

Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ»

(за матеріалами навчально-методичного відділу КНТЕУ)

Студента 2 курсу, б групи, факультету
обліку, аудиту та інформаційних
систем, денної форми
122 «Комп'ютерні науки»

Остапчук
Богдан
Валерійович

Науковий керівник
доктор технічних наук,
професор кафедри комп'ютерних наук

Яловець
Андрій
Леонідович

Гарант освітньої програми
доктор фізико-математичних наук,
професор

Пурський
Олег
Іванович

Київ – 2019

Київський національний торговельно-економічний університет

Факультет обліку, аудиту та інформаційних систем
Кафедра комп'ютерних наук
Спеціальність 122 "Комп'ютерні науки"

Затверджую

Зав. кафедри Пурський Олег Іванович

«__» _____ 2019 р.

Завдання на випускний кваліфікаційний проект студенту

Остапчуку Богдану Валерійовичу

1. Тема випускного кваліфікаційного проекту Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ

Затверджена наказом КНТЕУ від 07 листопада 2018 р. № 4181

2. Строк здачі студентом закінченого проекту _____

3. Цільова установка та вихідні дані до проекту

Мета проекту - обґрунтування та створення СУБД з урахуванням необхідного функціоналу та сучасного підходу.

Об'єкт дослідження - є процес створення системи управління бази даних

Предмет дослідження - засоби використовувані для створювання СУБД.

4. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)
В першому розділі було зібрано інформацію про досліджуваний об'єкт, розглянуто мову маніпуляції даними та різні види класифікації СУБД.

Другий розділ випускного кваліфікаційного проекту присвячений практичним аспектам реалізації інформаційної системи управління базою даних. Детально описані вимоги, проаналізовані мови серверного веб-програмування.

У третьому розділі розроблено інформаційну систему управління базою даних, проведено тестування системи.

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів роботи	
		За планом	фактично

1	2	3	4
1	<i>Вибір теми випускного кваліфікаційного проекту</i>	01.11.2018	01.11.2018
2	<i>Розробка та затвердження завдання на випускний кваліфікаційний проект</i>	15.11.2018	15.11.2018
3	<i>Вступ</i>	01.06.2019	01.06.2019
4	<i>Розділ 1.</i>	25.06.2019	25.06.2019
5	<i>Розділ 2.</i>	02.09.2019	02.09.2019
6	<i>Підготовка статті у збірник наукових статей магістрів</i>	09.09.2019	09.09.2019
7	<i>Розділ 3.</i>	21.10.2019	21.10.2019
8	<i>Висновки</i>	01.11.2019	01.11.2019
9	<i>Здача випускного кваліфікаційного проекту на кафедрі науковому керівнику</i>	05.11.2019	05.11.2019
10	<i>Попередній захист випускного кваліфікаційного проекту</i>	20.11.2019	20.11.2019
11	<i>Виправлення зауважень, зовнішнє рецензування випускного кваліфікаційного проекту</i>	22.11.2019	22.11.2019
12	<i>Представлення готового зшитого випускного кваліфікаційного проекту на кафедрі</i>	25.11.2019	25.11.2019

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	9
РОЗДІЛ 1. Теоретичні аспекти розробки інформаційної системи управління базою даних	11
1.1 Інформація про досліджуваний об'єкт	11
1.2 Мова маніпуляції даними та класифікація СУБД	12
1.3. Основні функції систем управління базами даних.....	16
ВИСНОВКИ ДО РОЗДІЛУ 1	18
РОЗДІЛ 2. Аналітична підготовка до розробки інформаційної СУБД НМВ КНТЕУ	19
2.1. Аналіз даних навчально-методичного відділу КНТЕУ	19
2.2. На основі аналізу – побудова структури БД	21
2.3. Аналіз та вибір засобів реалізації поставленої мети	25
ВИСНОВКИ ДО РОЗДІЛУ 2	30
РОЗДІЛ 3. Розробка інформаційної системи управління БД НМВ КНТЕУ	31
3.1. Створення інтерфейсу СУБД.....	31
3.2. Програмування серверної частини	37
3.3. Тестування СУБД	49
ВИСНОВКИ ДО РОЗДІЛУ 3	53
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56

					<i>КНТЕУ-122-2019</i>		
<i>Зм.</i>	<i>Аркуш</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Зав. кафедрою</i>		<i>Пурський О.І.</i>		<i>20.11.2019</i>	<i>Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ</i>	<i>Сторінка</i>	<i>Сторінок</i>
<i>Керівник</i>		<i>Яловець А.Л.</i>		<i>20.11.2019</i>		<i>5</i>	<i>71</i>
<i>Гарант</i>		<i>Пурський О.І.</i>		<i>20.11.2019</i>		<i>Кафедра комп'ютерних наук</i>	
<i>Розробив</i>		<i>Остапчук Б.В.</i>		<i>20.11.2019</i>			
<i>Перевірів</i>		<i>Яловець А.Л.</i>		<i>20.11.2019</i>	<i>Вступ</i>	<i>бм</i>	

АНОТАЦІЯ

Тема: «Розробка програмного захисту інформації підприємства».

Обсяг дипломної роботи 71 сторінки, на яких розміщені 11 рисунків і 4 таблиця. При написанні диплома використовувалося 22 джерела.

Ключові слова: система управління базою даних, база даних, сервер, модуль, клас.

Об'єктом дослідження є *навчально-методичного відділу КНТЕУ*.

Предметом дослідження є засоби використовувані для СУБД.

У дипломну роботу входить вступ, три розділи, висновки по написаним розділам, висновки та пропозиції.

У вступі розкривається актуальність дослідження за обраним напрямом, ставиться проблема, мета і завдання дослідження, визначаються об'єкт, предмет наукових пошуків, мета і завдання.

У першому розділі було зібрано інформацію про досліджуваний об'єкт, розглянуто мову маніпуляції даними та різні види класифікації СУБД..

У другому розділі розглянуто практичні аспекти реалізації інформаційної системи управління базою даних. Детально описані вимоги, проаналізовані мови серверного веб-програмування.

У третьому розділі здійснюється практична реалізація інформаційної системи управління базою даних, проведено тестування системи.

Висновок присвячено основним результатам і пропозиціям щодо включення модулю до безпеки інформації підприємства.

ABSTRACT

Theme: "Development of software protection of enterprise information".

The amount of the thesis 71 pages, which contains 11 figures and 1 table. When writing a diploma, 22 sources were used.

Keywords: database management system, database, server, module, class.

The object of study is the teaching and methodological department of KNTEU.

The subject of the study are the tools used for the DBMS.

The diploma thesis includes introduction, three sections, conclusions on written sections, conclusions and proposals.

The introduction reveals the relevance of the study in the chosen direction, sets the problem, purpose and objectives of the study, identifies the object, the subject of scientific research, the purpose and objectives.

The first section gathers information about the object under study, examines the language of data manipulation, and different types of DBMS classification.

The second section discusses the practical aspects of implementing a database management information system. Requirements, server web programming languages are described in detail.

In the third section, the practical implementation of the database management information system is carried out, the system is tested.

The conclusion is devoted to the main results and proposals for the inclusion of the module in the security of enterprise information.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

1. ІС — Інформаційна система.
2. БД — База даних.
3. СУБД — Система управління базами даних.
4. SQL — Structured Query Language.
5. PHP — Hypertext Preprocessor.
6. JS – javascript.
7. DBMS - Database Management System.
8. DML - Data Manipulation Language (Мова маніпулювання даними).
9. ЕОМ - електронно-обчислювальна машина.
10. ОС – операційна система.
11. ЦП – центральний процесор.
12. API – прикладний програмний інтерфейс.

					КНТЕУ-122-2019	Аркуш
						8
Зм.	Аркуш	№ документу	Підпис	Дата		

ВСТУП

В сучасному світі розробляються інформаційні системи для обслуговування різних систем діяльності, систем управління господарськими і технічними об'єктами, модельні комплекси для наукових досліджень, системи автоматизації проектування і виробництва, всілякі тренажери і навчальні системи. Однією з важливих передумов створення таких систем стала можливість оснащення їх «пам'яттю» для накопичення, зберігання та систематизація великих обсягів даних.

Дослідження і розробки, пов'язані з проектуванням, створенням і експлуатації баз даних, а також необхідних для цих цілей мовних і програмних інструментальних засобів привели до появи самостійної гілки інформатики, що отримала назву системи управління даними.

Такі програмні комплекси виконують досить складний набір функцій, пов'язаний з централізованими управліннями, даними в базі даних інтерфейсів всієї сукупності її користувачів. По суті, система управління базами даних служить посередником між користувачами і базою даних.

В даний час розроблені і використовуються на персональних комп'ютерах близько двадцяти систем управління базами даних. Вони представляють користувачеві зручні засоби інтерактивної взаємодії з БД і мають розвинену мову програмування

Актуальність теми полягає в тому, щоб розробити цільову, направлену систему управління базою даних (СУБД). Значущим фактором є скорочення часу відклику сервера, написання цільових функцій обробки даних, створення зручного та необхідного функціоналу для адміністратора БД.

Об'єктом дослідження є процес створення системи управління бази даних.

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедри		Пурський О.І.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка	Сторінок
Керівник		Яловець А.Л.		20.11.2019		9	71
Гарант		Пурський О.І.		20.11.2019		Кафедра комп'ютерних наук бм	
Розробив		Остапчук Б.В.		20.11.2019			
Перевірив		Яловець А.Л.		20.11.2019			

Предметом дослідження являються засоби використовувані для створення СУБД.

Мета дослідження полягає в обґрунтуванні та створенні СУБД з урахуванням необхідного функціоналу та сучасного підходу.

Для досягнення поставленої мети дослідження головними завданнями є:

- Аналіз сучасних СУБД.
- Визначення основоположних функцій.
- Обрання методу реалізації.

					КНТЕУ-122-2019	Аркуш
						10
Зм.	Аркуш	№ документу	Підпис	Дата		

РОЗДІЛ 1. Теоретичні аспекти розробки інформаційної системи управління базою даних

1.1 Інформація про досліджуваний об'єкт

Система управління базою даних - це комплекс програмних засобів, призначених для створення структури нової бази, наповнення її вмістом, редагування вмісту і візуалізації інформації. Під візуалізацією інформації бази розуміється відбір відображуваних даних відповідно до заданого критерію, їхнє упорядкування, оформлення і наступна видача на пристрій виведення або передача по каналах зв'язку.

Сутність - відображення об'єкта в пам'яті людини або комп'ютера.

Параметр - конкретне значення будь-якого з властивостей об'єкта.

Атрибут - конкретне значення будь-якого з властивостей сутності.

Таблиця - деяка регулярна структура, що складається з кінцевого числа записів (рядків). Як правило, в базах даних використовуються двовимірні масиви (матриці).

Запис - це один рядок таблиці (або кількох таблиць), повністю описує одну сутність. Кожен запис складається з кінцевого числа полів.

Поле - це один елемент запису, в якому зберігається конкретне значення атрибута.

Ключовим елементом даних (ключем) називається такий атрибут, за значенням якого можна визначити значення інших атрибутів.

Первинний ключ - це атрибут або група атрибутів, які однозначно визначають кожен запис в таблиці. Первинний ключ завжди повинен бути унікальним, тобто його значення не повинні повторюватися.

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедрою		Пурський О.І.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка	Сторінок
Керівник		Яловець А.Л.		20.11.2019		11	71
Гарант		Пурський О.І.		20.11.2019		Кафедра	
Розробив		Остапчук Б.В.		20.11.2019	Теоретичні аспекти розробки інформаційної системи управління базою даних	комп'ютерних наук	
Перевірив		Яловець А.Л.		20.11.2019		бм	

Альтернативний ключ - це відмінні від первинного ключа атрибут або група атрибутів, які також однозначно визначають кожен запис в таблиці. Наприклад: сутність «Працівник» має атрибути: ідентифікатор працівника, прізвище, ім'я, по батькові, посада, оклад. Первинним ключем призначимо поле «Ідентифікатор працівника». Альтернативним ключем призначимо групу полів «Прізвище», «Ім'я», «По батькові».

Процедура - це додаток (програма), яка об'єднує запити користувача і процедурну логіку і зберігається в базі даних.

Правило - це логічне умова, визначальне значення одного атрибута в залежності від значення іншого атрибута (або групи атрибутів).

За допомогою правила контролюється достовірність інформації, що вводить.

Обмеження - це логічна умова, що накладає обмеження (інтервал допустимих значень) на значення атрибута.

1.2 Мова маніпуляції даними та класифікація СУБД

Маніпулювання даними це пошук інформації, що зберігається в базі даних, додавання нової інформації в базу даних, видалення інформації з бази даних, модифікація інформації, що зберігається в базі даних.

Мова обробки даних (DML) - це мова, яка дозволяє користувачам отримувати доступ або маніпулювати даними, як організовані відповідною моделлю даних. В основному це два типи:

Процедурні DML вимагають від користувача уточнити, які дані потрібні та як їх отримати.

Деклараційні DML (також звані непроцедурними DML) вимагають від користувача вказати, які дані потрібні, не вказуючи, як їх отримати.

Декларативні DML зазвичай легше вивчити та використовувати, ніж процедурні DML.

					КНТЕУ-122-2019	Аркуш
						12
Зм.	Аркуш	№ документу	Підпис	Дата		

Однак, оскільки користувачеві не потрібно вказувати, як отримати дані, система бази даних має розробити ефективний засіб доступу до даних. Компонент DML SQL мова непроцедурна.

Запит - це звернення, що вимагає отримання інформації. Частина DML, що відповідає за пошук інформації називається мовою запиту.

