



## АНОТАЦІЯ

*Чупілко О.С.* Організація та аналіз даних в інформаційній системі управління освітніх закладів.

Дипломна робота на здобуття освітнього ступеня магістр за спеціальністю 122 «Комп'ютерні науки». – Університет митної справи та фінансів, Дніпро, 2024.

Обсяг дипломної роботи складає 82 сторінки, 26 рисунків, 16 джерел.

Дипломна робота присвячена дослідженню автоматизованих інформаційних систем для зберігання, вибірки, обробки різного плану інформації, пов'язаної з моніторингом та управлінням освітніми закладами.

У першому розділі проаналізовано наявні підходи до організації даних та їх аналізу в закладах освіти, тенденції розробки освітніх систем, сучасні моделі і технології проектування програмного забезпечення для реалізації автоматизованої обробки інформації.

У другому розділі описано програмні інструменти для ефективної розробки і впровадження автоматизованих інформаційних систем, можливості подальшої обробки і аналізу даних з використанням математичного і програмно-технологічного забезпечення.

В третьому розділі описано проектування і програмну реалізацію автоматизованої системи обліку про відвідування навчальних занять учнями закладу освіти та інформаційну систему обліку та аналізу результатів роботи викладачів.

Результати роботи апробовано і впроваджено.

Ключові слова: інформаційні системи, автоматизація, бази даних, проектування систем, моделювання даних, програмування.

Список публікацій здобувача:

1. Т. Чупілко, О. Чупілко, і М. Мормуль, «Проектування і програмна реалізація автоматизованої системи відвідуваності та аналізу даних в закладах освіти», ІТКІ, вип. 56, вип. 1, с. 35–43, Бер 2023.  
DOI: <https://doi.org/10.31649/1999-9941-2023-56-1>
2. Чупілко О.С. Інноваційні технології в автоматизованій системі моніторингу відвідуваності та аналізу інформації // Міжнародна науково-практична конференція «Економіко-правові та управлінсько-технологічні вимоги сьогодення: молодіжний погляд». УМСФ, Дніпро, 03 листопада 2023 р. С.310-312.
3. Чупілко О.С. Проблеми використання сервісів і технологій великих даних в задачах економіки і управління // Міжнародна науково-практична конференція «Економіко-правові та управлінсько-технологічні вимоги сьогодення: молодіжний погляд». УМСФ, Дніпро, 03 листопада 2023 р. С.308-310.

Результати роботи представлено у Звіті з наукової роботи кафедри комп'ютерних наук та інженерії програмного забезпечення за 2023 рік, виконаної в рамках Науково-дослідної роботи «Математичне і комп'ютерне моделювання даних, процесів і систем в умовах визначеності, ризику, невизначеності та багатокритеріальності» (N держреєстрації 0123U100900).

## ABSTRACT

Chupilko O.S. Organization and analysis of data in the management information system of educational institutions

Thesis for obtaining a master's degree in the specialty 122 "Computer Science". – University of Customs and Finance, Dnipro, 2024.

The volume of the thesis is 82 pages, 26 figures, 16 sources.

The thesis is devoted to the study of automated information systems for storage, selection, processing of various types of information related to monitoring and management of educational institutions.

The first chapter analyzes available approaches to data organization and their analysis in educational institutions, trends in the development of educational systems, modern models and software design technologies for the implementation of automated information processing.

The second chapter describes software tools for effective development and implementation of automated information systems, the possibility of further processing and analysis of data using mathematical and software-technological support.

The third section describes the design and software implementation of an automated accounting system for attending classes by students of an educational institution and an information system for accounting and analyzing the results of teachers' work.

The results of the work were tested and implemented.

Keywords: information systems, automation, databases, system design, data modeling, programming.

List of publications of the Master's degree holder:

1. T. Chupilko, O. Chupilko, and M. Mormul, "Design and software implementation of an automated system of attendance and data analysis in educational institutions", ITKI, vol. 56, issue 1, p. 35–43, May 2023.  
DOI: <https://doi.org/10.31649/1999-9941-2023-56-1>
2. Chupilko O.S. Innovative technologies in the automated system of attendance monitoring and information analysis // International scientific and practical conference "Economic-legal and management-technological requirements of today: youth perspective". UMSF, Dnipro, November 3, 2023. P.310-312.
3. Chupilko O.S. Problems of using services and technologies of big data in tasks of economy and management // International scientific and practical conference "Economic-legal and management-technological requirements of today: youth perspective". UMSF, Dnipro, November 3, 2023. P.308-310.

The results of the work are presented in the Report on the scientific work of the Department of Computer Sciences and Software Engineering for 2023, carried out within the framework of the Scientific Research Work "Mathematical and computer modeling of data, processes and systems under conditions of certainty, risk, uncertainty and multi-criteria" (N state registration 0123U100900).

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1 Інформаційні освітні системи: організація даних і технології проектування.....	10
1.1. Сучасні підходи до організації та аналізу даних в освітніх системах..	10
1.2. Моделі і технології проектування програмного забезпечення та аналізу даних .....	17
1.3. Висновки до першого розділу .....	25
РОЗДІЛ 2 Програмне та математичне забезпечення організації та обробки даних в інформаційній системі .....	26
2.1. Системи управління базами даних та їх роль в організації даних в інформаційних системах .....	26
2.2. Програмні інструменти та математичні засади моделювання даних в інформаційних системах .....	35
2.3. Висновки до другого розділу .....	51
РОЗДІЛ 3 Проектування і програмна реалізація інформаційної системи та аналізу даних в закладах освіти.....	52
3.1. Автоматизована системи відвідуваності та обробки інформації в закладах освіти .....	52
3.2. Автоматизована інформаційна система обліку та аналізу результатів роботи викладачів .....	61
3.3. Висновки до третього розділу .....	79
ВИСНОВКИ.....	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	81

## ВСТУП

Для інтелектуального розвитку суспільства велике значення має освіта, яка наділяє школярів та студентів знаннями та вміннями. Заклади освіти мають бути сучасними з огляду використання технологічних засобів та програмних інструментів для організації навчального процесу, автоматизації діловодства, збирання освітніх даних, аналізу інформації та прийняття відповідних обґрунтованих рішень. Сучасні інноваційні технології дозволяють автоматизувати багато процесів в закладі освіти та об'єднати управління в одну інформаційну систему. Це сприяє підвищенню ефективності моніторингу і аналізу даних, а також управління закладом в цілому.

В нашій країні немає тенденції у створенні та впровадженні відповідного програмного забезпечення, уніфікованого і стандартизованого для потреб освітніх закладів, будь то середні школи, коледжі, заклади вищої освіти. Кожний заклад за наявності можливостей та умов, має самостійно «спрощувати» управління та аналіз інформації завдяки власним розробленим інформаційним системам або окремим програмам, що дозволяють вирішувати деякі питання, і не є інтегрованими в загальну інформаційну автоматизовану систему управління закладом. Тому дослідження цієї проблеми в кваліфікаційній магістерській роботі є обґрунтованим, і тема є актуальною.

Об'єктом дослідження є автоматизовані інформаційні системи для зберігання, вибірки, обробки різного плану інформації, пов'язаної з моніторингом та управлінням освітніми закладами.

Предметом дослідження є сучасні методи проектування і комп'ютерні технології розробки і впровадження інформаційних систем, методи аналізу інформації, що зберігається в системах, пов'язаних з організацією навчального процесу в освітніх закладах.

Метою дипломної магістерської роботи є проектування, створення і впровадження двох автоматизованих інформаційних систем, які

використовують одні дані: автоматизована система відвідуваності та обробки інформації в закладах освіти та інформаційна система аналізу результатів роботи викладачів.

Для досягнення мети потрібно виконати наступні завдання:

1. Проаналізувати сучасні підходи до автоматизованої організації даних та їх аналізу в освітніх системах.
2. Розглянути сучасні моделі і технології проектування інформаційних систем.
3. Обрати ефективні програмні інструменти для розробки і впровадження інформаційних систем.
4. Розглянути можливості подальшої обробки і аналізу даних з використанням математичного і програмно-технологічного забезпечення
5. Спроекувати і програмно реалізувати автоматизовану систему обліку про відвідування навчальних занять учнями закладу освіти, що дозволяє автоматизувати збір інформації та її аналіз з метою отримання необхідних звітів, а також надсилання повідомлень батькам про прибуття учня в навчальний заклад.
6. Спроекувати і програмно реалізувати інформаційну систему обліку та аналізу результатів роботи викладачів, яка надає можливість вносити та зберігати необхідні дані про загальну інформацію про вчителя, його навчально-методичну роботу, підвищення кваліфікації, атестацію, участь в конференціях, МАН та інших заходах та її подальшого аналізу.

В роботі використовуються загальні теоретичні та емпіричні методи.

Для реалізації завдань кваліфікаційної роботи було обрано Microsoft Access. Microsoft Access має широкий спектр функцій та зв'язків з іншими додатками MS Office, і є найбільш розповсюдженим і офіційним пакетом для роботи з різними типами документів в закладах освіти. Для аналізу та



візуалізації інформації можуть бути використані інструменти програмного пакету MS Office, а також мови програмування і пакетів для роботи з даними, зокрема, Python і відповідні бібліотеки з великим функціоналом.

Наукова новизна дослідження полягає у застосуванні ефективних інструментів для розробки структури програмного продукту, обробки, моделювання даних.

Практичне значення дипломної роботи полягає у проектуванні, створенні та впровадженні в освітньому закладі автоматизації роботи з інформацією за допомогою інформаційної системи.

Автоматизовану систему відвідуваності та обробки інформації в закладах освіти було в основному розроблено і реалізовано під час науково-дослідницької практики у 2023 р..

Створення і впровадження інформаційної системи обліку і аналізу результатів роботи викладачів частково було виконано під час переддипломної практики у 2023 р.

Результати роботи апробовано і впроваджено в ТОВ «Комп'ютерна академія ІТ СТЕП Кам'янське» та Комунальному закладі "Ліцей №18" Кам'янської міської ради.

Апробація результатів роботи: видано статтю у співавторстві в фаховому журналі категорії В [7] та оприлюднено результати роботи на конференції [14]. Результати роботи представлено у Звіті з наукової роботи кафедри комп'ютерних наук та інженерії програмного забезпечення за 2023 рік, виконаної в рамках НДР «Математичне і комп'ютерне моделювання даних, процесів і систем в умовах визначеності, ризику, невизначеності та багатокритеріальності» (N держреєстрації 0123U100900).

Робота складається зі вступу, 3 розділів, висновків, містить 82 сторінки, 26 рисунків. Список використаних джерел включає 16 найменувань.

## РОЗДІЛ 1

# ІНФОРМАЦІЙНІ ОСВІТНІ СИСТЕМИ: ОРГАНІЗАЦІЯ ДАНИХ І ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ

### 1.1. Сучасні підходи до організації та аналізу даних в освітніх системах

Для інтелектуального розвитку суспільства велике значення має освіта, яка наділяє школярів та студентів знаннями та вміннями. Заклади освіти мають бути сучасними в плані не тільки проведення занять за допомогою інноваційних технологій, а й з огляду використання технологічних засобів та програмних інструментів у інформаційних системах для організації навчального процесу, збирання освітніх даних, аналізу інформації та прийняття відповідних обґрунтованих рішень. Сучасні інноваційні технології дозволяють автоматизувати багато процесів в закладі освіти та об'єднати управління в одну інформаційну систему. Це сприяє підвищенню ефективності моніторингу і аналізу даних, а також управління закладом в цілому.

Сучасні інформаційні системи управління освітою (Education Management Information System (EMIS)) сприяють забезпеченню рівного доступу до якісної освіти, справедливості у розподілі ресурсів, підвищення ефективності та якості освітніх послуг за допомогою ефективною системи моніторингу й аналізу даних. Функції EMIS включають збір, зберігання, інтеграцію, обробку освітніх даних та статистики із використанням сучасного програмно-апаратного забезпечення [1]. Однак, така управлінська інформаційна система, яка забезпечує суспільство актуальними та достовірними даними про стан розвитку освіти, функціонує в Україні, але не є поширеною і орієнтованою безпосередньо на заклади освіти. Можна зазначити, що в нашій країні немає тенденції у створенні та впровадженні

відповідного програмного забезпечення, уніфікованого і стандартизованого для потреб освітніх закладів, будь то, середні школи, коледжі, заклади вищої освіти. Кожний заклад за наявності можливостей та умов, має самостійно «спрощувати» управління та аналіз інформації завдяки власним розробленим інформаційним системам або, швидше за все, окремим програмам, що дозволяють вирішувати окремі питання, і не є інтегрованими в загальну систему управління закладом. Звісно, розробка, впровадження, підтримка, супроводження таких систем дорого коштує, потребує фахівців-програмістів, і державний освітній заклад не має можливості створити велику інтегровану систему, допрацьовувати повсякчас відповідно до змін в законодавстві, підтримувати безперебійну роботу. За наявності інноваційних інформаційних технологій і запитів сучасності розробка відповідного програмного забезпечення має бути в пріоритеті Міністерства освіти і науки, а також Міністерства цифрової трансформації.

Є іноземні приклади концепцій розвитку інформаційно-комунікаційних технологій в процесах управління освітою, інформатизації роботи закладів освіти. Поступово створюється єдине інформаційне середовище для використання всіма учасниками освітнього процесу, включаючи викладачів, школярів або студентів та їх батьків. Автоматизація управління має спростити проведення рутинних адміністративних операцій. Один з напрямів інформатизації – впровадження єдиної системи електронного документообігу, впровадження хмарних систем, що супроводжуються заходами, що забезпечують захист інформації.

Прикладом є система державна освітня інформаційно-аналітична система ŠVIS/EMIS Литовської Республіки, сучасна за своєю архітектурою, програмно-апаратним забезпеченням, особливостями підтримки, опорою на деперсоніфіковані індивідуальні дані здобувачів освіти, гнучкістю системи щодо її конфігурування, зміни форм збору статистичних даних, можливості дослідження кар'єри випускників закладів вищої освіти, вивчення зв'язку

ринку праці із результатами освітньої діяльності закладів освіти, насамперед закладів вищої і професійно-технічної освіти. [1, с. 22]. Система створювалась і продовжує розвиватись за рахунок донорів. Програмне забезпечення литовської ŠVIS/EMIS функціонує на ПЗ IBM Cognos Analytics з вартістю ліцензії близько 400 тис. євро. Фінансування (близько шести млн євро) здійснюється Фондом Європейської структури.

Аналогічні системи розвиваються в Таджикистані, країнах африканського континенту.

В Україні В межах ініціативи Міністерства освіти і науки України з відновлення освітнього процесу «Школа, ми разом!» ТОВ «ЕДДІ Україна» та ДНУ «Інститут освітньої аналітики» домовилися про під'єднання першої освітньої інформаційної системи (ОІС) до провідного Автоматизованого інформаційного комплексу освітнього менеджменту (АІКОМ) [2].

Система дозволяє дані е-діловодства подавати через нашу освітню інформаційну систему, а також оптимізувати збір даних, уникаючи повторного їхнього подання в інших інформаційно-комунікаційних системах і на паперових формах.

Програмно-апаратний комплекс «Автоматизований інформаційний комплекс освітнього менеджменту» (АІКОМ) – електронна система управління освітою зі збору, зберігання, управління, використання даних у сфері освіти, що дає змогу приймати обґрунтовані та ефективні управлінські рішення. Дана система генерує статистику в межах системи освіти, забезпечує її моніторинг завдяки розширеним функціональним можливостям [3]. Одним важливим елементом АІКОМ є запровадження системи е-журналів та е-щоденників в закладах загальної середньої освіти, без необхідності їх дублювання в паперовій формі.

АІКОМ розроблено в рамках гранту Світового банку для проекту «Зміцнення науково-обґрунтованої розробки державної політики на основі освітньої статистики та аналітики» і впроваджується Державною науковою

установою «Інститут освітньої аналітики» (ДНУ «ІОА») за підтримки Міністерства освіти і науки України [3]. Цілями проекту є: збір достовірних, актуальних даних про стан системи освіти за затвердженою системою показників та автоматичною генерацією звітності; забезпечення органів управління і громадськості статистичними даними для прийняття ефективних управлінських рішень щодо підвищення доступності та якості освіти, ефективності використання державних коштів; реалізація можливості взаємодії АІКОМ з різними освітніми системами, базами даних та державними реєстрами; розвантаження вчителів від необхідності заповнення різних форм звітності та паперової роботи; запровадження обліку дітей дошкільного, шкільного віку та закладів дошкільної освіти; запровадження процесів зарахування, відрахування та переведення учнів до закладів загальної середньої освіти на базі АІКОМ; створення електронного сервісу запису до закладу позашкільної освіти; забезпечення можливості підвищення кваліфікації педагогічних працівників через електронну систему на базі АІКОМ; запровадження електронної системи управління професійною (професійно-технічною) освітою [3].

