МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до вивчення курсу “Програмні засоби проектування ЕМС» (частина1) та виконання контрольної роботи для студентів спеціальності 141"Електроенергетика, електротехніка та електромеханіка" (освітні програми «Електричні та електронні апарати» та «Електромеханічне обладнання енергоємних виробництв») заочної форми навчання

2018

Методичні вказівки до вивчення курсу «Програмні засоби проектування ЕМС» (частина 1) та виконання контрольної роботи для студентів спеціальності 141 "Електроенергетика, електротехніка та електромеханіка" (освітні програми «Електричні та електронні апарати» та «Електромеханічне обладнання енергоємних виробництв») заочної форми навчання. /Укл.: Л.С. Скрупська. – Запоріжжя: ЗНТУ, 2018. с.66.

Укладачі: Л.С.Скрупська

Рецензент: Л.О.Бондаренко

Відповідальний за випуск: П.Д.Андрієнко.

Затверджено
на засіданні кафедри
“Електричні та електронні апарати”
Протокол № 11 від 1 червня 2018 р.

Рекомендовано до видання НМК
Електротехнічного факультету
протокол № 11 від 21 червня 2018 р.

ЗМІСТ

ЗАГАЛЬНІ МЕТОДИЧНІ ВКАЗІВКИ ТА ЗАВДАННЯ ДО КОНТРОЛЬНОЇ РОБОТИ

Курс «Програмне забезпечення проектування ЕМС», призначений для вивчення потужної системи управління базами даних Microsoft Access, викладається у 5 та 6 семестрах.

На настановній сесії 5 семестру студенти слухають лекції й виконують практичні роботи під керівництвом викладача. Всі лекції курсу й практичні роботи представлені на одній предметній галузі «Інформація про книги бібліотеки університету, про читачів цієї бібліотеки, та про книги, що знаходяться у читачів».

Протягом семестру студенти продовжують самостійне вивчання курсу використовуючи методичні вказівки до самостійної роботи (методичні вказівки № 5706е) та літературу з наведенного списку.

Методичні вказівки до самостійної роботи та до виконання практикуму було підготовлено з метою

- надання необхідної теоретичної інформації з вивчення даного курсу;
- надання студентам необхідної інформації з виконання контрольної роботи;
- надання прикладів на виконання всіх етапів контрольної роботи.

Студенти виконують одну контрольну роботу. Наприкінці семестру здають залік, до якого допускаються студенти, що одержали позитивні рецензії на контрольну роботу.

Ціль контрольної роботи: навчити студентів основним принципам проектування баз даних (БД), а також методології, технології, засобам формування БД у середовищі системи управління базами даними (СУБД) MS Access.

У результаті виконання контрольної роботи студент повинен знати методологію й технологію проектування БД, а також її створення в середовищі реляційної СУБД MSAccess, уміти спроектувати й створити конкретну БД, розробити найпростіші елементи користувальницького додатка.

Етапи виконання роботи

Етап 1. **Проектування та створення реляційної бази даних в СУБД MSAccess.** Питання проектування БД розглядається

викладачем в першій лекції на настановній сесії. Разом зі студентами викладач пректує БД «Бібліотека», з предметною галузі «Інформація про книги бібліотеки університету, про читачів цієї бібліотеки, та про книги, що знаходяться у читачів». Для створення БД «Бібліотека» на комп'ютері в СУБД MSAccess студенти виконують практичну роботу №1.

Етап 2. **Створення запитів на вибірку.** Для виконання цього етапу студенти виконують практичну роботу № 2.

Етап 3. **Створення форм і звітів за допомогою вбудованих у Access механізмів *Мастер форм* і *Мастер отчетов*.** Для виконання цього етапу студенти виконують практичну роботу № 3

Етап 4**. Створення форм із використанням *Конструктора форм*.** Для виконання цього етапу студенти виконують практичну роботу № 4

Етап 5. **Створення запитів на зміну.** Для виконання цього етапу студенти виконують практичну роботу**№ 5.**

Форма звітності передбачає:

– Демонстрацію створеної БД «Library» на екрані комп'ютера.

– Звіту по проведеній роботі в складі:

– схеми даних спроектованої та створеної БД «Бібліотека»;

– звіту по кожній практичній роботі.

Питання до курсу, володіння якими необхідно для виконання контрольної роботи та здачи заліку:

1	Призначення MS Access. Склад системи.
2	Реляційна модель баз даних. Приклади реляційних СУБД.
3	Проектування реляційних баз даних.
4	Види зв'язків між таблицями.
5	Види об'єднання таблиць.
6	Об'єкти бази даних. Їх призначення.
7	Поняття «Забезпечення цілісності даних».
8	Типи даних.
9	Основні властивості полів таблиць: Розмір поля, Формат,
10	Ключеві поля та індекси.
11	Об'єкт: Запит. Типи запитів й їхнє призначення.
12	Призначення кнопок панелі інструментів Конструктора

	Запросов.
13	Створення запитів по кільком таблицям.
14	Створення підсумкових запитів
15	Використання Построителя Выражений.
16	Створення параметричних запитів.
17	Використання функції IIF().
18	Ітогові функції.
19	Створення запитів на видалення інформації.
20	Створення запитів на доповнення інформації.
21	Запит на створення таблиці.
22	Запит на відновлення записів у таблиці
23	Правила створення умови відбору в рядках "Умова" та «Или» Конструктора Запитів.
24	Об’єкт форма, його призначення.
25	Способи створення нової форми.
26	Типи форм.
27	Основні елементи вікна Конструктора Форм.
28	Об’єкт форма. Елементи керування.
29	Вільні, зв’язані та обчислювальні елементи керування.
30	Найбільш важливі властивості форми.
31	Найбільш важливі властивості елементів керування.
32	Використання Майстрів елементів керування.
33	Використання Поля зі списком (Combo Box) для вводу даних в поле.
34	Використання Поля зі списком (Combo Box) для пошуку запису відповідного вибраному в Combo Box значенню.
35	Створення обчислювальних елементів керування.
36	Створення підпорядкованих форм.
37	Використання підпорядкованих форм.
38	Об’єкт звіт, його призначення.

1 ПРАКТИЧНА РОБОТА №1. ПРОЕКТУВАННЯ ТА СТВОРЕННЯ РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ В СУБД MS ACCESS

1.1 Мета роботи

Практичне вивчення СУБД Access на прикладі створення БД "Бібліотека"(по проекту розробленому на лекції):

– створення, редагування та форматування таблиць бази даних у середовищі СКБД Access;
– встановлення зв'язків між таблицями БД;
– створення форм, для вводу інформації до таблиць;
– пошук, сортування та фільтрування даних засобами СУБД Access.

1.2 Загальні відомості про Microsoft Access

Учені запевняють, що зберігання та використання великих обсягів даних , накопичених людством, виправдано тільки за умови, якщо пошук потрібних даних здійснюється швидко і подаються вони в доступній для розуміння формі. Ці умови забезпечують сучасні технології зберігання даних. Основою цих технологій є комп'ютеризовані бази даних (БД).

База даних **–** це впорядкований за певними правилами набір взаємопов'язаних даних.

Наш курс передбачає вивчення реляційної *Системи Управління Базами Даних MS Access* .

Основними «будівельними» блоками Access є:

– таблиці – для зберігання даних;

– запити – для пошуку й витягування тільки необхідних даних за деякими умовами;

– форми для перегляду, додавання й зміни даних у таблицях;

– звіти для аналізу й друку даних у певному вигляді;

– сторінки доступу до даних для перегляду, відновлення й аналізу даних із БД через Інтернет або Інтрамережу.

1.2.1 Початок роботи з Microsoft Access

Для початку роботи з Access необхідно виконати наступне: кнопка «Пуск» → «Все Программы» → «Microsoft Office» →

«Microsoft Access ».

Після даної операції має з'явитися діалогове вікно (рис. 1.1).

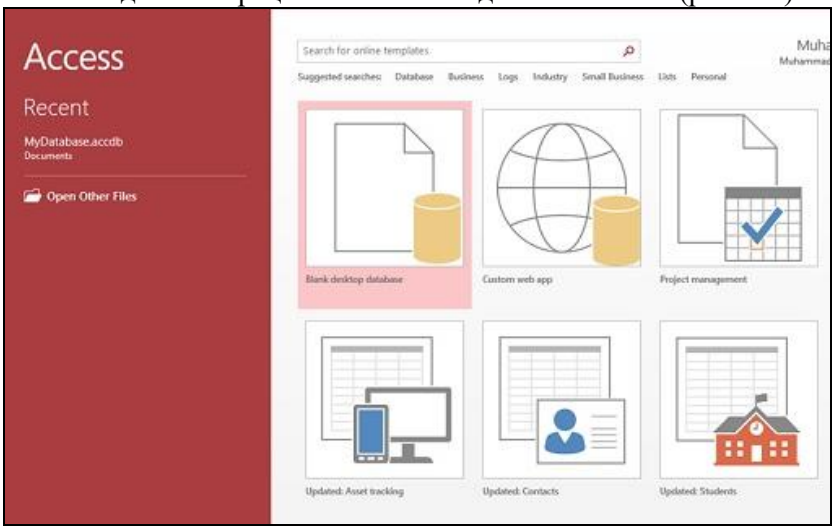

Рисунок 1.1 - Головне вікно Microsoft Access

При створенні нової бази даних з'являється наступне діалогове вікно (рис. 1.2):

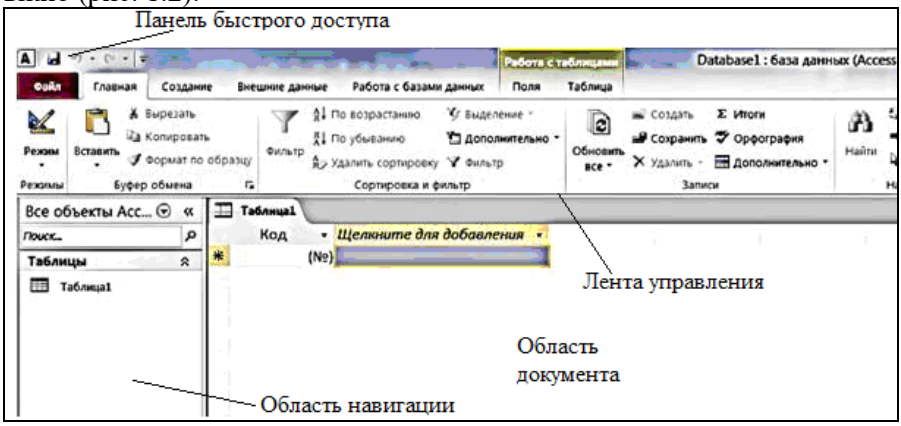

Рисунок 1.2 – Вікно створення Бази Даних

1. **Панель швидкого доступу** (Панель быстрого доступа**)** – забезпечує доступ до команд одним клацанням миші. Набір за замовчуванням включає команди «Сохранение», «Отмена» і «Возврат», при цьому можна налаштувати панель швидкого доступу

для додавання в неї найбільш часто використовуваних команд.

2. **Стрічка (**Лента управления) – багатосторінкова область, розташована у верхній частині головного вікна. Кожна сторінка ***«стрічки»*** містить набір логічно зв'язаних елементів. Деякі сторінки, такі як ***«Головна», «Створення», «Зовнішні дані», «Робота з базами даних»*** доступні весь час роботи з програмою, інші є контекстно-залежними, тобто з'являються тільки при роботі з певними об'єктами.

3. **Подання Backstage** (рис. 1.1) – набір команд на вкладці «Файл» на стрічці, які застосовуються до всієї бази даних. У поданні Backstage можна створювати або відкривати бази даних, публікувати їх в Інтернеті на сервері SharePoint Server і виконувати багато завдань обслуговування файлів і баз даних.

4. **Область навігації** (Область навигации) – область у лівій частині вікна Access, що призначена для роботи з об'єктами бази даних. Область навігації дозволяє організувати об'єкти бази даних і є основним засобом відкриття або зміни об'єктів бази даних.

5. **Рядок стану**– стандартний елемент інтерфейсу користувача, який використовується для відображення повідомлень про стан властивостей, індикаторів ходу виконання і т. д.

Для збереження баз даних **Access** починаючи з версії 2007 використовує як основний формат файлів **ACCDB** (англ. ***Acc****ess* ***D****ata****B****ase* – база даних **Access**), забезпечуючи роботу і з форматами файлів попередніх версій програми **Access 97** – **MDB** (англ. ***M****icrosoft* ***D****ata****B****ase* – база даних **Microsoft**).

1.2.2 Створення нової бази даних

Запустіть Microsoft Access із меню «Пуск» або за допомогою ярлика. З'явиться подання «Backstage» (рис. 1.1). Виконайте одну з наступних процедур:

Створення веб-бази даних

1. У групі «Доступные шаблоны» клацніть елемент «Пустая веб-база данных».

2. Праворуч у розділі «Пустая веб-база данных» у полі «Имя файла» введіть ім'я файла бази даних або використайте надане ім'я.

3. Натисніть кнопку «Создать». Буде створена нова база даних і відкрита нова таблиця в режимі таблиці.

Створення бази даних на комп'ютері

1. У групі «Доступные шаблоны» клацніть елемент «Пустая база данных».

2. Справа в розділі «Пустая база данных» у полі «Имя файла» введіть ім'я файла бази даних або використайте надане ім'я.

3. Натисніть кнопку «Создать». Буде створена нова база даних і відкрита нова таблиця в режимі таблиці.

Створення бази даних зі зразка шаблону

Microsoft Access містить ряд шаблонів. Шаблон Access являє собою готову базу даних з професійно розробленими таблицями, формами й звітами. Шаблони дозволяють швидко пройти початкові етапи створення бази даних.

1. Запустіть Microsoft Access із меню «Пуск» або за допомогою ярлика. З'явиться подання «Backstage» (рис. 1.1).

2. Клацніть елемент «Образцы шаблонов» і перегляньте доступні шаблони.

3. Клацніть необхідний шаблон.

4. У правій частині екрана у вікні «Имя Файла» введіть ім'я файлу або використайте запропоноване ім'я.

5. Натисніть кнопку «Создать». Microsoft Access створить на основі шаблону нову базу даних і відкриє її.

1.3 Теоретичні відомості про основний об’єкт БД - Таблиця

Одним із основних елементів СУБД Access, як і взагалі будь-якої СУБД, є таблиця. Саме в таблицях БД зберігають усю свою інформацію, і робота з новою БД починається зі створення таблиць. Відомо, що таблиця складається зі стовпців і рядків, які є в цілому розграфленою прямокутною областю, в якій пояснювальні надписи є найменуваннями стовпців, а безпосередньо сама інформація розташовується в рядках (рис. 1.3).

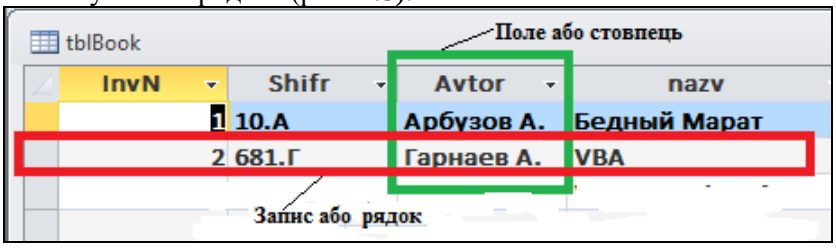

Рисунок 1.3 – Схема структурних елементів таблиці

Говорячи мовою СУБД, кожний рядок таблиці включає дані про

один об'єкт (книжки, читачи, трансформтори). Стовпці таблиці містять різні характеристики цих об'єктів – атрибути (наприклад, найменування книжок, адреси читачив, типовиконання трансформаторів). Рядки таблиці називаються записами; всі записи мають однакову структуру – вони складаються з полів (рис. 1.3), в яких зберігаються атрибути об'єкта. Кожне поле запису містить одну характеристику об'єкта й має чітко визначений тип даних (наприклад, текстовий рядок, число, дата), що визначає тип інформації, що зберігається в полі. Усі записи мають ті самі поля, тільки в них містяться різні значення атрибутів.

1.3.1 Створення таблиць (в режимі «Конструктор»)

В Access використовуються два способи створення таблиць:
– створення пустої таблиці в режимі таблиці;
– створення таблиці в режимі «Конструктор».

Додати нову таблицю в існуючу базу даних можна за допомогою засобів у групі «Таблицы» на вкладці «Создание» (рис.1.4).

