Форма № 17

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Запорізький національний технічний університет

Кафедра програмних засобів
(повна назва кафедри)

КУРСОВИЙ ПРОЕКТ
(РОБОТА)

з дисципліни «Бази даних»
(назва дисципліни)

на тему: «Автоматизована система обліку діяльності автошколи»

Студента 2 курсу КНТ-137 групи
напряму підготовки Комп'ютерні науки
спеціальності Інженерія програмного забезпечення
Козлова В.В.
(прізвище та ініціали)

Керівник доцент, к.т.н., Каплієнко Т.І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____
Кількість балів: _____ Оцінка: ECTS _____

Члени комісії: _____ Каплієнко Т.І.
(підпис) (прізвище та ініціали)

_____ _____
(підпис) (прізвище та ініціали)

_____ _____
(підпис) (прізвище та ініціали)

м. Запоріжжя

2019 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Запорізький національний технічний університет
(повне найменування вищого навчального закладу)

Інститут, факультет інститут інформатики та радіоелектроніки, факультет комп'ютерних наук і технологій

Кафедра програмних засобів

Ступінь вищої освіти (освітній ступінь) бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Курс 2 Група КНТ-137 Семестр IV

ЗАВДАННЯ

на курсовий проект студентові

Козлову Валерію Валентиновичу
(прізвище, ім'я, по батькові)

1. Тема проекту: Автоматизована система обліку діяльності автошколи

2. Термін здачі студентами закінченого проекту 14 червня 2019 року

3. Початкові дані до проекту звітні відомості про діяльність автошколи

4. Зміст розрахунково пояснювальної записки:

1 Вибір та обґрунтування проектних рішень

2 Концептуальне проектування предметної області

3 Логічне проектування предметної області

4 Розробка інтерфейсу користувача

5 Методика роботи користувача з системою

Висновки

Перелік посилань

Додаток А Тексти SQL-запитів

Додаток В Зразки звітів

5. Дата видачі завдання: 20 лютого 2019 року

КАЛЕНДАРНИЙ ПЛАН

№	Назви етапів курсового проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1.	Отримання та аналіз індивідуального завдання	1 - 2 тиждень	
2.	Аналіз програмних засобів, що будуть використовуватись в роботі	3 - 4 тиждень	
3.	Розробка логічної моделі системи. Заповнення даними	4 - 5 тиждень	
4.	Вивчення можливостей програмної реалізації інтерфейсу користувача	5 - 6 тиждень	
5.	Проектування додатків системи (звіти, макроси, запити)	6 - 8 тиждень	
6.	Оформлення відповідних пунктів пояснювальної записки	9 тиждень	Розділи 1 - 5 ПЗ
7.	Захист курсової роботи	10 тиждень	

Студент _____ _____Козлов В.В._____

Керівник _____ _____Каплієнко Т.І._____

«_____» _____ 2019 р.

РЕФЕРАТ

ПЗ: 53 с., 47 рис., 11 табл., 2 додатки, 9 посилань.

Об’єкт проектування – автоматизована система обліку діяльності автошколи.

Область дослідження – структура та діяльність автошколи.

Мета курсового проекту – автоматизована система обліку діяльності автошколи, що працює під управлінням системи керування базою даних Microsoft Access 2016 та новіше.

Розроблена система дозволяє вирішувати такі задачі:

- облік діяльності автошколи;
- автоматичний підрахунок вартості практичних занять;
- генерація звітної документації;
- підведення підсумків про навчальний процес;
- ведення довідкової підсистеми.

Автоматизовану систему було розроблено з використанням операційної системи Windows 10 та системи керування базою даних Microsoft Access 2016.

Автоматизована система працює на операційній системі Windows 10 чи Windows 7 з встановленою програмою Microsoft(R) Access 2016 та новіше.

БАЗА ДАНИХ, СИСТЕМА КЕРУВАННЯ БАЗОЮ ДАНИХ, ІНТЕРФЕЙС, ФОРМА, ЗАПИТ, ЗВІТ, МАКРОС, АВТОШКОЛА, НАВЧАННЯ, ВОДІЙСЬКЕ ПОСВІДЧЕННЯ.

ЗМІСТ

Завдання 2
Календарний план 3
Реферат 4
Зміст 5
Перелік умовних позначень і скорочень 7
Вступ 8
1 Вибір і обґрунтування проектних рішень 9
1.1 Підстава для розробки 9
1.2 Призначення розробки 9
1.3 Основні вимоги до системи, що розробляється 9
1.3.1 Вимоги до функціональних характеристик 9
1.3.2 Вимоги до надійності 10
1.3.3 Умови експлуатації 10
1.3.4 Вимоги до складу і параметрів технічних засобів 10
1.3.5 Вимоги до маркування й пакування 10
1.3.6 Вимоги до транспортування та збереження 11
1.3.7 Вимоги до програмної документації 11
1.4 Характеристика об’єкта проектування 11
1.4.1 Вхідні та вихідні дані 11
1.5 Структура інформації 12
1.6 Обґрунтування структури системи, що проектується 12
2 Концептуальне проектування предметної області 13
2.1 Аналіз концептуальних вимог та інформаційних потреб 13
2.1.1 Вимоги користувачів до бази даних 13
2.1.2 Виявлення існуючих задач по обробці інформації 13
2.1.3 Результати аналізу 13
2.2 Виявлення інформаційних об’єктів і зв’язків між ними 14
2.3 Концептуальна схема бази даних 17
3 Логічне проектування 18
3.1 Вибір і обґрунтування інструментарію проектування 18
3.2 Розробка схеми бази даних 18

3.3 Нормалізація схеми бази даних 22
3.4 Проектування запитів 23
4 Інтерфейс користувача системи 25
4.1 Проектування форм..... 25
4.2 Проектування звітів 36
4.3 Проектування макросів..... 37
5 Методика роботи користувача з системою 40
Висновки 44
Перелік посилань..... 45
Додаток А Тексти SQL-запитів..... 46
Додаток В Зразки звітів 51

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

АІС – автоматизована інформаційна система;

ОС – операційна система;

БД – база даних;

ТЗ – технічне завдання;

СКБД – система керування базою даних;

НФ – нормальна форма;

1НФ – перша нормальна форма;

2НФ – друга нормальна форма;

3НФ – третя нормальна форма;

НФБК – нормальна форма Бойса-Кодда;

4НФ – четверта нормальна форма;

SQL – Structured Query Language;

VBA – Visual Basic for Applications.

ВСТУП

Двадцять перше століття – це ера бурхливого розвиту інформаційних технологій і програмних засобів, що застосовуються в різних сферах діяльності людини. Сучасні інформаційні технології дозволяють організовувати роботу підприємства з максимальною ефективністю, а питання комплексної автоматизації рутинної роботи стає все більш актуальним.

З кожним роком конкуренція між автошколами збільшується, а тому наявність більш сучасних та ефективних підходів до організації діяльності автошколи та навчального процесу, не тільки покращує якість навчання, а й сприяє зменшенню вартості послуг, а отже і виділяє автошколу серед конкурентів.

Саме через це існує великий спрос серед комерційних фірм на якісне програмне забезпечення.

Актуальність роботи полягає в розробці такого підходу до організації діяльності підприємства, за якого мінімізується діяльність персоналу, що сприяє підвищенню ефективності праці та безпеки системи.

Мета роботи – розробка автоматизованої інформаційної системи, призначеної для автоматизації ведення обліку діяльності автошколи.

1 ВИБІР І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

Назва проекту: «Автошкола УченьОК».

В цьому розділі на підставі аналізу технічного завдання були визначені основні вимоги до системи та її функціональності.

1.1 Підстава для розробки

Підставою для розробки є завдання для курсового проекту на тему «Автоматизована система обліку діяльності автошколи».

1.2 Призначення розробки

Автоматизована інформаційна система (АІС) призначена для одержання, обробки та збереження інформації про учнів, викладачів, формування груп, проведення теоретичних та практичних занять, оформлення відповідної звітної документації.

1.3 Основні вимоги до системи, що розробляється

1.3.1 Вимоги до функціональних характеристик

Початковими даними є таблиці «Учень», «Свідоцтво», «Викладач», «Група», «Теоретичне_навчання», «Практичне_навчання».

Вихідними даними програми є звіти та результати виконання запитів.

При введенні даних програма повинна попередньо перевіряти правильність їх введення, щоб уникнути некоректної інтерпретації даних.

Система повинна мати зручний та інтуїтивно зрозумілий інтерфейс користувача, що забезпечує просте й ефективне спілкування користувача з програмним продуктом.

1.3.2 Вимоги до надійності

АІС має розпізнавати ситуації нерозуміння системою користувача (користувачем системи), а також мати обробку збійних ситуацій, для чого слід передбачити повідомлення про некоректний ввод даних, підказки для користувача (наприклад, маска введення номеру телефону).

1.3.3 Умови експлуатації

Програма може бути записана на пристрій зберігання інформації: магнітний чи твердотільний накопичувач.

Експлуатація програмного продукту здійснюється відповідно до експлуатаційної документації на розроблений програмний продукт, що відповідає стандартам і містить інформацію, необхідну для його освоєння та експлуатації.

1.3.4 Вимоги до складу і параметрів технічних засобів

Як апаратно-технічні засоби для експлуатації системи повинні використовуватися IBM-сумісний комп’ютер з характеристиками не нижче: процесор Intel Pentium Dual-Core E6800 2.5 ГГц, ОЗП 4 Гб, відеокарта з підтримкою DirectX 9 з WDDM драйвером, жорсткий диск об’ємом 32 Гб. На комп’ютері повинна бути встановлена операційна система (ОС) Windows 10, та Microsoft Access 2010 (чи новіше) [1].

1.3.5 Вимоги до маркування й пакування

Программа може поставлятися на диску чи на флеш накопичувачі. На упакуванні має бути зазначена назва програмного продукту – «Автошкола УченьОК», та логотип (рис. 1.1).

Рисунок 1.1 – Логотип програмного продукту «Автошкола УченьОК»

1.3.6 Вимоги до транспортування та збереження

Вимоги до транспортування та збереження аналогічні тим, що висуваються до накопичувачів на яких зберігається програмний продукт.

1.3.7 Вимоги до програмної документації

Програмне забезпечення повинне поставлятися з «Інструкцією користувача».

1.4 Характеристика об’єкта проектування

1.4.1 Вхідні та вихідні дані

Вхідними даними до системи є така інформація:

- інформація про учнів;
- інформація про викладачів;
- інформація про групи;
- інформація про навчальний процес.

Вихідними даними для системи є така інформація:

- ергономічний, інтуїтивно зрозумілий інтерфейс для відображення всієї інформації, що зберігається у АІС;
- звіти щодо діяльності автошколи.

1.5 Структура інформації

Оперативною інформацією в базі даних (БД) є інформація про практичне та теоретичне навчання.

Довідковою інформацією в БД є інформація про викладачів, учнів, отримані свідоцтва, групи.

Архівна інформація, на відміну від довідкової, досить не часто використовується. Саме тому архівна інформація в системі відсутня. При потребі можливе доопрацювання системи та перенесення старих записів з таблиць до архівних таблиць.

1.6 Обґрунтування структури системи, що проектується

Готове програмне забезпечення представляє собою один файл з розширенням accdb, що сприяє легкому розповсюдженню програми та підвищує безпеку системи.

У системі представлені: таблиці, запити, форми, звіти, макроси.

Таблиці призначені для збереження інформації. Запити дозволяють обирати серед збереженої інформації необхідну. Форми – графічно відображають дані та дозволяють користувачу взаємодіяти з системою. Макроси автоматизують певні дії та операції в системі. Звіти – демонструють інформацію щодо певної діяльності автошколи.

2 КОНЦЕПТУАЛЬНЕ ПРОЕКТУВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

З концептуального проектування починається створення концептуальної схеми БД, в основі якої лежить концептуальна модель даних. Концептуальна модель представляє загальний погляд на дані, а опис предметної області в термінах деякої моделі даних називають концептуальною схемою [2].

