

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет
науки і технологій**

Кафедра «Економічної інформатики»

В авторській редакції

**ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ТА
АДМІНІСТРУВАННЯ БАЗ ДАНИХ**

Навчально-методичні рекомендації
до виконання лабораторних робіт

Частина 1

ДНІПРО
2024

Упорядники:

Л. М. Бандоріна, Л. М. Савчук, К. О. Удачина, Р. В. Савчук

*Електронний аналог
друкованого видання*

Схвалено Групою забезпечення якості освітньої програми
«Комп'ютерні технології в бізнесі»
Протокол № 3 від 22.10.2024 р.
«Інформаційні технології та моделювання в економіці»
Протокол № 3 від 21.10.2024 р.

Т 38 Технологія проектування та адміністрування баз даних. Частина 1: навчально-методичні рекомендації до виконання лабораторних робіт / упоряд. Л. М. Бандоріна, Л. М. Савчук, К. О. Удачина, Р. В. Савчук ; Укр. держ. ун-т науки і технологій. – Дніпро : УДУНТ, 2024. – 57 с.

Навчально-методичні рекомендації призначені для використання студентами спеціальностей 126 «Інформаційні системи і технології і», 051 «Економіка» заочної та денної форм навчання (бакалаврський рівень) під час виконання лабораторних робіт з дисципліни «Технологія проектування та адміністрування баз даних».

Навчально-методичні рекомендації містять основні теоретичні положення для засвоєння матеріалу, інструкції до виконання лабораторних робіт, запитання до захисту лабораторних робіт.

ЗМІСТ

ПЕРЕДМОВА	4
ЛАБОРАТОРНА РОБОТА №1	5
ЛАБОРАТОРНА РОБОТА №2	12
ЛАБОРАТОРНА РОБОТА №3	27
ЛАБОРАТОРНА РОБОТА №4	41
ПЕРЕЛІК ПОСИЛАНЬ	56

ПЕРЕДМОВА

Мета навчальної дисципліни «Технологія проектування та адміністрування баз даних» полягає у формуванні базових знань з технології проектування та адміністрування баз даних та придбанні практичних навичок розробки автоматизованих баз даних.

Навчальна дисципліна забезпечує набуття таких передбачених освітньою програмою компетентностей:

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми в області інформаційних систем та технологій, або в процесі навчання, що характеризуються комплексністю та невизначеністю умов, які потребують застосування теорій та методів інформаційних технологій.

ФК3. Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними.

ФК4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

Відповідно до освітньої програми дисципліна спільно з іншими освітніми компонентами має забезпечити досягнення таких програмних результатів навчання:

ПРН3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

ПРН4. Проводити системний аналіз об'єктів проектування та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в інформаційних системах та технологіях.

Дане методичне видання містить методичні вказівки, інструкції до виконання завдань, запитання до захисту лабораторних робіт.

ЛАБОРАТОРНА РОБОТА №1

ТЕМА РОБОТИ: Проектування реляційної бази даних. Приведення даних до третьої нормальної форми.

МЕТА РОБОТИ: Отримання практичних навичок створення нормалізованих баз даних.

ТРИВАЛІСТЬ: 2 години.

МЕТОДИЧНІ ВКАЗІВКИ

Створення бази даних (БД) починається з проектування.

Етапи проектування БД:

- Дослідження предметної області;
- Аналіз даних (сутностей та їх атрибутів);
- Визначення відношень між сутностями та визначення первинних і вторинних (зовнішніх) ключів.

У процесі проектування визначається структура реляційної БД (склад таблиць, їх структура та логічні зв'язки) [1]. Структура таблиці визначається:

- вмістом стовпчиків,
- типом даних й розмірами стовпчиків,
- ключами таблиці.

При проектуванні бази мають виконуватися принципи нормалізації [2]:

- a) в кожній таблиці БД не повинно бути полів, що повторюються;
- b) в кожній таблиці має бути унікальний ідентифікатор (первинний ключ);
- c) кожному значенню первинного ключа має відповідати достатня інформація про тип сутності чи про об'єкт таблиці (наприклад, інформація про успішність, групу чи студентів);
- d) зміна значень в полях таблиці не має впливати на інформацію в інших полях (крім змін в полях ключа).

Як правило, бази даних MS Access є багатотабличними базами даних. Тому важливим і досить трудомістким процесом є проектування структури баз даних, що полягає, в першу чергу, у ефективному розподілі даних між таблицями. В базі даних, що складається з безлічі таблиць, встановлюються зв'язки (відносини) між цими таблицями за допомогою співпадаючих полів (ключових полів).

Існують **чотири типи відносин**: «*один-до-одного*», «*один-до-багатьох*», «*багато-до-одного*», «*багато-до-багатьох*».

Відношення «*один-до-одного*» створюється в тому випадку, якщо обидва пов'язаних поля виступають ключовими або мають унікальні індекси. Даний тип відношення означає, що кожен запис в одній таблиці відповідає тільки одному запису в іншій таблиці.

Відношення «*один-до-багатьох*» створюється в тому випадку, коли одне з полів виступає ключовим. Даний тип відношення передбачає, що кожному запису в одній таблиці відповідає багато записів в іншій таблиці.

Відношення «*багато-до-одного*» аналогічно попередньому типу «*один-до-багатьох*» і залежить від точки зору на відношення. Наприклад, багато замовлень надійшло на один і той же товар.

Відношення «*багато-до-багатьох*» фактично виступає двома відношеннями «*один-до-багатьох*» з третьою таблицею, первинний ключ якої складається з полів зовнішнього ключа двох інших таблиць.

Визначення умов цілісності даних

Умовою цілісності даних називають набір правил, що використовують в Access для підтримки зв'язків між записами в зв'язаних таблицях. Ці правила роблять неможливим випадкові видалення або зміни зв'язаних даних.

Умови цілісності даних виконуються при наступних умовах:

- зв'язане поле головної таблиці є ключовим полем;
- пов'язані поля мають один тип даних і однакові дані;
- обидві таблиці належать одній базі даних.

Для визначення цілісності даних потрібно у вікні діалогу **Зв'язки** установити прапорець «*Забезпечення цілісності даних*».

Дана умова робить доступним наступні **два режими**:

- каскадне оновлення пов'язаних полів;
- каскадне видалення пов'язаних полів.

Якщо такі режими не встановлені, то при видаленні запису з головної таблиці, будуть з'являтися повідомлення про неможливість виконання даної операції в тому випадку, якщо в підпорядкованій таблиці є пов'язані записи.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ 1

1. Спроекувати базу даних (БД) предметної області **Замовлення**. В БД мають зберігатися дані по клієнтах, товарах, замовленнях і підприємствах, на яких працюють клієнти.

2. Визначити сутності, атрибути сутностей і основні вимоги до функцій БД.

3. Навести спроектовану базу даних, встановити зв'язки.

Основними сутностями БД «**Замовлення**» виступають: *Підприємство, Клієнти, Товари, Замовлення*.

З аналізу даних предметної області випливає, що кожній сутності необхідно призначити найпростішу двовимірну таблицю. Далі необхідно встановити логічні зв'язки між таблицями. Між таблицями *Клієнти* та *Замовлення* слід встановити такий зв'язок, щоб кожному запису з таблиці *Клієнти* відповідало кілька записів у таблиці *Замовлення*, тобто зв'язок виду «*один-до-багатьох*», оскільки в кожного клієнта може бути кілька замовлень.

Логічний зв'язок між сутностями *Товари-Замовлення* також визначається як «*один-до-багатьох*», оскільки один і той же товар може бути замовлений декілька разів.

Між сутностями *Підприємство-Клієнти* встановлюється також логічний зв'язок «*один-до-багатьох*», оскільки на одному і тому ж підприємстві можуть працювати багато співробітників-клієнтів.

4. Виконати вимоги відповідно до правил нормалізації.

Правила першої нормальної форми (1НФ):

1. Кожна сутність повинна мати ключове поле.
2. Значення атрибута екземпляра сутності має бути єдиним (атомарним).
3. Відсутність повторювання певних груп даних.

Наприклад, є таблиця (сутність) *Підприємство* (табл. 1.1), фрагмент якої наведено нижче.

Таблиця 1.1 – Сутність Підприємство (до приведення до 1НФ)

Найменування підприємства	Керівник	Адреса	Телефон	Робітник
ДМетІ	Величко О.Г.	Дніпро, Науки, 4	(050)-123-45-64	Матвієнко Олена Сергіївна Коваленко Ігор Дмитрович
УДУНТ	Сухий К.М.	Дніпро, Лазаряна, 2	(050)-676-48-37	Іванов Іван Іванович Карпенко Володимир Григорович Шевченко Наталія Олександрівна

В даній сутності визначено ключове поле *Найменування підприємства*, значення якого є унікальними, також відсутні порожні комірки, що відповідає першому правилу 1НФ.

Але в полі *Робітник* значення атрибута екземпляра сутності не є атомарним: оскільки атрибут характеризується множиною значень. ППП кожного робітника можна записати в окремий екземпляр сутності (в окремий рядок), але тоді будуть дублюватися значення інших атрибутів (полів). Отже, щоб привести дані до першої нормальної форми пропонується інформацію по робітниках винести в іншу сутність (таблицю) – *Клієнти* (табл. 1.2), ключовим полем якої визначити *Код клієнта*.

Таблиця 1.2 – Сутність Клієнти

Код клієнта	Прізвище	Ім'я	По батькові	Дата народження	Підприємство
ABC111111	Матвієнко	Олена	Сергіївна	15.01.1985	ДМетІ
ABC111112	Коваленко	Ігор	Дмитрович	05.11.1977	ДМетІ
ABC111113	Іванов	Іван	Іванович	25.03.1980	УДУНТ
ABC111114	Карпенко	Володимир	Григорович	05.11.1983	УДУНТ
ABC111115	Шевченко	Наталія	Олександрівна	17.05.1989	УДУНТ

Таблиця *Підприємство* виглядатиме наступним чином (табл. 1.3).

Таблиця 1.3 – Сутність Підприємство (після приведення до 1НФ)

Найменування підприємства	Керівник	Адреса	Телефон
ДМетІ	Величко О.Г.	Дніпро, Науки, 4	(050)-123-45-64
УДУНТ	Сухий К.М.	Дніпро, Лазаряна, 2	(050)-676-48-37

Також є сутності *Замовлення* і *Товари* (табл. 1.4-1.5).

Таблиця 1.4 – Сутність Замовлення

Код клієнта	Код товару	Дата замовлення	Замовлено	Дата продажу	Продано	Ціна	Сума проданого товару
ABC111111	3	12.01.2024	5	15.01.2024	5	10537,00	52685,00
ABC111112	2	05.01.2024	10	05.01.2024	7	15528,50	108699,50

Таблиця 1.5 – Сутність Товари

Код товару	Категорія	Найменування
1	Канцтовари	Ручка
2	Ноутбуки	HP
3	ПК	Fujitsu

В таблиці *Замовлення* з метою уникнення дублювання даних окремих ключових полів, наприклад: *Код клієнта*, *Код товару*, *Дата замовлення* – визначено складений ключ, який включає три поля: *Код клієнта*, *Код товару*, *Дата замовлення*.

В таблиці *Товари* ключовим полем виступає *Код товару*.

На даному етапі виконані всі вимоги 1НФ.

До сутностей, які знаходяться в 1НФ, слід застосувати вимоги другої нормальної форми (2НФ):

1. База даних повинна відповідати вимогам 1НФ.
2. Неключові поля мають залежати від усього складеного ключа.

Отже, треба переконатися, чи виконується друге правило 2НФ для таблиці *Замовлення*. Наприклад, поля: *Замовлено*, *Дата продажу*, *Продано* – залежать від трьох полів складеного ключа: *Код клієнта*, *Код товару*, *Дата замовлення*, що відповідає 2НФ, а поле *Ціна* залежить від *Коду товару*, але не залежить від інших полів складеного ключа. Тому поле *Ціна* слід винести в таблицю *Товари*.

Для приведення до третьої нормальної форми (3НФ) слід дотримуватися наступних правил:

1. База даних повинна відповідати вимогам 2НФ.
2. Неключове поле не повинно характеризуватися іншим неключовим полем.
3. Не повинно бути обчислюваних полів.

На даному етапі всі сутності відповідають вимогам 3НФ, але в таблиці *Замовлення* є обчислюване поле *Сума проданого товару*. Дане поле треба видалити з таблиці і в подальшому його можна створити в інших об'єктах MS Access, наприклад, в запиті.

Отже, після приведення даних до 3НФ сутності визначатимуться наступними атрибутами.

Сутність *Клієнти* характеризується наступними атрибутами: код клієнта, прізвище, ім'я, по батькові, дата народження, телефон, індекс, країна, місто, адреса, підприємство, розміщення організації.

Сутність *Підприємство*: найменування підприємства, ПІБ керівника, адреса.

Сутність *Товари*: код товару, категорія, найменування товару, ціна.

Сутність *Замовлення*: код клієнта, код товару, дата замовлення, дата продажу, замовлено, продано, примітка.

Нижче представлено спроектовану базу даних, встановлено зв'язки, ключові поля виділено курсивом (рис. 1.1).



Рисунок 1.1 – Схема даних бази даних Замовлення

КОНТРОЛЬНІ ПИТАННЯ ДО ЗАХИСТУ ЛАБОРАТОРНОЇ РОБОТИ 1

1. Перелічити етапи проектування бази даних.
2. Пояснити, для чого застосовується нормалізація реляційних баз даних.
3. Навести правила нормалізації БД.

4. Перелічити види логічних зв'язків в БД.

5. Пояснити основні вимоги до полів, між якими встановлюються зв'язки. Чи можуть бути встановлені зв'язки між полями з різними іменами, з різними типами даних?

6. Перерахувати можливі типи відносин між таблицями бази даних. Пояснити типи відносин між таблицями створеної бази даних.

ЛАБОРАТОРНА РОБОТА № 2

ТЕМА РОБОТИ: Розробка бази даних в MS Access.

МЕТА РОБОТИ: Отримання практичних навичок створення структури таблиць, організації ефективного пошуку, заміни і фільтрації даних в таблицях MS Access.

ТРИВАЛІСТЬ: 2 години.

МЕТОДИЧНІ ВКАЗІВКИ

База даних – це набір взаємозв'язаних таблиць з інформацією, що складається з рядків (записів) і стовпців (полів). Усі записи складаються з однакових полів. Дані для одного поля у всіх записах мають однаковий тип, але різні поля можуть мати різні типи даних.

Кожен рядок таблиці повинен бути унікальним. Для забезпечення унікальності рядків використовуються ключі – поля, значення яких не можуть повторюватися в жодній записі бази даних.

MS Office Access складається з окремих компонентів, які використовуються для зберігання і представлення інформації:

Таблиці – об'єкти, що являються основою бази даних і використовуються для збереження даних.

Форми – об'єкти, що використовуються для введення даних і відображення їх на екрані в необхідному вигляді.

Звіти – об'єкти, призначені для створення документів, які згодом можуть бути роздруковані або включені в документи іншої програми.

Запит – об'єкт, який дозволяє вилучити вибірку інформацію з бази даних.

Макрос – об'єкт, призначений для автоматизації часто виконуваних дій користувача.

Тип даних:

Короткий текст. Буквено-цифрові дані (до 255 байт (символів) за замовчуванням).

Довгий текст. Може містити ті ж типи даних, що й прості текстові поля. Відмінністю є те, що розмір поля *МЕМО* не обмежений 255 символами, а може містити до 1 ГБ символів.

Числовий. Автоматично виконується перевірка на відповідність введеного значення числовому типу. Тільки над числовими полями можливо виконання математичних операцій.

Для вказання числа десяткових знаків використовується властивість з назвою *Кількість десяткових знаків*. За замовчуванням задається атрибут *Авто*.

3. *Грошовий.* За замовчуванням *грошовий* тип приймається з 2 знаками після коми, кількість яких при бажанні можна змінити (від 1 до 4 знаків після коми).

4. *Лічильник.* Поля типу *Лічильник* призначені для зберігання даних, значення яких не редагуються, а встановлюються автоматично при додаванні кожного нового запису в таблицю. Їх значення збільшуються на 1 при додаванні кожного нового запису або можуть бути будь-яким випадковим числом (дану властивість можна задати в рядку *Нові значення – Послідовні* або *Випадкові*).

5. *Дата та час.* Використовується для зберігання календарних дат значень часу, в яких дані надані в спеціальному форматі. Дозволяє виконувати обчислення в одиницях виміру часу: у хвилинах, секундах, годинах, днях, місяцях і роках. Можлива установка різних *Форматів* поля у вікні *Властивості* поля (*Повний, Довгий, Середній, Короткий* та ін.). При введенні дат буде виконуватися перевірка коректності даних.

6. *Логічний.* Використовується для зберігання даних, які можуть приймати одне з двох можливих значень – *Так\Ні*.

7. *Поле об'єкта OLE.* Дозволяє зберігати в таблицях зображення та інші двійкові малюнки (наприклад, ел. таблицю EXCEL, документи MS Word, картинки, діаграми, звукові фрагменти, відеофайли) розміром до 2 ГБ.

8. *Гіперпосилання.* Містить адресу посилання на документ або файл в Інтернеті, інтрамережі, локальній мережі або на локальному комп'ютері.

9. *Обчислюваний.* Дозволяє створювати вирази, які використовують дані з одного або кількох полів. На основі виразу можна призначити різні типи даних результату.

9. *Майстер підстановок.* Вибір цього типу даних запускає *Майстер підстановок*, призначений для створення поля, в якому пропонується вибір значень зі списку, що містить набір постійних значень або значень з іншої таблиці.

При завершенні створення таблиці в режимі конструктора слід закрити вікно *Конструктора*, потім перейти на вкладку *Таблиці*, виконати подвійне клацання по імені сформованої в конструкторі таблиці, щоб відкрити її для заповнення даними.

