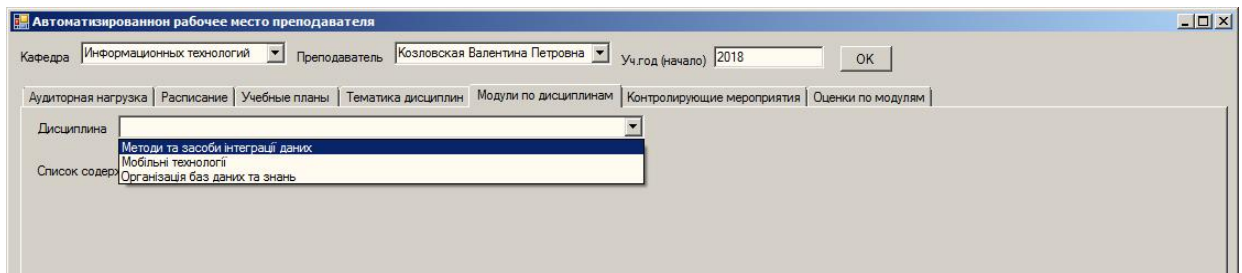


Рисунок 7.20 – Вкладка «Модули по дисциплінам»

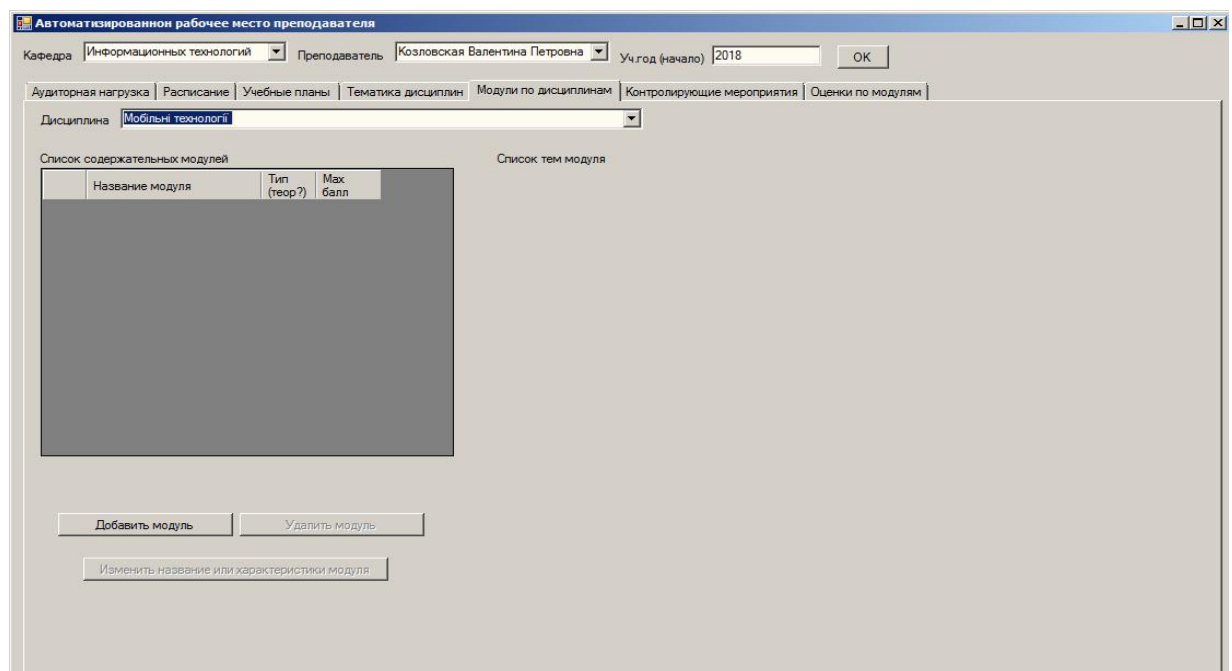


Рисунок 7.21 – Можливість додати модуль для дисципліни

При натисканні кнопки «Добавить модуль» відкривається вікно додавання модулю (рис. 7.22).

Рисунок 7.22 – Форма додавання змістовних модулів

Якщо в БД вже є записи про модулі для вибраної дисципліни, користувачеві надається перелік всіх наявних змістовних модулів даної дисципліни у лівій таблиці на формі програми (рис. 7.23).