Тож, в Україні інформаційно-аналітична система розглядається у плані зв'язку закладів освіти з Міністерством освіти та науки, з вимірюванням і періодичним моніторингом освітніх індикаторів та формуванням різноманітної звітності. У повсякденному використанні функції автоматизації обмежуються електронними щоденником та журналом. Нерідко ці програми допрацьовуються, позаяк користувачі (вчителі) зауважують, що інтерфейс є недостатньо зручним.

Останнім часом впроваджено інформаційну систему щодо перепідготовки кадрів в сфері освіти. Окремо можна зазначити розробку структури інтегрованого інформаційного середовища, що дає можливість здійснювати моніторинг підготовки наукових та науково-педагогічних кадрів у закладах вищої освіти та наукових установах. Ця система створена як

сукупність декількох розподілених баз даних, кожна з яких містить інформацію про обсяги державного замовлення аспірантури та докторантури, формування плану прийому, ведення особових справ здобувачів, мережу спеціалізованих вчених рад.

Загалом, українську інформаційну систему надання освітніх послуг розробляють на замовлення Міністерства освіти і науки окремі фірми за договорами, і пов'язують ці системи з питаннями формування баз даних для статистичної звітності, індикаторами для розрахунку результативної звітності в регіональному розрізі, коефіцієнтами порівняння досягнень за певними показниками освітніх закладів у міській та сільській місцевостях, показниками ефективності використання бюджетних коштів закладами освіти, показниками забезпеченості педагогічними працівниками, що викладають різні предмети і т. ін. На основі автоматизованих рішень розробляють і реалізують концептуальні засади комплексного оцінювання стану освіти. Використовують також математично-методологічне забезпечення для інформаційно-аналітичної системи ефективності освіти.

Відбувається формування інформаційної системи управління освітою у сфері професійно-технічної освіти. Зокрема, орієнтовані системи на створення інструментарію для оцінювання діяльності закладів освіти на основі визначених показників, зокрема, підготовки робітничих кадрів та технічних службовців відповідно до потреб ринку за категоріями, галузями підготовки та аналіз діяльності закладів по регіонах за інтегральними рейтинговими показниками.

В системі вищої освіти становлення інформаційної системи пов'язане з урахуванням процесів у вищій освіті та впливів зовнішнього середовища. Міністерство освіти і науки невпинно проводять реформування системи вищої освіти. На порядку дня стоїть розробка інформаційної системи, що враховує стан і тенденції розвитку. На сьогодні така система відсутня. Є окремі модулі, що забезпечують управління в освіті, але немає уніфікованих

форм обміну та сумісності інформації на різних рівнях, відсутня оптимізація структури системи інформаційного забезпечення, своєчасність отримання інформації, захист інформації та її вірогідність.

Щодо підготовки здобувачів вищої освіти комплекс складається з підсистем, що пов'язані з конкурсними пропозиціями прийому на рівні балакаврів та магістрів різних спрямувань, молодшого спеціалісту, обсягів державного замовлення. У зв'язку з наближенням стандартів вищої освіти до міжнародних компоненти системи зазнають щорічно змін і корегуються.

Тож, проблема створення і впровадження інформаційних освітніх систем стоїть перед державою на рівні Міністерства освіти і науки, не говорячи про окремі заклади освіти, де зберігається рутинна робота в ручному режимі. При цьому в державі є певні інституції, що мали б розробити такі системи і впровадити їх на державному рівні. Фахівцями ДНУ « Інституту освітньої аналітики» було зроблено деякі кроки у напрямку подальшого розвитку інформатизації освіти, але , на жаль, все залишається лише в планах і зведено до формулювання і визначення модулів, що мають бути у такій системі.

Задача, що стоїть у цій роботі є надзвичайно важливою, об'ємною і досить складною. Забезпечення комплексної автоматизації передбачає створення єдиного інформаційного простору державного управління освітою, в якому суб'єкти зможуть взаємодіяти, керуючись єдиними правилами надання і обробки інформації як у паперовому, так і електронному вигляді [1].

При розбудові аналітичної інформаційної системи, навіть в межах одного закладу, мають бути враховані наступні зауваження: інформація має інтенсивно і швидко потрапляти для обробки і донесення до споживача; інформація має бути уніфікованою, єдиною, структуризованою і оптимізованою, сумісною і відповідною на певних етапах формування і обробки; інформація має бути відкритою і доступною.

На першому етапі має бути розроблено загальне уявлення про інформаційну систему: які цілі має задовольняти інформаційна система та які умови її функціонування; яка вхідна інформація має використовуватися; які документи, форми, звіти потрібні для статистичної обробки та звітності; яку аналітику має виконувати інформаційна система і яка візуалізація даних потрібна; як забезпечити захист даних. Ці положення має бути зафіксовано у технічному завданні.

На другому етапі обираються програмні інструменти для реалізації системи; створюється структура; описуються модулі для організації і аналізу даних.

На третьому етапі відбувається налагодження системи, впровадження і тестування.

Важливо, щоб інформаційна система була такою, що дозволяє доповнення і допрацювання, а також внесення змін за потреби, тобто підтримка в актуальному стані.

В інформатизації закладів освіти можуть виникати певні проблеми, окрім відсутності програмістів–розробників або коштів на придбання готового програмного забезпечення. До них можна віднести відсутність єдиного підходу та цілісності у розробці та впровадженні програмного забезпечення, відсутність цілісного рішення про автоматизацію управлінської діяльності та документообігу на всіх рівнях; нестача електронних засобів навчання для більшості навчальних предметів і навіть інформатики; відсутність у багатьох установах і закладах освіти локальних обчислювальних та телекомунікаційних мереж, хмарних сервісів, що поєднують наявні технічні ресурси; відсутність інтеграції наявних програм, і освітніх, і спеціалізованого програмного забезпечення; недостатня кількість автоматизованих робочих місць для фахівців різних напрямів, як то керівників, викладачів і т.ін.; недостатній рівень підготовки викладачів, керівників закладів та інших спеціалістів для використання сучасних комп'ютерних технологій і т.ін.



В цій роботі запроваджено створення інформаційної системи для внутрішньої організації, обробки даних і результатів для потреб управління в самому закладі освіти, створенню власної звітності, аналізу і візуалізації успішності і т. ін. Створено також окремі можливості для ведення обліку відвідуваності учнів і аналізу, зокрема, цієї інформації для деяких задач управління. Роботу було виконано на замовлення закладу освіти. Інформаційна система пройшла тестування і є впровадженою в навчальний процес освітнього закладу (ліцею 18 в м. Кам'янське та комп'ютерної академії «СТЕП Кам'янське»).

## **1.2. Моделі і технології проектування програмного забезпечення та аналізу даних**

На сьогоднішній день не існує універсального процесу розробки програмного продукту. Кожен поточний процес розробки, який здійснюється в межах певного проекту, має велику кількість особливостей і індивідуальностей. Існує безліч аспектів у проектуванні, які мають бути узгоджені, але найголовнішими залишаються продумана модель і засоби реалізації програмного забезпечення.

Якщо 10 років тому повний життєвий цикл продукту в середньому становив 8—12 років, то в наш час — лише 2—4 роки завдяки значній швидкості розвитку і появи нових сучасних технологій. Однак розроблення програмного продукту потребує тривалого терміну і великих витрат. Отже, при проектуванні інформаційна структура повинна забезпечити оптимальну розробку з урахуванням матеріальних, часових, інших витрат та морального старіння. Крім того, впродовж життєвого циклу продукту потрібним є сервісне обслуговування, операції щодо вдосконалення його та інше.

Зазвичай, розробка інформаційної системи передбачає послідовні стандартні фази. Перша фаза – обговорення і визначення цілей інформаційної

системи для автоматизації, користувачів, очікуваних результатів, джерел, необхідних для створення інформаційної системи та ін. Ця стратегія часто фіксується у вигляді технічного завдання. Друга фаза – створення ІС та впровадження її. Третя фаза – розробник здійснює супроводження системи при переході системи після впровадження у повне розпорядження організації-користувача. У процесі супроводу розробник усуває помилки, виявлені після впровадження, здійснює адаптацію ІС за необхідністю, доопрацьовує її з метою підвищення якості функціонування. Супровід уповільнює моральне старіння програмного забезпечення інформаційної системи.

Але при роботі з закладами освіти виникає ряд труднощів, пов'язаних з тим, що вимоги не можуть бути сформульованими досить чітко і визначено одразу замовниками (адміністрацією), оскільки вони не завжди добре розуміють можливостей автоматизації процесів в принципі, що впливає на затвердження технічного завдання для розробки системи. В процесі роботи іде суттєве допрацювання вимог на окремих стадіях роботи. І окремо потрібно зауважити на відсутність коштів для залучення у процес розробки фахівців-програмістів. Тому створення інформаційної системи проводиться на добровільних засадах фахівцями, які вміють створювати програмні продукти. Окрім того, розробник має бути знайомим з процесом організації управління освітнім закладом.

### Моделі розробки програмного забезпечення

Існуючі моделі життєвого циклу різняться структурою і конкретним змістом фаз створення і впровадження автоматизованої системи (АС) або окремих її складових. Найпоширенішими моделями ЖЦП і методами є [ 4 ]:

- o каскадна модель (модель водоспаду);
- o спіральна модель;
- o метод швидкого прототипу;

- о метод послідовного нарощування функцій;
- о модель, заснована на повторному використанні компонентів;
- о модель, заснована на автоматизованому синтезі програм.

Каскадна модель характеризується чіткою впорядкованістю стадій створення і впровадження автоматизованої системи: визначення вимог, розробка технічного завдання, проектування, реалізація, супровід. Така впорядкованість передбачає виконання всіх стадій без їх перегляду і повернення до попередніх. Доопрацювання можливе тільки при поверненні на першу стадію після опрацювання всіх стадій проекту, визначених спочатку. Недоліком є саме неможливість додавання змін на стадії розробки.

Тому в роботі використано спіральну модель розробки інформаційної системи. Спіральна модель передбачає проходження одних і тих самих стадій розробки, поки створена система не буде задовольняти вимогам. На кожному етапі є можливість допрацювання і доповнення програмного забезпечення, що є і перевагою, і недоліком одночасно, позаяк існує деяка невизначеність у термінах готовності та реалізації проекту, а також у його розростанні та складності, що змінюється у процесі роботи.

Метод швидкого прототипу навряд чи можливо використовувати у проектуванні автоматизованої інформаційної системи для певного закладу освіти, оскільки програмне забезпечення у своїй структурі має специфічні особливості, на відміну від програмних продуктів, розроблених, наприклад, для бізнесу.

Метод послідовного нарощування функцій полягає у поетапному проектуванні та реалізації інформаційної автоматизованої системи. На кожному етапі пропонується для експлуатації варіант системи з більшим функціональним наповненням. Це дозволяє пришвидшити впровадження та експлуатацію програмного продукту, особливо, якщо розробкою проекту займається один-два розробники, а не великий колектив. Цей метод також використовувався у цій роботі.

Модель, заснована на повторному використанні компонентів, дає змогу суттєво скоротити час на розробку окремих модулів, у тому випадку, якщо частину задач сформулювати в термінах підзадач, де можна використати стандартні підпрограми, наприклад бібліотеки програм. Тоді процес зводиться до написання програми, що викликає ці підпрограми в потрібній послідовності, з указанням фактичних параметрів та організує між ними обмін даними. Такий підхід набуває все більшого значення, і його було використано в цій роботі для моделювання за допомогою бібліотек Python.

Автоматизований синтез програм заснований на представленні знань як про предметну область, так і про процес створення програмних засобів. На відміну від підходів, розглянутих вище, реалізація цього підходу потребує досить високих первинних витрат на побудову моделей знань і особливо на створення інструментальних засобів для підтримки їх, що пов'язано з ризиком значного подорожчання розробки. Автоматизований синтез програм за їх специфікаціями дає змогу різко скоротити всі види витрат і реалізувати високу якість програмного продукту [ 4 ]. Такий підхід не використовувався в цій роботі, оскільки вимагає додаткового часу розробки і лежить поза межами теми.

Сучасні технології розробки і проектування програмного забезпечення включають в себе різноманітні методи, інструменти та підходи, щоб забезпечити ефективний інноваційний процес створення програм.

Розглянемо деякі з найбільш важливих та актуальних технологій в сфері розробки програмних продуктів, деякі з них використано при створенні інформаційної системи.

Хмарні технології (cloud technologies) представляють собою використання розподіленого резервування ресурсів, обчислення, зберігання даних та послуг через Інтернет. Це надає користувачам доступ до обчислювальної потужності, а також різноманітних сервісів і функціоналу, без необхідності власної інфраструктури або локальних серверів. Основні аспекти

хмарних технологій включають надання віртуальної інфраструктури (сервери, мережеві ресурси, сховища) через Інтернет. Користувачі отримують доступ до ресурсів для створення власних середовищ, забезпечення масштабованості і виконання різних завдань. Як приклади, можна навести Microsoft Azure, Google Cloud Platform (GCP). Технологія надає платформи для розробки, випробування та розгортання додатків без необхідності управління інфраструктурою. Користувачі можуть концентруватися на розробці коду, а платформа відповідає за надання середовища виконання. Надання доступу до готових до використання програм та служб через Інтернет дає можливість використовувати програми, не встановлюючи їх локально, і працювати з даними через хмарний сервіс. Як приклад, Microsoft 365 (Office 365). Упаковка та розгортання програм та їх залежностей в контейнери дозволяє створювати консистентні середовища виконання для різних етапів розробки та виробництва. Хмарні ресурси можна використовувати для обробки та аналізу великих обсягів даних, що дозволяє реалізовувати складні аналітичні завдання. Хмарні технології забезпечують безпеку обчислювальних ресурсів та даних, що зберігаються у хмарі. В цій роботі використовуються хмарні технології в частині системи, що описано в п. 3.1.

Agile-технології є підходом до розробки програмного забезпечення, який спрямований на гнучкість, швидкість та взаємодію в команді для досягнення кращих результатів. Основні принципи та техніки Agile включають наступні аспекти розробки програмного забезпечення. Застосовується ітераційний та інкрементальний підхід, при якому розробка відбувається через короткі ітерації, які називаються спринтами. На кінець кожного спринта додається приріст функціональності продукту. Відзначимо акцент на взаємодії та комунікації розробників з зацікавленими сторонами, при якій відбуваються короткі зустрічі, спринт-планування та ретроспективи. Визначаються завдання, які можна виконати протягом короткого періоду (спринта). Відбувається постійне вдосконалення проекту і процесів, що є ключовим

елементом Agile. Важливими є визнання важливості задоволення потреб клієнта та змін вимог у будь-який момент розробки, а також відповідальність за виконання завдань та прийняття рішень. Тестування вбудоване в процес розробки, іноді відоме як тестування, орієнтоване на розробку. Agile використовується для покращення реакції на зміни вимог, швидкого впровадження нового функціоналу та створення продуктів, які краще відповідають потребам користувачів. Багато методологій розробки, таких як Scrum, Kanban і Extreme Programming (XP), ґрунтуються на принципах Agile. Приклади використання технології Agile можна знайти в наукових публікаціях, наприклад, в [ 5 ]. В цій роботі використовувалася саме така технологія для розробки системи, описаної в п.3.1 та 3.2.

Технології CI/CD відносяться до сучасних практик та інструментів Continuous Integration (CI) та Continuous Delivery (CD), які використовуються в розробці програмного забезпечення для автоматизації процесів розгортання та тестування в основному в колективах розробників. Основні поняття CI/CD включають Continuous Integration (CI) та Continuous Delivery (CD). У частині Continuous Integration відбувається: регулярне об'єднання коду з репозиторію розробки; запуск автоматизованих тестів для перевірки, чи не порушено функціональність вже існуючого коду після змін; завантаження зібраного коду на тестовий сервер для подальшого тестування та регулярне об'єднання коду з репозиторію розробки; запуск автоматизованих тестів для перевірки, чи не порушено функціональність вже існуючого коду після змін; завантаження зібраного коду на тестовий сервер для подальшого тестування. Continuous Delivery включає: автоматизоване розгортання програмного забезпечення на тестовий та/або продуктивний сервер; розгортання в середовищі виробництва, що забезпечує можливість автоматичного вивантаження коду може в середовищі виробництва. Інструменти CI/CD допомагають втілювати ці практики та автоматизувати весь цикл розробки і постачання програмного забезпечення. Деякі з популярних інструментів включають Jenkins, GitLab

CI/CD, Travis CI, CircleCI, GitHub Actions та інші. Процес CI/CD дозволяє командам розробників швидко та надійно впроваджувати новий функціонал та виправлення помилок у виробництві, зменшуючи ризик непередбачених проблем та покращуючи швидкість розробки та впровадження програмного забезпечення.