Цей запит мовою SQL знаходить ім'я клієнта, чий ідентифікатор клієнта є 192837465:

«SELECT customer.customer-name FROM customer WHERE customer.customer-id = 192-83-7465»

Запит вказує, що необхідно вибрати ім'я клієнту (customer.customer-name), якому надано ідентифікатор клієнта(customer.customer-id)192837465. Запити можуть включати інформацію з більш ніж однієї таблиці.

Мета мови маніпуляції даними – це дозволити людині ефективно взаємодіяти з системою.

1) Класифікація СУБД за видами програм

Під визначення СУБД може потрапити будь-який програмний продукт, здатний підтримувати процеси проектування, адміністрування та використання бази даних, тому була розроблена класифікація СУБД за видами програм:

1.1) повнофункціональні - найчисленніші і потужні за своїми можливостями програми, наприклад Microsoft Access, Microsoft FoxPro, Clarion Database Developer і ін.;

1.2) сервери баз даних - застосовуються для організації центрів обробки даних в мережах ЕОМ. Серед них програми Microsoft SQL Server, NetWare SQL фірми Novell;

1.3) клієнти баз даних - різні програми (повнофункціональні СУБД, електронні таблиці, текстові процесори і т. Д.), Що забезпечують більшу продуктивність обчислювальної мережі, якщо клієнтська і серверна частини бази даних будуть зроблені однією фірмою, але така умова не є обов'язковим.

					КНТЕУ-122-2019	Аркуш
						13
Зм.	Аркуш	№ документа	Підпис	Дата		

2) Класифікація СУБД з вигляду застосування

По виду застосування СУБД поділяються на персональні і розраховані на багато користувачів.

Персональні СУБД (наприклад, Visual FoxPro, Paradox, Access) використовуються при проектуванні персональних баз даних і недорогих додатків, що працюють з ними, які, в свою чергу, можуть застосовуватися в якості клієнтської частини розрахованої на багато користувачів СУБД.

Розраховані на багато користувачів СУБД (наприклад, Oracle та Informix) складаються з сервера баз даних і клієнтської частини і здатні працювати з різними типами ЕОМ і ОС різних фірм-виробників.

Найчастіше інформаційні системи будуються на основі архітектури клієнт-сервер, в яку входять обчислювальна мережа і розподілена база даних. Обчислювальна мережа використовується для організації наукової роботи на ПК і в мережах. Розподілена база даних складається з багатокористувацької бази даних, розміщеної на комп'ютері-сервері, і персональної бази даних, яка перебуває на робочих станціях. Сервер бази даних здійснює виконання основного обсягу обробки даних.

3) Класифікація СУБД за способом доступу до бази даних

3.1) Файл-серверні СУБД. У файл-серверних СУБД файли даних розташовуються централізовано на файл-сервері СУБД. Ядро СУБД розташовується на кожному клієнтському комп'ютері. Доступ до даних здійснюється через локальну мережу. Синхронізація читань і оновлень здійснюється за допомогою файлових блокувань. Перевагою цієї архітектури є низьке навантаження на ЦП сервера, а недоліком - високе завантаження локальної мережі. Клієнт-серверні СУБД. Такі СУБД складаються з клієнтської частини (яка входить до складу прикладної програми) і сервера СУБД. Клієнт-серверні СУБД, на відміну від файл-серверних, забезпечують розмежування доступу між користувачами і мало завантажують мережу і клієнтські машини. Сервер є

					КНТЕУ-122-2019	Аркуш
						14
Зм.	Аркуш	№ документу	Підпис	Дата		

зовнішньою по відношенню до клієнта програмою, і по потребі його можна замінити іншим. Недолік клієнт-серверних СУБД в самому факті існування сервера СУБД (що погано для локальних програм - в них зручніше вбудовуються СУБД) і великих обчислювальних ресурсах, споживаних сервером.

3.2) Вбудовувані СУБД. Вбудована СУБД - бібліотека, яка дозволяє уніфікованим чином зберігати великі обсяги даних на локальній машині. Доступ до даних може відбуватися через SQL або через особливі функції СУБД. Вбудовувані СУБД швидше звичайних клієнт-серверних і не вимагають установки сервера, тому затребувані в локальному ПЗ, яке має справу з великими обсягами даних.

4) За типом керованої бази даних СУБД поділяються на:

4.1) Ієрархічні СУБД - підтримують деревоподібну організацію інформації. Зв'язки між записами виражаються у вигляді відносин предок / нащадок, а у кожного запису є рівно один батьківський запис.

4.2) Мережеві СУБД - мережева модель розширює ієрархічну модель СУБД, дозволяючи групувати зв'язки між записами в безлічі. З логічної точки зору зв'язок - це не сам запис. Зв'язки лише висловлюють відносини між записами. Як і в ієрархічній моделі, зв'язки ведуть від батьківської записи до дочірньої, але на цей раз підтримується множинне спадкування.

4.3) Реляційні СУБД. У порівнянні з розглянутими вище моделями реляційна модель вимагає від сервера СУБД набагато більш високого рівня складності. У ній робиться спроба позбавити програміста від виконання рутинних операцій по управлінню даними, такі характерні для ієрархічної і мережної моделей.

У реляційної моделі база даних являє собою централізоване сховище таблиць, що забезпечує безпечний одночасний доступ до інформації з боку багатьох користувачів. У рядках таблиць частина полів містить дані, що стосуються безпосередньо до запису, а частина - посилання на записи інших таблиць. Таким чином, зв'язки між записами є невід'ємною властивістю реляційної моделі.

					КНТЕУ-122-2019	Аркуш
						15
Зм.	Аркуш	№ документу	Підпис	Дата		

1.3. Основні функції систем управління базами даних

1. Безпосереднє управління даними у зовнішній пам'яті.

Функція безпосереднього управління даними у зовнішній пам'яті включає надання необхідних структур зовнішньої пам'яті як для зберігання даних, які безпосередньо входять в базу даних, так і для службових цілей, наприклад для прискорення доступу до даних в деяких випадках (зазвичай для цього використовують індекси). Причому користувачам бази даних в загальному випадку не потрібно знати, чи застосовує СУБД файлову систему, а якщо застосовує, як організовані файли. Зазвичай СУБД підтримує власну систему іменування об'єктів бази даних. Залежно від способу реалізації СУБД може або скористатися наявними можливостями існуючих файлових систем, або працювати з пристроями зовнішньої пам'яті на низькому рівні.

2. Управління буферами оперативної пам'яті.

Обсяг інформації, що зберігається в базі даних, з якою працює СУБД, зазвичай досить великий і практично завжди перевищує доступний об'єм оперативної пам'яті. При цьому час доступу до даних, що зберігаються в оперативній пам'яті, істотно менше, ніж до даних, що зберігаються на пристроях зовнішньої пам'яті. Очевидно, що якщо при зверненні до будь-якого елементу даних потрібно обмін із зовнішньою пам'яттю, то вся система буде працювати зі швидкістю функціонування пристрою зовнішньої пам'яті.

Підвищення швидкості обміну даними можна досягти, використовуючи буферизацію даних в оперативній пам'яті. При цьому навіть якщо операційна система виконує загальносистемну буферизацію, цього недостатньо для цілей СУБД, у якій набагато більше інформації, що дозволяє судити про корисність буферизації тієї чи іншої частини бази даних. Тому в СУБД зазвичай підтримується власний набір буферів оперативної пам'яті з власним механізмом заміни буферів.

3. Управління транзакціями.

Транзакцією називається послідовність операцій над базою даних, що розглядаються СУБД як єдине ціле. Якщо всі операції успішно виконані, то

					КНТЕУ-122-2019	Аркуш
						16
Зм.	Аркуш	№ документу	Підпис	Дата		

транзакція також вважається успішно виконаним, і СУБД фіксує всі зміни даних, зроблені цією транзакцією (тобто заносить зміни в зовнішню пам'ять). Якщо ж хоча б одна операція транзакції закінчується невдачею, то транзакція вважається невиконаним, і проводиться її відкат зі скасуванням усіх змін даних, вироблених в ході виконання транзакції, і поверненням бази даних до стану до початку виконання транзакції. Управління транзакціями необхідне для підтримки логічної цілісності бази даних.

4. Протоколювання.

Одним з основних вимог до СУБД є надійність зберігання даних у зовнішній пам'яті. Під надійністю зберігання розуміється те, що СУБД повинна бути в змозі відновити останній узгоджений стан БД після будь-якого апаратного або програмного збою.

5. Підтримка мов баз даних.

Для роботи з інформацією, що зберігається в базі даних, використовуються спеціальні мови, що носять загальну назву мов баз даних. Тобто, система повинна мати функціонал виконання запитів на мові баз даних.

					КНТЕУ-122-2019	Аркуш
						17
Зм.	Аркуш	№ документу	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 1

В першому розділі було зібрано інформацію про досліджуваний об'єкт та визначено його види. Також, розглянуто мову маніпуляції даними, наведено приклад не процедурного запиту DML. Даний тип буде використано для створення зв'язку між СУБД та БД навчально-методичного відділу (НМВ). Вся процедурна частина буде реалізована на сервері, а адміністратору не потрібно буде вивчати процедурну частину, щоб отримати данні з бази.

СУБД необхідно реалізувати за допомогою зв'язку клієнт-сервер, що забезпечить роботу всіх працівників НМВ з БД одночасно.

Для реалізації поставленої цілі найбільш доречним буде використання реляційної СУБД, за типом керованої бази даних СУБД.

Необхідно реалізувати всі основні функції СУБД розглянуті в розділі. Реалізації даних функцій дозволить створити гнучку систему управління.

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документу	Підпис	Дата			
Зав. кафедрою		Пурський О.І.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка	Сторінок
Керівник		Яловець А.Л.		20.11.2019		18	71
Гарант		Пурський О.І.		20.11.2019		Кафедра	
Розробив		Остапчук Б.В.		20.11.2019		комп'ютерних наук	
Перевірів		Яловець А.Л.		20.11.2019	Висновки до розділу 1	бм	

РОЗДІЛ 2. Аналітична підготовка до розробки інформаційної СУБД НМВ КНТЕУ.

2.1. Аналіз даних навчально-методичного відділу КНТЕУ

За основу взято «План навчального процесу на 1 курс, I семестр 2016 - 2017 н.р., 17 тижнів (теоретичне навчання 15 тижнів)».

Основні поля для складання плану навчального процесу:

1. Назва навчальної дисципліни (курсової роботи, проекту).
2. Загальний обсяг по дисципліні:
 - 2.1.1. Навчальні години.
 - 2.1.2. Кредити ЄКТС.
3. Обсяг на семестр по дисципліні:
 - 3.1.1. Навчальні години.
 - 3.1.2. Кредити ЄКТС.
4. Навчальні години:
 - 4.1.Лекції.
 - 4.2.Лабораторні (у.т.ч. модульний контроль).
 - 4.3.Практичні (семінарські) у.т.ч. модульний контроль.
 - 4.4.Самостійна робота.
5. Форма контролю.
6. Код кафедри.

Для зручності оперування даними, поле «Назва навчальної дисципліни», «Форма контролю» та «Код кафедри» повинно зберігатись і інших таблицях (див. рис. 2.1), СУБД автоматично буде показувати можливі варіанти підстановок в дане поле. Тобто, користувач СУБД повинен обрати з вже доданих варіантів.

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедрою		Пурський О.І.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка	Сторінок
Керівник		Яловець А.Л.		20.11.2019		19	71
Гарант		Пурський О.І.		20.11.2019		Кафедра комп'ютерних наук бм	
Розробив		Остапчук Б.В.		20.11.2019			
Перевірів		Яловець А.Л.		20.11.2019			



Рис 2.1. Таблиці БД для створення записів навчального плану.

Дані, які будуть відноситись більше ніж до 1 поля необхідно винести в окремі таблиці, для зручності та уникнення дублювання інформації. Таким чином, інформація буде зберігатись в окремих таблицях та зв'язуватись унікальними ідентифікаторами.

Для текстового формату необхідно назначити тип даних varchar, а для числового формату – integer.

На основі даних таблиці «Навчальні плани» можна створювати звіти, наприклад, звіт по коду кафедри, по формі контролю, навантаженню та ін.

Також, назви навчальних дисциплін будуть фігурувати в звітах по навчальним та робочим навчальним програмам (П/РП). Для П та РП буде створена одна таблиця, яка буде містити поле, що визначатиме тип запису, а саме П чи РП. І, обов'язково, необхідно зняти обмеження довжини varchar для поля з примітками, назначивши тип даних – text.

Основний тип даних НМВ КНТЕУ – це текст, отже, необхідно визначитись з кодуванням бази даних. Для української мови найбільш доречним є використання utf8_general_ci.

					КНТЕУ-122-2019	Аркуш
						20
Зм.	Аркуш	№ документу	Підпис	Дата		

2.2. На основі аналізу – побудова структури БД

Структура БД буде складатись з 12 таблиць, пов'язаних через унікальні ідентифікатори.