Значення властивості *МАСКА ВВОДУ* може містити один обов'язковий і два необов'язкових розділи, відокремлених крапкою з комою (;):

- перший розділ (обов'язковий): містить безпосередньо маску вводу, що складається із символів, наприклад: (999)-000-00-00;

- другий розділ (необов'язковий): вказує на те, чи зберігаються символи в полі: якщо встановити 0, символи зберігаються з даними, а якщо 1 або нічого не вказувати – символи лише відображаються, але не зберігаються, що дозволяє економити місце в пам'яті;

- третій розділ (необов'язковий): містить символ, який вказує на окремий символ, що використовується як показник місця заповнення. За замовчуванням використовується символ підкреслення (); можна вказати будь-який символ ASCII, пробіл необхідно вказати в лапках (" ").

Символи, які використовуються для **МАСКИ ВВОДУ** [3]:

- 0 – користувач має ввести цифру (від 0 до 9).

- 9 – користувач може ввести цифру (від 0 до 9).

- # – користувач може ввести цифру, пробіл або знак плюса чи мінуса. У разі пропуску цієї позиції вводиться пробіл.

- L – користувач має ввести букву.

- ? – користувач може ввести букву.

- A – користувач має ввести букву або цифру.

- a – користувач може ввести букву або цифру.

- & - користувач має ввести або символ, або пробіл.

- C - користувач може ввести символи або пробіли.

- . , : ; - / показники місця заповнення десяткової коми та роздільника розрядів, роздільники дати й часу. Вибраний символ залежить від регіональних параметрів Microsoft Windows.

- > ссі наступні символи перетворюються на символи верхнього регістру.

- < усі наступні символи перетворюються на символи нижнього регістру.

- ! вимагає, щоб маска вводу заповнювалася зліва направо замість справа наліво.

- \ усі наступні суміжні символи відображаються літерами.

- "" символи, узяті в подвійні лапки, відображаються літерами.

Якщо користувач визначає *МАСКУ ВВОДА* і одночасно задає для того ж поля значення властивості *ФОРМАТ ПОЛЯ*, то при виведенні даних пріоритет має

властивість *ФОРМАТ ПОЛЯ*. Це означає, що при форматуванні даних збережена маска введення ігнорується. Дані в базовій таблиці при цьому не змінюються; властивість *ФОРМАТ ПОЛЯ* визначає тільки режим відображення даних.

У властивостях певного поля в *Конструкторі* можна вказати *Правило перевірки*, яке буде діяти при веденні даних користувачем в таблицю. Для створення правил можна використовувати наступні символи (табл. 1.1) [4].

Таблиця 1.1 – Символи для створення правил перевірки

Оператор	Функція	Приклад
NOT	Перевірка протилежних значень.	NOT > 10 (те ж саме, що <=10).
IN	Перевірка значень, які дорівнюють наявним елементам у списку. Значення порівняння має виглядати як список елементів, записаних через кому та взятих у дужки.	IN ("Токіо", "Париж", "Київ")
BETWEEN	Перевірка діапазону значень. Слід використовувати два значення для порівняння (менше та більше), записані через роздільник AND .	BETWEEN 100 AND 1000 (те ж саме, що >=100 AND <=1000)
LIKE	Зіставлення рядків структури в текстових полях.	LIKE "Geo*"
IS NOT NULL	Змушує користувачів вводити значення в полі. Результат такий же, як коли для властивості поля Обов'язково встановлено значення Так . Однак якщо властивість Обов'язково ввімкнено, а користувач не вводить значення, у програмі Access відображається не дуже зрозуміле повідомлення про помилку. Зазвичай, базою даних зручніше користуватися, якщо в ній використовується функція IS NOT NULL і введено зрозуміле повідомлення в полі властивості Текст перевірки .	IS NOT NULL
AND	Указує, що мають виконуватися всі частини правила перевірки.	>= #01.01.2024# AND <=#03.06.2024#
OR	Указує, що мають виконуватися не всі, а лише деякі правила перевірки.	січень OR лютий
<	Менше.	
<=	Менше або дорівнює.	
>	Більше.	
>=	Більше або дорівнює.	
=	Дорівнює.	
<>	Не дорівнює.	

Для **ПОШУКУ** і **ЗАМІНИ** значень полів, записів або імен файлів допускається застосування наступних символів в діалогових вікнах Пошук і Заміна або в запитах, командах і вираженнях (табл. 2.2) [5].

Таблиця 1.2 – Символи для пошуку і заміни даних

Символ	Пояснення	Приклад
*	Відповідає будь-якій кількості символів (навіть нульовій). Цей символ узагальнення можна використати як перший або останній символ у рядку символів.	За запитом "св*" буде знайдено "св", "свято", "світ" і "свіжий"
?	Відповідає будь-якій одній букві.	За запитом "с?н" буде знайдено слова "сон", "син" і "сан"
[]	Відповідає будь-якому одному символу в дужках.	За запитом "с[ао]н" буде знайдено слова "сан" і "сон", але не слово "син"
!	Відповідає будь-якому символу не в дужках.	За запитом с[!ао]н буде знайдено слово "син", але не слова "сан" і "сон"
-	Відповідає будь-якому символу з діапазону. Діапазон слід указувати у висхідному порядку (від А до Я, а не від Я до А).	За запитом "к[а-в]т" буде знайдено слова "кат", "кбт" і "квт"
#	Відповідає будь-якому одному числовому символу.	За запитом 1#3 буде знайдено 103, 113 і 123

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ 2

1. **Запустити MS Access.**
2. **Створити нову базу даних *Замовлення.***

Для створення бази даних потрібно виконати команду **Нова база**, потім задати ім'я нової бази даних з розширенням .accdb (наприклад, *Клієнти.accdb*). Відкриється вікно пустої бази даних.

3. **Створити таблицю *Контакти* за допомогою *Шаблонів*:** панель швидкого доступу *Створення\Таблиці\Шаблони*. Ввести в неї 5 записів.

Створити таблицю **Клієнти** в режимі *Таблиці: Створення\Таблиці\Таблиця*. У таблицю включити такі поля: *код клієнта, прізвище, ім'я, по батькові, дата народження, телефон, індекс, країна, місто, адреса, підприємство, розміщення організації.*

4. **Створити дві таблиці** («Підприємство» та «Товари») за допомогою режиму таблиці: **Створення\Таблиці\Таблиця**.

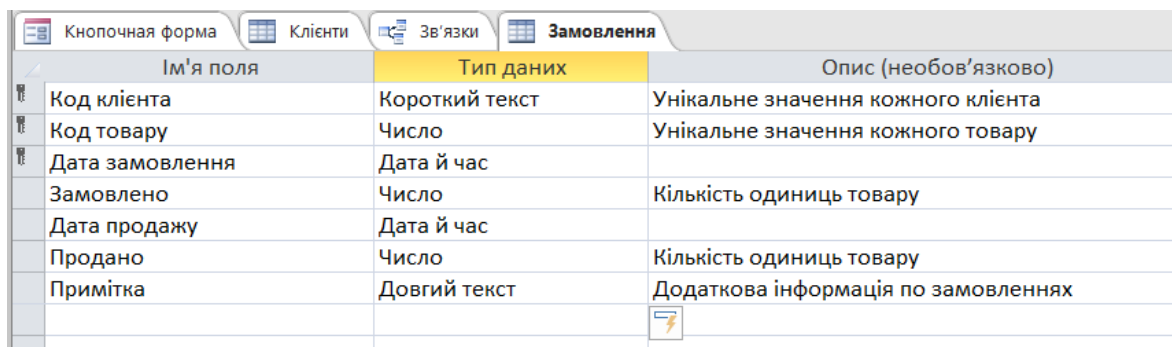
Таблиця «Підприємство» складається з наступних полів: *найменування підприємства, ПІБ керівника, адреса*.

Таблиця з ім'ям «Товари» містить поля: *код товару, категорія, найменування товару, ціна*.

При введенні даних в таблицю «Підприємство» ввести дані по підприємствах, які були вказані в таблиці «Клієнти». В якості коду підприємства використовувати тризначні числа.

Після введення кількох рядків даних можна зберегти таблицю (**Кнопка «Office»\Зберегти** або кнопка **Зберегти** на панелі інструментів).

5. **Створити таблицю Замовлення** за допомогою конструктора: панель швидкого доступу **Створення\Таблиці\Конструктор таблиць**. Таблиця **Замовлення** має наступну структуру (рис. 2.6):



Ім'я поля	Тип даних	Опис (необов'язково)
Код клієнта	Короткий текст	Унікальне значення кожного клієнта
Код товару	Число	Унікальне значення кожного товару
Дата замовлення	Дата й час	
Замовлено	Число	Кількість одиниць товару
Дата продажу	Дата й час	
Продано	Число	Кількість одиниць товару
Примітка	Довгий текст	Додаткова інформація по замовленнях

Рисунок 2.6 – Конструктор таблиці **Замовлення**

Створення таблиць у вікні **Конструктора** надає найбільш широкі можливості з визначення параметрів створюваної таблиці.

При виборі режиму **Конструктор таблиці** з'явиться вікно конструктора таблиці (рис. 2.7):

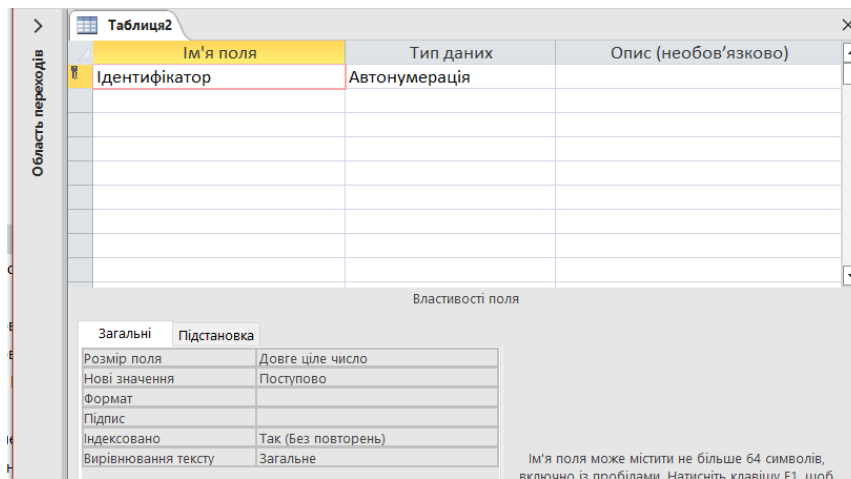



Рисунок 2.7 – Створення таблиці в режимі **Конструктора**

6. **Встановити для числових полів таблиці кількість знаків після коми** – 2, використовуючи при цьому *Властивості поля: Розмір поля, Формат поля, Кількість знаків після коми*.

7. **Для поля *Ціна* у властивості *Підпис*** ввести текст «Ціна, грн». Перевірте дію даної властивості в Режимі таблиці.

8. **Заповнити *Правило перевірки* для всієї таблиці** (команда **Аркуш властивостей** на панелі інструментів вікна **Конструктора** ) , в яких забезпечити контроль даних за наступними умовами:

- значення поля *Дата продажу* не повинно бути раніше значення поля *Дата замовлення*;
- значення поля *Продано* не повинно перевищувати значення поля *Замовлено*.

9. **Заповнити *Правило перевірки* для поля *Дата замовлення*** так, щоб база даних містила замовлення від 02.02.2024 року по сьогоднішню дату; у властивості ***Текст перевірки*** вказати текст, який буде виведено на екран у випадку введення даних, що не відповідають правилу перевірки (рис. 2.8):

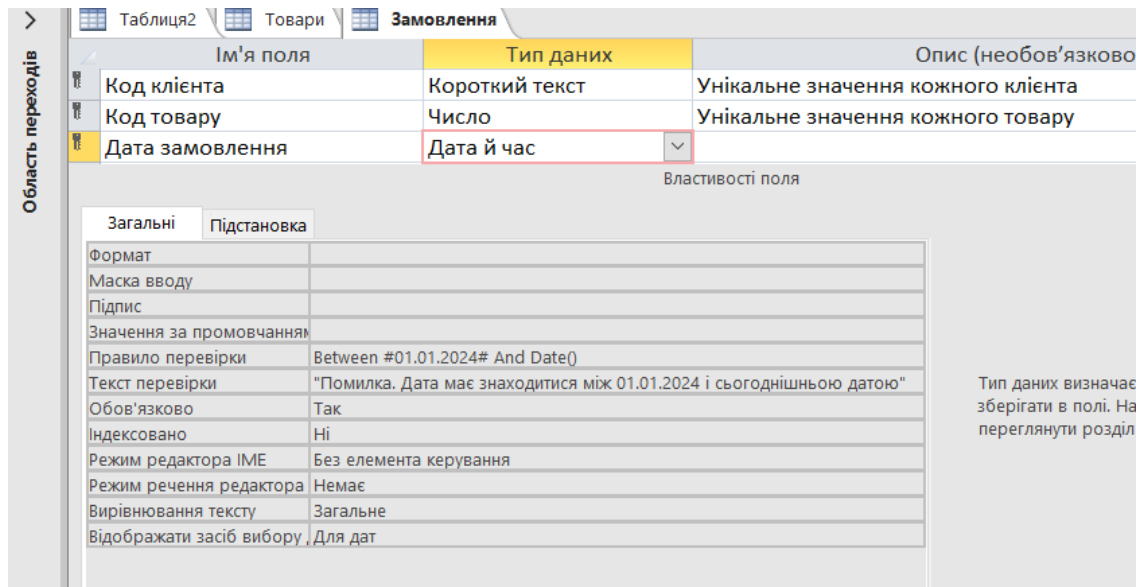


Рисунок 2.8 – Налаштування властивостей поля

10. У **властивостях поля** *Код клієнта* створити **маску вводу**, що складається з двох обов'язкових і однієї необов'язкової букви, розташованих на початку коду (букви повинні автоматично переводитися у верхній регістр) і п'яти обов'язкових цифр;

11. **Заповнити таблицю *Замовлення*** не менше ніж 10 записами.

12. **Виконати модифікацію структури таблиць:**

- у таблиці «**Підприємство**» додайте поле **Телефон**
- в таблиці «**Клієнти**» видаліть поле **Розміщення організації**.

13. **За допомогою режиму *Конструктор*** (кнопка **Конструктор** вікна бази даних) перевірте типи даних полів таблиць «**Клієнти**». Зверніть особливу увагу на тип даних поля дата народження. Відкорегуйте типи даних тих полів, для яких не встановлено відповідність між значеннями даних, що вводяться, і їх типом (список, що розкривається стовпця *Тип даних* у вікні **Конструктор**).

14. **За допомогою вікна *Конструктор* для полегшення введення даних в таблицю** ввести значення за замовчуванням для поля ***Країна*** – *Україна*, а для поля ***Місто*** – *Дніпро*.

15. Заповнити таблицю **Клієнти** не менш ніж 15 записами так, щоб:

- серед клієнтів були громадяни різних країн, у тому числі австрійці, і жителі різних міст, в тому числі *Києва*;
- кілька людей працювали в *ДМетІ*;

– серед клієнтів були співробітники СП «Айрит», що працюють у філіях Києва та Австрії;

– серед клієнтів були народжені в березні і в день виконання лабораторної роботи.

16. **Зберегти структуру створених таблиць в базі «Клієнти».**

17. **Відкрити таблицю Клієнти.**

18. **Використовуючи команди панелі Головна \Додатково** (Режим таблиці):

– закріпити поля *Прізвище, Ім'я, По батькові, Телефон* так, щоб вони завжди були видні на екрані. Для цього помістіть поля *Прізвище, Ім'я, По батькові, Телефон* поруч один з одним; виділіть їх і виконайте команду **Закріпити поля**;

– приховати поле *Примітка*. Для цього виділити поле і виконати команду **Приховати поля**;

– змінити розмір шрифту, колір букв, фон таблиці (**Головна \ Шрифт**).