Дисциплина: Организация баз данных та знань

| Название модуля | Тип (теор?) | Max балл |
|-----------------|-------------------------------------|----------|
| ЗМ-П1 | <input type="checkbox"/> | 20 |
| ЗМ-П2 | <input type="checkbox"/> | 20 |
| ЗМ-П3 | <input type="checkbox"/> | 20 |
| ЗМ-П1 | <input checked="" type="checkbox"/> | 38 |
| ЗМ-П2 | <input checked="" type="checkbox"/> | 28 |
| ЗМ-П3 | <input checked="" type="checkbox"/> | 34 |
| ЗМ-П3 | <input checked="" type="checkbox"/> | 40 |

Сумма баллов по всем модулям: **200**

| Название темы | Ауд.час | СРС(час) |
|---|---------|----------|
| Системы баз данных. Основные понятия та архитектура | 2 | 1 |
| Модели данных | 2 | 1 |
| Реляционная модель данных | 4 | 2 |
| Методы анализа та проектирования баз данных | 2 | 1 |
| Теория нормализации реляционной модели | 2 | 1 |

Всего (час): **12** **6**

| Название работы | Max балл | Ауд.(час) | СРС(час) |
|-----------------|----------|-----------|----------|
| КР-1 | 20 | 0 | 5 |

Всего за модуль: **20** **0** **5**

Рисунок 7.23 – Вид вкладки «Модули по дисциплинам» при наявності даних по модулям у БД

Під таблицею виводиться максимальна сума балів, яку може набрати студент при виконанні всіх модулів даної дисципліни. Також під таблицею є кнопки для маніпулювання даними у таблиці змістовних модулів. При натисканні кнопки «Изменить название или характеристики модуля» відкривається віно форми для зміни даних існуючого у БД модулю (рис. 7.24).

Рисунок 7.24 – Форма редагування змістовних модулів

Праворуч від списку модулів на вкладці головної форми знаходяться таблиці, в яких відображається зміст модулів (рис. 7.23, 7.25 – 7.27).

| Название модуля | Тип (теор?) | Max балл |
|-----------------|-------------------------------------|----------|
| ЗМ-П1 | <input type="checkbox"/> | 20 |
| ЗМ-П2 | <input type="checkbox"/> | 20 |
| ЗМ-П3 | <input type="checkbox"/> | 20 |
| ЗМ-П1 | <input checked="" type="checkbox"/> | 38 |
| ЗМ-П2 | <input checked="" type="checkbox"/> | 28 |
| ЗМ-П3 | <input checked="" type="checkbox"/> | 34 |
| ЗМ-13 | <input checked="" type="checkbox"/> | 40 |

| Название работы | Max балл | Ауд. час | СРС(час) |
|-----------------|----------|----------|----------|
| Курсовая работа | 40 | 0 | 20 |

Рисунок 7.25 – Видяг форми для практичного модулю

Зверху праворуч на формі знаходиться таблицю, у яку виводяться теми, що вивчаються на лекціях поточного лекційного змістовного модулю.

Під таблицею тем модулю виводяться два числа: сума аудиторних годин, відведених на вивчення тем модулю, та сума годин самостійного вивчення тем модулю (рис. 7.23).

Якщо поточний змістовний модуль у таблиці ліворуч є практичним, теми модулю можуть бути відсутніми (рис. 7.25 – 7.26).

| Название модуля | Тип (теор?) | Мак балл |
|-----------------|-------------------------------------|----------|
| ЗМ-П1 | <input type="checkbox"/> | 20 |
| ЗМ-П2 | <input type="checkbox"/> | 20 |
| ЗМ-П3 | <input type="checkbox"/> | 20 |
| ▶ ЗМ-П1 | <input checked="" type="checkbox"/> | 38 |
| ЗМ-П2 | <input checked="" type="checkbox"/> | 28 |
| ЗМ-П3 | <input checked="" type="checkbox"/> | 34 |
| ЗМ-П3 | <input checked="" type="checkbox"/> | 40 |

Сумма баллов по всем модулям **200**

| Название работы | Мак балл | Ауд.час | СРС(час) |
|---|----------|---------|----------|
| ▶ ПР1 Створення БД у MS SQL Server | 8 | 4 | 3 |
| ПР2 Прості запити до БД | 10 | 4 | 3 |
| ПР3 Складні запити до БД | 12 | 4 | 3 |
| ПР4 Маніпулювання даними у БД засобами мови SQL | 8 | 4 | 3 |

Всего за модуль **38 16 12**

Рисунок 7.26 – Наявність декількох складових практичного модулю

Також теми можуть бути відсутні, якщо користувач не заповнив у БД список тем модулю (рис. 7.27).