Low-code програмування - це метод розробки програмного забезпечення, при якому використовуються візуальні елементи та графічний інтерфейс для створення програм, зазвичай з мінімальним написанням коду. Ця технологія спрощує процес розробки та дозволяє людям з обмеженими навичками програмування швидше створювати функціональні додатки. Основні характеристики low-code програмування: візуальний інтерфейс, використання готових компонент; швидкість розробки; є зменшення кількості написаного коду; легка інтеграція з іншими системами та сервісами. Замість традиційного програмування кодом, розробники можуть використовувати графічний інтерфейс для створення додатків. Це може включати в себе перетягування та розміщення компонентів, вибір опцій та налаштування параметрів. Low-code платформи надають готові компоненти та модулі, які можна легко використовувати та конфігурувати. Це включає в себе готові блоки коду, базові функції та інші елементи. Розробка з використанням low-code може бути значно швидшою порівняно з традиційним програмуванням, оскільки вимагає менше написання коду та дозволяє швидко створювати прототипи. Основною ідеєю є зменшення кількості написаного коду. Це може полегшити розуміння та підтримку програми в майбутньому. Low-code робить розробку програмного забезпечення доступною для широкого кола користувачів, включаючи тих, хто не має глибоких знань в області програмування. Ці платформи надають можливості для легкої інтеграції з іншими системами та сервісами. І хоча low-code розробка може бути ефективною для швидкої постановки завдань та їх реалізації для користувачів, важливо враховувати її обмеження та відповідально використовувати в

конкретних випадках, де вона відповідає вимогам проекту. Ця технологія пов'язана з використанням готових застосунків, зокрема, бібліотек, а також можливостей використання Wizards у відповідних ситуаціях, хоча при цьому генерується відповідний код, але без безпосереднього написання коду. Тож, можна вважати, що в цій роботі також частково присутня технологія low-code розробки.

Інтелектуальний аналіз даних (Intelligent Data Analysis) - це галузь, яка поєднує методи і техніки аналізу даних, машинного навчання та штучного інтелекту з метою отримання корисних висновків та прийняття обґрунтованих рішень. Основні аспекти інтелектуального аналізу даних включають наступні можливості. В обробці даних дуже важливим етапом є збір та очищення даних, що включає отримання даних (інколи з різних джерел) та їх підготовка для подальшого аналізу, включаючи видалення аномалій та обробку пропущених значень. Далі - етап інтеграції даних – об'єднання даних з різних джерел для створення повних та комплексних датасетів. Тільки після цього можливо проводити аналіз даних: визначення ключових факторів, структури і характеристик даних; застосування алгоритмів для виявлення залежностей, прогнозування та класифікації на основі вхідних даних. Сюди можна віднести машинне навчання або більш класичне моделювання на основі методів математичної статистики та теорії ймовірностей. Тут можуть бути задіяні відповідні програмні інструменти: або спеціальні пакети прикладних програм, або модулі інформаційної системи, в яких використовуються бібліотеки з відповідними алгоритмами. Саме такий підхід у обробці інформації було задіяно в цій роботі. До інтелектуальної обробки даних, окрім описаного аналізу, відноситься виявлення та інтерпретація структур та закономірностей в даних, аналіз та розуміння текстової інформації для отримання смислової інформації, тобто розпізнавання образів та обробка мовних даних. Результатом обробки даних зазвичай є візуалізація даних, тобто корисна інформація, подана у зрозумілій формі для сприйняття та визначення



конкретних рекомендацій для надання та обґрунтованих висновків. Реалізація результатів полягає у впровадженні знань та рекомендацій у робочі процеси та системи організації. В цій роботі демонструються деякі результати обробки даних в інформаційній системі, що буде описано в п. 3.2.

### **1.3. Висновки до першого розділу**

Проведено аналіз наявних в Україні освітніх інформаційних автоматизованих систем. Існують деякі державні освітні системи для періодичного моніторингу освітніх індикаторів та формування різноманітної звітності по закладам освіти. У повсякденному використанні функції автоматизації обмежуються електронними щоденником та журналом, і не забезпечують поточних проблем автоматизації у роботі закладів освіти, що пов'язана зі зберіганням, обробкою, вибором інформації для її ефективного аналізу та створенню необхідної документації.

Стандартизоване програмне забезпечення для потреб освітніх закладів, будь то, середні школи, коледжі, заклади вищої освіти, відсутнє. Кожен заклад самостійно може організувати управління та аналіз інформації завдяки власним розробкам програмного забезпечення або придбаних систем, розроблених приватними розробниками, але постає питання відсутності грошового забезпечення.

Сучасні інноваційні технології дозволяють автоматизувати багато процесів в закладі освіти та об'єднати управління в одну інформаційну систему. Це сприяє підвищенню ефективності моніторингу і аналізу даних, а також управління закладом в цілому.

Сучасні моделі проектування та комп'ютерні технології є інструментами для розв'язання задач даної роботи

## РОЗДІЛ 2

### ПРОГРАМНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ОРГАНІЗАЦІЇ ТА ОБРОБКИ ДАНИХ В ІНФОРМАЦІЙНІЙ СИСТЕМІ

#### 2.1. Системи управління базами даних та їх роль в організації даних в інформаційних системах

Інформаційні системи будуються на основі баз даних, в яких знаходиться вся необхідна інформація. Інформаційна система містить у собі множину програмних рішень автоматизації процесів конкретної організації. Одна з найбільш важливих вимог до такої інформаційної системи – це здатність до адаптації і подальшого розвитку.

Інформаційна система – це комунікаційна система збору, передачі, переробки інформації про об'єкт, що постачає працівникам різного рангу інформацію для реалізації функцій управління. Переважна більшість інформаційних систем працює в режимі діалогу з користувачем.

Структура інформаційної системи визначається інформацією та задачами, що мають розв'язуватися за допомогою організації даних.

Незалежно від сфер застосування інформаційні системи мають декілька класів компонентів: забезпечуючі компоненти, функціональні підсистеми, організаційні компоненти.

Забезпечуючі компоненти - це сукупність мережної, технічної, програмної, інформаційної, організаційної, правової, ергономічної, технологічної підсистем.

Вимоги до технічної компоненти комп'ютерної системи стандартизовані та єдині. Інформаційне забезпечення включає системи класифікації і кодування, системи документації, оптимізацію документообігу формування звітності. Обов'язковим компонентом є програмне забезпечення тобто сукупність програмних засобів для створення, підтримки,

супроводження систем організації та обробки даних, що складається з базового системного і спеціального прикладного програмного забезпечення. Технічне забезпечення слугує для забезпечення функціонування систем організації та обробки даних. Технологічне забезпечення утворюють різні технічні документи, що регламентують процес обробки даних.

Для можливості обробки інформації обов'язковим є математичне забезпечення, тобто сукупність математичної складової у вигляді методів, моделей, алгоритмів.

Обов'язковим в інформаційній системі є організаційне забезпечення, що дозволяють взаємодію працівників з технічними засобами і між собою при експлуатації системи. Організаційне забезпечення виконує ряд функцій, таких, як аналіз завдань, що підлягають автоматизації, створення структури інформаційної системи, визначення працівників на різних рівнях, які будуть користуватися інформаційною системою для аналізу результатів; складання технічного завдання на проектування інформаційної системи, обґрунтування її ефективності, зокрема, що є більш економічним: придбати готову систему чи створити для своєї організації індивідуальну; розробка управлінських рішень щодо методології розв'язання проблем і задач, зокрема, визначення тих форм і звітів, що дають змогу зробити управління більш ефективним.

Функціональні, організаційні компоненти інформаційної системи залежать від організації і типу задач. Функціональні підсистеми можуть бути відокремленими, наприклад, бухгалтерський облік, облік відділу кадрів, облік успішності, якщо мова йде про навчальний заклад, система документів і звітів щодо вимог міністерства і т. ін.

Невід'ємною частиною автоматизованих інформаційних систем є бази даних, а також системи управління базами даних.

Існують різні типи баз даних в залежності від задач, для яких вони призначені, і відповідно, працюють по-різному, надають користувачу різні

функції і засоби. Але більшість систем управління базами даних спирається на єдиний сталий комплекс основних понять.

База даних – це сукупність взаємопов'язаних даних, які зберігаються у вигляді таблиць. До даних належить не лише інформація в цих таблицях, але й інші об'єкти бази даних. Основні характеристики баз даних включають логічний зв'язок, нормалізацію даних, незалежність даних від програм, підтримку цілісності та захист від несанкціонованого доступу.

Системи управління базами даних – це програмні засоби, що дозволяють проектувати та створювати структури баз даних, вносити та вилучати необхідні дані, а також працювати з ними. Існує численні спеціалізовані мови програмування для таких систем, такі як FoxPro, Dbase+, Clipper, Paradox та інші. Це набір програмних інструментів, необхідних для створення, використання та підтримки баз даних.

Створювані форми, запити і звіти впродовж користування Системою Управління Базами Даних (СУБД) надають можливість оперативно та ефективно оновлювати, видаляти або додавати необхідні дані, аналізувати інформацію, друкувати звіти, створювати діаграми та виконувати обчислення безпосередньо в межах СУБД. Також існує можливість імпортувати дані з інших джерел та експортувати результати до інших програм.

Різні Системи Управління Базами Даних (СУБД) можуть взаємодіяти з об'єктами та надавати різні функції та інструменти для користувача. Більшість СУБД базуються на загальних концепціях та принципах.

До основних моделей даних належать:

- ієрархічна,
- мережева,
- реляційна.

В ієрархічній моделі ( Рисунок 2.1) даних інформацію про об'єкти предметної галузі подають у вигляді дерева. Це сукупність об'єктів різного рівня, причому об'єкти нижнього рівня підпорядковані об'єктам верхнього

рівня. Усі вершини дерева мають певні рівні. На найвищому (першому) рівні знаходиться тільки одна вершина, яка називається коренем дерева – вузлом. Вона з'єднується ребрами з усіма вершинами, що знаходяться на наступному – другому рівні, і лише з ними. Вершини другого рівня з'єднуються з вершинами третього рівня ребрами так, що кожна вершина третього рівня з'єднана тільки з однією вершиною другого рівня, і т. д.

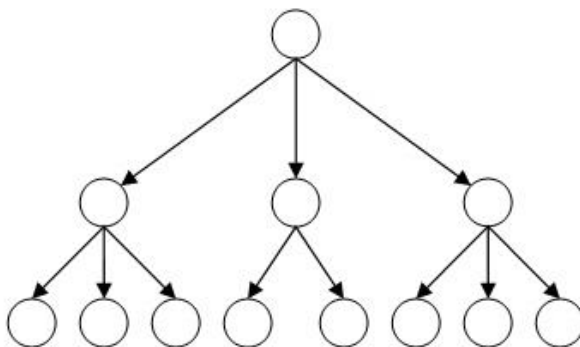


Рисунок 2.1 – Ієрархічна модель даних

У мережевій моделі (Рисунок 2.2) основною структурою є мережі, тобто довільні графи, у вершинах яких записано деяку інформацію, а ребра відповідають зв'язкам між вершинами (граф — це математична конструкція, що складається з вершин і ребер). Кожне ребро з'єднує дві вершини.

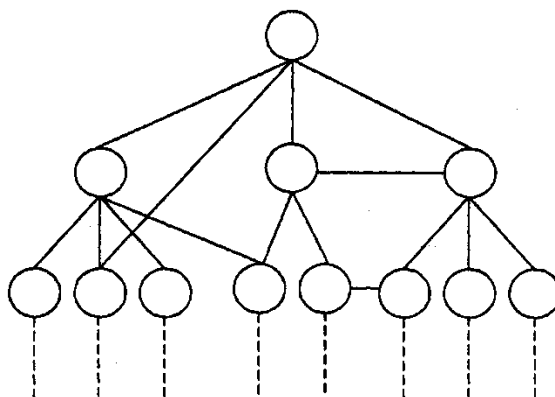


Рисунок 2.1 – Мережева модель даних

Мережева база даних відрізняється більшою гнучкістю, оскільки в ній існує можливість встановлювати додатково до вертикальних ієрархічних зв'язків горизонтальні зв'язки. Це полегшує процес пошуку потрібних елементів даних, оскільки вже не вимагає обов'язкового проходження кількох ієрархічних рівнів.

Найбільше поширена реляційна модель даних (РМД), яку в 1970 р. запропонував англійський математик Кодд (Рисунок 2.3). Реляційною БД називається БД, що містить інформацію, подану у відповідних прямокутних таблицях. Така таблиця називається відношенням. Таблиця складається з рядків – записів (або кортежів). Таблиця має заголовок (назву відношення) і стовпці, або поля таблиці, що також мають заголовки. Кожен рядок таблиці містить інформацію про один окремих об'єкт системи (про конкретну книжку, співробітника установи тощо), а кожен стовпчик — певні характеристики (властивості, атрибути) цих об'єктів. Наприклад, атрибутами об'єктів можуть бути автор книжки, посада співробітника, відділ, в якому він працює, тощо. Рядки такої таблиці називаються – записами, а стовпчики – полями. Кожен запис повинен відрізнятися від інших значеннями головного ключа – певного поля або сукупності полів, що ідентифікують запис, роблять його унікальним.

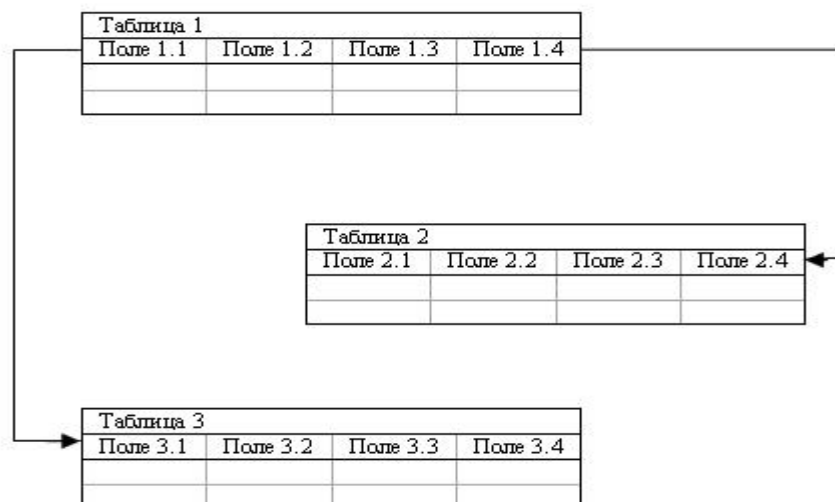


Рисунок 2.3 – Реляційна модель даних

Основне обмеження реляційної моделі: будь-яка таблиця не має містити однакових записів. Для дотримання цього обмеження існує ключ – це поле таблиці або сукупність полів, в яких значення не повторюються і таким чином забезпечується унікальність кожного запису таблиці.

Реляційна модель БД має такі властивості: кожен елемент таблиці – один елемент даних; всі дані в одному і тому ж стовпчику таблиці мають один тип; кожен стовпчик має унікальне ім'я; однакові рядки в таблиці відсутні.

Ключ може бути простим (складається із одного поля), або складеним (складається з кількох полів). Ключі використовуються для зв'язку між таблицями.

Одним із прикладів системи з реляційною базою даних є Microsoft Access. У відмінну від інших настільних СУБД, Access зберігає всі дані в одному файлі, хоча самі дані можуть бути розміщені у різних таблицях. До даного поняття належить не лише інформація в таблицях, але й інші об'єкти бази даних. Практично всі основні операції у Access можна виконати за допомогою Майстрів, які допомагають уникнути рутинних завдань при роботі з даними і створенні додатків, або за допомогою вбудованої мови програмування Visual Basic for Applications,

Microsoft Access має широкий спектр функцій, що включають зв'язані запити, зв'язок із зовнішніми таблицями та базами даних. Вбудована мова програмування Visual Basic for Applications дозволяє створювати програмні коди і додатки, які будуть працювати зі створеними в MS Access базами даних.

MS Access також володіє розвиненою системою довідки та інтуїтивно зрозумілим інтерфейсом, що сприяє полегшенню процесів проектування та створення баз даних, а також виконання вибірок із них. У програмі також наявні різноманітні діалогові інструменти, які допомагають швидше створювати додатки, уникаючи необхідності розробки запитів на мові SQL чи написання макросів або модулів на мові VBA.