19. **Знайти дані про клієнтів** (піктограма *Знайти* 

– працюючих в *ДМетІ* (рис. 2.9);

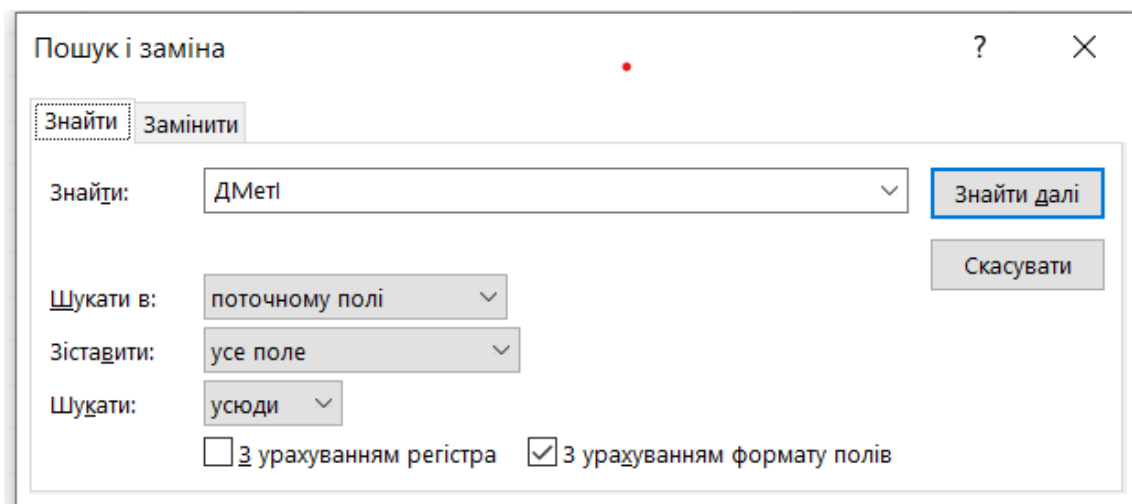


Рисунок 2.9 – Пошук клієнтів працюючих в ДМетІ

– народжених сьогодні; припустимо, сьогодні 12.10. 2024 (рис. 2.10);

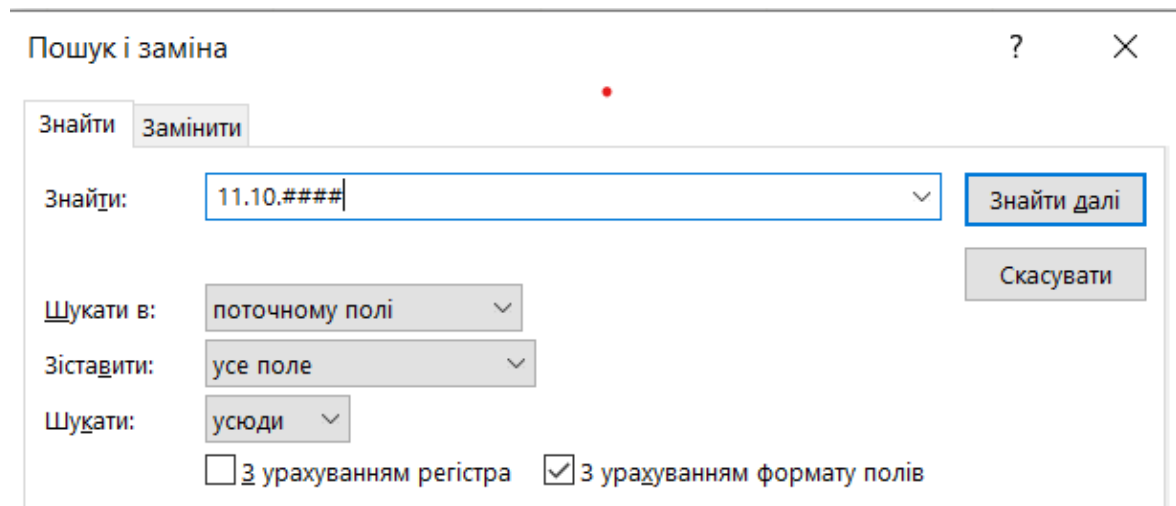


Рисунок 2.10 – Пошук клієнтів, що народилися сьогодні

– у яких цього року ювілей (вік кратний 10). Припустимо, сьогодні 12.10. 2024 (рис. 2.11);

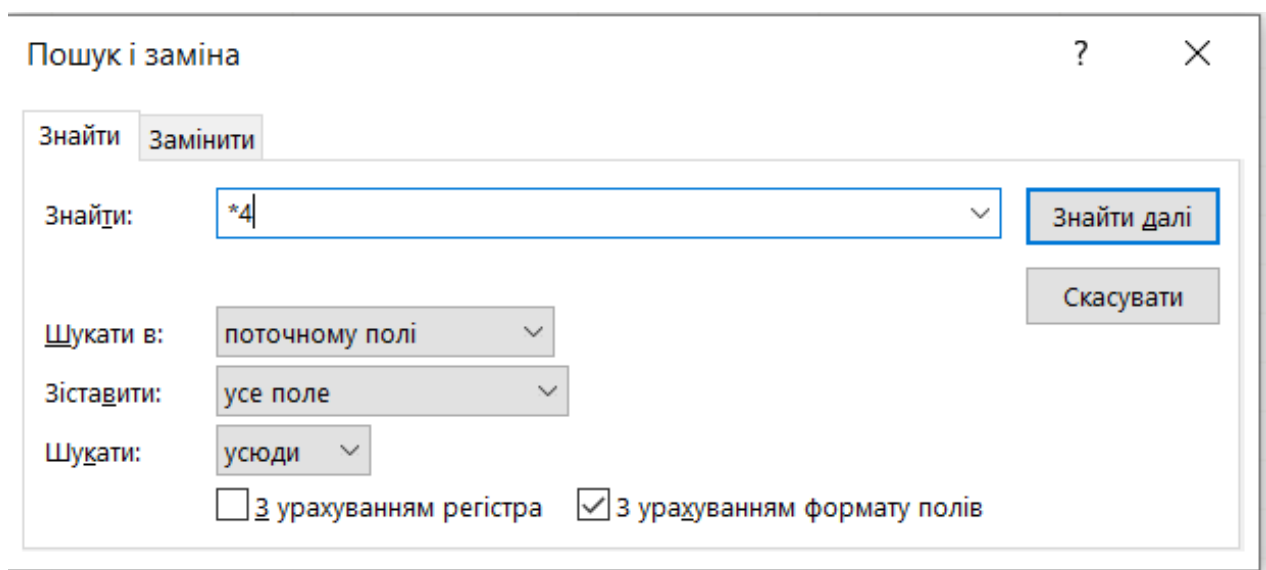


Рисунок 2.11 – Пошук клієнтів, вік яких кратний 10

20. **За допомогою фільтра за виділенням відфільтрувати дані** тільки по тим клієнтам:

– які проживають в *Україні*. Для цього виділіть у полі **Країна** значення **Україна** і виконайте **Головна\Сортування і фільтр\Виділення\Дорівнює Україна**. Для скасування фільтра виконайте команду: **Головна\Сортування і фільтр\Додатково\Очистити всі фільтри**;

– які проживають за межами України. Для цього виділити в поле **Країна** значення **Україна** і виконати команду **Головна\Сортування і фільтр\Виділення\Не дорівнює Україна**. Для скасування фільтра виконайте команду: **Головна\Сортування і фільтр\Додатково\Очистити всі фільтри**;

– імена яких починаються з букви **С**. Для цього виділити першу букву **С** будь-якого імені та виконати наступну команду: **Головна\Сортування і фільтр\Виділення\Починається з «С»**;

21. по жінках. Для цього виділити буквосполучення «вна» в кінці будь-якого по батькові і виконати наступну команду: **Головна\Сортування і фільтр\Виділення\Закінчується на «вна»**;

22. за допомогою команди **Головна\Сортування і фільтр \Додатково\Змінити фільтр** показати дані тільки по клієнтах:

– з Дніпра, які працюють в ДМетІ. Для цього виконайте команду **Змінити фільтр**, потім зі списків полів **Місто** та **Підприємство** обрати відповідно: **Дніпро** і **ДМетІ**, як показано на рисунку (рис. 2.12); після виконати команду: **Головна\Сортування і фільтр\Застосувати фільтр**. Для скасування фільтра виконати команду: **Головна\Сортування і фільтр\Додатково\Очистити всі фільтри**;

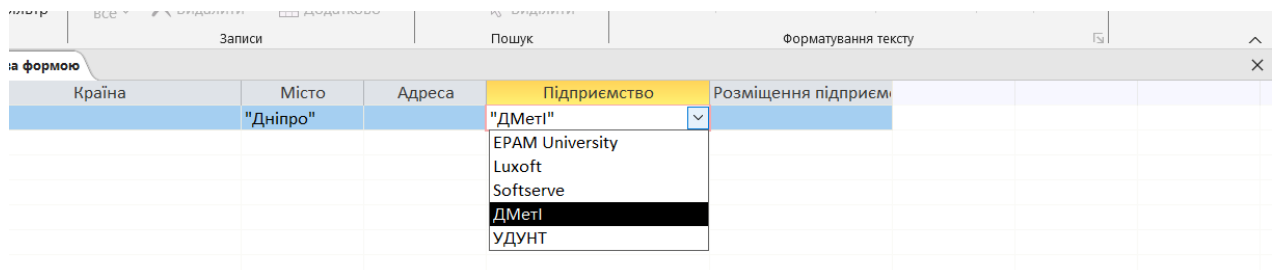


Рисунок 2.12 – Фільтр клієнтів, які проживають у Дніпрі і працюють і ДМетІ

– клієнтів, в прізвищах яких не зустрічається буква **а**. Для цього виконати команду: **Змінити фільтр**; у полі **Прізвище** ввести такий вираз: **Not Like '*а*'** (рис. 2.13-2.14); після очистити фільтри;

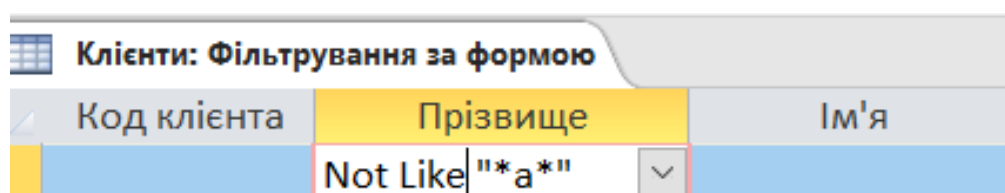


Рисунок 2.13 – Фільтр клієнтів, в прізвищах яких не зустрічається літера **а**

Код клієнта	Прізвище	Ім'я	По батькові	Дата народж.	Телефон	Індекс	
ABC00007	Творніченко	Денис	Семенович	05.03.1995			
ABC00003	Пилипенко	Андрій	Олександрович	15.03.1990	(067)-485-74-83	47865	Україна, І
ABC00004	Герцен	Евгеній	Евгенович	15.04.1985		47857	

Рисунок 2.14 – Результати фільтрації

– прізвища яких починаються з літер А, Б або В (реалізувати дану умову двома способами).

Перший спосіб: виконайте команду **Змінити фільтр**, у полі *Прізвище* ввести вираз: Like "[АБВ]*" (рис. 2.15-2.16).

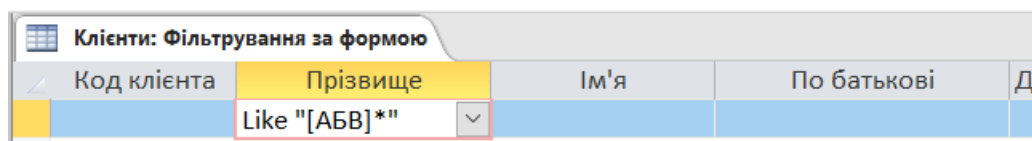


Рисунок 2.15 – Фільтр клієнтів, прізвища яких починаються з літер А, Б, В (спосіб 1)

Код клієнта	Прізвище	Ім'я	По батькові
ABC00001	Бабенко	Денис	Анатолійович
*			

Рисунок 2.16 – Результати фільтрації

Очистити фільтри.

Другий спосіб: у полі *Прізвище* ввести вираз: Like "А*", перейти на вкладку **АБО** і в полі *Прізвище* ввести вираз: Like "Б*", перейти на вкладку **АБО** і в полі *Прізвище* ввести вираз: Like "В*" (рис. 2.17).

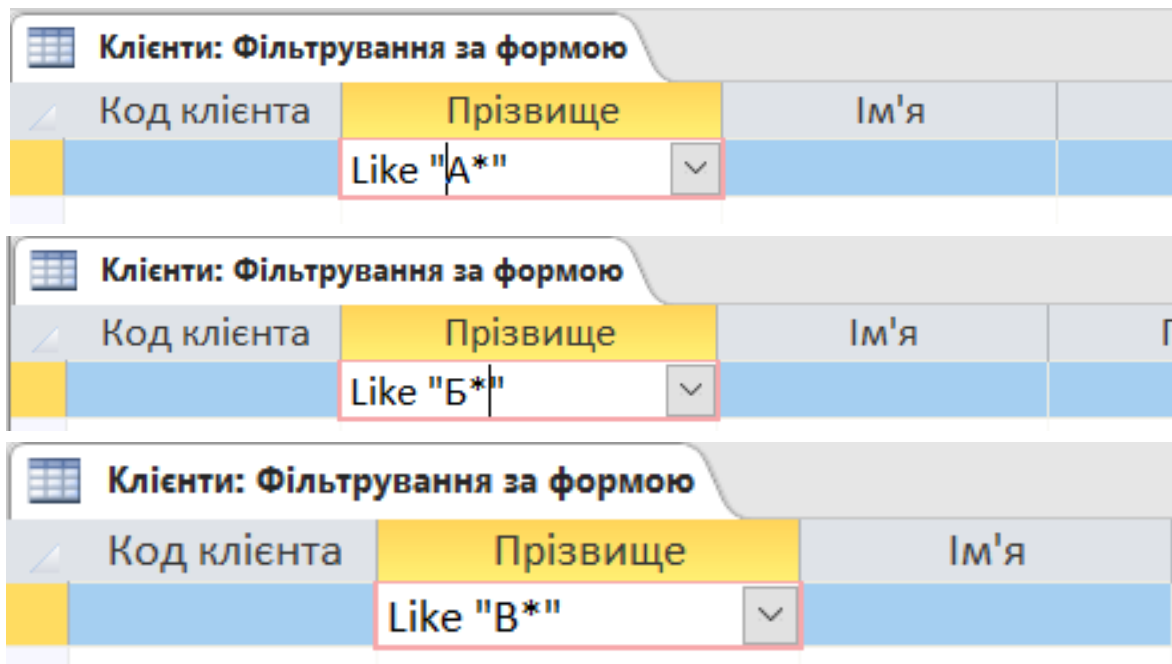


Рисунок 2.17 – Фільтр клієнтів, прізвища яких починаються з літер А, Б, В (спосіб 2)

23. За допомогою команди **Головна\Сортування і фільтр \ Додатково\Розширений фільтр** відобразити дані по клієнтах:

- у яких немає телефону: в нижній частині бланка додати поле **Телефон** (в критеріях в полі **Телефон** вказати Is Null) (рис. 2.18-2.19);

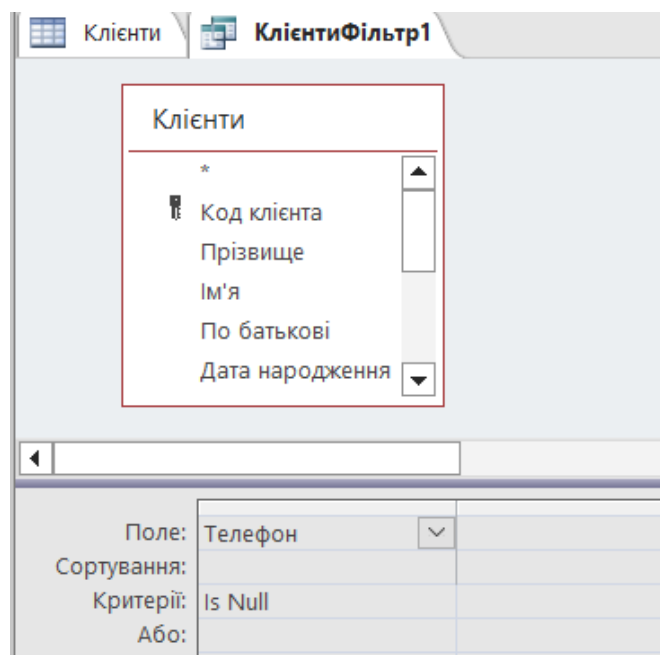


Рисунок 2.18 – Розширений фільтр клієнтів, у яких немає номера телефону

Код клієнта	Прізвище	Ім'я	По батькові	Дата народу	Телефон
ABC00007	Творніченко	Денис	Семенович	05.03.1995	
ABC00001	Бабенко	Денис	Анатолійович	28.01.1987	
ABC00004	Герцен	Евгеній	Євгенович	15.04.1985	
ABC00005	Іваненко	Наталія	Іванівна	09.02.2000	

Рисунок 2.19 – Результати фільтрації

– які мають роботу (вказати вираз Is not Null в критеріях поля **Підприємство**);

– які проживають в Києві або Дніпрі і працюють в Luxoft (використовувати рядок АБО).

24. Сформувати різні варіанти критерію пошуку для визначення номера телефона і знайти у таблиці клієнта, для якого в шестизначному номері телефона ви пам'ятаєте перші дві цифри; четверту і останню цифри.

25. Зберегти зміни в базі даних.

КОНТРОЛЬНІ ПИТАННЯ ДО ЗАХИСТУ ЛАБОРАТОРНОЇ РОБОТИ 2

1. У якому випадку для створення структури таблиці доцільно використовувати **Шаблони**?

1. Як визначається тип даних полів при створенні таблиці в **Режимі таблиці**?

2. Перелічити правила створення імен полів таблиці.

3. Як в **Режимі таблиці** можна **змінити** найменування поля?
Продемонструвати зміни імені поля.

4. Перерахувати можливі варіанти додавання нових полів в таблицю в **Режимі таблиці**.

5. Як здійснюється **видалення** поля таблиці в **Режимі таблиці**?

6. Що потрібно зробити, щоб перемістити поле таблиці з одного місця на інше?

7. Для чого призначена **Маска вводу**, де вона задається?

8. Перерахувати основні символи, використовувані в **Масці вводу**, охарактеризуйте їх призначення.

9. Які типи даних є допустимими в Access?

10. Як у вікні конструктора таблиць встановлюється тип даних для полів таблиці?

11. Для чого призначений атрибут **Підпис**? Що станеться, якщо не вказувати значень цього атрибута при створенні структури таблиці у вікні конструктора?

12. Для чого призначена властивість поля **Правило перевірки**, приведіть приклад використання цієї властивості.

13. Як пов'язані між собою такі властивості поля **Правило перевірки** і **Текст перевірки**, наведіть приклад.

14. Яким чином можна за допомогою команди **Знайти** здійснювати пошук даних по всій таблиці, а не тільки в поточному стовпці.

15. Навести приклад використання фільтра за **виділенням** в таблиці «Клієнти».

16. У якому випадку необхідно використовувати команду **Змінити фільтр** при пошуку даних в таблиці.

17. Охарактеризувати відмінності у використанні критерію **I** і критерію **АБО** при фільтрації даних в таблиці.

18. Для чого здійснюється **закріплення** стовпців таблиці і як воно виконується?

19. Перерахувати можливості зміни зовнішнього вигляду таблиці бази даних.

ЛАБОРАТОРНА РОБОТА №3

ТЕМА РОБОТИ: Створення запитів в інформаційній системі.

МЕТА РОБОТИ: Отримання практичних навичок зі створення простих запитів і запитів в режимі Конструктора.

ТРИВАЛІСТЬ: 2 години.

МЕТОДИЧНІ ВКАЗІВКИ

Запит на вибірку – це інтерактивний засіб для вибору даних з однієї або декількох таблиць. Результат запиту називається вибіркою. Вибірка не зберігається в вашій базі даних. Вона створюється знову при виконанні запиту і знищується при його закритті.