2.1 Аналіз концептуальних вимог та інформаційних потреб

2.1.1 Вимоги користувачів до бази даних

БД повинна виконувати та надавати доступ користувачам до усіх, необхідних відповідно до технічного завдання (ТЗ), функцій системи, а саме:

- облік, додавання, видалення, редагування інформації щодо учнів, отриманих ними свідоцтв, викладачів, груп, теоретичних і практичних занять;
- ведення довідкової системи;
- автоматизований підрахунок вартості практичних занять;
- формування звітної інформації щодо діяльності автошколи.

2.1.2 Виявлення існуючих задач по обробці інформації

Існуючого програмного забезпечення для вирішення задач, що розглядаються у цьому проекті, не було виявлено у вільному доступі. Це досить специфічна сфера, програмні рішення під яку виконуються під замовлення.

2.1.3 Результати аналізу

За виконаним аналізом вимог користувачів і враховуючи факт відсутності існуючих програмних продуктів було зроблено такі висновки:

- система повинна задовольняти всі вимоги користувачів;

- інтерфейс розробленої системи повинен бути зручним та інтуїтивно зрозумілим навіть для користувачів, які не мають базові навички роботи з комп’ютером;

- інтерфейс розробленої системи не повинен бути інформаційно перенавантаженим.

2.2 Виявлення інформаційних об’єктів і зв’язків між ними

Серед інформаційних об’єктів предметної області було виділено такі сутності (об’єкти) та атрибути (властивості, що характеризують сутність):

- сутність «Учень» з атрибутами: код, прізвище, ім'я, по батькові, паспорт, дата народження, адреса проживання, телефон;

- сутність «Свідоцтво» з атрибутами: код, номер, учень, категорія, оцінка з теорії, оцінка з практики, оцінка з будови автомобіля, дата видачі;

- сутність «Викладач» з атрибутами: код, прізвище, ім'я, по батькові, категорії для навчання, оклад, паспорт, дата народження, адреса проживання, телефон;

- сутність «Група» з атрибутами: код, назва, категорія для навчання, викладач теорії, вартість навчання, дата початку навчання, дата закінчення навчання;

- сутність «Теоретичне_навчання» з атрибутами: код, група, тема, початок заняття, кінець заняття;

- сутність «Практичне_навчання» з атрибутами: код, викладач, група, учень, тема, вартість, початок заняття, кінець заняття, сума.

Існують три види зв'язків (відношень) між сутностями:

- «один до одного» – позначається як «1 – 1»;

- «один до багатьох» – позначається як «1 – оо»;

- «багато до багатьох» – позначається як «оо – оо».

Автоматизована система включає в себе два типи зв'язків, а саме: «один до багатьох» та «багато до багатьох».

Зв’язок «один до багатьох» наявний, коли одному запису в першій таблиці може відповідати кілька записів у другій таблиці, проте запису у другій таблиці не може відповідати більше ніж один запис у першій таблиці. Одне з полів, за якими здійснюється зв'язок, – ключове [3].

Зв’язок «багато до багатьох» існує тоді, коли одному запису в першій таблиці відповідає кілька записів у другій таблиці, і навпаки – одному запису в другій таблиці можуть відповідати кілька записів у першій таблиці. Таку схему зв'язку реалізують за допомогою додаткової (зв’язної) таблиці, де одне поле зв’язної таблиці створює зв'язок з першою таблицею, а друге поле зв’язної таблиці створює зв'язок з другою таблицею.

Кожен учень може отримати будь-яку кількість свідоцтв, про закінчення автошколи, тобто одному запису з таблиці «Учень» може відповідати будь-яка кількість записів з таблиці «Свідоцтво». Проте кожне свідоцтво може належати тільки одному учню, тобто одному запису з таблиці «Свідоцтво» може відповідати тільки один запис з таблиці «Учень». Тому між таблицями «Учень» та «Свідоцтво» існує зв’язок «один до багатьох».

Кожен викладач може проводити лекції з теорії у будь-якій кількості груп, тобто одному запису з таблиці «Викладач» може відповідати будь-яка кількість записів з таблиці «Група». Проте за однією групою може бути закріплений тільки один викладач, тобто одному запису з таблиці «Група» може відповідати тільки один запис з таблиці «Викладач». Тому між таблицями «Викладач» та «Група» існує зв’язок «один до багатьох».

Кожен викладач може проводити будь-яку кількість практичних занять, тобто одному запису з таблиці «Викладач» може відповідати будь-яка кількість записів з таблиці «Практичне_навчання». Проте кожне практичне заняття проводиться тільки одним викладачем, тобто одному запису з таблиці «Практичне_навчання» може відповідати тільки один запис з таблиці «Викладач». Тому між таблицями «Викладач» та «Практичне_навчання» існує зв’язок «один до багатьох».

Кожен учень може відвідати будь-яку кількість практичних занять, тобто одному запису з таблиці «Учень» може відповідати будь-яка кількість

записів з таблиці «Практичне_навчання». Проте кожне практичне заняття проводиться тільки з одним учнем, тобто одному запису з таблиці «Практичне_навчання» може відповідати тільки один запис з таблиці «Учень». Тому між таблицями «Учень» та «Практичне_навчання» існує зв'язок «один до багатьох».

Усі учні групи можуть відвідати будь-яку кількість практичних занять, тобто одному запису з таблиці «Група» може відповідати будь-яка кількість записів з таблиці «Практичне_навчання». Проте кожне практичне заняття проводиться тільки з учнем однієї групи, тобто одному запису з таблиці «Практичне_навчання» може відповідати тільки один запис з таблиці «Група». Тому між таблицями «Група» та «Практичне_навчання» існує зв'язок «один до багатьох».

Кожен учень може навчатися у будь-якій кількості груп, тобто одному запису з таблиці «Учень» може відповідати будь-яка кількість записів з таблиці «Група». Кожна група може складатися з будь-якої кількості учнів, тобто одному запису з таблиці «Група» може відповідати будь-яка кількість записів з таблиці «Учень». Тому між таблицями «Учень» та «Група» існує зв'язок «багато до багатьох». Для реалізації цього зв'язку додатково введено сутність «Група_Учень», що складається з таких атрибутів: код, учень, група. Відповідно до цього, між таблицями «Учень» та «Група_Учень» існує зв'язок «один до багатьох», між таблицями «Група» та «Група_Учень» існує зв'язок «один до багатьох».

Для кожної групи може проводитися будь-яка кількість теоретичних занять, тобто одному запису з таблиці «Група» може відповідати будь-яка кількість записів з таблиці «Теоретичне_навчання». Проте кожне теоретичне заняття проводиться тільки для однієї групи, тобто одному запису з таблиці «Теоретичне_навчання» може відповідати тільки один запис з таблиці «Група». Тому між таблицями «Група» та «Теоретичне_навчання» існує зв'язок «один до багатьох».

Кожен учень може пропустити будь-яку кількість теоретичних занять, тобто одному запису з таблиці «Учень» може відповідати будь-яка кількість

записів з таблиці «Теоретичне_навчання». Кожне теоретичне заняття може пропустити будь-яка кількість учнів, тобто одному запису з таблиці «Теоретичне_навчання» може відповідати будь-яка кількість записів з таблиці «Учень». Для реалізації цього зв’язку додатково введено сутність «Пропуск_заняття», що складається з таких атрибутів: код, заняття, учень, коментар. Відповідно до цього, між таблицями «Учень» та «Пропуск_заняття» існує зв’язок «один до багатьох», між таблицями «Теоретичне_навчання» та «Пропуск_заняття» існує зв’язок «один до багатьох».

2.3 Концептуальна схема бази даних

Концептуальну схему БД зображено на рисунку 2.1.

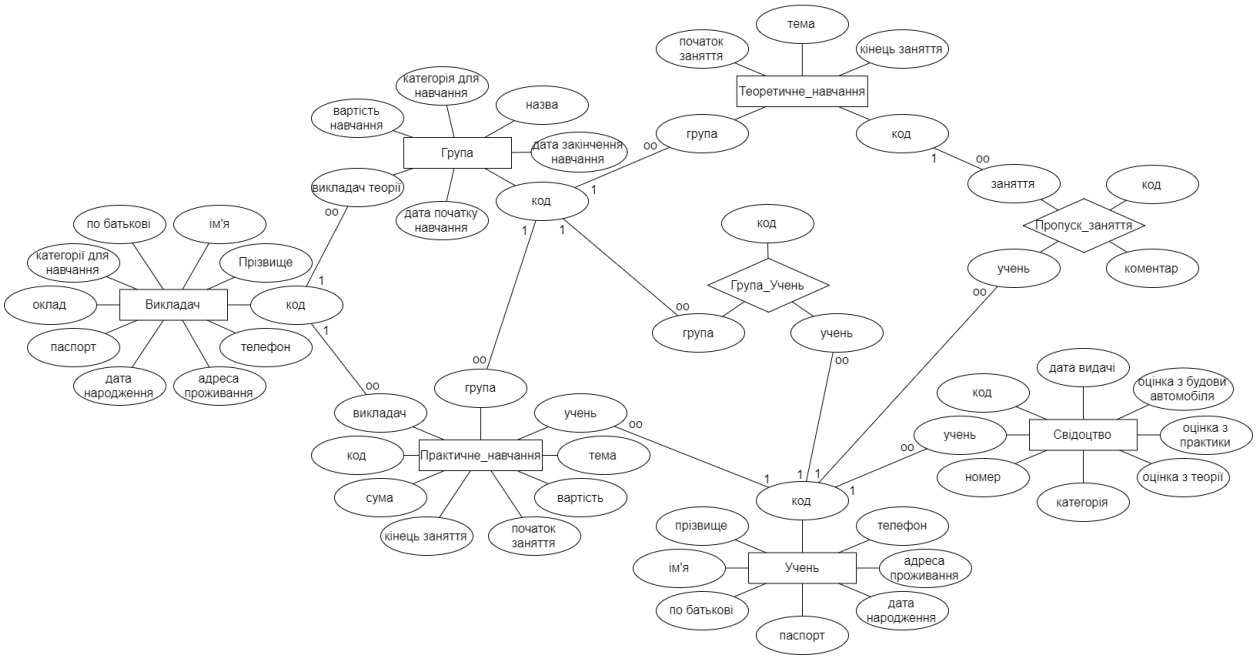

Рисунок 2.1 – Концептуальна модель предметної області

3 ЛОГІЧНЕ ПРОЕКТУВАННЯ

3.1 Вибір і обґрунтування інструментарію проектування

При розробці бази даних актуальним є питання вибору системи керування бозою даних (СКБД). Сьогодні існує великий вибір СКБД, наприклад: MySQL, Oracle RDBMS, PostgreSQL, Microsoft Office Access, MongoDB.

Для розробки системи обрано Microsoft Office Access. Було обрано саме цю СКБД, бо вона має широкий спектр функцій, включаючи зв'язані запити, сортування по різних полях, зв'язок із зовнішніми таблицями і базами даних, вбудовану мову VBA.

СКБД Access є системою управління реляційною базою даних, що включає всі необхідні інструментальні засоби для створення локальної бази даних, загальної бази даних в локальній мережі або в мережі інтернет [4].

Слід зазначити, що для використання готових додатків, створених за допомогою Access, ліцензія не потрібна. Для роботи такого додатку необхідна runtime-версія Access [5], яка розповсюджується безкоштовно. Готова система представлена одним файлом, що сприяє легкому розповсюдженню програми та підвищує безпеку системи, бо майже неможливо втратити файли системи при копіюванні.

3.2 Розробка схеми бази даних

Перед розробкою схеми БД необхідно мати чітке розуміння про структуру її таблиць, які є ототожненням сутностей. Структури таблиць показано в табл. 3.1 – 3.8.

