



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

С. І. ДОЦЕНКО, Є. М. БУДНІЧЕНКО

ЛАБОРАТОРНИЙ ПРАКТИКУМ
до виконання лабораторних робіт з дисципліни
«ОРГАНІЗАЦІЯ ТА СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ»

Навчальний посібник

Харків 2023

УДК 004.65(075)

Д 714

*Рекомендовано вченою радою Українського державного університету
залізничного транспорту як навчальний посібник
(витяг з протоколу № 11 від 2 грудня 2021 р.)*

Рецензенти:

професори К. О. Метешкін (ХНУМГ ім. О. М. Бекетова),
В. О. Філатов (ХНУРЕ),
доцент О. І. Морозова (НАУ «ХАІ» ім. Жуковського)

Д 714

Доценко С. І., Будніченко Є. М. Лабораторний практикум до виконання лабораторних робіт з дисципліни «Організація та системи керування базами даних»: Навч. посібник. – Харків: УкрДУЗТ, 2023. – 174 с., рис. 124, табл. 33.

ISBN

Навчальний посібник призначено для здобувачів вищої освіти денної та заочної форм навчання освітнього рівня 1 (бакалавр), галузь знань 12 – КІ, освітня програма - СКС.

Викладено теоретичні основи організації та систем керування базами даних. Наведено методичні рекомендації з виконання лабораторних робіт.

УДК 004.65(075)

ISBN

© Український державний університет
залізничного транспорту, 2023.
© С. І. Доценко, Є. М. Будніченко, 2023.

ЗМІСТ

ВСТУП.....	5
Розділ 1. Вступ до бази даних.....	6
1.1. Основні поняття бази даних.....	6
1.2. Обробка інформації за допомогою системи керування базами даних.....	8
1.3. Реляційна модель баз даних.....	9
1.4. Особливості системи керування базами даних Access.....	10
1.5. Структура системи керування базами даних Access.....	12
Контрольні запитання до розділу 1.....	13
Розділ 2. Створення бази даних і основні операції з таблицями.....	14
2.1. Створення БД.....	14
2.2. Створення таблиць.....	16
2.3. Робота з полями.....	23
2.4. Зв'язки між таблицями.....	29
2.5. Використання індексів.....	37
2.6. Операції з таблицями.....	40
2.7. Стискання бази даних.....	42
Контрольні запитання до розділу 2.....	42
Розділ 3. Обробка даних у таблицях.....	43
3.1. Можливості обробки даних.....	43
3.2. Форматування таблиці.....	54
3.3. Використання підтаблиць.....	57
3.4. Застосування підстановки.....	60
Контрольні запитання до розділу 3.....	65
Розділ 4. Використання запитів.....	66
4.1. Запит на вибірку.....	66
4.2. Типи запитів на зміну.....	86

4.3. Властивості запитів.....	96
Контрольні запитання до розділу 4.....	99
Розділ 5. Форми.....	100
5.1. Види форм.....	100
5.2. Створення форми.....	103
5.3. Вибір полів.....	105
5.4. Підлеглі форми.....	117
5.5. Використання кнопок.....	120
5.6. Умовне форматування.....	123
Контрольні запитання до розділу 5.....	126
Розділ 6. Звіти.....	127
6.1. Базові відомості про звіти.....	127
6.2. Створення звіту.....	127
Контрольні запитання до розділу 6.....	137
Розділ 7. Друк в Access.....	138
7.1. Друк опису об'єктів БД.....	138
7.2. Друк даних у режимі таблиці.....	140
7.3. Друк форм.....	143
7.4. Друк звітів.....	145
Контрольні запитання до розділу 7.....	146
Лабораторна робота 1. Створення незв'язаних таблиць.....	147
Лабораторна робота 2. Створення зв'язаних таблиць.....	152
Лабораторна робота 3. Створення запитів.....	157
Лабораторна робота 4. Робота з груповими операціями.....	162
Лабораторна робота 5. Створення форм.....	167
Лабораторна робота 6. Розроблення звітів.....	170
Бібліографічний список.....	173

ВСТУП

Навчальний посібник використовується при підготовці до лабораторних і практичних занять і самостійної роботи з дисципліни «Організація та системи керування базами даних». Містить теоретичні відомості щодо проектування і створення баз даних за допомогою Microsoft Access, фільтрування даних, побудови запитів і звітів, конструювання форм для зручного інтерфейсу бази даних, а також детальну послідовність виконання лабораторних робіт, що дають змогу здобувачам вищої освіти всебічно опанувати роботу в Access.

Розділ 1. ВСТУП ДО БАЗИ ДАНИХ

1.1. Основні поняття бази даних

У наш час життя людини настільки сильно насичене різною інформацією, що для її обробки необхідне створення величезної кількості сховищ і банків даних різного призначення. Зараз практично будь-яке завдання пов'язане з маніпулюванням інформацією й даними. Через це останнім часом з'явилося безліч різних комп'ютерних систем, які називаються системами керування базами даних (СКБД) і призначені саме для цього.

По суті база даних (БД) – це деяка подоба електронної картотеки, електронного сховища даних, що зберігається в комп'ютері у вигляді одного або декількох файлів [1–5].

Водночас виникає необхідність виконання ряду операцій з БД, зокрема [1]:

- додавання нової інформації в існуючі файли БД;
- додавання нових порожніх файлів у БД;
- зміна (модифікація) інформації в існуючих файлах БД;
- пошук інформації в БД;
- видалення інформації з існуючих файлів БД;
- видалення файлів із БД.

Комп'ютеризована інформаційна система являє собою програмний комплекс, завданням якого є підтримка надійного зберігання БД у комп'ютері, виконання перетворень інформації і відповідних обчислень, надання користувачам зручного і легкого в освоєнні інтерфейсу. Традиційно обсяги інформації, з якими доводиться мати справу таким системам, досить великі, а самі БД мають доволі складну структуру. Прикладами інформаційних систем є системи замовлення залізничних або авіаційних квитків, банківські системи та інші.

Як ілюстрацію можна розглянути невелику БД (рис. 1.1), у якій зберігається інформація про стипендії студентів.

KOD	NAME	GROUP	BALL	STIP
525122	Батракова І.Н.	5-4-УПП	4.5	1000
525123	Баулін А.В.	5-4-УПП	4.0	0.00
525124	Белянин Д.А.	5-4-УПП	5.0	1000
525125	Бондаренко О.В.	5-4-УПП	4.5	1000
525126	Зеленський І.В.	5-4-УПП	3.0	1000
525127	Климова В.В.	5-4-УПП	4.5	1000
525128	Кричун О.В.	5-4-УПП	4.0	0.00
525129	Кузнецова В.В.	5-4-УПП	4.5	1000
525130	Нестреляй А.М.	5-4-УПП	3.0	1000
525131	Салова О.П.	5-4-УПП	4.5	1000

Запис →

↑
Поле

Рис. 1.1. Приклад таблиці «Стипендія»

Дані про студентів записуються в таблицю, що має чітку структуру даних. Інформація всередині таблиці складається з *полів*, кожне з яких має власне ім'я. Поля, що є в розглянутому прикладі, наведено в табл. 1.1.

Таблиця 1.1

Опис полів таблиці «Стипендія»

Поле	Опис
KOD	Номер студентського квитка
NAME	Прізвище й ініціали студента
GROUP	Найменування навчальної групи студента
BALL	Середній бал успішності студента
STIP	Розмір нарахованої стипендії

Кожен рядок таблиці потрібно розглядати як *запис*, водночас інформація в таблицю заноситься у відповідні поля. Усі записи складаються з однакових полів, а характер інформації для одного поля у всіх записах однаковий. Водночас різні поля мають різний характер збережених даних.

Основним призначенням БД насамперед є швидкий пошук інформації, що зберігається в ній. При значному розмірі БД ручний пошук, а також модифікація інформації, що в ній зберігається, забирає значний час. Використання комп'ютера для обробки БД усуває перераховані вище проблеми при пошуку і вибірці інформації. У цьому випадку її модифікація здійснюється досить швидко й ефективно, а сама БД легко розміщується на будь-якому диску і за потреби може бути перенесена на інший комп'ютер.

1.2. Обробка інформації за допомогою системи керування базами даних

Існує велика кількість програм, призначених для структурування інформації, розміщення її в таблицях і маніпулювання наявними даними – такі програми отримали назву СКБД. Основна особливість СКБД – це наявність засобів для введення й зберігання не тільки самих даних, але й описів їхньої структури [6–9].

Якщо говорити більш детально, то до функцій СКБД відносять нижченаведені функції.

Керування даними безпосередньо в БД – функція, яка забезпечує зберігання даних, що безпосередньо входять у БД, і службової інформації, що забезпечує роботу СКБД [6].

Керування даними в пам'яті комп'ютера – функція, пов'язана насамперед з тим, що СКБД працюють з БД великого розміру. Для прискорення роботи СКБД використовується буферизація даних в

оперативній пам'яті комп'ютера. Водночас користувач СКБД використовує тільки необхідну для його конкретного завдання частину БД, а за потреби одержує нову «порцію» даних [6].

Керування транзакціями – функція СКБД, що здійснює ряд операцій над БД як над єдиним цілим. Як правило, такі операції виконуються в пам'яті комп'ютера. Насамперед транзакції необхідні для підтримки логічної цілісності БД у багатокористувальницьких системах. Якщо транзакція (маніпуляція над даними) успішно виконується, то СКБД вносить відповідні зміни в БД. В іншому випадку жодна зі зроблених змін ніяк не впливає на стан БД [6].

Підтримка мов БД – для роботи з БД використовуються спеціальні мови, які мають спільну назву *мови баз даних*. У СКБД звичайно підтримується єдина мова, що містить усі необхідні засоби – від створення БД до забезпечення користувальницького інтерфейсу при роботі з даними. Найпоширенішою на цей час мовою СКБД є мова SQL (Structured Query Language) [6].

1.3. Реляційна модель баз даних

Останнім часом у більшості БД використовуються реляційні моделі даних, і практично всі сучасні СКБД орієнтовані саме на таке подання інформації [9–12].

Реляційну модель можна подати як особливий метод розгляду даних, що містить як власне дані (у вигляді таблиць), так і способи роботи й маніпуляції з ними (у вигляді зв'язків). Інакше кажучи, у реляційній БД використовується кілька таблиць, між якими встановлюються зв'язки. Отже, інформація, введена в одну таблицю, може бути пов'язана з одним або кількома записами з іншої таблиці [9].

Між записами двох таблиць (наприклад таблиць А і В) можуть існувати такі види зв'язку [9]:

- «один до одного» (кожному запису з А відповідає один певний запис із В, наприклад у кожного студента є лише одна залікова книжка);
- «один до багатьох» (кожному запису з А відповідає кілька записів з В, наприклад у будинку проживає багато мешканців);
- «багато до одного» (множині записів з А відповідає один певний запис із В, наприклад декілька студентів навчаються в одній групі);
- «багато до багатьох» (множині записів з А відповідає множина записів з В, наприклад у різних студентів заняття проводять різні викладачі).

Важливим поняттям у теорії реляційних БД є *нормалізація*, принципи якої можна сформулювати у вигляді таких основних правил, що використовуються при розробленні структури БД [9]:

- 1) у кожному полі таблиці має перебувати унікальний вид інформації, тобто в тій самій таблиці не мають бути поля, що повторюються;
- 2) у кожній таблиці має бути *первинний ключ* або унікальний ідентифікатор, що однозначно визначає даний запис серед безлічі записів таблиці;
- 3) кожному значенню первинного ключа має відповідати вичерпна інформація про об'єкт таблиці (зокрема об'єктом таблиці може бути студент, повна інформація про якого подана в таблиці БД «Стипендія»);
- 4) зміна значення кожного поля таблиці, що не входить до складу первинного ключа, не має впливати на інформацію в інших її полях.

1.4. Особливості системи керування базами даних Access [1]

Додаток Access є реляційною СКБД, що підтримує всі засоби і можливості з обробки даних, властиві реляційним моделям. Водночас інформація, яку потрібно зберігати у відповідних БД, може бути подана в

практично будь-якому форматі, зокрема текстовому, графічному, числовому, грошовому, дата або час тощо.

Серед засобів, пропоновані СКБД Access, не можна не виділити можливість динамічного обміну даними (DDE) між Access та іншими додатками, які підтримують цю технологію. Також є можливість застосування технології ActiveX, що дає змогу використовувати розробнику у своєму програмному продукті не тільки ті об'єкти, властиві цьому додатку (зокрема Access), але й об'єкти інших додатків (наприклад Excel або Word) [1].

Досить зручною можливістю є той факт, що користувач при обробці даних може працювати не тільки з БД оброблюваного в Access формату, але й експортувати дані інших СКБД, що мають зовсім інший формат подання, зокрема формат FoxPro, Paradox та інші.

Крім того, за допомогою Access користувач може обробляти БД, що підтримують відкритий доступ до даних (стандарт ODBC), зокрема досить популярних останнім часом серверів баз даних Oracle та SQL Server [1].

При обробці даних в Access використовується структурована мова запитів SQL, яку без перебільшення можна назвати стандартною мовою БД. За її допомогою можна виконувати найрізноманітнішу обробку наявних даних, зокрема створювати вибірки потрібної структури, вносити необхідні зміни в наявні БД, перетворювати або видаляти таблиці, формувати дані для звітів і багато іншого.

Важливою перевагою СКБД Access є те, що за її допомогою можна розробляти системи, які обробляють БД як на окремому комп'ютері, так і в локальній мережі підприємства або в інтернеті, використовуючи режим обробки даних «клієнт-сервер» [1].

Слід також зазначити, що Access дає широкі можливості для створення додатків, пов'язаних з обробкою БД. Водночас розробнику не обов'язково бути програмістом високого класу, а цілком достатньо мати

увявлення про створення таких додатків у середовищі Windows, а також володіти деякими навичками програмування мовою Visual Basic. У цьому разі розробник досить швидко зможе опанувати навички створення додатків в Access, що дасть змогу виконувати автоматизацію як простих, так і досить складних завдань, пов'язаних з обробкою даних.

1.5. Структура системи керування базами даних Access

СКБД Access при обробці інформації розглядає БД як набір декількох структурних елементів, кожен з яких може містити один або кілька об'єктів. Серед основних складових БД, з погляду Access, можна виділити такі об'єкти [1, 6]:

1. *Таблиці*. Являють собою об'єкти, створювані користувачем для зберігання інформації про предмети або суб'єкти в певній структурі. Будь-яка таблиця складається з *полів* (стовпчиків) і *записів* (рядків) (рис. 1.1).

2. *Запити*. Є об'єктами, призначеними для одержання необхідних даних з наявних у БД таблиць. Як правило, при створенні запитів використовується мова SQL. За допомогою запитів можна створювати вибірки даних, додавати або видаляти інформацію в певній таблиці. Крім того, за допомогою запиту можливе також створення нових таблиць на підставі однієї або декількох наявних у БД таблиць.

3. *Форми*. Являють собою об'єкти, використовувані для розроблення інтерфейсу, за допомогою якого відбувається введення даних користувачем, а також відображення наявної в БД інформації на екрані. Крім того, форми застосовуються для керування розробленим додатком, наприклад для виконання якоїсь дії при виникненні певної події.

4. *Звіти*. Є об'єктами, використовуваними для підбивання будь-яких підсумків на підставі наявних даних і виведення цих підсумків у певному форматі на друк.

5. *Сторінки*. Являють собою об'єкти, які забезпечують доступ до інформації, наявної в БД, з мережі Інтернет за допомогою браузера. Кожна сторінка, як правило, являє собою HTML-файл, за допомогою якого користувачі інтернету одержують доступ до наявної БД.

6. *Макроси*. Є об'єктами, призначеними для виконання певних дій при виникненні тієї або іншої події. Наприклад, за допомогою макросів можна створювати запити, формувати звіти, відкривати таблиці, обробляти форми тощо.

7. *Модулі*. Являють собою об'єкти, що містять одну або кілька процедур, написаних мовою Visual Basic у системі розроблення додатків Visual Basic for Applications. За допомогою модулів можна реалізовувати широкий спектр можливостей з обробки даних, недоступних розробнику при використанні макросів.

Контрольні запитання до розділу 1

1. Чим є по суті база даних?
2. Як потрібно розглядати кожен рядок таблиці?
3. Визначити основне призначення БД.
4. Навести зміст функції СКБД *«керування даними безпосередньо в БД»*.
5. Навести зміст функції СКБД *«керування даними в пам'яті комп'ютера»*.
6. Навести зміст функції СКБД *«керування транзакціями»*.
7. Навести визначення Реляційної моделі.
8. Чи є додаток Access реляційною СКБД?
9. Чи дає Access можливості щодо створення додатків, пов'язаних з обробкою БД?
10. Як СКБД Access розглядає БД при обробці інформації?

Розділ 2. СТВОРЕННЯ БАЗИ ДАНИХ І ОСНОВНІ ОПЕРАЦІЇ З ТАБЛИЦЯМИ

2.1. Створення БД

Після того як додаток Access запущений, на екрані з'явиться діалогове вікно початку роботи з цією СКБД (рис. 2.1). За його допомогою можна створити нову БД (*Нова база даних*) або відкрити вже наявну, недавно використовувану (*Відкрити базу даних*). Якщо потрібної БД нема в списку, то потрібно викликати діалогове вікно *Відкриття файлу бази даних* за допомогою команди *Інші файли....* Водночас потрібно враховувати, що файли БД мають розширення *.accdb [1, 6].

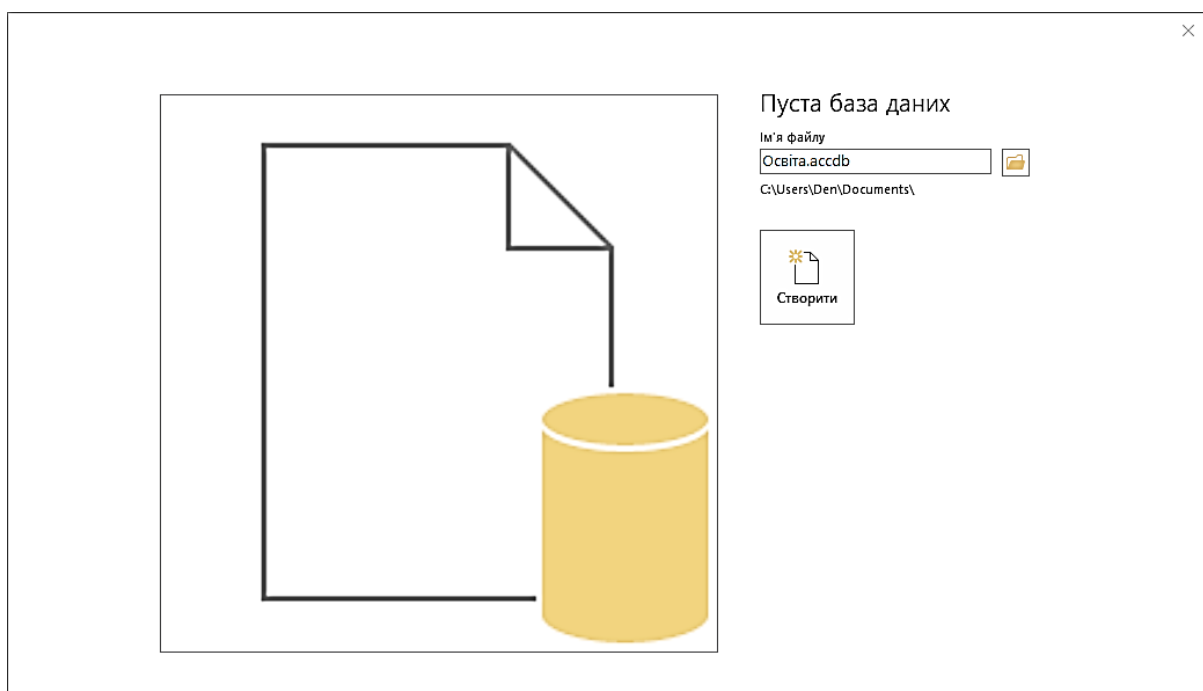


Рис. 2.1. Початок роботи з Access

При виборі команди *Нова база даних* на екрані буде відображено діалогове вікно створення нової БД, у якому потрібно в полі *Ім'я файлу*

вказати ім'я БД, що створюється. Як приклад буде розглянуто створення навчальної БД **Освіта.accdb**, що містить інформацію про студентів, предмети, що вони вивчають, викладачів, а також успішність студентів.

Після введення імені БД потрібно натиснути кнопку *Створити*, після чого на екрані з'явиться вікно створеної БД **Освіта** (рис. 2.2), необхідне при роботі з будь-якою БД.

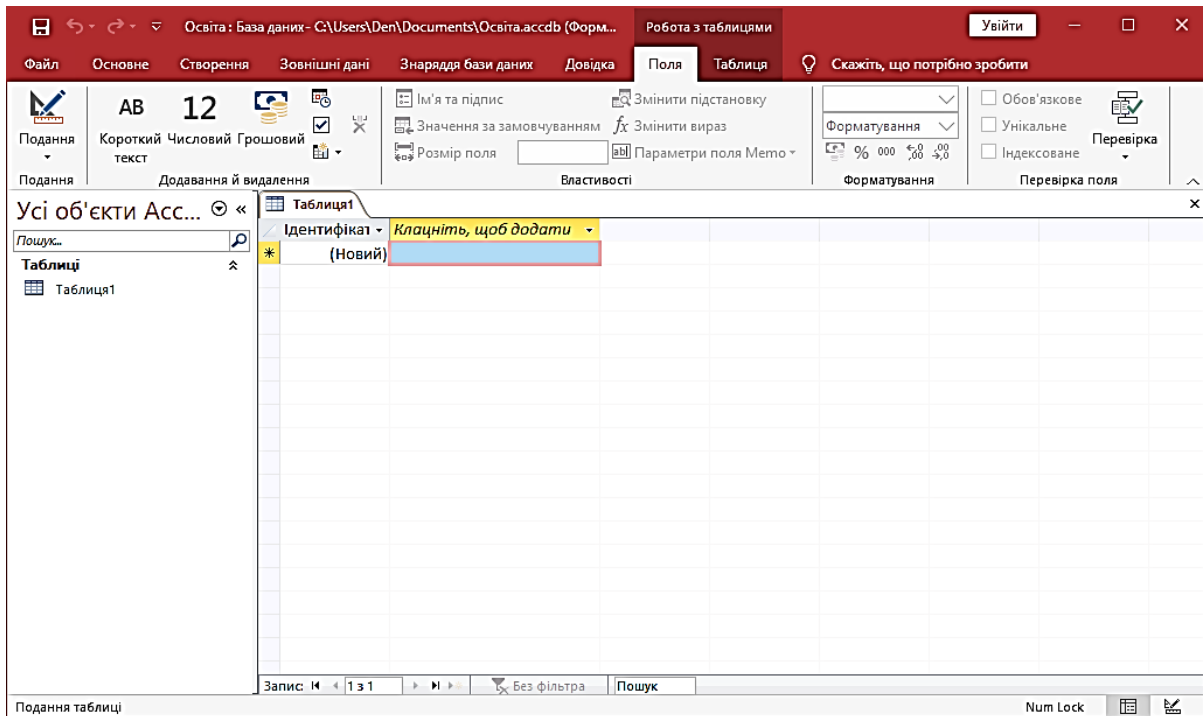





Рис. 2.2. Нова БД **Освіта**

Щоб за його допомогою створити новий об'єкт або змінити вже наявний, наприклад таблицю, потрібно натиснути кнопку *Таблиці* в списку *Об'єкти*. Після цього в правій частині вікна будуть відображені всі можливі режими створення обраного об'єкта (у цьому випадку таблиці), а також список усіх наявних у цій БД таблиць. Крім того, у верхній частині вікна є кілька кнопок для роботи з обраним об'єктом (табл. 2.1).

Кнопки роботи з таблицею

Кнопка	Опис
 Подання	відкрити таблицю в <i>режимі таблиці</i>
 Подання	відкрити таблицю в <i>режимі конструктора</i>
 Таблиця	створити нову таблицю

2.2. Створення таблиць

Щоб створити нову таблицю, потрібно у вікні відкритої БД вибрати в меню *Створення* і натиснути кнопку *Таблиця* (рис. 2.2). Після цього відобразиться нова таблиця (рис. 2.3).

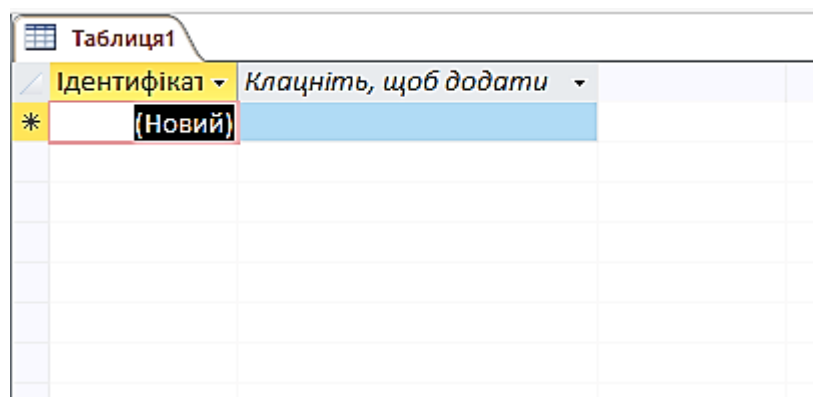


Рис. 2.3. Нова таблиця

Ще один варіант створення таблиці полягає у виборі бажаного варіанта створення таблиці (створення таблиці в режимі *конструктора*, створення списків SharePoint).

Режим таблиці. Дає змогу вводити дані безпосередньо в таблицю (зручно використовувати для редагування наявної таблиці).

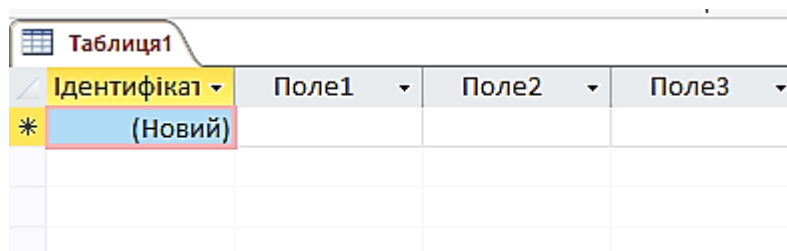
Конструктор. Дає можливість перед введенням даних самостійно визначити структуру нової таблиці, тобто які саме поля в ній будуть і типи використовуваних у цих полях даних.

Імпорт таблиць. Застосовується для створення нової таблиці БД з якого-небудь файлу даних, наявного на диску, причому, можливо, навіть в іншому форматі (не Access).

Зв'язок з таблицями. Дає змогу створити в БД таблицю, що буде пов'язана з таблицею з другого файлу даних, інакше кажучи, зовнішню таблицю. Дуже зручно використовувати цей режим для підключення до будь-якого стороннього сервера БД, наприклад Microsoft SQL Server.

Далі розглянемо перші три режими створення нових таблиць у БД.

Режим таблиці. У разі вибору цього режиму буде відображене вікно з новою таблицею, що має стандартне ім'я *Таблиця1* (рис. 2.4). Назви стовпчиків також мають стандартні назви: *Поле1*, *Поле2* і т. д.



Ідентифікат ▾	Поле1 ▾	Поле2 ▾	Поле3 ▾
*	(Новий)		

Рис. 2.4. Частина вікна Таблиця1





При заповненні нового запису в таблиці дані потрібно вводити в комірки на перетині відповідного рядка і стовпчика. Область таблиці, розташована ліворуч від комірок, називається *областю виділення рядка*, а область над комірками – *областю виділення стовпчика*. При переміщенні покажчика миші в одну з цих областей він набуває вигляду чорної стрілки, а натискання лівою кнопкою миші призведе до виділення відповідного



рядка або стовпчика. З іншого боку, натискання в цьому ж положенні правої кнопки миші призведе до відображення списку можливих дій, які можна виконати для виділеного рядка (стовпчика).

В області виділення рядка може перебувати один із символів, що наочно відображує поточний стан поля таблиці, на якому в цей момент перебуває курсор (табл. 2.2).

Таблиця 2.2

Кнопки в області виділення рядка

Кнопка	Опис
	поточний запис, уже збережений
	новий запис, до якого можна вводити дані
	запис, що змінюється, ще не збережений
	запис у цей момент заблокований

Після того як до будь-якого поля порожнього запису введено перший символ, в області виділення рядка замість зображення поточного збереженого запису  з'явиться зображення запису, що редагується . Після закінчення введення значення до поточного поля запису можна скористатися клавішами **Tab**, **Enter** або клавішею керування курсором → для переходу до наступного поля. Інакше кажучи, введення і редагування даних у комірках таблиці Access виконується так само, як і в Excel. Після завершення введення до останнього поля поточного запису (рядка) виконується автоматичний перехід до першого поля наступного запису (наступного рядка).

Щоб змінити назву стовпчика в таблиці, потрібно виділити необхідний стовпчик, натиснувши лівою кнопкою миші на його області виділення, і вибрати команду *Перейменувати стовпчик* у меню *Формат*

(або в контекстному меню для цього стовпчика). Після цього в області виділення стовпчика можна задати для нього нове ім'я. Слід зазначити, що в попередніх версіях Access при зміні назви поля в одній з таблиць БД доводилося змінювати вручну відповідні назви, пов'язані з цим полем, в інших таблицях, запитах, звітах, формах і т. д. У нових версіях Access існує налаштування, що дає змогу виконувати такі зміни автоматично. Режим увімкнений за замовчуванням. Щоб вимкнути режим автоматичної заміни імен для полів таблиць, потрібно перейти в *Параметри*, після чого на екрані відобразиться діалогове вікно *Параметри*, потім вибрати вкладку *Поточна база даних*, на якій є секція *Параметри автовиправлення імен* (рис. 2.5).

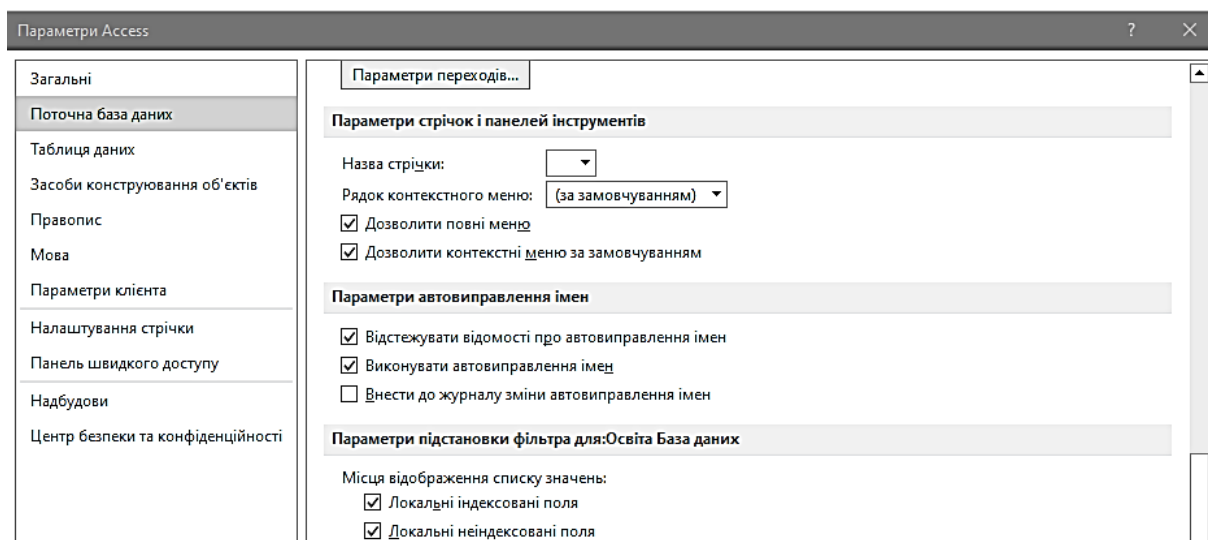


Рис. 2.5. Діалогове вікно *Параметри*. Вкладка *Поточна база даних*

На режим автоматичної заміни імен впливають прапорці категорії *Автозаміна імен*:

– *відстежувати відомості про автовиправлення імен* – зміни відслідковуються, фіксуються, але водночас заміна імен не виконується автоматично для всіх об'єктів БД – таблиць, запитів, форм і т. д.;

– *виконувати авто виправлення імен* – при зміні імені поля в таблиці для всіх інших об'єктів БД виконується автоматична заміна цього імені;

– *внести до журналу зміни авто виправлення імен* – у таблиці «Журнал автозаміни» будуть фіксуватися всі зміни в БД при виконанні автоматичної заміни імен.

Після встановлення необхідних опцій потрібно натиснути кнопку *ОК* (як правило, перші дві опції в категорії *авто виправлення імен* установлені за замовчуванням). Водночас потрібно мати на увазі, що параметри, встановлювані для автоматичної заміни імен, впливають тільки на поточну відкриту БД. Отже, при роботі з кожною БД слід перевіряти, чи встановлені для неї параметри автоматичної заміни.






Щоб додати до таблиці новий стовпчик, слід спочатку виділити той стовпчик, перед яким потрібно вставити інший, після чого виконати команду *Стовпчик* меню *Вставка* (або в контекстному меню для цього стовпчика вибрати *Вставити поле*).

Для видалення стовпчика треба спочатку його виділити, після цього в меню *Виправлення* або в контекстному меню для цього стовпчика виконати команду *Видалити поле* і у вікні *Попередження*, що з'явилося, вибрати кнопку *Так*.

За потреби зміни порядку проходження стовпчиків у таблиці можна виконати перетягування якого-небудь стовпчика в інше місце таблиці. Для цього потрібно спочатку виділити стовпчик або групу стовпчиків, які потрібно перемістити, після цього встановити курсор на виділеному заголовку і перетягнути стовпчики на нове місце в таблиці, утримуючи натиснутою кнопку миші.

Слід також зазначити, що в цьому режимі для переходу до того або другого запису в нижній частині вікна таблиці є кнопки керування курсором, тобто кнопки, що дають змогу встановити курсор на той або інший запис (табл. 2.3).

Кнопки керування курсором у таблиці

Кнопка	Опис
	Перший запис
	Попередній запис
	Наступний запис
	Останній запис
	Новий запис

Крім того, можна також скористатися *полем номера запису*, у якому відображується порядковий номер поточного запису. Для того щоб перейти до запису з певним порядковим номером, потрібно ввести цей номер до *поля номера запису* і натиснути **Enter**.

Конструктор. Цей режим застосовується для створення структури нової таблиці. При виборі режиму конструктора буде відображене вікно (рис. 2.6), у якому потрібно визначити структуру нової таблиці, тобто набір полів і їхній формат. Щоб описати чергове поле в структурі таблиці, потрібно спочатку вказати назву поля *Ім'я поля* і після цього визначити в списку, що розкривається, тип даних, які будуть у ньому зберігатися. Крім того, можна також описати призначення інформації, що буде вводиться до цього поля **Опис**. Слід зазначити, що вміст параметра *Опис* буде відображуватися в рядку стану вікна Access у момент редагування цього поля таблиці.

Після того як визначено структуру таблиці, можна перейти до безпосереднього введення даних. Для перемикання між режимами **Конструктор** і **Таблиці** можна скористатися *командами* меню *Вид: Конструктор* і *Режим таблиці*. Крім того, з тією ж метою зручно користуватися кнопкою *Вид* панелі інструментів (табл. 2.4).

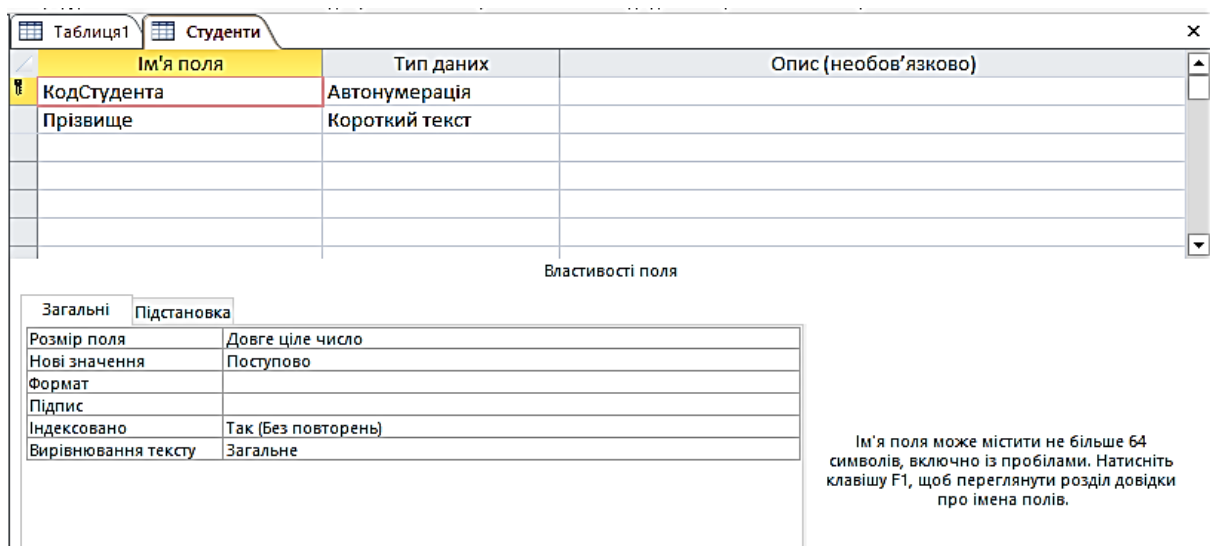

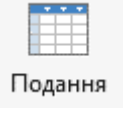


Рис. 2.6. Режим конструктора таблиці


Таблиця 2.4


Кнопка *Вид панелі інструментів*

Кнопка	Опис
	перехід до <i>режиму конструктора</i>
	перехід до <i>режиму таблиці</i>

Для будь-якої таблиці в реляційній БД має бути заданий так званий *первинний ключ*, або *ключове поле*, що дає змогу однозначно визначити той або інший запис у таблиці. Інакше кажучи, ключ необхідний для дотримання унікальності наявних у таблиці записів. Насправді таблиця може існувати і без ключового поля, однак у цьому разі можливі проблеми з підтримкою цілісності даних.

Щоб створити в таблиці первинний ключ, потрібно відкрити таблицю в режимі **Конструктор**, потім встановити курсор у поле, яке потрібно призначити ключовим, натиснувши лівою кнопкою миші на області

виділення рядка, що відповідає цьому полю, після цього виконати команду *Виправлення / Ключове поле* або натиснути кнопку *Ключове поле*  на панелі інструментів, що призведе до появи відповідного зображення в області виділення рядка. Щоб скасувати призначення первинного ключа для поля, потрібно ще раз виконати описані вище дії.

Для збереження змін, внесених до таблиці, потрібно виконати команду *Файл / Зберегти* або натиснути кнопку *Зберегти*  на панелі інструментів.

2.3. Робота з полями

При створенні структури таблиці потрібно визначати типи даних, які будуть у ній використовуватися. Щоб задати тип даних, які будуть зберігатися в певному полі, потрібно натиснувши в *режимі Конструктор* таблиць лівою кнопкою миші в стовпчику *Тип даних* відповідного рядка, після чого вибрати у списку необхідний тип. Наприклад, щоб визначити тип даних для поля **Код Студента** таблиці **Студенти**, потрібно виконати натискання в стовпчику *Тип даних* (рис. 2.7), після чого скористатися стрілкою, що з'явилася, у цьому рядку [1].

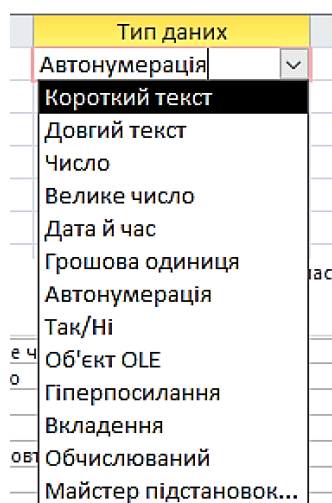


Рис. 2.7. Типи даних

Потім потрібно вибрати у списку необхідний тип даних і задати параметри обраного типу в розділі *Властивості поля* вікна конструктора.

У табл. 2.5 наведено всі наявні в Access типи даних [1].

Для кожного типу даних можна встановити значення властивостей, що будуть визначати особливості обробки відповідного поля. Серед усіх властивостей можна виділити загальні, які можна задати для більшості типів (табл. 2.6).

Слід також зазначити, що в Access існує можливість введення даних у певному форматі. Для цього відповідному полю потрібно задати *маску введення*, що визначає формат введення інформації, що, наприклад, може обмежити діапазон можливих символів, які вводитимуться в поле, тільки цифрами або тільки буквами, встановити автоматичне перетворення введеного тексту до нижнього або верхнього регістру (тобто змінювати всі введені букви на малі або на великі) тощо [1].

Таблиця 2.5

Типи даних СКБД Access

Тип	Опис	Обсяг пам'яті
Текстовий	Текст – один рядок	До 255 байт
Поле МЕМО	Текст – кілька рядків	До 65535 байт
Числовий	Число	1, 2, 4 або 8 байт
Дата/час	Дата й/або час	8 байт
Грошовий	Грошові суми (числа з чотирма десятковими знаками)	8 байт
Лічильник	Унікальне ціле число, що позначає порядковий номер запису в таблиці	4 байт
Логічний	Логічні значення	1 біт
Поле об'єкта OLE	Об'єкти інших додатків Windows	До 1 Гбайт
Гіперпосилання	Посилання на файл (у цьому комп'ютері, локальній мережі, в інтернет)	До 2048 символів
Майстер підстановок	Вибирає значення полів з іншої таблиці або запиту, або ж дає змогу ввести фіксований набір варіантів	Залежить від обраного варіанта підстановки

Основні властивості більшості типів даних

Властивість	Опис
Підпис	Встановлюється інформативна назва поля, яка автоматично буде використовуватися при створенні форм і звітів
Значення за замовчуванням	Для поля задається значення, що буде використовуватися за замовчуванням, тобто в тому разі, якщо до цього поля не буде введена інша інформація
Умова на значення	Встановлює обмеження на введені дані, інакше кажучи, не дозволяє вводити до поля такі дані, які не відповідають зазначеній умові
Повідомлення про помилку	Задає текст повідомлення, що буде відображатися в тому разі, якщо дані, введені в поле, не відповідають типу даних або обмеженням, зазначеним у властивостях <i>Умова на значення</i> , <i>Обов'язкове поле</i> , <i>Індексоване поле</i> тощо
Обов'язкове поле	Визначає режим обов'язкового введення інформації до цього поля
Індексоване поле	Встановлює режим використання індексу для цього поля, що дає змогу прискорити процедуру доступу до інформації в полі, а також задати режим, при якому до поля не можна вводити повторювані значення

Нижче наведено опис основних типів даних, використовуваних у таблицях СКБД Access [1].

Текстовий. Використовується для зберігання рядкових даних з довжиною до 255 символів. Властивість *Розмір* дає змогу задати максимальну кількість символів (до 255), які будуть міститися в рядку (за замовчуванням встановлюється *Розмір* = 50).

Поле МЕМО. Застосовується для зберігання тексту, що перевищує 255 символів і/або складається з декількох рядків (тобто тексту, у якому зустрічається символ *повернення каретки*).

Основні властивості цих типів даних, встановлюваних у нижній частині вікна Конструктора (рис. 2.8):

Формат поля – задається спеціальний формат, у якому дані будуть подані на екрані, наприклад текст буде виведений із заданими кольорами (скажімо, синім) і вирівняний по лівому краю.

Порожні рядки – встановлюється режим, при якому можна вводити в поле порожній рядок (тобто значення “”).

Порожнім значенням в Access називається спеціальне значення Null, що має ряд особливостей. Зокрема не можна поставити знак рівності між значенням Null і будь-яким іншим значенням, у тому числі ще одним Null. Крім того, значення Null не обробляються при використанні підсумкових функцій, зокрема функції *Sum* (підсумовування), наприклад при створенні звітів. Однак полям формату *Текстовий* і *Поле МЕМО* можна привласнювати порожні рядки, тобто значення “”, які потім можна використовувати при обчисленнях.

Числовий. Використовується для обробки чисел різного розміру і формату. Розмір визначається за допомогою властивості *Розмір поля* (табл. 2.7), формат числа задається властивістю *Формат поля* (табл. 2.8), який також використовується для типів даних *Грошовий* і *Лічильник*. Потрібно зазначити, що для форматів *Числовий* і *Грошовий* за замовчуванням встановлюється значення 0.