Для завдання в запитах складних логічних виразів можна використовувати *Побудовник виразів*. Для цього в бланку конструктора в рядку *Критерії* потрібно встановити курсор і натиснути кнопку «Конструктор» на панелі інструментів: *Настроювання запиту / Конструктор* (рис. 3.1).

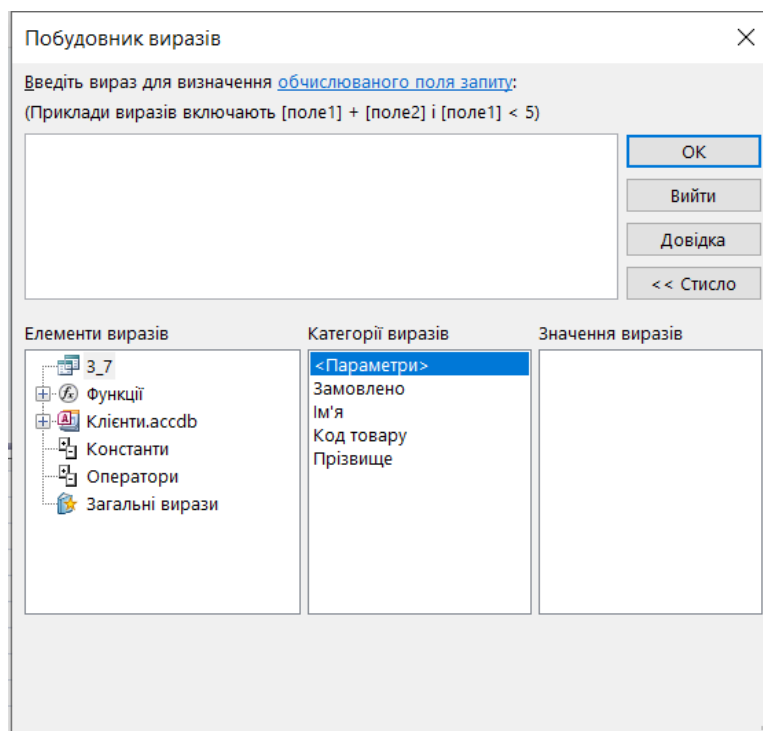


Рисунок 3.1 – Діалогове вікно *Побудовника виразів*

Вікно *Побудовника виразів* складається з декількох областей.

Поле для виразів. У верхній частині вікна *Побудовника* розташоване поле, в якому створюється вираз (*Вираз* – це поєднання математичних і логічних операторів, констант, функцій, імен полів, елементів управління і властивостей, в результаті обробки якого виходить єдине значення. Вираз може виконувати обчислення, обробляти текст або перевіряти дані). Нижче знаходиться розділ, призначений для створення елементів вираження та їх подальшої вставки в поле вираження. Допускається безпосереднє введення частини вираження у полі для виразів.

Елементи виразів. У нижній частині вікна *Побудовника* знаходяться три поля: *ліво поле, середнє поле та праве поле.*

В лівому полі розміщуються **Елементи виразів** – напки, що містять таблиці, запити, форми, об'єкти бази даних, функції, константи, оператори. (Константа – постійне (не обчислювальне) значення. Наприклад, число 210 і текст «Квартальна премія» є константами. Вираження і результат обчислення вираження константами не є).

В **середньому полі** вказуються категорії виразів – певні елементи або типи елементів для напки, заданої в лівому полі. Наприклад, якщо обрати в лівому полі вбудовані функції, то в середньому полі з'явиться список всіх категорій функцій Microsoft Access.

В **правому полі** виводиться список значень (якщо вони існують) для елементів, заданих в лівому і середньому полях. Наприклад, якщо обрати в лівому полі *Вбудовані функції* і категорію функції в середньому, то в правому полі буде виведений список всіх вбудованих функцій обраного типу.

Використання функцій у запитах

Функція **DateSerial** (*year; month; day*) – повертає значення типу Variant (Date), що відповідає вказаному року, місяцю і дню. Аргументи функції:

year – Обов'язковий. Значення типу Integer. Число від 100 до 9999 включно або числовий вираз.

month – Обов'язковий. Значення типу Integer. Будь-який числовий вираз.

day – Обов'язковий. Значення типу Integer. Будь-який числовий вираз.

Значення кожного аргументу функції **DateSerial** має лежати у відповідному діапазоні: 1 – 31 для днів і 1 – 12 для місяців. Крім того, можна використовувати числові вирази для опису відносної дати, тобто на певне число днів, місяців і років пізніше або раніше за вказану.

Наприклад.

DateSerial (Year ([Клієнти]! [Дата народження]); 3; 14) будуть визначені всі абоненти, що народилися 14 березня з роком народження, зазначеним у таблиці **Клієнти**.

Функція **Date()** – повертає значення типу Variant (Date), що містить поточну системну дату.

Функція **Day (дата)** – повертає значення типу Variant (Integer), що містить ціле число (від 1 до 31 включно), яке представляє день місяця.

Функція **Month (дата)** – повертає значення типу Variant (Integer), що містить ціле число (від 0 до 12 включно), яке представляє місяць в значенні дати.

Функція **Year (дата)** – повертає значення типу Variant (Integer), що містить ціле число, що представляє рік.

Функція **Hour (дата)** – повертає ціле число від 0 до 23, що представляє значення години.

Функція **Weekday (дата)** – повертає ціле число від 1 до 7 (неділя □ суботу), відповідне дню тижня.

Функція **DateAdd (інтервал; число; дата)** – повертає значення типу Variant (Date), що містить дату, до якої додано вказаний інтервал часу.

У функцію **DateAdd** входять три обов'язкові аргументи [6]:

інтервал (interval) – символічний вираз, що визначає інтервал часу, який нужно додати;

число (number) – числовий вираз, що визначає кількість інтервалів часу, які потрібно додати. Цей аргумент може приймати як додатні, так і від'ємні значення;

дата (date) – змінна типу Variant (Date) або літерал, що представляє дату, до якої додається інтервал.

Аргумент *інтервал* приймає такі значення (табл. 3.1).

Таблиця 3.1 – Опис параметрів, які може приймати *інтервал*

Параметр	Опис
уууу	Рік
q	Квартал
m	Місяць
y	День року
d	День
w	День тижня
ww	Тиждень
h	Година
n	Хвилина
s	Секунда

Функцію **DateAdd** можна використовувати для додавання або віднімання з дати певного інтервалу часу. Наприклад, функцію **DateAdd** можна використовувати для обчислення дати, яка настане через 30 днів після сьогоднішнього, або для обчислення часу, який настав 45 хвилин тому.

Для додавання днів до дати можна використовувати «День року» («у»), «День» («d») або «День тижня» («w»).

Функція **DateAdd** не повертає неприпустимі значення дат. Наприклад, додати один місяць до дати «31 січня»:

```
DateAdd ("m"; 1; "31-Jan-16")
```

В даному випадку функція **DateAdd** поверне дату «28-Feb-23», а не «31-Feb-23». Якщо вихідна дата – «31-Jan-24», то функція поверне «29-Feb-24», так як 2024 рік є високосним.

Якщо обчислена дата передеє 100-му році (тобто з дати було вирахувано занадто багато років), виникає помилка.

Якщо для завдання місяців використовуються їхні імена, ці імена повинні відповідати поточній властивості Календаря. Щоб мінімізувати ймовірність конфлікту, пов'язаного з невідповідністю імен місяців поточної властивості Календаря, рекомендується використовувати числові позначення місяців (формат Short Date).

Нижче наведено приклади використання функцій (табл. 2.3).

Таблиця 3.2 – Приклади використання функцій

Вираз	Пояснення
Year([Клієнти]![Дата народження])=1987	Клієнти, які народилися у 1987 році
Month(Date()) > Month([Клієнти]![Дата народження])	Клієнти, у яких день народження цього року минув
Day([Клієнти]![Дата народження])>=21	Клієнти, що народилися в останній декаді місяця
<Date()-30	Замовлення, що були оформлені не пізніше, ніж за 30 днів до поточної дати
Month([Клієнти]![Дата народження])=6	Клієнти, день народження яких у червні
DateAdd("d",10;DateSerial(Year([Клієнти]![Дата народження]);3;5))	Клієнти, що народилися у вказаному місяці, на 10 днів пізніше вказаної дати
Between DateAdd("d",-12;Date()) And Date()	Замовлення, які були виконані за останні 12 днів

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ 3

1. **Запустити MS Access.**

2. **Відкрити базу даних Замовлення.** Створити наступні варіанти простих запитів.

Створити простий запит з ім'ям «Клієнт 1» за допомогою Майстра запитів. Для створення простого запиту необхідні тільки поля *Прізвище, Ім'я, По батькові і Телефон* таблиці «Клієнти». Відсортувати дані за прізвищем.

Для цього виконати такі дії:

– На стрічці виконати *Створення \ Запити \ Майстер запитів \ Майстер простих запитів* (рис. 3.2).

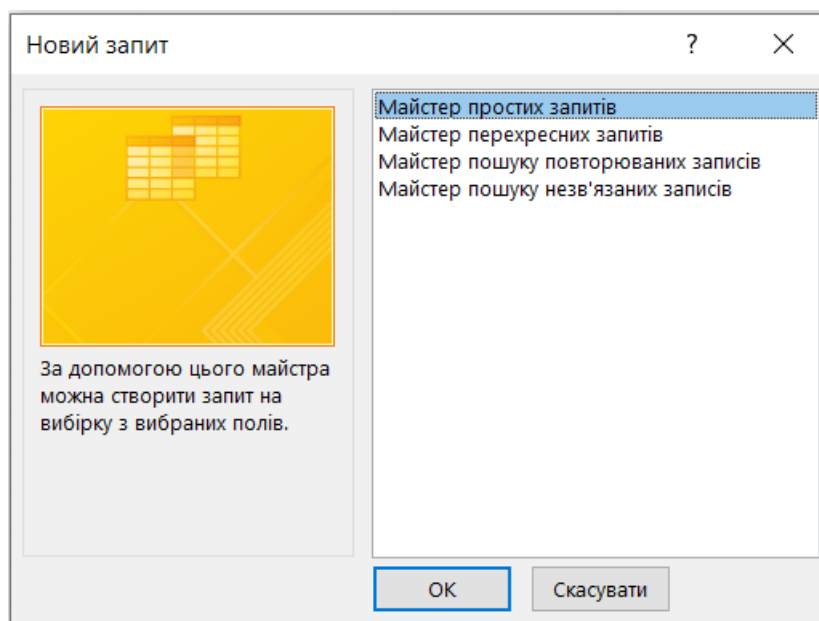



Рисунок 3.2 – Стартове діалогове вікно створення нового запиту за допомогою Майстра запитів

На екрані з'явиться майстер створення простих запитів, в якому зі списку таблиць та запитів вибрати таблицю **Клієнти**, після чого список «Доступні поля» буде відображати перелік полів з таблиці **Клієнти**. За допомогою кнопки  перемістити потрібні поля зі списку «Доступні поля» в список «Вибрані поля» та натиснути кнопку Далі (рис. 3.3).

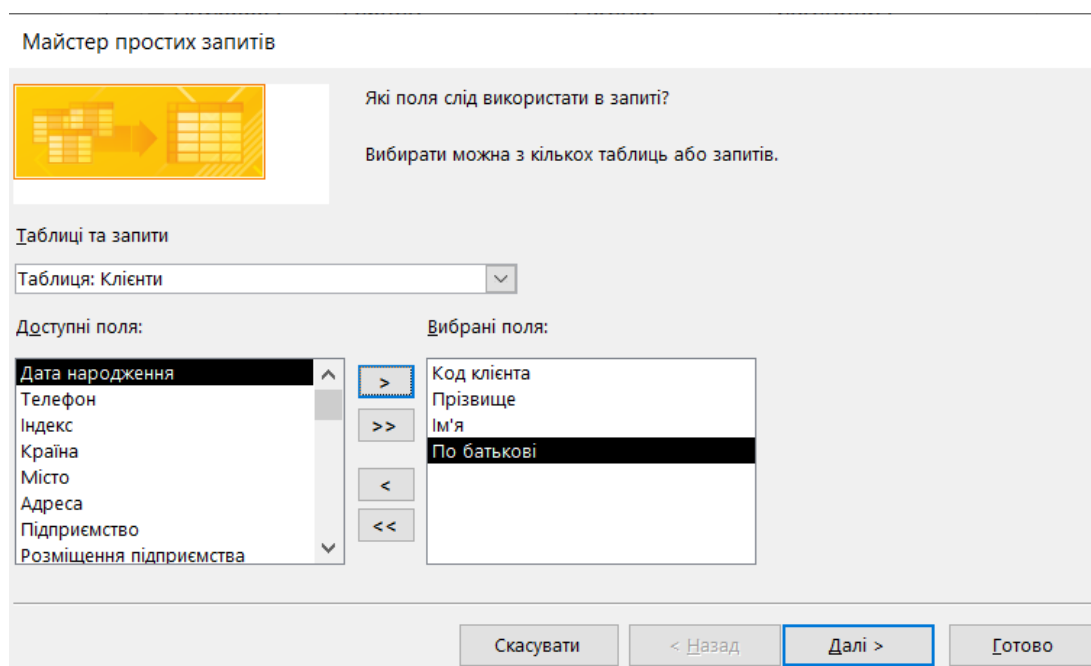


Рисунок 3.3 – Діалогове вікно вибору полів при створенні простого запиту за допомогою Майстра запитів

– На наступному кроці майстра вказати ім'я створюваного запиту. Для редагування макету запиту вибрати перемикач Змінити структуру. Натиснути кнопку Готово (рис. 3.4).

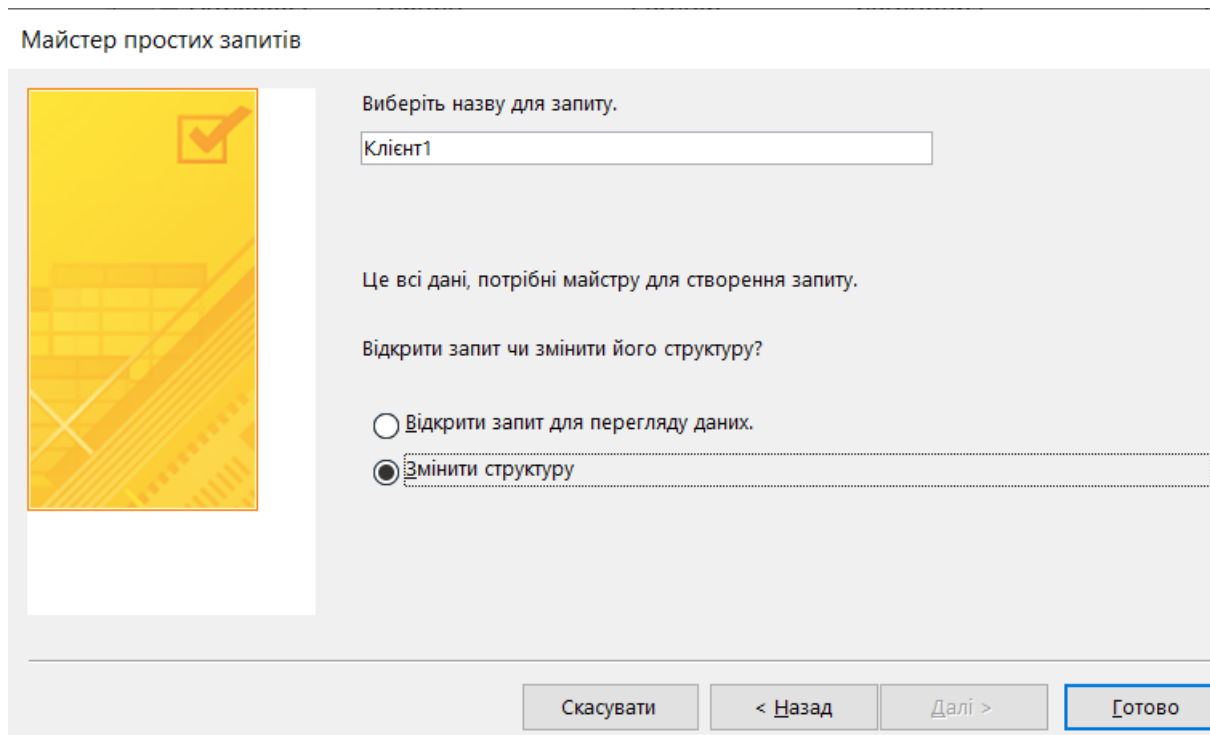


Рисунок 3.4 – Діалогове вікно введення та вибору подальшого режиму перегляду запиту за допомогою Майстра запитів

– На екрані з'явиться вікно конструктора запиту *Клієнт1*, в верхній частині знаходиться область таблиць, в нижній – бланк запиту. Для сортування даних за прізвищем в алфавітному порядку слід в бланці запиту в рядку *Сортування* для поля *Прізвище* встановити *За зростанням* (рис. 3.5).

