

РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет математики та інформатики

Кафедра інформатики та прикладної математики

«До захисту допущено»

Завідувач кафедри

_____ Батишкіна Ю.В.

протокол № _____

від _____

Дипломний проект (робота)

Ступеня «Магістр»

З спеціальності _____ 122- Комп'ютерні науки _____

(шифр та назва спеціальності)

На тему: Автоматизація роботи наукового відділу Рівненського державного гуманітарного університету _____

Виконав: здобувач освіти 2 курсу, групи КН-М-21

_____ Порохняк Олександр Петрович _____

(прізвище, ім'я, по батькові)

(підпис)

Керівник: Сінчук Алеся Михайлівна, канд.техн. наук, доцент _____

(посади, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент: Шахрайчук Микола Йович, канд.фіз.мат.наук,проф. _____

(посади, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент: Шинкарчук Назар Володимирович, канд.техн.наук,доцент _____

(посади, науковий ступінь, прізвище та ініціали)

(підпис)

Рівне-2021 року

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ I. АНАЛІЗ ІНСТРУМЕНТАРІЮ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ.....	6
1.1. Вивчення можливостей розробки програмного забезпечення.....	6
1.1.1. Система управління базами даних Microsoft SQL Server.....	12
1.1.2. Середовище SQL Server Management Studio.....	19
1.1.3. Середовище розробки Microsoft Visual Studio.....	25
1.2. Розгляд вимог до акредитаційної діяльності в РДГУ.....	32
1.2.1. Постановка технічного завдання.....	37
РОЗДІЛ II. Реалізація технічного завдання.....	38
2.1. Опис структури Баз даних.....	38
2.2. Розробка Баз даних в середовищі SQL Management Studio.....	40
2.3. Розробка Інтерфейсу та логіки програми за допомогою середовища Microsoft Visual Studio.....	45
РОЗДІЛ III. Тестування програми.....	50
3.1. Тестування програмного комплексу.....	50
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДОДАТКИ.....	57

ВСТУП

Актуальність. У сучасному світі інформація є одним із найважливіших елементів життєдіяльності, яка здійснює вплив на суспільство. Збір, обробка, зберігання та подальше використання даних неможливе без застосування новітніх засобів обчислювальної техніки. З кожним роком кількість інформації збільшується стрімкими темпами, що призводить до необхідності пошуку нових автоматизованих методів її збереження, структуризації та групування. Таким середовищем, яке задовольняє вищевказані вимоги є сучасні бази даних.

Попри це, розв'язання даного завдання забезпечить досягнення стратегічної мети нашої держави, яка полягає в тому, щоб побудувати інформаційне суспільство, у якому кожен міг би створювати й накопичувати певну інформацію, мати до неї вільний доступ і можливість користуватися й обмінюватися нею.

Для того щоб розв'язати цю проблему, кожному закладу вищої освіти необхідно здійснити суттєві кроки у напрямі розробки та впровадження такої автоматизованої інформаційно-аналітичної системи, яка б змогла поєднати у собі роботу адміністративних, фінансових, господарських, навчальних та наукових підрозділів. а також сприяла інтеграції всіх процесів, які спрямовані на забезпечення належних умов щодо задоволення інформаційних потреб усіх учасників освітнього процесу, розвитку їх інтелектуального потенціалу, самореалізації і самовдосконалення, а також на якісну підготовку студентської молоді до повноцінної майбутньої професійної діяльності і життя в інформаційному суспільстві.

Дослідженням проблем автоматизації роботи науковим відділом у ЗВО займалися такі вітчизняні та зарубіжні вчені: А. А. Амосов [1], Н. С. Бахвалов [2], А. И. Белоусов[3], А. И. Боровков [5], Р. Виейра [6], М.І. Жалдак [8], О. Ісаєнко [11], Т. Кайт [12], Н. Каліберда [13], Д.І. Карпенко [14], Т. Коваль [18], Е.М. Коломейчик [19], В.В. Крюков [20], О.М. Левченко

[22], Л. Г. Лойцянский [23], К. Нейгл [24], Д. Скит [26], Г. А. Циммерман [27].

Аналіз численних праць вітчизняних і зарубіжних науковців у даній галузі засвідчують наявність значної кількості функціонуючих систем такого призначення, але немає єдиного підходу до проектування якісних автоматизованих систем управління науковим відділом у ЗВО. З огляду на це, проблема створення таких програмних засобів, які дозволяють контролювати освітній процес і сприяють його ефективності, надає актуальності окресленому питанню, як такому, що має соціальну значущість.

Мета магістерської роботи – розробити функціональну базу даних для автоматизованої роботи наукового відділу РДГУ.

Відповідно до мети роботи, було поставлено і виконано такі **завдання**:

- Проаналізувати та охарактеризувати можливості розробки програмного забезпечення;
- проаналізувати інструментарій створення програмного забезпечення: СУБД Microsoft SQL Server, середовище Server Management Studio, середовище розробки Microsoft Visual Studio.
- розробити технічне завдання;
- розглянути вимоги до акредитаційної діяльності в РДГУ;
- створити базу даних наукових співробітників, їх наукових здобутків;
- розробити інтерфейс для користувачів;
- протестувати роботу;

Об’єкт дослідження: процес створення та застосування програмного засобу в роботі наукового відділу РДГУ.

Предмет дослідження: автоматизація роботи наукового відділу Рівненського державного гуманітарного університету.

Наукова новизна дослідження полягає в тому, що визначено та обґрунтовано впровадження програмних засобів у науковий відділ Рівненського державного гуманітарного університету.

Практичне значення дослідження може бути використане у науково-дослідній сфері, у педагогічній діяльності, у навчальному процесі.

Методи дослідження теми. У дослідженні використовуються різноманітні методи, зокрема: методи історизму, наукового аналізу і синтезу (для вивчення аспектів виникнення, формування та розвитку процесів автоматизації інформаційної діяльності у хронологічній послідовності); системного підходу (оскільки увесь матеріал нашого дослідження розглядеться у цілісній системі, і ми за рахунок дослідження одиничних аспектів проблеми доходимо до загальних тенденцій у процесах автоматизації ЗВО); методи обробки, групування і узагальнення даних при дослідженні напрямків застосування автоматизованих технологій та дослідженні ключових аспектів процесів автоматизації).

Апробація результатів дослідження.

XIV Всеукраїнська науково–практична конференція Інформаційні технології у професійній діяльності Рівненського державного гуманітарного університету, яка відбулася 1 листопада 2021 року. Рівне.

Структура роботи. Магістерська робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ I. АНАЛІЗ ІНСТРУМЕНТАРІЮ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ

1.1. Вивчення можливостей розробки програмного забезпечення

Описуючи вивчення можливостей розробки програмного забезпечення, слід зазначити, що можливості сучасних ПК настільки великі, що все більше людей використовують його у своїй роботі, навчанні, житті. Найважливішою якістю сучасного комп'ютера є його спрямованість на зручність «спілкування» з користувачем. Застосовувати можливості комп'ютера стало простим та зрозумілим. Комп'ютер сам підказує користувачеві, що робити в тій чи іншій ситуації, допомагає вийти зі складних ситуацій. Це можливо завдяки комп'ютерному програмному забезпеченню [5, с. 34].

Скориставшись аналогією між комп'ютером і людиною. Новонароджена нічого не знає. Знання та навички приходять саме в процесі розвитку, навчання, накопичення інформації в своїй пам'яті. Комп'ютер, зібраний на заводі з мікросхем, проводів, плат тощо, схожий на новонароджену людину. Варто зазначити, що завантаження в пам'ять комп'ютера програмного забезпечення симетрична процесу навчання дитини. Саме програмне забезпечення створюють програмісти.

Вся сукупність програм, які зберігаються на пристроях довготривалої пам'яті комп'ютера, становить його програмне забезпечення [5, с. 35].

Програмне забезпечення комп'ютера постійно оновлюється, розвивається, удосконалюється. Ціна встановлених програм на сучасному ПК часто перевищує ціну його технічних пристроїв. Розробка сучасного програмного забезпечення вимагає від програмістів надзвичайно високої кваліфікації.

Системні програми призначені для управління роботою комп'ютерної системи, відтворюють різноманітні допоміжні функції (копіювання, видача сертифікатів, тестування, форматування тощо).

Базове ПЗ включає в себе [1, с. 201]:

1. Операційні системи;
2. Оболонки;
3. Мережеві операційні системи.

Сервісне ПЗ включає в себе програми (утиліти):

1. Діагностики;
2. Антивірусні;
3. Обслуговування носіїв;
4. Архівування;
5. Обслуговування мережі.

Головною частиною системного програмного забезпечення є операційна система (ОС). Ось назви деяких поширених ОС для персональних комп'ютерів: MS-DOS, Windows, Linux.

Операційна система - це набір програм, які керують оперативною пам'яттю, процесором, зовнішніми пристроями і файлами, з яким працює користувач.

Операційна система має багато роботи, і вона постійно працює. Наприклад, щоб запустити прикладну програму, її необхідно шукати у зовнішній пам'яті (на диску), помістити в оперативну пам'ять, знайти там вільне місце, «запустити» процесор для запуску програми, контролювати роботу всіх пристроїв машини під час виконання і в разі збоїв вивести діагностичні повідомлення. Усі ці проблеми бере на себе операційна система [1, с. 202].

Якщо повернутися до аналогії і сказати, що системне програмне забезпечення з'єднує всі пристрої та забезпечує безперервний потік інформації між ними, то це є вірне рішення. Більше того, це «нервова система», яка має можливість безперервно стежити за поведінкою

користувача і вчасно забезпечувати його всім необхідним. Тому операційна система забезпечує інтерактивний режим.

Розробка програмного забезпечення для закладів вищої освіти має ряд особливостей, адже це є програмні засоби, основною метою застосування яких є забезпечення навчання за індивідуальними і оптимальними навчальними програмами з керуванням процесу навчання [30].

За призначенням методу можна розрізнити програмне забезпечення різних типів. Оскільки сучасний інформаційний простір надає багато програм, які привертають увагу саме освітян, більшість з яких є низькоякісними, тому дуже важливо створити програмне забезпечення в навчальному процесі безкоштовної освіти, яке має бути конкурентоспроможним не лише в навчанні, а й в інформаційному потоці.

Процес розробки неважливо якого програмного продукту включає етапи постановки завдання, вибору засобів розробки, проектування відповідної моделі, реалізації проекту, тестування, документування та впровадження. Тому зупинимося на особливостях деяких етапів розробки прикладних програм.

В першу чергу потрібно мати на увазі, що саме майбутній програмний продукт повинен зробити працівника ЗВО не об'єктом демонстрації використання ІТ, а активним суб'єктом використання ІТ з метою заохочення пізнавальної діяльності та зацікавленості матеріалом. На етапі постановки завдання доцільно ретельно проаналізувати шляхи зацікавленості працівників освіти, які допоможуть в автоматизації роботи кожного відділу в ЗВО [27, с. 88].

На етапі реалізації проекту необхідно вибрати найбільш підходящі інструменти розробки програмного забезпечення для повного використання сучасних ІТ. Це навіть можуть бути інструменти розробки, доступні викладачам, щоб підвищити інтерес співробітників до процесу створення додатків (наприклад, тимчасових проектів). Варто відзначити, що існує

широкий спектр можливостей для створення мультимедійних додатків з використанням флеш-технологій, а також інструменти дуже зручні.

Дуже важливим є етап тестування та тестування програмних продуктів. Адже саме на цьому етапі він перевіряє доцільність обраних методів, засобів і творчих прийомів у реальному житті. На наш погляд, головним критиком є викладач.

Таким чином процес розробки ПЗ вимагає особливої прицільної уваги розробників до вибору не лише самої моделі, а і шляхів її реалізації [27, с. 89].

Попри популярність спроектованих і впроваджених автоматизованих інформаційно-аналітичних Web-орієнтованих систем управління освітнім процесом у діяльність сучасних ЗВО, слід зазначити, що залежно від рівня розробки, вони мають низку характерних особливостей, а також свої переваги та недоліки, які подано у таблиці 1.1 [15].

Таблиця 1.1.

Особливості розробки автоматизованих систем управління освітнім процесом

Рівень розробки	Недоліки	Переваги
Незалежна розробка	<ul style="list-style-type: none"> – надмірна комерціалізація; – надлишкова універсальність; – складність налаштування; – громіздкий та незрозумілий інтерфейс 	<ul style="list-style-type: none"> – якісна документованість; – сервісна та консультаційна підтримка; – широкий набір функціональних можливостей
Індивідуальна розробка	<ul style="list-style-type: none"> – менша стійкість; – залежність від персонального супроводу своїми розробниками; – неякісна документованість 	<ul style="list-style-type: none"> – точніша відповідність специфічним умовам та вимогам вишу; – більш гнучкіша й оперативніша реакція щодо змін у законодавстві; – зручність для використання персоналом

		певного конкретного закладу вищої освіт
--	--	--

В основі роботи всіх інформаційно-аналітичних Web-орієнтованих систем, спроектованих вищезазначеними установами, лежить один і той же принцип, про який наголошено у низці нормативних документів – принцип доступності до будь-якого роду інформації [15, с. 123].

Найкраще використовувати допоміжні комп'ютерні засоби для контролю процесу розробки програмного забезпечення, щоб ефективно організувати гнучку методологію Scrum. Основні функції цих інструментів включають: здатність планувати завдання, можливість планувати та контролювати виконання завдань, правильний розподіл ресурсів, здатність працювати разом (побудова команд, груп), швидке спілкування, керування системою та документування. Усі перераховані вище функції є частиною викладання технічних предметів. Ми розглянемо популярні програми контролю розробки програмного забезпечення.

Redmine — одне з найкращих і найпопулярніших веб-додатків з відкритим кодом. Його головне призначення — керувати проектом і досягати можливості відстежувати помилки. Веб-додатки виконують багато різних функцій: можливість одночасного запуску кількох проектів; відстеження помилок у виконанні завдань (проектів); можливість побудови діаграми Ганта; ведення власного календаря подій; можливість додавати новини до проект, обробляйте документи та файли, використовуйте канали RSS та електронну пошту для негайного сповіщення про зміни, підтримка множинної автентифікації LDAP, багатомовний інтерфейс; підтримка СУБД MySQL, PostgreSQL, SQLite, Oracle [15, с. 124].

Конкретизуючи зміст вище наведених характеристик якості програмного продукту, сформулюємо такі основні вимоги до програмного забезпечення [9, с. 89]:

- можливість швидкого управління визначеним користувачем завданнями з мінімальними інформаційними та апаратними ресурсами та

програмним забезпеченням (мінімальна кількість операцій, які система використовує для вирішення завдань користувача; логіка та технічна досконалість алгоритмів вирішення проблем);

– можливість задовольнити вимоги користувача щодо якості отриманого за допомогою програмного забезпечення результату та його конфігурації (максимальна відповідність отриманого результату уявленням користувача щодо його якості та форми подання; логічність розбудови архітектури програмного забезпечення та доцільність його використання; естетичність та лаконічність інтерфейсів користувача та адміністратора; можливість подання результатів використання програмного продукту у різних формах, а саме у вигляді тексту, графіків, діаграм тощо);

– узгодженість архітектури конкретного програмного продукту зі стандартною (стандартною) архітектурою існуючого програмного забезпечення (технічна сумісність між пропонованим програмним продуктом та існуючим програмним продуктом може бути використана для вирішення конкретних завдань на рівні допоміжних засобів)

– можливість вдосконалення програмного продукту відповідно до перспективних (очікуваних) змін пріоритетів замовника щодо функціоналу запропонованих розробником алгоритмів (модульна структура програмного забезпечення з можливістю її зміни (доповнення) за можливості включення до неї нових елементів; доцільність використання окремих модулів програмного продукту в межах інших проектів);

– можливість програмного продукту забезпечувати захист і безпеку інформації та архітектури програмного продукту від несанкціонованої зміни їх змісту (захист інформації та алгоритмів роботи програмного продукту від несанкціонованої зміни його конфігурації) тощо [9, с. 90].