Загальні	Підстановка
Розмір поля	Довге ціле число
Нові значення	Поступово
Формат	
Підпис	
Індексовано	Так (Без повторень)
Вирівнювання тексту	Загальне

Рис. 2.8. Закладка *Властивості поля*

Таблиця 2.7

Основні значення властивості *Розмір поля типу даних Числовий*

Значення	Зміст	Діапазон	Обсяг
Байт	Цілі невід'ємні числа	$[0; 2^8-1]$	1 байт
Ціле	Цілі числа	$[-2^{15}; 2^{15}-1]$	2 байт
Довге ціле	Цілі числа	$[-2^{31}; 2^{31}-1]$	4 байт
Одинарне з рухомою крапкою	Дійсні числа з точністю до семи знаків після коми	$[-2^{128}; -2^{-149}]$ і $[2^{-149}; 2^{128}-1]$	4 байт
Подвійне з рухомою крапкою	Дійсні числа з точністю до 15 знаків після коми	$[-2^{1024}; -2^{-1074}]$ і $[2^{-1074}; 2^{1024}-1]$	8 байт
Дійсне	Дійсні числа з заданою точністю (за замовчуванням точність = 18 знаків)	$[-10^{28}; 10^{28}]$	12 байт

Таблиця 2.8

Властивість *Формат поля типів даних Числовий, Грошовий Лічильник*

Формат	Приклад
Основний	123,456
Грошовий	123,45 грн
Євро	1234,56 €
Фіксований	1234,56
З роздільниками розрядів	1 234,56
Процентний	12,34 %
Експонентний	12,3E+02

Дата/час. Застосовується при обробці значень календарних дат і часу. Цей тип даних дає змогу зберігати значення в декількох форматах (табл. 2.9). За замовчуванням встановлюється *Повний формат дати*.

Властивість *Формат поля* типу даних *Дата/Час*

Формат	Приклад
Повний формат дати	15.11.99 14:11:30
Довгий формат дати	15 листопада 1999 р.
Середній формат дати	15-ноя-99
Короткий формат дати	15.11.99
Довгий формат часу	14:11:30
Середній формат часу	2:11
Короткий формат часу	14:11

Грошовий. Використовується для зберігання грошових значень і передбачає точність до чотирьох знаків після десяткової коми і 15 знаків перед комою.

Лічильник. Застосовується для опису поля, яке використовується як первинний ключ і генерує для кожного нового запису в таблиці унікальне значення формату *Довге ціле*.

Логічний. Призначений для зберігання й обробки логічних значень, тобто *Неправда* або *Істина*. Цей тип даних передбачає використання одного з таких форматів, як властивість *Формат поля*: *Істина/Неправда*, *Так/Ні*, *Вкл/Викл* (За замовчуванням встановлюється формат *Так/Ні*).

У БД **Освіта**, що буде використовуватися як приклад, таблиця **Студенти** має поля таких форматів:

Код Студента – Лічильник (первинний ключ);

Прізвище Студента – Текстовий (розмір поля = 20 символів);

Ім'я Студента – Текстовий (розмір поля = 10 символів);

По батькові Студента – Текстовий (розмір поля = 15 символів).

2.4. Зв'язки між таблицями

Структура реляційної БД завжди розробляється так, щоб кожна таблиця, що в ній перебуває, не містила надлишкової інформації. Наприклад, у БД **Освіта** потрібно зберігати дані про оцінки, одержані кожним студентом з тієї чи іншої дисципліни [1].

Як наслідок, потрібно також якось зберігати і назви дисциплін, за які ці оцінки ставляться, крім того, зберігати прізвище, ім'я та по батькові викладачів із зазначенням дисципліни, що веде кожен з них.

Якщо для цього буде використовуватися одна таблиця, то стане очевидним нераціональне використання пам'яті комп'ютера, тому що для кожного студента доведеться зберігати у відповідному записі назви дисциплін і прізвища викладачів, які будуть повторюватися багато разів. Тому зазначену інформацію потрібно розбити на кілька таблиць, які будуть між собою взаємозалежні [1].

Студенти – інформація про студента (унікальний код студента, прізвище, ім'я та по батькові, номер групи, дата народження тощо).

Викладачі – інформація про викладача (унікальний код викладача, прізвище, ім'я та по батькові, кафедра, науковий ступінь, вчене звання тощо).

Дисципліни – інформація про дисципліни (унікальний код дисципліни, код викладача, що веде цю дисципліну, назва дисципліни і кількість годин).

Оцінки – інформація про оцінки студентів (унікальний код для кожного запису з оцінкою, код студента, код предмета та оцінка, що отримана цим студентом з певної дисципліни).

Отже, структура таблиць, що використовуються у БД **Освіта**, буде мати вигляд як табл. 2.10–2.13.

Таблиця 2.10

Структура таблиці **Студенти**

Поле	Опис
Код Студента	Лічильник (первинний ключ)
Прізвище Студента	Текстовий (20 символів)
Ім'я Студента	Текстовий (10 символів)
По батькові Студента	Текстовий (15 символів)
Дата народження Студента	Дата/час
Група Студента	Текстовий (15 символів)
...	...

Таблиця 2.11

Структура таблиці **Викладачі**

Поле	Опис
Код Викладача	Лічильник (первинний ключ)
Прізвище Викладача	Текстовий (20 символів)
Ім'я Викладача	Текстовий (10 символів)
По батькові Викладача	Текстовий (15 символів)
Кафедра Викладача	Текстовий (10 символів)
...	...

Таблиця 2.12

Структура таблиці **Дисципліни**

Поле	Опис
Код Дисципліни	Лічильник (первинний ключ)
Код Викладача	Числовий (довге ціле)
Дисципліна	Текстовий (40 символів)
Години	Числовий (ціле)

Структура таблиці **Оцінки**

Поле	Опис
Код Дисципліни	Числовий (довге ціле)
Код Студента	Числовий (довге ціле)
Оцінка	Числовий (байт)
Дата Складання	Дата/час (короткий формат дати)


Слід зазначити, що при визначенні імен можна використовувати назви, що складаються з декількох слів.

Для створених таблиць потрібно встановити зв'язки, щоб, наприклад, для кожного студента (запису з таблиці **Студенти**) визначити всі його оцінки (записи в таблиці **Оцінки**). Зв'язки між таблицями в БД використовуються при формуванні запитів, створенні звітів або розробленні форм. Як видно з табл. 2.10-2.13, у розроблюваній структурі даних існують такі взаємозв'язки:

1) таблиця **Студенти**, поле **Код Студента** ↔ таблиця **Оцінки**, поле **Код Студента**;

2) таблиця **Викладачі**, поле **Код Викладача** ↔ таблиця **Дисципліни**, поле **Код Викладача**;

3) таблиця **Дисципліни**, поле **Код Дисципліни** ↔ таблиця **Оцінки**, поле **Код Дисципліни**.

Для встановлення зв'язків можна скористатися функцією *Зв'язки*, що міститься в меню *Знаряддя бази даних*  на панелі інструментів [1]. При виконанні цієї команди на екрані з'явиться порожнє вікно. Для додавання таблиць до цього вікна потрібно вибрати таблиці та натиснути кнопку *Додати* (рис. 2.9). Потім потрібно закрити вікно *Додавання таблиці*.

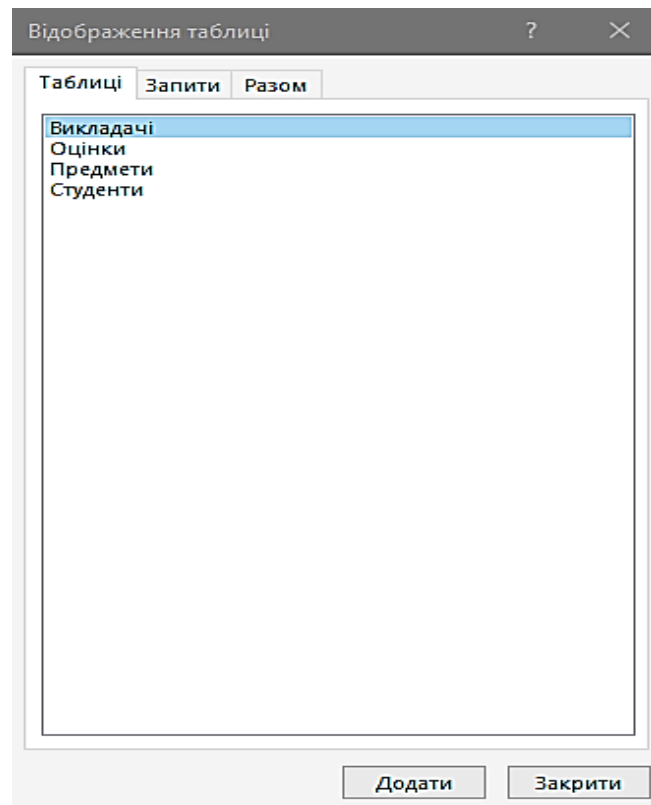


Рис. 2.9. Діалогове вікно *Додавання таблиці*

Після виконання зазначених дій на схемі даних будуть розташовані всі таблиці БД Освіта, розміщення яких можна змінити на свій розсуд (рис. 2.10) [1].

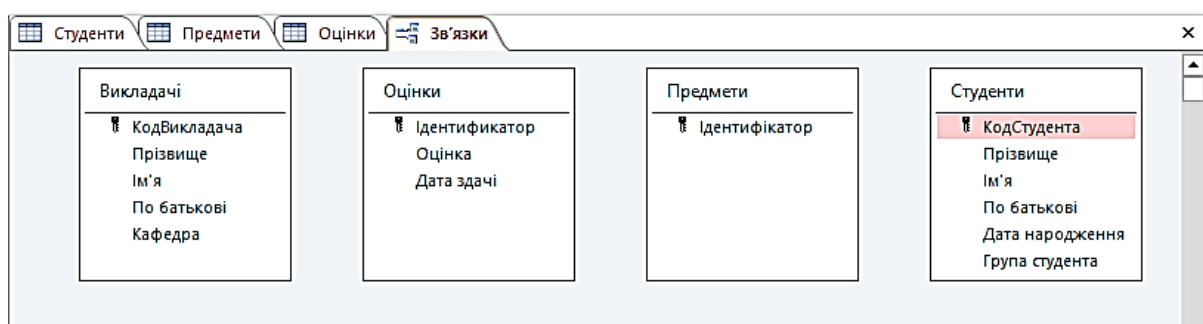
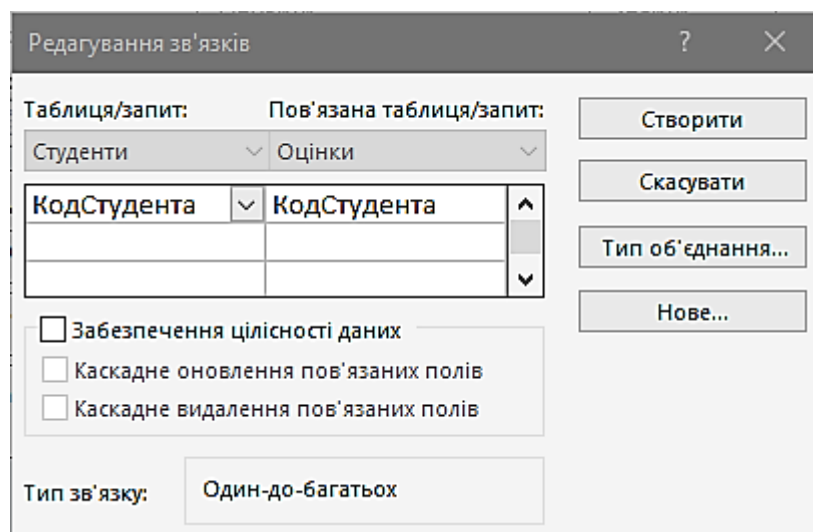


Рис. 2.10. Таблиці БД **Освіта** на схемі даних

Наступним етапом у розробленні схеми даних буде встановлення зв'язків між таблицями. Зокрема потрібно зв'язати таблиці **Студенти** та **Оцінки** за полем **Код Студента** так, щоб кожному запису з таблиці

Студенти відповідало декілька записів таблиці **Оцінки** (тому що в кожного студента може бути кілька оцінок). Інакше кажучи, між зазначеними таблицями потрібно встановити зв'язок виду «один до багатьох». Аналогічно потрібно встановити зв'язок «один до багатьох» за полем **Код Дисципліни** між таблицями **Дисципліни** і **Оцінки** (тому що з кожної дисципліни може бути кілька оцінок, виставлених різним студентам), а також зв'язок «один до багатьох» за полем **Код Викладача** між таблицями **Дисципліни** і **Викладачі** (зважаючи на припущення, що кожен викладач веде заняття не тільки з однієї дисципліни).

Щоб установити зв'язок, наприклад, по полю **Код Студента** між таблицями **Студенти** та **Оцінки**, потрібно виділити у вікні *Схема даних* поле **Код Студента** в таблиці **Студенти**, за допомогою миші «перетягнути» його зображення на поле **Код Студента** таблиці **Оцінки** і потім відпустити кнопку миші. Водночас з'явиться діалогове вікно *Зміна зв'язків*, у якому для обох таблиць має бути зазначене поле **Код Студента** (рис. 2.11) [1].



Якщо кнопка миші була відпущена в той момент, коли зображення перебувало над іншим полем, то в стовпчики *Зв'язана таблиця/запит*

(таблиця **Оцінки**) буде відображена назва цього поля. У такому разі для зміни поля, за яким потрібно зв'язати таблиці, можна вибрати назву поля **Код Студента** для таблиці **Оцінки** у списку полів, що розкривається, яке можна відобразити, попередньо клацнувши в полі *Зв'язана таблиця/запит* вікна *Зміна зв'язків*. Крім того, можна закрити вікно *Зміна зв'язків* і знову виконати «перетягування» поля.

Для того щоб у таблиці **Оцінки** не зберігалися оцінки неіснуючих студентів (тобто запису, для яких значення поля **Код Студента** не збігається з жодним зі значень поля **Код Студента** в таблиці **Студенти**), треба у вікні *Зміна зв'язків* установити прапорець *Забезпечення цілісності даних*. Після цього в таблицю **Оцінки** неможливо буде ввести запис із таким значенням поля **Код Студента**, для якого не можна буде знайти відповідний запис у таблиці **Студенти**.

Якщо встановлено прапорець *Забезпечення цілісності даних*, то у вікні *Зміна зв'язків* автоматично стають доступні ще два прапорці [1]:

– *каскадне відновлення зв'язаних полів*. При зміні значення ключового поля в «головній» таблиці (**Студенти**) автоматично змінюються значення зв'язаного поля для відповідних записів у «підлеглий» таблиці (**Оцінки**). У розглянутому випадку ключове поле **Код Студента** є лічильником, отже, значення цього поля не можуть бути змінені користувачем. Тому нема потреби встановлювати цей прапорець. А от якщо для зв'язаного поля передбачається його подальше редагування, прапорець рекомендується встановити [1];

– *каскадне видалення зв'язаних полів*. При видаленні запису з «головної» таблиці (**Студенти**) автоматично видаляються ті записи з «підлеглої» таблиці (**Оцінки**), для яких значення поля **Код Студента** збігається із значенням у вилученому записі з таблиці **Студенти**. Якщо в таблиці **Оцінки** не потрібно зберігати оцінки відрахованих студентів, то прапорець *Каскадне видалення зв'язаних полів* потрібно встановити. Якщо ж

ми передбачаємо можливість подальшого використання цих даних, або ж для запобігання помилковому видаленню даних про студента, у якого виставлено оцінки в таблиці **Студенти**, тоді цей прапорець встановлювати не потрібно.

Після того як обрано поля для встановлення зв'язку між таблицями, у вікні *Зміна зв'язків* потрібно натиснути кнопку *Створити*, після чого на схемі даних у вигляді лінії буде відображений створений зв'язок.

Слід зазначити, що тип відносин між зв'язаними таблицями Access визначає автоматично. У цьому випадку, якщо був встановлений прапорець *Забезпечення цілісності даних* (рис. 2.11), то поруч із полем **Код Студента** таблиці **Студенти** над лінією зв'язку відображується символ «1», а поруч з таблицею **Оцінки** – символ нескінченності «∞», тобто встановлено зв'язок типу «один до багатьох» (рис. 2.12).

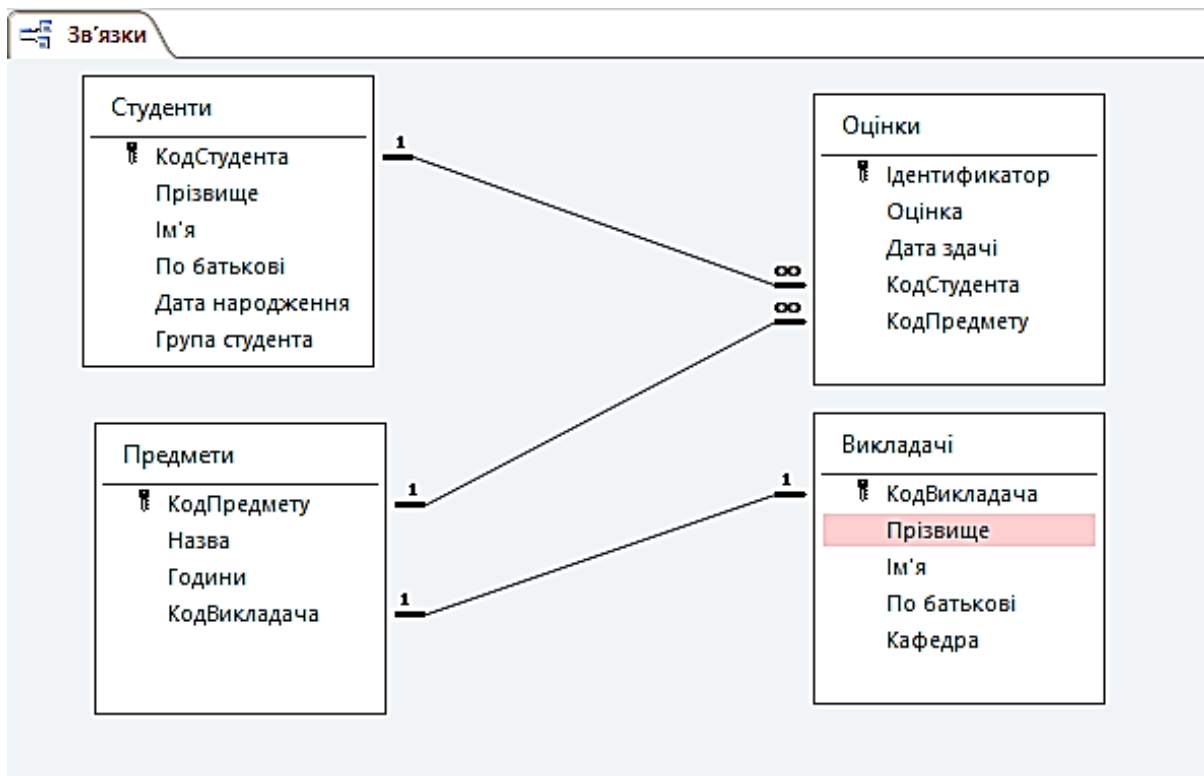






Рис. 2.12. Підсумкова схема даних БД Освіта

Щоб змінити встановлений зв'язок, потрібно виділити відповідну лінію, натиснувши на клавішу миші. Після цього зв'язок можна видалити командою *Виправлення / Видалити* або за допомогою клавіші **Delete**, а також змінити, викликавши вікно *Зміна зв'язку* командою *Зв'язки / Змінити зв'язок* або подвійним натисканням лівою клавішею миші на лінії зв'язку.

Після того як усі зв'язки між таблицями встановлено, схема даних для БД **Освіта** матиме вигляд як на рис. 2.12.

Отриману схему даних потрібно зберегти. Для цього закрити вікно *Схема даних* і відповісти ствердно на питання про збереження схеми даних у системному повідомленні про зміну макета.

При роботі з БД, що містять велику кількість таблиць, буває зручно відображувати на схемі даних тільки певну групу зв'язаних таблиць.

Наприклад, для БД **Освіта** потрібно на схемі даних відобразити тільки таблицю **Студенти** і таблицю, пов'язану з нею (**Оцінки**). Для цього потрібно очистити макет схеми даних кнопкою , додати за допомогою кнопки *Відобразити таблицю*  до схеми даних таблицю **Студенти** і після цього відобразити на схемі всі таблиці, з якими вона пов'язана, за допомогою кнопки *Прямі зв'язки* . Водночас на макеті додається таблиця **Оцінки**. З іншого боку, щоб відобразити на схемі даних усі таблиці, які мають хоча б один зв'язок, можна скористатися кнопкою *Усі зв'язки* . Для БД **Освіта** в цьому випадку на схемі даних відобразяться всі таблиці, тому що кожна з них має хоча б один зв'язок (рис. 2.12).

Якщо потрібно сховати якусь таблицю на схемі даних, то для цього потрібно натиснути на таблиці і вибрати команду *Приховати таблицю*.

Слід зазначити, що використання майстра підстановок, про який буде розказано далі, при виборі типу поля дає змогу автоматично встановлювати необхідні зв'язки між таблицями.

2.5. Використання індексів

Кожен запис, що додається до таблиці, розташовується в її кінці, отже, записи в таблиці ніяк не впорядковані. Однак найчастіше буває потрібно виконувати їхнє сортування. Це можна зробити, відсортувавши всі записи таблиці і перезаписавши їх у новому порядку. Але цю операцію довелося б робити щоразу при додаванні і зміні записів у таблиці, що значно сповільнило б обробку інформації. Тому для прискорення процесу сортування був розроблений метод, пов'язаний з перезаписом окремої таблиці, яку називають *індексом*, що складається тільки з двох полів вихідної таблиці [1].

Будь-який індекс являє собою автоматично створювану й оновлювану внутрішню таблицю Access, у якій є два стовпчики.

У першому містяться значення виразів, що являють собою вміст індексованого поля або декількох полів, що входять до індексу.

У другому стовпчику зберігається фізичний номер кожного запису таблиці, для якої обчислено значення індексу. Отже, при використанні індексів записи в таблиці розглядаються не в тому порядку, у якому вони в ній зберігаються (тобто в порядку, у якому вони вводилися до таблиці), а з урахуванням їхнього розташування в індексі.

Наприклад, для прискорення пошуку прізвища студента в таблиці **Студенти** можна встановити індекс для поля **Прізвище**.

Щоб створити індекс для одного поля (**Прізвище**), потрібно відкрити таблицю **Студенти** в режимі **Конструктор**, вибрати поле, яке ми плануємо зробити індексованим (у нашому випадку **Прізвище**), і в розділі *Властивості поля* встановити у властивості *Індексоване поле* значення *Так*.


Оскільки декілька студентів можуть мати те саме прізвище, то потрібно вибрати значення *Так (Збіги допускаються)*. З іншого боку, якщо індекс створюється для первинного ключа або будь-якого іншого

унікального параметра (наприклад **Номер залікової книжки**), то потрібно вибрати *Так (Збіги не допускаються)*, тому що значення первинного ключа для кожного запису є унікальним.

Після того як буде створений індекс для поля **Прізвище**, пошук того або другого студента за **Прізвищем** у таблиці **Студенти** буде виконуватися, зважаючи на те міркування, що всі записи прізвищ студентів відсортовані за абеткою.

Часто для прискорення процесу пошуку того або іншого запису потрібно створювати індекс на підставі не одного, а декількох полів таблиці одночасно. У такому разі індекс називається *складовим*.

Наприклад, у таблиці **Студенти** потрібно відсортувати студентів за абеткою їхніх прізвищ, імен і по батькові, а для цього створити індекс, що включає поля **Прізвище**, **Ім'я** та **По батькові**.

Щоб побудувати складовий індекс, потрібно відкрити таблицю **Студенти** в режимі **Конструктор** і потім скористатися кнопкою *Індекси* . Після цього на екрані з'явиться вікно індексів для таблиці **Студенти** – *Індекси: Студенти* (рис. 2.13) [1].

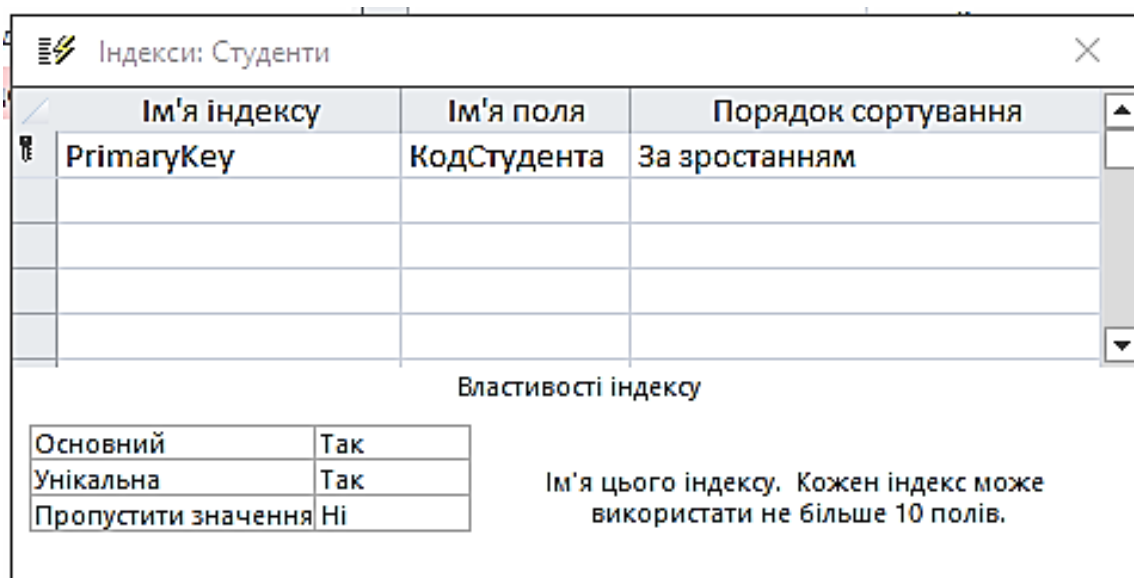


Рис. 2.13. Вікно *Індекси: Студенти*. Індекс *Прізвище*

Тут стовпчик *Індекс* містить назву індексу; у стовпчику *Ім'я поля* вибирається поле, за яким будується індекс; стовпчик *Порядок сортування* визначає напрямок сортування в індексі: за зростанням або за зменшенням.

У першому рядку розташований індекс за ключовим полем **Код Студента**, яке не можна змінювати або видаляти. У другому рядку замість індексу **Прізвище** буде створений складовий індекс **ПІБ**. Для цього спочатку потрібно ввести нову назву індексу в стовпчику *Індекс* (**ПІБ** замість **Прізвище**). Потім потрібно, залишивши назву поля **Прізвище**, у двох наступних комірках стовпчика *Ім'я поля* по черзі вибрати зі списку, що розкривається, назви полів **Ім'я** та **По батькові** (рис. 2.14). Після цього можна закрити вікно індексів і зберегти зміни в структурі таблиці **Студенти**.

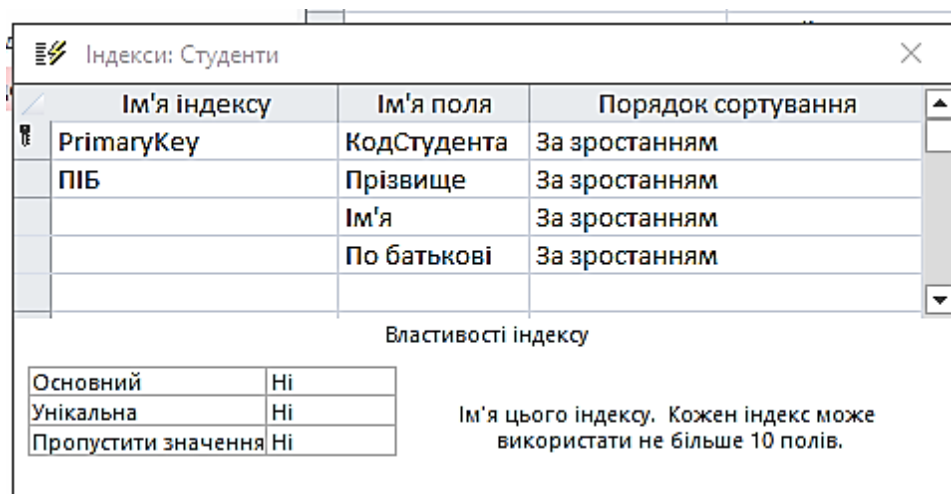


Рис. 2.14. Вікно *Індекси: Студенти*. Індекс *ПІБ*

Щоб видалити той або інший індекс, наприклад з індексу **ПІБ** виключити поле **По батькові**, потрібно у вікні індексів відобразити для потрібного поля (**По батькові**) правою кнопкою миші контекстне меню й вибрати в ньому команду *Видалити рядок* або виділити поле, натиснувши на його області виділення і клавішу **Delete**. З іншого боку, аналогічно можна вставити порожній рядок для індексу. Для цього потрібно виділити рядок,

перед яким потрібно виконати вставку, і натиснути клавішу **Insert** або виконати в контекстному меню цього ж рядка команду *Додати рядок*.

2.6. Операції з таблицями

При обробці БД часто виникає необхідність виконати операції над створеними таблицями, наприклад перейменувати, створити копію або видалити таблицю. Нижче буде докладно розглянута реалізація цих дій [1].

Перейменування. Щоб перейменувати створену таблицю, потрібно виділити її у вікні БД і в контекстному меню вибрати *Перейменування* або просто натиснути на виділеній назві. Після цього буде увімкнений режим **Перейменування таблиці**, у якому їй можна задати нове ім'я (рис. 2.15) [1].

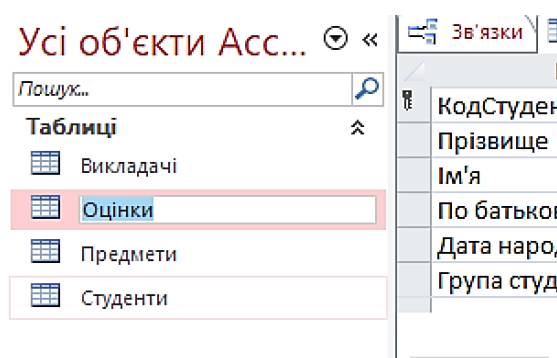




Рис. 2.15. Перейменування таблиці Оцінки у БД Освіта

Копіювання. Якщо потрібно внести серйозні корективи до наявної таблиці, то можна попередньо створити її резервну копію з іншим ім'ям, і потім, якщо буде потреба, відновити цю копію під її старим ім'ям, для цього видалити невдалий варіант старої таблиці, після чого перейменувати створену резервну копію, відновивши її початкове ім'я.

Наприклад, щоб створити копію таблиці **Оцінки**, потрібно виділити її у вікні БД **Освіта**, після чого виконати операцію копіювання в буфер кнопкою *Копіювати* , потім вставити з буфера кнопкою *Вставити* .

Після вставлення на екрані з'явиться діалогове вікно *Вставлення таблиці* (рис. 2.16), у якому потрібно вказати нове ім'я таблиці (наприклад **Копія Успішність**), вибрати як *Параметр вставлення* пункт *структура і дані* і натиснути кнопку *ОК*. У результаті у вікні БД **Освіта** з'явиться нова таблиця з назвою **Копія Успішність**, до якої будуть скопійовані структура і дані таблиці **Оцінки**.

Якщо в БД уже існує таблиця з зазначеним для копії ім'ям, то в цьому випадку відобразиться вікно повідомлення, у якому буде запропоновано замінити наявну таблицю. Однак у такій ситуації потрібно перервати операцію копіювання таблиці, тобто натиснути кнопку *Ні* і задати у вікні *Вставлення таблиці* інше ім'я для копії.

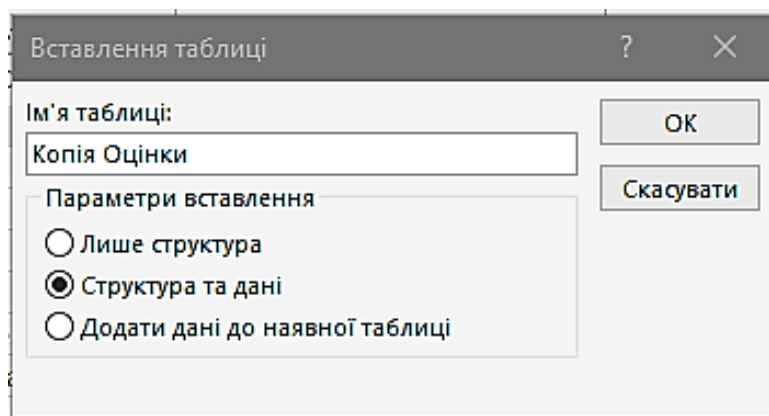


Рис. 2.16. Діалогове вікно *Вставлення таблиці*

Видалення. Щоб видалити таблицю із БД, її спочатку потрібно виділити у вікні БД, після чого скористатися клавішею **Delete** або виконати команду *Виправлення / Видалити*. Потім з'явиться вікно попередження, у якому потрібно підтвердити видалення таблиці, вибравши кнопку *Так*. У тому випадку, якщо видаляється таблиця, яка мала зв'язки з іншими таблицями, Access відобразить ще одне повідомлення про необхідність видалення всіх зв'язків цієї таблиці. Щоб підтвердити видалення зв'язків, потрібно вибрати кнопку *Так*.

2.7 Стискання бази даних

При інтенсивній обробці інформації в БД, наприклад при видаленні наявних і створенні нових об'єктів (таблиць, запитів і т. д.), файл БД рано або пізно стане зайво фрагментованим, тобто в ньому будуть виникати «порожнечі». Оскільки наявність таких «порожнеч» у файлі призводить до збільшення його обсягу, то час від часу БД потрібно «стискати», тобто видаляти з файлу БД ділянки пам'яті, що не використовуються [1].

Щоб стиснути БД, потрібно закрити всі відкриті БД і виконати команду *Сервіс | Службові програми | Стиснути і відновити базу даних*. Після цього з'явиться діалогове вікно *База даних для стиску*, у якому потрібно вибрати БД (наприклад **Освіта**) і натиснути кнопку *Стиснути*.

Слід зазначити, що описану команду для стискання БД можна виконати й при відкритій БД. У цьому разі операція стискання буде виконана для БД, що була відкрита.

Контрольні запитання до розділу 2

1. Що дозволяє команда *Режим таблиці*?
2. Що дозволяє команда *Майстер таблиць*?
3. Що дозволяє команда *Конструктор*?
4. Що дозволяє команда *Імпорт таблиць*?
5. Що дозволяє команда *Зв'язок з таблицями*?

Розділ 3. ОБРОБКА ДАНИХ У ТАБЛИЦЯХ

3.1. Можливості обробки даних

Як приклад у цьому розділі буде розглянута навчальна БД **Освіта**, структура якої була докладно описана в попередньому розділі. Дані, що зберігаються в її таблицях, наведені в табл. 3.1–3.4.

Таблиця 3.1

Таблиця Студенти

КодСтудента	Прізвище	Ім'я	По батькові
1	Жукова	Марина	Володимирівна
2	Морозов	Ігор	Олегович
3	Петров	Михайло	Олександрович
4	Ігнатов	Максим	Валерійович
5	Волков	Юрій	Олексійович
6	Волошина	Вікторія	Іванівна
7	Литвинов	Євгеній	Валерійович
8	Гущина	Олена	Валеріївна
9	Кузнецов	Руслан	Олексійович
10	Семененко	Світлана	Станіславівна

Таблиця 3.2

Таблиця Викладачі

КодВикладача	Прізвище	Ім'я	По батькові
1	Вікуліна	Валентина	Іванівна
2	Костиркін	Олег	Володимирович
3	Казанко	Віталій	Володимирович

Таблиця 3.3

Таблиця **Предмети**

КодПредмета	КодВикладача	Назва	КількістьГодин
1	1	Фізика	34
2	2	Хімія	68
3	3	Математика	68
4	4	Економіка	17
5	5	Філософія	17

Таблиця 3.4

Таблиця **Оцінки**

КодОцінки	КодСтудента	КодПредмета	Оцінка	ДатаЗдачі
1	1	1	5	02.01.21
2	1	2	5	02.01.21
3	2	1	4	02.01.21
4	2	2	4	02.01.21
5	3	5	3	02.01.21
6	3	4	4	02.01.21
7	3	3	5	02.01.21
8	4	1	5	02.01.21
9	4	2	4	02.01.21



Ключові поля в таблицях (наприклад **КодСтудента** в таблиці **Студенти**) заповнюються автоматично при введенні нових даних.

Далі будуть розглянуті основні можливості обробки даних у режимі **Таблиці**. Водночас як приклад буде використовуватися таблиця **Студенти** (рис. 3.1).

	КодСтудент	Прізвище	Ім'я	По батькові	Дат
+	1	Жукова	Марина	Володимирів	
+	2	Морозов	Ігор	Олегович	
+	3	Петров	Михайло	Олександров	
+	4	Ігнатов	Максим	Валерійович	
+	5	Волков	Юрій	Олексійович	
+	6	Волошина	Вікторія	Іванівна	
+	7	Литвинов	Євген	Валерійович	
+	8	Гущина	Олена	Валеріївна	
+	9	Кузнецов	Руслан	Олексійович	
+	10	Семененко	Світлана	Станіславівна	
*	(Новий)				


Рис. 3.1. Таблиця Студенти

Сортування даних. Коли в Access відкривається будь-яка таблиця, то записи в ній подані в тому порядку, у якому вони були введені. Щоб змінити порядок відображення записів, можна скористатися можливостями *сортування*. Щоб відсортувати записи в таблиці за зростанням або зменшенням значень певного стовпчика, цей стовпчик спочатку потрібно виділити, потім вибрати пункт *Записи / Сортування* і вибрати одну з команд [1]:

- сортування за зростанням (кнопка 
- сортування по зменшенням (кнопка 

Водночас потрібно мати на увазі, що встановлюване в такий спосіб сортування впливає тільки на відображення записів, але не впливає на їхнє фізичне розташування в таблиці або наявні індекси.

Пошук і заміна даних. Щоб знайти або замінити в таблиці дані певного виду, наприклад у всіх датах, які зберігаються в таблиці, замінити рік 99 на

1999, потрібно натиснути кнопку  Знайти, після чого буде відображене діалогове вікно *Пошук і заміна* (рис. 3.2). У цьому вікні є нижче наведені поля для настроювання параметрів пошуку [1].

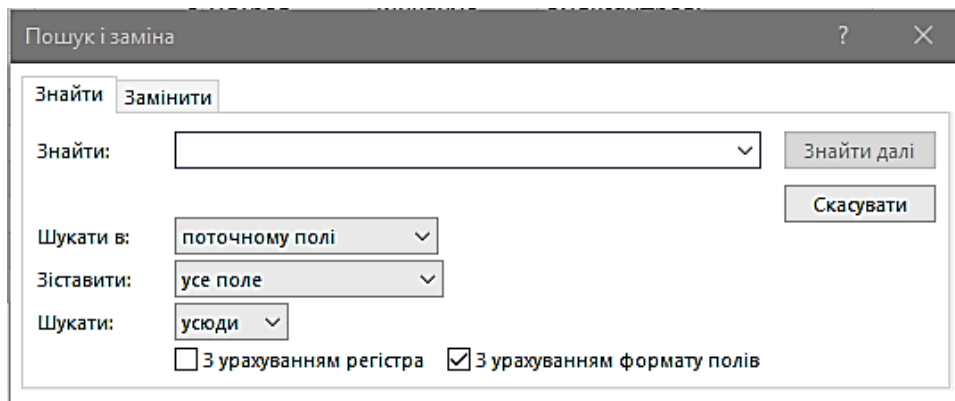


Рис. 3.2. Діалогове вікно *Пошук і заміна*

Зразок – рядок для пошуку. Наприклад, щоб знайти всі сполучення символів “99”, потрібно в це поле ввести значення 99. Можна також використати спеціальні символи:

* – заміняє рядок будь-якої довжини, що складається з довільних символів;

? – заміняє один будь-який символ.

Зокрема, щоб знайти всі рядки, у яких є текст, що починається на «а», можна в поле **Зразок** ввести значення *a**.

Пошук в – визначає режим пошуку або в поточному полі активної таблиці, або у всій таблиці.

Збіг – визначає тип пошуку. Наприклад, якщо в поле **Зразок** введено 99, то для значень 99, 1998 і 991 вибір того або другого режиму призведе до таких результатів:

- *З будь-якою частиною поля* – знайде значення і **99**, і **1998**, і **991**;
- *Поля цілком* – знайде тільки значення **99**;
- *З початку поля* – знайде значення **99** і **991**.

Перегляд – визначає напрямок пошуку від поточного запису:

- *Нагору* – до першого запису;
- *Вниз* – до останнього запису;
- *Всі* – пошук у всій таблиці.

З урахуванням реєстра – при пошуку враховується реєстр (тобто рядкові і прописні букви), що використався при зазначенні рядка в полі *Зразок*.


З урахуванням формату полів – при пошуку враховується формат відображення, встановлений для того або іншого поля. Наприклад, дата, збережена у форматі 10.02.99, відображується в таблиці як 10-лют.-99.

Замінити на – рядок для заміни знайденого значення (тільки на вкладці *Заміна*).


Після того як знайдене чергове значення, для пошуку наступного використовується *кнопка Знайти далі*. З іншого боку, у режимі **Заміни** використовуються ще дві кнопки – *Замінити* (заміна чергового знайденого значення на зазначене) і *Замінити всі* (заміна всіх знайдених значень на зазначене) [1].

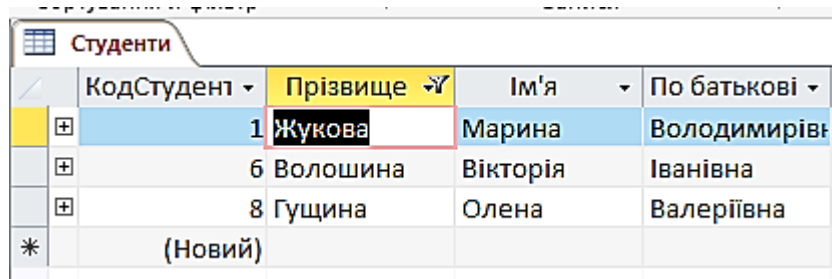
Фільтрація даних. При роботі з таблицями часто буває потрібно відображувати не всі записи, а тільки ті, які відповідають певним умовам. Для цього використовуються режими **Фільтрація** [1].

1. *Фільтр по виділеному*. Щоб задати режим, при якому виводилися б тільки ті записи, у яких у певному полі наявне певне значення, потрібно виділити цю частину поля і скористатися режимом **Фільтр по виділеному** [1].

Нехай потрібно встановити такий фільтр, щоб у таблиці відображувалися тільки ті студенти, прізвища яких закінчуються на букву «а». Для цього в полі **Прізвище** потрібно виділити останню букву «а» у прізвищі, наприклад Жукова. Потім потрібно натиснути кнопку . Після цього будуть відображені записи для студентів: Жукова, Волошина і Гущина (рис. 3.3).

Найчастіше буває потрібно встановити режим фільтрації не за однією, а за декількома умовами. Наприклад, щоб і у прізвищі, і в імені останньою була буква «а». У цьому випадку потрібно спочатку задати режим **Фільтр**

по виділеному для прізвища (рис. 3.3), а в отриманій таблиці задати той самий режим для імені, тобто виділити останню «а» у будь-якому імені і натиснути кнопку  (рис. 3.4).




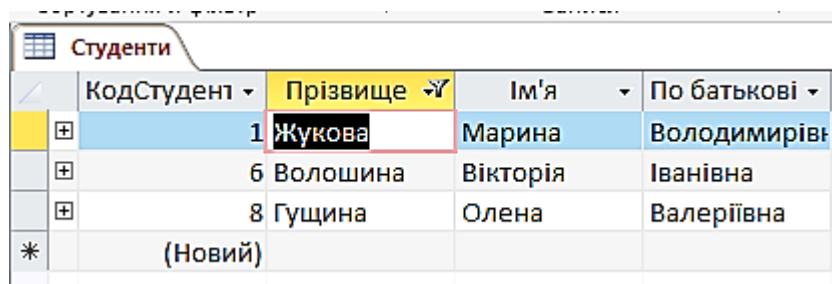
	КодСтудент	Прізвище 	Ім'я	По батькові
+	1	Жукова	Марина	Володимирів
+	6	Волошина	Вікторія	Іванівна
+	8	Гущина	Олена	Валеріївна
*	(Новий)			

Рис. 3.3. Фільтрація за останньою буквою «а»







	КодСтудент	Прізвище 	Ім'я 	По батькові
+	1	Жукова	Марина	Володимирів
+	6	Волошина	Вікторія	Іванівна
+	8	Гущина	Олена	Валеріївна
*	(Новий)			

Рис. 3.4. Фільтрація за ім'ям і прізвищем

Щоб скасувати режим фільтрації, можна скористатися кнопкою *Видалити фільтр* . Слід зазначити, що вмикання цієї кнопки виконує установлення останнього застосованого фільтра (*Застосування фільтра*), а її відключення – скасування фільтрації (*Видалити фільтр*).

2. *Звичайний фільтр*. Режим фільтрації по виділеному використовується в тому випадку, коли потрібно відображувати тільки ті записи, що задовольняють усі зазначені обмеження (наприклад остання буква «а» в прізвищі) [1].