У лекційних модулів повинні бути перелічені теми курсу, що в робочій програмі дисципліни віднесені до цього модулю. Тому потрібно у базі даних додати теми у кожен лекційний модуль.

Для додавання нової теми дисципліни у модуль призначена кнопка «Добавить тему модулю». При натисканні на цю кнопку відкривається вікно форми для додавання теми до модулю (рис. 7.28).

На формі додавання теми для модулю виводиться назва дисципліни та назва модулю, для якого додається тема.

Рисунок 7.27 – Відсутність тем лекційних модуль для дисципліні у БД

Рисунок 7.28 – Форма для додавання теми для змістовного модулю

Користувач повинен вибрати тему дисципліни з випадного списку тем (рис. 7.29), а також ввести у текстові поля форми кількість аудиторних годин вивчення даної теми та кількість годин самостійного вивчення теми студентами (рис. 7.28).

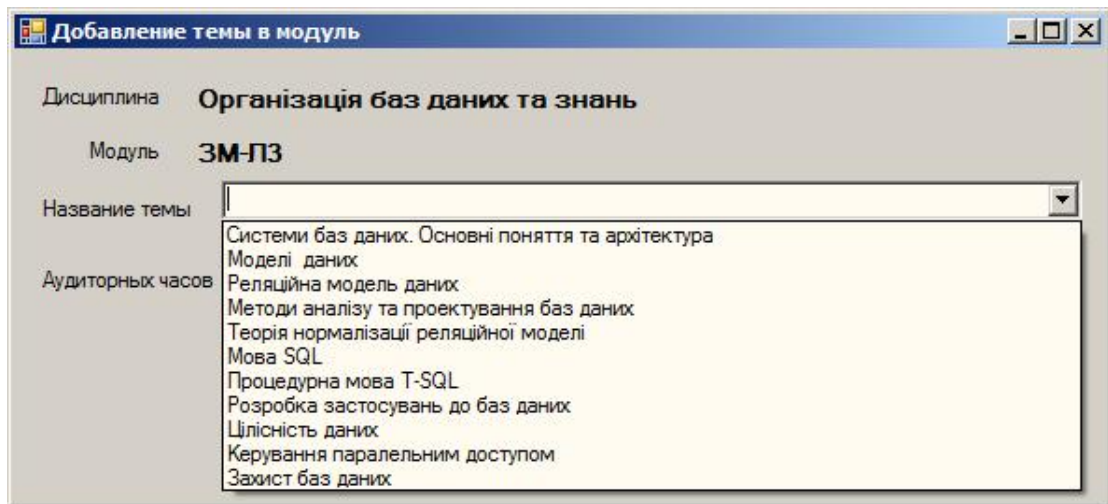


Рисунок 7.29 – Вибір теми дисципліни для додання її у змістовний модуль

Наявну тему модулю можна редагувати, якщо натиснути кнопку «Изменить тему или часы на тему у модуля» на формі (рис. 7.30).