1. Для створення таблиці «Навчальні плани», відповідно до вимог, отримаємо наступний SQL запит:

```
DROP TABLE IF EXISTS `Навчальні плани`;  
CREATE TABLE `Навчальні плани` (  
  `id` INTEGER NOT NULL AUTO_INCREMENT,  
  `навчальна_дисципліна` INTEGER NULL DEFAULT NULL,  
  `Порядковий номер` INTEGER NULL DEFAULT NULL,  
  `Заг_обсяг_навчальні_години` INTEGER NULL DEFAULT NULL,  
  `Заг_обсяг_кредити` INTEGER NULL DEFAULT NULL,  
  `години_в_семестрі` INTEGER NULL DEFAULT NULL,  
  `кредити_в_семестрі` INTEGER NULL DEFAULT NULL,  
  `НВ_лекції` INTEGER NULL DEFAULT NULL,  
  `НВ_лабораторні` INTEGER NULL DEFAULT NULL,  
  `НВ_практичні` INTEGER NULL DEFAULT NULL,  
  `НВ_самостійна_робота` INTEGER NULL DEFAULT NULL,  
  `форма_контролю` INTEGER NULL DEFAULT NULL,  
  `код_кафедри` INTEGER NULL DEFAULT NULL,  
  PRIMARY KEY (`id`));
```

2. Створення таблиці навчальні дисципліни. Таблиця міститиме ідентифікатор та назву дисципліни. SQL запит для створення:

```
DROP TABLE IF EXISTS `Навчальні дисципліни`;  
CREATE TABLE `Навчальні дисципліни` (  
  `id` INTEGER NOT NULL AUTO_INCREMENT DEFAULT NULL,  
  `Назва` VARCHAR(655) NOT NULL,  
  PRIMARY KEY (`id`)  
  ) COMMENT 'Назва навчальної дисципліни (курсової роботи, проекту)';
```

					KHTEU-122-2019	Аркуш
						21
Зм.	Аркуш	№ документу	Підпис	Дата		

3. Створення таблиці «Форми контролю». Таблиця включиє в себе ідентифікатор та назву форми контролю.

```
DROP TABLE IF EXISTS `Форми контролю`;
```

```
CREATE TABLE `Форми контролю` (  
  `id` INTEGER NOT NULL AUTO_INCREMENT,  
  `Назва` VARCHAR(655) NULL DEFAULT NULL,  
  PRIMARY KEY (`id`));
```

4. Створення таблиці «Кафедри». Обов'язковими полями є код кафедри та її назва, ідентифікатор для зав'язків з іншими таблицями.

```
DROP TABLE IF EXISTS `Кафедри`;
```

```
CREATE TABLE `Кафедри` (  
  `id` INTEGER NOT NULL AUTO_INCREMENT,  
  `Назва_кафедри` VARCHAR(655) NULL DEFAULT NULL,  
  `Код_кафедри` INTEGER NULL DEFAULT NULL,  
  PRIMARY KEY (`id`, `Код_кафедри`));
```

5. Створення таблиці «Освітні рівні». Таблиця повинна містити назву та ідентифікатор.

```
DROP TABLE IF EXISTS `Освітні рівні`;
```

```
CREATE TABLE `Освітні рівні` (  
  `id` INTEGER NOT NULL AUTO_INCREMENT,  
  `Назва` VARCHAR(655) NULL DEFAULT NULL,  
  PRIMARY KEY (`id`));
```

6. Створення таблиці «Програми». До таблиці входять поля, що відображають назву програм, її автора, тип запису, освітній рівень та дату.

```
DROP TABLE IF EXISTS `Програми`;
```

```
CREATE TABLE `Програми` (  
  `id` INTEGER NOT NULL AUTO_INCREMENT DEFAULT NULL,  
  `Назва_програми` VARCHAR(655) NOT NULL DEFAULT 'NULL',
```

					КНТЕУ-122-2019	Аркуш
						22
Зм.	Аркуш	№ документу	Підпис	Дата		

```
`Автор_програми` INTEGER NULL DEFAULT NULL,  
`Тип_запису` INTEGER NULL DEFAULT NULL,  
`Освітній рівень` INTEGER NULL DEFAULT NULL,  
`Дата (вхід)` INTEGER NULL DEFAULT NULL,  
PRIMARY KEY (`id`));
```

7. Створення таблиці «Тип запису програм». Таблиця буде характеризувати програму, а саме її тип. Тому, таблиця включає в себе назву типу та його ідентифікатор.

```
DROP TABLE IF EXISTS `Тип_запису_програм`;  
CREATE TABLE `Тип_запису_програм` (  
`id` INTEGER NOT NULL AUTO_INCREMENT DEFAULT NULL,  
`Назва` INTEGER NULL DEFAULT NULL,  
PRIMARY KEY (`id`));
```

8. Створення таблиці «Автори програм». Таблиця створюється для зберігання інформації про авторів програм, включає в себе поле з ФІО автора, кафедру до якої він відноситься та ідентифікатор в системі.

```
DROP TABLE IF EXISTS `Автори_програм`;  
CREATE TABLE `Автори_програм` (  
`id` INTEGER NOT NULL AUTO_INCREMENT,  
`ФІО` VARCHAR(655) NULL DEFAULT NULL,  
`Кафедра` INTEGER NULL DEFAULT NULL,  
PRIMARY KEY (`id`));
```

9. Створення таблиці «Зауваження експертів». До таблиці входить id, id експерта, текстове зауваження.

```
DROP TABLE IF EXISTS `Зауваження_експертів`;  
CREATE TABLE `Зауваження_експертів` (  
`id` INTEGER NOT NULL AUTO_INCREMENT,  
`Експерт` INTEGER NULL,  
`Текст_зауваження` VARCHAR(655) NULL,
```

					КНТЕУ-122-2019	Аркуш
						23
Зм.	Аркуш	№ документу	Підпис	Дата		

PRIMARY KEY (`id`));

10. Створення таблиці «Зауваження НМВ». До таблиці входить id, id експерта (саме працівника НМВ), текстове зауваження.

DROP TABLE IF EXISTS `Зауваження НМВ`;

CREATE TABLE `Зауваження НМВ` (
`id` INTEGER NOT NULL AUTO_INCREMENT DEFAULT NULL,
`Зауваження` INTEGER NULL DEFAULT NULL,
`Експерт` INTEGER NULL DEFAULT NULL,
PRIMARY KEY (`id`));

11. Створення таблиці «Працівники НМВ». В таблиці будуть відображені працівники НМВ, їх ФІО, логін та пароль входу, а також тип доступу користувача.

DROP TABLE IF EXISTS `Працівники НМВ`;

CREATE TABLE `Працівники НМВ` (
`id` INTEGER NOT NULL AUTO_INCREMENT DEFAULT NULL,
`ФІО` VARCHAR(655) NULL DEFAULT NULL,
`ЛОГІН` VARCHAR(655) NULL DEFAULT NULL,
`ПАРОЛЬ` VARCHAR(655) NULL DEFAULT NULL,
`Тип\_доступу` `INTEGER NOT NULL,
PRIMARY KEY (`id`));

12. Створення таблиці «Типи доступу». Доступ для користувачів необхідно розбити на категорії «Адміністратор» та «Працівник», де група користувачів «Адміністратор» матиме повний доступ до функціоналу, а інша група матиме обмежений доступ.

DROP TABLE IF EXISTS `Типи доступу`;

CREATE TABLE `Типи доступу` (
`id` INTEGER NOT NULL AUTO_INCREMENT DEFAULT NULL,
`Доступи` INTEGER NULL DEFAULT NULL,
PRIMARY KEY (`id`));

					КНТЕУ-122-2019	Аркуш
						24
Зм.	Аркуш	№ документа	Підпис	Дата		

Отже, після виконання всіх запитів на сервері було отримано наступну структуру БД



Рис. 2.2. Структура БД НМВ КНТЕУ

2.3. Аналіз та вибір засобів реалізації поставленої мети

Першим і вагомим фактором є орієнтація програми на ВЕБ напрямок. Отже, необхідно обрати одну з серверних мов ВЕБ-програмування для досягнення мети. Звичайно для структури та стилізації буде використано HTML та CSS.

Необхідно проаналізувати дані статистики щодо серверних мов програмування та розглянути трьох лідерів. Найпопулярнішими мовами серверного програмування є (дані на 1 квітня 2018р.): PHP, ASP.NET, Java, static files, ColdFusion(рис. 2.3).

					КНТЕУ-122-2019	Аркуш
						25
Зм.	Аркуш	№ документа	Підпис	Дата		

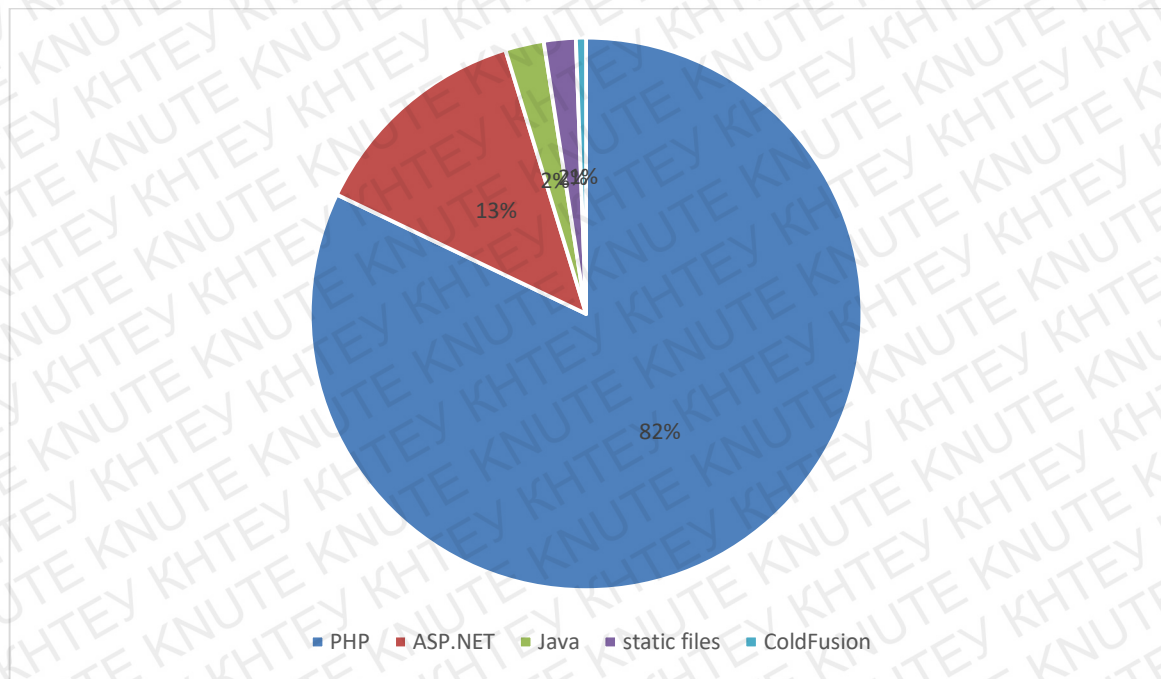


Рисунок 2.3. Найпопулярніші мови ВЕБ-програмування

Мова Java розроблена компанією Sun Microsystems, творцем якого був Джеймс Гослінг, і випущена в 1995 році в якості основних компонентів компанії Sun Microsystems - Java платформ. Мова має багато переваг та недоліків. Першою і вагомою перевагою є Сі-подібний синтаксис, який забезпечує низький поріг для вивчення мови. Знаючи хоча б одну мову з цього сімейства можна вивчити іншу набагато простіше. Java відноситься до мов загального призначення, на якій можна написати практично будь-яку програму. На сьогоднішній день на Java створюються програми будь-якого рівня складності: від програмування побутової техніки та мобільних пристроїв, до складних високо навантажених сервісів.

Всі комп'ютерні програми намагаються описати процеси нашого світу в командах процесора. Оскільки комп'ютер нічого не знає про об'єкти нашого світу, програмісту завжди доводилося «переводити» все на мову цифр і інструкцій. Так було до 70-х років, поки не з'явилася об'єктно орієнтована парадигма програмування. З її допомогою стало можливим програмування на рівні об'єктів з нашого світу, а не на рівні цифр зі світу комп'ютерного. Програми написані на об'єктно орієнтованих мовах виконувалися повільніше і це було основною

					КНТЕУ-122-2019	Аркуш
						26
Зм.	Аркуш	№ документа	Підпис	Дата		

причиною того, чому ця парадигма стала популярною лише в 90-х, коли сервери стали до цього готові. Java, як майже всі сучасні мови, це об'єктно-орієнтована мова, на якому досить просто моделювати процеси реального світу, і користуватися всіма перевагами об'єктно орієнтованого підходу.

Суворі типізація змінних створює перелік правил, порушувати які не можна, всі необхідні перетворення доводиться виконувати явно. Оскільки типи всіх змінних в Java визначаються статично на етапі компіляції, то є можливість перевірити чи правильно ці типи використані, але це питання вирішується після компіляції.

Переваги Java:

- універсальність;
- об'єктно-орієнтована;
- є вільним програмним забезпеченням;
- є велика кількість бібліотек і розширень мови.

Недоліки Java:

- поступається швидкості більшості мов серверного-програмування;
- суворі типізація;
- реалізовано з використанням інтерпретації Р-коду (байт-коду). Тобто програма спочатку транслюється в Р-код, а потім інтерпретується деякою програмою-інтерпретатором (віртуальна Java-машина, JVM).

ASP.NET - це частина технології .NET, використовувана для написання потужних клієнт-серверних інтернет додатків. Вона дозволяє створювати динамічні сторінки HTML. ASP.NET виникла в результаті об'єднання старішої технології ASP (активні серверні сторінки) і .NET Framework. Вона містить безліч готових елементів управління, використовуючи які можна швидко створювати інтерактивні web-сайти. У ASP.NET використовують компільовані мови. Під час компіляції перевіряється синтаксична коректність вихідного тексту. Компіляція відбувається на сервері в момент першого звернення користувача до сторінки. Якщо програміст змінив текст сторінки, програма перекомпілюється автоматично.

					КНТЕУ-122-2019	Аркуш
						27
Зм.	Аркуш	№ документа	Підпис	Дата		

При написанні коду можна використовувати набір компонентів, що поставляються з .NET.

ASP.NET використовує технологію доступу до даних ADO.NET, яка забезпечує єдиний інтерфейс для доступу до баз даних SQL Server і файлам XML. Крім того, посилена модель безпеки дозволяє забезпечувати захист клієнта і сервера від несанкціонованого доступу.

Переваги ASP.NET:

- наявність засобів візуального програмування;
- реалізація об'єктної моделі програми;
- підтримка технології Microsoft.NET вбудована в ОС Microsoft;
- вбудована підтримка технології Ajax - будь-який елемент форми може виконаний в «класичному» native варіанті або за допомогою Ajax.

