ІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Уманський державний педагогічний університет імені Павла Тичини

MS ACCESS 2019: ТЕОРІЯ ТА ПРАКТИКА

Навчально-методичний посібник

Укладачі: М. О. Медведєва, І. І. Криворучко, М. С. Ковтанюк

Умань
2021

УДК 004.65(075.8)
М14

Рецензенти:

Ткачук Г.В., доктор педагогічних наук, доцент, професор кафедри інформатики і інформаційно-комунікаційних технологій Уманського державного педагогічного університету імені Павла Тичини;

Дякон В.М., кандидат фізико-математичних наук, доцент, директор Уманської філії Європейського університету.

Рекомендовано до друку вченою радою
факультету фізики, математики та інформатики Уманського
державного педагогічного університету імені Павла Тичини
(протокол № 5 від 25 листопада 2021 року)

М14 MS ACCESS 2019: теорія та практика : навч.-метод. посіб. / МОН України, Уманський держ. пед. ун-т імені Павла Тичини ; уклад.: М. О. Медведєва, І. І Криворучко., М. С. Ковтанюк. – Умань : Візаві, 2021. – 124 с.

Навчально-методичний посібник «MS Access 2019: теорія та практика» висвітлює всі основні теми згідно навчальної програми, а також створений на основі досвіду викладачів факультету фізики, математики та інформатики. Особлива увага надається формуванню в майбутніх вчителів закладів загальної середньої освіти знань і умінь щодо роботи з офісним програмним забезпеченням, оволодінню основними навичками роботи з ПК та підготовці до ІІ етапу Всеукраїнської студентської олімпіади з інформатики.

УДК 004.65(075.8)

© М. О., Медведєва, І. І., Криворучко, М. С., Ковтанюк, уклад., 2021

ЗМІСТ

ПЕРЕДМОВА		5
РОЗДІЛ І.	ТЕОРЕТИЧНИЙ БЛОК	6
Лекція 1.	Створення таблиць бази даних в Access	6
1.1	Основні об’єкти СУБД Access	6
1.2	Порядок створення БД у MS Access	8
1.3	Типи даних полів	10
1.4	Способи створення таблиць	14
Лекція 2.	Імпортування даних та встановлення зв’язків між таблицями	23
2.1	Використання зовнішніх даних – імпортування таблиць	23
2.2	Створення у таблицях полів підстановки	26
2.3	Створення зв’язків між таблицями	31
Лекція 3.	Сортування та фільтрування даних	35
3.1	Сортування даних	35
3.2	Фільтрування даних	38
3.3	Правила формування критеріїв відбору	40
Лекція 4.	Створення простих запитів	43
4.1	Способи формування запитів	43
4.2	Створення простого запиту у режимі Макет запиту	44
4.3	Режим SQL	47
4.4	Створення простого запиту за допомогою Майстра запитів	48
4.5	Створення обчислюваних полів	51
Лекція 5.	Типи запитів Access	56
5.1	Типи створення запитів в Access	56
5.2	Створення перехресного запиту	56
5.3	Створення запиту на відбір з параметром	58
5.4	Запит на оновлення записів	59
5.5	Запит на видалення записів	60
5.6	Запит на створення таблиці	61
5.7	Запит на додавання даних	63
Лекція 6.	Створення форм	64
6.1	Способи створення форм	64
6.2	Використання інструмента Форма	66
6.3	Використання Майстра форм	66
6.4	Використання пустої форми	68
6.5	Розробка форми у режимі Конструктор	69

6.6	Створення підформ (підпорядкованих форм)	71
Лекція 7.	Створення звітів	76
7.1	Огляд звітів у програмі Access	76
7.2	Частини звіту	76
7.3	Способи створення звітів	78
7.4	Послідовність створення звіту у програмі Access	83
7.5	Виділення даних за допомогою умовного форматування	83
7.6	Надання звітові професійного зовнішнього вигляду за допомогою тем	85
7.7	Додавання емблеми або зображення тла	85
7.8	Попередній перегляд і друк звіту	86
Лекція 8.	Створення макросів	88
8.1	Засоби програмування в Access	88
8.2	Специфіка використання макросів та засоби їх створення	89
8.3	Створення командної кнопки з макросом	91
8.4	Створення самостійного (глобального) макросу	93
8.5	Створення складного макросу	96
Лекція 9.	Створення форм навігації бази даних	100
9.1	Створення форм навігації бази даних	100
9.2	Редагування форми навігації	101
9.3	Налаштування параметрів відображення форми навігації	102
РОЗДІЛ ІІ.	ЛАБОРАТОРНИЙ БЛОК	103
	Лабораторна робота 1	103
	Лабораторна робота 2	107
	Лабораторна робота 3	111
	Лабораторна робота 4	113
	Лабораторна робота 5	116
	Лабораторна робота 6	118
	Лабораторна робота 7	120
	Лабораторна робота 8	121
	Лабораторна робота 9	122
	Рекомендована література	123

ПЕРЕДМОВА

Третє тисячоліття в усіх сферах життєдіяльності людини і суспільства в цілому висуває нові, раніше невідомі виклики. Це ставить перед людиною, а отже, і перед освітою, нові завдання, зумовлені переходом людства до нового типу цивілізації.

Освіта XXI століття – це освіта для людини, відповідальної особистості, яка здатна до самоосвіти і саморозвитку, вміє критично мислити, опрацьовувати різноманітну інформацію, використовувати набуті знання і вміння для творчого розв'язання проблем, прагне змінити на краще своє життя і життя своєї країни.

Сучасна освіта направлена на формування цілісної інформаційної культури майбутнього громадянина інформаційного суспільства. Рішення цієї задачі можливе лише тоді, коли в освіту прийде вчитель інформатики, який сам володіє цілісною інформаційною культурою, інформаційним світоглядом і здатний сформувати їх у своїх учнів.

Тому формування інформаційного світогляду вчителя інформатики – одне з пріоритетних завдань сучасної педагогічної освіти.

У навчально-методичному посібнику надаються теоретичні відомості та практичні прийоми по проєктуванню та створенню реляційних баз даних і додатків баз даних, як головної частини програмного забезпечення сучасних інформаційних систем. Питання розглядаються на прикладах системи управління базою даних Access 2019. Приведено етапи створення бази даних та основні принципи нормалізації. Розглянуто створення таблиць різної структури, установка зв'язків між таблицями, створення запитів. Значна увага приділена в навчально-методичному посібнику створенню різних по призначенню та структурі форм, порядку налагодження форм, побудові та модифікації звітів. Пропонується перелік лабораторних робіт.

Особлива увага надається формуванню в майбутніх вчителів закладів загальної середньої освіти знань і умінь щодо роботи з офісним програмним забезпеченням, оволодінню основними навичками роботи з ПК та підготовці до II етапу Всеукраїнської студентської олімпіади з інформатики.

РОЗДІЛ І. ТЕОРЕТИЧНИЙ БЛОК

Лекція 1. СТВОРЕННЯ ТАБЛИЦЬ БАЗИ ДАНИХ В ACCESS

1.1. Основні об’єкти СУБД Access

Access працює з такими типами об’єктів: таблиці, запити, форми, звіти, макроси і модулі. Всі вони згруповані за категоріями і відображаються в області переходів базиданих.

Таблиці – це основний об’єкт бази даних, оскільки саме в них містяться всі дані. Наприклад, база даних нотаріуса може містити таблицю *Клієнти*, в якій зберігаються імена клієнтів, адреси електронної пошти та номери телефонів. У термінології Accessрядки таблиці називають **записами**, а стовпці – **полями** (Рис. 1.1). Для кожного поля таблиці при його створенні призначається своє ім’я, вибирається тип даних і можуть задаватися різноманітні параметри форматування даних. Записи таблиці містять дані у тих форматах полів, які призначає їм розробник. Тому при визначенні таблиці реляційної бази даних кажуть, що **таблиця БД – це сукупність однотипних записів**.

Реальні бази даних містять безліч пов’язаних таблиць. Група пов’язаних таблиць називається **схемою бази даних**. Завдяки зв’язкам інформація з однієї таблиці стає доступною для іншої, що дозволяє забезпечити цілісність даних. У добре спроєктованої бази даних у кожній таблиці зберігаються дані про конкретний об’єкт, наприклад, про клієнтів чи договори

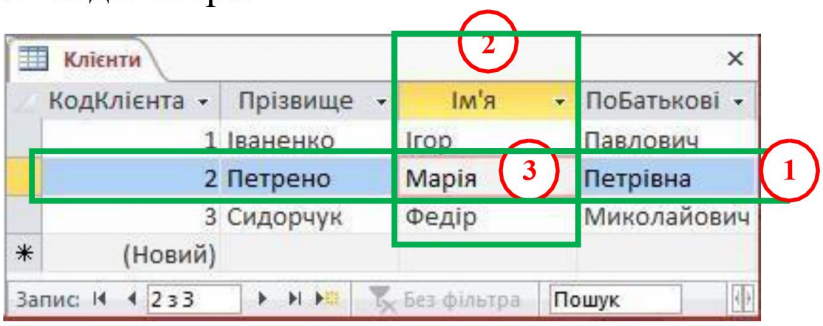


Рис. 1.1. Структура таблиці: 1 – запис, який містить конкретні дані про певний об’єкт; 2 – поле з даними про один аспект елемента таблиці; 3 – значення поля.

Форми – це об’єкт бази даних, який можна використовувати для створення інтерфейсу користувача для програми бази даних. Форма подібна до бланка, що підлягає заповненню, або до маски для на

введення даних. Бланк-форма дозволяє спростити процес заповнення бази даних. Маска-форма дозволяє обмежити обсяг інформації, доступної користувачеві, який звертається до БД. Звичайно форма безпосередньо з'єднана з джерелом даних, наприклад з таблицею або запитом, і може використовуватися для введення, змінення або відображення даних із джерела даних.Проте можна створити «вільну» форму, яка не пов'язана безпосередньо з джерелом даних, але яка може містити кнопки, текстові поля, підписи та інші елементи керування, необхідні для роботи програми.

Запити призначені для модифікації і відбору записів на основі певних критеріїв.Можна використовувати запити за зразком (Queries By Example) – QBE-запити, параметри яких задаються у вікні конструктора запитів. Також для створення запиту можна використовувати інструкції структурованої мови запитів (Structured Query Language) – SQL-запити.

За допомогою запитів можна вибирати дані з однієї або декількох пов'язаних таблиць, здійснювати логічні та арифметичні операції над даними, групувати дані у нові таблиці за певними ознаками.

Звіти використовуються для відображення даних з таблиць і запитів у зручномудля перегляданння вигляді з використанням засобів форматування, підведення підсумків, фільтрації і графічного перетворення на гістограми та діаграми. Часто звітвиступає як об'єкт, призначений для створення документа, який у подальшому може бути роздрукований або включений у документ іншого додатка. За допомогою звітів можна переглядати, форматувати і групувати інформацію у БД Access. Наприклад, можна створити простий звіт у вигляді списку номерів телефонів усіх контактів або зведений звіт за підсумками продажу компанії у різних регіонах за різні періоди.

Макроси – це найпростіші програми мовою макрокоманд (мовою сценаріїв), які дозволяють при звертанні до заздалегідь створених запитів, форм, звітів, виконувати певні дії. До складу БД доцільно включати макроси, які автоматизують процеси її заповнення, відбору інформації тощо. У програмі Access макроси не створюються за допомогою макрорекордера, а призначаються розробником. Тобто для створення макросу треба звернутися до режиму конструктора, в якому з'являється можливість вибору дії та аргументів для макросу.

Модулі являють собою об'єкти, які містять процедури опрацювання подій або виконання обчислень, написані мовою Visual Basic for Application (VBA). За допомогою модулів можна вирішити

широкий клас завдань з пошуку та перетворення інформації в базі даних.

Відмінною особливістю СУБД Access є те, що вся інформація, що відноситься доконкретної БД, зберігається в одному файлі. Це є доволі зручним, особливо для користувачів.

Зараз СУБД Access є безумовним лідером серед десктоп-продуктів по роботі з великими БД. Завдяки вивіреному інтерфейсу, кількості наочних підказок, інтегрованим шаблонам й ефективним інструментам, програму може освоїти навіть початковий користувач, який не мав раніше досвіду роботи з БД.

Access – це однаково просте рішення як для ведення домашньої бухгалтерії і невеликого бізнесу, так і супроводу БД великих промислових підприємств.

1.2. Порядок створення БД у MS Access

Інтерфейс Access 2019 має деякі змінення порівняно з попередніми версіями Access 2007, Access 2010 та Access 2013, Access 2016. Два основних компоненти інтерфейсу користувача – стрічка та область переходів – вперше з'явилися в Access 2007. В Access 2013 став можливим український інтерфейс, стрічка дещо змінилася і додано третій компонент інтерфейсу користувача – подання Backstage (набір команд навкладці *Файл* на стрічці).

Саме подання Microsoft Office Backstage відкривається при запуску Access 2013, Access 2016 або Access 2019 (Рис. 1.2). Backstage містить команди для створення чи відкривання БД, публікації їх в Інтернеті на сервері SharePoint Server і виконання багатьох завдань з обслуговування файлів і БД. У поданні Backstage також доступні команди налаштування, обслуговування і спільного використання БД. Звичайно команди в поданні Backstage застосовуються до всієї БД, а не до окремих її об'єктів.

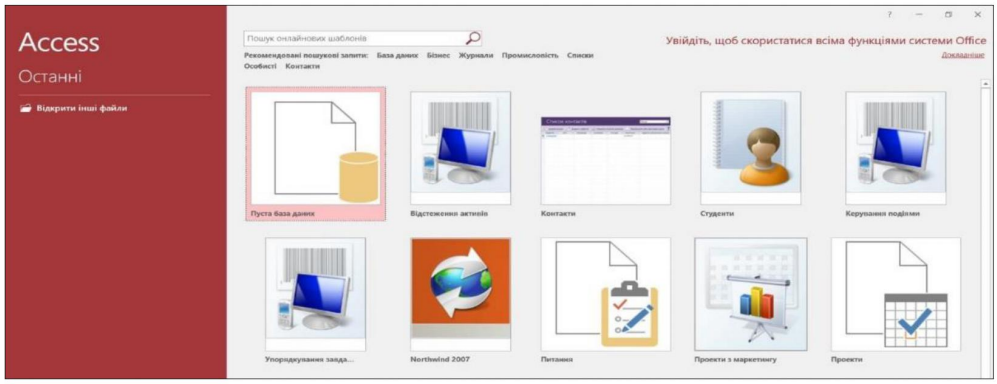


Рис. 1.2. Подання Backstage при відкриванні MS Access 2019

У Microsoft Access є значна кількість шаблонів БД, які дозволяють прискорити створення БД.

Шаблон – це заготовка БД, яка містить структуру таблиць, запитів, форм і звітів, необхідних для виконання певного завдання. Наприклад, існують шаблони, використовувані для відстеження проблем, управління контактами чи то записамивитрат. Шаблони БД можна використовувати як вони є або налаштувати відповідно до конкретних вимог.

Послідовність дій для **створення БД за допомогою шаблона** може бути такою:

1) запустити Access 2019 і вибрати один із доступних шаблонів, наприклад *Студенти*, а у полі *Ім'я файла* ввести ім'я, наприклад, *Студенти*;

2) також треба клацнути по значку папки поряд з полем *Ім'я файла* та вибрати папку для зберігання БД;

3) натиснути кнопку *Створити*.

Після цього відкриється вікно з новою порожньою таблицею (Рис. 1.3).

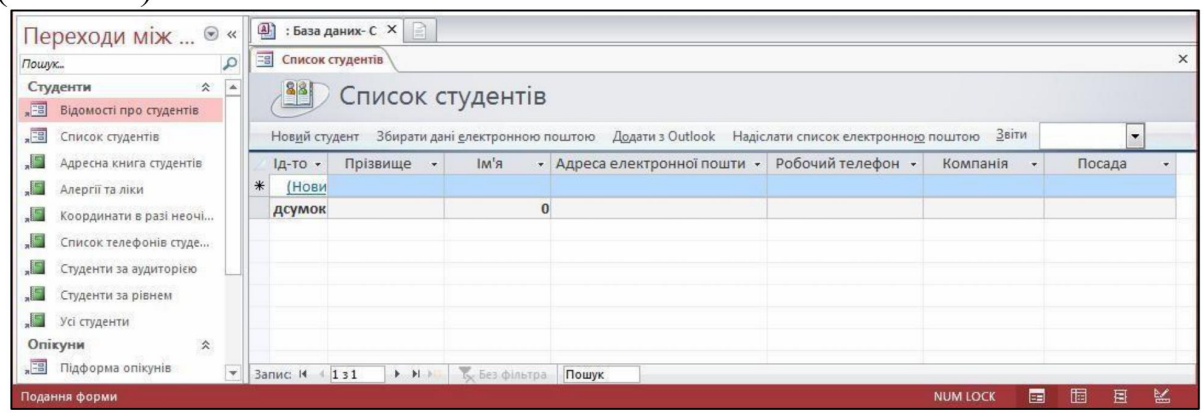


Рис. 1.3. Вікно БД, створеної за допомогою шаблона *Студенти*

Створити нову (порожню) БД можна за допомогою команди *Файл – Створити – Пуста настільна база даних* і задати ім'я та місце розміщення нової БД. Після цих дій відкриється вікно БД.

Розглянемо основні елементи інтерфейсу СУБД Access 2019 за допомогою цього вікна.

Головним елементом інтерфейсу користувача MS Access 2019 є стрічка, яка розміщена вздовж верхньої частини вікна (Рис. 1.4). Стрічка управління містить вкладки: **Файл, Основне, Створення, Зовнішні дані, Знаряддя бази даних, Довідка**.

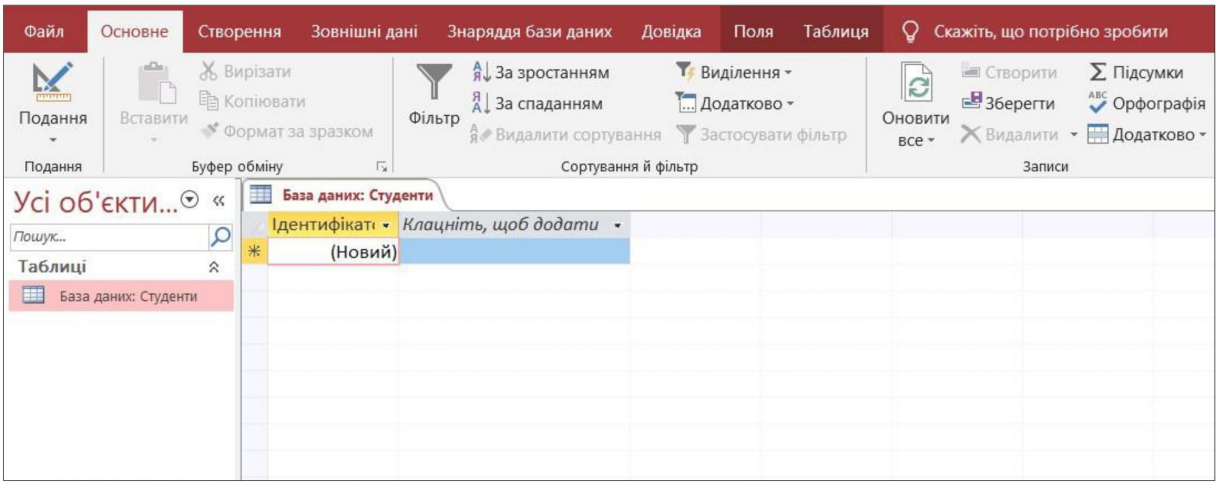


Рис. 1.4. Основні елементи головного вікна програми Access

Кожна вкладка пов’язана з видом виконуваної дії. Вибравши одну з вкладок на стрічці, можна побачити розміщені на ній команди у вигляді піктограм кнопок з позначенням виконуваних функцій. При цьому деякі з кнопок на вкладках стрічки надають вибір дій, якщо на піктограмі є стрілка. Крім того, деякі з груп команд на вкладках, наприклад, групи *Буфер обміну* і *Форматування тексту* на вкладці Основне, у своєму правому нижньому куті мають піктограму-запускач для відкривання відповідного діалогового вікна (Рис. 1.4).

Крім основних п’яти вкладок, стрічка передбачає роботу з контекстними вкладками, які з’являються лише тоді, коли їхнє використання допустимо. Наприклад, при конструюванні таблиць на стрічці з’являється додаткова вкладка Робота з таблицями, яка включає дві вкладки *Поля* і *Таблиця* (Рис. 1.4).

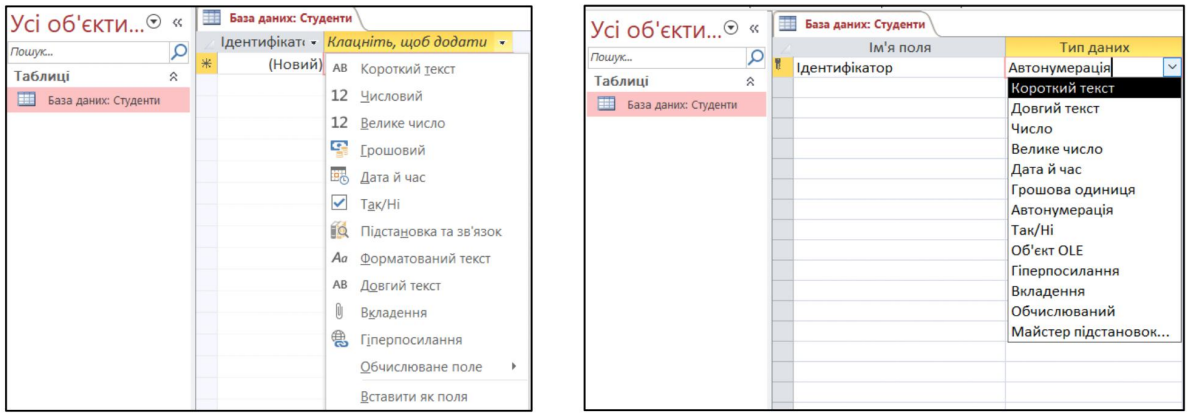
1.3. Типи даних полів

При створенні полів таблиці кожному з них треба задати відповідний тип даних. В Access 2019 існує ціла низка типів даних, деякі з них мають свої підтипи.

В Access 2019 при відкриванні режиму створення таблиці можна відразу вибрати необхідний формат поля. При натисканні на стрілку в рядку з найменуванням полівтаблиці відкриється список з доступними типами даних (за замовчуванням задається тип *Короткий текст*)

(Рис. 1.5, а).

Іншим способом вибору типу даних конструйованого поля є перехід з режиму *Таблиця* до режиму *Конструктор*. У цьому режимі для кожного поля можна вибрати свій тип даних (Рис. 1.5, б).



а) б)

Рис. 1.5. Список з можливими типами даних для поля таблиці:
а) список типів, які можна вибрати для поля у режимі *Таблиця*;
б) список типів, які можна вибрати для поля у режимі *Конструктор*

Короткий текст використовується для зберігання таких даних, як імена та адреси,а також для чисел, які не потребують обчислень, таких як номери телефонів, інвентарніномери або поштові індекси. За замовчуванням усім полям таблиці в Access 2019 надається саме цей тип даних. У текстовому полі може розміщуватись до 255 знаків, алеза замовчуванням встановлюється розмір поля 50 знаків. Саме властивість Розмір поля визначає максимальну кількість знаків, які можна ввести у текстове поле. Для економії пам'яті при використанні текстових даних доцільно для кожного поля задати його розмір (кількість символів).

Довгий текст *(поле Мемо)* – тип даних для текстових полів з довжиною понад 255символів. У полі цього типу може зберігатися до 65 536 символів. Використовуються поля такого типу для зберігання абзаців тексту: резюме, коротких описів тощо. При цьому зникає потреба піклуватися про зазначення довжини запису, який буде зберігатись у пам'яті машини, оскільки для полів з типом даних Довгий текст буде виділено рівно стільки пам'яті, скільки буде потрібно для зберігання усіх введених символів. Поля цього типу, як і текстові поля, можна сортувати або групувати, але при цьому Access використовує тільки перші 255 знаків.

Число – тип, який використовується для зберігання числових даних, що братимутьучасть у математичних обчисленнях, за винятком грошових розрахунків. Тип і розмір значень числового поля можна змінити у властивості *Розмір поля*. Щоб вказати конкретний спосіб зберігання даних у полях числового типу, передбачено декілька підтипів: *Байт, Ціле число, Довге ціле число, Одинарне значення, Подвійне значення, Ідентифікатор реплікації, Десяткове значення*. Наприклад, поле типу Байт допускає введення тільки цілих чисел (без дійсної частини) від 0 до 255. Найпоширеніші значення – *Подвійне значення* (Double) та *Довге ціле число* (Long Integer). Якщо поле буде пов'язане з полем лічильника зв'язком «один-до-багатьох», тип даних цього поля має бути *Довге ціле число* (Рис. 1.6).

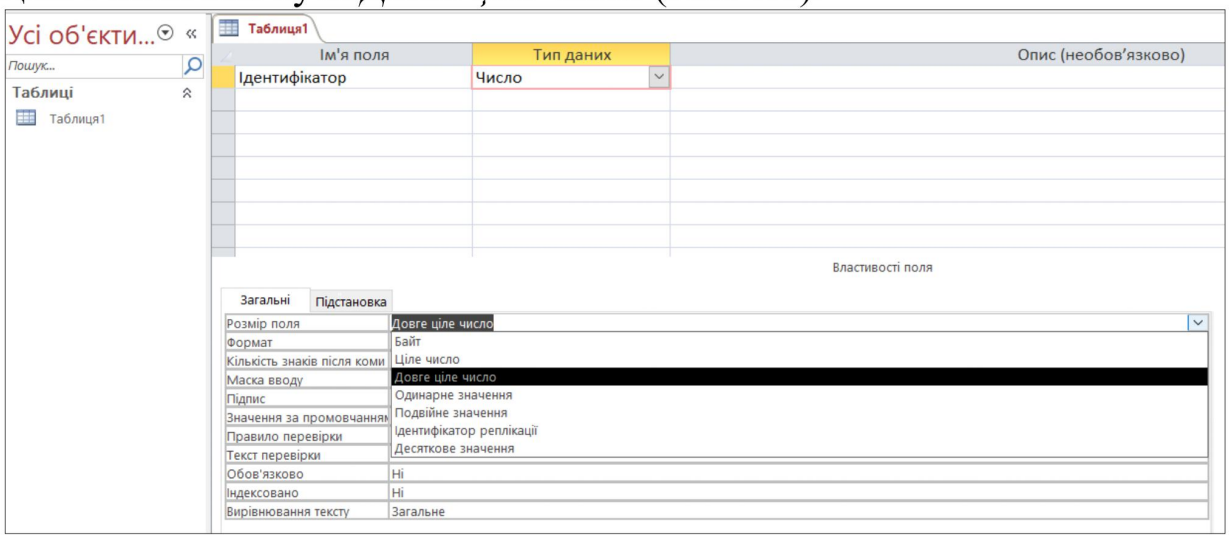


Рис. 1.6. Тип даних поля *Довге ціле число*

Грошовий *(Грошова одиниця)* тип даних використовується в базі даних MS Access для проведення розрахунків з грошовими значеннями або для обчислень з фіксованою комою, в яких потрібна висока точність. Грошовий тип поля використовують для запобігання округлення під час обчислень. У полях цього типу забезпечується 15 знаків ліворуч від десяткової коми і 4 знаки праворуч. Грошове поле займає 8 байтів на диску. Над такими даними можна виконувати різноманітні арифметичні операції, як над типом даних *Число*.

Дата й час – тип даних, призначений для зберігання значень дати й часу, який дозволяє здійснювати обчислення над цими даними. Для різноманітного відображенняданих й часу існують спеціальні формати у вигляді шаблонів, наприклад: *Повний форматдати* (за замовчуванням) – «12.11.2015 17:34:23», *Довгий формат дати* – «12 вересня 2021 р.», *Короткий формат дати* – «12.09.2021», *Довгий формат часу* –

«12:42:46», *Короткий формат часу* – «12:42» тощо.

Автонумерація – тип даних поля, в якому при додаванні нового запису автоматично сформується унікальне ціле число, за замовчуванням на одиницю більше попереднього: 1, 2, 3, … Створений номер не може бути змінений. Тип автонумерації (лічильника) зручно використовувати як первинний ключ таблиці. Крім послідовно зростаючих на одиницю чисел, поле автонумерації може генерувати ідентифікатори реплікації (які ще називають GUID – глобальні унікальні ідентифікатори), для чого треба у властивості *Розмір поля* вибрати відповідне значення.

Так/Ні – це логічний тип даних поля ( ), за якого поле може мати одне з двох значень: «Так» або «Ні». Крім формату *Так / Ні*, заданого за замовчуванням, в Access передбачено ще два формати (властивість *Формат*) для відображення даних логічного типу*: Істина/Хибність* та *Увімк./Вимк.* Значення *Null* для логічних даних не допускається.

Об'єкт OLE – тип даних, використовуваний для зберігання об'єктів з інших додатків, зв'язаних або вбудованих у БД Microsoft Access. Термін OLE розшифровується як Object Linking and Embedding – зв'язування і вбудовування об'єкта. У полі можуть розміщуватись рисунки, електронні таблиці, відеофільми у двійковому форматі. Такий тип даних зберігає дані розміром до 1 Гб.

Гіперпосилання забезпечує зв'язок з вебсторінкою або будь-яким файлом. При виборі цього типу даних Access автоматично запускає додаток, в якому можна відобразити цей файл, наприклад, браузер для відображення вебсторінки або програмний модуль з розширенням *exe*.

Вкладення – тип даних, який з'явився в Access 2007/2010, для зберігання безпосередньо у базі даних файлів значних розмірів (рисунків, додатків MS Office та інших видів) у стислому вигляді розміром до 2 Гб. вибрати опції поля, яке буде або запозичено з іншої таблиці, або заповнюватись значенням із самостійно створеного списку.

Майстер підстановок (*Підстановка і зв'язок*) – за суттю це не тип даних, а властивість поля, використовувана для більш ефективного і коректного введення даних. При виборі майстра підстановок запуститься вікно *Майстер підстановок*, в якому можна вибрати опції поля, яке буде або запозичено з іншої таблиці, або заповнюватись значенням із самостійно створеного списку (Рис. 1.7).

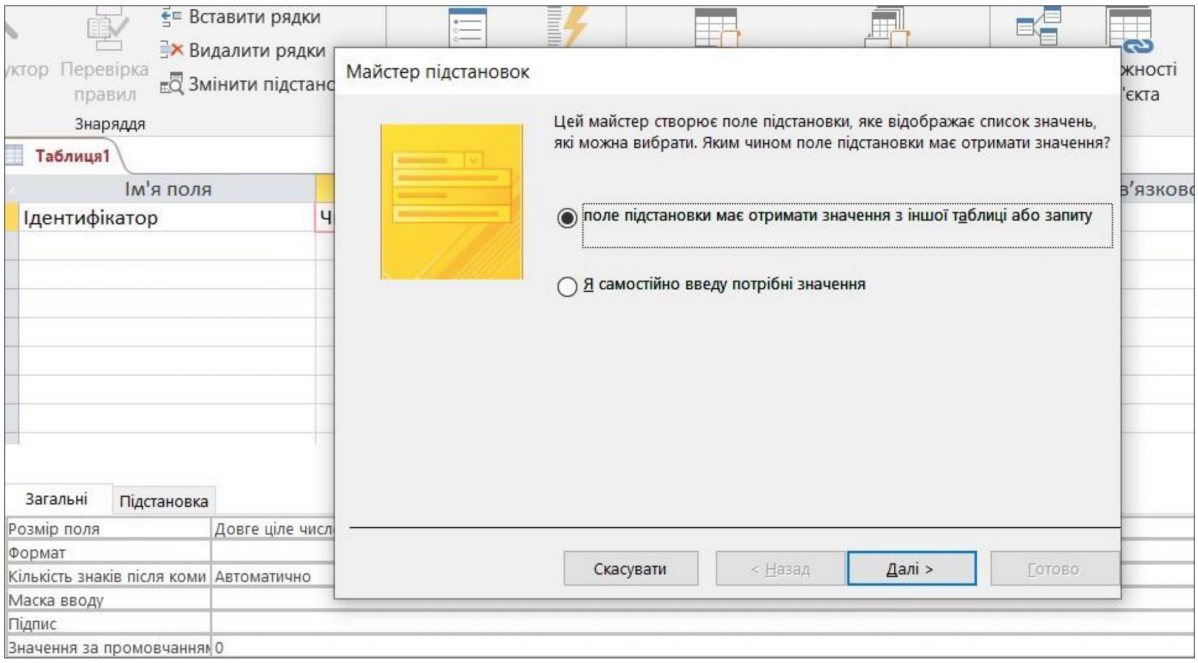


Рис. 1.7. Майстер підстановок

1.4. Способи створення таблиць

В Access для створення (конструювання) таблиць передбачено режими *Таблиця* і *Конструктор таблиць*. Далі на конкретних прикладах створення таблиць будуть розглянуті особливості кожного режиму. Але крім цих режимів, таблиці можна імпортувати із зовнішніх джерел, наприклад, таблиць Excel або інших БД

Створення таблиць у режимі *Таблиця*

Цей режим зручний тим, що розробник може сам задавати назву полів, вибиратидля них типи даних і безпосередньо вводити дані.

Розглянемо порядок створення таблиці *Надбавки* з розмірами залежно від стажу роботи працівників фірми у режимі *Таблиця* (табл. 1.1)

Таблиця 1.1.

Код	**Стаж**	**Надбавка (грн.)**
1	Менше 1 року	0
2	Від 1 року до 3-х років	150
3	Від 3-х років до 5 років	400
4	Понад 5 років	600

Порядок створення таблиці *Надбавки* у новій БД в Access буде таким:

1. Створити нову БД з ім’ям *Працівники*.
2. Вибрати на вкладці *Створення* у групі *Таблиці* команду *Таблиця*. Це призведедо відкривання заготовки такого вигляду:

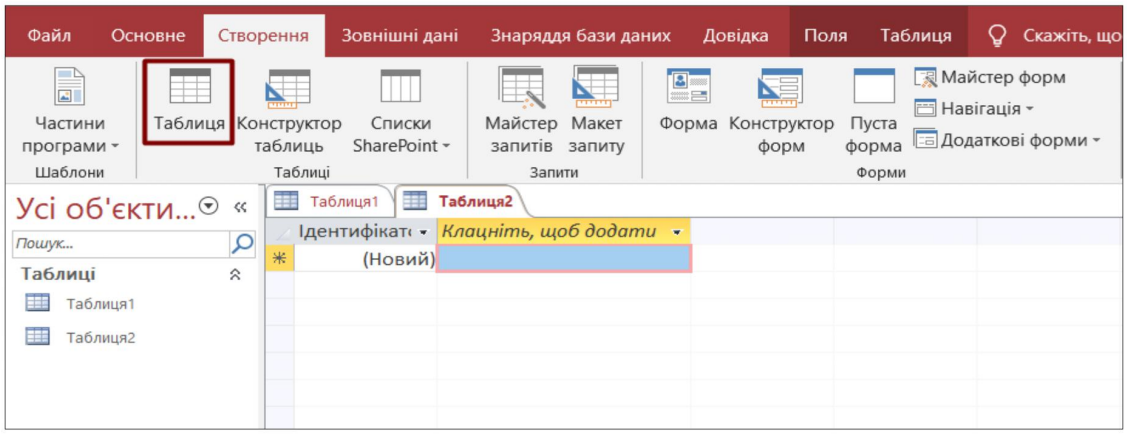


Рис. 1.8. Створення таблиць у режимі *Таблиця*

При цьому першим полем автоматично створиться поле *Ідентифікатор* з типом даних *Автонумерація*, яке пропонується використовувати як ключ.

3. Натиснувши на полі *Клацніть, щоб додати*, вибрати з розкривного списку(Рис. 1.5, а) тип другого створюваного поля *Стаж – Короткий текст*.
4. Змінити імена полів *Ідентифікатор* на *Код*, а *Поле1* на *Стаж* (згідно з табл. 1.1) за допомогою команди контекстного меню *Перейменувати поле*.
5. Додати третє поле з ім’ям *Надбавка* і форматом даних *Грошовий*.
6. Заповнити створену таблицю даними з табл. 1.1, у результаті чого заповненатаблиця набуде вигляду (Рис. 1.10):

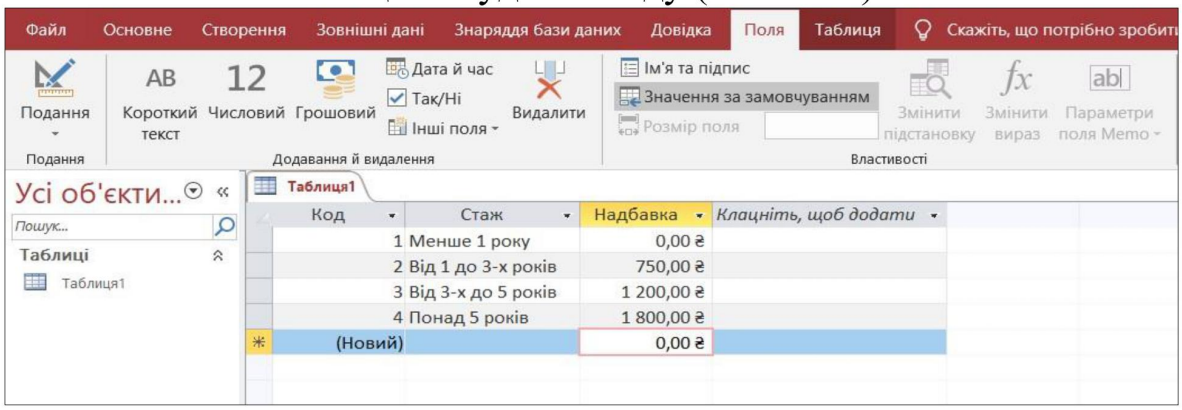


Рис. 1.10. Створена таблиця у режимі *Таблиця*

7. Закрити таблицю, натиснувши кнопку у правому верхньому куті таблиці.

При цьому система задасть питання про доцільність зберігання таблиці. Якщо відповісти «Так», з'явиться діалогове вікно *Зберігання*, в якому треба змінити ім'я *Таблиця1* на *Надбавки* і натиснути кнопку *ОК*.

8. У подальшому змінити ім'я таблиці можна з області переходів. Тільки слід пам'ятати, що таблиця у цей момент має бути закритою. При натисканні на праву кнопку миші з'явиться контекстне меню, в якому слід вибрати команду *Перейменувати*.

Створення таблиці у режимі *Конструктор таблиць*

Створення нової таблиці у цьому режимі дозволяє виконувати такі операції: додавати поля, задавати формати полів, змінювати ключове поле таблиці, вибирати дані з готовихтаблиць, будувати арифметичні і логічні вирази для контролю даних при введенні.

Особливості режиму *Конструктор таблиць* розглянемо на прикладі створення таблиці *Працівники* з даними про працівників організації (прізвища, імена, посади, датиприйому на роботу тощо).

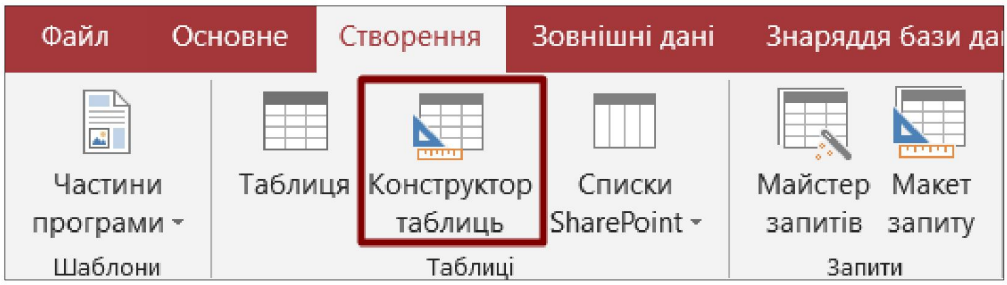


Рис. 1.11. Створення таблиці у режимі *Конструктор таблиць*

Порядок створення цієї таблиці буде таким:

1. Для запуску цього режиму в Access 2019 треба на вкладці *Створення* у групі *Таблиці* вибрати команду *Конструктор таблиць*.

Вікно конструктора структурно складається з трьох розділів (стовпців): *Ім'я поля*, *Тип даних* і *Опис*.

2. У розділі *Ім'я поля* треба почергово вписати імена створюваних полів, а у розділі *Тип даних* для кожного поля вибрати зі списку (Рис. 1.12) відповідний тип даних. У розділі *Опис*, який не є обов'язковим для заповнення, розробник може задати коментар, щоб булозрозуміло, які дані будуть зберігатися у цьому полі.



Рис. 1.12. Вигляд вікна при створенні полів таблиці *Працівники* у режимі *Конструктор таблиць*

Примітка. Хоча Access дозволяє пробіли у назвах полів, таблиць та інших об’єктів БД, краще все-таки створювати імена без пробілів, оскільки ті можуть при деяких обставинах спричиняти конфлікти у роботі з іншими системами (наприклад, в SQL-запитах, у програмах VBA тощо). Саме тому пробіли в іменах об’єктів баз даних не рекомендується використовувати.

3. Для зберігання і присвоєння таблиці імені треба закрити таблицю, клацнувши по значку, який знаходиться на сірій панелі з ім’ям таблиці *Таблиця1* праворуч. При цьому система сформує повідомлення з пропозицією зберегти змінення макета або структури таблиці. Натиснути кнопку *Так*.

Відразу після цього буде сформовано діалогове вікно, в якому, замість імені *Таблиця1*, яке пропонується за замовчуванням, треба ввести ім’я нової таблиці, наприклад *Працівники*, і натиснути кнопку *ОК* (Рис. 1.13).

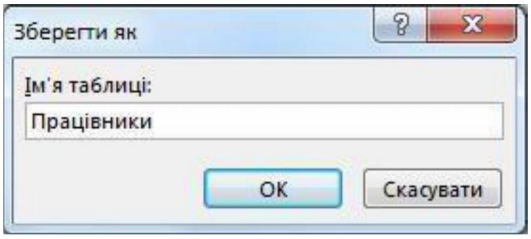


Рис. 1.13. Діалогове вікно зміни імені таблиці

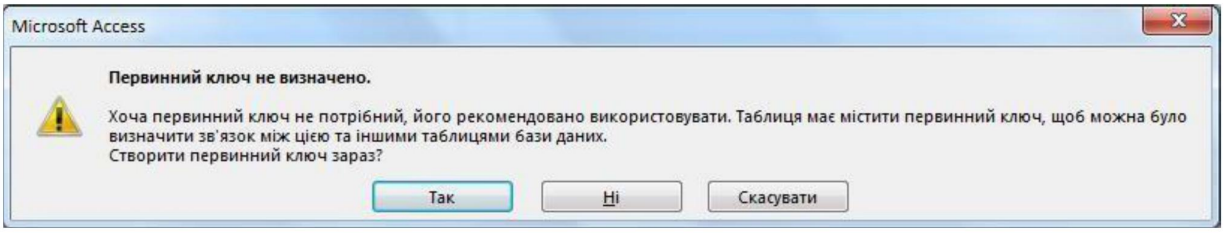


Рис. 1.14. Створення первинного ключа

Внаслідок цього система сформує діалогове вікно з пропозицією створення ключового поля. Натиснути кнопку *Так* (Рис. 1.14).