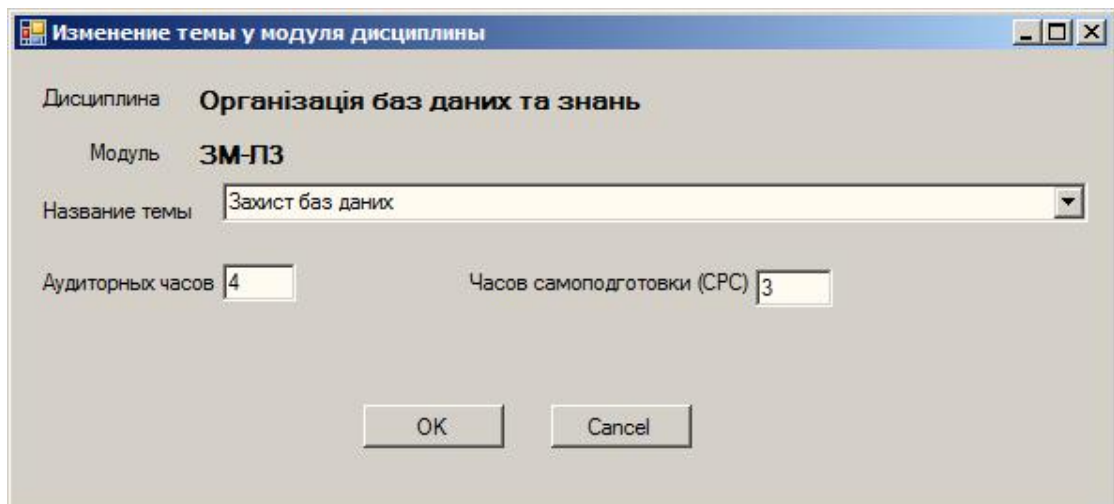


Рисунок 7.30 – Форма для редагування теми для змістовного модулю

Можна видалити тему модулю, якщо натиснути кнопку «Удалить тему у модуля». У переліку тем дисципліни тема зберігається, коли її видаляють з модулю.

Змістовний модуль може мати декілька складових робіт, які виконують студенти і отримують бали за їх виконання. Ці складові частини модулю виводяться у таблицю у правому нижньому куті форми програми (рис. 7.23, 7.25 – 7.27). Для лекційних змістовних модулів у якості цих складових частин модулю виступають контролюючі заходи для модулю, наприклад, усне опитування, або модульна контрольна робота.

Під таблицею виводяться сумарна кількість балів, що може отримати студент при виконанні всіх перелічених робіт по модулю, а також сумарні години, що відводяться на виконання всіх робіт для поточного модулю. Під цими даними знаходяться кнопки, за допомогою яких можна додавати, змінювати та видаляти складові роботи (контролюючі заходи) для модулю.

На шостій вкладці виводиться графік здачі контролюючих заходів з вибраної дисципліни (рис. 7.31).

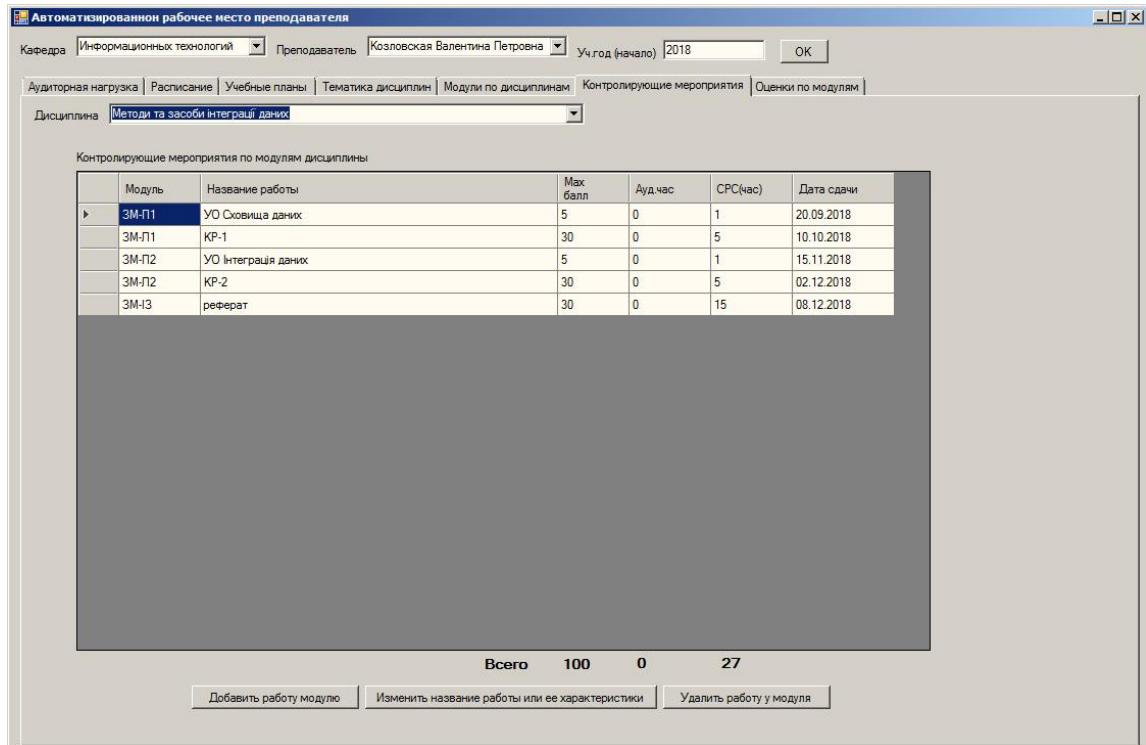


Рисунок 7.31 – Вкладка «Контролирующие мероприятия»

Для кожного контролюючого заходу виводиться назва модулю, до якого відноситься цей захід, назва роботи, кількість балів, що нараховується за виконання заходу, години підготовки до цього заходу та дата здачі. Під таблицею контролюючих заходів виводяться сумарні дані: сумарна кількість балів по всім контролюючим заходам, загальна кількість годин підготовки до всіх заходів.