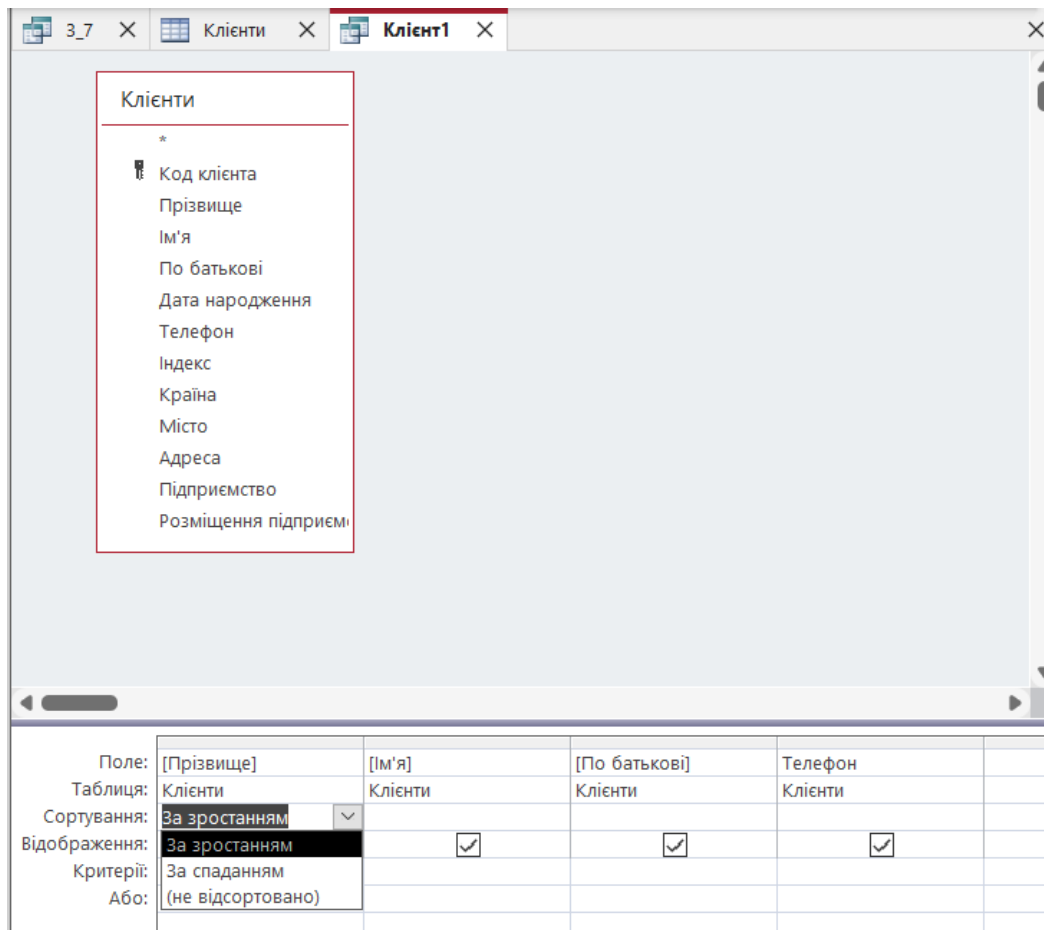


Рисунок 3.5 – Бланк запиту «Клієнт 1»

Для виконання запити слід натиснути команду *Пуск* (кнопка з червоним знаком оклику на стрічці).

3. **За допомогою Макета запити створити запит «Клієнт 2»**, в якому будуть відомості тільки про клієнтів з України, при цьому поле *Країна* не виводити на екран.

Для цього виконати наступні дії:

- На стрічці виконати *Створення \ Запити \ Макет запитів*.
- Додайте таблицю **Клієнти**, натиснувши в контекстному меню Додавання таблиці і вибравши потрібну таблицю.
- Додати в бланк запити поля: *Код клієнта*, *Прізвище*, *Ім'я*, *По батькові*, *Дата народження*, *Телефон* і *Країна*. В полі *Країна* в рядку **Критерії** ввести «Україна», в рядку **Відображення** прибрати позначку в полі *Країна* (рис. 3.6).

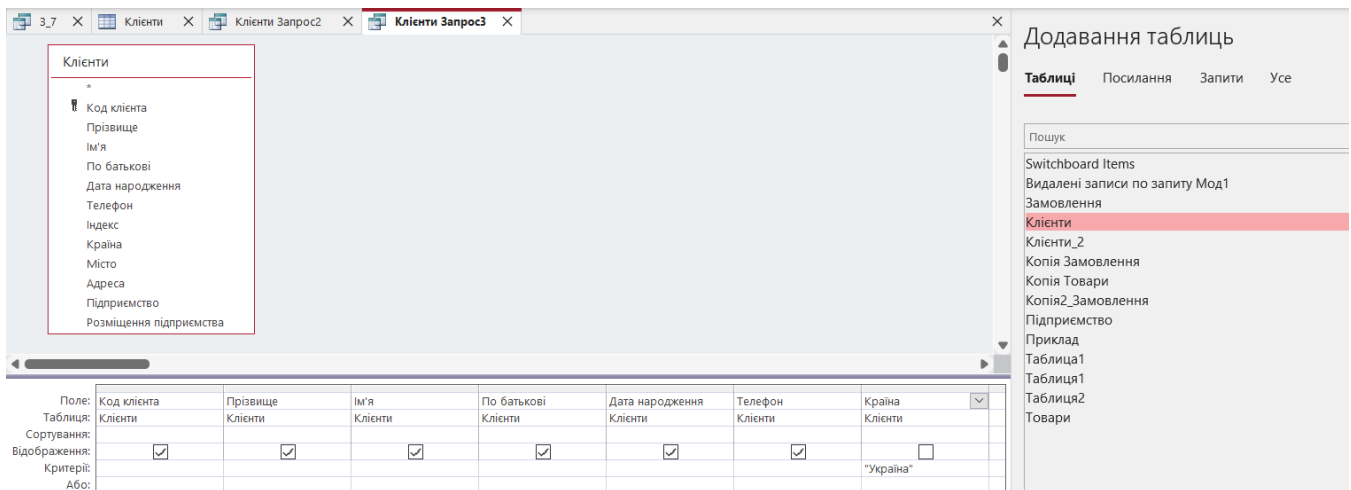


Рисунок 3.6 – Бланк запити «Клієнт 2»

- Зберегти запит, вказавши ім'я «Клієнт 2».
- Переглянути результат запити (кнопка Пуск).

1. **Створити запит «Клієнт 3»**, в якому будуть відомості про клієнтів, що знаходяться за межами України, при цьому поле *Країна* виводити на екран.

Для цього виконати наступні дії:

- Скопіювати запит «Клієнт 2», вказавши ім'я «Клієнт 3».
- Відкрити запит «Клієнт 3» в режимі конструктора.
- В бланку запити для поля *Країна* в рядку **Критерії** ввести $\langle \rangle$ *Україна*, що

означає «Не Україна» (рис. 3.7).

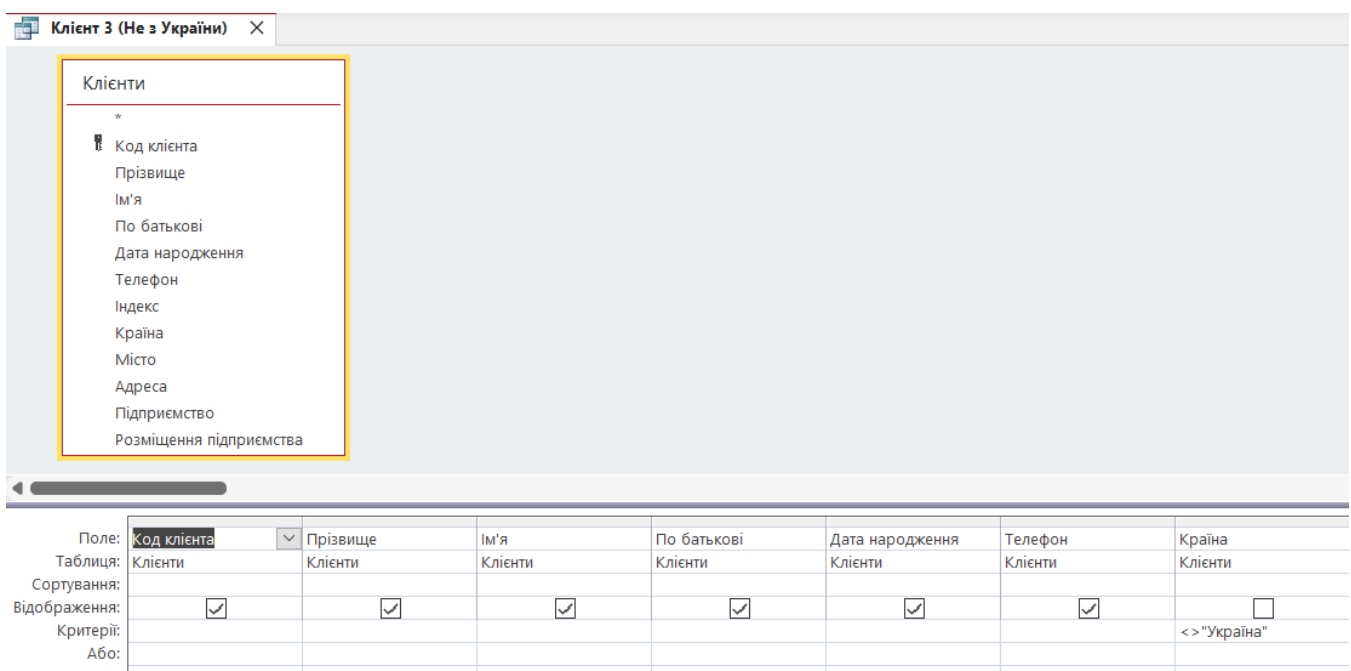


Рисунок 3.7 – Бланк запити «Клієнт 3»


- Зберегти запит.
- Переглянути результат запиту (кнопка Пуск).

2. **Створити запит «Клієнт 4»** так, щоб з його допомогою можна було знаходити клієнтів, що народилися в останній декаді березня (тобто народилися між 21-им і 31-им числами березня включно), з обчисленням віку клієнтів в окремому полі.

Для цього виконати наступні дії:

- На стрічці виконати **Створення \ Запити \ Макет запитів**.
- Додати таблицю **Клієнти**.
- Додати в бланк запиту поля: *Код клієнта, Прізвище, Ім'я, По батькові, Дата народження*.

Дата народження.

- Збережіть запит, привласнивши ім'я «Клієнт 4».
- Для поля *Дата народження* у рядку **Критерії** викликати «Побудовник виразів» (встановити курсор в рядку **Критерії** і натиснути кнопку «Конструктор»  на панелі інструментів).

- У лівому полі області «Елементів виразів» Побудовника виразів вибрати «Оператори»; в середньому полі області «Категорії виразів» вибрати тип оператора «Порівняння»; в правому полі області «Значення виразів» вибрати оператор **Between**; двічі натиснути праву кнопку миші (рис. 3.8).

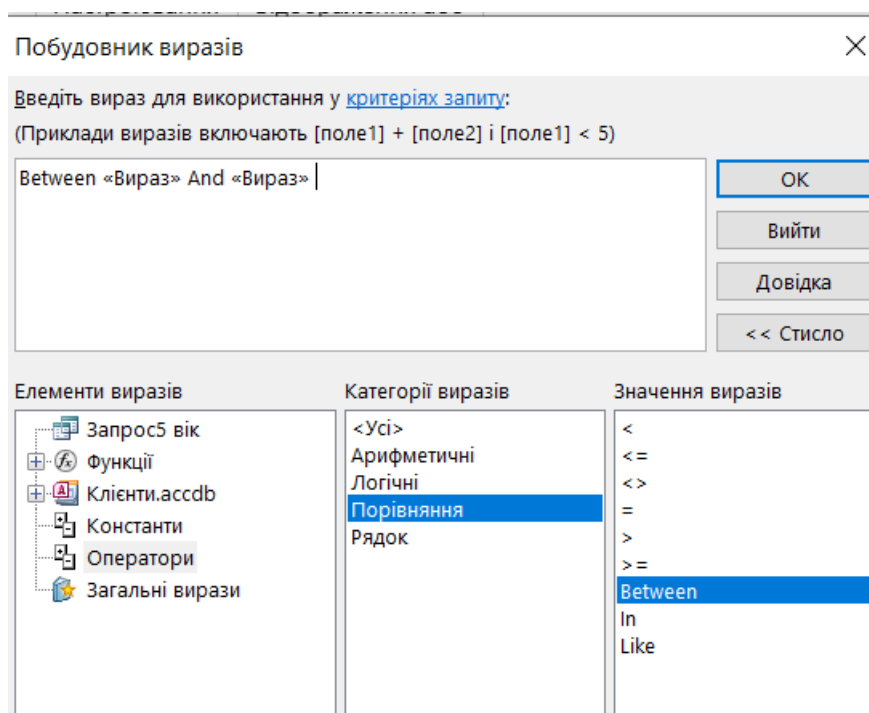


Рисунок 3.8 – Використання оператора порівняння для запиту «Клієнт 4»

Замість першого аргумента оператора **Between** необхідно викликати функцію *DateSerial*; для цього:

4. подвійним клацанням виділити в операторі **Between** перше слово «Вираз»;
5. в лівому полі області «Елементів виразів» будівника виразів вибрати Функції \ Вбудовані функції;
6. в середньому полі області «Елементів виразів» вибрати тип функцій *Дата й час*;
7. в правому полі області «Значення виразів» вибрати функцію *DateSerial()*; двічі натиснути праву кнопку миші (рис. 3.9).
8. замість першого аргументу функції *DateSerial()* викликати функцію *Year()*: *Функції / Вбудовані функції / Дата й час / Year()*; в якості аргументу функції *Year()* вказати поле *Дата народження* із таблиці **Клієнти**;
9. замість другого аргументу функції *DateSerial()* вказати 3 (березень);
10. замість третього аргументу функції *DateSerial()* вказати 21 (21 число);
11. Скопіювати вираз *DateSerial(Year([Дата Народження]);3;20)*, яке міститься в першому аргументі оператора **Between**. Вставити скопійований вираз на місце другого аргумента оператора **Between**. Змінити в другому аргументі значення 21 на 31 (рис. 3.9).

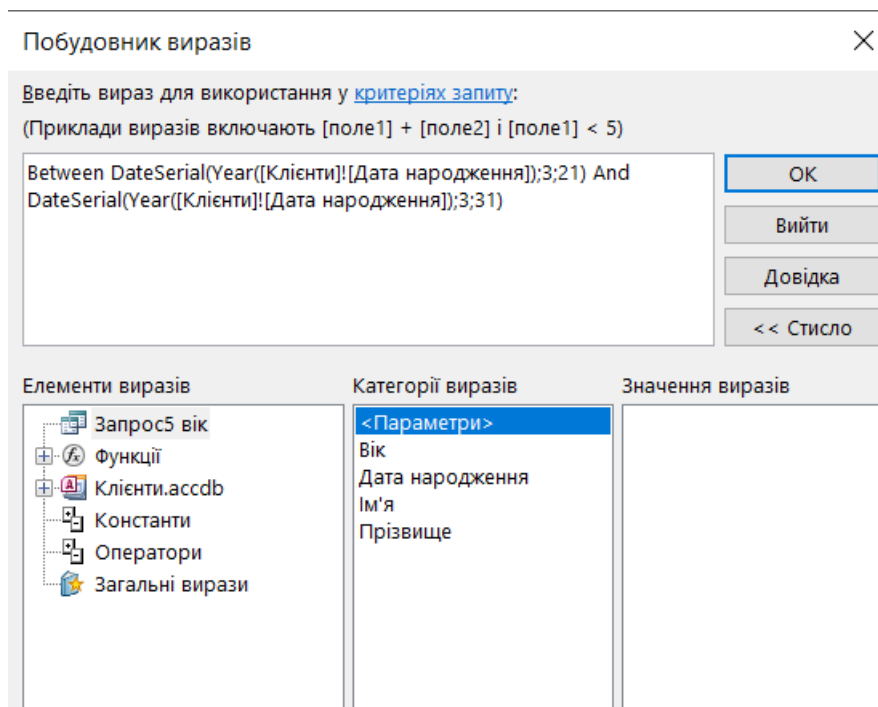


Рисунок 3.9 – Критерій для поля *Дата народження* для запиту «Клієнт 4»

12. Двічі натиснути праву кнопку миші. Вид запиту «Клієнт 4» в режимі конструктора після введення критерію для поля *Дата народження* (рис. 3.10).

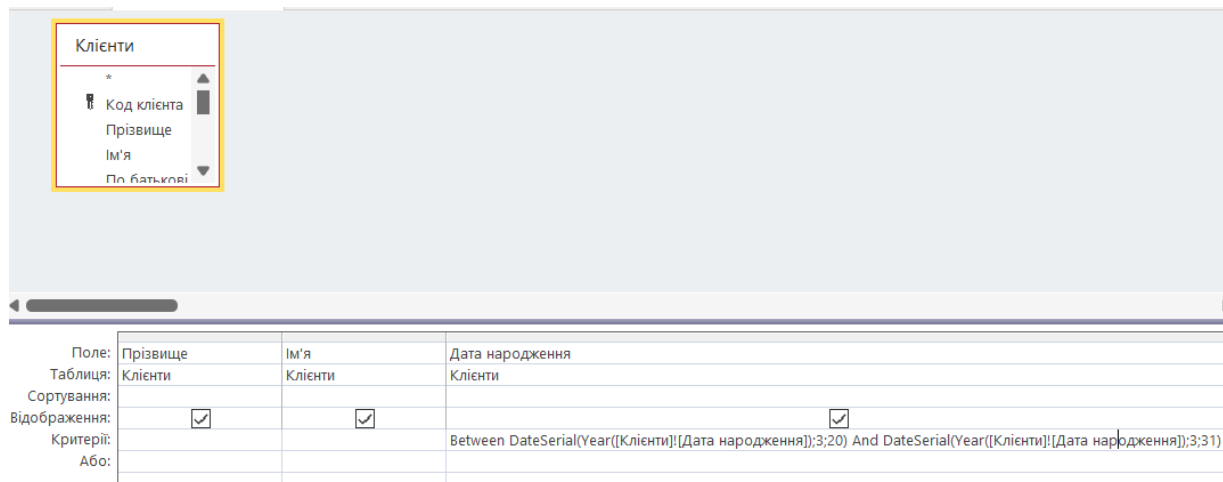


Рисунок 3.10 – Бланк запити «Клієнт 4»

13.3 метою підвищення точності визначення віку клієнтів, при створенні обчислюваного поля *Вік* доцільно використовувати функцію:

$\text{If}(\langle \text{вираз} \rangle; \langle \text{якщо_істина} \rangle; \langle \text{якщо_хибність} \rangle)$

14. Ця функція дозволить виконати порівняння поточного місяця з місяцем народження клієнта і число поточного місяця з числом народженням клієнта (рис. 3.11).