Таблиця 3.1 – Структура таблиці «Учень»

Поле	Тип	Розмір	Обов’язкове	Індексоване
1	2	3	4	5
код	Лічильник	Ціле	Так	Так

Продовження таблиці 3.1

1	2	3	4	5
прізвище	Текстовий	32	Так	Ні
імя	Текстовий	32	Так	Ні
по_батькові	Текстовий	32	Так	Ні
паспорт	Текстовий	8	Так	Ні
дата_народження	Дата і час		Так	Ні
адреса_проживання	Текстовий	64	Так	Ні
телефон	Текстовий	10	Ні	Ні

Таблиця 3.2 – Структура таблиці «Свідоцтво»

Поле	Тип	Розмір	Обов’язкове	Індексоване
код	Лічильник	Ціле	Так	Так
код_учня	Числовий	Ціле	Так	Ні
номер	Текстовий	9	Так	Ні
категорія	Числовий	Ціле	Так	Ні
оцінка_теорія	Числовий	Ціле	Так	Ні
оцінка_практика	Числовий	Ціле	Так	Ні
оцінка_будова_автомобіля	Числовий	Ціле	Так	Ні
дата_видачі	Дата і час		Так	Ні

Таблиця 3.3 – Структура таблиці «Викладач»

Поле	Тип	Розмір	Обов’язкове	Індексоване
1	2	3	4	5
код	Лічильник	Ціле	Так	Так
прізвище	Текстовий	32	Так	Ні
імя	Текстовий	32	Так	Ні
по_батькові	Текстовий	32	Так	Ні
категорії_для_навчання	Числовий	Ціле	Так	Ні
оклад	Грошовий		Так	Ні
паспорт	Текстовий	8	Так	Ні

Продовження таблиці 3.3

1	2	3	4	5
дата_народження	Дата і час		Так	Ні
адреса_проживання	Текстовий	64	Так	Ні
телефон	Текстовий	10	Ні	Ні

Таблиця 3.4 – Структура таблиці «Група»

Поле	Тип	Розмір	Обов’язкове	Індексоване
код	Лічильник	Ціле	Так	Так
назва	Текстовий	16	Так	Ні
категорія_для_навчання	Числовий	Ціле	Так	Ні
код_викладача_теорії	Числовий	Ціле	Так	Ні
вартість_навчання	Грошовий		Так	Ні
дата_початку_навчання	Дата і час		Ні	Ні
дата_закінчення_навчання	Дата і час		Ні	Ні

Таблиця 3.5 – Структура таблиці «Група_Учень»

Поле	Тип	Розмір	Обов’язкове	Індексоване
код	Лічильник	Ціле	Так	Так
код_групи	Числовий	Ціле	Так	Ні
код_учня	Числовий	Ціле	Так	Ні

Таблиця 3.6 – Структура таблиці «Теоретичне_навчання»

Поле	Тип	Розмір	Обов’язкове	Індексоване
код	Лічильник	Ціле	Так	Так
код_групи	Числовий	Ціле	Так	Ні
тема	Числовий	Ціле	Так	Ні
початок_заняття	Дата і час		Так	Ні
кінець_заняття	Дата і час		Так	Ні

Таблиця 3.7 – Структура таблиці «Пропуск_заняття»

Поле	Тип	Розмір	Обов’язкове	Індексоване
код	Лічильник	Ціле	Так	Так
код_заняття	Числовий	Ціле	Так	Ні
код_учня	Числовий	Ціле	Так	Ні
коментар	Текстовий	32	Ні	Ні

Таблиця 3.8 – Структура таблиці «Практичне_навчання»

Поле	Тип	Розмір	Обов’язкове	Індексоване
код	Лічильник	Ціле	Так	Так
код_викладача	Числовий	Ціле	Так	Ні
код_групи	Числовий	Ціле	Так	Ні
код_учня	Числовий	Ціле	Так	Ні
тема	Числовий	Ціле	Так	Ні
вартість	Грошовий		Так	Ні
початок_заняття	Дата і час		Так	Ні
кінець_заняття	Дата і час		Так	Ні
сума	Грошовий		Так	Ні

Для зручного додавання однотипних даних до системи було додано довідкові таблиці «»Категорія», «Тема_теорія», «Тема_практика», структуру яких показано в табл. 3.9 – 3.11.

Таблиця 3.9 – Структура таблиці «Категорія»

Поле	Тип	Розмір	Обов’язкове	Індексоване
код	Лічильник	Ціле	Так	Так
категорія	Текстовий	4	Так	Ні
вартість_навчання	Грошовий		Так	Ні

Таблиця 3.10 – Структура таблиці «Тема_теорія»

Поле	Тип	Розмір	Обов’язкове	Індексоване
Код	Лічильник	Ціле	Так	Так
Номер	Числовий	Ціле	Так	Ні
Назва	Текстовий	128	Так	Ні

Таблиця 3.11 – Структура таблиці «Тема_практика»

Поле	Тип	Розмір	Обов’язкове	Індексоване
Код	Лічильник	Ціле	Так	Так
Номер	Числовий	Ціле	Так	Ні
Назва	Текстовий	128	Так	Ні

Розглянувши структуру всіх таблиць та всі типи зв’язків, що існують між ними (розділ 2.3) було складено схему БД у СКБД Access «Автошкола УченьОК» (рис. 3.1).

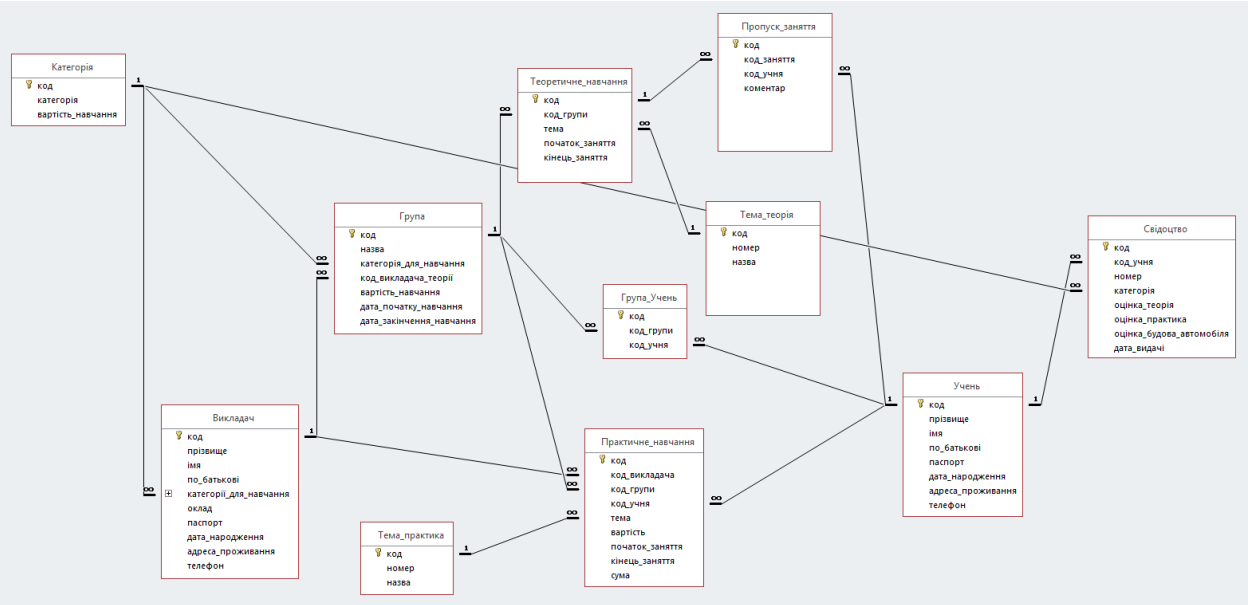

Рисунок 3.1 – Схема БД «Автошкола УченьОК»

3.3 Нормалізація схеми бази даних

Нормалізація – це процес організації даних в БД, що включає встановлення відносин між таблицями (сутностями) відповідно до правил, які

забезпечують захист даних і роблять базу даних більш гнучкою, усуваючи надмірність і неузгоджені залежності [6].

Існує декілька правил нормалізації баз даних. Кожне правило називається нормальною формою (НФ). Якщо виконується, наприклад, перше правило нормалізації – говорять, що база даних представлена в «першій нормальній формі».

Існують такі нормальні форми:

- перша нормальна форма (1НФ) – всі неключові атрибути кожного відношення прості, тобто містять атомарні (неподільні) значення;

- друга нормальна форма (2НФ) – кожне відношення знаходиться у 1НФ і кожний неключовий атрибут в кожному відношенні функціонально повно залежить від будь-якого множинного ключа;

- третя нормальна форма (3НФ) – кожне відношення заходиться у 2НФ і в кожному відношенні відсутні транзитивні залежності;

- нормальна форма Бойса-Кодда (НФБК) – кожне відношення знаходиться у 3НФ і в кожному відношенні детермінанти є потенційними ключами;

- четверта нормальна форма (4НФ) – кожне відношення знаходиться у НФБК і в кожному відношенні відсутні незалежні багатозначні залежності.

Проаналізувавши схему даних БД «Автошкола УченьОК» можна зробити висновок, що розроблена БД відповідає визначенню 4НФ, а, отже, не потребує нормалізації.

3.4 Проектування запитів

Запити – спеціальні об’єкти, призначені для вибірки даних з таблиць бази, а також для виконання обчислень та інших операцій з базовими таблицями, включаючи їхнє перетворення. У результаті запиту утвориться таблиця. Інформація є актуальною, відбиває останні зміни даних [7].

Важливою умовою реалізації запитів є встановлення зв’язків між таблицями, а саме, полями різних таблиць. За допомогою таких зв’язків і реалізуються запити, звернені відразу до декількох таблиць бази даних.

Запроси реалізуються за допомогою використання мови структурованих запитів – Structured Query Language.

В БД «Автошкола УченьОК» було використано запити на вибірку:

- викладачів, які можуть проводити заняття за вказаною категорією;
- груп, для яких викладач може проводити заняття за вказаними категоріями навчання у викладача;
- учнів за групою;
- учнів за теоретичним заняттям;
- доступних категорій для отримання свідоцтва про закінчення автошколи за учнем;
- практичних занять за вказаний рік та місяць.

В БД «Автошкола УченьОК» було використано запити на вибірку та підрахунок:

- кількості учнів за групою чи категорією;
- кількості робочих годин викладачів практики за рік;
- кількості теоретичних годин за групами за вказаний рік;
- викладачів та середнього балу свідоцтв, отриманих їхніми учнями;
- кількість пропусків теоретичних занять учнями.

Усі тексти запитів наведено у додатку А.

4 ІНТЕРФЕЙС КОРИСТУВАЧА СИСТЕМИ

4.1 Проектування форм

Форма Access – це об’єкт бази даних, за допомогою якого можна створити користувацький інтерфейс для програми бази даних. Форми можна порівняти з вікнами, через які користувачі бачать БД і звертаються до неї. Ефективна форма прискорює роботу з базою, оскільки позбавляє від необхідності шукати потрібну інформацію [8]. Візуально приваблива форма сприяє більш приємній та ефективній роботі з БД, а також допомагає запобігти введенню невірних даних, а отже інтерфейс БД повинен бути реалізований інтуїтивно зрозумілим для користувача – непрофесіонала в комп’ютерній галузі. Від користувача потрібні лише пізнання в його предметній області.

При розробці форм за мету було визначено створення функціонального, проте простого, неперенавантаженого зайвою інформацією, інтерфейсу. Інтерфейс, створений за такими принципами сприятиме зручному використанню БД, та підвищить конкурентоздатність системи.

В системі наявні такі форми:

- Меню;
- Практичне навчання;
- Теоретичне навчання;
- Пропуски занять;
- Викладачі;
- Учні;
- Формування груп;
- Свідоцтва;
- Категорії;
- Теми з теорії;
- Теми з практики;
- Форма підтвердження закриття БД;
- Форма для вибору групи для формування звіту про її склад;
- Форма для вибору групи для формування звіту про проведені лекції.

В процесі розробки інтерфейсу перевага була віддана спокійним, проте яскравим тонам, які привертають увагу користувача. Для полегшення процесу візуального сприйняття, основні елементи управління мають закруглені кути, так як вони не просто приємні для погляду, а й полегшують сприйняття графіки і обробку інформації людиною.

Всі розроблені форми мають спільну кольорову гамму, стилі шрифтів та підходи до розташування інтерфейсу. Це візуально підтримує цілісний образ системи.