Для лекційних модулів контролюючими заходами, як правило, є усне опитування та модульна контрольна робота. Для цих заходів не відводяться аудиторні години, тільки години самостійної підготовки студентів (рис. 7.31).

Для лабораторних робіт, виконання та захист яких також є контролюючими заходами, відводяться аудиторні години на виконання роботи та години самостійної підготовки для оформлення звіту (рис. 7.32).

| Модуль | Название работы | Макс балл | Ауд.час | СРС(час) | Дата сдачи |
|--------|---|-----------|---------|----------|------------|
| ЗМ-П1 | ПР1 Створення БД у MS SQL Server | 8 | 4 | 3 | 10.09.2018 |
| ЗМ-П1 | ПР2 Прості запити до БД | 10 | 4 | 3 | 20.09.2018 |
| ЗМ-П1 | ПР3 Складні запити до БД | 12 | 4 | 3 | 29.09.2018 |
| ЗМ-П1 | КР-1 | 20 | 0 | 5 | 05.10.2018 |
| ЗМ-П1 | ПР4 Маніпулювання даними у БД засобами мови SQL | 8 | 4 | 3 | 06.10.2018 |
| ЗМ-П2 | ПР5 Розробка БД та представлень до неї | 8 | 4 | 3 | 16.10.2018 |
| ЗМ-П2 | ПР6 Розробка серверного ПЗ | 12 | 4 | 3 | 30.10.2018 |
| ЗМ-П2 | ПР7 Створення застосування для роботи з БД у середовищі Visu... | 8 | 2 | 3 | 08.11.2018 |
| ЗМ-П2 | КР-2 | 20 | 0 | 5 | 10.11.2018 |
| ЗМ-П3 | ПР8 Використання збережених процедур БД у застосуванні | 10 | 6 | 3 | 17.11.2018 |
| ЗМ-П3 | ПР9 Створення форми для додавання записів у таблиці-довідники | 11 | 6 | 3 | 30.11.2018 |
| ЗМ-П3 | КР-3 | 20 | 0 | 5 | 05.12.2018 |
| ЗМ-П3 | ПР10 Створення форми для додавання записів у робочі таблиці | 13 | 7 | 3 | 10.12.2018 |
| ЗМ-П3 | Курсова робота | 40 | 0 | 20 | 12.12.2018 |

Рисунок 7.32 – Контролюючі заходами з лабораторними роботами

Понизу форми розташовані кнопки, що призначені для додавання, редагування та видалення контролюючих заходів. Для додавання та редагування контролюючих заходів використовується окрема форма програми (рис. 7.33).

Рисунок 7.33 – Форма для додавання та редагування контролюючих заходів

Остання форма програми «Оценки по модулям» призначена для внесення в базу даних та перегляду викладачем оцінок, що отримали студенти за виконання контролюючих заходів (рис. 7.34). Користувач повинен вибрати з випадних списків дисципліну, академічну групу, контролюючий захід, після цього на форму виводиться список студентів групи з оцінками за цей захід (рис. 7.35).