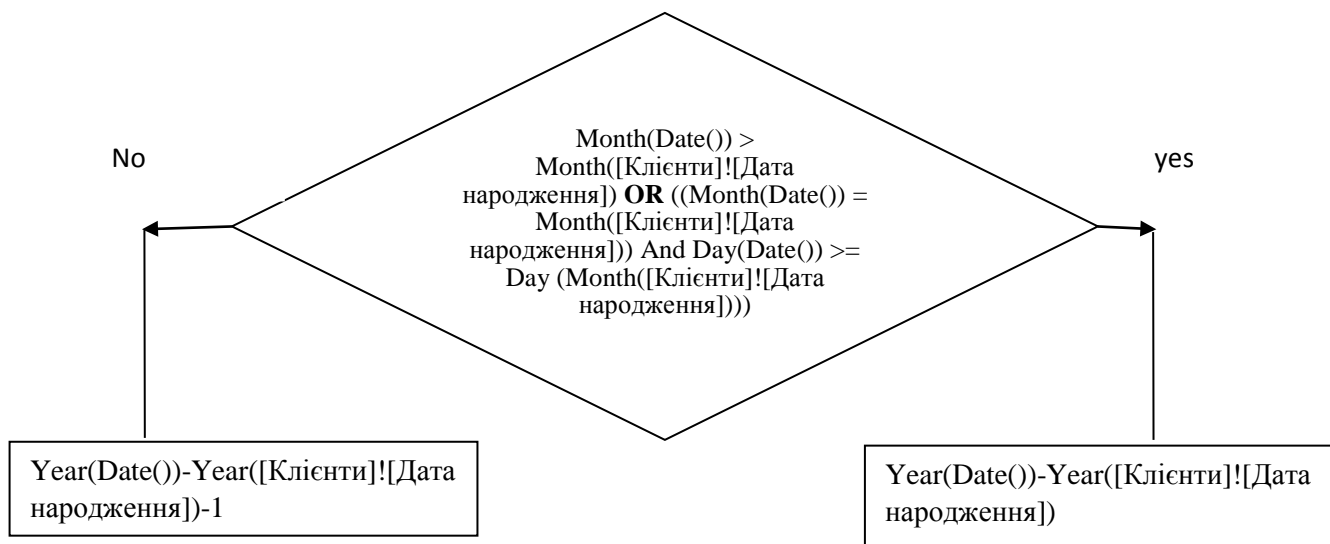


Рисунок 3.11 – Фрагмент алгоритму обчислення віку клієнта

Для створення обчислювального поля «Вік» встановити курсор в порожнє поле; ввести ім'я даного поля **Вік**: (двокрапка відокремлює назву поля від обчислень в даному полі) і викликати *Побудовник виразів*. Ввести вираз за наведеним вище фрагментом алгоритму і натиснути «Ок» (рис. 3.12-3.13).

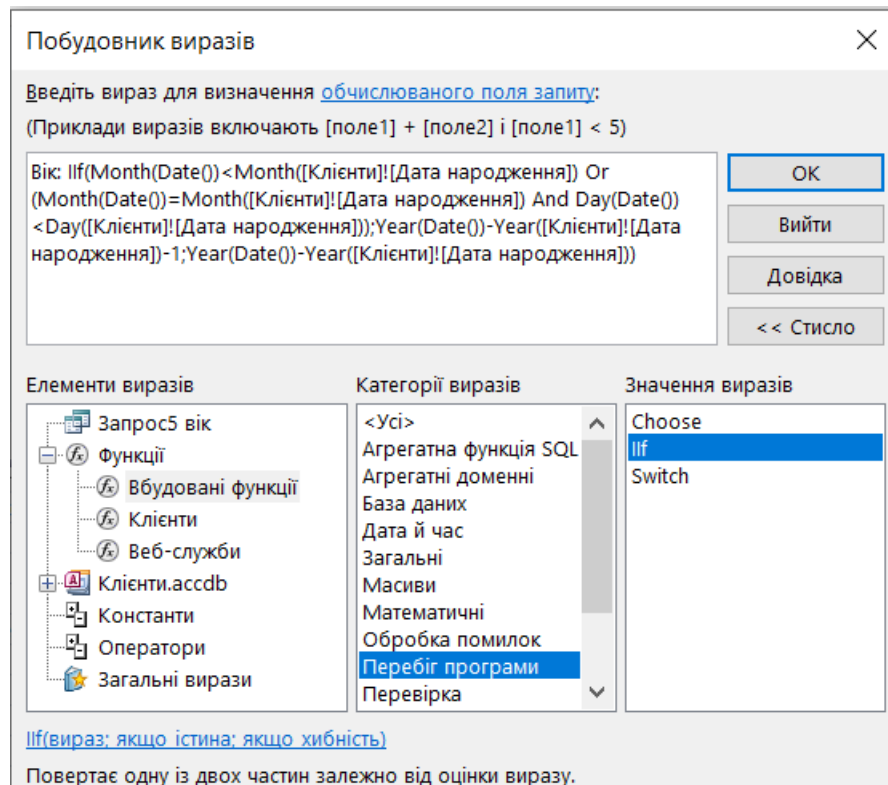


Рисунок 3.12 – Вигляд виразу для обчислення віку клієнта в *Побудовнику виразів*



Рисунок 3.13 – Вигляд виразу для обчислення віку клієнта в бланку запитів

3. Створити запит «Клієнт 5» для пошуку клієнтів, прізвища яких починаються з літер в діапазоні від А до К, що проживають в Києві або Дніпрі і народжені у квітні. Розв'язати цю задачу двома способами: без використання оператора OR і з використанням цього оператора.

Вид запиту «Клієнт 5 спосіб 1» в режимі конструктора з використанням оператора OR (рис. 3.14):

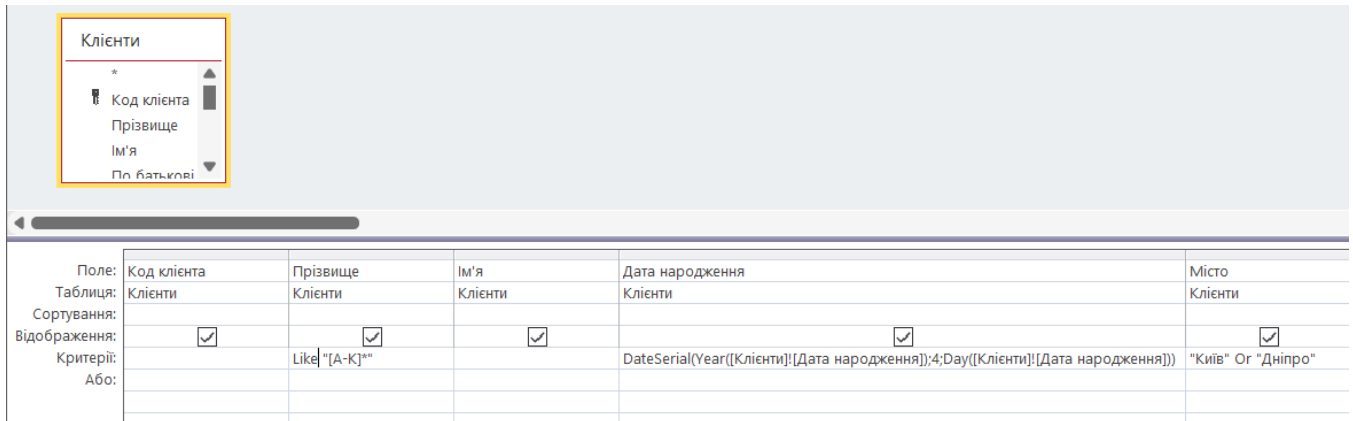


Рисунок 3.14 – Бланк запиту *Клієнт 5 спосіб 1*

Самостійно створити запит «Клієнт 5 спосіб 2» в режимі конструктора без використання оператора OR.

4. Зберегти запити в базі даних.

КОНТРОЛЬНІ ПИТАННЯ ДО ЗАХИСТУ ЛАБОРАТОРНОЇ РОБОТИ 3

1. Визначити особливості запитів на вибірку, перерахувати варіанти створення запитів за зразком.

2. Відкрити вікно Конструктора запитів і охарактеризувати його елементи.

3. Охарактеризувати призначення рядків бланка запиту у вікні Макета запиту.

4. Як виконуються операції додавання, видалення і зміни порядку полів у бланку запиту, продемонструвати виконання цих операцій.

5. Як забезпечується можливість сортування результатів вибірки за запитом, показати на прикладі одного зі створених запитів.

6. Навести приклад запиту з критерієм по діапазону значень.

7. Навести приклад запиту, в якому можна використовувати оператори OR і AND.

8. Охарактеризувати особливості створення в запитах обчислювальних полів, навести приклад такого запиту.

ЛАБОРАТОРНА РОБОТА №4

ТЕМА РОБОТИ: Конструювання форм.

МЕТА РОБОТИ: Отримання практичних навичок зі створення форм.

ТРИВАЛІСТЬ: 2 години

МЕТОДИЧНІ ВКАЗІВКИ

Форми – це більш зручний спосіб відображення інформації.

Створювати форми можна за допомогою спеціальних вбудованих майстрів, а також самостійно [7].

Налаштування форми


Для налаштування основних властивостей форми слід відкрити потрібну форму в режимі конструктора, виконати команду **Конструктор форм/Аркуш властивостей**:

– у вкладці *Формат* визначити, чи будуть виводитися Кнопки віконного меню (взагалі), які кнопки будуть активними (Кнопки *Розмірів вікна*) і чи буде на екрані Кнопка Закриття. У рядку *Надпис* даної вкладки можна визначити надпис у заголовку вікна Форми.

– у вкладці *Дані* в рядку *Джерело записів* можна змінити таблицю або запит, на базі якого створюється Форма. Вміст вікна **Список полів** автоматично зміниться. Крім того, в даній вкладці можна визначити, чи буде форма використовуватися тільки для перегляду даних, або в ній можна виконати редагування, видалення і додавання записів. Для цього використовуються властивості: *Дозволити редагування; Дозволити додавання; Дозволити видалення*.

Розміщення полів введення в формі

Розмістити (або додати) поля введення у формі можна двома способами.

1-й спосіб. Увійти в конструктор форми, вибрати на панелі **Елементи управління** кнопку *Текстове поле*  і клацнути мишею в тому ж місці області даних, де має з'явитися поле (рис.4.1).

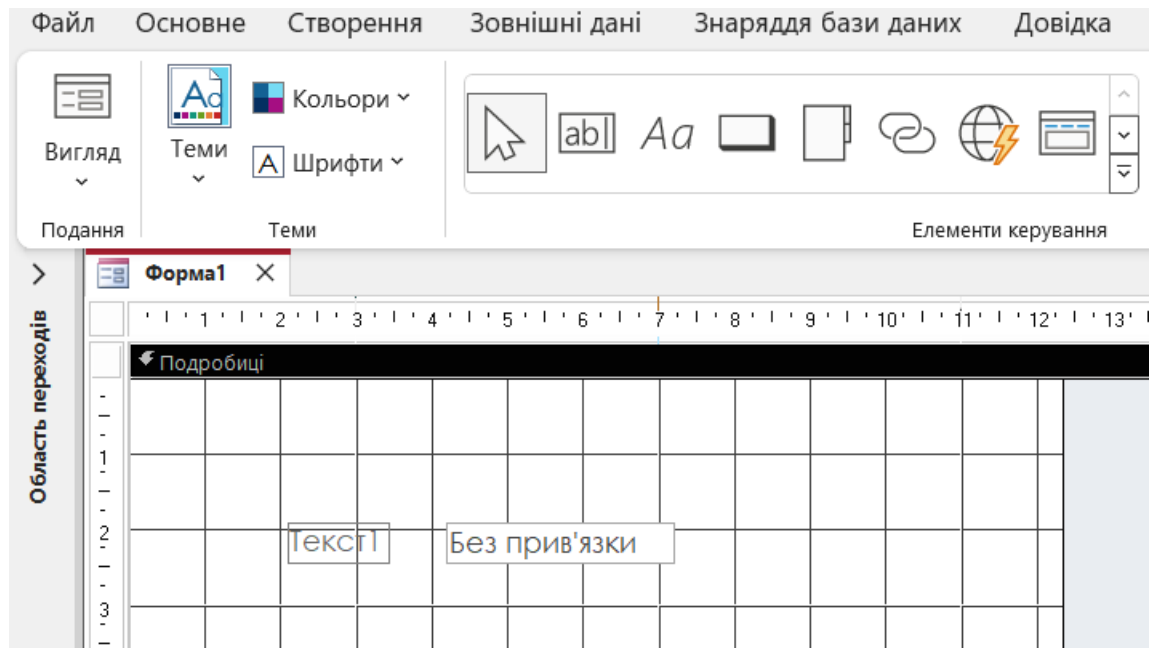



Рисунок 4.1 – Розміщення полів у формі

Виділивши **Текст1** (надпис поля) або **Без прив'язки** (значення поля), можна переміщувати ці елементи одночасно або окремо один від одного.

Для встановлення зв'язку цього поля з відповідним полем таблиці слід виділити **Без прив'язки** (рис.4.2) і, клацнувши правою кнопкою миші, в контекстному меню обрати команду **Властивості**. У вікні **властивостей** у вкладці **Дані** в рядку **Дані** потрібно вказати те поле таблиці, дані якого повинні виводитися на екран. У разі необхідності використання арифметичних виразів, у рядку **Дані** потрібно викликати **Побудовник виразів** (натиснувши на кнопку ).

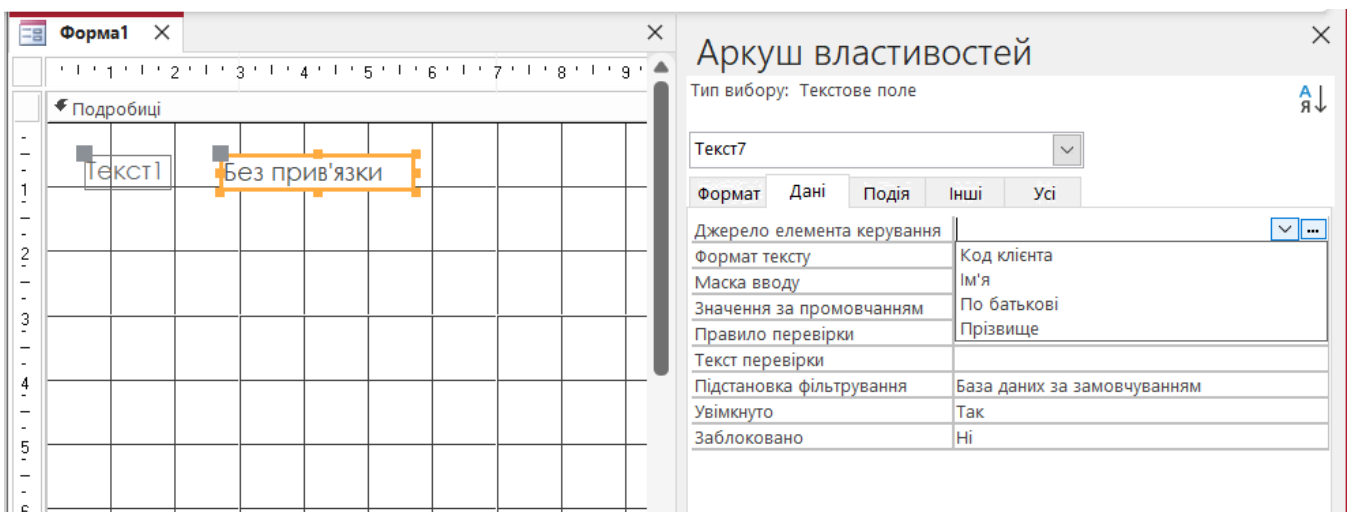


Рисунок 4.2 – Встановлення зв'язку поля з відповідним полем таблиці

Головною умовою того, щоб дані відображалися в полі введення, являється наступне: у формулах і математичних виразах **не повинні** бути присутніми імена таблиць, оскільки поля обираються з **форми**. Наприклад, обчислювальне поле **Залишок** у формі буде задано у вигляді виразу:

= [Замовлено] – [Продано]


У вкладці **Інші** можна встановити коротке пояснення до поля в рядку *Текст підказки елемента керування*.

Виділивши **Надпис** поля введення в режимі *Конструктора форми* і виконавши команду контекстного меню **Властивості**, можна на вкладці **Формат** задати *Підпис* поля (наприклад, замість *Текст1* вказати **Залишок**).

2-й спосіб. У вікні конструктора **Додавання наявних полів** відкрити потрібну таблицю, виділити ім'я поля і перетягнути його в *область даних Форми*. У цьому випадку **Поле вводу** буде вже пов'язане з відповідним полем таблиці, а **Надпис** до нього буде відповідати імені поля (**Увага! Даний спосіб не можна застосувати для обчислювальних полів**).

Створення кнопок управління

Кнопки, що використовуються у формах для виконання певної дії або ряду дій (наприклад, відкриття іншої форми, відкриття запиту), називаються *кнопками управління*. Для того, щоб кнопка виконувала будь-яку дію, необхідно створити макрос або процедуру обробки події і пов'язати їх з властивістю кнопки *Натиснення кнопки*. Наприклад, для створення кнопки управління, що викликає на виконання запит, слід виконати наступні дії.

1. На панелі швидкого доступу вибрати **Конструктор/Елементи управління/ Кнопка**  і намалювати кнопку в *Області даних форми*.

2. Закрити діалогове вікно майстра створення кнопки. Виділити кнопку, виконавши клацання правою кнопкою миші і в контекстному меню обрати команду **Властивості**.

3. У вікні **Кнопка** перейти на вкладку **Подія**, встановити курсор в рядку *Після клацання* і викликати *Побудовник* .

4. У діалоговому вікні **Побудовник** вибрати **Макроси**.