Дані зберігаються і, за необхідності, вилучаються для подальшого аналізу, що може включати моделювання соціальних або фінансово-економічних та показників, вивчення зв'язків між факторами, побудову короткотривалих і довготривалих прогнозів і інші завдання, використовуючи прикладне програмне забезпечення або створене на мові програмування.

Зручність роботи з даними підтримується мовою запитів SQL (Structured Query Language), призначеною для взаємодії користувачів з реляційними базами даних.

Розрізняють дві основні мови в контексті баз даних: мова визначення даних та мова керування даними.

Мова визначення даних надає можливість створювати та видаляти таблиці в базі даних, визначати індекси (ключі), описувати зв'язки між таблицями та накладати умови на таблиці. Ці операції також можуть бути виконані за допомогою конструктора таблиць та схеми даних.

Мова керування даними призначена для виконання запитів, оновлення, додавання нових записів та видалення існуючих. До команд мови керування даними входять інструкції, такі як: SELECT (вибірка даних), UPDATE (оновлення даних), DELETE (видалення даних), INSERT (додавання даних).

Для формування запитів використовуються конструкції з FROM, ORDER BY, DESC, WHERE. За допомогою цих конструкцій можна створювати складні запити, які об'єднують кілька умов та виконують різні операції, такі як сумування.

Результати запиту виводяться у просту таблицю. Для зручності аналізу результатів можна створити екранні форми на основі запиту. Для друку використовують спеціально розроблені форми – звіти. Звіти можуть містити докладні відомості про окремі записи, зведені відомості за групою записів або об'єднувати обидві ці можливості. Можна створювати наклейки для списку розсилок та виконувати інші завдання. Як і форми, звіти створюються на підставі даних таблиць чи даних запитів.



Для розробки інформаційно-аналітичної системи було обрано Microsoft Access, з вбудованою мовою програмування Visual Basic, та програмні інструменти Python та його бібліотеки.

Microsoft Access має широкий спектр функцій, що включають зв'язані запити, зв'язок із зовнішніми таблицями та базами даних. Вбудована мова програмування Visual Basic for Applications дозволяє створювати програмні коди і додатки, які будуть працювати зі створеними в MS Access базами даних.

MS Access також володіє розвиненою системою довідки та інтуїтивно зрозумілим інтерфейсом, що сприяє полегшенню процесів проектування та створення баз даних, а також виконання вибірок із них. У програмі також наявні різноманітні діалогові інструменти, які допомагають швидше створювати додатки, уникаючи необхідності розробки запитів на мові SQL чи написання макросів або модулів на мові VBA.

MS Access відноситься до реляційних баз даних. Основні вимоги, які необхідно враховувати при проектуванні бази даних, включають наступне. Кожна таблиця повинна бути вільною від повторюючихся записів. Всі дані в одному стовпці таблиці мають однаковий тип. Кожен стовпчик має унікальне ім'я. Кожна інформація повинна заноситися тільки один раз. У кожному таблицю включаються лише ті поля, які стосуються предметної області таблиці. Якщо головна таблиця має поле з повторюючимися значеннями, потрібно організувати підстановку із іншої таблиці-довідника. Довідник, у свою чергу, має включати тільки ті поля, які безпосередньо стосуються об'єкту довідника. При проектуванні необхідно забезпечити цілісність даних.

Для зв'язку між таблицями використовуються ключі - поле таблиці або сукупність полів, в яких значення не повторюються і, таким чином, забезпечують унікальність кожного запису таблиці. Ключ може складатися з одного поля або з кількох полів.

Зв'язування таблиць дозволяє:

- о автоматично контролювати цілісність введених у базу даних даних за допомогою засобів СУБД;
- о спростити доступ до даних при виконанні операцій пошуку, перегляду, редагування, вибірки та підготовки звітів завдяки автоматичному зверненню до полів пов'язаних записів.

Основні типи зв'язку між таблицями включають:

- о один до багатьох (1 :  $\infty$ );
- о багато до одного ( $\infty$  : 1);
- о один до одного (1 : 1);
- о багато до багатьох ( $\infty$  :  $\infty$ ).

Контроль цілісності зв'язків зазвичай передбачає аналіз вмісту двох таблиць для визначення дотримання таких правил [ 6]:

- о кожному запису основної таблиці відповідає нуль або більше записів додаткової таблиці.
- о кожен запис додаткової таблиці має рівно один батьківський запис основної таблиці.
- о контроль цілісності реалізується під час виконання основних операцій з даними двох таблиць, таких як введення нових записів, модифікація записів та видалення записів.

Можна зазначити наступні етапи проектування бази даних:

1. Дослідження предметної області та формулювання основних допущень (накладених умов). На цьому етапі складається список всіх форм і звітів, які можуть бути затребувані користувачами вашої БД.
2. Аналіз даних. Складання переліку усіх елементів, що входять у форми і звіти, та згрупування їх в таблиці БД.
3. Встановлення взаємозв'язків між елементами даних. Визначення первинних й вторинних (зовнішніх) ключів та відносин. Організація полів даних в таблицях відповідно до правил нормалізації

Структура бази даних обумовлена даними та інформацією, яку необхідно систематизувати та обробити, а також завданнями, що стоять перед розробником і дослідником. Базу даних можна доповнювати потрібними запитами і звітами в процесі роботи з даними. СУБД може бути використана для аналізу даних.

## **2.2. Програмні інструменти та математичні засади моделювання даних в інформаційних системах**

База даних є моделлю організації структурованих даних. На основі запитів і звітів можна отримувати потрібну інформацію та аналізувати її класичними математичними методами, використовуючи при цьому сучасні програмні інструменти.

Сучасними інструментами аналітики, які можуть бути використані в аналізі даних, зокрема, в освітньому закладі, є кластерний і факторний аналіз, кореляційний, регресійний, економетричний аналіз, нейронні мережі, штучний інтелект, машинне навчання і т. ін., тобто можна застосувати відповідні до задачі відомі технологічні, програмні, математичні методи та моделі, що допомагають в аналогічних задачах обробки, аналізу, прогнозування даних для прийняття обґрунтованих рішень, зокрема, в управлінні закладом.

### **Використання інструментів MS Office для аналізу даних**

Для обробки структурованих даних використовуються різні технології, зокрема, одним з основних інструментів є реляційні бази даних, які дозволяють створити потрібний для вивчення і дослідження даних формат зберігання. База даних представляє собою сукупність взаємопов'язаних даних, що зберігаються в таблицях. До даних відноситься не тільки інформація в

таблицях, але й інші об'єкти бази даних. Основними властивостями реляційних баз даних є логічний зв'язок, нормалізація даних, незалежність даних від програм, наявність засобів підтримки цілісності даних та захисту від несанкціонованого доступу.

За допомогою систем управління базами даних можна проектувати і створювати структури бази даних, вносити і вилучати потрібні дані і працювати з ними. Існує багато спеціальних мов програмування для систем управління базами даних, зокрема, FoxPro, Dbase+, Clipper, Paradox та інші. Це набір програмних засобів, необхідних для створення, використання і підтримки баз даних.

Створювані форми, запити і звіти дозволяють швидко й ефективно оновлювати дані, вилучати або додавати потрібні дані, аналізувати дані, друкувати звіти, діаграми і робити обчислення безпосередньо в СУБД, імпортувати дані та експортувати результати в інші програми. Різні СУБД можуть працювати по-різному з об'єктами, надаючи різні функції і засоби. Більшість СУБД спираються на єдиний сталий комплекс основних понять.

Одним із прикладів є система управління базами даних MS ACCESS. На відміну від інших настільних СУБД, Access зберігає всі дані в одному файлі, хоч дані розподіляються по різних таблицях. До даних відноситься не тільки інформація в таблицях, але й інші об'єкти бази даних. Майже всі основні операції Access можна виконувати за допомогою Майстрів (Wizards), які допомагають уникнути рутинних дій при роботі з даними і створенні додатків. Також MS Access має розвинену систему довідки і зрозумілий інтерфейс, що полегшує проектування, створення БД і вибірку даних з неї. Різноманітні діалогові засоби в MS Access сприяють швидшому створенню додатків без необхідності розробки запитів на мові SQL або програмування макросів чи модулів на мові VBA.

MS Access відноситься до реляційних баз даних, і при проектуванні бази даних дотримуються основних обмежень. Будь-яка таблиця не може містити

однакових записів, усі дані в одному стовпчику таблиці мають один тип, кожний стовпчик має унікальне ім'я. Інформація повинна вноситися тільки один раз, а в кожену таблицю включаються лише ті поля, що відносяться до предметної області таблиці. Якщо головна таблиця має поле, значення якого повторюються, використовується підстановка із іншої таблиці – довідника. В свою чергу, довідник має містити тільки ті поля, що безпосередньо відносяться до об'єкту довідника. При проектуванні необхідно забезпечити цілісність даних.

На прикладі СУБД Access розглянемо застосування баз даних до регресійного аналізу безпосередньо в середовищі, що докладно описано в [4].

В п.1.2 було описано можливість використання мови SQL для моделювання потрібної для аналізу інформації, тобто інформація може бути вилучена, доповнена, відсортована за необхідністю. Результати запиту виводяться у зручну таблицю.

Для огляду та аналізу оброблених даних зручно будувати екранні форми на основі запиту. Для друку результатів найкраще використовувати спеціально розроблену форму – звіт. Звіт є об'єктом бази даних, який використовується для відображення даних на екрані та для їх друку. В звітах можуть міститися детальні відомості про окремі записи, зведені дані за великою групою записів або будь-яке поєднання обох. Крім того, звіти Access можна використовувати для створення наклейок для списку розсилок та інших завдань. Як і форми, звіти створюються на основі даних таблиць або даних запитів. Для їх створення використовують "Майстер" або "Конструктор" звітів, принцип роботи яких аналогічний відповідним інструментам для форм.

Структура бази даних для конкретного завдання обумовлена даними та інформацією, яку потрібно систематизувати і обробити, а також завданнями, що стоять перед розробником і дослідником. Створену базу даних можна доповнювати необхідними запитамі і звітами, які виникають також під час роботи з даними.

Розглянемо можливість елементарного регресійного аналізу в MS Access. Для більш докладного моделювання більш доцільним є використання інших інструментів, на яких зупинимось далі. Але, все ж таки навіть в середовищі СУБД є можливість побудови моделей та аналізу даних на їх основі [ 6].

На першому етапі проектується система реляційних таблиць для збереження даних.

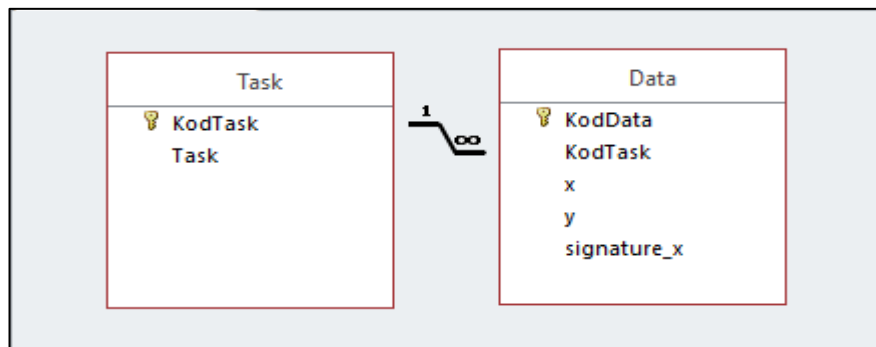


Рисунок 2.4 – Проста схема даних

Таблиця Task призначена для збереження назв задач аналізу:

Унікальний код проекту – *KodTask*

Назва проекту – *Task*

Вхідні дані зберігаються у таблиці Data.

Унікальний код запису – *KodData* ,

Код проекту – *KodTask*

*x*, *y* – масиви вхідних даних даних проекту.

Таким чином, програма дозволить швидко знаходити потрібну задачу, змінювати за необхідності дані та розраховувати необхідні параметри.

Розрахунок параметрів моделі проводиться за допомогою SQL-мови.

Фрагменти коду приведені нижче.

Для параметрів лінійної моделі перетину і нахилу:

```

SELECT DataTask.KodTask, Count(DataTask.KodTask) AS n, Sum(DataTask.x)
AS SumX, Sum(DataTask.y) AS SumY, Sum([x]*[y]) AS SumXY, Sum([x]*[x]) AS
SumX2, Avg(DataTask.y) AS Ysr, Round((([n]*[SumXY]-
[SumX]*[SumY])/([n]*[SumX2]-([SumX]^2),4) AS a1, Round((([SumY]-
[a1]*[SumX])/[n],4) AS a0
FROM DataTask
GROUP BY DataTask.KodTask

```

Для обчислення статистики (коефіцієнт детермінації та стандартна похибка регресії):

```

SELECT Count(DataTaskStat.KodTask) AS n, DataTaskStat.KodTask,
Sum(DataTaskStat.delY_Yr) AS SumDely_Yr, Sum(DataTaskStat.delY_Ysr) AS
SumDely_Ysr, Sum(DataTaskStat.delYr_Ysr) AS SumDelyr_Ysr,
Round([SumDelyr_Ysr]/[SumDely_Ysr],4) AS R2,
Round(Sqr([SumDely_Yr]/([n]-2)),4) AS s
FROM DataTaskStat
GROUP BY DataTaskStat.KodTask;

```

Після внесення даних задачі за допомогою кнопки Аналіз результат виводяться в окрему екрану форму (Рисунок 2.5).

Це одна з можливостей аналізу даних, але є більш зручні інструменти для цієї задачі.

основна статистика для оцінювання результатів моделювання

Більше можливостей для моделювання і аналізу дає зручний, зрозумілий та доступний інструмент майже у всіх закладах освіти MS Excel, що має пакет аналізу з широкими можливостями, придатний для отримання результатів у першому наближенні для уявлення про характер і розподіл даних, а також основні характеристики моделі даних. Цей пакет дозволяє розв'язувати задачі

не тільки регресійного характеру, а й задачі оптимізації. Досить велика потужність пакету обумовлена можливістю обробляти порядка 40000 записів і варіювати в оптимізаційних задачах більше 20 змінних. Доступність MS Office з додатками на комп'ютерах освітніх закладів в Україні робить його популярним інструментом для більшості задач, що виникають в процесі роботи і адміністративного управління кожного дня.

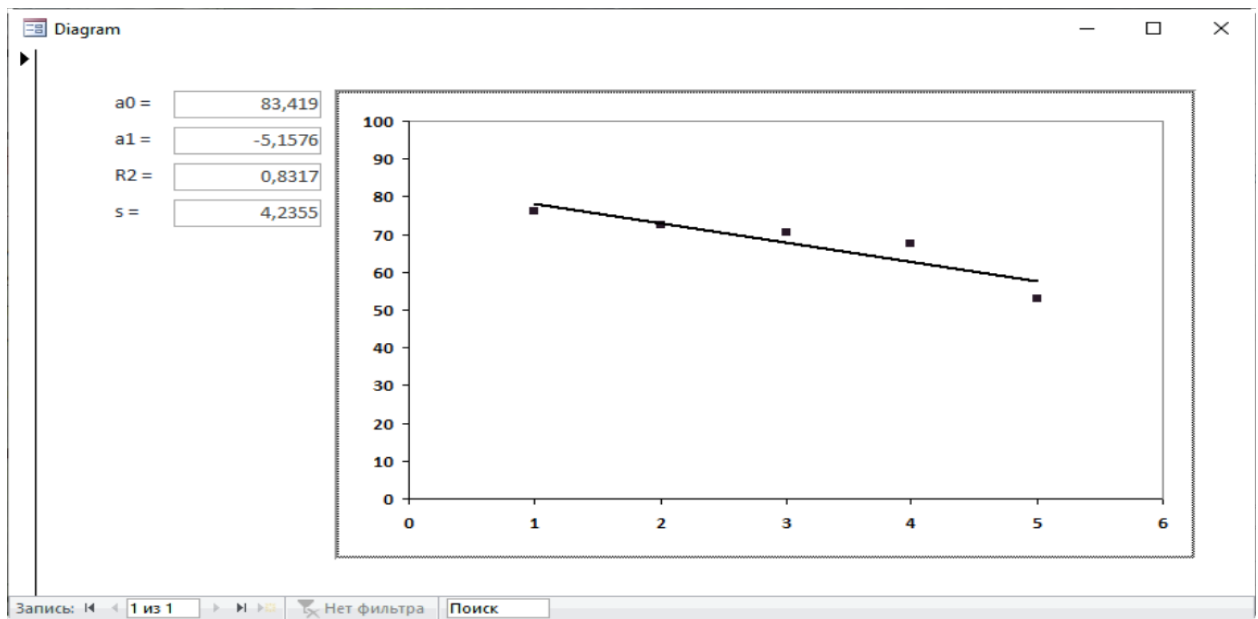


Рисунок 2.5 – Розподіл вхідних даних, регресія, визначені параметри моделі,

Для автоматизованого управління і програмування в MS Office використовують VBA for application, що робить цей програмний інструмент дуже важливим і універсальним для освітніх закладів. Окрім цього, при роботі з базами даних є можливість використання Майстрів та Конструкторів. Це розширює можливості програмного забезпечення для моделювання даних навіть не фахівцями-програмістами, а звичайними користувачами, більшість з не володіють глибокими комп'ютерними знаннями і навичками.