Недоліки ASP.NET

- для роботи з ASP.NET потрібно ліцензійне програмне забезпечення;
- закладена шаблонізація;
- сувора типізація.

Отже, ASP.NET складна та багатофункціональна мова, яка потребує придбання ліцензії на її використання.

RHP (Hypertext PreProcessor, препроцесор гіпертексту) - мова програмування, що виконується на стороні веб-сервера, спроектований Расмусом Лердорфом (Rasmus Lerdorf) в якості інструменту створення динамічних та інтерактивних веб-сайтів.

Ця мова виявився досить гнучкою та потужною, тому набула великої популярності і використовується в проектах будь-якого масштабу: від простого блогу до найбільших веб-додатків в Інтернеті.

Переваги RHP:

- є вільним програмним забезпеченням;
- легкий в освоєнні на всіх етапах;
- підтримується великою спільнотою користувачів і розробників;

					КНТЕУ-122-2019	Аркуш
						28
Зм.	Аркуш	№ документа	Підпис	Дата		

- має розвинену підтримку баз даних;
- є велика кількість бібліотек і розширень мови;
- може використовуватися в ізольованому середовищі;
- є досить повною заміною середовища ASP (Active Server Pages) від Microsoft;
- може бути розгорнутий майже на будь-якому сервері;
- розроблено під велику кількість апаратних платформ і операційних систем.

Недоліки PHP:

- не підходить для створення десктопних додатків або системних компонентів;
- має слабкі засоби для роботи з винятками;
- глобальні параметри конфігурації впливають на базовий синтаксис мови, що ускладнює настройку сервера і розгортання додатків;
- об'єкти передаються за значенням, що бентежить багатьох програмістів, які звикли до передачі об'єктів по посиланню, як це робиться в більшості інших мов;

Отже, перевага на стороні PHP, яка є найпоширенішою, безкоштовною, універсальною для серверів та має розвинену підтримку баз даних.

Проекти, що використовують PHP: Yahoo, Facebook, Google, NASA, BlaBlaCar, W3C.

					КНТЕУ-122-2019	Аркуш
						29
Зм.	Аркуш	№ документа	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 2

Робота з базами даних потребує побудови чіткої архітектури та використання сучасних драйверів доступу. Правильно створена БД зменшує ризик втрати інформації. Забезпечення створення резервних копій знижує ризик безповоротної втрати даних.

Дані НМВ зберігаються в форматі .xls, що спрощує роботу з даними файлами. Відкривається можливість створення парсеру, який заповнить базу даних інформацією та в майбутньому зможе вносити необхідні зміни в БД.

На основі проведеного аналізу визначено кількість таблиць, їх поля, вміст, тип та саму структуру бази даних. На основі створених зв'язків через ідентифікатори виключається фактор дублювання інформації, що зменшить розмір БД. Наприклад, в таблицю зауважень не потрібно вносити ФІО та кафедру експерта, а лише його унікальний ідентифікатор. СУБД звернувшись до таблиці експертів зможе взяти інформацію та відобразити її користувачу.

На сьогоднішній день існує велика кількість мов серверного програмування. В результаті дослідження було проаналізовано трьох лідерів в даному напрямку та надано перевагу мові PHP, яка підтримується на більшості серверів та є найпоширенішою.

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедрою		Пурський О.І.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка	Сторінок
Керівник		Яловець А.Л.		20.11.2019		30	71
Гарант		Пурський О.І.		20.11.2019		Кафедра комп'ютерних наук бм	
Розробив		Остапчук Б.В.		20.11.2019			
Перевірив		Яловець А.Л.		20.11.2019			
					Висновки до розділу 2		

РОЗДІЛ 3. Розробка інформаційної системи управління БД НМВ КНТЕУ

3.1. Створення інтерфейсу СУБД

Створення інтерфейсу СУБД - це один з найважливіших етапів, тому необхідно виділити його основні принципи:

1. Інтерфейс повинен бути простий і зрозумілий, завдання повинні вирішуватися мінімальним числом дій, все повинно бути зрозуміло і очевидно.
2. У будь-якому сучасному інтерфейсі є багато елементів управління, доцільніше використовувати звичні елементи і візуальні образи.
3. Групування. Необхідно створити декілька логічних блоків (груп), так користувачеві легше орієнтуватися.
4. Розуміння краще запам'ятовування (нативно зрозумілий інтерфейс).
5. Всі важливі елементи інтерфейсу повинні бути на виду і відповідним чином виділені.
6. Повинно бути не більше 3-х дій для переходу з одного розділу в інший. Це ж правило застосовується до головної сторінки: будь-яка важлива інформація повинна бути доступна не більше ніж в 3 кліка.
7. У великих проектах часто зустрічається однорідний функціонал в різних частинах ІС (наприклад, шаблон редагування), він не повинен відрізнятися. Це ж стосується і стилю.

Отже, необхідно відштовхуватись від стандартної структури подібних систем – шапка, тіло, ліва колонка, права колонка, подвал. Шапка системи повинна містити логотип та назву університету, назву відділу та кнопку авторизації (або виходу, якщо користувач авторизований).

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедри		Пурський О.І.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка	Сторінок
Керівник		Яловець А.Л.		20.11.2019		31	71
Гарант		Пурський О.І.		20.11.2019		Кафедра комп'ютерних наук бм	
Розробив		Остапчук Б.В.		20.11.2019			
Перевірив		Яловець А.Л.		20.11.2019			

Панель керування необхідно розмістити в доступному місці, оскільки шапка сайту займатиме не велику область то можна розмістити інструменти в лівій колонці. А для зручності користування інструментами, їх буде згруповано в категорії (відповідно до принципу групування).

Для переходу на сторінку певного інструменту кількість кліків не повинна перевищувати трьох. Отже, перший клік користувача обирає групу інструменту, а другий клік – вибір самого інструменту.

Система матиме 4 групи, основну сторінку та інструмент прямого запиту SQL(у табл. 3.1).

Таблиця 3.1.

Групування інструментів

Порядковий номер	Група	Інструменти
1.	Основне	1. Програми. 2. Навчальні плани. 3. Кафедри. 4. Дисципліни.
2.	Зауваження	1. Зауваження НМВ КНТЕУ. 2. Зауваження експертів.
3.	Для налаштувань	1. Освітні рівні. 2. Форми контролю. 3. Типи програм.
4.	Користувачі	1. Автори програм. 2. Співробітники. 3. Групи працівників

Головна панель системи включатиме в себе кругову діаграму розміру таблиць відносно загального розміру БД. Та детальну інформацію по таблицям (див. рис. 3.1.).

					КНТЕУ-122-2019	Аркуш
						32
Зм.	Аркуш	№ документу	Підпис	Дата		

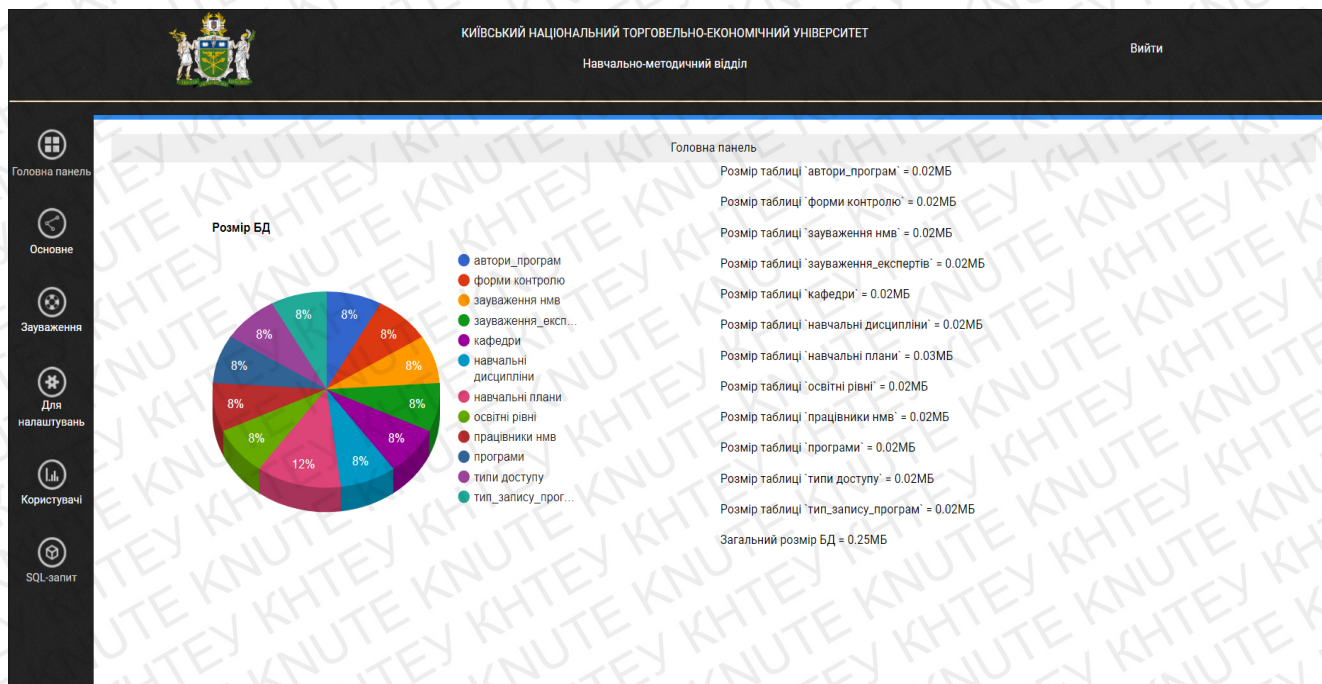


Рис. 3.1. Головна панель системи

Для розширення функціоналу створено пункт «SQL запит», сторінка якого містить поле для вводу тексту на мові SQL. Після запиту нище буде виведено результат у структурованому форматі масиву (див. рис. 3.2.).

КІЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
Навчально-методичний відділ

Головна панель

Основне

Зауваження

Для налаштувань

Користувачі

SQL-запит

SQL-запит

Введіть SQL запит

SELECT * FROM 'Програми'

Отправить

Результат Вашого запиту:

```

Array
(
    [0] => Array
        (
            [id] => 4
            [Назва_програм] => Програма для Комп. наук
            [Автор_програм] => 4
            [Тип_запису] => 3
            [Освітній_рівень] => 3
            [Дата_(v1d)] => 2019-11-18
            [зауваження_експертів] => 2
            [зауваження_нмв] => 2
        )
    [1] => Array
        (
            [id] => 5
            [Назва_програм] => test2
            [Автор_програм] => 3
            [Тип_запису] => 1
            [Освітній_рівень] => 2
            [Дата_(v1d)] => 2019-11-23
            [зауваження_експертів] => 2
            [зауваження_нмв] => 2
        )
)

```

					КНТЕУ-122-2019	Аркуш
						33
Зм.	Аркуш	№ документа	Підпис	Дата		

Рис. 3.2. Інструмент «SQL-запит»

Відповідно до останнього принципу побудови інтерфейсу системи необхідно розробити сторінку для одного інструменти з групи основне, яка буде шаблоном для решти інструментів.

Отже, сторінка «Навчальні плани» повинна включати в себе кнопки «Додати» та «Видалити», які виконуватимуть відповідний функціонал. Для зручності масового видалення буде використано чекбокси. Тобто, всі програми які будуть відмічені, після натискання на кнопку «Видалити» відповідно будуть видалені з системи.

Навчальні плани будуть побудовані за допомогою <table> (рис. 3.3.). Шапка таблиці включатиме наступні поля:

1. Чекбокс. При натисканні будуть відмічені всі навчальні плани в системі.
2. Порядковий номер.
3. Навчальна дисципліна.
4. Загальний обсяг – навчальні години.
5. Загальний обсяг – кредити.
6. Годин в семестрі.
7. Кредитів в семестрі.
8. Навчальні години – лекції.
9. Навчальні години – лабораторні (у.т.ч. модульний контроль).
10. Навчальні години – практичні (семінарські) у.т.ч. модульний контроль.
11. Навчальні години – самостійна робота.
12. Форма контролю.
13. Код кафедри.
14. Дія.

Остання колонка таблиці слугуватиме для розміщення кнопки для зміни програм – «Змінити».

					КНТЕУ-122-2019	Аркуш
						34
Зм.	Аркуш	№ документу	Підпис	Дата		

Видалити

№	Порядковий номер	Навчальна дисципліна	Загальний обсяг - навчальні години:	Загальний обсяг - кредити:	Годин в семестрі:	Кредити в семестрі:	Начальні години - лекції:	Начальні години - лабораторії (у т.ч. модульний контроль) :	Начальні години - практичні (семінарські) у т.ч. модульний контроль:	Начальні години - самостійна робота:	Форма контролю:	Код кафедри:	Дія
□	1	Податкове право	225	7.5	225	7.5	32	0	28	165	Е/п	14 : Економічної кібернетики	Змінити
□	2	Юридична компаративістика	135	4	135	4	30	0	14	91	Е/у	67 : Адміністративного, фінансового та інформаційного права	Змінити
□	3	Комерційне право	135	4	135	4	16	0	14	60	Е/у	68 : Загальноправових дисциплін	Змінити
□	4	Правове регулювання альтернативних методів вирішення юридичних спорів	135	4	135	4	16	0	14	105	3	66 : Міжнародного приватного, комерційного та цивільного права	Змінити
□	5	Тлумачення правових актів	90	3	90	3	16	0	14	60	3	68 : Загальноправових дисциплін	Змінити
□	6	Транснаціональне торгове право (Lex mercatoria)	90	3	90	3	16	0	14	60	3	66 : Міжнародного приватного, комерційного та цивільного права	Змінити
□	7	Методологія і організація наукових досліджень	90	3	90	3	16	0	14	60	3	68 : Загальноправових дисциплін	Змінити

При натисканні на кнопку «Додати» користувача буде направлено на сторінку додавання нового навчального плану. Сторінка повинна мати кнопки збереження та повернення на попередню сторінку, поля для вводу, які є необхідними для заповнення всіх полів, що стосуються навчального плану (рис. 3.4.). Поля «Дисципліна», «Форма контролю» та «Код кафедри» необхідно реалізувати за допомогою полю з вибором. Даними для вибору слугуватимуть відповідні дані, які будуть вибрані з БД відповідно до поля. Таким чином, для поля «Форма контролю» буде поле, в якому можна обрати один з варіантів «Е/У», «Е/З», «З».

