

ВИДАВНИЦТВО
РАНОК

ІНФОРМАТИКА

В. Д. Руденко

Бази даних

МОДУЛЬ ДЛЯ УЧНІВ 10–11 КЛАСІВ

РІВЕНЬ СТАНДАРТУ



ІНФОРМАТИКА

В. Д. Руденко

Бази даних

**ДЛЯ УЧНІВ 10–11 КЛАСІВ
РІВЕНЬ СТАНДАРТУ**

ВИДАВНИЦТВО
РАНОК

УДК 004.65-053.5
Р83

Схвалено для використання у загальноосвітніх навчальних закладах
(лист Інституту модернізації змісту освіти
Міністерства освіти і науки України
від 27.03.2018 № 22.1/12-Г-181)

В. Д. Руденко

Р83 Інформатика : бази даних (модуль для учнів 10–11 класів, рівень стандарту) / В. Д. Руденко. — Харків: Вид-во «Ранок», 2019. — 112 с.

ISBN 978-617-09-4776-5




Зміст посібника відповідає вимогам навчальної програми вибірково-обов'язкового курсу з інформатики для учнів 10 (11) класу загальноосвітніх навчальних закладів (рівень стандарту) і призначений для вивчення вибіркового модуля «Бази даних».

Пропонований посібник складається із семи розділів і містить основні теоретичні відомості про бази даних, опис структури та порядок роботи в середовищі Access, алгоритми і приклади створення різних об'єктів за допомогою СУБД Access, завдання для самостійного виконання, практичні роботи.





Призначено для учнів 10–11 класів закладів загальної середньої освіти та вчителів інформатики.

УДК 004.65-053.5

Різнорівневі питання для перевірки знань і завдання для самостійного виконання відповідають рівням навчальних досягнень:

-  — початковий і середній рівні
-  — достатній рівень
-  — високий рівень

У тексті використано такі позначки:

-  — означення, висновок
-  — зверніть увагу
-  — завдання для виконання в парах або групах
-  — вправи для домашнього виконання



Разом дбаємо
про екологію та здоров'я

ISBN 978-617-09-4776-5

© Руденко В. Д., 2018
© ТОВ Видавництво «Ранок», 2019

Присвячується моєму другу і соратнику
Мірче Олександровичу Патланжоглу.

З повагою, автор

ПЕРЕДМОВА

Посібник відповідає вимогам навчальної програми вибірково-обов'язкового курсу з інформатики для учнів 10 (11) класу загально-освітніх навчальних закладів (рівень стандарту) і призначений для вивчення модуля «Бази даних». Видання розраховане на вивчення баз даних обсягом 35 годин і містить теоретичний матеріал, завдання для самостійного виконання, диференційовані за рівнем складності, та практичні роботи.

У процесі опанування матеріалу цього модуля потрібно пам'ятати, що база даних є однією з основних складових сучасних інформаційних систем, які бурхливо розвиваються й використовуються в багатьох сферах людської діяльності. Крім того, це досить складна система, що містить низку принципово нових для вас абстрактних понять. Для кращого усвідомлення змісту матеріалу необхідно кожне поняття ретельно розглядати на конкретних прикладах, наведених у посібнику.

Посібник складається із семи розділів. У *розділі 1* описано сутність баз даних, поняття реляційної моделі даних, порядок проектування баз даних та їх місце в інформаційних системах, а також основні відомості про систему управління базами даних Access 2016. У *розділі 2* розглянуто порядок створення структури таблиць і їх модифікації, правила встановлення ключового поля, індексації і зв'язування таблиць; стисло наведено порядок уведення, редагування і пошук даних у таблицях, методика сортування та фільтрації даних. У *розділі 3* наведено класифікацію запитів, на прикладах описано методика створення запитів різних типів. У *розділі 4* розглянуто форми, у *розділі 5* — звіти. Основну увагу приділено конструкторам для створення форм і звітів. У *розділі 6* описано призначення, основні поняття й можливості мови SQL, розглянуто створення найпростіших запитів мовою SQL, запитів з умовою і порядок їх групування. У *розділі 7* розглянуто сутність імпорту й експорту об'єктів із однієї бази даних Access 2016 в іншу.