Якщо ж потрібно відображувати всі записи, які задовольняють хоча б одну з зазначених умов (наприклад остання буква «а» в імені *Або* прізвищі),

то в цьому випадку потрібно використати звичайний фільтр, який можна встановити кнопкою . У лівому нижньому кутку вікна фільтрації є вкладки *Знайти* і *Або*, при виборі яких у полях відображуються встановлені обмеження. Наприклад, після установалення розглянутих вище обмежень на ім'я і прізвище (остання буква «а») на вкладці *Знайти* в полях **Прізвище** і **Ім'я** будуть встановлені фільтри Like “*а” (рис. 3.5).

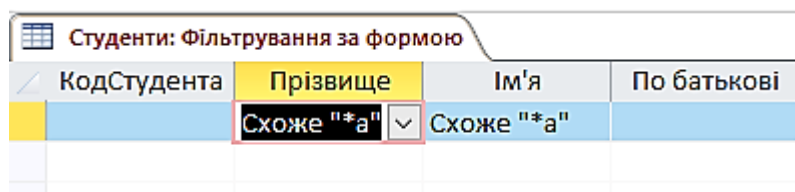


Рис. 3.5. Зміна фільтра

Для задавання будь-яких умов в Access використовуються декілька операторів порівняння (табл. 3.5), а також логічні оператори (табл. 3.6).

Таблиця 3.5

Оператори порівняння

Оператор	Опис
=	Дорівнює
<	Менше
>	Більше
<=	Менше або дорівнює
>=	Більше або дорівнює
<>	Не дорівнює
IN	Перевірка на рівність значення якому-небудь елементу списку, що вказується в дужках оператора IN
BETWEEN	Перевірка на знаходження значення в зазначеному діапазоні (границі діапазону вказуються за допомогою логічного оператора AND)
LIKE	Перевірка значення на відповідність зазначеному шаблону

Логічні оператори

Оператор	Опис
AND	Перевіряє відповідність значення обом зазначеним умовам
OR	Перевіряє відповідність значення одній із зазначених умов

Як приклад застосування операторів будуть розглянуті кілька умов (табл. 3.7). Слід зазначити, що назви операторів прийнято писати або великими буквами (LIKE), або починати з великих (Like) [1].


Наприклад, потрібно встановити єдине обмеження, що полягає в тому, що перша буква поля **Прізвище** має перебувати в діапазоні від «А» до «М». У цьому випадку можна в режимі зміни фільтра задати для поля **Прізвище** обмеження `>= «А*» AND <=«М*»` або обмеження `BETWEEN «А*» AND «М*»` (рис. 3.6).

Приклади використання операторів

Приклад	Опис
LIKE «*а»	Вміст текстового поля закінчується на «а»
LIKE «*а» OR «*в»	Вміст текстового поля закінчується на «а» або на «в»
IN («Волков»; «Морозов»)	Вмістом текстового поля є або рядок «Волков», або рядок «Морозов»
BETWEEN 1 AND 10	Вміст числового поля міститься в діапазоні від 1 до 10
BETWEEN «А*» AND «М*»	Перша буква текстового поля міститься в діапазоні від «А» до «М»
>= «А*» AND <=«М*»	Перша буква текстового поля міститься в діапазоні від «А» до «М»

Студенти: Фільтрування за формою		
КодСтудента	Прізвище	Ім'я
	>="А*" And <="М*"	

Рис. 3.6. Задавання умови фільтрації

Після застосування цього фільтра (кнопка ) у таблиці **Студенти** будуть відображені тільки ті записи, у яких значення поля **Прізвище** починається на букву, що міститься в діапазоні від А до М (рис. 3.7).

Студенти				
КодСтудент	Прізвище	Ім'я	По батькові	
1	Жукова	Марина	Володимирівна	
4	Ігнатов	Максим	Валерійович	
5	Волков	Юрій	Олексійович	
6	Волошина	Вікторія	Іванівна	
7	Литвинов	Євген	Валерійович	
8	Гущина	Олена	Валеріївна	
9	Кузнецов	Руслан	Олексійович	
*	(Новий)			

Рис. 3.7. Результат застосування фільтра

Щоб задати обмеження на кілька полів, при яких відображувалися б усі записи, які задовольняють хоча б одне з зазначених обмежень, можна скористатися вкладкою *Або* вікна фільтра. Наприклад, потрібно відобразити тільки тих студентів, у яких ім'я або прізвище закінчується на букву «а». Для цього спочатку потрібно на вкладці *Знайти* вікна фільтра вказати для поля **Прізвище** умову LIKE «*а» (рис. 3.8). Після цього вибрати вкладку *Або* і вказати для поля *Ім'я* умову LIKE «*а» (рис. 3.9).

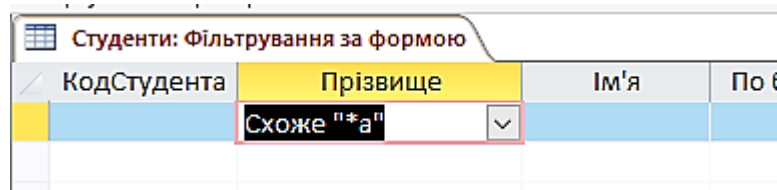


Рис. 3.8. Вкладка *Знайти* Вікна фільтра

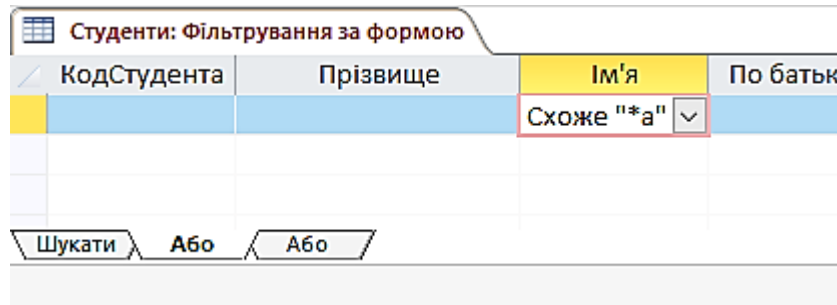




Рис. 3.9. Вкладка *Або* Вікна фільтра

Слід також зазначити, що при виборі вкладки *Або* автоматично з'являється наступна вкладка *Або*, тобто можна вказати кілька умов, одна з яких має виконуватися. Якщо потрібно видалити всі встановлені обмеження, то при відкритому вікні фільтра потрібно скористатися кнопкою *Очистити бланк* . [1]

Після встановлення обмежень потрібно застосувати фільтр до таблиці (кнопка ) , у результаті чого будуть відображені всі записи, для яких виконується хоча б одна з зазначених умов (рис. 3.10).

Студенти					
	КодСтудент	Прізвище	Ім'я	По батькові	Дата
	1	Жукова	Марина	Володимирівна	
	6	Волошина	Вікторія	Іванівна	
	8	Гущина	Олена	Валеріївна	
	10	Семененко	Світлана	Станіславівна	
*	(Новий)				

Рис. 3.10. Результат застосування фільтра

3. *Розширений фільтр*. Якщо можливостей звичайної фільтрації не вистачає, наприклад потрібно не тільки відображувати певні записи, що задовольняють ті або інші умови, але й виконувати сортування даних по декількох стовпчиках, то в цьому випадку можна використати розширений фільтр [1].

Для цього потрібно натиснути кнопку *Розширений фільтр*, після чого відобразиться вікно розширеного фільтра (рис. 3.11). Потім вибрати поля, на які будуть накладатися обмеження (*Поле*), а також напрямок їх сортування (*Сортування*). Після цього вказати обмеження на вміст обраного поля (*Умова відбору*).

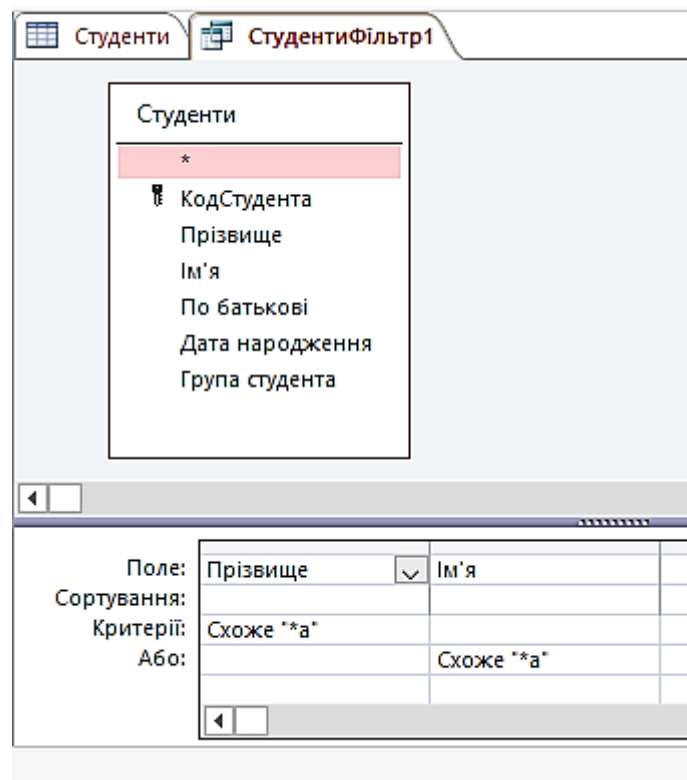



Рис. 3.11. Таблиця **Студенти**. Розширений фільтр

Наприклад, потрібно відображувати тільки тих студентів, імена яких починаються на букву «М», а по батькові – на букву «В», причому їх відсортувати за абеткою (прізвище, ім'я і по батькові). Для цього треба в першу комірку параметра *Поле* вибрати зі списку поле **Прізвище** (або

перетягнути його в цю комірку зі структури таблиці у верхній частині вікна розширеного фільтра). Потім в другій комірці вибрати поле **Ім'я** і встановити умову відбору Like «М*». Після цього в третій комірці параметра *Поле* вибрати **По батькові** й вказати умову відбору Like «В*». На закінчення потрібно вказати для всіх полів режим сортування за зростанням (рис. 3.12) і скористатися кнопкою  - в таблиці будуть відображені тільки ті записи, що задовольняють зазначені умови, відсортовані в зазначеному напрямку (спочатку – сортування за прізвищем, потім – за ім'ям, і після цього – по батькові) (рис. 3.13).

Поле:	Прізвище	Ім'я	По батькові	
Сортування:	За зростанням	За зростанням	За зростанням	
Критерії:		Схоже "М*"	Схоже "В*"	
Або:				

Рис. 3.12. Таблиця **Студенти**. Параметри розширеного фільтра

КодСтудент	Прізвище	Ім'я	По батькові	Д
	Жукова	Марина	Володимирів	
4	Ігнатов	Максим	Валерійович	
*	(Новий)			

Рис. 3.13. Результат застосування розширеного фільтра

3.2. Форматування таблиці

Як правило, не завжди ті налаштування для подання таблиці, встановлені за замовчуванням, повністю влаштовують користувача. Перебуваючи в режимі перегляду інформації таблиці, за допомогою команд

меню *Формат* можна відформатувати відкриту таблицю, наприклад встановити нові параметри шрифту, сховати або відобразити стовпчики тощо. У меню формат є нижченаведені основні команди [1].

Шрифт, Режим таблиці та Висота рядка. Можна встановити параметри шрифту, що буде використовуватися в меню *Основне*, у якому можна встановити необхідні параметри шрифту: тип, розмір, вид або кольори, а також вибрати вид оформлення таблиці, відобразити або приховати лінії сітки, а також встановити для звичайного типу оформлення кольори фону і ліній сітки або вид границь і ліній у таблиці (рис. 3.14)

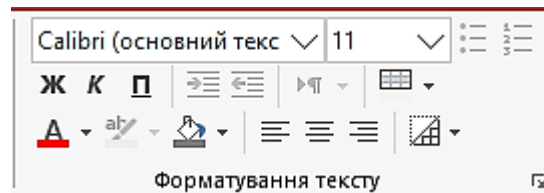


Рис. 3.14. Діалогове вікно *Шрифт*

Ширина стовпчика. Для зміни ширини стовпчика потрібно його виділити (або встановити в ньому курсор) і вибрати контекстне меню *Ширина поля* [1].

Після цього в діалоговому вікні *Ширина стовпчика* (рис. 3.15) встановити необхідну ширину і натиснути кнопку *ОК*. Можна також скористатися кнопкою *По ширині даних*, однак слід мати на увазі, що водночас ширина буде встановлена за максимальним значенням цього стовпчика з тих його значень, які в цей момент відображені на екрані.

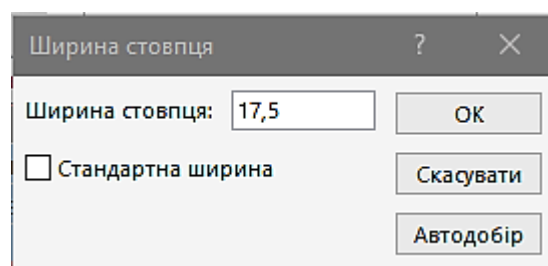


Рис. 3.15. Діалогове вікно *Ширина стовпчика*

Якщо жодне з установлених значень висоти рядка або ширини стовпчика неприйнятне, можна скористатися прапорцем *Стандартна*, щоб повернути стандартне значення ширини стовпчика або висоти рядка.

Сховати стовпчики. Щоб не відображувати в таблиці деякі стовпчики (один або декілька), спочатку їх потрібно виділити, а потім виконати команду *Формат / Сховати стовпчики* [1].

Для виділення декількох стовпчиків, розташованих поруч, тобто діапазону стовпчиків, потрібно виділити один із крайніх стовпчиків діапазону (наприклад лівий) і потім протягнути виділення до правого стовпчика. З іншого боку, щоб виділити довільно розташовані стовпчики, потрібно при виділенні чергового стовпчика утримувати натиснутою клавішу **Ctrl**. Можна також зменшити ширину стовпчика таким чином, щоб його не було видно, за допомогою миші або встановити для нього ширину 0.

Відобразити стовпчики. Якщо потрібно показати заховані раніше стовпчики, то в цьому випадку потрібно в контекстному меню вибрати *Відобразити поля*, потім у діалоговому вікні *Відображення стовпчиків* відзначити стовпчики, які потрібно показати (рис. 3.16), і натиснути кнопку *Закрити* [1].

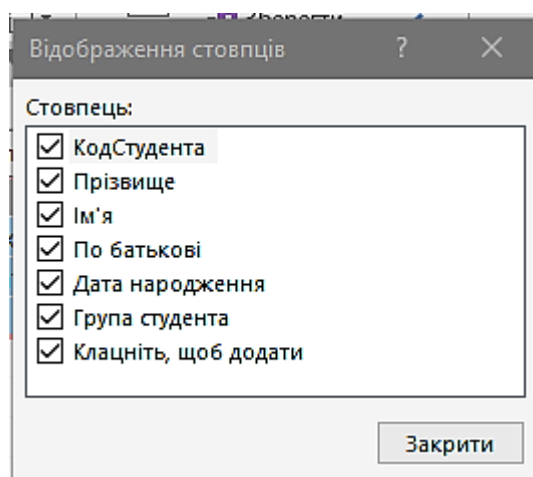


Рис. 3.16. Діалогове вікно *Відображення стовпчиків*

Закріпити стовпчики. Часто всі стовпчики таблиці не вміщаються у вікні, тому доводиться виконувати прокручування вмісту полів таблиці (ліворуч-праворуч). У цьому випадку може виникнути необхідність того, щоб певний стовпчик (або кілька стовпчиків) не прокручувалися, а залишалися на місці. Щоб виконати закріплення стовпчика, потрібно його виділити або встановити на ньому курсор, і потім виконати команду *Формат | Закріпити стовпчики.*

Звільнити всі стовпчики. Якщо у встановленому раніше закріпленні стовпчиків уже нема потреби, можна скасувати цей режим за допомогою команди *Формат | Звільнити всі стовпчики.*

3.3. Використання підтаблиць

Структура реляційної БД, як правило, створюється таким чином, що в одних таблицях поряд з інформативними даними містяться коди, за якими можна знайти зв'язані дані з інших таблиць [4, 5].

Наприклад, у БД **Освіта** таблиця **Студенти** пов'язана за полем **Код Студента** з таблицею **Оцінки**, у якій містяться оцінки студентів. Очевидно, працювати з таблицею **Студенти** було б зручніше, якби в ній для кожного студента можна було б відобразити його оцінки, тобто записи з таблиці **Оцінки**, які відповідають його індивідуальному коду.

В Access є можливість відображення в одній таблиці даних з іншої таблиці, якщо між цими таблицями є зв'язок [1].

Одержувана в такий спосіб структура «таблиця в таблиці» називається *підтаблицею*. Вони автоматично створюються відразу після визначення зв'язків між таблицями. Водночас поруч із кожним записом у таблиці з'являється зображення хрестика, при натисканні на який буде розгорнута структура підтаблиці. Наприклад, для таблиці **Студенти** така структура буде створена на основі таблиці **Оцінки** (рис. 3.17). Слід зазначити, що дані

в підтаблицях можна редагувати так само, як і у звичайних таблицях (наприклад додати нову оцінку, змінити або видалити введену раніше).

The screenshot shows a database interface with a table named 'Студенти'. The main table has columns: КодСтудент, Прізвище, Ім'я, По батькові, and Дата народ. The first row shows student 1, Жукова Марина Володимирівна. A sub-table 'Оцінки' is expanded for student 1, with columns: Ідентифікація, Оцінка, Дата здачі, КодПредме, and Кількість. The sub-table contains three rows: one with ID 1, score 5, and quantity 1; another with ID 2, score 5, and quantity 2; and a new row marked with an asterisk and '(Новий)', with score 0 and quantity 0. Below the sub-table, other students are listed: 2 Морозов Ігор Олегович, 3 Петров Михайло Олександрович, and 4 Ігнатов Максим Валерійович.

КодСтудент	Прізвище	Ім'я	По батькові	Дата народ																				
1	Жукова	Марина	Володимирівна																					
<table border="1"> <thead> <tr> <th>Ідентифікація</th> <th>Оцінка</th> <th>Дата здачі</th> <th>КодПредме</th> <th>Кількість</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>5</td> <td></td> <td>1</td> <td></td> </tr> <tr> <td>2</td> <td>5</td> <td></td> <td>2</td> <td></td> </tr> <tr> <td>*</td> <td>(Новий)</td> <td></td> <td>0</td> <td></td> </tr> </tbody> </table>					Ідентифікація	Оцінка	Дата здачі	КодПредме	Кількість	1	5		1		2	5		2		*	(Новий)		0	
Ідентифікація	Оцінка	Дата здачі	КодПредме	Кількість																				
1	5		1																					
2	5		2																					
*	(Новий)		0																					
2	Морозов	Ігор	Олегович																					
3	Петров	Михайло	Олександрович																					
4	Ігнатов	Максим	Валерійович																					

Рис. 3.17. Таблиця **Студенти** з підтаблицею **Оцінки**

У меню *Основне / Записи* є команди для роботи з підтаблицями поточної таблиці. Наприклад, щоб для кожного запису відкрити підтаблиці, можна скористатися командою *Записи | Підтаблиця | Розгорнути все*. З іншого боку, щоб згорнути всі розкриті підтаблиці, потрібно виконати команду *Записи | Підтаблиця | Згорнути все* [1].

Однак не у всіх випадках автоматично створювані підтаблиці будуть відповідати вимогам користувача. Щоб видалити наявну підтаблицю, потрібно скористатися командою *Записи | Підтаблиця | Видалити*. У цьому випадку підтаблиця буде вилучена, і користувач для цієї таблиці може створювати підтаблицю на свій розсуд [1].

Щоб створити нову підтаблицю, потрібно у відкритій таблиці виконати команду *Записи | Підтаблиця*. Потім у діалоговому вікні *Вставка підтаблиці* (рис. 3.18) можна вибрати таблицю (або запит), дані з якої будуть відображатися у підтаблиці. Для таблиці **Студенти** як підтаблицю потрібно на вкладці *Таблиці* вибрати **Оцінки**. Потім як параметри *Підлеглі поля* і *Основні поля* вказати поля, за якими виконується зв'язок (у цьому випадку це **Код Студента**), і натиснути кнопку *ОК*. У результаті буде сформована підтаблиця, створена раніше автоматично (рис. 3.18) [1].

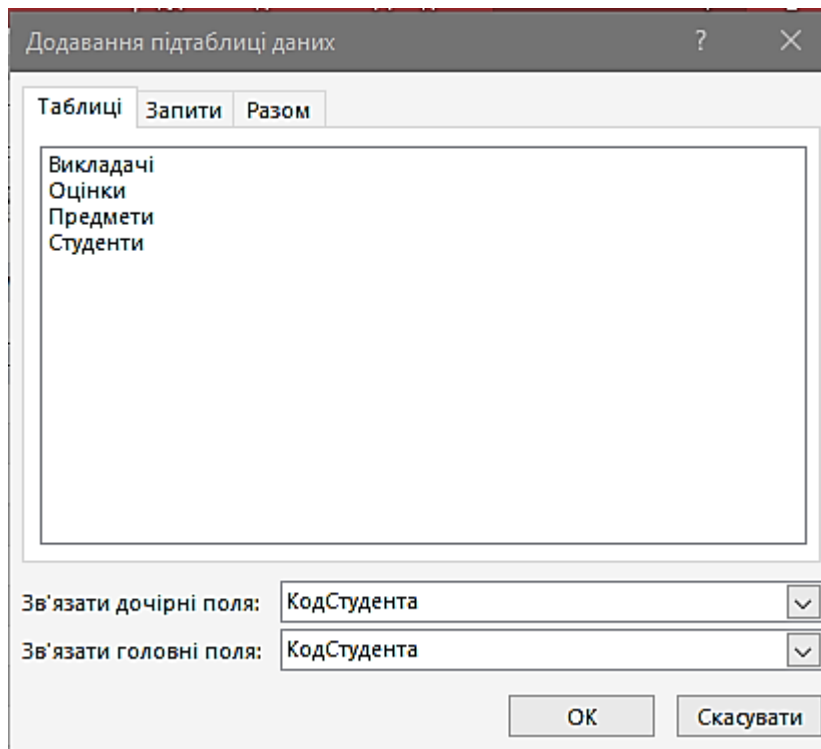


Рис. 3.18. Діалогове вікно *Вставка підтаблиці*

Можна також створити складну структуру, що складається з декількох підтаблиць [1].

Наприклад, було б зручно відображувати для кожної оцінки назву дисципліни, за яку вона отримана. Для цього потрібно розгорнути підтаблицю для одного з записів (рис. 3.19) і встановити в ній курсор. Потім виконати команду *Записи / Підтаблиця*, вибрати для зв'язку таблицю **Предмети** і вказати як сполучне поле **Код Предмета**. У результаті буде отримана структура «підтаблиця у підтаблиці», тобто для кожної оцінки студента, що цікавить, можна відобразити відповідний предмет. Водночас, якщо в отриманій підтаблиці «другого рівня» автоматично була створена власна підтаблиця «третього рівня», її потрібно видалити, тому що в результаті може вийти, що підтаблиці посилаються одна на одну. Зокрема для оцінок відображуються предмети, для яких відображуються оцінки, і т. д.

Можна створити підтаблицю третього рівня, у якій для кожної дисципліни буде відображатися прізвище викладача. Для цього треба в таблиці **Студенти** розгорнути підтаблицю **Оцінки**, у якій для будь-якого запису відобразити підтаблицю **Предмети**. Потім встановити в останню підтаблицю курсор, знову виконати команду *Записи | Підтаблиця*. Після цього вибрати таблицю **Викладачі** і вказати поле для зв'язування **Код Викладача**. У результаті вийде структура, що складається з чотирьох рівнів: основна таблиця **Студенти** і три вкладені підтаблиці **Оцінки**→**Предмети**→**Викладачі** (рис. 3.19) [1].

Студенти																																																																													
КодСтудент	Прізвище	Ім'я	По батькові	Дата народ																																																																									
1	Жукова	Марина	Володимирів																																																																										
<table border="1"> <thead> <tr> <th>Ідентифика</th> <th>Оцінка</th> <th>Дата здачі</th> <th>КодПредме</th> <th>Клацність, t</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>5</td> <td></td> <td>1</td> <td></td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Назва</th> <th>Години</th> <th>КодВикладч</th> <th>Клацність, t</th> </tr> </thead> <tbody> <tr> <td>Фізика</td> <td>34</td> <td>1</td> <td></td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Прізвище</th> <th>Ім'я</th> <th>По батькові</th> <th>Кафедра</th> <th>Клацніт</th> </tr> </thead> <tbody> <tr> <td>Зікулина</td> <td>Валентина</td> <td>Іванівна</td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table> </td> </tr> <tr> <td>2</td> <td>5</td> <td></td> <td>2</td> <td></td> </tr> <tr> <td>*</td> <td>(Новий)</td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table> </td> </tr> <tr> <td>2</td> <td>Морозов</td> <td>Ігор</td> <td>Олегович</td> <td></td> </tr> <tr> <td>3</td> <td>Петров</td> <td>Михайло</td> <td>Олександров</td> <td></td> </tr> <tr> <td>4</td> <td>Ігнатов</td> <td>Максим</td> <td>Валерійович</td> <td></td> </tr> </tbody> </table>					Ідентифика	Оцінка	Дата здачі	КодПредме	Клацність, t	1	5		1		<table border="1"> <thead> <tr> <th>Назва</th> <th>Години</th> <th>КодВикладч</th> <th>Клацність, t</th> </tr> </thead> <tbody> <tr> <td>Фізика</td> <td>34</td> <td>1</td> <td></td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Прізвище</th> <th>Ім'я</th> <th>По батькові</th> <th>Кафедра</th> <th>Клацніт</th> </tr> </thead> <tbody> <tr> <td>Зікулина</td> <td>Валентина</td> <td>Іванівна</td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table> </td> </tr> <tr> <td>2</td> <td>5</td> <td></td> <td>2</td> <td></td> </tr> <tr> <td>*</td> <td>(Новий)</td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table>					Назва	Години	КодВикладч	Клацність, t	Фізика	34	1		<table border="1"> <thead> <tr> <th>Прізвище</th> <th>Ім'я</th> <th>По батькові</th> <th>Кафедра</th> <th>Клацніт</th> </tr> </thead> <tbody> <tr> <td>Зікулина</td> <td>Валентина</td> <td>Іванівна</td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table>					Прізвище	Ім'я	По батькові	Кафедра	Клацніт	Зікулина	Валентина	Іванівна			*					*		0		0	2	5		2		*	(Новий)	0		0	2	Морозов	Ігор	Олегович		3	Петров	Михайло	Олександров		4	Ігнатов	Максим	Валерійович	
Ідентифика	Оцінка	Дата здачі	КодПредме	Клацність, t																																																																									
1	5		1																																																																										
<table border="1"> <thead> <tr> <th>Назва</th> <th>Години</th> <th>КодВикладч</th> <th>Клацність, t</th> </tr> </thead> <tbody> <tr> <td>Фізика</td> <td>34</td> <td>1</td> <td></td> </tr> <tr> <td colspan="5"> <table border="1"> <thead> <tr> <th>Прізвище</th> <th>Ім'я</th> <th>По батькові</th> <th>Кафедра</th> <th>Клацніт</th> </tr> </thead> <tbody> <tr> <td>Зікулина</td> <td>Валентина</td> <td>Іванівна</td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table> </td> </tr> <tr> <td>2</td> <td>5</td> <td></td> <td>2</td> <td></td> </tr> <tr> <td>*</td> <td>(Новий)</td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table>					Назва	Години	КодВикладч	Клацність, t	Фізика	34	1		<table border="1"> <thead> <tr> <th>Прізвище</th> <th>Ім'я</th> <th>По батькові</th> <th>Кафедра</th> <th>Клацніт</th> </tr> </thead> <tbody> <tr> <td>Зікулина</td> <td>Валентина</td> <td>Іванівна</td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table>					Прізвище	Ім'я	По батькові	Кафедра	Клацніт	Зікулина	Валентина	Іванівна			*					*		0		0	2	5		2		*	(Новий)	0		0																														
Назва	Години	КодВикладч	Клацність, t																																																																										
Фізика	34	1																																																																											
<table border="1"> <thead> <tr> <th>Прізвище</th> <th>Ім'я</th> <th>По батькові</th> <th>Кафедра</th> <th>Клацніт</th> </tr> </thead> <tbody> <tr> <td>Зікулина</td> <td>Валентина</td> <td>Іванівна</td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table>					Прізвище	Ім'я	По батькові	Кафедра	Клацніт	Зікулина	Валентина	Іванівна			*					*		0		0																																																					
Прізвище	Ім'я	По батькові	Кафедра	Клацніт																																																																									
Зікулина	Валентина	Іванівна																																																																											
*																																																																													
*		0		0																																																																									
2	5		2																																																																										
*	(Новий)	0		0																																																																									
2	Морозов	Ігор	Олегович																																																																										
3	Петров	Михайло	Олександров																																																																										
4	Ігнатов	Максим	Валерійович																																																																										

Рис. 3.19. Таблиця **Студенти** з декількома підтаблицями

3.4. Застосування підстановки

У попередньому підрозділі було розглянуто використання підтаблиць, звичайно застосовуваних у тому випадку, коли для запису в таблиці потрібно відображувати більш повну інформацію, що міститься в іншій таблиці. Зокрема для кожного студента можна відображувати отримані ним оцінки.

У зв'язаних таблицях реляційних БД є поля, за якими встановлюється зв'язок [4, 5].

Наприклад, щоб у БД **Освіта** зв'язати таблиці **Оцінки** і **Предмети**, використовується поле **Код Предмета**. Водночас у таблиці **Оцінки** відображуються тільки коди предметів, але не їхні назви. Щоб у цьому випадку відображувати інформацію зі зв'язаної таблиці, тобто **Предмети**, можна створити підтаблицю, як було розглянуто раніше, але простіше і наочніше в цьому випадку використати *підстановку*.

Наприклад, потрібно в таблиці **Оцінки** відображувати не коди предметів, а їхні назви. Для цього потрібно відкрити таблицю **Оцінки** у режимі **Конструктор** і вибрати для поля **Код Предмета** у списку *Тип даних* пункт *Майстер підстановок*. Ця дія запустить майстер підстановок, на першому етапі роботи якого (рис. 3.20) слід встановити, з якого джерела стовпчик підстановки одержуватиме дані.

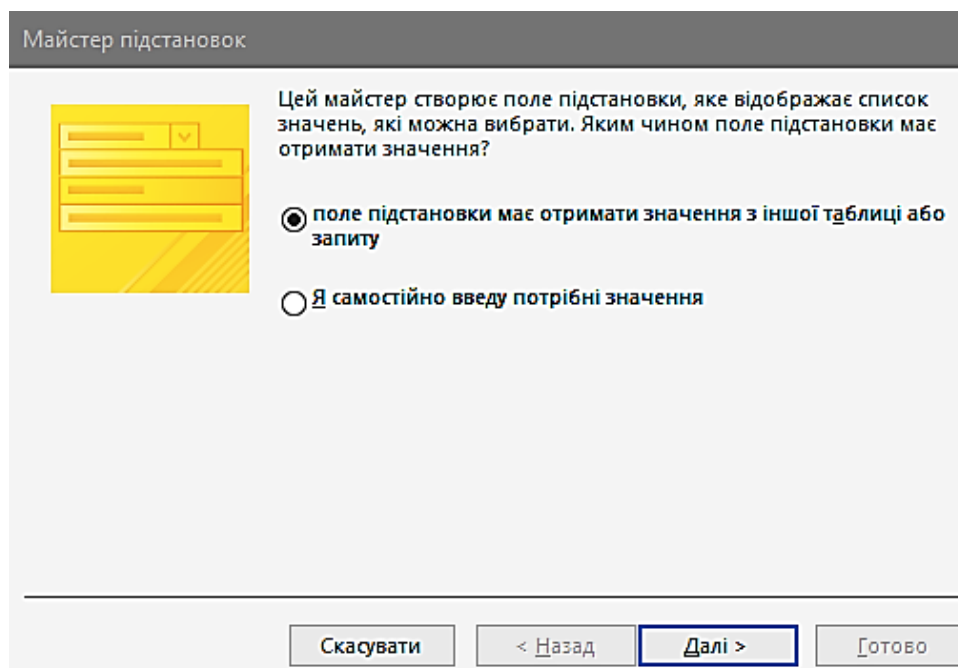


Рис. 3.20. Майстер підстановок. Крок 1

Оскільки назва дисципліни міститься і в таблиці **Предмети**, то потрібно вибрати перший пункт (з *таблиці або запиту*) і натиснути кнопку *Далі*>.

На другому етапі потрібно вибрати таблицю (або запит), що буде використовуватися як джерело даних, тобто таблицю **Предмети** (рис. 3.21) і натиснути кнопку *Далі*>.

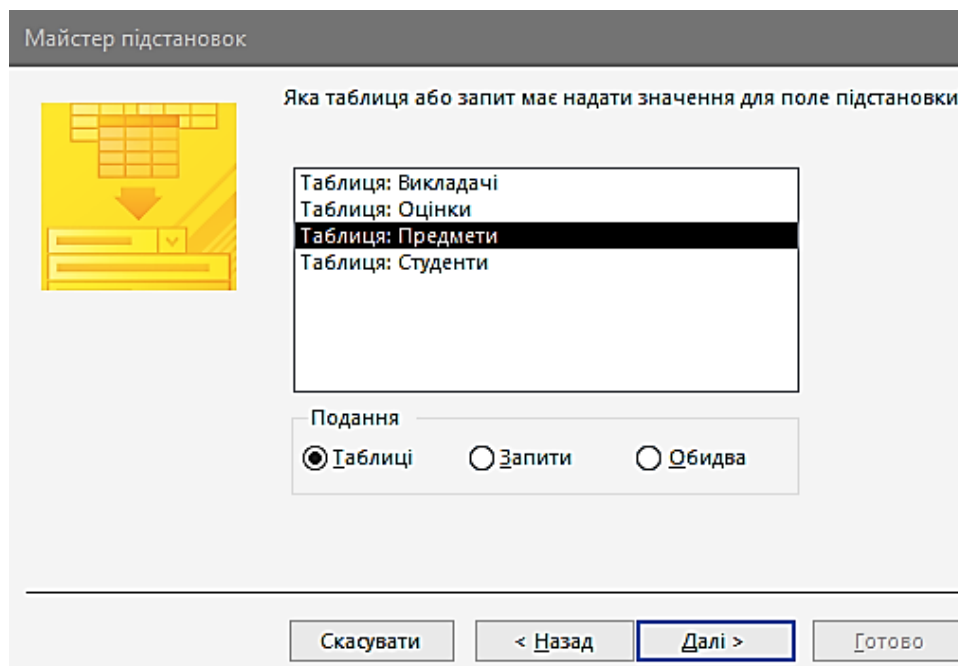


Рис. 3.21. Майстер підстановок. Крок 2

Наступний етап припускає вибір поля (або полів), значення якого (яких) будуть використовуватися у стовпчику підстановки. У цьому випадку потрібно в список *Обрані поля* перемістити поле **Назва** (рис. 3.22) [1] і натиснути *Далі*>.

На четвертому етапі роботи майстра можна встановити ширину полів, що відображатимуться у стовпчику підстановки, а також встановити галочку *Сховати ключовий стовпчик*, для того щоб не відображувалися коди предметів (рис. 3.23), і потім натиснути *Далі*>.

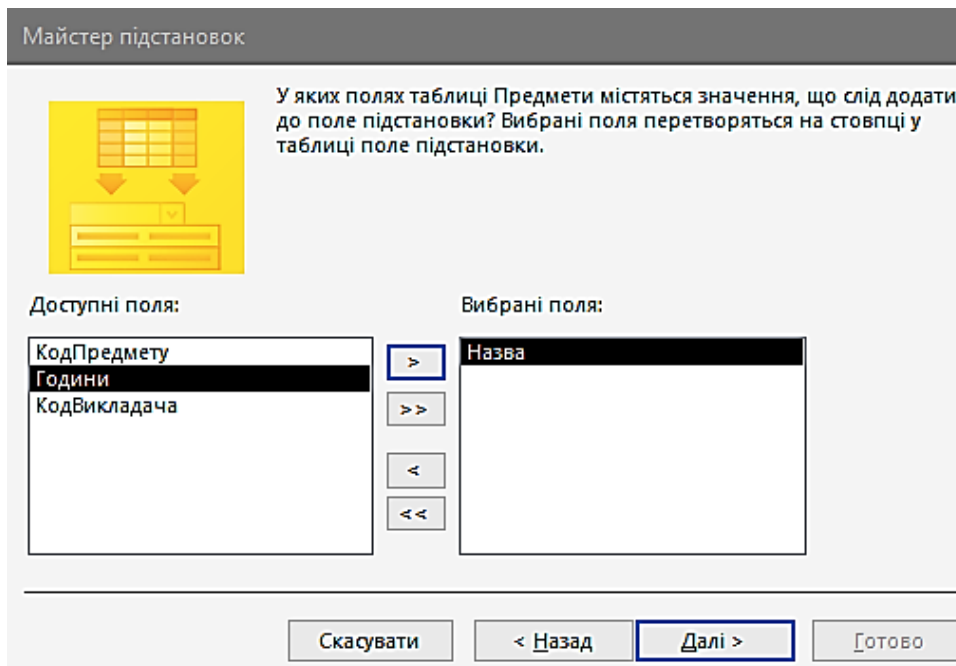


Рис. 3.22. Майстер підстановок. Крок 3

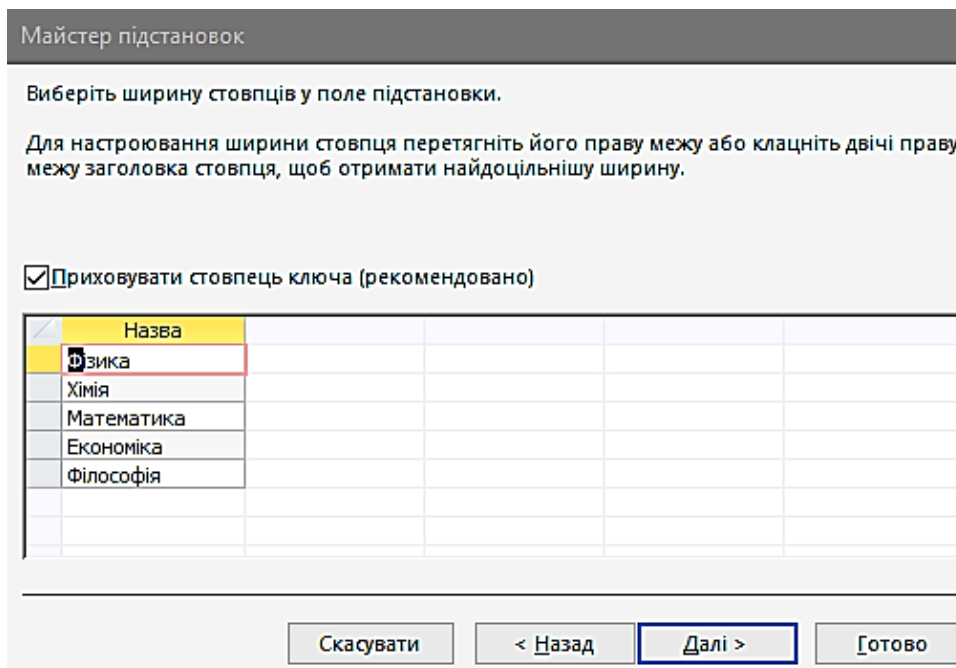


Рис. 3.23. Майстер підстановок. Крок 4

На останньому етапі створення підстановки можна визначити підпис, що відображатиметься для стовпчика підстановки. Наприклад, можна

задати підпис *Предмет* (рис. 3.24), і потім, для завершення роботи майстра, натиснути кнопку *Готово*. Після цього зберегти таблицю, як це буде запропоновано зробити у вікні повідомлення.

У результаті при перегляді вмісту таблиці **Оцінки** замість стовпчика **Код Предмета** буде відображатися стовпчик **Предмет**, у якому міститься не код предмета, а його назва.

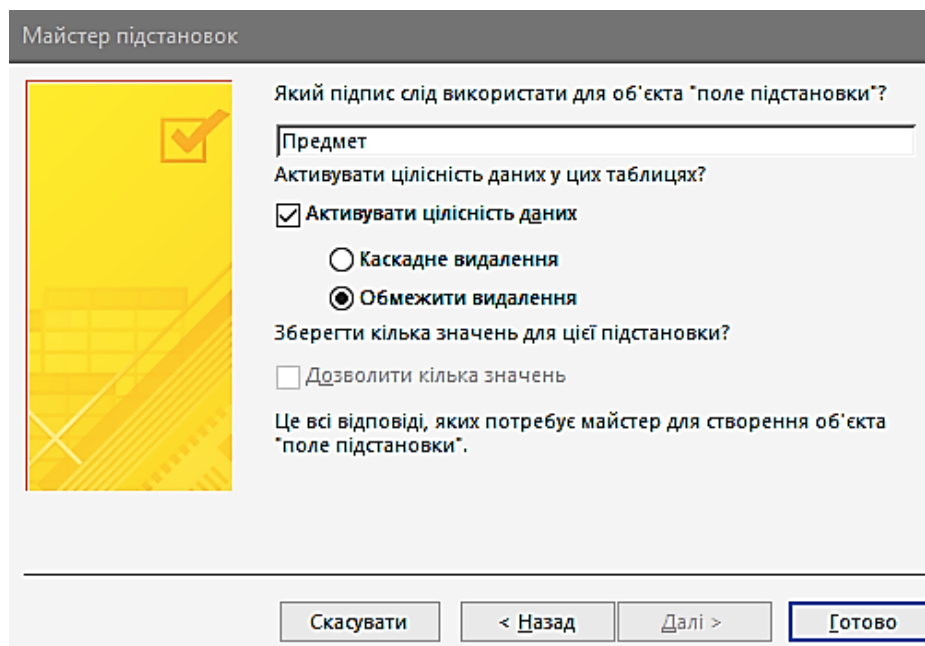


Рис. 3.24. Майстер підстановок. Крок 5

Щоб змінити вміст стовпчика підстановки для якого-небудь запису, наприклад першого, потрібно встановити курсор у стовпчик **Предмет**, розгорнути за допомогою стрілки список наявних дисциплін (рис. 3.25) і вибрати необхідну назву дисципліни. Однак це не означає, що в полі підстановки будуть записуватися текстові значення назв дисциплін.

Як сполучне поле так само використовується поле **Код Предмета**, просто в цьому випадку змінюється спосіб відображення інформації. У цьому легко можна переконатися, якщо змінити назву дисципліни в таблиці **Предмети**. Ця дія призведе до того, що внесена зміна автоматично відобразиться в полі підстановки таблиці **Оцінки**.

Оцінки						
Ідентифіка	Оцінка	Дата здачі	КодСтудент	Предмет	Клацніть, и	
1	5			1 Фізика		
2	5			1 Фізика		
3	4			2 Хімія		
4	4			2 Математика		
5	3			3 Економіка		
6	4			3 Філософія		
7	5			3 Математика		
8	5			4 Фізика		
9	4			4 Хімія		
*	(Новий)	0		0		

Рис. 3.25. Таблиця **Оцінки** зі стовпчиком підстановки *Предмет*

Контрольні запитання до розділу 3

1. Як здійснюється операція Сортування?
2. Що потрібно виконати для реалізації команди *Пошук і заміна даних*?
3. Що забезпечує виконання команди *Фільтрація даних*?
4. Що забезпечує виконання оператора AND?
5. Що забезпечує виконання оператора OR?
6. У якому режимі здійснюють перегляд інформації таблиці?
7. За яких умов в Access можливе відображення в одній таблиці даних з іншої таблиці?

Розділ 4. ВИКОРИСТАННЯ ЗАПИТІВ

4.1. Запит на вибірку

Як правило, нема потреби у відображенні всієї інформації, що зберігається в тій або іншій таблиці БД. У третьому розділі було розглянуто можливості щодо забезпечення потрібного режиму обробки даних, зокрема фільтрація, сортування, приховування стовпчиків, застосування підстановок, використання підтаблиць.

Однак є більш потужний засіб обробки даних в одній або декількох таблицях, що дає змогу відображувати потрібну інформацію, що відповідає зазначеним вимогам [1].

Наприклад, у БД **Освіта** потрібно знайти всіх студентів, які мають відмінні оцінки, або відобразити прізвища всіх студентів, які склали фізику. Для цього в Access використовуються *запити*.

Існує кілька типів запитів:

- на вибірку;
- відновлення;
- додавання
- видалення;
- перехресний і створення таблиці.

Найпоширенішим є запит на вибірку. При його виконанні створюється безліч записів, у яких містяться зазначені дані, наприклад певні поля з однієї або декількох таблиць. Інформацію, відображану в запиті, можна змінювати так само, як і при роботі зі звичайною таблицею. Однак слід зазначити, що наведений набір записів створюється тимчасово, тільки під час виконання запиту, і не зберігається в БД, на відміну від вмісту таблиць [1].