Отже, описуючи дослідження можливостей розробки програмного забезпечення, слід зазначити, що програмне забезпечення — це сукупність програм, призначених для вирішення комп'ютерних задач. Програма – це впорядкований набір команд. Програмне та апаратне забезпечення

взаємопов'язані і постійно взаємодіють. Будь-який апаратний пристрій керується програмним забезпеченням. Програмне забезпечення можна розділити на три категорії: системи, програми та інструменти. Проблема розробки програмного забезпечення в ЗВО має багато характерних рис і пов'язана зі зростанням попиту на інформаційно-освітній простір для підвищення якості освіти шляхом залучення ІТ, що відповідає сучасним вимогам до реалізації навчального процесу.

1.1.1. Система управління базами даних Microsoft SQL Server

Описуючи систему управління базами даних Microsoft SQL Server, слід зазначати, що база даних – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує найменше одну з областей застосування. В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних [21, с. 145].

База даних класифікується за різними стандартами. Відповідно до моделі організації даних база даних поділяється на такі категорії:

- Ієрархічна база даних представлена у вигляді дерева, що складається з об'єктів різних рівнів. Між об'єктами існує зв'язок «предок-нащадок». У цьому випадку може виникнути ситуація, коли об'єкт не має нащадків або має кілька нащадків, а об'єкт-нащадок повинен мати лише одного предка.
- Мережна. Така база даних подібна до ієрархічної бази даних, за винятком того, що кожен об'єкт може мати кілька предків.
- Реляційна. Реляційні бази даних зберігають дані у вигляді таблиць.
- Об'єктно-орієнтована. У цьому типі бази даних дані оформляються у вигляді об'єктної моделі.

Система управління базами даних — сукупність програмних і лінгвістичних засобів загального або спеціального призначення, що забезпечують управління створенням та використанням баз даних [7, с. 456].

Основне призначення системи управління базами даних (СУБД) - створити і підтримувати поточний стан бази даних, а також її зв'язок з програмами, що вирішують економічні завдання (прикладні програми користувачів) [7, с. 458].

Завдяки створенню системи управління базами даних спеціального програмного продукту можна організувати єдину базу даних. У світі існує багато СУБД. Хоча вони можуть по-різному обробляти різні об'єкти та надавати користувачам різні функції та інструменти, більшість СУБД базується на наборі встановлених базових концепцій. Це дає нам можливість розглянути систему та поширити її концепції, прийоми та методи на весь клас СУБД. Найпоширенішими базами даних є Visual FoxPro і Microsoft Access [10, с. 231].

Сучасні бази даних – це переважно програми Windows, оскільки це середовище дозволяє використовувати функції персонального комп'ютера (ПК) ширше, ніж DOS (Дискова операційна система). Зменшення вартості потужного ПК не тільки призведе до широкого переходу до середовища Windows, де розробники програмного забезпечення можуть не піклуватися про розподіл ресурсів, але також призведе до зниження цінності програмного забезпечення ПК (особливо СУБД) до апаратних ресурсів.

Сучасні бази даних здебільшого є додатками Windows, оскільки це середовище дозволяє ширше використовувати функції персонального комп'ютера (ПК) порівняно з DOS (Дискова операційна система). Зменшення вартості потужного ПК не тільки призводить до широкого переходу до середовища Windows, в якому розробники програмного забезпечення можуть не піклуватися про розподіл ресурсів, але й зменшує значення програмного забезпечення ПК, особливо СУБД, для апаратних ресурсів.

Можна відзначити, що найбільш яскравими представниками системи управління базами даних є:

- Borland dBase;
- PostgreSQL;
- Microsoft SQL Server;
- Oracle Databas;
- Microsoft Access.

На даний момент не має значення, на якій мові та пакеті написаний той чи інший додаток, і який формат даних у ньому використовується. Тому при сучасній швидкості розвитку інформаційних технологій у програмуванні, все частіше згадується «класична» СУБД, а також мови програмування Visual Basic і Visual C++, вони можуть дозволити швидко створювати необхідні компоненти програми. швидкість дуже важлива, це важко. Іноді неможливо розробити через «класичну» СУБД. Сучасні методи управління базами даних також передбачають широке використання технології «клієнт-сервер».

Microsoft SQL Server — це система керування базами даних, розроблена Microsoft. Сервер даних виконує основну функцію зберігання та надання даних у відповідь на запити інших програм, які можуть працювати на тому ж сервері або в мережі.

Microsoft SQL Server використовує версію SQL під назвою Transact-SQL (скорочено від T-SQL) як мову запитів і є реалізацією SQL-92 (стандарту ISO для SQL) з багатьма розширеннями. T-SQL забезпечує використання додаткового синтаксису для збережених процедур і забезпечує підтримку транзакцій (взаємодія з базою даних, яка керує програмою). Microsoft SQL Server і Sybase ASE використовують для взаємодії з мережею протокол рівня програми, який називається Tabular Data Stream (TDS, протокол передачі табличних даних).

SQL Server – це поширена програма, яка може працювати на комп'ютерах, серверах, ноутбуках, віртуальних машинах і навіть у хмарі.

Він дозволяє підключатися до нього локально або через мережу, відправляти команди та отримувати відповіді за спеціальним протоколом TDS. Це потужна та сучасна платформа, яка дозволяє зберігати та обробляти дані. Все, що він робить, це відкриває мережевий порт і готується отримувати команди від користувача, а потім відповідати. Найважливіше під час використання SQL Server – це мати можливість підключатися до нього та виконувати команди, коли це необхідно, тому що, очевидно, якщо це не працює, інакше це безглуздо [29].

При роботі в локальній мережі програму необхідно встановити на робочому місці кожного користувача. Режим роботи залежить від обраної бази даних. У найпростішому випадку завдання полягає у використанні бази даних Microsoft Access. Цей режим використовується за замовчуванням відразу після встановлення програми і підходить для більшості програм, якщо кількість користувачів не перевищує 10-12 осіб. У мережевому середовищі файл бази даних знаходиться на загальнодоступному ресурсі, а підключення до ресурсу надходить від користувачів інших комп'ютерів, на яких встановлено необхідне програмне забезпечення.

Коли одну базу даних одночасно використовують більше 12 користувачів або швидкість сервера значно гірша, має сенс перевести базу даних у формат Microsoft SQL. Це зріла архітектура клієнт-сервер, в якій мережа сама не витягує таблицю представлення бази даних, а повертає результати певного запиту. Усі обчислення виконуються на сервері бази даних, а не на клієнтському комп'ютері. Швидкість значно збільшиться, що особливо важливо при великих обсягах роботи.

У 2019 році з'явилась MS SQL Server 2019. Розробники MS SQL Server більше не позиціонують свій продукт з простою системою керування базами даних. Основна мета даної версії MS SQL Server – стати системою для роботи з Big Data [4, с. 101].

Окремо слід виділити продукт PostgreSQL. Функціональність цієї бази даних не така хороша, як у Microsoft SQL Server. Однак, оскільки вона

безкоштовна, ця база даних стала особливо популярною в останні роки. Крім того, PostgreSQL дозволяє розробникам створювати нові типи даних і розробляти компоненти. Наприклад, існує компонент, розроблений для PostgreSQL, який дозволяє використовувати відеокарти Nvidia для обробки даних. Розрахунок проводиться за технологією CUDA. З одного боку, здатність писати власні компоненти надає програмістам багато можливостей, з іншого боку, підтримувати ці компоненти іноді буває дуже складно. Немає гарантії, що наступна версія продукту підтримуватиме компоненти минулої версії.

Компанія Oracle не поступалася Microsoft, а купила продукт MySQL, що дало їй додаткові можливості для її розвитку. Багато технологій компанії Microsoft мають аналоги в продуктах компанії Oracle.

Окремим напрямком розвитку є хмарні технології. Спрощує передачу всіх завдань керування та виконує лише програми для збільшення обчислювальної потужності. Адже в таких умовах почали розвиватися CosmosDB від Microsoft та DynamoDB від Amazon [4, с. 102].

СУБД Oracle зарекомендувала себе як хороший спосіб розробки великих баз даних, оскільки персональні комп'ютери є частиною обчислювальної потужності підприємства чи організації. Oracle може запускати будь-яку операційну систему: Windows і Unix-подібну. Це робить його більш гнучким і адаптованим до будь-якої інформаційної системи. Її неодноразово називали однією з найпотужніших баз даних на ринку програмного забезпечення. Також слід додати, що Oracle-система не є безкоштовною [12, с. 345].

Firebird — це компактна, кросплатформна, безкоштовна система керування базами даних, яка працює на Linux, Microsoft Windows та різних платформах Unix. Firebird також є сервером баз даних. Один сервер Firebird може обробляти сотні незалежних баз даних. Навіть для комерційного використання це абсолютно безкоштовно. Firebird найбільш широко використовується в операційних системах Linux і Microsoft Windows [28].

База даних складається з механізму бази даних, фактичної бази даних і прикладних програм, які використовуються для керування даними та компонентами реляційної бази даних. Microsoft SQL Server керує даними у вигляді зв'язаних таблиць, що складаються з рядків і стовпців. База даних SQL Server — це реляційна база даних, сумісна з інтегрованою підтримкою SQL XML для Інтернет-додатків. Однак, на відміну від Firebird, ця система управління базами даних є платною [6, с. 401].

Використовуючи постійну семантику запиту або методи виконання запитів для зміни синтаксичної структури запитів SQL, ми можемо побачити збільшення швидкості отримання даних. Відповідних змін можна досягти шляхом рефакторингу програмного коду, запиту інструкцій щодо виконання планів або використання інформації служби бази даних (метаданих). Ці інструменти вимагають багато часу та фінансових витрат, крім того, певні методи можуть зашкодити загальній продуктивності IP. З метою збереження цілісності системи запропоновано спосіб заміни SQL-запиту на етапі отримання бази даних.

Реалізація цього методу вимагає використання спеціального програмного забезпечення (ПЗ). Характеристики програмного забезпечення: не залежить від моделі даних бази даних; адаптованість до різних реляційних баз даних (Firebird 1.5, Firebird 2.1.4, MySQL); не залежить від типу, версії та кількості клієнтського програмного забезпечення.

В даний час існує безліч програмних рішень, які можуть перехоплювати запити на виконання бази даних [6, с. 406]:

- UIB SQL Monitor;
- FBHook;
- Sinatica Monitor for Firebird;
- JdbMonitor;
- SQL Server Profiler.

Усі вищевказані програмні реалізації зазвичай використовуються для моніторингу та аналізу запитів SQL, але програмне забезпечення не дозволяє редагувати запити, тому що для реальної реалізації цієї мети вам слід розробити та впровадити власний алгоритм перехоплення SQL-запитів, який буде відповідати наступним вимогам:

- перехопити запит перед виконанням в СУБД;
- можливість редагування та заміни запитів на його виконання в базі даних;
- перехоплювати та редагувати запити в режимі реального часу.

Алгоритм перехоплення запитів SQL перед виконанням СУБД [6, с. 407]:

1. Перевірте запущені компоненти сервера.
2. Встановіть та перехопіть WinSock API-реалізацію функцій за допомогою віддаленої потокової передачі.
3. Дочекайтеся запиту від клієнта.
4. У разі запиту клієнта отримати SQL-запит і замінити його в базу даних.
5. Виконати запит СУБД.
6. Надіслати результат виконання клієнтській частині.

Тому, описуючи управління базою даних Microsoft SQL Server, слід зазначити, що SQL Server – це програма загального призначення, яка працює на комп'ютерах, серверах, ноутбуках, віртуальних машинах і навіть у хмарі. Він дозволяє підключатися до нього локально або через мережу, відправляти команди та отримувати відповіді за спеціальним протоколом TDS. Це потужна та сучасна платформа, яка дозволяє зберігати та обробляти дані. Все, що він робить, це відкриває мережевий порт і готується отримувати команди від користувача, а потім повертає результат.

1.1.2. Середовище SQL Server Management Studio

SQL Server Management Studio (SSMS) — це нова інтегрована програма, яка надає доступ до більшості графічних інструментів, які можна використовувати для виконання завдань керування та розробки на SQL Server. SSMS є заміною Enterprise Manager, Query Analyzer та Analysis. Менеджер доступний у SQL Server 2000. Корпорація Майкрософт об'єднає всі ці інструменти в один, зосереджуючись на забезпеченні інструментів, які відповідають потребам розробників і адміністраторів баз даних (DBA).

SSMS — це складний інструмент, який забезпечує точку входу майже для всіх функцій SQL Server. SSMS надає занадто багато функцій, щоб їх можна було розглянути в одній главі. Мета цього розділу — надати базовий огляд SSMS, зосередившись на нових можливостях SQL Server. Решта книги розуміє компоненти SSMS та надає більш детальну інформацію [10, с. 421].

Основним інструментом SQL Server Management Studio є Object Explorer, який дозволяє переглядати, видаляти і комплексно керувати об'єктами сервера.

Також є SQL Server Management Studio Express для експрес-сервера, який є безкоштовною. Однак у неї немає жодної допомоги від ряду компонентів (Analysis Services, Integration Services, Notification Services, Reporting Services) і SQL Server 2005 Mobile Edition.

Середовище опрацювання SQL Server Management Studio складається з декількох компонентів, що використовуються для адміністрування і керування всією системою. Основні такі компоненти наведені в наступному списку:

- Registered Servers (Зареєстровані сервери);
- Object Explorer (Оглядач об'єктів);
- Query Editor (Редактор запитів);
- Solution Explorer (Оглядач рішень).

Під час запуску середовища SQL Server Management Studio відкривається діалогове вікно Connect to Server (З'єднання з сервером), в ньому потрібно встановити необхідні параметри для підключення до сервера [10, с. 424]:

Server type (Тип сервера). Ви можете використовувати SQL Server Management Studio для керування об'єктами в системі баз даних і службах Analysis Services. Ми розглянемо лише використання SQL Server Management Studio для керування об'єктами системи баз даних.

Server name (Ім'я сервера). Виберіть зі списку або введіть з клавіатури ім'я сервера, до якого потрібно підключитися. (Зазвичай, середовище SQL Server Management Studio можна підключити до будь-якого встановленого продукту на певному сервері.)

Authentication (Перевірка справжності). Виберіть один із наступних двох типів автентифікації:

Windows Authentication (Перевірка автентичності Windows). Підключитися до SQL Server за обліковим записом Windows. Це найкращий варіант підключення, що рекомендується компанією Microsoft;

SQL Server Authentication (Перевірка справжності SQL Server). Використовуйте автентифікацію системи баз даних за допомогою імені користувача та пароля.