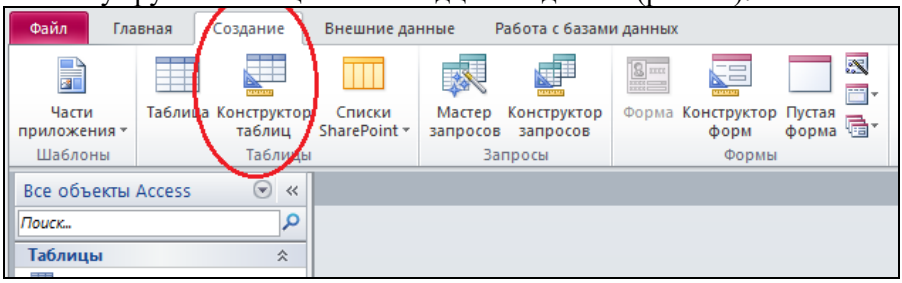

Рисунок 1.4 – Вкладка «Создание»

Для створення нової таблиці можна скористатися кожним із описаних способів. При цьому можна керуватися наступними правилами:

1. У режимі таблиці можна відразу ж приступити до введення даних, дозволивши за допомогою додатка Access сформувати структуру таблиці автоматично. Імена полів задаються номерами («Поле1», «Поле2» і т. д.), а тип даних поля визначається на основі типу введених даних.

2. У режимі «Конструктор» спочатку слід створити структуру нової таблиці. Потім необхідно переключитися в режим таблиці для введення даних або ввести дані іншим способом, наприклад за допомогою форми.

«Конструктор таблиц» призначений для задання й зміни структури таблиці (рис. 1.5). За допомогою конструктора можна формувати таблиці будь-якої складності, з полями будь-якого типу.

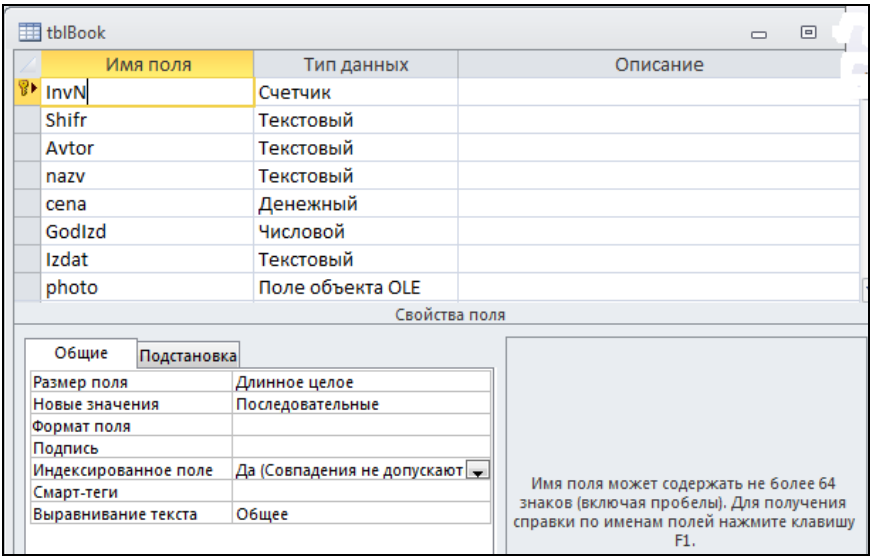

Рисунок 1.5 – Вікно таблиці в режимі «Конструктор»

Створення таблиці в режимі «Конструктор» здійснюється в кілька етапів:

1. Завдання імен полів (стовпців) створюваної таблиці.
2. Визначення типу даних для кожного поля таблиці.
3. Завдання (при необхідності) додаткових властивостей кожного поля, або використання значень властивостей, що установлені за замовчуванням.
4. Завдання ключових полів.
5. Збереження таблиці під обраним ім'ям.

Імена полів

При заданні імен полів таблиць необхідно дотримуватися ряду правил:

1. Імена полів у таблиці не повинні повторюватися, тобто мають бути унікальними.

2. Імена полів можуть містити не більше 64 символів, включаючи пробіли. Бажано заміняти пробіл « » («Назва книжки») в іменах на знак підкреслення «_» («Назва_книжки»), або писати кожне слово з великої букви («НазваКнижки»).

3. Бажано уникати вживання імен полів, що збігаються з іменами вбудованих функцій або властивостей Access (наприклад, Name – ім'я).

4. Ім'я поля не повинне починатися із пробілу або керуючого символу.

5. Імена полів можуть містити будь-які символи, включаючи букви, цифри, пробіли, спеціальні символи, за винятком крапки (.), знаку оклику (!), апострофа (') і квадратних дужок ([,]). Цього ж правила необхідно дотримуватися для імен таблиць, а також інших об'єктів Access.

Тип даних полів

Після введення імені поля необхідно задати тип даних, які будуть знаходитися в цьому полі. Найбільш зручним способом є вибір типу зі списку, що доступний у режимі «Конструктор» (рис. 1.6). В Access є наступні типи даних:

1. «Текстовый» – символьні або числові дані, які не потребують обчислень. Поле даного типу може містити до 255 символів. Розмір текстового поля задається за допомогою властивості «Размер поля», в якому вказується максимальна кількість символів, які можуть бути введені в дане поле.

2. «Поле МЕМО» – призначено для введення текстової інформації, яка по обсягу перевищує 255 символів. Таке поле може містити до 65 535 символів. Поле типу «МЕМО» не може бути ключовим.

3. «Числовой» – числовий тип застосовується для зберігання числових даних, що використовуються в математичних розрахунках. Має багато підтипів. Від вибору підтипу (розміру) даних числового типу залежить точність обчислень. Для установки підтипу числових даних служить властивість «Размер поля». Дані цього типу можуть утримуватися в 1, 2, 4, 8 або 16 байтах. Звичайно за замовчуванням використовується підтип «Длинное целое», що займає 4 байти та являє собою число в межах від –2 147 483 648 до +2 147 483 647. Але, крім цього типу, можна вказати «Байт» – 1 байт, «Целое» – 2 байти, «Одинарное с плавающей точкою» – 4 байти, «Двойное с плавающей

точкою» – 8 байтів, «Код репликации», «Действительное», які можуть містити певний діапазон значень.

4. «Дата/Время» – тип для представлення дати й часу. Дозволяє вводити дати з 100 по 9999 рік. Дані цього типу утримуються в 8 байтах.

5. «Денежный» – тип даних, що призначений для зберігання даних, точність подання яких коливається від 1 до 4 десяткових знаків. Ціла частина даного типу може містити до 15 десяткових знаків. Звичайно використовується для полів, що містять дані про ціни, вартість і т. д.

6. «Счетчик» – поле містить унікальний номер, який Access визначає автоматично для кожного нового запису або випадково, або шляхом збільшення попереднього значення на 1. Значення полів типу «Счетчик» обновляти не можна. Максимальне число записів у таблиці з полем «Счетчик» не повинно перевищувати двох мільярдів. Звичайно використовується для ключових полів (наприклад: «КодКнижки», «НомерКвиткаЧитача» та інше).

7. «Логический» – логічне поле, що може містити тільки два значення, які інтерпретуються як Так/Ні, Істина/Неправда. Поля логічного типу не можуть бути ключовими.

8. «Поле объекта OLE» – містить посилання на OLE-об'єкт (аркуш Microsoft Excel, документ Word, звук, рисунок і т. д.).

9. «Гиперссылка» – дає можливість зберігати в поле посилання, за допомогою якої можна посилатися на довільний фрагмент даних усередині файла або Web-сторінки на тому ж комп'ютері, в інтранет або в Інтернет.

10. «Мастер подстановок» – не є типом ланих. Це інструмент, що служить для створення поля підстановок. Поле підстановок дозволяє вибирати значення поля зі списку, що містить набір постійних значень або значень із іншої таблиці.

Поле підстановки на основі списку підстановки отримує дані з існуючої таблиці або запиту бази даних. При використанні підстановки цього типу таблиці є пов'язаними, і при зміні значень джерела даних поточні дані відразу стають доступні в поле підстановки.

Інформація, сформована майстром підстановок, відображається на вкладці «Подстановка» нижній панелі конструктора таблиць.

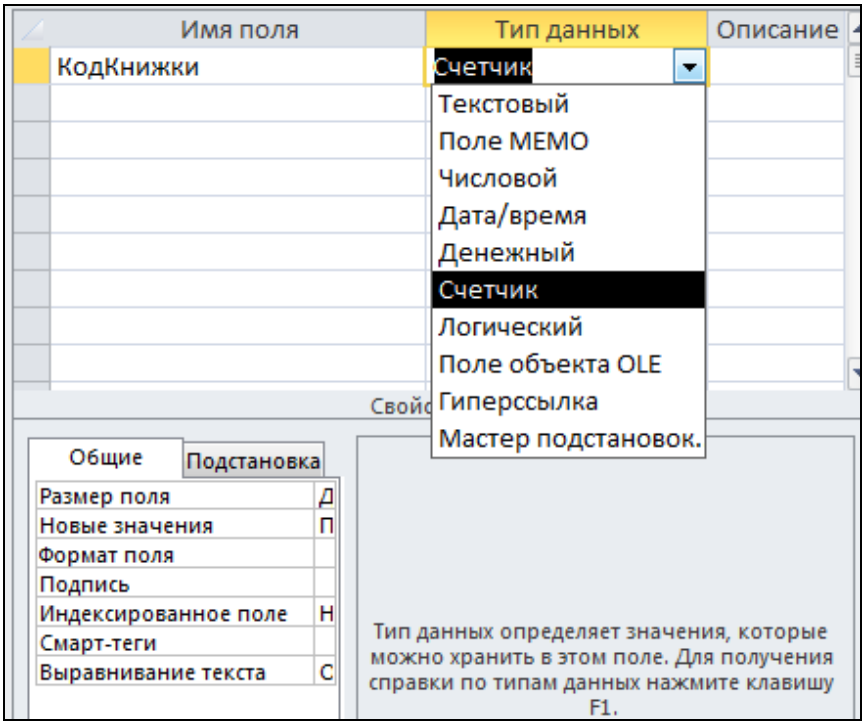

Рисунок 1.6 - Вибір типу поля

Властивості полів таблиці

Властивості полів таблиці використовуються для визначення виду даних, що вводяться й відображаються в полі. «Формат поля» Дозволяє вказати формати виводу тексту, чисел, дат і значень часу на екран і на друк у таблицях, і в пов'язаних із даним полем елементах керування у формах і звітах (якщо для цих елементів керування не встановлений інший формат). Наприклад, задану дату (15 травня 2018 року) можна відобразити (якщо задати певний формат поля) на екрані/при друку різними способами:

– 15 травня 2012 року; – 15.05.18;

– 15.05.18 р.;

– 15-травень-2018 і т. д.

Властивість «Формат поля» визначає тільки спосіб відображення даних, не впливаючи на спосіб їхнього збереження. Значення властивості «Формат поля» залежить від «Типа данных» поля, для якого встановлюється формат.

Для текстового поля визначає максимальну довжину. За замовчуванням дорівнює 50 символів.

Для визначення формату числового поля можуть бути використані стандартні значення властивості (таблиця 1.1).

Таблиця 1.1 – Вбудовані числові формати для числових і грошових типів даних

Значення	**Опис**	**Дані**	**Вид на екрані**
«Основной»	Використовується як значення за замовчуванням. Числа відображаються так, як вони були введені	3456,789$ $ 213,21 –3456,789	3456,789$ $ 213,21 –3456,789
«Денежный»	Число виводиться з роздільниками розрядів і символом національної валюти, що введений у локальних установках	3456,789	3,456,79 грн
«Процентный»	Значення множиться на 100; додається символ відсотка (%)	3 0,45	300 % 45 %
«Экспоненциальный»	Числа виводяться в експонентному вигляді	3456	3,46E+03

Властивість Подпись дозволяє задати заголовок поля при виведенні таблиці на екран, якщо заголовок не заданий, то виводиться ім’я поля.

Властивість «Условия на значения» дозволяють встановити обмеження для значень, що вводяться в поле (наприклад, для поля *Стоимость* можна поставити **<100,** якщо вартість книги не повинна

перевищувати 100 гривень).

Властивість «Сообщение об ошибке» (парне до властивості **Условия на значения)** містить повідомлення користувачу при введенні в поле помилкових значень.

Властивість «Обязательное поле» визначає поле, яке обов'язкове для заповнення. Якщо ця властивість має значення «Да», то при введенні нового запису необхідно ввести значення в це поле або в будь-який приєднаний до нього елемент керування. Порожні («Null») значення в цьому полі не допускаються.

Властивість «Индексирование» визначає індекс для поля. Індекси використовуються для прискорення пошуку. Access підтримує два типові індекси: **Індекс-первинний ключ** (збіги не допускаються) та **Індекс-поле** (збіги допускаються)

Визначення ключових полів

Ключове поле – це одне або кілька полів, комбінація значень яких однозначно визначає кожний запис у таблиці. Якщо для таблиці визначені ключові поля, то Access запобігає дублюванню або введенню порожніх значень у ключове поле. Ключові поля використовуються для швидкого пошуку й зв'язку даних із різних таблиць за допомогою запитів, форм і звітів.

В Access можна виділити три типи ключових полів: лічильник, простий ключ і складений ключ. Розглянемо кожний із цих типів.

1. Для створення ключового поля типу «Счетчик», необхідно в режимі «Конструктор таблиц»:

– включити в таблицю поле лічильника (вказавши для поля тип даних «Счетчик»);

– задати для нього автоматичне збільшення на 1;

– указати це поле в якості ключового, шляхом натискання на кнопку «Ключевое поле» на стрічці «Конструктор таблиц».

Якщо до збереження створеної таблиці ключові поля не були визначені, то при збереженні буде видано повідомлення про створення ключового поля. При натисканні кнопки «Да» буде створено ключове поле «Счетчик» з ім'ям «Код» і типом даних «Счетчик».

2. Для створення простого ключа досить мати поле, що містить унікальні значення (наприклад, коди або номери). Якщо обране поле містить повторювані або порожні значення, його не можна визначити як ключове. Для визначення записів, що містять повторювані дані,

можна виконати запит на пошук повторюваних записів. Якщо усунути повтори шляхом зміни значень неможливо, потрібно або додати в таблицю поле «Счетчик» й зробити його ключовим, або визначити складений ключ.

3. «Составной ключ» необхідний у випадку, якщо неможливо гарантувати унікальність запису за допомогою одного поля. Він являє собою комбінацію декількох полів. Використовується в складних проектах БД.

Увага! Ключеве поле повинно бути призначено **Обязательным** і необхідно вказати, що це поле є **Индексированным** (**Да. Совпадений не допускается**)..

1.3.2 Поняття зв'язків між таблицями

Зв'язки дозволяють установити правила взаємодії між таблицями, завдяки чому часто вдається значно зменшити обсяг будь-якої бази даних, особливо в тих випадках, коли інформація повторюється. Крім того, забезпечується високий рівень захисту від некоректного введення інформації, і з'являється можливість уникнути багатьох проблем, пов'язані зі зміною даних у таблицях.

Зв'язки в реляційних базах даних визначаються за збігом значень полів у різних таблицях. У теорії реляційних СУБД використовуються 3 варіанти зв'язків між двома таблицями (які звичайно називаються відносинами):

1. Один-до-одного (1: 1) – кожному запису першої таблиці відповідає не більше одного запису в другій таблиці й навпаки. Відповідність записів встановлюється в результаті пошуку в полі, що є первинним ключем однієї з таблиць, значення поля, що має назву зовнішнього ключа другої таблиці.

2. Один-до-багатьох (1: N) – первинний ключ таблиці А (поле, що містить унікальні значення), зв'язується зі зовнішнім ключем таблиці В (значення поля можуть повторюватися). При цьому кожному запису першої таблиці може відповідати кілька записів другої.

3. Багато до багатьох (N: M). Однією записи в таблиці А може відповідати кілька записів у таблиці В, а одного запису в таблиці В може відповідати багато записів в таблиці А. Зв'язок багато до багатьох в Access не реалізований і перетворюється на два зв'язку один до багатьох за допомогою таблиці зв'язки.

Для роботи зі зв'язками між таблицями в Access використовується схема даних . Щоб відкрити схему даних, необхідно в групі «Связи» натиснути кнопку «Схема данных» (рис. 1.7).

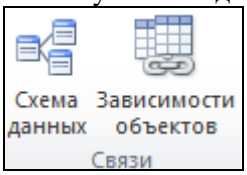

Рисунок 1.7 – Щоб відкрити схему даних, необхідно на вкладці «Работа с Базами Даннях» в групі «Связи» натиснути кнопку «Схема данных»

Забезпечення цілісності даних

В Access забезпечується можливість автоматичної перевірки цілісності даних у зв'язаних полях.