Можна відзначити для задач оптимізації процедуру «Пошук рішення» з аналізом коефіцієнтів цільової функції та обмежень, перевірка стійкості



розв'язків; пакет «Аналіз даних» для проведення однофакторного дисперсійного аналізу, двофакторного дисперсійного аналізу з повтореннями та без повторень, визначення кореляцій, коваріацій, описувальної статистики, для побудови експоненціального згладжування, деяких тестів: двухвибіркового тесту Фішера для дисперсії, аналізу Фур'є, генерації випадкових чисел та ін. На жаль, аналіз результатів економетричного моделювання є обмеженим тільки деякими видами моделей. Є можливості побудови діаграм та графіків різного вигляду; проведення автоматизації розрахунків за допомогою макросів та програмування на VBA. Особливо актуальним це є з огляду на новий застосунок Power BI, інтегрований в MS Office, можливий для використання в безкоштовній версії, для обробки, моделювання та візуалізації даних.

Для розрахунку статистики моделювання і перевірки лінійних регресійних моделей, до яких мають бути зведені будь-які нелінійні моделі, в електронних таблицях EXCEL використовувати вбудовані статистичні функції, побудовані на запрограмованих формулах, що є відомими із теорії ймовірностей, математичної статистики та економетричного аналізу, і які є основою для прогнозних моделей, побудованих на відповідних математичних засадах. Аналогічні формули використовуються і в модулі інформаційної системи, що відповідає за аналіз даних відповідно до указаної методики. Частина роботи з аналізу можна проводити і в ручному режимі на основі відповідних даних, що потребують аналізу та вилучених із бази даних.

Функція ЛИНЕЙН в електронних таблицях EXCEL дозволяє отримати необхідну інформацію по моделі в одному масиві.

ЛИНЕЙН(известные\_значения\_у;известные\_значения\_х;конст;статистика)

Задаються для конкретної задачі:

известные\_значения\_у – значення показника

известные\_значения\_х – значення фактору або декількох одним масивом

конст – значення, яке показує, чи необхідно, щоб константа мала нульове значення чи обчислювалася звичайним способом

статистика – значення, яке показує, чи необхідно, щоб виводилась додаткова статистика по регресії або тільки значення параметрів регресії

Регресійна статистика виводиться у масив значень:

$a_n$	$a_{n-1}$	...	$a_1$	$a_0$
$S_n$	$S_{n-1}$	...	$S_1$	$S_0$
$R^2$	$S_y$			
$F$	$Df$			
$ssr$	$sse$			

де

$a_n, a_{n-1}, \dots, a_0$	параметри лінійної регресії
$S_n, S_{n-1}, \dots, S_0$	стандартні похибки параметрів $a_n, a_{n-1}, \dots, a_0$
$R^2$	коефіцієнт детермінації
$S_y$	стандартна похибка регресії
$F$	F-статистика, розраховане значення для даних задачі
$Df$	ступені вільності
$ssr$	сума квадратів похибок, визначених регресією
$sse$	сума квадратів залишків похибок відхилень

Для визначення критичного значення критерію Фішера використовується функція FРАСПОБР ( $\alpha$ ;  $p$ ;  $n-p-1$ ) для надійності  $(1-\alpha)$  та при  $p$  факторах, де  $n$  – кількість спостережень.

Визначення критичного значення критерію Стьюдента відбувається за допомогою функції СТЬЮДРАСПОБР ( $\alpha$ ;  $n-p-1$ ) для надійності  $(1-\alpha)$ , де  $n$  – кількість спостережень,  $p$  – кількість факторів.

В регресійному аналізі використовуються функції: для визначення добутку матриць – МУМНОЖ, для обчислення оберненої матриці – МОБР, для транспонування матриць –ТРАНСП.

Для інших дій, наприклад, визначення суми значень, добутків відповідних елементів масивів, максимальних та мінімальних значень та ін. виконуються за допомогою відповідних стандартних функцій MS EXCEL. Для аналізу даних можна користатися «Пакетом аналізу», а також процедурою «Пошук рішення», якщо виникає необхідність моделювання задач оптимізації, яка допускає варіації більше 20 змінних.

MS Excel є зручним інструментом для візуалізації результатів досліджень у вигляді графіків, різного типу діаграм, особливо із застосунком Power BI, який наразі має цілий ряд безкоштовних інструментів і доповнює можливості електронних таблиць. Разом з Power Point і Word MS Office є незамінним інструментом для освітян, більшість з яких дуже обмежено володіють навичками роботи з програмними застосунками.

Тож, такі переваги, як доступність застосунків та зручність передачі даних, пакети аналізу даних, можливість програмування у вбудованій мові VBA для автоматизації розрахунків, робить MS Office популярним інструментом в закладах освіти.

Використання мов програмування і пакетів для роботи з даними

Функціонал інформаційної системи для аналізу реалізується за допомогою мови програмування Python і відповідних бібліотек. Коротко наведемо можливості основних програмних модулів, що використовуються при аналізі даних [7, 8, 10].

NumPy - основний пакет для виконання наукових розрахунків на Python. Зверху NumPy побудовано інші бібліотеки.

Основні можливості пакету:

- швидкий і ефективний об'єкт багатовимірного масиву ndarray;
- функції для виконання обчислень з елементами одного масиву або математичних операцій з декількома масивами;
- засоби для читання і запису на диски наборів даних, представлених у вигляді масивів;
- операції лінійної алгебри, перетворення Фур'є і генератор випадкових чисел;
- засоби для інтеграції з кодом, що написаний на C, C++ або Fortran.

Крім прискорення роботи з масивами, однією з основних цілей NumPy стосовно аналізу даних є організація контейнера для передачі даних поміж алгоритмами. Як засіб зберігання і маніпуляції даними масиви NumPy значно ефективніші вбудованих в Python структур даних.

Багато засобів обчислень орієнтовано на Python та використовують масиви NumPy як основні структури даних.

Бібліотека pandas надає структури даних і функції для роботи зі структурованими даними. Основні об'єкти pandas - DataFrame - двовимірна таблиця, в якій рядки і стовпці мають мітки, і Series - об'єкт одновимірного масиву з мітками.

У бібліотеці pandas поєднуються висока продуктивність засобів роботи з масивами, притаманна NumPy, і гнучкі можливості маніпулювання даними, властиві електронним таблицям і реляційним базам даних (наприклад, на основі SQL). Оскільки маніпулювання даними, їх підготовка і очищення грають дуже велику роль в аналізі даних [7, 10], pandas є одним з основних інструментів.

Основні можливості бібліотеки:

- розвинені засоби індексування, що дозволяють просто змінювати форму наборів даних, формувати зрізи, виконувати агрегування і вибирати підмножини;

- структури даних з позначеними осями підтримують автоматичне або явне вирівнювання даних, що виключає появу типових помилок при роботі з невіривняні даними і даними з різних джерел, які по-різному індексовані;
- вбудована функціональність часових рядів;
- одні і ті ж самі структури даних підтримують як тимчасові ряди, так і дані інших видів;
- арифметичні операції;
- гнучка обробка відсутніх даних;
- інтеграція даних;
- підтримка з'єднання і інших реляційних операцій, наявних в популярних базах даних (наприклад, на основі SQL).

Сама назва `pandas` утворена від `panel data` (панельні дані), що застосовуються в економетричних дослідженнях для позначення багатовимірних структурованих наборів даних.

`matplotlib` - інструмент Python для створення графіків і інших способів візуалізації двовимірних даних. Вона підходить для створення графіків, придатних для публікації. Існують також інші бібліотеки візуалізації, але `matplotlib` використовується найчастіше і тому добре інтегрована з іншими частинами екосистеми.

`SciPy` - збір пакетів, призначених для вирішення різних стандартних обчислювальних задач:

- `scipy.integrate` - підпрограми чисельного інтегрування і розв'язання диференціальних рівнянь;
- `scipy.linalg` - підпрограми лінійної алгебри і розкладання матриць, доповнюють ті, що включені в `numpy.linalg`;
- `scipy.optimize` - алгоритми оптимізації функцій (знаходження екстремумів) і пошуку коренів;
- `scipy.signal` - засоби обробки сигналів;

- `scipy.sparse` - алгоритми роботи з розрідженими матрицями і розв'язання розріджених систем лінійних рівнянь;
- `scipy.special` - обгортка навколо SPECFUN, написаної на Fortran-бібліотеці, що містить реалізації багатьох стандартних математичних функцій, в тому числі гамма-функції;
- `scipy.stats` - стандартні безперервні і дискретні розподіли ймовірностей (функції щільності ймовірності, формування вибірки, функції безперервного розподілу ймовірності), різні статистичні критерії і додаткові описові статистики.

`scikit-learn` – основний інструментарій машинного навчання на Python. Є підмодулі для наступних моделей:

- класифікація: метод опорних векторів, метод найближчих сусідів, випадкові ліси, логістична регресія і т. ін.;
- регресія: Lasso, гребнева регресія і т. ін.;
- кластеризація: метод k середніх, спектральна кластеризація і т. ін.;
- зниження розмірності: метод головних компонент, відбір ознак, матрична факторизація і т. ін.;
- вибір моделі: пошук на сітці, перехресний контроль, метрики;
- попередня обробка: виділення ознак, нормування.

`scikit-learn` орієнтований головним чином на передбачення.

`Statsmodels` - пакет статистичного аналізу, містить алгоритми класичної статистики та економетрики. У нього входять наступні підмодулі:

- регресійні моделі: лінійна регресія, узагальнені лінійні моделі, лінійні моделі зі змішаними ефектами і т. ін.;
- дисперсійний аналіз (ANOVA);
- аналіз часових рядів: AR, ARMA, ARIMA, VAR і інші моделі;
- непараметричні методи: ядерна оцінка щільності, ядерна регресія;
- візуалізація результатів статистичного моделювання.

Пакет `statsmodels` орієнтований більшою мірою на статистичне виведення результатів, він дає оцінки невизначеності і р-значення параметрів.

Використовується разом з `NumPy` і `pandas`.

Найбільш інформативним для моделювання є використання пакету `statsmodels`, при виконанні якого в лістингу результатів можна бачити різноманітну статистику відповідно до обраної моделі.

Для прикладу розглянемо результат, отриманий за допомогою `statsmodels` при виборі однієї із запропонованих моделей у цій бібліотеці – лінійної регресії для одного фактору (модель OLS). Результат виглядає так, як показано на Рисунку 2.6. У якості метрики обрано метод найменших квадратів, який дає найкраще наближення у класі незміщених та ефективних оцінок.

На роздруківці бачимо значення багатьох критеріїв. Зокрема, найбільш загальним критерієм для оцінювання якості моделі є критерій Фішера. Статистичну значимість параметрів моделі можна оцінити за критерієм Стюдента.

Адекватність моделі оцінимо за допомогою критерію Фішера. Оцінювання статистичної значимості параметрів регресій, а також довірчих інтервалів регресій, проведемо на основі критерію Стюдента. Окрім цього, отримаємо іншу статистику по моделі і застосуємо модель для прогнозу. Статистика Дарбіна–Уотсона показує ще один важливий для ефективності використання методів моделювання, а саме, наявність або відсутність автокореляції, від чого залежить вибір методів. В результатах можна побачити корельованість факторів і правильність матриці кореляцій. Також є розраховані показники для більш досконалого аналізу моделі.

Значення фактору  $X$  і показника  $Y$  наведено в лістингу на Рисунку 2.6. у відповідних списках.

Зробимо коротко висновки щодо моделі. Модель OLS (метод `Least squares`) обрано для моделювання.  $X$  – фактор, незалежна змінна,  $Y$  – показник, залежна змінна. Коефіцієнт детермінації 0.958 показує сильну кореляцію між

фактором і показником. Скорегований коефіцієнт детермінації також спостерігаємо в результатах. Він є дещо меншим – 0.955, і загалом, має більш вагоме значення для багатofакторних моделей. З довірчою імовірністю 0.975 розраховано значення F-статистики 338, та t- статистики 18.396 для нахилу 1.3 а також аналогічні дані для перетину регресії. Тож, можна вважати модель адекватною, параметри статистично значимо відрізняються від нуля. Вірно специфіковано коваріаційну матрицю. Відсутність автокореляції в моделі обґрунтовується статистикою Дарбіна–Уотсона 1.875.

Критичні значення критеріїв не виводяться в цьому лістингу, але є відповідні функції в бібліотеці для їх отримання.

В прикладі показано розрахунок довірчого інтервалу фактору за заданим значенням показника. Аналогічно, можна отримати точкову оцінку прогнозу показника та надійний інтервал за заданого рівня значимості.

Можна оцінити також за даними розрахунку еластичність показника, що є одною з важливих характеристик інтенсивності зміни. Для указаних даних маємо середній коефіцієнт еластичності 0.609, що свідчить про нееластичність показника по фактору.

В якості X та Y можна обирати будь-які дані для побудови моделі. Звісно, при цьому дані мають задовольняти нормальному розподілу, в системі має бути відсутня мультиколінеарність, що є суттєвим для багатofакторних моделей.

В пакеті statsmodels використовуються стандартні методи класичного економетричного моделювання, що є найбільш прийнятним для аналізу статичних даних в інформаційній системі.



```

Номер фактору: 1
Значення для прогнозу показника y: 120000
Дані показника y: [ 35340. 45140. 53513. 61354. 41659. 65355. 78222. 77231. 88677.
90922. 95130. 86811. 97991. 99536. 105327. 115619. 120000.]
Дані фактора x 1 [[12046. 15039. 17015. 20803. 17501. 23083. 36370. 38295. 40323. 43783.
48836. 45415. 51196. 55240. 65371. 68158. 70200.]]
Результат розрахунку:

```

```

                                OLS Regression Results
=====
Dep. Variable:                    y      R-squared:                0.958
Model:                            OLS    Adj. R-squared:           0.955
Method:                            Least Squares  F-statistic:             338.4
Date:                               Mon, 10 May 2021  Prob (F-statistic):      1.06e-11
Time:                               20:41:06    Log-Likelihood:         -169.43
No. Observations:                  17      AIC:                    342.9
Df Residuals:                      15      BIC:                    344.5
Df Model:                           1
Covariance Type:                   nonrobust
=====
                                coef      std err          t      P>|t|      [0.025      0.975]
-----+-----
x1                                1.3107      0.071      18.396      0.000      1.159      1.463
const                            2.832e+04  3102.448      9.127      0.000      2.17e+04  3.49e+04
=====
Omnibus:                          1.110      Durbin-Watson:          1.875
Prob(Omnibus):                     0.574      Jarque-Bera (JB):       0.993
Skew:                               -0.474      Prob(JB):               0.609
Kurtosis:                          2.291      Cond. No.                1.02e+05
=====

```

#### Warnings:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 1.02e+05. This might indicate that there are strong multicollinearity or other numerical problems.

#### Кореляція:

```
[[1.    0.9785]
 [0.9785 1.    ]]
```

```
Значення фактору для заданого прогнозу показника: [[67773.14371161]
[77416.62667164]]
```

Рисунок 2.6 – Лістинг розрахунку при моделюванні в пакеті statsmodels з використанням OLS

### Математичне забезпечення аналізу даних

За допомогою програмних інструментів можна реалізувати математичну модель, відношення між реальними елементами подаються у формі рівнянь, нерівностей, систем.

Для кожної задачі потрібно провести певні етапи. Узагальнено:

- Постановка проблеми та її якісний аналіз: формулюються сутність проблеми, припущення, визначаються питання, які потрібно розв'язати
- Побудова математичної моделі: вибір методів розв'язання, формалізація задачі. Зважаючи на постановку і цілі задачі, можуть бути використані методи диференціального та інтегрального числення, математичного програмування, ймовірнісні та статистичні і т.ін.
- Математичний аналіз моделі: за допомогою математичних прийомів та методів виявлення загальних властивостей
- Підготовка вхідних даних: нормалізація статистичних даних, усунення викидів і т. ін.
- Числове розв'язання: алгоритмізація, вибір методів і програмних інструментів, безпосередня реалізація і отримання розрахунків.
- Аналіз результатів: оцінка якості моделі за різними параметрами, коректності і повноти результатів, практична значимість.
- Апробація моделі: на нових даних перевірка моделі, визначення прогнозів або ін.