Рис.1.1. Підключення до SQL Server

Після вказівки цих необхідних налаштувань натисніть кнопку «Підключити», і механізм баз даних підключиться до вказаного сервера. Після підключення до сервера баз даних відкриється головне вікно SQL Server Management Studio. За зовнішнім виглядом це вікно схоже на головне вікно середовища розробки Visual Studio 2010, тому користувачі застосовують досвід Visual Studio в цьому середовищі. На наступному малюнку показано головне вікно SQL Server Management Studio з кількома панелями:

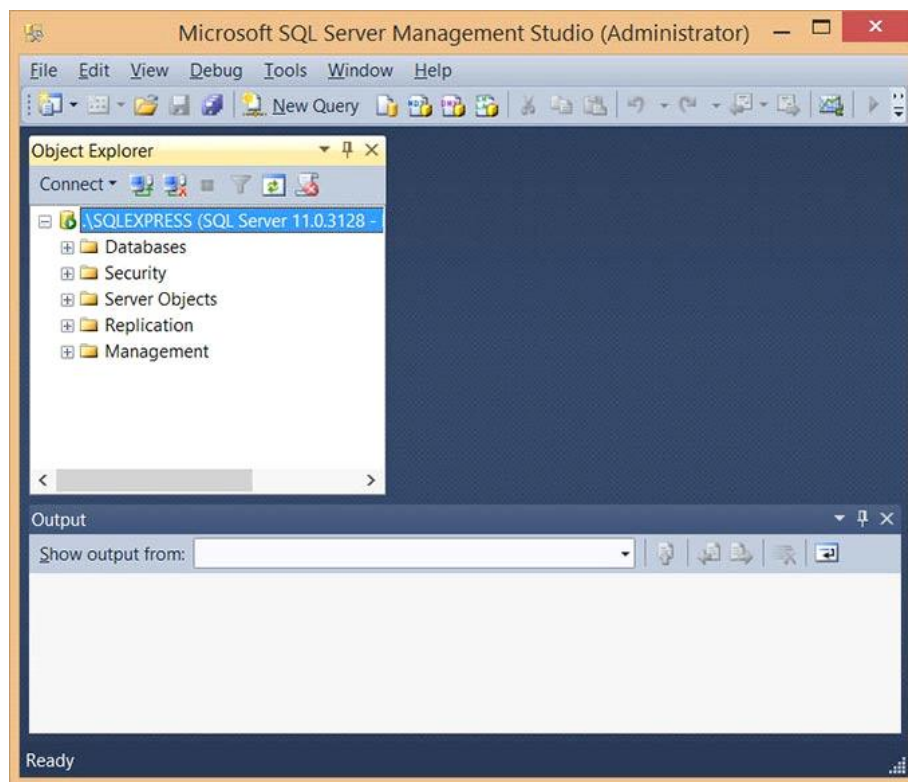


Рис.1.2. Середовище SQL Server Management Studio

SQL Server Management Studio надає єдиний інтерфейс для керування серверами та запитамі для всіх компонентів SQL Server. Іншими словами, механізм баз даних, служба аналізу, служба інтеграції та служба звітності використовують один і той же графічний інтерфейс.

Компонент Registered Servers. Компонент Registered Servers (Зареєстровані сервери) наведені у вигляді панелі, що дозволяє працювати з

серверами, які вже використані. Якщо панель Registered Servers відсутня, її можна відкрити, вибравши її назву в меню View. Ви можете використовувати ці з'єднання для перевірки стану сервера або керування його об'єктами. Кожен користувач використовує окремий зареєстрований список серверів, який зберігається локально [21, с. 89].

Ми можемо додати нові сервери до списку або видалити наявні сервери зі списку. Сервери можна сортувати за групами. Кожна група повинна розміщувати сервери з логічними з'єднаннями. Сервери також можна згрупувати за типами, наприклад, сервери для баз даних, служби аналізу, служби звітності та служби інтеграції.

Компонент Object Explorer. Панель Object Explorer (Обозреватель об'єктів) є у вигляді дерева представлення всіх об'єктів баз даних сервера. Якщо панель Object Explorer не відображена, її можна відкрити, вибравши послідовність команд з меню View --> Object Explorer. Ця деревоподібна вистава відображає ієрархію об'єктів на сервері. Якщо її розгорнути, буде показано логічну її структуру відповідного сервера.

Організація панелей середовища SQL Server Management Studio та переміщення по них. Усі панелі середовища SQL Server Management Studio доцільно закріплювати у головному вікні або приховувати з поля зору. Рекомендується виправити всі панелі SQL Server Management Studio в головному вікні або приховати їх. Клацніть правою кнопкою миші рядок заголовка панелі, щоб надати панелі такі параметри відображення:

Floating (Плаваюча область) - панель вільно переміщується поверх інших панелей середовища SQL Server Management Studio, і її можна залишити в будь-якому місці на екрані [21, с. 90];

Dockable (Закріпити) – панель можна переміщати і закріплювати на місці. Щоб перемістити панель у потрібне положення, клацніть її рядок заголовка, а потім перетягніть панель у потрібне положення, не відпускаючи кнопку миші;

Tabbed Document (Закріпити як вкладки) - панелі можна організувати у вигляді вкладок документа, коли стан панелі змінюється із вкладки, що закріплюється на документ;

Hide - ми можемо приховати панель. Крім того, ми можемо приховати панель, натиснувши хрестик у верхньому правому куті. Щоб знову показати закриту панель, виберіть її назву в меню «Вид»;

Auto Hide (Автоматично приховувати) – Панель згорнута і прикріплена як вкладка в лівій частині екрана. Щоб відкрити таку панель, наведіть курсор миші на вкладку в лівій частині екрана і тримайте панель відкритою, натисніть кнопку на кнопці у верхньому правому куті панелі [21, с. 91].

SQL Server Management Studio дозволяє виконувати одне і те ж завдання кількома способами. Різним людям подобаються різні методи – декому подобається двічі клацати, комусь подобається натискати піктограми «+» і «-», хтось використовує праву кнопку, хтось отримує доступ до меню, хтось любить ярлики тощо. Гуляйте в навколишньому середовищі і пробуйте різні способи, поки не виберете найбільш підходящий спосіб.

У Analysis Services ви можете створювати проекти сценаріїв для Analysis Server у SQL Server Management Studio та групувати пов'язані сценарії для спільної розробки, керування та контролю версій. Якщо проект наразі не завантажено в SQL Server Management Studio, нове рішення буде створено автоматично, коли буде створено новий проект сценарію сервера аналізу даних. В іншому випадку ви можете створити або додати новий проект сценарію сервера аналізу даних у нове рішення [29].

Для того щоб створити проект сценаріїв сервера аналізу даних у SQL Server Management Studio використовуйте такі основні кроки:

- У меню Файл вкажіть «Створити», а потім натисніть «Проект».
- Виберіть шаблон проекту Скрипти служб Analysis Server , а потім вкажіть ім'я та розташування нового проекту.

Клацніть правою кнопкою миші елемент З'єднання , щоб створити нове з'єднання в папці «З'єднання» у проекті «Скрипти служб Analysis Server» у браузері рішень.

Ця папка містить рядки підключення для екземплярів служб Analysis Services , щодо яких можуть виконуватися скрипти, які містяться у проекті скриптів Analysis Server. У проекті скриптів сервера аналізу даних можуть використовуватися кілька з'єднань, і можна вибирати з'єднання, щодо якого запускати скрипт, що міститься в проекті під час виконання.

- Клацніть правою кнопкою миші елемент Запити , щоб створити скрипти багатовимірних виразів, розширень інтелектуального аналізу даних та XML для аналітики (XMLA) у папці «Скрипти» проекту скриптів служб Analysis Server у браузері рішень.

- Клацніть правою кнопкою миші проект, вкажіть пункт Додати та виберіть пункт Існуючий елемент, щоб додати інші файли, наприклад текстові файли, що містять примітки до проекту, у папці Різне проекту «Скрипти служб Analysis Server» у браузері рішень. Ці файли не враховуються середовищем SQL Server Manag

Таким чином, SQL Server Management Studio (SSMS) є інтегрованим середовищем для управління будь-якою інфраструктурою SQL. Використовуйте SSMS, щоб отримати доступ до всіх компонентів SQL Server, SQL Azure та Azure Synapse Analytics, налаштувати й керувати ними, а також керувати ними. SSMS надає єдину повнофункціональну допоміжну програму, яка поєднує в собі велику кількість графічних інструментів і безліч чудових редакторів сценаріїв для всіх професійних розробників і адміністраторів баз даних для доступу до SQL Server.ement Studio [29].

1.1.3. Середовище розробки Microsoft Visual Studio

Microsoft Visual Studio — це інтегроване середовище розробки (IDE) Microsoft. Він використовується для розробки комп'ютерних програм, а також веб-сайтів, веб-додатків, веб-сервісів та мобільних додатків. Visual Studio використовує платформи розробки програмного забезпечення Microsoft, такі як Windows API, Windows Forms, Windows Presentation Foundation, Windows Store і Microsoft Silverlight. Він може генерувати користувацький код і керований код. Приклад середовища показано на рисунку 1.3.

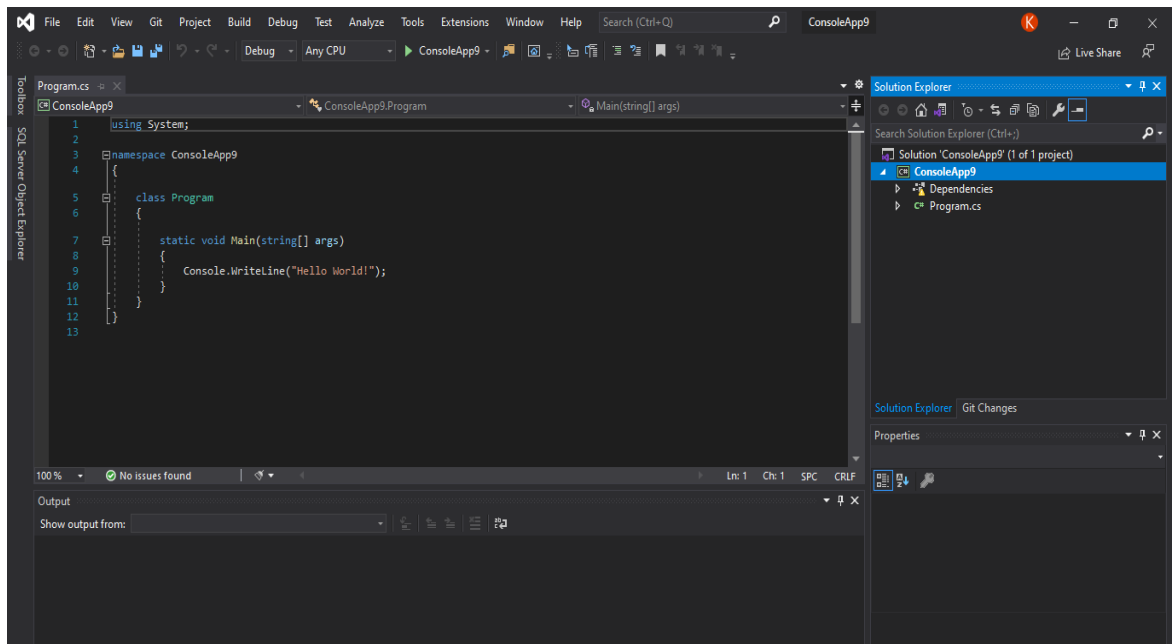


Рис. 1.3. Середовище Microsoft Visual Studio

Продукт, присвячений розробці настільних і серверних додатків Windows, інтегроване середовище розробки (IDE) Microsoft Visual Studio все більше і більше нагадує Swiss Army Knife, здатний підтримувати кілька обчислювальних платформ, мов і середовищ виконання.

Це перша версія IDE, яка включає компілятор Roslyn нового покоління, - сказав С. Сома Сомасегар, віце-президент з розвитку Microsoft.

Roslyn розроблено Microsoft і надає багато розширених функцій налагодження для мов програмування C# і Visual Basic [25, с.321].

Аналітик програмного забезпечення IDC Аль Хілва написав у електронному листі, що вдосконалення, розроблені Roslyn, є «одною з найважливіших нових функцій для професійних розробників». Наприклад, функція лампочки на основі Roslyn може активно пропонувати способи відновлення пошкодженого коду або покращення робочого коду. Аналітик програмного забезпечення IDC Аль Хілва написав у електронному листі, що вдосконалення, розроблені Roslyn, є «одною з найважливіших нових функцій для професійних розробників».

Наприклад, функція лампочки на основі Roslyn може активно пропонувати способи відновлення пошкодженого коду або покращення робочого коду.

Крім того, завдяки Roslyn Visual Studio також дозволяє розробникам налаштовувати сповіщення та рекомендації. Командні проекти можуть використовувати цю функцію для створення набору сповіщень, щоб запобігти розробникам типових помилок або порушення організаційних інструкцій [26, с. 209].

Visual Studio дає змогу розробнику написати єдину програму для роботи на кількох платформах Windows, таких як мобільний, настільний і навіть експериментальний середовище Microsoft HoloLens. Він також забезпечує спосіб створення додатків, які взагалі не працюють на комп'ютерах Windows, але замість цього вони працюють на пристроях iOS або у веб-додатках у хмарі.

Visual Studio містить редактор коду, який підтримує IntelliSense (компонент завершення коду) і рефакторинг коду. Вбудований налагоджувач працює як налагоджувач на рівні виводу та як налагоджувач на рівні машини. Інші вбудовані інструменти включають програми кодування, конструктори форм для побудови графічних інтерфейсів, веб-дизайнери, дизайнери класів і схем баз даних. Він використовує плагіни, які

покращують функції майже на всіх рівнях, включаючи підтримку систем управління ресурсами (таких як Subversion і Git) і додавання нових наборів інструментів, таких як редактори мов і розробники візуалізації або інші аспекти періоду розробки програмного набору інструментів (наприклад, Team Foundation Server: Team Explorer).

Visual Studio дозволяє розробляти програми з використанням різних мов програмування: Visual C#, Visual Basic, Visual F#, Visual C++, Python тощо. (рисунок 1.4). Є можливість не тільки розробляти програми для Windows, а також можн розробляти програми для інших популярних платформ: Android, iOS [29].

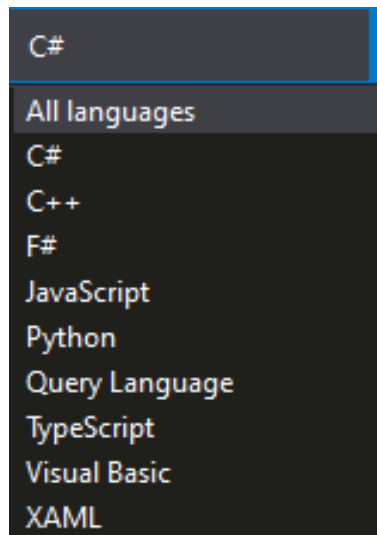


Рис.1.4. Мови програмування: Visual C#, Visual Basic, Visual F#, Visual C++, Python і т.д.

Visual Studio Community є повністю безкоштовним для студентів і розробників програмного забезпечення з відкритим кодом.