Цілісність даних означає систему правил, що використовуються для підтримки зв'язків між записами у зв'язаних таблицях, а також для забезпечення захисту від випадкового видалення або зміни зв'язаних даних. Установити перевірку цілісності даних можна, якщо виконані наступні умови:

– зв'язане поле головної таблиці є ключовим полем або має унікальний індекс;

– зв'язані поля мають один тип даних;

– обидві таблиці належать одній БД Access.

Забезпечення цілісності даних призводить до обов'язкового дотримання наступних правил:

1. Неможливо ввести у зв'язане поле підлеглої таблиці значення, що відсутнє у зв'язаному полі головної таблиці.

2. Не допускається видалення запису з головної таблиці, якщо існують пов'язані з нею записи в підлеглій таблиці.

3. Неможливо змінити значення ключового поля в головній таблиці, якщо існують записи, що пов'язані з даною таблицею.

Щоб установити ці правила для конкретного зв'язку, після створення, необхідно на схемі даних (рис. 1.8), двічі клацнути на зв'язку (лінії, що з'єднує таблиці), для якого необхідно встановити цілісність даних. У діалоговому вікні (рис. 1.9) задати параметри цілісності, установивши необхідні прапорці.

1. «**Обеспечение целостности данных**» – будь-яка спроба виконати дію, що порушує одне з перерахованих вище правил, призведе до виводу на екран попередження, а сама дія виконана не

буде;

2. «**Каскадное обновление связанных полей**» – при зміні ключового поля головної таблиці автоматично будуть змінені й відповідні значення поля зв'язаних записів.

3. «**Каскадное удаление связанных записей**» – при видаленні запису в головній таблиці видаляються й всі зв'язані записи в підлеглій таблиці.

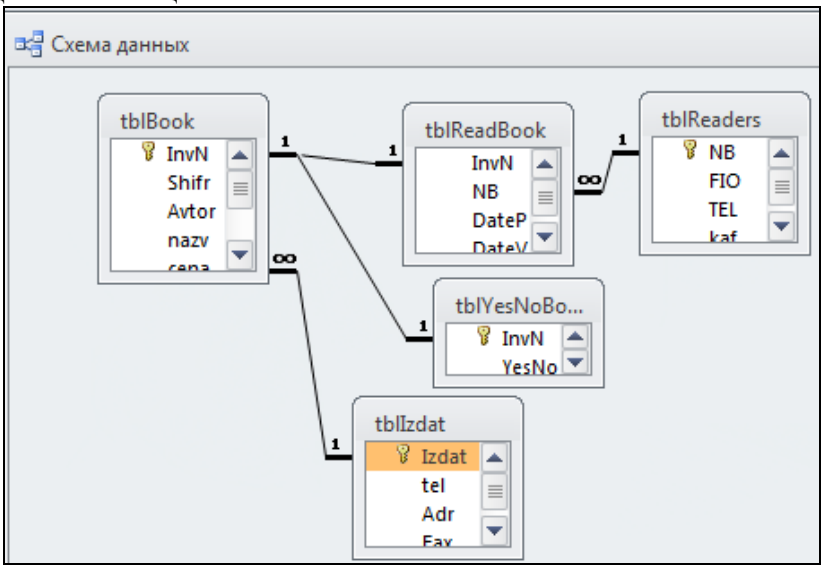

Рисунок 1.8 – Приклад схеми даних БД

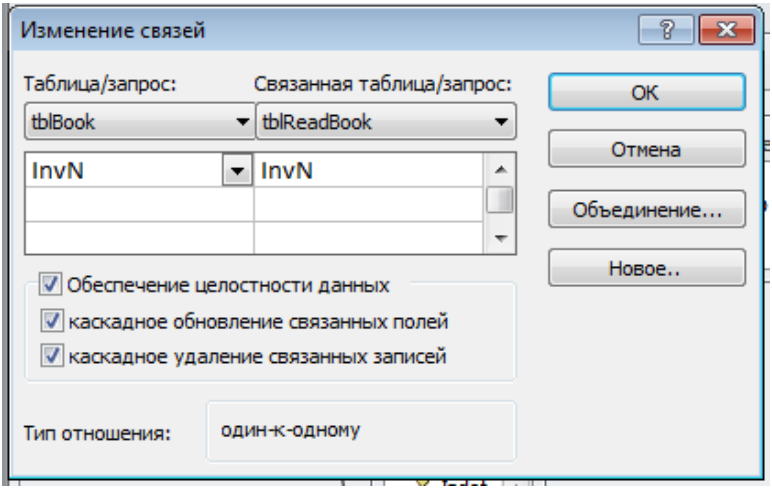

Рисунок 1.9 – Завдання параметрів забезпечення цілісності даних

1.4 Структура БД «Бібліотека»

Перед створенням нової БД обов'язково необхідно добре продумати наступні моменти:

– які дані будуть зберігатися в БД і як їх організувати найкраще (це дозволить визначити, які будуть потрібні таблиці та які зв'язки необхідно організувати між ними);

– **які дії з даними необхідно буде робити** в процесі експлуатації БД (це дозволить визначити, які форми будуть потрібні);

– **які документи необхідно буде виводити на друк** (це дозволить визначити, які звіти будуть потрібні).

Розглянемо структуру бази даних “Бібліотека”, проектування котрої було виконане на лекції:

1. кількість таблиць відповідає кількості об’єктів(сутностей) моделі «сутність–зв’язок». Їх три: **«Книжки»** - містить інформацію про книжки, що зберігаються в бібліотеці; **«ЧитКнижки»** - містить інформацію про книжки, що одержані читачами; **«Читачи»** - містить інформацію про читачів бібліотеки;

2. перелік полів для кожної з таблиць відповідає переліку властивостей для кожного з об’єктів(сутностей). Наведемо перелік полів для зазначених таблиць:

– **Книжки:** Ідентифікаційний номер (IndexBook), Шифр, Автор, Назва книги, Видавництво, Рік, Ціна;
– **Читачи:** № Читацького квитка, Прізвище й ім'я; Телефон, Адреса, Кафедра;
– **ЧитКнижки:** Ідентифікаційний номер книжки, № Читацького квитка, Дата видачі книжки, Дата , якою читач повинен повернути книжку;

3. типи даних для кожного з полів таблиці **Книжки**:
– **Ідентифікаційний номер** - тип даних - *лічильник,* **ключове поле**;
– **Шифр** – тип даних – *текст*, розмір – *20 символів*;
– **Автор** – тип даних – *текст*, розмір – 5*0 символів;*
– **Назва** – тип даних – *текст*, розмір – 7*0 символів;*
– **Видавництво** – тип даних – *текст*, розмір – 2*0 символів;*
– **Рік** – тип даних – *число*, формат *Short Integer* (англ. *Short Integer* – коротке ціле);
– **Ціна** – тип даних – *грошовий*;

4. типи даних для кожного з полів таблиці **Читачи**:
– **Читацький квиток** – тип даних – текст, розмір – 10 символів; **ключове поле**;
– **Прізвище й ім'я** – тип даних – *текст*, розмір – 30 символів;
– **Телефон** - тип даних – *текст*, розмір – 30 символів;
– **Адреса** – тип даних – *текст*, розмір – 50 символів;
–**Кафедра** - – тип даних – *текст*, розмір – 20 символів;

5. типи даних для кожного з полів таблиці **ЧитКнижки** (ключового поля не має):
– **Ідентифікаційний номер книжки** - тип даних –число, довге ціле, індексований, збігу не допускає **(** використати *майстер підстановок*. Цей інструмент передбачає використання даних з таблиці Книжки);
– **Читацький квиток** - тип даних –текст, **індексований, збіг допускає (**використати *майстер підстановок*. Цей інструмент передбачає використання даних з таблиці Читачі);
– **Дата видачі книжки** - тип даних –дата/час;
– **Дата повернення книжки** - - тип даних –дата/час;

4. типи зв'язків між таблицями:

1) таблиця **Книжки** поле **Ідентифікаційний номер** і таблиця **ЧитКнижки** поле **Ідентифікаційний номер** – *зв'язок один до одного*;

2) таблиця **Читачи** поле **№ Читацького квитка** і таблиця **ЧитКнижки** поле **№ Читацького квитка** – *зв'язок один до багатьох*.

Робота з таблицями складається з трьох основних етапів:

1) створення таблиць (опис структури);

2) установлення зв'язку між ними;

3) робота з даними в таблицях: уведення, перегляд, зміна, пошук і т. д.

При створені об'єктів на комп'ютері будемо виходити з наступних **рекомендацій (що до імен цих об'єктів):**

імена об'єктів любого рівня (назва таблиць, назва полів, і т.д.) краще(необов'язково) записувати буквами англійського алфавіту;

імена об'єктів не повинні включати спеціальні символи, крім знаку підкреслення (пробіл допускается, але не рекомендується);

при створенні імен об'єктів вищого рівня (таблиця, запит, форма, звіт) рекомендується починати ім.'я з префіксу, що вказує на приналежність імені об'єкту. Наприклад, для об'єкту Таблиця (Table) префікс – tbl, обєкту Форма (Form) – frm.

Згідно цим рекомендаціям дамо назву таблицям БД «Бібліотека»:

tblBook – імя таблиці Книжки,

tblReadBook – імя таблиці ЧитКнижки

tblReaders – імя таблиці Читачи.

1.5 Послідовність дій по створенню БД «Бібліотека»

1. Відкрийте Access, створіть нову базу даних. Збережіть її під ім'ям «Бібліотека».

2. Створюємо таблиці. У вікні Конструкторі таблиць створіть :

таблицю tblBook,

таблицю tblReaders,

таблицю tblReadBook.

Заповніть структуру кожної таблиці відповідними полями (**дивись розділу 1.4 «Структура БД «Бібліотека»)**. Згідно до змісту і типу полів встановіть потрібні значення властивостей для кожного поля. Неодмінно, для таблиць tblBook і tblReaders призначте ключові поля, для таблиці tblReadBook ключове поле можна не призначати.

Запам’ятайте, що ключові поля повинні бути індексованими й обов’язковими.

3. Встановіть зв’язки між таблицями

Далі потрібно встановити постійні зв’язки між таблицями для того, щоб можна було вибирати дані з декількох таблиць у відповідності зі значеннями збіжних полів. Для цього:

– Клацніть на інструменті **Схема данных** та додайте до вікна схеми даних три створені таблиці. На екрані з’явилося схематичне зображення трьох таблиць;

– Зв’язки між ними встановлюються за допомогою миші за методом Drop and Drag. Зачепіть ключове поле у таблиці **tblBook** і протягніть до такого ж поля в таблиці **tblReadBook**. На схемі з’явиться лінія, що з’єднує ці поля. Аналогічно встановіть зв’язок таблиць **tblReaders і tblReadBook** по полю № Читацького квитка, яке для таблиці **tblReaders** є ключовім полем**;**

– Для кожного зв’язка між таблицями, у вікні **Изменение связей** перевірити **Тип отношения**, який повинен збігатися з типом відношень між цими таблицями вашого проекту БД (дивись розділу 1.3 «Структура БД «Бібліотека»)**;**

– У вікні **Изменение связей** увімкнути три опції: **Обеспечение целостности данных, Каскадное обновление данных, Каскадное изменение записей.**

Получена схема даних повинна мати наступний вигляд(рис. 1.10):

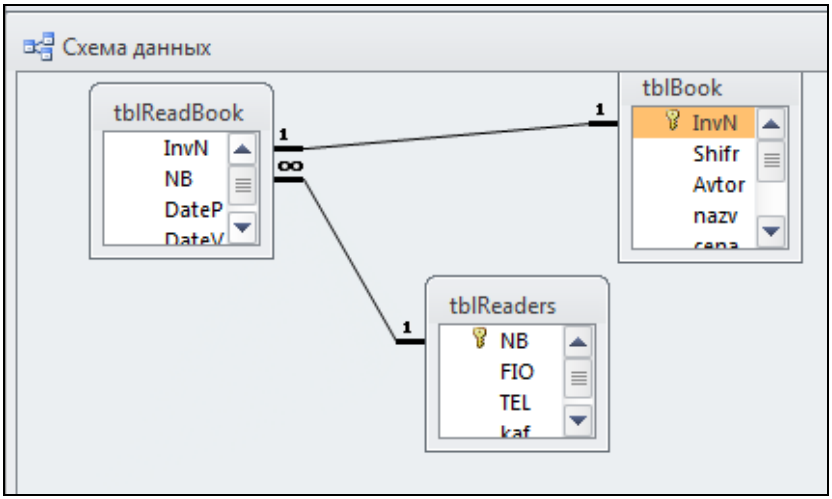

Рисунок 1.10 – Схема данних БД «Бібліотека»

4. Заповніть таблиці даними

Введіть дані в таблиці в такій послідовності:

– спочатку заповніть таблицю **tblBook** інформацією про 10 книжок. Приклад інформації , потрібної для таблиці **tblBook** наведений в табл. 1.2;

– потім заповніть таблицю **tblReaders** даними про читачів (як мінімум 7 читачів);

– В останню чергу заповніть таблицю **tblReadBook.**

Стежте за тим, щоб дані у всіх трьох таблицях були узгоджені, тобто не видавайте книг, котрих немає в бібліотеці.

Стежте за тим, щоб не видавати книги неіснуючим читачам.

Таблиця 1.2 – Приклад інформація для таблиці tblBook

№ п/п	Назва поля	Зміст поля
1	Інв№	1
	Шифр	681.3.06/Г32
	Автор книги	Гарнаев А.Ю.
	Назва книги	Самоучитель VBA

	Місто	Санкт-Петербург
	Видавництво	БХВ
	Рік видання	2010
	Вартість	190
	Анотація	Це посібник з мови програмування, що використо-вується в пакеті Microsoft Access
2	Інв.№	2
	Шифр	618.3/К54
	Автор книги	Камарада, Билл
	Назва книги	Использование Microsoft Word 97: Пер. с англ.
	Місто	Киев
	Видавництво	Издательский дом "Вильямс"
	Рік видання	2017
	Вартість	360
	Анотація	Мета цієї книги зробити вас кваліфікованим користувачем

5. Відкрийте одну з трьох таблиць БД в режимі перегляду та на практиці оволодійте такими інструментами:

– навігацією по таблиці – використовуючи клавіші переміщення курсору, сполучення клавіш (*Ctrl-Pgup, Ctrl-Pgdn, Ctrl-Home, Ctrl-End*), за допомогою миші, використовуючи при необхідності горизонтальну і вертикальну смуги прокручування. Для переміщення по записам служить навігаційна панель, яка розташована ліворуч в нижній частині вікна таблиці;

– вводом та редагуванням даних**;**

– видаленням рядків і стовпців**;**

– зміною порядку проходження стовпців**;**

– пошуком і заміною даних**;**

– сортуванням**;**

– прихованням і відображенням стовпців;

– закріпленням і звільненням стовпців**;**

– підстановкою**;**

– застосуванням фільтрів**.**

1.6 Зміст звіту

1. Опис створенної БД:

– схема БД;

– призначення кожної таблиці наведеної схеми;

– опис встановленого зв’язку для кожної пари зв’язаних таблиць.

2. Стислі відповіді на такі запитання:

– визначте основні поняття: СУБД, база даних, таблиця, поле, запис;

– типи полів та стисла характеристика їх;

– властивості полів;

– ключеві поля;

– що таке зв’язок між таблицями? Типи зв’язків;

– поняття цілісності даних;

– поняття фільтра. Створення і застосування фільтрів;

– пошук та заміна даних в таблиці;

– сортування даних в таблиці.

2 ПРАКТИЧНА РОБОТА №2. СТВОРЕННЯ ЗАПИТІВ ДЛЯ ДОБОРУ ДАНИХ

2.1 Мета роботи

Вивчення прийомів роботи з конструктором запитів на прикладах створення запитів для добору даних з однієї або декількох таблиць.

.

2.2 Загальні відомості про створення запитів для добору даних та приклади їх створення

Access підтримує різні типи запитів: запити на вибірку, групові запити, запити на зміну (видалення, додавання і відновлення записів), підсумкові запити, запити на об'єднання, запити з параметрами.

Запити на вибірку – це найбільш розповсюджений вид запитів. Вони вибирають інформацію (на основі заданих умов відбору записів) з однієї чи декількох таблиць. Завдяки цим запитам можна витягти з всієї інформації БД саме ті дані, що потрібні, і в тому порядку, що необхідний.

Об'єкт Запит– це програма написана на мові **Structure Queries Language (SQL)**. MSAccess дозволяє створювати запити в **Конструкторі запитів**, що значно полегшує роботу користувача. Для перегляду відібраних даних запит потрібно запустити на виконання. Дані, відібрані запитом, доступні користувачу в віртуальній таблиці, тобто в таблиці, яка відкривається в оперативній пам'яті і не зберігається на магнітних носіях.