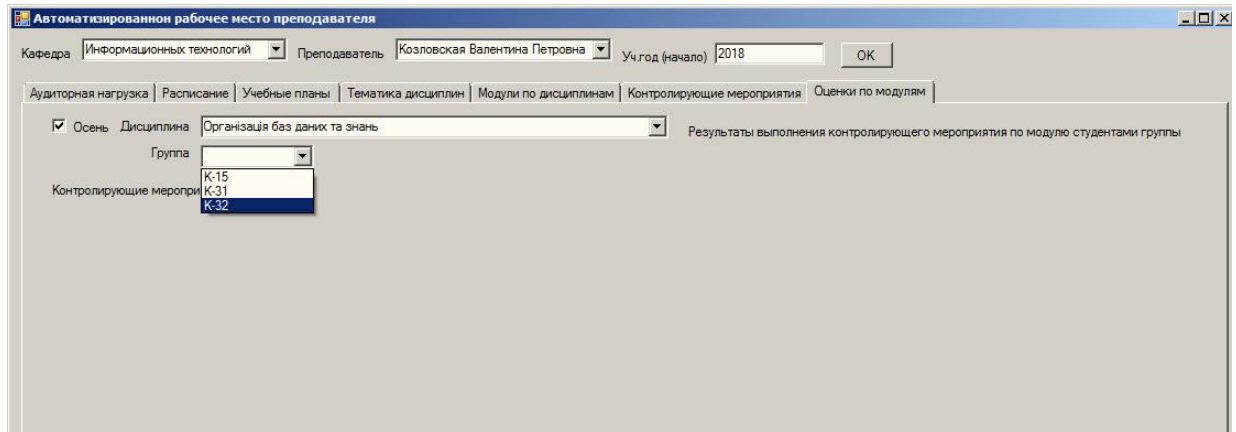


Рисунок 7.34 – Вкладка «Оценки по модулям»

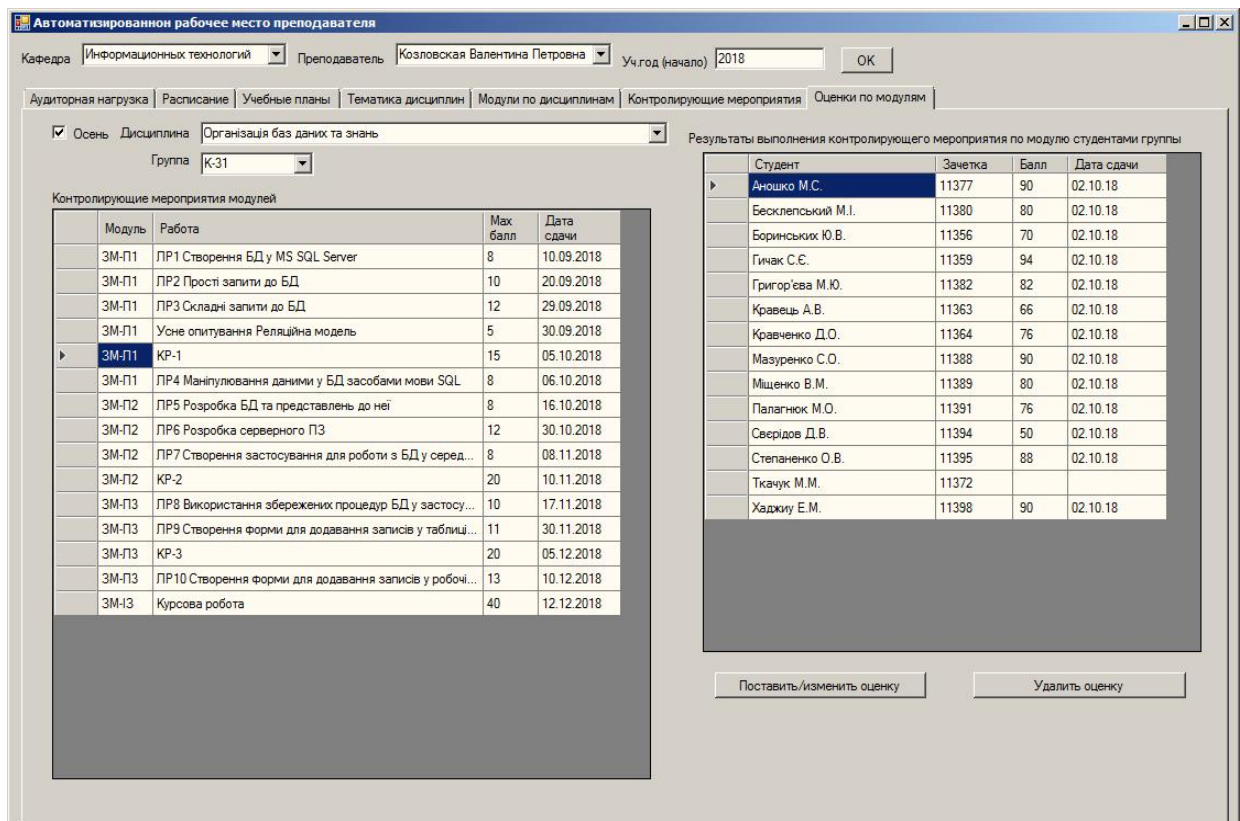


Рисунок 7.35 – Перегляд результатів здачі студентами контролюючого заходу

Якщо студент не здавав контролюючий захід, він є у списку студентів групи на формі, але не має оцінки та дати здачі. Для додавання оцінки використовується кнопка «Поставить/изменить оценку» у правому нижньому куті форми. При натисканні на цю кнопку під нею з'являються елементи форми, необхідні для додавання або редагування запису в таблиці результатів виконання контролюючого заходу (рис. 7.36).