Для того, щоб створити проект потрібно натиснути «Файл — Создать — Проект» (или «File — New — Project» (рис. 1.5):

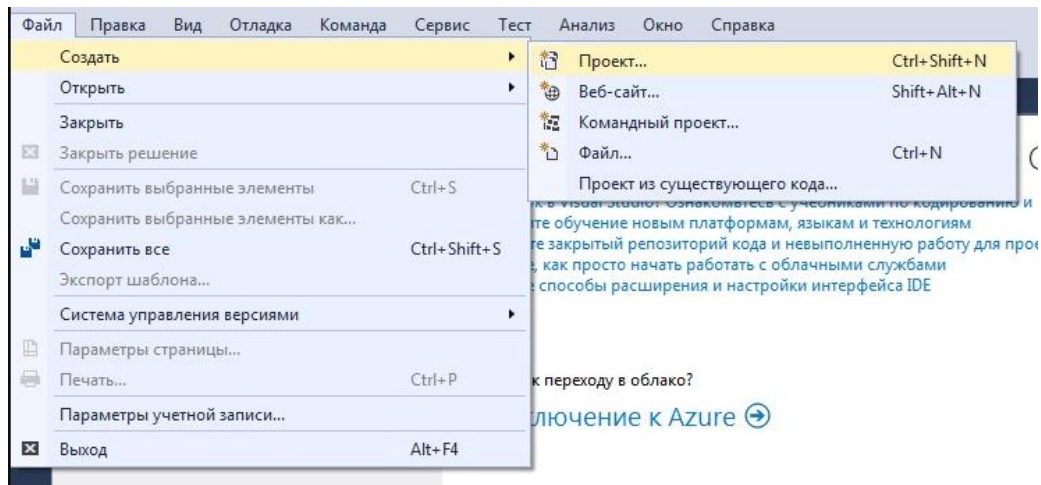


Рис.1.5. Створення програм у Visual Studio

Повинно відкритися вікно «Створення проекту» (рис. 1.6), в якому потрібно:

- у лівій області вибрати "Шаблони - Visual C # - Windows";
- у головній робочій області вибрати «Консольний додаток»;
- у текстовому полі "Ім'я" вказати назву проекту (у нашому випадку це буде "FirstProject");
- у текстовому полі "Розташування" вказати папку, де зберігатимуться проекти додатків (у моєму випадку це "D:\Програмування\Projects");
- натиснути кнопку «ОК».

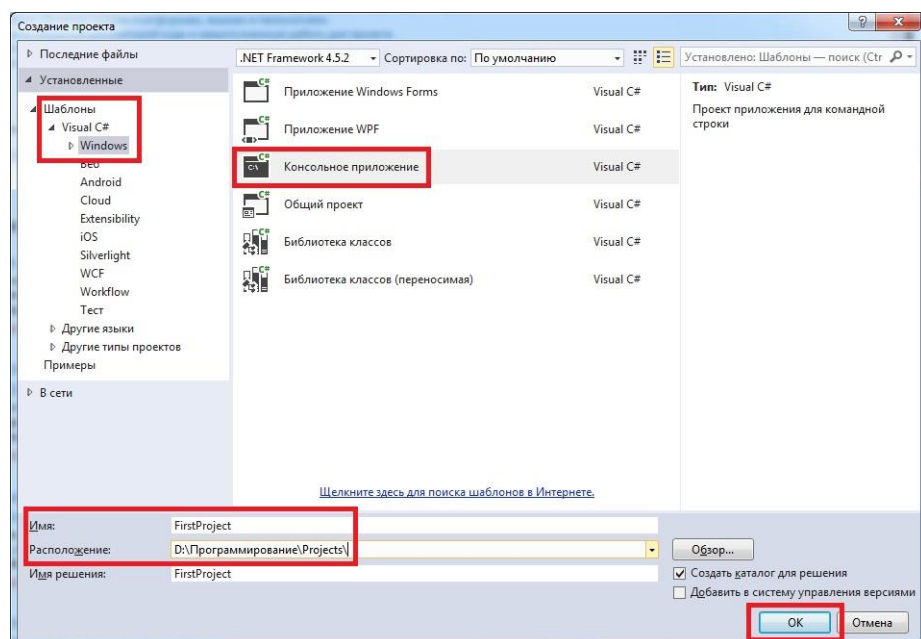


Рис.1.6. Вікно створення нового проекту Visual Studio

У головній робочій області вже є код програми, це стандартна заготовка, яку створює Visual Studio за замовчуванням (рис. 1.7). Ця заготовка поки що нічого не робить, і нам потрібно її доопрацювати. Для цього в робочій області знайдемо основний «метод» (з точки зору об'єктно орієнтованого програмування), цей метод називається Main. Метод Main це так звана точка входу програми, будемо вважати, що це перше місце, яке буде виконуватися в програмі!

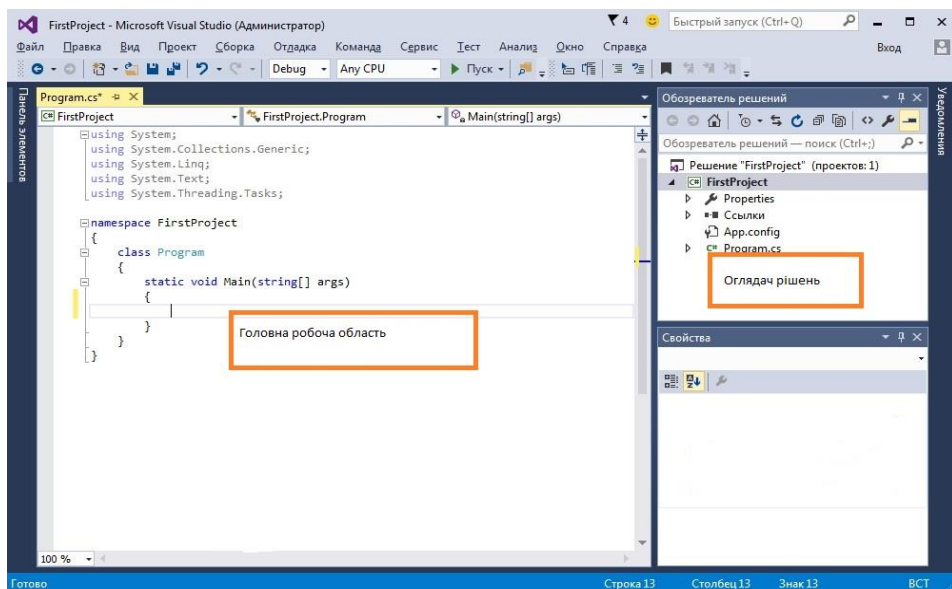


Рис.1.7. Головне вікно Visual Studio

Доповнимо наш код, так як показано нижче (доданий рядок виділено):