2.2.1 Створення запиту в Конструкторі запитів

1. Виконати: вкладка «Создание» →група «Запроси» → «Конструктор запросов» (рис. 2.1).

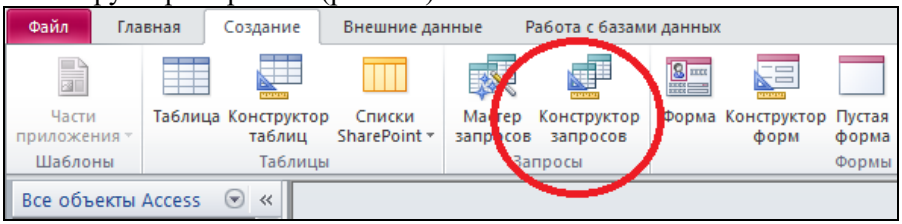

Рисунок 2.1 – Виклик Конструктора запитів

2. Відкривається незаповнений бланк конструктора запиту і діалогове вікно *Добавление таблицы* для вибору об'єктів, на яких буде засновано запит (рия.2.2).

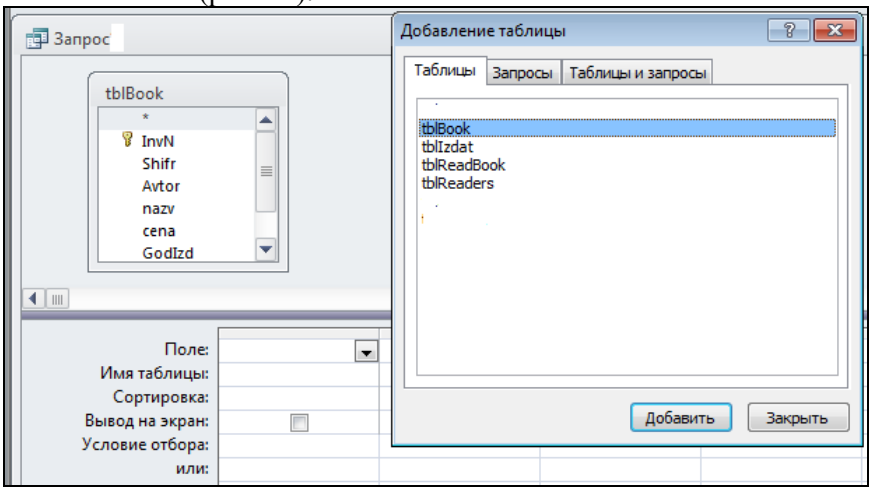

Рисунок 2.2 - Бланк конструктора запиту і діалогове вікно *Добавление таблицы*

3. У діалоговому вікні *Добавление таблицы* вибрати потрібні таблицю/таблиці або запити, які будуть джерелом даних для запита, що створюється.

4. *Бланк запиту* за зразком має дві панелі. На *верхній панелі* розташовані списки полів тих таблиць, на яких ґрунтується запит.

5. У *нижній панелі* визначається структура запиту, тобто структура результуючої таблиці, в якій будуть міститись дані, одержані по результатах запиту.

6. Вибір поля здійснюється одним із таких способів:

–двічі клацнути на потрібному полі у списку полів;

–виділити поля, які необхідно відобразити в таблиці, і при затисненій лівій кнопці миші перенести в бланк запиту;

–для перенесення всіх полів, виділити їх, двічі клацнувши на рядку заголовка, і перетягти;

–натиснути на зірочку в першому рядку і не відпускаючи перенести в бланк запиту. В цьому випадку, в режимі Конструктора не видно кожне поле, але після запуску запиту всі вони будуть вибрані.

7. Клацнувши в рядку *Сортировка*, можна задати сортування результуючої таблиці по вибраному полю.

8. Скориставшись рядком *Условие отбора* можна задати критерій, за яким вибирають записи для включення в результуючу таблицю. По кожному полю можна задати свої критерії.

9. Запуск на виконання створенного запита виконують клацнувши на кнопці *Выполнить* або кнопкою *Режим/Режим таблицы* (рис.2.3). При запуску утворюється результуюча таблиця.

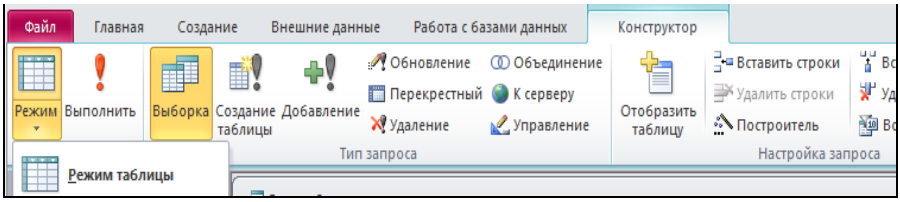

Рисунок 2.3 - Запуск на виконання створенного запита виконують клацнувши на кнопці *Выполнить* або кнопкою *Режим/Режим таблицы*

2.2.2 Задавання умов відбору даних

Насамперед потрібно навчитися записувати умови відбору даних. Без умов дія запиту не має сенсу.

Для **числових полів** умови записують як арифметичні або логічні вирази. Наприклад, «<100» , «>40». В умовах можна використати логічні функції «**And» , «Or», «Not**». Наприклад, умова «>40 And <60» відбирає значення поля, що є більші за 40 і менші за 60.

При формуванні запитів, що мають відбирати значення, перевіряючи їх належність до інтервалу, зручно застосовувати функцію:

«Beetween (НижняяГраница) And (ВерхняяГраница)»

Наприклад, вираз «Beetwen 40 And 60» відбирає значення, що знаходяться в інтервалі від 40 до 60.

При роботі з **текстовими полями** часто виникає потреба перевіряти інформацію на відповідність певному шаблону. Прикладом такого запиту може бути вибір з базі даних всіх осіб, чиє прізвище починається з «КАР».

Таку дію можна реалізувати з використанням функції **«Like“Рядок пошуку із символами шаблона”».** Знак «*» у шаблоні заміняє довільну кількість символів, що можуть стояти на місці даної позиції. Знак шаблону «**?**» заміняє поодинокий символ у позиції, де він знаходиться. Символ «#» вказує, що на його місці має стояти цифра.

Наприклад, умова «**Like "КАР***"» , будучі накладеною на поле прізвищ відбирає з бази даних записи, чиє прізвище починається з «КАР».

Докладне описання умов відбору даних дивись в методичних вказівках до самостійної роботи.

Приклад 1. **Запит, що відбирає дані з однієї таблиці за умовою.**

Побудуємо запит, що відбирає книжки любого автора для вивчення язика програмування VBA. Начебто, є така книга за авторством Горнаева (чи може Гарнаева?). Формулюємо критерії відбору:

Потрібна книжка в назві котрої присутнє слово VBA;

Прізвище автора любе, вдовольняє також Г**а**рнаев або Г**о**рнаев;

Видані с 2000 до 2015 року;

Найдену інформацію вивести в порядку убування року видання.

Створення запиту

Відкриваємо Конструктор для створення нового запиту.

У вікні діалогу «**Додавання таблиці**» на вкладці «**Таблиці**» вибираємо таблицю, з якої потрібно відібрати дані. У нашому випадку такою таблицею є «**tblBook**». Натисніть кнопку «**Добавить**». Закрийте вікно «Добавление таблицы».

Виберіть поля, які треба включити до запиту, а саме: «Автор», «Назва», «Год видавництва». Щоб помістити їх до бланку запиту, двічі клацніть кнопкою миші на імені поля у таблиці. Вибрати поле таблиці можна безпосередньо у вікні запиту, вибравши їх з випадаючого списку.

Записуємо критерії відбору. Всі критерії відбору записані в одному рядку об'єднуються в один логічний вираз за допомогою операнда **AND**, в різних рядках - за допомогою операнда **OR**. Тому маємо два варіанта виконання створення уловного виразу для відбору потрібної інформації:

1 варіант.У рядку «**Условия отбора**»:

– в полі «Автор» ставимо умову «Like "Г?рнаев" OR Like"*"»,

– в полі «Назва» - «Like "*VBA*"»,

– в полі «Год видавництва» - «Between 2000 AND 2015»,

2 варіант. У рядку «**Условия отбора**»:

– в полі «Автор» ставимо умову «Like "Г?рнаев"»,

– в полі «Назва» - «Like "*VBA*"»,

– в полі «Год видавництва» - «Between 2000 AND 2015»,
У рядку **ИЛИ**
– в полі «Назва» - «Like “*VBA*”»,
– в полі «Год видавництва» - «Between 2000 AND 2015»,

У рядку **Сортировка** встановіть ознаки сортування за полем **Рік видання,** виберіть значення “**По убыванию**”. Якщо ви сортуєте за декількома полями, то Access завжди починає сортування з крайнього ліворуч поля;

Щоб переглянути результат натисніть кнопку «!» або виберіть пункт меню «**Вид→Режим таблицы**».

2.2.3 Підсумкові запити

Щоб вибрати не окремі записи, а підсумкові значення для певної групи даних (наприклад, кількість книг по інформатиці, по кожному видавництву, кількість читачів по кафедрах тощо), треба скористатись підсумковими запитами. Таким запитам треба вказати характер групування даних і тип групової операції. Щоб конкретизувати характер групування входимо до «**Груповые операции → Группировка**». Це можна зробити так:

- ввійти до режиму **Конструктора» запиту**;
- сформувати поля, що будуть складати запит;
- натиснути на піктограму, що знаходиться на панелі інструментів (виглядає як значок «**Σ**»). Замість піктограми можна клацнути правою кнопкою миші й вибрати з контекстного меню пункт «**Групповые операции**».

Access додасть рядок з іменем «**Групповая операция**». Щоб створити групу записів з однаковими значеннями по деякому полю, треба в рядку «**Групповая операция**» цього поля ввести «**Группировка**». Access вибере множину записів з однаковим значенням поля групування й підрахує для них підсумки. Такі дії Access повторить для кожної групи. Спосіб рахування підсумків вказують у тому ж рядку «**Групповая операция**». Для цього переходимо до поля, по якому планують визначити підсумкове значення. Клацаємо лівою кнопкою миші на списку «**Группировка**». Список розкриється, з нього вибираємо тип операції для одержання підсумків. Список операцій має такі функції:

SUM – обчислення суми значень для групи;
AVG – середнє значення поля для даних із групи записів;

MIN – мінімальне значення для даних із групи записів;
MAX – максимальне значення для даних із групи записів;
COUNT – кількість записів, у яких є значення із групи;
STDEV – стандартне відхилення;
VAR – дисперсія;
FIRST – значення в першому записі групи;
LAST – значення в останньому записі групи.

Приклад 2. Продемонструємо процес будування підсумкового звіту на такому прикладі: порахуємо кількість читачів з кожної кафедри. Спочатку будуємо звичайний запит у режимі «Конструктора», додаємо до нього таблицю «tblReaders». Включаємо до результуючого звіту поля «Кафедра» і «Прізвище». Натискаємо на панелі інструментів піктограму «Групповые операции». У полі «Кафедра» вказуємо «Группировка». Переходимо до поля «Фамилия». У рядку «группировка» для цього поля вибираємо функцію «Count» .

Важливо! Щоб при перегляді запиту поля мали зрозумілу назву, треба скоригувати їх властивості. Для цього ставимо курсор на колонку поля у вікні «Конструктора запитів», входимо до меню «Вид→Свойства». У полі «Подпись» друкуємо заголовок.

Перехресний запит

Це особливий тип підсумкового запиту, який дозволяє організувати підсумкові значення у таблицю, що нагадує зведену таблицю в Excel. Головною перевагою перехресного запиту є те, що його можна будувати по двом ключовим полям. Таким чином, перехресний запит може трансформувати лінійну таблицю у таблицю з двома вимірами.

Наприклад, побудуємо запит для визначення сумарної вартості книжок для різних років видання по кожному видавництву. Найзручніший спосіб будування перехресних запитів — скористатися кнопкою «**Создать**» у вікні роботи із запитами . Натискаємо її. На екрані з’явиться вікно з переліком типів запитів. Вибираємо «**Перекрестный запрос**».

Кожний етап діалогу щодо будування перехресного запиту представлено в окремому вікні. По черзі вибираємо таблицю, назви рядків, назви стовпців і функцію обробки даних, кожного разу натискаючи кнопку «Далее». У нашому випадку вказуємо ім’я

таблиці «Книги», найменування рядків «Год»; найменування колонок «Издательство», для поля «Стоимость» вказуємо функцію обробки даних «Sum». Встановимо прапорець «Итоговое значение по срокам» в активне положення. Закінчивши формування запиту, натискаємо на кнопку «Виконати».

2.2.4 Запити, що відбирають дані із декількох таблиць

Для формування таких запитів треба включити до верхньої частини вікна «Конструктора запитів» декілька таблиць. Якщо між таблицями існують зв'язки (було побудовано «схему даних»), вони відобразяться у цьому вікні автоматично. Тепер можна включати до запиту поля із декількох таблиць.

Існує два різновиди об'єднання двох таблиць: «внутрішнє (inner join)» і «зовнішнє(left join та right join)». Якщо об'єднання внутрішнє (саме його ви встановили), то у запиті будуть об'єднані записи, у яких співпадають значення полів, що зв'язані. Якщо будь-який запис в одній таблиці не має відповідного запису в іншій таблиці, він не буде включений до результуючого звіту. Зовнішнє об'єднання ми розглянемо пізніше.

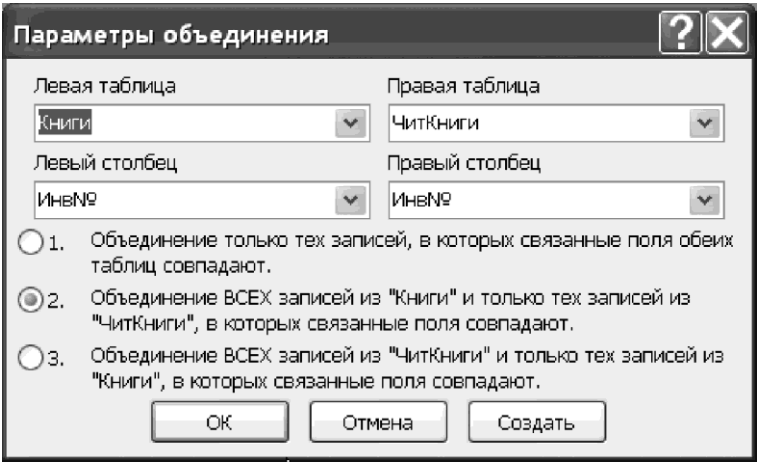

Рисунок 2.4 – Вікно вибору виду об'єднання між таблицями

Приклад 3. Запит, що потребую внутрішнього об'єднання. Побудуємо запит, що знаходить список книг, які видані на руки читачам. До запиту включимо дві таблиці «tblBook» і «tblReadBook».

Нам знадобляться поля «Інвентарний номер», «Шифр», «Автор», «Назва» з таблиці «tblBook. Знайдемо ті записи, які є в таблиці «tblBook» і присутні в таблиці «tblRB».

Для відбору тільки таких книжок (записів), у котрих поля Інвентарний номер в обох таблицях мають одні і ті ж значення, потрібно для цих таблиць встановити **Внутрішнє об'єднання (Inner Join)(рис.2.4).** Цей вид об'єднання встановлюється при створенні Схеми даних «по умолчанию», тому для виконання завдання достатньо в Конструктор запитів внести список потрібних полів і запустити запит на виконання.

Формування запитів типу «Записи без підлеглих»

Ці запити відбирають записи, що не зв'язані із записами в іншому списку. Для роботи з такими запитами треба змінити **тип об'єднання таблиць на «зовнішній»**. При «зовнішньому» зв'язуванні до запиту обов'язково потраплять всі записи із першої таблиці. Записи з другої таблиці будуть додані, коли значення зв'язаних полів співпадають. Якщо в другій таблиці такого запису немає, то до запису з першої таблиці додаються поля з порожніми значеннями. У конструкторі таблиць зовнішнє зв'язування зображається лінією зі стрілкою.