Кольорова гама переважно складається з чотирьох кольорів:

- Шістнадцятковий код кольору #292929 – темно сірий відтінок кольору, що використовується для оформлення заголовків форм;
- Шістнадцятковий код кольору #505050 –сірий відтінок кольору, що використовується для оформлення заголовків таблиць та описових даних;
- Шістнадцятковий код кольору #1D1D1D – темно сірий відтінок кольору, що використовується для оформлення даних з БД, що відображаються у формах;
- Шістнадцятковий код кольору #FFDE40 – відтінок жовтого кольору, що використовується для оформлення фону форми.

На відміну від чорного кольору, темно сірі та сірі відтінки виглядають більш нейтрально і легко поєднується з іншими кольорами в інтерфейсі. Тому саме їх було обрано для відображення інформації.

Для фону форм було використано відтінок жовтого кольору, бо у призначеному для користувача інтерфейсі жовтий колір привертає увагу. Також цей колір асоціюється у користувача з деякими дорожними знаками, що ненав’язливо пов’язує програму з предметною областю на рівні асоціацій.

Як основний шрифт для відображення текстової інформації було обрано «Century Schoolbook».

Майже на кожну форму (де є необхідність) додано кнопки навігації по записам у таблиці: на перший, назад, вперед, на останній (рис. 4.1) та кнопки керування записами у таблиці: пошук, додавання й видалення запису (рис. 4.2).

Рисунок 4.1 – Кнопки навігації по записам у таблиці

Рисунок 4.2 – Кнопки керування записами у таблиці

Для закриття форми користувач може натиснути на хрестик у верхній правій частині форми, чи на кнопку закриття форми (рис. 4.3)

Рисунок 4.3 – Кнопка закриття форми

Головною формою є форма «Меню» (рис. 4.4). Вона автоматично відкривається при запуску системи і є своєрідним контейнером з кнопками, які реалізують перехід до всіх інших, доступних для користувача системи, форм.

Рисунок 4.4 – Форма «Меню»

У верхній частині форми «Меню» зображено логотип продукту, а у нижній частині – стилізовані іконки транспортних засобів, що ненав’язливо пов’язує програму з предметною областю на рівні асоціацій.

На початковому меню доступна кнопка «Вихід», що відкриває форму «Підтвердження закриття БД», яка у формі діалогового вікна запитує підтвердження у користувача на закриття системи (рис. 4.5). В підрозділах (підменю) доступна кнопка «Назад», яка активує початкове меню.

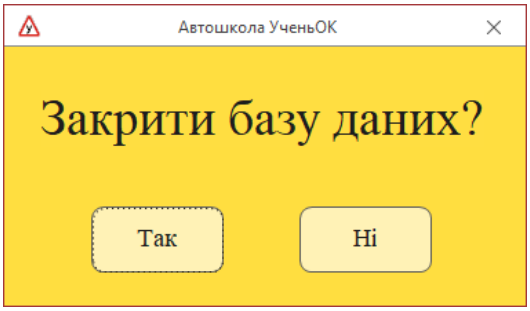

Рисунок 4.5 – Форма «Підтвердження закриття БД»

В підменю «Заняття» форми «Меню» (рис. 4.6) наявні кнопки, які відповідають за відкриття відповідних до назв кнопок форм: «Практичне навчання», «Теоретичне навчання», «Пропуски занять».

Рисунок 4.6 – Підменю «Заняття» форми «Меню»

Завдяки формі «Практичне навчання» (рис. 4.7) користувач може додавати, редагувати, видаляти інформацію про практичні заняття, а також переглядати більш детальну інформацію про викладача, учня та групу.

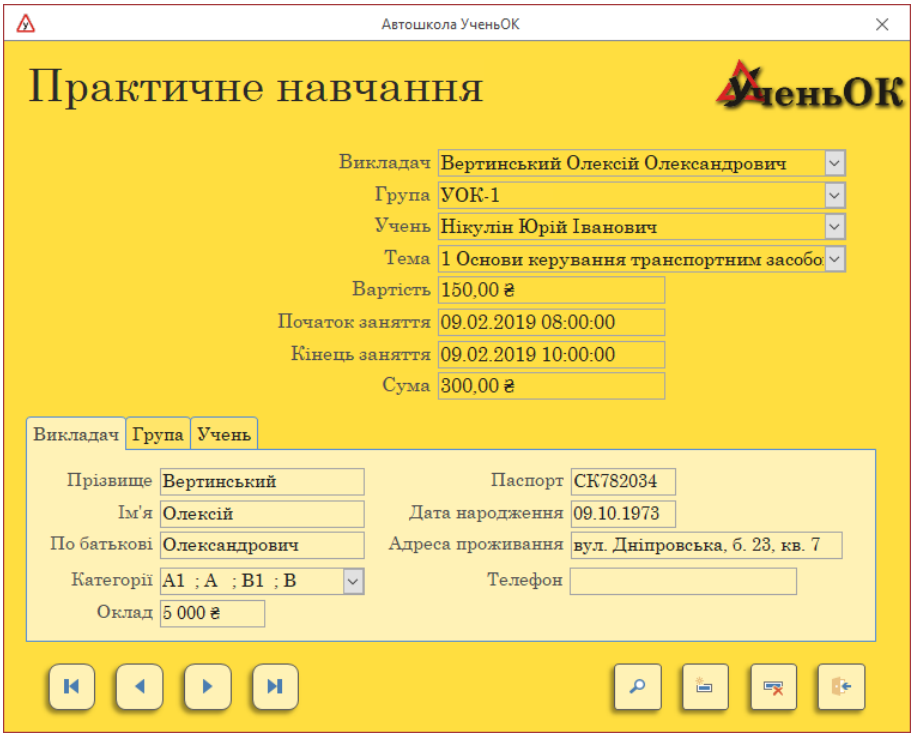

Рисунок 4.7 – Форма «Практичне навчання»

Завдяки формі «Теоретичне навчання» (рис. 4.8) користувач може додавати, редагувати, видаляти інформацію про теоретичні заняття, а також переглядати інформацію про учнів, які пропустили ці заняття.

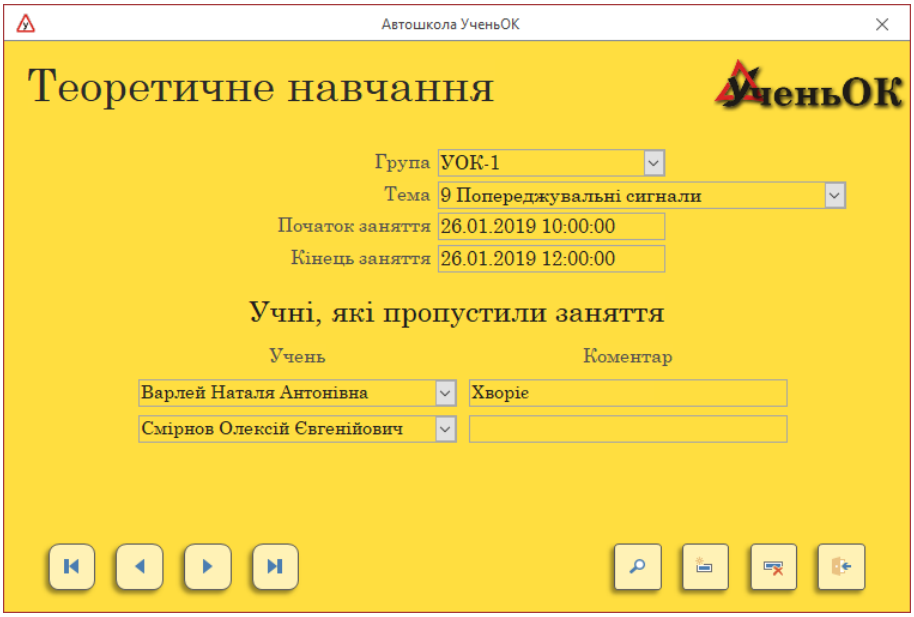

Рисунок 4.8 – Форма «Теоретичне навчання»

Завдяки формі «Пропуски занять» (рис. 4.9) користувач може додавати, редагувати, видаляти інформацію про пропуски теоретичних занять учнями, а також переглядати більш детальну інформацію про заняття та учня.

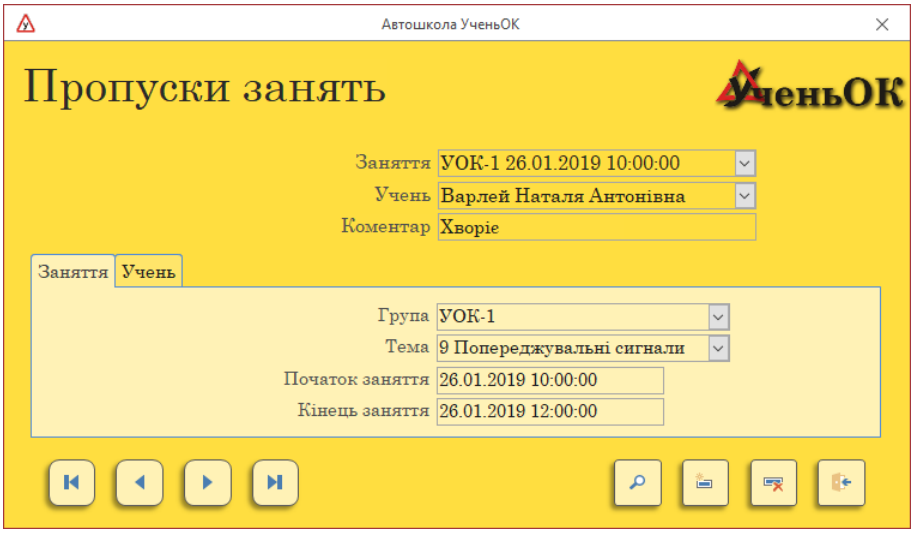

Рисунок 4.9 – Форма «Пропуски занять»

В підменю «Склад автошколи» форми «Меню» (рис. 4.10) наявні кнопки, які відповідають за відкриття відповідних до назв кнопок форм: «Викладачі», «Учні», «Групи», «Формування груп».

Рисунок 4.10 – Підменю «Склад автошколи» форми «Меню»

Завдяки формі «Викладачі» (рис. 4.11) користувач може додавати, редагувати, видаляти інформацію про викладача, а також переглядати інформацію про групи, в яких він викладав теорію.

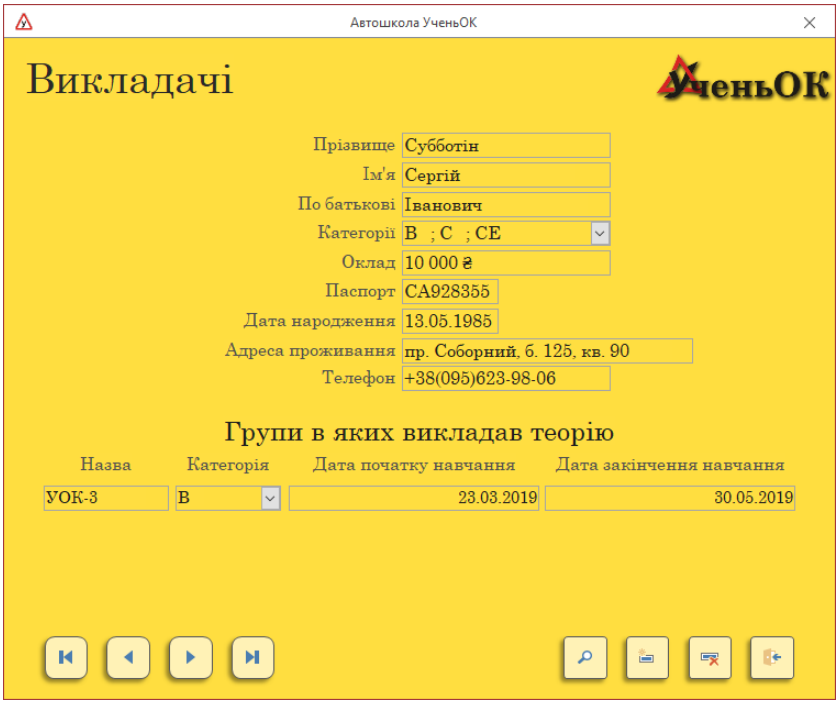

Рисунок 4.11 – Форма «Викладачі»

Завдяки формі «Учні» (рис. 4.12) користувач може додавати, редагувати, видаляти інформацію про учня, а також переглядати інформацію про групи, в яких він навчався.