5. У стовпці **Макрокоманда** обрати макрокоманду **Відкрити запит**.

6. Перейти до **аргументів макрокоманди** (в нижню частину вікна). У рядку **Ім'я запиту** вказати той запит, який буде виконуватися при натисненні кнопки управління.

7. Закрити вікно, зберегти макрос, перейменувати кнопку управління і перейти з *режиму конструктора* в *режим форми*.

8. Спробувати дію кнопки управління в режимі форми.

Для створення кнопки управління, дія якої полягатиме у **Відкритті форми** слід виконати ті самі дії, тільки в *пункті 6* обрати макрокоманду **Відкрити форму**, а в *пункті 7* у рядку **Ім'я форми** вказати ім'я відкриваючої форми.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ 4

1. Запустити **Access**.
2. Відкрити базу даних **Замовлення**.
3. Створити форму за допомогою **Майстра форм** для таблиці «Клієнти» (На панелі інструментів виконати *Створення / Форми / Майстер форм*). На екрані з'явиться вікно першого кроку майстра створення форми (рис. 4.3).

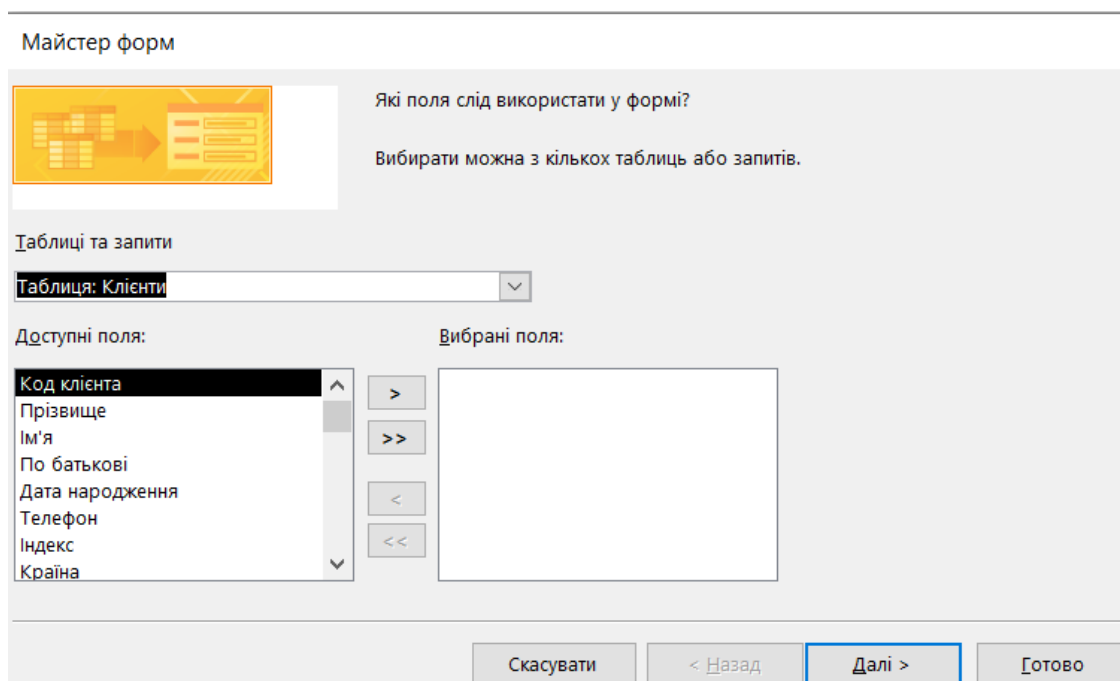



Рисунок 4.3 – Початкове вікно Майстра форм

Натиснути кнопку  для перенесення в форму всіх записів з таблиці «Клієнти» (рис. 4.4).

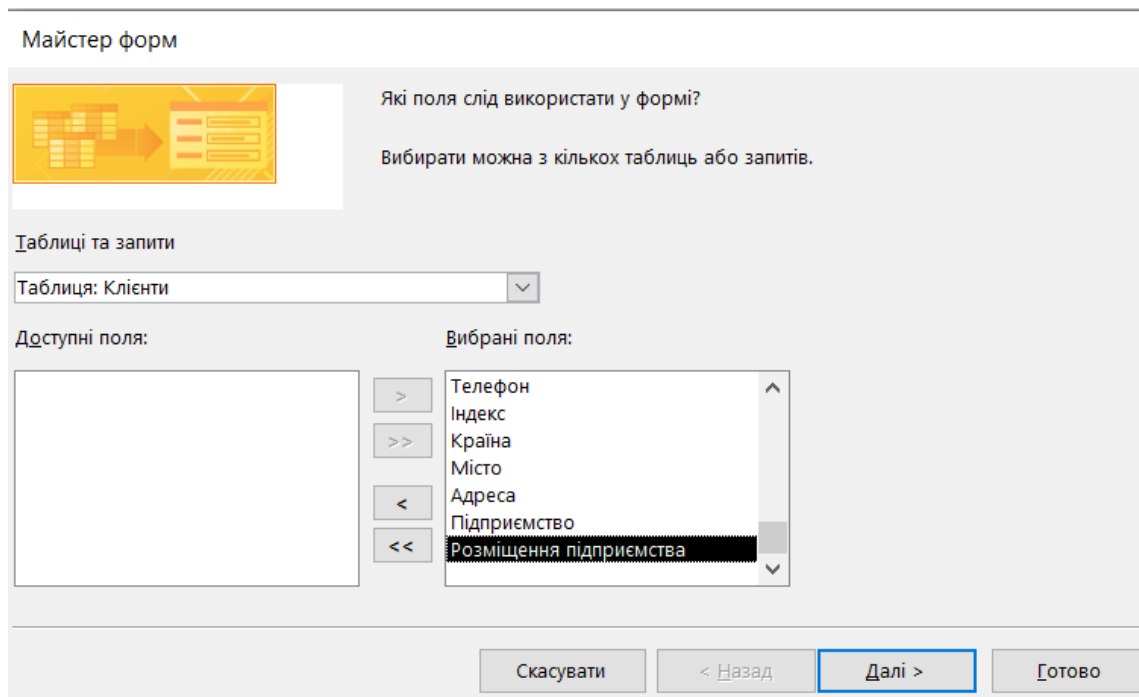


Рисунок 4.4 – Перенесення полів в Майстрі форм

Натиснути кнопку *Далі*, щоб перейти на другий крок майстра створення форм. Обрати зовнішній вигляд для майбутньої форми (наприклад, в один стовпець) і натиснути кнопку *Далі*, щоб перейти на наступний крок (рис. 4.5).

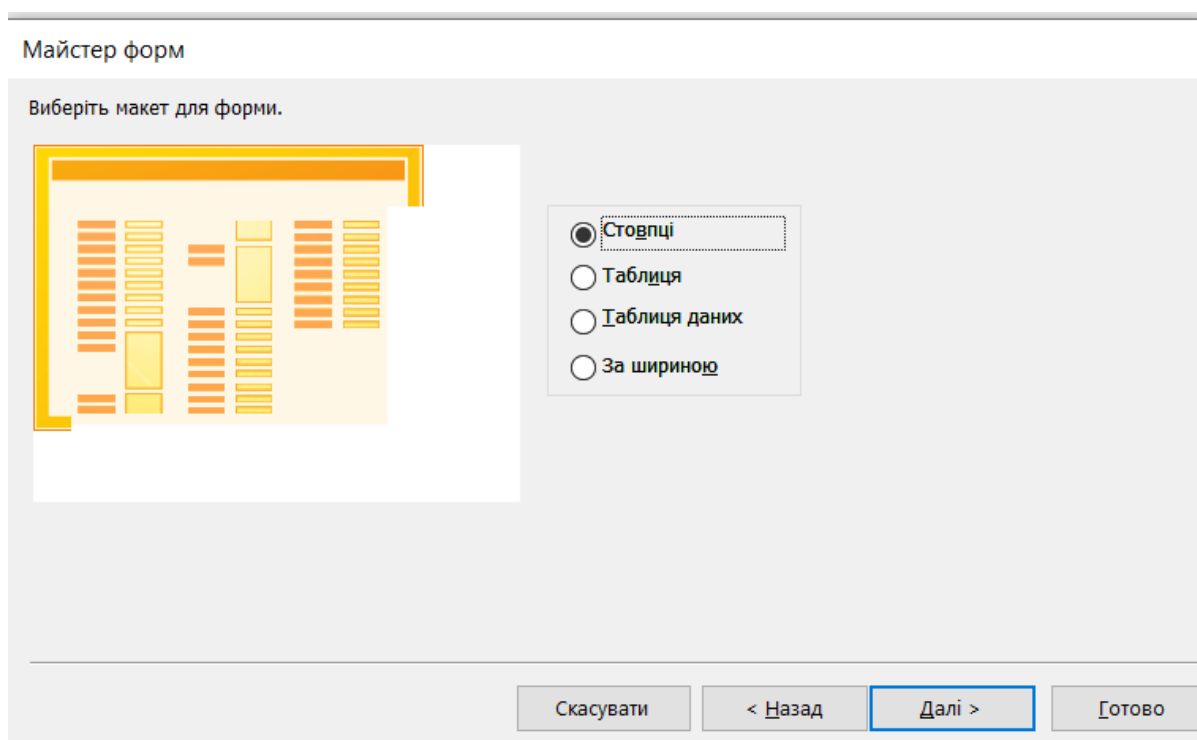


Рисунок 4.5 – Налаштування макету для форми в Майстрі форм

На останньому кроці майстра форм вказати назву форми і встановити перемикач для виконання певної наступної дії. Якщо додаткових змін в макеті форми не передбачається, то поставити перемикач на «Відкрити форму для перегляду або введення даних». Натиснути кнопку *Готово* (рис. 4.6).

Майстер форм

Виберіть назву для форми.

Клієнти1

Це всі дані, потрібні майстру для створення форми.

Відкрити форму чи змінити її макет?

Відкрити форму для перегляду або введення даних

Змінити макет форми

Скасувати < Назад Далі > Готово

Рисунок 4.6 – Зазначення назви форми в Майстрі форм

4. У режимі Конструктора створити форму для таблиці «Клієнти», для чого виконати наступні кроки.

– Спроекувати макет форми. Для цього на панелі інструментів виконати **Створення / група Форми / Конструктор форм**. На екрані з'явиться порожній макет форми. Для відображення списку доступних полів виконати: **Конструктор форм / Знаряддя / Додавання наявних полів**. У списку полів розкрити певну таблицю (рис. 4.7).

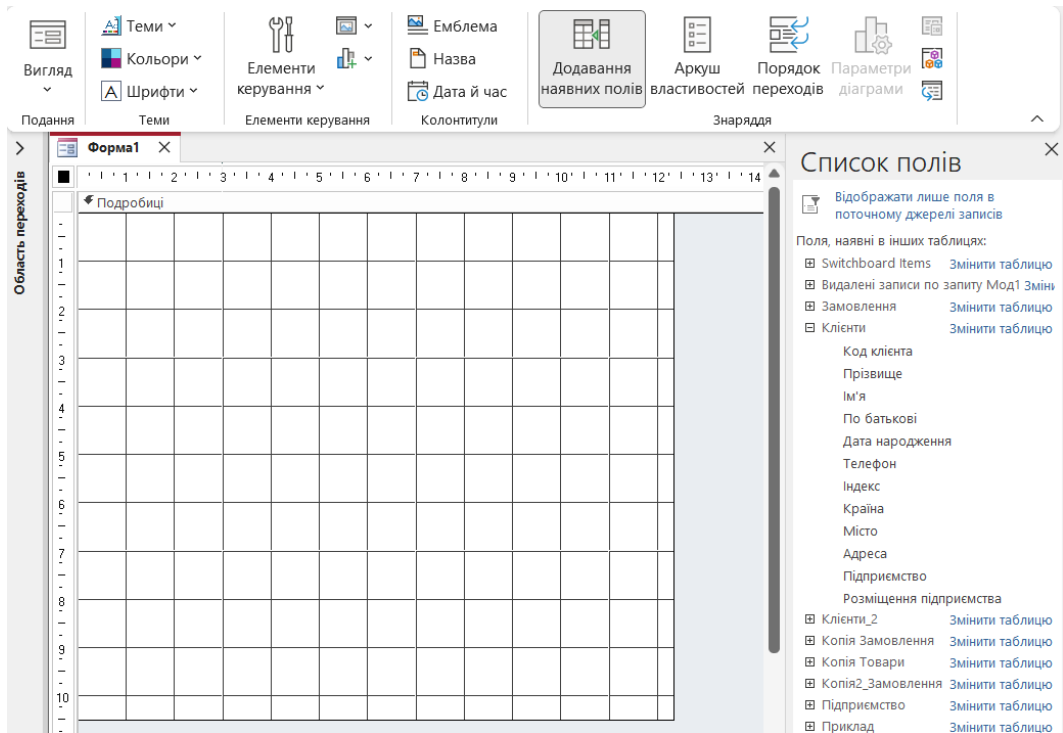


Рисунок 4.7 – Вибір об’єкта для створення форми

– Методом перетягування зі списку доступних полів в область даних форми перенести поля: *Код клієнта, Прізвище, Ім’я, По батькові, Індекс, Країна, Місто, Адреса, Телефон, Дата народження, Підприємство*. Розташувати так, як це показано на рисунку (рис. 4.8).



Рисунок 4.8 – Розміщення полів на формі



– Додати назви до груп за допомогою об'єкта *Підпис*  на панелі інструментів *Елементи управління* і поміст їх в рамки, окресливши їх прямокутниками (кнопка  на панелі інструментів *Елементи управління*) (рис. 4.9).



Рисунок 4.9 – Редагування форми

– У бланк форми додати область заголовка – команда *Назва* в Конструкторі форм.

– Вставити в області заголовка форми емблему для форми (*Колонтитули / Емблема*) (рис. 4.10);

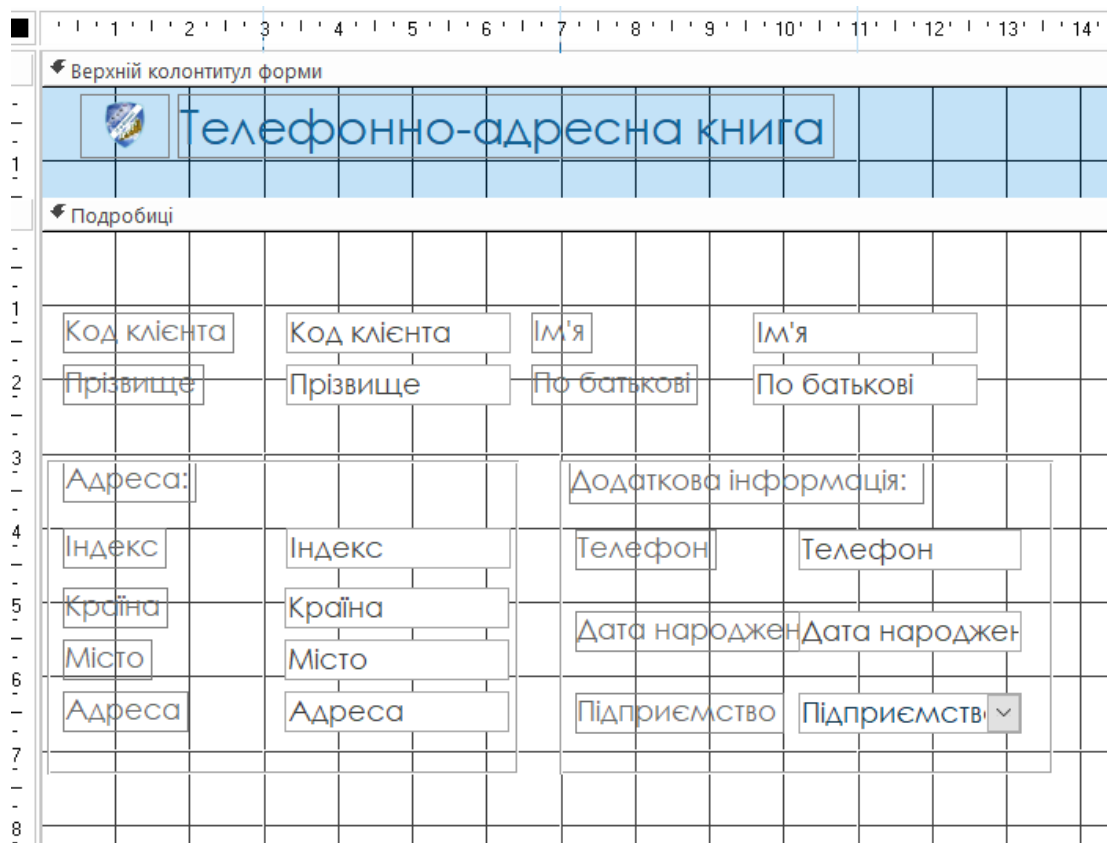



Рисунок 4.10 – Додавання області заголовка на форму

- Зберегти форму в базі даних, присвоївши їй ім'я «Телефонно-адресна книга»
- Створити **обчислювальне поле** для розрахунку віку клієнтів. Для цього на панелі *Елементів керування* натиснути кнопку Текстове Поле ; натиснути лівою кнопкою миші в тому місці форми, де хочете, щоб перебувало нове поле; на рисунку це поле *Текст20* (рис. 4.11).