Приклад 4. Для прикладу побудуємо запит, що знаходить список книг, які не видані на руки читачам. До запиту включимо дві таблиці «tblBook» і «tblReadBook». Нам знадобляться поля «Інвентарний номер», «Шифр», «Автор», «Назва» з таблиці «tblBook» і поле «Інвентарний номер» з таблиці «tblReadBook». Знайдемо ті записи, які є в таблиці «tblBook» і відсутні в таблиці «tblReadBook». Змінимо параметри об'єднання таблиць. Ставимо курсор на лінію зв'язку між таблицями, клацаємо правою кнопкою миші й обираємо з контекстного меню пункт «Параметры объединения». Із запропонованих варіантів обираємо «Объединение ВСЕХ записей из «tblBook» и только тех записей из «tblReadBook», в которых связанные поля совпадают».(рис. 2.2).

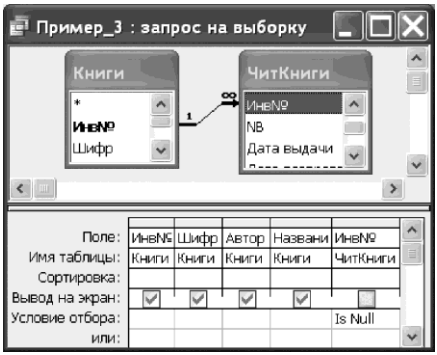

Рисунок 2.5 - Запит, що знаходить список книг, які не видані на руки читачам

Ми створили **Ліве зовнішнє об'єднання (Left join),** до якого включені всі записи з таблиці «tblBook». Для книг, що не мають записів у таблиці «tblReadBook» встановлюється значення «**Null**». Тому для поля «Інвентарний номер» з таблиці «tblReadBook» встановимо параметр «**Условие отбора**» в значення «**Is Null**» і знімемо прапорець «**Вывод на экран**». Параметри запиту показано на рис. 2.5. Виконуємо запит.

2.2.5 Створення полів для обчислень

На відміну від електронних таблиць у таблицях баз даних ніколи не зберігають інформацію, яку можна одержати в результаті обчислень. Це пов’язано з тим, що здебільшого такий підхід вимагає значних витрат пам'яті й сповільнює пошук і обробку даних. Тому всі необхідні обчислення виконують у запитах, створюючи для цього спеціальні поля. Щоб створити таке поле у вільному стовпці конструктора запитів записують нове ім'я, знак “:”, а потім вираз, що обчислює необхідне значення. У виразі можна використати знаки операцій, функції, звертатися до значень інших полів. Наприклад, щоб обчислити вартість замовлення, маючи значення «СтоимостьЕдиницы» і «КоличествоЕдиниц» треба у новому стовпці записати формулу

«СтоимостьЗаказа:[СтоимостьЕдиницы]*[КоличествоЕдиниц]».

Імена полів бази даних записують у квадратних дужках. У формулах можна використати вбудовані функції Access. Ось деякі з них:

iif(умова; вираз1; вираз2) — якщо вираз «умова» виконується ,

функція обчислює значення «вираз1», якщо умова не виконується — обчислює значення «вираз2». Функція iif() подібна до ЕСЛИ() в Excel.

DateDiff(“d”; дата1; дата2) — знаходить різницю між двома датами, результат представляє у днях. Якщо перший аргумент дорівнює «m», різницю буде представлено у місяцях, якщо він дорівнює “y” — у роках.

Повний перелік функцій та їх параметрів можна побачити у довідковій системі або у вікні «**Построитель выражений**». «Построитель виражений» — це спеціальний інструмент, що суттєво полегшує набір складних формул. Оскільки основні складові формули при використанні «Построителя…» набираються автоматично, використання цього інструменту суттєво знижує кількість помилок.

Приклад 5. Ми проілюструємо використання обчислювальних полів на запиті, що відбирає й поєднує записи з двох таблиць: «tblBook» і «tblReadBook». Мета запиту — показати, які книги були видані читачам. У запиті створимо поле для обчислення з іменем «Пеня». У цьому полі буде нараховуватись пеня на кожну книгу, яка не булла повернута вчасно. Розмір пені дорівнює 1% від вартості книги за кожний прострочений день.

Переходимо на закладку «Запросы». Натискаємо кнопку «Создать». У вікні, що з’явиться, вибираємо варіант «Конструктор». Це означає, що параметри запиту будемо формувати в режимі «Конструктора…». Додаємо до запиту таблиці «tblBook» й «tblReadBook». Оскільки для вказаних таблиць було створено схему даних, її буде відображено у верхній частині вікна запиту. Якщо схеми даних не існує, створіть її. Це можна зробити прямо у вікні запиту. Нагадаємо, що ми маємо зв’язати таблиці за інвентарним номером книги (поле «Інвентарний номер»). Тип зв’язку таблиць «tblBook» та «tblReadBook» — «один до багатьох».

До запиту включаємо такі поля: «Автор», «Название», «Стоимость», «Інвентарний номер», «Дата выдачи», «Дата возврата», «NB». Останнє поле нам знадобиться для організації зв’язку з «tblReaders». Зберігаємо запит з іменем «**List1**». У першому вільному стовпчику нижньої частини вікна створіть поле, що обчислює пеню, з ім'ям «**F**». Для цього наберіть у верхньому рядку (де розташовано ім'я поля) такий текст:

F: IIf([ДатаВозврата]<Date();DateDiff("d";[ДатаВозврата];Date())*0,01*[Цена];0)

Для створення виразу можна використати «**Построитель**

выражений». Натискаємо кнопку «Построить» на панелі інструментів. Відкриється вікно «Построителя…». У лівій частині вікна перелічені доступні об'єкти Access для будування формули: «Таблицы», «Запроси», «Формы», «Функції», «Отчеты» тощо. Якщо ліворуч від назви об'єкту стоїть позначка «+»(«плюс»), вказаний об'єкт має багато значень. Розкриваючи потрібні закладки, ви одержите доступ до окремих елементів поточної бази даних, з яких можна побудувати формулу. Наприклад, розкриваючи таблиці або звіти, ви можете дістатись до полів, що до них входять. Натискаючи кнопку «Вставить» об'єкт буде перенесено до спеціального вікна, де будується формула. Лишається розставити знаки операцій поміж об'єктами і завершити формулу. Детальне опанування «Посторителя выражений» ми лишаємо для самостійного опрацювання. Збережіть запит з іменем «List1». Виконайте запит і перегляньте результат. Розмір пені, зрозуміло, залежить від поточної дати. Спробуйте змінити дату повернення книги. Переконайтесь, що сума пені також змінилась.

Приклад 6. Спробуємо обчислити для любого читача (тобто призвище читача буде параметром). е кількість взятих ним книг, їх загальну вартість і пеню, що нарахована.

Для цього сформуємо новий запит. Включаємо до нього таблиці «tblReaders» і запит «List1». З таблиці «tblReaders» до запиту включаємо поля «Фамилия» із запиту «List1» — поля «Інвентарний номер», «Стоимость», поле «Пеня» (Назва цього поля – **«F»)**.

На панелі інструментів вибираємо піктограму «**Групповые операции**», у бланку запиту з'являється **рядок «Групповая операция»**.

В рядку **«Групповая операция»** для поля «Фамилия» вибираємо «**Группировка**», для поля «Стоимость» операцію підрахунку суми – «**SUM**», для поля «Инв№» — операція обчислення кількості «**Count**» і для поля «Пеня» — операцію «**SUM**».

В рядку **Условия отбора** для поля «Фамилия» вводим параметр: **[Введіть призвище читача]**.

Щоб результат мав придатний вигляд, поставимо зрозумілі підписи для кожної колонки. Для цього треба ввійти у коригування властивостей полів (перейти до режиму «Конструктора…», встановити курсор на поле, натиснути праву кнопку миші й вибрати з

контекстного меню «**Свойства**»). У переліку «Свойства» треба поставити зрозумілий текст у параметрі «**Подпись**

2.3 Завдання на самостійну роботу

1. Виведіть список читачів каф. ЕА, на руках у яких є бібліотечні книги**.**

2. Виведіть список читачів каф. ЕА, на руках у яких, в даний час, не має бібліотечних книжок**.**

3. Сконструйований запит перетворіть у такий, котрий виводив би всіх читачів бібліотеки поза залежністю від того, чи користуються вони послугами бібліотеки в даний час. Для читачів, що мають книги з бібліотеки, вивести «Інвентарний номер» книги.

4. В отриманому запиті, змінюючи умови добору, перегляньте інформацію:

– за заданою кафедрою і за датою видачі книги,

– за заданими двома кафедрами,

– за боржниками з відсортованою інформацією по даті повернення книги в бібліотеку.

5. Виведіть список читачів і автора, і назву книжок, що отримані читачами в бібліотеці (запит до трьох таблиць).

6. Кількість читачів по кожній кафедрі, затримавших книжки більше ніж на 3 місяця.

7. Створіть запит з ім’ям **quList1** для вибору списку книг, що отримані читачами в бібліотеці.

В списку повинна виводитись інформація про пеню, котру повинен сплатити бібліотеці читач в разі затримки книжки. Цей запит буде використовуватись в наступній лабораторній роботі як джерело даних при створенні підпорядкованої форми.

2.4 Зміст звіту

Для створених запитів з розділу «Загальні відомості про створення запитів на вибірку та приклади їх створення» (5 запитів) і «Для самостійного виконання» (7 запитів) занотовувати:

а) зміст запиту;

б) запит в режимі Конструктора;

в) результат виконання запиту.

3 ПРАКТИЧНА РОБОТА №3. СТВОРЕННЯ ФОРМ І ЗВІТІВ ЗА ДОПОМОГОЮ МАЙСТРІВ

3.1 Мета роботи

Формування вмінь та навичок створення звичайних, підпорядкованих форм та звітів за допомогою майстра для корегування та аналізу даних. Усвідомлення ролі зв'язків між таблицями при створенні складних форм.

3.2 Загальні відомості

Досі були вивчені об'єкти баз даних, що кінцевому користувачу не видні, і, як правило, він навіть не знає про їхнє існування. Користувач взаємодіє винятково з інтерфейсом, створеним для нього розробником бази даних. Одним з основних елементів інтерфейсу є форма. За допомогою форм можна організувати доступ до всіх інших об'єктів бази даних: таблиць, запитів, інших форм, звітів, контекстних меню і т.д. Звіти багато в чому схожі з формами, тому що звіт – це об'єкт бази даних, призначений для ефективного представлення даних у друкованій формі. Створити нову форму чи новий звіт, як і будь-який об'єкт бази даних можна декількома способами.

Способи створення форм

Форми створюються з набору окремих елементів керування: текстових полів для введення і редагування даних, кнопок, перемикачів, списків, підписів полів, а також рамок об'єктів для відображення графіки та об'єктів OLE.

Створити форму в базі даних Access можна кількома способами, що пропонуються на вкладці ***Створення*** стрічки у групі ***Форми*** (рис. 3.1).

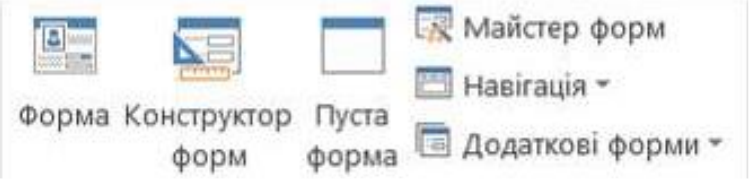

Рисунок 3.1 – Інструменти для створення форм

Щоб швидко створити форму, треба вибрати таблицю або запит в області навігації і на основі цього об'єкта створити форму за допомогою команди ***Форма.***

Використання ***Майстра форм*** є швидким і зручним способом створення різноманітних за складністю форм, з можливістю формування рівнів групування.

Самостійне створення форми здійснюється за допомогою ***Конструктора форм*** або ***Пустої форми***. При цьому розробник може використовувати елементи керування (кнопки, текстові вікна тощо), підключати макроси або писати програмні модулі VBA.

Команда ***Додаткої форми*** дозволяє створення різних варіантів форм фактично в найпростішому режимі Автоформа:

Форма кількох елементів (також називається стрічковою) дозволяє відображати інформацію відразу з декількох записів. Ця форма подібна до таблиці, але надає більше можливостей з керування елементами, наприклад форматування тексту, додавання графічних об'єктів, кнопок та інших елементів керування.

Таблиця – форма, що відображає кілька записів у таблиці по одному запису в рядку.

Розділена форма відображає дані у режимі таблиці і форми одночасно. Найчастіше користувачеві бази даних з інтерфейсом у вигляді форми потрібні додаткові засоби для швидкого пошуку необхідних даних (наприклад, сортування даних по полях, фільтрація даних.

Форма навігації являє собою форму, що містить елемент навігації. Форми навігації особливо важливі для навігації в БД, що будуть опубліковані в Інтернеті, оскільки область переходів Access не відображається в браузері.

Форми можуть виводитись на екран в трьох режимах: ***Режим форми, Режим розмітки*** та ***Конструктор***. Для переходу з одного режиму до іншого використовуються команди групи ***Подання*** або кнопки-піктограми у правому нижньому куті вікна форми .

Режим розмітки дозволяє вносити змінення у форму і при цьому дані можна переглядати, що дуже зручно, якщо треба відрегулювати розмір елементів керування або внести інші змінення у структуру, що впливатимуть на зовнішній вигляд форми і зручність

роботи з нею. Режим ***Конструктор*** дозволяє вносити змінення, які вимагають детального розгляду структури форми.

3.3 Порядок виконання лабораторної роботи

Виконайте нижче приведені завдання, занотуйте в звіті макети створених форм та прокоментуйте особливість використаного кожного інструменту для створення форми та полученого результату.

3.3.1 Створення форм для вводу та перегляду даних

Завдання 1. Створіть форми для перегляду даних у таблиці *tblBook* використовуючи наступні інструменти :

Форма,
Додаткові форми/ Форма нескольких элементов,
Додаткові форми/ Таблица,
Додаткові форми/ Разделенная форма

Завдання 2. Створіть форму для перегляду даних у таблиці *tblBook* за допомогою ***Майстра форм.***

Майстер форм дозволяє при створенні форми вибирати певні поля з однієї чи декількох взаємопов'язаних таблиць і запитів для відображення на формі. У ***Майстрі*** можна задавати рівні групування та сортування даних, вибирати макет форми.

Для запуску майстра форм треба на вкладці ***Створення*** у групі ***Форми*** натиснути кнопку ***Майстер***. Далі слідувати інструкціям на сторінках ***Майстра форм***.

У *Майстрі форм* можна здобути усілякі результати залежно від вибраних параметрів. Тому рекомендується запустити майстер кілька разів, поперемінно експериментуючи з параметрами, поки не буде здобуто потрібний результат. Окрім того, у *Майстрі* можна задавати параметри групування і сортування даних.

Особливість використання *Майстра форм* полягає в тому, що всі поля, які були вибрані для форми, будуть оформлені без участі розробника. Але при бажанні можна перейти до режиму *Конструктор* і відкоригувати зовнішній вигляд форми. Поза жодним сумнівом, зручність використання *Майстра форм* полягає у швидкості створення форми.

3.3.2 Створення форм з підпорядкованою формою за допомогою ***Майстра форм.***

При роботі з реляційними даними (тобто коли пов'язані один з одним дані зберігаються в окремих таблицях) нерідко потрібно переглянути дані з декількох таблиць або запитів на одній формі. Наприклад, потрібно переглянути запис клієнта з однієї таблиці та відомості про його замовлення з іншої. Підфор-

ми (підпорядковані форми) – зручний інструмент для подібних завдань, і в

Access їх можна швидко створити декількома способами.

Підформа – це форма, вставлена в іншу форму. Первинна форма називається головною формою. Комбінація з форми та підформи іноді називається ієрархічною формою, формою зі зв’язком "головний-другорядний" або формою зі зв’язком "батьківський-дочірній".

Підформи особливо ефективні, коли потрібно відобразити дані з таблиць або запитів зі зв’язком "один-до-багатьох".

Головна форма та підформа пов’язані таким чином, що у підформі відображаються лише записи, пов’язані з поточним записом у головній формі.

Виконайте нижче приведені завдання, занотуйте в звіті макети створених форм та прокоментуйте особливість використаного кожного інструменту для створення форми та полученого результату.

Завдання 3. Створіть форму, що для кожного читача виводить список книг, які у нього на руках. При цьому вказується автор книги, назва книги, дата видачі і дата повернення. Крім того, створіть поле, що обчислюється. Дайте йому назву – **Пеня**. Пеня нараховується в розмірі одного відсотка від вартості книги за кожний прострочений день. Підрахуйте також загальну суму пені для кожного читача. Форма буде мати підпорядковану форму зі списком книг читача. Дані в підпорядковану форму будуть братися з запиту **quList1**, який вже створено в попередній лабораторній роботі.

Початковий макет форми створіть за допомогою майстра форм, а потім поліпшіть його за допомогою конструктора.