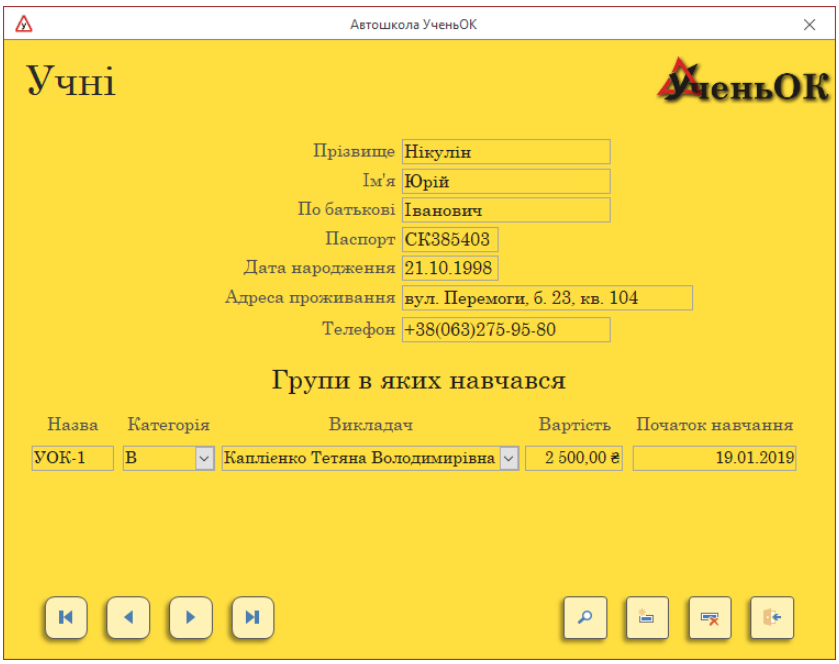

Рисунок 4.12 – Форма «Учні»

Завдяки формі «Групи» (рис. 4.13) користувач може додавати, редагувати, видаляти інформацію про групу, а також переглядати інформацію про теми лекцій, які вже відбулися.

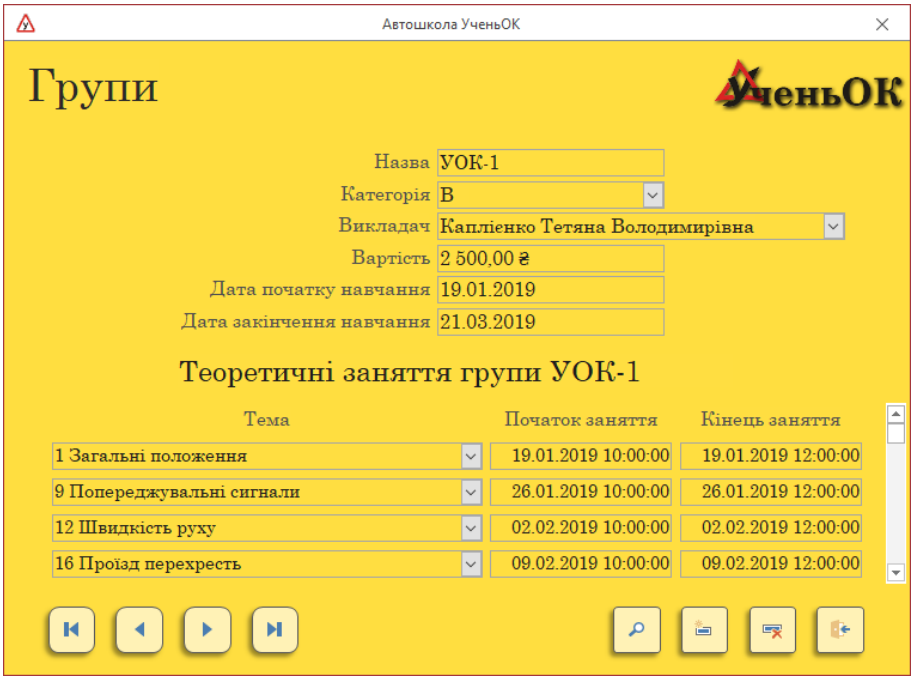

Рисунок 4.13 – Форма «Групи»

Завдяки формі «Формування груп» (рис. 4.14) користувач може додавати, редагувати, видаляти інформацію про склад групи, а також переглядати більш детальну інформацію про групу та учня.

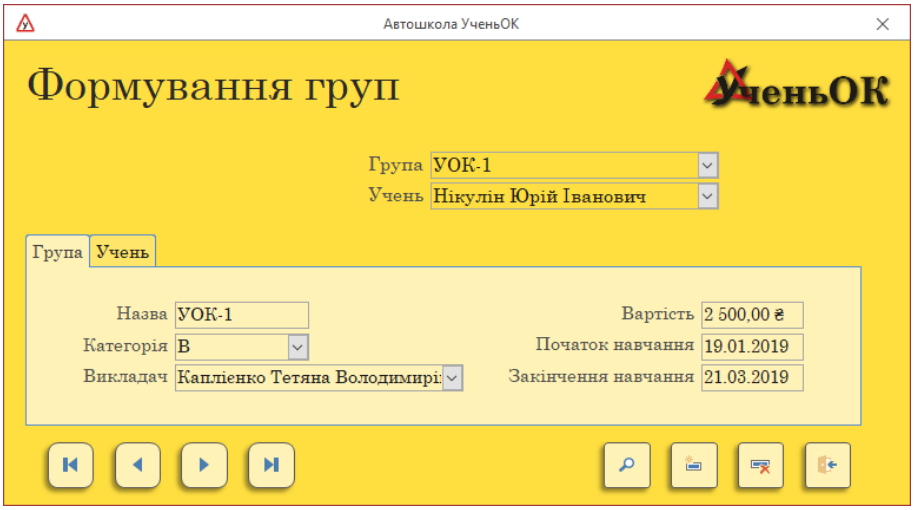

Рисунок 4.14 – Форма «Формування груп»

Кнопка «Свідоцтва» початкового меню форми «Меню» відкриває форму «Свідоцтва». Завдяки формі «Свідоцтва» (рис. 4.15) користувач може

додавати, редагувати, видаляти інформацію про свідоцтва, отримані учнем після закінчення автошколи, а також переглядати більш детальну інформацію про самого учня.

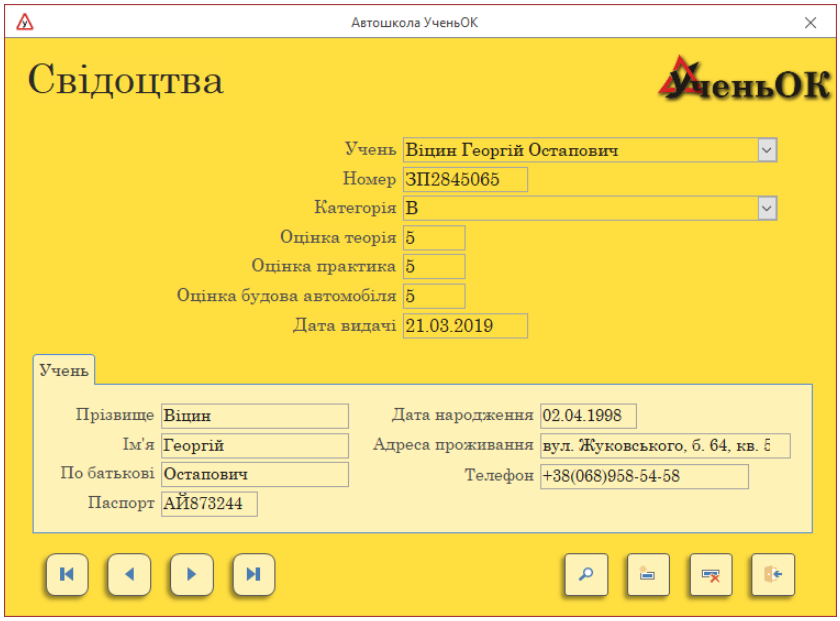

Рисунок 4.15 – Форма «Свідоцтва»

В підменю «Довідкові дані» форми «Меню» (рис. 4.16) наявні кнопки, які відповідають за відкриття відповідних до назв кнопок форм: «Категорії», «Теми з теорії», «Теми з практики».

Рисунок 4.16 – Підменю «Довідкові дані» форми «Меню»

Завдяки формі «Категорії» (рис. 4.17) користувач може додавати, редагувати, видаляти інформацію про категорії для навчання у автошколі.

Рисунок 4.17 – Форма «Категорії»

Завдяки формі «Теми з теорії» (рис. 4.18) користувач може додавати, редагувати, видаляти інформацію про теми для теоретичних занять.

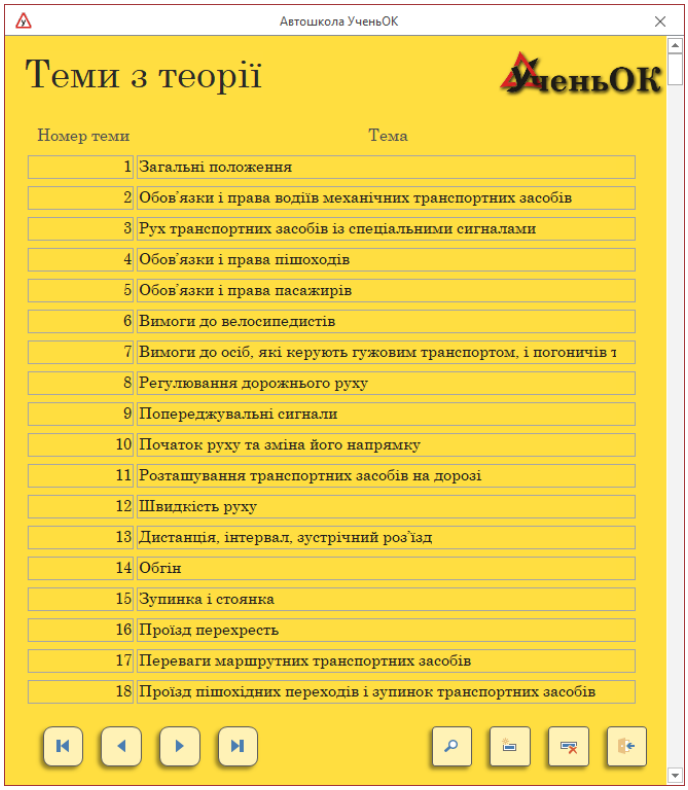

Рисунок 4.18 – Форма «Теми з теорії»

Завдяки формі «Теми з практики» (рис. 4.19) користувач може додавати, редагувати, видаляти інформацію про теми для практичних занять.

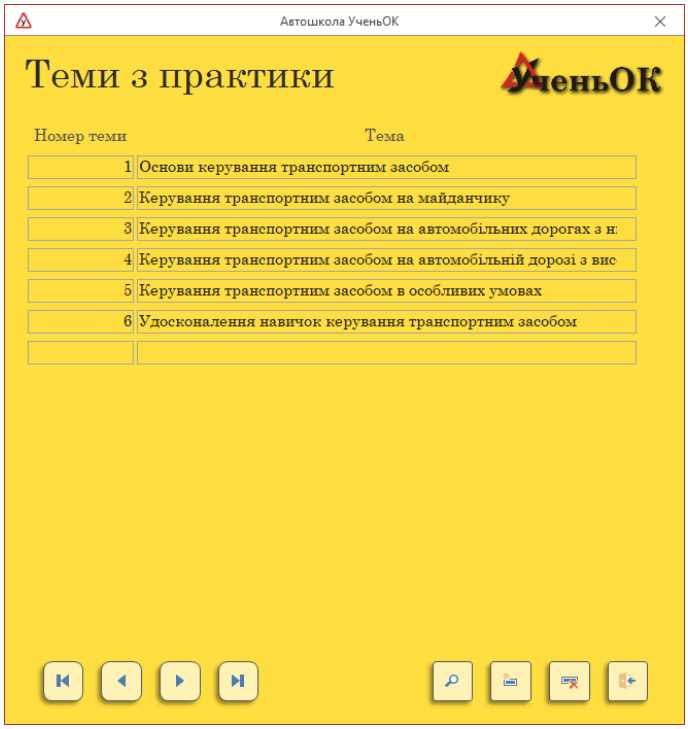

Рисунок 4.19 – Форма «Теми з практики»

В підменю «Звіти» форми «Меню» (рис. 4.20) наявні кнопки, які відповідають за формування і відкриття звітів. Кнопки «Склад групи» та «Лекції групи» відкривають форму для вибору групи для формування звіту про її склад, та форму для вибору групи для формування звіту про проведені лекції.