Основні задачі, що виникають при аналізі результатів роботи в закладі освіти, пов'язані з трендами тих чи інших показників, залежністю показників від певних факторів (регресійні моделі), визначенням інтегральних показників для задач різних рейтингів і взагалі з різноманітними задачами обробки статистичних даних.

Даними можуть бути або результати запитів, отриманих із інформаційної системи, або будь-які статистичні дані, що є цікавими для аналізу.

Тип моделі обирається за початковим розподілом даних. Дані обов'язково мають бути нормалізовані для отримання незміщених та ефективних оцінок параметрів, їх кількість також впливає на якість результатів.

В зазначених вище програмних інструментах використовуються класичні методи і алгоритми для регресійного аналізу і прогнозування показників, кластерного, факторного, інтегрального аналізу. Відповідні алгоритми можуть бути запрограмовані і додані як окремі модулі для обробки даних за необхідності.

### **2.3. Висновки до другого розділу**

Основою інформаційної системи є база даних, а також система управління базою даних. Структура інформаційної системи визначається інформацією та задачами, що мають розв'язуватися за допомогою організації даних. Інформаційна система містить у собі множину програмних рішень автоматизації процесів конкретної організації. Важливим є здатність до адаптації і подальшого розвитку.

Microsoft Access має переваги для створення СУБД в освітньому закладі. Microsoft Access має широкий спектр функцій, зв'язок із зовнішніми таблицями та базами даних, іншими додатками та є найбільш розповсюдженим і офіційним пакетом для роботи з різними типами документів. Перевагами є також вбудована мова програмування Visual Basic for Applications, розвинена система довідки, інтуїтивно зрозумілий інтерфейс, різноманітні діалогові інструменти.

Для аналізу та візуалізації інформації можуть бути використані інструменти програмного пакету MS Office, а також мови програмування і пакетів для роботи з даними, зокрема, Python і відповідні бібліотеки з великим функціоналом. В основі моделювання даних лежать класичні математичні методи і алгоритми для регресійного аналізу та прогнозування показників, кластерного, факторного, інтегрального аналізу. Відповідні алгоритми можуть бути запрограмовані і додані як окремі модулі для обробки даних за необхідності.

## РОЗДІЛ 3

### ПРОЕКТУВАННЯ І ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА АНАЛІЗУ ДАНИХ В ЗАКЛАДАХ ОСВІТИ

#### 3.1. Автоматизована системи відвідуваності та обробки інформації в закладах освіти

Автоматизовану систему відвідуваності та обробки інформації в закладах освіти було розроблено під час науково-дослідницької практики у 2023 р., що відображено у звіті з практики. За результатами допрацювань цієї системи було опубліковано статтю у співавторстві в українському науковому фаховому журналі «Інформаційні технології та комп'ютерна інженерія» [7].

#### Схеми проектування системи

Найпростіший варіант інформаційної системи обліку і аналізу відвідування навчальних занять (ІС) складається з одного комп'ютера, до якого через USB-порт приєднаний зчитувач магнітних карток, смарт-карт або QR-кодів.



Рисунок 3.1 – Найпростіша схема

Принцип роботи вказаної системи простий: інформація про учня, що прийшов до школи, зчитується сканером та пересилається на локальний комп'ютер (ЛК), де вона і зберігається. На цьому ж комп'ютері проводиться аналіз збереженої інформації та формуються необхідні звіти.

Але ця проста система має суттєві недоліки:

- доступ до інформації про відвідування занять може бути тільки з комп'ютера, де зберігаються дані;
- розташування комп'ютера, який є інструментом для аналізу даних по відвідуванню учнями школи, не найкраще – біля входних дверей,
- труднощі аналізу даних у випадку, якщо учні входять до школи через два-три входи, кожен з яких повинен реєструвати учнів, які зайшли до школи.

Вказані недоліки усуваються, якщо локальні комп'ютери, до яких приєднані сканери, включити до мережі закладу освіти.

В цьому випадку вся інформація зі сканерів через локальні комп'ютери (ЛК) може одразу передаватися на сервер даних (СД), який буде зберігати інформацію з декілька сканерів і який забезпечить доступ до інформації з робочих місць (РМ), де треба мати необхідні звіти.

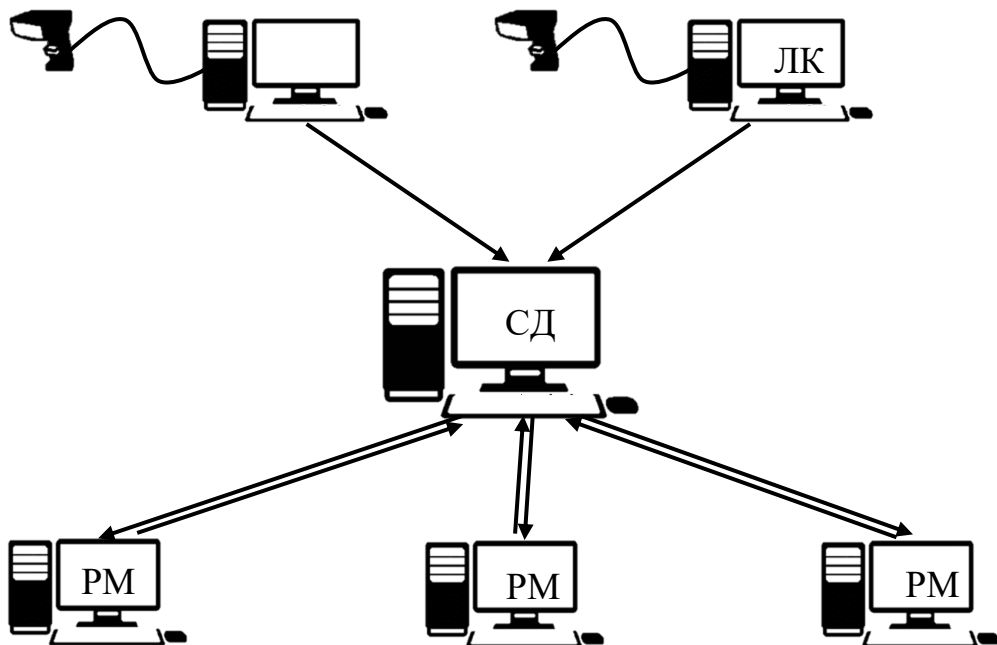


Рисунок 3.2 – Інформаційна система з сервером даних закладу

Але, якщо сервер працює не цілодобово, а включається на початку робочого дня, дана система має суттєвий недолік, який пов'язаний з тим, що більшість учнів приходять до академії до включення сервера даних, і тому їх приход зареєструвати в системі буде неможливо.

Якщо сервер даних розташувати у хмарному середовищі, то проблеми з часом роботи сервера існувати не буде.

Дані зі сканерів будуть одразу пересилатися у хмарне сховище і там зберігатися. Доступ до них буде постійним. Крім того, доступ до них можна організувати з будь-яких пристроїв: комп'ютерів, планшетів, мобільних телефонів.

Але, ця система має недолік: при поганому інтернет-з'єднанні будуть проблеми зі збереженням вхідної інформації на сервері, що може привести до повної непрацездатності програми збереження вхідної інформації.

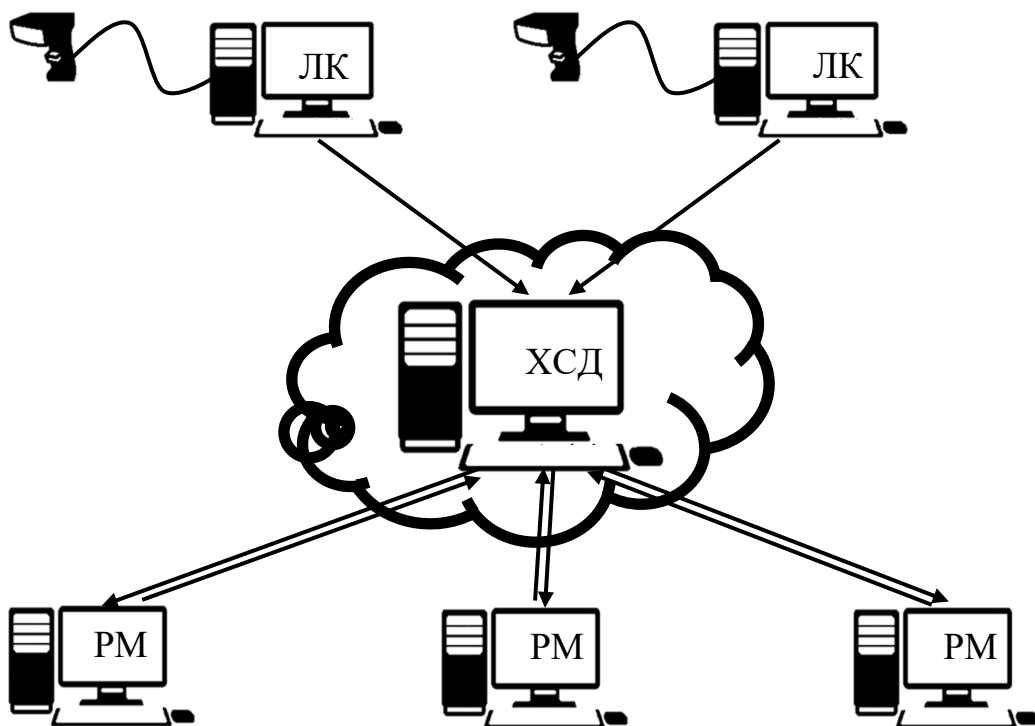


Рисунок 3.3 – Інформаційна система з хмарним сервером даних

Створена інформаційна система мінімізує проблеми збереження вхідних даних наступним чином:

- дані, які скануються, зберігаються в локальній базі даних на комп'ютері, до якого безпосередньо підключено сканер;
- в якості комп'ютера використовується ноутбук, який може працювати автономно не менше чотирьох годин;
- на одному з робочих комп'ютерів (або на локальному сервері) встановлюється програма, яка пересилає дані з локальної бази даних в базу даних, розташовану на сервері;
- всі робочі комп'ютери для обробки даних використовують дані сервера.

Вхідні дані при такому рішенні буде оброблено з деякою затримкою, але така схема вирішує проблему поганого інтернет-зв'язку та відсутність електропостачання.

#### Основні етапи створення програмного забезпечення

Для збереження інформації на комп'ютер, до якого приєднаний сканер, було розглянуто два незалежних підходи.

Було розглянуто два варіанти.

У першому варіанті використовувалась мова Python. Дані зберігалися у файлі формату .csv. Цей формат підтримується багатьма мовами програмування і дозволяє легко переносити дані в різні бази даних, включаючи Access та MySQL. Головна доцільність використання цього підходу в тому, що на вході замість відносно дорогих ноутбуків можна використовувати недорогі мікрокомп'ютери Raspberry PI або Arduino. В них використовується Linux і вони добре підтримують мову Python, а також формат файлів .csv.

Другий варіант використовує СУБД Access: сканована інформація одразу зберігається у базі даних Access, яка розташована на комп'ютері, до якого приєднаний сканер. Якщо в якості комп'ютера використовувати ноутбук, то система збору інформації буде повністю автономною. Вона не буде залежати від інтернету, від електрики. Це значно підвищує її надійність.

Для збереження даних про відвідуваність на локальному комп'ютері створюється база даних, яка містить всього три об'єкти:

- таблицю Signal, яка зберігає ID-коди учнів, які відвідали школу,
- таблицю User Local, яка містить відомості про учнів,
- форму для відображення інформації про учня, який на даний момент проходить реєстрацію (Рисунок 4):
  - прізвище учня,
  - час і дату приходу у школу.

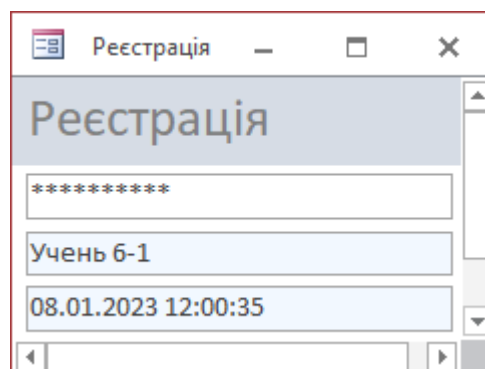
The image shows a screenshot of a software window titled "Реєстрація" (Registration). The window has a title bar with standard Windows controls (minimize, maximize, close). The main content area has a header "Реєстрація" and three input fields. The first field contains a masked string "\*\*\*\*\*". The second field contains the text "Учень 6-1". The third field contains the timestamp "08.01.2023 12:00:35". There are scroll bars on the right and bottom of the form area.

Рисунок 3.4 – Форма відображення інформації при реєстрації

Якщо сервер даних знаходиться у хмарному середовищі, то інформацію потрібно туди скопіювати з локальних баз даних, які знаходяться на локальних комп'ютерах. Це реалізується за допомогою запитів, які можуть бути розташовані на локальних комп'ютерах, або на одному з робочих комп'ютерів (РМ), які використовуються персоналом для аналізу даних про відвідування. У першому випадку запити передають тільки дані на локальних комп'ютерах.



У другому – дані спочатку збираються зі всіх локальних комп'ютерів, а потім передаються в базу даних, яка розташована у хмарному середовищі.

Якщо сервер даних знаходиться в приміщенні закладу, то можна не переносити дані з локальних комп'ютерів на сервер, на якому може бути розміщено тільки програми для створення необхідних звітів. При цьому програму спочатку збирає дані зі всіх локальних комп'ютерів, а потім обробляє їх разом, генеруючи необхідні звіти.

Краще, щоб всі дані зберігалися в одному місці. В цьому випадку перенесення даних на локальний сервер даних проводиться аналогічно переносу даних на сервер у хмарному середовищі.

У створеній інформаційній системі всі дані знаходяться на сервері даних закладу. Їх обробка може проводитися на комп'ютерах, які підключені до шкільної мережі і мають доступ до бази даних. Ці комп'ютери можуть бути розташовані на робочих місцях, де необхідно аналізувати інформацію про відвідування, наприклад, на комп'ютерах директора, завуча. Саме адміністрацію цікавить інформація з відвідування учнями закладу.

Окрім цього, програма генерує деякі спеціальні звіти, які використовуються для вирішення інших задач. Прикладом є звіт про загальну кількість дітей в школі, загальну кількість присутніх та відсутніх в абсолютних значеннях та відносних показниках, відсотках. Ця інформація може бути підставою для прийняття певних адміністративних рішень, наприклад, оголошення карантину в закладі, інформація для шкільної їдальні або інше.

#### Надсилання повідомлень батькам

Оповіщення батьків реалізується в інформаційній системі через бота, створеного за допомогою BotFather в месенджері Telegram.

Коли учень прийшов до школи і зареєструвався в системі, за його ідентифікаційним номером визначаються батьки. Їм надсилається

повідомлення про те, що їх дитина прийшла до школи і вказується дата та час її реєстрації в школі.

Для реалізації використано функцією месенджеру Telegram, що дозволяє створювати автоматичні боти для виконання завдань розсилки користувачам.

Боти – це спеціальні акаунти у Telegram, створені для автоматичної обробки та надсилання повідомлень. Користувачі можуть взаємодіяти з ботами за допомогою повідомлень, що надсилаються через звичайні або групові чати. Логіка боту контролюється за допомогою HTTPS запитів до API системи для роботів.

BotFather є офіційним інструментом для створення ботів та управління ними. За допомогою цього інструменту створюємо бота, даємо йому назву та нікнейм. Далі BotFather надає посилання на створений бот і токен для звернення до нього.