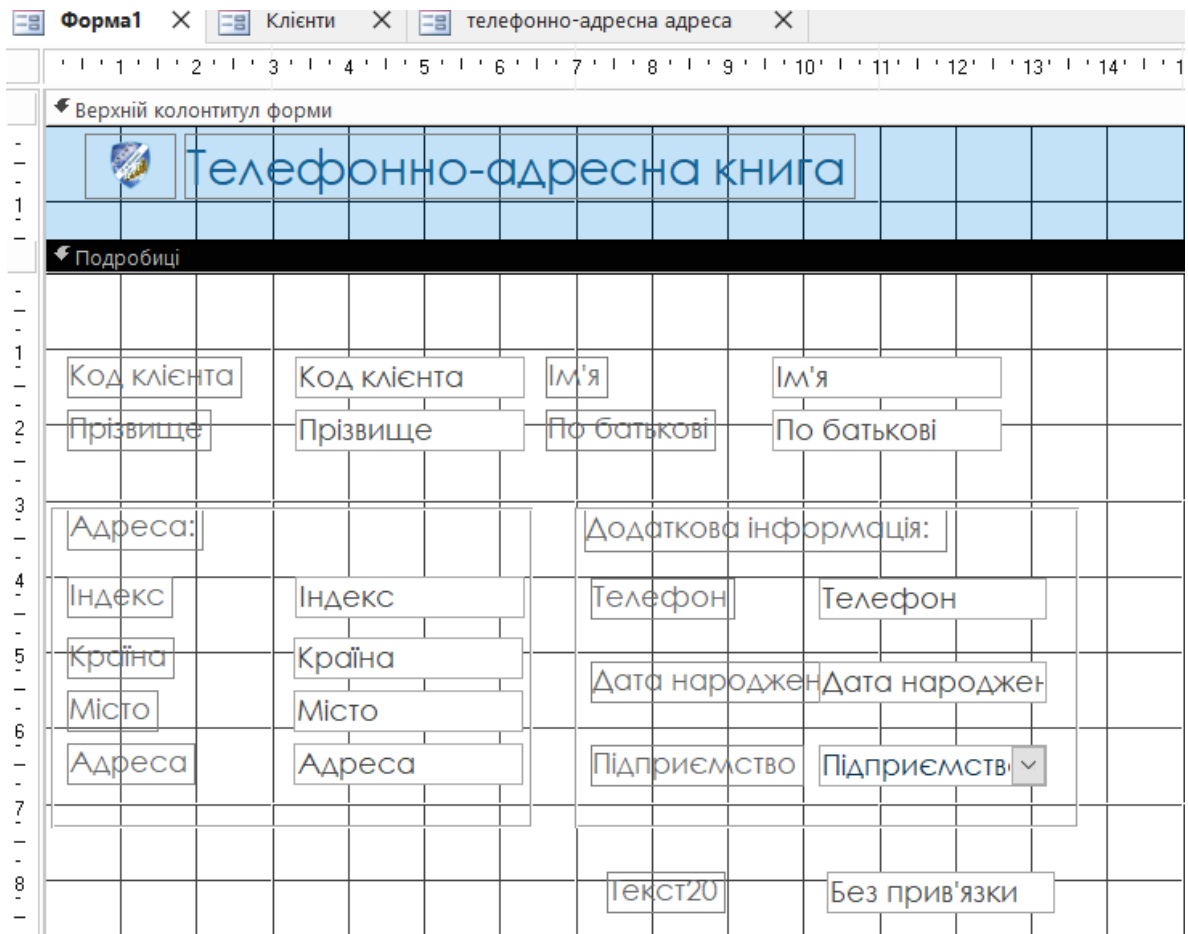


Рисунок 4.11 – Додавання елемента для обчислюваного поля

- Змінити надпис поля «Текст20» на «Вік».
- Зайти у властивості поля (в контекстному меню викликати *Властивості*); перейти на вкладку *Дані / Джерело елемента керування* і викликати *Побудовник виразів*, натиснувши на три крапки. Побудувати вираз, за допомогою якого буде проводитися розрахунок віку клієнтів (зверніть увагу на те, що у виразі для обчислень вибрати поля **з поточної форми**, а не з таблиці). Використовувати алгоритм розрахунку віку клієнтів з лабораторної роботи 3 рис. 3.11;

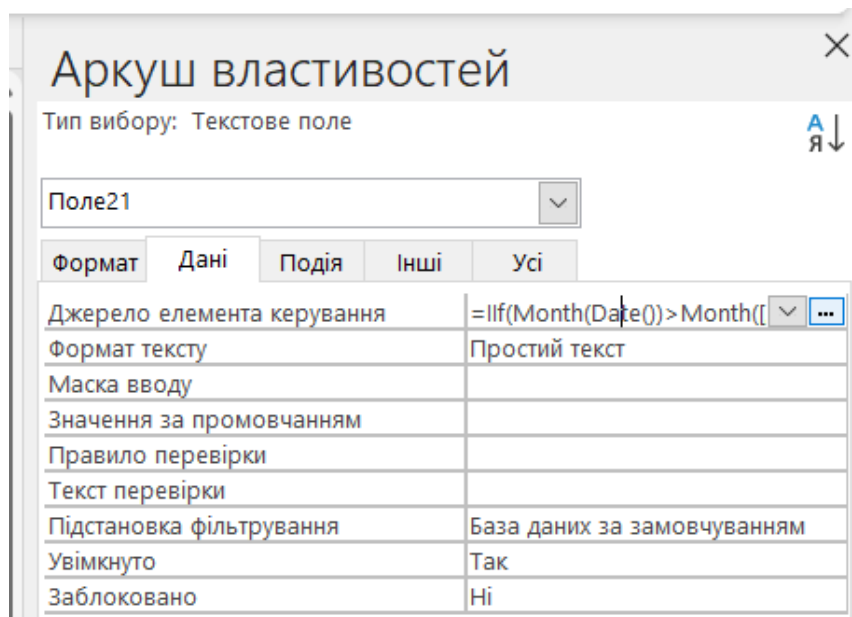


Рисунок 4.12 – Побудова виразу для обчислювального поля

– Створити **Кнопку управління** в примітці форми, клацання по якій дозволило б викликати на виконання запит "*Клієнт2*" (інформація про всіх клієнтів з України). Для цього на панелі *Елементів керування* натиснути на кнопку «Кнопка»; в нижньому колонтитулі форми намалюйте кнопку. На екрані відкриється діалогове вікно Майстра кнопок, в якому вибрати категорію *Інше* та дію *Виконати запит*, як це показано на малюнку 4.13:

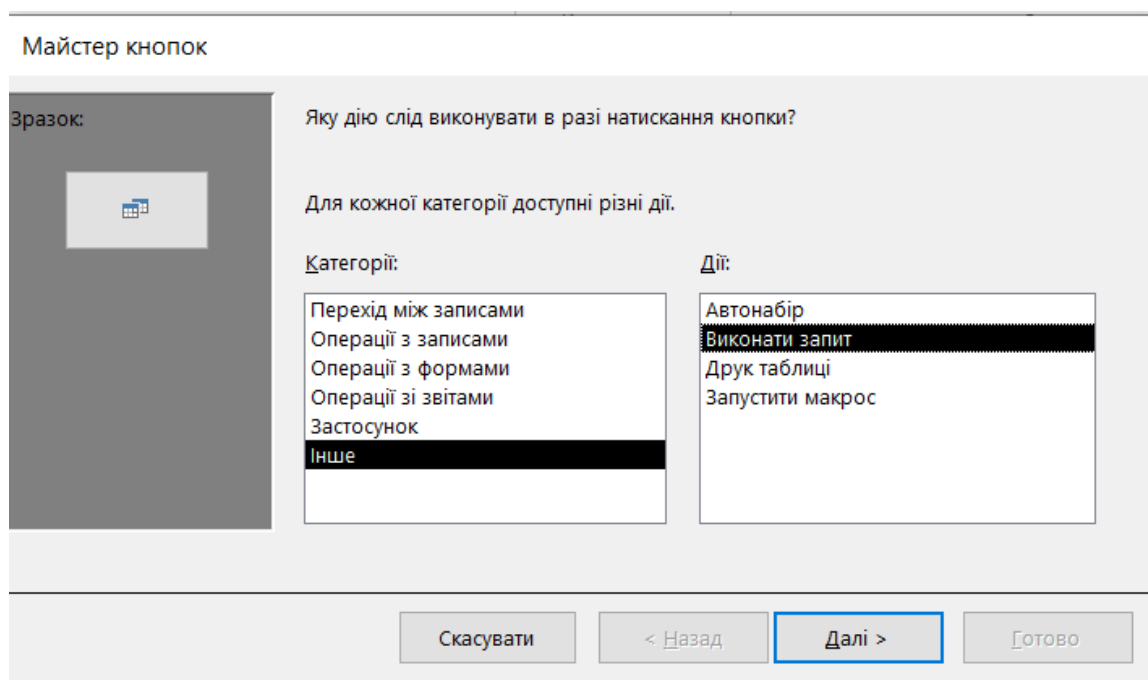


Рисунок 4.13 – Діалогове вікно Майстра кнопок

– Натиснути на кнопку *Далі*, вибрати запит «Клієнт2_Україна». Натиснути знову на кнопку *Далі* (рис. 4.14).

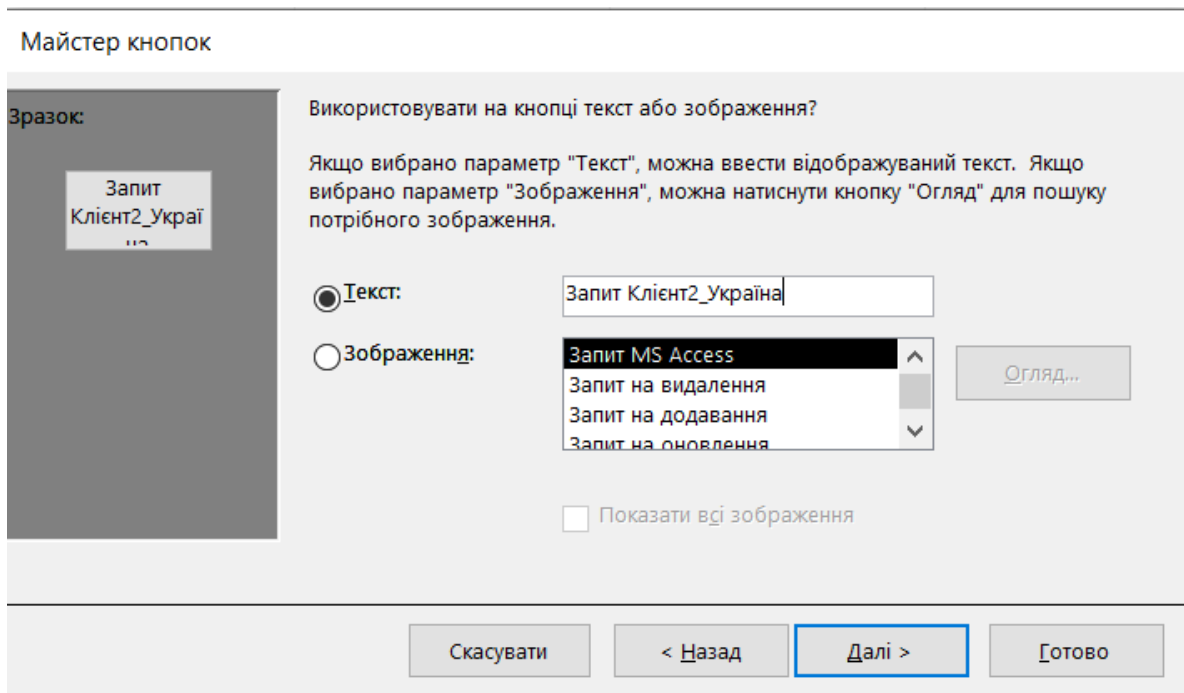


Рисунок 4.14 – Введення назви кнопки

- Ввести текст, який буде відображатися на кнопці, наприклад, *Клієнти з України*. Натиснути кнопку *Далі* (рис. 4.15).
- На останньому етапі можна задати ім'я для кнопки, яка створюється. Натиснути кнопку *Готово*.
- створеній кнопці керування задати властивість *Текст підказки елемента керування* (вкладка *Інші* вікна *Властивості* вибраного об'єкта).
- перевірити *Порядок переходів* об'єктів в режимі форми при натисканні клавіші Tab (за необхідності цей порядок можна змінити *Конструктор форм / Знаряддя / Порядок переходів*);
- виконати редагування форми відповідно до макету, а саме: змінити розміри об'єктів; перемістити їх у потрібне місце форми; виконати вирівнювання об'єктів; змінити установки кольору об'єктів форми; використовувати лінії і прямокутники для об'єднання в групи схожих за змістом об'єктів і для поліпшення зовнішнього вигляду форми;
- ввести в таблицю «**Клієнти**» за допомогою створеної форми кілька нових записів;
- зберегти форму в базі даних (рис. 4.15).

Форма1 X Клієнти X телефонно-адресна адреса X

Телефонно-адресна книга

Код клієнта: Ім'я:
 Прізвище: По батькові:

Адреса: Індекс:
 Країна:
 Місто:
 Адреса:

Додаткові відомості: Телефон:
 Дата народження:
 Підприємство:

Вік:

Рисунок 4.15 – Зовнішній вигляд форми «Телефонно-адресна книга»

5. Створити форми за допомогою команд: *Створення / Форми / Додаткові форми / Кілька елементів (Таблиця, Розділена форма)* (рис. 4.16-4.18).

Замовлення1 X

Замовлення

Код клієнта	Код товару	Дата замовлення	Замовлено	Дата продажу	Продано
ABC00001	1	13.02.2023	80	14.02.2023	70
ABC00001	1	01.03.2024	11	01.03.2024	11
ABC00001	1	06.06.2024	10	12.06.2024	7
ABC00001	2	05.05.2022	15	15.09.2022	15
ABC00003	1	25.02.2024	23	01.03.2024	20
ABC00003	2	02.09.2023	12	07.09.2023	12
ABC00004	1	01.02.2023	10	10.02.2023	10

Рисунок 4.16 – Створення форми на основі шаблону «Кілька елементів»

Підприємство1

Найменування підприємства: EPAM University

Керівник: Іваненко А.В.

Адреса: Дніпро, Гагаріна, 7

Телефон: (066)-738-47-83

Найменування підприємства	Керівник	Адреса	Телефон
EPAM University	Іваненко А.В.	Дніпро, Гагаріна, 7	(066)-738-47-83
Luxoft	Петренко П.М.	Київ, Шишкевича, 125	(097)-565-46-58
Softserve	Пилипенко П.Г	Дніпро, Поля,34	(097)-657-64-58
ДМетІ	Величко О.Г.	Дніпро, Науки, 4	(050)-123-45-64
УДУНТ	Сухий К.М.	Дніпро, Лазаряна, 4	(050)-676-48-37
*			

Рисунок 4.17 – Створення форми на основі шаблону «Розділена форма»

Код клієнта	Код товару	Дата замовл.	Замовлено	Дата продаж	Продано
ABC00001	1	13.02.2023	80	14.02.2023	70
ABC00001	1	01.03.2024	11	01.03.2024	11
ABC00001	1	06.06.2024	10	12.06.2024	7
ABC00001	2	05.05.2022	15	15.09.2022	15
ABC00003	1	25.02.2024	23	01.03.2024	20
ABC00003	2	02.09.2023	12	07.09.2023	12
ABC00004	1	01.02.2023	10	10.02.2023	10
ABC00005	1	01.03.2023	5	02.03.2023	5
ABC00006	2	01.01.2017	0		0
ABC00008	2	01.04.2024	7	04.04.2024	5
*			0		0

Рисунок 4.18 – Створення форми на основі шаблону «Таблиця»

КОНТРОЛЬНІ ПИТАННЯ ДО ЗАХИСТУ ЛАБОРАТОРНОЇ РОБОТИ 4

1. Перерахувати способи створення форм.
2. Пояснити метод додавання полів на форму.

3. Перелічити елементи керування при роботі з формою.
4. Продемонструвати, як здійснюється виділення, зміна розмірів, переміщення й видалення об'єктів у формі? Виконати ці операції на прикладі своєї форми.
5. Пояснити, для чого і як виконується зміна *Порядку обходів* об'єктів форми.
6. Продемонструвати створення кнопки на формі.
7. Пояснити, як створюються обчислювальні поля на формі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Лосєв М. Ю., Федько В. В. Бази даних : навч.-практ. посіб. для самост. роботи студентів. Електрон. ресурс. Харків : ХНЕУ ім. С. Кузнеця, 2018. 233 с.
2. Ситник Н. В. Організація баз та сховищ даних : практикум. Електрон. ресурс. Київ, 2017. 148 с.
3. Керування форматами введення даних за допомогою масок вводу. *Microsoft Support*. URL : <http://surl.li/jjkbdl> (дата звернення: 12.08.2024).
4. Обмеження на введення даних за допомогою правил перевірки. *Microsoft Support*. URL : <http://surl.li/zvudmw> (дата звернення: 12.08.2024).
5. Використання символів узагальнення в запитах і параметрах Access перевірки. *Microsoft Support*. URL : <http://surl.li/qvbael> (дата звернення: 12.08.2024).
6. Функція DateAdd. *Microsoft Support*. URL : <http://surl.li/zdjbok> (дата звернення: 21.08.2024).
7. Основи організації баз даних : навч. посіб. до вивчення дисципліни «Організація баз даних» / уклад.: Б. Л. Голуб, Д. Ю. Ящук. Київ, 2017. 139 с.

Навчально-методичне видання

Бандоріна Лілія Микодаївна,
Савчук Лариса Миколаївна,
Удачина Катерина Олександрівна,
Савчук Роман Вячеславович

**ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ТА
АДМІНІСТРУВАННЯ БАЗ ДАНИХ**

Навчально-методичні рекомендації
до виконання лабораторних робіт
Частина 1

В авторській редакції
Комп'ютерна верстка К. О. Удачина

Експертний висновок склав канд. екон. наук, проф. Л. М. Савчук

Зареєстровано НМВ УДУНТ (№ 805 від 02.12.2024)

Формат 60x84 1/16. Ум. друк. арк. 3,31. Обл.-вид. арк. 1,42.
Зам. № 127

Видавець: Український державний університет науки і технологій
вул. Лазаряна, 2, ауд. 2216, м. Дніпро, 49010.
Свідоцтво суб'єкта видавничої справи ДК №7709 від 14.12.2022

Адреса видавця та дільниці оперативної поліграфії:
вул. Лазаряна, 2, Дніпро, 49010