Як наслідок цього в таблиці буде автоматично сформовано нове ключове поле *Ідентифікатор* з типом даних *Автонумерація*, що надалі дозволить встановити зв'язки з іншими таблицями БД. На відміну від такого способу створення ключового поля таблиці, розробник і сам може задавати ключові поля у режимі *Конструктор*, вибираючи тип (здебільшого *Автонумерація*, *Довге ціле* або *Текст*) та команду *Ключове поле* у контекстному меню або на вкладці *Конструктор*.

Отже, таблиця *Працівники* створена, але ще не заповнена даними. Вводити дані у таблицю можна різними способами: вручну у режимі *Таблиця*, за допомогою майстра підстановок у режимі *Таблиця*, за допомогою імпорту даних з інших джерел,за допомогою долучення файлів, за допомогою майстра підстановок у режимі *Конструктор*.

4. Якщо Ви вийшли з режиму *Конструктор таблиць*, доцільно повернутися до нього, після чого додати коментар у розділі *Опис* та уточнити властивості полів таблиці *Працівники*.

Відкрити таблицю у режимі *Конструктор* можна, натиснувши на назві таблиці правою кнопкою миші і вибравши з контекстного меню команду *Конструктор* (Рис. 1.15).

Задати властивості полів таблиці можна у нижній частині конструктора у розділі

Властивості поля, де є дві вкладки *Загальні* і *Підстановка* (Рис. 1.7). На вкладці *Загальні* відображається перелік властивостей виділеного поля, наприклад: для текстових полів у властивості *Розмір поля* система за замовчуванням задає 255 символів, а користувач для економії місця може зменшити це значення до доцільного. Зауважте, що в таблиці з'явилось нове поле *Ідентифікатор*.

4.1. Для поля *Ідентифікатор* (тип даних *Автонумерація*) змінити ім'я поля на *КодПрацівника* (без пробілів).

4.2. Для поля *Прізвище* (тип даних *Короткий текст*):

– задати у властивості *Розмір поля* значення 30 (як максимальну кількістьсимволів у прізвищах працівників);

– у властивості *Обов'язково* вибрати зі списку значення *Так*;

у розділі *Опис* вписати «Обов'язкове поле (до 30 символів)».

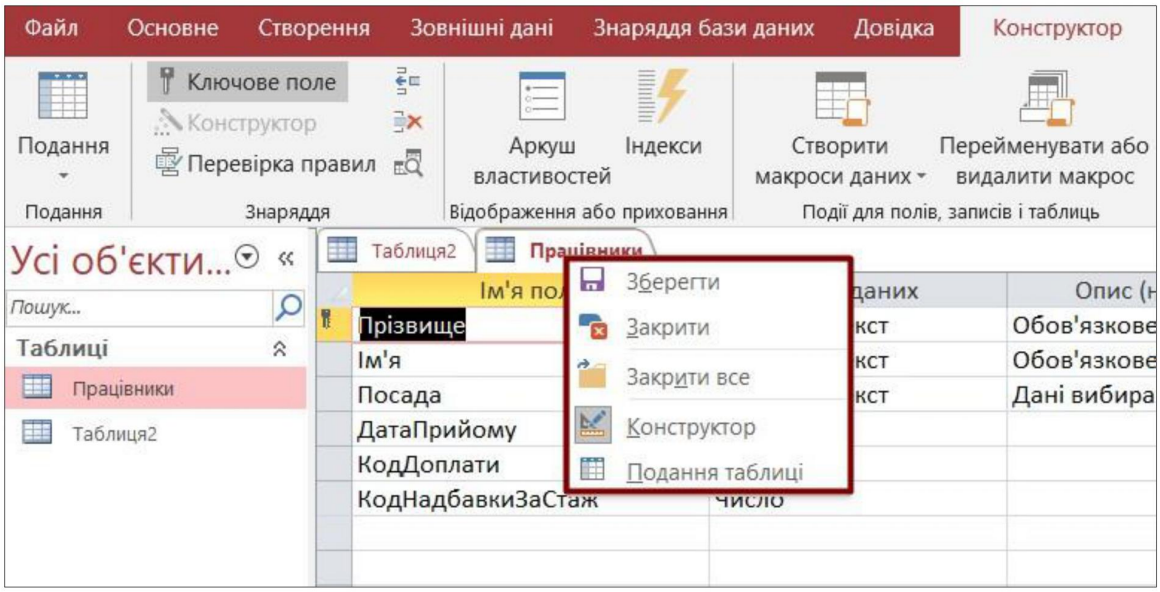


Рис. 1.15. Відкриття таблиці у режимі *Конструктор* за допомогою контекстного мені

4.3. Для поля *Ім'я* (тип даних *Короткий текст*):

– задати у властивості *Розмір поля* значення 15;

– у властивості *Обов'язково* вибрати зі списку значення *Так*;

– у розділі *Опис* вписати «Обов'язкове поле (до 15 символів)».

4.4. Для поля *Посада* (тип даних *Короткий текст*):

– задати у властивості *Розмір поля* значення 25;

– у розділі *Опис* вписати «Дані вибиратимуться з таблиці».

4.5. Для поля *ДатаПрийому* (тип даних *Дата й час*):

– у властивості *Обов'язково* вибрати зі списку значення *Так*;

– у властивості *Текст перевірки* вписати «Вкажіть дату прийому на роботу».

4.6. Для полів *КодДоплати* і *КодНадбавкиЗаСтаж* (тип *Число*):

– у властивості *Розмір поля* вибрати зі списку значення *Довге ціле*;

– у властивості *Обов'язково* вибрати зі списку значення *Так*.

Після цього вікно конструювання таблиці набуде вигляду, який показано на рис. 1.16

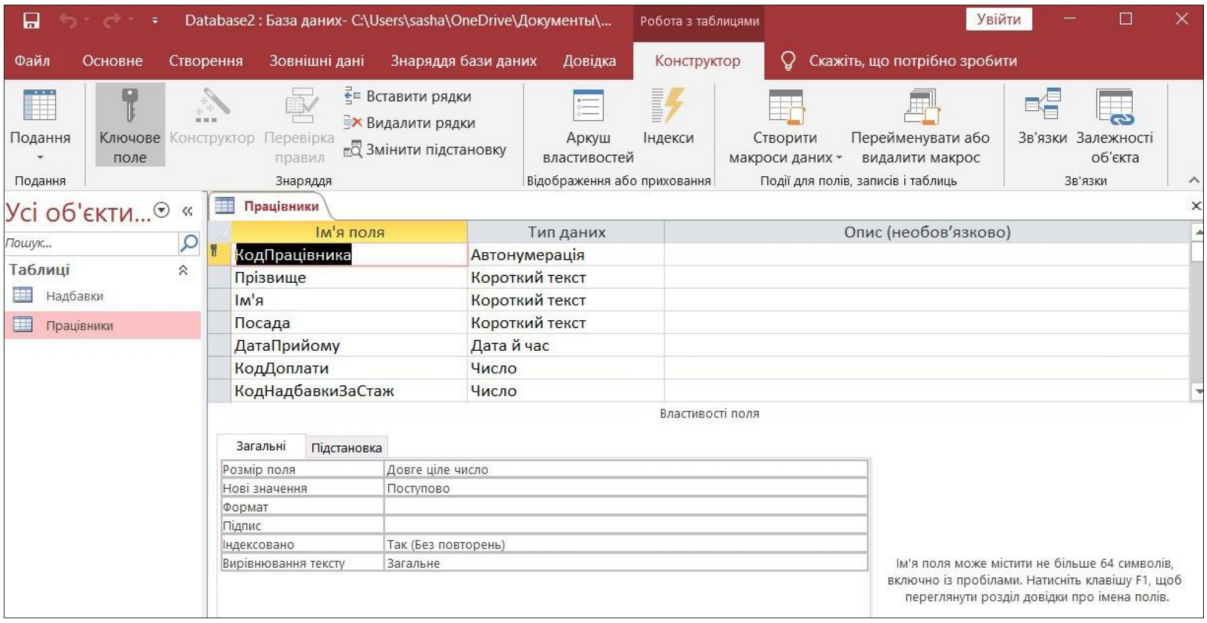


Рис. 1.16. Вигляд вікна конструктора таблиці *Працівники* після уточнення властивостей полів

5. Для заповнення створеної таблиці треба перейти до режиму таблиці, скориставшись на вкладці *Основне* командою *Подання – Подання таблиці*. На повідомлення з пропозицією зберігання таблиці натиснути кнопку *Так*. Заповнити таблицю довільними даними (не менше десяти записів). При цьому у необов'язкових полях можна заповняти не всі записи. Поле *Посада* буде заповнятися пізніше шляхом імпорту даних з іншої таблиці. А поле *КодДоплати* треба заповняти значеннями від 1 до 4, як і поле *КодНадбавки*, що зумовлено набором значень у відповідних таблицях.

Зауважте, що у заголовку кожного поля праворуч є значок (стрілка донизу), це розкривний список з деякими функціями, аналогічними до Excel, які дозволяють сортувати дані, здійснювати пошук даних у виділеному полі, використовувати фільтрацію даних (Рис. 1.17).

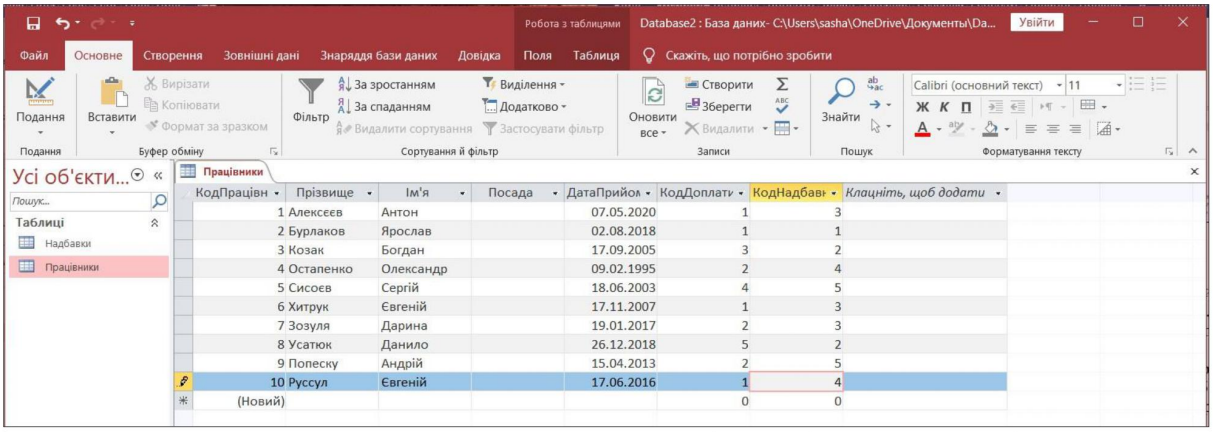


Рис. 1.17. Розкривний список з деякими функціями, аналогічними до Excel

Закрити таблицю, натиснувши кнопку у правому верхньому куті таблиці. При цьому система задасть питання про доцільність зберігання змінень макета таблиці *Працівники*. Треба відповісти *Так*.

6. У режимі *Конструктор таблиць* треба створити у цій самій БД ще одну таблицю *Анкети* з даними особових справ працівників організації: код працівника, дата народження, адреса, освіта, телефон, фотографія, телефон, сімейний стан (Рис. 1.18):

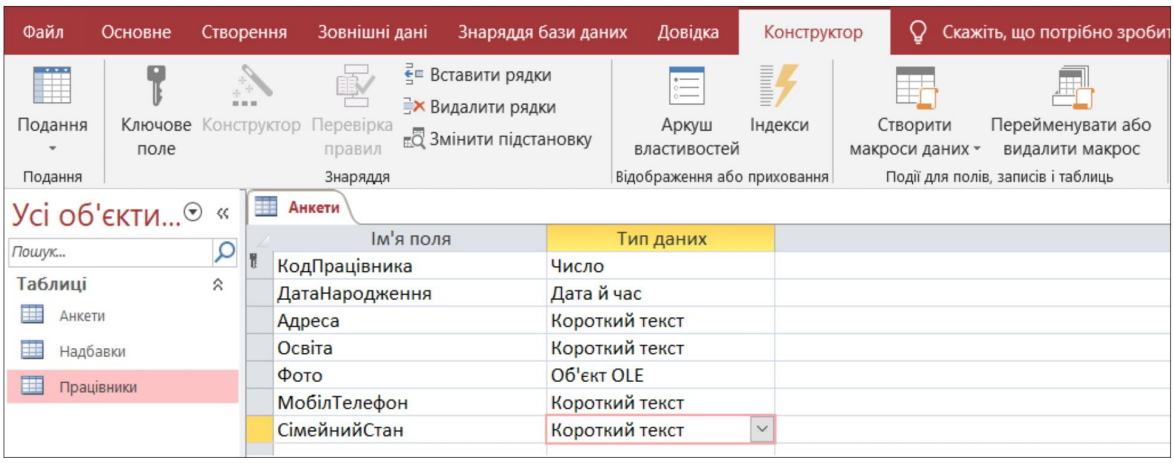


Рис. 1.18. Створення таблиці *Анкети*

Для поля *КодПрацівника* задати тип *Число* розмір поля *Довге ціле.*

Для поля *МобілТелефон* (тип *Короткий текст*) доцільно задати маску введення, для цього треба у властивості *Маска вводу* вписати: (###)###-###-##. Система переведецю маску до вигляду: \(###\)###\-

##\-##. У цій масці перші 3 цифри – код мобільного оператора, далі 7 цифр – номер телефону. При заповненні даних в це поле користувач вводитиме тільки цифри номера телефону, наприклад, 0502353360, а створена маска перетворить цей номер до вигляду: (050)235-33-60. Крім цього, для економії місця, можна обмежити розмір цього поля, задавши у властивості *Розмір поля* значення 10 символів.

Після конструювання таблиці *Анкети* треба перейти до режиму таблиці і заповнити її даними (Рис. 1.19.). При цьому поле *Освіта* поки що не треба заповнювати, оскільки дані уньому вводитимуться пізніше із поля підстановки (див. п. 5 цих теоретичних відомостей).

А при заповненні поля *Фото* (тип *Об'єкт OLE*) треба за допомогою будь-якого графічного редактора відкрити файл з фотографією працівника, виконати команду копіювання ([Ctrl] + [C]), перейти у клітинку поля *Фото* таблиці БД відповідного працівника і виконати команду вставки ([Ctrl] + [V]).

Анкети

КодПрацівн	ДатаНарод	Адреса	Освіта	Фото	МобілТелефон	СімейнийСт
1	18.11.2002	вул. Європейська, 8, кв. 50	вища	Package	(097)658-842-3	неодружений
2	28.06.2000	вул. Тищика, 22, кв. 17	середня	Package	(050)845-692-0	неодружений
3	06.05.1999	вул. Гонти, 17, кв. 1	базова середн	Package	(099)873-022-5	неодружений
4	06.07.2003	вул. Велика Фонтанна, 45, кв. 6	базова середн	Package	(091)745-803-6	неодружений
5	18.07.2003	вул. Стуса, 117, кв. 65	науковий ступ	Package	(073)569-842-5	неодружений
6	12.03.2000	вул. Садова, 30. кв. 25	вища	Package	(067)582-020-1	неодружений
7	05.05.2001	вул. Жовтнева, 10 кв. 118	вища	Package	(063)587-445-5	неодружений
8	09.09.2001	вул. Центральна, 184, кв. 243	вища	Package	(073)558-866-9	незаміжня
9	01.02.2001	вул. Б. Хмельницького, 21, кв. 36	середня	Package	(099)556-644-1	неодружений
10	27.10.1999	просп. Богуна, 16, кв. 3	науковий ступ	Package	(098)636-363-6	неодружений
11						

Рис. 1.19. Заповнена таблиця *Анкети*

Лекція 2. ІМПОРТУВАННЯ ДАНИХ ТА ВСТАНОВЛЕННЯ ЗВ'ЯЗКІВ МІЖ ТАБЛИЦЯМИ

2.1. Використання зовнішніх даних – імпортування таблиць

З «нуля» таблиці БД розробляти зовсім не обов'язково, оскільки на підприємствах звичайно є файли, створені у різних додатках, які можна використовувати при формуванні БД в Access. Припустимо, що у відділі кадрів підприємства ведеться реєстр номенклатури посад в Excel, відомості у реєстрі можуть змінюватись або доповнюватись новими записами. Дані з цього реєстру потрібні і для ведення БД про працівників підприємства. Засоби Access дозволяють не лише імпортувати дані з таблиці Excel у таблицю Access, але і, за потреби, зберегти зв'язок з вихідними даними Excel. В Access це називається – робота із зовнішніми даними.

Продемонструємо ці засоби на прикладах створення таблиць *Посади* та *Доплати*, імпортуючи дані з таблиць Excel. Причому, в одному випадку буде збережений зв'язок з вихідними даними Excel, а в іншому – дані будуть просто скопійовані як джерело даних.

Імпортування таблиці з установленням зв'язку

Створити Excel-файл, окремий аркуш якого перейменувати на *Посади*, і створити на цьому аркуші таблицю з переліком посад і базових окладів (Рис. 2.1).

	A	B
1	**Посади**	**Оклад**
2	методист	3600,00
3	фахівець І категорії	3500,00
4	провідний фахівець	3550,00
5	юрист	4200,00
6	інспектор	4000,00
7	програміст	8000,00
8	адміністратор	7500,00
9	модератор	6000,00
10	економіст	7250,00
11	лаборант	3200,00

Рис. 2.1. Створена таблиця з переліком посад та базових окладів

1. Для імпорту цієї таблиці до БД *Працівники* треба в Access на вкладці *Зовнішні дані* у групі *Імпорт і зв'язування* вибрати нове джерело даних і обрати команду *Excel* (піктограма). Це призведе до відкривання діалогового вікна *Отримати зовнішні дані – Таблиця Excel*. За допомогою кнопки *Огляд* треба вибрати потрібний Excel-файл.

За замовчуванням у цьому діалоговому вікні встановлена перша з трьох опцій *Імпортувати дані джерела до нової таблиці в поточній базі даних*. Ця опція дуже зручна для простого копіювання усіх даних з таблиці Excel до нової таблиці Access.

У нашому випадку, оскільки дані у реєстрі можуть змінюватись або доповнюватись новими записами, доцільно створити таблицю, пов’язану з джерелом даних, включивши третю з опцій (Рис. 2.2):

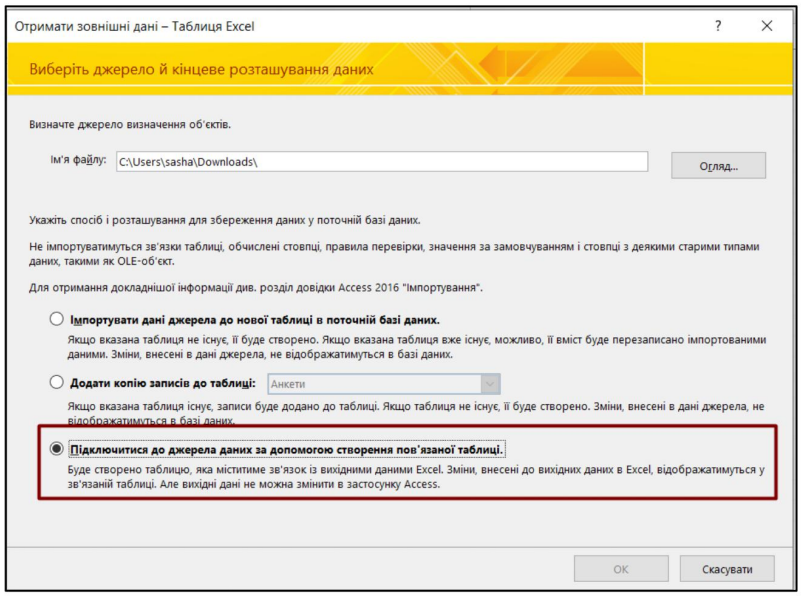


Рис. 2.2. Отримання зовнішніх даних

Після цього натиснути кнопку *ОК*.

2. Наступним кроком з’явиться діалогове вікно *Майстер зв’язування електронних таблиць* з назвами аркушів в імпортованій таблиці Excel, в якому після вибору потрібного аркуша треба натиснути кнопку *Далі*.

3. На наступному кроці слід встановити прапорець опції *«Перший рядок містить заголовки стовпців»* для використання даних з першого рядка в якості імен полів таблиці та натиснути кнопку *Далі*.

4. У наступному діалоговому вікні ввести ім’я нової таблиці, наприклад, *Посади*, і натиснути кнопку *Готово*. Після цього з’явиться повідомлення про завершення зв’язування таблиці *Посади* як джерела даних з Excel з базою даних Access, в якому натиснути кнопку *ОК*.

Після закривання повідомлення у переліку таблиць БД з’явиться новий значок з ім’ям створеної таблиці. Стрілка біля значка таблиці означає, що таблиця має зв’язок із зовнішнім джерелом даних Excel (Рис. 2.3).

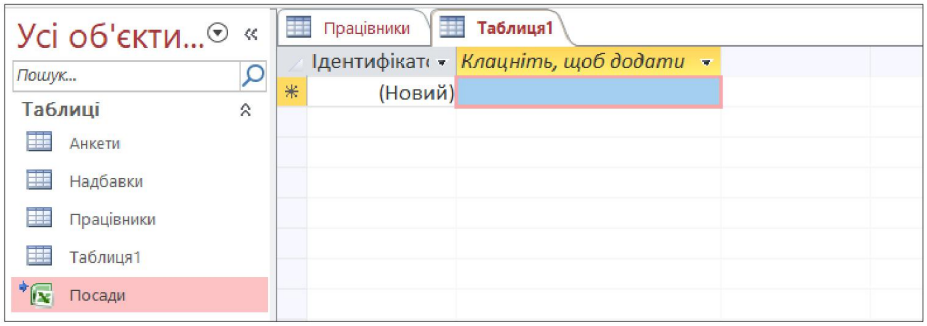


Рис. 2.3. Зв'язок створеної таблиці із зовнішнім джерелом даних Excel

5. Відкрити таблицю *Посади*, двічі клацнувши по значку з її ім'ям, впевнитись у правильності імпортованих даних і закрити таблицю (Рис. 2.4).

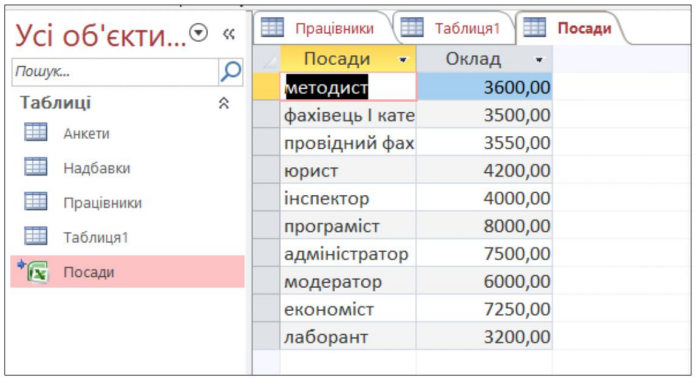


Рис. 2.4. Перевірка правильності імпортування даних

Імпортування (копіювання) таблиці із зовнішнього джерела

1. Створити новий Excel-файл та перейменувати його на *Доплати* та створити таблицю з можливими варіантами доплат (рис. 2.5). Зберегти файл і закрити Excel.

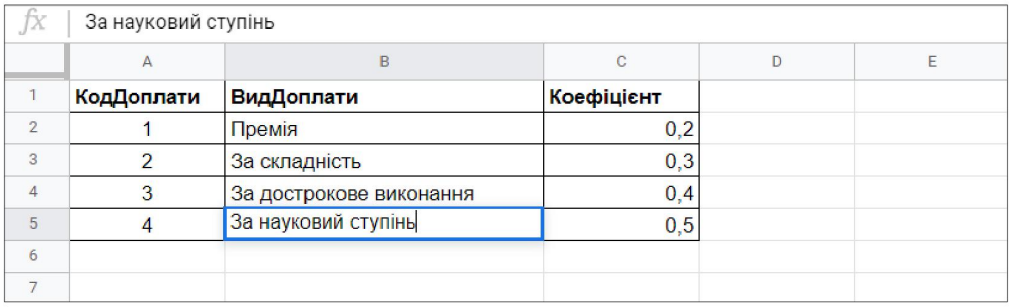


Рис. 2.5. Таблиця з можливими варіантами доплат

2. Для імпорту цієї таблиці в Access треба повернутися до БД *Працівники* і виконати команду *Зовнішні дані – Excel*. У діалоговому вікні *Отримати зовнішні дані – Таблиця Excel* за

допомогою кнопки *Огляд* вибрати файл Excel-файл з таблицею *Доплати* і простежити, щоб була встановлена перша з трьох опцій – *Імпортувати дані джерела до нової таблиці в поточній базі даних* для простого копіювання всіх даних з таблиці Excel до нової таблиці Access. Натиснути кнопку *ОК*.

3. Далі слід встановити опцію *Перший рядок містить заголовки стовпців* для використання даних з першого рядка в якості імен полів таблиці та натиснути кнопку *Далі*.

4. На наступному кроці система запропонує уточнити тип кожного поля імпортованої таблиці. Треба для поля *КодДоплати* вибрати тип *Довге ціле число*, а також зі списку *Індексовано* вибрати *Так (Без повторень)* та натиснути кнопку *Далі.*

5. Далі система порадить задати ключове поле у новій таблиці. Треба увімкнути другу опцію *Вибрати власний первинний ключ* і вибрати зі списку поле *КодДоплати*. Натиснути кнопку *Далі.*

6. У наступному діалоговому вікні задати ім'я *Доплати* для нової таблиці і натиснути кнопку *Готово*. Після цього з'явиться повідомлення про завершення імпортування таблиці. Після закривання повідомлення у переліку таблиць БД з'явиться значок з ім'ям створеної таблиці.

7. Відкрити таблицю *Доплати*, двічі клацнувши по значку з її ім'ям, впевнитись у правильності імпортованих даних і закрити таблицю.

2.2. Створення у таблицях полів підстановки

Доволі часто доводиться вибирати зі списку певні дані, які зафіксовані. Такі списки створюють безпосередньо при проєктуванні полів таблиці. Розглянемо чотири способи створення полів підстановки:

а) при створенні нового поля у режимі *Таблиці*,

б) фіксованого списку в режимі *Конструктора*;

в) при зміненні налаштувань існуючого поля у режимі *Конструктора* з вибиранням значень зі списку іншої таблиці;

г) за допомогою *Майстра підстановок*.

Створення поля з фіксованим списком підстановки

а) у режимі *Таблиця*

Припустимо, виникла потреба здійснити штатну розстановку працівників у відповідності з наявними найменуваннями відділів. Для створення нового поля *Відділ* з фіксованим списком з назвами відділів

організації доцільно скористатися майстром підстановки у режимі *Таблиця*, виконавши декілька кроків:

1. Відкрити таблицю *Працівники* у режимі таблиці.
2. Клацнути у заголовку на стрілці останнього поля *Клацніть, щоб додати* та вибрати команду *Підстановка та зв'язок* (також цю команду можна вибрати на вкладці *Поля* у групі *Додавання й видалення* – *Інші поля*). Це призведе до відкривання діалогового вікна *Майстер підстановки*, в якому на першому кроці треба увімкнути опцію *Я самостійно введу потрібні значення* і натиснути кнопку *Далі*.
3. На другому кроці майстра підстановки треба ввести список для поля підстановки. У нашому прикладі треба створити один стовпець і заповнити три рядки значеннями:

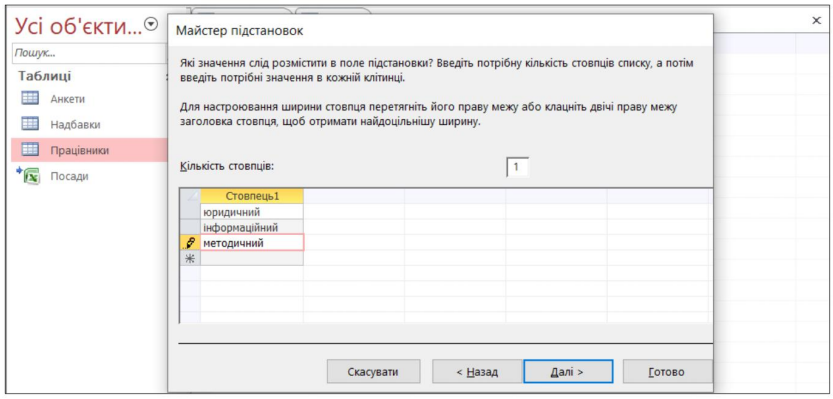


Рис. 2.6. Список для поля підстановки

Натиснути кнопку *Далі*.

4. Наступним кроком треба задати ім'я нового поля *Відділ* замість імені *Поле1*, пропонованого системою за замовчуванням, і натиснути кнопку *Готово*.
5. Заповнити клітинки поля *Відділ* даними, вибираючи їх зі списку (Рис. 2.7).

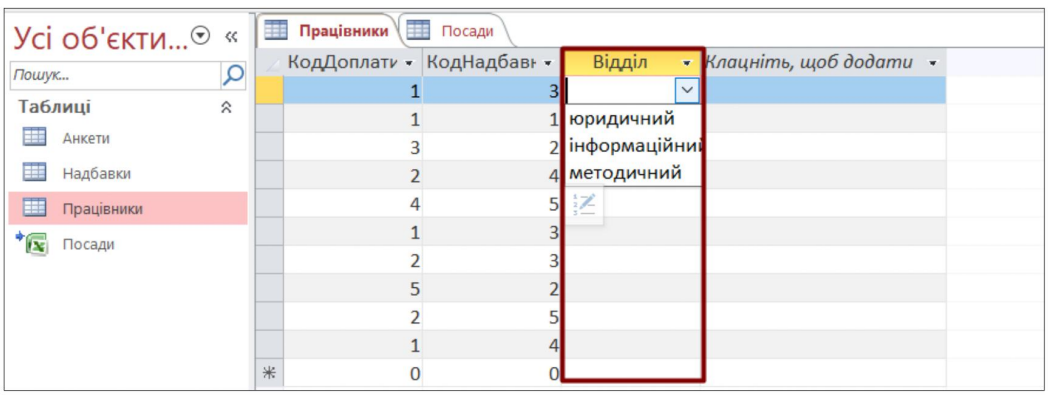


Рис. 2.7. Заповнення поля *Відділ* із запропонованого списку

б) у режимі *Конструктор*

Для створення списку з різновидами освіти працівників організації (вища, середня спеціальна, середня та ін.) для уже існуючого поля *Освіта* в таблиці *Анкети* треба відкрити таблицю *Анкети* у режимі *Конструктор*. Встановити курсор в поле *Освіта*, у властивостях поля перейти на вкладку *Підстановка* і вибрати для властивості *Відобразити елемент керування* значення *Поле зі списком*, а для властивості *Тип джерела рядків* – значення *Список значень*. Після цього у властивості *Джерело рядків* ввести такі значення (роздільник ";" – крапка з комою):

«середня»; «середня спеціальна»; «вища»; «науковий ступінь»

Після чого лишилось заповнити поле *Освіта* відповідними значеннями зі сформованого списку (Рис. 2.8).

Анкети

КодПрацівн	ДатаНарожд	Адреса	Освіта	Фото	МобілТелефон	СімейнийСт
1	18.11.2002	вул. Європейська, 8, кв. 50	вища	Package	(097)658-842-3	неодружений
2	28.06.2000	вул. Тищика, 22, кв. 17	середня	Package	(050)845-692-0	неодружений
3	06.05.1999	вул. Гонти, 17, кв. 1	базова середн	Package	(099)873-022-5	неодружений
4	06.07.2003	вул. Велика Фонтанна, 45, кв. 6	вища	Package	(091)745-803-6	неодружений
5	18.07.2003	вул. Стуса, 117, кв. 65	науковий ступ	Package	(073)569-842-5	неодружений
6	12.03.2000	вул. Садова, 30. кв. 25	вища	Package	(067)582-020-1	неодружений
7	05.05.2001	вул. Жовтнева, 10 кв. 118	вища	Package	(063)587-445-5	неодружений
8	09.09.2001	вул. Центральна, 184, кв. 243	вища	Package	(073)558-866-9	незаміжня
9	01.02.2001	вул. Б. Хмельницького, 21, кв. 36	середня	Package	(099)556-644-1	неодружений
10	27.10.1999	просп. Богуна, 16, кв. 3	науковий ступ	Package	(098)636-363-6	неодружений
11						

Рис. 2.8. Заповнення поле *Освіта*

в) Створення вкладеної таблиці даних – списку підстановки для існуючого поля з вибором значень списку з іншої таблиці (пов'язаного джерела даних)

Використання полів підстановки, використовуючи значення пов'язаних існуючих у БД таблиць, дозволяє: по-перше, не вводити, а просто вибирати значення, а, по-друге, не думати про помилкові значення, неіснуючі в таблиці, підтримуючи, тим самим, цілісність даних:

1. Відкрити таблицю *Працівники* у режимі *Конструктора* і встановити курсор у полі *Посада*.
2. У нижній частині вікна конструктора перейти на вкладку *Підстановка*. Для властивості *Відобразити елемент керування* вибрати значення *Список*.
3. Далі для властивостей вибрати зі списків такі значення:

– для властивості *Тип джерела рядків* – значення *Таблиця/Запит*;

– для властивості *Джерел рядків* – значення *Посади* (це ім'я таблиці).

4. Перейти до режиму таблиці і заповнити клітинки поля *Посада* даними, вибираючи їх зі списку.

Створене таким чином поле підстановки зі значеннями з пов'язаної таблиці БД дозволить користувачеві не піклуватися про актуалізацію і ведення таблиці, яка поповнюється в даному випадку з таблиці Excel (Рис. 2.9).

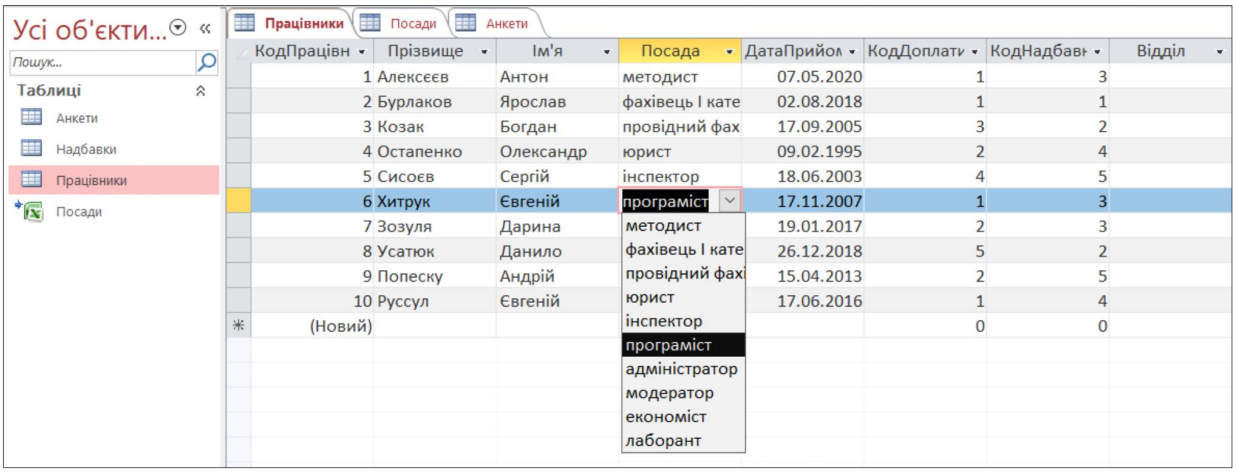


Рис. 2.9. Заповнення поля *Посада*

Як наслідок цих дій автоматично створиться зв'язок між таблицями *Посади* (головна) і *Працівники* (підпорядкована).

г) Створення у підпорядкованій таблиці поля підстановки для вибирання значень із головної таблиці за допомогою Майстра підстановок

Розглянемо порядок створення такого поля підстановки на прикладі поля *КодДоплати* у підпорядкованій таблиці *Працівники* з вибиранням значень із головної таблиці *Доплати*:

1. Відкрити підпорядковану таблицю *Працівники* у режимі конструктора, попередньо вилучивши зв'язок між таблицями *Працівники* і *Доплати*, якщо він був встановлений.
2. Виділити поле вторинного ключа *КодДоплати* і як тип даних для цього поля вибрати замість типу *Число* значення *Майстер підстановок*. Це призведе до відкривання діалогового вікна майстра створення поля підстановки.
3. На першому кроці створення поля підстановки за допомогою Майстра треба просто натиснути кнопку *Далі*, погодившись зі встановленою за замовчуванням опцією *поле підстановки має отримати значення з іншої таблиці або запиту*.

4. На другому кроці створення поля підстановки треба вибрати таблицю *Доплати* з первинним ключем (при цьому типи даних для первинного і вторинного ключів мають бути однаковими). Натиснути кнопку *Далі*.

5. На наступному кроці треба відібрати те поле або поля таблиці *Доплати*, що мають відображатися при користуванні полем підстановки у таблиці *Працівники*. Наприклад, можна вибрати поля *ВидДоплати* та *Коефіцієнт* або ж вказати тільки поле *ВидДоплати*. Натиснути кнопку *Далі*.

6. Далі можна (хоча і не обов'язково) вказати порядок сортування елементів списку, наприклад, по полю *КодДоплати*. Натиснути кнопку *Далі*.

7. Наступним кроком рекомендується за допомогою покажчика миші задати ширину стовпців у полі підстановки. Натиснути кнопку *Далі*.

8. На останньому кроці Майстра створення поля підстановки треба задати підпис поля і встановити опцію для перевірки цілісності даних, яка передбачає каскадне видалення. Натиснути кнопку *Готово*.

9. Залишилось погодитись з необхідністю зберігання таблиці, перейти до режиму *Таблиця* та впевнитись у зміненні значень поля *КодДоплати* на текстові види доплати, що є інформативнішим при роботі з таблицею.

10. За потреби заповнення або змінення значень у цьому полі у правому краю вибраної клітинки автоматично з'являтиметься стрілочка для відкривання розкривного списку з елементами списку (Рис. 2.10).

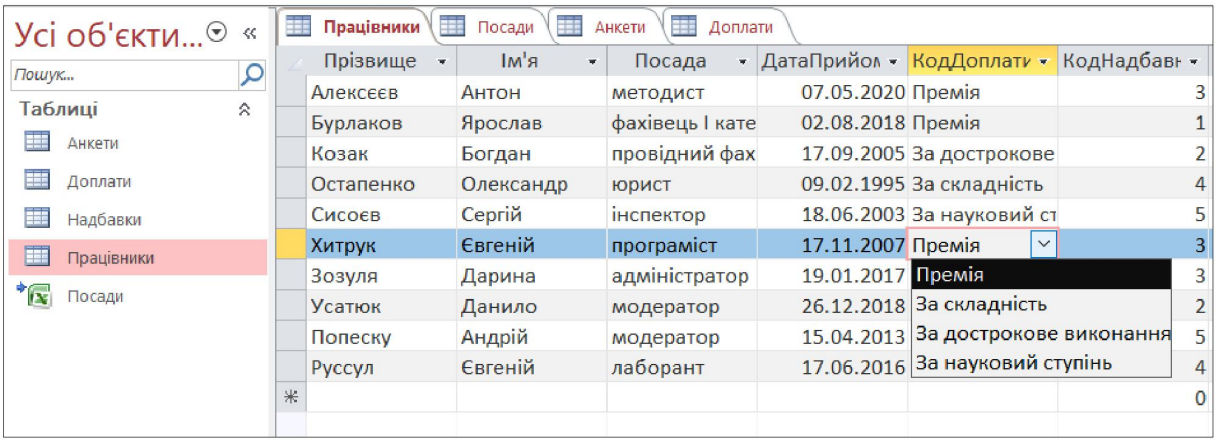


Рис. 2.10. Розкривний список заповнення або зміни значень

2.3. Створення зв'язків між таблицями

При створенні БД відомості розподіляються по таблицях, у кожній з яких є первинний ключ. Після цього до пов'язаних таблиць додаються зовнішні ключі, які посилаються на первинні ключі. Ці пари зовнішнього і первинного ключів формують основу для міжтабличних зв'язків і багатотабличних запитів. Тому важливо, щоб посилання *«зовнішній ключ – первинний ключ»* залишалися синхронізованими. Цілісність даних допомагає впевнитись, що посилання залишаються синхронізованими, і визначається міжтабличними зв'язками.

Access дає змогу організовувати і відображати міжтабличні зв'язки за допомогою вікна *Зв'язки* (схема даних), яке можна відкрити однойменною командою на вкладці *Знаряддя бази даних.*

Використовуються зв'язки між таблицями для коректного створення і подальшого використання інших об'єктів бази даних (форм, запитів і звітів), оскільки міжтабличні зв'язки є основою, за допомогою якої можна забезпечити цілісність даних.

Зв'язки встановлюються шляхом зв'язування ключового поля головної (батьківської) таблиці з відповідним їй полем підпорядкованої (дочірньої) таблиці. Часто ці поля в таблицях мають однакові імена, але в загальному випадку це не обов'язково.

Обов'язковими для створення зв'язків є такі вимоги:

1. В одній або в обох таблицях має бути поле, яке містить унікальні (без повторів) значення в усіх записах, це і є первинний ключ.
2. Зв'язувані поля повинні мати однакові типи даних, крім таких винятків:

– поле типу *Автонумерація* (*Код*) можна зв'язувати з числовим полем, якщо в числовому полі у властивості *Розмір поля* задано значення *Довге ціле*;

– поле типу *Автонумерація* можна зв'язувати з числовим полем, якщо для обох полів у властивості *Розмір поля* задано значення *Ідентифікатор реплікації*.

3. Зв'язувані поля числового типу повинні мати однакові значення властивості *Розмір поля*.
4. Для полів типу *Об'єкт OLE*, *Так/Ні* та *Довгий текст* первинний ключ призначити не можна.

Для відображення, коригування та вилучення зв'язків між таблицями в Access існує вікно *Зв'язки* (схема даних), яке відкривається командою *Знаряддя бази даних – Зв'язки* (Рис. 2.11).

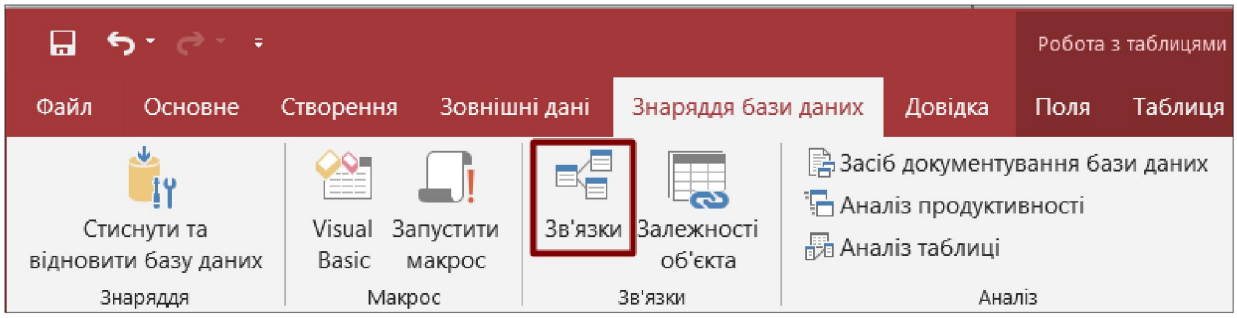


Рис. 2.11. Вікно *Зв'язки* (схема даних)

Вікно *Зв'язки* в Access є не тільки засобом графічного відображення логічної структури БД, воно активно використовується системою при роботі з БД.