Для теоретичного матеріалу, завдань для самостійного виконання і практичних робіт розроблено різні варіанти прикладів. Усі приклади об'єктів, розроблені в теоретичній частині на початку посібника, використовуються в усіх наступних розділах. Цей принцип поширюється й на завдання для самостійного виконання та практичні роботи. Тому уважно ставтеся до процесу збереження розроблених об'єктів упродовж усього часу вивчення баз даних.

Бажаємо успіхів у набутті навичок користувача бази даних.

1

ЗАГАЛЬНІ ВІДОМОСТІ ПРО БАЗИ ДАНИХ

1.1

ПОНЯТТЯ БАЗИ ДАНИХ

Термін *database* (база даних) з'явився на початку 60-х років ХХ ст. на симпозиумах фірми System Development Corporation (США), хоча самими базами даних людство користувалося здавна. Широкого розповсюдження термін набув у 1970-ті роки з розвитком ЕОМ.



База даних — це сховище даних різного типу про об'єкти та взаємозв'язки між ними певної предметної області, що зберігаються, оновлюються та опрацьовуються за встановленими правилами.

Базами даних (БД), наприклад, є статті з принципів будови та функціонування ОС Windows, укази Президента України, дані про випускників школи, відомості про наявність ліків у аптеці та їх ціни.

Предметною областю називають сферу застосування конкретної БД. Наприклад, предметною областю може бути будівельна фірма, школа, Міністерство освіти і науки, аеропорт, банк, поліклініка тощо.

Об'єкт предметної області — це те, про кого або про що зберігаються дані в БД. Якщо предметною областю є, наприклад, школа, то її об'єктами можуть бути: учні, вчителі, директор школи, кабінети. Об'єктами вищого закладу освіти можуть бути факультети, кафедри, викладачі, студенти, аудиторії, а об'єктами аптек — завідувач, провізори, постачальники ліків.

- ✓ Кожен об'єкт БД характеризується сукупністю властивостей (атрибутів).

Так, об'єкт АВТОМОБІЛЬ може мати такі властивості: модель, потужність двигуна, максимальна швидкість, вантажопідйомність, а об'єкт УЧЕНЬ — такі атрибути: прізвище, ім'я, рік народження, домашня адреса, школа, клас, зріст.

Далі об'єкти БД будемо позначати так: назва об'єкта великими літерами, а в круглих дужках — перелік його атрибутів, які відокремлюються один від одного комою. Наприклад, об'єкт ПОТЯГ можна позначити так: ПОТЯГ (номер потягу, станція відправлення, час відправлення, кінцева станція, час прибуття на кінцеву станцію).

За структурою даних БД поділяються на дві основні групи: документальні та фактографічні.

Документальні БД — це звичайні документи, наприклад, постанови Кабінету Міністрів, інструкції до ліків, юридичні документи, статті, накази директора школи й ін. Найчастіше із документальних БД отримується перелік документів або тексти документів, у яких слід шукати

відповідь на поставлене запитання. Пошук документа може здійснюватися за його назвою, ключовими словами або фрагментом тексту.

Фактографічні БД зберігаються за чітко встановленою формою. Наприклад, дані про автомобілі, що продаються в автомагазині, можуть зберігатися за такою формою (табл. 1).

Таблиця 1. ФОРМА ЗБЕРІГАННЯ ФАКТОГРАФІЧНОЇ БД

Марка	Двигун	Пробіг (км)	Колір	Ціна (грн)
BMW 316 Ti	бензиновий	190000	чорний	60000
Volkswagen Arteon	дизельний	100000	білий	85000

У фактографічних БД кожен атрибут об'єкта має певну сукупність значень, тобто елементів даних, які є найменшими неподільними одиницями даних.

Наприклад, атрибут *марка* в наведеному прикладі має значення BMW 316 Ti і Volkswagen Arteon, атрибут *двигун* — значення *бензиновий* і *дизельний*, а атрибут *колір* — значення *чорний* і *білий*.

База даних є однією з найважливіших складових сучасної інформаційної системи, побудованої на основі комп'ютерних систем і мереж. Робота з БД у цих системах здійснюється за допомогою спеціальної мови БД або програмного забезпечення, яке має назву **система управління базами даних (СУБД)**.