					КНТЕУ-122-2019	Арқуш
						35
Зм.	Арқуш	№ документи	Підпис	Дата		



КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
 Навчально-методичний відділ

Головна панель
 Основне
 Зауваження
 Для налаштувань
 Користувачі
 SQL-запит

Зберегти
Повернутись

Навчальні плани

*Порядковий номер:

*Дисципліна:

Тлумачення правових актів

*Загальний обсяг - навчальні години:

*Загальний обсяг - кредити:

*Годин в семестрі:

*Кредити в семестрі:

*Начальні години - лекції:

*Начальні години - лабораторні (у.т.ч. модульний контроль) :

*Начальні години - практичні (семінарські) у.т.ч. модульний контроль:

*Начальні години - самостійна робота:

*Форма контролю:

E/т

*Код кафедри:

68 : Загальноправових дисциплін

Рис. 3.4. Додавання нового навчального плану.

Сторінка з додаванням та редагуванням навчального плану будуть однакові, лише з однією відмінністю. Якщо система отримає ідентифікатор запису, який потрібно редагувати, то поля введення будуть вже заповнені інформацією пов'язаною з даним ідентифікатором.

Відповідно до розробленого шаблону сторінки «Навчальні плани» та останнього принципу побудови інтерфейсу буде реалізовано решту інструменту системи.

Також, необхідно розробити шаблон для фільтрування даних, що й слугуватиме звітами. Базуючись на останньому принципі побудови інтерфейсу, розробимо фільтр для однієї сторінки, що слугуватиме шаблоном для побудови фільтрів для інших інструментів.

Отже, для інструменту «Програми», фільтр включатиме поля «Назва», «Автор», «Тип програми», «Освітній рівень» та кнопку «Фільтр». Останні три поля будуть реалізовані за допомогою випадаючого списку існуючих варіантів

					КНТЕУ-122-2019	Аркуш
						36
Зм.	Аркуш	№ документу	Підпис	Дата		

фільтрації. Наприклад, поле «Освітній рівень» матиме в собі 3 варіанти: «Всі», «Бакалавр», «Магістр» (див. рис. 3.4).

Програми

Додати Видалити

Фільтр "Програми":

Назва:

Автор:

Тип програми:

Освітній рівень:

Фільтрувати

	Назва	Автор програми	Тип програми	Освітній рівень	Зуваження НМВ	Зуваження експертів	Дата	Дія
☐	Програма для Комп. наук	Івановський Богдан Богданович	Збірник тестових завдань	Бакалавр	Автор зауваження: Іванов Петро Іванович Текст зауваження: Зауваження1	Автор зауваження: Валеріанов Валерій Валерійович Текст зауваження: Зауваження1 експерт	2019-11-18	Змінити

Рис. 3.4. Інструмент «Програми»

3.2. Програмування серверної частини

Першим кроком є створення підключення до Баз Даних. Для роботи з мовою SQL необхідно скачати та встановити офіційну бібліотеку та драйвера для зв'язку з БД. Редагувати офіційні файли не рекомендовано, тому необхідно створити файл конфігурації, який буде ініціалізувати зв'язок системи з Базою Даних та міститиме налаштування з'єднання (налаштування збережені в класі). Отже, файл конфігурації матиме наступний вигляд:

```
<?php class DBConfig
{
    protected $HOSTNAME = 'localhost'; // хост БД
    protected $username= 'root'; // користувач
    protected $password= ""; // пароль
    protected $database = 'nmv_knteu'; //Назва БД
    function getHOSTNAME()
    {
        return $this->HOSTNAME;
    }
    function getUserUserName()
```

					КНТЕУ-122-2019	Аркуш
						37
Зм.	Аркуш	№ документа	Підпис	Дата		

```

    { return $this->username; }
function getPassword()
    { return $this->password; }
function getDataBase()
    { return $this->database; }
};
include 'models/library/db.php'; // підключення бібліотеки SQL
$DBConfig = new DBConfig();
$db = new DB('mysql', $DBConfig->getHOSTNAME(), $DBConfig->
    >getUserName(), $DBConfig->getPassword(), $DBConfig->getDataBase());
include 'models/DB.php';
$DB = new work_with_DB($db); ?>

```

Останній рядок коду звертається до класу, конструктор якого приймає підключення БД, та буде використаний для роботи з БД. Клас має приватну змінну, в яку конструктор записує зв'язок з БД. Код конструктора:

```

private $db = null;
public function __construct($driver_db) {
    $this->db = $driver_db;
}

```

Необхідно виділити функції для інструментів системи.

1. Get{Назва інструменту}s() – функція, що повертає всі записи в таблиці інструменту.
2. Get{Назва інструменту}(\$id) – функція приймає ідентифікатор та обирає в БД запис відповідно до \$id.
3. Add{Назва інструменту}({параметри}) – функція додає новий запис в таблицю, обов'язковим є наявність переданих параметрів.
4. Edit{Назва інструменту}(\$id,{параметри}) – функція редагує існуючий запис з ідентифікатором, а саме – оновлює передані параметри.

					КНТЕУ-122-2019	Аркуш
						38
Зм.	Аркуш	№ документу	Підпис	Дата		

5. Remove{Назва інструменту}(\$id) – функція видаляє запис з таблиці у якого ідентифікатор дорівнює переданому параметру \$id.

Отже, маючи дванадцять інструментів необхідно написати 60 функцій обробки даних відповідно до інструменту. Додатково необхідні функції збору статистики БД та пряме виконання SQL запиту.

1. sendSQL(\$sql) – функція виконує запит, який було передано через змінну \$sql. Код функції:

```
public function sendSQL($sql){  
    $result = $this->db->query($sql);  
    return $result->rows;  
}
```

2. getSQL() – функція не приймає параметрів, а лише збирає статистику по таблицям БД. Код функції:

```
public function getSQL(){  
    $result = $this->db->query("SELECT  
    table_name,  
    round((((data_length + index_length) / 1024 / 1024), 2) `Size in MB`  
FROM information_schema.TABLES  
WHERE table_schema = \"nmv_knteu\"");  
    return $result->rows;  
}
```

Після побудови класу обробки БД (повний код класу див. дод. А) можна перейти до побудови файлової структури та контролерів.

Оскільки система модульна то це необхідно відобразити і в структурі файлової системи, яка включатиме в себе:

1. image – папка з зображеннями.
2. Scripts – папка зі скриптами js та jQuery.
3. Models – включає в себе бібліотеку SQL, драйвера роботи з БД та клас роботи з базою.

					КНТЕУ-122-2019	Аркуш
						39
Зм.	Аркуш	№ документу	Підпис	Дата		

4. Styles – стилі сайту.
5. Templates – шаблони для формування сторінок.
6. Контроллери з розширенням .php.

Першими шаблоном буде header.php в який необхідно перенести шаблон шапки сайту. На кожен сторінку системи буде підключено контролер header.php, який відповідно буде виводити шапку. Таким самим чином необхідно створити шаблон з лівою колонкою, яка відповідає за меню інструментів.

Наступним кроком є створення контролера головної сторінки. Першим пунктом є підключення раніше створеного файлу конфігурації, щоб сторінка мала зв'язок з БД. Наступним, і обов'язковим, пунктом є підключенням лівої колонки та шапки з папки шаблонів. Оскільки, було прийнято рішення виводити статистику по таблицям БД на головну то звернемося до раніше написаної функції getSQL(). Отримаємо статистику в вигляді масива, який необхідно вивести в вигляді розгорнутої історії до діаграми. Саму діаграму створено за допомогою сервісу Google – «jsapi». Необхідний масив даних для API сформуємо за допомогою php безпосередньо в скрипті js. Отже, отримаємо наступний код:

```
<script>
```

```
google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
    var data = google.visualization.arrayToDataTable([
        ['БД', 'розмір'],
        <?php
        foreach ($db_infos as $db_info){
            echo ['\'.'.$db_info['table_name'].'.\',\'.'.$db_info['Size in MB'].'.'],';
        }
        ?>
        ['0', 0]]);
    var options = {
```

					КНТЕУ-122-2019	Аркуш
						40
Зм.	Аркуш	№ документа	Підпис	Дата		

```

        title: 'Розмір БД',
        is3D: true,
        pieResidueSliceLabel: 'Решта таблиць'
    };

    var chart = new google.visualization.PieChart(document.getElementById('air'));
    chart.draw(data, options);
}
</script>

```

Переходимо до контролера інструменту, необхідно розробити основний модуль на основі якого будуть будуватися інші контролери системи. Обширним інструментом являється «Програми» та «Навчальні плани». Обриємо другий варіант та пишемо код який буду компонувати головну сторінку інструменту. Особливу увагу варто приділити полям, які зберігають лише ідентифікатор звязку елемента з існуючими записами.

Першим, обовязковим, пунктом є підключення файлу конфігурації, шапки та меню системи. Перед початком формування сторінки необхідно перевіряти вхідні дані, в випадку, якщо значення action (дія) існує та не пуста, необхідно виконати дію залежно від змінної \$action.

Контролер розуміє 3 дії: додати, редагувати, видалити.

Якщо обрана дія «add», то ініціалізується перевірка додаткових параметрів запиту методом POST. Перевіряється їх наявність та зміст (якщо змінна пуста – дію не виконувати), щоб уникнути помилок SQL. Якщо перевірка повернула успішний результат – звертаємось до класу «Робота з БД», викликаємо необхідну функцію запису та передаємо необхідні параметри. Після запису в БД перезавантажуємо сторіну, контролер сформує головну сторінку, яка все міститиме новий запис.

Якщо обрана дія «edit» - виконуються аналогічні дії, як в вище описаній дії «add» з однією відмінністю, а саме – до параметрів додається ідентифікатор запису.

Якщо обрана дія «remove» - перевіряється наявність відмічених записів, цикл проходить по них та відправляє запит для їх видалення, параметром являється унікальний ідентифікатор запису.

					КНТЕУ-122-2019	Аркуш
						41
Зм.	Аркуш	№ документу	Підпис	Дата		

Отримаємо наступний код обробки дій:

```
include_once('config.php');
if(!empty($_POST)){
    $action = $_GET['action'];
    switch ($action){
        case 'add':
            if(!empty($_POST['discipline'])    &&    !empty($_POST['number'])    &&
!empty($_POST['total_sem_time'])&& !empty($_POST['total_sem_credit']))){
                $DB-
>addEdPlan($_POST['discipline'],$_POST['number'],$_POST['total_sem_time'],$_POS
T['total_sem_credit'],$_POST['total_sem_time_semester'],$_POST['total_sem_credit_se
mester'],$_POST['ed_time_lection'],$_POST['ed_time_labs'],$_POST['ed_time_pract'],
$_POST['ed_time_self'],$_POST['program_type'],$_POST['ed_level']);
            }
            header('Location: /ed_plans.php');
            exit;
            break;
        case 'edit':
            if(!empty($_POST['plan_id'])&&    !empty($_POST['discipline'])    &&
!empty($_POST['number'])    &&    !empty($_POST['total_sem_time'])&&
!empty($_POST['total_sem_credit']))){
                $DB-
>editEdPlan($_POST['plan_id'],$_POST['discipline'],$_POST['number'],$_POST['total
_sem_time'],$_POST['total_sem_credit'],$_POST['total_sem_time_semester'],$_POST['
total_sem_credit_semester'],$_POST['ed_time_lection'],$_POST['ed_time_labs'],$_PO
ST['ed_time_pract'],$_POST['ed_time_self'],$_POST['program_type'],$_POST['ed_leve
l']);
            }
            header('Location: /ed_plans.php');
```

					KHTEV-122-2019	Аркуш
						42
Зм.	Аркуш	№ документу	Підпис	Дата		


```

        exit;
        break;
    case 'remove':
        if(isset($_POST['selected'])){
            foreach ($_POST['selected'] as $id){
                $DB->removeProgram($id);
            }
            header('Location: /ed_plans.php');
            exit;
        }
        break;
    default:
        break;
    }
}??>

```

Якщо контролер не знайшов параметра дії то відбувається формування головної сторінки інструменту. Контролер знаходить та підключає шаблон головної сторінки, який також необхідно запрограмувати.

Отже, для кожного контролера інструменту необхідно створити три шаблони. Перший шаблон відповідатиме за головну сторінку, другий – за сторінку додавання, а третій – за сторінку редагування. Другий та третій шаблон можна об'єднати в один і наповнювати даними, якщо передано ідентифікатор. Але для гнучкості і модульності буде створено саме три шаблони.

Переходимо до створення шаблону головної сторінки, що завершити її формування. В шаблон необхідно перенести раніше створену розмітку та не втратити прив'язки стилізації. Головна сторінка повинна включати в себе фільтр та всі записи, що наявні в БД відповідно до даного інструменту.

Спочатку виконаємо вивід всіх записів. Для цього необхідно звернутись до функції роботи з БД, а саме до функції GetEdPlans. Функція не приймає даних, а

					КНТЕУ-122-2019	Аркуш
						43
Зм.	Аркуш	№ документа	Підпис	Дата		

лише повертає їх. Якщо масив даних пустий то необхідно повідомити користувача що «Записів не знайдено». Якщо масив не пустий, то отримані дані необхідно записати в змінну, та пройтись по ній циклом, щоб структуровано вивести їх в таблицю. Нова ітерація циклу – це новий рядок в таблиці.