Послідовність встановлення зв'язків між таблицями БД:

1. Закрити всі таблиці в БД і виконати команду *Знаряддя бази даних – Зв'язки*.

2. У переліку таблиць виділити одну з них, наприклад *Анкети*, і просто перетягнути її на бланк зв'язків мишею, а тоді так само перетягнути решту таблиць. Крім такого способу, на вкладці *Знаряддя для зв'язків / Конструктор* можна вибрати команду *Відобразити таблицю*, після чого відкриється вікно для додавання таблиць на бланк конструктора зв'язків (схему даних). Натискаючи на кнопку *Додати*, треба почергово винести на бланк конструктора всі таблиці БД, після чого натиснути кнопку *Закрити* (рис. 2.12).

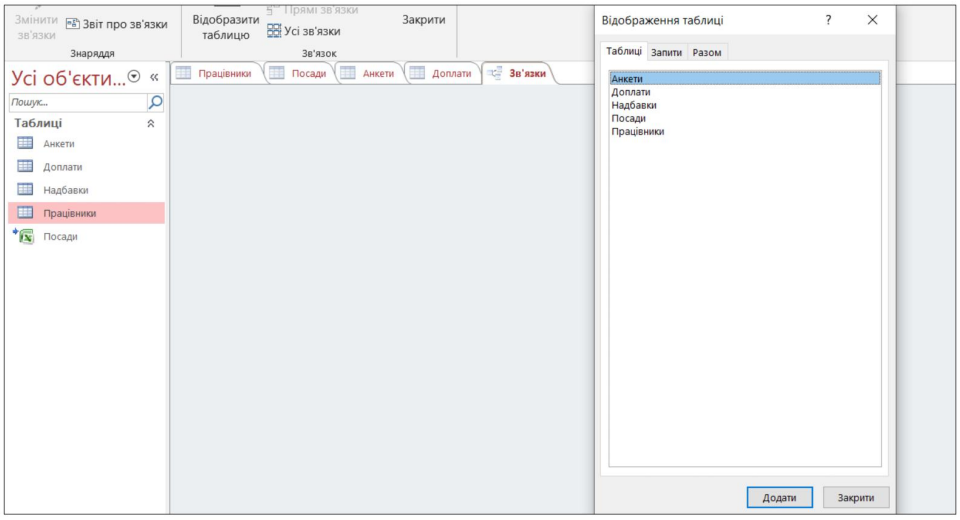


Рис. 2.12. Відображення таблиці

До речі, як видно з рис. 2.12, зв'язки можна встановлювати не лише з таблицями, а й із запитами, які на виході формують нові таблиці

(створення запитів розглядатиметься на подальших лабораторних роботах).

Отже, на полі *Зв'язки* з'являться п'ять таблиць БД, причому три з них – *Посади, Доплати* і *Працівники* – уже пов'язані засобами створених нами раніше у таблиці *Працівники* полів підстановок, і залишилось пов'язати решту таблиць.

3. Для зв'язування таблиць *Надбавки* (головна таблиця) і *Працівники* (підпорядкована таблиця) треба лівою кнопкою миші виділити в таблиці *Надбавки* поле *Код* і перетягнути мишею на поле

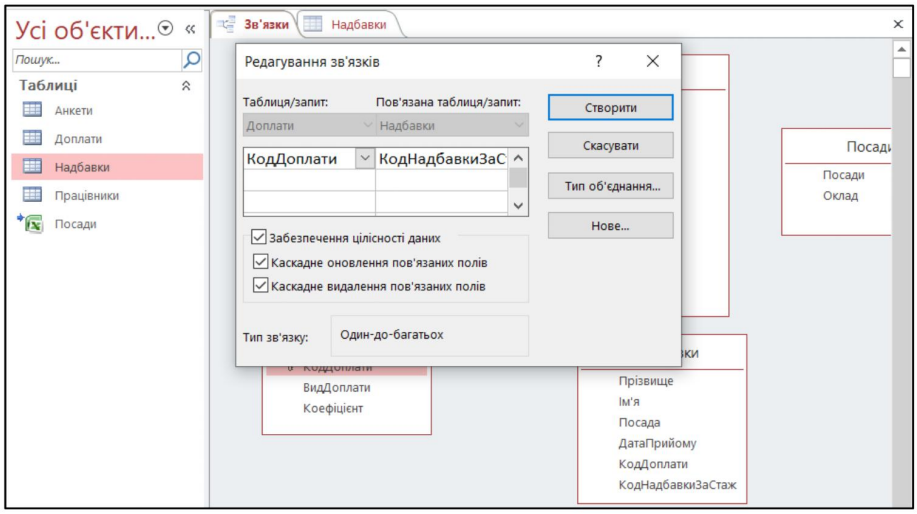


Рис. 2.13. Редагування зв'язків

КодНадбавкиЗаСтаж таблиці *Працівники*, після чого з'явиться вікно *Редагування зв'язків*, в якому обов'язково треба увімкнути опції *Забезпечення цілісності даних*, а також *Каскадне оновлення пов'язаних полів* і *Каскадне видалення пов'язаних полів*. Залишилось натиснути кнопку *Створити* (Рис. 2.13).

Після цього у вікні *Зв'язки* з'явиться лінія зв'язку, на одному кінці якої буде стояти одиниця, а на другому – знак: зв'язок «один-до-багатьох».

4. Для створення зв'язку між таблицями Працівники та Анкети треба виконати такі самі дії, проте, на відміну від попередніх, тут створиться зв'язок типу «один-до-одного». Після цього вікно Зв'язки для цього прикладу БД набуде вигляду:

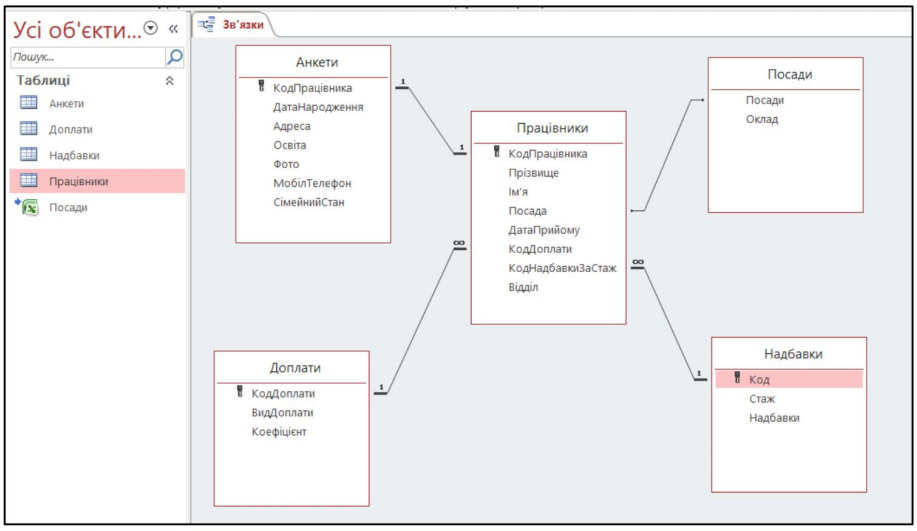


Рис. 2.14. Створений зв’язок між таблицями

Закрити вікно Зв’язки (схему даних) можна, натиснувши кнопку у правому верхньому куті. Якщо після цього відкрити будь-яку з пов’язаних таблиць, наприклад *Надбавки*, то тепер у ній з’являться маркери, клацання по яким розкриє рядки (записи) з підпорядкованої таблиці (в даному прикладі показані записи з відомостями про працівників, які мають стаж роботи від 3-х до 5-ти років) (Рис. 2.15).

Відтак, яскраво продемонстровано створений зв’язок типу «один-до багатьох»: кожен рядок у таблиці *Надбавки* є унікальним, у той час як у таблиці працівники декілька записів містять однакові значення з кодом стажу роботи працівника.

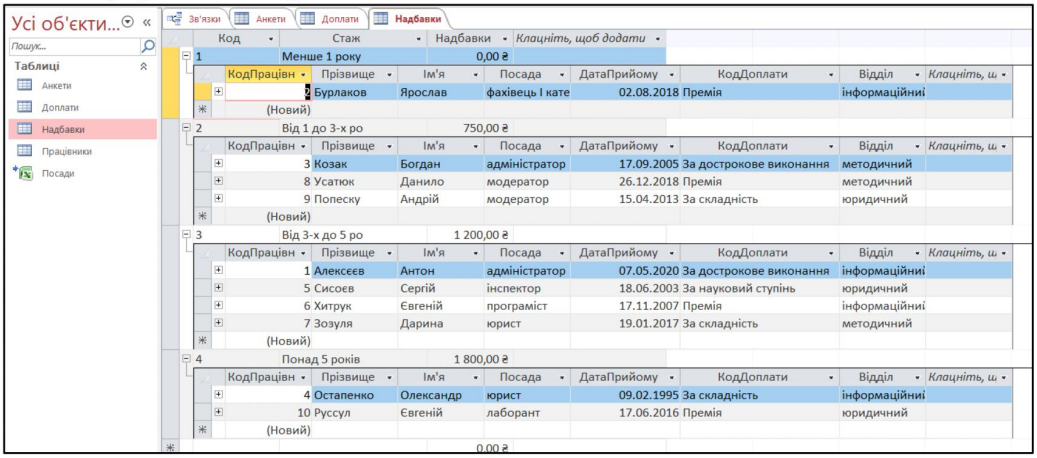


Рис. 2.15. Відкриття пов’язаних таблиць

Нерідко розробнику бази даних доводиться змінювати імена ключових полів, змінювати або вилучати зв’язки. Для цього або просто для того, щоб побачити існуючі зв’язки між таблицями, достатньо закрити всі таблиці, активізувати вкладку *Знаряддя бази даних* та вибрати команду *Зв’язки*.

Лекція 3. СОРТУВАННЯ ТА ФІЛЬТРУВАННЯ ДАНИХ

3.1. Сортування даних

Сортування даних відіграє важливу роль у створенні ефективних і простих у використанні звітів і форм. Наприклад, каталог товарів, в якому товари вказано в алфавітному порядку або впорядковано за ціною, зручніше використовувати, ніж каталог, який містить елементи в довільному порядку.

Деякі списки, з якими зазвичай працюють користувачі, уже впорядковано, або їх можна впорядкувати у потрібний спосіб. Календар або планувальник відсортований за датою. З іншого боку, під час пошуку авіаквитків або повідомлень у папці *Вхідні* можна виконати сортування за одним із багатьох доступних полів. Квитки можна сортувати за ціною, часом відправлення або авіакомпанією, а повідомлення в поштовій скриньці – за відправником, темою або датою. Упорядкований список дозволяє користувачам переглядати та знаходити потрібні дані, не переглядаючи всі дані.

Записи в таблиці, запиті, формі або звіті можна сортувати за одним або кількома полями. Сортувати можна за будь-яким відображуваним полем, окрім полів, які містять вкладення або об’єкти OLE. Вибір способу сортування записів у звіті або формі не потребує виконання складних дій зі структурою.

Сортувати записи можна як під час створення таблиці, запиту, форми або звіту, так і під час їх переглядання. Порядок, вибраний під час створення об’єкта, установлюється порядком сортування даних об’єкта за замовчуванням. Під час переглядання запиту або звіту кожен користувач може сортувати записи відповідно до власних потреб, наприклад за ціною або постачальником.

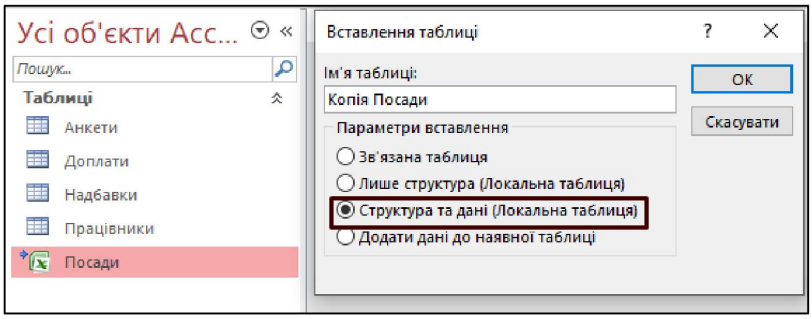


Рис. 3.1. Вставлення таблиці *Посади* як локальної

Як приклад, скопіюємо таблицю *Посади* і вставимо її копію як локальну таблицю, зберігши структуру та дані (слід на її назві клацнути правою кнопкою миші та вибрати з контекстного меню команду *Копіювати*, після чого виконати команду контекстного меню *Вставити*; система запропонує задати ім'я цієї таблиці *Копія Посади*), після чого задамо порядок сортування за зростанням окладів (Рис. 3.1). Для цього слід клацнути в будь-якому місці у полі *Оклад* і вибрати на вкладці *Основне* у групі *Сортування й фільтр* команду *За зростанням* (Рис. 3.2).

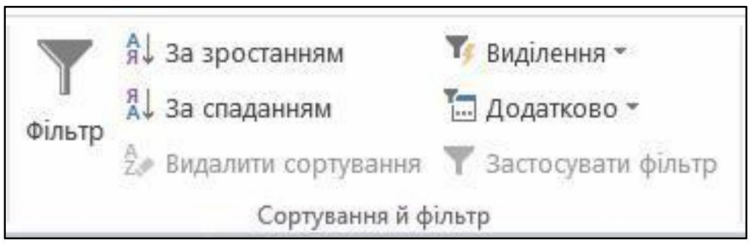


Рис. 3.2. Сортування й фільтр

Подібну за дією команду можна вибрати і з контекстного меню, проте назва команди буде дещо іншою і залежатиме від типу даних сортованого поля (рис. 3.3).



Рис. 3.3. Види сортування полів з різними типами

Види сортування полів з різними типами:
1 – типи даних: *Число*, *Грошова одиниця*, *Автонумерація*;
2 – типи даних: *Текст*, *Примітка*, *Гіперпосилання*;
3 – тип даних *Так/Ні*;
4 – тип даних *Дата й час*

Такі самі параметри сортування можна задати, якщо натиснути на стрілці праворуч від імені поля:

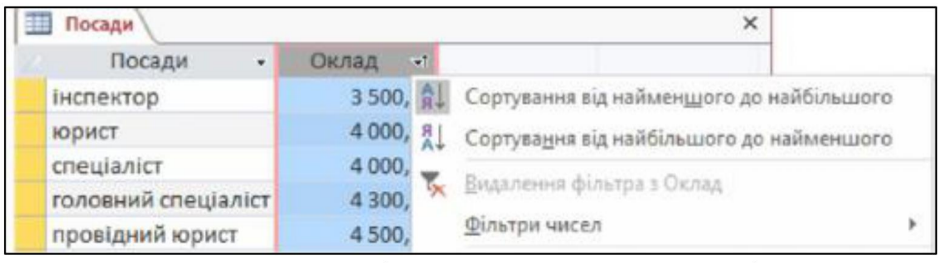


Рис. 3.4. Варіанти сортування полів

Під час сортування слід пам’ятати, що порядок сортування чисел, тексту та спеціальних символів залежить від установлених на комп’ютері мови й регіональних параметрів, указаних у діалоговому вікні *Параметри Access*.

Щоб скасувати порядок сортування таблиці, запиту або форми, треба на вкладці *Основне* у групі *Сортування й фільтр* натиснути кнопку *Видалити сортування*. Це вилучить порядок сортування в усіх полях подання.

Сортувати можна відразу за кількома полями, наприклад, сортувати таблицю *Анкети* за полями *Освіта* (первинне сортування) та *ДатаНародження* (вторинне сортування). Особливістю такого сортування є те, що спочатку треба задати параметри вторинного сортування (поле *ДатаНародження* – тип сортування *від найстаршого до наймолодшого*), а вже тоді – параметри первинного сортування (поле *Освіта* – тип сортування *від А до Я*). Тобто записи буде впорядковано полем *Освіта*, а в разі однакових значень цього поля записи виводитимуться за спаданням віку працівників.

3.2. Фільтрування даних

Фільтрування (фільтрація) – це один зі способів відбирання даних за певним критерієм, що є базовою операцією з базою даних (Рис. 3.5).

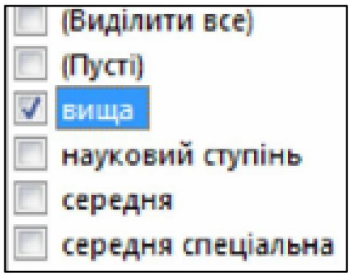


Рис. 3.5. Відбирання даних за певним критерієм

Існує багато типів фільтрів: деякі з них можна застосовувати лише до одного типу даних, а інші підтримують кілька типів даних. Використання того чи іншого фільтра залежить від типу та значень поля, яке потрібно відфільтрувати. Існує кілька способів фільтрування: використання загальних фільтрів, фільтрування за вибраним та розширене фільтрування.

Загальні фільтри найбільш поширені, оскільки забезпечують просте фільтрування для більшості типів даних. Наприклад, для фільтрування даних *Дата й час* можна використовувати загальний фільтр *Перед*, щоб виключити дати, раніші або пізніші за потрібну дату. Загальні фільтри одночасно працюють тільки в одному полі. Щоб застосувати фільтр до кількох полів або елементів керування, треба до кожного поля чи елемента керування застосовувати фільтр окремо або скористатися додатковим

параметром фільтра.

Щоб задати такий фільтр для певного поля, слід клацнути на заголовок цього поля та встановити критерії відбору. Наприклад, щоб у таблиці *Копія Анкети* відібрати працівників лише з вищою освітою, слід клацнути у заголовку поля *Освіта* та вибрати значення «*вища»* як критерій фільтрації.

Щоб задати у цій таблиці інший фільтр, наприклад у полі *ДатаНародження* (працівників, що святкуватимуть день народження наприкінці року у четвертому кварталі), слід клацнути по заголовку цього поля і вибрати *Фільтри дат – Усі дати за період – 4-й квартал*, після чого натиснути кнопку *ОК* (Рис. 3.6).

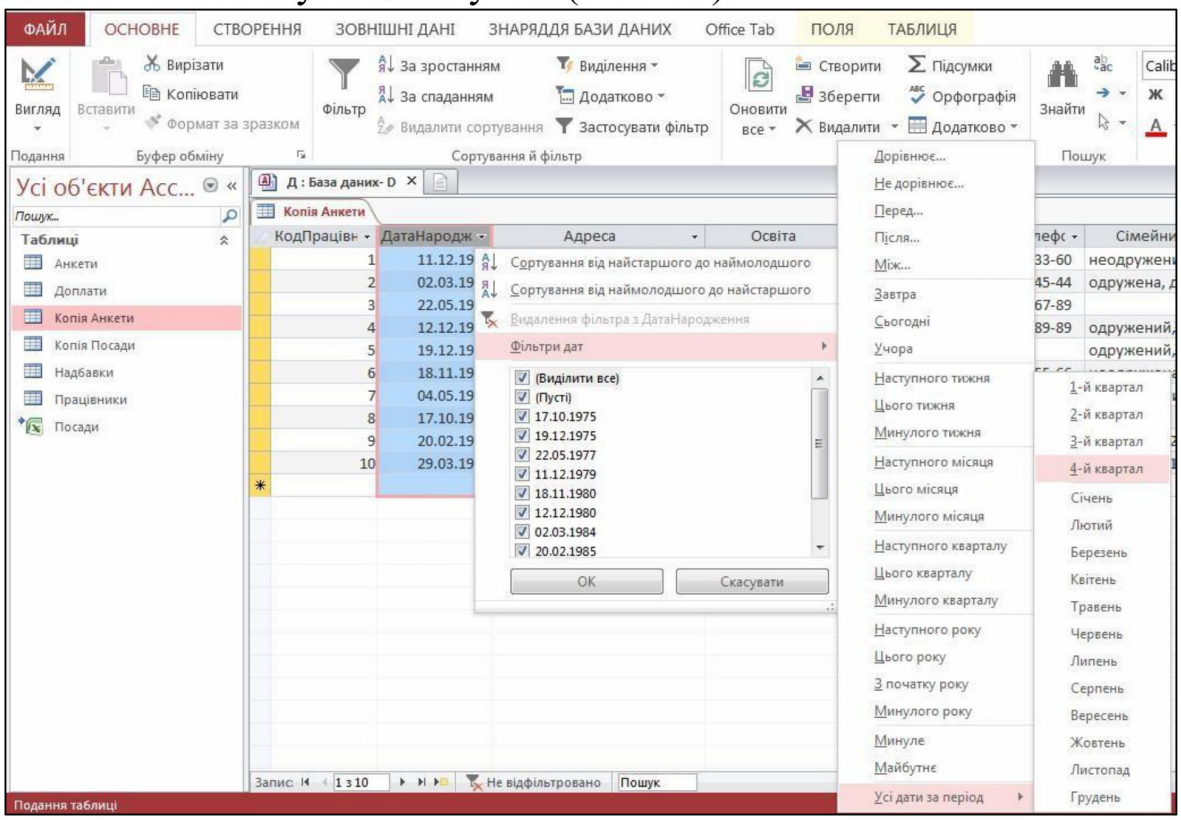


Рис. 3.6. Фільтр працівників, що святкуватимуть день народження наприкінці року в четвертому кварталі

Слід зазначити, що загальні фільтри недоступні для полів типів *Так/Ні*, *Об'єкт OLE* та *Вкладення*, а список значень недоступний для полів типу *Memo*. Окрім того, загальні фільтри доступні лише в поданні таблиці, звіту або макета.

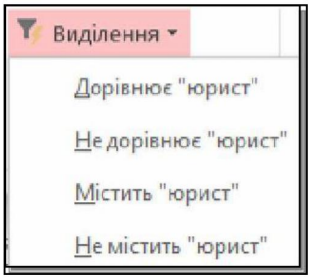


Рис. 3.7. Кнопка *Виділення*

Фільтрування за виділенням доцільно застосувати, якщо потрібно використати певне

значення як фільтр. Щоб застосувати фільтр на основі вибраного зараз значення, слід вибрати на стрічці вкладку *Основне* й у групі *Сортування й фільтр* натиснути кнопку *Виділення* (Рис. 3.7). У розкривному списку відобразяться доступні варіанти фільтрування. Як і загальні фільтри, варіанти фільтра за виділенням залежать від типу даних. Також можна клацнути правою кнопкою миші вибране значення, щоб фільтрувати за виділенням.

Розширений фільтр має більше можливостей для створення умов фільтрації, ніж в інших способах фільтрування, і застосовується для створення різноманітних складних критеріїв фільтрації. Процес створення такого фільтра подібний до створення запиту. Щоб задати розширений фільтр, слід скористатись кнопкою *Додатково* на вкладці *Основне* у групі *Сортування й фільтр*. Застосовувати таку фільтрацію можна як до даних таблиць, так і до запитів, форм або звітів у вікні табличного подання даних, поданні форми, звіту або розмітки.

Наприклад, щоб за допомогою фільтрації відібрати працівників, які у цьому році святкуватимуть ювілей (вік кратний 5), не обійтися без розширеного фільтра. Проте до його створення доцільно переконатися, що до даних не використовуються якісь інші фільтри, вилучивши їх командою *Додатково – Очистити всі фільтри*. На вкладці *Основне* стрічки у групі *Сортування й фільтр* натиснути кнопку *Додатково* та вибрати пункт *Розширений фільтр – Сортування*. При цьому відкриється нова вкладка для створення шаблона умови на кшталт вікна конструювання запиту. У цьому вікні вибрати поле *ДатаНародження* у рядку *Поле*, а в рядку *Критерії* клацнути правою кнопкою миші та вибрати команду контекстного меню *Побудувати*. У діалоговому вікні *Побудовник виразів* створити вираз:

(*Date()* - [*Копія Анкети*]![*ДатаНародження*]) / 365,25 *Mod* 5 = 0

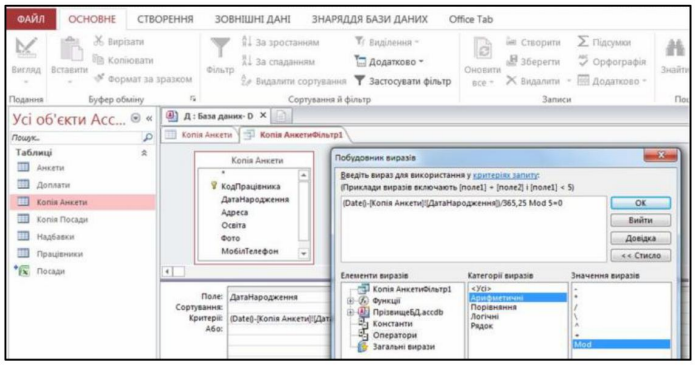


Рис. 3.8. Створення виразу за допомогою *Побудовника виразів*

При цьому назви функції *Date()* арифметичного оператора *Mod* (залишок від цілочислового ділення), а також назву поля *ДатаНародження* доцільно вибирати за допомогою інструментарію *Елементи виразів* (Рис. 3.8).

Після цього доцільно натиснути кнопку *ОК* та виконати команду *Застосувати фільтр* на вкладці *Основне* у групі *Сортування й фільтр*.

Рядок *Або* в конструкторі фільтра застосовується як альтернатива умовам у рядку *Критерії*.

При створенні розширеного фільтра стає доступною команда *Додатково – Зберегти як запит*, яка дає змогу зберігати налаштування фільтра як новий запит (Рис. 3.9).

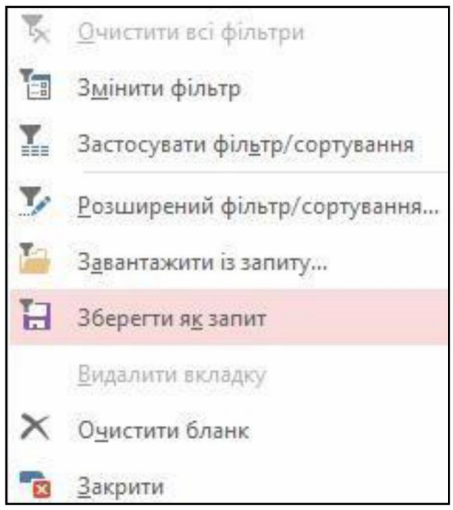


Рис. 3.9. Команда *Додатково – Зберегти як запит*

Фільтр можна очистити, вилучивши його командою *Додатково – Очистити всі фільтри*.

Незважаючи на простоту і потужність інструментарію створення фільтрів, здебільшого для відбирання даних за критерієм використовують запити, оскільки запити можна згодом використовувати для створення форм, звітів та інших запитів (усіх об’єктів, для яких потрібні джерела даних).

3.3. Правила формування критеріїв відбору

Критерій фільтрації за суттю є умовою відбору, яку можна задавати в одному або в декількох полях. Умова може бути як простою, так і складним виразом, як набраним з клавіатури, так і сконструйованим за допомогою конструктора виразів, що відкривається командою *Побудувати* з контекстного меню.

У критерії відбору можуть використовуватись:

– *оператори порівняння*: >, <, >=, <=, =, <> (не дорівнює), *BETWEEN* (діапазон), *IN* (множина значень), *LIKE* (шаблон подібності, в якому можуть використовуватись в якості маски * та ?);

– *логічні оператори*: *AND*, *OR*, *NOT*;

– *арифметичні оператори*: +, –, *, /, \ (ділення націло), ^ (піднесення до степеню) і *MOD* (остача від цілочислового ділення);

– *рядковий оператор* конкатенації (злиття) рядків – *&*;

– *константи*: *Істина*, *Хибність*, "" (порожній рядок), *NULL* (незаповнена клітинка);

– *вбудовані функції* Access, наприклад, функція *Date()* повертає поточну системну дату.

При формуванні виразів імена полів записуються у квадратних дужках [].

У наведеній нижче таблиці подані деякі приклади умов, а також результати їх застосування.

Таблиця 3.1.

Умова	**Результат**
> 234	Повертає всі числа більші 234
>= "Одеса"	Повертає всі записи від "Одеса" і до кінця алфавіту
#02.02.2014#	Находить всі записи за 2 лютого 2014 р.
< *Date*() – 30	Повертає записи, від дати яких минуло понад 30 днів. Порівняння здійснюється з поточною системною датою
BETWEEN #21.12.2014# *AND* #12.01.2015#	Повертає дати між 21.12.2014 та 12.01.2015
NOT "Італія"	Находить всі записи, в яких вміст поля не відповідає значенню "Італія", тобто будуть найдені записи, які містять будь-який текст, крім "Італія", наприклад, "Чехія" або "Європа (Італія)"
NOT "*т"	Находить записи, які не закінчуються літерою "т"
IN("Канада"; "США"; "Китай")	Находить записи, які містять одне зі значень, зазначениху списку
LIKE "Т*"	Находить всі записи, які починаються з літери "Т"
LIKE "*ар*"	Находить записи, які містять сполучення літер "ар"

LIKE "[А-Г]*"	Находить у полі з типом даних *Короткий текст* записи, які починаються з літер "А-Г"
LIKE "Григорій Чорн??"	Находить записи, які починаються з імені "Григорій" і з прізвищем з 6-ти літер, з яких перші 4 літери невідомі, а останні дві літери – ні (на це вказують знаки питання). Наприклад, прізвища "Чорний" або "Чорняк"
BETWEEN Date() *AND DateAdd* ("m"; 3; *Date*( ))	Повертає всі записи з датою, яка знаходиться між поточною датою і датою, яка на 3 місяці пізніше поточної
IS NOT NULL	Повертає записи, які містять непорожнє (заповнене) значення
""	Повертає записи, які містять рядок (тип *Короткий текст*) нульової довжини. Такі рядки звичайно використовуються, коли поле обов'язкове для заповнення, але значення ще невідоме
"" *OR IS NULL*	Повертає записи, які містять порожні (незаповнені) значення
LEN([*Країна*]) >10	Повертає записи із назвами країн певної довжини – понад10 символів

Лекція 4. СТВОРЕННЯ ПРОСТИХ ЗАПИТІВ

4.1. Способи формування запитів

Однією з основних переваг реляційних БД – швидкий пошук даних в них, найбільш потужним засобом для цього є запити. Вони дозволяють вибирати дані з однієї чи декількох взаємопов'язаних таблиць БД, здійснювати над ними обчислення та здобувати результати у вигляді таблиць.

В Access існують різноманітні **способи формування запитів** (Рис. 4.1):

1) у режимі *Макет запиту*,
2) за допомогою *Майстра запитів,*
3) формування запитів мовою *SQL*.

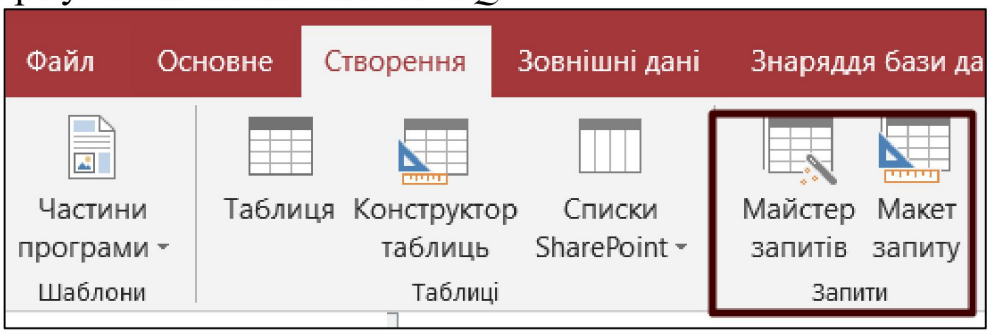


Рис. 4.1. Способи формування запитів

Для вибору режиму формування запитів *Майстер* або *Макет* в Access треба на вкладці *Створення* вибрати відповідну команду у групі *Запити*.

Вже існуючий запит в Access можна відкрити одним із режимів подання запитів: *Подання таблиці*, *Режим SQL*, *Конструктор* (Рис. 4.2).

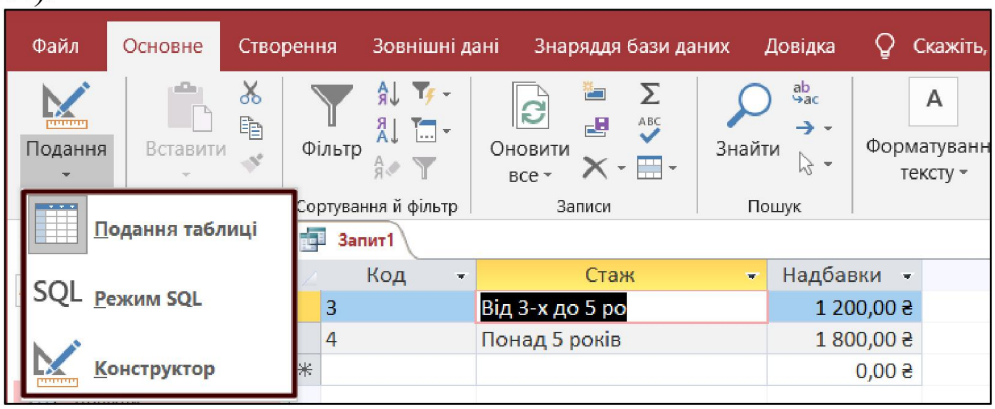


Рис. 4.2. Режими подання запитів

Перейти з режиму *Конструктор* до іншого режиму відкривання

можна скориставшись на панелі *Конструктор* у групі *Результати* командою *Вигляд*.

4.2. Створення простого запиту у режимі *Макет запиту*

Макет запиту є простим і наочним засобом створення досить складних запитів. Бланк запиту в цьому режимі складається з двох областей. У верхній відображається структура таблиць, до яких запит адресований, а нижня область розбита на стовпці – по одному стовпцю на кожне поле майбутньої результуючої таблиці (рис. 4.3).

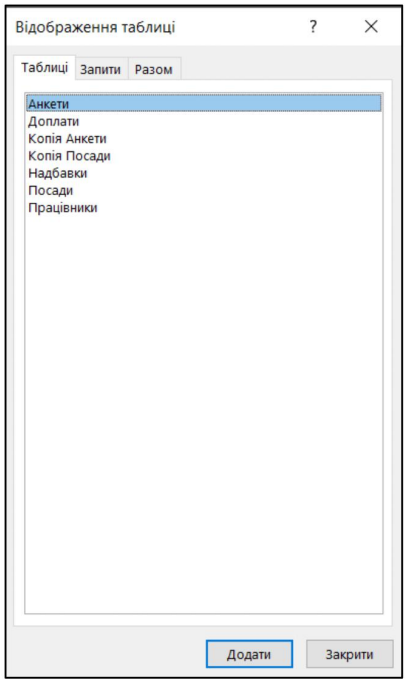


Рис. 4.3. Макет запиту

Для винесення однієї або декількох таблиць на бланк запиту слід скористатися командою контекстного меню *Відображення таблиці* і за допомогою кнопки *Додати* вибрати необхідні для створення запиту таблиці, після чого натиснути кнопку *Закрити.*

Для кожного відібраного в запиті поля у конструкторі треба обов'язково вказати ім'я таблиці та ім'я вибраного з неї поля (перші два рядки у нижній частині бланка запиту).

Винести на бланк запиту поля можна або за допомогою подвійного клацання по іменах полів, або перетягуванням імен полів, або вибиранням зі списку імен, розкривши його за допомогою. Причому, для обчислюваних полів ім'я таблиці не вказується. А для швидкого винесення всіх полів таблиці, треба двічі клацнути зірочку (*) у верхній частині списку полів таблиці.

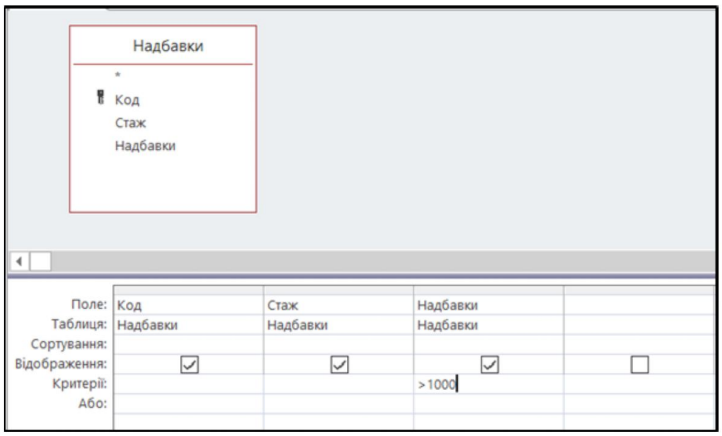


Рис. 4.4. Вікно *Макет запиту* з прикладом заповнення бланка

Необов'язковими для заповнення, але доцільними для конструювання запитів, є параметри (рядки у нижній частині конструктора) *Сортування*, *Відображення* та *Критерії* (Рис. 4.4).

Параметр *Сортування* дозволяє впорядковувати для зручності відібрані у запиті дані, наприклад, в алфавітному порядку прізвищ співробітників або в порядку зменшення значень їхніх окладів.

Параметр *Відображення* на екран містить прапорці, які відповідають за виведення на екран полів, включених до запиту. За замовчуванням усі прапорці увімкнені, тобто всі поля виводитимуться на екран після запуску запиту на виконання. Якщо вимкнути прапорець, тоді дані цього поля будуть оброблятися у запиті, але на екран виводитися не будуть.

Параметр *Критерії* дозволяє сформувати умову відбирання даних. Правила формування критеріїв розглядались на попередній лабораторній роботі. Наприклад, для відбору всіх даних таблиці *Надбавки* понад 1000 грн. слід задати критерій: >1000 (Рис. 4.4).

Запустити запит на виконання можна командою *Запуск*, яка міститься на вкладці *Конструктор* у групі *Результати*. Результат виконання запиту буде виведено в режимі *Таблиця* (Рис. 4.5).

При створенні запиту в режимі *Макет* ім'я запиту зазвичай задають наприкінці його створення, натиснувши у правому верхньому куті. При цьому з'явиться повідомлення з пропозицією зберегти зміни до структури запиту, в якому натиснути кнопку Так, потім у діалоговому вікні задати ім'я запиту. Після цього в області об'єктів БД у розділі *Запити* з'явиться новий запис *НадбавкиПонад1000*, подвійне клацання по якому відкриє запит у режимі *Таблиця* з даними, відібраними відповідно до умови (Рис. 4.5).

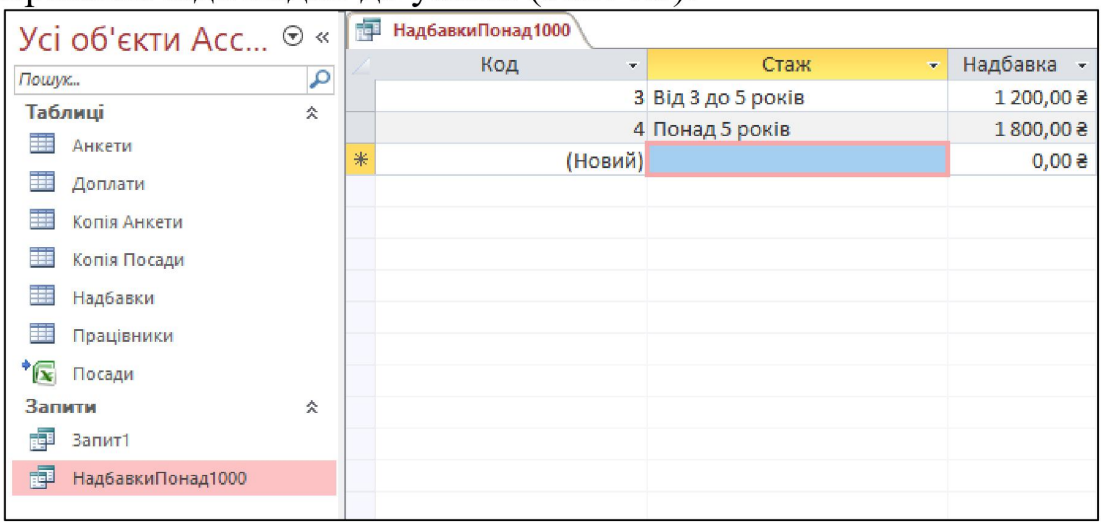


Рис. 4.5. Результат виконання запиту з відбором даних

Критерії відбору можна організовувати різними способами. Наприклад, відібрати ті дані з текстового поля *Посада*, які збігаються з двома певними значеннями («юрист» або «провідний юрист»), можна кількома способами:

– використовуючи критерій "юрист" *OR* "провідний юрист";
– використовуючи критерій *IN* ("юрист"; "провідний юрист");
– використовуючи критерій *LIKE* ("*юрист");
– використовуючи параметр *АБО* у вікні *Конструктора запитів*.

Критерії відбору можуть бути доволі складними. Наприклад, відібрати дані з певного діапазону дат – дата прийому була здійснена в межах 10-ти останніх років можна за допомогою такого критерію:

BETWEEN #01.01.2010# *AND* #31.12.2020#

Але більш «працездатною» буде умова, яка не буде «прив'язаною» до конкретної дати, і буде працювати в подальшому, оскільки буде опрацьовувати поточну системну дату (функція *Date*()) (Рис. 4.6, рис. 4.7):

Поле:	ДатаПрийому	Посада	Прізвище	Відділ
Таблиця:	Працівники	Працівники	Працівники	Працівники
Сортування:				
Відображення:	☑	☑	☑	☑
Критерії:	Between Date() And DateAdd("yyyy";-10;Date())	"юрист" Or "провідний		
Або:				

Рис. 4.6. Умова опрацьовування поточної системної дати

BETWEEN Date() *AND DateAdd*("yyyy"; -10; *Date*())

або

BETWEEN Date() *AND Date*() - 365,25*10

ЮристиПрацюютьДо10років

ДатаПрийому	Посада	Прізвище	Відділ
19.01.2015	провідний юрист	Зозуля	юридичний
15.04.2013	юрист	Попеску	інформаційний

Рис. 4.7. Таблиця відбору за певним критерієм

Якщо відкрити цей запит у режимі SQL командою *Подання – Режим SQL*, запит набуде вигляду (Рис. 4.8):

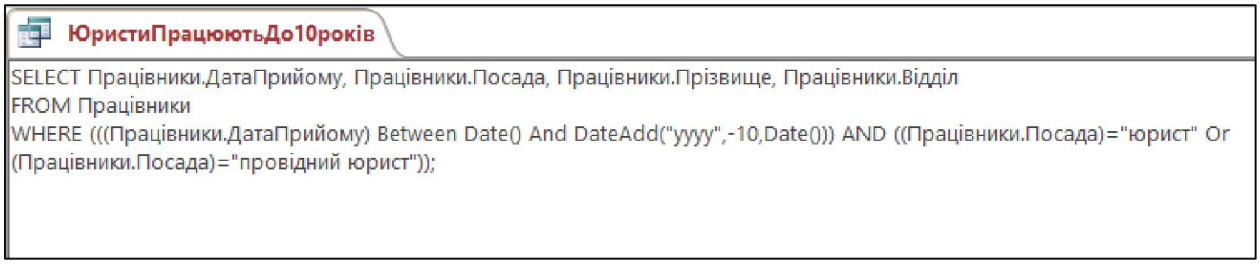


Рис. 4.8. запит у режимі SQL

4.3. Режим SQL

У **режимі SQL** можна не тільки переглядати вже створені запити, але й створювати нові «з нуля». Для створення такого запиту треба виконати команду *Створення – Макет запитів*, відмовитись від пропозиції додавання таблиць до бланка конструктора запиту, натиснувши кнопку *Закрити*, і перейти до режиму SQL командою *Подання – Режим SQL*. У вікні конструювання запиту дописати після слова *SELECT* весь текст запиту. Наприклад, для визначення середнього окладу всіх працівників запит може бути таким:

SELECT Avg (*Оклад*) *AS СереднійОклад FROM Посади*;

SQL-запит для визначення кількості працівників у кожному з відділів:

SELECT Відділ, *Count* (*КодПрацівника*) *AS КількістьПрацівниківУВідділі FROM Працівники GROUP BY Відділ*;

У цьому SQL-запиті для визначення кількості працівників на кожній посаді використовується агрегатна функція *Count*() та групування *GROUP BY* (Рис. 4.9).