СУБД — це інструмент, призначений насамперед для створення структури БД, введення й оновлення даних, пошуку необхідних даних та їх опрацювання за певним алгоритмом.

Оскільки до БД може звертатися велика кількість користувачів, то важливою функцією СУБД є забезпечення цілісності та безпечності даних.

Окрім функцій, безпосередньо пов'язаних зі створенням і підтримкою БД, окремі СУБД виконують також функцію підтримки спеціалізованих мов програмування, які мають загальну назву «мови баз даних». Наприклад, СУБД Access 2016 підтримує мову запитів SQL.

Нині фактичним стандартом мови баз даних є мова SQL. Однак у деяких випадках доводиться користуватися й іншими мовами програмування, наприклад, мовою VBA. Професійні розробники БД засобами СУБД та іншими мовами програмування можуть розробляти прикладні програми, за допомогою яких користувач простим натисканням однієї кнопки може отримати з БД необхідні дані або виконати їх опрацювання за певним алгоритмом.

Наприклад, натисканням лише однієї кнопки можна обчислити суму реалізованих у супермаркеті конкретних назв товарів за добу, нарахувати заробітну платню працівникам фірми або отримати інформацію про наявність вільних місць у готелях міста Відня вартістю не дороже 200 євро на добу.

Отже, взаємодія користувача з БД може здійснюватися як засобами СУБД, так і за допомогою прикладних програм, що пояснюється схемою, зображеною на рис. 1.

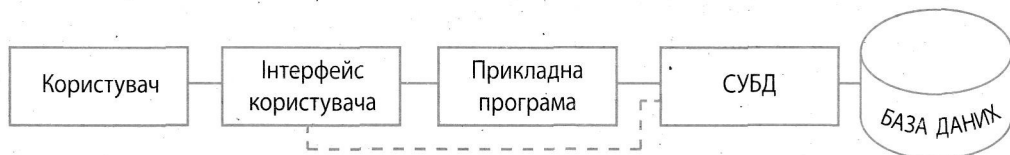


Рис. 1. Варіанти взаємодії користувача з БД

Сучасні БД мають величезні обсяги даних і зберігаються в комп'ютерних системах на жорстких магнітних дисках. Але користувач позбавлений необхідності знати тонкощі фізичного розміщення даних на таких пристроях. Ця функція повністю реалізується СУБД разом з операційною системою.

Важливою функцією СУБД є також керування транзакціями. **Транзакція** — це послідовність операцій над даними, яка сприймається СУБД як єдине ціле. Якщо всі операції цієї послідовності виконані успішно, то вважається, що й транзакція завершена успішно, й усі зміни даних, виконані за цією транзакцією, вносяться в зовнішню пам'ять. Але якщо хоча б одна операція послідовності завершена невдало, транзакція вважається невиконаною, і здійснюється відкат, тобто скасування змін у всіх даних, виконаних у процесі транзакції, і повернення БД до стану початку її виконання. Наприклад, якщо в системі продажу квитків на потяг із будь-яких причин не виконано замовлення на квиток, ніяких змін до БД про наявність квитків внесено не буде, тобто відбудеться відкат.

Ще однією важливою функцією СУБД є так звана **журналізація**, під якою розуміють уведення змін до БД. Перед тим як виконати необхідні зміни в БД, ці зміни заносять до спеціального журналу. Якщо відбудеться апаратний або програмний збій, то за допомогою архівної копії і журналу можна повністю відновити БД.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. ▲ Що називають предметною областю БД?
2. ▲ Наведіть приклади властивостей об'єкта *смайтфон*.
3. ▲ Як позначають об'єкти БД?
4. ▲ Як БД поділяються за структурою?
5. ✦ Які БД називають фактографічними?
6. ✦ Наведіть означення БД.
7. ★ Назвіть основні функції СУБД.
8. ★ Поясніть сутність транзакції.

РЕЛЯЦІЙНА МОДЕЛЬ ДАНИХ

Раніше вже було зазначено, що об'єкти предметної області характеризуються сукупністю атрибутів (властивостей) та їх значеннями. Одне значення атрибута називають елементарною одиницею даних. Наприклад, для об'єкта *автомобіль* його елементарними одиницями можуть бути: *марка* — Volkswagen, *двигун* — дизельний.