Щоб створити новий запит, потрібно натиснути кнопку *Запити* у вікні БД **Освіта** і після цього вибрати режим створення запиту з використанням конструктора або майстра. Також для створення запиту можна скористатися кнопкою *Створити*, після чого вибрати режим створення нового запиту. Існує кілька режимів створення запиту. Нижче буде докладно розглянуто кожен з них [1].

Створення запиту в режимі конструктора. Після вибору режиму конструктора буде відображене діалогове вікно *Додавання таблиці* для вибору таблиці (або таблиць), що буде служити джерелом даних при виконанні цього запиту. Як приклад буде розглянуто створення запиту на основі даних з таблиці **Оцінки**. Після додавання таблиці **Оцінки** потрібно закрити вікно *Додавання таблиці*, після чого на екрані буде відображено вікно конструктора нового запиту (рис. 4.1), що зовні нагадує розглянуте раніше вікно розширеного фільтра [1].

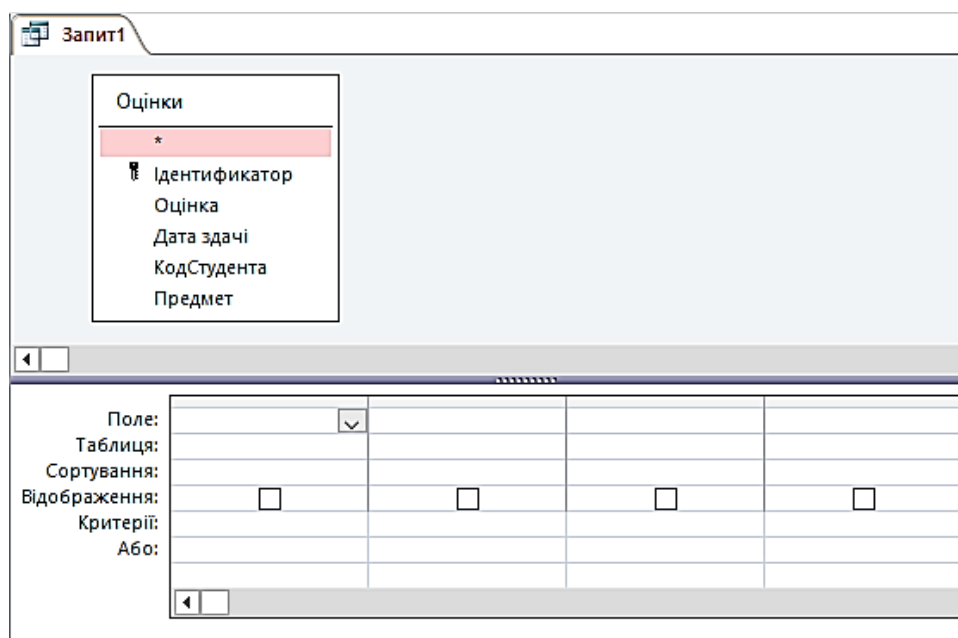


Рис. 4.1. Конструктор запитів

У верхній частині вікна *Конструктора* розташовуються таблиці (або таблиця), використовувані як джерело даних. У нижній частині вікна вказуються параметри даних, які відбирають до вибірки.

Поле – вказується назва поля, вміст якого буде додано в результати запиту. Щоб додати поле в запит, можна перетягнути його в рядок *Поле* з макета таблиці, розташованого у верхній частині вікна, або вибрати необхідне поле у списку, що випадає, який можна розкрити у рядку *Поле*. Щоб додати в запит усі поля з таблиці, потрібно двічі натиснути клавішею миші в таблиці на рядку з символом «*».

Ім'я таблиці – за допомогою списку вибирається назва таблиці, у якій є поле, що використовується як джерело даних. Якщо вихідна таблиця одна або назва поля унікальна для всіх таблиць, то у використанні цього параметра нема потреби, тому що його значення буде встановлюватися автоматично.

Сортування – вказується напрямок сортування для цього поля – *за зростанням* або *убуванням*.

Виведення на екран – задається режим, при якому на екрані буде відображатися вміст обраного поля. На практиці можливі випадки, коли певне поле бере участь у запиті, наприклад для виконання відносно нього якої-небудь умови, однак у результатах запиту це поле може і не відображатися.

Умова відбору, або – вказуються одна або кілька умов, відповідно до яких записи будуть додаватися до вибірки.


Наприклад, потрібно відобразити тільки оцінки «4» і «5», отримані студентами (без зазначення прізвищ), і дату складання. При створенні такого запиту джерелом даних буде служити таблиця **Оцінки** (рис. 4.1).

У запиті використовуватимуться поля **Оцінка** й **ДатаСкладання**, назви яких потрібно вибрати в рядку *Поле*. Назву таблиці в поле **Ім'я таблиці** буде встановлено автоматично.

Для поля **ДатаСкладання** можна встановити напрямок сортування за зростанням. Обидва зазначених поля потрібно виводити на екран, тому для кожного з них встановити прапорець *Виведення на екран*. У рядку *Умова відбору* для поля **Оцінка** потрібно вказати таку умову: 4 OR 5 (рис. 4.2) або вказати перше значення (4) у поле **Умова відбору**, а друге (5) – у поле **Або**.

Поле:	Оцінка	Дата здачі	
Таблиця:	Оцінки	Оцінки	
Сортування:		За зростанням	▼
Відображення:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Критерії:			
Або:			

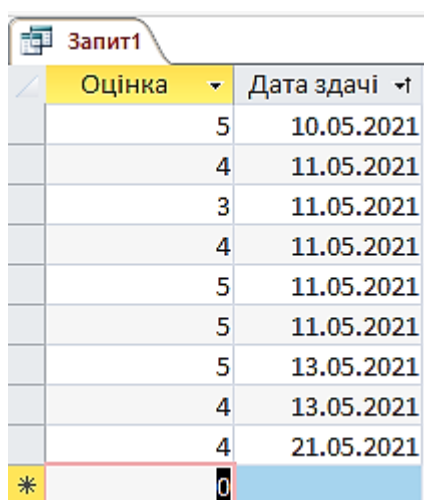
Рис. 4.2. Запит на вибірку з таблиці Оцінки

Створений у такий спосіб запит потрібно виконати. Для цього натиснути кнопку *Запуск*  або виконати команду *Запит | Запуск*. У результаті буде отримано набір записів з необхідним набором полів, значення яких відповідають зазначеній умові (рис. 4.3). Для збереження запиту виконати команду *Файл | Зберегти*, після чого вказати в діалоговому вікні ім'я нового запиту, наприклад **ОцінкаДата**. У результаті в списку запитів БД **Освіта** з'явиться запит **ОцінкаДата**, який можна відкрити для редагування кнопкою *Відкрити* або змінити запит за допомогою кнопки *Конструктор* вікна БД.

Розглянутий запит створювався на основі однієї таблиці. Існують також запити, які використовують дані з декількох таблиць. Такі запити називаються *багатотабличними* [1].


Наприклад, потрібно змінити запит **ОцінкаДата** таким чином, щоб відображувалися не тільки оцінки, але й прізвища студентів, які їх

одержали, а також назви відповідних предметів. Водночас у запиті мають відображатися тільки оцінки з дисципліни «Фізика».



Оцінка	Дата здачі
5	10.05.2021
4	11.05.2021
3	11.05.2021
4	11.05.2021
5	11.05.2021
5	11.05.2021
5	13.05.2021
4	13.05.2021
4	21.05.2021
*	

Рис. 4.3. Результат виконання запиту

Для цього потрібно відкрити запит **ОцінкаДата** в режимі **Конструктора** і додати до макета запиту ще дві таблиці, які містять необхідну інформацію: **Студенти** і **Предмети**. Щоб додати таблицю в макет запиту, потрібно виконати команду *Запит | Додати таблицю* або натиснути кнопку *Відобразити таблиці* , і за допомогою діалогового вікна *Додавання таблиці* по черзі додати таблиці **Студенти** і **Предмети** в макет. Водночас на схемі будуть автоматично відображені зв'язки між зазначеними таблицями. Слід зазначити, що в макеті запиту можна визначити зв'язки між таблицями, якщо вони не були раніше встановлені. Для цього потрібно виділити назву поля зв'язку в одній таблиці і перетягнути його на макет іншої таблиці, аналогічно роботі зі схемою даних.

Після додавання всіх таблиць у нижній частині вікна *Запиту* можна перед полем **Оцінка** вставити два стовпчики – для прізвищ студентів і назв предметів. При перетягуванні поля з таблиці, розташованої в макеті, у рядок *Поле*, автоматично з'являється новий стовпчик, у якому можна вказувати

параметри цього поля. Для поля **Прізвище** з таблиці **Студенти** потрібно вказати напрямок сортування за зростанням, а для поля **Назва** з таблиці **Предмети** – «Фізика». Водночас сортування для поля **ДатаСкладання** можна скасувати, вибравши в полі **Сортування** значення *(відсутній)* (рис. 4.4) [1].

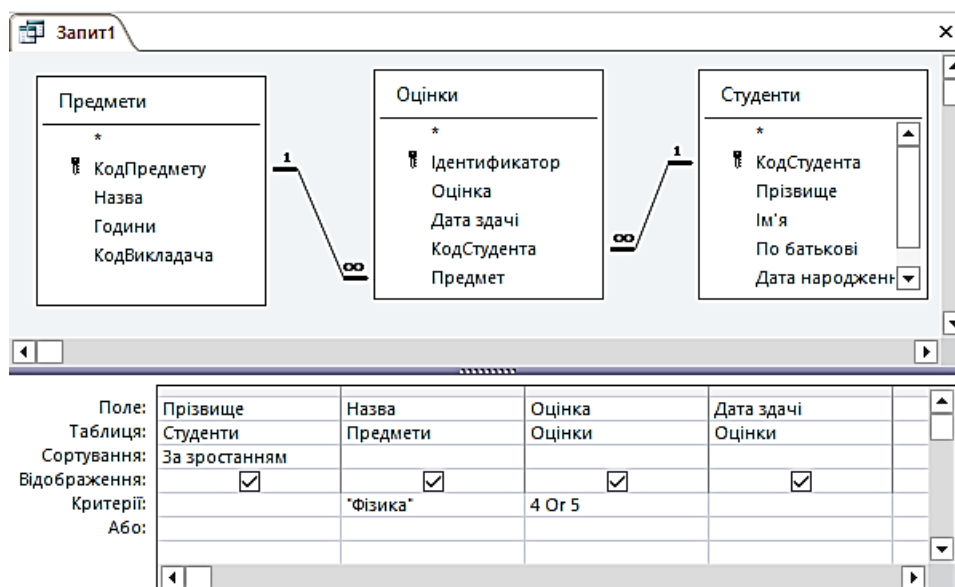



Рис. 4.4. Запит на вибірку з таблиць **Студенти**, **Предмети** і **Оцінки**

Створений запит потрібно виконати, скориставшись кнопкою *Запуск* . У результаті будуть отримані записи з зазначеним набором полів з обраних таблиць. Водночас дані відображуються тільки для дисципліни «Фізика» (рис. 4.5) [1].

Прізвище	Назва	Оцінка	Дата здачі
Жукова	Фізика	5	11.05.2021
Ігнатов	Фізика	5	10.05.2021
Морозов	Фізика	4	11.05.2021
*			

Рис. 4.5. Результат виконання запиту

Отриманому багатотабличному запиту можна дати назву *ОцінкиЗФізики*, скориставшись для цього командою *Файл |Зберегти як*.

Створення запиту за допомогою майстра

Щоб запуснути майстер запитів, треба у вікні БД двічі натиснути кнопкою миші на пункті *Створення запиту за допомогою майстра* в розділі *Запити*.

Водночас першим кроком у створенні запиту є вибір полів, які будуть у ньому відображатися, з наявних у БД таблиць і запитів. Наприклад, потрібно створити запит, що містить прізвища викладачів і назви дисциплін, які вони викладають. Для цього потрібно спочатку у списку, що розкривається, *Таблиці і запити* вибрати таблицю **Викладачі**, і додати поле **Прізвище** в розділ *Обрані поля*. Потім вибрати таблицю **Предмети** і додати в запит поле **Назва** (рис. 4.6).

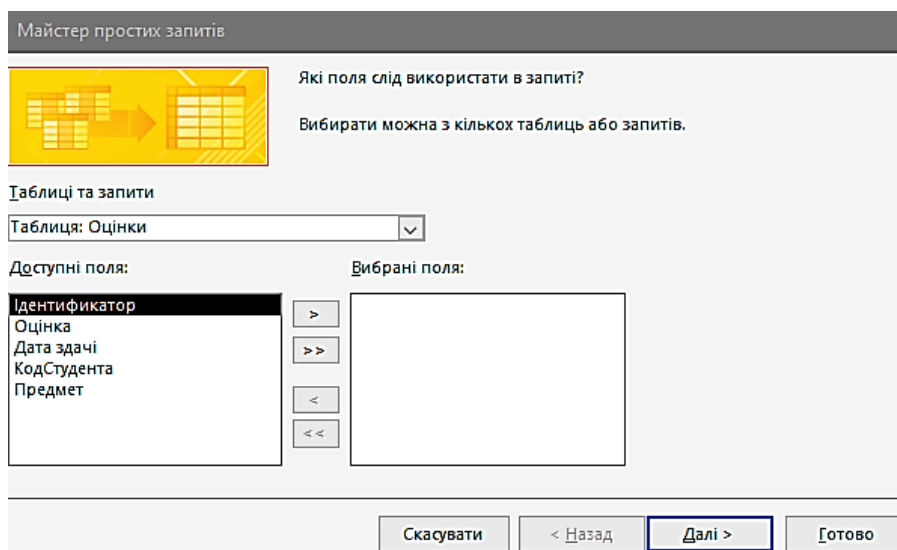
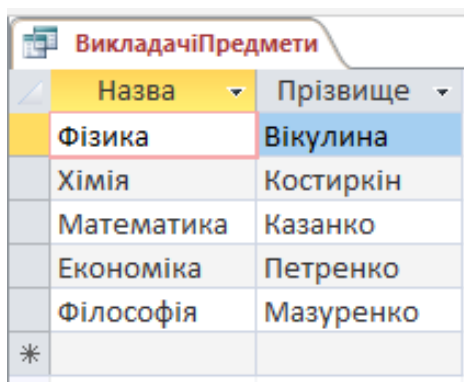


Рис. 4.6. Майстер запитів. Вибір полів

Після натискання кнопки *Далі >* буде відображено завершальне вікно майстра, у якому потрібно вказати назву створеного запиту і визначити подальший режим роботи з ним – перегляд даних або зміну макета. Запиту можна дати назву **ВикладачіПредмети**, і після вибору режиму перегляду

даних і натискання кнопки *Готово* будуть відображені записи створеного запиту, що містять прізвища викладачів і назви дисциплін (рис. 4.7).



Назва	Прізвище
Фізика	Вікулина
Хімія	Костиркін
Математика	Казанко
Економіка	Петренко
Філософія	Мазуренко
*	

Рис. 4.7. Запит **ВикладачіПредмети**

Поля, що обчислюються. При створенні запитів часто виникає необхідність не тільки використати наявні поля таблиць, але й створювати на їхній основі інші поля, які називаються полями, що обчислюються. Наприклад, якщо в таблиці будь-якого магазину є поле ціни на товар і поле кількості цього товару, то, зважаючи на це, можна створити поле, що обчислюється, у якому підраховуватиметься загальна вартість для кожного товару перемноженням значень ціни і кількості. Інакше кажучи, у полі, що обчислюється, можуть використовуватися арифметичні оператори.

З іншого боку, при створенні поля, що обчислюється, на основі вмісту текстових полів, як правило, використовується операція об'єднання текстових значень, що називається *конкатенацією*. У табл. 4.1 наведені оператори, використовувані в Access при побудові виразів.

Очевидно, що розглянуті оператори надають досить вузький спектр можливостей для створення виразів. Щоб створити складний вираз, використовують потужний засіб, що називається *Будівник виразів*.

Основні оператори, використовувані при створенні виразів

Оператор	Опис
+	Додавання чисел
-	Вирахування чисел
*	Множення чисел
/	Ділення чисел
\	Ділення першого числа на друге з округленням результату
^	Піднесення до ступеня
MOD	Обчислення залишку від цілочислового ділення
&	Конкатенація рядків

Ця програма дає змогу побудувати вираз будь-якої складності з використанням найрізноманітніших функцій (рис. 4.8).

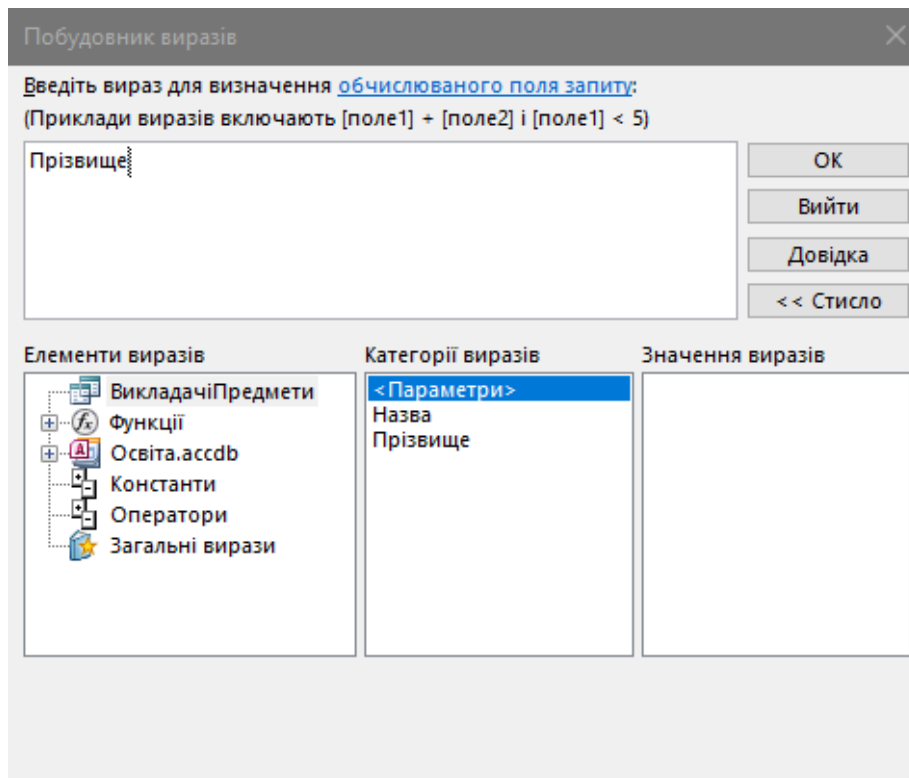


Рис. 4.8. Будівник виразів

За допомогою *Будівника виразів* можна виконувати різні обчислення з використанням даних, що перебувають у БД – полів таблиць, запитів, форм і звітів. Усі перераховані об'єкти БД перебувають відповідно в папках *Таблиці, Запити, Форми та Звіти* у лівій частині *Будівника*.

Щоб вибрати необхідний об'єкт, наприклад таблицю, потрібно виконати подвійне натискання лівої клавіші миші на папці *Таблиці*, у результаті чого розкриється список усіх таблиць БД. Після вибору потрібної таблиці (або другого об'єкта) у середній частині *Будівника* відображується список її полів, значення яких можна включити до створюваного виразу. Для цього потрібно вибрати необхідне поле й натиснути у вікні *Будівника* кнопку *Вставити*.

При побудові виразів, крім назв полів, можуть використовуватися різні оператори (табл. 4.1), додавання яких виконується при натисканні відповідних кнопок у вікні *Будівника*.

Також можливе використання у виразі різних функцій Access [1].

Для цього в лівій частині вікна потрібно відкрити папку *Функції* і вибрати пункт *Вбудовані функції*. Потім у середній частині *Будівника* визначити категорію, у якій перебуває необхідна функція, і вибрати після цього у правій частині вікна назву потрібної функції (рис. 4.9, 4.10). Створюваний у такий спосіб вираз відображується у верхній частині вікна *Будівника виразів*.

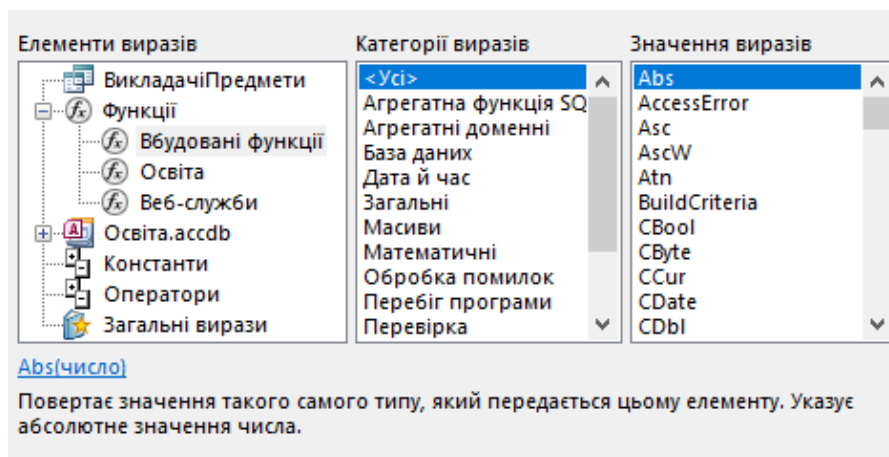


Рис. 4.9. *Будівник виразів*. Вбудовані функції

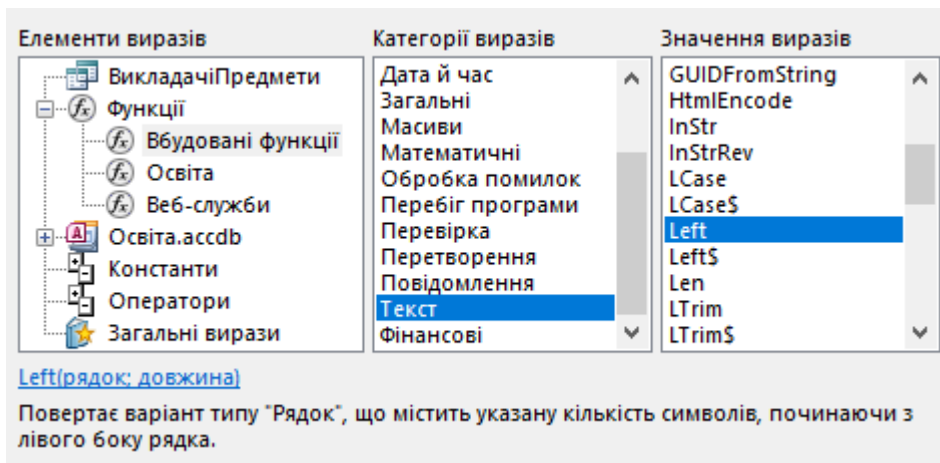



Рис. 4.10. Будівник виразів. Текстова функція Left

Наприклад, потрібно в розглянутому вище запиті **ВикладачіПредмети** замість поля **Прізвище** додати поле, що обчислюється, у якому містилися б прізвище та ініціали викладача.

Для цього відкрити запит у режимі **Конструктор**. Потім установити курсор у рядок *Поле* першого стовпчика, тобто в поле **Прізвище**, і скористатися кнопкою *Побудувати* , у результаті чого буде відображене вікно *Будівника виразів* (рис. 4.10).

Слід зазначити, що за допомогою кнопки *Побудувати* можна як створити новий вираз, так і змінити вже існуючий, якщо він перебуває в поточному полі [1].

Назви полів у виразі вказують у квадратних дужках, однак якщо поле буде зазначено без дужок, і в імені поля відсутні пропуски, знаки арифметичних операцій та деякі інші спеціальні символи, то Access ці дужки вставить автоматично. За наявності в іменах полів вищезазначених символів квадратні дужки потрібно вставити вручну, бо інакше СКБД може розставити ці дужки некоректно.

Для створення текстового рядка з прізвищем та ініціалами потрібно кілька разів послідовно виконати операцію конкатенації такої структури: *Прізвище & Символ Пропуску&Ініціал імені&Символ крапки&Ініціал по*

батькові&Символ крапки. Усі текстові символи вказуються в подвійних лапках, тому структура виразу набуде такий вигляд: *Прізвище & «» &Ініціал імені& «.» &Ініціал по батькові& «.»*.

Отже, початок виразу буде таким [1]:

Прізвище &«»&.

Для роботи з текстовими значеннями використовуються функції категорії *Текстові*. Щоб одержати ініціал імені, потрібно скористатися однією з функцій цієї категорії, а саме функцією *Left*, що повертає перші *n* символів зазначеного рядка. Щоб додати функцію у вираз, потрібно двічі натиснути на ліву кнопку миші на папці *Функції* і вибрати після цього розділ *Вбудовані функції* (рис. 4.9).

Потім у другому стовпчику вибрати необхідну категорію, у цьому випадку *Текстові*, і після цього в третьому стовпчику вибрати потрібну функцію, тобто *Left* (рис. 4.10).

Після того як функція обрана, встановити курсор у те місце виразу, куди вона має бути вставлена, і натиснути кнопку *Вставити* у вікні *Будівника виразів*.

У результаті цих дій у вираз буде додано шаблон функції *Left* [1]:

Left («stringexpr»; «n»),

у якому вказати значення потрібних параметрів. Зокрема замість *stringexpr* потрібно вказати поле **Ім'я**, а замість *n* – кількість символів, які потрібно одержати з рядка, тобто 1. Отже, текст виразу має набути такого вигляду [1]:

Прізвище & «» &Left (Ім'я; 1).

Слід зазначити, що для поля **Ім'я** не вказувалася назва таблиці **Викладачі**, тому що воно було обрано раніше в рядку *Ім'я таблиці* вікна *Конструктора запитів*. З іншого боку, щоб додати до виразу ім'я поля конкретної таблиці, можна розкрити папку *Таблиці* і вибрати в таблиці **Викладачі** поле **Ім'я** (рис. 4.11), після чого встановити курсор у відповідне

місце виразу і натиснути кнопку *Вставити*. Отже, назву поля з зазначенням імені таблиці слід відображувати у такому форматі:

[Ім'я таблиці]![Назва поля].

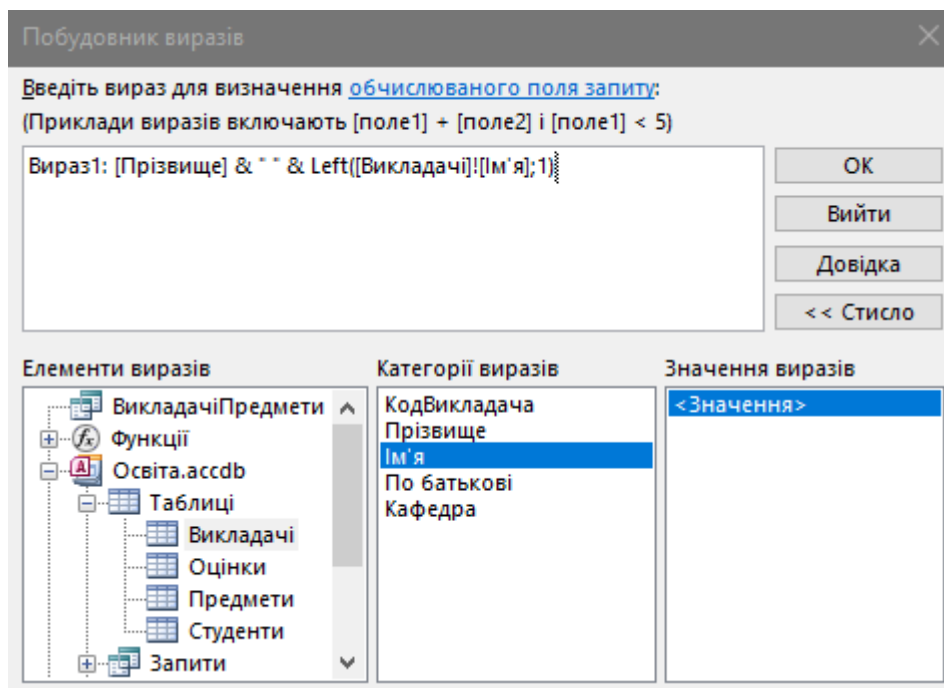



Рис. 4.11. Будівник виразів. Поле Ім'я таблиці Викладачі

Після визначення першого ініціала потрібно поставити крапку, тобто вказати у виразі рядок «.». Потім аналогічно додати ініціал **По батькові** і знову поставити крапку - буде отримано такий вираз [1]:

Прізвище & « » &Left(Ім'я;1) & «.» &Left (ПоБатькові; 1) & «.».

Щоб повернутися до вікна *Конструктора запитів*, потрібно натиснути кнопку *ОК*. У результаті створений вираз буде перебувати в рядку *Поле* першого стовпчика запиту. Для виконання запиту потрібно скористатися кнопкою *Запуск* , після чого в запиті буде створено поле, що обчислюється, *Вираз1*, яке міститиме прізвище та ініціали викладача.

Назви полів запиту формуються автоматично. Для полів таблиць назви збігаються з назвами відповідних полів запиту (наприклад **Поле Назва**). З іншого боку, назви полів, що обчислюються, формуються

автоматично: *Вираз1*, *Вираз2* і т. д. Очевидно, що такі назви не є інформативними.


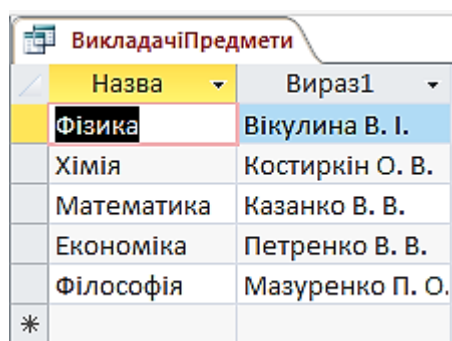
Щоб установити підпис для поля в таблиці або запиті, потрібно в режимі **Конструктор** встановити курсор у необхідне поле (зокрема в поле **Вираз1**) і натиснути кнопку *Властивості*  [1]. Після цього в діалоговому вікні *Властивості поля* можна на вкладці *Загальні* вказати в рядку *Підпис* ту назву поля, що використовуватиметься замість вихідного в режимі **Редагування запиту** (або таблиці), а також у формах і звітах. Зокрема для поля, що обчислюється, *Вираз1* можна вказати підпис ПІБ (рис. 4.12) [1].



Рис. 4.12. Діалогове вікно *Властивості поля*

Отже, був створений і виконаний запит на вибірку з використанням поля, що обчислюється (рис. 4.13).





Назва	Вираз1
Фізика	Вікулина В. І.
Хімія	Костиркін О. В.
Математика	Казанко В. В.
Економіка	Петренко В. В.
Філософія	Мазуренко П. О.
*	

Рис. 4.13. Запит на вибірку з полем, що обчислюється

Групові операції. При обробці даних у таблицях найчастіше буває потрібно враховувати в запитах не окремі записи, а підсумки, обчислювані або формовані певним чином для різних груп записів у таблиці.

Наприклад, потрібно виконати запит, у якому для кожного студента була б підрахована кількість отриманих ним оцінок, а також обчислений середній бал. І потім на підставі середнього бала визначена винагорода, наприклад з розрахунку 10 грн за кожен бал. Інакше кажучи, для відмінника (середній бал = 5) розмір винагороди має становити 50 грн.

Щоб виконати такі обчислення, потрібно створити новий запит у режимі **Конструктор**, додати до нього за допомогою кнопки *Відобразити таблицю*  таблиці **Студенти** й **Оцінки**, і потім скористатися кнопкою *Групові операції* . Після цього серед параметрів запиту з'явиться рядок *Групові операції*.

Щоб обчислити для кожного студента його середній бал, потрібно спочатку додати в запит поле **Прізвище** таблиці **Студенти**. Водночас у рядку *Групові операції* за замовчуванням встановлюється значення *Групування*.

Отже, для кожного прізвища студента в другому стовпчику потрібно розташувати загальну кількість оцінок, отриманих кожним студентом, у третьому – середній бал, четвертому – розмір нарахованої стипендії. Для цього до другого і третього стовпчиків запиту потрібно додати поле **Оцінка** таблиці **Оцінки**, тому що на підставі саме цього поля буде визначатися загальна кількість оцінок, а також обчислюватись середній бал кожного студента.

При виконанні групових операцій можна використовувати так звані підсумкові функції (табл. 4.2), які потрібно вибирати зі списку в цьому полі **Групові операції** [1].


Для підрахунку кількості отриманих оцінок потрібно вибрати функцію **Count**, а для визначення середнього бала – функцію **Avg**. Потім створений запит потрібно зберегти, вказавши для нього ім'я **СереднійБал**.

Основні підсумкові функції

Функція	Опис
Sum	Підсумовування значень у групі
Avg	Середнє значення для групи
Min	Мінімальне значення для групи
Max	Максимальне значення для групи
Count	Підрахунок кількості значень у групі
First	Значення поля в першому записі групи
Last	Значення поля в останньому записі групи

Щоб обчислити розмір винагороди, потрібно створити вираз за допомогою *Будівника виразів*.

Для цього спочатку потрібно вибрати в рядку *Групові операції* четвертого стовпчика запиту за допомогою списку, що розгортається, пункт *Вираз*.

Потім, перебуваючи в цьому полі, натиснути кнопку *Побудувати*  і вказати в *Будівнику виразів* формулу для обчислення розміру винагороди. У цій формулі буде використовуватися поле групової операції з функцією Avg, отримане раніше.

Слід зазначити, що подібно до полів у запитах для полів, що обчислюються, Access також автоматично присвоює імена. Зокрема поле, що містить кількість оцінок, було названо *Count-Оцінка*, а поле з середнім балом – *Avg-Оцінка*.

Отже, у вікні *Будівника виразів* потрібно у формулі для стипендії використати поле *Avg-Оцінка* збереженого запиту *СереднійБал*. Тобто вставити до формули поле *Avg-Оцінка* з папки *СереднійБал* і помножити це значення на 10 для визначення розміру винагороди (рис. 4.14). Однак слід мати на увазі, що якби розглянутий запит не був збережений, то в *Будівнику виразів* не з'явилися б поля *Count-Оцінка* і *Avg-Оцінка*.

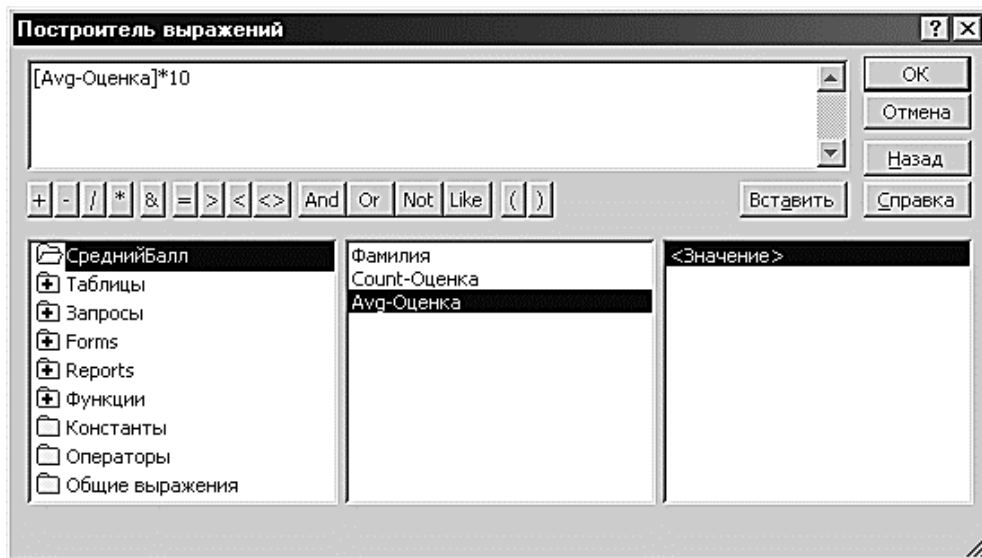


Рис. 4.14. Будівник виразів. Визначення розміру винагороди

Після установлення всіх описаних параметрів макет запиту буде мати вигляд, зображений на рис. 4.15.

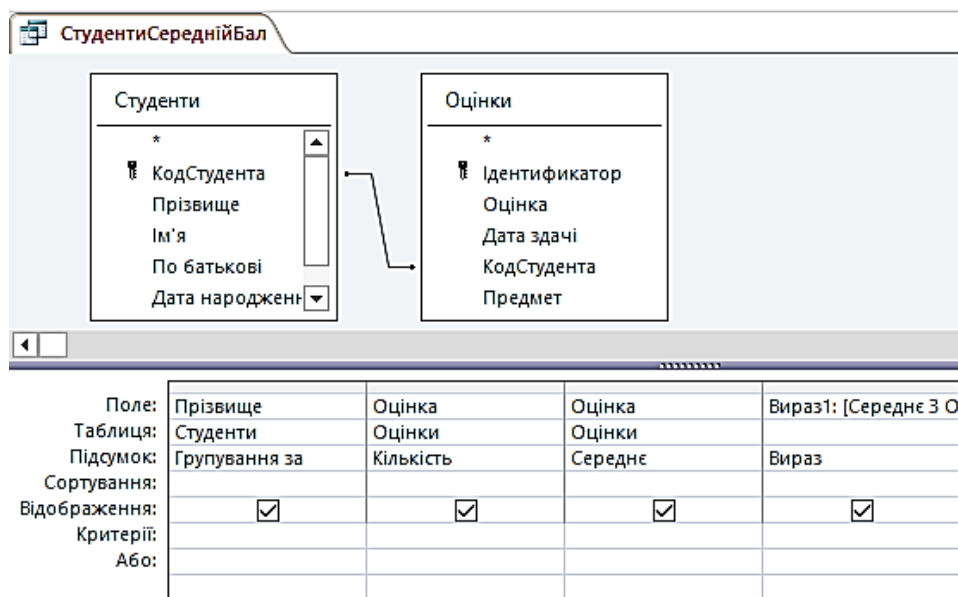



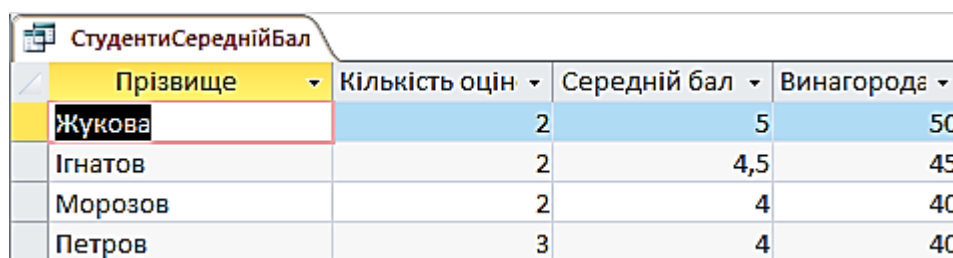
Рис. 4.15. Макет запиту з використанням групових операцій

При виконанні створеного таким способом запиту буде визначена кількість оцінок, середній бал і розмір винагороди для кожного студента.

Однак назви полів із груповими операціями (кількість оцінок і середній бал) і виразу (винагорода) були створені автоматично і не є достатньою мірою інформативними. Тому потрібно змінити властивості полів, встановивши їм підписи.

Для цього відкрити запит у режимі **Конструктор** і у вікні *Властивостей* (кнопка ) для кожного з цих полів указати в рядку підпис назви *КількістьОцінок*, *СереднійБал* і *Винагорода* відповідно.

Результатом виконання запиту **СереднійБал** буде набір записів, поданий на рис. 4.16.



Прізвище	Кількість оцін	Середній бал	Винагорода
Жукова	2	5	50
Ігнатов	2	4,5	45
Морозов	2	4	40
Петров	3	4	40

Рис. 4.16. Результат виконання запиту СереднійБал

Перехресні запити. В Access є можливість створення запитів, у яких використовуються групові операції, у вигляді електронних таблиць [1].

Тобто як назви рядків і стовпчиків такої «таблиці» використовуються ті або інші поля таблиць, а в її «комірках» будуть відображуватися обчислювані значення, що відповідають значенням, зазначеним у назві рядка й стовпчика.

Як приклад такої електронної таблиці можна розглянути набір оцінок, одержаних кожним студентом з усіх дисциплін.

Інакше кажучи, заголовками рядків такої таблиці будуть прізвища студентів, заголовками стовпчиків – назви дисциплін, а на перетині рядків і стовпчиків буде відображуватися оцінка того або другого студента з цього предмета, аналогічно розташуванню даних у робочих аркушах Excel.

Водночас слід зазначити, що обчислювані значення не можуть бути відредаговані, тому використання перехресного запиту для зміни даних не має рації. Однак перехресні запити можуть виявитися досить корисними при створенні різних звітів [1].

Для створення перехресного запиту, що відображує таблицю оцінок студентів, потрібно спочатку створити новий запит у режимі **Конструктор**. Потім до макета запиту додати таблиці, у яких містяться вихідні дані запиту: **Студенти, Оцінки і Предмети**.

Після цього виконати команду *Запит | Перехресний*, у результаті чого в макеті запиту серед параметрів з'явиться рядок *Перехресна таблиця*, у якій для полів запиту можна за допомогою списку, що розгортається, вибрати одне з нижченаведених значень.

Заголовки рядків – вибирається для поля запиту, значення якого потрібно використати як заголовки рядків. Як заголовки рядків визначають хоча б одне поле, водночас у рядку *Групові операції* вибирають значення *Групування*, одну з підсумкових функцій або *Вираз*.

Заголовки стовпчиків – вибирається для поля запиту, значення якого потрібно використати як заголовки стовпчиків.

З цією метою використовується тільки одне поле, вимоги до якого аналогічні вимогам до поля, у якому встановлене значення *Заголовки рядків*.

Значення – вибирається для поля запиту, у якому обчислюється підсумкове значення, відображуване в «комірках» перехресного запиту. Таке поле має бути єдиним, водночас у рядку *Групові операції* для нього потрібно вибрати одну з підсумкових функцій або задати вираз, у якому використовуються підсумкові функції.

Не відображується – вибирається для того поля, значення якого не мають відображуватися в перехресному запиті (використовується для завдання деяких умов відбору даних).

У першому стовпчику запиту потрібно вибрати поле **Прізвище** таблиці **Студенти**, одночасно задати для нього значення *Групування* в поле **Групова операція**, а також *Заголовки рядків* у поле **Перехресна таблиця**. Також можна вибрати напрямок сортування *за зростанням*.

Для другого стовпчика запиту потрібно вибрати поле **Назва** таблиці **Предмети** і встановити для нього значення *Групування* в поле **Групова операція** і *Заголовки стовпчиків* у поле **Перехресна таблиця**.

У третьому стовпчику запиту потрібно вибрати поле **Оцінка** таблиці **Оцінки** і задати для нього функцію *Sum* у поле **Групова операція**, а також *Значення* в поле **Перехресна таблиця**.

Створений запит можна зберегти під ім'ям **ОцінкиЗаПредметами** (рис. 4.17).

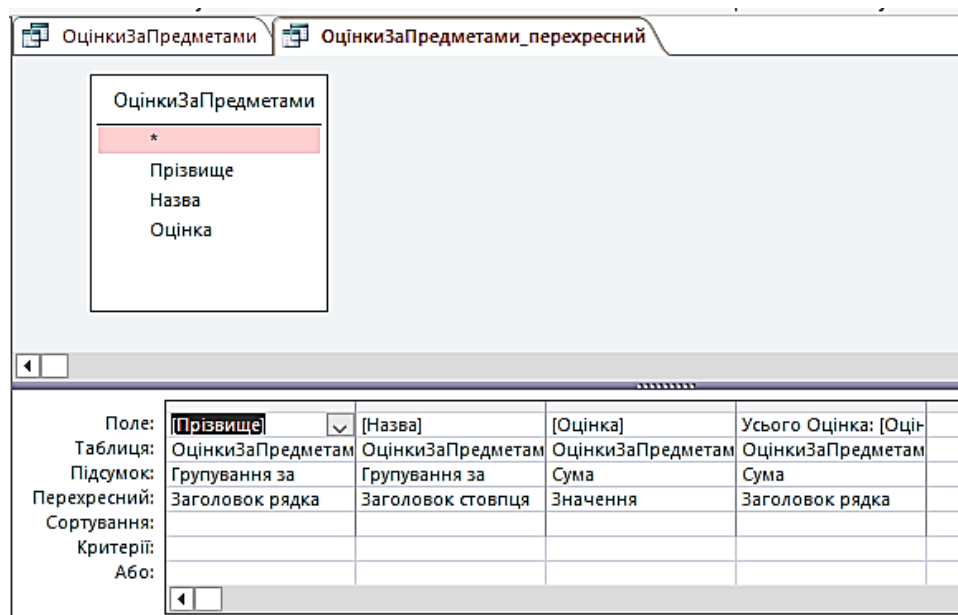


Рис. 4.17. Макет перехресного запиту **ОцінкиЗаПредметами**

У результаті виконання перехресного запиту **ОцінкиЗаПредметами** буде отримана електронна таблиця, яка містить оцінки студентів з усіх дисциплін (рис. 4.18). Цей запит може бути використаний, наприклад, у звіті, що має відображувати підсумки попередньої сесії.