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace FirstProject
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Hello, World!");
14         }
15     }
16 }

```

Згенеруємо наш код програми! Йдемо в головне меню Visual Studio і вибираємо пункт «Сборка — Собрать решение» (или як в англійській версії Visual Studio «Build — Build Solution») (рис.1.8).

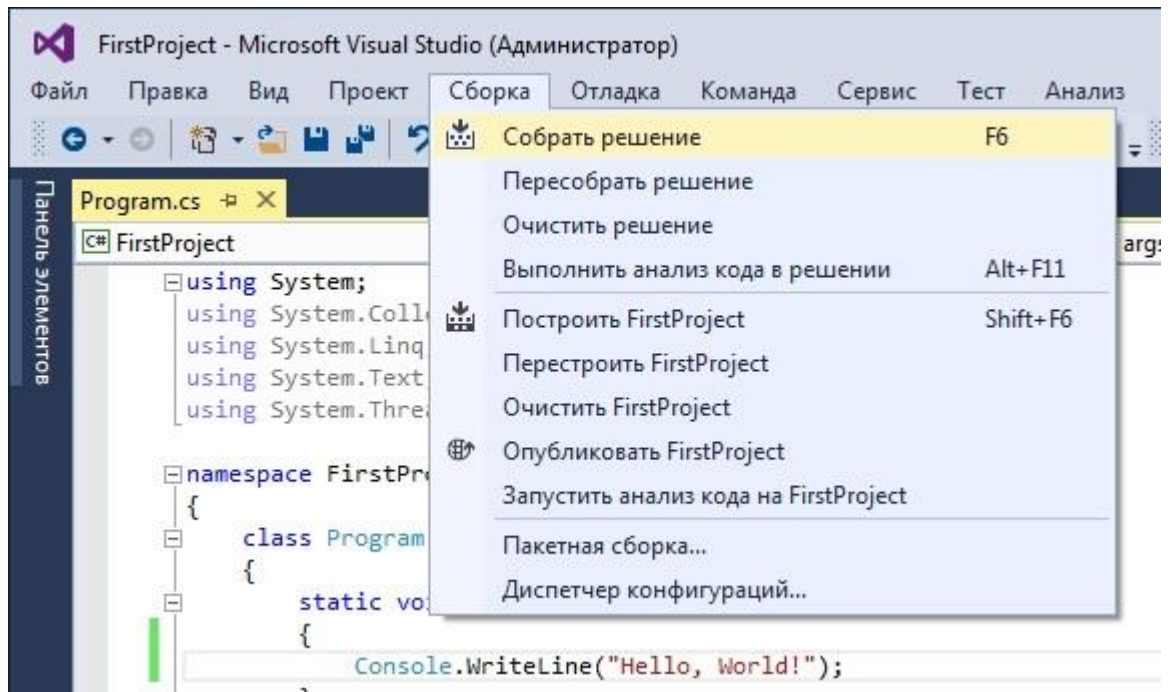


Рис.1.8. Сборка первого проекта Visual Studio

Тепер залишилося тільки запустити наш додаток, для цього знову йдемо в головне меню і вибираємо пункт «Отладка — Запуск без отладки» (или «Debug — Start Without Debugging» в английской версии студии) (рис. 1.9.).

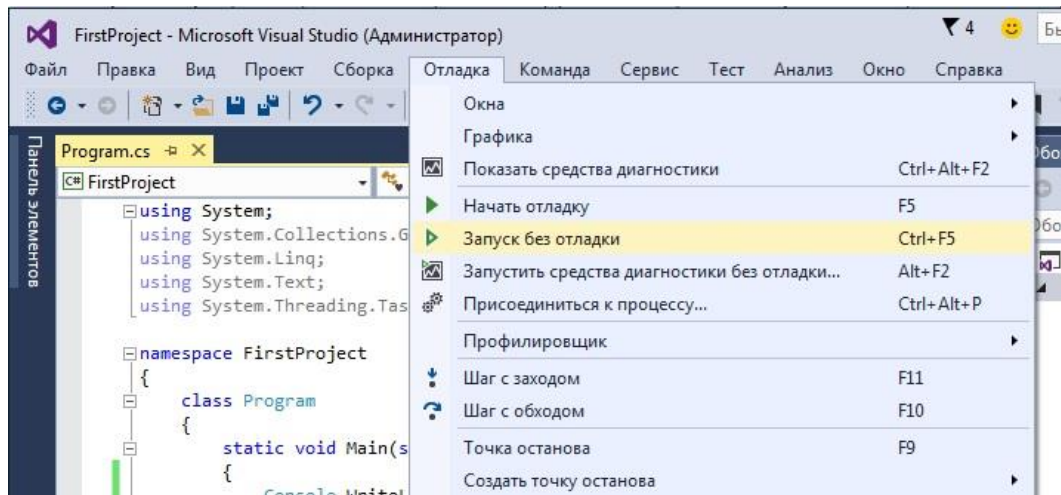


Рис.1.9. Запуск программы через головное меню Visual Studio

В результаті має з'явитися вікно такого вигляду (рис. 2):

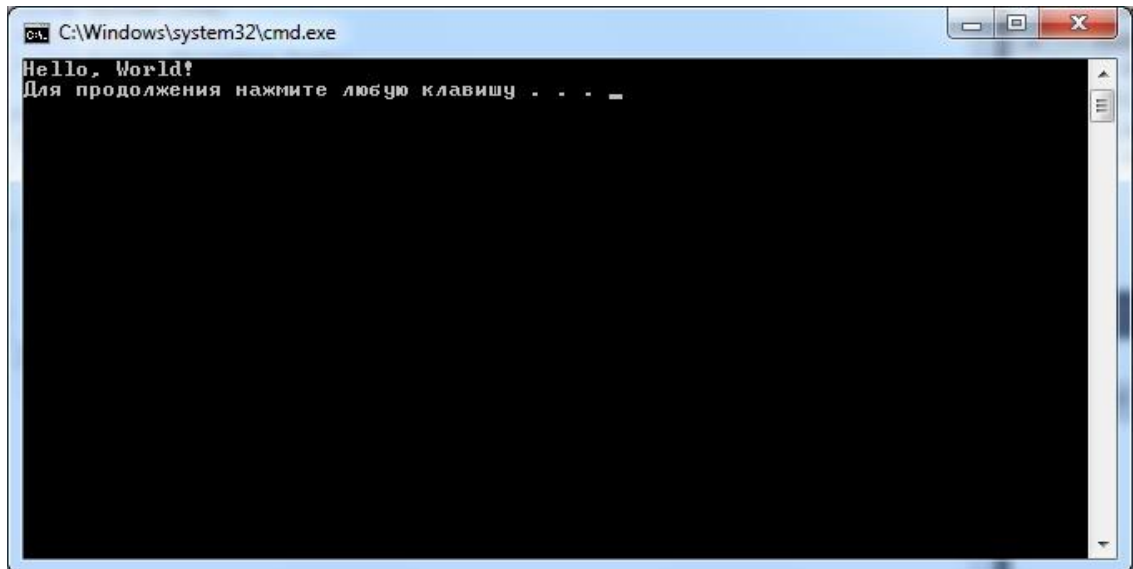


Рис.2. Консольний додаток Visual Studio

У Visual Studio є фонові компіляції (також відома як додаткова компіляція). Коли ви пишете код, Visual Studio компілює його у фоновому режимі, щоб надати зворотній зв'язок щодо синтаксису та помилок компіляції, позначених червоним хвилястим підкресленням. Попередження виділені зеленим кольором. Тональна компіляція не генерує виконуваний код, оскільки для генерації виконуваного коду потрібен компілятор, а не компілятор. Компіляція довідки Microsoft Visual Basic спочатку була представлена, але тепер поширюється на всі включені мови.

Отже, Microsoft Visual Studio — це серія продуктів Microsoft, включаючи інтегроване середовище розробки програмного забезпечення та багато інших інструментів. Ці продукти дозволяють розробляти консольні та графічні додатки, включаючи підтримку технології Windows Forms, а також веб-сайти, веб-додатки у рідному та керованому коді для всіх платформ, які підтримуються Microsoft Windows, Windows Mobile, Windows Phone та Windows. Веб-сервіс CE, .NET Framework, .NET Compact Framework.

1.2. Розгляд вимог до акредитаційної діяльності в РДГУ

Учасників акредитації освітньо-професійних програм (ОПП) у ЗВО ділять на кілька категорій:

- 1) національні установи (відомчі експертні групи (комітети), секретаріати національних установ та їх співробітники);
- 2) ЗВО (науково-дослідні установи) та їх представники, які здійснюють освітню діяльність у сфері вищої освіти, ректор університету (науково-дослідних установ), гаранті освітньо-професійних програм;
- 3) інші зацікавлені сторони (здобувачі освіти (зокрема, студентське самоврядування) та їх батьки, працівники тощо).

Права РДГУ (наукової установи) [31]:

- В період з 1 червня по 1 серпня кожного року в електронному вигляді повідомляти національний заклад про намір сертифікувати ОПП певної спеціальності на наступний навчальний рік.

- Заздалегідь почати підготовку до чергового проходження акредитації.

- Згідно з підтвердженими заявами РДГУ, особливо якщо конфлікт інтересів або обставини виправдовують розумне припущення про упередженість експерта, останнього може бути відкликано з комісії та одночасно замінено іншим експертом.;

- Проведення одночасної акредитації групи освітніх програм у межах спеціальності.

Обов'язки РДГУ [31]:

- представники акредитаційного відділу РДГУ сприяють сертифікації та діють сумлінно у відносинах з національними установами, їх галузевими експертними радами та експертами;

- РДГУ, затверджує навчальні плани та подає матеріали до національних закладів для атестації в електронному вигляді;

- Виконує вимоги плану роботи експертної групи;

- Публікувати інформацію про самооцінку освітніх проектів, рішення про атестацію чи відхилення, відповідні висновки експертів ГЕР, звіти експертних груп на офіційному веб-сайті РДГУ.;

- Оплачує вартість акредитаційних процедур.

Ректор РДГУ, права [32]:

-Гарант ОПП визначається його наказом (у разі відсутності відповідного наказу гарантом освітньої програми є особа, відповідальна за проектну групу та призначається РДГУ відповідно до Ліцензійних умов навчального закладу);

- Має право узгоджувати з керівником експертної групи виїзну акредитаційну частину процедури - отримувати звіт експертної групи в електронній формі протягом двох робочих днів;

- Протягом трьох робочих днів з дати отримання звіту надати Національному агентству обґрунтовані коментарі до звіту, які можуть включати інформацію про порушення експертною групою процесу сертифікації.;

- Запрошується брати участь на засіданнях ГЕР та нарадах Національних установ, під час яких будуть розглянуті відповідні справи про акредитацію.

Перелік вимог для акредитації, запитання для аналізу та орієнтовний перелік документів, що підтверджують виконання того чи іншого критерію.

Проаналізуємо вимоги до акредитаційної діяльності [30]:

Вимога 1. Право на акредитацію освітньої програми.

Освітня програма може бути акредитована у випадку якщо навчальний заклад, який приймає участь у її реалізації має ліцензію на здійснення освітньої діяльності у сфері вищої освіти.

Вимога 2. Створення освітньої програми, визначення компетентностей та результатів навчання.

У плані освіти є затверджена вищим навчальним закладом заява, в якій чітко прописані цілі, які сприяють реалізації місії та стратегії РДГУ. Опис

плану навчання відповідає потребам та вимогам здобувачів вищої освіти, враховуючи при цьому потреби всіх зацікавлених сторін. Основні зацікавлені сторони залучаються до визначення цілей плану та результатів його навчання, а також подальшого вдосконалення плану.

Вимога 3. Навчальний план, його структура, навантаження та можливості академічної мобільності студентів.

План і навчальна програма структуровані таким чином, щоб студенти могли досягати цілей і результатів навчання плану. План поділяється на навчальні частини (дисципліни, практики, модулі тощо), і кожна частина забезпечує досягнення загальних цілей плану та запланованих результатів навчання..

Вимога 4. Умови допуску до освітньої програми.

Вимоги та процедури вступу офіційно визначені, прозорі та однакові для всіх абітурієнтів та опубліковані на веб-сайті РДГУ. Вступні вимоги сформульовані таким чином, щоб випускники могли досягти запланованих результатів навчання. Існують програми, які дають можливість студентам отримати знання та навички, яких їм не вистачає для досягнення запланованих результатів навчання. Існують чіткі та прозорі правила, структури та відповідальні особи для визнання досягнень та навчальних досягнень інших (у тому числі іноземних РДГУ) для забезпечення академічної мобільності.

Вимога 5. Викладання та навчання. Академічна підтримка здобувачів освіти.

Форми та методи навчання є найбільш корисними для досягнення очікуваних цілей і запланованих результатів навчання, а також відповідають вимогам навчання, орієнтованого на студента. Підходи викладання та навчання передбачають обґрунтоване поєднання контактних годин (навчальних занять) та часу для самостійного навчання, а також проведення наукових досліджень. РДГУ надає поради, рекомендації та допоміжні

послуги (технічні та нетехнічні), щоб допомогти досягти запланованих результатів навчання та свідомо обирати освітні траєкторії.

Вимога 6. Оцінювання здобувачів освіти.

Стандарти оцінювання знань студентів чіткі та прості для розуміння, і можна зрозуміти, наскільки здобувачі вищої освіти можуть досягти запланованих результатів навчання (отримати відповідні знання, навички та можливості розвитку), і вони є загальнодоступними. Для перевірки результатів навчання за різними програмами використовуються різні методи та формати оцінювання (наприклад, усний і письмовий). Перед навчанням усі студенти ознайомлюються з вимогами навчальної частини оцінювання (попередній огляд, завдання тощо).

Вимога 7. Науково-педагогічні та педагогічні працівники.

Склад, наукова спрямованість та кваліфікація (академічна та професійна) наукового та адміністративного персоналу є достатніми для успішного виконання програми. Він достатньо великий, щоб виконувати всі викладацькі обов'язки, надавати допомогу та консультації студентам, а також виконувати всі адміністративні завдання протягом періоду атестації. Встановлено правила, що гарантують кваліфікацію запрошених викладачів (іноземних викладачів, викладачів з практичним досвідом тощо). РДГУ забезпечує умови для підвищення кваліфікації викладачів, безперервного навчання та доступу до всіх ресурсів, необхідних для підвищення кваліфікації.

Вимога 8. Матеріальне забезпечення.

Кількість та якість інфраструктури, обладнання, навчально-методологічного та науково-методичного забезпечення є достатніми для досягнення цілей та результатів навчання програми. Викладачі та студенти можуть отримувати найновіші публікації, особливо англійські, та рецензовані журнали. РДГУ забезпечує доступ до всіх необхідних освітніх ресурсів (наприклад, друкованих примірників, електронних видань, електронних журналів тощо).

Вимога 9. Фінансове забезпечення.

Бюджет плану має бути прозорим і достатнім для забезпечення якісного виконання плану. Джерело коштів для плану (державні закупівлі, кошти, які використовуються фізичними та юридичними особами на навчання абітурієнтів, спонсорські кошти тощо) має бути чітко визначено. В разі об'єднання різних джерел, вам слід вказати співвідношення. Визначити джерело інформації для розрахунку витрат на навчання на основі плану. РДГУ має чітко розуміти ціну (вартість) програми та очікувані переваги, які вона надасть здобувачам вищої освіти.

Вимога 10. Внутрішня система забезпечення якості.

РДГУ розробила внутрішню політику та процедури забезпечення якості, які є публічними. РДГУ використовує різні методи для досягнення цілей плану, виявлення слабких місць і можливостей для вдосконалення. Для кожного проекту процес забезпечення якості здійснюється на постійній основі із залученням усіх зацікавлених сторін (таких як студенти, викладачі, випускники та потенційні роботодавці) з метою подальшого вдосконалення та модернізації. Результати будуть повідомлені відповідним зацікавленим сторонам. РДГУ заохочує зворотний зв'язок і повідомляє про свої висновки всім учасникам процесу.

Вимога 11. Повнота нормативного забезпечення та публічна інформація.

Для визначення прав та обов'язків учасників навчально-виховного процесу розроблено норми та правила. Ці правила поширюються на всіх зацікавлених сторін. Опублікувати опис навчального плану та його компонентів, включаючи інформацію про цілі плану та заплановані результати навчання (освітні компоненти), інформаційне забезпечення (підручники, посібники, монографії, рецензовані журнали тощо), методи навчання та методи оцінювання рівень запланованих результатів навчання.

1.2.1. Постановка технічного завдання

Зважаючи на вимоги які вказані в попередньому пункті та додаткові побажання представника навчального відділу для спрощення та часткової автоматизації процесу проходження акредитації було поставлено наступне завдання: Розробити інформаційну систему (базу даних) для переліку викладачів вищого навчального закладу, інформації про їх вчений ступінь та наукову спеціалізацію та реалізувати прикладний інтерфейс програми для автоматизації процесу пошуку та фільтрації записів в базі даних.

Для виконання поставленої задачі потрібно:

1. Змодельовати структуру бази.
2. Розробити базу даних за допомогою середовища SQL Management Studio.
3. Заповнити розроблену базу актуальною інформацією.
4. Створити користувацький інтерфейс.
5. Протестувати роботу програми.

РОЗДІЛ II. РЕАЛІЗАЦІЯ ТЕХНІЧНОГО ЗАВДАННЯ

2.1. Опис структури Баз даних

Для розробки бази даних навчального відділу нам потрібні такі таблиці:

1. Викладачі – це наша основна таблиця в якій буде міститися загальна інформація про викладача.
2. Наукова галузь – в цій таблиці буде інформація про галузь роботи викладача та його спеціалізацію.
3. Посада – ким працює викладач в університеті та його вчений ступінь.
4. Кафедри – таблиця яка містить в собі назви всіх кафедр університету.
5. Факультети – таблиця яка містить в собі назви всіх факультетів університету.

Тепер опишемо структуру полів кожної таблиці:

Таблиця 1.

Структура таблиці Vikladachi

Поле	Тип	Коментар
vikl_id	int	Ідентифікатор (первинний ключ)
vikl_fullname	Nvarchar (40)	Повне ім'я викладача
vikl_rik	int	Рік народження викладача
vikl_spec	Nvarchar (70)	Спеціалізація викладача
vikl_posada_id	int	Вторинний ключ на таблицю із посадами
vikl_nauk_galuz_id	int	Вторинний ключ на таблицю наукової галузі
vikl_naykpedstagrik	int	Педагогічний стаж викладача
vikl_temadis	Nvarchar (100)	Тема дисертації
vikl_facult_id	int	Посилання через вторинний ключ на таблицю факультетів

vikl_kafedra_id	int	Посилання через вторинний ключ на таблицю кафедр
-----------------	-----	--

Таблиця 2.

Структура таблиці Nauk_galuz

Поле	Тип	Коментар
nauk_galuz_id	int	Ідентифікатор (первинний ключ)
nauk_galuz_name	Nvarchar (70)	Повна назва наукової галузі
nauk_galuz_shifr	int	Шифр наукової галузі

Таблиця 3.

Структура таблиці Posada

Поле	Тип	Коментар
posada_id	int	Ідентифікатор (первинний ключ)
posada_name	Nvarchar (70)	Повна назва посади
posada_stup	Nvarchar (50)	Вчений ступінь викладача

Таблиця 4.

Структура таблиці Facult

Поле	Тип	Коментар
facult_id	int	Ідентифікатор (первинний ключ)
facult_name	Nvarchar (100)	Назва факультету

Таблиця 5.

Структура таблиці Kafedr

Поле	Тип	Коментар
kafedr_id	int	Ідентифікатор (первинний ключ)
kafedr_name	Nvarchar (100)	Назва кафедри

2.2. Розробка Баз даних в середовищі SQL Management Studio

Згідно описаної структури таблиць перейдемо в середовище SQL Management Studio та створимо таблиці за допомогою запитів **Transact SQL**:

1. Створимо таблицю Vikladachi:

```
use n
go
Create Table Vikladachi
(
vikl_id int PRIMARY KEY NOT NULL,
vikl_fullname nvarchar(40) NOT NULL,
vikl_rik int,
vikl_spec Nvarchar (70),
vikl_posada_id int,
vikl_nauk_galuz_id int,
vikl_naykpedstagrik int,
vikl_temadis Nvarchar (100),
vikl_facult_id int,
vikl_kafedra_id int,
)
```

2. Створимо таблицю Nauk_galuz:

```
use n
go
Create Table Nauk_galuz
(
nauk_galuz_id int PRIMARY KEY NOT NULL,
nauk_galuz_name Nvarchar (70),
nauk_galuz_shifr int
)
```

3. Створимо таблицю Posada:

```
use n
go
Create Table Posada
```



```
(
posada_id int PRIMARY KEY NOT NULL,
posada_name Nvarchar (70),
posada_stup Nvarchar (50)
)
```

4. Створимо таблицю Facult:

```
use n
go
Create Table Facult
(
facult_id int PRIMARY KEY NOT NULL,
facult_name Nvarchar (100)
)
```

5. Створимо таблицю Kafedr:

```
use n
go
Create Table Kafedr
(
kafedr_id int PRIMARY KEY NOT NULL,
kafedr_name Nvarchar (100)
)
```

Після створення таблиць можна створити діаграму бази даних щоб оглянути всю базу.

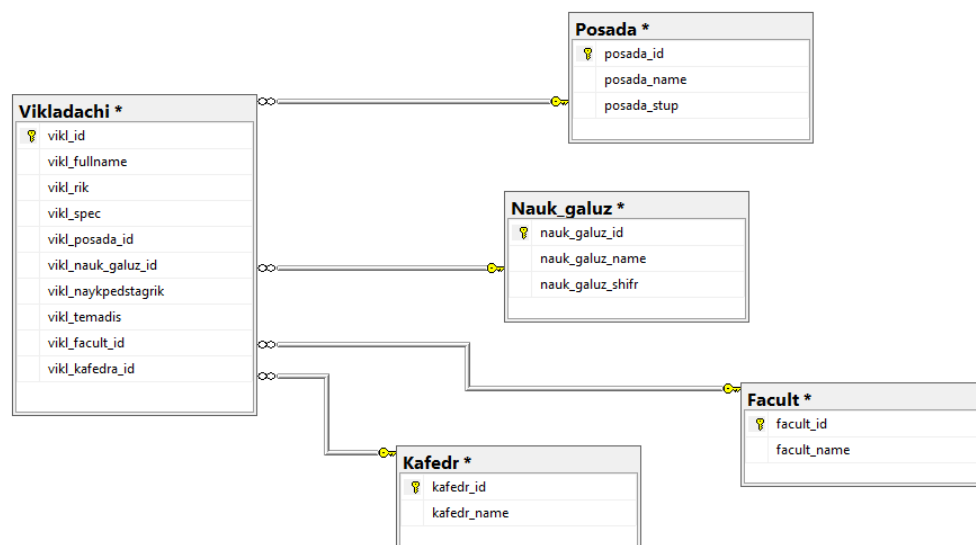
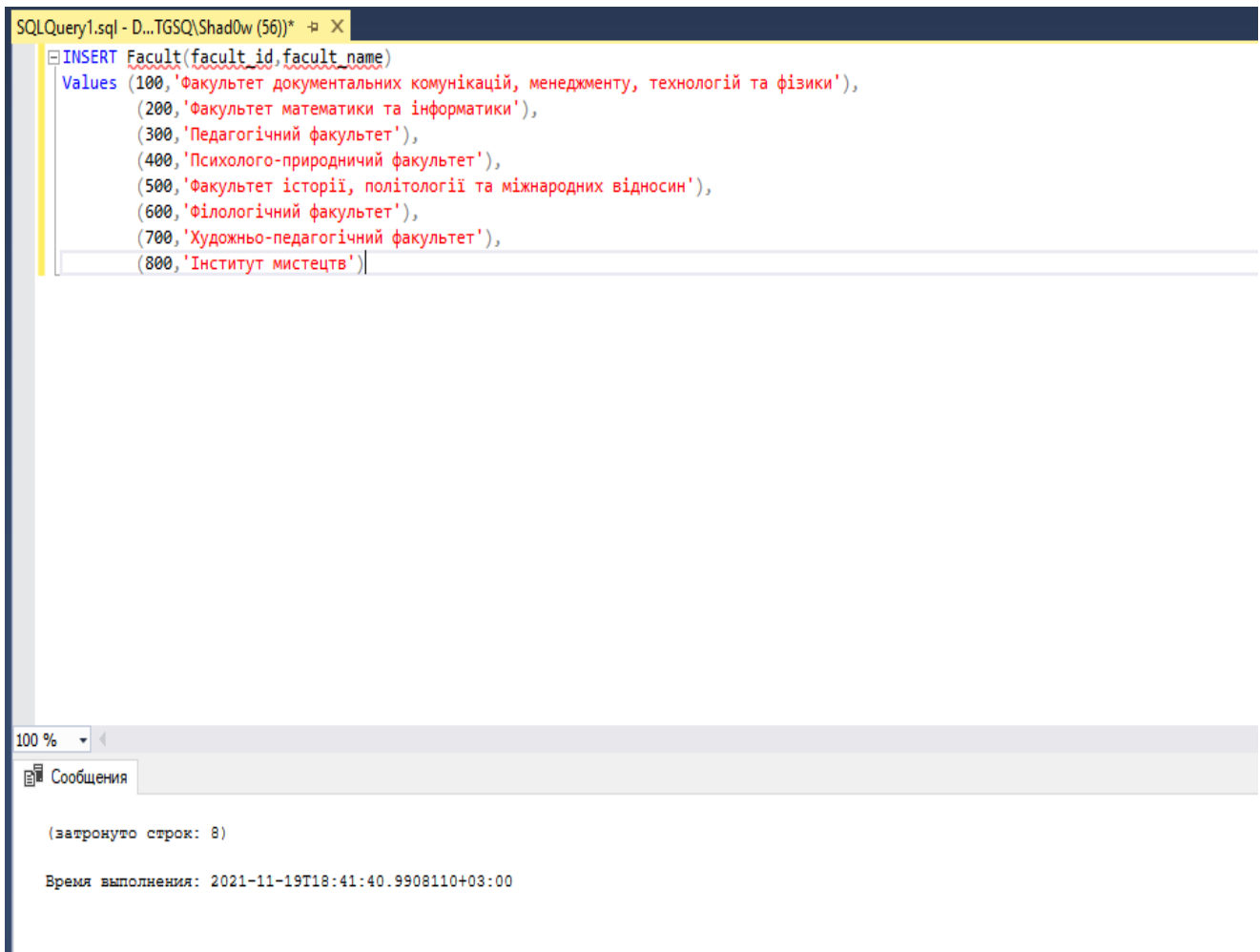


Рис. 2.1 (Схема бази даних)

Після того як таблиці і схема даних створені можна почати їх заповнення, спочатку заповним найголовніші елементи, почнемо з таблиці факультетів. Оскільки факультети є основними підрозділами які здійснюють підготовку студентів та аспірантів а також здійснюють керівництво науково-дослідною діяльністю кафедр, і викладачів яких вони об'єднують.