Виконайте такі операції для створення форми майстром форм:

а) виберіть команду ***Создать*** групу ***Формы*** і клацніть на кнопці ***Майстер;***

б) у вікні **Создание форм** розкрийте список таблиць та запитів

БД і виберіть таблицю **tblReaders***;*

в) Зі списку **Доступные поля** перенесіть у список **Выбранные поля** поля : **NB, Прізвище, Кафедра** і **Телефон**. Потім розкрийте список таблиць та запитів, виберіть запит **quList1**. Його поля з'являться в списку ***Доступные поля,*** перенесіть у список **Выбранные поля** всі поля запиту;

г) у наступному вікні необхідно вибрати тип представлення даних. Виберіть по **Читачі**, тому що головною формою буде форма, яка показує відомості про читачів. Відзначте перемикач **Подчиненные формы**, щоб інші дані були приміщені в підпорядковану форму, і клацніть на кнопці **Далее***;*

д) у наступному вікні виберіть вид підпорядкованої форми. Тому що зручніше усього було б бачити дані про книги, що читаються, наданими у вигляді таблиці, відзначте перемикач **Ленточный** або **Табличный** і клацніть на кнопці **Далее***;*

е) виберіть стиль для головної форми - **Проста**. Стиль показується у вікні вибору відразу ж, як тільки ви відзначите один із них. Клацніть на кнопці **Далее**;

ж) у наступному вікні необхідно задати імена форм – головної і підпорядкованої. Access створив дві форми, пов'язані одна з одною. Ви можете корегувати їх у режимі конструктора незалежно одна від одної, а також користуватися підпорядкованою формою незалежно від головної. Дайте головній формі ім'я **frmReaders**, а підпорядкованій – **frmList1**. Клацніть на кнопці **Готово** і ви побачите на екрані створену форму. На наступному кроці поліпшіть створену форму за допомогою **Конструктора форм**.

Корегування форми в режимі Конструктор

Додамо у підпорядковану форму **frmList1** поле, де обчислюють загальну суму пені для кожного читача. Вигляд форми у режимі «Конструктора…» наведено на рис.3.2.

Конструктором форм додано поле «Всего пени», яке розташовано в області «Примечание формы». Розглянемо послідовність цих дій детальніше:

1. Відкрийте підлеглу форму **frmList1** в режимі «**Конструктора...**». В області «**Примечание формы**» створіть нове поле, що буде обчислюватись. Змініть текст перед цим полем на «Всего пени».

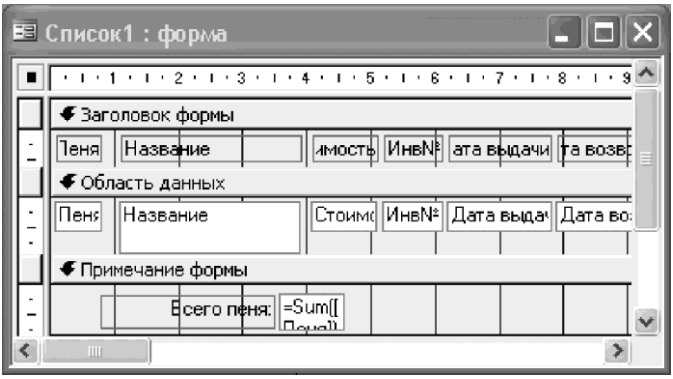

Рисунок 3.2 – Форма frmList1 в режимі конструктора

2. У вікні «**Свойства**»:
   – в рядку «**Данные**» введіть формулу: **=Sum([Пеня])** — це буде сума пені,
   – в рядку «**Имя»** введіть ім’я створеного об’єкта Поле, наприклад **-VP**.

3. Збережіть зміни і відкрийте форму в режимі перегляду. У вікні форми ви побачите записи про видані книги, а якщо форма **frmList1 – ленточная,** то а у нижній частині вікна — загальну суму пені по всіх читачах. Зверніть увагу на зміну розмірів полів, в яких виводиться інформація про книги. Це зроблено для того, щоб автори і назви книг були показані у декілька рядків.

4. У нижній частині кожної форми розташовано групу елементів управління, що називаються «**Кнопки перехода**». Вони дозволяють переміщатися по записах, додавати до таблиці порожні записи, щоб у подальшому заповнити їх інформацією. Цей елемент управління з'являється у формі, якщо властивість форми «**Кнопки перехода**» має значення «**Да**». Якщо його змінити на «**Нет**», елементу управління «**Кнопки перехода**» е буде на формі. У підлеглій формі цей елемент не потрібен. Тому, знову відкрийте форму **frmList1** в режимі «**Конструктора…**». Клацніть правою кнопкою миші на вільному місці у формі. З контекстного меню виберіть «**Свойства**», зайдіть на закладку «**Все»** і знайдіть параметр «Кнопки перехода» і встановіть його значення на «**Нет**». Збережіть форму.

Поліпшимо тепер вид форми «frmReaders» (рис.3.3)

Для цього відкрийте її у режимі **Конструктора** і виконайте такі дії:

1. Збільшить область **Заголовок формы**. Додайте елемент управління **Надпись**, напишіть у ньому текст «Информация про читателей и виданные книги». Змініть шрифт і його розмір для кращого вигляду;

2. Перемістіть поля «Кафедра» і «Телефон» праворуч, щоб звільнити більше місця для підлеглої форми;

3. Розтягніть підлеглу форму на все вільне місце.

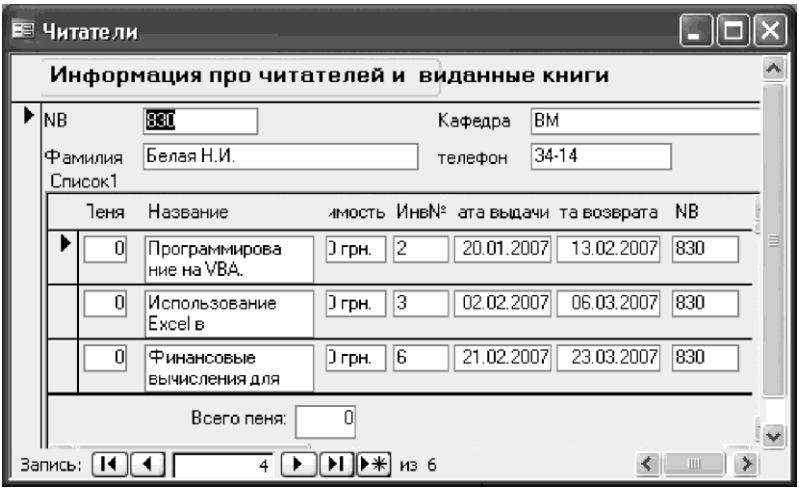

Рисунок 3.3 – Форма з підпорядкованою в режимі перегляду

Важливо: Якщо підпорядкована форма **frmList1** подана в табличному виді, то створеного поля **VP(Усього пені)** не видно, так як в табличній формі не передбачається показ полів з області **Примечание.** Проте поле у формі є присутнім і обчислення проводяться. Тому, потрібно створимо поле у формі **frmReaders**, що буде показувати нам дані, обчислені в полі **VP (Усього пені)** форми **frmList1**. Для цього відчиніть форму **frmReaders** у режимі конструктора і додайте елемент управління **Текстовое поле**. Відзначте створене текстове поле, викличте правою кнопкою миші

контекстне меню, виберіть **Свойства** та у рядку **Данные** впишіть такий текст:

=[Forms]![frmReaders]![frmList1].[Form]![VP],

який значить наступне: в колекції форм нашої БД є форма з назвою **frmReaders, в** цій формі є підпорядкована форма **frmList1,** в підпорядкованій формі є поле **VP,** значення цього поля буде відображено в створеному текстовому полі головної форми **frmReaders.**

3.3.3 Створення звітів за допомогою *Мастера отчетов*

Звіт – це об'єкт бази даних, який використовується для відображення й узагальнення даних. Звичайно звіт є кінцевим продуктом БД, призначених для друку, і при створенні звіту можна комбінувати дані з таблиць, запитів і навіть форм. За допомогою звітів можна переглядати, форматувати та підсумовувати дані, можна використовувати такі параметри як сортування, групування та зведення даних. Звіти можуть містити докладні відомості про окремі записи, зведені відомості про великі групи записів або і про те, і про інше. Інформацію на звітах можна формувати на потрібному рівні деталізації і в кількох форматах. Окрім того, звіти Access також можна використовувати при створенні наклейок для списків розсилок і багато чого іншого.

Створювати звіти в базі даних Access можна декількома способами, які пропонуються на вкладці *Створення* стрічки у групі *Звіти*.

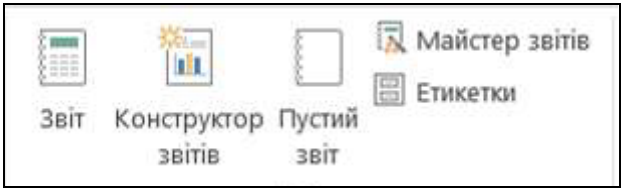

Рисунок 3.4 – Інструменти для створення звітів

Звіт дозволяє створити найпростіший табличний звіт, що містить усі поля з джерела записів, вибраного в області переходів.

Майстер звітів запускає покроковий майстер, за допомогою якого можна задавати поля, рівні групування/сортування та параметри макета. Як наслідок роботи майстра буде створено звіт на базі вибраних параметрів.

Завдання 4. Перейдіть на вкладку **Отчеты** і створіть звіт за допомогою кнопки ***Звіт*** про наявні у бібліотеці книги.

Завдання 5. Створіть простий звіт за доромогою ***Майстра звітів*** , що виводить на принтер список книг, згрупованих за видавництвами, і підраховує загальну вартість книг кожного видавництва.

Для цього виконайте такі дії:

а) У вікні **Мастера отчетов** виберіть у якості джерела даних таблицю **tblBook.** Для переходу до наступного кроку клацніть на кнопці **Далее;**

б) Тепер необхідно відібрати поля, що будуть відображені в звіті. Виберіть усі поля таблиці **tblBook**. Зауважимо, що на цьому кроці можна вибирати поля не тільки з зазначеної вище таблиці, але з будь-яких таблиць і запитів поточної бази даних. Для цього у вікні **Таблицы / Запрос** розкрийте список таблиць і запитів і виберіть потрібний об'єкт. Список полів обраної таблиці з'явиться у вікні **Доступные поля**, що дасть можливість перенести потрібні поля у вікно **Отобранные поля;**

в) На цьому кроці необхідно визначити, чи хочете ви групувати дані в звіті за значенням якогось поля. Access часто сам пропонує поле, за яким виконувати угрупування. Виберіть поле **Видавництво**;

г) Наступний екран пропонує вам вибрати порядок сортування й обчислення, що необхідно виконати для запису. Сортування можна виконувати за чотирма полями. Виберіть у першому вікні поле **Шифр**, а в другому **Рік видання**. Це означає, що для кожного видавництва книги будуть упорядковані за шифром, а для кожного шифру – за роком видання;

д) Клацніть на кнопці **Результаты**, щоб організувати обчислення підсумкових значень для потрібних полів. Access запропонує вам усі числові поля серед відібраних у звіт. У вашому випадку будуть запропоновані поля **Рік і Вартість**. Напроти поля **Вартість** відзначте прапорці під написами **sum** і **avg**, щоб прорахувати сумарну і середню вартості книг для кожного видавництва і по бібліотеці в цілому. Відзначте перемикач **Показать данные** и

результаты і прапорець **Вычислить проценты**, якщо це необхідно;

е) Виберіть вид **макета** для звіту. Макет показується на екрані і ви можете вибрати його за своїм смаком. Виберіть **стиль** звіту серед тих, що пропонуються у вікні. Назвіть ваш звіт **rptListBook** і клацніть на кнопці **Готово**.

Перегляньте створений звіт і переконайтеся, що він задовольняє усім вимогам.

Завдання 6. Створіть звіт, що виводить список читачів бібліотеки, згрупованих за кафедрами, список книг кожного читача, рахує пеню та підводить по ній результат для кожного читача і по всіх читачах. Для створення такого звіту використовується таблиця **tblReaders** і запит **quList1**, створений у попередній роботі, у якому є поле, що обчислюється, **– Пеня**. Звіт буде мати аналогічний формі вигляд, але його можна друкувати.

Для створення звіту виконайте такі дії:

а) виберіть **Мастер отчетов** і таблицю **tblReaders**;

б) виберіть з таблиці **tblReaders** поля: Прізвище, Кафедра і Телефон, а із запиту **quList1** усі його поля;

в) тип представлення даних – **по Читачі**;

г) додати рівні угрупування – за полем **Кафедра**;

д) сортувати за полем **Дата видачі**, а результати підводити за полем **Пеня**;

е) виберіть вид макета і стиль;

ж) дайте звіту ім’я **rptListReaders** і **Готово**.

Уважно перегляньте створений звіт.

3.4 Зміст звіту

1. Занотуйте в звіті макети створених форм та прокоментуйте особливість використаного кожного інструменту для створення форми та полученого результату.

2. Стисло відповеділь на наступні запитання:

- призначення об’єкта **Форма**. Типи форм і їхні особливості (проста, стрічкова і т.д.);
- інструменти створення форм, режими перегляду форм;
- поняття **підпорядкованої форми**. Створення підпорядкованої форми і впровадження її в основну форму;
- створення полів, що обчислюються, у формі;
- призначення об’єкта **Звіт**. Інструментарій створення звітів в MS

Access.

ПРАКТИЧНА РОБОТА №4. СТВОРЕННЯ ТА КОРЕГУВАННЯ ВЛАСТИВОСТЕЙФОРМ І ЇХ ЄЛЕМЕНТІВ КЕРУВАННЯ В РЕЖИМІ КОНСТРУКТОРА

4.1 Мета роботи

Формування вмінь і навичок створення форм та їх елементів керування в режимі конструктора. Закріплення вмінь та навичок використування майстрів СУБД. Застосування на практиці знань структури реляційної моделі при створені підпорядкованих форм.

4.2 Загальні відомості

Форма є основним засобом організації інтерфейсу користувача в додатку. Форми можна створювати з різною метою, а саме: перегляду, уведення, виведення і редагування даних. Це найбільш розповсюджена сфера застосування форм. Прості форми з використанням *Мастера форм* створювались в попередніх роботах. *Конструктор форм* дозволить спроектувати форми з різними елементами управління, що спрощують організацію перегляду і уведення потрібних даних.

Використання *Конструктора форм* дозволяє створювати форму “з нуля”, цілком керуючи всім процесом. Форма може містити п’ять розділів:

– заголовок форми;
– верхній колонтитул;
– область даних;
– нижній колонтитул;
– примітка форми.

На формі можна розташувати *Елементы управления*. Список доступних елементів управління обмежений стандартними і додатковими елементами (елементи *Active*), які містяться на *Панели елементов*. Вікно конструктора форм представлено на рисунку 4.1. Панель елементів управління представлена на рисунку 4.2.

Усі властивості форми, її розділів, елементів управління, розташованих на формі, відображаються і редагуються в *Окне Свойств*(рис.4.3). Щоб відобразити на екрані вікно властивостей якого-небудь елемента (у тому числі форми чи будь–якого з її розділів), необхідно його виділити, а потім клацнути на кнопці

Свойства панелі інструментів чи вибрати відповідний пункт із контекстного меню. Якщо вікно властивостей уже присутнє на екрані, то для відображення в ньому властивостей якого-небудь конкретного елемента досить просто його виділити.