Прізвище	Усього Оцін	Економіка	Математики	Фізика	Філософія	Хімія
Жукова	10			5		5
Ігнатов	9			5		4
Морозов	8			4		4
Петров	12	4	5		3	

Рис. 4.18. Результат виконання запиту ОцінкиЗаПредметами

4.2. Типи запитів на зміну

Як правило, при роботі з таблицями використовується запит на вибірку. Однак у багатьох випадках потрібно за допомогою запитів виконувати різні зміни в БД, а саме створювати нові таблиці, оновлювати, додавати і видаляти записи в існуючих таблицях [1].

Для цього використовуються *запити на зміну*, які можна поділити відповідно на такі категорії: *створення таблиці, оновлення, додавання і видалення*. Слід зазначити, що перед створенням будь-якого запиту на зміну спочатку створюється запит на вибірку з потрібною структурою, після чого вибирається необхідний тип запиту [1].

Створення таблиці. У тому випадку, коли при багаторазовому виконанні запиту на вибірку відбувається відбір тих самих даних, то їх доцільно оформити у вигляді нової таблиці за допомогою запиту на створення.

Такий запит буде особливо ефективним, якщо процес одержання даних досить трудомісткий, тобто коли доводиться виконувати велику кількість обчислень або працювати зі значними обсягами інформації.

Щоб створити за допомогою запиту нову таблицю, потрібно за допомогою конструктора побудувати запит на вибірку, після чого виконати команду *Створення | Макет запиту | Створення таблиці* (рис. 4.19).

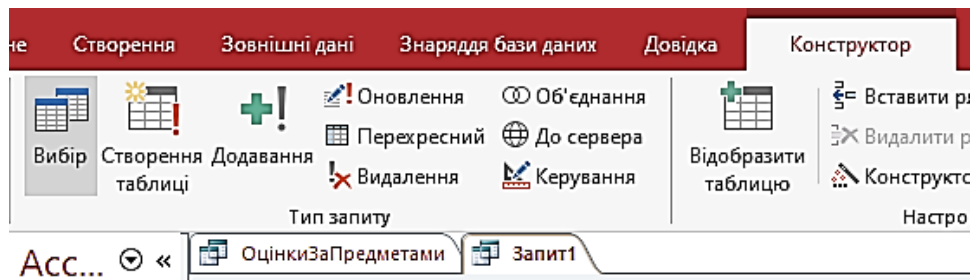


Рис. 4.19. Кнопка *Тип запиту*

Якщо обрано режим **Створення таблиці**, то в цьому разі з'явиться діалогове вікно *Створення таблиці* (рис. 4.20), у якому потрібно вказати ім'я нової таблиці, а також її розташування, тобто таблиця буде створена в поточній БД або в тій, яка зазначена в рядку *Ім'я файлу*.

Наприклад, потрібно створити нову таблицю на підставі розглянутого раніше запиту на вибірку **ВикладачіПредмети**. Для цього відкрити цей запит у режимі **Конструктор** і вибрати тип запиту *Створення таблиці*. Потім вказати ім'я створюваної таблиці, наприклад **ПредметиВикладачі**, і натиснути кнопку *ОК*.

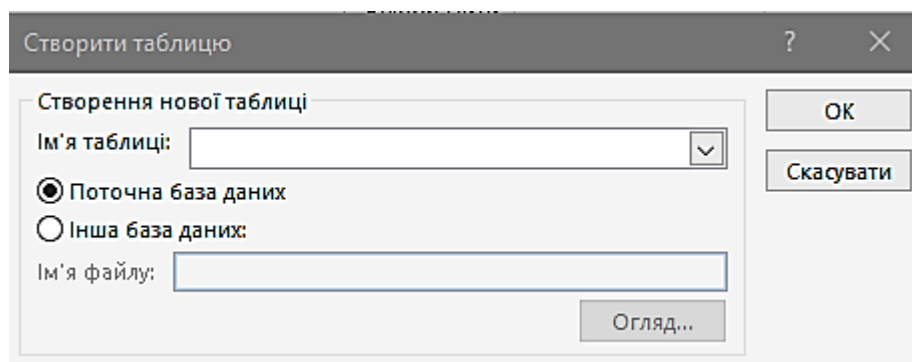

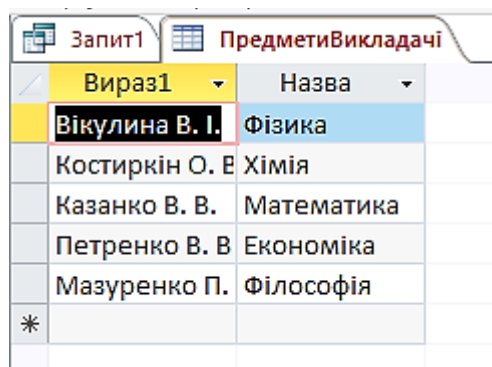


Рис. 4.20. Діалогове вікно *Створення таблиці*

При виконанні цього запити за допомогою кнопки *Запуск*  буде відображено вікно попередження про створення нової таблиці і розміщення

в ній обраних записів. Після підтвердження, тобто вибору кнопки *Так*, у БД **Освіта** буде створена нова таблиця **ПредметиВикладачі** (рис. 4.21).



Вираз1	Назва
Вікулина В. І.	Фізика
Костиркін О. Е	Хімія
Казанко В. В.	Математика
Петренко В. В	Економіка
Мазуренко П.	Філософія
*	

Рис. 4.21. Нова таблиця **ПредметиВикладачі**

Слід зазначити, що в БД не може бути двох об'єктів з однаковими іменами (це стосується тільки таблиць і запитів). Тому якщо в описаному прикладі новій таблиці було б присвоєно ім'я **ВикладачіПредмети**, що збігається з ім'ям запиту, то було б відображено попередження про те, що об'єкт з таким ім'ям у цій базі вже є.

Оновлення. Цей тип запиту на зміну використовується тоді, коли потрібно виконувати відбір записів з наступною зміною для них значення певного поля.

Як приклад можна розглянути таблицю **Студенти** в БД **Освіта**. Кожен студент у цій БД може перебувати в одному з чотирьох станів: *навчається, в академічній відпустці, відрахований, закінчив*.

Щоб мати можливість змінювати стан студента, тобто відраховувати, відновлювати, відправляти в академічну відпустку або «повертати» з відпустки, можна в таблицю **Студенти** додати поле **Стан**. Якщо студент навчається, то це поле має бути порожнім. Якщо ж студент відрахований, то потрібно в полі **Стан** вказати ознаку «в», якщо в академічній відпустці – ознаку «а», якщо закінчив – ознаку «з».

Для додавання нового поля потрібно відкрити таблицю **Студенти** в режимі **Конструктор**. Потім задати нове поле **Стан**, для якого вибрати тип *Текстовий* і визначити такі властивості [1]:

Розмір поля: 1;

Умова на значення: "" Or "a" Or "в" Or "з";

Порожні рядки: Так.

У полі **Стан** можуть перебувати як символи «a», «в» або «з», так і порожній рядок, тобто «». Тому у властивості *Умова на значення* потрібно вказати, що в це поле може бути введено тільки одне з цих значень (або не введено нічого). Крім того, у властивості *Порожні рядки* вказати, що в полі **Стан** можуть зберігатися порожні рядки.

Після установаження властивостей і переходу в режим таблиці буде відображено попередження про можливе недотримання цілісності даних, установленної при створенні зв'язків між таблицями. У цьому випадку можна не виконувати перевірку на цілісність, тому що її порушення відбулося через додавання нового поля, що не має спричинити небажані зміни у БД [1, 4, 5].

У режимі таблиці можна внести деякі зміни в поле **Стан** для окремих студентів (табл. 4.3). Водночас слід зазначити, що при спробі введення в поле **Стан** значення, що не відповідає зазначеній умові, буде показано вікно попередження. Також потрібно мати на увазі, що при перевірках і обчисленнях букви, які мають однаковий вигляд у латиниці і кирилиці, розрізняються. Інакше кажучи, якщо ввести в поле **Стан** латинську букву «a», то Access це сприйме як спробу запису в поле помилкового значення.

У подальших міркуваннях можна зробити допущення, що на початку кожного навчального року всі «академісти» повертаються до навчання, тому для них потрібно в полі **Стан** видаляти ознаку «a». У подібних ситуаціях зручно використати запит на оновлення, у якому для всіх записів, у яких значення поля **Стан** дорівнює «a», буде виконане видалення цієї ознаки.

Таблиця Студенти з новим полем **Стан**

КодСтудента	Прізвище	Ім'я	По батькові	Стан
1	Жукова	Марина	Володимирівна	
2	Морозов	Ігор	Олегович	
3	Петров	Михайло	Олександрович	
4	Ігнатов	Максим	Валерійович	
5	Волков	Юрій	Олексійович	а
6	Волошина	Вікторія	Іванівна	
7	Литвинов	Євгеній	Валерійович	в
8	Гущина	Олена	Валеріївна	а
9	Кузнецов	Руслан	Олексійович	в
10	Семененко	Світлана	Станіславівна	в

Для виконання описаних дій потрібно створити новий запит у режимі **Конструктор** і додати до макета таблицю **Студенти**. Потім по черзі додати в запит усі поля цієї таблиці, причому для поля **Стан** вказати параметр *Умова відбору*: «а». Після цього можна виконати цей запит, щоб переконатися в його функціональності. Заключним етапом у створенні запиту буде зміна його типу з *Вибірка* на *Оновлення* кнопкою *Тип запиту* (рис. 4.19) після чого в макеті з'явиться новий параметр – *Оновлення*. Для поля **Стан** потрібно вказати *Оновлення*: «» (порожній рядок), щоб для всіх студентів-академістів була вилучена ознака «а». Отриманий таким способом запит (рис. 4.22) можна зберегти під ім'ям **Академісти**.

При виконанні запиту **Академісти** в цьому випадку буде відображено повідомлення про зміну двох записів, що відповідає поточній кількості академістів. Після підтвердження оновлення записів усі значення, рівні «а», будуть вилучені з поля **Стан** (рис. 4.23).

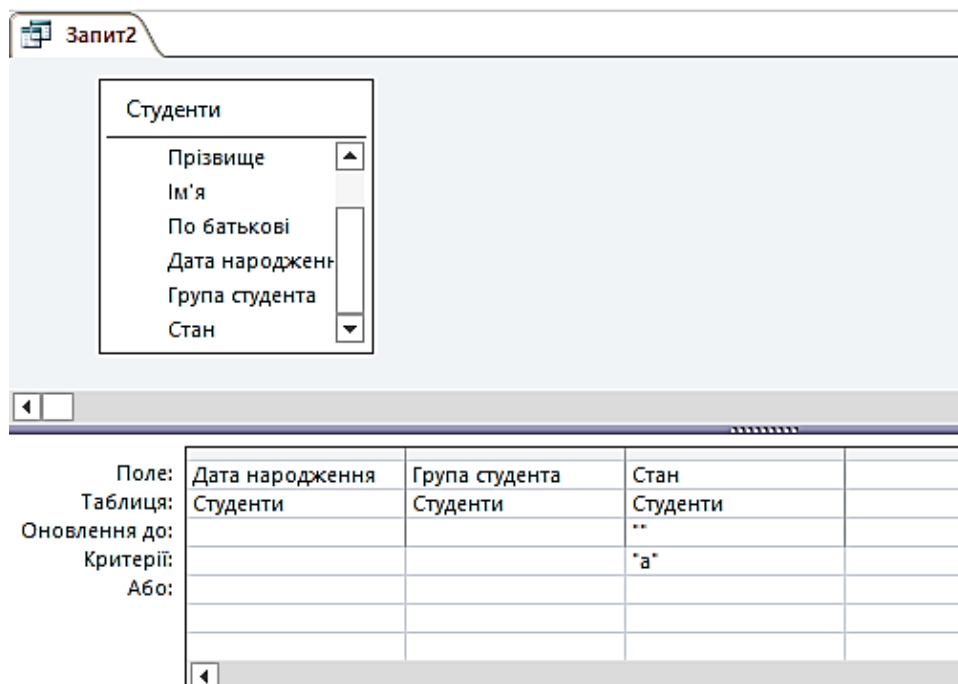


Рис. 4.22. Макет запиту Академісти

КодСтудент	Прізвище	Ім'я	По батькові	Дата народ	Група студе	Стан
1	Жукова	Марина	Володимирів			
2	Морозов	Ігор	Олегович			
3	Петров	Михайло	Олександрови			
4	Ігнатов	Максим	Валерійович			
5	Волков	Юрій	Олексійович			
6	Волошина	Вікторія	Іванівна			
7	Литвинов	Євген	Валерійович			в
8	Гущина	Олена	Валеріївна			
9	Кузнецов	Руслан	Олексійович			в
10	Семененко	Світлана	Станіславівна			в
*	(Новий)					

Рис. 4.23. Оновлена таблиця Студенти (без академістів)

Слід також зазначити, що в макеті запиту **Академісти** після того, як його тип був змінений на *Оновлення*, автоматично будуть вилучені всі поля, що не оновлюються, тобто залишиться тільки поле **Стан**, яке буде оновлюватись.

Додавання. Іноді виникає необхідність тимчасового зберігання якихось даних з можливістю їхнього відновлення. Наприклад, з таблиці

Студенти потрібно час від часу видаляти відрахованих студентів. Однак відрахований студент може бути відновлений, і тому дані про всіх відрахованих студентів можна зберігати в архівній таблиці **СтудентиВідраховані**, і якщо виникне потреба у відновленні інформації в БД про того або іншого студента, можна буде просто скопіювати відповідний запис із таблиці **СтудентиВідраховані** у таблицю **Студенти**.

Для обробки такого архіву спочатку потрібно створити в БД **Освіта** нову таблицю **СтудентиВідраховані**, структура якої буде збігатися зі структурою таблиці **Студенти**. Тому найкраще в цій ситуації скористатися операцією копіювання.

Тобто потрібно у вікні БД **Освіта** виділити таблицю **Студенти**, натиснути кнопку *Копіювати*, і потім натиснути кнопку *Вставити*. Після цього буде відображене вікно *Вставка таблиці*, у якому потрібно вказати ім'я нової таблиці **СтудентиВідраховані** і вибрати режим вставки **Тільки структура** (рис. 4.24).

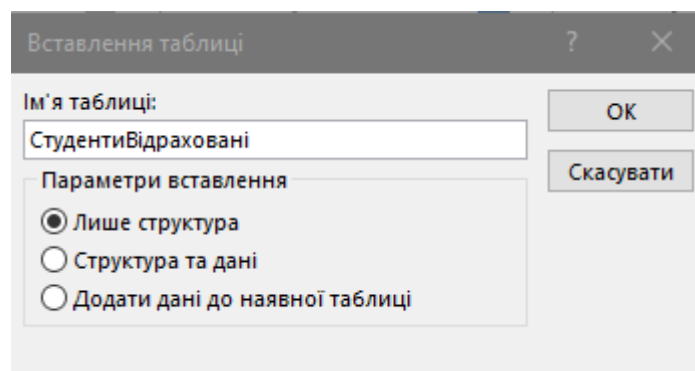


Рис. 4.24. Діалогове вікно *Вставка таблиці*

Потім створити новий запит у режимі **Конструктор**, додати до його макета таблицю **Студенти**, і по черзі додати до структури запиту всі поля, крім **КодСтудента**, тому що значення поля **КодСтудента** формуються автоматично і їх не потрібно переносити з таблиці **Студенти** в таблицю **СтудентиВідраховані**. З іншого боку, через те, що поле **КодСтудента** є

первинним ключем, при спробі повторного копіювання тих самих студентів буде виникати помилка, тому що значення первинного ключа має бути унікальним для кожного запису.

Після цього вибрати тип запиту **Додавання**, у результаті чого буде відображене діалогове вікно *Додавання*, у якому потрібно в ролі «приймача» даних вибрати таблицю **СтудентиВідраховані** (рис. 4.25).

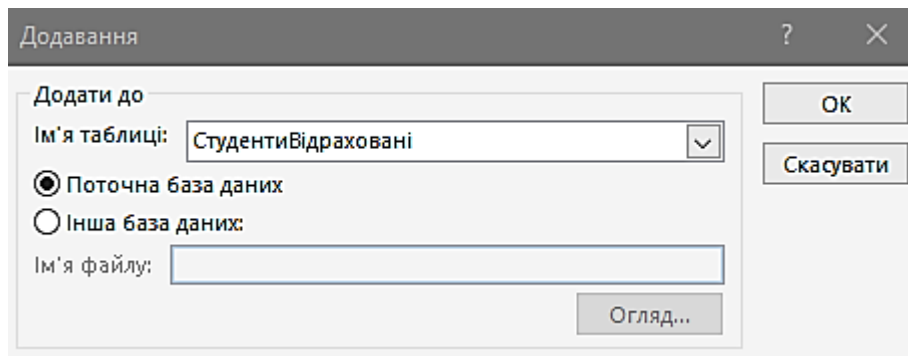


Рис. 4.25. Діалогове вікно *Додавання*

Водночас у макеті запити з'явиться новий рядок *Додавання*, до якого автоматично будуть додані імена відповідних полів таблиці **СтудентиВідраховані**. Потім треба для поля **Стан** указати параметр *Умова відбору*: «в» і зберегти створений запит під ім'ям **ДодаванняВідрахованих** (рис. 4.26).

Тепер при кожному виконанні запити **ДодаванняВідрахованих** до таблиці **СтудентиВідраховані** будуть додаватися відраховані студенти.

Видалення. Разом з додаванням записів часто виникає необхідність видалення записів з таблиці при виконанні запити. Наприклад, у розглянутому випадку з відрахованими студентами при кожному новому виконанні запити **ДодаванняВідрахованих** будуть додаватися вже скопійовані раніше записи, тому що умовою для запити є тільки наявність ознаки «в» у полі **Стан**.

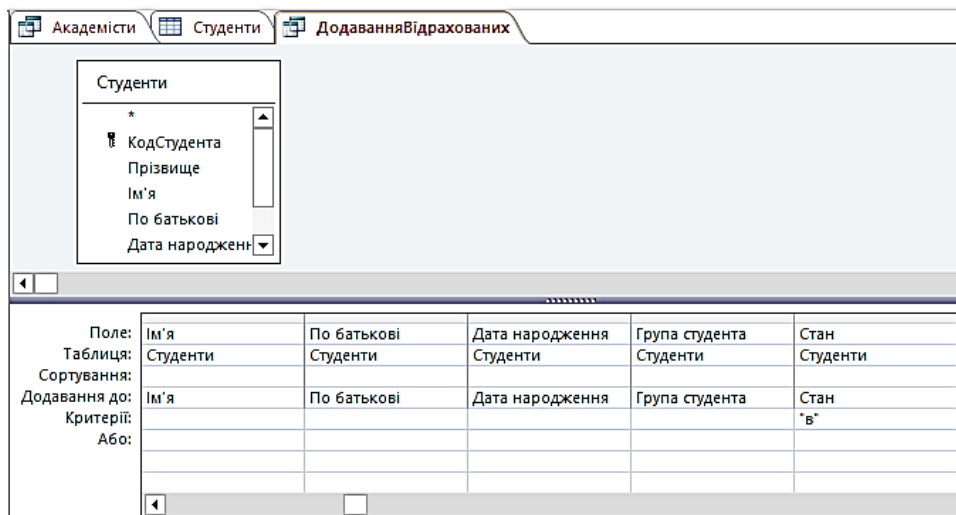


Рис. 4.26. Макет запити ДодаванняВідрахованих

Тому зручним доповненням до запити на додавання є *запит на видалення*, за допомогою якого в цьому випадку можна після додавання до таблиці **СтудентиВідраховані** видалити всіх відрахованих студентів з таблиці **Студенти**.

При створенні запити на видалення слід мати на увазі, що з записами, що видаляються, можуть бути пов'язані дані з іншою таблицею. Наприклад, студенти **Жукова** і **Морозов** мають кілька оцінок, тобто існує безпосередній зв'язок між даними в таблицях **Студенти** та **Оцінки**. Інакше кажучи, неможливо просто видалити записи для цих студентів з таблиці **Студенти**, тому що в цьому разі для декількох записів з таблиці **Оцінки** у полі **КодСтудента** буде міститися посилання на неіснуючі записи в таблиці **Студенти**.

Для запобігання подібній ситуації потрібно перевірити, який вид зв'язку є між полями **КодСтудента** таблиць **Студенти** та **Оцінки**, і переконатися, що для цього зв'язку в діалоговому вікні *Зміна зв'язків* (рис. 4.27) встановлено режим **Каскадне видалення зв'язаних записів**.

Створення запити виконується так само, як і у попередніх випадках. Для простоти можна в цьому випадку скористатися готовим запитом

Додавання Відрахованих, зберігши його під новим ім'ям
Видалення Відрахованих. Водночас встановити тип запиту **Видалення** в
 меню *Конструктор* (рис. 4.28).

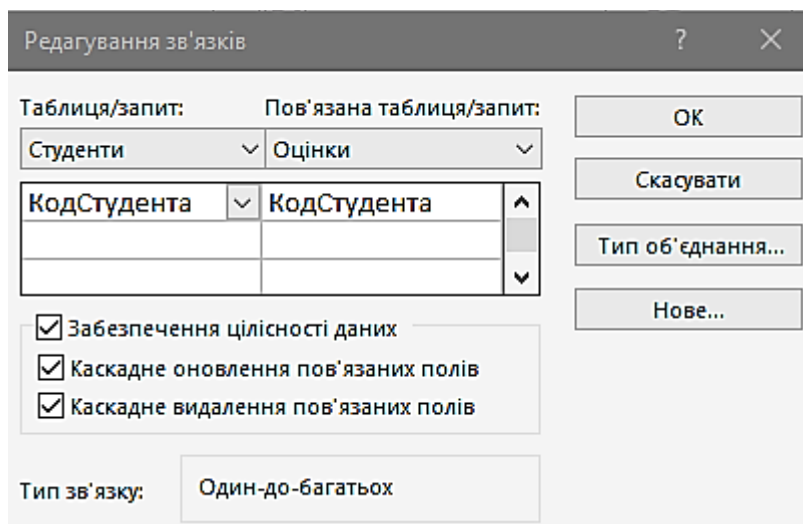


Рис. 4.27. Діалогове вікно *Зміна зв'язків*

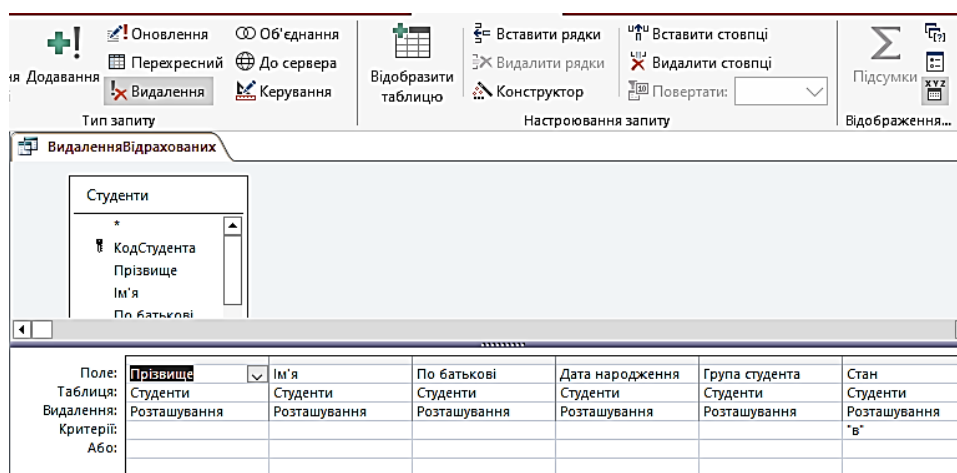



Рис. 4.28. Макет запиту **Видалення Відрахованих**

Тепер при кожному запуску цього запиту з таблиці **Студенти** будуть видалятися всі відраховані студенти. Крім того, будуть також видалятися всі пов'язані з ними записи з таблиці **Оцінки**. Інакше кажучи, із БД буде видалятися і студент, і його оцінки.

4.3. Властивості запитів

Для будь-якого запиту можуть бути задані певні параметри, якими визначаються особливості його виконання. Інакше кажучи, для кожного запиту існує набір властивостей, використовуваних за замовчуванням, і заміна значень цих властивостей тим або іншим способом буде впливати на процес виконання запиту.

Щоб змінити будь-яку властивість, потрібно відкрити запит у режимі конструктора і, не встановлюючи курсор ні на яке поле в макеті запиту, скористатися командою *Вид | Властивості* або натиснути кнопку *Властивості* . Після цього буде відображено вікно *Властивості запиту* (рис. 4.29). Далі будуть розглянуті основні властивості запитів.

Виведення всіх полів – відображення в запиті всіх полів з таблиць, використовуваних як джерело даних. Як правило, для цієї властивості встановлюється значення *Немає*.

Набір значень – виведення на екран перших *n* записів (5, 25 або 100) або *n*% загальної кількості записів (5 %, 25 % або 100 %). Цю властивість корисно використовувати в тих випадках, коли кількість знайдених записів у виконаному запиті буде достатньо великою, тому можна відображувати вже відібрані записи, не чекаючи на закінчення виконання запиту.

Унікальні значення – використовується для відображення в запиті тільки унікальних значень серед знайдених записів. Інакше кажучи, якщо для цієї властивості встановлено значення *Так*, то в запиті будуть відображатися тільки записи з унікальними сполученнями значень у відображуваних полях.

Унікальні записи – застосовується для відображення унікальних записів тільки в багатотабличних запитах. При створенні багатотабличного запиту може виникнути ситуація, коли у відібраних записах має відображатися поле (або кілька полів) однієї з таблиць, тоді як поле

пов'язаної з нею таблиці використовується в структурі запити тільки для зазначення умови для відбору записів. Тобто поля другої таблиці не будуть включені до результату виконання запиту.

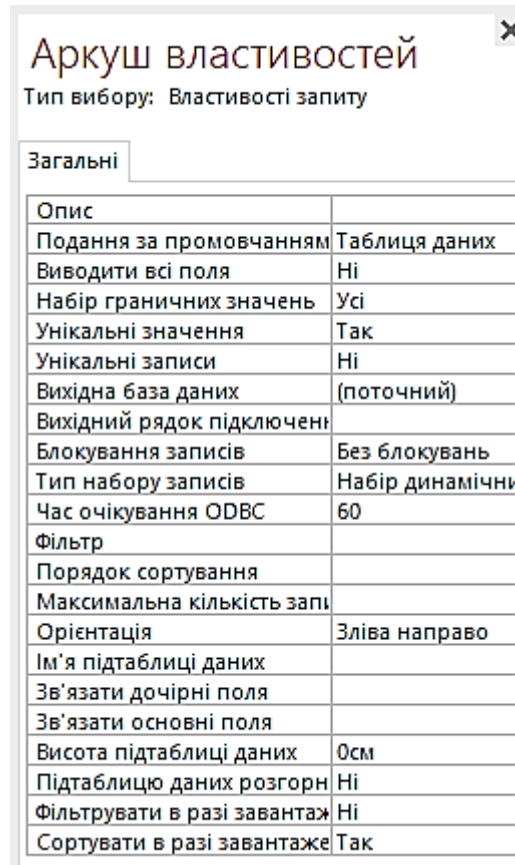


Рис. 4.29. Вікно *Властивості запиту*

Фільтр – застосування зазначеного фільтра для знайдених записів, тобто відображення тільки тих записів, які будуть відповідати зазначеній умові.

Порядок сортування – визначення напрямку сортування: *за зростанням* або *убуванням*.

Як приклад використання властивостей *Унікальні значення* та *Унікальні записи* можна розглянути два запити. Перший запит буде включати тільки поле **Оцінка** таблиці **Оцінки**. Якщо в цьому випадку для властивості *Унікальні значення* буде встановлене значення *Так*, то в

результаті виконання цього запиту будуть відібрані тільки ті записи, для яких значення в полі **Оцінка** не повторюються (рис. 4.30).

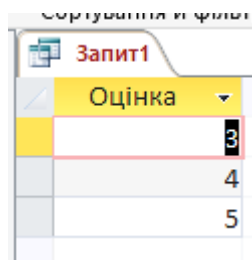


Рис. 4.30. Використання властивості *Унікальні значення*

Другий запит буде являти собою доповнення першого. До структури запиту потрібно додати поле **Назви** таблиці **Предмети**, водночас встановити для нього прапорець *Виведення на екран*, а для поля **Оцінка** цей прапорець прибрати, щоб після виконання запиту у знайдених записях відображувалися тільки назви дисциплін. Якщо властивості *Унікальні записи* для цього запиту присвоєно значення *Ні*, то в результаті його виконання будуть знайдені всі записи, у тому числі і з повторюваними назвами (рис. 4.31). Якщо ж встановити для властивості *Унікальні записи* значення *Так*, то будуть відображені тільки записи з унікальними значеннями в полі Предмет (рис. 4.32).

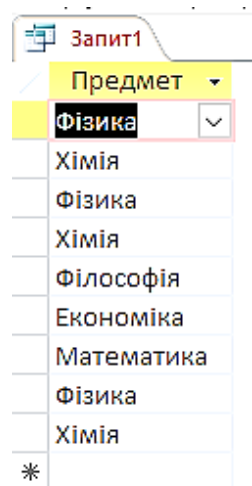


Рис. 4.31. *Унікальні записи* не встановлено

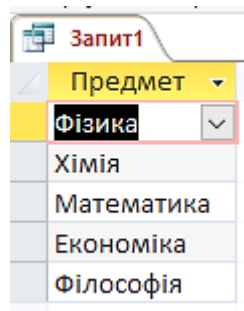


Рис. 4.32. Унікальні записи встановлено

Слід зазначити, що властивість *Унікальні значення* використовується тільки для полів, відображуваних у запиті, водночас джерелом даних служить одна таблиця. З іншого боку, властивість *Унікальні записи* встановлюється для багатотабличних запитів, у яких поля деяких таблиць присутні в запиті, але не відображуються в знайдених записах. Тому через розходження у сфері дії даних властивостей для того самого запиту неможливо одночасно встановити значення *Так* і для властивості *Унікальні значення*, і для властивості *Унікальні записи*.

Контрольні запитання до розділу 4

1. Навести типи запитів.
2. Чи існують запити, що використовують дані з декількох таблиць?
3. Що є першим кроком у створенні запиту за допомогою майстра?
4. Чи використовуються при створенні запитів поля, що обчислюються?
5. Чи можна за допомогою *Будівника виразів* виконувати різні обчислення з використанням даних?
6. Що потрібно виконати для створення текстового рядка?
7. Що потрібно виконати, щоб установити підпис для поля в таблиці або запиті?
8. Які функції потрібно використовувати при виконанні групових операцій?

Розділ 5. ФОРМИ

5.1. Види форм

Робота з даними в БД безпосередньо здійснюється в таблицях і запитах. Водночас пропонувані можливості зі зміни структур таблиць і даних, що містяться в них, корисні скоріше розробнику, ніж користувачеві. *В Access є широкі можливості для створення інтерфейсу користувача за допомогою форм [1].*

Створюючи форму в БД, розробник може досягти одну з декількох цілей [1]:

1) *відображення і зміна даних.* Найчастіше форми створюються саме для виведення і зміни даних у БД. Водночас розробник може задати будь-який вид відображення інформації, тобто довільний набір полів з однієї або декількох таблиць чи запитів (якщо поля перебувають у різних таблицях, то між цими таблицями має бути зв'язок). Також можна встановити будь-які обмеження на редагування даних у формі, тобто дозволити редагування одних полів і заборонити зміну інших;

2) *введення даних.* Форми можуть використовуватися для введення нових даних до таблиці, водночас користувач буде вводити дані в тому форматі, який задав розробник. Інакше кажучи, користувач може вводити значення тільки для тих полів, які наведені у формі;

3) *друк.* Незважаючи на те, що для друку даних у БД використовуються звіти, вміст форми також можна роздрукувати;

4) *повідомлення.* Додаткова можливість використання форм полягає у створенні за їхньою допомогою різних повідомлень, що відображатимуться згодом у тій або іншій ситуації для полегшення роботи користувача. У такому вигляді форми використовуються в додатках.

Отже, можна зробити висновок, що форми в Access є головним засобом розроблення користувальницького інтерфейсу. Залежно від того, з якою метою створюється форма, розробник визначає параметри її зовнішнього вигляду. Існує кілька основних видів форм, наведених нижче.

Багатосторінкова (у стовпчик) – поля запису виводяться в один стовпчик, водночас у цей самий момент часу у формі відображується вміст тільки поточного запису. На рис. 5.1 наведена багатосторінкова форма **Студенти**, у якій відображені дані про студентів з таблиці **Студенти**.

Прізвище	Ім'я	По батькові	Стан
Жукова	Марина	Володимирівна	

Рис. 5.1. Багатосторінкова форма

Рядкова – всі поля запису виводяться в один рядок, водночас у формі відображуються всі записи. У рядковій формі **Студенти**, наведеній на рис. 5.2, відображені всі записи з таблиці **Студенти**.

Таблична – відображення записів здійснюється в режимі таблиці. На рис. 5.3 подано табличну форму **Студенти**, у якій усі записи таблиці **Студенти** виведені в табличному режимі.

СтудентиФорма			
Прізвище	Ім'я	По батькові	Стан
Жукова	Марина	Володимирівна	
Морозов	Ігор	Олегович	
Петров	Михайло	Олександрович	
Ігнатов	Максим	Валерійович	
Волков	Юрій	Олексійович	
Волошина	Вікторія	Іванівна	
Литвинов	Євген	Валерійович	в
Гущина	Олена	Валеріївна	
Кузнєцов	Руслан	Олексійович	в
Семененко	Світлана	Станіславівна	в
*			

Рис. 5.2. Рядкова форма

СтудентиФормаТаблиця			
Прізвище	Ім'я	По батькові	С
Жукова	Марина	Володимирівна	
Морозов	Ігор	Олегович	
Петров	Михайло	Олександрович	
Ігнатов	Максим	Валерійович	
Волков	Юрій	Олексійович	
Волошина	Вікторія	Іванівна	
Литвинов	Євген	Валерійович	в
Гущина	Олена	Валеріївна	
Кузнєцов	Руслан	Олексійович	в
Семененко	Світлана	Станіславівна	в
*			

Рис. 5.3. Таблична форма

Підлегла – використовується за потреби виведення даних, пов'язаних відношенням «один до багатьох», водночас у підлеглій формі

відображуються записи з таблиці, що відповідає слову «багатьох». У формі **ОцінкиСтудентів** (рис. 5.4), крім даних з таблиці **Студенти**, є також підлегла форма, у якій відображуються записи з таблиці **Оцінки**.

Оцінка	Предмет
5	Фізика
5	Хімія
*	0

Рис. 5.4. Підлегла форма

Слід звернути увагу на той факт, що у стовпчику **Предмет** цієї форми виводяться назви предметів, тоді як у таблиці **Оцінки**, що служить джерелом даних для підлеглої форми, є тільки коди предметів. Подібний ефект досягається за допомогою майстра підстановок для таблиці **Оцінки**, причому як підстановка для поля **КодПредмета** було використано поле **Назва** з таблиці **Предмети**.

5.2. Створення форми

Щоб створити форму, потрібно в меню **Створення** вибрати секцію *Форми* (рис. 5.5). Основними режимами створення будь-якої форми є **Конструктор**, **Майстер форм** і **Форма** [1].

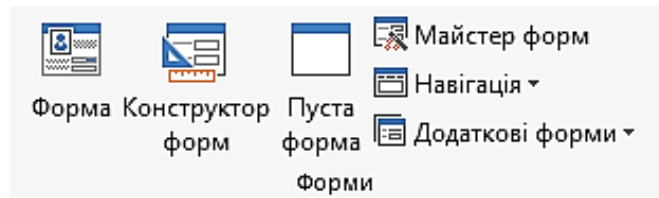


Рис. 5.5. Меню **Форми**

Майстер форм. Багато форм можуть бути створені в режимі конструктора, який буде розглянуто нижче, не вдаючись до допомоги додаткових засобів побудови. Однак більш простим і наочним методом розроблення форм є використання майстра, за допомогою якого можна створити основну частину форми (зокрема задати набір полів для відображення), після чого в режимі конструктора зробити «косметичні» зміни (наприклад змінити розташування полів на формі) [1].

Якщо в якомусь полі таблиці використовується підстановка, то при додаванні такого поля до форми майстер створення форм автоматично встановлює режим відображення так, щоб замість реального значення в цьому полі використовувалась підстановка. При розробленні форми в режимі конструктора таке використання підстановки буде складно реалізувати, тому що в цьому випадку доведеться змінювати значну кількість властивостей того елемента, у якому має відображатися значення підстановки [1].

Як приклад роботи майстра буде розглянуто створення форми **Студенти**, у якій відображатиметься інформація про студентів. Після вибору режиму **Майстер форм** і таблиці-джерела **Студенти** (у списку, що розкривається) потрібно натиснути **ОК**.

На першому етапі роботи майстра потрібно вибрати поля, що будуть використовуватися у створюваній формі. У цьому випадку вибрати поля **Прізвище**, **Ім'я**, **по батькові** та **Стан** (рис. 5.6) і потім натиснути *Далі*>. Слід зазначити, що поля можуть додаватися до форми з декількох таблиць

– для цього потрібно вибирати потрібну таблицю у списку, що розкривається, *Таблиці та запити* вікна **Майстри форм**.

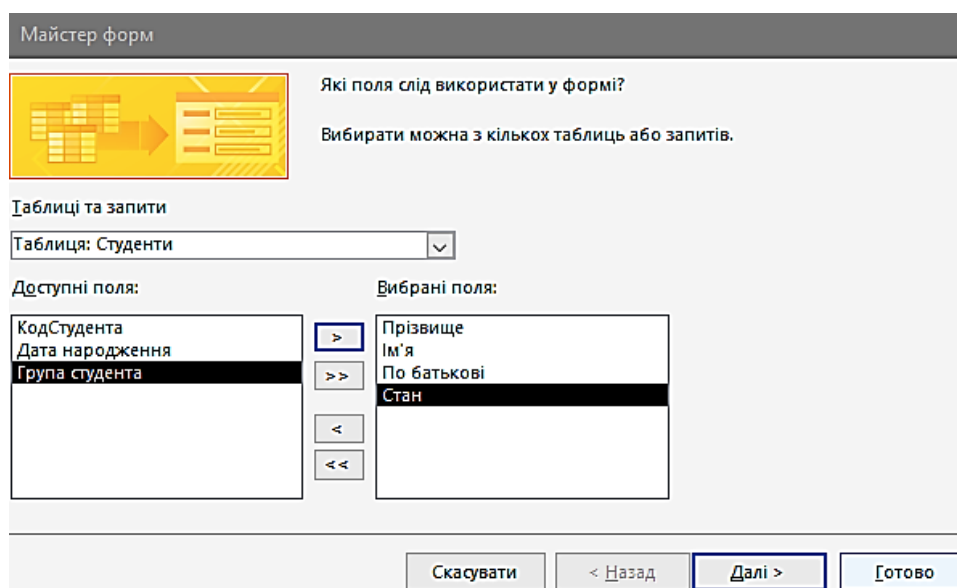


Рис. 5.6. Перший етап роботи майстри форм

5.3. Вибір полів

Другий етап передбачає вибір зовнішнього вигляду форми. У нашому випадку можна вибрати режим **В один стовпчик** і натиснути кнопку *Далі >*.

На третьому етапі роботи майстра потрібно вибрати стиль дизайну форми, що створюється (рис. 5.7). Для форми **Студенти** можна скористатися стилем *Стандартний* і потім натиснути *Далі >*.

Останній етап припускає задавання імені новій формі. Тут потрібно ввести ім'я **Студенти** і як подальші дії вибрати пункт *Відкрити форму для перегляду та введення даних*. Для завершення роботи майстра потрібно натиснути кнопку *Готово* [1].

У результаті буде отриманий найпростіший варіант багатосторінкової форми **Студенти**, у якій відображується інформація про студентів (рис. 5.8).

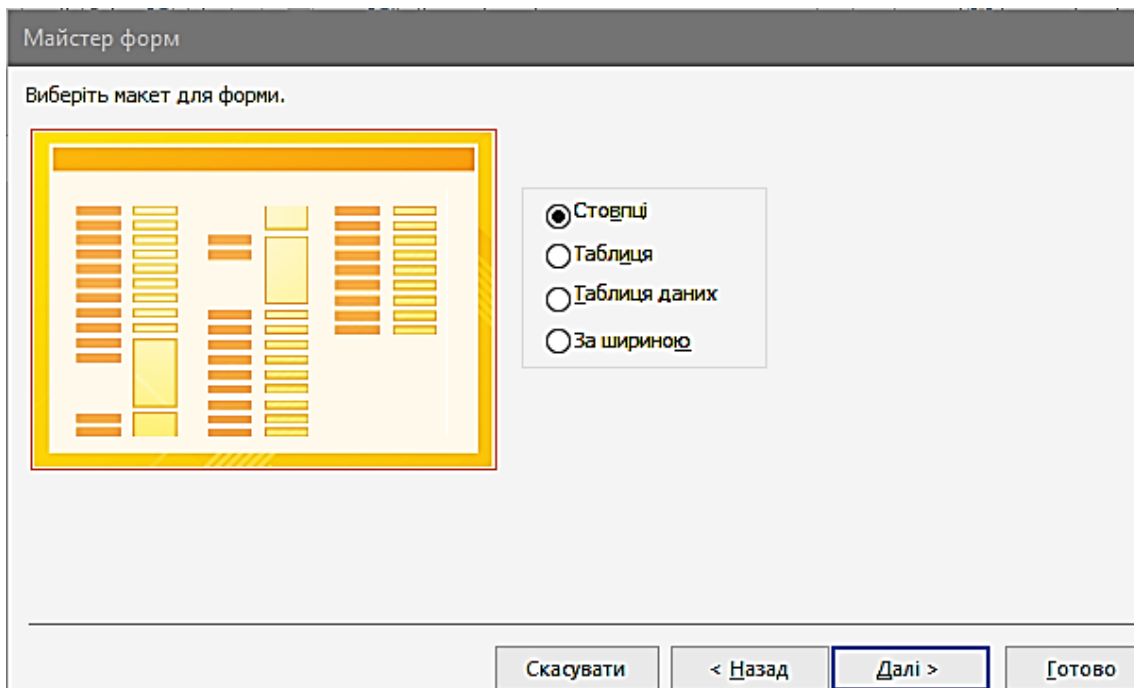






Рис. 5.7. Третій етап роботи майстра форм. Вибір стилю

Прізвище	Жукова
Ім'я	Марина
По батькові	Володимирівна
Стан	


Рис. 5.8. Форма Студенти

Слід зазначити, що переміщення між записами, а також додавання нових записів виконується за допомогою засобу *Запис* аналогічно роботі в режимі таблиці (табл. 5.1).

Кнопки в області виділення рядка



Кнопка	Опис
	Поточний запис, уже збережений
	Новий запис, у який можна вводити дані
	Запис, що змінюється, ще не збережений
	Запис у цей момент заблокований

Автоформа. У випадку, коли джерелом даних для форми є одна таблиця або запит, можна скористатися режимом **Автоформа** [1].

У цьому випадку Access автоматично обирає поля і розміщує їх на формі. Для створення форми в автоматичному режимі можна скористатись описаним раніше діалоговим вікном *Нова форма* або ж спеціальною кнопкою на панелі інструментів *Автоформа* . Форма **ОцінкиСтудентів** була створена саме так.

Конструктор. Як правило, режим конструктора використовується для вдосконалення зовнішнього вигляду форми, створеної за допомогою майстра [1].

Робота з конструктором форм буде розглянута на прикладі створеної форми **Студенти**.

Щоб викликати режим конструктора для відкритої форми, потрібно виконати команду *Вид | Конструктор* або скористатися кнопкою *Вид*, вибравши зі списку, що розкривається, *режим Конструктор* . З іншого боку, щоб вибрати режим перегляду форми, перебуваючи в конструкторі, можна виконати команду *Вид | Режим форми* або задати цей режим за допомогою кнопки *Вид* . Структура будь-якої форми (рис. 5.9) являє собою набір нижченаведених компонентів [1].