```
SQLQuery1.sql - D:\TGSQ\Shadow (56)*  X
INSERT Facult(facult_id,facult_name)
Values (100,'Факультет документальних комунікацій, менеджменту, технологій та фізики'),
(200,'Факультет математики та інформатики'),
(300,'Педагогічний факультет'),
(400,'Психолого-природничий факультет'),
(500,'Факультет історії, політології та міжнародних відносин'),
(600,'Філологічний факультет'),
(700,'Художньо-педагогічний факультет'),
(800,'Інститут мистецтв')
```

100 %

Сообщения

(затронута строк: 8)

Время выполнения: 2021-11-19T18:41:40.9908110+03:00

Рис. 2.2 (заповнення факультетів)

Далі заповнимо таблицю кафедр:

```

SQLQuery4.sql - D...TGSQ\Shad0w (51))* ✖ ✕
INSERT Kafedr(kafedr_id,kafedr_name)
Values (101,'Кафедра фізики, астрономії та методики викладання'),
(102,'Кафедра ЗТД, технологій та цивільної безпеки'),
(103,'Кафедра професійної освіти, трудового навчання та технологій'),
(104,'Кафедра документальних комунікацій та бібліотечної справи'),
(105,'Кафедра менеджменту'),
(106,'Кафедра економічної теорії'),
(107,'Кафедра економіки та управління бізнесом'),
(201,'Кафедра вищої математики'),
(202,'Кафедра математики з методикою викладання'),
(203,'Кафедра інформатики та прикладної математики'),
(204,'Кафедра інформаційно-комунікаційних технологій та методики викладання інформатики'),
(301,'Кафедра теорії та методик початкової освіти'),
(302,'Кафедра педагогіки початкової, інклюзивної та вищої освіти'),
(303,'Кафедра дошкільної педагогіки та психології та спеціальної освіти імені проф. Т.І. Поніманської'),
(304,'Кафедра теорії і методики виховання'),
(305,'Кафедра теорії і практики фізичної культури і спорту'),
(401,'Кафедра біології, здоров'я людини та фізичної терапії'),
(402,'Кафедра природних наук з методиками навчання'),
(403,'Кафедра екології, географії та туризму'),
(404,'Кафедра загальної психології та психодіагностики'),
(405,'Кафедра вікової і педагогічної психології'),
(406,'Кафедра практичної психології та психотерапії'),
(407,'Кафедра педагогіки, освітнього менеджменту та соціальної роботи'),
(501,'Кафедра всесвітньої історії'),
(502,'Кафедра історії України'),
(503,'Кафедра філософії'),
(504,'Кафедра політології та соціології'),
(505,'Кафедра політичних наук'),
(601,'Кафедра української мови імені проф. К.Ф. Шульжук'),
(602,'Кафедра стилістики та культури української мови'),
(603,'Кафедра української літератури'),
(604,'Кафедра теорії та історії світової літератури '),
(605,'Кафедра практики німецької і французької мови'),
(606,'Кафедра практики англійської мови'),
(607,'Кафедра романо-германської філології'),
(608,'Кафедра методики викладання іноземних мов'),
(609,'Кафедра іноземних мов'),
(701,'Кафедра театральної режисури'),
(702,'Кафедра культурології та музеєзнавства'),
(703,'Кафедра хореографії'),
(704,'Кафедра образотворчого та декоративно-прикладного мистецтва'),
(801,'Кафедра історії, теорії музики та методики музичного виховання'),
(802,'Кафедра гри на музичних інструментах'),
(803,'Кафедра пісенно-хорової практики та постановки голосу'),
(804,'Кафедра народних інструментів'),
(805,'Кафедра хорового диригування'),
(806,'Кафедра духових та ударних інструментів'),
(807,'Кафедра музичного фольклору'),
(808,'Кафедра естрадної музики')

```

Рис.2.3 (заповнення кафедр)

Після кафедр потрібно заповнити основну і найбільшу таблицю Викладачів, для заповнення таблиці використовуємо наступну Transact SQL

команду:

INSERT

```
Vikladachi(vikl_id,vikl_fullname,vikl_rik,vikl_spec,vikl_kval,vikl_posada,vikl_nauk_galuz,vikl_nauk_shifr,vikl_nauk_nazva,vikl_naykpedstagrik,vikl_v_chenezvannya_id,vikl_temadis,vikl_kafedra_id,vikl_facult_id)
```

Values

('2','Колупаєв Борис Сергійович','1937','математика і фізика','','професор, радник ректора','доктор хімічних наук','01,04,19','фізика і механіка полімерів','52','1111','Дослідження фізико-механічних і теплофізичних властивостей наповненого і пластифікованого полівінілхлориду, Енергообмінні процеси в металонаповнених гнучколанцюгових полімерах','Кафедра фізики, астрономії та методики викладання','Факультет документальних комунікацій, менеджменту, технологій та фізики')

Рис.2.4 (заповнення таблиці викладачів)

2.3. Розробка інтерфейсу та логіки програми за допомогою середовища Microsoft Visual Studio

Для розробки інтерфейсу програми використаємо технологію Windows Form, створимо новий макет проекту WF.

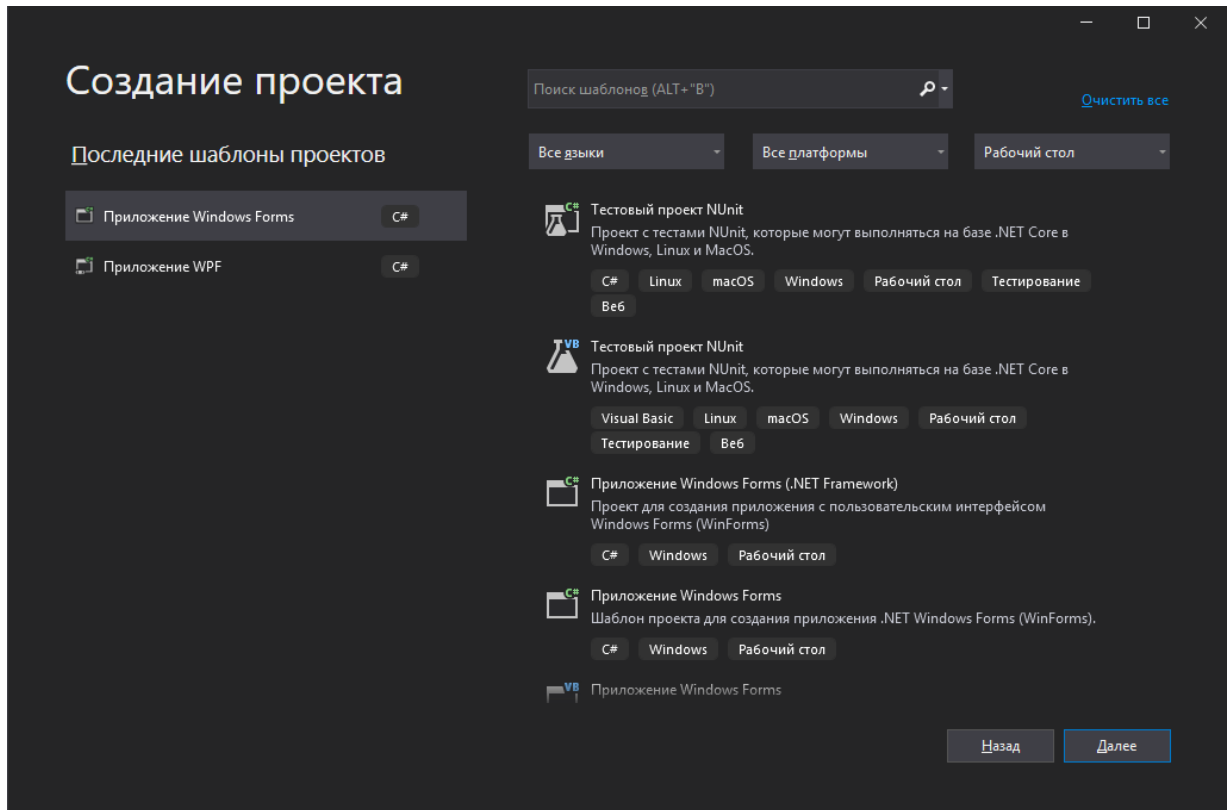


Рис.2.5 (Створення проекту)

Після завантаження проекту і стартової форми додамо на форму елемент DataGridView і перейдемо до написання коду за допомогою клавіші F7. Опишемо поля для з'єднання з базою даних

```

public partial class Form1 : Form
{
    private SqlConnection sqlConnection = null;
    private SqlCommandBuilder sqlBuilder = null;
    private SqlDataAdapter sqlDataAdapter = null;
    private DataSet dataSet = null;
    private bool newRowAdding = false;
    ссылка:1
    public Form1()
    {
        Program.f1 = this;
        InitializeComponent();
    }
}

```

Рис.2.6 (Створення sql полів)

Далі опишемо функцію для завантаження на сітку DataGridView інформації з нашої бази даних. Спочатку за допомогою SqlDataAdapter отримаємо запит на мові SQL з бази даних і зведемо його до команди з використанням SqlCommandBuilder, також ініціалізуєм елемент dataSet який буде приймати в себе цю інформацію і потім безпосередньо загрузити на DataGridView.

```

Ссылка: 1
private void LoadData() //Загрузка на датагрид
{
    try
    {
        this.WindowState = FormWindowState.Maximized; //фулл скрін
        sqlDataAdapter = new SqlDataAdapter("Select vikl_id, vikl_fullname, vikl_rik, vikl_spec, vikl_posada, vikl_nauk_galuz,
        sqlBuilder = new SqlCommandBuilder(sqlDataAdapter);

        sqlBuilder.GetInsertCommand();
        sqlBuilder.GetUpdateCommand();
        sqlBuilder.GetDeleteCommand();

        dataSet = new DataSet();
        sqlDataAdapter.Fill(dataSet, "Vikladachi");
        dataGridView1.DataSource = dataSet.Tables["Vikladachi"];

        for (int i = 0; i < dataGridView1.Rows.Count; i++)
        {
            DataGridViewLinkCell linkCell = new DataGridViewLinkCell();
            dataGridView1[12, i] = linkCell; //??
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
Ссылка: 2

```

Рис.2.7 (Загрузка на датагрид)

Створимо метод RELoadData() який буде відповідати за поновлення даних з бази коли нам це буде потрібно.

```

private void RELoadData() //Обновления данных на датагрид
{
    try
    {
        dataSet.Tables["Vikladachi"].Clear();
        sqlDataAdapter.Fill(dataSet, "Vikladachi");
        dataGridView1.DataSource = dataSet.Tables["Vikladachi"];

        for (int i = 0; i < dataGridView1.Rows.Count; i++)
        {
            DataGridViewLinkCell linkCell = new DataGridViewLinkCell();
            dataGridView1[12, i] = linkCell; //??
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
Ссылка: 1

```

Рис.2.8 (Метод для оновлення даних)

В подальшому потрібно описати подію завантаження форми та створення SQL з'єднання з базою даних по посиланню Data Source.

```

ссылка: 1
private void Form1_Load(object sender, EventArgs e)
{
    sqlConnection = new SqlConnection(@"Data Source=DESKTOP-8PNTG5Q;Initial Catalog=NavchViddil;Integrated Security=True;User=sa;Password=1qaz!WSX@");
    sqlConnection.Open();
    LoadData();
}

```

Рис.2.9 (SQL з'єднання)

Потім додамо на форму кнопку яка викликає метод для оновлення даних який був описаний раніше. Та опрацюємо подію кліка на елемент таблиці Data Grid.

```

ссылка: 1
private void toolStripButton1_Click(object sender, EventArgs e) // Кнопка оновлення таблиці
{
    ReloadData();
}

ссылка: 1
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (e.ColumnIndex == 12)
        {
            string task = dataGridView1.Rows[e.RowIndex].Cells[12].Value.ToString();
            if (task == "Видалити")
            {
                if (MessageBox.Show("Видалити цей запис?", "Видалення", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    int rowIndex = e.RowIndex;
                    dataGridView1.Rows.RemoveAt(rowIndex);
                    dataSet.Tables["Vikladachi"].Rows[rowIndex].Delete();
                    sqlDataAdapter.Update(dataSet, "Vikladachi");
                }
                newRowAdding = false;
            }
            else if (task == "Внести дані")//
            {
                int r = e.RowIndex;

                dataSet.Tables["Vikladachi"].Rows[r]["vikl_id"] = dataGridView1.Rows[r].Cells["vikl_id"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_fullname"] = dataGridView1.Rows[r].Cells["vikl_fullname"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_rik"] = dataGridView1.Rows[r].Cells["vikl_rik"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_spec"] = dataGridView1.Rows[r].Cells["vikl_spec"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_posada"] = dataGridView1.Rows[r].Cells["vikl_posada"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_nauk_galuz"] = dataGridView1.Rows[r].Cells["vikl_nauk_galuz"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_nauk_shifr"] = dataGridView1.Rows[r].Cells["vikl_nauk_shifr"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_nauk_nazva"] = dataGridView1.Rows[r].Cells["vikl_nauk_nazva"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_naykpedstagrik"] = dataGridView1.Rows[r].Cells["vikl_naykpedstagrik"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_temadis"] = dataGridView1.Rows[r].Cells["vikl_temadis"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_facult_id"] = dataGridView1.Rows[r].Cells["vikl_facult_id"].Value;
                dataSet.Tables["Vikladachi"].Rows[r]["vikl_kafedra_id"] = dataGridView1.Rows[r].Cells["vikl_kafedra_id"].Value;
            }
        }
    }
}

```

Рис.3 (Подія кліку на поле таблиці)

По алгоритму наступним кроком буде створення події зміни даних в певній клітині таблиці та обробка цієї події.