Для полів які мають лише посилання на запис іншого інструменту було обрано виведення полями з вибором. Для цього необхідно звернутись до функції іншого інструменту, яка міститься в класі роботи з БД, та запросити запис по ідентифікатору запису ітерації циклу.

Отримаємо наступну розмітку з кодом:

```
<?php $sed_plans = $DB->getEdPlans();
if(isset($sed_plans) && !empty($sed_plans)){
    foreach ($sed_plans as $sed_plan){ ?>
        <tr>
            <td style="text-align: center;"> <input type="checkbox"
name="selected[]" value="<?php echo $sed_plan['id']; ?>">
            </td>
            <td class="left">
                <?php echo $sed_plan['Порядковий номер']; ?>
            </td>
            <td class="left">
                <?php $discipline = $DB->getDiscipline($sed_plan['навчальна_дисципліна']);
                echo $discipline['Назва'];
                ?>
            </td>
            <td class="left">
                <?php echo $sed_plan['Заг_обсяг_навчальні_години']; ?>
            </td>
            <td class="left">
                <?php echo $sed_plan['Заг_обсяг_кредити']; ?>
```

					КНТЕУ-122-2019	Аркуш
						44
Зм.	Аркуш	№ документу	Підпис	Дата		

```

</td>
<td class="left">
    <?php echo $ed_plan['години_в_семестрі']; ?>
</td>
<td class="left">
    <?php echo $ed_plan['кредити_в_семестрі']; ?>
</td>
<td class="left">
    <?php echo $ed_plan['НВ_лекції']; ?>
</td>
<td class="left">
    <?php echo $ed_plan['НВ_лабораторні']; ?>
</td>
<td class="left">
    <?php echo $ed_plan['НВ_практичні']; ?>
</td>
<td class="left">
    <?php echo $ed_plan['НВ_самостійна_робота']; ?>
</td>
<td class="left">
    <?php $form_type = $DB->getFotmType($ed_plan['форма_контролю']);
    if(isset($form_type)){
        echo $form_type['Назва'];
    }
    ?>
</td>
<td class="left">
    <?php $chair = $DB->getChair($ed_plan['код_кафедри']);
    if(isset($chair)){

```

					КНТЕУ-122-2019	Аркуш
						45
Зм.	Аркуш	№ документа	Підпис	Дата		


```

        echo $chair['Код_кафедри'] . ' : ' . ' ';
        echo $chair['Назва_кафедри'];}

    ?>
</td>

<td class="right"> <a href="/ed_plans.php?action=edit&plan_id=<?php
echo $ed_plan['id']; ?>">Змінити</a>

</td>

</tr>

<?php }

}else{echo "Записів не знайдено.";}?>

```

Після створення головної сторінки, та її побудови отримаємо повідомлення, що «Записів не знайдено». Необхідно створити сторінку, яка зможе додавати записи в БД та викликатиметься раніше описаними діями(\$action). Для побудови такого шаблону візьмемо вже готову шапку таблиці головної сторінки, структуруємо відповідно до раніше створеної розмітки та зв'язуємо з базою даних.

Відповідно сторінку редагування можна створити скопіювавши сторінку додавання записів та внести зміни в значення полів. У сторінки додавання всі поля мають значення value="". В лапки необхідно внести наступний код, відповідно до поля, <?php echo \${значення_поля}; ?>.

Наприклад, поле з порядковим номером:

```

<tr>

<td><span class="required">*</span> Порядковий номер:</td>

<td><input type="text" name="number" value="<?php echo
$plan['Порядковий номер']; ?>" required>

</td>

</tr>

```

Для поля з вибором використовується інша конструкція, наприклад, поле «Дисципліна»:

					КНТЕУ-122-2019	Аркуш
						46
Зм.	Аркуш	№ документа	Підпис	Дата		

```

<tr>
<td><span class="required">*</span>Дисципліна:</td>
<td>
<select name="discipline">
<?php $disciplines = $DB->getDisciplines(); ?>
<?php foreach ($disciplines as $discipline){
    ?>
    <option value="<?php echo $discipline['id']; ?>" <?php
if($discipline['id']==$plan['навчальна_дисципліна']) { ?> selected <?php } ?><?php
echo $discipline['Назва']; ?></option>
    <?php } ?>
</select>
</td>
</tr>

```

Отриманий код форми редагування можна переглянути в додатку Б.

Заповнивши декілька записів, отримаємо наступний вигляд головної сторінки інструменту:

 КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ Навчально-методичний відділ													
Навчальні плани													
Головна панель Освоєне Зуваження Для налаштувань Користувачі SQL-запит													
Додати Видалити													
Порядковий номер	Навчальна дисципліна	Загальний обсяг - навчальні години:	Загальний обсяг - кредити:	Годин в семестрі:	Кредити в семестрі:	Начальні години - лекції:	Начальні години - лабораторії (у т.ч. модульний контроль):	Начальні години - практичні (семінарські) у т.ч. модульний контроль:	Начальні години - самостійна робота:	Форма контролю:	Код кафедри:	Дія	
☐ 1	Податкове право	225	7.5	225	7.5	32	0	28	165	Е/п	14: Економічної кібернетики	Змінити	
☐ 2	Юридична компаративістика	135	4	135	4	30	0	14	91	Е/у	67: Адміністративного, фінансового та інформаційного права	Змінити	
☐ 3	Комерційне право	135	4	135	4	16	0	14	60	Е/у	68: Загальноправових дисциплін	Змінити	
☐ 4	Правове регулювання альтернативних методів вирішення юридичних спорів	135	4	135	4	16	0	14	105	3	66: Міжнародного приватного, комерційного та цивільного права	Змінити	

Рис. 3.2. Головна сторінка інструменту «Навчальні плани»

					КНТЕУ-122-2019	Аркуш
						47
Зм.	Аркуш	№ документа	Підпис	Дата		

Залишилось інтегрувати фільтр для необхідних інструментів. Оскільки, «Навчальні плани» вже розглянуто, то фільтр буде розроблено для «Програми» та інтегровано в інструмент «Навчальні плани» з відповідними змінами.

Виконавши дії побудови шаблонів редагування, додавання та головної сторінки для інструменту «Програми» отримаємо наступний вид головної сторінки, на яку потрібно інтегрувати фільтр:

 КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ Навчально-методичний відділ									
Головна панель Оснoвнe Зaуваження Длa налаштувань Користувачі SQL-запит	Програми								
	Додати Видалити								
	☐	Назва	Автор програми	Тип програми	Освітній рівень	Зауваження НМВ	Зауваження експертів	Дата	Дія
	☐	Програма для Комп. наук	Івановський Богдан Богданович	Збірник тестових завдань	Бакалавр	Автор зауваження: Іванов Петро Іванович Текст зауваження: Зауваження1	Автор зауваження: Валеріанов Валерій Валерійович Текст зауваження: Зауваження1 експерт	2019-11-18	[Змінити]
	☐	Програма тестова	Валеріанов Валерій Валерійович	Програма	Магістр	Автор зауваження: Іванов Петро Іванович Текст зауваження: Зауваження1	Автор зауваження: Валеріанов Валерій Валерійович Текст зауваження: Зауваження1 експерт	2019-11-23	[Змінити]

Рис. 3.3. Головна сторінка інструменту «Програми»

Перед таблицею даних інтегруємо розмітку фільтра та наповнимо його даними для фільтрації по інструменту.

Поле з назвою буде фільтрувати по співпадінню, від користувача не вимагається введення точної назви. Наприклад, ввівши «Ком», частину назви «Комп. наук» система знайде та відобразить запит з такою назвою. Для полів з випадаючими списками першим пунктом повинна бути опція «Всі записи», щоб фільтр не впливав на перше відпрацювання контролера інструменту. Код поля з вибором буде наступним:

```
<select name="author_filter">
```

```
<?php $authors = $DB->getAuthors(); ?>
```

					КНТЕУ-122-2019	Аркуш
						48
Зм.	Аркуш	№ документу	Підпис	Дата		


```

<option value="" <?php if($data_author_filter == ""){ echo "selected";
}>>Bci</option>

<?php foreach ($authors as $author){
    ?>

    <option value="<?php echo $author['id']; ?>" <?php if($author['id']==
$data_author_filter){ echo "selected"; } ?>><?php echo $author['FIO']; ?></option>

    <?php } ?>
</select>

```

Після такої інтеграції отримаємо наступний вигляд сторінки:

Рис. 3.4. Вигляд інструменту «Програми» з інтегрованим фільтром

Решту інструменту розроблено відповідно до вище описаних особливостей.

3.3. Тестування СУБД

Першим етапом тестування буде швидкість побудови сторінки складних інструментів. Відповідно, під час отримання запиту від користувача – фіксуємо час (з точністю до мілісекунд). Після завершення формування сторінки – надсилаємо відповідь про швидкість.

Код при отриманні запиту:

```
$start_time = microtime();
```

					КНТЕУ-122-2019	Аркуш
						49
Зм.	Аркуш	№ документа	Підпис	Дата		

```
$start_array = explode(" ", $start_time); // обробка секунд та мілісекунд
$start_time = $start_array[1] + $start_array[0];
```

Під час відповіді користувачеві виконати наступний код:

```
$end_time = microtime();
$end_array = explode(" ", $end_time);
$end_time = $end_array[1] + $end_array[0];
$time = $end_time - $start_time;
Printf("Сторінка сформована за %f секунд", $time);
```

Аналіз швидкодії фіксуємо в таблиці для зручності порівняння даних.

Таблиця 3.2.

Швидкодія сторінок СУБД

№	Назва сторінки/інструменту	Швидкість (с.)
1.	Головна панель	0,000907 с.
2.	Програми	0,002678 с.
3.	Навчальні плани	0,002511 с.
4.	Кафедри	0,000775 с.
5.	Дисципліни	0,000876 с.
6.	Зауваження НМП КНТЕУ	0,000743 с.
7.	Зауваження Експертів	0,000743 с.
8.	Освітні рівні	0,000826 с.
9.	Форми контролю	0,000425 с.
10.	Типи програм	0,000555 с.
11.	Автори програм	0,000729 с.
12.	Співробітники	0,000623 с.
13.	SQL запит	0,000228 с.

Проведені тести відбувались на сервері з процесором Intel i7-8750H, 16ГБ ОЗУ та на швидкісному SSD M.2 PCIe 3.0 x4 V-NAND MLC.

Очевидно, що найшвидшою сторінкою є «SQL-запит», оскільки містить найменше компонентів, а найповільнішими є два інструменти під номером 2 та 3

					КНТЕУ-122-2019	Аркуш
						50
Зм.	Аркуш	№ документу	Підпис	Дата		

та головна сторінка з діаграмою. Отже, середня швидкість системи складає 0,000970 с.

Наступним етапом буде тестування найповільнішої сторінки системи «Програми», а саме – запис, зміна, видалення запиту та фільтрація (див. таб. 3.3.).

Таблиця 3.3

Тестування інструменту «Програми»

№	Вид тесту	Швидкість (с.)
1.	Фільтрування по назві	0,001930 с.
2.	Фільтрування по всім параметрам	0,002046 с.
3.	Додавання запису	0,002925 с.
4.	Редагування запису	0,003325 с.
5.	Видалення запису	0,002599 с.

Найшвидша дія це фільтрування по назві, навіть швидше ніж звичайне формування сторінки. Це обумовлене тим, що відібраних записів менше, отже, системі потрібно менше ітерацій циклу. Найповільніша – редагування запису, оскільки ця дія передає додатково ідентифікатор запису.

Наступний етап тестування – це тестування самого класу звязків з БД, пряме виконання SQL-запиту та виведення структурованого масиву даних(див. таб. 3.4).

Таблиця 3.4

Тестування прямими SQL-запитами

№	SQL-запит	Швидкість (с.)
1.	SELECT * FROM `Програми`	0,000662 с.
2.	SELECT * FROM `Програми` WHERE `Дата (вхід)` = '2019-08-18'	0,000697 с.
3.	SELECT * FROM `Програми` WHERE `Дата (вхід)` > '2019-07-18'	0.000802 с.
4.	SELECT p.`id`,p.`Назва_програми`,ар.`ФІО` FROM `Програми` p	0,000877 с.

					КНТЕУ-122-2019	Аркуш
						51
Зм.	Аркуш	№ документу	Підпис	Дата		

	LEFT JOIN `автори_програм` ar ON (p.`Автор_програми` = ar.`id`) WHERE `Дата (вхід)` > '2019-07-18'	
5.	SELECT p.`id`,p.`Назва_програми`,ar.`ФІО` as `ФІО автора`,tp.`Назва` as `Назва типу запису` FROM `Програми` p LEFT JOIN `автори_програм` ar ON (p.`Автор_програми` = ar.`id`) LEFT JOIN `тип_запису_програм` tp ON (p.`Тип_запису` = tp.`id`) WHERE `Дата (вхід)` > '2019-07-18'	0,000898 с.

Отже, виконання запитів не перевищило 0,001 с. виконання, що свідчить про її швидкість та стійкість.

					КНТЕУ-122-2019	Аркуш
						52
Зм.	Аркуш	№ документа	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 3

На основі двох попередніх розділів було побудовано інформаційну систему управління базою даних навчально-методичного відділу Київського національного торговельно-економічного університету. Створено необхідну кількість інструментів для додавання, видалення, редагування записів бази даних

Було розроблено та реалізовано простий, модульний та гнучкий інтерфейс системи. Інструменти згруповані по їх призначенню. Система має фільтри, що слугують звітами по записам. Також система має функцію прямої взаємодії з SQL завдяки інструменту «SQL-запит».