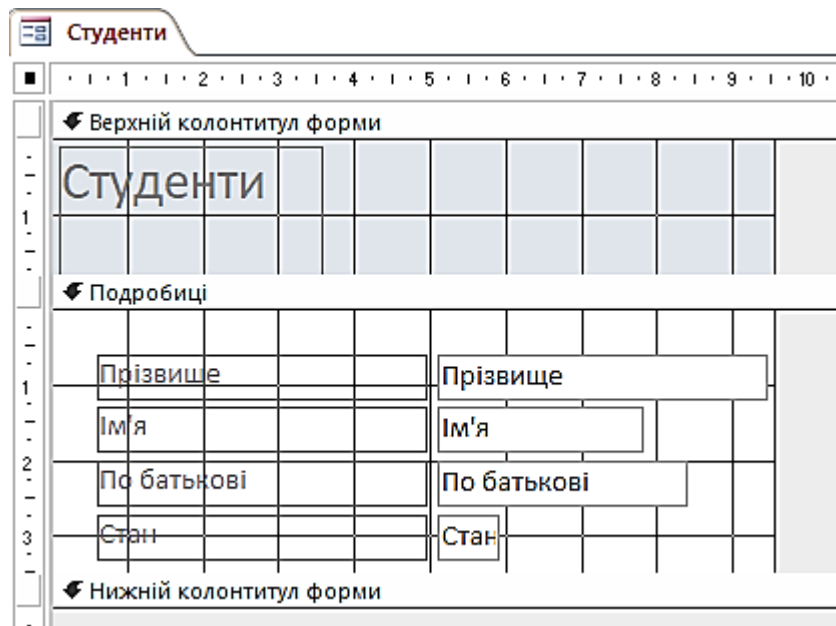



Рис. 5.9. Форма Студенти в режимі конструктора

Заголовок форми – область у верхній частині форми, у якій міститься інформація, що не залежить від вмісту відображуваних записів. Зокрема в заголовку форми може бути розташований напис із назвою форми (наприклад «Дані про студентів»), а також назви стовпчиків, тобто полів записів (**Прізвище**, **Ім'я**, **По батькові** і **Стан**). Приклад заголовка з зазначенням назви форми і підписів до стовпчиків подано в рядковій формі **Студенти**.

Область даних – центральна частина форми, у якій відображується інформація з таблиць або запитів, а також пояснювальні написи (зокрема назви полів і їхній вміст).

Примітка форми – область у нижній частині форми, що містить не залежну від записів інформацію, аналогічно заголовку.

Уся інформація відображується у формі за допомогою так званих *елементів керування*. Щоб додати до форми новий елемент керування, потрібно скористатися *Панеллю елементів* (рис. 5.10), відображуваною командою *Вид | Панель елементів* або кнопкою *Панель елементів*  [1].

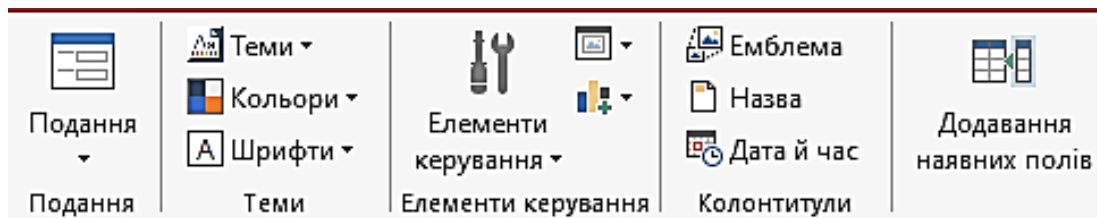











Рис. 5.10. Панель елементів

Перелік усіх елементів керування, а також їхній опис наведено в табл. 5.2.


Таблица 5.2

Список елементів керування [1]


Кнопка	Назва	Опис
1	2	3
	Напис	Містить текст для опису елемента керування або іншого об'єкта (заголовки форми, підписи до полів)
	Поле	Відображення будь-яких значень (як правило, полів з таблиць і запитів)
	Група перемикачів	Використовується для об'єднання в групу декількох елементів вибору (вимикачів, перемикачів або прапорців)
	Вимикач	Застосовується для вибору користувачем значення типу Так/Ні або Вкл/Викл
	Перемикач	Використовується в тому випадку, коли з декількох значень або параметрів потрібно вибрати тільки один
	Галочка	Застосовується для вибору декількох значень або параметрів з наявного набору
	Поле зі списком	Використовується для роботи з усіма значеннями певного поля таблиці або запиту. Водночас в полі відображується поточне обране значення, у той час як за допомогою списку, що розкривається, можна вибрати інше значення поля з наявного набору записів
	Список	Застосовується для відображення всіх значень певного поля таблиці або запиту, водночас у списку виділене поточне значення цього поля. Вибір іншого значення виконується за допомогою виділення відповідного елемента списку
	Кнопка	Використовується для виконання будь-якої дії, наприклад відображення іншої форми або звіту

1	2	3
	Рисунок	Дозволяє розмістити на формі рисунок (як ілюстрацію)
	Вільна рамка об'єкта	Застосовується для додавання до форми зовнішнього об'єкта ActiveX (діаграма, рисунок або звуковий файл), які не містяться безпосередньо в таблиці БД
	Приєднана рамка об'єкта	Використовується для додавання до форми об'єкта ActiveX, що перебуває в таблиці БД. Для об'єктів, які не можуть зберігатися в таблиці, у формі відображується іконка (значок) додатка, за допомогою якого обробляється цей об'єкт
	Розрив сторінки	Застосовується з метою вставлення розривів у багатосторінковій формі
	Набір вкладок	Служить для створення у формі декількох вкладок (або сторінок), на яких можуть розташовуватися інші елементи
	Підлегла форма/звіт	Дозволяє додати до форми іншу форму або звіт. Водночас додається форма, що називається «вкладеною» або підлеглою
	Лінія	Призначена для відображення у формі розділових ліній між іншими елементами, розділяючи їх у такий спосіб на логічні категорії
	Прямокутник	Служить для додавання до форми прямокутників з метою поліпшення сприйняття інформації, відображуваної на формі
	Інші елементи	Використовується для відображення повного списку елементів ActiveX, які можуть бути додані до форми

Слід зазначити, що на панелі елементів є також дві кнопки (рис. 5.10), що визначають режим роботи з формою:

 – *Вибір об'єктів*. Якщо кнопка перебуває в натиснутому стані, то це означає, що жоден елемент керування на панелі не обраний, і можна виконувати зміни об'єктів, які вже є на формі. З іншого боку, якщо вибрати на панелі будь-який елемент керування, то кнопка набуде вигляду «відтиснутої».

Це буде означати, що обрано об'єкт (тобто елемент керування) для додавання на форму. Щоб скасувати вибір елемента, потрібно знову натиснути кнопку *Вибір об'єктів*;

 – *Майстри*. Натиснутий стан кнопки означає, що в цей момент встановлено режим автоматичного використання майстрів, які будуть запускатися у випадку створення нового об'єкта на формі. Рекомендується в процесі розроблення макета форми тримати цю кнопку в натиснутому стані.

Щоб додати будь-який елемент керування до форми (наприклад напис), потрібно виконати таку послідовність дій:

1. Вибрати кнопку *Напис* на панелі елементів.
2. Помістити покажчик миші на форму. Водночас замість стрілки як покажчик з'явиться хрестик з буквою *A*, що зображує напис.
3. Натиснути ліву кнопку й перетягнути покажчик миші так, щоб зображення напису досягло необхідних розмірів, після чого відпустити кнопку миші.
4. У полі, що з'явилося, ввести напис, після чого натиснути лівою кнопкою миші за її межами.

У результаті на формі буде відображено зазначений напис. На рис. 5.11 наведено виділений напис *Прізвище*.

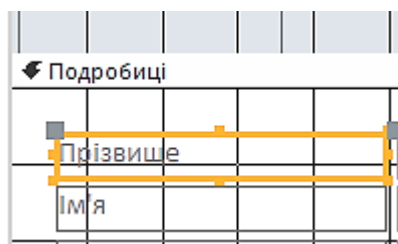



Рис. 5.11. Виділений напис *Прізвище*

Щоб змінити розміри об'єкта, наприклад напису, потрібно цей об'єкт виділити, потім установити покажчик миші на його границю, виділену чорними квадратами, щоб з'явилася подвійна стрілка. Після цього перетягнути границю, збільшивши або зменшивши об'єкт до необхідних розмірів.

Для переміщення об'єкта потрібно його виділити, потім установити на великому чорному квадраті в лівому верхньому куті покажчик миші так, щоб з'явилося зображення розкритої долоні, після чого можна перетягнути об'єкт на потрібне місце.

Якщо об'єкт потрібно видалити, то його спочатку виділяють, і після цього можна скористатися клавішею **Delete** [1].

Кожен елемент керування являє собою об'єкт з набором властивостей. Щоб змінити ті або інші властивості виділеного об'єкта, наприклад напису *Прізвище*, потрібно скористатися командою *Вид | Властивості* або кнопкою *Властивості* , у результаті чого з'явиться вікно властивостей цього напису (рис. 5.12).

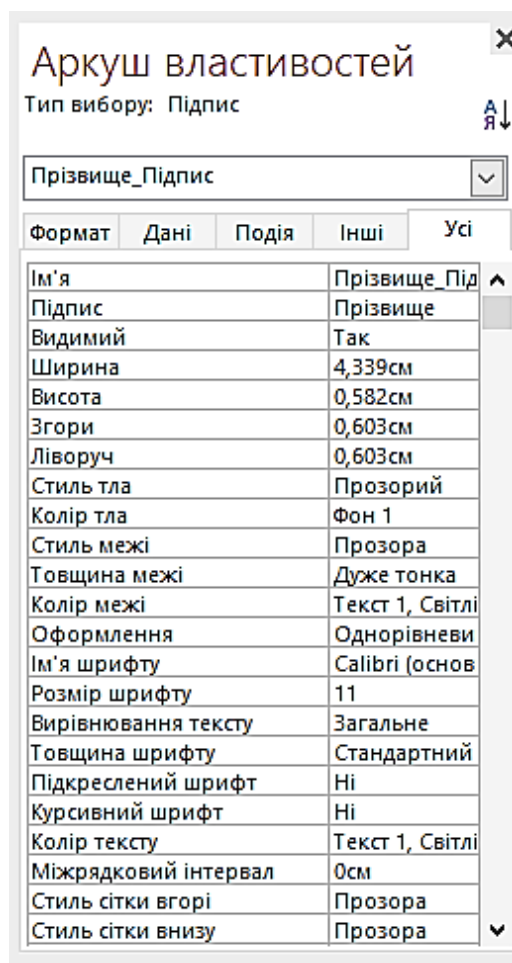


Рис. 5.12. Властивості напису *Прізвище*

Слід зазначити, що кожен об'єкт на формі має ім'я, використовуване при створенні додатків на VBA. У розглянутому випадку для напису було обране ім'я **НадписПрізвище**.

Властивість *Підпис* містить значення, що відображується у вигляді напису на формі (у цьому випадку зазначене значення – *Прізвище*).

Для об'єкта може також бути визначено властивість виведення на екран у формі (властивість *Виведення на екран*), а також режим виведення (властивість *Режим виведення*), у якому можна визначити, коли буде відображатися цей напис: *Завжди*, *Тільки при друці* або *Тільки на екран* [1].

Для оформлення зовнішнього вигляду об'єкта є ряд властивостей, що впливають на ті або інші особливості його відображення на формі: *Ширина*, *Висота*, параметри тла, параметри границі, а також властивості шрифту.

У вікні властивостей можуть відображатися відразу всі властивості обраного об'єкта або тільки з певної категорії. Для вибору того або іншого режиму відображення властивостей використовуються вкладки *Макет*, *Дані*, *Події*, *Інші* та *Всі* (рис. 5.13).

Якщо на форму додається поле **аб1**, то для нього автоматично «приєднується» пояснювальний напис. Наприклад, потрібно відобразити у формі вміст поля **Прізвище** таблиці **Студенти**. Для цього спочатку додати на форму нове поле (рис. 5.13).

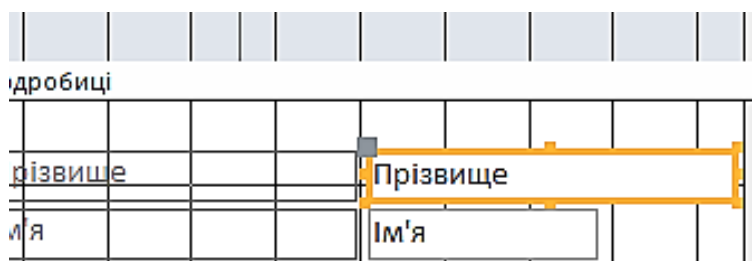



Рис. 5.13. Нове поле у формі


Потім змінити властивості поля і приєднаного напису. Для властивості *Ім'я* можна встановити значення *НаписПрізвище*, а для властивості *Підпис* – значення **Прізвище**. Після цього потрібно закрити вікно властивостей, у результаті чого приєднаний напис буде мати вигляд, як показано на рис. 5.14.



Рис. 5.14. Приєднаний до поля напис

Останнім етапом у створенні поля буде зміна його властивостей. Для властивості *Ім'я* можна вказати значення *ПолеПрізвище*. Досить важливою властивістю для об'єкта **Поле** є властивість *Дані* (рис. 5.15), у якому вибирається джерело даних для цього поля.

За допомогою кнопки  можна відобразити вікно *Будівника виразів*, у якому потрібно вказати вираз для виділеного поля.

За допомогою кнопки  у списку, що розкривається, можна вибрати поле таблиці або запиту, що буде служити джерелом даних для об'єкта **Поле**. Слід зазначити, що в цьому списку відображуються тільки таблиці і запити, встановлені у властивості *Джерело записів* для форми, тому в нашому випадку список, що розкривається, буде порожній [1].

Оскільки як джерело даних для поля **Поле Прізвище** потрібно вибрати таблицю **Студенти**, то додаємо цю таблицю до властивості *Джерело записів* форми. Щоб вибрати той або інший об'єкт для зміни його властивостей, наприклад поле, напис або форму, можна або натиснути на потрібному об'єкті, або скористатися кнопкою *Об'єкт* панелі інструментів *Формат (форма/звіт)* (рис. 5.16), після чого вибрати зі списку потрібний об'єкт (у нашому випадку – об'єкт **Форма**).

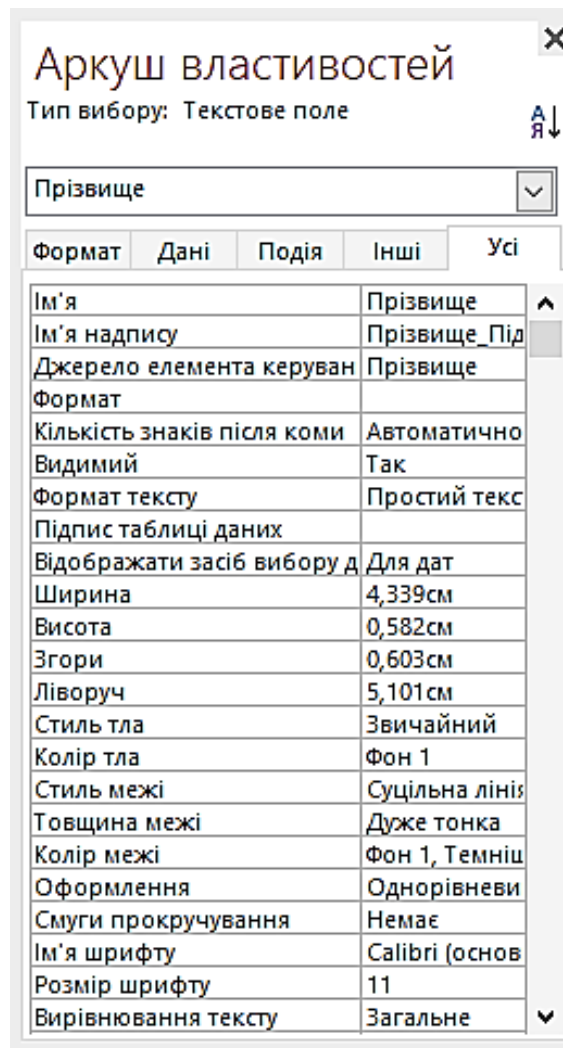


Рис. 5.15. Властивості поля **Поле Прізвище**

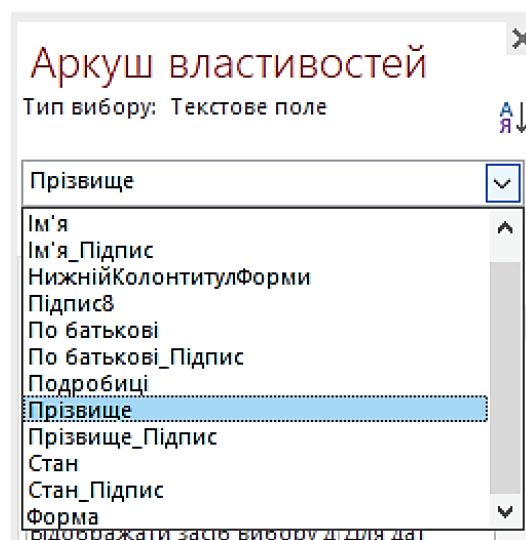




Рис. 5.16. *Кнопка Об'єкт*

Потім для властивості *Джерело записів* (вкладка *Дані* або *Усі*) вибрати таблицю **Студенти**. Для цього в рядку цієї властивості скористатися кнопкою  і вибрати зі списку всіх таблиць і запитів БД **Освіта** таблицю **Студенти**. З іншого боку, якщо потрібно було вибрати кілька таблиць для властивості *Джерело записів*, то скористатися кнопкою , після чого створити запит відповідної структури, додавши в нього потрібні поля необхідних таблиць.


Після встановлення властивості *Джерело записів* форми для властивості *Дані* об'єкта **ПолеПрізвище** вибрати зі списку, що розкривається, поле **Прізвище**. У результаті об'єкти форми, тобто напис і поле, будуть мати такий вигляд, як на рис. 5.17.



Рис. 5.17. Об'єкти форми

Потім так само можна додати на форму об'єкти **ПолеІм'я**, **ПолеПоБатькові** і **ПолеСтан**, вибравши для кожного з них як джерело даних (властивість *Дані*) поля **Ім'я**, **ПоБатькові** і **Стан** відповідно.

На наступному етапі створення цієї форми можна в заголовку форми відобразити напис **Дані про студентів**. Щоб одержати доступ до областей форми **Заголовок** і **Примітка**, потрібно виконати команду *Вид / Заголовок/Примітка форми*. Після цього можна в область заголовка додати напис, указавши для неї властивості *Підпис* значення **Дані про студентів**, а також установивши параметри шрифту і границі.


Отриману форму потрібно зберегти під ім'ям **Студенти**. Для її перегляду користуватися кнопкою **Вид** , після чого на екрані з'явиться створена форма **Студенти**, що відображує значення записів у таблиці.

5.4. Підлеглі форми

Форма з попереднього прикладу мала досить просту структуру, тобто джерелом даних для неї служила одна таблиця **Студенти**. У тому випадку, коли потрібно відображувати і змінювати дані відразу в декількох таблицях, використати підлеглі форми [1].

Наприклад, потрібно створити форму, у якій для кожного студента (таблиця **Студенти**) виводилися б назви предметів (таблиця **Предмети**), отримані оцінки й дата складання (таблиця **Оцінки**). Для реалізації описаного завдання можна створити форму такої структури: вмістом головної форми буде інформація про студентів, а в підлеглій формі будуть відображатися інші дані, що містяться в таблиці **Оцінки**.

Першим кроком у створенні описаної складної форми буде створення нової форми **ОцінкиСтудентів** або використання вже наявної форми **Студенти**, яку можна зберегти під новим ім'ям – **ОцінкиСтудентів**. Якщо форма створюється за допомогою майстра, то як джерело даних потрібно вибрати таблицю **Студенти**, з якої вибрати поля **Прізвище**, **Ім'я**, **ПоБатькові** і **Стан**.

На наступному етапі потрібно додати на створену форму об'єкт **Підлегла форма** за допомогою кнопки *Підлегла форма/звіт*, у результаті чого виконуватиметься майстер підлеглих форм, якщо попередньо *кнопка*  була встановлена в натиснутий стан [1].

На початку роботи майстра буде відображено вікно, у якому потрібно вибрати джерело даних для підлеглої форми: *таблиці і запити* або *форми*, які є в БД. Вибираємо пункт *Наявні таблиці і запити* і тиснемо *Далі*>.

Другий етап припускає вибір полів з однієї або декількох таблиць, які потрібно включити до підлеглої форми. У розглянутому прикладі потрібно з таблиці **Оцінки** вибрати поля **КодПредмета**, **Оцінка** і **ДатаСкладання**. Слід зазначити, що замість назв предметів до форми були додані тільки їхні

коди (поле **КодПредмета**). Однак у режимі форми замість кодів будуть відображатися назви предметів, тому що для поля **КодПредмета** таблиці **Оцінки** була використана підстановка поля **Назва** з таблиці **Предмети**.

На наступному кроці роботи майстра потрібно визначити поле зв'язку між головною і підлеглою формами. У цьому випадку майстер автоматично визначив існуючий зв'язок за полем **КодСтудента** між таблицями **Студенти** і **Оцінки**. Отже, потрібно в режимі **Вибір зі списку** вибрати зв'язок **'Оцінки'** для кожного запису **'Студенти'**, поле зв'язку **'КодСтудента'** (рис. 5.18).

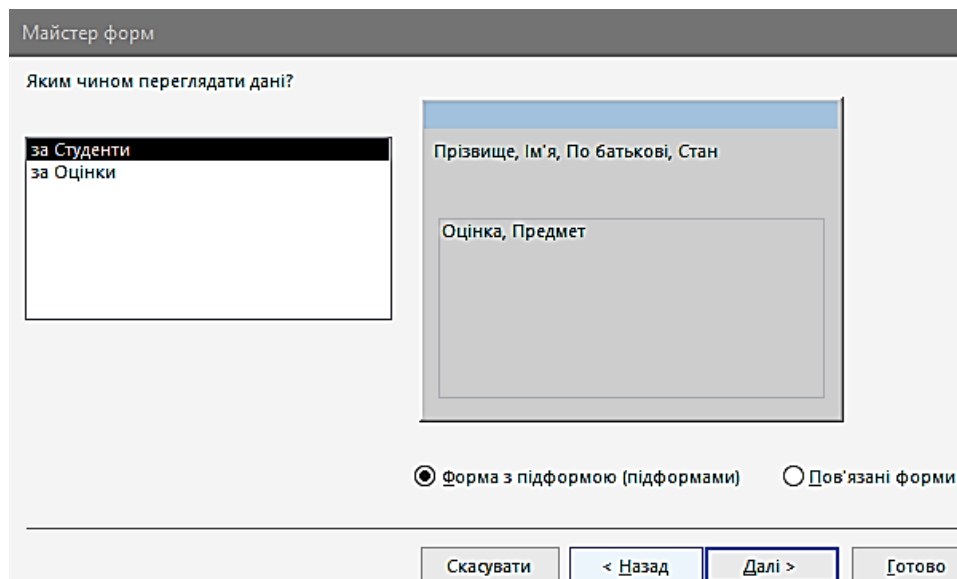


Рис. 5.18. Майстер підлеглих форм. Визначення зв'язків

На останньому етапі потрібно вказати ім'я для підлеглої форми, наприклад **підлегла форма Оцінки**, і натиснути **Готово**.

У результаті отримано основу складної форми **Оцінки Студентів** з використанням підлеглої форми. Для завершення створення форми можна в область заголовка додати напис *Оцінки за предмети*, а також зробити підписи для даних: *Студенти* і *Оцінки*. Крім того, можна додати розділову лінію (об'єкт **Лінія**) між даними головної і підлеглої форм.

Слід зазначити, що для відображення назв предметів було автоматично створене поле зі списком (рис. 5.19), водночас у його властивості *Дані* отримується значення **Код Предмета**.

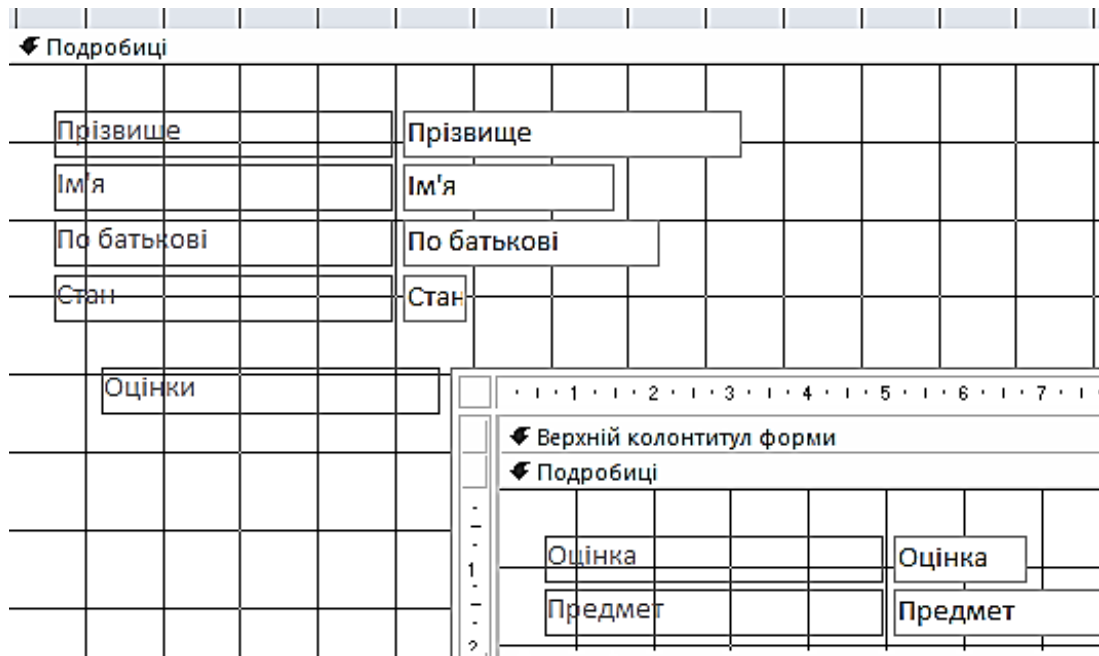


Рис. 5.19. Підлегла форма Оцінки

З іншого боку, у властивості *Джерело рядків* цього об'єкта перебуває запит, що містить два поля з таблиці **Предмети: КодПредмета і Назва**. Відображення у формі тільки назв предметів було досягнуто автоматичним установленням нижченаведених властивостей поля на вкладці *Макет* (рис. 5.20).

Кількість стовпчиків – 2.

Ширина стовпчиків – 0 см; 2,54 см.

Предмети

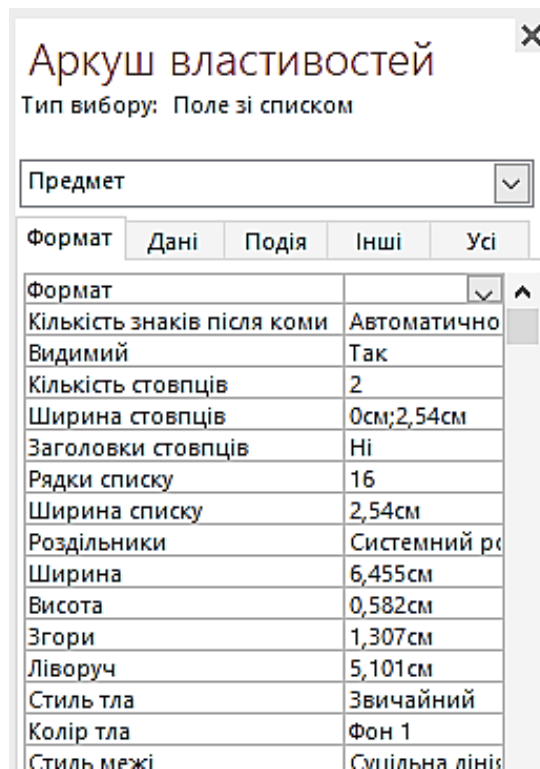



Рис. 5.20. Властивості поля зі списком *КодПредмета*. Вкладка *Макет*

5.5. Використання кнопок

В Access існує можливість виконання тих або інших команд обробки записів, форм, звітів тощо за допомогою додавання на форму спеціального об'єкта – *кнопки*. Натискання на цю кнопку в режимі форми призводить до виконання певної дії.

Наприклад, у формі **ОцінкиСтудентів** можна додавати оцінки студентам, але не можна додавати нові предмети, тому що для поля зі списком установлена властивість *Обмежитися списком* (вкладка *Дані*). Для додавання нових предметів до таблиці **Предмети** можна створити на формі кнопку, при натисканні на яку будуть відображуватися записи таблиці. Отже, можна буде вносити додавання до списку предметів.

При додаванні на форму об'єкта **Кнопка** буде запущений майстер створення кнопок, якщо *кнопка*  перебуває в натиснутому стані.

На першому етапі його роботи потрібно вибрати категорію *Робота з формою*, у списку дій якої вибрати *Відкрити форму* (рис. 5.21), щоб при натисканні на кнопку виконувалося відкриття певної форми.

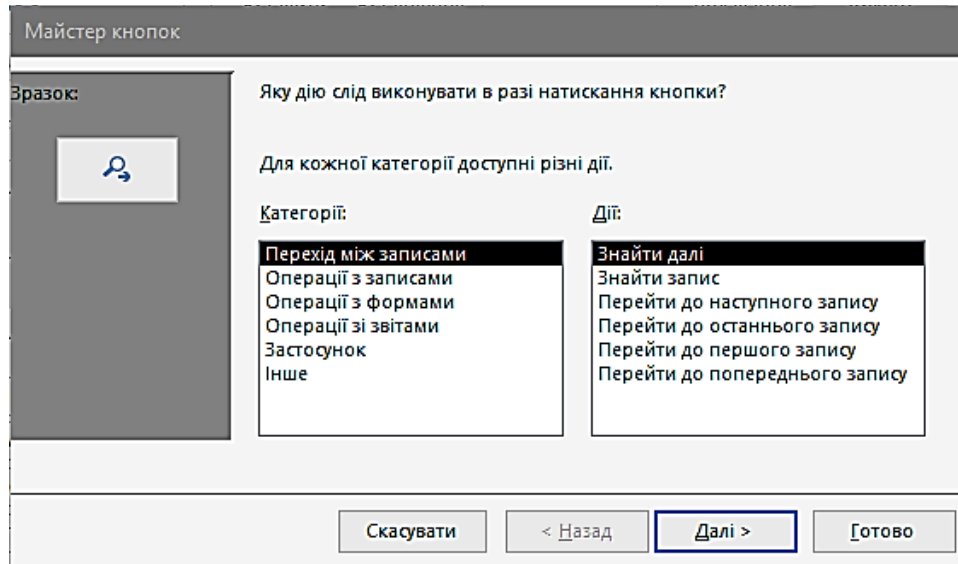


Рис. 5.21. Створення кнопок. Вибір дії

Слід зазначити, що для кнопки можна вибрати дію, що міститься в одній з таких категорій [1]:

- *Переходи по записах* – містить дії, пов'язані з переміщенням по записах таблиці (*Перший запис*, *Останній запис* тощо);
- *Обробка записів* – пропонує набір дій, пов'язаних з обробкою записів (*Додати запис*, *Видалити запис* тощо);
- *Робота з формою* – містить список дій, використовуваних при роботі з формою (*Відкрити форму*, *Закрити форму*, *Друк* тощо);
- *Робота зі звітом* – пропонує перелік дій для обробки звітів (*Друк звіту*, *Перегляд звіту* тощо);
- *Додаток* – містить набір дій для керування додатком (*Виконати додаток*, *Вийти з додатка* тощо);
- *Різне* – містить корисні дії, які не стосуються жодної з описаних вище категорій (*Виконати запит*, *Друк таблиці* тощо).

Другий етап припускає вибір таблиці, що має бути відкрита. У Нашому випадку потрібно вибрати таблицю **Предмети**.

На наступному етапі можна встановити режим, при якому у формі будуть відображатися тільки ті записи, що задовольняють певні умови. Тому, якщо нема потреби в установленні обмежень на список предметів, потрібно вибрати режим **Відкрити форму і показати всі записи**.

Черговий етап припускає вибір тексту або рисунка, який потрібно розмістити на кнопці. У цьому прикладі можна в полі **Текст** вказати підпис **Предмети** (рис. 5.22).

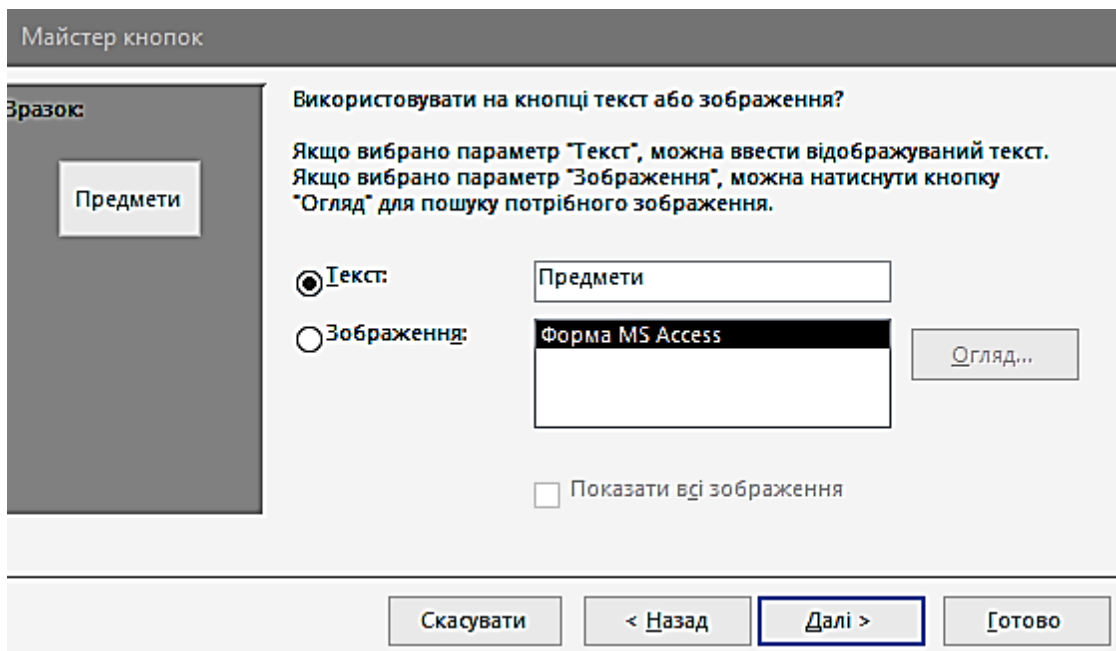


Рис. 5.22. Створення кнопок. Встановлення підпису

На останньому етапі потрібно вибрати **Ім'я** для об'єкта **Кнопка**, наприклад **Предмети**.

У результаті всіх виконаних дій на формі **ОцінкиСтудентів** буде створена *кнопка Предмети*. Натискання на цю кнопку в режимі **Форми** призведе до відображення форми **Предмети**, у якій буде виведено список наявних предметів з можливістю додавання нових назв.

Можна також для кнопки *Предмети* вказати текст підказки, що спливає. Для цього потрібно в режимі конструктора форми **ОцінкиСтудентів** виділити кнопку і встановити для неї у властивості **Спливаюча підказка** значення *Список предметів*.

Отримана в результаті всіх виконаних дій форма **ОцінкиСтудентів** наведена на рис. 5.23.

The screenshot shows a web form titled "Студенти1". It contains several input fields: "Прізвище" (Last name) with the value "Жукова", "Ім'я" (First name) with "Марина", and "По батькові" (Patronymic) with "Володимирівна". There is a "Стан" (Status) field which is empty. A blue button labeled "Предмети" is positioned to the right of the name fields. Below these fields is a table with two columns: "Оцінка" (Grade) and "Предмет" (Subject). The table has three rows of data: the first row shows a grade of "5" for "Фізика" (Physics), the second row shows a grade of "5" for "Хімія" (Chemistry), and the third row shows a grade of "0" for an empty subject. A "*" symbol is visible in the first column of the third row. The table has dropdown arrows in the header cells.

Рис. 5.23. Форма *ОцінкиСтудентів*

5.6. Умовне форматування

Часто при відображенні даних у формі буває зручно тим або іншим способом виділяти значення в полях залежно від виконання певних умов. Такі дії називають *умовним форматуванням* [1].

Наприклад, потрібно залежно від оцінки, яку одержав студент, виводити її значення шрифтом установленого формату:

- якщо «5», то жирним, чорного кольору;
- якщо «4», то жирним, сірого кольору;
- якщо «3» або «2», то жирним, з підкресленням, чорного кольору.

А також якщо за предмет отримана оцінка «3» або «2», то назва предмета має відображатися жирним підкресленим шрифтом.

Спочатку потрібно встановити умовне форматування поля **Оцінка**. Для цього його виділити, перебуваючи в режимі **Конструктор**, потім виконати команду *Формат | Умовне форматування*. Далі в діалоговому вікні в рядку *Умова 1* потрібно встановити параметрам такі значення: *Значення поля, дорівнює, 5*. Після цього вибрати необхідні параметри для шрифту, яким буде відображена оцінка (жирний, чорного кольору).

Щоб указати наступну умову і відповідний формат, потрібно натиснути у вікні *Умовне форматування* кнопку *Додати>>*. З іншого боку, щоб видалити будь-яку з установлених умов, потрібно скористатися кнопкою *Видалити* і вибрати непотрібні умови. Слід зазначити, що для одного поля можна встановити до трьох умов.

Після того як у вікні *Умовне форматування* будуть зазначені всі умови і формати (рис. 5.24), натиснути кнопку *ОК*.

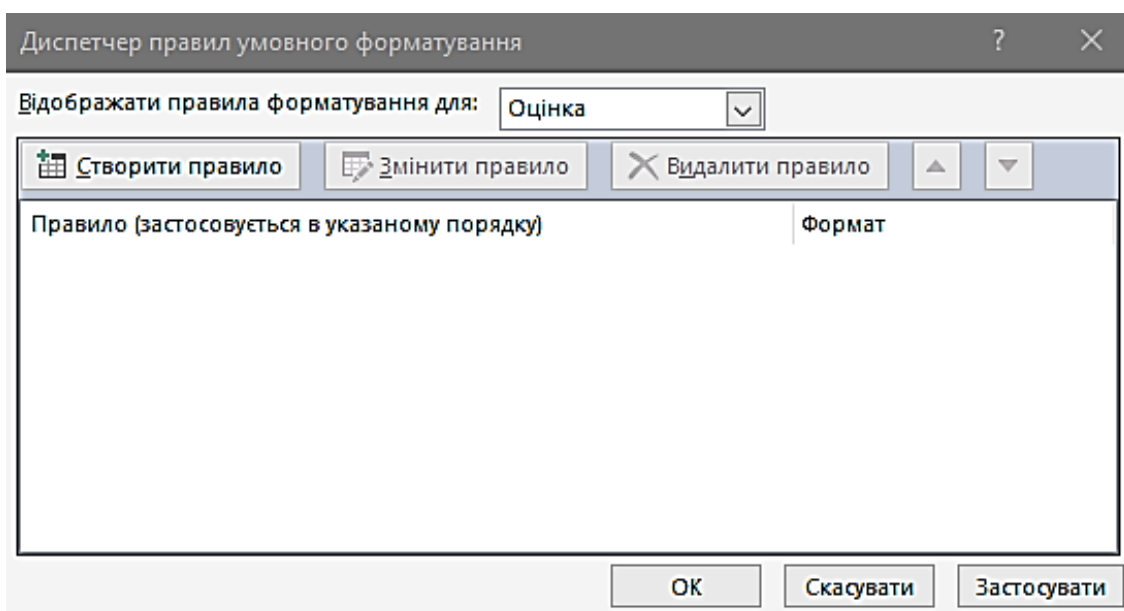


Рис. 5.24. Діалогове вікно *Умовне форматування*

При установленні формату для поля зі списком предметів потрібно вибрати параметр *Вираз* і вказати умову

$$[Оцінка]=3 \text{ Or } [Оцінка]=2,$$

після чого встановити необхідний формат шрифту (рис. 5.25).

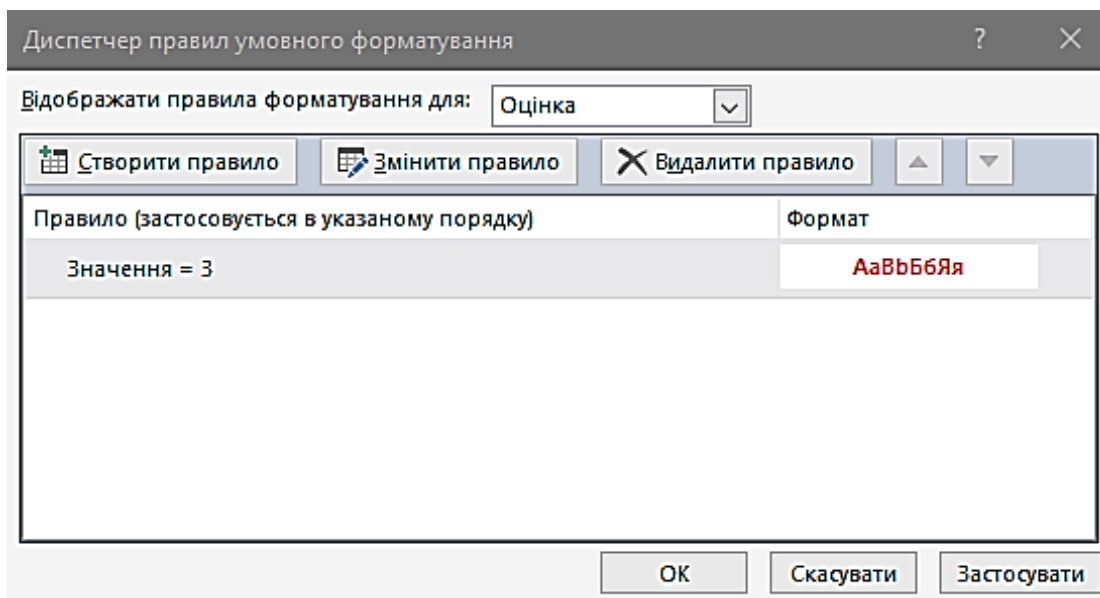


Рис. 5.25. Умове форматування для поля зі списком предметів

Результати використання умовного форматування можна побачити в режимі **Форми**. Зокрема на рис. 5.26 наведено оцінки і назви предметів для студента Петрова, водночас дані відображуються із урахуванням умовного форматування.

Стан

Оцінки

Оцінка	Предмет
3	Філософія
4	Економіка
5	Математика
*	0

Рис. 5.26. Результати умовного форматування

Контрольні запитання до розділу 5

1. Що має на меті розробник, створюючи форму в БД?
2. Чим є форми в Access при розробленні користувальницького інтерфейсу?
3. Що потрібно виконати, щоб створити форму?
4. Що передбачає другий етап створення форми?
5. Яким режимом можна скористатися у випадку, коли джерелом даних для форми є одна таблиця або запит?
6. Що потрібно використати, коли потрібно відображувати і змінювати дані відразу в декількох таблицях?
7. Чи використовується в Access можливість виконання тих або інших команд за допомогою додавання на форму спеціального об'єкта – **Кнопки**?
8. Вказати категорії дій для кнопки, які можна вибрати.

Розділ 6. ЗВІТИ

6.1. Базові відомості про звіти

Щоб подати у друкованому вигляді інформацію з БД (таблиць або запитів), в Access використовуються *звіти*, що мають порівняно з іншими способами друку (з таблиць або форм) такі переваги [1]:

1. Крім відображення вмісту полів з таблиць або запитів, у звітах також можна обчислювати різні підсумкові значення, а також групувати дані.

2. За допомогою звітів можна барвисто оформляти різноманітні документи, необхідні в тих або інших сферах професійної діяльності.

Для відображення основних даних і обчислених підсумкових значень у звітах можна існують такі можливості:

1. Здійснення обчислень будь-якої складності з інформацією, що міститься не тільки в одній, але також і в різних групах даних.

2. Використання ієрархічної структури у відображенні груп даних (у структурі може перебувати до 10 рівнів).

3. Друк верхнього і нижнього колонтитулів на кожній сторінці звіту.

4. Для звіту можна встановити текст заголовка і приміток.

5. У кожній групі даних можуть перебувати власні заголовки і примітки.

Крім перерахованих можливостей, у звітах (аналогічно створенню форм) також застосовуються впроваджені рисунки, об'єкти і підлеглі звіти.

6.2. Створення звіту

Як приклад можна розглянути створення звіту **Оцінки** (рис. 6.1), у якому для кожного студента відображуються його оцінки з усіх предметів, а також його середній бал.

СтудентиЗвіт

Прізвище	Ім'я	По батькові	Дата здачі за Місяць	Предмет	Оцінка ,ачі
Жукова	Марина	Володимирів	Май 2021		
				Фізика	5 ###
				Хімія	5 ###
Зведення для 'Дата здачі' = 11.05.2021 (2 записи з відомостями)					
			Середнє		5
Зведення для 'Прізвище' = Жукова (2 записи з відомостями)					
			Середнє		5
Ігнатов	Максим	Валерійович	Май 2021		
				Фізика	5 ###
				Хімія	4 ###
Зведення для 'Дата здачі' = 11.05.2021 (2 записи з відомостями)					
			Середнє		4,5

Рис. 6.1. Звіт Оцінки

Якщо як вихідні дані для звіту використовуються таблиці або запити, то потрібно створити запит на вибірку **Оцінки**, у якому мають відображатися поля **Прізвище**, **Ім'я** та **По батькові** таблиці **Студенти**, поле **Назва** таблиці **Предмети**, а також поля **Оцінка** і **ДатаСкладання** таблиці **Оцінки**. У результаті запуску цього запиту буде отриманий набір записів, наведений на рис. 6.2.