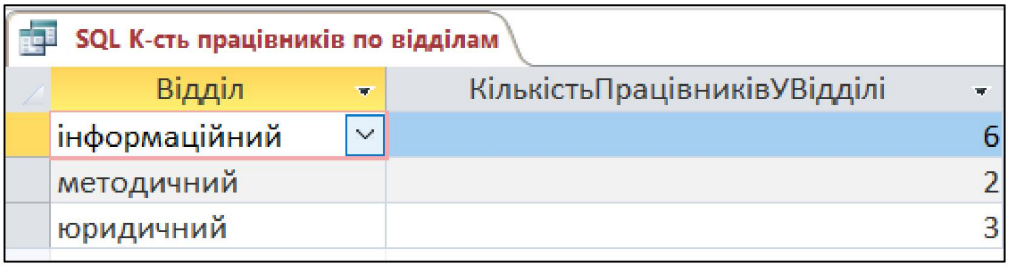


Рис. 4.9. SQL-запит для визначення кількості працівників на кожній посаді

SQL-запит для відбирання з двох взаємопов'язаних таблиць прізвищ та окладів працівників, які пропрацювали на підприємстві

менше 10 років (Рис. 4.10):

SELECT Прізвище, ДатаПрийому, Оклад FROM Працівники, Посади
WHERE Посади.Посади = Працівники.Посада
AND Abs(Date() – ДатаПрийому)/365.25 < 10;

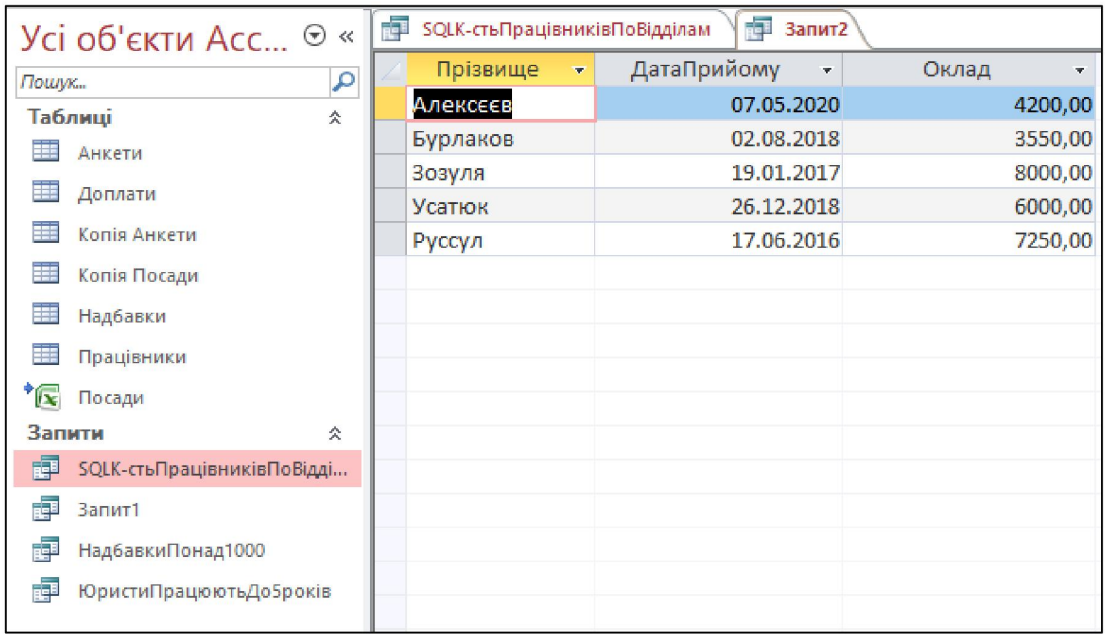


Рис. 4.10. SQL-запит для відбирання з двох взаємопов'язаних таблиць прізвищ та окладів працівників, які пропрацювали на підприємстві менше 10 років

4.4. Створення простого запиту за допомогою *Майстра запитів*

Майстер запитів дозволяє послідовно формувати бланк запиту, використовуючи готові шаблони. Окрім того, з'являється можливість створювати вирази без використання побудовника, що дуже зручно. При відкриванні *Майстра запитів* відкриється вікно *Новий запит*, в якому можна вибрати один з чотирьох варіантів створення запитів: *Майстер простих запитів*, *Майстер перехресних запитів*, *Майстер пошуку повторюваних запитів* та *Майстер пошуку незв'язаних записів*.

Розглянемо специфіку роботи з *Майстром* при створенні простого запиту на відбір даних зі значеннями сумарних виплат окладів по кожному відділу і визначенням кількості працівників у кожному з них. Для створення такого запиту за допомогою *Майстра* треба виконати такі дії:

1) На вкладці *Створення* у групі *Запити* виконати команду *Майстер запитів*, вибрати тип *Майстер простих запитів* і натиснути кнопку *ОК*.

2) У діалоговому вікні *Майстер простих запитів* вибрати таблицю *Працівники*, з якої за допомогою кнопки-стрілки > перенести у праве вікно поле *Відділ*. Після цього вибрати таблицю *Посади*, з якої перенести у праве вікно два поля *Посади* та *Оклад*. Натиснути кнопку *Далі* (Рис. 4.11).

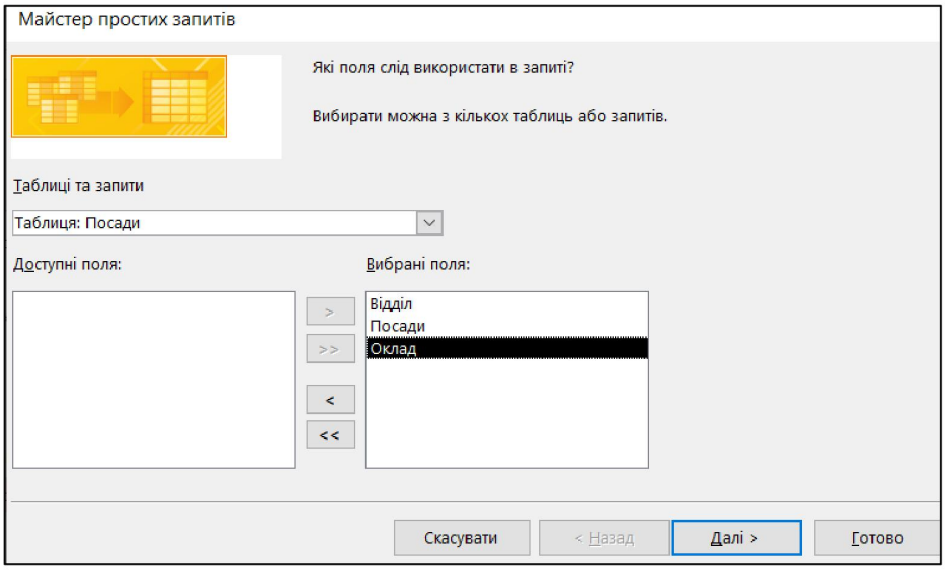


Рис. 4.11. Вікно *Майстер простих запитів*

3) На наступному кроці треба встановити опцію *Зведення* і натиснути кнопку *Параметри зведення*.

4) У вікні *Параметри зведення* треба поставити позначки напроти *Sum* та у полі *Кількість записів у Працівники*, після чого натиснути кнопку *ОК* (рис. 4.12).

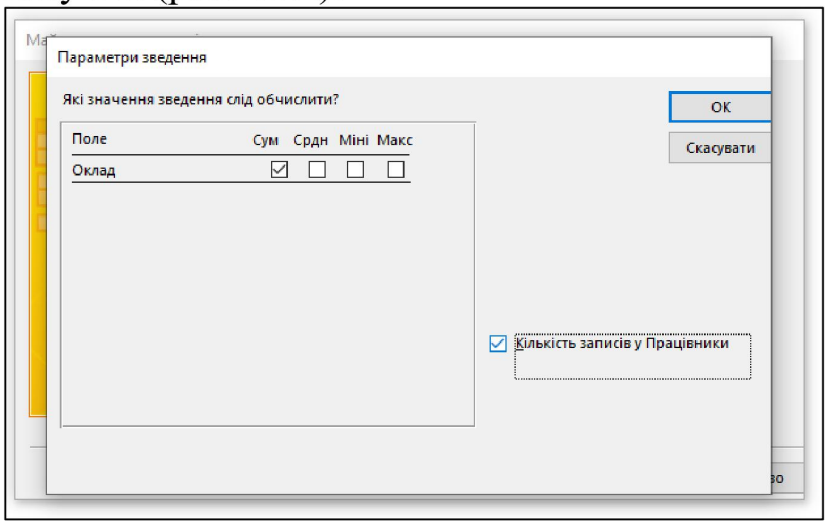


Рис. 4.12. Вікно *Параметри зведення*

5) Решта кроків особливих коментарів не потребують. Ім’я цього запиту – *Сумарні виплати по відділах* (Рис. 4.13).

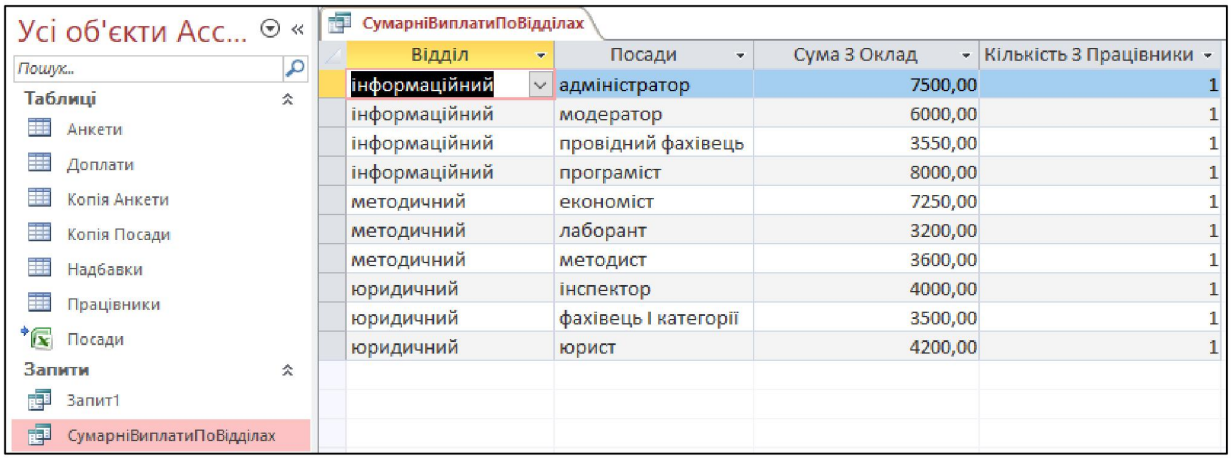


Рис. 4.13. Сумарні виплати по відділах

Як наслідок виконання цього запиту буде сформована таблиця, в якій поле *Сума З Оклад* матиме сумарні значення окладів усіх працівників на тій чи іншій посаді, а поле *Сума З Працівники* – сумарну кількість *Конструктор*, можна побачити цей запит у такому вигляді (Рис. 4.14).

При бажанні можна перейти до режиму SQL і побачити цей запит у вигляді оператора *SELECT* (Рис. 4.15).

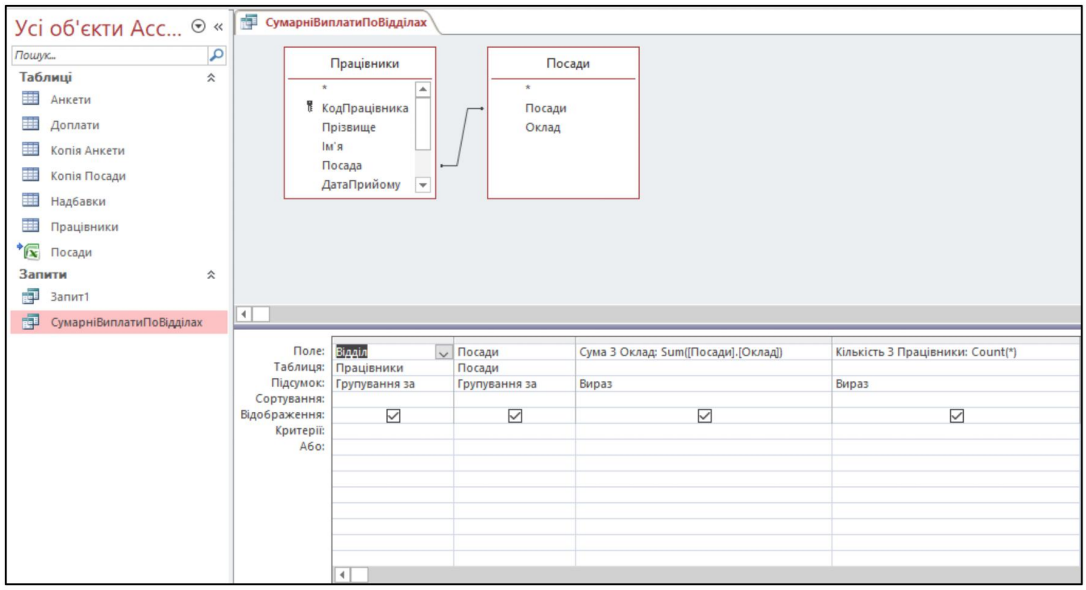


Рис. 4.14. Сформована таблиця *Сумарних виплат по відділах*



Рис. 4.15. Запит у вигляді оператора *SELECT*

4.5. Створення обчислюваних полів

Звичайно таблиці БД не використовуються для зберігання обчислюваних значень, оскільки це може суперечити іншим збереженими даними і тим самим порушувати цілісність даних. Наприклад, не має сенсу створювати поле *Вік* у таблиці, оскільки доведеться оновлювати це значення кожен рік; замість цього можна зберігати дату народження, а потім використовувати запит для обчислення віку. Також при створенні БД *Співробітники організації* треба враховувати те, що надбавка за стаж має періодично або автоматично коригуватися при збільшенні робочого стажу працівників.

Обчислювані поля організовують за потребою у запитах. Наприклад, у БД *Співробітники організації*, щоб дізнатися сумарне значення щомісячної зарплати кожного працівника, треба у запиті вибрати чи сформувати такі поля:

1) вибрати поле *Прізвище* з таблиці *Працівники*;
2) вибрати поле *Оклад* з таблиці *Посади*;
3) вибрати поле *Надбавка* за стаж з таблиці *Надбавки*;
4) обчислити значення доплати за індивідуальним кодом доплати (поле *Код-Доплати*) – помножити поле *Оклад* з таблиці *Посади* на поле *Коефіцієнт* з таблиці *Доплати*, тобто у четвертому полі у рядку бланка запиту *Поле* ввести вираз (Рис. 4.18):

Доплата: [*Оклад*] * [*Коефіцієнт*]

5) обчислити сумарне значення усіх нарахувань, тобто ввести вираз (Рис. 4.18):

Нараховано: [*Оклад*] + [*Надбавка*] + [*Доплата*]

До речі, для наведеного прикладу формування запиту доцільно встановити більш коректний зв’язок між таблицями без ключового

поля. Слід нагадати, що таблиця *Посади* має зв’язок з файлом Excel, і з цієї причини ключове поле у таблиці не створюється. Зв’язок таблиці *Працівник* з таблицею *Посади* не показує відношення «один-до-багатьох».

Щоб запит спрацював коректно, треба змінити зв’язок, виділивши лінію зв’язку між таблицями і клацнувши двічі по ній. Це призведе до відкривання діалогового вікна *Параметри об’єднання*, в якому треба увімкнути другу опцію і натиснути кнопку *ОК* (Рис. 4.16).

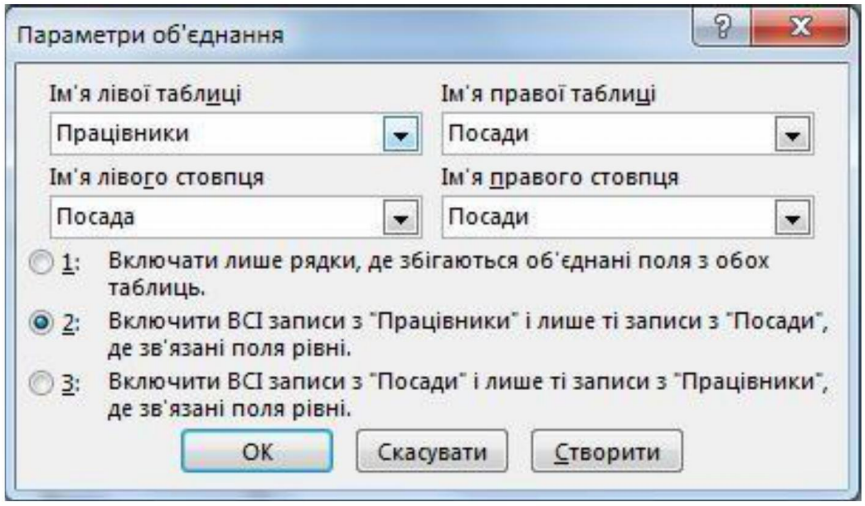


Рис. 4.16. Зміна зв’язку

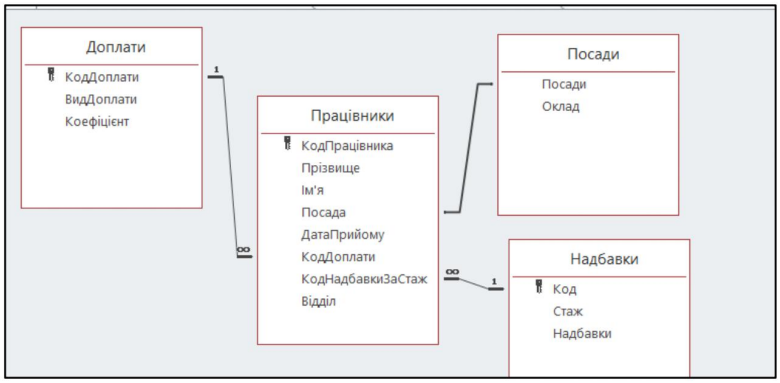


Рис. 4.17. Встановлені зв’язки

Поле:	Прізвище	Оклад	Надбавка	Доплата: [Оклад]*[Коефіцієнт]	Нараховано: [Оклад]+[Надбавка]+[Доплата]
Таблиця:	Працівники	Посади	Надбавки		
Сортування:					
Відображення:	☑	☑	☑	☑	☑
Критерії:					
Або:					

Рис. 4.18. Обчислення значення доплати за індивідуальним кодом доплати та обчислення сумарного значення всіх нарахувань

Можна перейти до режиму *Таблиця* і впевнитись у правильності роботи обох створених обчислених полів (Рис. 4.19).

Надбавки Запит

Прізвище	Оклад	Надбавки	Доплата	Нараховано
Алексєєв	6000,00	1 200,00 ₴	6000	12001
Бурлаков	3500,00	0,00 ₴	3500	7001
Козак	6000,00	750,00 ₴	6000	12001
Остапенко	4200,00	1 800,00 ₴	4200	8401
Сисоєв	8000,00	1 200,00 ₴	8000	16001
Хитрук	7500,00	1 200,00 ₴	7500	15001
Зозуля	4200,00	1 200,00 ₴	4200	8401
Усатюк	7250,00	750,00 ₴	7250	14501
Попеску	7250,00	750,00 ₴	7250	14501
Руссул	4000,00	1 800,00 ₴	4000	8001

Рис. 4.19. Перевірка правильності створених полів

Лишилося закрити запит, натиснувши у правому верхньому куті, і зберегти його з ім'ям *Зарплатня (з надбавками)*.

Розглянемо ще один приклад створення обчислюваного поля запиту за допомогою *Побудовника виразів*, наприклад, звіту з даними про тих працівників, які у поточному році святкують ювілей – вік кратний числу 5.

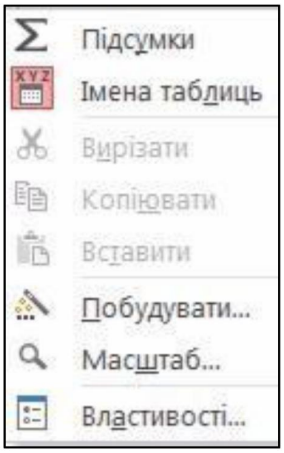


Рис. 4.20. Створення обчислюваного поля запиту за допомогою *Побудовника виразів*

Послідовність створення такого запиту може бути такою:

1) Створити новий запит у режимі *Макет запиту* вибрати за основу таблицю *Анкети*.

2) У перший стовпець бланка запиту занести поле *КодПрацівника* з таблиці *Анкети*.

3) У другий стовпець занести поле *ДатаНародження*, а в рядку *Критерії* цього стовпця натиснути праву кнопку миші та вибрати команду *Побудувати*.

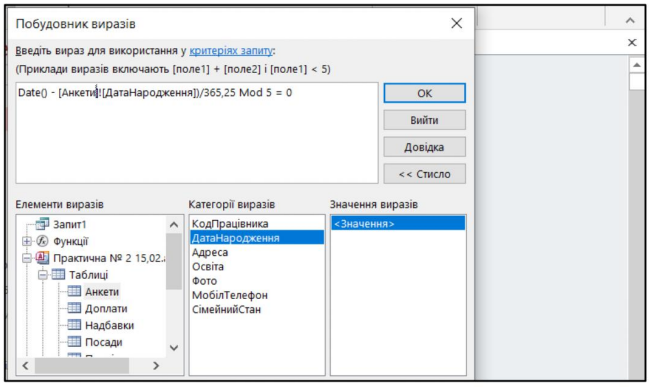


Рис. 4.21. Побудовник виразів

Це призведе до відкривання діалогового вікна *Побудовник виразів* (Рис. 4.21).

4) У вікні *Побудовника виразів* створити такий вираз, скориставшись вибором назви функції *Date()* (поточна системна дата), назви поля *[Анкети]![Дата-Народження]* та оператора *Mod* (залишок від цілочислового ділення):

Date() – [Анкети]![ДатаНародження])/365,25 Mod 5 = 0

Натиснути кнопку *ОК* (Рис. 4.22).

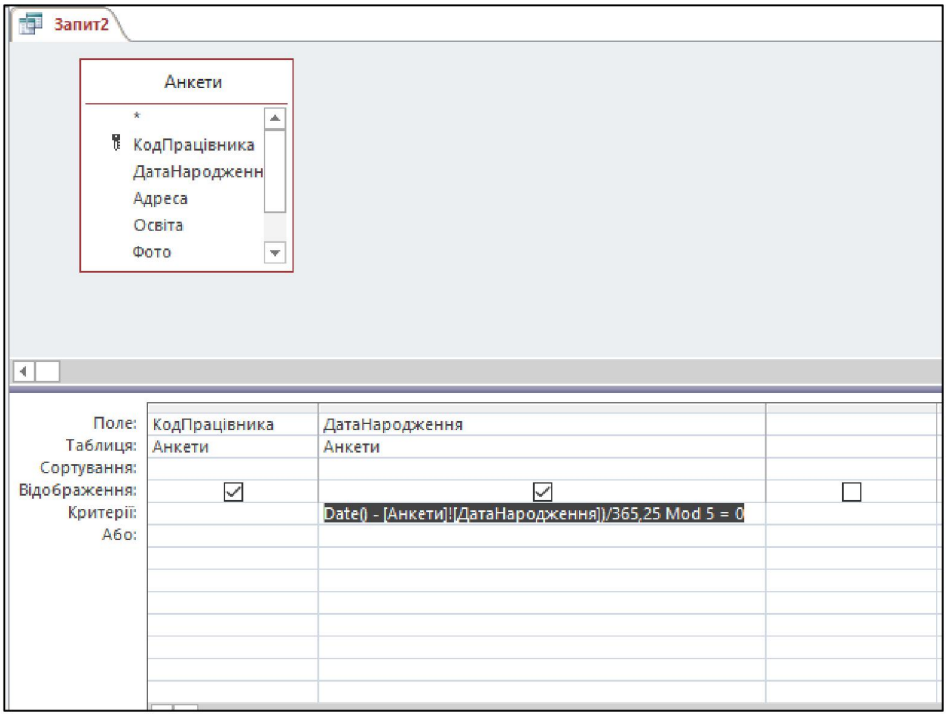


Рис. 4.22. Створення виразу з вибором назви функції *Date()*

5) Запустити запит на виконання і впевнитись у правильності відбору даних.

6) Повернутися до режиму конструктора цього запиту та для третього стовпця у першому його рядку створити такий вираз для обчислення віку ювіляра:

Вік: *DateDiff*("yyyy"; [*ДатаНародження*]; *Date*())

Розтлумачимо цей вираз:

– *Вік* – назва обчислюваного поля у запиті;

– *DateDiff* – функція, яка саме з параметром "yyyy" обчислює кількість років в інтервалі між двома датами;

– [*ДатаНародження*] – поле, яке задає початок часового інтервалу;

– *Date*() – функція, яка повертає поточну системну дату.

Для спрощення можна скористатись *Побудовником виразів*, де назви полів і назви функцій можна вибирати, а не вписувати власноруч (Рис. 4.23).

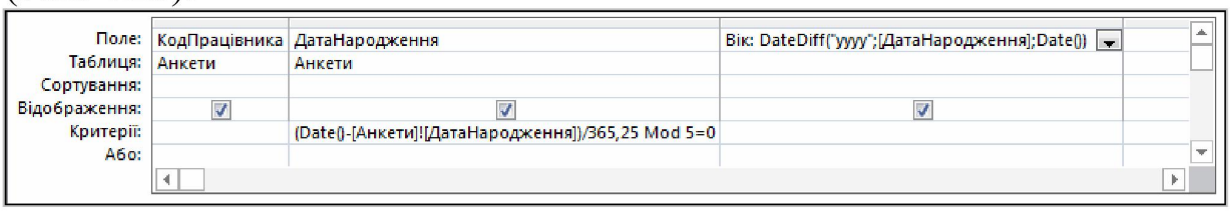


Рис. 4.23. Побудовник виразів

7) Запустити запит на виконання та впевнитись у правильності роботи створеного обчисленого поля (Рис. 4.24).

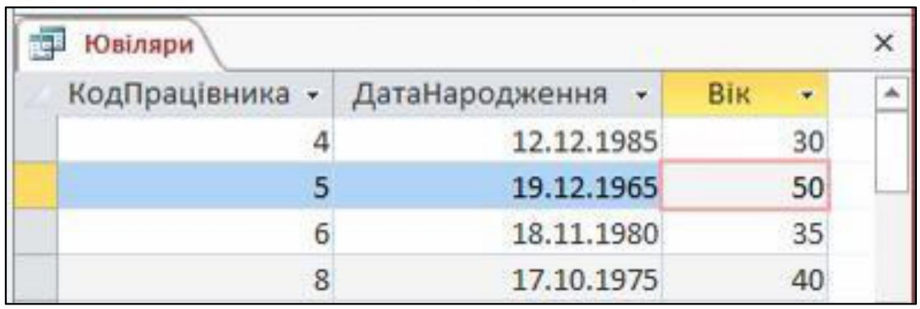


Рис. 4.24. Виконаний запит

Лекція 5. ТИПИ ЗАПИТІВ ACCESS

5.1. Типи створення запитів в Access

На попередній лекції розглядались запити на відбір, оскільки вони відбирають дані за певними критеріями з різних таблиць. Проте запити – це дуже гнучкий інструмент, що дозволяє також додавати, редагувати або вилучати дані таблиць. У програмі Access існують різні типи створення запитів оскільки вони відбирають дані за певними критеріями з різних таблиць. Проте запити – це дуже гнучкий інструмент, що дозволяє також додавати, редагувати або вилучати дані таблиць. У програмі Access існують різні типи створення запитів (*Створення таблиці*, *Додавання*, *Оновлення*, *Перехресний*, *Видалення*, *Об'єднання* тощо), і кожен тип використовується з урахуванням конкретного завдання.

Вибрати той чи інший тип запиту можна на вкладці *Конструктор* у групі *Тип запиту* (Рис. 5.1).

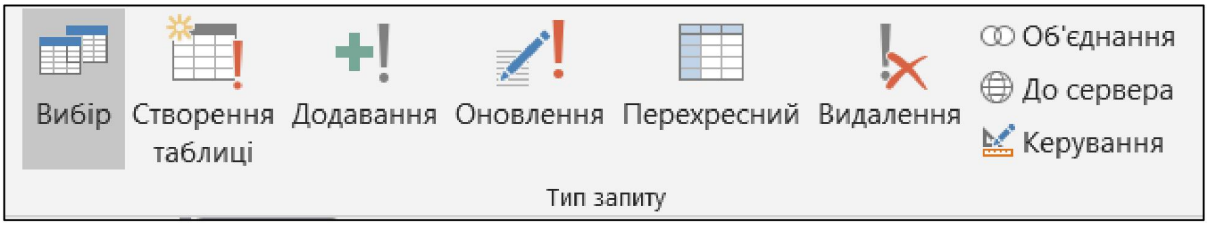


Рис. 5.1. Типи запиту

5.2. Створення перехресного запиту

Перехресні запити дозволяють більш наочно подавати дані підсумкових запитів, які передбачають групування за кількома ознаками: при цьому значення полів за першою ознакою групування можуть стати заголовками рядків, а за другою – заголовками стовпців. Тобто, перехресні запити доцільно будувати на базі таблиць, в яких виконувалися групові операції. У клітинках перехресного запиту будуть відображатися дані, обчислені за допомогою однієї з підсумкових операцій: *Sum* – сума, *Avg* – середнє, *Count* – кількість, *Max* – максимальне значення, *Min* – мінімальне значення. Результати перехресного запиту легко експортувати до Excel або використовувати для створення демонстраційних діаграм.

Розглянемо на конкретному прикладі технологію створення перехресного запиту. Припустимо, що виникла потреба підготувати доповідь, в яку доцільно включити ілюстративний матеріал про розподіл зарплатного фонду працівників по відділах і посадах. Як було

зазначено вище, перехресні запити доцільно будувати на базі таблиць, в яких виконувалися групові операції. У розглянутій БД *Співробітники організації* був створений запит *Сумарні виплати по відділах*, який може бути використаний для розв'язання поставленої задачі, оскільки в нього включені дані з різних таблиць і виконується групова операція підсумовування. Послідовність дій для створення перехресного запиту може бути такою:

1) Створити новий запит за допомогою *Майстра*, вибрати тип *Майстер перехресних запитів* та натиснути кнопку *ОК*.

2) У діалоговому вікні *Майстер перехресних запитів* вибрати запит *Сумарні виплати по відділах* та натиснути кнопку *Далі*.

3) Далі у діалогових вікнах почергово вибрати по одному полю:

– для заголовків рядків – поле *Відділ*;

– для заголовків стовпців – поле *Посада*;

– для обчислень – поле *Оклад*, для функції обчислення – *Сума*, прапорець *Обчислити суму для кожного рядка* має бути увімкненим (значення *Так, включати суми рядків*);

– задати ім'я запиту *Сумарні виплати по відділах перехресний* та натиснути кнопку *Готово* (Рис. 5.2).

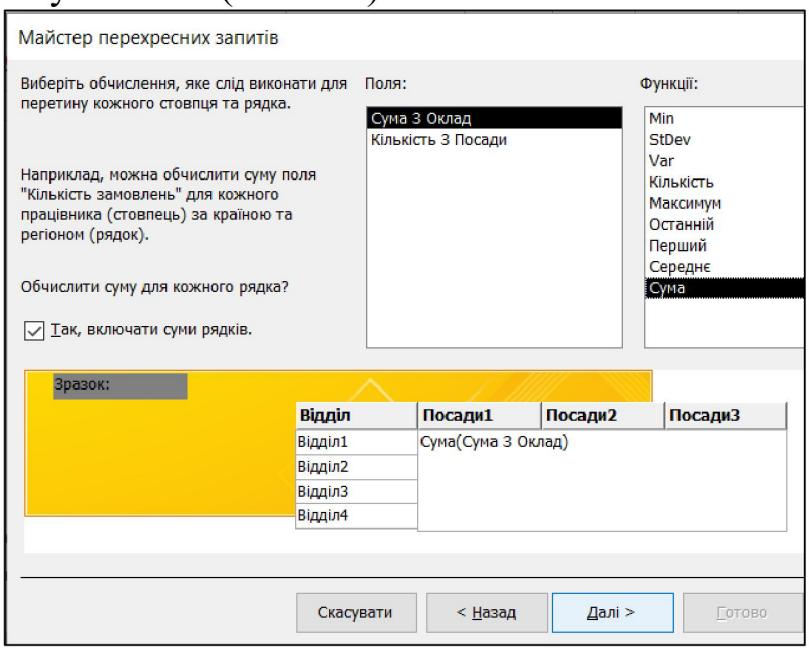
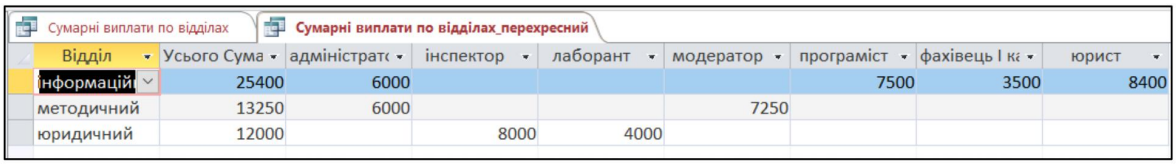


Рис. 5.2. Майстер перехресних запитів

Результати цього перехресного запиту будуть наочно показувати розподіл окладів по категоріях посад і відділах (Рис. 5.3):



Відділ	Усього Сума	адміністрат	інспектор	лаборант	модератор	програміст	фахівець І ка	юрист
інформаційн	25400	6000				7500	3500	8400
методичний	13250	6000			7250			
юридичний	12000		8000	4000				

Рис. 5.3. Розподіл окладів по категоріях посад і відділах

5.3. Створення запиту на відбір з параметром

Запит з параметром запитує у користувача значення одного чи декількох полів, які використовуватимуться як умови на відбір даних. Послідовність створення запиту з параметром відбору працівників певного відділу буде такою:

1) Створити новий запит у режимі *Макет запиту*.

2) Винести таблицю *Працівники* та вибрати з неї до бланка запиту поля: *Відділ*, *Прізвище*, *Ім'я*, *Посада*.

3) У рядку *Критерії* для поля *Відділ* ввести текст у квадратних дужках:

[*З якого відділу?*]

Текст у квадратних дужках є пропозицією ввести параметр відбору даних.

4) На вкладці *Конструктор* у групі *Результати* натиснути кнопку *Запуск*. Запит запропонує ввести назву відділу, наприклад *методичний*(Рис. 5.4).

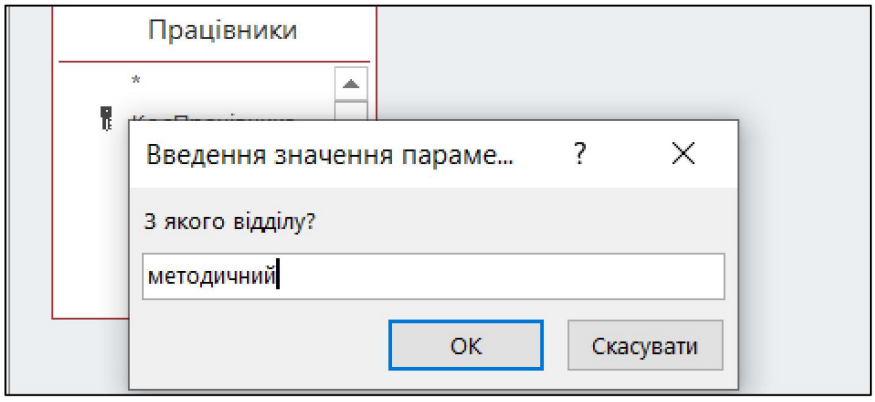


Рис. 5.4. Запит для введення назву відділу

Після натискання кнопки *ОК* система сформує таблицю з даними працівників зазначеного відділу (Рис. 5.5).

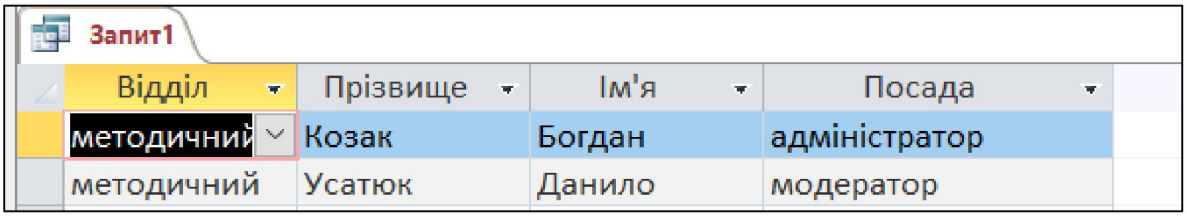


Рис. 5.5. Таблиця з даними працівників зазначеного відділу

5.4. Запит на оновлення записів

Запит на оновлення дозволяє вносити змінення в групу записів однієї чи декількох взаємопов'язаних таблиць. Запит на оновлення неможливо відмінити. Можливо, перед оновленням слід створити резервні копії всіх таблиць, які будуть оновлені таким запитом.

Розглянемо специфіку роботи із запитом на оновлення при створенні запиту для збільшення премії на 10% (з 20% до 30% від розміру окладу) з перерахунком в усіх таблицях і запитах. У нашій БД крім таблиці *Доплати* значення премії фігурує у полях запитів *Оклади та доплати* і *Зарплатня (з надбавками)*. Отже, запит на оновлення має збільшити коефіцієнт нарахування премії з 0,2 до 0,3 у таблиці *Доплати* та в обох запитах.

1) Створити новий запит у режимі *Макет*, включивши в нього поля, які підлягають оновленню, і поля необхідні для критерію відбору. У нашому прикладі треба з таблиці *Доплати* перенести на бланк запиту поля *ВидДоплати* та *Коефіцієнт*.

2) Вибрати на вкладці *Конструктор* у групі *Тип запиту* команду *Оновлення* (або найти цю команду в контекстному меню). Після цього в бланку запиту з'явиться новий рядок *Оновлення до*. Для поля *ВидДоплати* у рядку *Критерії* треба ввести значення *Премія*, а для поля *Коефіцієнт* у рядку *Оновлення до* ввести формулу:

[*Коефіцієнт*] + 0,1

Закрити запит і задати його ім'я *Премія30%* (Рис. 5.6).



Рис. 5.6. Запит *Премія*

3) Після цього у полі переходів БД з'явиться запит *Премія*.

Подвійним клацанням треба запустити його на виконання, при цьому система сформує повідомлення для підтвердження виконання запиту, оскільки дані будуть безповоротно змінені і відновлення змінених даних буде неможливим. Після натискання кнопки *Так* система не сформує ніякої результуючої таблиці і треба самостійно відкрити таблицю *Доплати* (або запити *Оклади та доплати* і *Зарплатня (з надбавками))* та впевнитись в автоматичному зміненні даних у них.

5.5. Запит на видалення записів

Запит на видалення дозволяє вилучити групу записів на основі вказаних критеріїв з однієї чи декількох таблиць. Запит на видалення даних неможливо відмінити. При цьому видаляється весь запис, а не окремі поля всередині нього. Такого роду дії доволі часто доводиться здійснювати користувачеві БД без участі її адміністратора, тому критерії видалення можна або чітко задавати в запиті, або вводити за допомогою діалогового вікна. Наприклад, можна видалити рядки, які мають порожні клітинки певного поля (*IS NULL*), або ж при звільненні співробітника організації треба вилучити записи про цього працівника з БД.

1) Створити новий запит у режимі *Макет*, додати в нього таблицю *Працівники* та винести лише одне поле *Прізвище.*

2) Вибрати на вкладці *Конструктор* у групі *Тип запиту* команду *Видалення* (або найти цю команду в контекстному меню). Після цього в бланку запиту з'явиться новий рядок *Видалення.*

3) У рядку *Критерії* для поля *Прізвище* ввести у квадратних дужках текст повідомлення, яке буде виводитися при виконанні запита (Рис. 5.7):

[*Введіть прізвище працівника, якого треба видалити з БД*]

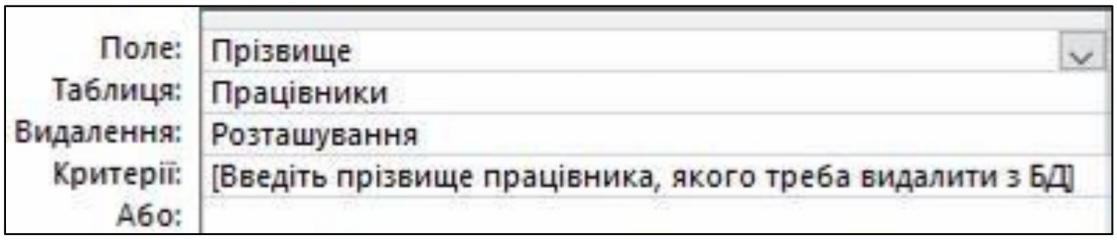


Рис. 5.7. Видалення з БД

4) Закрити запит і задати його ім'я *ВидаленняПрацівника*.

Після цього у вікні переходів бази даних з'явиться новий вид запиту *ВидаленняПрацівника*. Після запуску цього запиту система сформує повідомлення про підтвердження виконання запиту, оскільки дані будуть безповоротно вилучені. Після того, як буде натиснута кнопка *Так*, з'явиться діалогове вікно для введення прізвища як параметра видалення запису з БД (Рис. 5.8).

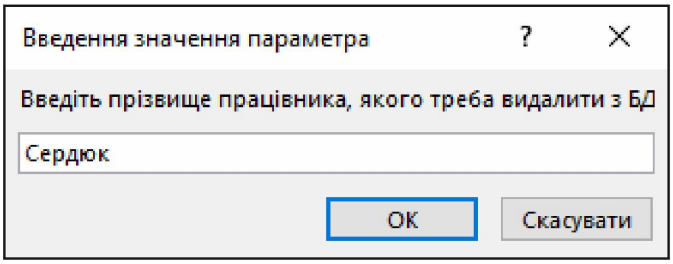


Рис. 5.8. Введення значення параметра

Після введення прізвища працівника для видалення і натискання кнопки *ОК*, система сформує попередження, що відновлення видалених даних буде неможливим. Після натискання кнопки *Так* система не виведе ніякої результуючої таблиці і треба самостійно відкрити таблицю *Працівники*, щоби впевнитися у коректності вилучення даних з неї.

До речі, у бланку запиту можна задавати умови на видалення у декількох полях, використовуючи при цьому різноманітні логічні операції.

При видаленні записів з таблиць, які мають зв'язки з іншими таблицями і запитами, видалення записів буде проведено каскадом в усіх пов'язаних об'єктах. Саме для цього при створенні зв'язків між таблицями встановлюють опцію «Каскадне видалення пов'язаних полів». Тому, якщо при створенні схеми даних така опція не була встановлена, то її можна включити. Для цього треба відкрити вікно *Зв'язки* на вкладці *Знаряддя бази даних* двічі клацнути по лінії зв'язку між таблицями та виконати необхідні правки у вікні *Редагування зв'язків*.