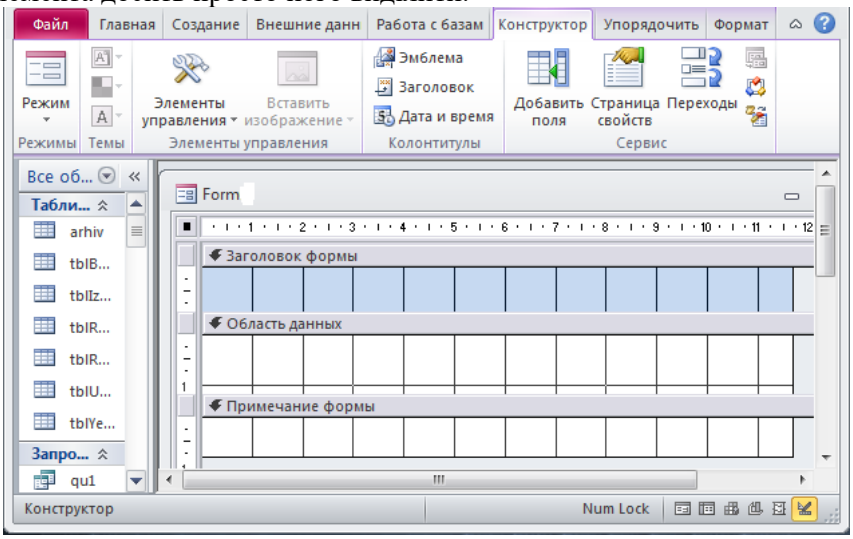

Рисунок 4.1 – Вікно конструктора форм

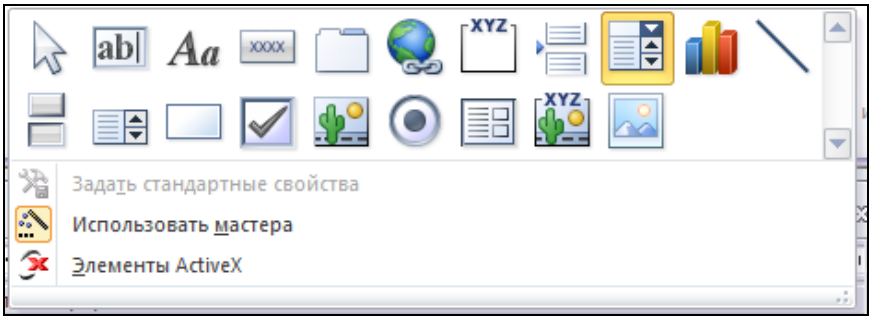

Рисунок 4.2 - Панель елементів управління

Вікно властивостей містить п’ять вкладок: *Макет, Данные, События, Другие, Все* (рис.4.3).

Макет. Тут зібрані властивості, що стосуються зовнішнього вигляду об’єкта (форми, розділу форми, елемента управління).

Данные. На цій вкладці представлені властивості, що відносяться до джерела даних елемента і до засобів маніпулювання

даними.

События. Тут представлені посилання на процедури, функції чи макроси, які викликаються при наставанні для елемента якої-небудь події, наприклад, відкритті чи закритті форми, утраті чи одержанні фокуса, зміні даних і т.д. Самі функції розташовані в модулі форми, але перейти до їх перегляду і редагування можна, клацнувши на кнопці *Построитель* (кнопка з трьома горизонтальними крапками, розташованими праворуч від кожної властивості).

Другие. Тут зібрані властивості, що не ввійшли в перші три категорії.

Все. На цій вкладці представлені абсолютно усі властивості елемента з перших чотирьох вкладок.

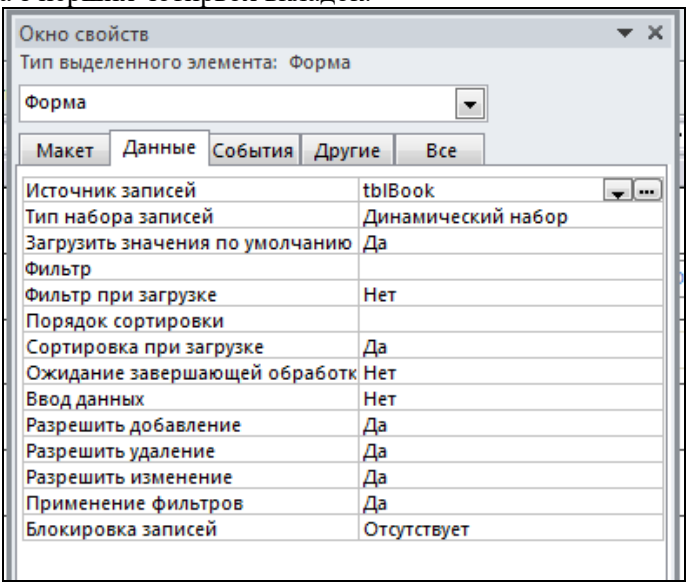

Рисунок 4.3 - Вікно властивостей форми

4.3 Приклади створення та корегування форм Конструктором для БД «Library»

4.3.1 Форми уведення даних

Форми введення даних до таблиць t`blBook` і `tblReaders`

Якщо потрібно створити відносно простий інструмент уведення даних, то можна заощадити зусилля, використовуючи *Мастер форм*

(Form Wizard) для побудови базової структури, і потім змінити властивість *Ввод данных* на вкладці *Данные* і додати окремі елементи.

Таблиці t**blBook** і **tblReaders** добре працюють із простими формами. Після завершення роботи *Мастера форм* (по створенню форм **frmBook** і **frmReaders**) можна змінити структуру і форму елементів управління, а також додати при необхідності заголовок і рисунки. Поліпшення форм можна провести також за рахунок додавання вибору можливої інформації для поля **Видавництво** і для поля **Телефон кафедри**.

Форма введення даних до таблиці tblReadBook

Створюючи форму **frmReadBook**, для уведення даних у таблицю **tblReadBook,** потрібно пам'ятати, що *Мастер форм* заміняє поля уведення під **Інвентарний номер** і **Номер читацького квитка** на елемент управління *Поле со списком* (*ComboBox*), якщо властивість *Тип данных* відповідних полів у таблицях установлена за допомогою *Мастера подстановки*. На жаль для заповнення таблиці **tblReadBook** цього виявилося недостатньо, тому що при уведенні інформації в поле **Інвентарний номер** потрібно відслідковувати, щоб обрана читачем книга дійсно знаходилася в бібліотеці, а не була уже видана іншому читачу. Рішенням проблеми буде заміна *Источника данных* для поля із списком **Інвентарний номер** форми з таблиці **tblReadBook** на запит (наприклад, під ім'ям **quListReadBook**), що створює набір записів з інформацією тільки про книги, які є у наявності в бібліотеці в даний момент. Запит **quListReadBook** приведений на рисунку 4.4.

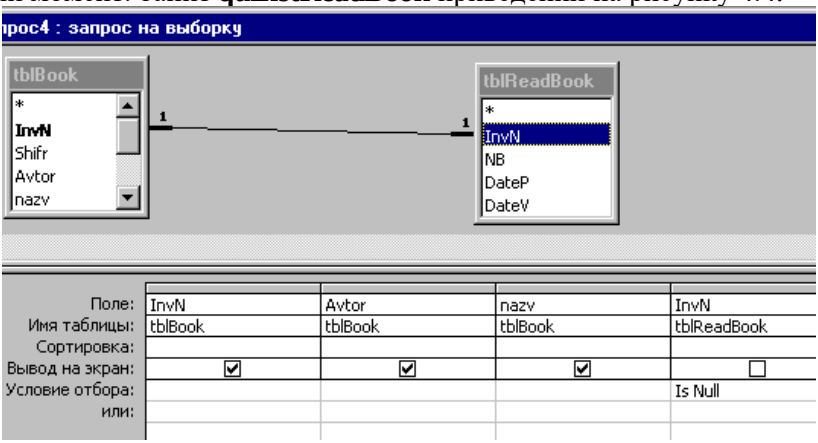

Рисунок 4.4– Запит quListReadBook

Послідовність дій:

1. Створюємо запит **quListReadBook.**

2. Відкриваємо форму **frmReadBook** у режимі *Конструктора.*

3.Виділяємо елемент управління *Поле із списком,* яке призначене для вводу інформації в поле **Інвентарний номер** таблиці **tblReadBook.** В вікні *Свойства* цього елемента в рядку *Данные* вписуємо **quListReadBook.**

4.3.2 Форми перегляду та корегування даних за різними критеріями вибору

Для створення таких форм використовується об'єкт *Поле со списком*. Цей об'єкт створюємо за допомогою *Мастера*:

1. Клацнувши попередньо у панелі елементів на кнопці *Мастер,* виберіть елемент *Поле со списком (ComboBox*) і помістіть його на форму. На екрані монітора з'явиться перше вікно майстра.

2. На першому кроці пропонується вибрати спосіб одержання рядків значень. Необхідно вибрати один із трьох варіантів:

–об'єкт *Поле со списком* буде використовувати значення з таблиці чи запиту. У цьому випадку список значень буде базуватися на запиті, що вибирає необхідні рядки з базової таблиці чи запиту;

–*Будет введен фиксированный набор значений.* Джерелом рядків для поля зі списком буде фіксований набір значень, введений користувачем;.

–*Поиск записи в форме на основе значения, которое содержит поле со списком*. Поле зі списком може використовуватися для переходу на запис у джерелі даних форми. У цьому випадку джерелом рядків для поля зі списком автоматично стає джерело даних форми.

3. Клацніть на кнопці *Далее*, щоб перейти до наступного вікна майстра. На цьому етапі потрібно вибрати поля, значення яких будуть складати стовпці списку, що випадає. Зробіть необхідний вибір і клацніть на кнопці *Далее*.

4. Подальша послідовність вікон *Мастера* дозволить таке:

–установити бажану ширину стовпців (це можна зробити і пізніше, коли поле зі списком уже буде створене);

–указати поле, значення якого буде представляти собою значення поля зі списком. Іншими словами, ви повинні вибрати стовпець, значення якого зберігається в перемінній

форми;

–вказати ім’я створеного поля зі списком.

Приклад створення форми frmLookBook для пошуку потрібної книги “За прізвищем автора”, “За назвою книги”.

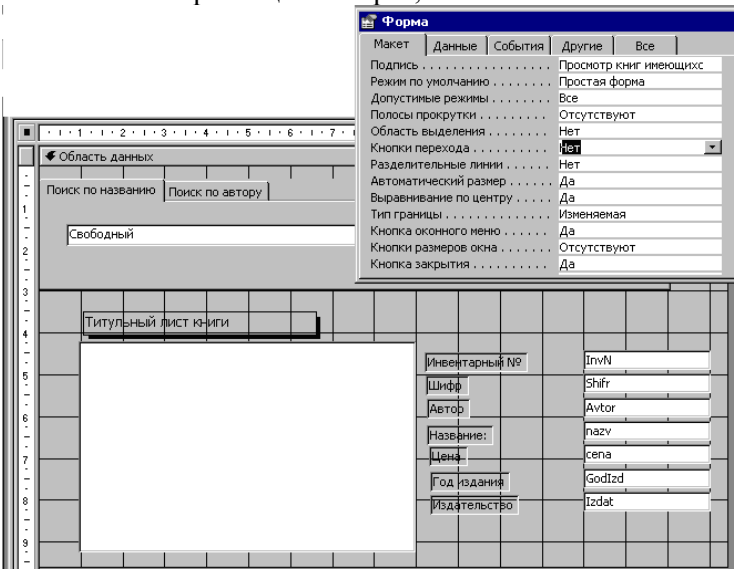

Рисунок 4.5 – Форма frmLookBook у режимі конструктора з виведеним вікном властивостей форми

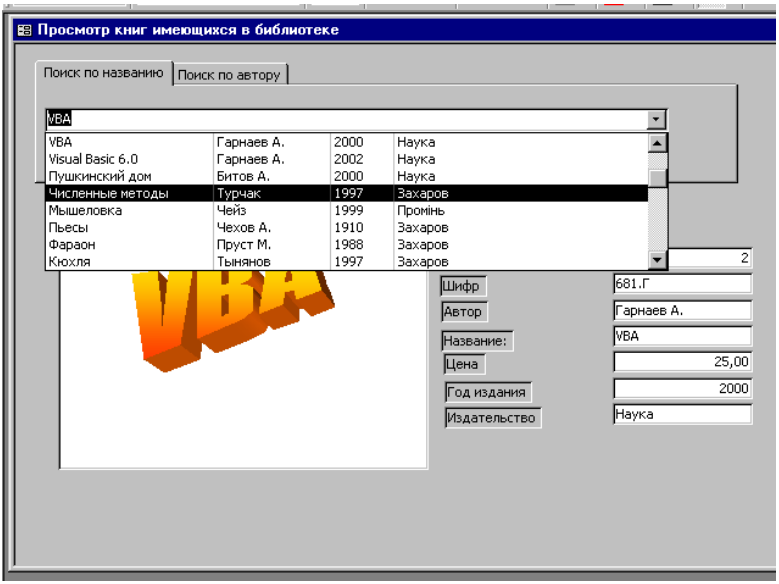

Рисунок 4.6 – Форма frmLookBook у режимі перегляду і пошуку книг за назвою

Послідовність дій:

1. Джерелом даних для цієї форми є таблиця **tblBook.**

1. Відкриємо форму в режимі конструктора і розташуємо на ній елемент управління *Набор вкладок (Tab).*

2. На кожній із вкладок розташуємо по одному елементу управління *Поле со списком (ComboBox)* для введення критерію пошуку. Привласнимо їм імена – **cmbAvtor** і **cmbNazv**.

3. Нижче *Набора вкладок* розмістимо приєднану рамку об'єкта (для виводу титульного листа книжки) і текстові поля для виведення інформації про знайдену, відповідно до обраного критерію пошуку, книгу.

4. Настроїмо елементи управління **cmbAvtor** і **cmbNazv.** Як описано вище, для кожного з цих елементів потрібно визвати *Мастера* і вибрати *Поиск записи в форме на основе значения, которое содержит поле со списком*. У цьому випадку джерелом рядків для поля зі списком автоматично стає джерело даних форми, тобто таблиця **tblBook.** Поле зі списком **cmbAvtor** буде використовуватися для переходу на запис у таблиці **tblBook** по вибраному автору**,** .а **cmbNazv –** по вибраній назві книжки**.**

5. Вид створених елементів управління **cmbAvtor** і **cmbNazv** визначається через властивості, які потрібно установити у вікні *Свойства поля со списком.*

6. Вид всієї форми визначається через властивості, які потрібно установити у вікні *Свойства форми.*

Форма **frmLookBook** на різних етапах роботи системи представлена на рисунках 4.5, 4.6.

4.3.3 Форми для редагування даних

У таких формах користувачу повинна бути надана можливість редагування інформації в таблиці (чи таблицях), що є джерелом даних для шуканої форми, а саме:

– переміщуватися на потрібний запис;

– змінювати інформацію безпосередньо в полі;

– додавати нові записи;

– видаляти записи.

Як приклад створіть форму **frmEditBook** (див. рисунок 4.7) для редагування інформації в таблиці **tblBook**.

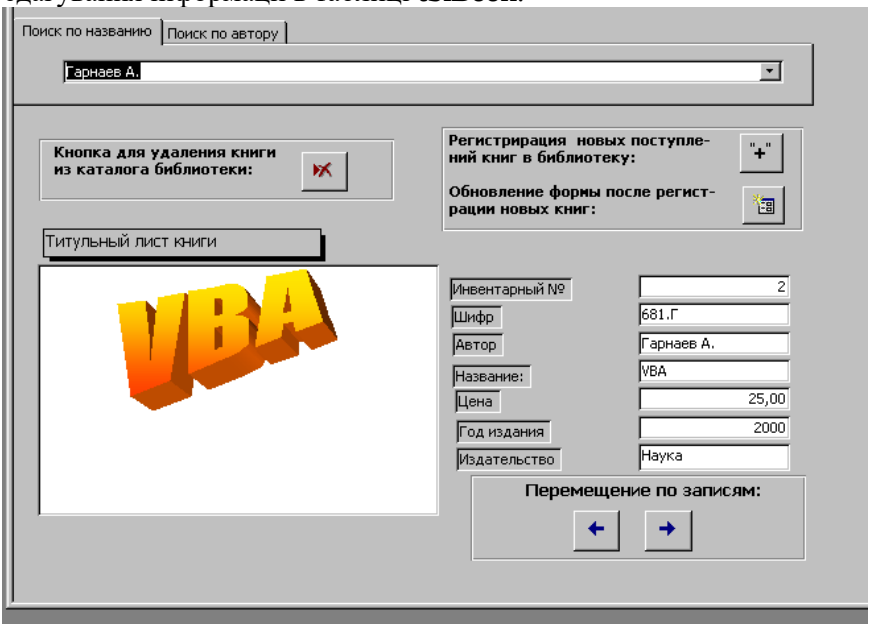

Рисунок 4.7 – Приклад форми для редагування даних у таблиці tblBook

З цією метою доробіть форму **frmLookBook**. Розташуйте на ній

такі кнопки:

– cmdAdd – для додавання нового запису в таблицю **tblBook**;

– cmdRefresh – для відновлення інформації, що відображається у формі;

– **cmdDelete** – для видалення запису таблиці **tblBook**;

– mdGco – для переміщення по записах.