Таким чином, об'єкти БД характеризуються сукупністю елементарних одиниць даних, між якими мають бути встановлені однозначні зв'язки. Це означає, що основою будь-якої структури даних є відображення елементарної одиниці даних у вигляді трійки: <об'єкт, атрибут об'єкта, значення атрибута>, наприклад: <учень, прізвище, Костирко>; <учень, клас, 10>.

Отже, дані, що зберігаються в БД, мають певну логічну структуру, тобто описуються певною моделлю даних, яка підтримується відповідною СУБД.

Існують різні способи відображення зв'язків між даними, тобто існують різні моделі даних. Зараз є три класичні моделі даних: *ієрархічна*, *мережна* і *реляційна*.

Розвиваються й інші моделі, що засновані на класичних, наприклад, об'єктно-реляційна модель даних.

- ✓ Таким чином, модель даних визначає, як відбувається об'єднання даних у структури. Вона також визначає можливі операції над даними та обмеження на їх значення.

Ієрархічна й мережна моделі засновані на поняттях: рівень, вузол, зв'язок. Приклад їх структури і стислий опис сутності цих моделей зображено на рис. 1.



Рис. 1. Структури ієрархічної і мережної моделей даних

Із рис. 1 видно, що ієрархічна модель у цьому випадку містить три рівні об'єктів. На верхньому рівні розташовано головний об'єкт. На другому рівні містяться два вузли, на третьому — три вузли. Зв'язки між вузлами зображені стрілками.

Вузли верхнього рівня мають зв'язки з вузлами найближчого нижнього рівня. Мережна модель на рис. 1 містить два рівні (реально їх може бути скільки завгодно), на кожному з яких є два вузли. Звернемо увагу, що в мережній моделі кожен вузол може мати зв'язок із будь-яким іншим вузлом.

Основним недоліком ієрархічних і мережних БД є складність їх розроблення. Найпоширенішою нині є **реляційна модель даних**. Її основна перевага полягає в простоті розроблення БД і систем управління ними.

У реляційних моделях об'єкти та взаємозв'язки між даними представляються за допомогою відношень. Порядок розміщення рядків і стовпців у таблиці довільний. Таблиці в теорії БД називають відношеннями, рядки — записами, а стовпці — полями. Найпростіша БД містить одну таблицю, а складні — десятки й навіть сотні.

Нижче наведено приклад найпростішої реляційної БД, що містить лише одну таблицю УЧНІ (табл. 1).

Таблиця 1. УЧНІ

Номер	Прізвище	Дата народження	Адреса	Клас	Зріст
1	Колот А.І.	07.02.2002	Зоряна, 2, кв. 7	10	172
2	Федорчук Ю.А.	30.05.2003	Лісова, 5	9	165
3	Таранов С.О.	02.06.2003	Поштова, 3, кв. 9	9	174

Зазначимо, що не кожна таблиця може бути об'єктом БД. Щоб таблиця була об'єктом БД, необхідно виконати їх **нормалізацію**.

Сутність нормалізації полягає в тому, що таблиця має бути перетворена так, щоб вона відповідала таким основним вимогам:

- кожне поле повинно мати унікальне ім'я;
- усі поля мають бути однорідними. Це означає, що значення елементів одного поля можуть бути лише одного типу, наприклад, тільки числовими, тільки рядковими або іншими;
- у таблиці не може бути однакових записів, вони мають відрізнятися значеннями хоча б одного поля;
- у таблиці має бути унікальне поле або набір кількох полів, які однозначно ідентифікують записи. Такі поля називають **ключем**.

Ключове поле використовується для швидкого пошуку необхідних даних, а також для зв'язування даних із різних таблиць.

Ключ, який містить тільки одне поле, називають простим, а кілька полів — складним. У таблиці УЧНІ складним ключем можна вважати поля Прізвище і Дата народження, оскільки вони однозначно ідентифікують записи.

У таблиці може бути кілька ключів, але тільки один із них може бути первинним. Найкраще первинним ключем вибрати простий ключ. Бажано, щоб він мав цілочисловий тип. У такому випадку операції

опрацювання даних виконуватимуться швидше. У таблиці УЧНІ простим є поле з іменем Номер.