Рисунок 4.20 – Підменю «Звіти» форми «Меню»

Завдяки формі для вибору групи для формування звіту про її склад (рис. 4.21) користувач може вибрати групу і отримати відповідний до вибору звіт.

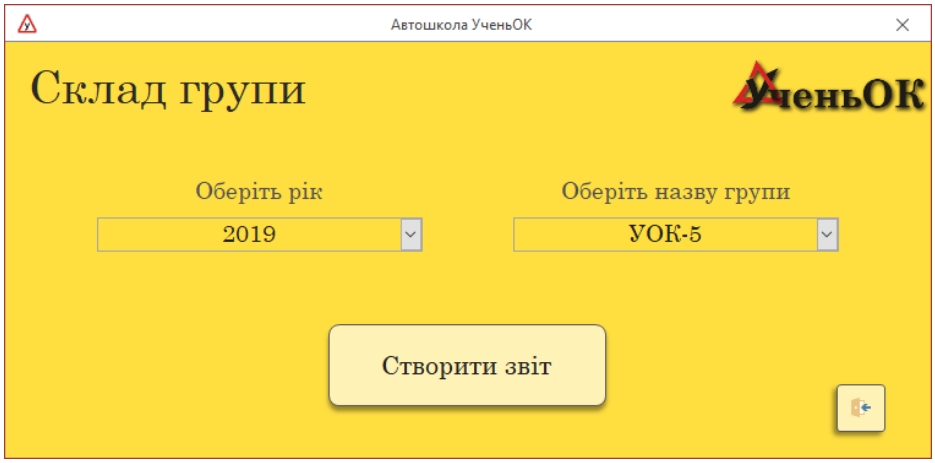

Рисунок 4.21 – Форма для вибору групи для формування звіту про її склад

Завдяки формі для вибору групи для формування звіту про проведені лекції (рис. 4.22) користувач може вибрати групу і отримати відповідний до вибору звіт.

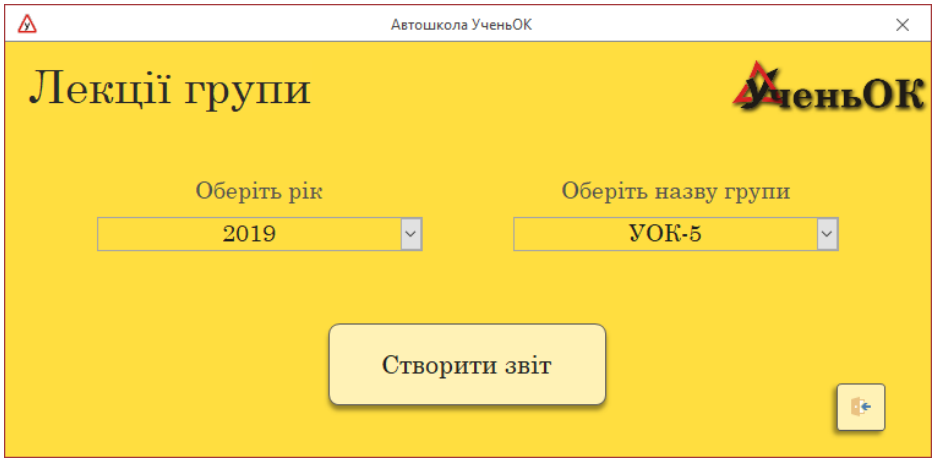

Рисунок 4.22 – Форма для вибору групи для формування звіту про лекції

4.2 Проектування звітів

Звіт – це представлення даних у спеціальному форматі, яке виводиться на екран або на друк. За допомогою звітів можна переглядати, форматувати та підсумовувати дані, адже звіти служать для формування документів,

створених на основі таблиць, запитів. На відміну від форм, вони не можуть бути використані для введення даних у таблиці.

Звіти дозволяють отримувати з бази потрібні відомості й представляти їх у вигляді, зручному для сприйняття. Також звіти надають широкі можливості для узагальнення й аналізу даних.

Звіти було розроблено в мінімалістичному стилі з використанням якомога меншої кількості кольорів, з метою економії ресурсів, що можуть бути витрачені на фарбу при друку.

В БД «Автошкола УченьОК» передбачено можливість генерувати такі звіти:

- «Склад групи» – відображення списку учнів, що навчаються у вибраній користувачем групі;
- «Лекції групи» – відображення інформації про всі теоретичні заняття проведені у вибраній користувачем групі, а також вивчені теми;
- «Затребуваність категорій» – відображення статистичної інформації щодо затребуваності категорій серед учнів;
- «Дохід з практики» – відображення інформації щодо роботи викладачів та дохід з практичних занять за вибраним користувачем місяцем та роком.

Вигляд звітів наведено у додатку В.

4.3 Проектування макросів

Макрос – це інструмент програми Access, за допомогою якого можна автоматизувати завдання та розширити можливості форм, звітів і елементів керування [9].

Макроси програми Access – це ніби спрощена мова програмування, за допомогою якої складається список дій, які потрібно виконати. Макроси допомагають розширити можливості форм, звітів і елементів керування без написання коду в модулі Visual Basic for Applications (VBA). Макроси являють собою підмножину команд, доступних у VBA.

При розробці БД «Автошкола УченьОК» було використано такі макроси:

- оновлення деяких полів;
- визначення вартості навчання за категорією (рис. 4.23);
- відкриття звіту «Склад групи» (рис. 4.24);
- відкриття звіту «Лекції групи» (рис. 4.25);
- визначення суми практичного заняття за вартістю і датами (рис. 4.26);
- повідомлення при закритті бази даних (рис. 4.27).

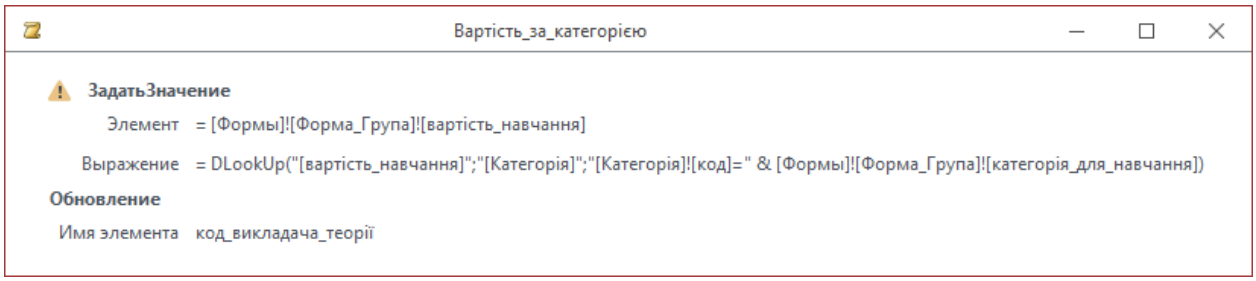

Рисунок 4.23 – Конструктор макросу «Визначення вартості навчання за категорією»

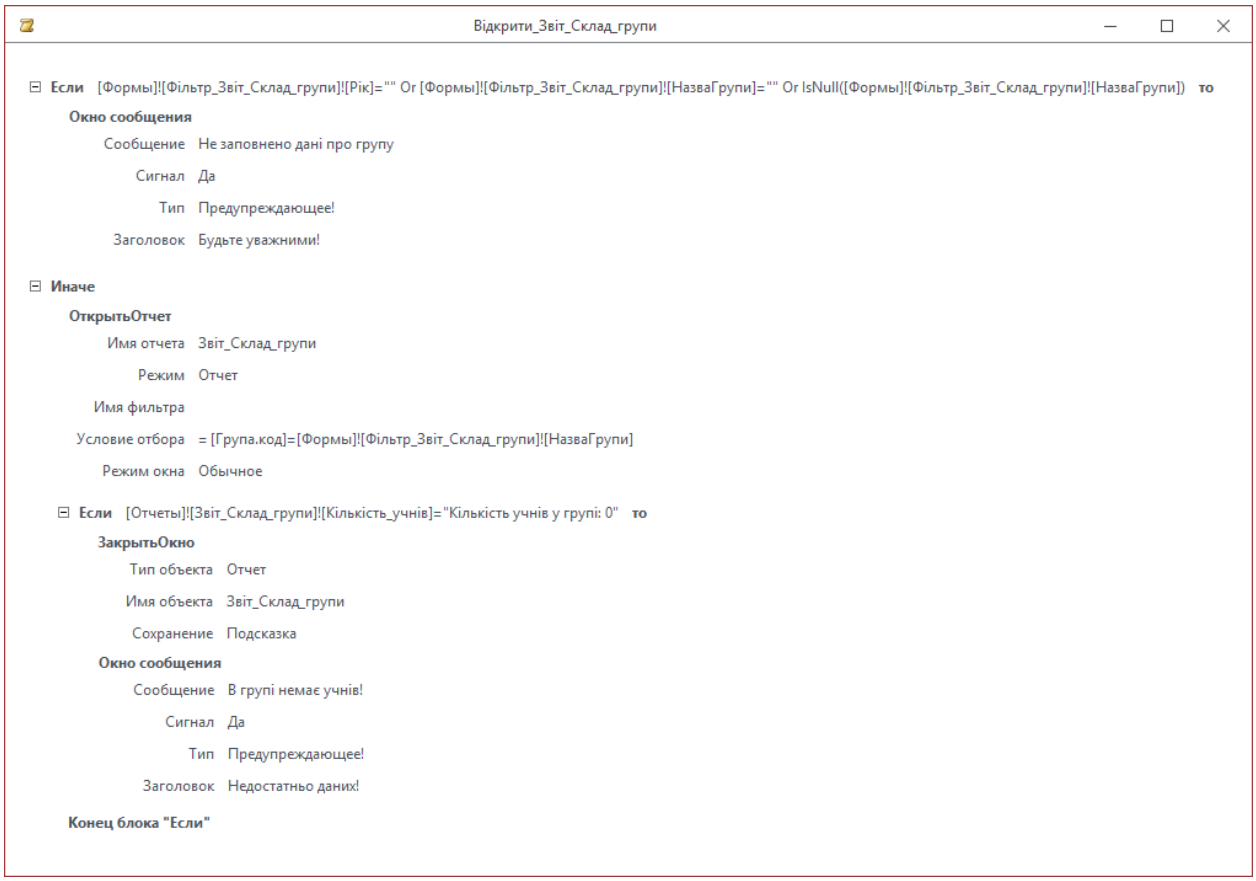

Рисунок 4.24 – Конструктор макросу «Відкриття звіту «Склад групи»

Відкрити_Звіт_Лекції_групи

Если [Формы]![Фільтр_Звіт_Лекції_групи]![Рік]="" Or [Формы]![Фільтр_Звіт_Лекції_групи]![НазваГрупи]="" Or IsNull([Формы]![Фільтр_Звіт_Лекції_групи]![НазваГрупи]) **то**

Окно сообщения

Сообщение Не заповнено дані про групу

Сигнал Да

Тип Предупреждающее!

Заголовок Будьте уважними!