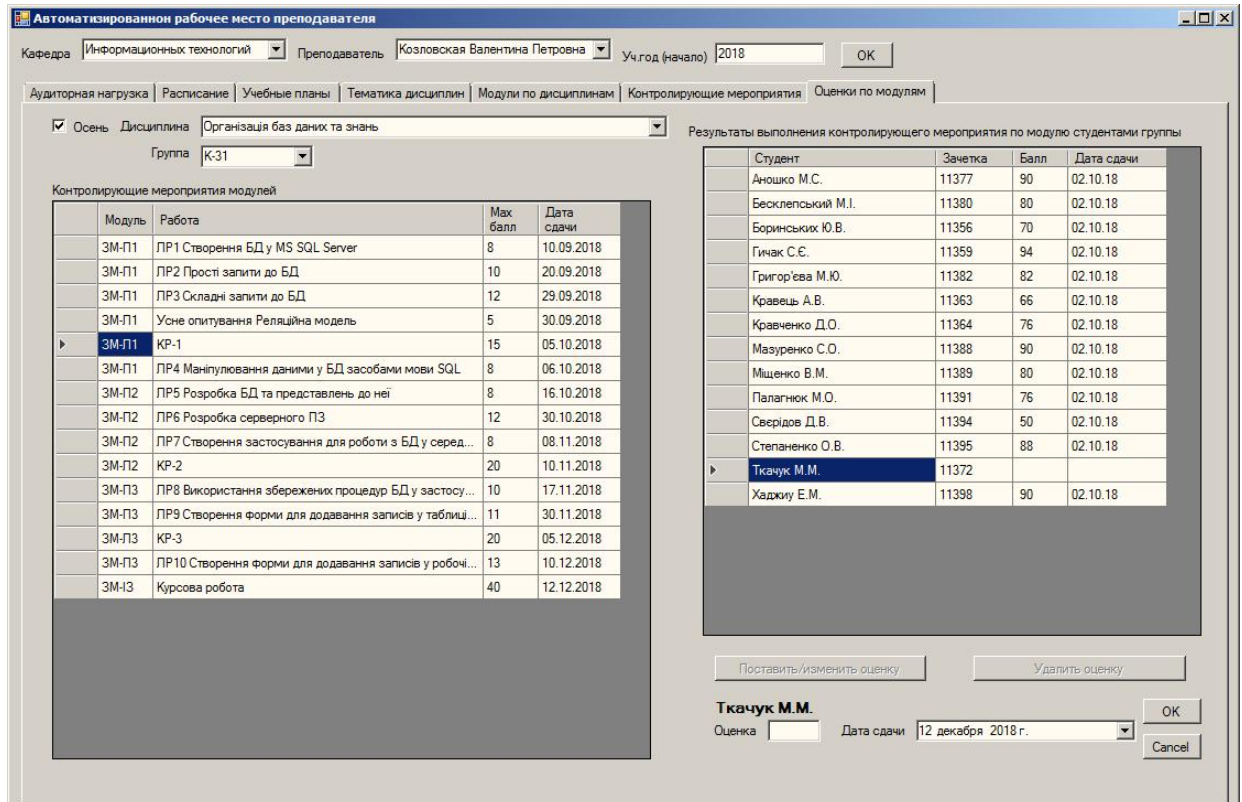


Рисунок 7.36 – Перегляд результатів здачі студентами контролюючого заходу

Оцінка ставиться студенту, що є поточним у списку студентів групи.

За наявності застосування виду «Органайзер студента», що розроблений у вигляді веб-застосування або мобільного застосування студенти університету мають можливість переглядати теми курсу, що необхідно вивчити для підготовки до модульної контрольної роботи або усного опитування, графік контролюючих заходів з дисциплін, оцінки, отримані за виконання контролюючих заходів.