```

ссылка:1
private void dataGridView1_CellValueChanged(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        if (newRowAdding == false)
        {
            int rowIndex = dataGridView1.SelectedCells[0].RowIndex;
            DataGridViewRow editingRow = dataGridView1.Rows[rowIndex];

            DataGridViewLinkCell linkCell = new DataGridViewLinkCell();
            dataGridView1[12, rowIndex] = linkCell;
            editingRow.Cells["Видалити"].Value = "Внести дані";
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Рис.3.1 (Обробка події зміни значення в таблиці)

В подальшому потрібно реалізувати зручний пошук по базі даних за допомогою сітки Data grid, для цього додамо поля для пошуку тексту та напишемо відповідний метод для кожного поля.

```

//Швидкий пошук
ссылка:1
private void toolStripTextBox1_TextChanged(object sender, EventArgs e)
{
    (dataGridView1.DataSource as DataTable).DefaultView.RowFilter = $"vik1_fullname LIKE '{toolStripTextBox1.Text}%'";
}

ссылка:1
private void toolStripTextBox2_TextChanged(object sender, EventArgs e)
{
    (dataGridView1.DataSource as DataTable).DefaultView.RowFilter = $"vik1_spec LIKE '{toolStripTextBox2.Text}%'";
}

ссылка:1
private void toolStripTextBox3_TextChanged(object sender, EventArgs e)
{
    (dataGridView1.DataSource as DataTable).DefaultView.RowFilter = $"vik1_posada LIKE '{toolStripTextBox3.Text}%'";
}

ссылка:1
private void toolStripTextBox4_TextChanged(object sender, EventArgs e)
{
    (dataGridView1.DataSource as DataTable).DefaultView.RowFilter = $"vik1_nauk_galuz LIKE '{toolStripTextBox4.Text}%'";
}

ссылка:1

```

Рис.3.2 (Реалізація пошуку)

Фінальним етапом написання коду програми стане сортування інформації по факультету і кафедрі на екрані за допомогою ComboBox

```

ссылка:1
private void toolStripComboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (toolStripComboBox1.Text == "*Всі*")
    {
        dataGridView1.Refresh();
    }
    else
    {
        (dataGridView1.DataSource as DataTable).DefaultView.RowFilter = $"vikl_facult_id LIKE '{toolStripComboBox1.Text}%'";
    }
}

ссылка:1
private void toolStripComboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    if (toolStripComboBox2.Text == "*Всі*")
    {
        dataGridView1.Refresh();
    }
    else
    {
        (dataGridView1.DataSource as DataTable).DefaultView.RowFilter = $"vikl_kafedra_id LIKE '{toolStripComboBox2.Text}%'";
    }
}
}

```

Рис.3.3 (Сортування)

Тепер розмістим всі графічні елементи на формі дизайнера і отримаєм такий інтерфейс.

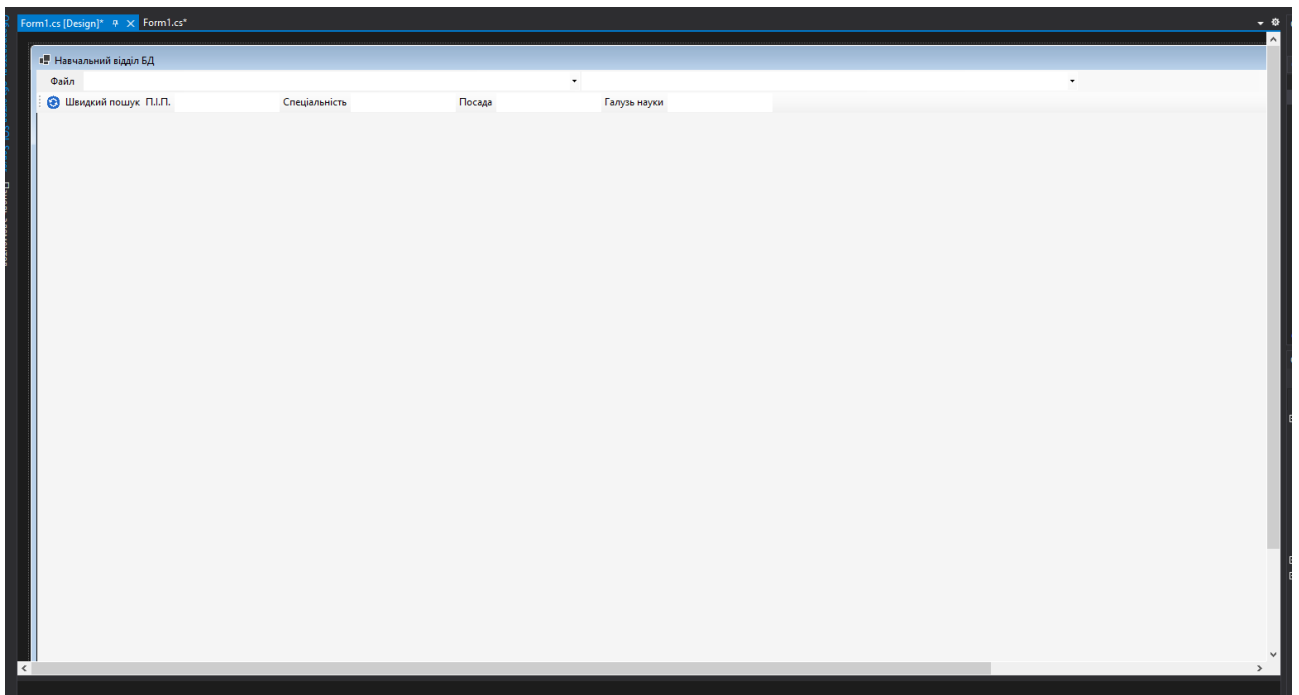


Рис.3.4 (Інтерфейс програми)

РОЗДІЛ III. ТЕСТУВАННЯ ПРОГРАМИ

3.1. Тестування програмного комплексу

Для завершення циклу розробки програми потрібно її протестувати, отже виконаємо перевірку функцій програми:

1. Зв'язок з базою даних та можливість змінити інформацію в таблиці.

Щоб перевірити цей пункт запустимо програму та спробуємо змінити якийсь запис і перевірити на реакцію бази даних. Наприклад змінимо рік народження

Колупаєва Бориса Сергійовича з 1937р. на 2000р.

№	Прізвище Ім'я Побатькові	Рік народження	Спеціальність	Посада
1	Колупаєв Борис Сергійович	1937	математика і фізика	професор, радник ректора
3	Мислінчук Володимир Олександрович	1977	фізика і математика	доцент
4	Максимцев Юрій Романович	1973	фізика та інформатика	доцент, зав. кафедри
5	Кривцов Валентин Валерійович	1977	фізика, хімія основи інформатики	доцент
6	Сідлецький Валентин Олександрович	1970	математика і фізика	доцент
7	Шевчук Тетяна Миколаївна	1981	фізика і хімія	доцент
8	Левчук Василь Васильович	1972	математика і фізика	доцент
9	Галатюк Юрій Михайлович	1962	математика і фізика	професор
10	Нечипорук Богдан Дмитрович	1954	математика і фізика	доцент
11	Віднічук Микола Антонович	1958	математика і фізика, облік і аудит, економіка підприємства	професор, директор ІПДО
12	Сяський Андрій Олексійович	1946	механіка	професор, довант по каф. ІПМ
13	Ястремський Сергій Вікторович	1964	механізація гідромеліорат. робіт	доцент
14	Шамсутдинов Баходир Фахридінович	1952	промислове і цивільне будівництво	доцент
15	Трофімчук Володимир Миколайович	1979	трудове навчання і фізика	доцент, зав. кафедри
16	Шевчук Олена Анатоліївна	1969	культурологія, практична психологія, вазеологія	доцент, довант. на каф.ПНМН на каф.БЗЛФТ
17	Глінчук Юлія Олександрівна	1978	математика	доцент
18	Тимощук Олександр Станіславович	1986	трудове навчання,БЖД та основи інформатики	доцент
19	Новак Сергій Миколайович	1959	машини і апарати легкої промисловості	доцент
20	Гавриш Володимир Степанович	1954	механізація гідромеліоративних робіт	доцент
21	Янцур Микола Сергійович	1952	загально технічні дисципліни	професор
22	Сингаївський Дмитро Васильович	1958	ЗДТ і праця	доцент
23	Герасименко Олександр Анатолійович	1976	Культурологія, економіка підприємства	доцент
24	Фешук Юрій Вікторович	1975	Трудове навчання і профорієнтація	доцент
25	Лісова Світлана Валеріївна	1946	ЗДТ і праця	зав.кафедри, професор
26	Симонович Наталія Василівна	1974	трудове навчання, креслення	доцент
27	Кирильчук Юрій Васильович	1959	ЗДТ і праця	доцент
28	Дупак Надія Василівна	1966	рос. мова та література	доцент
29	Шурин Олена Іванівна	1984	початкове навчання, біологія, практ. психолог.	доцент
30	Поліщук Наталія Володимирівна	1978	фізика і хімія, технології, швейна справа, обслуговуюча праця, креслення	доцент
31	Сілкова Галина Василівна	1953	бібліотеко-знавство та бібліографія	зав.кафедри, доцент
32	Бучковська Олена Юріївна	1961	бібліотеко-знавство та бібліографія	доцент

Рис.3.5 (Перевірка зміни запису)

Ми змінили рік в таблиці та натиснули кнопку “Внести зміни” в кінці рядку.

1	Прізвище Ім'я Побатькові	Рік народження	Спеціальність	Посада
2	Колупаєв Борис Сергійович	2000	математика і фізика	професор, радник ректора
3	Мислінчук Володимир Олександрович	1977	фізика і математика	доцент
4	Максимцев Юрій Романович	1973	фізика та інформатика	доцент, зав. кафедри
5	Кривцов Валентин Валерійович	1977	фізика, хімія основи інформатики	доцент
6	Сідлецький Валентин Олександрович	1970	математика і фізика	доцент
7	Шевчук Тетяна Миколаївна	1981	фізика і хімія	доцент
8	Левчук Василь Васильович	1972	математика і фізика	доцент
9	Галатюк Юрій Михайлович	1962	математика і фізика	професор
10	Нечипорук Богдан Дмитрович	1954	математика і фізика	доцент
11	Віднічук Микола Антонович	1958	математика і фізика, облік і аудит, економіка підприємства	професор, директор ІПДО
12	Сяський Андрій Олексійович	1946	механіка	професор, довант по каф. ІПМ
13	Ястремський Сергій Вікторович	1964	механізація гідромеліорат. робіт	доцент
14	Шамсутдинов Баходир Фахриддинович	1952	промислове і цивільне будівництво	доцент
15	Трофімчук Володимир Миколайович	1979	трудове навчання і фізика	доцент, зав. кафедри
16	Шевчук Олена Анатолівна	1969	культурологія, практична психологія, вазеологія	доцент, довант. на каф.ПНМН на каф.БЗЛФТ
17	Глінчук Юлія Олександрівна	1978	математика	доцент
18	Тимошук Олександр Станіславович	1986	трудове навчання,БЖД та основи інформатики	доцент
19	Новак Сергій Миколайович	1959	машини і апарати легкої промисловості	доцент
20	Гавриш Володимир Степанович	1954	механізація гідромеліораттвних робіт	доцент
21	Янцур Микола Сергійович	1952	загально технічні дисципліни	професор
22	Сингаївський Дмитро Васильович	1958	ЗДТ і праця	доцент
23	Герасименко Олександр Анатолійович	1976	Культурологія, економіка підприємства	доцент
24	Фещук Юрій Вікторович	1975	Трудове навчання і профорієнтація	доцент
25	Лісова Світлана Валеріївна	1946	ЗДТ і праця	зав.кафедри, професор
26	Симонович Наталія Василівна	1974	трудове навчання, креслення	доцент
27	Кирильчук Юрій Васильович	1959	ЗДТ і праця	доцент
28	Дупак Надія Василівна	1966	рос. мова та література	доцент
29	Шурин Олена Іванівна	1984	початкове навчання, біологія, практ. психолог.	доцент
30	Поліщук Наталія Володимирівна	1978	фізика і хімія, технології, швейна справа, обслуговуюча праця, креслення	доцент
31	Сілкова Галина Василівна	1953	бібліотеко-знавство та бібліографія	зав.кафедри, доцент

Рис.3.6 (Змінили рік в таблиці)

Тепер заходимо в SQL Management Studio і перевіряєм значення в базі даних. І бачимо що значення змінилось отже база даних приймає зміни які були застосовані в програмі

vikl_id	vikl_fullname	vikl_rlik	vikl_spec	vikl_kval	vikl_posada
1	Прізвище Ім'я Побатькові	Рік народження	Спеціальність		Посада
2	Колупаєв Борис Сергійович	2000	математика і фізика		професор, радник ректора
3	Мислінчук Володимир Олександрович	1977	фізика і математика		доцент
4	Максимцев Юрій Романович	1973	фізика та інформатика		доцент, зав. кафедри
5	Кривцов Валентин Валерійович	1977	фізика, хімія основи інформатики		доцент
6	Сідлецький Валентин Олександрович	1970	математика і фізика		доцент
7	Шевчук Тетяна Миколаївна	1981	фізика і хімія		доцент
8	Левчук Василь Васильович	1972	математика і фізика		доцент
9	Галатюк Юрій Михайлович	1962	математика і фізика		професор
10	Нечипорук Богдан Дмитрович	1954	математика і фізика		доцент
11	Віднічук Микола Антонович	1958	математика і фізика, облік і аудит, економіка пі...		професор, директор ІПДО
12	Сяський Андрій Олексійович	1946	механіка		професор, довант по каф. ІПМ
13	Ястремський Сергій Вікторович	1964	механізація гідромеліорат. робіт		доцент
14	Шамсутдинов Баходир Фахриддинович	1952	промислове і цивільне будівництво		доцент
15	Трофімчук Володимир Миколайович	1979	трудове навчання і фізика		доцент, зав. кафедри
16	Шевчук Олена Анатолівна	1969	культурологія, практична психологія, вазеологія		доцент, довант. на каф.ПНМН на каф.БЗЛФТ
17	Глінчук Юлія Олександрівна	1978	математика		доцент
18	Тимошук Олександр Станіславович	1986	трудове навчання,БЖД та основи інформатики		доцент

Рис.3.7 (Зміна в базі відбулася)

2. Перевірка функції швидкого пошуку.

Функція швидкого пошуку це інструмент який дозволяє знаючи певні характеристики особи швидко знайти її в базі даних. Фільтрування відбувається за 4 категоріями: Прізвище Ім'я По батькові(П. І. П), Спеціальність, Посада або Галузь науки.

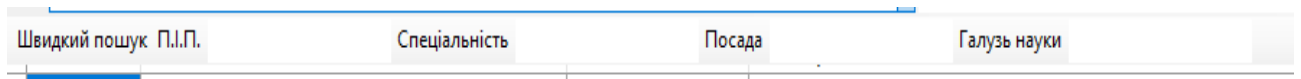


Рис.3.8 (Панель швидкого пошуку)

Отже протестуємо функціонал пошуку, допустим мені потрібно знайти викладача який спеціалізується на математиці. Як бачимо за допомогою пошуку в базі даних було знайдено цілих 12 записів.