Прізвище	Ім'я	По батькові	Предмет	Оцінка	Дата здачі
Жукова	Марина	Володимирів	Фізика	5	11.05.2021
Жукова	Марина	Володимирів	Хімія	5	11.05.2021
Морозов	Ігор	Олегович	Фізика	4	11.05.2021
Морозов	Ігор	Олегович	Хімія	4	21.05.2021
Петров	Михайло	Олександров	Філософія	3	11.05.2021
Петров	Михайло	Олександров	Економіка	4	13.05.2021
Петров	Михайло	Олександров	Математика	5	13.05.2021
Ігнатов	Максим	Валерійович	Фізика	5	10.05.2021
Ігнатов	Максим	Валерійович	Хімія	4	11.05.2021
*					

Рис. 6.2. Запит Оцінки

Аналогічно створенню форм створювати звіти можна як за допомогою майстра, так і конструктора. Як правило, майстер використовується для створення основної частини звіту, тобто структури, тоді як конструктор застосовується для вдосконалення зовнішнього вигляду звіту [1].

Щоб створити звіт, потрібно у вікні БД у розділі **Звіти** скористатися кнопкою *Створити* і вибрати необхідний режим створення нового звіту.

Майстер звітів. На першому етапі роботи майстра потрібно вибрати у списку, що випадає, *Таблиці і запити* запит **Оцінки** і додати з нього всі поля до списку *Обрані поля* (рис. 6.3). Після цього натиснути *Далі*>.

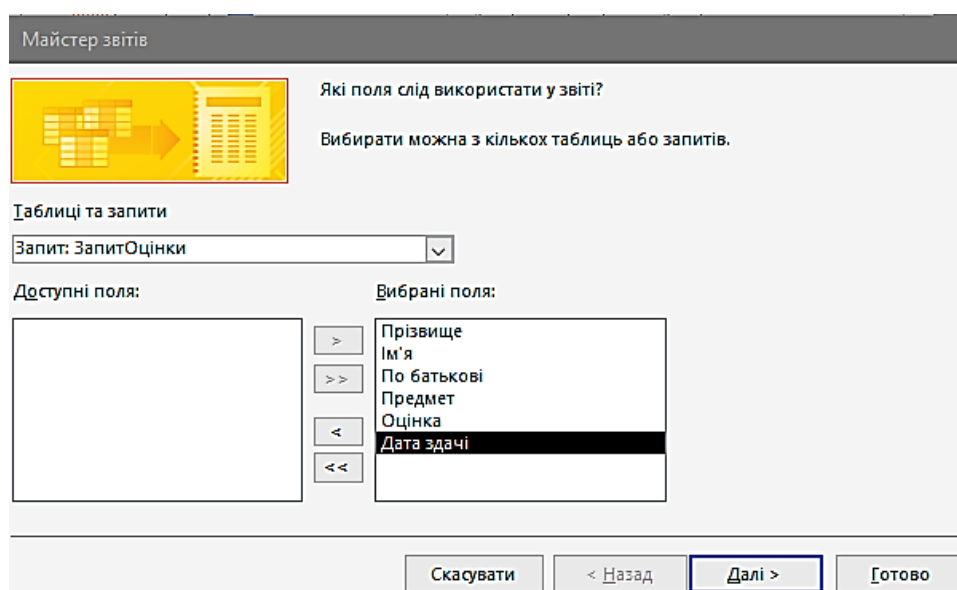


Рис. 6.3. Створення звіту. Вибір полів

Потім потрібно вибрати вид подання даних, тобто розбиті на групи дані у звіті. У розглянутому прикладі потрібно для кожного студента відображувати список оцінок. Ієрархія в цьому випадку буде такою: **Студенти**→**Предмети**, **Оцінка** і **Дата**, тобто потрібно у списку вибрати пункт *Студенти* (рис. 6.4), після цього натиснути *Далі*>.

На наступному етапі, якщо буде потреба, можна додати рівні групування відображуваних даних, наприклад оцінки, одержані студентом у той або інший день, тобто додати рівень **ДатаСкладання**. Після цього

стане доступною кнопка *Групування*, що дасть змогу встановити інтервали групування для полів, включених до ієрархічної структури для створення нових рівнів групування. Наприклад, якщо додати рівень групування **ДатаСкладання**, то натискання кнопки *Групування* дасть змогу вибрати у діалоговому вікні *Інтервали групування* за допомогою списку, що розкривається, один з наявних інтервалів, наприклад за *місяцями* (рис. 6.5).

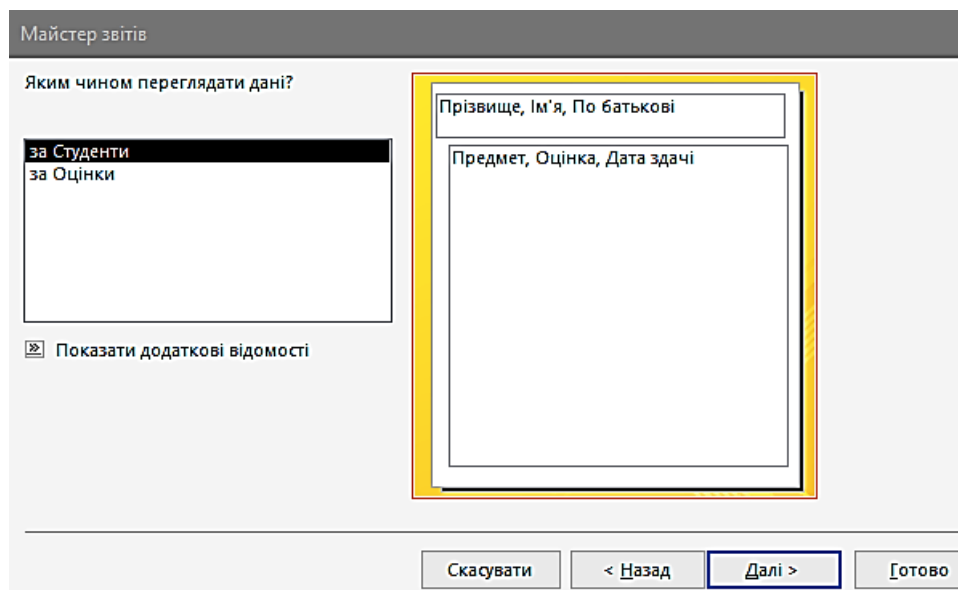


Рис. 6.4. Створення звіту. Вид подання даних

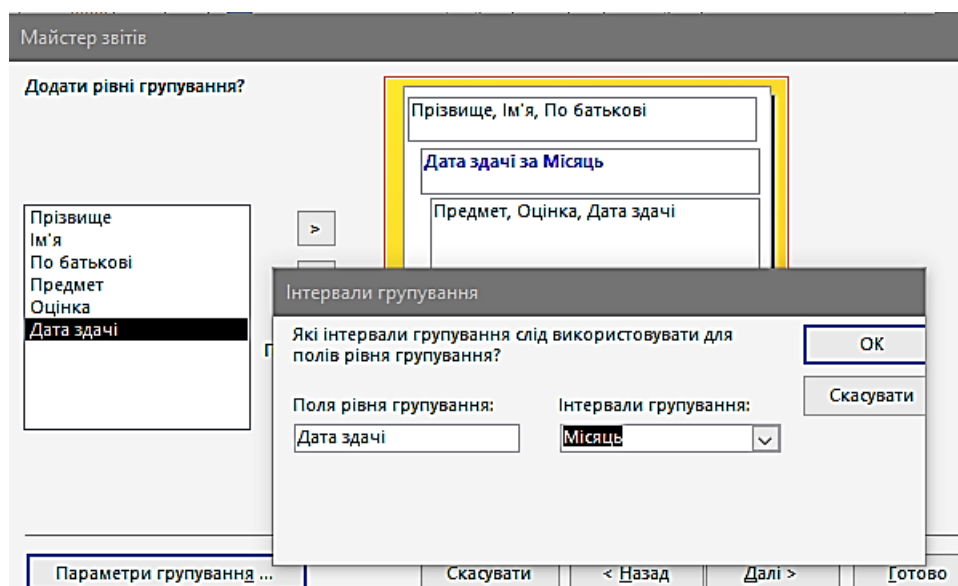


Рис. 6.5. Створення звіту. Інтервали групування

У розглянутому прикладі можна не додавати рівні групування, а відразу натиснути *Далі*>.

Черговий етап припускає визначення порядку сортування записів, що відображуються у звіті (рис. 6.6), а також підсумкові обчислення, виконання яких можна встановити за допомогою кнопки *Підсумки*. У діалоговому вікні *Підсумки*, який викликається натисканням відповідної кнопки (рис. 6.7), можна обчислити суму (*Sum*), середнє значення (*Avg*), а також мінімум (*Min*) або максимум (*Max*) для даних, що містяться в зазначеному полі.

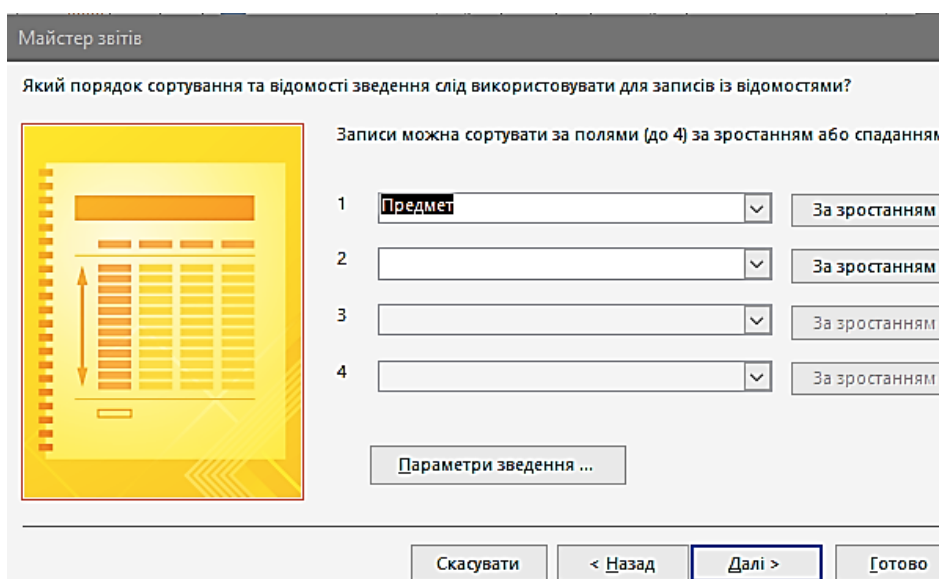


Рис. 6.6. Створення звіту. Порядок сортування

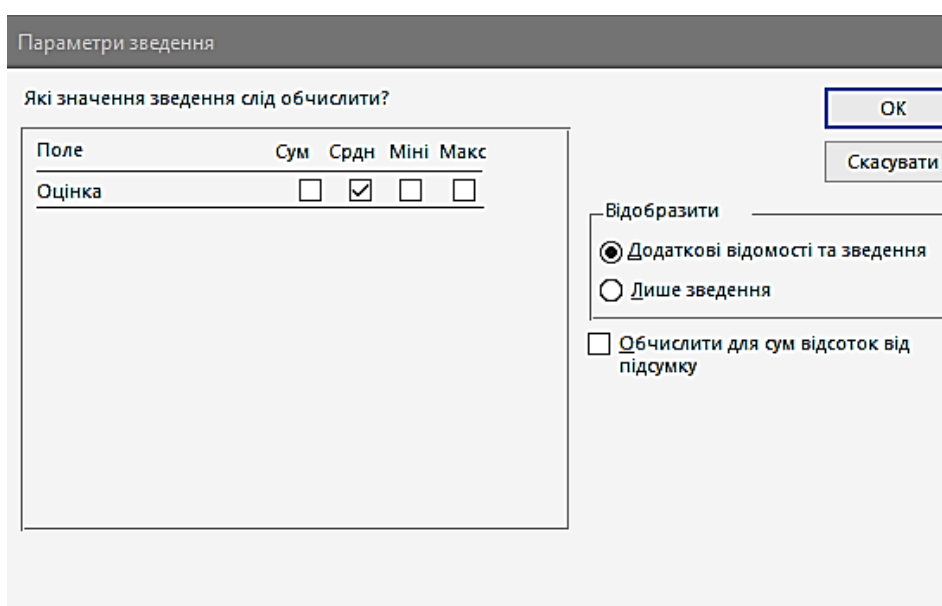


Рис. 6.7. Створення звіту. Підсумки

Для поля **Назва** можна встановити сортування за алфавітом, тобто за зростанням, а у вікні *Підсумки* потрібно вибрати обчислення середнього бала (функція *Avg*) [1].

На наступному етапі можна вибрати вид макета, що буде використовуватися у звіті, а також орієнтацію сторінки (*книжкова* або *альбомна*). У списку *Макет* можна вибрати *східчастий* і встановити книжкову орієнтацію (рис. 6.8).

Черговий крок роботи майстра дає змогу вибрати стиль для оформлення звіту, аналогічний тому, як визначається зовнішній вигляд при створенні форм. У нашому випадку можна вибрати стиль *Діловий* і натиснути *Далі* > [1].

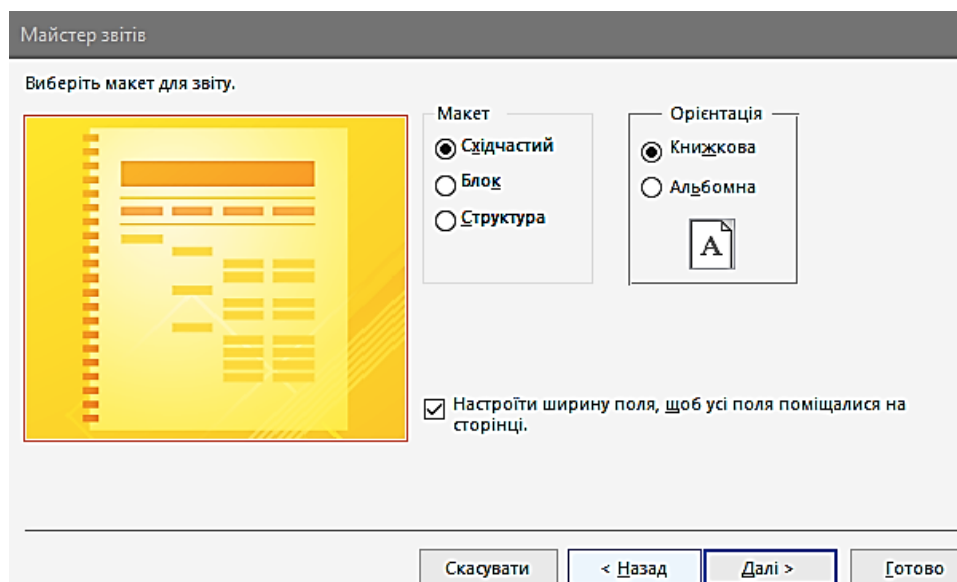



Рис. 6.8. Створення звіту. Вид макета

На останньому етапі потрібно вказати ім'я створеного звіту: **Оцінки**, а також вибрати подальші дії: *Переглянути звіт* (режим **Попереднього перегляду**) або *Змінити макет звіту* (режим **Конструктор**). Слід зазначити, що перемикання між режимом **Попередній перегляд** і **Конструктор** виконується за допомогою кнопки *Вид* , використання якої аналогічне раніше розглянутим об'єктам БД, тобто таблицям, запитам і формам.

Після того як за допомогою майстра створена структура звіту, для внесення змін, пов'язаних з його оформленням, потрібно встановити для звіту **Оцінки** режим **Конструктор**.

Конструктор. У режимі конструктора будь-який звіт являє собою набір таких розділів: верхній і нижній колонтитули (друкуються відповідно вгорі і внизу кожної сторінки звіту) і область даних [1].

Водночас у звіт можна включити заголовок, що буде друкуватися тільки вгорі першої сторінки, а також примітку, що буде відображатися наприкінці останньої сторінки звіту. Слід також зазначити, що область даних може перебувати всередині однієї або декількох ієрархічних груп, кожна з яких має свій заголовок і примітку. Наприклад, у звіті **Оцінки**, створеному за допомогою майстра (рис. 6.9), є тільки одна група – *Прізвище*, для якого встановлені заголовок і примітка.

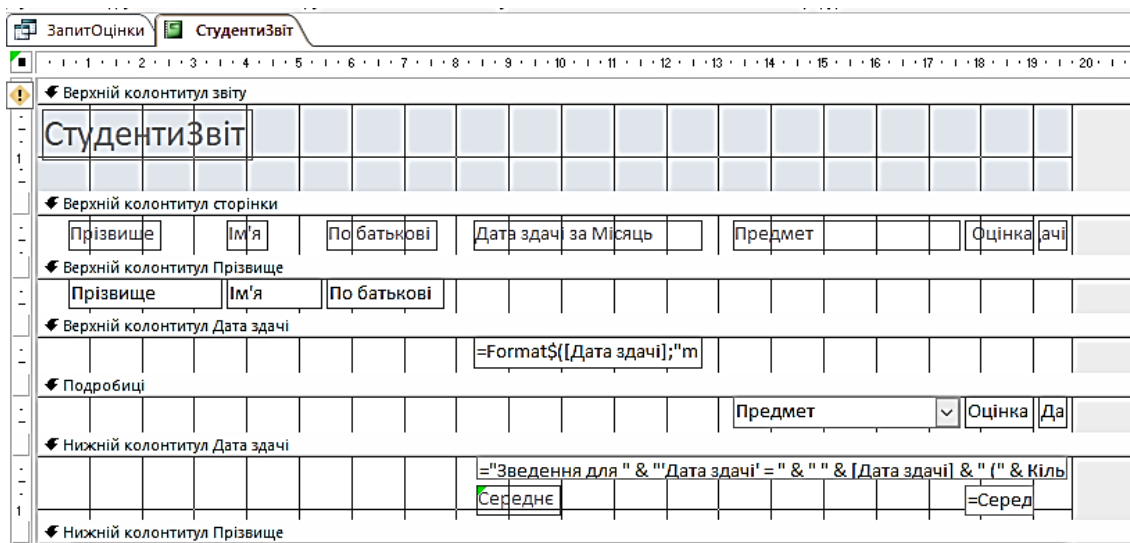


Рис. 6.9. Звіт Оцінки. Режим конструктора

Як ми вже зазначали раніше, за допомогою майстра зручно створювати звіти, а за допомогою конструктора звітів – виправляти помилки. Наприклад, можна поставити завдання внесення за допомогою конструктора ряду змін в отриманий завдяки майстрові звіт [1]:

1. Вирівняти заголовок звіту по центру сторінки.
2. Замість підпису **Назва** встановити підпис **Предмет**.
3. Змінити Підпис стовпчика з **ДатаСкладання** на **Дата складання**.
4. Видалити у розділі приміток групи *Прізвище* рядок коментарю «Підсумки для ...».

5. Замість напису *Avg*, що позначає обчислення середнього значення для оцінок студента, ввести напис *Середній бал* і розташувати його у стовпчику *Предмет*.

6. Перемістити напис із поточною датою в лівий верхній кут сторінки і видалити напис у правому нижньому куті: “*Сторінка ...*”.

7. Додати поруч із датою номер сторінки.

Щоб перемістити будь-який об'єкт, наприклад напис *Оцінки*, у межах одного розділу звіту, достатньо встановити на нього покажчик миші і, натиснувши ліву кнопку, перетягнути об'єкт у необхідне місце. Якщо ж потрібно перемістити об'єкт з одного розділу в інший, наприклад із примітки в заголовок, то виконати нижченаведену послідовність дій [1]:

1. Виділити об'єкт.
2. Вирізати його в буфер командою *Виправлення | Вирізати*.
3. Натиснути кнопкою миші в тому розділі, куди потрібно перемістити об'єкт (у нашому випадку на заголовку).
4. Вставити об'єкт з буфера командою *Виправлення | Вставити*.
5. Перемістити об'єкт на необхідне місце даного розділу (заголовка).

Зокрема описані дії потрібно виконати для напису з поточною датою. Водночас слід зазначити, що поточна дата визначається за допомогою вбудованої функції *Now* без параметрів, тобто значенням властивості *Дані* для відповідного поля в цьому випадку є значення $=Now()$. Загалом якщо в полі потрібно обчислити будь-який вираз, то для властивості *Дані* потрібно вказувати перед цим виразом знак “=”, щоб Access сприймав його не як


текст, а як значення, що обчислюється, аналогічно редактору електронних таблиць Excel [1].

Щоб змінити написи *Назва*, *ДатаСкладання* і *Avg*, потрібно вказати для кожної з них відповідні значення у властивості *Дані*.

Для відображення номерів сторінок у звіті потрібно скористатися функцією *Page*, вказавши її у властивості *Дані* для нового напису, який можна розташувати у верхньому або нижньому колонтитулі.

Крім усіх перерахованих змін, для поліпшення сприйняття цього звіту можна додати кілька ліній, що відокремлюють логічні розділи. Зокрема можна відокремити лініями всіх студентів, а також рядок із визначенням середнього бала кожного студента [1].

У результаті виконання всіх описаних дій буде отриманий звіт **Оцінки**.

На закінчення слід зазначити, що для створеного звіту можна встановлювати порядок групування і сортування даних у групах за допомогою кнопки *Сортування і групування* , при натисканні на яку відображується діалогове вікно *Сортування і групування* (рис. 6.10). У цьому вікні можна додавати, змінювати або видаляти рівні групування так само, як і при роботі в режимі таблиці, а також встановлювати порядок сортування для того або другого рівня [1].

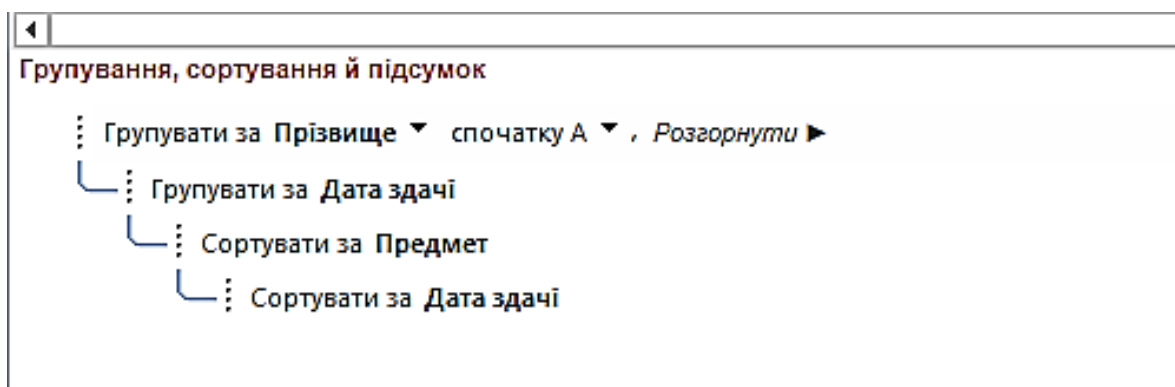


Рис. 6.10. Діалогове вікно *Сортування і групування*

Крім роботи з рівнями групування, для полів звіту можна також використати умовне форматування, аналогічно роботі з формами. Наприклад, якщо середній бал студента більший або дорівнює «4,5», то в цьому випадку потрібно відображувати значення середнього бала, а також прізвище, ім'я та по батькові студента напівжирним шрифтом.

Щоб установити умовне форматування для поля, потрібно його виділити і виконати після цього команду *Формат | Умове форматування*. Для поля з обчисленням середнього бала потрібно вказати як *Умова1* такі параметри: *Значення поля, більше або дорівнює, 4,5* і встановити після цього формат шрифту: напівжирний (рис. 6.11).

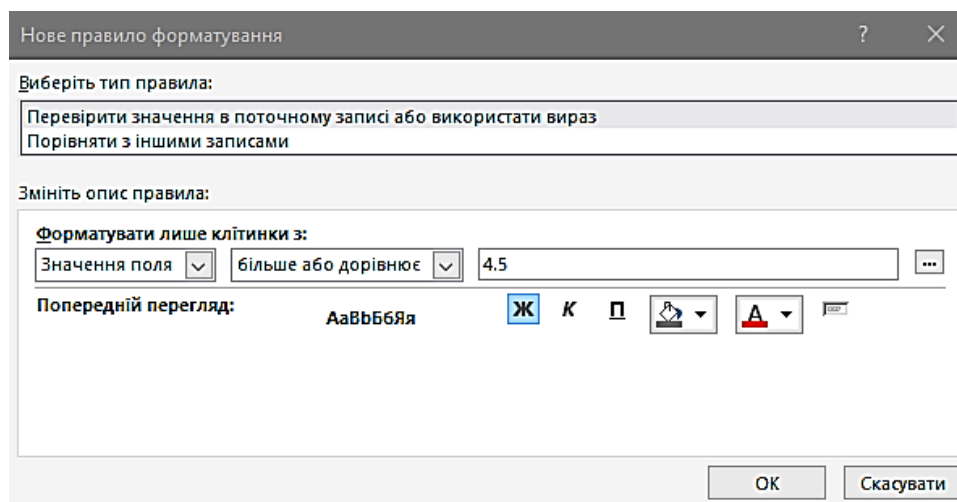


Рис. 6.11. Умове форматування для середнього бала

З іншого боку, для полів **Прізвище**, **Ім'я** та **ПоБатькові** потрібно вказати умову з використанням поля, у якому обчислюється середній бал студента. За замовчуванням цьому полю було присвоєно ім'я **Avg – оцінка**. Тому у вікні *Умове форматування* для полів **Прізвище**, **Ім'я** та **По батькові** потрібно як параметр для *Умови1* вибрати *Вираз* і вказати умову $[Avg - оцінка] \geq 4,5$. Потім для цієї умови також вибрати формат шрифту напівжирний.

На рис. 6.12 наведено звіт Оцінки з застосуванням описаного вище умовного форматування.

СтудентиЗвіт					
Прізвище	Ім'я	По батькові	Дата здачі за Місяць	Предмет	Оцінка ,ачі
Жукова	Марина	Володимиріт	Май 2021	Фізика	5 ###
				Хімія	5 ###
Зведення для 'Дата здачі' = 11.05.2021 (2 записи з відомостями)					
Середнє					5
Зведення для 'Прізвище' = Жукова (2 записи з відомостями)					
Середнє					5
Ігнатов	Максим	Валерійович	Май 2021	Фізика	5 ###
				Хімія	4 ###
Зведення для 'Дата здачі' = 11.05.2021 (2 записи з відомостями)					
Середнє					4,5

Рис. 6.12. Звіт **Оцінки** із застосуванням умовного форматування

Контрольні запитання до розділу 6

1. Для чого в Access використовуються *звіти*?
2. Які методи використовуються для створення звітів?
3. Чи передбачено у звіті визначення порядку сортування записів, що відображуються у звіті?
4. Чи передбачено у звіті вибір виду макета, що буде використовуватися у звіті?
5. Набір яких розділів у режимі конструктора складає звіт?
6. Що потрібно виконати для переміщення будь-якого об'єкта в межах одного розділу звіту?
7. Чи можна для створеного звіту встановлювати порядок групування і сортування даних у групах?
8. Що потрібно виконати для установлення умовного форматування для поля?

Розділ 7. ДРУК В ACCESS

7.1. Друк опису об'єктів БД

В Access існує можливість створення звіту з описом того або іншого об'єкта обраної БД, наприклад таблиць, запитів тощо [1].

Щоб створити звіт з описом, потрібно скористатися командою *Сервіс / Аналіз | Архіваріус*, після чого буде відображено діалогове вікно *Архіваріус* (рис. 7.1). У цьому вікні можна, відкривши будь-яку вкладку (*Таблиці*, *Запити* тощо), вибирати об'єкти, для яких потрібно роздрукувати опис. Крім того, за допомогою кнопки *Параметри* можна встановлювати, які саме властивості об'єкта будуть відображені у звіті з описом [1].

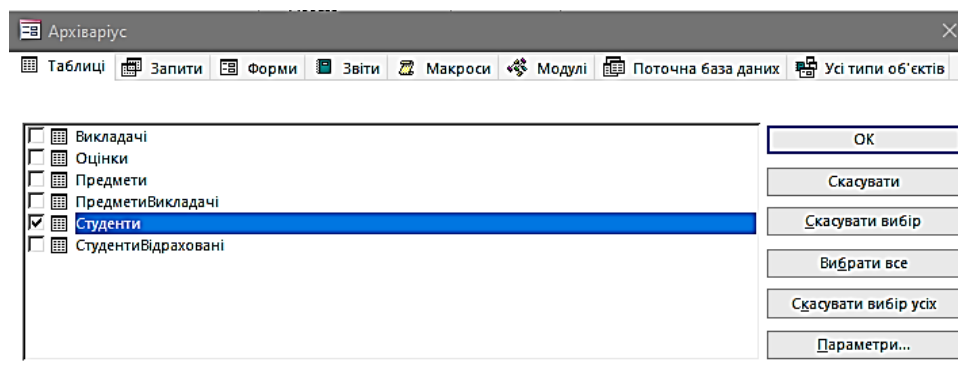


Рис. 7.1. Діалогове вікно *Архіваріус*. Вкладка *Таблиці*

Наприклад, потрібно створити звіт опису таблиці **Студенти**, до якого включити тільки імена полів, типи даних і формати. Для цього треба у вікні *Архіваріус* установити прапорець поруч із таблицею **Студенти** (вкладка *Таблиці*) і натиснути кнопку *Параметри*. Потім у діалоговому вікні *Друк опису таблиці* в розділі *Включити в опис полів* вибрати пункт *імена, типи даних і формати* (рис. 7.2) і натиснути *ОК*. Після цього, якщо буде потреба, можна вибрати ще будь-які об'єкти, для яких також вказати параметри

опису. Якщо всі потрібні об'єкти обрані, то для створення звіту з описом потрібно у вікні *Архіваріус* натиснути *OK*. У розглянутому прикладі створюється звіт з описом полів таблиці **Студенти** (рис. 7.3).

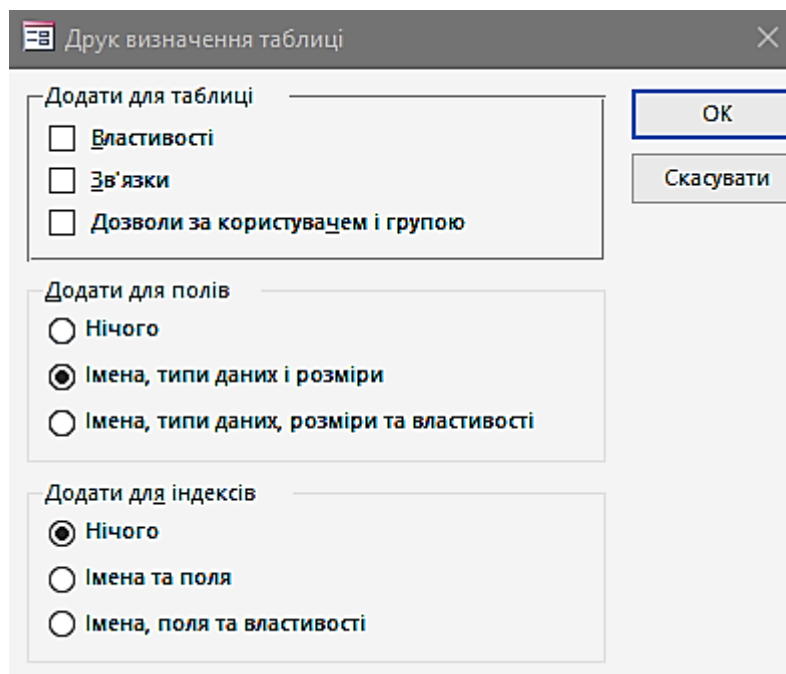


Рис. 7.2. Діалогове вікно *Друк опису таблиці*

звіт

C:\Users\Deni\Documents\Ocsbita.accdb
Таблиця: Студенти

2 июня 2021 г.
Сторінка: 1

Стовпці	Ім'я	Тип	Розмір
	КодСтудента	Довге ціле число	
	Прізвище	Короткий текст	
	Ім'я	Короткий текст	
	По батькові	Короткий текст	
	Дата народження	Дата й час	
	Група студента	Короткий текст	
	Стан	Короткий текст	

Рис. 7.3. Опис таблиці **Студенти**

7.2. Друк даних у режимі таблиці

Дані, які зберігаються в таблицях, можуть бути роздруковані. Водночас на інформацію, що друкується, впливають режими фільтрації і сортування, встановлені для цієї таблиці, а також формат самої таблиці, тобто висота рядків, ширина стовпчиків тощо [1].

Наприклад, потрібно роздрукувати інформацію про студентів, тобто прізвище, ім'я та по батькові студента. Для цього потрібно відкрити таблицю **Студенти** і сховати в ній стовпчики **КодСтудента** і **Стан**, які не мають відображатися. Можна також за допомогою команди *Формат / Режим таблиці* встановити в діалоговому вікні *Формат таблиці* параметри для відображення таблиці (рис. 7.4).

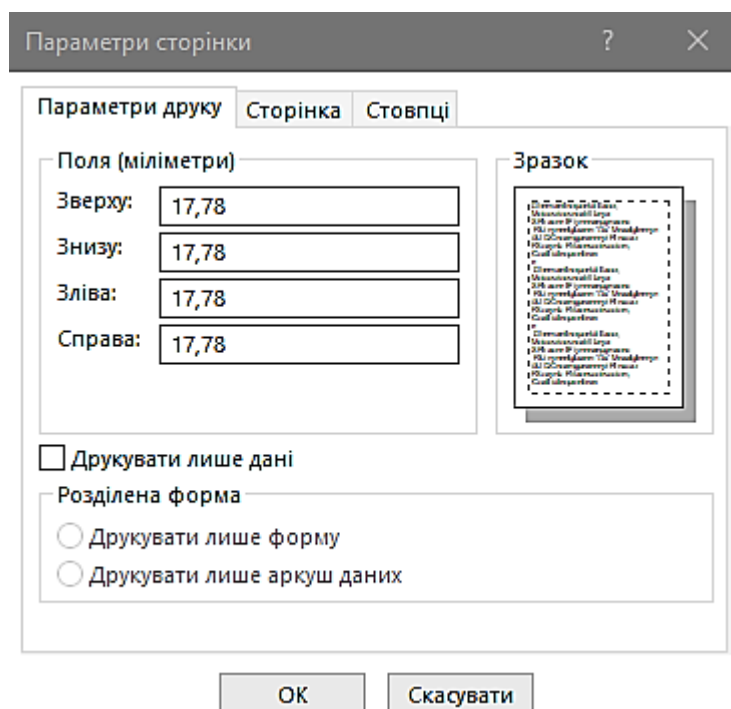


Рис. 7.4. Діалогове вікно *Формат таблиці*

Після того як встановлено формат таблиці, потрібно скористатися командою *Файл | Попередній перегляд*. У цьому режимі можна побачити таблицю в тому вигляді, у якому вона буде надрукована.

В Access існує панель інструментів *Попередній перегляд*, що відображується для перегляду даних перед друком (рис. 7.5) [1].

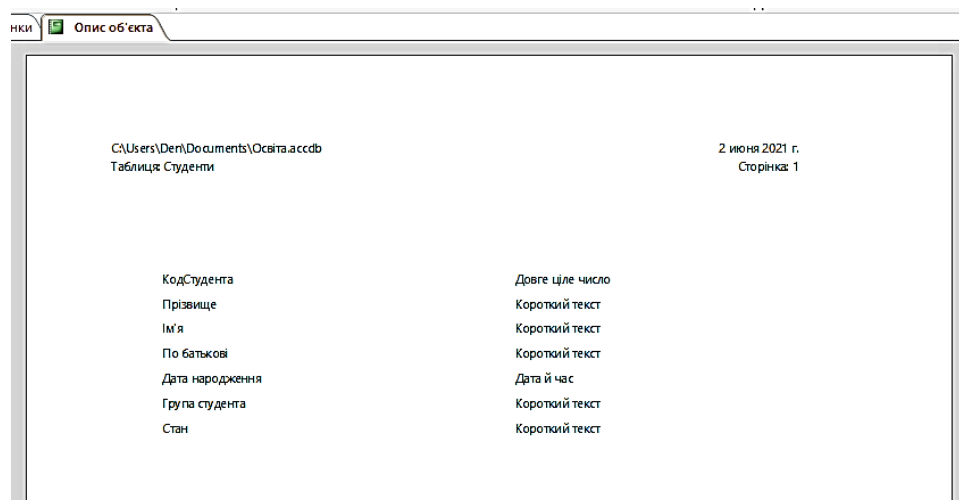







Рис. 7.5. Таблиця Студенти. Режим *Попередній перегляд*

Опис основних кнопок цієї панелі наведено в табл. 7.1.

Таблиця 7.1

Основні кнопки панелі інструментів *Попередній перегляд*

Кнопка	Назва	Опис
	Друк	Відправлення документа на друк
	Одна сторінка	Відображення даних з розташуванням на екрані однієї поточної сторінки
	Дві сторінки	Відображення даних з розташуванням на екрані двох сторінок
	Кілька сторінок	Виведення на екран одночасно декількох сторінок
	Масштаб	Зміна масштабу сторінки
	Закрити	Закриття режиму Попередній перегляд

Слід також зазначити, що для сторінок, які друкують, можуть бути задані параметри: поля, розмір, орієнтація тощо. Для встановлення параметрів потрібно виконати команду *Файл | Параметри сторінки*, після чого буде відображене діалогове вікно *Параметри сторінки* (рис. 7.6) [1].

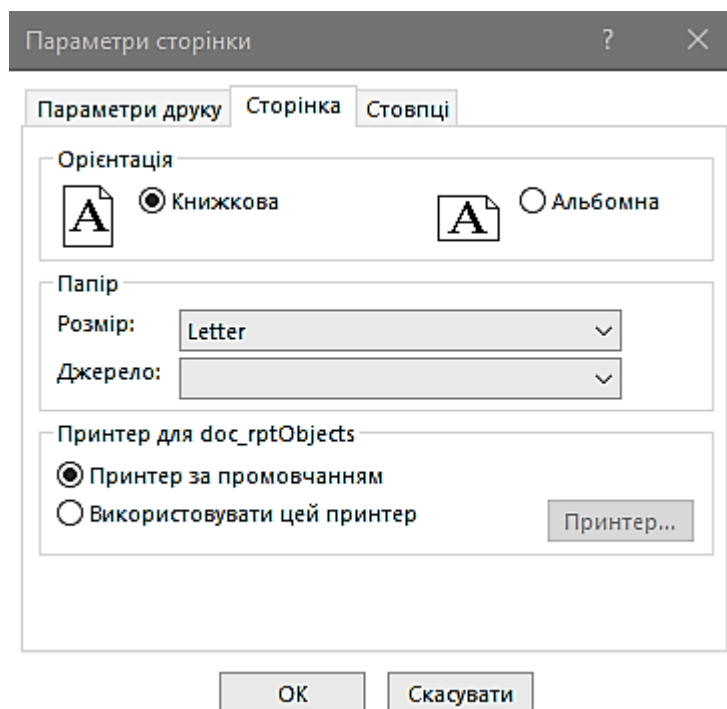


Рис. 7.6. Діалогове вікно *Параметри сторінки*. Вкладка *Сторінка*

На вкладці *Поля* для сторінок, що друкуються, можна встановити розмір полів, а на *вкладці Сторінка* – орієнтацію сторінки (книжкова або альбомна), розмір аркуша тощо.

Щоб відправити дані на друк, можна скористатися кнопкою *Друк* панелі *Попередній перегляд*. У цьому разі документ буде надрукований на поточному принтері з тими налаштуваннями, які для нього встановлені за замовчуванням. З іншого боку, щоб задати деякі параметри друку, наприклад кількість сторінок, що друкуються, або принтер, використовуваний для друку, потрібно скористатися *командою Файл | Друк*, після чого буде відображено діалогове вікно *Друк* (рис. 7.7). У цьому

вікні встановити всі необхідні параметри для друку, після чого натиснути кнопку *OK*. У результаті документ буде відправлений на друк.

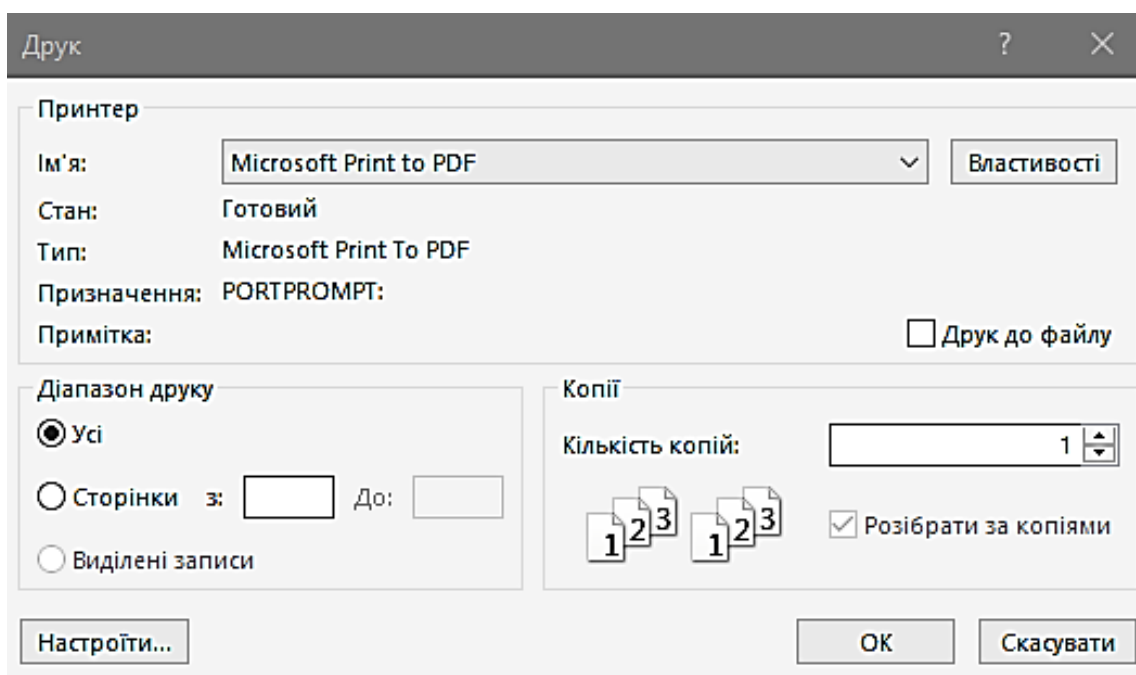


Рис. 7.7. Діалогове вікно *Друк*

7.3. Друк форм

Форма в Access може бути використана не тільки для введення і відображення даних, але і для друку тієї інформації, що подана у формі.

Наприклад, потрібно вивести на друк інформацію про студентів, що відображується у формі **Студенти**. Для цього потрібно змінити форму так, щоб кожен запис відображувався в одному рядку, і назви стовпчиків розмістити в заголовку форми. Після цього можна скористатися кнопкою *Попередній перегляд*, щоб побачити, який вигляд матиме форма після роздрукування (рис. 7.8). Слід зазначити, що в цьому режимі для форми так само, як і для таблиці, доступна панель інструментів **Попередній перегляд**.

	Прізвище	Ім'я	По батькові	Стан
▶	Жукова	Марина	Володимирівна	
	Морозов	Ігор	Олегович	
	Петров	Михайло	Олександрович	
	Ігнатов	Максим	Валерійович	
	Волков	Юрій	Олексійович	
	Волошина	Вікторія	Іванівна	
	Литвинов	Євген	Валерійович	в
	Гущина	Олена	Валеріївна	
	Кузнецов	Руслан	Олексійович	в
	Семененко	Світлана	Станіславівна	в
*				

Рис. 7.8. Попередній перегляд форми **Студенти**

Якщо буде потреба, можна змінити параметри сторінки для форми, що друкується. Для цього, як і в разі друкування таблиці, потрібно скористатися командою *Файл | Параметри сторінки*. У діалоговому вікні *Параметри сторінки* для форми (рис. 7.9) є розглянуті вище вкладки *Поля* і *Сторінка*. Однак є також вкладка *Стовпчики*, у якій можна встановлювати кількість і розмір стовпчиків для відображення даних, тобто в скількох стовпчиках буде розташована інформація на сторінці, а також макет стовпчика [1]:

– *зверху вниз* – дані спочатку виводяться в одному стовпчику, і коли сторінка закінчується, то виведення здійснюється в наступному стовпчику і т. д.;

– *зліва направо* – дані виводяться в першому рядку першого стовпчика, потім у першому рядку другого стовпчика і т. д.