ВИСНОВКИ

Дана робота є складовою частиною розробки інформаційної системи «Університет», яка призначена для автоматизації усіх стадій навчального процесу в університеті.

В роботі розглянуто інтеграцію у єдину ІС «Університет» окремих прикладних систем баз даних, що вже розроблялись в Одеському державно-му екологічному університеті при дипломному проектуванні та виконанні магістерських робіт, а саме інтеграції прикладних баз даних «Навантаження викладача» та «Розклад занять».

Інтеграція всіх прикладних систем баз даних, що відносяться до навчального процесу університету, у єдину інформаційну систему дозволить зменшити документопотік між підрозділами університету, спростить процес отримання різних статистичних даних щодо навчального процесу, дозволить різним групам користувачів легко отримувати різноманітну інформацію: контингенти студентів, списки навчальних планів факультетів, навчальне навантаження викладачів, розклади занять академічних груп та викладачів університету, завантаження аудиторного фонду університету.

Результатами виконання даної частини магістерської роботи є:

- розроблена концептуальна модель підсистеми «Викладач» у складі ІС «Навчальний процес університету»;
- відповідно до моделі реструктурована база даних: додано у схему 4 нових таблиці;
- розроблено серверне програмне забезпечення для підсистеми «Викладач»;
- розроблено серверне програмне забезпечення включає в себе 6 збережених процедур і 19 уявлень на мові T-SQL для СУБД MS SQL Server 2008;
- розроблено клієнтське застосування «Автоматизоване робоче місце викладача» у форматі офісного додатку до бази даних інформаційної системи «Навчальний процес університету»;
- клієнтське застосування дозволяє викладачам переглядати аудиторне навантаження викладача та розклад занять;
- застосування дозволяє користувачеві переглядати та оновлювати, що складають робочу програму дисципліни: змістовні модулі, контролюючі заходи

Програма «Автоматизоване робоче місце викладача» розроблялась як частина інформаційної системи «Навчальний процес університету», тому не вимагає введення великих масивів вихідних даних.

Необхідні дані від підрозділів університету, що потрібні для роботи викладача з інформаційною системою, повинні оновлюватись у базі даних інформаційної системи групами користувачів, що мають відповідні права на оновлення визначеного виду даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. Куликов Г. Г., Никулина Н. О., Речкалов А. В. Управление проектами на основе системного моделирования: Учебное пособие. – Уфа: УГАТУ, 2009. – 171 с.
2. Клиффорд Ф. Грей, Эри У. Ларсон. Управление проектами. – М: ДиС, 2007. – 608 с.
3. Мазур И. И., Шапиро В. Д, Ольдерогге Н. Г. Управление проектами: учеб. пособие. 2-е изд. М.: Омега-Л, 2004. – 664 с.
4. Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. М: «Финансы и статистика», 1998. – 98 с.
5. Томас Конноли, Каролин Бегг. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание.: Пер. с англ. М. : Издательский дом «Вильямс», 2003. – 1440 с.
6. Дейт К. Дж.. Введение в системы баз данных, 6-е издание: Пер. с англ. – К.; М.; СПб: Издательский дом «Вильямс», 2000 – 848 с.
7. Сравнение SQL баз данных. <http://webarty.net/databases/sravnenie-sql-baz-dannyh>
8. Мамаев Е.В. Microsoft SQL Server 2000. – СПб.: БХВ-Петербург, 2004. – 1260 с.
9. Дэвидсон Л. Проектирование баз данных на SQL Server 2000 / Л.Дэвидсон; пер. с англ. – М.:БИНОМ. Лаборатория знаний, 2003. – 680 с.
- 10.Роберт Виейра. Программирование баз данных Microsoft SQL Server 2005. Базовый курс. М.: «Диалектика», 2007. – с. 832.
- 11.Введение в MS SQL Server и T-SQL <https://metanit.com/sql/sqlserver/1.1.php>
- 12.Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7. RL: <https://metanit.com/sharp/tutorial/>
- 13.Руководство по программированию на C#. URL: <https://docs.microsoft.com/ru-ru/previous-versions/67ef8sbd>
- 14.Руководство по программированию на C#. URL: <https://docs.microsoft.com/ru-ru/previous-versions/67ef8sbd>
- 15.Руководство по C# - Часть 1. URL: https://professorweb.ru/my/csharp/charp_theory/level1/index.php