5.6. Запит на створення таблиці

Запит на створення таблиці можна використовувати для створення нової таблиці на основі даних з інших таблиць, включаючи обчислювані поля, відібравши їх за якимось критерієм з однієї чи декількох таблиць. У подальшому сформовані таким чином таблиці можна експортувати до інших систем, наприклад, для звітності

підприємств.

Розглянемо специфіку створення запиту на створення таблиці у БД *Працівник організації* на прикладі формування таблиці з прізвищами і віком працівників з юридичного відділу:

1) Створити новий запит у режимі *Макет*, додавши на бланк запиту таблиці *Працівники* та *Анкети* та включивши поля з цих таблиць у такій послідовності:

– поле *Прізвище* з таблиці *Працівники*;
– поле *Відділ* з таблиці *Працівники*;
– поле *ДатаНародження* з таблиці *Анкети*.

2) Задати параметри відбору даних для цих трьох полів:

– для поля *Прізвище* задати сортування *за зростанням*;
– для поля *Відділ* у рядку *Критерії* ввести текст «методичний».

3) У четвертому полі (стовпці) у рядку *Поле* сформувати такий вираз:

Вік: (*Date*() - [*ДатаНародження*]) / 365,25

4) Утримуючи курсор над полем *Вік*, натиснути праву кнопку миші і виконати команду контекстного меню *Властивості.* Це призведе до відкривання діалогового вікна *Аркуш властивостей*, в якому для властивості *Формат* вибрати значення *Фіксований*, а для властивості *Кількість знаків після коми* – значення 1.

5) До того як використовувати дані для створення таблиці, доцільно впевнитись у правильності їх відбору, переключившись до режиму *Таблиця* або натиснувши кнопку *Запуск* на вкладці *Конструктор* у групі *Результати*.

6) Натиснути клавіші [Ctrl] + [S], щоб зберегти запит, ввести ім’я запиту *Нова таблиця Вік працівників юрвідділу*.

7) Повернутися до режиму *Конструктор* і для поля *Відділ* у рядку *Відображення* вимкнути «галочку» (Рис. 5.9).

Поле:	Прізвище	Відділ	ДатаНародження	Вік: (Date()-[ДатаНародження])/365,25
Таблиця:	Працівники	Працівники	Анкети	
Сортування:	За зростанням			
Відображення:	☑	☐	☑	☑
Критерії:		"юридичний"		
Або:				

Рис. 5.9. У рядку *Відображення* вимкнення «галочки».

8) Виконати команду *Створення таблиці* на вкладці *Конструктор* у групі *Тип запиту*. У діалоговому вікні в полі *Ім’я таблиці* ввести *Працівники юрвідділу* і натиснути кнопку *ОК* (Рис. 5.10).

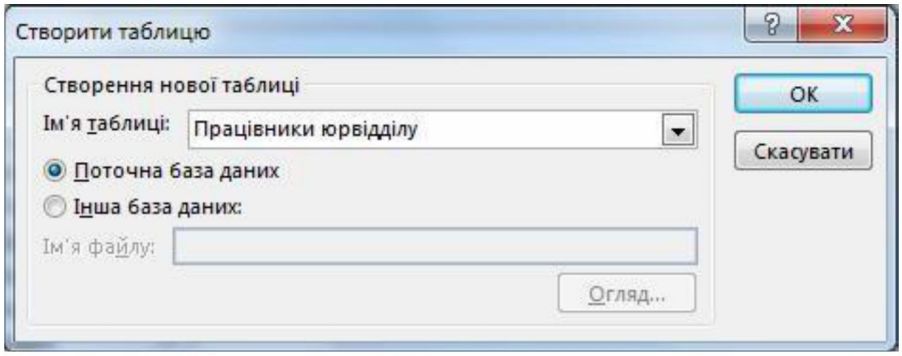


Рис. 5.10. Створення нової таблиці

9) На вкладці *Конструктор* у групі *Результати* натиснути кнопку *Запуск*. У діалоговому вікні натиснути кнопку *Так*, після чого в області переходів з'явиться нова таблиця (Рис. 5.11).

Нова таблиця Вік працівників юрвідділу

Прізвище	ДатаНародж	Вік
Антонов	11.12.1979	35,7
Кондратюк	12.12.1980	34,7
Лисенко	19.12.1975	39,7
Пурич	29.03.1992	23,4

Рис. 5.11. Створена нова таблиця

5.7. Запит на додавання даних

Запит на додавання даних можна використовувати для відбирання даних з однієї або декількох таблиць і додавання їх в іншу таблицю.

Розглянемо специфіку створення запиту на додавання даних на прикладі додавання у таблицю *Працівники юрвідділу* даних про співробітників з інформаційного відділу.

1) Відкрити запит *Вік працівників юрвідділу* в конструкторі.

2) На вкладці *Конструктор* у групі *Тип запиту* вибрати команду *Додавання*. У діалоговому вікні *Додавання* клацнути стрілку в полі *Ім'я таблиці* та вибрати зі списку *Працівники юрвідділу*, натиснути *ОК*.

3) У бланку запиту в рядку *Критерії* поля *Відділ* видалити значення «юридичний» і ввести «інформаційний».

4) На вкладці *Конструктор* у групі *Результати* натиснути кнопку *Запуск*. При цьому система сформує повідомлення про кількість доданих у таблицю записів та запитом на підтвердження додавання цих записів.

Лекція 6. СТВОРЕННЯ ФОРМ

6.1. Способи створення форм

Форми Access дозволяють створювати інтерфейс користувача для таблиць бази даних. Хоча бази даних можуть створюватися без форм, більшість користувачів вважають за краще використовувати форми для перегляданння, введення та редагування табличних даних. Якщо база даних використовуватиметься декількома користувачами, добре продумані форми – запорука точності даних і ефективності роботи з ними.

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.

Звичайно форми будуються на основі таблиць або запитів, а вміст форм завжди відповідає інформації в таблицях і запитах. Вигідною перевагою форм є їхня багатофункціональність, вони дозволяють виконувати завдання, які не можна виконати в режимі таблиці. Наприклад, можна створювати форми, на яких розташовуватимуться інші форми (форма всередині іншої форми називається **підпорядкованою** або **підформою**). Також форми дозволяють перевіряти коректність даних у таблицях. Крім того, форми дозволяють організовувати і виводити на екран різного роду обчислення над даними.

Створити форму в базі даних Access можна кількома способами, що пропонуються на вкладці *Створення* стрічки у групі *Форми* (Рис. 6.1).



Рис. 6.1. Інструменти для створення форм

Щоб швидко створити форму, треба вибрати таблицю або запит в області переходів і на основі цього об'єкта створити форму за допомогою команди ***Форма***. Наприклад, відкривши запит *Зарплатня (з надбавками)* і натиснувши кнопку *Форма*, матимемо швидке створення форми, яку можна зберегти з ім'ям *Зарплата*. 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.

При використанні різних варіантів створення форми є можливість включати до неї поля з різних таблиць і запитів.

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. Створюється така форма командою *Додаткові форми – Кілька елементів* на вкладці *Створення*.

Розділена форма відображає дані у режимі таблиці і поданні 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). *Розділена форма* дозволяє вивести на екран форму разом з вихідною таблицею з можливістю використання додаткових засобів. Створюється така форма командою *Додаткові форми – Розділена форма* на вкладці *Створення*.

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. Різні види конструювання форм навігації можна вибрати командою *Навігація* на вкладці *Створення*.

Форми можуть виводитись на екран в чотирьох режимах: *Режим форми*, *Подання таблиці*, *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 *Подання таблиці*, повертає до перегляду інформації у звичайному табличному режимі.

Режим *Конструктор* дозволяє вносити змінення, які вимагають

детального розгляду структури форми, наприклад, розділів *Верхній колонтитул*, *Область даних* та *Нижній колонтитул*. При внесенні змінень у режимі *Конструктор* переглядати базові дані не можна. Корисними можливостями режиму *Конструктор* є: 1) можливість додавання різних елементів керування, таких як приєднані рамки об'єкта, розриви сторінок та діаграми; 2) змінення джерел для текстових полів без використання вікна властивостей; 3) змінення розмірів розділів форми; змінення певних властивостей форми, які не можна змінити у *Режимі розмітки*.

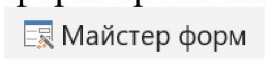
6.2. Використання інструмента *Форма*

Інструмент ***Форма*** (в деяких версіях *Автоформа*) – це найшвидший спосіб створення форми одним клацанням миші.

Прикладом, для створення форми *Працівники* за однойменною таблицею слід один раз клацнути по цій таблиці в області переходів (не відкриваючи її) і на вкладці *Створення* у групі *Форми* клацнути команду *Форма* (Рис. 6.1). Після цього з'явиться готова форма *Працівники*, яку за потреби можна відредагувати в режимі *Конструктор* або в режимі *Подання*. Наприклад, можна зменшити ширину всіх полів за максимальним вмістом, а також зменшити висоту полів списків, щоб відображалось лише одне значення.

6.3. Використання *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 

. Далі слідувати інструкціям на сторінках *Майстра форм*. При цьому для додавання полів на форму з декількох таблиць або запитів не треба натискати кнопку *Далі* або *Готово* після вибирання полів із першої таблиці або запиту на першій сторінці *Майстра форм*. Замість цього треба повторити зазначені дії для вибирання таблиці або запиту і клацанням вибрати додаткові поля, які потрібно винести на форму. Для продовження натиснути кнопку *Далі* або *Готово*.

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.

Приміром, для створення форми *Анкети* за однойменною таблицею за допомогою *Майстра форм* послідовність дій може бути такою:

1) На вкладці *Створення* у групі *Форми* клацнути команду *Майстер форм* (див. Рис. 6.1).

2) У вікні *Майстра форм* вибрати зі списку таблиць і запитів таблицю *Анкети* і за допомогою кнопки >> вибрати усі поля. Натиснути кнопку *Далі*.

3) Вибрати макет форми *Стовпці* і натиснути кнопку *Далі*.

4) На останньому кроці *Майстра форм* з пропозицією *Відкрити форму для перегляду або введення даних* просто натиснути кнопку *Готово*.

Після цього з'явиться готова форма *Анкети*, яку за потреби можна відредагувати в режимі *Розмітки* або в режимі *Конструктор* (Рис. 6.2).

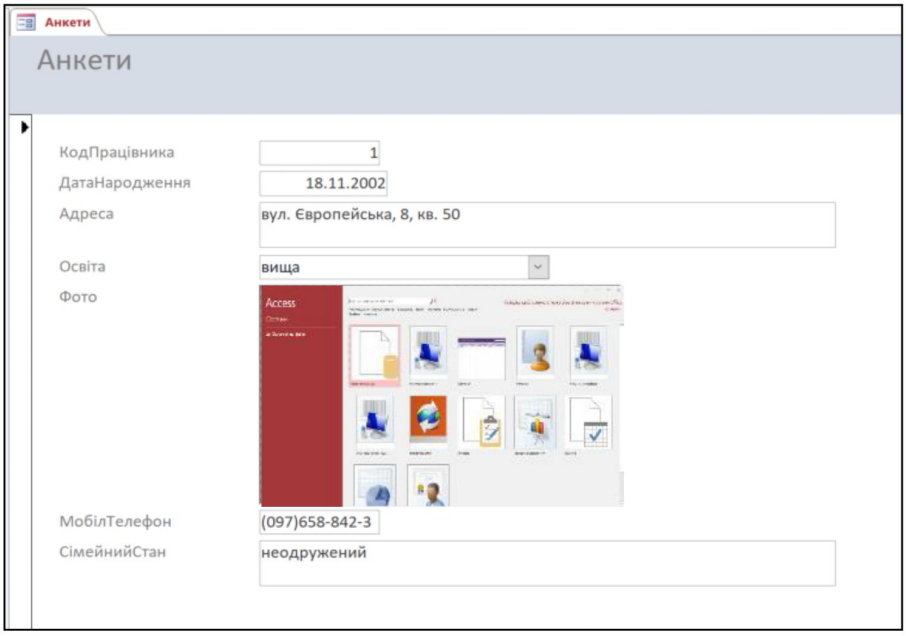


Рис. 6.2. Форма *Анкети*

6.4. Використання пустої форми

Розробляти форму для додавання у таблицю записів або внесення змінень даних зручно на основі пустої (порожньої) форми, проектування якої починається клацанням лівою кнопкою миші по піктограмі *Пуста форма* на вкладці *Створення*.

Припустимо, що для внесення змінень і доповнень у таблицю *Доплати* потрібно розробити однойменну форму:

1) Натиснути кнопку *Пуста форма* на вкладці *Створення*.

2) Перенести з вікна *Список полів* на форму всі поля таблиці *Доплати* (порядок перенесення полів не важливий). Для цього треба по черзі двічі клацати на іменах полів у списку полів або перетягнути їх на форму.

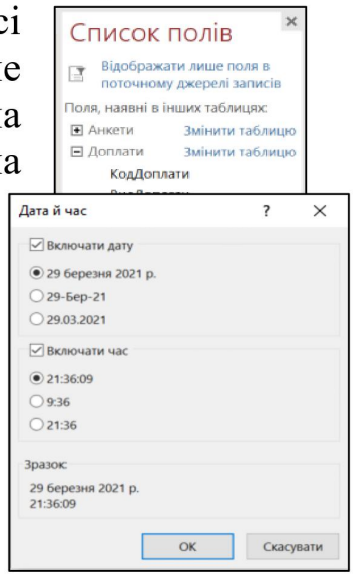


Рис. 6.3. Налаштування форми

3) Змінити дизайн створеної форми можна різними способами, наприклад, відкривши вкладку *Основне* і скориставшись панеллю *Форматування тексту*, виділяючи необхідне поле. Інший спосіб форматування є використання режиму *Конструктор*. Перейти до цього режиму можна, натиснувши кнопку-піктограму у нижньому правому куті вікна форми. Можна скористатися і третім способом, залишитися у *Режимі розмітки* (піктограма у нижньому правому куті форми) і відкрити вікно властивостей, в якому змінити параметри форми та елементи на ній.

За допомогою засобів групи *Елементи керування* на вкладці *Конструктор* можна додати на форму емблему, назву, номери сторінок або дату й час. Приміром, щоб додати у верхній колонтитул форми дату й час треба на вкладці *Конструктор* у групі *Колонтитули* виконати команду *Дата й час*. У діалоговому вікні вибрати формат відображення дати й часу. При бажанні можна перемістити поле дати й часу до нижнього колонтитула форми та виконати інші налаштування форми (Рис. 6.3).

4) Зберегти форму, задавши ім’я форми *Доплати*.

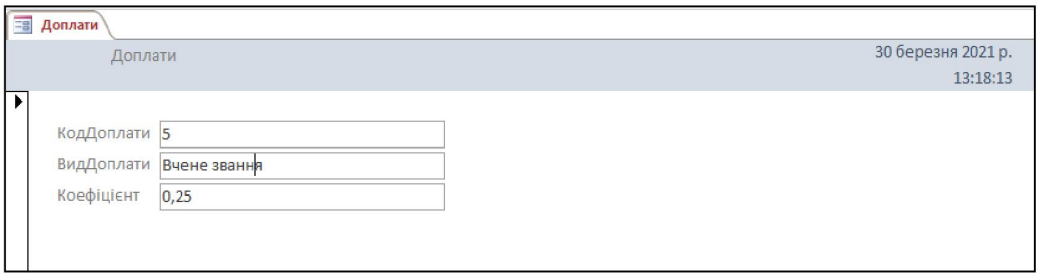


Рис. 6.4. Форма *Доплати*

Залишилося запустити готову форму, переглянути на ній види доплат, додати новий вид доплат, наприклад «Вчене звання», і подивитися, як змінилася основна таблиця *Доплати* (Рис. 6.4).

6.5. Розробка форми у режимі *Конструктор*

Звичайно режим *Конструктор форм* (Рис. 6.5) використовується для удосконалення зовнішнього вигляду форми, створеної за допомогою *Майстра форм*, але може використовуватися і при створенні форми «з нуля». Для запуску *Конструктора форм* можна скористатися командою *Конструктор форм* на вкладці *Створення* у групі *Форми*. Це призведе до відкривання вікна з порожньою заготовкою форми.

Структурно форми у цьому режимі можуть включати такі області:

1) *Верхній колонтитул форми* (у деяких версіях *Заголовок*) – область у верхній частині форми, в якій можна розташовувати інформацію, яка не залежить від змісту записів, наприклад, назву організації.

2) *Подробиці* (у деяких версіях *Область даних*) – центральна частина форми, на яку виводяться дані з таблиць або запитів, пояснювальні тексти, елементи керування.

3) *Нижній колонтитул форми* (у деяких версіях *Примітка форми*) – область у нижній частині форми, в яку можна виводити інформацію у вигляді пояснень.

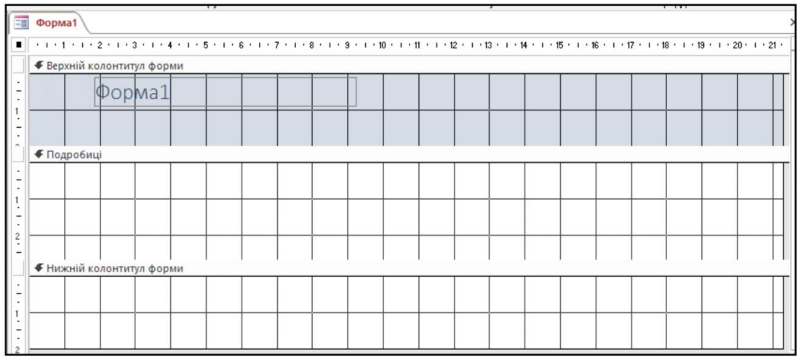


Рис. 6.5. Режим *Конструктора форм*

Кожна з областей є автономною, що дає можливість задавати їм власний фон та формат подання даних. Ту чи іншу область можна приховати або відобразити за допомогою відповідної команди контекстного меню. Якщо не потрібно мати на формі області колонтитулів, їх можна прибрати, і навпаки, якщо форма не містить цих областей, їх можна додати. Для цього клацнути правою кнопкою миші по області даних (*Подробиці*) на формі і вибрати з контекстного меню команду *Заголовок / примітка форми*. Для вибору джерел записів для форми треба на вкладці *Знаряддя конструктора форм / Конструктор* виконати команду  *Додавання наявних полів*, після чого праворуч з'явиться вікно *Список полів*.

Перенесення необхідних найменувань полів на форму здійснюється захопленням лівою кнопкою миші на імені поля у *Списку полів* з подальшим перетягуванням його на форму. При цьому, одночасно з ім'ям поля на формі з'являтиметься текстове вікно для виведення вмісту поля. Наприклад, для створення форми *Надбавки* слід перенести на форму поля таблиці *Надбавки*.

Розміри, спосіб подання інформації, колірну гаму, положення та інші властивості можна змінювати, використовуючи інструменти на вкладках панелі *Знаряддя конструктора форм* або у вікні властивостей.

Після усіх налаштувань слід закрити форму, надавши їй ім'я *Надбавки* (Рис. 6.6).

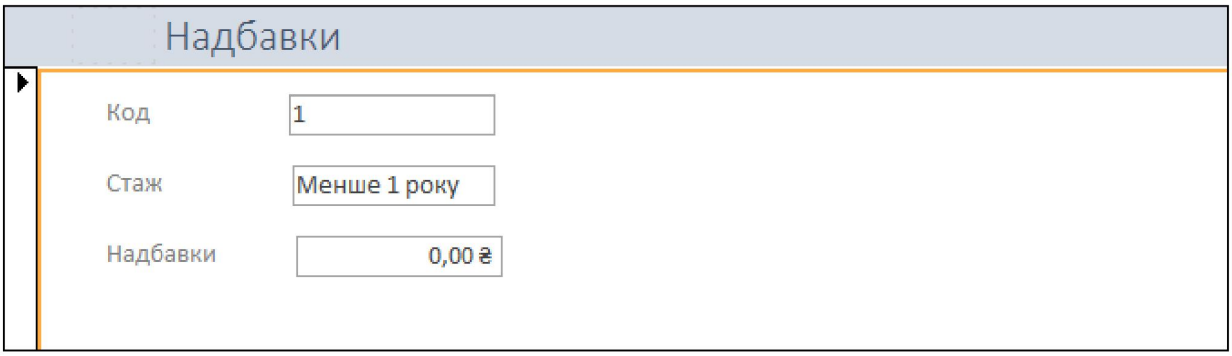


Рис. 6.6. Форма *Надбавки*

До речі, яка завгодно форма, використовувана для переглядання даних, внизу має панель інструментів, що дозволяє здійснювати аналіз записів на формі, а також вікно *Пошук* для введення пошукових виразів, щоб швидко віднайти потрібний запис.

Форма, створена у режимі *Конструктор*, дозволяє доповнювати вихідну таблицю даними тільки у тому разі, якщо на ній присутні поля,

контрольовані з таблиці джерела.

6.6. Створення підформ (підпорядкованих 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. Підформи (підпорядковані форми) – зручний інструмент для подібних завдань, і в Access їх можна швидко створити декількома способами.

Підформа – це форма, яка містить 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». Наприклад, можна створити форму з відомостями про доплати, яка містить підпорядковану форму з інформацією про працівників, які отримують цей вид доплати. Дані у таблиці *Доплати* є стороною «один», а дані в таблиці *Працівники* – стороною «багато»: кожен вид доплати може отримувати декілька співробітників.

Головна форма та підформа пов'язані таким чином, що у підформі відображаються лише записи, пов'язані з поточним записом у головній формі. Наприклад, коли на головній формі відображаються відомості про доплату у вигляді премії, на підформу виводяться дані про працівників, які отримують цей вид доплати. Якби форма та підформа не були пов'язані, у підформі відображалися б усі працівники, а не лише премійовані.

Access 2019 пропонує декілька інструментів для створення підформ, кожен з яких має свою специфіку, і, залежно від конкретної ситуації, можна вибрати найбільш зручний спосіб.

1) **Створення форми, яка містить підформу, за допомогою *Майстра форм*** – це найшвидший спосіб почати роботу, якщо ані форму, яку потрібно використовувати як головну форму, ані підформу ще не створено. У цій процедурі створюється нове поєднання форми та підформи за допомогою команди *Майстер форм* на вкладці *Створити* у групі *Форми*. При цьому Access створить дві форми: одну для головної форми, а другу власне для підформи.

При цьому способі створення послідовність дій може бути такою:

– на вкладці *Створення* у групі *Форми* натиснути кнопку *Майстер форм*;

– на першій сторінці майстра з розкривного списку *Таблиці та запити* вибрати потрібну таблицю або запит, наприклад, таблицю *Надбавки* (Рис. 6.7);

– вибрати з таблиці (або запиту) поля *Стаж* та *Надбавка*, які потрібно включити до форми. На цьому ж етапі у цьому ж вікні змінити таблицю, наприклад, на таблицю *Працівники*, та вибрати ще і з неї поля для підформи, наприклад, поля *Прізвище*, *Ім'я*, *Посада*, *Відділ*;

– на наступному кроці вибрати зовнішній вигляд (макет) підформи;

– на останньому кроці *Майстра* задати назви форм *Надбавки з підформою* та *Працівники підформа.*

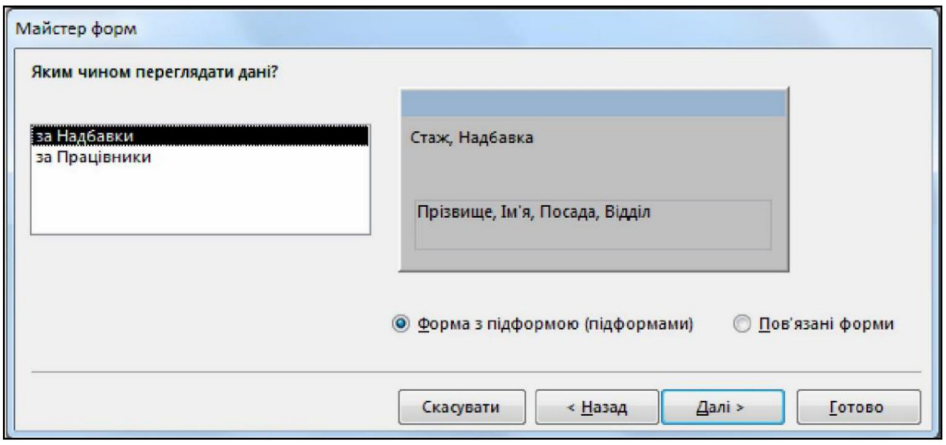


Рис. 6.7. Майстер форм

Така послідовність дій створить дві форми: головну *Надбавки з підформою*, що містить елемент керування *Підформа*, та другу *Працівники підформа* (Рис. 6.8).

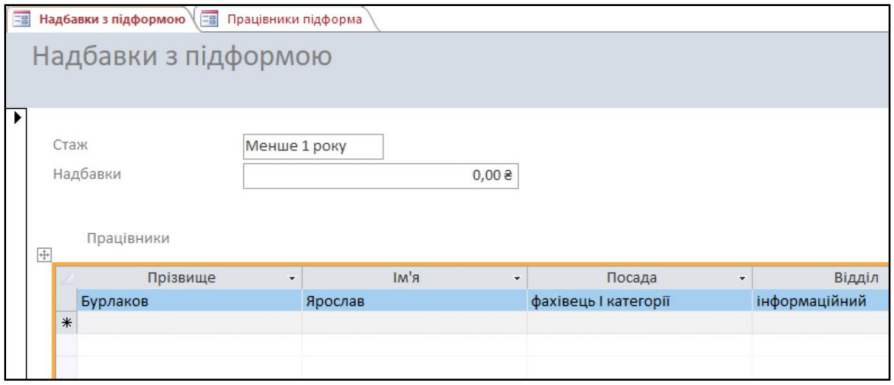


Рис. 6.8. Створені форми

2) Додавання однієї або кількох підформ до наявної форми за допомогою майстра підформ. Для кожної підформи у програмі Access можна на вибір створити нову форму або використати вже наявну форму. Наприклад, щоб до наявної форми *Доплати* додати підформу *Працівники* потрібно:

– в області переходів клацнути правою кнопкою миші на наявній формі *Доплати* і вибрати команду контекстного меню *Конструктор*;

– на вкладці *Знаряддя конструктора форм / Конструктор* у групі *Елементи керування* натиснути стрілку розкривного списку для відображення колекції *Елементи керування* (Рис. 6.9) та переконатися, що вибрано параметр *Застосувати майстри елементів керування*;

– на вкладці *Конструктор* у колекції *Елементи керування* натиснути кнопку *Підформа / підзвіт*;

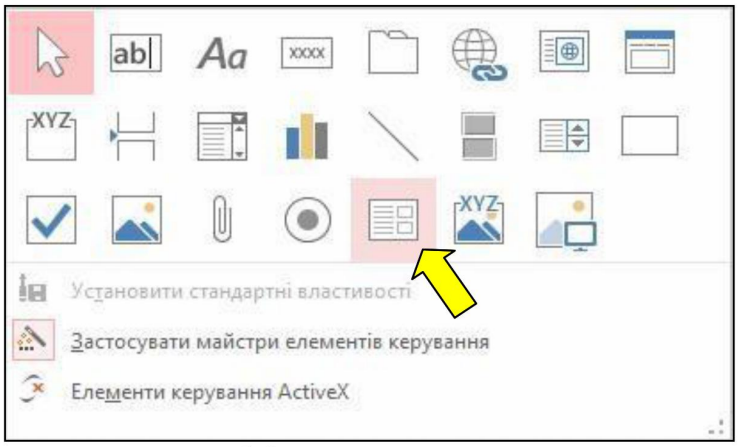


Рис. 6.9. Елементи керування

– клацнути на форму у те місце, куди розмістити підформу;

– дотримуватись вказівок майстра підпорядкованих форм;

– погодитись з пропозицією *Використовувати наявні таблиці та запити*, натиснувши кнопку *Далі*;

– вибрати зі списку таблиць і запитів таблицю *Працівники* та перенести всі або лише уподобані її поля. Натиснути кнопку *Готово*.

Після натискання кнопки *Готово* програма Access додасть до форми елемент керування *Підформа*, а у переліку форм в області переходів з'явиться форма з назвою *Підпорядкована форма Працівники* (Рис. 6.10). Залишилось перейти до режиму форми та впевнитись у працездатності форми *Доплати*.

Якщо замість пропозиції майстра підпорядкованих форм *Використовувати наявні таблиці та запити* вибрати опцію *Використовувати наявну форму*, це дозволить вибрати одну із наявних

форм у базі даних. Наприклад, в такий спосіб можна на формі *Працівники* розмістити форму *Анкети*.

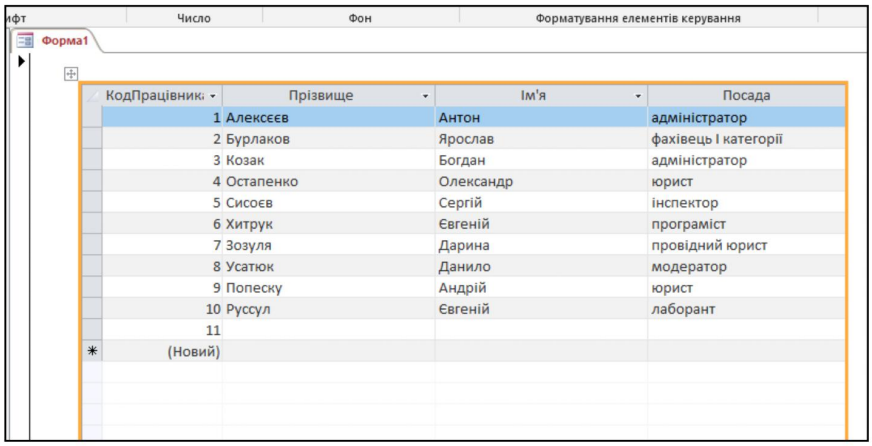


Рис. 6.10. Форма *Працівники*

3) Створення підформи перетягуванням однієї форми в іншу – це найшвидший спосіб створення підформ і його використовують, коли на існуючу (головну) форму треба винести одну чи декілька вже наявних форм (таблиць або запитів) в якості підформ. При цьому неможна при створенні підформи вибирати, які саме поля переносити, оскільки форма переноситься цілком з усіма її полями і форматуванням (Рис. 6.11).

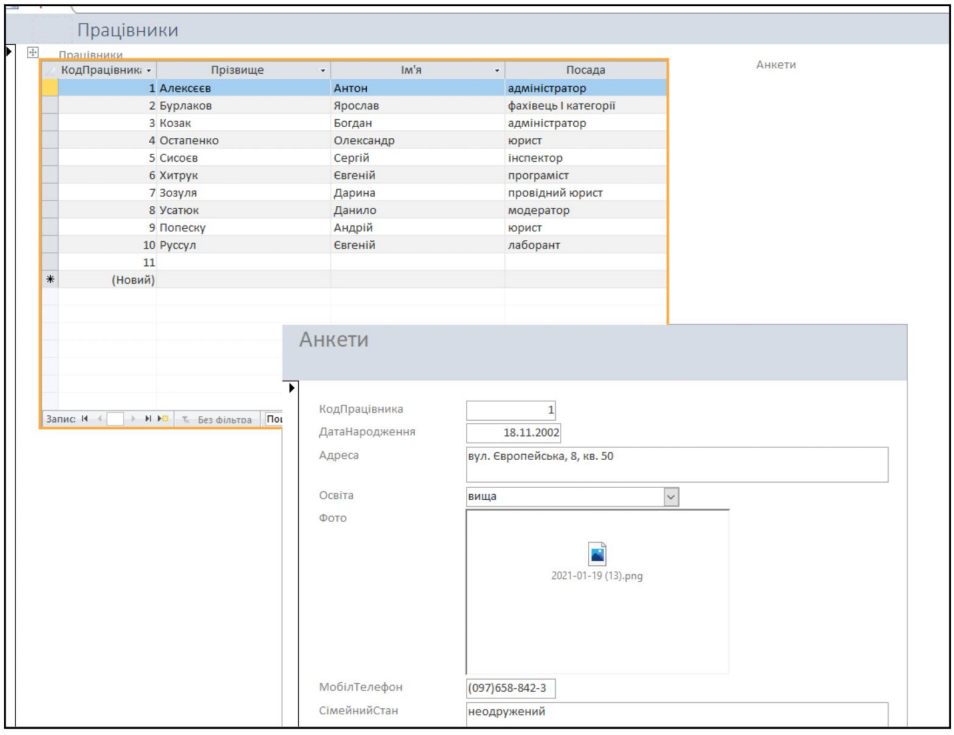


Рис. 6.11. Створення підформи

Наприклад для створення на формі *Доплати* підформи *Працівники*

послідовність дій може бути такою:

– в області переходів клацнути правою кнопкою миші форму *Доплати*, яку потрібно використовувати як головну форму, і вибрати режим *Конструктор* або *Режим розмітки*;

– перетягнути форму *Працівники*, яку слід використовувати як підформу, з області переходів до головної форми. Перетягувати можна не лише наявні форми, а й таблиці. Після перетягування на головній формі буде створено елемент керування *Підформа*. Також буде виконана спроба пов'язати підформу з головною, виходячи зі встановлених у БД зв'язків;

– за потреби додавання й інших підформ до головної форми треба повторити попередній крок;

– щоб переконатися в успішності зв'язування, на вкладці *Основне* у групі *Подання* натиснути кнопку *Подання* і вибрати пункт *Режим форми*, а потім за допомогою селектора записів головної форми переглянути декілька записів. Якщо у підформі правильно виводяться (змінюються) дані для кожного виду оплати, це означає, що процедура завершена успішно. Якщо ж ця перевірка завершилася невдачею, це свідчить про те, що програмі Access не вдалося визначити як зв'язати підформу з головною формою. В такому разі властивості *Зв'язати дочірні поля* та *Зв'язати головні поля* елемента керування *Підформа* залишилися порожніми та їх слід задати вручну (Рис. 6.12);

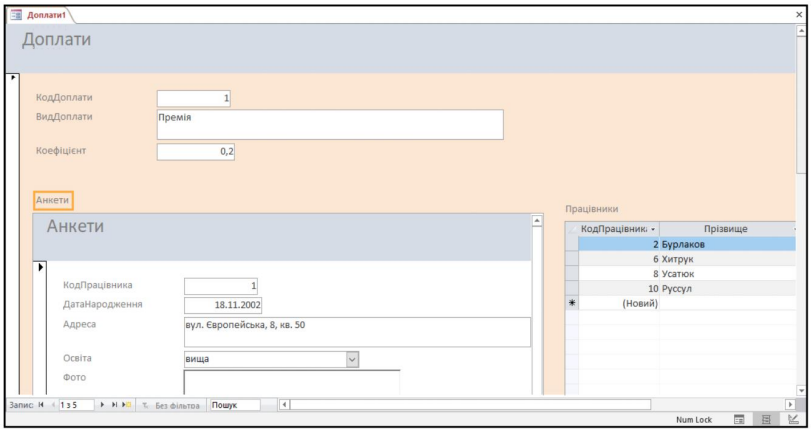


Рис. 6.12. Форма *Доплати*

– зберегти головну форму *Надбавки*, перейти до режиму форми і переконатися, що форма працює належним чином.

Лекція 7. СТВОРЕННЯ ЗВІТІВ

7.1. Огляд звітів у програмі Access

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.

7.2. Частини звіту

За своєю структурою макет звіту поділено на розділи, які можна побачити у режимі конструктора. Для того щоб створити звіт на основі певної таблиці достатньо перейти в пункт меню *Створення* та у групі *Звіти* обрати *Звіт*, після чого буде автоматично створено новий звіт на основі обраної таблиці. У режимі розмітки розділи виділені не так чітко, тому краще редагувати звіти в режиму конструктора. Розглянемо й охарактеризуємо призначення кожного з семи розділів у порядку їх розташування у звіті.

1) **Заголовок звіту** (верхній колонтитул звіту) містить інформацію, що виводиться тільки один раз на початку звіту на титульній сторінці, наприклад емблему, назву звіту або дату. Заголовок звіту друкується перед верхнім колонтитулом сторінки (Рис. 7.1).

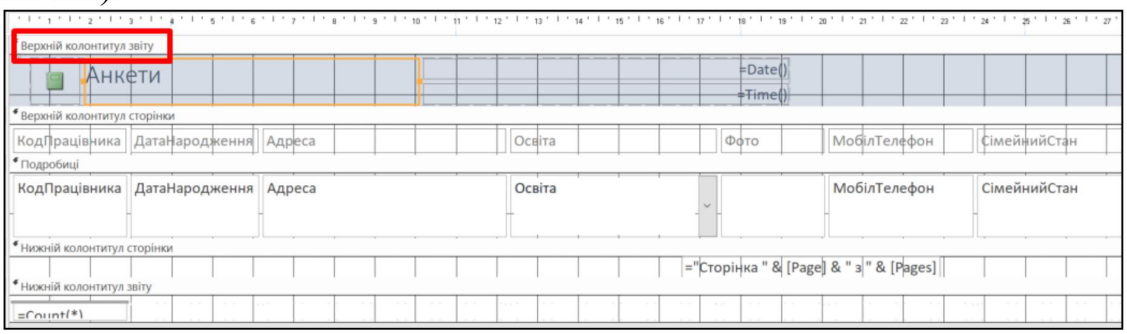


Рис. 7.1. Заголовок звіту

2) **Верхній колонтитул сторінки** містить інформацію, що буде виводитися на друк вгорі кожної сторінки звіту, наприклад: назву звіту, дату або час формування поточного звіту тощо (Рис. 7.2).

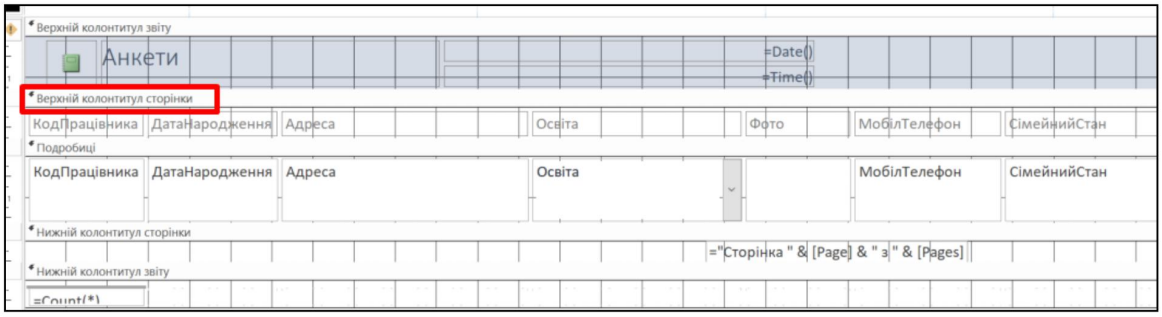


Рис. 7.2. Верхній колонтитул сторінки

3) **Заголовок групи** (верхній колонтитул групи) друкується перед кожною новою групою записів. Цей розділ використовується для друку назви групи. Наприклад, якщо звіт згрупований по відділах, у заголовках груп можна вказати їхні назви. У звіті може бути кілька заголовків груп залежно від кількості рівнів групування.

4) **Подробиці** (область даних) містять елементи керування, які становлять тіло звіту (Рис. 7.3).

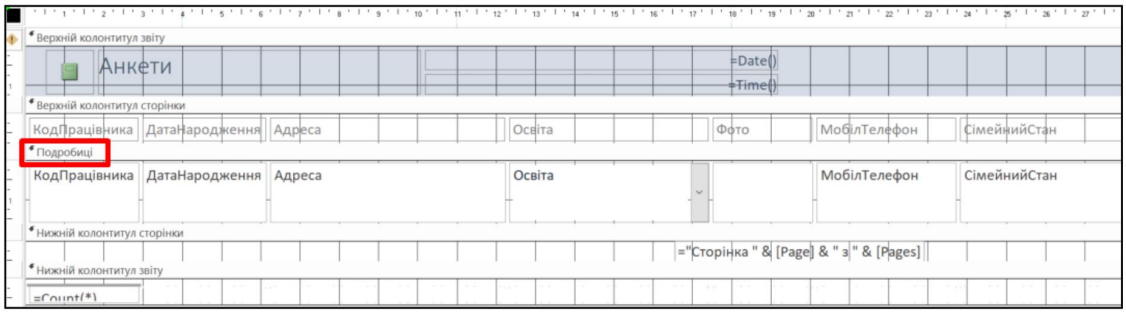


Рис. 7.3. Подробиці

5) **Примітка групи** (нижній колонтитул групи) виводиться на друк наприкінці кожної групи записів. Тут відображаються зведені дані для групи. Якщо помістити сюди обчислюваний елемент керування, в якому використовується агрегатна функція підсумовування, то сума буде розраховуватись для поточної групи. У звіті може бути кілька приміток груп залежно від кількості рівнів групування.

6) **Нижній колонтитул** друкується внизу кожної сторінки. Використовується для нумерації сторінок або даних, які стосуються кожної сторінки. На рис. 7.1 у нижньому колонтитулі розміщено поточну дату та номер поточної сторінки із зазначенням загальної

кількості сторінок (Рис. 7.4).

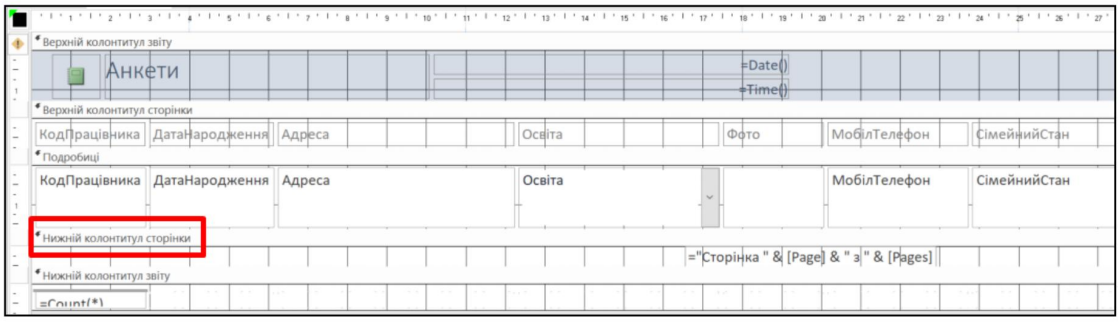


Рис. 7.4. Нижній колонтитул сторінки

7) **Примітка звіту** (нижній колонтитул звіту) виводиться один раз наприкінці звіту і використовується для відображення підсумків та іншої зведеної інформації по всьому звіту. У режимі конструктора цей розділ відображається під нижнім колонтитулом сторінки. Проте в усіх інших режимах (наприклад, у режимі розмітки, при друкуванні або попередньому перегляді) цей розділ буде розташовуватися над нижнім колонтитулом сторінки, відразу після примітки останньої групи або області даних на останній сторінці.

Створювати змістовні звіти набагато простіше, коли база даних має добре спроектовану структуру таблиці та правильно визначені зв’язки.

7.3. Способи створення звітів

Створювати звіти в базі даних Access можна декількома способами, які пропонуються на вкладці *Створення* стрічки у групі *Звіти* (Рис. 7.5).