### Взаємодія з ботом за допомогою Python

Для взаємодії з ботом Telegram використано мову програмування Python. Додаток має модульну структуру. За відправку повідомлень відповідає клас telegram, який забезпечує взаємодію з платформою за допомогою API. Дані для надсилання повідомлень програма отримує з бази даних mysql, розташованої на віддаленому сервері. За виконання запитів на сервер відповідає бібліотека requests

```
import requests
```

Метод setChat класу telegram задає чат одержувача і зберігає його ID у властивість класу:

```
def setChat (self, chat_id):
    self.chat_id = chat_id
    pass
```

Метод `sendMessage` відповідає за відправлення повідомлень: формує посилання нашого бота з методом надсилання повідомлень і передає дані методом POST:

```
def sendMessage(self, text):
    if self.chat_id == 0:
        raise Exception('chat_enter_error')
        link='https://api.telegram.org/bot'+
        self.bot_id+':'+self.token+'/sendMessage'
    answer = requests.post(link, data = {
        'chat_id': self.chat_id,
        'text': text
    })
    if answer.status_code != 200:
        raise Exception('post_text_error')
    return answer
```

За взаємодію з базою даних відповідає клас `database`, який реалізовано за допомогою стандартної бібліотеки  `pymysql`  мови програмування Python:

```
import pymysql
```

Метод `connect` відповідає за з'єднання з базою даних SQL:

```
def connect(self, host, user, password, db):
    connection = pymysql.connect(
        host = host,
        user = user,
```

```

    password = password,
    db = db
)
print ('connect successful!!')
return connection

```

Метод `getRows` відповідає за отримання ID чатів одержувачів та текст, який потрібно їм відправити:

```

def getRows(self, connection):
    try:
        with connection.cursor() as cursor
            sql = 'SELECT `chat_id`, `text` FROM `signal`'
            cursor.execute(sql)
            rows = []
            for row in cursor:
                rows.append(row)
            return rows
    except:
        print('Database Connection Error')
    finally:
        print('end')

```

Сам код програми виглядає так:

```

from classes.database import database
from classes.telegram import telegram
db = database()
messenger = telegram()

```

```
connection = db.connect('localhost', 'root', 'root', 'signal')
rows = db.getRows(connection)
for row in rows:
    messenger.setChat(row[0])
    result = messenger.sendMessage(row[1])
print(result)
```

У ньому створюються об'єкти класів, викликаються методи, задаються потрібні параметри та виводиться результат.

### **3.2. Автоматизована інформаційна система обліку та аналізу результатів роботи викладачів**

Для створення ефективної інформаційної системи необхідна систематизація, структурування даних, можливість їх зручного введення, вилучення і аналізу для формування різного плану звітності, що допомагає в організації навчального процесу і прийнятті обґрунтованих управлінських рішень. Для оцінки професійного рівня педагогічних і керівних кадрів проводиться аналіз за різними показниками: освітою, педагогічним стажем, атестаціями і підвищенням кваліфікації, соціальними виплатами і стимулюванням. В інформаційній системі також має зберігатися особиста інформація по викладачам та їх показники навчально-методичної і наукової роботи і т. ін.

Частково роботу зі створення інформаційної системи було виконано під час переддипломної практики у 2023 р.

## Постановка задачі

Розробити програмне забезпечення для введення, зберігання, редагування та обробки інформації про вчителів та їх навчально-методичну діяльність, яка включає:

- загальну інформацію про вчителя,
- його навчально-методичну роботу,
- підвищення кваліфікації,
- атестацію вчителів.

Програмний комплекс повинен надавати можливість вносити та зберігати необхідні дані, які мають накопичуваний характер. Для цього потрібно створити необхідні форми, які дозволять опрацьовувати вхідну інформацію. Для обробки даних і формування звітів створити відповідні екранні форми, згенерувати необхідні запити, які будуть обробляти вхідну інформацію і розміщувати результат обробки в екранних формах для перегляду на екрані комп'ютері та у формі звітів для їх друку на принтері.

## Аналіз вимог

На першому етапі визначаються вимоги до застосування. Це включає визначення того, які дані необхідно зберігати, що потрібно мати в результаті обробки внесених даних.

Система "Вчитель" за вимогами повинна містити наступну інформацію:

### 1. загальна інформація про викладачів:

- логін та пароль вчителя;
- прізвище, ім'я, по-батькові;
- стать;
- номер телефону;
- домашня адреса;

- посада;
  - категорія;
  - звання;
  - тижневе навантаження;
  - дані про прийняття на роботу;
  - дані про звільнення;
  - загальний та педагогічний стаж на момент прийняття на роботу;
  - дату народження.
2. документи викладачів (паспорт, атестат та інше);
3. інформація про річне навантаження викладача:
- період (навчальний рік);
  - предмет;
  - годин на тиждень.
4. інформація про атестацію:
- період;
  - дати атестації;
  - категорія;
  - звання;
5. інформація про підвищення кваліфікації:
- період;
  - дати початку та закінчення підвищення;
  - організатор курсів;
  - кількість годин курсу;
  - документи по закінченню курсів;
6. інформація про участь в МАН:
- період;
  - рівень;
  - ПІБ учня;
  - клас учня;

- секція, відділ;
- назва роботи;
- місце;

7. інформація про статті:

- період;
- дата;
- прізвище учня-співавтора, клас;
- назва статті;
- видання;
- кількість сторінок;
- назва роботи;

8. інформація про конференції:

- період;
- назва конференції;
- дати проведення конференції;
- учень;
- клас;
- назва роботи;
- кількість сторінок;
- рівень конференції;

9. інформація про організацію культурно-масових заходів:

- період;
- дата проведення;
- назва заходу;

10. інформація про організацію спортивно-оздоровчих заходів:

- період;
- дата проведення;
- назва заходу.



Основні етапи створення програмного додатку з використанням баз даних

В якості програмного інструменту обрано MS Access. Створення програмного додатку для роботи з даними - це розробка програмного забезпечення, що використовує базу даних для зберігання та управління даними. Цей процес включає декілька етапів.

### Проектування бази даних

На етапі проектування бази даних необхідно визначити структуру фізичних таблиць, які будуть зберігати необхідну інформацію, виявити та створити зв'язки між таблицями.

Структура є дуже важливою у проектуванні. Неякісна структура бази даних може бути причиною неефективної роботи додатку або взагалі неможливості реалізації деяких вимог до системи в цілому.

Загалом, структура даних є визначаючим фактором успіху всієї програми. Е. Ф. Кодд довів, що, дотримуючись при створенні зв'язаних таблиць деяких правил, можна забезпечити простоту і ефективність маніпулювання даними.

Його методика отримала найменування нормалізації даних, суть яких полягає у наступному:

1. Поля таблиць повинні бути атомарними, тобто зберігати одну найменшу одиницю інформації.

Наприклад, якщо прізвище, ім'я та по-батькові робітника буде завжди використовуватися у повній формі, то достатньо одного поля для зберігання цієї інформації. Але, якщо поряд з цим потрібно використовувати окремо прізвище, або прізвище разом з ім'ям, або прізвище та ініціали, тоді ця інформація буде складатися з трьох одиниць інформації: прізвища окремо, ім'я

окремо, по-батькові окремо, для зберігання яких потрібно проектувати три поля.

2. Таблиця не повинна містити повторювану групу стовпців. Для збереження відповідної інформації створюється окрема таблиця.

3. В таблицю потрібно включати тільки ті поля, які відносяться тільки до предметної області таблиці. Поле, що представляє факт з іншої предметної області, має належати до іншої таблиці.

4. Якщо таблиця, яку назвемо головною, має поле, значення якого повторюються, то ці значення повинні зберігатися в окремій таблиці, яку у подальшому будемо називати довідником, а в полі головної таблиці потрібно організувати підстановку з цих значень з довідника. Крім того, таблиця-довідник має містити ті поля, які безпосередньо відносяться до об'єкту довідника

Якщо список значень поля заздалегідь відомий і є постійним, то джерелом підстановки у поле головної таблиці може бути фіксований список. У цьому випадку створювати окрему таблицю-довідник не має потреби.

5. Якщо будь-яка інформація може бути отримана за допомогою іншої інформації, то її зберігати не слід.

6. Кожна інформація має заноситися тільки один раз.

#### Створення таблиць. Схема даних

Створення таблиці здійснюється за допомогою конструктора таблиць, який дозволяє створити таблицю будь-якої складності.

При створенні таблиці потрібно:

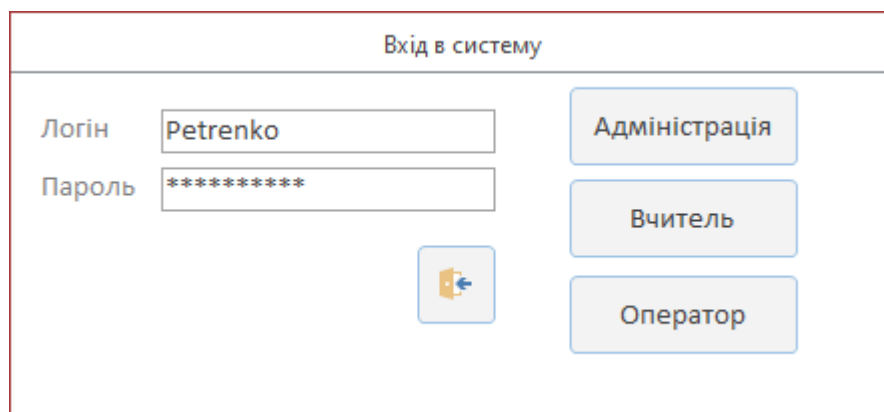
- вказати необхідні поля, яка повинна мати таблиця;
- встановити тип даних для полів таблиці;
- вказати необхідні властивості кожного з полів;



форматах: в форматі текстового редактора Word, табличного редактора Excel, редактора презентацій PowerPoint.

### Основні складові графічного інтерфейсу

При запуску програми першим з'являється вікно ідентифікації користувача:



The screenshot shows a login window titled "Вхід в систему". It features two input fields: "Логін" (Login) containing the text "Petrenko" and "Пароль" (Password) containing masked characters "\*\*\*\*\*". To the right of the password field is a blue button with a left-pointing arrow. Further to the right are three stacked buttons: "Адміністрація" (Administration), "Вчитель" (Teacher), and "Оператор" (Operator).

Рисунок 3.6 – Вхід в систему

Логін, пароль, а також права (Адміністратора, вчителя або оператора) задаються при реєстрації вчителя. Надалі ці права можна змінити.

Оператори мають права на реєстрацію користувачів, внесення деяких даних, а також на обслуговування системи.

Користувач с правами вчителя може вносити та переглядати особисті дані. Дані інших вчителів, а також загальні звіти для них не є доступними. Адміністрація має право на перегляд інформації про будь-якого вчителя. Але дані, які безпосередньо відносяться до вчителів і які вносяться самими вчителями, адміністрація може тільки переглядати без права редагування.

Основною формою є картка вчителя, і вона призначена для введення інформації про вчителів. Вона дозволяє вносити, змінювати та видаляти, всі дані, які відносяться до вчителів та їх діяльності.

Картка вчителя

### Особиста картка вчителя

працючі
  звільнені
  всі

admin
  оператор
  вчитель
 Телефон: +38(056)111-22-33

логін: 1      посада: вчитель інформатики  
 пароль: \*      статус: постійно

прізвище: Петренко      дата прийому: 01.09.2009  
 ім'я: Тетяна      дата звільнення:

по-батькові: Степанівна      причина звільнення:

стать: ж      на пенсії з:

Дані вчителя | Атестація | Навантаження | Курси | МАН | Статті | Конференції | КМЗ | Спорт | Змінити пароль

**1**

**2**

**3**

Документи:

Назва документа	
диплом	0(0)
паспорт	0(0)
*	0(0)

Запись: 1 из 2      Нет фильтра      Поиск

Рисунок 3.7 – Екранна форма «Особиста картка вчителя»

Візуально екран має три зони.

Зона 1 містить список вчителів, завдяки якому можна знайти необхідного вчителя швидко.

В зоні 2 розміщена така інформація:

- фільтр, який дозволяє обрати один з варіантів висвітлення інформації тільки по працюючим вчителям, тільки по звільненим або по всім вчителям (і працюючим на поточний час і звільненим)
- фотографія вчителя та основні відомості про вчителя (пропорції 4x3),
- основна інформація про вчителя:
  - ✓ логін, пароль,
  - ✓ прізвище, ім'я, по батькові,

- ✓ стать,
- ✓ посада,
- ✓ статус,
- ✓ дата прийому,
- ✓ дата і причина звільнення,
- ✓ дата виходу на пенсію (для пенсіонерів)

Зона 3 являє собою систему вкладок, які відображують:

- вкладка "Дані вчителя":
  - ✓ освіту вчителя,
  - ✓ навчальний заклад, який закінчував вчитель,
  - ✓ дату народження,
  - ✓ адресу проживання,
  - ✓ загальний та педагогічний стаж на вказану дату,
  - ✓ необхідні документи: диплом, паспорт та інше...

Для прикладу декілька вкладок показано на рисунках.

- вкладка "Атестація":
  - ✓ період,
  - ✓ дата,
  - ✓ категорія,
  - ✓ звання,
  - ✓ розряд,
  - ✓ виконання.

Період	Дата	Категорія	Кат ***	Звання	Звання ***	Розряд	Роз ***	Вик.
2021-2022	01.09.2021	Ща	new	вчитель-методист	new	12	old	<input type="checkbox"/>
*								<input type="checkbox"/>

Рисунок 3.8 – Вкладка "Атестація"

- вкладка "Навантаження":
  - ✓ період,
  - ✓ предмет,
  - ✓ кількість годин на тиждень в указаному періоді,

Дані вчителя	Атестація	Навантаження	Курси	МАН	Статті	Конференції	КМЗ	Спорт	Змінити пароль
		Період	Предмет	Годин					
		2021-2022	хімія	18					
		*		0					

Рисунок 3.9 – Вкладка "Навантаження"

- вкладка "Курси":
  - ✓ період, за який будуть зараховуватися курси підвищення кваліфікації,
  - ✓ номер і дата свідоцтва про підвищення кваліфікації,
  - ✓ місце проходження,
  - ✓ кількість годин,
  - ✓ скановані документи про проходження курсів,

Дані вчителя	Атестація	Навантаження	Курси	МАН	Статті	Конференції	КМЗ	Спорт	Змінити пароль	
		Period	№ дс	від	з	по	Місце проходження	год.	Заг	📎
		2021-2022	345899	09.02.2024	05.02.2024	09.02.2024	ДДТУ	30	<input checked="" type="checkbox"/>	📎(1)

Рисунок 3.10 – Вкладка "Курси"

- вкладка "МАН":
  - ✓ період,
  - ✓ номер і дата свідоцтва про підвищення кваліфікації,
  - ✓ місце проходження,
  - ✓ кількість годин,

✓ скановані документи.

Дані вчителя		Атестація	Навантаження	Курси	МАН	Статті	Конференції	КМЗ	Спорт	Змінити пароль
Період	Рівень	Учень	кл	Секція	Відділ	Тема роботи		Місце	В	
2021-202	міський	Іванов	9	1. Комп'ютерні системи	07. Комп'юте	Створення автоматизованої с		1		
*										

Рисунок 3.11 – Вкладка "Курси"

– вкладка "Статті":

- ✓ період,
- ✓ дата видання статті,
- ✓ назва статті,
- ✓ назва видання,
- ✓ кількість сторінок

Дані вчителя		Атестація	Навантаження	Курси	МАН	Статті	Конференції	КМЗ	Спорт	Змінити пароль
Період	Дата	Стаття		Видання		Стор.	Ви	Заг		
						0	<input type="checkbox"/>	<input type="checkbox"/>		
*						0	<input type="checkbox"/>	<input type="checkbox"/>		

Рисунок 3.12 – Вкладка "Курси"

– вкладка "Конференції":

- ✓ період,
- ✓ прізвище учня (якщо треба),
- ✓ клас, в якому вчиться учень,
- ✓ назва конференції,
- ✓ початок та закінчення конференції,
- ✓ назва роботи на конференцію,



Дані вчителя		Атестація		Навантаження		Курси		МАН		Статті		Конференції		КМЗ		Спорт		Змінити пароль	
Період	Учень	Кл	Конференція	Початок	Закінчення	Нзва роботи		Сто	Зар										
2021-2022	Іанченко	10а	Конференція 1	06.01.2022	06.01.2022	Тут буде тема виступу		0	<input checked="" type="checkbox"/>										
*								0	<input type="checkbox"/>										

Рисунок 3.13 – Вкладка "Конференції"

- вкладка "КМЗ" (організація культурно-масових заходів):
  - ✓ період,
  - ✓ дата заходу,
  - ✓ назва заходу (події)

Дані вчителя		Атестація		Навантаження		Курси		МАН		Статті		Конференції		КМЗ		Спорт		Змінити пароль	
Період	Дата	Назва події		Вик.	Зар														
*				<input type="checkbox"/>	<input type="checkbox"/>														

Рисунок 3.14 – Вкладка "КМЗ"

- вкладка "спорт" (організація спортивно-масових заходів):
  - ✓ період,
  - ✓ дата заходу,
  - ✓ назва події,

Дані вчителя		Атестація		Навантаження		Курси		МАН		Статті		Конференції		КМЗ		Спорт		Змінити пароль	
Період	Дата	Назва події		Вик.	Зар.														
*				<input type="checkbox"/>	<input type="checkbox"/>														

Рисунок 3.15 – Вкладка "Спорт"

- вкладка "Змінити пароль" (зміна особистого паролю входу):
  - ✓ поточний пароль,
  - ✓ новий пароль,

- ✓ новий пароль (підтвердження),
- ✓ кнопка "Змінити" для зміни паролю

Рисунок 3.16 – Вкладка "Змінити пароль"

### Екранні форми для довідникової інформації

Порядок створення екранних форм особливого значення не має, але рекомендується спочатку створити форми внесення даних в довідники, після чого можна ввести деякі дані, необхідні для тестування основних форм.