Навчальний відділ БД				
Файл				
Швидкий пошук П.І.П.				
Спеціальність математика				
Посада професор				
Галузь науки				
2	Колупаєв Борис Сергійович	1937	математика і фізика	професор, радник ректора
3	Мисліччук Володимир Олександрович	1977	фізика і математика	доцент
6	Сідлецький Валентин Олександрович	1970	математика і фізика	доцент
8	Левчук Василь Васильович	1972	математика і фізика	доцент
9	Галатюк Юрій Михайлович	1962	математика і фізика	професор
10	Нечипорук Богдан Дмитрович	1954	математика і фізика	доцент
11	Віднічук Микола Антонович	1958	математика і фізика, облік і аудит, економіка підприємства	професор, директор ІПДО
17	Глінчук Юлія Олександрівна	1978	математика	доцент
69	Лукомська Оксана Іванівна	1979	математика та основи економіки	викладач
72	Юськів Богдан Миколайович	1956	прикладна математика, автоматизація проектування	професор, довант. по кафедрі політ. наук
77	Коханевич Тетяна Петрівна	1979	математика і фізика, фінанси і кредит	викладач
81	Свистун Тетяна Леонідівна	1979	математика та основи інформатики	викладач

Рис.3.9 (Пошук по значенням)

3. Функція сортування за факультетами та кафедрами

З використанням цієї функції можна спростити пошук записів по певній кафедрі або факультету. Сортування відбувається за допомогою двох Combobox елементів, перший відповідає за факультет а другий за кафедри на цьому факультеті. Застосуємо сортування, оберемо факультет документальних комунікацій, менеджменту, технологій та фізики та відсортуємо викладачів з кафедри менеджменту на цьому факультеті.

Навчальний відділ БД

Файл Факультет документальних комунікацій, менеджменту, технологій та фізики Кафедра менеджменту

Швидкий пошук П.П.П. Спеціальність Посада Галузь науки

№	П.П.П.	Спеціальність	Посада	Галузь науки
42	Микитин Тарас Миронович	1965	гідромеліорація	зав. кафедри, доцент
43	Максимчук Надія Станіславівна	1983	менеджмент організації	доцент
44	Самборський Ігор Олександрович	1962	політична економія	доцент, довант по каф. ТПФКС
45	Пелех Оксана Богданівна	1979	облік та аудит	професор
46	Берташ Борис Миколайович	1969	гідромеліорація	доцент, довант по каф. ЗЛФТ
47	Хижнякова Надія Олександрівна	1976	економіка підприємства	доцент
48	Сяська Олена Василівна	1981	менеджмент організації	доцент
49	Поліщук Олена Юрївна	1981	менеджмент організації	викладач
50	Машта Надія Олександрівна	1984	товар-во та експертиза в митній справі	доцент
51	Дайнега Олександр Вікторович	1962	економіка і організація водного господарства	професор, проректор з наукової роботи
52	Дейнега Інна Олександрівна	1976	економіка підприємства	професор
53	Юхименко-Назарук Ірина Анатолівна	1982	економіка підприємства	професор, декан ФДКМТФ
54	Шимко Ольга Володимирівна	1961	товар-во і організація торгівлі непрод. тов.	доцент
55	Савченко Ольга Ростиславівна	1978	менеджмент організації	доцент, довант. по кафедрі культура та МЗ
56	Мединська Тетяна Ігорівна	1981	економіка підприємства	доцент
57	Бобровський Анатолій Леонідович	1951	гідротехн. будівн.річк. споруд та ГЕС	професор
58	Кнолле-Нікшич Світлана Міланівна	1978	облік та аудит, психологія	доцент
59	Поляк Катерина Юрївна	1979	менеджмент у невиробн. сфері	доцент
60	Орлов Олександр Геннадійович	1975	менеджмент у невиробн. сфері	викладач
61	Трало Ірина Миколаївна	1980	менеджмент організації	старший викладач
62	Пашніна Анна Олександрівна	1988	економічна кібернетика	доцент, довант по кафедрі ІКТ та МВІ
63	Марценюк Віктор Васильович	1978	маркетинг	викладач
64	Щесюк Сергій Вікторович	1974	менеджмент у невиробничій сфері	старший викладач

Рис.4 ч.1 (Сортування за кафедрою)

В кінці записів вказується факультет і кафедра, отже ми можемо побачити що сортування працює.

Оптимізація параметрів водоприймачів осушувальних систем Західного Полісся України	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Економічні форми активізації людського фактору	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Міжнародна трудова міграція (порівняльний аналіз Чеської Республіки та України)	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Еколого-економічні засади використання осушених земель при створенні природно-заповідних територій	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Інвестиційне забезпечення реформування систем поводження з твердими побутовими відходами	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Економічне регулювання водокористування в комунальному господарстві	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Заробітна плата в умовах ринкової трансформації економіки України	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Вдосконалення асортименту і поліпшення якості плавлених сирних продуктів	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Інформаційна підтримка прийняття маркетингових рішень промисловим підприємством в умовах невизначеності	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Формування комплексу маркетингових комунікацій закладів вищої освіти: теорія, методологія, практика	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Розвиток теорії та методології бухгалтерського обліку в контексті неінституційної теорії	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Організація торгівлі текстильними товарами в умовах ринкової економіки	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Ризик-менеджмент на підприємствах хімічної промисловості	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
Обґрунтування параметрів дренажу з врахуванням глибокого розпушування в Західних районах України	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту
	Факультет документальних комунікацій, менеджменту, технологій та фізики	Кафедра менеджменту

Рис.4.1 ч.2 (Сортування за кафедрою)

ВИСНОВКИ

Отже, охарактеризувавши та дослідивши інструментарій створення програмного забезпечення та наведені вимоги щодо трудомісткості процесу акредитаційної діяльності РДГУ, для даної роботи було сформовано технічне завдання метою якого була часткова автоматизація та пришвидшення пошуку інформації в навчальному відділі, за допомогою використання бази даних для переліку викладачів вищого навчального закладу, інформації про їх вчений ступінь та наукову спеціалізацію.

В цій дипломній роботі я розробив базу даних на основі SQL сервера та прикладну програму яка є зручним інтерфесом для неї і проводить виконання таких операцій як: додавання, редагування, видалення, пошук та сортування даних за певними критеріями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сінчук А.М., Порохняк О.П. Автоматизація роботи процесу наукового відділу Рівненського державного гуманітарного університету. XIV Всеукраїнської науково-практичної конференції «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРОФЕСІЙНІЙ ДІЯЛЬНОСТІ», 1 листопада, 2021 рік. Рівне, Україна. С.60-61в
2. Амосов А. А. Вычислительные методы для инженеров / А. А. Амосов, Ю. А. Дубинский, Н. П. Копченлова. — М: Мир, 1998. — 644 с.
3. Бахвалов Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. — Москва: Бинном, 2001. — 363 с.
4. Белоусов А. И., Ткачев С. Б. Дискретная математика. — М.: МГТУ, 2006. — С. 460-587.
5. Боб Уорд, Инновации SQL Server 2019. Использование технологий больших данных и машинного обучения / пер. с англ. Желновой Н. Б. — М.: ДМК Пресс, 2020. — 408 с.
6. Боровков А. И. Компьютерный инжиниринг / А. И. Боровков. — СПб: Изд-во Политехн. ун-та, 2012. — 93 с.
7. Виейра Р. Программирование базы данных Microsoft SQL Server 2005 / Р. Виейра// — М. ООО «И.Д. Вильямс», 2007. — 832 с.
8. Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс = Database Systems: The Complete Book. — Вильямс, 2003. — 1088 с. — ISBN 5-8459-0384-X.
9. Жалдак М.І. Михалін Г.О.Елементи стохастики з комп'ютерною підтримкою. Посібник для вчителів. Видання 3-тє, доповнене — К.: «Шкільний світ». 2003. — 120 с.
- 10.Згуровський М.З. Інформаційні мережеві технології в науці та освіті, 5 липня 2002, Дзеркало тижня, № 25.

11. Информатика: Комп'ютерна техніка. Комп'ютерні технології. Посіб./ За ред. О.І. Пушкаря – К.: Видавничий центр "Академія", 2001. – 696 с.
12. Ісаєнко О. Типово-видова структура інформаційного обслуговування сучасної бібліотеки / О. Ісаєнко // Вісн. Кн. палати. – 2008. – №12. – С. 17-19.
13. Кайт Т. Oracle для професіоналов / Том Кайт// – СПб. : ООО "ДиаСофтЮП", 2003. – 672 с.
14. Калашян А.Н., Калянов Г.Н. Структурные модели бизнеса: DFD – технологии. – М., 2003. – 339 с.
15. Карпенко Д.І. Комп'ютерне моделювання процесу випромінювання звуку при русі вісесиметричних тіл в морському середовищі / Д.І. Карпенко // Сучасні проблеми наукового забезпечення енергетики, 2018. — 137 с.
16. Карплюк С. "Інформаційно-педагогічний менеджмент вищої школи: сучасний стан та перспективи розвитку", Науковий вісник Ужгородського університету. Серія: Педагогіка. Соціальна робота., №241, с. 122-125, 2017.
17. Коваль Т. Інноваційні процеси бібліотечно-інформаційного обслуговування читача наукової бібліотеки / Т. Коваль, Л. Туровська // Наук. пр. Нац. б-ки України ім. В.І. Вернадського. – Київ: НБУВ, 2007. – Вип. 19. – С. 99-107.
18. Коваль Т. Український читач на тлі суспільних перетворень / Т. Коваль // Вісн. Кн. палати. – 2003. – №2. – С. 28-34.
19. Коваль Т. Читач у системі обслуговування електронними інформаційними ресурсами наукової бібліотеки / Т. Коваль // Бібл. форум. – 2006. – №2. – С. 22-23.
20. Коломейчук Е.М. Структура обслуговування в універсальних наукових бібліотеках Росії / Е.М. Коломейчик // Наук.Техн. Бібл.. –Москва: НТБ, 2002. –№ 5. – С. 5-11.

- 21.Крюков В.В., Шахгельдян К.И. Корпоративная информационная среда ВУЗА: Методология, модели, решения: [монография] / В.В. Крюков, К.И. Шахгельдян. – Владивосток: Дальнаука, 2007. – 308 с.
- 22.Кузнецов С. Д. Основы баз данных. — 2-е изд. — М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2017. — 484 с. — ISBN 978-5-94774-736-2.
- 23.Левченко О.М. та ін. Основи створення комп'ютерних презентацій: [навчальний посібник] / О.М. Левченко, І.В. Коваль, І.О. Завадський. – К.: Вид. група ВНУ, 2009. – 368 с.
- 24.Лойцянский Л. Г. Механика жидкости и газа / Л. Г. Лойцянский. — М: Дрофа, 2003. — 840 с.
- 25.Нейгл К. С# 5,0 и платформа .NET 4.5 для профессионалов/ Кристиан Нейл. — М: Вильямс, 2013. — 1440 с.
- 26.Просиз Д. Программирование для Microsoft .NET / Джеф Просиз. — М: Русская редакция, 2003. — 704 с.
- 27.Скит Д. С# для профессионалов: тонкости программирования, 3-е издание, новый перевод С# in Depth / Джонни Скит. — М: Вильямс, 2014. — 608 с.
- 28.Циммерман Г. А. Проблема розробки та використання програмного забезпечення навчального призначення у складі сучасних освітніх технологій / Г. А. Циммерман, О. В. Циммерман / [Електронний ресурс] - Режим доступу: <http://virtkafedra.ucoz.ua/el> HYPERLINK "http://virtkafedra.ucoz.ua/el_gurnal/pages/wp7/konf4/Cimmerman.pdf" gurnal/pages/wp7/konf4/Cimmerman.pdf.
- 29.Почему SQL? MySQL: [Електроний ресурс]. – Режим доступа : http://citforum.ru/database/sql_kg/1-2.shtml
- 30.Середовище розробки Microsoft Visual Studio [Електроний ресурс]. – Режим доступу : https://informatics.in.ua/programming_csharp/part_01.php

31. Методичні-рекомендації-з-акредитації-освітніх-програм університету [Електронний ресурс]. - Режим доступу: <https://projects.lnu.edu.ua/quaere/wp-content/uploads/sites/6/2018/10/Методичні-рекомендації-з-акредитації-освітніх-програм.pdf>
32. МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАКАЗ [Електронний ресурс]. - Режим доступу: http://search.ligazakon.ua/1_doc2.nsf/link1/RE33851.html
33. Статус Рівненського державного гуманітарного університету [Електронний ресурс]. - Режим доступу: https://rshu.edu.ua/images/st_rshu_2017.pdf
34. Що таке Педагогічний Програмний Засіб [Електронний ресурс].- Режим доступу: <http://www.znanius.com/3875.html>

ДОДАТОК А.

Тези доповідей на тему «Автоматизація процесів управління роботою наукового відділу Рівненського державного гуманітарного університету»

Анотація. Інформаційні технології в сучасних освітніх вимірюваннях є важливою складовою запровадження автоматизованого підходу до роботи будь-якого виду діяльності, зокрема, у закладах вищої освіти.

Ключові слова: автоматизована система, науковий відділ ЗВО.

Sinchuk A., Porokhnyak O. Automation of work management processes of the scientific department of the rivne state humanities university

Abstract. Information technologies in modern educational measurements are an important component of the introduction of an automated approach to the work of any activity, in particular, in higher education institutions.

Key words: automated system, research department of a higher education institution

Одним з найперспективніших способів ефективного управління освітнім процесом є впровадження інформаційно-комунікативних технологій в процес менеджменту на базі міжнародних стандартів. Розвиток сфери інформаційних технологій дав поштовх до вдосконалення всіх сфер життєдіяльності суспільства [2, с. 8-12]. Досить актуальним є питання автоматизації навчального процесу, зокрема, автоматизації процесів управління у закладах вищої освіти (ЗВО).

Актуальність даної проблеми є дуже висока оскільки за допомогою автоматизації можна значно прискорити процес управління інформацією та знизити трудомісткість роботи працівників закладу вищої освіти і зберегти час для інших задач.

Програмна реалізація являється інформаційною системою, розробленою на основі бази даних MS SQL та програмного забезпечення розробленого в середовищі Visual Studio. Основною метою цієї програми є структурування інформації про навчальний відділ, а саме пошук, сортування та редагування даних про викладачів Рівненського державного гуманітарного університету по відповідним критеріям.

Створена, в ній, база даних (англ. database) – сукупність даних, організованих відповідно до концепції, яка описує характеристику відповідних даних і взаємозв'язки між їх елементами; ця сукупність

підтримує щонайменше одну з областей застосування. В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки.

Середовище SQL Server Management Studio – це основний, стандартний і повнофункціональний інструмент для роботи з Microsoft SQL Server [4, с. 55], розроблений компанією Microsoft, який призначений як для розробників, так і для адміністраторів SQL Server [1, с. 68-82]. Це безкоштовне графічне середовище. За допомогою SSMS можна розробляти бази даних, виконувати інструкції T-SQL [3, с. 24], а також адмініструвати Microsoft SQL Server.

Microsoft Visual Studio – інтегроване середовище розробки програмного забезпечення від фірми Microsoft. Це середовище дозволяє створювати різноманітні програмні продукти: консольні програми, програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх підтримуваних платформ – Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.