Иначе

ОткрытьОтчет

Имя отчета Звіт_Лекції_групи

Режим Отчет

Имя фильтра

Условие отбора = [Група.код]=[Формы]![Фільтр_Звіт_Лекції_групи]![НазваГрупи]

Режим окна Обычное

Если [Отчеты]![Звіт_Лекції_групи]![Кількість_лекцій]="Загальна кількість лекцій: 0" **то**

ЗакрытьОкно

Тип объекта Отчет

Имя объекта Звіт_Лекції_групи

Сохранение Подсказка

Окно сообщения

Сообщение В групі не було лекцій!

Сигнал Да

Тип Предупреждающее!

Заголовок Недостатньо даних!

Конец блока "Если"

Конец блока "Если"

Рисунок 4.25 – Конструктор макросу «Відкриття звіту «Лекції групи»

Сума_за_вартістю_і_датами

Если [Формы]![Форма_Практичне_навчання]![вартість]<>"" And [Формы]![Форма_Практичне_навчання]![початок_заняття]<>"" And [Формы]![Форма_Практичне_навчання]![кінець_заняття]<>"" **то**

ЗадатьЗначение

Элемент = [Формы]![Форма_Практичне_навчання]![сума]

Выражение = [вартість]*(Hour([кінець_заняття])-Hour([початок_заняття]))

Конец блока "Если"

Рисунок 4.26 – Конструктор макросу «Визначення суми практичного заняття за вартістю і датами»

Повідомлення_при_закритті_бд

Окно сообщения

Сообщение Дякую за користування програмою! До нових зустрічей!

Сигнал Да

Тип Информационное

Заголовок Допобачення...

ЗапускКомандыМеню

Команда ЗакрытьБазуДанных

Рисунок 4.27 – Конструктор макросу «Повідомлення при закритті бази даних»

5 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ

Система «Автошкола УченьОК» при запуску автоматично відкриває меню, структурну схему якого зображено на рисунку 5.1.

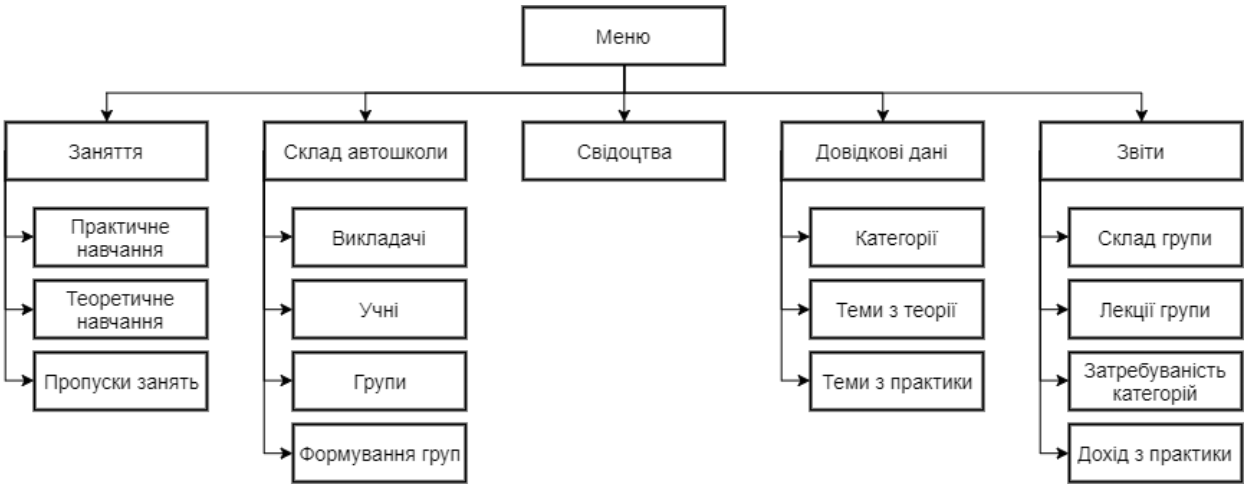

Рисунок 5.1 – Структурна схема меню системи

Для обирання пунктів меню треба натиснути на кнопку, що розташована ліворуч від напису потрібного пункту, або безпосередньо на напис. Кнопка вибору позначена квадратом із закругленими кутами (рис. 5.2).

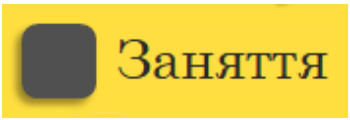

Рисунок 5.2 – Приклад кнопки з написом для вибору пункту меню

На початковому меню передбачено кнопку виходу з програми (рис. 5.3), після натискання на яку в користувача у діалоговому вікні буде запитано підтвердження його намірів закриття БД.

Рисунок 5.3 – Кнопка для виходу з бази даних

В підрозділах (підменю) доступна кнопка «Назад» (рис. 5.4), яка активує початкове меню.

Рисунок 5.4 – Кнопка для повернення у початкове меню

Майже на кожну форму (де є необхідність) додано кнопки навігації та керування записами у таблиці: «На перший» (рис. 5.5) – для переходу на перший запис, «Назад» (рис. 5.6) – для переходу на попередній запис, «Вперед» (рис. 5.7) – для переходу на наступний запис, «На останній» (рис. 5.8) – для переходу на останній запис, «Пошук» (рис. 5.9) – для переходу на запис, за заданими критеріями пошуку, «Додавання запису» (рис. 5.10) – для переходу до пустого запису для заповнення даними, «Видалення запису» (рис. 5.11) – для видалення поточного запису.

Рисунок 5.5 – Кнопка «На перший»

Рисунок 5.6 – Кнопка «Назад»

Рисунок 5.7 – Кнопка «Вперед»

Рисунок 5.8 – Кнопка «На останній»

Рисунок 5.9 – Кнопка «Пошук»

Рисунок 5.10 – Кнопка «Додавання»

Рисунок 5.11 – Кнопка «Видалення»

Для закриття форми користувач може натиснути на хрестик у верхній правій частині форми, чи на кнопку закриття форми (рис. 5.12).

Рисунок 5.12 – Кнопка закриття форми

Для додавання нового запису потрібно відкрити форму з суміжними за типом записами, перейти на останню сторінку форми або натиснути на відповідну навігаційну кнопку і потім ввести нові значення.

Для редагування певного запису потрібно відкрити відповідну форму, за допомогою навігаційних кнопок перейти до потрібного запису і змінити необхідні значення.

Для видалення певного запису потрібно відкрити відповідну форму, за допомогою навігаційних кнопок перейти до потрібного запису і натиснути на кнопку, що відповідає за видалення запису.

Слід зауважити, що в системі передбачено захист від помилкового вводу даних.

Частина перевірок відбувається без повідомлення користувачу, з метою зменшення інформаційного навантаження на нього. Наприклад в формі «Практичне навчання» при додаванні чи редагуванні даних про учня користувачу буде запропоновано учнів тільки з вибраної групи. Це зроблено з метою уникнення ситуації, плутанини даних.

Інша частина перевірок розрахована на діалог з користувачем. Це перевірки на коректність вхідних даних, на заповнення усіх обов’язкових

полів, перевірки від створення «пустих» звітів. В разі виникнення помилки користувачу буде надано повідомлення з описом дій, які треба зробити для усунення проблем. Приклад такого повідомлення зображено на рисунку 5.13.

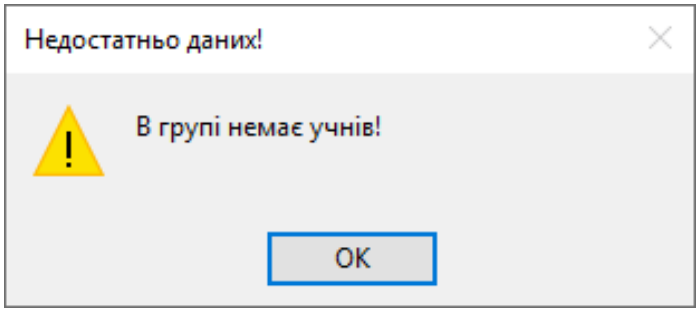

Рисунок 5.13 – Приклад повідомлення про помилку при створенні звіту

ВИСНОВКИ

У даному курсовому проекті було вирішено такі задачі:

- аналіз предметної області;
- розробка концептуальної моделі предметної області;
- вибір інструментарію для розробки системи;
- розробка користувацького інтерфейсу системи;
- заповнення системи реальними даними та її тестування.

В результаті проведеної роботи було створено автоматизовану систему обліку діяльності автошколи, що відповідає усім вимогами ТЗ, які були висунуті замовником.

Завдяки простому, інтуїтивно зрозумілому, неперенавантаженому інтерфейсу розроблену систему з легкістю може використовувати навіть людина, яка має лише базові навички роботи з комп'ютером. Передбачено генерування звітної документації, що є суттєвою допомогою при роботі з документацією на підприємстві, а також дозволяє проводити статистичні аналізи і розрахунки.

Було проведено випробування системи, яке показало, що створений програмний продукт відповідає усім вимогам замовника і є економічно доцільним для впровадження в робочий процес.

В якості середовища розробки було використано СКБД Microsoft Access 2016. Використана СКБД працює на ОС Windows 10 чи ОС Windows 7 з встановленою програмою Microsoft(R) Access 2016 та новіше.

ПЕРЕЛІК ПОСИЛАНЬ

1. Системні вимоги для Office [Електронний ресурс]. – Режим доступу: https://products.office.com/uk-ua/office-system-requirements

2. Малахов Є. В. Основи проектування баз даних / Є.В. Малахов. – О: Наука и техника, 2006. – 156 с.

3. Корнієнко, С. К. Системи баз даних: організація та проектування : навч. посібник / С.К. Корнієнко. – Запоріжжя: ЗНТУ, 2006. – 252 с.

4. Бекаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2016 / Ю.Б. Бекаревич, Н.В. Пушкина. – СПб.: БХВ-Петербург, 2017. – 480 с.

5. Балтер Э. Microsoft Office Access 2007: профессиональное программирование / Э. Балтер. – М.: Вильямс, 2008. – 1296 с.

6. Кириллов В. В., Громов Г. Ю. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб.: БХВ-Петербург, 2009. – 464 с.

7. Карпова И. П. Базы данных. Учебное пособие / И.П. Карпова. – СПб.: Питер, 2013. – 240 с.

8. Довідка з Access [Електронний ресурс]. – Режим доступу: https://support.office.com/uk-ua/access

9. Макрокоманда [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Макрокоманда

ДОДАТОК А
ТЕКСТИ SQL-ЗАПИТІВ

А1 Викладачі, які можуть проводити заняття за вказаною категорією

```
SELECT
    Викладач.код,
    RTrim([прізвище]) & " " & RTrim([імя]) & " " & RTrim([по_батькові]) AS ПІБ
FROM
    Викладач
WHERE
    Викладач.категорії_для_навчання.Value
=[Формы]![Форма_Група]![категорія_для_навчання];
```

А2 Групи, для яких викладач може проводити заняття за вказаними категоріями навчання у викладача

```
SELECT
    Група.код,
    Група.назва,
    Викладач.код
FROM
    (Категорія INNER JOIN Викладач
        ON Категорія.код = Викладач.категорії_для_навчання.Value)
            INNER JOIN Група
                ON Категорія.код = Група.категорія_для_навчання
WHERE
    Викладач.код = [Forms]![Форма_Практичне_навчання]![код_викладача];
```

А3 Учні за групою

```
SELECT
    Учень.код,
    RTrim([прізвище]) & " " & RTrim([імя]) & " " & RTrim([по_батькові]) AS ПІБ
FROM
    Учень INNER JOIN Група_Учень
        ON Учень.код = Група_Учень.код_учня
WHERE
    Група_Учень.код_групи = [Forms]![Форма_Практичне_навчання]![код_групи];
```

А4 Учні за теоретичним заняттям

```
SELECT
    Учень.код,
    RTrim([прізвище]) & " " & RTrim([імя]) & " " & RTrim([по_батькові]) AS ПІБ
FROM
    ((Група INNER JOIN Група_Учень
        ON Група.код = Група_Учень.код_групи)
            INNER JOIN Теоретичне_навчання
```

```
                ON Група.код = Теоретичне_навчання.код_групи)
                    INNER JOIN Учень
                        ON Учень.код = Група_Учень.код_учня
WHERE
    Теоретичне_навчання.код = [Forms]![Форма_Пропуск_заняття]![код_заняття];
```