Якщо розглядати прості форми, то для довідників вони найчастіше мають форми "в один стовпчик" або "стрічковий". Якщо полів таблиці відносно не багато і вони без прокрутки розміщуються по ширині екрану, обирається, як правило тип "стрічковий".

Спочатку форму для довідникової інформації створюємо за допомогою майстра форми, після чого редагуємо її у конструкторі. Типову довідникову форму показано на Рисунку 3.17.

Типова довідникова форма має поля для внесення даних довідника, а також кнопки управління записами (зліва-направо):

- перехід на перший запис,
- перехід на останній запис,
- пошук необхідної інформації,
- додавання нового запису,

- видалення нового запису,
- вихід з форми довідника.

Довідник предметів

Предмети

біологія
інформатика
фізкультура
хімія

Navigation icons: back, forward, search, add, delete.

Рисунок 3.17 – Довідник предметів

Разом з однотобличними довідниками у роботі використовуються довідники, які засновані на двох таблицях.

Наприклад, довідник секцій МАН, які згруповані по секціям. Форма для введення даних для довідника секцій має вигляд, показаний на Рисунок 3.18.

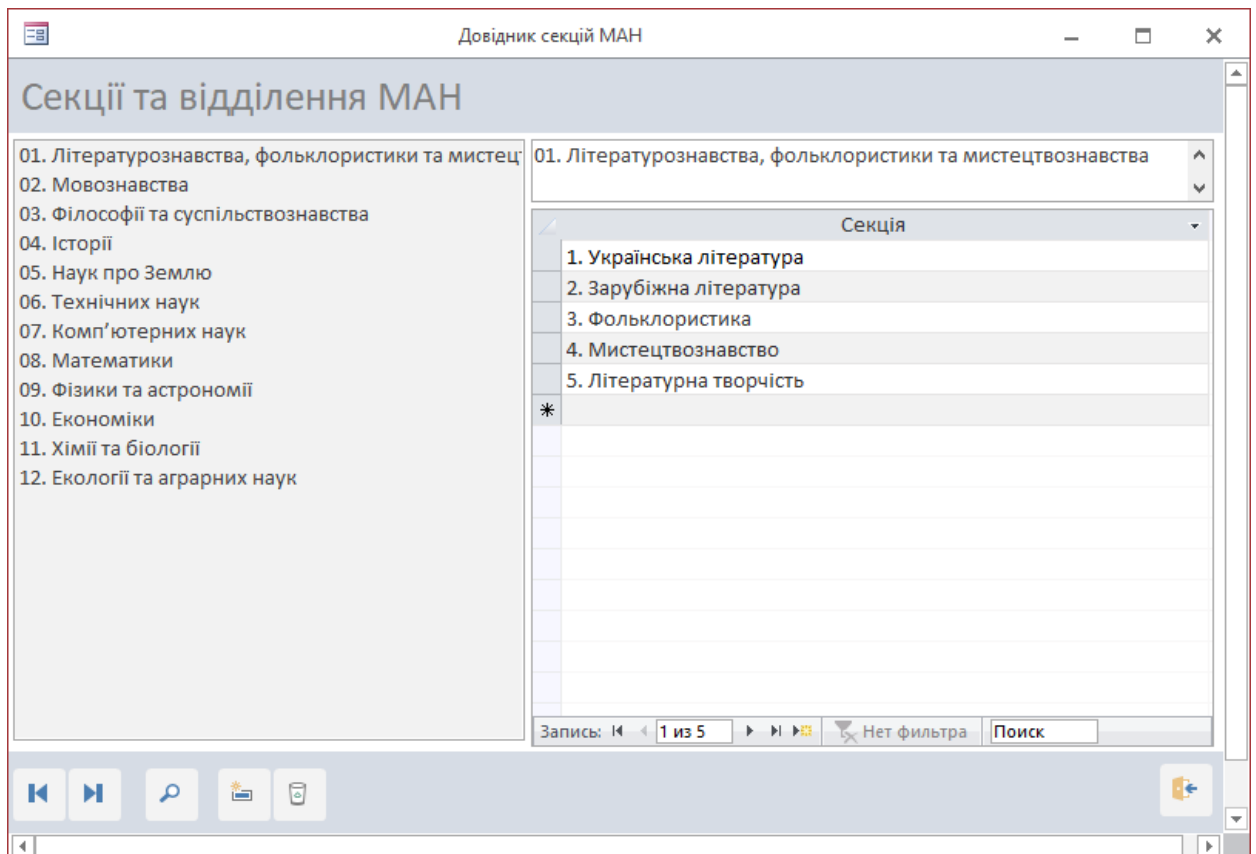


Рисунок 3.18 – Довідник секцій МАН

### Аналіз даних

Метою аналізу даних є отримання необхідних для школи звітів. Обробка даних проводиться з використанням SQL-запитів. Кінцевий результат запиту – це робота не одного запиту, а декількох послідовних SQL-команд, остання з яких, як правило, і надає кінцевий результат.

В програмі практично всі запити формуються за встановлений період. Період задається користувачем і може бути як одним днем, так і декілька років: задається початкова та кінцева дата для запитів.

При цьому результати запитів можуть бути:

- виведені на екран;
- роздруковані;

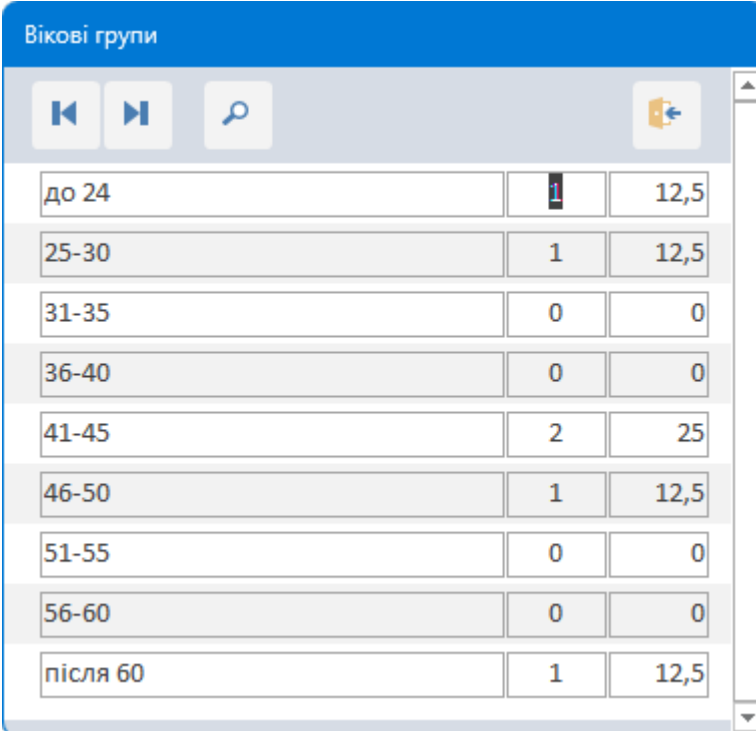
– експортовані в таблиці Excel.

Якщо звіт про аналіз даних виводиться на екран, то спочатку будується екрана форма, за допомогою якої ці результати виводяться на екран користувача.

За необхідності створюються декілька екранних форм.

Наведемо декілька прикладів:

1. Звіт по віковим групам вчителів:



The screenshot shows a software window titled "Вікові групи" (Age Groups). It contains a table with three columns: Age Group, a count, and a percentage. The table data is as follows:

Вік	Кількість	Відсоток
до 24	1	12,5
25-30	1	12,5
31-35	0	0
36-40	0	0
41-45	2	25
46-50	1	12,5
51-55	0	0
56-60	0	0
після 60	1	12,5

Рисунок 3.19 – Вікові групи

2. Звіт по обчисленню загального та педагогічного стажу вчителів на задану дату:

Вчителі. Стаж на 08.09.2023				на день прийому						на день звіту								
				загальний стаж			пед. стаж			стаж у школі			загальний стаж			педстаж		
				рр	мм	дд	рр	мм	дд	рр	мм	дд	рр	мм	дд	рр	мм	дд
				2	3	5	7	9	12	14	16	18	20	22	25	27	29	31
▶	2	Іванченко П.С.	30.03.2020	49	4	1	4	4	1	3	5	9	52	9	10	7	9	10
	3	Крутий П.І.	30.11.2012	12	0	0	5	6	2	10	9	8	22	9	8	16	3	10
	5	Карпенко Т.І.	26.11.2018	0	0	0	0	0	0	4	9	12	4	9	12	4	9	12
	7	Сидоров І.П.	16.04.2019	7	10	4	7	10	4	4	4	22	12	2	26	12	2	26
	1	Петренко Т.С.	01.09.2009	6	8	10	3	2	15	14	0	7	20	8	17	17	2	22
	8	Петренко О.І.	13.04.2000	0	0	0	0	0	0	23	4	25	23	4	25	23	4	25
	9	Буряк О.П.	08.08.2022	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0

Рисунок 3.20 – Стаж вчителів на задану дату

Результати обробки даних можна друкувати. Результати форми друкувати не дуже зручно, тому краще спочатку створити окремий об'єкт для друку результатів – звіт. Перед друком звіт можна спочатку переглянути на екрані, а потім роздрукувати.

Зручним варіантом є виведення результатів запиту в таблицю Excel. В цьому випадку кінцевий користувач може самостійно змінювати вигляд форми звіту, а також проводити подальшу обробку результатів.

### Тестування, розгортання і підтримка програми

Тестування функціональності, продуктивності і безпеки програми є обов'язковим. Тестування програми двох-етапне:

- на першому етапі програму тестує розробник,
- на другому етапі користувачі тестують всі можливості програми; простий інтерфейс програми дозволяє одразу працювати з програмою.

На цьому етапі необхідно розгорнути додаток у середовищі її використання, тобто у закладі. Це включає встановлення програми на сервері і налаштування його для використання.

На етапі тестування програма може бути розташована на одному комп'ютері, після чого бази даних розміщуються на сервері, а робочі місця для виконання конкретних дій – на локальних комп'ютерах. У подальшому необхідно підтримувати програму після її розгортання. Це включає виправлення помилок, додавання нових функцій і забезпечення безпеки програми.

### **3.3. Висновки до третього розділу**

Інформаційна система обліку (п.3.1.) дозволяє автоматизувати збір інформації про відвідування навчальних занять учнями закладу освіти та аналізувати її з метою отримання необхідних звітів. Створена система має простий інтерфейс і є простою у використанні. Результат роботи представлено у вигляді файлу БД Access, який містить структуру спроектованих таблиць, схему даних зі зв'язками між таблицями, екранні форми для занесення вхідних даних та екранні форми для перегляду результатів їх обробки, запити для аналізу даних, звіти. В систему інтегровано програму, створену на Python, для надсилання повідомлень батькам про прибуття учня в навчальний заклад.

Інформаційна системи обліку і аналізу результатів роботи викладачів (п.3.2) надає можливість вносити, зберігати, вилучати і обирати для аналізу необхідні загальні дані про вчителя, його навчально-методичну роботу, підвищення кваліфікації, атестацію, участь в конференціях, МАН та інших заходах. Спроектовано і реалізовано базу даних за принципом нормалізації даних, що забезпечує їх цілісність і захищеність. Створено необхідні екранні форми для опрацювання інформації і формування звітів. Згенеровано необхідні запити, які дозволяють обробити вхідну інформацію і розмістити результат обробки в екранних формах для перегляду на екрані комп'ютера та у формі звітів для їх друку на принтері, або передати дані в файли Excel для подальшого аналізу. Необхідними є підтримка і супроводження програми.

## ВИСНОВКИ

Тема кваліфікаційної роботи є актуальною внаслідок відсутності інформаційних систем в закладах освіти, що дозволяють автоматизувати роботу, що пов'язана зі зберіганням, обробкою, вибором інформації, її аналізом та створенням необхідної документації. Сучасні інноваційні технології дозволяють автоматизувати багато процесів в закладі освіти та об'єднати управління в одну інформаційну систему. Це сприяє підвищенню ефективності моніторингу і аналізу даних, а також управління закладом в цілому.

Для реалізації завдань кваліфікаційної роботи було обрано Microsoft Access. Microsoft Access має широкий спектр функцій та зв'язків з іншими додатками MS Office, і є найбільш розповсюдженим і офіційним пакетом для роботи з різними типами документів в закладах освіти. Для аналізу та візуалізації інформації можуть бути використані інструменти програмного пакету MS Office, а також мови програмування і пакетів для роботи з даними, зокрема, Python і відповідні бібліотеки з великим функціоналом.

Розроблено і впроваджено інформаційну систему обліку про відвідування навчальних занять учнями закладу освіти, що дозволяє автоматизувати збір інформації та аналізувати її з метою отримання необхідних звітів. В систему інтегровано програму, створену з використанням Python, для надсилання повідомлень батькам про прибуття учня в навчальний заклад.

Спроектовано і реалізовано інформаційну систему обліку та аналізу результатів роботи викладачів, яка надає можливість вносити та зберігати необхідні дані про загальну інформацію про вчителя, його навчально-методичну роботу, підвищення кваліфікації, атестацію, участь в конференціях, МАН та інших заходах.

Завдання кваліфікаційної роботи виконано.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розвиток інформаційних систем управління освітою як інструмент реалізації державної освітньої політики: монографія / за ред. С. Л. Лондара; ДНУ «Інститут освітньої аналітики». Київ, 2020. 258 с.
2. Цифровізація освіти. URL: <https://mon.gov.ua/ua/news/cifrovizaciya-osviti-pershu-osvitnyu-informacijnu-sistemu-pidyednali-do-aikom> (дата звернення 4.01.2024).
3. Приєднання першої освітньої системи України до АІКОМ. URL: <https://aikom.iea.gov.ua/site/about> (дата звернення 4.01.2024).
4. Бородкіна І.Л., Бородкін Г.О. Інженерія програмного забезпечення. Київ, 2018 . 251 с.
5. В.Яковенко, Ю.Ульяновська, Т.Яковенко, і Т.Чупілко, «Адаптація принципів AGILE методології для правління проєктом розробки програмного застосунку», ІТКІ, вип.52, вип.3, с. 44-52, Груд 2021. DOI: <https://doi.org/10.31649/1999-9941-2021-52-3-44-52>
6. Чупілко Т.А. Моделювання і обробка структурованих даних із застосуванням сучасних комп'ютерних технологій: Дипломна робота магістра. Дніпро, УМСФ. 2020. 76 с.
7. Т. Чупілко, О. Чупілко, і М. Мормуль, «Проектування і програмна реалізація автоматизованої системи відвідуваності та аналізу даних в закладах освіти», ІТКІ, вип. 56, вип. 1, с. 35–43, Бер 2023. DOI: <https://doi.org/10.31649/1999-9941-2023-56-1>
8. Christian Hill, Learning Scientific Programming with Python. Second Edition. Cambridge University Press, 2020, 646 p.
9. William McKinney. Python for Data Analysis. Data Wrangling with Pandas, NumPy, and IPython. Second Edition. O'REILLY, 2018. 365 p.

10. Т.Чупілко, Ю.Ульяновська, М. Мормуль і А.Лагода, "Python для обробки даних і моделювання фінансово-економічних показників", ІТКІ, т. 51, вип. 2, с. 68-77, 2021. DOI: <http://doi.org/10.31649/1999-9941-2021-51-2-68-77>
11. Лавріщева К.М. Програмна інженерія. К. , 2008. 319 с.
12. Бардус І.О., Лазарєв М.І., Ніценко А.О. Бази даних у схемах (на основі фундаменталізованого підходу): навч. посіб. Харків: Вид-во «Діса плюс» Б, 2017. 133 с.
13. Завадський І.О. Основи баз даних: Навч. посіб. К. , 2011. 192 с.
14. Чупілко О.С. Інноваційні технології в автоматизованій системі моніторингу відвідуваності та аналізу інформації // Міжнародна науково-практична конференція «Економіко-правові та управлінсько-технологічні вимоги сьогодення: молодіжний погляд». УМСФ, Дніпро, 03 листопада 2023 р. С.310-312.
15. Яковенко А.В. Основи програмування. Python. Частина 1. Київ: КПІ, 2018. 195 с.
16. William McKinney Python for Data Analysis. Data Wrangling with Pandas, NumPy, and IPython. Second Edition. – O'REILLY. – 2018. – 365 p.