Мова програмування PHP дала змогу повноцінно виконати створення класу для забезпечення зв'язку з БД. Оскільки система модульна, це забезпечує легке інтегрування додаткових модулів з функціоналом, гнучкість в розвиненні системи.

Система успішно пройшла тестування, показала швидке виконання запитів, стабільність роботи. Всі запити відпрацьовують без затримок, в швидкість було включено, як виконання запиту так і побудова сторінки. Тобто, швидкість було проаналізовано від запиту клієнта до надсилання відповіді. А швидкість виконання прямих запитів не перевищувала 0,001с.. Складний груповий запит, який звертався до трьох таблиць, збирав данні та записував їх під відповідним ключем, був виконаний лише за 0,000898 секунди.

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документу	Підпис	Дата			
Зав. кафедрую		Пурський О.І.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка	Сторінок
Керівник		Яловець А.Л.		20.11.2019		53	71
Гарант		Пурський О.І.		20.11.2019		Кафедра	
Розробив		Остапчук Б.В.		20.11.2019	Висновки до розділу 3	комп'ютерних наук	
Перевірів		Яловець А.Л.		20.11.2019		бм	

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В епоху діджиталізації постає питання переведення інформації в цифрову форму, тому виникла необхідність в розробці та створенні інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ.

В першому розділі було зібрано інформацію про досліджуваний об'єкт, розглянуто мову маніпуляції даними та різні види класифікації СУБД. Виходячи із зібраної інформації було надано перевагу передовим та ефективним методам розробки СУБД.

На основі розглянутих функцій СУБД були визначені та реалізовані інструменти для роботи. Функції реалізовано підповідно до останніх вимох діджиталізації. Система будувалась на принципах простоти, ефективності, модульності та гнучкості.

В другому розділі було проаналізовано данні НМВ КНТЕУ відповідно до чого побудована структура самої бази даних. Також, на основі характеристик та статистики мов серверного програмування було обрано РНР, як засіб реалізації поставленої мети. Дана мова забезпечила ефективність та створення саме модульної системи.

В третьому розділі було розроблено інформаційну систему управління базою даних навчально-методичного відділу Київського національного торговельно-економічного університету, відповідно до вимог поставлених в попередніх пунктах.

В епоху діджиталізації, швидких змін та зростання темпу розвитку інформаційних технологій, необхідно використовувати переваги цифрової інформації, що дає змогу йти в ногу з часом та уникнути найпоширеніших пробле

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедраю		Пурський О.І.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка	Сторінок
Керівник		Яловець А.Л.		20.11.2019		54	71
Гарант		Пурський О.І.		20.11.2019		Кафедра	
Розробив		Остапчук Б.В.		20.11.2019	Висновки та пропозиції	комп'ютерних наук	
Перевірів		Яловець А.Л.		20.11.2019		бм	

устарілих технологій. Рано чи пізно держава вводить закони та оцифровує інформацію, вимагаючи цього від приватних та державних підприємств.

Після завершення мінімально необхідної діджиталізації з'являються очевидні переваги, наприклад, економія часу. Використавши фільтр даних можна за доли секунди отримати необхідну інформацію.

Обрана архітектура дозволяє використовувати СУБД одночасно великою кількістю користувачів, виключає протиріччя записів, підтримує цілісність даних, інформація стає доступнішою.

Але система вимагає від адміністратора безпосереднього контролю, створення резервних копій, контроль доступу користувачів, створення груп доступу та інше. Це є основний недолік, який перекидає значущу кількість переваг.

Зі стрімким зростанням обчислювальної потужності - зростає і швидкість роботи СУБД, основні виробники-гіганти пропонують широкий спектр процесорів для різних потреб. Їх конкуренція є важелем для цін, що робить процесори доступнішими.

Отже, переваги діджиталізації, використання баз даних та систем їх управління є очевидними. Питання створення подібних систем постає все гостріше з кожним кварталом року, але до розробки СУБД необхідно підходити обережно та з детальним аналізом даних, поставлених цілей та методів реалізації.

					КНТЕУ-122-2019	Аркуш
						55
Зм.	Аркуш	№ документа	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нормативний документ технічного захисту інформації 1.1-002-99 Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу.
2. Кудінов М.В. Аналіз методик навчання дисципліни "Основи автоматизованого проектування складних об'єктів і систем (CASE)" – Бердянськ, 2009 р. — 284 с.
3. Курнієнко М.М., Іванова І.Д. Бази даних. Системи управління базами даних – Ранок, 2009 – 52 с.
4. Тарасов О. В. Лосєв М. Ю., Федько. В. В. Використання мови SQL для роботи з сучасними системами керування базами даних — Харків : ХНЕУ, 2013. — 347 с.
5. Анісімов А.В., Кулябко П.П. Інформаційні системи та бази даних – ВHV, Київ. – 2017. – 110 с.
6. Жалдак М.І., Радкевич Ю.С., Рамський В.О., Новітні комп'ютерні технології – Кривий Ріг, 2017 – 280 с.
7. Зав'ялець Ю.А. Web-технології та web-дизайн. Конспект лекцій /Укл.: – Чернівці, 2014. – 90 с.
8. Програмна інженерія : підруч. для студ. вищ. навч. закл. / К.М. Лавріщева. – К. : Київ. УДК, 2008. – 266 с.
9. Інтернет-портали: збірник наукових статей, збірник 2 / А. Н. Тихонов – К. : Київ., 2008. – 499 с.

					КНТЕУ-122-2019		
Зм.	Аркуш	№ документа	Підпис	Дата			
Зав. кафедри		Пурський О.І.		20.11.2019			
Керівник		Яловець А.Л.		20.11.2019	Розробка інформаційної системи управління базою даних навчально-методичного відділу КНТЕУ	Сторінка 56	Сторінок 71
Гарант		Пурський О.І.		20.11.2019		Кафедра комп'ютерних наук бм	
Розробив		Остапчук Б.В.		20.11.2019			
Перевірів		Яловець А.Л.		20.11.2019			

- 10.Бази даних в Інтернеті: практичний посібник зі створення Web-додатків з базами даних / А.В. Фролов, Г.В. Фролов, – К. : Київ., 2009. – 89 с.
- 11.Web-дизайн : навч. посіб., збірник 3 / Т. Пауелл – К. : Вінниця., 2008. - 1084 с.
- 12.Web-програмування та web-дизайн. Технологія XML : навч. посіб. / О.Б. Проценко. – К. : Суми. СумДУ, 2009. – 127 с.
- 13.MySQL: лабораторний практикум : нав. книга / Н.Р.Балик, В.І. Мандзюк. - К. : Тернопіль. 2008. - 7 с.
- 14.Економіко-математичне моделювання соціально-економічних систем: збірник наукових праць МННЦ ITiC, збірник 14 / Т. П. Подчасова – К. : Київ. УДК, 2009. – 24 с.
- 15.Системи управління базами даних та знань: книга 2 / 8. А. Ю. Берко, О. М. Верес, В. В. Пасічник. – К. : Львів. «Магнолія-2006», 2012. – 584 с.
- 16.Coar K., Bowen R. Apache Cookbook – O'Reilly, 2007 – 310 p.
- 17.Martin Kleppmann. Designing Data-Intensive Applications – O'Reilly, 2017 – 559 p.
- 18.International research and practice journal - Software products and systems, №3 2016y.
- 19.Matt Zandstra. PHP Objects, Patterns, and Practice 5th ed. – Liverpool, 2016 – 565 p.
- 20.MacIntyre P., Danchilla B., Gogala M. Pro PHP Programming – Apress, 2011 – 440 p.
- 21.Економіко-математичне моделювання соціально-економічних систем [Електронний ресурс]. – Режим доступу: http://ua.kursoviks.com.ua/metodychni_vkazivky/article_post/274-v-rtualn-p-dpri-mstva-yak-suchasna-forma-organ-zac-virobnictva
- 22.Поняття і функції інтернет-магазину. [Електронний ресурс]. – Режим доступу: <http://ukrbukva.net/page,6,78319-Saiyt-sportivnogo-kluba-bodibilinga-s-internet-magazinom.html>

					KHTEU-122-2019	Аркуш
						57
Зм.	Аркуш	№ документа	Підпис	Дата		

Додатки

Додаток А

```
<?php
class work_with_DB
{
    private $db = null;
    public function __construct($driver_db) {
        $this->db = $driver_db;
    }
    /* personal block */
    public function getPersonals(){
        $result = $this->db->query("SELECT * FROM `працівники нмв` ORDER BY
`працівники нмв`.`ФІО` ASC")->rows;
        return $result;
    }
    public function getPersonal($id){
        $result = $this->db->query("SELECT * FROM `працівники нмв` WHERE
`працівники нмв`.`id` = ".$id."")->row;
        return $result;
    }
    public function addPersonal($FIO,$login,$password, $role = null)
    {
        $result = $this->db->query("INSERT INTO `працівники нмв` SET `ФІО` =
'".$FIO."', `ЛОГІН` = '".$login."', `ПАРОЛЬ` = '".$password."', `Тип доступу` =
'".$role."");
    }
    public function editPersonal($id, $FIO,$login,$password, $role = null)
    {
        $result = $this->db->query("UPDATE `працівники нмв` SET `ФІО` =
'".$FIO."', `ЛОГІН` = '".$login."', `ПАРОЛЬ` = '".$password."', `Тип доступу` =
'".$role."' WHERE `працівники нмв`.`id` = ".$id."");
    }
}
```

```

}

public function removePerson($id){

    $this->db->query("DELETE FROM `працівники нмв` WHERE `працівники нмв`.`id` = '". $id.'"'");

}

/* !personal block */

/* role block */

public function getRoles(){

    $result = $this->db->query("SELECT * FROM `типи доступу`")->rows;

    return $result;

}

public function getRole($id){

    $result = $this->db->query("SELECT * FROM `типи доступу` WHERE `типи доступу`.`id` = '". $id.'"'")->row;

    return $result;

}

public function addRole($user_group, $access)

{

    $result = $this->db->query("INSERT INTO `типи доступу` SET `Назва` = '". $user_group.'', `Доступи` = '". $access.'"'");

}

public function editRole($id,$user_group,$access)

{

    $result = $this->db->query("UPDATE `типи доступу` SET `Назва` = '". $user_group.'', `Доступи` = '". $access.'" WHERE `типи доступу`.`id` = '". $id.'"'");

}

public function removeRole($id){

    $this->db->query("DELETE FROM `типи доступу` WHERE `типи доступу`.`id` = '". $id.'"'");

}

```

```

}

/* !role block */

/* ed_level block */

public function getEdLevels(){

    $result = $this->db->query("SELECT * FROM `освітні рівні` ORDER BY
`освітні рівні`.`Назва` ASC")->rows;

    return $result;

}

public function getEdLevel($id){

    $result = $this->db->query("SELECT * FROM `освітні рівні` WHERE `освітні
рівні`.`id` = ".$id."")->row;

    return $result;

}

public function addEdLevel($name)

{

    $result = $this->db->query("INSERT INTO `освітні рівні` SET `Назва` =
".$name."");

}

public function editEdLevel($id,$name)

{

    $result = $this->db->query("UPDATE `освітні рівні` SET `Назва` = ".$name."
WHERE `освітні рівні`.`id` = ".$id."");

}

public function removeEdLevel($id){

    $this->db->query("DELETE FROM `освітні рівні` WHERE `освітні рівні`.`id` =
".$id."");

}

/* !ed_level block */

```



```

/* type_program block */

public function getProgramTypes(){
    $result = $this->db->query("SELECT * FROM `тип_запису_програм` ORDER
BY `тип_запису_програм`.`Назва` ASC")->rows;
    return $result;
}

public function getProgramType($id){
    $result = $this->db->query("SELECT * FROM `тип_запису_програм` WHERE
`тип_запису_програм`.`id` = '".$id."'")->row;
    return $result;
}

public function addProgramType($name)
{
    $result = $this->db->query("INSERT INTO `тип_запису_програм` SET `Назва`
= '".$name."'");
}

public function editProgramType($id,$name)
{
    $result = $this->db->query("UPDATE `тип_запису_програм` SET `Назва` =
'".$name.'" WHERE `тип_запису_програм`.`id` = '".$id."'");
}

public function removeProgramType($id){
    $this->db->query("DELETE FROM `тип_запису_програм` WHERE
`тип_запису_програм`.`id` = '".$id."'");
}

/* !type_program block */

/* chair block */

public function getChairs(){

```

```

$result = $this->db->query("SELECT * FROM `кафедри` ORDER BY
`кафедри`.`Назва_кафедри` ASC")->rows;

    return $result;
}

public function getChair($id){
    $result = $this->db->query("SELECT * FROM `кафедри` WHERE
`кафедри`.`id` = '". $id. "'")->row;

    return $result;
}

public function addChair($name, $chair_code)
{
    $result = $this->db->query("INSERT INTO `кафедри` SET `Назва_кафедри` =
'".$name."', `Код_кафедри` = '".$chair_code."'");
}

public function editChair($id,$name,$chair_code)
{
    $result = $this->db->query("UPDATE `кафедри` SET `Назва_кафедри` =
'".$name."', `Код_кафедри` = '".$chair_code."' WHERE `кафедри`.`id` = '". $id. "'");
}

public function removeChair($id){
    $this->db->query("DELETE FROM `кафедри` WHERE `кафедри`.`id` =
'".$id. "'");
}

/* !chair block */
/* form_types block */

public function getFotmTypes(){
    $result = $this->db->quer
public function getFotmTypes(){
    $result = $this->db->query("SELECT * FROM `форми контролю`")->rows;