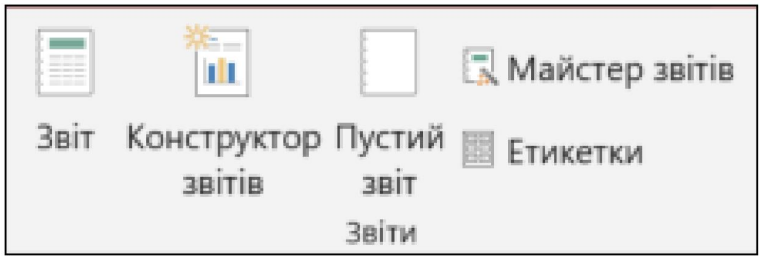


Рис. 7.5. Створення звітів

Звіт дозволяє створити найпростіший табличний звіт, що містить усі поля з джерела записів, вибраного в області переходів. Наприклад, для виконання п. 2 завдання можна просто клацнути в області переходів на таблиці *Анкети*, не відкриваючи її, та виконати команду *Звіт* на вкладці *Створення* у групі *Звіти.* Після цього буде сформовано такий звіт (Рис. 7.6):

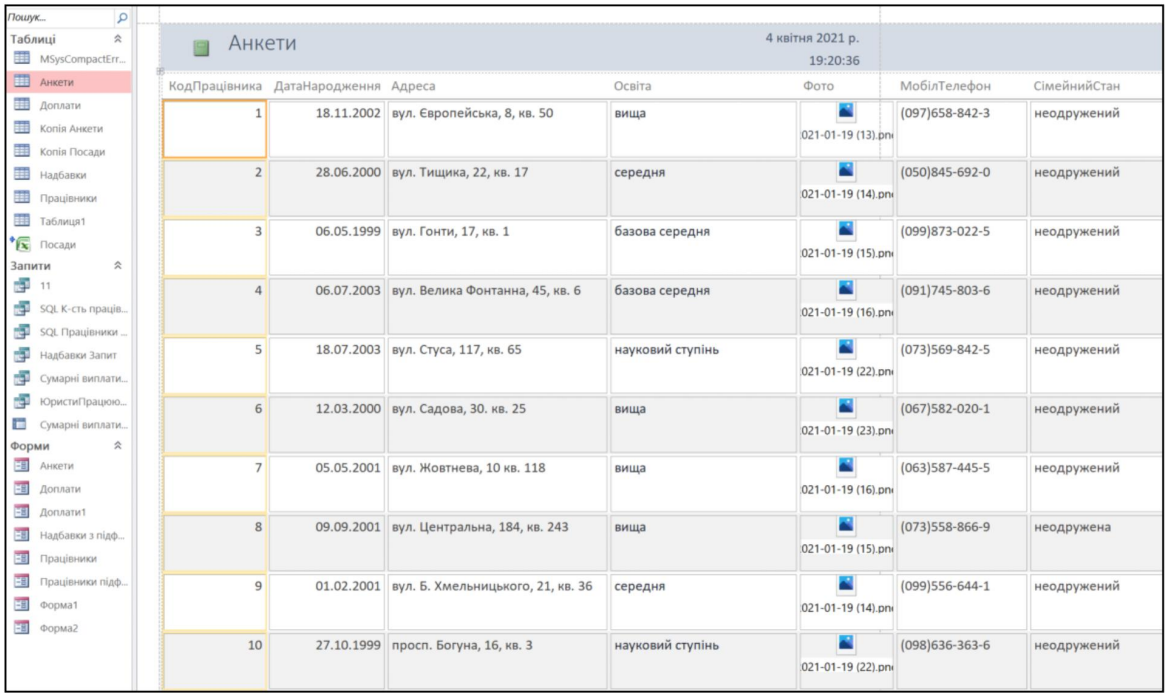


Рис. 7.6. Створений звіт *Анкети*

Конструктор звітів відкриває порожній звіт у режимі конструктора і дозволяє додати в нього потрібні поля й елементи керування.

Пустий звіт відкриває порожній звіт у режимі розмітки і відкриває список полів, за допомогою якого можна додавати поля до звіту. При перетягуванні поля на звіт створюється вбудований запит.

Етикетки викликає майстер, в якому можна вибирати стандартний або настроюваний розмір підписів, а також указувати, які поля потрібно відображати та як їх слід сортувати. Як наслідок майстер створить звіт з підписами на базі вибраних параметрів (Рис. 7.7).

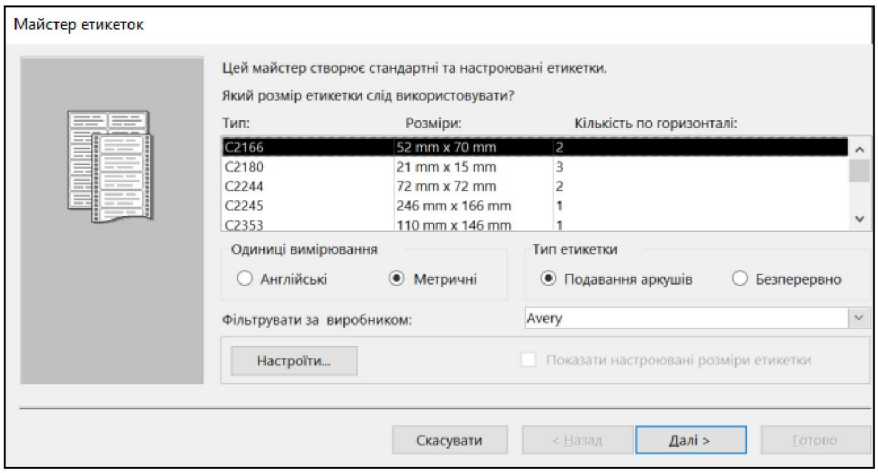


Рис. 7.7. Майстер етикеток

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.

Щоб за допомогою *Майстра звітів* створити звіт *Працівники* з групуванням по відділах слід клацнути в області переходів на таблиці *Працівники*, не відкриваючи її, та виконати команду *Майстер звітів* на вкладці *Створення* у групі *Звіти.* У діалоговому вікні *Майстер звітів* вибрати поля у такій послідовності: 1) *Відділ*; 2) *Прізвище*; 3) *Ім'я*; 4) *Посада*. Натиснути кнопку *Далі* (Рис. 7.8).

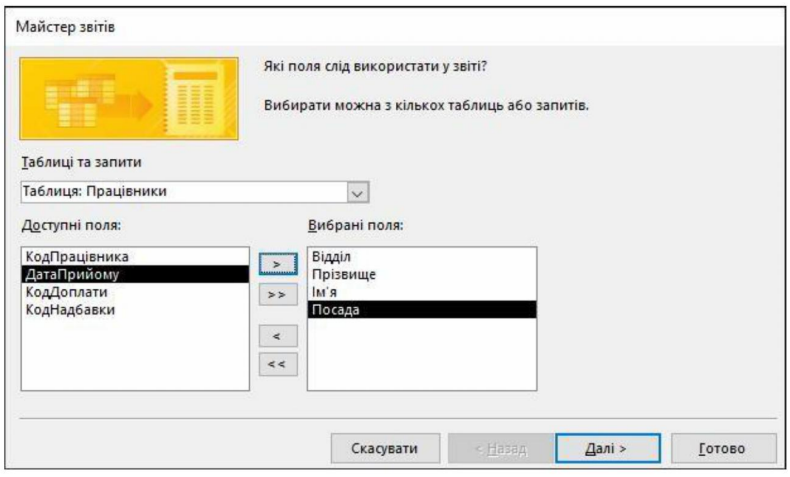


Рис. 7.8. Майстер звітів

На наступному кроці додати рівень групування – поле *Відділ* перенести кнопкою > (Рис. 7.9).

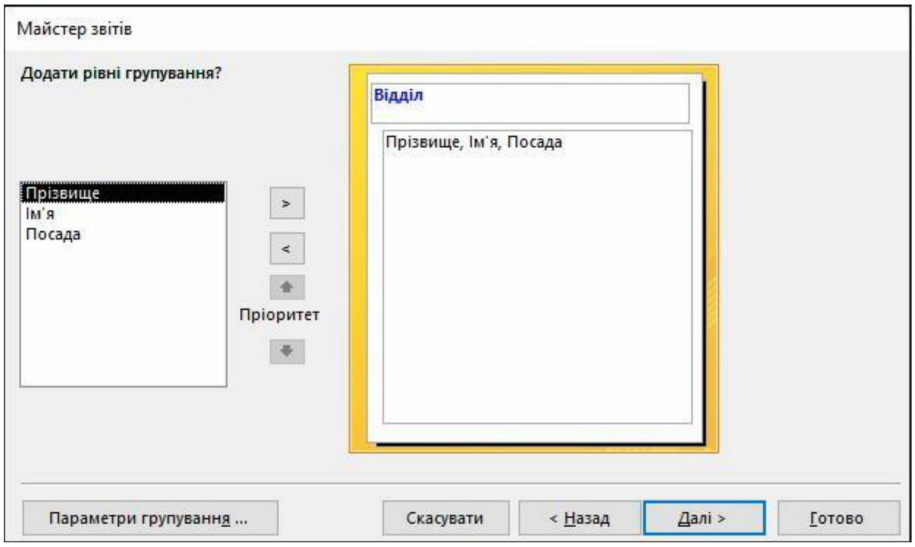


Рис. 7.9. Додавання рівнів групування

Натиснути кнопку *Готово* або *Далі*, щоб на наступних двох кроках задати додаткове сортування та зовнішній вигляд звіту. Перейти до режиму *Конструктор* (Рис. 7.10), за потреби зменшити висоту поля *Посада*, щоб відображалось лише одне значення (а не весь список), і зменшити висоту області даних *Подробиці*.

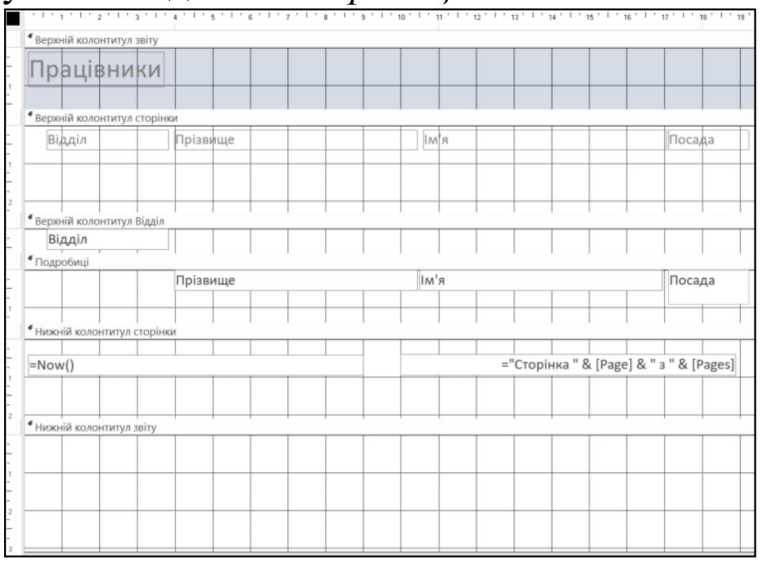


Рис. 7.10. Відображення звіту в режимі *Конструктора*

Щоб додати обчислення кількості працівників для кожного з відділів та загальну кількість працівників на підприємстві, слід виокремити поле *Працівники* в області даних *Подробиці* та на панелі *Знаряддя конструктора звітів* на вкладці *Конструктор* у групі *Групування та підсумки* вибрати команду *Підсумки – Кількість записів* (Рис. 7.11).

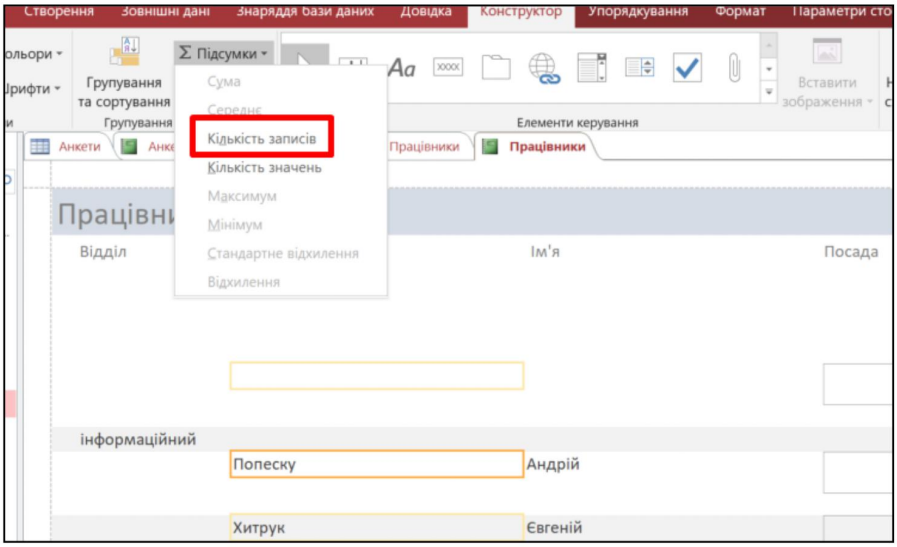


Рис. 7.11. Обчислення кількості працівників

Після цього слід перейти до режиму *Конструктор*, вибрати елемент керування *Підпис*, розмістити його в область *Нижній колонтитул Відділ* ліворуч від обчисленого поля *=Count(*)* та вписати текст *Всього працівників у відділі*. За потреби можна перемістити праворуч обчислене поле (вирізати та вставити). Подібний підпис з текстом *Всього працівників* створити і в області *Нижній колонтитул звіту* (Рис. 7.12).

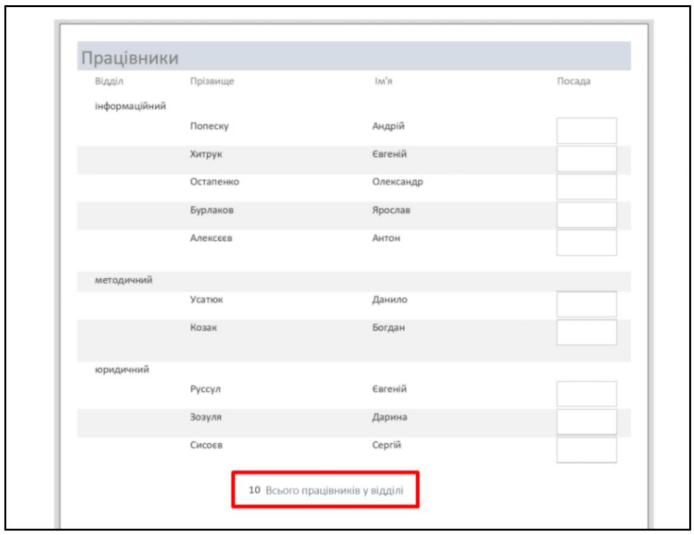


Рис. 7.12. Елемент *Підпис*

7.4. Послідовність створення звіту у програмі Access

Виконавши описані нижче кроки, можна створити звіти для локальної бази даних Access.

1) В області переходів **вибрати (виділити) джерело записів для звіту** – таблицю або запит. Джерело записів має містити всі рядки та стовпці, дані з яких треба додати до звіту. Якщо джерело записів ще не створено, можна скористатись засобом *Пустий звіт* або ж створити таблицю (таблиці або запит) із потрібними даними.

2) **Вибрати спосіб для створення звітів**, натиснувши відповідну кнопку на вкладці *Створення* стрічки у групі *Звіти.* Якщо запуститься майстер, виконати його вказівки, а на останньому етапі натиснути кнопку *Готово*. Звіт буде відкрито у режимі розмітки.

3) **Відформатувати звіт**, щоб домогтися бажаного зовнішнього вигляду. Для цього можна змінювати розмір полів і підписів, виділяючи їх і перетягуючи їхні краї. Можна розташовувати поля в потрібному порядку, виділяючи їх (і відповідні підписи, якщо вони є) і перетягуючи в інше місце. Також можна за допомогою команд контекстного меню розділяти або поєднувати клітинки, вилучати або вибирати поля чи виконувати інші завдання форматування.

При бажанні на звіті можна задати групування, сортування або підсумки, клацнувши правою кнопкою миші на полі, до якого треба застосувати відповідну операцію, і вибравши потрібну команду з контекстного меню. Окрім того, коли звіт відкрито в режимі розмітки або конструктора, функції групування, сортування й підсумків можна задати за допомогою засобів групи *Групування та підсумки* на вкладці *Конструктор* стрічки. Ці та інші засоби дозволять зробити звіт ще привабливішим і зручнішими для читання.

7.5. Виділення даних за допомогою умовного форматування

У програмі Access передбачені потужні функціональні інструменти для виділення даних у звітах. Для кожного елемента керування або їх групи можна задати до 50 правил умовного форматування, а у звітах можна використовувати для порівняння гістограми даних із різних записів. Додати умовне форматування можна в такий спосіб:

1) Відкрити звіт у режимі розмітки (подання макета).

2) Вибрати усі елементи керування, до яких потрібно застосувати умовне форматування, і потім на вкладці *Формат* у групі *Форматування елементів керування* натиснути кнопку *Умовне*

форматування.

3) У діалоговому вікні *Диспетчер правил умовного форматування* натиснути кнопку *Створити правило* (Рис. 7.13).

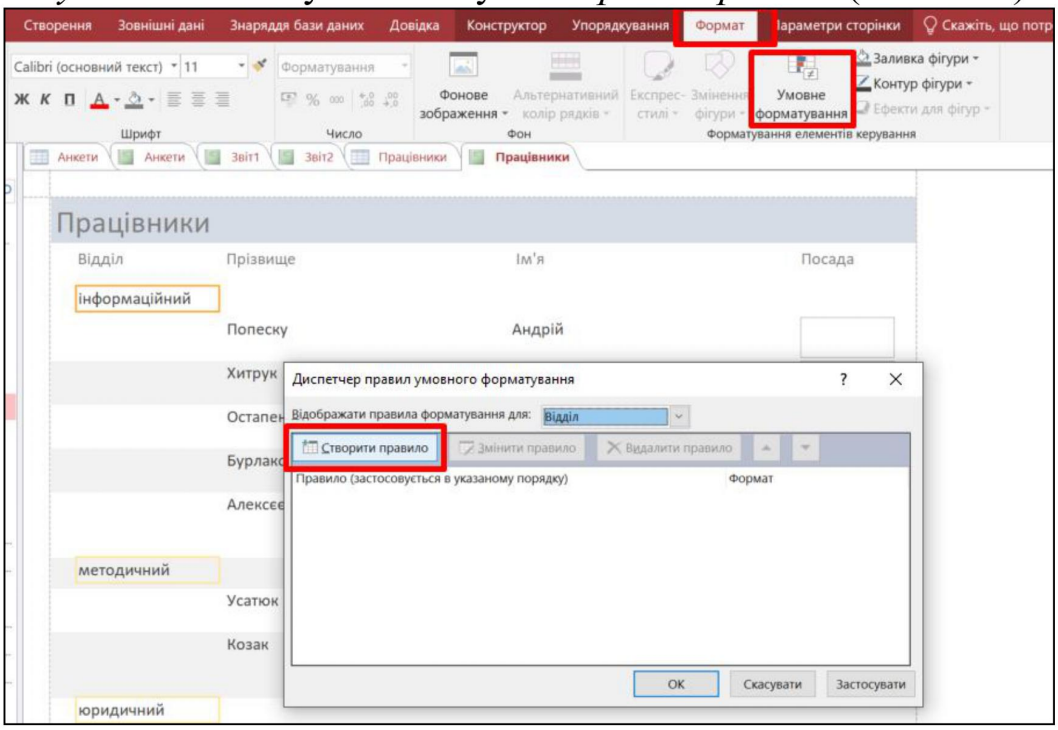


Рис. 7.13. Створення правила

4) У діалоговому вікні *Нове правило форматування* в області *Виберіть тип правила* вибрати потрібне значення:

– щоб створити правило, яке обчислюватиметься для кожного запису окремо, вибрати значення *Перевірити значення в поточному записі або використати вирази* (Рис. 7.14);

– щоб створити правило, яке порівнюватиме записи між собою за допомогою гістограм, вибрати значення *Порівняти з іншими записами*.

5) В області Змініть опис правила сформувати правило, яке визначатиме, коли слід застосовувати форматування та яким воно має бути, а потім натиснути кнопку *OK*.

6) Щоб для цього ж елемента керування або набору елементів керування створити додаткове правило, слід повторити цю процедуру, починаючи з кроку 4.

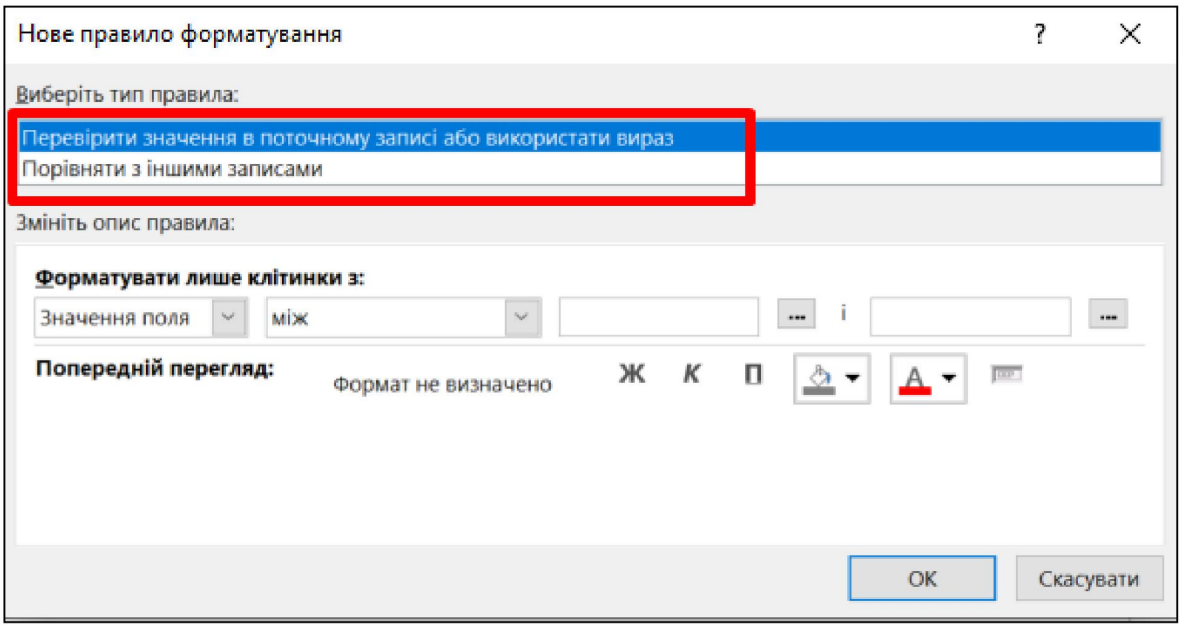


Рис. 7.14. Налаштування правила

7.6. Надання звітові професійного зовнішнього вигляду за допомогою тем

Access дає змогу до баз даних застосовувати теми, при цьому вибрана тема

Office, шрифт або колір буде застосована не лише до поточного звіту, а й до всіх форм та звітів у базі даних:

1) Відкрити звіт у режимі розмітки.

2) На вкладці *Знаряддя конструктора звітів / Конструктор* у групі *Теми* вибрати потрібну тему, колір або шрифт. З колекції *Теми* можна вибрати кольори та шрифти відповідно до готової схеми (Рис. 7.15).

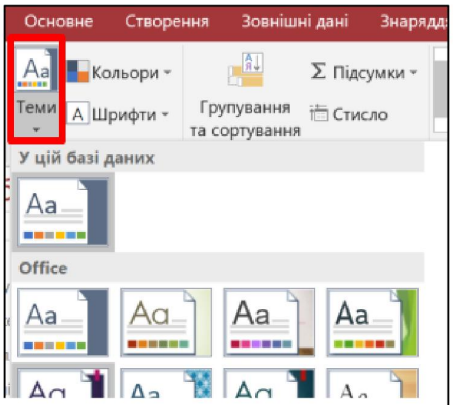


Рис. 7.15. Зміна зовнішнього вигляду

Підводячи покажчик миші до різних тем у колекції, можна переглянути ефекти. Щоб вибрати тему, слід клацнути її, а потім зберегти звіт. За допомогою колекцій *Кольори* і *Шрифти* можна задавати кольори і шрифти незалежно один від одного.

7.7. Додавання емблеми або зображення тла

Зображення в Access традиційно прив’язувалися до окремих елементів керування на формах і звітах. Щоб змінити часто використовуване зображення, присутнє на декількох формах і звітах,

кожен елемент керування доводилося оновлювати вручну. Тепер Access дозволяє прикріпити до бази даних зображення, а потім використовувати його з декількома об'єктами. При цьому оновлення цього зображення призведе до його оновлення в масштабах усієї бази даних. Це доволі зручно при роботі з такими об'єктами, як емблеми компаній і фонові зображення, використовувані в усій базі даних. Крім зображення емблеми, до звіту можна додати зображення тла. Послідовність дій:

1) В області переходів клацнути правою кнопкою миші на звіті, до якого потрібно додати зображення, і вибрати команду *Режим розмітки*.

2) Клацнути на те місце звіту, де потрібно розмістити зображення.

3) Цей крок залежить від виду зображення:

– для вставлення **зображення** на вкладці *Конструктор* у групі *Елементи керування* натиснути кнопку *Вставити зображення*;

– для вставлення **емблеми** на вкладці *Конструктор* у групі *Колонтитул* натиснути кнопку *Емблема*;

– для вставлення зображення **тла** на вкладці *Формат* у групі *Фон* натиснути кнопку *Фонове зображення*.

4) Вибрати зображення з колекції зображень або за допомогою кнопки *Огляд*. Після цього вибране зображення буде додано до звіту.

7.8. Попередній перегляд і друк звіту

Щоб виконати **попередній перегляд** звіту треба відкрити звіт, вибравши його в області переходів, і на вкладці *Файл* виконати команду *Друк*, а потім – *Попередній перегляд*. Після цього звіт буде відкрито у режимі попереднього перегляду. За допомогою команд на вкладці *Попередній перегляд* можна виконати такі дії:

– надрукувати звіт;
– змінити розмір або макет сторінки;
– змінити масштаб або переглянути декілька сторінок;
– оновити дані у звіті;
– експортувати звіт до файла іншого формату.

Щоб повернутися в робочу область бази даних, на вкладці *Попередній перегляд* у групі *Закрити* треба вибрати команду *Закрити вікно.*

Друк звіту можна виконати не тільки з режиму попереднього перегляду, а й за допомогою команди *Друк* на вкладці *Файл*. При цьому

можна буде задати додаткові параметри друку, наприклад кількість сторінок і копій, а також указати принтер.

Щоб надіслати звіт на принтер за замовчуванням не змінюючи параметри друку, можна скористатися командою *Файл – Швидкий друк* або клацнути правою кнопкою миші звіт в області переходів і вибрати команду *Друк* (Рис. 7.16).

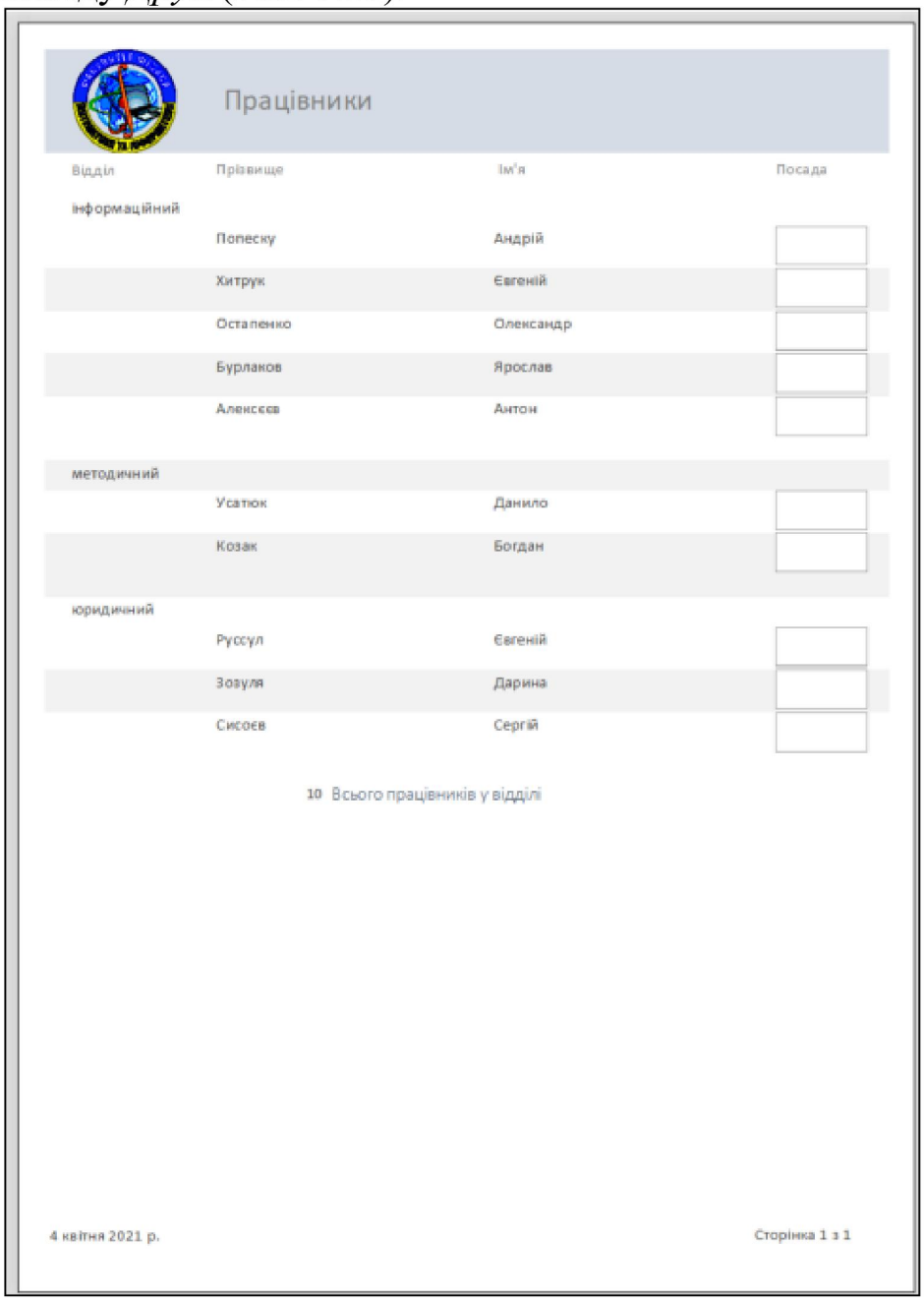


Рис. 7.16. Готовий звіт

Лекція 8. СТВОРЕННЯ МАКРОСІВ

8.1. Засоби програмування в Access

Створення бази даних зазвичай починається зі створення декількох об'єктів, наприклад таблиць, форм і звітів, але рано чи пізно доводиться вдаватися до засобів програмування, щоб автоматизувати деякі процеси та пов'язати об'єкти бази даних.

В Access програмуванням називається процес розширення функціональності бази даних з використанням ***макросів*** або ***модулів***. **Макроси** – це невеликі програми ***мовою макрокоманд*** (мовою сценаріїв), інакше кажучи, це набір готових дій, які можна вибирати і призначати тим або іншим об'єктам на формах і звітах за допомогою конструктора макросів. **Модулі** – це набори процедур ***мовою програмування VBA*** (Visual Basic for Applications), тобто це об'єкти, які містять програми мовою Visual Basic. Припустимо, що користувач створив форму і звіт та прагне додати на форму кнопку, при натисканні якої буде відкриватися звіт. У цьому разі програмування буде включати створення макросу або процедури VBA і налаштування події *OnClick* для кнопки таким чином, щоб при її натисканні запускався відповідний макрос або процедура. Для реалізації простих операцій, таких як відкривання звіту, можна використовувати майстер кнопок, але можна відключити його і самостійно запрограмувати все на VBA. Відтак, на відміну від інших програм Microsoft Office, таких як Word чи Excel, в яких макросом називається код VBA, в Access існує відмінність між поняттями макросу і модуля.

Основним призначенням макросів і модулів в Access є створення зручного інтерфейсу програми, в якому форми і звіти відкривалися б при натисканні кнопок на формах або на панелях інструментів.

Модулі є більш потужним засобом створення програмних розширень у середовищі Microsoft Office. Застосування модулів вимагає від користувачів знань основних принципів об'єктно-орієнтованого програмування. У свою чергу, конструктор макросів має більш структурований інтерфейс, ніж редактор VBA, дозволяючи додавати програмний код до елементів керування та об'єктів без знань особливостей програмування VBA, оскільки кожна дія (команда макросу) просто вибирається зі списку. Саме тому більшість користувачів надають перевагу створенню саме макросів, а ніж написанню коду VBA.

Прийняття рішення щодо використання макросів, VBA-коду або і

того, й іншого залежить в першу чергу від того, де буде використовуватися база даних, і рішення слід приймати на основі вимог до безпеки і функціональності додатка. Наприклад, якщо база даних буде використовуватися одним користувачем на одному комп'ютері, можна виконати більшість завдань програмування за допомогою VBA. Якщо ж планується розмістити базу даних на сервері і спільно використовувати її з іншими користувачами, краще відмовитися від застосування VBA з міркувань безпеки. Також треба враховувати ту особливість, що код VBA несумісний з функцією веб-публікації, і, якщо планується публікація бази даних у якості веб-додатка Access, для виконання завдань програмування доцільно використовувати макроси замість коду VBA.

В якості рекомендацій забезпечення безпеки бази даних можна порадити розробнику БД віддавати перевагу використанню макросів і вдаватися до коду VBA для виконання лише тих операцій, які неможна виконати за допомогою макрокоманд. Крім того, слід намагатися використовувати тільки ті дії, для виконання яких не потрібно висловлювати довіру до БД. Таке обмеження на використання макрокоманд дозволяє користувачам бути впевненими в тому, що БД не містить програмні засоби, які можуть зашкодити даним та іншим файлам на комп'ютерах.

У свою чергу, без модулів VBA не обійтися при виконанні деяких дій на рівні системи, наприклад, за потреби запуску з Access іншої програми – Word чи Excel, оскільки макрос неможна використовувати для виконання операцій поза Access. За допомогою VBA можна не лише запустити відповідну програму, а ще й перевірити наявність файла на комп'ютері, використовувати автоматизацію або динамічний обмін даними (DDE) для взаємодії або викликати функцію з бібліотеки динамічного компонування (DLL) Windows.

8.2. Специфіка використання макросів та засоби їх створення

Макрос – іменований набір макрокоманд для виконання певних операцій, що дозволяє автоматизувати роботу додатка. Використовуючи макроси для виконання рутинних операцій, можна істотно заощадити час і сили. Крім того, оскільки всякий раз при запуску макросу буде здійснюватися одна і та сама послідовність дій, макрос зробить роботу з базою даних більш ефективною і технологічною. Макроси можуть бути як простими, що складаються з однієї дії, так і складними, що включають логічні операції і кілька

взаємопов’язаних дій.

Об’єктам (наприклад, формам, звітам) і елементам керування на них (наприклад, кнопкам, текстовим полям) можна на виконання різних подій призначати різні макроси. Прикладами подій можуть бути клацання кнопки миші, відкривання або закривання форми, запуск звіту або змінення даних у текстовому полі тощо. Наприклад, якщо до форми додати кнопку, подія *OnClick* кнопки пов’язуватиметься з макросом, що містить команди, які мають виконуватися при кожному натисканні кнопки.

Події можуть запускатися зовнішніми по відношенню до Access факторами, такими як системні події, макроси або процедури, пов’язані з іншими подіями. Якщо додати значну кількість макросів до подій різних об’єктів на формах і звітах, база даних може стати вельми складною, але в більшості випадків можна досягти необхідного результату майже без програмування.

Призначати макроси, пов’язані з подіями, можна за допомогою вікна властивостей форм та звітів (до речі, таблиці і запити подіями не володіють).

Починаючи з версії Access 2010, в програму включено багато нових дій (макрокоманд), які дозволяють створювати більш ефективні макроси, ніж у попередніх версіях Access. Наприклад, за допомогою макрокоманд можна створювати і використовувати тимчасові глобальні змінні, а також більш гнучко обробляти помилки. У попередніх версіях Access ці можливості були доступні тільки при використанні VBA. Крім того, можна вбудовувати макроси безпосередньо у властивості подій об’єктів або елементів керування. Вбудований макрос стає частиною об’єкта або елемента керування і залишається з ним при його переміщенні або копіюванні.

Для створення макросів за допомогою *Конструктора макросів*, який можна запустити командою *Макрос* на вкладці *Створення* у групі *Макроси та код* (Рис. 8.1).

Вибрати ту чи іншу макрокоманду можна з розкривного списку.

Створювані макроси можуть бути від початку пов’язані з якимось елементом, наприклад, з кнопкою на формі, або бути самостійними об’єктами, багаторазово використовуваними у різних об’єктах на різних формах і звітах, наприклад, один і той самий макрос може виконуватися при виборі пункту меню на формі або при натисканні кнопки на звіті.

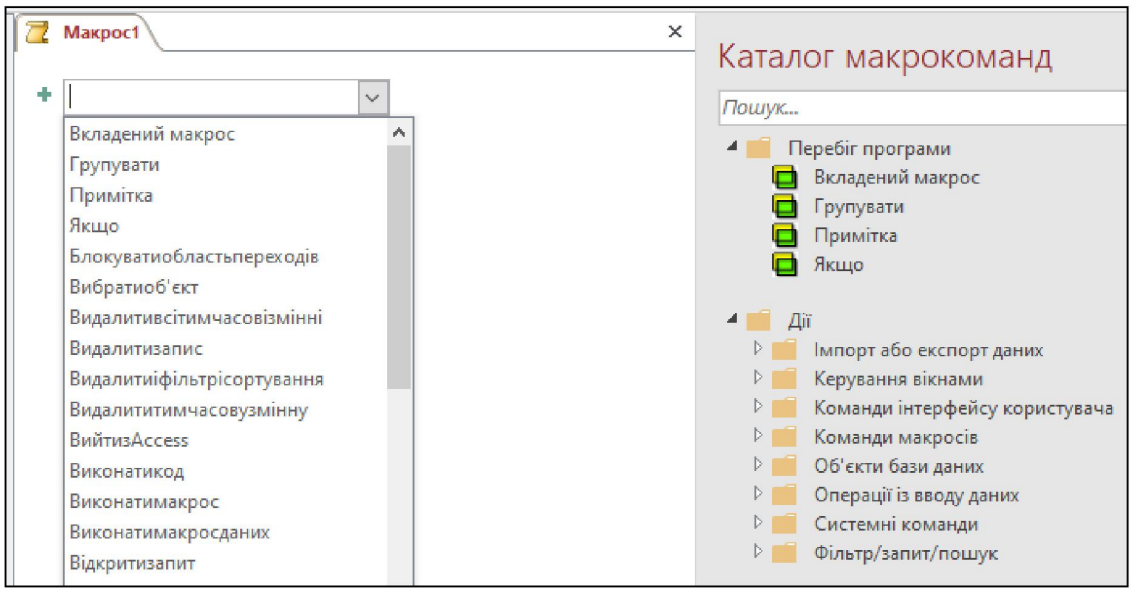


Рис. 8.1. Створення нового макросу

8.3. Створення командної кнопки з макросом

Послідовність дій для створення макросу розглянемо на прикладі створення форми *Ювіляри* на основі однойменного запиту та додавання на цю форму кнопки, при натисканні на яку відкриється форма *Працівники* з докладними анкетними даними працівника:

1. В області переходів натиснути один раз на запиті *Ювіляри* та виконати команду *Форма* на вкладці *Створення*. Після цього створиться та відкриється форма, яку треба зберегти з ім'ям *Ювіляри* (не закриваючи її).
2. Перейти до режиму *Конструктор* та збільшити ширину форми.
3. На вкладці *Конструктор* у групі *Елементи керування* (Рис. 8.2) клацнути стрілку вниз, щоб відобразити колекцію елементів керування, і вибрати елемент *Кнопка*, після чого розмістити кнопку на вільному місці форми.
4. При цьому відкриється вікно майстра створення кнопок.

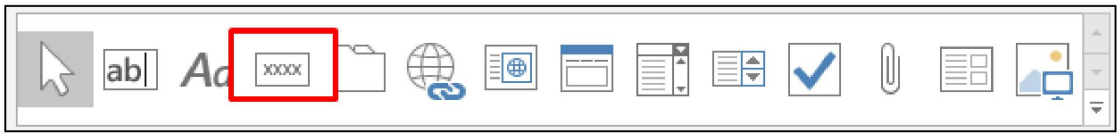


Рис. 8.2. Панель *Елементи керування*

5. На першій сторінці майстра кнопок в області *Категорії* вибрати пункт *Операції з формами*, а в області *Дії* вибрати *Відкрити форму* і натиснути кнопку *Далі* (Рис. 8.3).

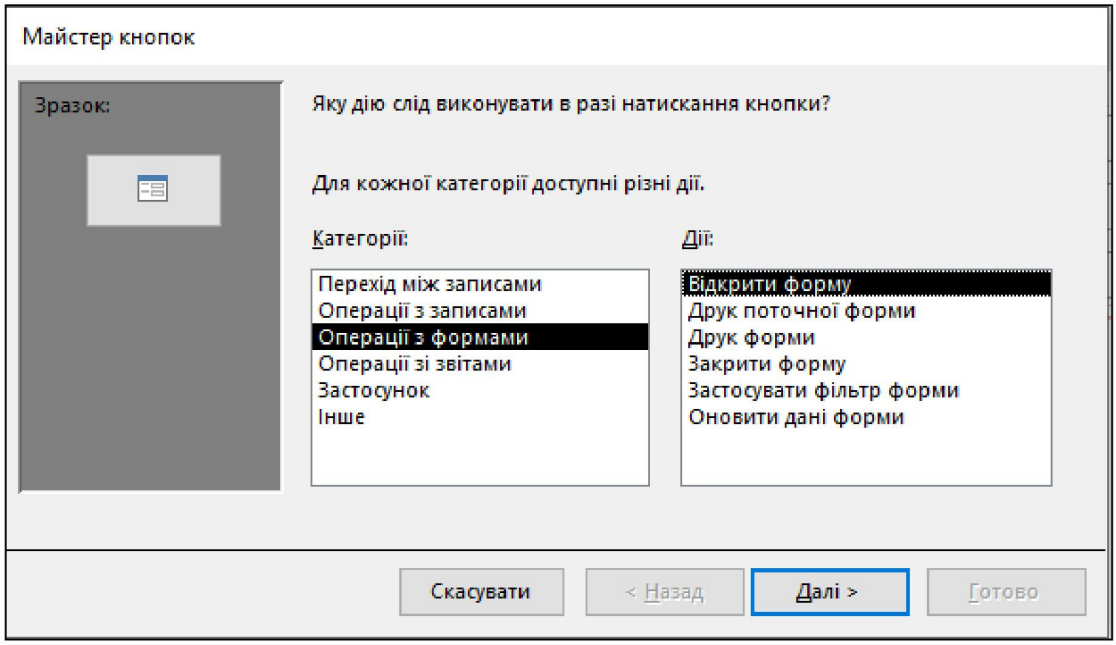


Рис. 8.3. Майстер створення кнопок

6. Вибрати назву відкритої форми *Працівники* і натиснути кнопку *Далі*.

7. На наступному кроці встановити опцію *Відкрити форму та знайти певні дані для відображення* і натиснути кнопку *Далі*.