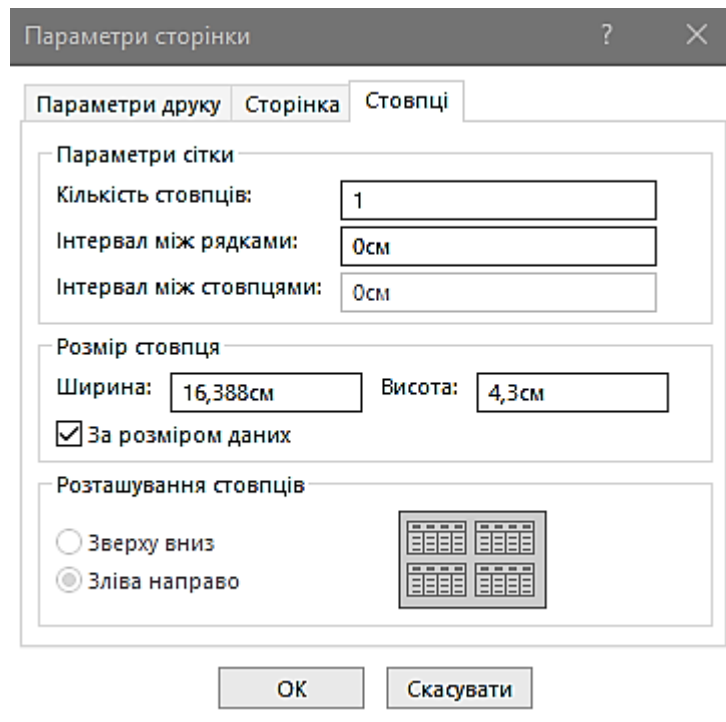


Рис. 7.9. Діалогове вікно *Параметри сторінки*. Вкладка *Стовпчики*

7.4. Друк звітів

Створені звіти можна переглянути перед друком за допомогою кнопки *Попередній перегляд*, аналогічно роботі з таблицями і формами. Водночас також відображується панель інструментів *Попередній перегляд*, кнопки якої дають змогу встановити параметри перегляду сторінок звіту [1].

Для зміни параметрів сторінок звіту потрібно скористатися командою *Файл | Параметри сторінки*, після чого встановити необхідні параметри. Вкладки діалогового вікна *Параметри сторінки* для звіту аналогічні вкладкам для форми. Зокрема на вкладці *Стовпчики* також можна встановити кількість стовпчиків, яка знадобиться для відображення інформації на сторінці, а також встановити макет стовпчика. З іншого боку, на вкладці *Поля* (рис. 7.10) можна змінити розмір полів, встановлені за замовчуванням.

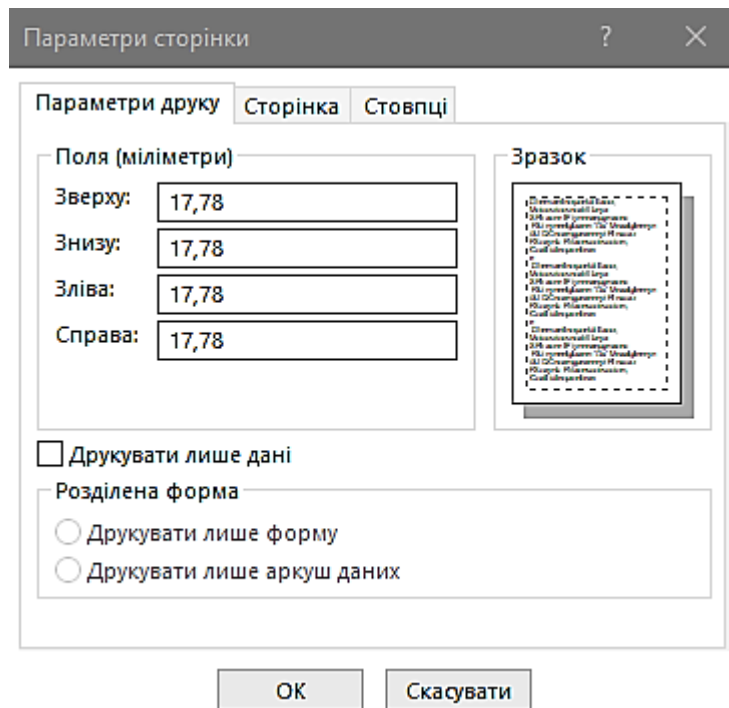


Рис. 7.10. Діалогове вікно *Параметри сторінки*. Вкладка *Поля*

Контрольні запитання до розділу 7

1. Чи можуть бути роздруковані дані, які зберігаються в таблицях?
2. Для чого в Access існує панель інструментів *Попередній перегляд*?
3. Чи можуть бути задані параметри поля, розміру, орієнтації для сторінок, що друкують?
4. Чи може бути використана форма в Access для введення даних?
5. Чи можна змінити параметри сторінки для форми, що друкується?
6. Якою командою потрібно скористатися для зміни параметрів сторінок звіту?
7. Чи можна створені звіти переглянути перед друком?
8. Чи може бути використана форма в Access для відображення даних?

Лабораторна робота 1. СТВОРЕННЯ НЕЗВ'ЯЗАНИХ ТАБЛИЦЬ

1.1. Ознайомлення з інтерфейсом програми Access.

1.2. Створення таблиць у режимі конструктора і заповнення даними.

Порядок виконання роботи

1.1. Ознайомлення з інтерфейсом програми Access

У наш час життя людини настільки сильно насичене різною інформацією, що для її обробки потрібне створення величезної кількості сховищ і банків даних різного призначення. Зараз практично будь-яке завдання пов'язано з маніпулюванням інформацією й даними. Через це останнім часом з'явилося безліч різних комп'ютерних систем, які називаються системами керування базами даних, призначеними саме для цього.

По суті БД – це деяка подоба електронної картотеки, електронного сховища даних, що зберігається в комп'ютері у вигляді одного або декількох файлів. Водночас виникає необхідність у виконанні ряду операцій з БД:

- додавання нової інформації до існуючих файлів БД;
- додавання нових порожніх файлів до БД;
- зміна (модифікація) інформації в існуючих файлах БД;
- пошук інформації в БД;
- видалення інформації з існуючих файлів БД;
- видалення файлів із БД.

Комп'ютеризована інформаційна система являє собою програмний комплекс, завдання якого полягає в підтримці надійного зберігання БД у комп'ютері, виконанні перетворень інформації і відповідних обчислень, наданні користувачам зручного і легкого в освоєнні інтерфейсу.

Традиційно обсяги інформації, з якими доводиться мати справу таким системам, досить великі, а самі БД мають доволі складну структуру. Прикладами інформаційних систем є системи замовлення залізничних або авіаційних квитків, банківські системи та ін.

Як ілюстрацію можна розглянути невелику БД (рис. 1.1), у якій зберігається інформація про стипендії студентів.

KOD	NAME	GROUP	BALL	STIP
525122	Батракова І.Н.	5-4-УПП	4.5	1000
525123	Баулін А.В.	5-4-УПП	4.0	0.00
525124	Белянин Д.А.	5-4-УПП	5.0	1000
525125	Бондаренко О.В.	5-4-УПП	4.5	1000
525126	Зеленський І.В.	5-4-УПП	3.0	1000
525127	Климова В.В.	5-4-УПП	4.5	1000
525128	Кричун О.В.	5-4-УПП	4.0	0.00
525129	Кузнецова В.В.	5-4-УПП	4.5	1000
525130	Нестреляй А.М.	5-4-УПП	3.0	1000
525131	Салова О.П.	5-4-УПП	4.5	1000

Запис →

↑
Поле

Рис. 1.1. Приклад таблиці «Стипендія»

Дані про студентів записуються до таблиці, що має чітку структуру даних. Інформація всередині таблиці складається з *полів*, кожне з яких має власне ім'я. Поля, які є в розглянутому прикладі, подано в табл. 1.1.

Кожен рядок таблиці потрібно розглядати як *запис*, водночас інформація до таблиці заноситься у відповідні поля. Усі записи складаються з однакових полів, а характер інформації для одного поля у всіх записах однаковий. Водночас час різні поля мають різний характер збережених даних.

Опис полів таблиці «Стипендія»


Поле	Опис
KOD	Номер студентського квитка
NAME	Прізвище та ініціали студента
GROUP	Навчальна група студента
BALL	Середній бал успішності студента
STIP	Розмір нарахованої стипендії

Основним призначенням БД насамперед є швидкий пошук інформації, що зберігається в ній. При значному розмірі БД ручний пошук, а також модифікація інформації, що в ній зберігається, забирає значний час. Використання комп'ютера для обробки БД усуває перераховані вище проблеми при пошуку і виборі інформації. У цьому випадку її модифікація здійснюється досить швидко й ефективно, а сама БД легко розміщується на будь-якому диску і за потреби може бути перенесена на інший комп'ютер.

1.2. Створення таблиць у режимі конструктора і заповнення даними

1. За допомогою програми Access створити нову базу даних **Освіта**.

2. У режимі конструктора *під керівництвом викладача* створити таблицю **тблСтуденти** з полями:

– **КодСтудента**. *Тип даних* – Лічильник (*Счетчик*). Це поле буде нашим первинним ключем. Щоб визначити зазначене поле ключовим, потрібно натиснути на іконку *Ключове поле* . На вкладці *Загальне (Загальні)* у розмірі поля залишаємо за замовчуванням *Довге ціле (Довге ціле)*. Індексоване поле (*Індексоване поле*) – Так (Збіги не допускаються);

– **ПрізвищеСтудента**: *Тип даних* – Текстовий (20 символів). *Обов'язкове поле* – Так. *Індексоване поле* – Так (Збіги допускаються);

– **Ім'я Студента:** *Тип даних* – Текстовий (10 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги - допускаються);

– **По батькові Студента:** *Тип даних* – Текстовий (15 символів); *Обов'язкове поле:* – Ні; *Індексоване поле* – Ні;

– **Група Студента:** *Тип даних* – Текстовий (10 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги допускаються);

– **Дата Народження:** *Тип даних* – Дата/час; *Формат поля* – Короткий формат дати; *Обов'язкове поле* – Так; *Індексоване поле* – Так (Співпадіння допускаються);

– **Адреса Студента:** *Тип даних* – Текстовий (20 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги допускаються).

3. Зберігаємо створену таблицю під іменем **тбл Студенти**.

Самостійна робота

4. З режиму конструктора потрібно перейти в режим таблиці і ввести дані про якнайменше вісім студентів:

- *три студенти з Вашої групи;*
- *три студенти з інших груп вашого курсу (поток);*
- *два студенти не з вашого курсу (поток).*

5. Створити нову таблицю (**тбл Викладачі**) у режимі конструктора з полями:

– **Код Викладача:** *Тип даних* – Лічильник; Первинний ключ; *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги не допускаються);

– **Прізвище Викладача:** *Тип даних* – Текстовий (20 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги допускаються);

– **Ім'я Викладача:** *Тип даних* – Текстовий (10 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги допускаються);

– **По батькові Викладача:** *Тип даних* – Текстовий (15 символів); *Обов'язкове поле* – Ні; *Індексоване поле* – Ні;

– **Кафедра Викладача:** *Тип даних* – Текстовий (15 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги допускаються);

– **Вчене Звання:** *Тип даних* – Майстер підстановок. Обираємо «Буде введено фіксований набір значень». Далі. У комірці (кожне слово в новому рядку) вписуємо: доцент, професор, академік. Далі. У наступному вікні залишаємо все за замовчуванням. *Готово*. *Обов'язкове поле* – Ні; *Індексоване поле* – Так (Збіги допускаються).

6. Зберігаємо таблицю з ім'ям **тбл Викладачі**.

7. До таблиці **тбл Викладачі** потрібно ввести не менше шести викладачів з різних кафедр. Щонайменше одного викладача вказати без вченого звання і хоча б два – три викладачі (але не всі) мають бути з однієї кафедри.

Лабораторна робота 2. СТВОРЕННЯ ЗВ'ЯЗАНИХ ТАБЛИЦЬ

2.1. Встановлення зв'язків між таблицями з використанням майстра підстановок і заповнення даними.

2.2. Сортування та фільтрація даних. Ознайомлення з елементом Схема даних.

Порядок виконання роботи

2.1. Напередодні ознайомитись з теоретичним матеріалом розд. 3.

Як приклад у цьому розділі розглянута навчальна БД **Освіта**, структура якої була докладно описана в попередньому розділі. Дані, які зберігаються в її таблицях, наведені в табл. 2.1 – 2.4.

Таблиця 2.1

Таблиця Студенти

КодСтудента	Прізвище	Ім'я	По батькові
1	Жукова	Марина	Володимирівна
2	Морозов	Ігор	Олегович
3	Петров	Михайло	Олександрович
4	Ігнатов	Максим	Валерійович
5	Волков	Юрій	Олексійович
6	Волошина	Вікторія	Іванівна
7	Литвинов	Євгеній	Валерійович
8	Гущина	Олена	Валеріївна
9	Кузнєцов	Руслан	Олексійович
10	Семененко	Світлана	Станіславівна

Таблиця 2.2

Таблиця Викладачі

КодВикладача	Прізвище	Ім'я	По батькові
1	Вікулина	Валентина	Іванівна
2	Костиркін	Олег	Володимирович
3	Казанко	Віталій	Володимирович
4	Позднякова	Любов	Олексіївна
5	Загарійчук	Ігор	Дмитрович

Таблиця 2.3

Таблиця Предмети

КодПредмета	КодВикладача	Назва	КількістьГодин
1	1	Фізика	34
2	2	Хімія	68
3	3	Математика	68
4	4	Економіка	17
5	5	Філософія	17

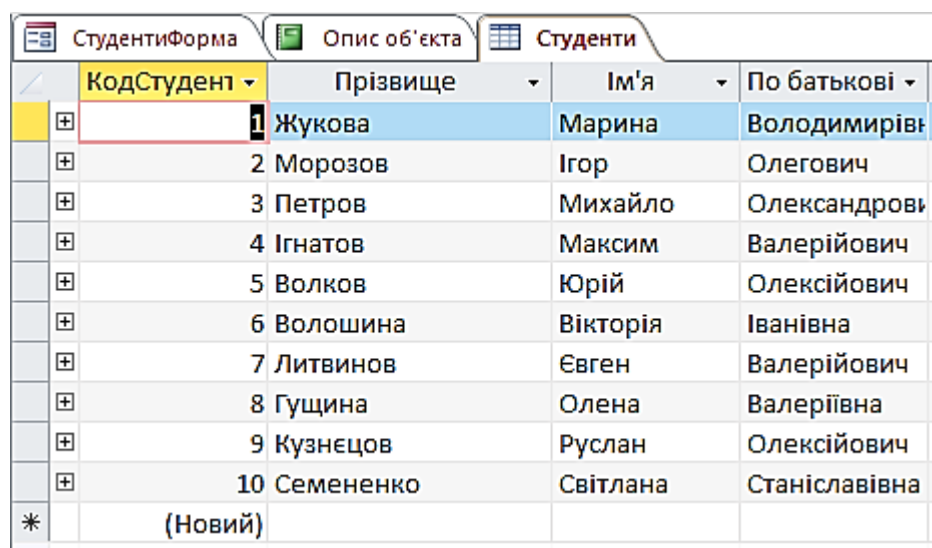
Таблиця 2.4

Таблиця Оцінки

КодОцінки	КодСтудента	Код Предмета	Оцінка	Дата Складання
1	1	1	5	15.01.21
2	1	2	5	20.01.21
3	2	1	4	15.01.21
4	2	2	4	20.01.21
5	3	5	3	10.02.21
6	3	4	4	25.01.21
7	3	3	5	22.01.21

Ключові поля в таблицях (наприклад **КодСтудента** в таблиці **Студенти**) заповнюються автоматично, у процесі введення нових даних.


Далі будуть розглянуті основні можливості обробки даних у режимі таблиці. Водночас як приклад буде використовуватися таблиця **Студенти** (рис. 2.1).



КодСтудент	Прізвище	Ім'я	По батькові
1	Жукова	Марина	Володимирів
2	Морозов	Ігор	Олегович
3	Петров	Михайло	Олександров
4	Ігнатов	Максим	Валерійович
5	Волков	Юрій	Олексійович
6	Волошина	Вікторія	Іванівна
7	Литвинов	Євген	Валерійович
8	Гущина	Олена	Валеріївна
9	Кузнєцов	Руслан	Олексійович
10	Семененко	Світлана	Станіславівна
*	(Новий)		

Рис. 2.1. Таблиця Студенти

Сортування даних. Коли в Access відкривається будь-яка таблиця, то записи в ній подані в тому порядку, у якому вони були введені. Щоб змінити порядок відображення записів, можна скористатися можливостями *сортування*. Щоб відсортувати записи в таблиці за зростанням або зменшенням значень певного стовпчика, цей стовпчик спочатку потрібно виділити, потім вибрати пункт *Записи / Сортування* і вибрати одну з команд:

Сортування за зростанням (кнопка ) або

Сортування за зменшенням (кнопка )

1. У режимі конструктора (під керівництвом викладача) створити таблицю **тбл Дисципліни** з полями:

– **Код Дисципліни:** *Тип даних:* Лічильник (Счетчик). Первинний ключ; *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги не допускаються);

– **Код Викладача:** *Тип даних:* Майстер підстановок. З двох можливих варіантів обираємо той, у якому пропонується використовувати значення з таблиці або запиту (*Об'єкт столбец подстановки будет использовать значения из таблицы или запроса*). Тиснемо *Далі*. Із запропонованого списку вибираємо таблицю **тбл Викладачі**. Тиснемо *Далі*. У списку *Доступні поля* вибираємо **Код Викладача** і **Прізвище Викладача**. Тиснемо *Далі*. Обираємо порядок сортування списку – **Прізвище Викладача** (можливі варіанти сортування в алфавітному порядку або ж навпаки). Тиснемо *Далі*. За потреби можна змінити ширину стовпчика, який містить стовпчик підстановки. Тиснемо *Далі*. У наступному вікні залишаємо все за замовчуванням. Тиснемо *Готово*. У вікні, що випадає, погоджуємось з пропозицією зберегти таблицю і вводимо її назву (**тбл Дисципліни**). В описі параметрів цього поля визначаємо: *Обов'язкове поле:* Так; *Індексоване поле:* Так (Співпадання не допускаються);

– **Назва Дисципліни:** *Тип даних:* Текстовий (20 символів); *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги допускаються);

– **Кількість Годин:** *Тип даних:* Числове (ціле); *Обов'язкове поле:* Так; *Індексоване поле:* Ні.

2. Зберігаємо таблицю (назву вже не вводимо, оскільки ми ввели її під час виконання попереднього пункту).

3. У режимі таблиці потрібно ввести дані про 5-7 дисциплін (у таблиці має бути щонайменше один викладач, що викладає більше однієї дисципліни).


4. У режимі конструктора створити таблицю **тбл Оцінки** з полями:

- **Код Дисципліни:** *Тип даних:* Майстер підстановок (усі дії у **Майстрі** підстановок виконуються самостійно аналогічно з попереднім завданням, ключове поле не задається). Для підстановки вибираються поля **Код Дисципліни** і **Назва Дисципліни**. *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги допускаються);

– **Код Студента:** *Тип даних:* Майстер підстановок (усі дії у **Майстрі** підстановок виконуються самостійно аналогічно з попереднім завданням). Для підстановки вибираються поля **Код Студента**, **Прізвище Студента**, **Ім'я Студента** і **Група Студента**. *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги допускаються);

– **Оцінка:** *Тип даних:* Числовий (ціле); *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги допускаються);

– **Дата Складання:** *Тип даних:* Дата/час (Короткий формат дати); *Обов'язкове поле:* Так; *Індексоване поле:* Так (збіги допускаються).

5. Для таблиці **тбл Оцінки** ключовими є пара полів **Код Дисципліни** і **Код Студента**. Потрібно виділити за допомогою мишки обидва поля і натиснути на панелі інструментів кнопку *Ключове поле* .

6. У режимі таблиці потрібно ввести інформацію про дві – три оцінки, виставлені кожному студенту з різних дисциплін, використовуючи підстановку в полях **КодСтудента** і **КодДисципліни**.

7. Відкрити вікно *схема даних*. Перевірити коректність структури даних і встановити цілісність даних у вікні *Зміна зв'язків*. Показати результат викладачеві.

8. Самостійно, використовуючи теоретичний матеріал розд. 3, застосувати різні варіанти сортування і фільтрації (за прізвищем, датою народження, номером групи і т. ін.).

Лабораторна робота 3. СТВОРЕННЯ ЗАПИТІВ

Виведення запиту на екран:

- у вікні відкритої БД вибираємо в списку *Об'єкти* кнопку *Запити*;
- виділяємо *Створення запиту* в режимі конструктора. Натискаємо ОК;
- додаємо за допомогою кнопки *Показати таблицю* ТБЛ *Студенти*;
- для виведення всіх даних таблиці *Студенти* натискаємо на «*» (у самому верхньому полі) подвійним натисканням лівою кнопкою миші;
- відображуємо запит на екрані за допомогою кнопки *Запуск «!»*;
- зберігаємо наш перший запит. **Файл, Зберегти як, «ЗПРСтуденти».**

Ознайомлення з конструктором:

- створюємо запит у режимі конструктора;
- додаємо за допомогою кнопки *Показати таблицю* ТБЛ *Студенти*;
- для роботи нам знадобляться прізвище, ім'я та група студента.

Існує три різні способи вибору потрібних нам комірок з тієї чи іншої таблиці:

1. Подвійне натискання лівою кнопкою миші на потрібній нам комірці (додаємо таким способом комірку *ПрізвищеСтудента*).

2. Затиснувши потрібну комірку лівою кнопкою миші і перетягнувши на вільне поле (додамо комірку *Ім'яСтудента*).

3. Вибравши потрібну нам комірку з випливаючого списку (проведемо цю процедуру над коміркою *ГрупаСтудента*):

- запускаємо запит, натиснувши на кнопку *Запуск «!»*;
- переконуємося, що всі три колонки з прізвищами, іменами та групами студентів відобразилися коректно;
- переходимо в режим конструктора;

– ознайомимося з його структурою.

У верхній частині вікна конструктора розташовано таблиці (або таблиця), використовувані як джерело даних. У нижній частині вікна вказуються параметри, які відбираються до вибірки даних:

- 1) поле;
- 2) ім'я таблиці;
- 3) сортування;
- 4) виведення на екран;
- 5) умова відбору.

Для детальшого пояснення останньої комірки, слід виконати такі дії:

- створюємо багатотабличний запит;
- додаємо за допомогою кнопки *Показати таблицю* **ТБЛ Успішність**

і ТБЛ Дисципліни;

- вибираємо комірки **НазваДисципліни, Оцінка;**
- тепер у нас в нижній частині вікна розташовуються п'ять колонок:

ПрізвищеСтудента, Ім'яСтудента, ГрупаСтудента, Дисципліна, Оцінка.

Завдання 1. Вивести всіх відмінників на екран:

– згадуємо параметри запису розширеного фільтра. Оскільки студенти оцінюються за стобальною шкалою, відмінники – це всі студенти, які отримали 90 балів і вище. Тому в колонці Оцінка в комірці Умова відбору записуємо Like «9 *». Але оскільки фільтр відсіє все значення, крім діапазону від 90 до 99, нам потрібно ще додати значення 100. Вписуємо в комірку цифру 100. Альтернатива запису: у комірці Умова відбору вписуємо Like «9 *» or 100;

- відображуємо запит на екрані за допомогою кнопки *Запуск «!»*;
- зберігаємо запит. *Файл, Зберегти як, «ЗПРВідмінники»*;
- не закриваючи запит, переходимо в режим конструктора;

- перевіримо, чи вчасно студент склав дисципліну і чи законно поставлена оцінка відмінно. Додаємо комірку *Датаскладання*;
- виводимо запит на екран за допомогою «!»;
- звертаємо увагу, що прізвища розташовані в довільному порядку;
- для того щоб упорядкувати прізвища за алфавітом, переходимо в режим конструктора;
- у комірці *Сортування* виставляємо **За зростанням**.
- виводимо запит на екран і переконуємося, що сортування виконалося коректно;
- зберігаємо запит під ім'ям **ЗПРСтуденти_Відмінники**;
- переходимо в режим конструктора;
- додаємо в умови відбору конкретну групу студентів і виводимо запит на екран за допомогою «!»;
- зберігаємо запит **ЗПРСтуденти_Відмінники_** (маркування групи).

Завдання 2. Подивитися успішність студентів у всіх групах за двох умов:

- щоб і прізвища студентів, і групи впорядкувати за зростанням;
- так згрупувати студентів, щоб вони відобразилися в алфавітному порядку у своїх групах, а не єдиним алфавітним списком на весь ЗВО.

Розв'язання:

- переходимо в режим конструктора;
- додаємо за допомогою кнопки *Показати таблицю ТБЛ Студенти, ТБЛ Дисципліни, ТБЛ Успішність*;
- для роботи знадобляться комірки **ПрізвищеСтудента, Ім'яСтудента, По батьковіСтудента, ГрупаСтудента, Дисципліна, Оцінка**;
- у колонках **ПрізвищеСтудента** і **ГрупаСтудента** виставляємо сортування *За зростанням*;

– виводимо запит на екран і бачимо, що всі студенти ЗВО розмістилися в алфавітному порядку, але ніяк не за групами.

Переходимо назад у режим **Конструктор**, щоб вирішити цю проблему і впоратися з поставленим завданням:

- переносимо колонку ГрупаСтудента на перше місце;
- отже, у нас у першу чергу відсортовуються групи, а в них вже в другу чергу будуть сортуватися студенти за прізвищами;
- відображуємо запит на екрані. Завдання виконано (табл. 3.1).

Таблиця 3.1

Група Студента	ІМ'Я Студента
2-3-СКС	Іванов Іван Іванович
2-3-СКС	Лаврентьєва Ольга Миколаївна
2-3-СКС	Семенов Валентин Сергійович
4-4-СКС	Антонов Михайло Олегович
4-4-СКС	Козлова Анна Миколаївна
4-4-СКС	Петренко Катерина Семенівна
4-4-СКС	Татарський Віктор Олександрович
4-4-СКС	Якушев Леонід Данилович
4-5-УПП	Краснокутська Євгенія Василівна
4-5-УПП	Романенко Олексій Федорович

Завдання 3. Відобразити на екрані успішність студентів з таким самим сортуванням, як зазначено в завданні 2, але водночас домогтися того, щоб на першому місці виводилася на екран колонка з прізвищем студента, далі йшла колонка з ім'ям, далі по батькові і тільки після цього йшла колонка з групою студента.

Завдання вирішити самостійно. Зберегти запит.

Параметричний запит з клавіатури:

- для роботи будемо використовувати попередню форму запиту;

- для прикладу створимо параметричний запит з клавіатури на прізвище студента;
- у колонці **ПрізвищеСтудента**, в комірці **Умова** відбору пишемо [Введіть прізвище студента];
- натискаємо кнопку Запуск «!»;
- тепер, щоб знайти конкретного студента в нашому списку, досить буде ввести лише його прізвище і натиснути кнопку *OK*;
- також можлива маніпуляція не з одним видом даних, а відразу з декількома для звуження кола пошуку, наприклад поставити параметричний запит з клавіатури на **Прізвище, Ім'я та Групу студента**.

Самостійна робота

Створити запит за вибіркою. Вивести інформацію про навантаження викладачів із сортуванням за кафедрою і прізвищем (вид сортування аналогічний завданню 2). Створити параметричний запит з введенням назви кафедри.

Зберегти запит.

Лабораторна робота 4. РОБОТА З ГРУПОВИМИ ОПЕРАЦІЯМИ

У вікні відкритої БД вибираємо в списку Об'єкти кнопку *Запити*:

– виділяємо *Створення запиту* в режимі конструктора. Натискаємо ОК;

– додаємо за допомогою кнопки *Показати* таблицю **ТБЛ Студенти, ТБЛ Успішність**;

– працювати будемо з комірками: **КодСтудента, ПрізвищеСтудента, Ім'яСтудента, По батьковіСтудента, ГрупаСтудента, Дисципліна, Оцінка**;

– для роботи з груповими операціями натискаємо іконку Σ (*Totals*).

Завдання. Дізнатися середній бал кожного студента:

- у колонці Оцінка вибрати в меню Avg, що впливає;
- запитувати інформацію («!»);
- бачимо, що нічого не змінилося, тому що БД вивела середнє значення (середній бал студента) з кожної дисципліни.

Питання

Як вивести загальний середній бал з усіх дисциплін у кожного студента?

Завдання вирішити самостійно.

Завдання 1. Підрахувати загальну кількість оцінок кожного студента:

- додаємо ще одну колонку *Оцінка*;
- у меню, що впливає, вибираємо *Count*;
- переходимо в режим ТБЛ;
- бачимо, що «заголовки» колонок названі за замовчуванням Avg-оцінка і Count-оцінка.

Питання

Як перейменувати «заголовок» колонок Avg-оцінка і Count-оцінка?

Розв'язання

Досить у запиті у верхній комірці поставити знак двокрапки («:») перед назвою використовуваної таблиці (у цьому випадку **ТБЛ Успішність**). А перед «:» написати нове ім'я «заголовка» цієї колонки.

Тепер комірки мають такий вигляд:

Середній бал: Оцінка

Кількість оцінок: Оцінка

Зберігаємо запит.

Питання

Як вивести успішність кожної з груп?

Завдання вирішити самостійно.

Самостійна робота 1. Підрахувати і вивести кількість студентів у групі і другим запитом - кількість студентів в університеті.

Самостійна робота 2. Вивести навантаження викладачів і другим запитом – навантаження кафедри.

Для запиту знадобляться:

– **ТБЛ Викладачі** та **ТБЛ Дисципліни**;

– комірки **КодВикладача, ПрізвищеВикладача, Ім'яВикладача,**

По батьковіВикладача, КафедраВикладача, НазваДисципліни, Години.

Зберегти обидва запити.

Робота з перехресними запитами. Створення зведеної відомості:

– створюємо новий запит у режимі конструктора;

– додаємо за допомогою кнопки *Показати таблицю* **ТБЛ Студенти,**

ТБЛ Успішність;

– працювати будемо з комірками **ПрізвищеСтудента, КодДисципліни, Оцінка;**

– тип запиту виставляємо CrossTab (перехресний);

– підключаємо групові операції натисканням на іконку Σ (*Totals*);

– у комірці **ПрізвищеСтудента** виставляємо crosstab, заголовки рядків (маючи на увазі, що як заголовки рядків потрібно визначити хоча б одне поле);

– у комірці **КодДисципліни** виставляємо crosstab, заголовки стовпчиків (ураховуючи, що як заголовки стовпчиків потрібно визначити тільки одне поле);

– у комірці **Оцінка** виставляємо crosstab, значення (ураховуючи, що як значення потрібно визначити тільки одне поле) і groupby (Avg) для виведення середнього бала студента;

– виводимо запит на екран;

– бачимо, що всі дисципліни вивелися в цифровому вигляді (за кодом кожної дисципліни);

– додаємо **ТБЛ Дисципліни** (тому що комірки **НазваДисципліни** в **ТБЛ Успішність** нема);

– видаляємо в конструкторі запити комірку **КодДисципліни** і замість неї додаємо комірку **Назва Дисципліни**, виставляємо crosstab, заголовки стовпчиків;

– виводимо запит на екран.

У зведених відомостях прізвища студента, як правило, недостатньо. Потрібно вписати як ім'я, по батькові студента, так і його групу. Для цього:

– додаємо комірки **Ім'яСтудента, По батьковіСтудента, ГрупаСтудента;**

– виставляємо в доданих комірках поле crosstab, заголовки рядків (пам'ятаємо, що як заголовки рядків ми маємо право визначити одне і більше значень);

- виводимо запит на екран;
- бачимо, що прізвище, ім'я, по батькові та група кожного студента містяться в окремій комірці.

Питання

Як об'єднати комірки з прізвищем, ім'ям, по батькові і групою студента в одну загальну?

Розв'язання:

– потрібно підключити будівник виразів за допомогою кнопки *Побудувати*;

– додаємо (подвійним натисканням лівою кнопкою миші) комірки **ІмяСтудента**, **По-батьковіСтудента** і **ГрупаСтудента**. Дані комірки відобразяться в середньому полі;

- у верхньому полі замінюємо всі «expr» на & (знак об'єднання);
- натискаємо Ок;
- виводимо запит на екран;
- для того щоб розставити пропуски, знову переходимо в будівник виразів і у верхньому полі поруч зі знаками & вставляємо пропуск у подвійних лапках («»). Рядок повинен мати такий вигляд: [...] [...] & «» & [...] [...] і т. д.;

- виводимо запит на екран;
- подвійним натисканням лівою кнопкою миші по межах комірок (у режимі ТБЛ) робимо автоматичний підбір ширини колонок.

Самостійно перейменувати назву колонки з **ПБ і групою студентів**.

Зберегти запит.

Самостійна робота 3. Вивести зведену відомість викладачів і їхнє навантаження. У полі Значення вказати загальну кількість годин.

Зберегти запит.

Завдання. Створити запит виведення ПБ студентів і групу кожного студента. Замінити повні імена та по батькові студентів на ініціали:

- створюємо новий запит у режимі конструктора;
- додаємо за допомогою кнопки *Показати таблицю ТБЛ Студенти*;
- додаємо комірки **ПрізвищеСтудента, ІмяСтудента, По-батькові студента, ГрупаСтудента**;
- встановлюємо курсор в рядку **ПолеПрізвищеСтудента**;
- натискаємо кнопку *Побудувати*, тим самим запускаючи будівник виразів;
- у верхньому полі після **ПрізвищеСтудента** ставимо & «» &, щоб додати пробіл і можливість додавання функції;
- вибираємо в лівому полі Функції, далі Вбудовані функції;
- у середньому полі вибираємо Текстові;
- у правому полі вибираємо Left;
- устанавлюємо курсор у верхньому полі в те місце виразу, куди ця функція має бути вставлена, і натискаємо кнопку Вставити у вікні **Будівника виразів**;
- у результаті цих дій до виразу буде додано шаблон функції Left: Left («stringexpr»; «n»);
- замість «stringexpr» вписуємо Ім'яСтудента. Замість «n» – кількість символів, які слід залишити зліва. Оскільки в ініціалах, як правило, використовують лише першу букву імені – ставимо замість «n» цифру 1. Вираз повинен мати такий вигляд:


```
[ПрізвищеСтудента] & "" & Left ([Ім'яСтудента]; 1);
```
- після ініціалу імені слід поставити крапку. Тепер вираз має такий вигляд:


```
[ПрізвищеСтудента] & "" & Left ([ІмяСтудента]; 1) & ".".
```
- Самостійно виконати аналогічні дії і додати ініціал по батькові до прізвища та ініціалів імені студента.

Лабораторна робота 5. СТВОРЕННЯ ФОРМ

Пояснення, чому часто дуже незручно працювати з таблицями, які, по-перше, самі по собі об'ємні, по-друге, вони ще й містять підтаблиці.

У вікні відкритої БД вибираємо в списку Об'єкти кнопку *Таблиці*:

- виділяємо **ТБЛСтуденти**. Натискаємо на іконку *Автоформа*;
- ознайомлення з вмістом автоформи: робочі поля, кнопки.

Переходимо в режим конструктора:

- ознайомлення з Областю даних;
- ознайомлення з панеллю елементів.

Пояснення, як виділяти об'єкти, перетягувати їх по полю, змінювати розмір як об'єктів, так і всієї форми в цілому, за допомогою чого можна змінити розмір / колір / нахил / жирність написів.

Самостійна робота 1

1. У режимі конструктора змінити зовнішній вигляд форми на свій смак за допомогою маніпуляції над розмірами і кольором шрифту, а також розташуванням робочих комірок.

2. У режимі форми змінити інформацію одного зі студентів.

3. У режимі форми додати нового студента і повністю заповнити інформацію про нього.

Зберігаємо форму. Називаємо **ФРМСтуденти**:

– у вікні відкритої БД вибираємо в списку Об'єкти кнопку *Таблиці*;
– виділяємо **ТБЛОцінки**. Відкриваємо список, що впливає в іконці *Автоформа*;

– вибираємо Форма;

– вибираємо Автоформа: рядкова. У нижньому полі має бути обрана **ТБЛОцінки**. Натискаємо *ОК*;

- переходимо в режим конструктора форми;
- ознайомлення з зовнішнім виглядом рядкової автоформи;
- ознайомлення з областями Тема форми, Область даних, Примітка форми;
- у Примітці форми створюємо поле **Середній бал**: вибираємо на панелі елементів іконку *Поле* і перетягуємо її на область примітки форми. Замість слова Поле записуємо **Середній бал**. У комірці **Вільний** записуємо = avg ([Оцінка]);
- переходимо в режим форми і перевіряємо – чи коректно відображується середній бал студентів;
- застосовуємо сортування і фільтри в рядковій формі.

Самостійна робота 2

1. Додати в примітки форми підрахунок кількості оцінок.
2. Замінити розташування / розмір / колір / жирність шрифту на свій смак. Зберегти форму.
3. Створити стрічкову форму таблиці Викладачі. Підрахувати кількість викладачів у ВНЗ. Зберегти форму.

Питання

Як вивести інформацію про викладачів і підрахувати їхнє підсумкове навантаження?

Розв'язання:

- у вікні відкритої БД вибираємо в списку Об'єкти кнопку *Таблиці*;
- виділяємо **ТБЛОценкі**. Відкриваємо список, що впливає в іконці *Автоформа*;
- вибираємо Форма;
- вибираємо **Майстер форм**. Натискаємо ОК;

– вибираємо в полі **Таблиці та Запити** ТБЛ**Викладачі**. У полі **Доступні поля** вибираємо **КодВикладача, ФаміліяВикладача, ІмяВикладача, По-батьковіВикладача, КафедраВикладача**;

– у полі **Таблиці та Запити** переходимо на ТБЛ**Дисципліни**. У полі **Доступна поля** вибираємо **НазваДісципліни, КількістьГодин**;

– натискаємо *Далі*;

– вибираємо зовнішній вигляд форми: **Рядковий**;

– натискаємо *Далі*;

– самостійно вибираємо стиль форми, який сподобався;

– натискаємо *Далі*;

– виділяємо *Відкриваємо форму* для перегляду і введення даних;

– натискаємо **Готово**;

– переходимо в режим конструктора форми;

– вибираємо в **Панелі елементів** іконку *Поле* і перетягуємо її на область примітки форми. Замість слова **Поле** записуємо **Підсумкове навантаження**. У комірці **Вільний** записуємо = sum ([КількістьГодин]);

– зберігаємо форму.

Лабораторна робота 6. РОЗРОБЛЕННЯ ЗВІТІВ

Усе виконується так само, як у лабораторній роботі 5.

Пояснення, чому часто дуже незручно працювати з таблицями, які, по-перше, самі по собі об'ємні, по-друге, вони ще й містять підтаблиці.

У вікні відкритої БД вибираємо в списку Об'єкти кнопку *Таблиці*:

- виділяємо **ТБЛСтуденти**. Натискаємо на іконку *Автозвіт*;
- ознайомлення з вмістом автозвіту: робочі поля, кнопки.

Переходимо в режим **Конструктор**:

- ознайомлення з **Областю даних**;
- ознайомлення з **Панеллю елементів**.

Пояснення, як виділяти об'єкти, перетягувати їх по полю, змінювати розмір як об'єктів, так і всього звіту в цілому, за допомогою чого можна змінити розмір / колір / нахил / жирність написів.

Самостійна робота 1

1. У режимі конструктора змінити зовнішній вигляд звіту на свій смак за допомогою маніпуляції над розмірами і кольором шрифту, а також розташуванням робочих комірок.

2. У режимі звіту змінити інформацію одного зі студентів.

3. У режимі звіту додати нового студента і повністю заповнити інформацію про нього.

Зберігаємо звіт. Називаємо **ФРМСтуденти**:

- у вікні відкритої БД вибираємо в списку Об'єкти кнопку *Таблиці*;
- виділяємо **ТБЛОцінки**. Відкриваємо список, що впливає в іконці *Автозвіт*;
- вибираємо **Звіт**;

- вибираємо Автозвіт: рядкова. У нижньому полі має бути обрана **ТБЛЮцінки**. Натискаємо ОК;
- переходимо в режим **Конструктор звіту**;
- ознайомлення з зовнішнім виглядом стрічкового автозвіту;
- ознайомлення з областями: Тема звіт, Область даних, Примітка звіти;
- у Примітці звіти створюємо поле **Середній бал**: вибираємо на панелі елементів іконку *Поле* і перетягуємо її на область примітки звіту. Замість слова Поле записуємо Середній бал. До комірки **Вільний** записуємо = avg ([Оцінка]);
- переходимо в режим **Звіт** і перевіряємо, чи коректно відображується середній бал студентів;
- застосовуємо сортування і фільтри в рядковому звіті.

Самостійна робота 2

1. Додати в **Примітки звіту** підрахунок кількості оцінок.
2. Замінити розташування / розмір / колір / жирність шрифту на свій смак. Зберегти звіт.
3. Створити рядковий звіт таблиці **Викладачі**. Підрахувати кількість викладачів у ЗВО. Зберегти звіт.

Питання

Як вивести інформацію про викладачів і підрахувати їхнє підсумкове навантаження?

Розв'язання:

- у вікні відкритої БД вибираємо в списку Об'єкти кнопку *Таблиці*;
- виділяємо **ТБЛЮцінки**. Відкриваємо список, що впливає в іконці *Автозвіт*;
- вибираємо Звіт;

- вибираємо **Майстер звіту**. Натискаємо ОК;
- вибираємо в полі Таблиці та Запити **ТБЛВикладачі**. У полі **Доступні поля** вибираємо **КодВикладача, ПрізвищеВикладача, Ім'яВикладача, По батьковіВикладача, КафедраВикладача**;
- у полі **Таблиці та Запити** переходимо на **ТБЛДисципліни**. У полі **Доступні поля** вибираємо **НазваДісципліни, КількістьГодин**;
- натискаємо *Далі*;
- вибираємо зовнішній вигляд звіту: **Рядковий**;
- натискаємо *Далі*;
- самостійно вибираємо стиль звіту, який сподобався;
- натискаємо *Далі*;
- виділяємо *Відкриваємо звіт* для перегляду і введення даних;
- натискаємо **Готово**;
- переходимо в режим конструктора звіту;
- вибираємо в Панелі елементів іконку *Поле* і перетягуємо її на область примітки звіту. Замість слова **Поле** записуємо **Підсумкове навантаження**. У комірці **Вільний** записуємо = sum ([КількістьГодин]);
- зберігаємо звіт.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Roger Jennings Special Edition Using Microsoft Access 2000 — USA: Indianapolis, Indiana 46290, 1999. 1147 p.
2. Федунець А. Д. Пакети програм офісного призначення. Кіровоград: Народне слово, 2000. 272 с.
3. Історія розвитку баз даних. URL: helpiks.org/2-9486.html.
4. Гайна Г. А. Основи проектування баз даних: навч. посіб. Київ: Кондор, 2008. 200 с.
5. Пасічник В. В., Резніченко В. А. Організація баз даних та знань. Київ: Видавнича група ВНУ, 2006. 384 с.
6. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань: навч. посіб. Кн. 2. Системи управління базами даних та знань. Львів: Магнолія-2006, 2012. 584 с.
7. Грицунов О. В. Інформаційні системи та технології: навч. посіб. Харків: ХНАМГ, 2010. 222 с.
8. Завадський І. О. Основи баз даних: навч. посіб. Київ: Видавець І. О. Завадський, 2011. 192 с.
9. Зарицька О. Л. Бази даних та інформаційні системи : метод. посіб. Житомир: Вид-во ЖДУ ім. І. Франка, 2009. 132 с
10. Створення зв'язку. URL: <https://support.office.com/uk-UA/article/Створення-зв-язку-40c998dd-4875-4da4-98c8-8ac8f109b85b>. Назва з екрана.
11. Класифікація моделей даних. URL: <https://helpiks.org/2-9488.html>. Назва з екрана.
12. Access 2016 Bible 1st Edition by Michael Alexander, Richard Kusleika. 1152 p. ISBN: 978-1119086543.
13. Exploring Microsoft Office Access 2019 Comprehensive 1st Edition by Mary Anne Poatsy, Jerri Williams, Amy Rutledge. 640 p. ISBN: 978-0135435816.

14. MS ACCESS 2016 SQL COMPREHENSIVE by Dr. Pindaro E. Demertzoglou. 461 p. ISBN: 978-0988330092.

15. Problem Solving Cases In Microsoft Access & Excel 16th Edition by Ellen Monk, Joseph Brady, Emilio Mendelsohn. 256 p. ISBN: 978-0357138632.

16. Looseleaf for Microsoft Access 365 Complete: In Practice, 2019 Edition 1st Edition by Annette Easton, Randy Nordell. 928 p. ISBN: 978-1260818260.

Навчальний посібник

Доценко Сергій Ілліч,
Будніченко Єгор Миколайович

ЛАБОРАТОРНИЙ ПРАКТИКУМ

до виконання лабораторних робіт з дисципліни
«ОРГАНІЗАЦІЯ ТА СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ»

Відповідальний за випуск Доценко С. І.

Редактор Ібрагімова Н. В.

Підписано до друку 11.06.2021 р.

Умовн. друк. арк. 9,0. Тираж . Замовлення № .

Видавець та виготовлювач Український державний університет
залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.

Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.