```

```

return $result;
}

public function getFotmType($id){
    $result = $this->db->query("SELECT * FROM `форми контролю` WHERE
`форми контролю`.`id` = '".$id."'")->row;
    return $result;
}

public function addFormType($name)
{
    $result = $this->db->query("INSERT INTO `форми контролю` SET `Назва` =
'".$name."'");
}

public function editFormType($id,$name)
{
    $result = $this->db->query("UPDATE `форми контролю` SET `Назва` =
'".$name.'" WHERE `форми контролю`.`id` = '".$id."'");
}

public function removeFormType($id){
    $this->db->query("DELETE FROM `форми контролю` WHERE `форми
контролю`.`id` = '".$id."'");
}

/* !form_types block */
/* disciplines block */

public function getDisciplines(){
    $result = $this->db->query("SELECT * FROM `навчальні дисципліни` ORDER
BY `навчальні дисципліни`.`Назва` ASC")->rows;
    return $result;
}

public function getDiscipline($id){

```



```

$result = $this->db->query("SELECT * FROM `навчальні дисципліни` WHERE
`навчальні дисципліни`.`id` = '". $id.'")->row;

return $result;

}

public function addDiscipline($name)
{
    $result = $this->db->query("INSERT INTO `навчальні дисципліни` SET `Назва`
= '". $name.'")");
}

public function editDiscipline($id,$name)
{
    $result = $this->db->query("UPDATE `навчальні дисципліни` SET `Назва` =
'". $name.'" WHERE `навчальні дисципліни`.`id` = '". $id.'")");
}

public function removeDiscipline($id){
    $this->db->query("DELETE FROM `навчальні дисципліни` WHERE
`навчальні дисципліни`.`id` = '". $id.'")");
}

/* !disciplines block */
/* author block */

public function getAuthors(){
    $result = $this->db->query("SELECT * FROM `автори_програм` ORDER BY
`автори_програм`.`ФІО` ASC")->rows;

    return $result;
}

public function getAuthor($id){
    $result = $this->db->query("SELECT * FROM `автори_програм` WHERE
`автори_програм`.`id` = '". $id.'")->row;

    return $result;
}

```

```

    }

    public function addAuthor($FIO, $chair = null)
    {
        $result = $this->db->query("INSERT INTO `автори_програм` SET `ФІО` =
        ' ".$FIO."', `Кафедра` = ' ".$chair."");
    }

    public function editAuthor($id, $FIO, $chair = null)
    {
        $result = $this->db->query("UPDATE `автори_програм` SET `ФІО` = ' ".$FIO."
        ', `Кафедра` = ' ".$chair." WHERE `автори_програм`.`id` = ' ".$id."");
    }

    public function removeAuthor($id){
        $this->db->query("DELETE FROM `автори_програм` WHERE
        `автори_програм`.`id` = ' ".$id."");
    }

    /* !author block */

    /* remark_nmv block */

    public function getNMVRemarks(){
        $result = $this->db->query("SELECT * FROM `зауваження нмв`")->rows;
        return $result;
    }

    public function getNMVRemar($id){
        $result = $this->db->query("SELECT * FROM `зауваження нмв` WHERE
        `зауваження нмв`.`id` = ' ".$id."")->row;
        return $result;
    }

    public function addNMVRemar($remark, $person)
    {

```

```

$result = $this->db->query("INSERT INTO `зауваження нмв` SET `Зауваження`
= ' ".$remark."', `Експерт` = ' ".$person."");
}

public function editNMVRemar($id,$remark,$person)
{
    $result = $this->db->query("UPDATE `зауваження нмв` SET `Зауваження` =
' ".$remark."', `Експерт` = ' ".$person.'" WHERE `зауваження нмв`.`id` = ' ".$id."");
}

public function removeNMVRemar($id){
    $this->db->query("DELETE FROM `зауваження нмв` WHERE `зауваження
нмв`.`id` = ' ".$id."");
}

/* !remark_nmv block */
/* remark expert block */

public function getExpertRemarks(){
    $result = $this->db->query("SELECT * FROM `зауваження_експертів`")->rows;
    return $result;
}

public function getExpertRemark($id){
    $result = $this->db->query("SELECT * FROM `зауваження_експертів` WHERE
`зауваження_експертів`.`id` = ' ".$id."")->row;
    return $result;
}

public function addExpertRemark($remark, $person)
{
    $result = $this->db->query("INSERT INTO `зауваження_експертів` SET
`Зауваження` = ' ".$remark."', `Експерт` = ' ".$person."");
}

public function editExpertRemark($id,$remark,$person)

```



```

{
    $result = $this->db->query("UPDATE `зауваження_експертів` SET
`Зауваження` = ".$remark.", `Експерт` = ".$person." WHERE
`зауваження_експертів`.`id` = ".$id."");
}

public function removeExpertRemark($id){
    $this->db->query("DELETE FROM `зауваження_експертів` WHERE
`зауваження_експертів`.`id` = ".$id."");
}

/* !remark expert block */

/* programs block */

public function getPrograms($where){
    $result = $this->db->query("SELECT * FROM `програми` ".$where." ORDER
BY `програми`.`Дата (вхід)` ASC")->rows;
    return $result;
}

public function getProgram($id){
    $result = $this->db->query("SELECT * FROM `програми` WHERE
`програми`.`id` = ".$id."")->row;
    return $result;
}

public function addProgram($name, $author, $program_type,
$ed_level,$nmv_remark = NULL,$expert_remark = NULL)
{
    $result = $this->db->query("INSERT INTO `програми` SET
        `Назва_програми` = ".$name.",
        `Автор_програми` = ".$author.",
        `Тип_запису` = ".$program_type.",
        `Освітній рівень` = ".$ed_level.",

```

```

        `зауваження_HMB` = ".$nmv_remark." ,
        `зауваження_експертів` = ".$sexpert_remark." ,
        `Дата (вхід)` = NOW());
    }

    public function editProgram($id, $name, $author, $program_type,
    $ed_level,$nmv_remark = NULL,$sexpert_remark = NULL)
    {
        $result = $this->db->query("UPDATE `програми` SET
            `Назва_програми` = ".$name." ,
            `Автор_програми` = ".$author." ,
            `Тип_запису` = ".$program_type." ,
            `Освітній рівень` = ".$ed_level." ,
            `зауваження_HMB` = ".$nmv_remark." ,
            `зауваження_експертів` = ".$sexpert_remark."
            WHERE `програми`.`id` = ".$id."");
    }

    public function removeProgram($id){
        $this->db->query("DELETE FROM `програми` WHERE `програми`.`id` =
        ".$id."");
    }

    /* !programs block */
    /* ed_plans block */

    public function getEdPlans(){
        $result = $this->db->query("SELECT * FROM `навчальні плани` ORDER BY
        `навчальні плани`.`навчальна_дисципліна` ASC")->rows;
        return $result;
    }

    public function getEdPlan($id){

```

```

$result = $this->db->query("SELECT * FROM `навчальні плани` WHERE
`навчальні плани`.`id` = ".$id."")->row;

return $result;
}

public function addEdPlan($discipline, $number, $total_sem_time,
$total_sem_credit,$total_sem_time_semester,$total_sem_credit_semester,$sed_time_lect
ion,$sed_time_labs,$sed_time_pract,$sed_time_self,$program_type,$sed_level)
{
    $result = $this->db->query("INSERT INTO `навчальні плани` SET
        `навчальна_дисципліна` = ".$discipline." ,
        `Порядковый номер` = ".$number." ,
        `Заг_обсяг_навчальні_години` = ".$total_sem_time." ,
        `Заг_обсяг_кредити` = ".$total_sem_credit." ,
        `години_в_семестрі` = ".$total_sem_time_semester." ,
        `кредити_в_семестрі` = ".$total_sem_credit_semester." ,
        `НВ_лекції` = ".$sed_time_lection." ,
        `НВ_лабораторні` = ".$sed_time_labs." ,
        `НВ_практичні` = ".$sed_time_pract." ,
        `НВ_самостійна_робота` = ".$sed_time_self." ,
        `форма_контролю` = ".$program_type." ,
        `код_кафедри` = ".$sed_level."");
}

public function editEdPlan($id, $discipline, $number, $total_sem_time,
$total_sem_credit,$total_sem_time_semester,$total_sem_credit_semester,$sed_time_lect
ion,$sed_time_labs,$sed_time_pract,$sed_time_self,$program_type,$sed_level)
{
    $result = $this->db->query("UPDATE `навчальні плани` SET
        `навчальна_дисципліна` = ".$discipline." ,
        `Порядковый номер` = ".$number." ,

```



```

        `Заг_обсяг_навчальні_години` = ".$total_sem_time.",
        `Заг_обсяг_кредити` = ".$total_sem_credit.",
        `години_в_семестрі` = ".$total_sem_time_semester.",
        `кредити_в_семестрі` = ".$total_sem_credit_semester.",
        `НВ_лекції` = ".$sed_time_lection.",
        `НВ_лабораторні` = ".$sed_time_labs.",
        `НВ_практичні` = ".$sed_time_pract.",
        `НВ_самостійна_робота` = ".$sed_time_self.",
        `форма_контролю` = ".$program_type.",
        `код_кафедри` = ".$sed_level."
        WHERE `навчальні плани`.`id` = ".$Sid."");
    }

    public function removeEdPlan($id){
        $this->db->query("DELETE FROM `навчальні плани` WHERE `навчальні
        плани`.`id` = ".$id."");
    }

    /* !ed_plans block */

    public function getSQL(){
        $result = $this->db->query("SELECT table_name, round((((data_length +
        index_length) / 1024 / 1024), 2) `Size in MB` FROM information_schema.TABLES
        WHERE table_schema = \"nmv_knteu\"");

        return $result->rows;
    }

    public function sendSQL($sql){
        $result = $this->db->query($sql);
        return $result->rows;
    }
}

```

```
<form action="/ed_plans.php?action=edit" method="post" enctype="multipart/form-
data" id="form" style="overflow: auto;">
```

```
<table class="form">
```

```
<?php
```

```
if(isset($_GET['plan_id']) && !empty($_GET['plan_id'])){
```

```
$plan = $DB->getEdPlan($_GET['plan_id']);
```

```
?>
```

```
<input type="hidden" name="plan_id" value="<?php echo $plan['id']; ?>"/>
```

```
<tbody>
```

```
<tr>
```

```
<td><span class="required">*</span> Порядковий номер:</td>
```

```
<td><input type="text" name="number" value="<?php echo
$plan['Порядковий номер']; ?>" required>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td><span class="required">*</span> Дисципліна:</td>
```

```
<td>
```

```
<select name="discipline">
```

```
<?php $disciplines = $DB->getDisciplines(); ?>
```

```
<?php foreach ($disciplines as $discipline){
```

```
?>
```

```
<option value="<?php echo $discipline['id']; ?>" <?php
if($discipline['id']==$plan['навчальна_дисципліна']) { ?> selected <?php } ?> ><?php
echo $discipline['Назва']; ?></option>
```

```
<?php } ?>
```

```
</select>
```

```
</td>
```

```
</tr>
```

```
<tr>
    <td><span class="required">*</span>Загальний обсяг - навчальні
години:</td>
    <td><input type="text" name="total_sem_time" value="<?php echo
$plan['Заг_обсяг_навчальні_години']; ?>" required>
    </td>
</tr>
<tr>
    <td><span class="required">*</span>Загальний обсяг - кредити:</td>
    <td><input type="text" name="total_sem_credit" value="<?php echo
$plan['Заг_обсяг_кредити']; ?>" required>
    </td>
</tr>
<tr>
    <td><span class="required">*</span>Годин в семестрі:</td>
    <td><input type="text" name="total_sem_time_semester" value="<?php echo
$plan['години_в_семестрі']; ?>" required>
    </td>
</tr>
<tr>
    <td><span class="required">*</span>Кредити в семестрі:</td>
    <td><input type="text" name="total_sem_credit_semester" value="<?php
echo $plan['кредити_в_семестрі']; ?>" required>
    </td>
</tr>
<tr>
    <td><span class="required">*</span>Начальні години - лекції:</td>
    <td>
        <input type="text" name="ed_time_lection" value="<?php echo
$plan['НВ_лекції']; ?>" required>
```



```
</td>
</tr>
<tr>
<td><span class="required">*</span>Начальні години - лабораторні (у.т.ч.
модульний контроль) :</td>
<td>
<input type="text" name="ed_time_labs" value="<?php echo
$plan['HB_лабораторні']; ?>" required>
</td>
</tr>
<tr>
<td><span class="required">*</span>Начальні години - практичні
(семінарські)у.т.ч. модульний контроль:</td>
<td>
<input type="text" name="ed_time_pract" value="<?php echo
$plan['HB_практичні']; ?>" required>
</td>
</tr>
<tr>
<td><span class="required">*</span>Начальні години - самостійна
робота:</td>
<td>
<input type="text" name="ed_time_self" value="<?php echo
$plan['HB_самостійна_робота']; ?>" required>
</td>
</tr>
<tr>
<td><span class="required">*</span>Форма контролю:</td>
<td>
<select name="program_type">
```

```

<?php $form_types = $DB->getFotmTypes(); ?>
<?php foreach ($form_types as $form_type){
    ?>
    <option value="<?php echo $form_type['id']; ?>" <?php
if($form_type['id']==$plan['форма_контролю']) { ?> selected <?php } ?> ><?php echo
$form_type['Назва']; ?></option>
    <?php } ?>
</select>
</td>
</tr>
<tr>
<td><span class="required">*</span>Код кафедри:</td>
<td>
<select name="ed_level">
<?php $chairs = $DB->getChairs(); ?>
<?php foreach ($chairs as $chair){
    ?>
    <option value="<?php echo $chair['id']; ?>" <?php
if($chair['id']==$plan['код_кафедри']) { ?> selected <?php } ?> ><?php echo
$chair['Код_кафедри']. ' : '. $chair['Назва_кафедри']; ?></option>
    <?php } ?>
</select>
</td>
</tr>
<?php } ?>
</tbody></table>
</form>

```