8. Оскільки запит *Ювіляри* і таблиця *Працівники*, на основі яких створені обидві форми, не були безпосередньо пов'язані, треба вибрати поля, які слід використовувати для відбирання даних. Для цього вибрати поле *КодПрацівника* в області *Ювіляри* й однойменне поле в області *Працівники*, після чого натиснути кнопку <-> і кнопку *Далі*.

9. На наступному кроці вибрати опцію *Текст* та вписати текст, який буде відображатися на створюваній кнопці, наприклад: «Показати анкетні дані співробітника». Натиснути кнопку *Далі*.

Проте можна скористатися варіантом *Зображення*, який дозволить вибрати і відобразити на кнопці файл-малюнок формату *bmp* або *ico*.

10. Ввести ім'я кнопки *ВідкрПрацівники1* і натиснути кнопку *Готово*. Після цього кнопка буде вставлена на форму (Рис. 8.4).

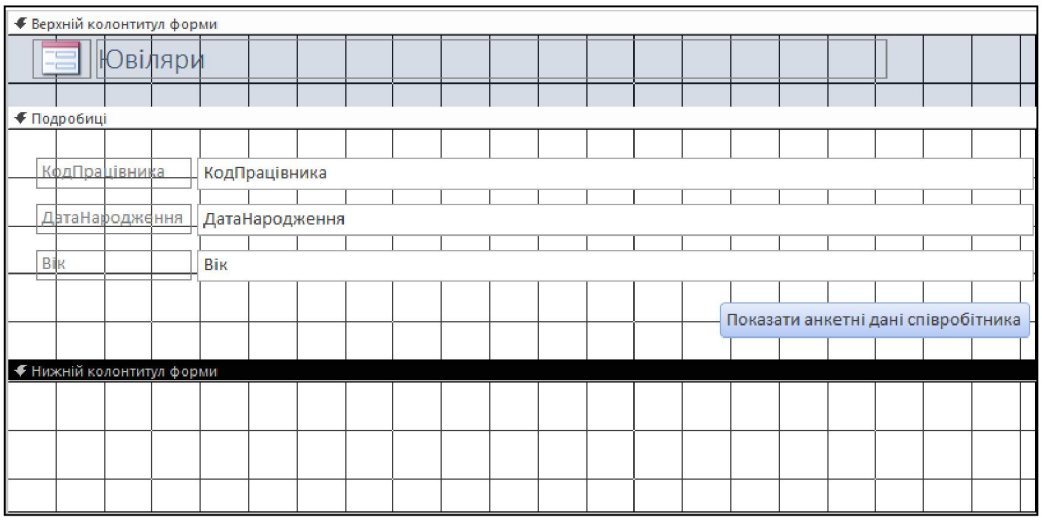


Рис. 8.4. Готова форма

11. Для додавання ще однієї дії, яка має виконуватися при натисканні цієї кнопки, треба в режимі *Конструктор* натиснути [F4] для відображення вікна властивостей. На вкладці *Подія* аркуша властивостей для події *Після клацання* натиснути кнопку побудови ..., що призведе до відкривання конструктора макросу. Далі треба за допомогою списку вибрати подальшу дію, наприклад *Закрити вікно*, і заповнити рядки у бланку макросу: у рядку *Тип об'єкта* вибрати *Форма*, а в рядку *Ім'я об'єкта* вибрати *Ювіляри* (Рис. 8.5).

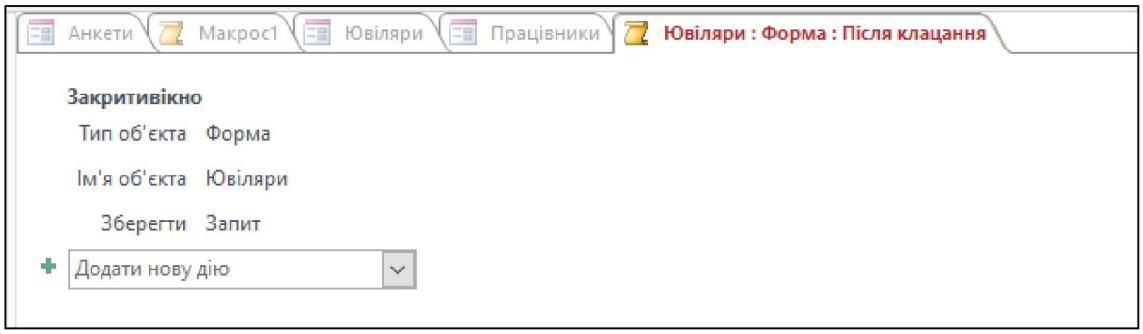


Рис. 8.5. Форма *Ювіляри*

12. Залишилося закрити макрос, перейти до режиму форми *Ювіляри* і натиснути на створену кнопку, щоб переконатися, що вона працює саме так, як очікувалось.

8.4. Створення самостійного (глобального) макросу

Макрос є об'єктом бази даних, який можна використовувати неодноразово в різних варіантах. До того як створити макрос, слід подумати щодо його подальше використання. Наприклад, перед тим,

як приступити до змінення даних в таблиці *Працівники*, слід попередити про це користувача, а потім відкрити потрібну таблицю. Отже, треба виконати дві дії (дві макрокоманди): 1) запустити вікно з попередженням; 2) відкрити потрібну таблицю.

Розглянемо порядок створення такого макросу, а потім використаємо цей макрос для форми *Ювіляри* і звіту *Працівники*:

1. На вкладці *Створення* у групі *Макроси та код* натиснути на значок *Макрос*.
2. У вікні *Макрос2* розкрити список і вибрати макрокоманду *Вікноповідомлення*, що призведе до відкривання бланка складання повідомлення. Наприклад, у рядку *Повідомлення* ввести текст «Буде відкрита таблиця, в яку можна вносити зміни» (без лапок), у рядку *Тип* вибрати зі списку *Попередження!*, у рядку *Назва* ввести «Увага!» (перша частина макросу готова) (Рис. 8.6).

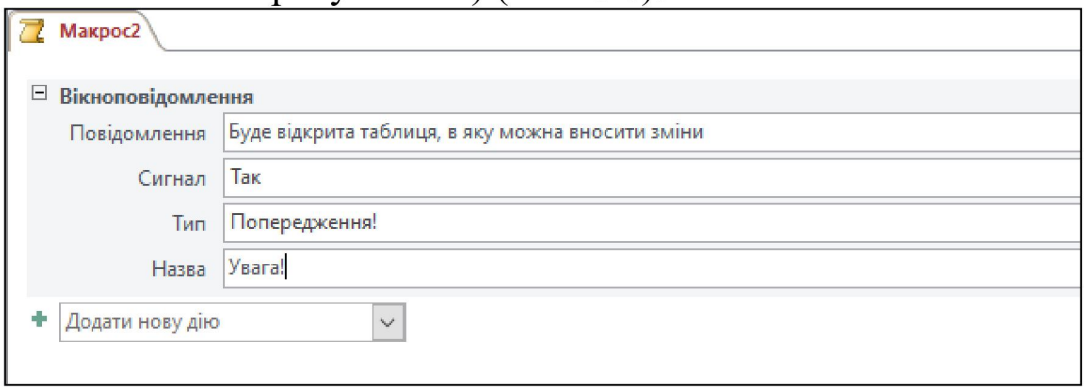


Рис. 8.6. Форма *Ювіляри*

3. Щоб додати до макросу ще одну макрокоманду, треба у вікні макросу зі списку 

вибрати дію *Відкрити таблицю*, після чого вибрати зі списку ім'я таблиці *Працівники*.

Сам макрос можна відкрити в режимі *Конструктор* і відкоригувати його.

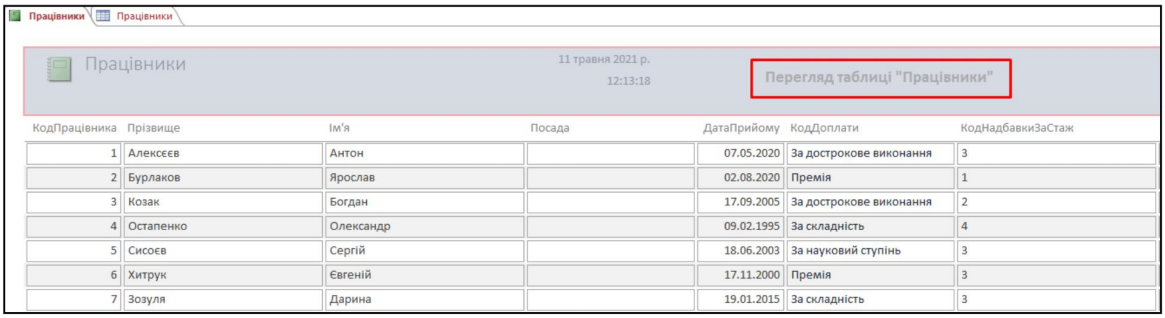
4. Зберегти макрос з ім'ям *ВідкрТаблПрацівники*, після чого в області переходів з'явиться зазначене ім'я створеного макросу.

Залишилося найголовніше, прив'язати макрос до якогось об'єкта (або об'єктів).

5. Створити спеціальний підпис у звіті *Працівники*, клацання по якому запустить створений макрос. Для цього треба відкрити цей звіт у режимі *Конструктор*, розмістити у верхньому колонтитулі звіту (заголовку) (чи іншому місці) новий елемент керування *Підпис* Aa. Відкрити вікно властивостей цього елемента, у полі *Підпис* ввести

текст «Перегляд таблиці «Працівники»». Також можна змінити інші властивості, наприклад, колір фону, розмір шрифту тощо.

Для прив'язки макросу до цього підпису треба на вкладці *Подія* у рядку *Після клацання* вибрати зі списку ім'я потрібного макросу *ВідкрТаблПрацівники*. Зберегти звіт, перейти до режиму *Подання звіту* та перевірити працездатність макросу (Рис. 8.7).



КодПрацівника	Прізвище	Ім'я	Посада	ДатаПрийому	КодДоплати	КодНадбавкиЗаСтаж
1	Алексєєв	Антон		07.05.2020	За дострокове виконання	3
2	Бурлаков	Ярослав		02.08.2020	Премія	1
3	Козак	Богдан		17.09.2005	За дострокове виконання	2
4	Остапенко	Олександр		09.02.1995	За складність	4
5	Сисоєв	Сергій		18.06.2003	За науковий ступінь	3
6	Хитрук	Євгеній		17.11.2000	Премія	3
7	Зозуля	Дарина		19.01.2015	За складність	3

Рис. 8.7. Перевірка працездатності макросу

6. Для ще одного використання цього макросу і для форми *Анкети*, слід відкрити форму в режимі *Конструктор* і розмістити на формі елемент керування 

Кнопка, що призведе до відкривання майстра створення кнопок. На першій сторінці майстра в області *Категорії* клацанням вибрати пункт *Інше*, зі списку *Дія* вибрати *Запустити макрос* і натиснути кнопку *Далі* (Рис. 8.8).

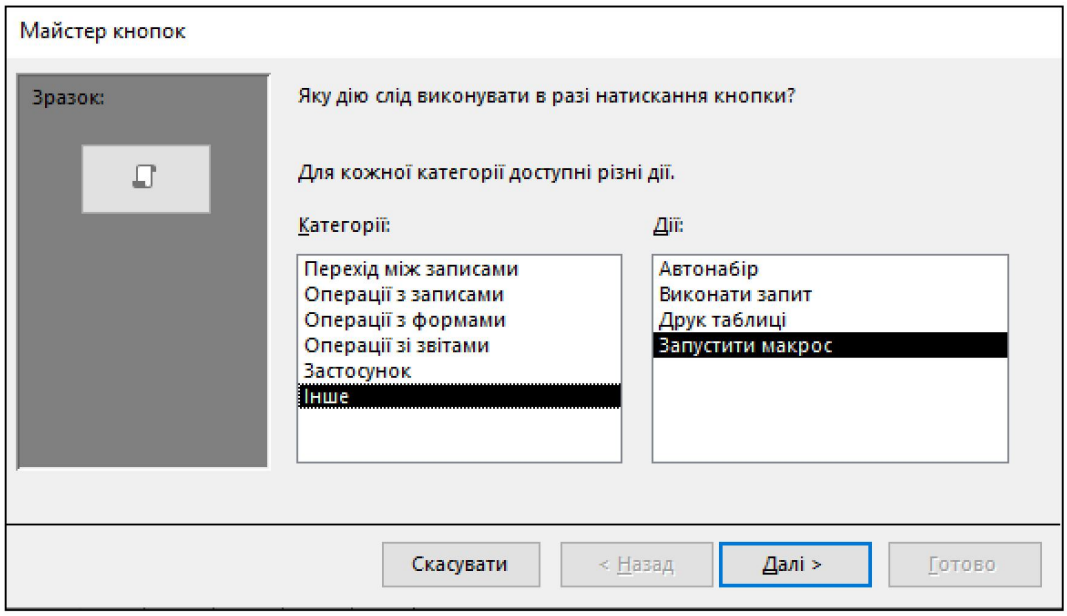


Рис. 8.8. Використання макросу

Після цього вибрати ім'я макросу *ВідкрТаблПрацівники* і натиснути кнопку *Далі*. Вибрати відповідний рисунок і/або вписати

текст «Відкрити таблицю «Працівники»», який буде відображатися на створюваній кнопці. Натиснути кнопку *Готово*. Залишилося перейти до режиму форми і перевірити правильність роботи макросу. Отже, зручністю використання макросу є ще й те, що він поширюється на всі типи об'єктів бази даних (Рис. 8.9).

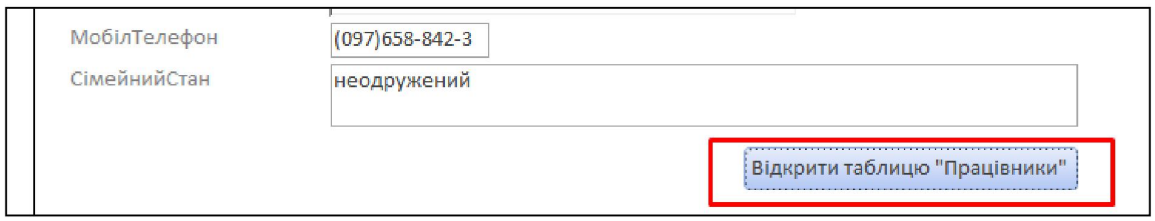


Рис. 8.9. Робота з таблицею *Працівники*

8.5. Створення складного макросу

Оскільки макроси в Access складаються зі стандартних дій (макрокоманд), то можна перекласти весь пошук необхідних даних на макроси замість того, щоб створювати складні умови відбору даних у запитах. Припустимо, керівництво організації цікавлять конкретні дані про працівників, які займають певні посади. Швидше за все, керівництво вже заплуталось у назвах посад на підприємстві, це пов'язано з постійною реорганізацією і створенням наказів щодо внесення змінень до штатного розкладу. Напевно, керівнику слід надати форму з переліком існуючих посад на підприємстві, а далі дати можливість за назвою посади віднайти прізвища працівників, їхнє місце роботи та інші відомості. Спробуємо створити взаємодію між формами за допомогою макросу.

Для розв'язання поставленої задачі потрібно дві форми. Перша форма (головна) міститиме список посад на певний момент часу в організації, а друга форма (підпорядкована форма, підформа) – дані, які цікавлять керівника. З підформи мають вибиратися дані, які задовольнятимуть умові, заданій керівником. Отже, зв'язок між формами покладається на програму, яка буде формуватися за допомогою макросів. Виконаємо кілька кроків для розв'язання поставленої задачі.

Крок 1. Розробка головної форми *Посади*:

1. Створити форму за таблицею *Посади*, виділивши відповідну таблицю в області переходів (не відкриваючи її) та вибравши на вкладці *Створення* у групі *Форми* команду *Майстер форм*. Далі треба вибрати обидва поля та на наступному кроці вибрати макет *Таблиця* (у попередніх версіях Access – вид форми *Стрічковий* (рос. *Ленточный*), а на останньому кроці задати ім'я форми – *Посади*. Ширину полів

налаштувати власноруч у режимі розмітки або конструктора. При відкриванні форма матиме такий вигляд.

2. Для внесення змінень до створеної форми потрібно перейти до режиму *Конструктор*, посунути праворуч поля в області *Подробиці* (даних), розмістити на вільному місці елемент керування *Кнопка*. На запуск майстра створення кнопок слід відреагувати натисканням кнопки *Скасувати*. У вікні властивостей ввести у властивості *Підпис* текст «Працівники на цій посаді».

В області форми *Нижній колонтитул* (або *Примітка*) розташувати елемент керування *Підпис*, у властивості *Підпис* якого ввести текст «Щоб переглянути список працівників, які займають відповідну посаду, потрібно натиснути кнопку ліворуч від цієї посади». При бажанні можна задати інші параметри форматування форми і полів на ній, після чого зберегти готову форму (Рис. 8.10).

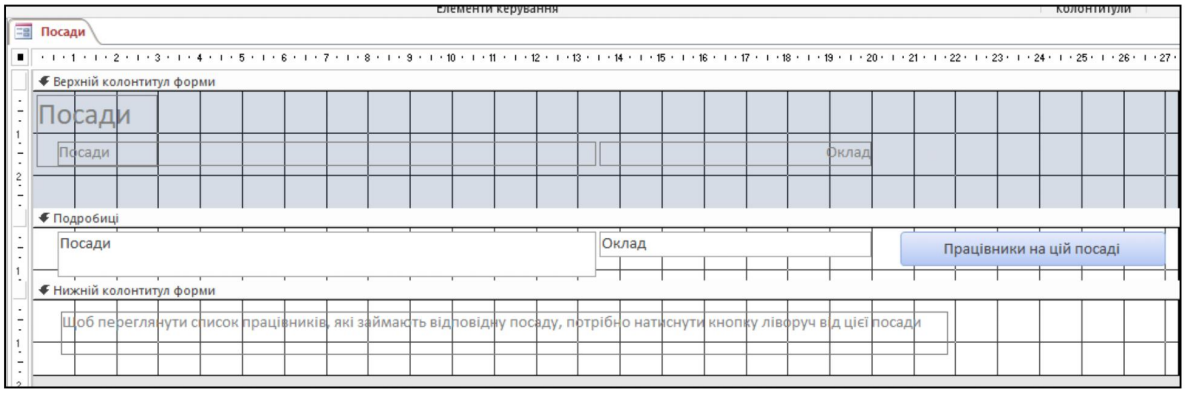


Рис. 8.10. Форматування макросу

3. Для створення макросу, який буде виконуватися при натисканні щойно створеної кнопки, треба для форми *Посади* у режимі *Конструктор* клацнути правою кнопкою миші на кнопці *Працівники на цій посаді* і вибрати з контекстного меню команду *Обробка подій*. У вікні вибрати параметр *Побудовник макросів* і натиснути кнопку *ОК*. Далі у вікні конструктора макросу треба зі списку вибрати дію *УстановитиТимчасовуЗмінну*.



У відкритому бланку макросу в рядку *Ім'я* ввести ім'я тимчасової змінної *Z*, за допомогою якої будуть передаватися дані з головної форми до підформи. Щоб це відбувалося, у рядку *Вираз* бланка слід вписати ім'я поля [*Посади*], значення якого будуть поєднувальною ланкою між головною формою і підформою. Причому, ім'я цього поля можна не вписувати, а вибрати зі списку полів, скориставшись кнопкою .

Закрити і зберегти бланк макросу.

Крок 2. Вибір підформи і пов'язування її з головною формою:

1. Вибір підформи можна здійснити з уже наявних форм або створити нову форму. У даному прикладі доцільно створити спеціальну форму з найменуванням *Анкетні дані*, яка міститиме поля з таблиць *Працівники* та *Анкети*. До речі, у ній обов'язково має бути поле *Посада*, оскільки у подальшому саме за його значенням буде відбуватися відбір записів для відображення (Рис. 8.11).

Перейти до режиму форми, зберегти та закрити її.

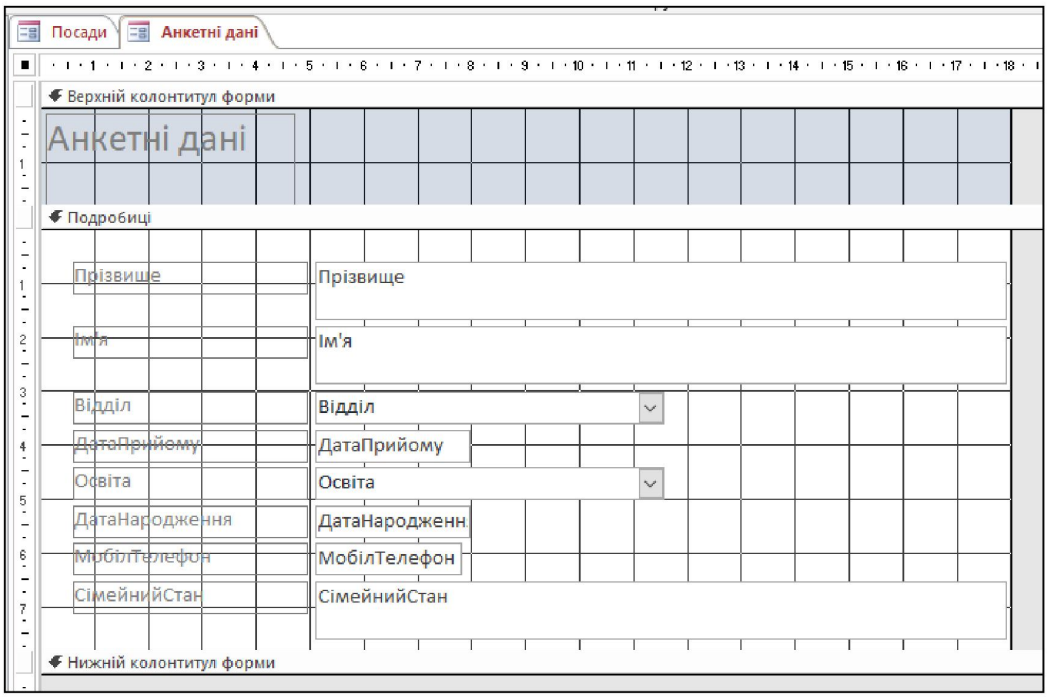


Рис. 8.11. Форматування макросу

2. Тепер можна продовжити процес створення макросу. Треба форму *Посади* відкрити у режимі *Конструктор*, натиснути кнопку *Працівники на цій посаді* й у вікні властивостей на вкладці *Подія* у рядку *Після клацання* натиснути значок ... . Це призведе до повторного відкривання бланка макросу.

3. Для додавання нової дії треба зі списку вибрати рядок з найменуванням *ВідкритиФорму*. У бланку макросу треба зі списку вибрати ім'я форми *Анкетні дані*, а в рядку *Умова відбору* ввести вираз: [*Посада*]=[*TempVars*]![*Z*] (Рис. 8.12).

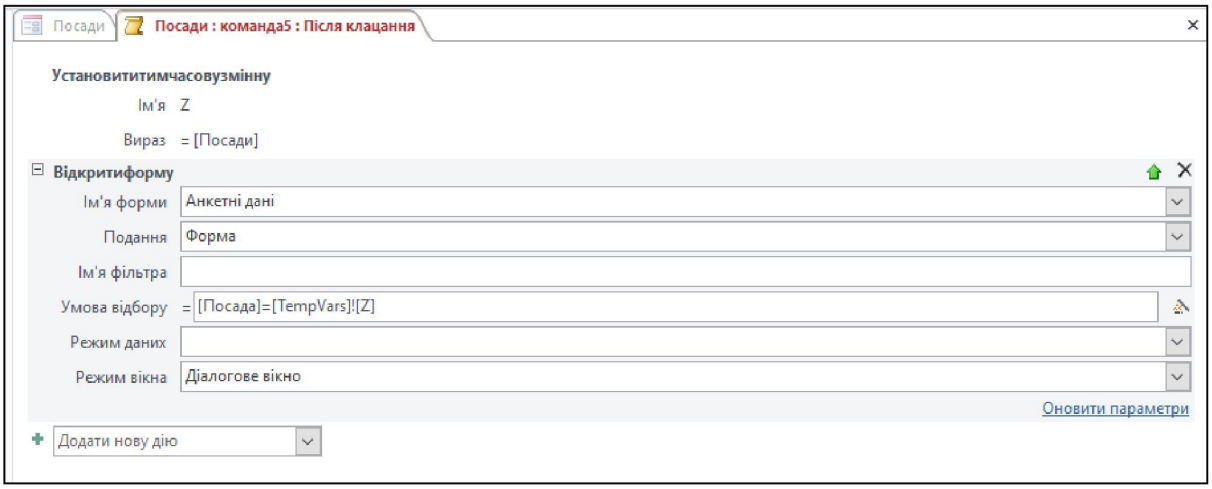


Рис. 8.12. Форматування макросу

У рядку *Режим вікна* вибрати зі списку значення *Діалогове вікно*. Це робиться для того, щоб користувач міг переглядати дані, зібрані з підпорядкованої форми, у відповідності до поставленої умови.

4. Зберегти макрос і закрити його Перевірити правильність робо ти кнопки *Працівники на цій посаді* на формі *Посади*.

Так, при клацанні по цій кнопці біля посади *Адміністратор* відкриється підформа *Анкетні дані* з відображенням даних про працівників саме на цій посаді (Рис. 8.13).

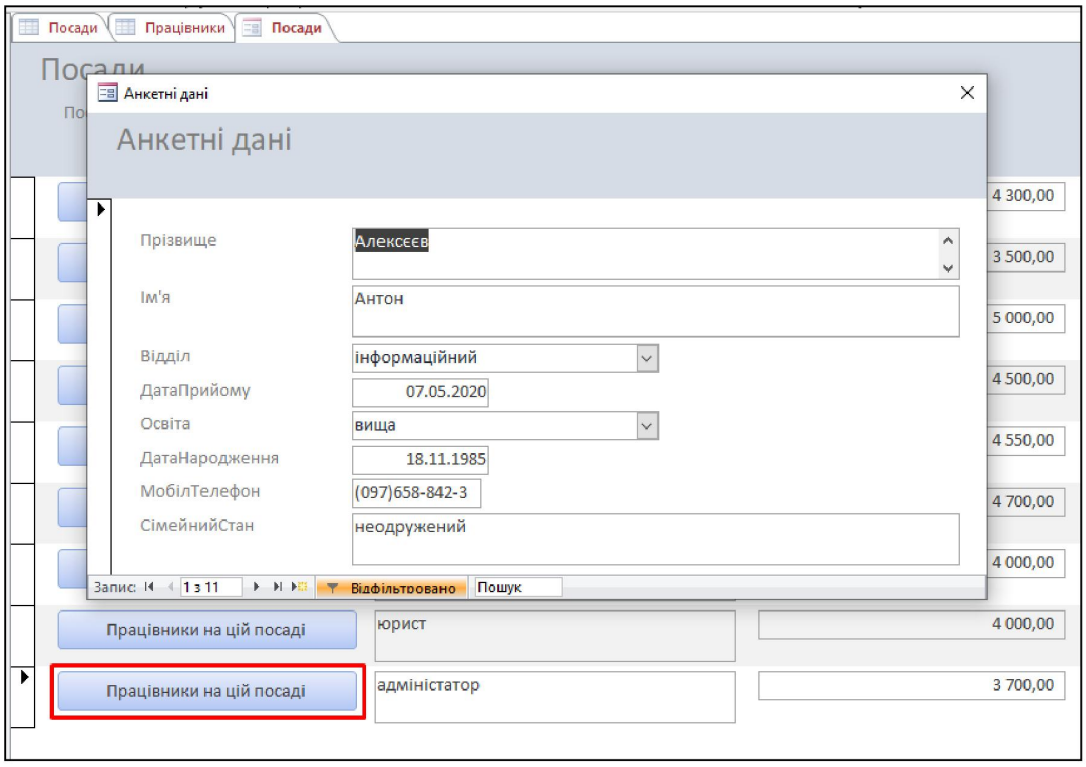


Рис. 8.13. Перевірка роботи макросу

Лекція 9. СТВОРЕННЯ ФОРМ НАВІГАЦІЇ БАЗИ ДАНИХ

9.1. Створення форм навігації бази даних

Форма навігації – це звичайна форма, яка містить навігаційні елементи керування і спрощує перемикання між різними формами та звітами в базі даних. Особливо важливу роль створення форми навігації відіграє в базі даних, якщо її планується публікувати в Інтернеті, оскільки в браузері область переходів Access не відображається.

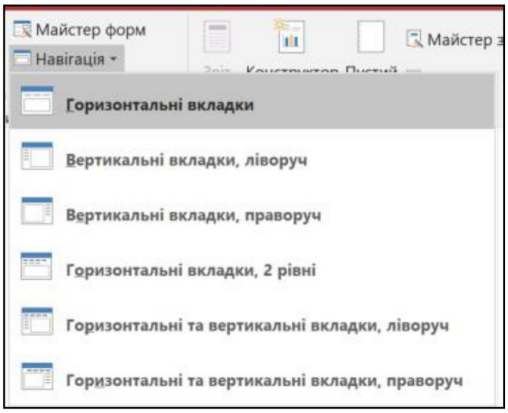


Рис. 9.1. Навігація

Для **створення форми навігації** слід на вкладці *Створення* у групі *Форми* вибрати команду *Навігація* та потрібний стиль форми навігації, наприклад: *горизонтальні та вертикальні вкладки, ліворуч*. Після цього Access створить форму з навігаційним елементом керування. Звичайно створена форма навігації відкривається у режимі розмітки (Рис. 9.1).

Якщо область переходів не відображається, слід натиснути клавішу [F11].

На робочому полі форми, а саме на полі [додати нову] натиснути праву кнопку миші та вибрати команду *Властивості*. Праворуч відкриється аркуш властивостей, де на вкладці *Формат* у полі *Підпис* слід записати назву вкладки, наприклад, *Працівники*. Перейти на вкладку *Дані* та в полі *Ім'я цільового об'єкта навігації* вибрати форму *Працівники*. Натиснути клавішу [Enter], щоб завершити змінення властивостей. Аналогічно слід винести решту форм Вашої БД на горизонтальні вкладки, а на вертикальні вкладки – звіти.

Крім того, додавати об'єкти на форму навігації можна і шляхом перетягування їх з панелі *Всі об'єкти Access*.

Після формування усіх потрібних кнопок слід перейти до режиму форми, щоб впевнитись у працездатності створеної форми навігації, натискаючи на відповідні кнопки.

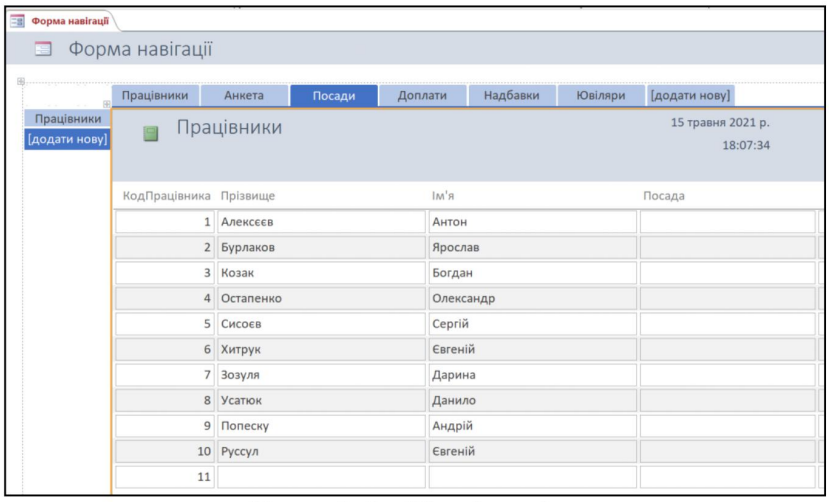


Рис. 9.2. Навігація

9.2. Редагування форми навігації

Під час створення форми навігації програма Access 2019 за замовчуванням додає до заголовка форми підпис *Форма навігації*. Щоб змінити підпис, треба відкрити форму навігації в поданні *Режим розмітки*, вибрати підпис заголовка форми, двічі його клацнувши. Відредагувати підпис та натиснути клавішу [Enter] (Рис. 9.2).

Для перейменування форми потрібно натиснути праву кнопку миші на заголовку форми і з контекстного меню вибрати команду *Властивості форми*. На вкладці *Формат* у полі *Підпис* записати нове ім’я форми навігації *База даних Прізвище* і натиснути клавішу [Enter]. Закрити вікно властивостей форми. Після цього у списку об’єктів бази даних у розділі *Форми* з’явиться форма *База даних Прізвище*.

Для змінення оформлення форми навігації можна застосовувати готові теми Office, які дають змогу швидко змінювати всі кольори та шрифти бази даних. Для вибору теми треба відкрити форму навігації в режимі розмітки та у розділі *Знаряддя для макетів форм* перейти на вкладку *Конструктор* та вибрати уподобану тему командою *Теми*.

Щоб змінити тільки кольори фону, не зачіпаючи налаштування шрифту, потрібно скористатись колекцією *Кольори* на вкладці *Конструктор* у групі *Теми*. Щоб змінити тільки параметри шрифтів, не зачіпаючи кольори, потрібно вибрати уподобаний шрифт з колекції *Шрифти*.

Крім того, існує ціла низка експрес-стилів для навігаційних кнопок, доступ до яких здійснюється командою *Експрес-стилі* на вкладці *Формат* у групі *Форматування елементів керування* (Рис. 9.3)

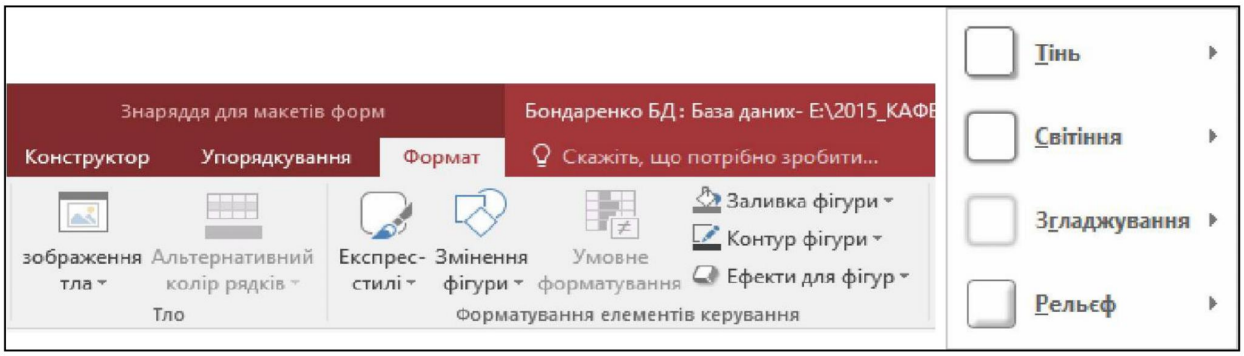


Рис. 9.3. Форматування елементів керування

Можна змінити фігуру кнопки, натиснувши кнопку *Змінення фігури*. Можна також змінити колір заливки, натиснувши кнопку *Заливка фігури*, або натиснувши кнопку *Контур фігури*. Також можна застосовувати різноманітні ефекти тіні, світіння, рельєфу або згладжування .

9.3. Налаштування параметрів відображення форми навігації

Форма навігації використовується у базах даних як «домашня сторінка», тому доцільно увімкнути її відображення за замовчуванням при відкриванні бази даних. Для цього слід виконати так у послідовність дій.

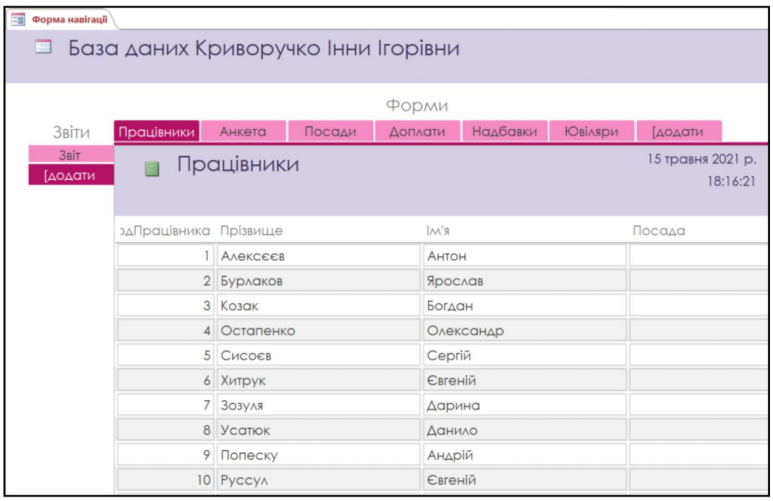


Рис. 9.4. Готова база даних

На вкладці *Файл* вибрати пункт *Параметри*. У вікні *Параметри Access* вибрати пункт *Поточна база даних*. В області *Параметри застосунку* вибрати свою форму навігації зі списку *Форма перегляду*, щоб призначити її за замовчуванням, та натиснути кнопку *ОК*. На екрані з’явиться повідомлення про необхідність перезапуску бази даних. Виконати перезапуск своєї бази даних (Рис. 9.4).

РОЗДІЛ ІІ. ЛАБОРАТОРНИЙ БЛОК

Лабораторна робота № 1

Завдання

1. Запустити програму MS Access та створити у своїй папці базу даних (БД) з ім'ям *ПрізвищеБД*.

2. У режимі *Таблиця* створити таблицю *Надбавки* з розмірами тарифів, залежно від стажу роботи працівників фірми, з трьома полями:

– поле *Код* – тип даних *Автонумерація* (ключове поле);
– поле *Стаж* – тип даних *Короткий текст*;
– поле *Надбавка*
– тип даних *Грошова одиниця*.

Заповнити таблицю даними (Рис. 1.1):

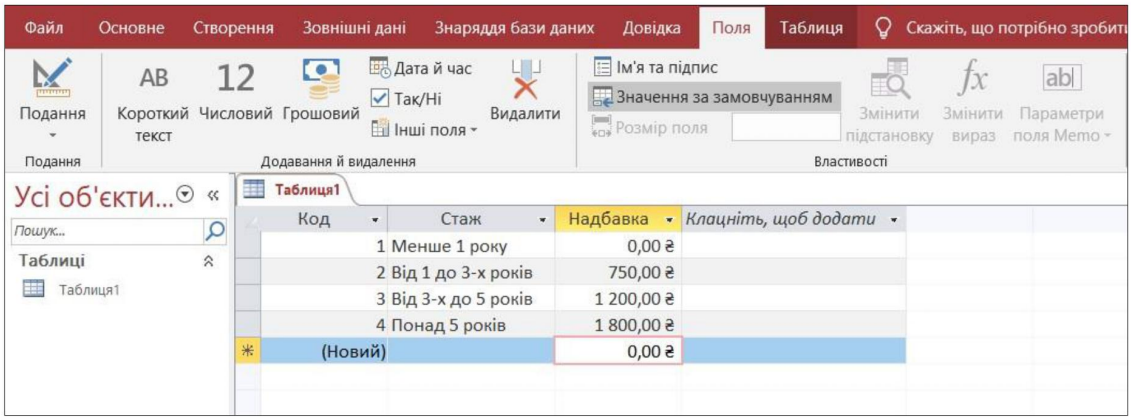


Рис. 1.1. Створення таблиці *Надбавки* з розмірами залежно від стажу роботи працівників фірми у режимі *Таблиця*

3. У режимі *Конструктор таблиць* створити таблицю *Працівники* з такими полями (Рис. 1.2):



Рис. 1.2. Вигляд вікна при створенні полів таблиці *Працівники* у режимі *Конструктор таблиць*

Після створення полів, вибору типів і опису полів задати такі властивості полів таблиці:

– для поля *Прізвище* (тип даних *Короткий текст*):

– властивість *Розмір поля* – значення 30 (як максимальна кількість символів у прізвищах працівників);

– властивість *Обов'язково* – значення *Так*;

– для поля *Ім'я* (тип даних *Короткий текст*):

– властивість *Розмір поля* – значення 15;

– властивість *Обов'язково* – значення *Так*;

– для поля *Посада* (тип даних *Короткий текст*) властивість *Розмір поля* – значення 25;

– для поля *ДатаПрийому* (тип даних *Дата й час*):

– властивість *Обов'язково* – значення *Так*;

– властивість *Текст перевірки* – "Вкажіть дату прийому на роботу".

– для полів *КодДоплати* і *КодНадбавкиЗаСтаж* (тип *Число*):

– властивість *Розмір поля* – значення *Довге ціле*;

– властивість *Обов'язково* – значення *Так*.

Заповнити таблицю (крім поля *Посада*) довільними даними (не менше десяти записів), при чому значення прізвищ, імен та по батькові мають відповідати даним ваших колег по групі, а першим у списку зазначити власне прізвище, ім'я та по батькові (Рис. 1.3):

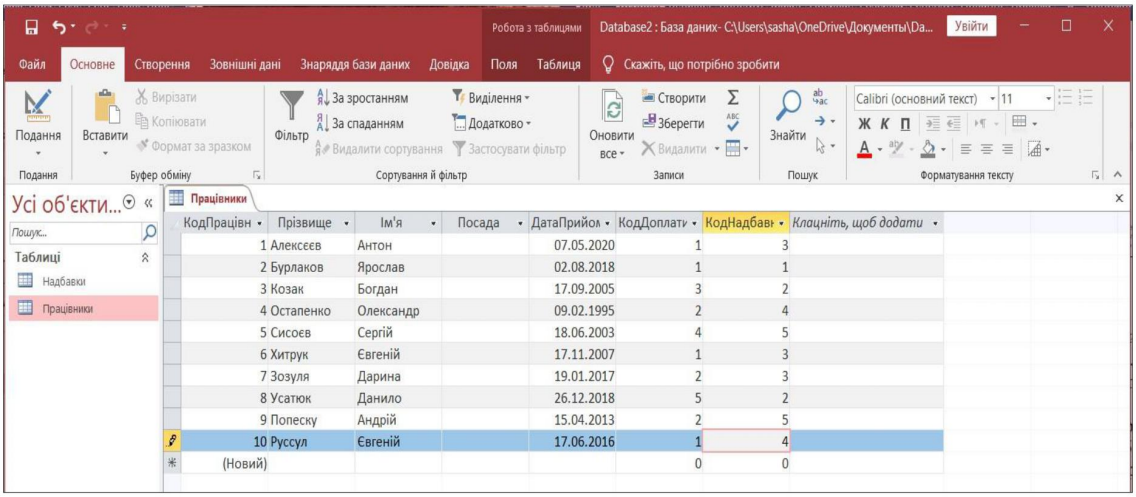


Рис. 1.3. Заповнення таблиці (крім поля *Посада*) довільними даними

Створити у цій само БД третю таблицю *Анкети* з даними особових справ працівників організації з такими полями (Рис. 1.4):

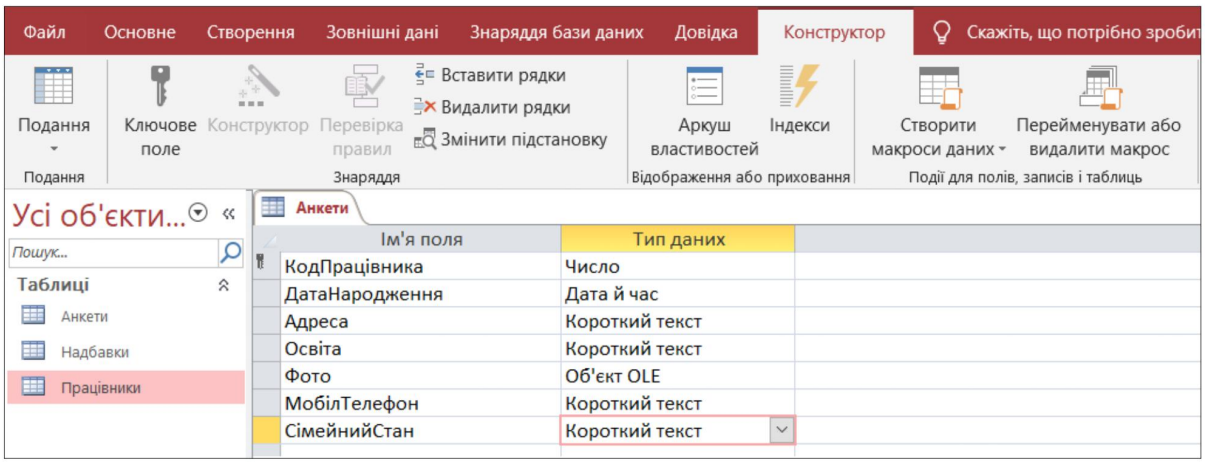


Рис. 1.4. Створення таблиці *Анкети*

код працівника, дата народження, адреса, освіта, телефон, фотографія, телефон, сімейний стан:

Після створення полів і вибору типів даних задати такі властивості:

– для ключового поля *КодПрацівника* (тип даних *Число*) задати для властивості *Розмір поля* значення *Довге ціле*;

– для поля *МобілТелефон* (тип *Короткий текст*):

– властивість *Маска вводу* – значення (###)###-###-##;

– властивість *Розмір поля* – значення 10.