У таблиці часто використовують поле, яке називається лічильником. Він використовується для того, щоб зробити кожний запис унікальним. Окрім того, лічильник забезпечує нумерацію записів. Лічильником у таблиці УЧНІ є поле з іменем Номер.

Важливо усвідомити, що БД будь-якої складності можна створити на основі однієї таблиці. Але в такому випадку таблиця може містити сотні полів і тисячі записів. Розібратися і працювати з такою таблицею досить складно. Крім того, це може призвести до значного дублювання даних. Тому для кожного об'єкта розробляється власна таблиця. А щоб можна було одночасно отримати дані з кількох таблиць, слід установити зв'язки між ними.

Наприклад, для БД фірми в одній таблиці можуть зберігатися дані про співробітників фірми, у другій — дані про їх заробітну платню, у третій — відомості про постачальників продукції. Такий підхід спрощує подальшу модифікацію БД і має низку інших переваг.

Зв'язки можуть установлюватися між двома, трьома й більшою кількістю таблиць.

Для встановлення зв'язків між двома таблицями одна із них вважається **основною** (батьківською), а друга — **допоміжною** (дочірньою).

- ✓ В основній таблиці вибирають первинний ключ, а в допоміжній — зовнішній ключ.

Головна вимога до зовнішнього ключа така: у ньому не може бути даних, відсутніх у первинному ключі, інакше зв'язок буде некоректним. Але часто зовнішній ключ вибирають із тим самим іменем, що й первинний ключ основної таблиці.

Часто для забезпечення зв'язку між таблицями в допоміжну таблицю спеціально вводять поле із таким самим іменем, що й ім'я первинного ключа основної таблиці. У такому випадку деякі СУБД, наприклад, Access 2016, автоматично встановлюють зв'язок між цими таблицями. Якщо ж імена вказаних полів різні, користувач має сам установити зв'язок між ними.

Пояснимо сутність зв'язків між двома таблицями на прикладі.

Приклад 1

У БД будівельної компанії є дві таблиці — ПОСТАЧАЛЬНИКИ (табл. 2) і ТОВАРИ (табл. 3).

Таблиця 2. ПОСТАЧАЛЬНИКИ

Фірма	Директор	Телефон
РПЗ-1	Сопко А.І.	345-23-51
БУТ-5	Маслов В.М.	295-44-87
ДМК-2	Бондаренко К.О.	454-98-56

Таблиця 3. ТОВАРИ

Матеріал	Вага (тонни)	Фірма
Бетон	100	РПЗ-1
Бетон	120	БУТ-5
Бетон	200	ДМК-2
Цемент	50	БУТ-5

У таблиці ПОСТАЧАЛЬНИКИ первинним ключем є поле з іменем Фірма. У таблиці ТОВАРИ поле із цим іменем не може бути первинним ключем, бо в ньому повторюються назви фірм, але воно може бути зовнішнім ключем, оскільки його значення збігаються зі значеннями однойменного поля таблиці ПОСТАЧАЛЬНИКИ. Більше того, вони мають однакове ім'я. За даними цього поля можна встановити зв'язок між таблицями ПОСТАЧАЛЬНИКИ й ТОВАРИ. Якщо потрібно дізнатися телефон і прізвище директора фірми, яка постачає 120 тонн бетону і 50 тонн цементу, то слід із таблиці ТОВАРИ вибрати назву фірми БУТ-5 і за її назвою в таблиці ПОСТАЧАЛЬНИКИ знайти прізвище Маслов В. М., телефон 295-44-87.

Таким чином, зв'язки між таблицями дають змогу отримати дані з кількох таблиць. Окрім того, вони забезпечують цілісність даних у пов'язаних таблицях, якщо з деяких причин сталися зміни в одній таблиці.

Пояснимо сутність цілісності даних на прикладі вже розглянутих раніше таблиць.

Приклад*2

Припустимо, що між табл. 2 і 3 зв'язок не встановлено, з табл. 2 видалено запис, де директором фірми є Маслов В. М., а в табл. 3 усі дані є цілісними. Ця ситуація відображена в табл. 4.