А5 Доступні категорії для отримання свідоцтва про закінчення автошколи за учнем

```
SELECT
    Категорія.код,
    Категорія.категорія
FROM
    ((Група INNER JOIN Група_Учень
        ON Група.код = Група_Учень.код_групи)
            INNER JOIN Категорія
                ON Категорія.код = Група.категорія_для_навчання)
                    INNER JOIN Учень
                        ON Учень.код = Група_Учень.код_учня
WHERE
    Учень.код = [Формы]![Форма_Свідоцтво]![код_учня];
```

А6 Практичні заняття за вказаний рік та місяць

```
SELECT
    RTrim([Викладач.прізвище]) & " " & RTrim([Викладач.імя]) & " " &
RTrim([Викладач.по_батькові]) AS ПІБ_викладача,
    Практичне_навчання.код_групи,
    Практичне_навчання.вартість,
    Практичне_навчання.початок_заняття,
    Практичне_навчання.кінець_заняття,
    Практичне_навчання.сума,
    RTrim([Учень.прізвище]) & " " & RTrim([Учень.імя]) AS ПІБ_учня
FROM
    (Викладач INNER JOIN Практичне_навчання
        ON Викладач.код = Практичне_навчання.код_викладача)
            INNER JOIN Учень
                ON Учень.код = Практичне_навчання.код_учня
WHERE
    Year([початок_заняття]) = [Введіть рік]
        AND MonthName(Month([початок_заняття])) Like "*"+[Введіть назву місяця]+"*";
```

А7 Кількість учнів за групою чи категорією

```
SELECT
    Категорія.категорія,
     RTrim([назва]) AS назва_групи,
    Популярність.кількість_учнів,
    RTrim([прізвище]) & " " & RTrim([імя]) & " " & RTrim([по_батькові]) AS
ПІБ_Викладача
FROM
    (((SELECT
```

```
        Група.код AS група_код,
        Count(Учень.код) AS кількість_учнів
    FROM
        (Група INNER JOIN Група_Учень
            ON Група.код = Група_Учень.код_групи)
                INNER JOIN Учень
                    ON Учень.код = Група_Учень.код_учня
    GROUP BY
        Група.код) AS Популярність
        INNER JOIN Група
            ON Група.код = Популярність.група_код)
                INNER JOIN Викладач
                    ON Викладач.код = Група.код_викладача_теорії)
                        INNER JOIN Категорія
                            ON Категорія.код = Група.категорія_для_навчання;
```

А8 Кількість теоретичних годин за групами за вказаний рік

```
SELECT
    Група.назва,
    Sum(Hour(кінець_заняття) - Hour(початок_заняття)) AS Кількість_теоретичних_годин
FROM
    Група INNER JOIN Теоретичне_навчання
        ON Група.код = Теоретичне_навчання.код_групи
WHERE
     Year([Теоретичне_навчання.початок_заняття]) = [Введіть рік]
GROUP BY
    Група.назва;
```

А9 Викладачі та середній бал свідоцтв, отриманих їхніми учнями

```
SELECT
    Успішність.ПІБ,
    Avg(Успішність.Середній_бал) AS середня_оцінка
FROM
    (SELECT
        RTrim([Викладач.прізвище]) & " " & RTrim([Викладач.імя]) & " " &
RTrim([Викладач.по_батькові]) AS ПІБ,
        (Свідоцтво.оцінка_теорія + Свідоцтво.оцінка_практика +
Свідоцтво.оцінка_будова_автомобіля) / 3 AS Середній_бал
    FROM
        (((Викладач INNER JOIN Група
            ON Викладач.код = Група.код_викладача_теорії)
                INNER JOIN Група_Учень
                    ON Група.код = Група_Учень.код_групи)
                        INNER JOIN Учень
                            ON Учень.код = Група_Учень.код_учня)
                                INNER JOIN Свідоцтво
                                    ON Учень.код = Свідоцтво.код_учня) AS Успішність
GROUP BY
    Успішність.ПІБ
ORDER BY
    Avg(Успішність.Середній_бал) DESC;
```

А10 Кількість пропусків теоретичних занять учнями

```
SELECT
    RTrim([прізвище]) & " " & RTrim([імя]) & " " & RTrim([по_батькові]) AS ПІБ,
    Count(*) AS кількість_пропусків
FROM
    Учень INNER JOIN Пропуск_заняття
        ON Учень.код = Пропуск_заняття.код_учня
GROUP BY
    RTrim([прізвище]) & " " & RTrim([імя]) & " " & RTrim([по_батькові])
ORDER BY
    Count(*) DESC;
```

ДОДАТОК В
ЗРАЗКИ ЗВІТІВ

Автошкола УченьОК

Склад групи УОК-1

УченьОК

№	П.І.Б учня	Дата народження	Телефон
1	Нікулін Юрій Іванович	21.10.1998	+38(063)275-95-80
2	Леонов Євгеній Федорович	05.03.1998	+38(050)932-42-94
3	Крамаров Савелій Генадійович	07.06.1999	
4	Варлей Наталя Антонівна	11.02.1999	+38(098)634-29-34
5	Смірнов Олексій Євгенійович	18.08.1997	

Кількість учнів у групі: 5

Станом на 17.05.2019 11:56:25

Стр. 1 из 1

Рисунок В.1 – Склад групи

Автошкола УченьОК

Лекції групи УОК-1

УченьОК

№	Тема	Початок заняття	Кінець заняття
1	1 Загальні положення	19.01.2019 10:00:00	19.01.2019 12:00:00
2	9 Попереджувальні сигнали	26.01.2019 10:00:00	26.01.2019 12:00:00
3	12 Швидкість руху	02.02.2019 10:00:00	02.02.2019 12:00:00
4	16 Проїзд перехресть	09.02.2019 10:00:00	09.02.2019 12:00:00
5	21 Перевезення пасажирів	16.02.2019 10:00:00	16.02.2019 12:00:00
6	24 Навчальна їзда	23.02.2019 10:00:00	23.02.2019 12:00:00
7	33 Дорожні знаки	02.03.2019 10:00:00	02.03.2019 12:00:00
8	34 Дорожня розмітка	09.03.2019 18:00:00	09.03.2019 20:00:00
9	35 Основи безпеки руху	16.03.2019 18:00:00	16.03.2019 20:00:00
10	38 Будова автомобіля	21.03.2019 18:00:00	21.03.2019 21:00:00

Загальна кількість лекцій: 10

Станом на 17.05.2019 12:00:46

Стр. 1 из 1

Рисунок В.2 – Лекції групи

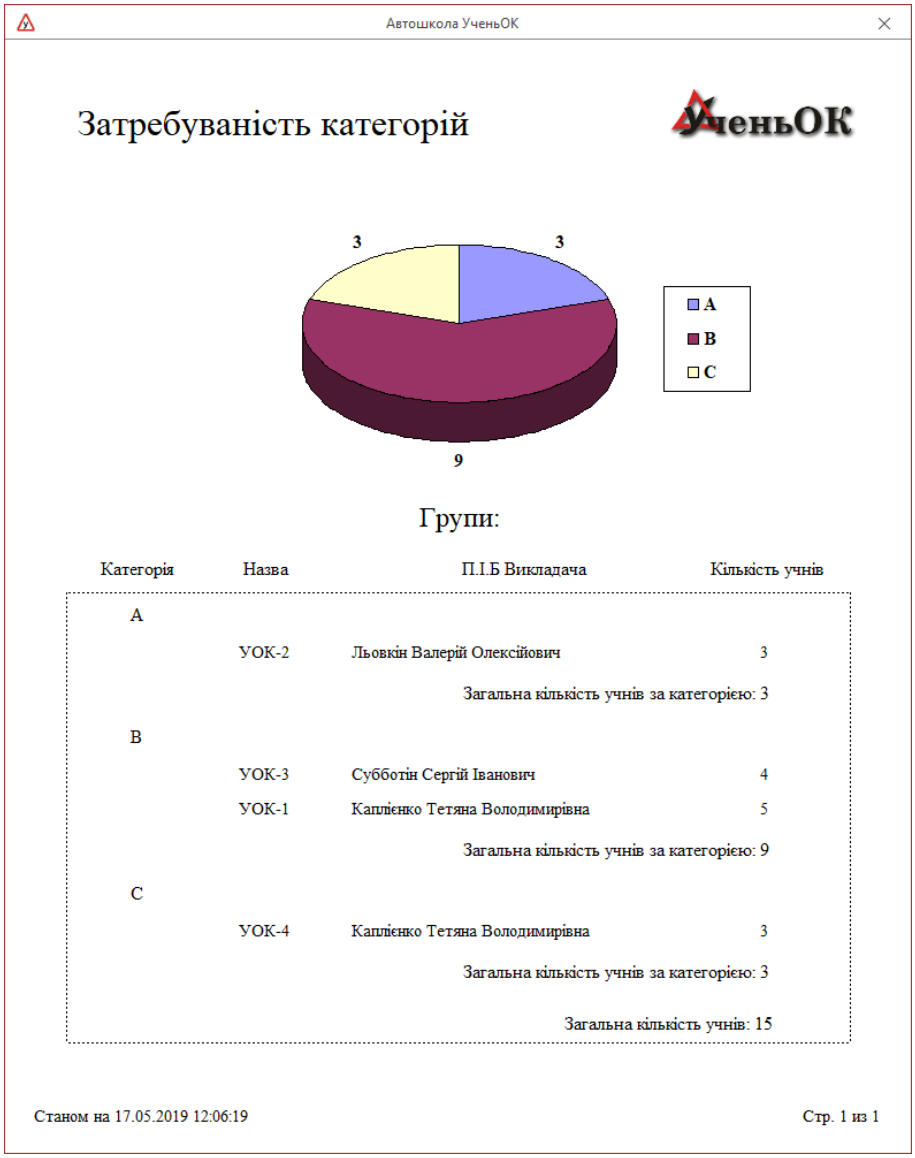

Рисунок В.3 – Затребуваність категорій

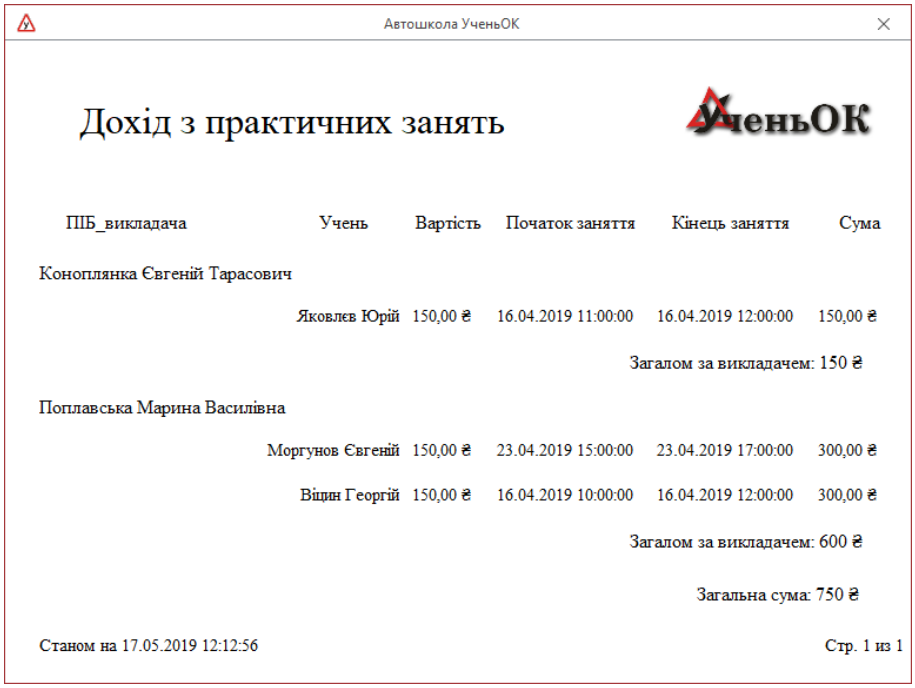

Рисунок В.4 – Дохід з практики