Заповнити таблицю (окрім поля *Освіта*) довільними даними (не менше десяти записів), при чому для значень поля *Фото* треба за допомогою будь-якого графічного редактора відкрити графічний файл, виконати команду копіювання, перейти у клітинку поля *Фото* таблиці БД відповідного працівника і виконати команду вставлення. Необов'язкові для заповнення поля можна заповнювати не повністю (Рис. 1.5):

Анкети

КодПрацівн	ДатаНарод	Адреса	Освіта	Фото	МобілТелефон	СімейнийСт
1	18.11.2002	вул. Європейська, 8, кв. 50	вища	Package	(097)658-842-3	неодружений
2	28.06.2000	вул. Тищика, 22, кв. 17	середня	Package	(050)845-692-0	неодружений
3	06.05.1999	вул. Гонти, 17, кв. 1	базова середн	Package	(099)873-022-5	неодружений
4	06.07.2003	вул. Велика Фонтанна, 45, кв. 6	базова середн	Package	(091)745-803-6	неодружений
5	18.07.2003	вул. Стуса, 117, кв. 65	науковий ступ	Package	(073)569-842-5	неодружений
6	12.03.2000	вул. Садова, 30. кв. 25	вища	Package	(067)582-020-1	неодружений
7	05.05.2001	вул. Жовтнева, 10 кв. 118	вища	Package	(063)587-445-5	неодружений
8	09.09.2001	вул. Центральна, 184, кв. 243	вища	Package	(073)558-866-9	незаміжня
9	01.02.2001	вул. Б. Хмельницького, 21, кв. 36	середня	Package	(099)556-644-1	неодружений
10	27.10.1999	просп. Богуна, 16, кв. 3	науковий ступ	Package	(098)636-363-6	неодружений
11						

Рис. 1.5. Заповнена таблиця *Анкети*

Питання для самоконтролю

1. Назвати та охарактеризувати всі основні об’єкти Access.
2. Що таке шаблони БД Access і для чого вони призначені?
3. Як можна задавати типи даних полів таблиці БД?
4. Охарактеризувати типи даних, доступні у режимі Конструктор?
5. Які типи даних полів призначені для зберігання текстової інформації? У чому їх відмінність?
6. Як сприймає система логічний тип даних Так/Ні? Який формат і яке значення задається за замовчуванням для логічного типу?
7. У чому полягає специфіка використання типу даних Автонумерація?
8. Чи можна змінювати значення поля з типом Автонумерація? Для яких полів доцільно використовувати саме цей тип даних?
9. Чи можна після вилучення рядків таблиці, в якій є поле з типом Автонумерація, повторно створити рядки зі значенням вилученого номера? Якщо так, як це зробити?
10. Який з типів даних використовується для зберігання зображень? Яким чином такі поля заповнюються даними?
11. З яких структурних елементів складається вікно конструктора таблиць в Access, яке їх основне призначення?
12. У чому полягають особливості створення таблиць у режимі конструктора?
13. Як можна додати нове поле у таблицю?
14. Чи можна змінювати порядок розташування полів у таблиці? Якщо так, то яким чином?
15. Для чого треба встановлювати первинний ключ у таблиці? Яким чином можна це зробити?
16. Поля яких типів не можна (некоректно) встановлювати як ключові?
17. Чи можна поле з типом даних Короткий текст призначати первинним ключем?
18. Поля яких типів можна встановлювати як ключові?
19. Що означає поняття «ключ» (ключове поле)?
20. Які вимоги висуваються до ключового поля?

Лабораторна робота № 2

Завдання

1. Створити Excel-файл, окремий аркуш якого перейменувати на *Посади*, і створити на цьому аркуші таблицю з переліком посад і базових окладів. Зберегти цей файл.

	A	B
1	**Посади**	**Оклад**
2	методист	3600,00
3	фахівець І категорії	3500,00
4	провідний фахівець	3550,00
5	юрист	4200,00
6	інспектор	4000,00
7	програміст	8000,00
8	адміністратор	7500,00
9	модератор	6000,00
10	економіст	7250,00
11	лаборант	3200,00

Рис. 1.6. Створена таблиця з переліком посад та базових окладів

2. Запустити базу даних MS Access з ім'ям *ПрізвищеБД*, створену Вами на попередній лабораторній роботі.

3. Імпортувати до цієї БД таблицю *Посади*, створену в Excel на аркуші *Посади* з переліком посад і базових окладів, установивши зв'язок з першоджерелом, для того щоб при зміненні даних або доповненні нових записів у реєстрі (Excel-файлі) вони відображались і в базі даних Access.

4. Відкрити Excel-файл, створений при виконанні п. 1, або створити новий Excel-файл, додати новий аркуш і перейменувати його на *Доплати*, створити на цьому аркуші таблицю з можливими варіантами доплат (Рис. 1.7):

fx За науковий ступінь

	A	B	C	D	E
1	**КодДоплати**	**ВидДоплати**	**Коефіцієнт**		
2	1	Премія	0,2		
3	2	За складність	0,3		
4	3	За дострокове виконання	0,4		
5	4	За науковий ступінь	0,5		
6					
7					

Рис. 1.7. Таблиця з можливими варіантами доплат

Зберегти файл і закрити Excel.

5. Імпортувати (скопіювати) до БД таблицю *Доплати* з відповідного аркуша таблиці Excel (без встановлення зв'язку з першоджерелом).

6. У таблиці *Працівники* створити нове поле *Відділ* з фіксованим списком підстановки з назвами відділів. Заповнити клітинки поля *Відділ* даними, вибираючи їх зі списку (Рис. 1.8).

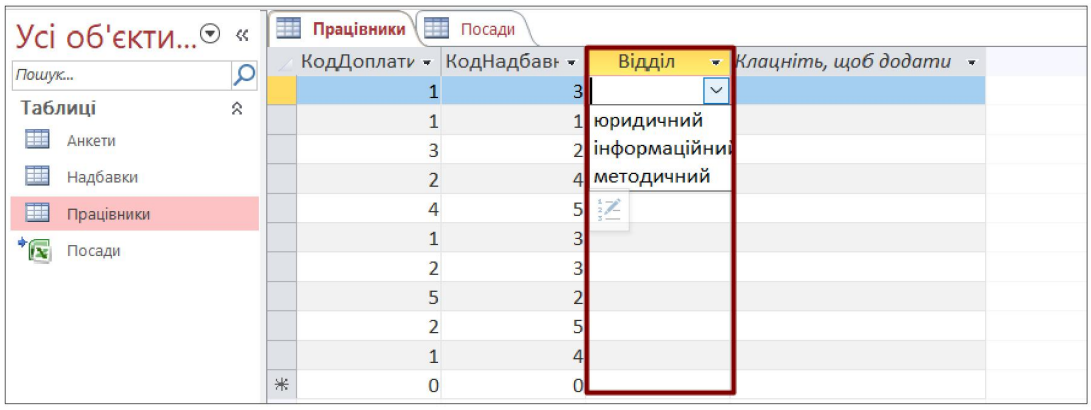
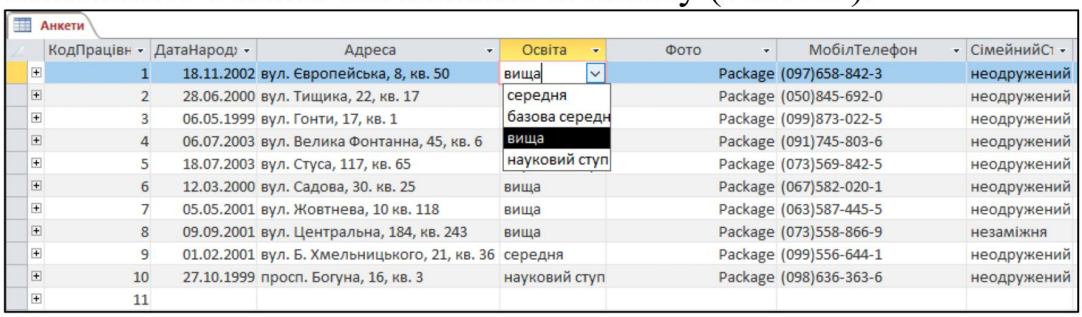


Рис. 1.8. Заповнення поля *Відділ* із запропонованого списку

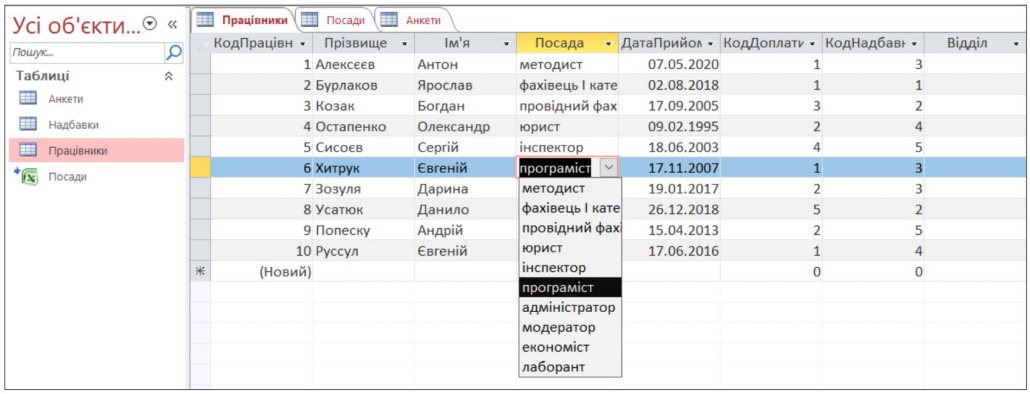
7. У таблиці *Анкети* для існуючого поля *Освіта* створити фіксований список з різновидами освіти працівників організації (науковий ступінь, вища, середня спеціальна, середня), після чого заповнити поле *Освіта* значеннями зі списку (Рис. 1.9).



КодПрацівн	ДатаНародж	Адреса	Освіта	Фото	МобілТелефон	СімейнийСт
1	18.11.2002	вул. Європейська, 8, кв. 50	вища	Package	(097)658-842-3	неодружений
2	28.06.2000	вул. Тищика, 22, кв. 17		Package	(050)845-692-0	неодружений
3	06.05.1999	вул. Гонти, 17, кв. 1		Package	(099)873-022-5	неодружений
4	06.07.2003	вул. Велика Фонтанна, 45, кв. 6		Package	(091)745-803-6	неодружений
5	18.07.2003	вул. Стуса, 117, кв. 65		Package	(073)569-842-5	неодружений
6	12.03.2000	вул. Садова, 30. кв. 25	вища	Package	(067)582-020-1	неодружений
7	05.05.2001	вул. Жовтнева, 10 кв. 118	вища	Package	(063)587-445-5	неодружений
8	09.09.2001	вул. Центральна, 184, кв. 243	вища	Package	(073)558-866-9	незаміжня
9	01.02.2001	вул. Б. Хмельницького, 21, кв. 36	середня	Package	(099)556-644-1	неодружений
10	27.10.1999	просп. Богуна, 16, кв. 3	науковий ступ	Package	(098)636-363-6	неодружений
11						

Рис. 1.9. Заповнення поле *Освіта*

8. У таблиці *Працівники* для існуючого поля *Посада* створити список підстановки з вибором значень з іншої таблиці *Посади* (пов'язаного джерела даних), встановивши тим самим зв'язок з цією таблицею. Заповнити клітинки поля *Посада* даними, вибираючи їх зі списку (Рис. 1.10).



КодПрацівн	Прізвище	Ім'я	Посада	ДатаПрийом	КодДоплати	КодНадбавк	Відділ
1	Алексєєв	Антон	методист	07.05.2020	1	3	
2	Бурлаков	Ярослав	фахівець І кате	02.08.2018	1	1	
3	Козак	Богдан	провідний фах	17.09.2005	3	2	
4	Остапенко	Олександр	юрист	09.02.1995	2	4	
5	Сисоєв	Сергій	інспектор	18.06.2003	4	5	
6	Хитрук	Євгеній	програміст	17.11.2007	1	3	
7	Зозуля	Дарина		19.01.2017	2	3	
8	Усатюк	Данило		26.12.2018	5	2	
9	Попеску	Андрій		15.04.2013	2	5	
10	Руссул	Євгеній		17.06.2016	1	4	
(Новий)					0	0	

Рис. 1.10. Заповнення поля *Посада*

9. У таблиці *Працівники* для поля *КодДоплати* створити поле підстановки для вибирання текстових (а не числових) значень із таблиці *Доплати* для кращої наочності (Рис. 1.11).

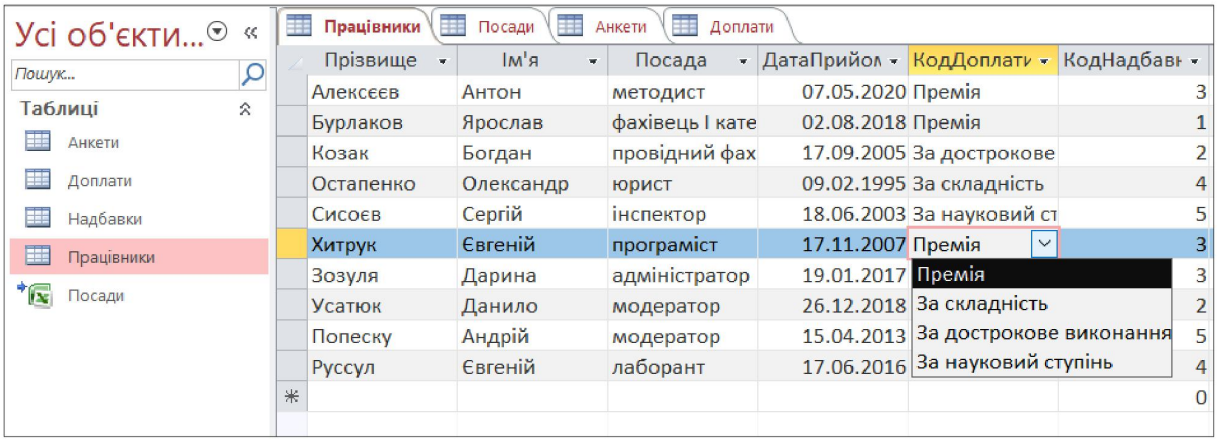


Рис. 1.11. Розкривний список заповнення або змінени значень

10. Встановити зв’язки типів «один-до-одного» та «один-до-багатьох» між усіма створеними таблицями БД, забезпечуючи цілісність даних (каскадне оновлення та вилучення пов’язаних полів) (Рис. 1.12):

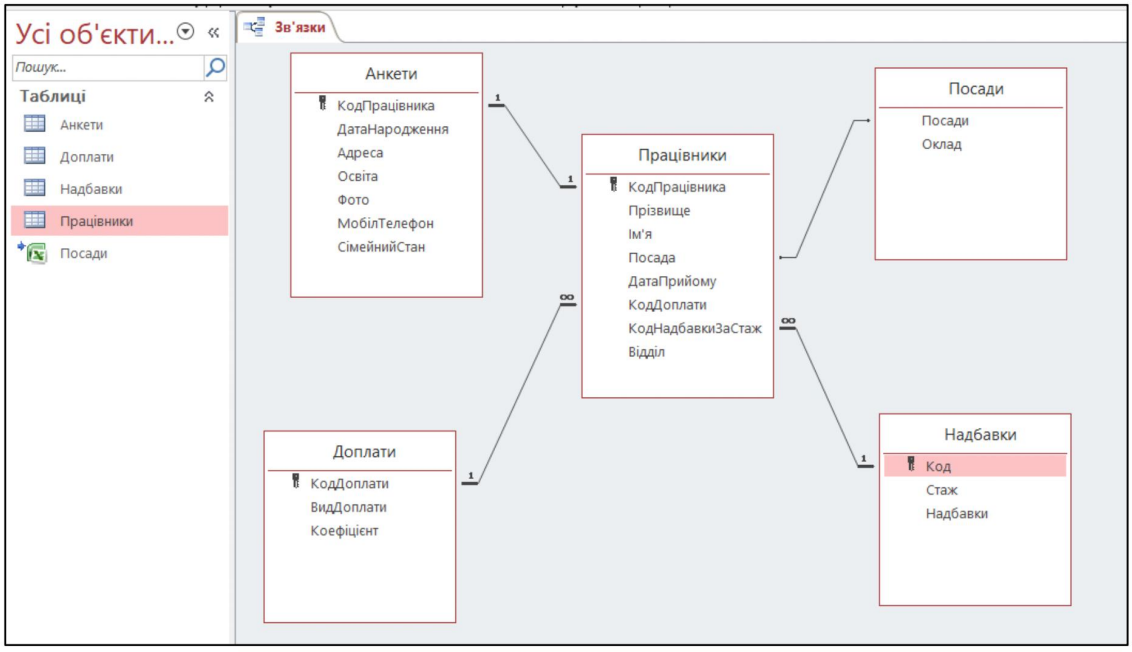


Рис. 1.12. Створений зв’язок між таблицями

Питання для самоконтролю

1. Назвати та охарактеризувати всі типи зв’язків між таблицями БД.
2. Для чого використовуються зв’язки між таблицями?
3. Які типи зв’язків між таблицями є допустимими в Access, а які ні і чому?
4. Які об’єкти БД можуть бути основою для створення полів підстановки (розкривних списків з можливими значеннями) у таблицях Access?
5. Як створити поле підстановки з фіксованим списком можливих значень?
6. Чи можна створити поле підстановки зі списком значень, взятих з іншої таблиці БД? Якщо так, то як це зробити?
7. Як імпортувати таблицю з іншої програми або бази даних зі встановленням зв’язку з першоджерелом?
8. Як імпортувати (скопіювати) таблицю з іншої програми або бази даних без встановлення зв’язку з першоджерелом?
9. Як побачити схему зв’язків між таблицями БД?
10. Як в Access створити зв’язки між таблицями?
11. Між якими полями таблиць можна встановити зв’язок?
12. Які обов’язкові вимоги при встановленні зв’язків між таблицями?
13. Як в Access можна вилучити існуючий зв’язок між таблицями?
14. Що означає маркер у таблиці, відкритій у режимі Таблиця?
15. Що означає символ на лінії зв’язку між таблицями на схемі зв’язків?

Лабораторна робота № 3

Завдання

1. Запустити базу даних MS Access з ім'ям *ПрізвищеБД*, створену Вами на попередній лабораторній роботі.

2. Створити копію таблиці *Посади* як локальну таблицю, зберігши структуру та дані. Перейменувати копію таблиці на *Сортування й фільтр1* та задати в ній сортування записів за зростанням окладу.

3. Створити копію таблиці *Анкети* як локальну таблицю, зберігши структуру та дані. Перейменувати копію таблиці на *Сортування й фільтр2* та встановити сортування за двома полями: 1) полем *Освіта* (за абеткою – *від А до Я*) та 2) за віком – поле *ДатаНародження* (*від найстаршого до наймолодшого*). Тобто в разі однакових значень поля *Освіта* записи мають виводитись за спаданням віку працівників.

4. У таблиці *Сортування й фільтр1* увімкнути фільтрацію для виведення посад з окладом понад 4000 грн. Зберегти змінення та закрити таблицю.

5. У таблиці *Сортування й фільтр2* встановити фільтрацію для виведення тільки працівників з вищою освітою (фільтр за виділенням).

6. Також у таблиці *Сортування й фільтр2* встановити фільтрацію для ви- ведення тільки працівників, що святкуватимуть день народження наприкінці року у четвертому кварталі. При цьому попереднє фільтрування втратить силу.

7. За допомогою розширеного фільтра задати подвійну фільтрацію: іменинників 4-го кварталу, які мають вищу освіту. Зберегти змінення та закрити таблицю.

8. У таблиці *Анкети* за допомогою розширеного фільтра відібрати працівників, які у цьому році святкують ювілей (вік кратний 5). Зберегти цей фільтр як запит з ім'ям *Ювіляри*. Закрити таблицю *Анкети* без зберігання змінень.

Питання для самоконтролю

1. Яку роль відіграє сортування в БД? Чи можна задавати сортування під час створення таблиці? Якщо так, чи можна у подальшому змінювати параметри сортування?

2. Чи можна одночасно задавати сортування за кількома полями? Якщо так, якими є правила встановлення такого сортування?
3. Які за типом поля не підлягають сортуванню?
4. Яким чином тип даних сортованого поля впливає на назву команд сортування?
5. Як скасувати сортування полів у БД?
6. Для чого призначене фільтрування даних?
7. Які існують способи фільтрування? Який з цих способів найпоширеніший і чому?
8. Чи можна одночасно задавати фільтрування за кількома полями? Якщо так, якими є правила встановлення такого сортування?
9. Коли використовують розширений фільтр?
10. Як скасувати фільтрування полів?
11. Пояснити використання в критеріях відбору операторів /, \ та MOD. Навести приклади таких умов
12. Які оператори порівняння можна використовувати в критеріях відбору? Навести приклади таких умов.
13. Навести приклади критеріїв відбору з використанням оператора LIKE.
14. Для чого може використовуватись вираз у критерії відбору: IS NULL? Чим він відрізняється від виразу IS NOT NULL?
15. Навести приклади критеріїв відбору з використанням логічних операторів AND та OR і логічних операторів BETWEEN та IN.
16. Що символізує текст у квадратних дужках у конструкторі критерію відбору?

Лабораторна робота № 4

Завдання

1. Запустити базу даних MS Access з ім’ям *ПрізвищеБД*, створену Вами напопередній лабораторній роботі.

2. У режимі *Конструктор* створити запит з ім’ям *НадбавкиПонад1000* на відбір усіх даних таблиці *Надбавки* Вашої БД за умови, що значення певного числового поля перевищують певне значення, наприклад, понад 1000 грн.

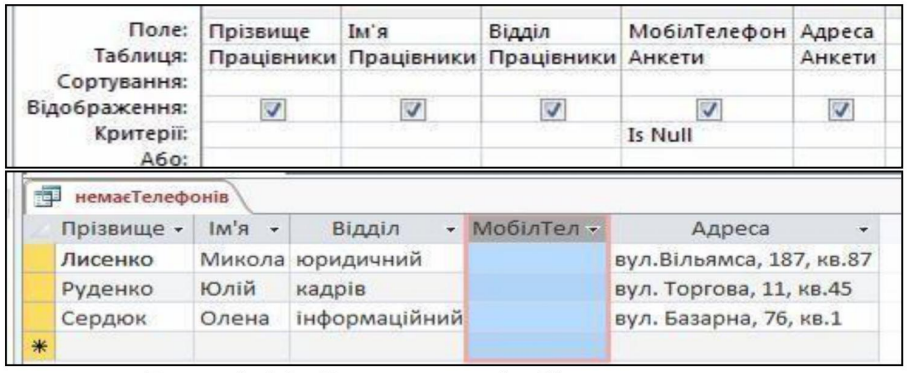
3. У режимі *Конструктор* створити запит на відбір даних лише чотирьох полів з таблиці *Працівники*: одне поле має бути полем дати (поле *ДатаПрийому*), друге – текстовим (поле *Посада*), а третє та четверте поля – якими завгодно на Ваш вибір. Двома умовами для відбору мають бути:

– у полі дати значення має належати до певного діапазону значень (наприклад, *ДатаПрийому* була здійснена в межах 5-ти останніх років);

– у полі *Посада* значення мають збігатися з двома чи трьома певними значеннями (наприклад, працівники на посадах "юрист" або "провідний юрист").

4. Відкрити ще раз створений у п. 3 запит та переглянути його в режимі SQL. Перейти до режиму *Конструктор* та для будь-якого поля встановити в рядку *Сортування* параметри сортування: *за зростанням* або *за спаданням*. Перейти до режиму *Таблиця*, впевнитися у правильності сортування, після чого перейти до режиму SQL та впевнитися в тому, що в запиті було долучено пара- метри сортування. Закрити цей запит.

У режимі *Конструктор* створити запит на відбір даних з двох чи трьох таблиць. В умові відбору використати перевірку на порожні клітинки – *IS NULL*.Наприклад, з таблиць *Працівники* та *Анкети*



Поле:	Прізвище	Ім'я	Відділ	МобілТелефон	Адреса
Таблиця:	Працівники	Працівники	Працівники	Анкети	Анкети
Сортування:					
Відображення:	☑	☑	☑	☑	☑
Критерії:				Is Null	
Або:					

немаєТелефонів

Прізвище	Ім'я	Відділ	МобілТел	Адреса
Лисенко	Микола	юридичний		вул.Вільямса, 187, кв.87
Руденко	Юлій	кадрів		вул. Торгова, 11, кв.45
Сердюк	Олена	інформаційний		вул. Базарна, 76, кв.1
*				

Рис. 1.13. Запит на відбір даних

відібрати дані про працівників з незаповненими даними номерів телефонів (Рис. 1.13):

5. Створити запити з формуванням обчислюваних полів:

– сумарне значення щомісячної зарплати кожного працівника;

– відібрати ювілярів – тих працівників, які у поточному році святкуютьювілей – вік кратний числу 5, та визначити їхній вік.

6. Сформувати три SQL-запити з використанням:

а) однієї з групових функцій *Avg*(), *Min*(), *Max*() або *Sum*(). Наприклад,визначити середній оклад усіх працівників;

б) агрегатної функції *Count*() та групування *GROUP BY*. Наприклад, визначити кількість працівників у кожному з відділів;

в) відбирання даних відразу з двох взаємопов'язаних таблиць даних за певною умовою. Наприклад, вибрати дані про прізвища та оклади працівників, які пропрацювали на підприємстві менше п'яти років.

7. Сформувати підсумковий запит за допомогою *Майстра*, наприклад, запит обчислення сумарних виплат значень окладів по кожному відділу і визначенням кількості працівників у кожному з них.

Питання для самоконтролю

1. Для чого призначені запити? Які види опрацювання даних виконують
запити? Як виводяться результати запитів?
2. У чому схожість і відмінність між запитами і таблицями БД?
3. Які існують способи формування запитів в Access?
4. Які режими подання запитів існують в Access?
5. З яких частин складається вікно Конструктора запитів?
6. Описати способи заповнення полів у бланку запиту.
7. Чим Майстер запитів відрізняється від Конструктора запитів?
8. Чи можна використовувати кілька таблиць БД при створенні простого
запиту у Майстрі запитів? Чому?
9. Роз'яснити використання в критеріях відбору операторів /, \ та MOD.

Навести приклади таких умов.

10. Які оператори порівняння можна використовувати в умовах відбору записів? Навести приклади таких умов.

11. Навести приклади умов у запитах з використанням оператора LIKE.

12. Для чого може використовуватись вираз в умові відбору: IS NULL? Чим
він відрізняється від виразу IS NOT NULL?

13. Навести приклади критеріїв відбору у запитах з використанням логічних операторів AND та OR.

14. Навести приклади критеріїв відбору у запитах з використанням операторів BETWEEN та IN.

15. З якою метою застосовують обчислювані поля? Як у запиті створити
обчислюване поле? Навести приклад виразу для обчислюваного поля.

16. Для чого у запитах застосовуються групові функції? Навести приклади

Лабораторна робота № 5

Завдання

1. Запустити базу даних MS Access з ім’ям *ПрізвищеБД*, створену Вами на попередній лабораторній роботі.
2. Сформувати перехресний запит обчислення сумарних значень окладів по відділах з розподілом за посадами.
3. У режимі *Конструктор* створити запит з параметром для відбору працівників певного відділу за таблицею *Працівники* (назву відділу вводитиме користувач при виконанні запиту).
4. Сформувати запит на оновлення даних, наприклад,збільшення значення премії на 10% (з 20% до 30% від розміру окладу) з перерахунком в усіх пов’язаних таблицях і запитах.
5. Створити запит на вилучення даних з БД з усіх пов’язаних таблиць і запитів, наприклад, вилучення певного працівника за його прізвищем.
6. Сформувати запит на створення нової таблиці, наприклад, з прізвищами і віком працівників юридичного відділу.
7. Сформувати запит на додавання даних, наприклад,додавання у таблицю *Працівники юрвідділу* даних про співробітників з інформаційного відділу.

Питання для самоконтролю

1. Які основні типи запитів існують в Access?
2. Що символізує текст у квадратних дужках у запиті з параметром?
3. Для чого застосовуються групові операції в запитах? Навести приклади.
4. Описати алгоритм створення перехресного запиту.
5. Як відображаються дані електронної таблиці для створення перехресного запиту?
6. Назвати п'ять підсумкових операцій, доступних для створення перехресного запиту.
7. Чи можна при створенні перехресного запиту вибрати дані з кількох таблиць? Чим це обумовлено?
8. Для яких цілей застосовуються запити на створення таблиць?
9. Описати алгоритм створення на створення таблиць.

10. Чим відрізняються запити на відбір від запитів на оновлення даних?
11. Описати алгоритм створення запиту на оновлення даних.
12. Як впливають запити на оновлення та видалення на дані у пов’язаних таблицях і запитах?
13. Описати алгоритм створення запиту на видалення даних.
14. Що відбувається із записами, які призначаються на видалення і чи можна їх відновити після виконання запиту на видалення?
15. Описати алгоритм створення запиту на додавання даних.
16. Для яких цілей застосовуються запити на додавання даних?

Лабораторна робота № 6

Завдання

1. Запустити базу даних MS Access з ім'ям *ПрізвищеБД*, створену Вами на попередній лабораторній роботі.

2. За допомогою інструмента *Форма* створити форму ***Працівники*** за таблицею *Працівники.* Перейти до режиму *Розмітки* (або *Макет*) та зменшити ширину всіх полів за максимальним вмістом, а також зменшити висоту поля списку *Посада*, щоб відображалось лише одне значення.

3. За допомогою *Майстра форм* створити форму ***Анкети*** за таблицею *Анкети* (макет форми *Стовпці*).

4. За допомогою *Пустої форми* створити форму за таблицею ***Доплати***. Змінити дизайн створеної форми на власний смак. За допомогою засобів групи *Елементи керування* на вкладці *Знаряддя конструктора форм / Конструктор* додати на форму тематичний рисунок, дату й час.

5. За допомогою *Конструктора* створити форму ***Надбавки*** за таблицею *Надбавки.* Змінити дизайн створеної форми на власний смак.

6. За допомогою *Майстра форм* створити форму ***Надбавки з підформою*** із підпорядкованою формою з даними про працівників, які отримують цей вид надбавки.

7. За допомогою *Майстра підформ* на формі *Доплати* створити підформу за таблицею *Працівники.*

8. За допомогою *Майстра підформ* на форму *Працівники* додати наявну форму *Анкети.*

9. За допомогою перетягування на формі *Надбавки* створити підформу *Працівники* (на якій до цього Ви розмістили підформу *Анкети*). Налаштувати правильну роботу підпорядкованих форм.

Питання для самоконтролю

1. Яке основне призначення форм при розробці БД?
2. Чому розробники БД намагаються забезпечити введення і редагування даних через форми, а не напряму в таблицях?
3. Назвати способи створення форм.
4. Який спосіб створення форм дозволить відобразити дані однієї таблиці і в режимі таблиці, і в поданні форми одночасно? Яка команда дозволить створити таку форму?

5. Яке призначення форм навігації?
6. У чому особливість використання форм кількох елементів? Як інакше називають такі форми?
7. У чому полягає специфіка використання Майстра форм?
8. Які засоби редагування форм можливі тільки у режимі Конструктор?
9. Як у режимі Конструктор можна винести поля на форму?
10. Порівняти розробку форм за допомогою Майстра та Конструктора.
11. Описати переваги та недоліки технологій розробки форм за допомогою Майстра та Конструктора.
12. Як змінити властивості елемента керування, розміщеного на формі?
13. Назвати три можливих режими виведення якої завгодно форми на екран. Якими командами можна переходити з одного режиму до іншого?
14. В якому з режимів виведення форми на екран можна редагувати колонтитули на формі? Як це зробити?
15. Що таке підформа? Як мають бути пов’язані між собою головна і підпорядкована форми?
16. Які існують способи створення підформ?

Лабораторна робота № 7

Завдання

1. Запустити базу даних MS Access з ім'ям *ПрізвищеБД*, створену Вами на попередній лабораторній роботі.
2. Сформувати простий табличний звіт за таблицею *Анкети* за допомогою команди *Звіт* на вкладці *Створення* у групі *Звіти*.
3. За таблицею *Працівники* сформувати звіт засобами *Майстра звітів* з групуванням по відділах.
4. На звіті *Працівники* обчислити загальну кількість працівників всіх відділів.
5. Додати на звіт *Працівники* емблему.
6. Змінити тему кольорової гами для звіту за власним смаком.
7. Засобами умовного форматування виділити червоним (чи то іншим) кольором кількість працівників у певному відділі, якщо їх менше трьох.
8. Організувати попередній перегляд звіту та за можливості роздрукувати його.

Питання для самоконтролю

1. Яке основне призначення звітів?
2. Порівняти призначення звітів і призначення форм.
3. З яких частин складається вікно конструктора звітів?
4. Охарактеризувати призначення розділів вікна конструктора звітів.
5. Як у вікні конструктора звітів додати відсутні розділи?
6. Як у вікні конструктора звітів вилучити непотрібні розділи?
7. Назвати способи створення звітів.
8. Назвати послідовність створення звіту.
9. Для чого у звіті формують рівні групування?
10. В якому режимі зручніше створювати рівні групування?
11. Як при роботі з Майстром звітів змінити групування даних?
12. Як елементу керування на звіті задати правила умовного форматування?
13. Як за допомогою тем надати звіту професійного зовнішнього вигляду?
14. Як на звіт винести емблему компанії?
15. Як на звіт винести фонове зображення?
16. Які команди можна виконувати у режимі Попередній перегляд звіту?

Лабораторна робота № 8

Завдання

1. Запустити базу даних MS Access з ім’ям *ПрізвищеБД*, створену Вами на попередній лабораторній роботі.
2. За запитом *Ювіляри* створити однойменну форму та додати до неї командну кнопку з макросом, при натисканні на яку відкриватиметься форма *Працівники* з докладними анкетними даними.
3. Створити самостійний (глобальний) макрос, який виконуватиме дві дії (макрокоманди): 1) запускатиме вікно з попередженням, наприклад таким: «Буде відкрита таблиця, в яку можна вносити змінення»; 2) відкриватиме таблицю *Працівники*.
4. Застосувати щойно створений макрос до форми *Ювіляри* (у вигляді пов’язаної з цим макросом кнопки, натискання якої запускатиме створений макрос) і до звіту *Працівники* (у вигляді пов’язаного з цим макросом елемента керування *Підпис*).
5. Створити форму з переліком існуючих посад на підприємстві (за таблицею *Посади*), а далі надати можливість за назвою посади віднайти прізвища працівників, їхнє місце роботи та інші відомості, створивши взаємодію між формами за допомогою макросу. Для розв’язання поставленої задачі створити не лише форму *Посади*, а ще й підформу з анкетними даними працівників. На підформі мають відображатися дані, які задовольнятимуть умову, задану на головній формі.

Питання для самоконтролю

1. Які засоби програмування підтримує MS Access?
2. Що являє собою макрос в Access?
3. Що являє собою модуль в Access?
4. Чим відрізняється модуль від макросу?
5. В яких випадках в Access доречніше використовувати макроси, а в яких – модулі?
6. Як створити макрос і через нього створити зв’язок з об’єктами бази даних?
7. Яка мова програмування використовується в MS Access для написання програмного коду?
8. Як створити на формі кнопку з макросом?
9. Як створити самостійний (глобальний) макрос, який можна використовувати неодноразово у різних об’єктах керування?
10. Де відображаються глобальні макроси, створені у БД?
11. Що треба зробити, щоб запустити глобальний макрос?

Лабораторна робота № 9

Завдання

1. Запустити базу даних MS Access з ім'ям *ПрізвищеБД*, створену Вами на попередній лабораторній роботі.

2. Створити форму навігації з навігаційними елементами керування для перемикання між створеними на попередніх заняттях формами: *Працівники*, *Анкети*, *Доплати*, *Надбавки*, *Посади*, *Ювіляри* і звітами: *Анкети* та *Працівники*.

3. Назвати форму навігації – *База даних Прізвище* (прізвище вказати власне).

4. Змінити оформлення форми навігації: додати емблему, на власний смак задати колір та форму навігаційних кнопок.

5. Призначити форму навігації формою перегляду за замовчуванням.

Питання для самоконтролю

1. Яке призначення форм навігації в Access?
2. Які існують стилі форм навігації?
3. Як відобразити область переходів на формі навігації?
4. В якому режимі відображення форми навігації можна додавати до неї об'єкти?
5. Як змінити заголовок форми навігації?
6. Як змінити фігуру кнопки на формі навігації?
7. Для чого у формах навігації застосовуються експрес-стилі?
8. Як налаштувати параметри відображення форми навігації?
9. Як додати емблему на форму навігації?
10. Як змінити оформлення форми навігації?

Рекомендована література

1. How to open a secured Access database in ADO through OLE. URL: https://support.microsoft.com/en-us/kb/191754.
2. PHP + MySQL. URL: http://sites.znu.edu.ua/webprog/lect/1222.ukr.html.
3. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань : навч. посіб. Львів : "Магнолія-2006", 2012. 584 с.
4. Введение в программирование Access. URL: http://office.microsoft.com/ruru/access-help/HA010341717.aspx?CTT=1#BMlearnaboutmacros .
5. Гайна Г. А. Основи проектування баз даних : навч. посіб. К. : Кондор, 2008. 200 с.
6. Дейт К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL. СПб. : Символ-Плюс, 2010. 480 с.
7. Джелен Б., Александер М. Сводные таблицы в Microsoft Excel 2010 / пер. с англ. М. : ООО "И.Д. Вильямс", 2011. 464 с.
8. Системи оброблення інформації. Бази даних. Терміни та визначення. К. : Держстандарт України, 1995. 29 с.
9. Системи оброблення інформації. Основні поняття. Терміни та визначення. К. : Держстандарт України, 1995. 32 с.
10. Зиборов В. В. Visual C# 2012 на примерах. СПб. : БХВ-Петербург, 2013. 480 с.
11. Знакомство с запросами. URL: http://office.microsoft.com/ru/accesshelp/HA102749599.aspx.
12. Експорт даних Access у документ Word. URL: http://office.microsoft.com/uk-ua/access-help/HA010341683.aspx.
13. Завадський І. О. Основи баз даних : навч. посіб. К. : Видавець І. О. Завадський, 2011. 192 с.
14. Зарицька О. Л. Бази даних та інформаційні системи : метод. посіб. Житомир : Вид-во ЖДУ ім. І. Франка, 2009. 132 с.
15. Кузнецов С. Базы данных. Вводный курс. URL: http://citforum.ru/database/advanced_intro.
16. Організація баз даних та знань: лабораторний практикум для студентів напряму 6.050101 “Комп’ютерні науки” ден. та заоч. форм навч. / уклад.: О.М. Мякшило та ін. К.: НУХТ, 2015. 86 с.
17. Основи баз даних. СКБД Access 2010 (2013) : практ. посіб. / Д. А. Покришень та ін. Чернігів : ТОВ НВП "Інтерсервіс", 2013. 225 с.

18. Павлов А. Н. Пособие для поддержки практических работ по освоению возможностей СУБД Access 2010. URL: http://pavlovrags.narod.ru/Accsess2010.

19. Поставщики данных .NET Framework. URL: https://msdn.microsoft.com/ruru/library/a6cd7c08.aspx.

20. Пошаговое руководство. Подключение к данным в базе данных Access (Windows Forms). URL: https://msdn.microsoft.com/ru-ru/library/ms171893.aspx.

21. Привязка элементов управления Windows Forms к данным в Visual Studio. URL: https://msdn.microsoft.com/ru-ru/library/2b4be09b.aspx.

22. Работа с базами данных на языке C#. Технология ADO .NET: учеб. пособ. / сост. О. Н. Евсеева, А. Б. Шамшев. Ульяновск: УлГТУ, 2009. 170 с.

23. Роб П., Коронер К. Системы баз данных : проектирование, реализация и управление. СПб. : БХВ, 2004. 1040 с.

24. Справочное руководство по MySQL. URL: http://www.MySQL.ru/docs/man/index.html.

25. Сучасні інформаційні та комунікаційні системи і технології: навч. посібн. / О. Г. Трофименко, Н. І. Логінова, О. В. Задерейко, О. Б. Козін; Одеса : Фенікс, 2016. 143 с.

26. Титенко С. В. СКБД MySQL і доступ до БД в PHP. URL: http://www.znannya.org/labs/?view=mysqli-intro.

27. Тарасов О. В., Федько В. В., Лосєв М. Ю. Використання мови SQL для роботи з сучасними системами керування базами даних. Х. : Вид. ХНЕУ, 2013. 348 с.

28. Тарасов О. В., Федько В. В., Лосєв М. Ю. Проектування баз даних : навч. посіб. Х. : Вид. ХНЕУ, 2011. 200 с.

29. Трофименко Е. Г., Швайко И. Г., Северин Н. В. Работа в MS Excel : метод. указания для лаб. и практ. работ. Одесса : ИЦ ОНАС им. А.С. Попова, 2013. 140 с.

30. Трофименко Е. Г., Буката Л. Н., Швайко И. Г. Создание и обработка баз данных : курс лекций. Одесса : ОНАС им. А. С. Попова, 2014. 140 с.

MS ACCESS 2019: ТЕОРІЯ ТА ПРАКТИКА

Навчально-методичний посібник

Укладачі: М. О. Медведєва, І. І. Криворучко, М. С. Ковтанюк

Видається в авторській редакції

Підписано до друку 17.11.2021 р. Формат 60х84/16.
Папір офсетний. Ум. друк. арк. 7,26
Тираж 100 прим. Замовлення № 329

Видавничо-поліграфічний центр «Візаві»
20300, м. Умань, вул. Садова,2
Свідоцтво суб'єкта видавничої справи
ДК № 2521 від 08.06.2006.
тел. (04744) 4-64-88, 4-67-77, (067) 104-64-88
vizavi-print.jimdo.com
e-mail: vizavisadova@gmail.com