Кнопки додайте за допомогою *Мастер кнопок (Command Button Wizard)*. Для створення кнопок **cmdAdd**, **cmdDelete**, **cmdGo**:

– у вікні конструктора форми переконайтеся, що активізована кнопка *Мастер (Control Wizards)* на панелі елементів, потім натисніть кнопку *Кнопка (Command Button)* і клацніть мишкою в тому місці форми, у якому збираєтеся створити кнопку. На екрані з'явиться діалогове вікно *Создание кнопок (Command Button Wizard*), що містить список категорій і дій для обраної категорії. Вікно *Создание кнопок* приведено на рисунку 4.8;

– виберіть пункт *Переходы по записям* чи *Обработка записей* у списку *Категории (Categories)*, потім пункт, відповідний до призначення кнопки в списку *Действия (Начальная запись, Следующая запись, Удалить запись, Добавить запись)*;

– для створення кнопки **cmdRefresh** виберіть пункт *Работа с формой (Form Operation)* у списку *Категории*, потім пункт *Обновить данные формы* в списку *Действия*.

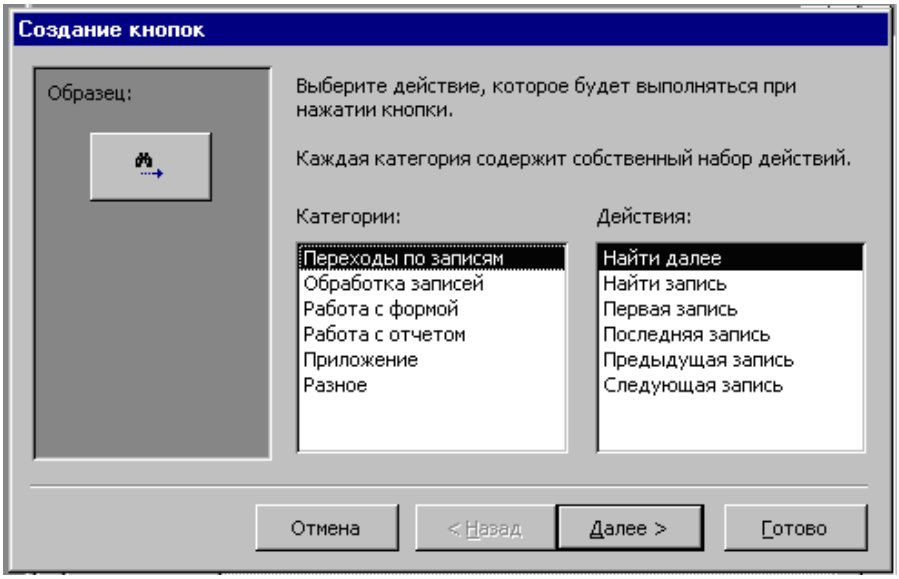

Рисунок 4.8 – Вікно Создание кнопок

4.4 Завдання на самостійну роботу

1. Виконайте побудову форм, що наведені у розділі 4.3 - **Приклади створення та корегування форм Конструктором для БД «Library»**.

2. Створіть для БД “Бібліотека” форми, що реалізують основні функції будь-якого додатка Access, а саме:

– форми для уведення даних у таблиці;

– форми для перегляду даних. Наприклад, для кожної кафедри вивести список читачів, що не здали книжки до бібліотеки вчасно, з вказівкою для кожного боржника кількості таких книжок, а для кафедри – кількості боржників;

– форми для редагування даних. Наприклад, форма, що дозволяє знайти потрібного читача (використання поля зі списком), переглянути список книжок, що знаходяться у цього читача (в підпорядкованій або зв’язаній формі), при необхідності провести редагування інформації (використання кнопок);

Використовуйте можливості режиму *Конструктор* для створення форм і інструментарій *Мастер* для побудови елементів

управління.

4.5 Зміст звіту

Звіт повинен містити:

1. Перелік та зображення створених форм.
2. Стислі відповіді на контрольні питання.

4.6 Контрольні питання

1. Створення форм у режимі *Конструктора*.
2. Структура і властивості форм.
3. Еелементи управління, призначення кожного елемента, їх властивості.
4. Три способи використання елемента “*Поле со списком*”;
5. Використання *Мастера* для створення кнопок. Види *Категорий* і *Действий* вікна *Создание кнопок*;
6. Поняття підлеглої форми, зв’язаної форми;
7. Створення полів, що обчислюються, з використанням “*Построителя выражений*”.

ПРАКТИЧНА РОБОТА №5.
СТВОРЕННЯ ЗАПИТІВ НА ЗМІНУ

5.1 Мета роботи

Вивчення можливостей створення БД і внесення в неї інформації з використанням зовнішніх БД.

5.2 Загальні відомості

Окрім запитів для добору даних, MSAccess пропонує, так звані **Запити на внесення змін.**

Запит на зміну за одну операцію вносить зміни або переміщує декілька записів. Існує чотири типи таких запитів:

Запит на створення таблиці – ґрунтується на запиті на вибирання і забезпечує формування і створення нової таблиці на основі всіх або частини даних із однієї або декількох таблиць. Запит на створення таблиці корисний для виконання наступних дій:

– створення звітів, що містять дані, відповідні певному моменту часу. Щоб отримати дані на вказаний момент часу, необхідно розробити запит на створення таблиці, в якому необхідні записи відбираються залежно від вказаного моменту часу і вміщуються в нову таблицю. Потім як основа для звіту потрібно використати цю таблицю, а не запит.

– створення архівної таблиці, що містить старі записи. Наприклад, можна створити таблицю, що зберігає всі старі замовлення, перш ніж видалити їх з поточної таблиці «Замовлення».

Запит на оновлення – дає змогу вносити зміни в групу записів, які вибираються за допомогою запиту на вибирання. (Підвищити заробітну плату деякій категорії робітників, наприклад, водіям або робітникам, які мають ставку менше зазначеної суми).

Запит на знищення – забезпечує виключення записів з однієї або кількох зв’язаних таблиць (Наприклад, ліквідувати на підприємстві деяку посаду).

Запит на додавання записів. Запит на додавання додає групу записів із однієї або декількох таблиць у кінець однієї або декількох таблиць. Для однієї і тієї ж таблиці можна створювати різні запити,

кожен з яких може вилучати з таблиці певну частину інформації, яка в даний момент необхідна.

5.3 Створення запитів на зміну

Запити на оновлення

Запити, які змінюють значення групи записів називаються запитами на оновлення даних. За допомогою такого запиту можна, наприклад, змінити вартість книг залежно від дати їх видання, збільшити вартість на 20% для книг 1998 року видання тощо. Оскільки запит на оновлення змінює інформацію у базі даних, для захисту від можливих помилок завжди робіть копії. Для цього перейдіть у вікно бази даних, виберіть таблицю і виконайте команду «*Правка→ Копировать*». Далі виберіть команду «*Правка→Вставить*» і у вікні діалогу дайте копії таблиці нове ім'я.

Запит на відновлення роблять на основі простого запиту. Тому спочатку сформуємо запит на вибірку. Ретельно переконаємось у тому, що створений запит без помилок вибирає потрібні записи. Після цього трансформуємо запит на вибірку у запит на оновлення.

Приклад створення запиту на оновлення.

До запиту включимо поля «**Год**» і «**Стоимость**» з таблиці **tblBoo**k. При цьому запит побудуємо так, щоб дати можливість вводити рік видання як параметр. Для цього в рядку *Условие отбора* у квадратних дужках запишемо вираз: **[Вкажіть рік].** Під час виконання запиту цей текст з'явиться в діалоговому вікні, куди можна ввести потрібне значення.

Щоб перетворити *запит на вибірку* у *запит на оновлення* переходимо до режиму *Конструктора запросов*, в основному меню вибираємо *Запрос→Обновление*. Бачимо новий рядок в параметрах запиту. Він називається *Обновление*. У цьому рядку для поля «**Стоимость**» вводимо формулу «[Стоимость]*([Наценка]+100)/100» для перерахунку вартості книг (рис.5.1). Цей вираз використовує ще один параметр «[**Наценка**]», що дозволяє вводити розмір націнки у відсотках.

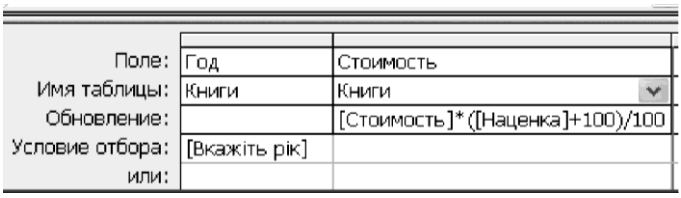

Рисунок 5.1 – Запит на оновлення в Конструкторі

Виконуємо запит. На екрані послідовно з'явиться вікно із запитом щодо розміру націнки, вводимо, наприклад, 50. Потім Access виведе вікно діалогу із запитом щодо року видання. Вводимо значення, наприклад, 2000. Access попередить, скільки буде змінено записів (згідно нашого прикладу). Підтверджуємо дозвіл на заміну. Відкриваємо таблицю і аналізуємо результат: вартість двох книжок відання 2000 р. збільшена на 50%.

Запит на формування нових таблиць

Запити цього типу створюють нову таблицю і записують до неї дані з вже існуючих таблиць або запитів.

Приклад створення запиту на створення нової таблиці

Для прикладу ми побудуємо за допомогою запиту таблицю з даними про читачів, які вчасно не повернули книги.

!. Як і у попередньому випадку, починаємо з формування запиту на вибірку. До запиту включимо таблицю **tblReaders** і запит **List1**, який знаходить суму пені для кожного читача, кількість заборгованих книг, сумарну вартість заборгованих книжок

З таблиці **tblReaders** вибираємо поля «NB», «Фамилия», «Кафедра», «Телефон». Із запиту **List1** вибираємо всі поля:

Щоб відібрати тільки читачів, що мають заборгованість, вказуємо для поля «Пеня» умову «<>0».

Виконуємо запит і переконуємось у тому, що він правильно відбирає записи.

2. Щоб перетворити запит на вибірку у запит на формування нової таблиці режим «*Конструктора запросов*», в меню вибираємо «*Запрос→Создание таблицы*». Access запропонує ввести ім'я нової таблиці, наприклад **tblDolg**.

Виконуємо запит. Access повідомить в окремому вікні про кількість записів, що будуть передані до нової таблиці. У цьому ж вікні пропонується підтвердити намір на створення таблиці або відмовитись від цієї дії. Натискаємо кнопку «Да». Переходимо на

закладку «Таблицы». У списку маємо побачити нову таблицю **tblDolg**.

Створення запитів на видалення даних

Такі запити дозволяють видаляти з таблиць групи записів, що задовольняють вказаним умовам.

Приклад запиту на видалення даних

Створимо запит, що видаляє з таблиці **tblReadBook** записи про читачів на момент повернення ними книги.

Важливо! Для захисту від можливих помилок зробіть резервну копію таблиці.

1. Як і в попередніх випадках, спочатку розробляємо запит на вибірку даних. В подальшому ми перетворимо його у запит на видалення даних. Запит побудуємо на основі таблиці **tblReadBook**, з якої візьмемо поля «Інвентарний номер» і «NB». Для цих полів вводимо параметри **«[Введите Инв№ книги]»** і **«[Введите номер читат. билета]»**. При виконанні цього запиту послідовно з'являться два вікна для введення значень цих полів у діалоговому режимі.

2. Щоб перетворити запит на вибірку у запит на видалення даних переходимо в режим *Конструктора запросов*, в меню вибираємо *Запрос→Удаление*. У параметрах запиту з'явиться рядок *Удаление*, в котрому виберіть із списку установку *Условие* для полів **Інвентарний номер** і **NB**.

Запустіть запит на виконання. На екрані з'явиться вікно з повідомленням про кількість записів, що будуть видалені з таблиці. Для підтвердження натисніть кнопку «Да». Відкрійте таблицю й переконайтесь, що записи видалено.

Створення запиту на додавання записів

Попереднє зауваження: Якщо у таблиці, в яку додаються записи, є ключове поле, то з таким ім'ям повинне бути поле і в таблиці – джерелі.

1. Створіть в режимі **Конструктора** запит, що містить таблицю – джерело, записи з якої необхідно додати в іншу.

2. У режимі Конструктора запиту натисніть стрілку поруч з кнопкою **Тип запросу** на панелі інструментів і виберіть команду **Добавление**. На екрані з'явиться діалогове вікно **Добавление**.

3. У поле **Имя таблицы** введіть ім'я таблиці, в яку необхідно

додати записи.

4. Виберіть параметр **В текущей Базе данных**, щоб вмістити таблицю у відкриту в даний момент базу даних, або виберіть параметр **В другой Базе данных** і введіть ім'я бази даних, в яку потрібно вмістити нову таблицю. У разі необхідності укажіть шлях.

5. Натисніть кнопку **ОК**.

6. Із списку полів (таблиці – джерела) в бланк запиту перемістіть за допомогою миші поля, які необхідно додати, а також ті, які будуть використані при визначенні умови відбору.

Якщо всі поля в обох таблицях мають однакові імена, то можна просто перемістити за допомогою миші символ «зірочка» (*) в бланк запиту.

7. Якщо в обох таблицях виділені поля мають однакові імена, відповідні імена автоматично вводяться в рядок **Добавление**. Якщо імена (але не зміст) полів двох таблиць відмінні один від одного, в рядок **Добавление** введіть імена полів, що додаються в таблицю.

8. Введіть умову відбору, по якій буде здійснюватись додавання, в рядок **Условие отбора**.

9. Для додавання записів натисніть кнопку **Запуск** на панелі інструментів. У діалоговому вікні **Access** повідомить про кількість записів, що додаються, і застереже про неможливість повернення.

10. Погодьтесь з виконанням безповоротного оновлення натисненням кнопки **Да**.

5.4 Завдання на самостійну роботу

1. Створити архівну таблицю, для інформації про книги, що списані за такими критеріями:

- книги з комп’ютерних технологій видані до 1990 году;
- книги, що не повернуті в бібліотеку більш як 3 роки.

2. Інформацію, для якої створена архівна таблиця в першому пункті завдання, видалити з основних таблиць БД.

3. В архівній таблиці перерахувати ціну книжок з комп’ютерної технології – зменшити на вказаний процент.

5.5 Зміст звіту

1. Створені запити занотовувати:

а) зміст створює мого запиту,

б) запит в режимі Конструктора;
в) результат виконання запиту.

2. Відповісти на наступні питання:
   а) Що таке запит?
   б) Які бувають типи запитів?
   в) Що таке запит на вибірку?
   г) Як створити запит з обчислювальним числовим полем?
   д) Як створити запит на вилучення записів з таблиці?
   е) Як створити запит на оновлення (зміну) значень окремих полів таблиці?
   ж) Як за допомогою запиту створити таблицю

ЛІТЕРАТУРА

Основна

1. Гурвиц Г. А. Microsoft Access 2010. Разработка приложений на реальном примере. – Издательство: BHV, 2010. – 496с.
3. Мирошниченко Г. А. Реляционные базы данных: практические приемы оптимальных решений. –СПб: БХВ-Петербург, 2005. –400с.
4. Єрьоміна Н. В. Проектування баз даних: Навчальний посібник. –К. : КНЕУ, 1998. – 208 с.
5. Джеффри Д. Ульман, Дженнифер Уидом. Введение в системы баз даннях. – Издательство «Лори», 2000. – 376 с.

Додаткова

1. Ivankov V. F. Calculation of CFD-thermal models of oil-cooled transformer equipment / V. F. Ivankov, A. V. Basova // Електротехніка та електроенергетика. - 2016. - № 2. - С. 19-32.
2. Ніценко В. В. Дослідження похибок трансформаторів струму у системах релейного захисту в усталених та перехідних режимах енергосистеми / В. В. Ніценко, Д. О. Кулагін, П. В. Махлін // Електротехніка та електроенергетика. - 2016. - № 2. - С. 59-71.
3. Ніценко В. В. Дослідження похибок трансформаторів струму у системах релейного захисту в усталених та перехідних режимах енергосистеми / В. В. Ніценко, Д. О. Кулагін, П. В. Махлін // Електротехніка та електроенергетика. - 2016. - № 2. - С. 59-71.

4. Дивчук Т. Е. Особенности определения параметров силовых трансформаторов методами схемно-полевого моделирования / Т. Е. Дивчук, Д. К. Мимоход, С. А. Кутилин, А. Э. Кузнецов, Ю. В. Гуразда, И. С. Сырых // Електротехніка та електроенергетика. - 2017. - № 1. - С. 61-70.
5. Дьяченко В. В. Формирование программы энергосбережения для систем электроснабжения / В. В. Дьяченко // Електротехніка та електроенергетика. - 2015. - № 1. - С. 70–76.
6. Методичні вказівки до самостійної роботи з курсу “Програмні засоби проектування електромеханічних систем” для студентів денної та заочної форм навчання напряму підготовки спеціальності 141 "Електроенергетика, електротехніка та електромеханіка" – / Укл.: Л.О.Бондаренко, О.О.Каплієнко. – Запоріжжя: ЗНТУ, 2015.– с.70. №5706е.