Таблиця 4. ПОСТАЧАЛЬНИКИ

Фірма	Директор	Телефон
РПЗ-1	Сопко А. І.	345-23-51
ДМК-2	Бондаренко К. О.	454-98-56
ДМК-2	Бондаренко К. О.	454-98-56

У такому випадку стає невідомим телефон і прізвище директора фірми БУТ-5, яка постачає 120 тонн бетону і 50 тонн цементу. Вважається, що в такому випадку сталося порушення цілісності даних і ця ситуація має бути автоматично виявлена.

Між таблицями можуть існувати такі типи зв'язків:

- **один до одного** — будь-якому запису першої таблиці відповідає тільки один запис другої таблиці, і навпаки. Наприклад, одним автомобілем може керувати лише один водій і керувати цим автомобілем може тільки один водій;
- **один до багатьох** — будь-якому запису першої таблиці відповідає кілька записів другої таблиці, але будь-якому запису другої таблиці відповідає тільки один запис першої таблиці. Наприклад, учень може мати з предмета «Інформатика» багато оцінок, і кожна з цих оцінок належить тільки одному учневі;
- **багато до одного** — будь-якому запису першої таблиці відповідає тільки один запис другої таблиці, але будь-якому запису другої таблиці відповідає кілька записів першої таблиці. Наприклад, водій працює тільки на одному тролейбусі, але цим тролейбусом можуть керувати кілька водіїв;
- **багато до багатьох** — будь-якому запису першої таблиці відповідає кілька записів другої таблиці, й будь-якому запису другої таблиці відповідає кілька записів першої таблиці. Наприклад, учень Рябко навчається в кількох учителів, і кожен учитель працює з кількома учнями.

Найчастіше між таблицями реляційної БД існує зв'язок **один до багатьох**.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. ▲ Що називають елементарною одиницею даних у БД?
2. ▲ Які існують основні моделі даних у БД?
3. ▲ Поясніть сутність ієрархічної моделі даних.
4. ▲ Які типи зв'язків існують між таблицями?
5. ★ Поясніть сутність реляційної моделі даних.
6. ★ Які поля таблиць називають ключем?
7. ★ Які ключі існують в таблицях БД?
8. ★ Наведіть означення моделі БД.
9. ★ Назвіть основні вимоги до таблиць БД.
10. ★ Поясніть сутність зв'язування таблиць БД.
11. ★ У чому полягає забезпечення цілісності даних БД?

1.3

ПРОЕКТУВАННЯ БАЗ ДАНИХ

Проектування БД — це досить відповідальний етап, від якості виконання якого залежить ефективність її функціонування. Проект БД доцільно спочатку розробити на папері, а потім реалізувати на комп'ютері. Слід зважати на те, що від того, як раціонально вона спроектована на папері, суттєво залежать ефективність її функціонування та подальші трудовитрати на її вдосконалення безпосередньо на комп'ютері. У створений проект БД можна вносити зміни й у процесі її експлуатації. Більше того,

такі зміни неминучі. Але такі зміни не мають бути зумовлені помилками проектування.

- ✓ Проектування БД здійснюється на фізичному та логічному рівнях.

Кінцевий користувач зазвичай проектує БД на логічному рівні, що не має жодного відношення до фізичного проектування. Ця функція покладена на СУБД і операційні системи. Тому далі розглядається сутність тільки логічного проектування. Існують кілька методів логічного проектування БД, але їх сутність мало відрізняється.

Одним із засобів моделювання предметної області на етапі проектування БД є модель **сутність—зв'язок**, яку ще називають ER-моделлю. У ній застосовуються графічні засоби. Основними поняттями такої моделі є сутність, атрибут і зв'язок.



Сутність — це деякий об'єкт реального світу. Вона має екземпляри, які відрізняються один від одного значеннями атрибутів.

Атрибут — це властивість сутності. Наприклад, сутність **ХОЛОДИЛЬНИК** характеризується такими атрибутами, як назва, вага, ціна, потужність. Конкретний холодильник є екземпляром сутності **ХОЛОДИЛЬНИК**. Атрибут, який є унікальним, тобто однозначно визначає екземпляр сутності, називають ключем. У ER-діаграмах сутність зображується прямокутником, а зв'язки між сутностями — ромбом. Зв'язок фактично встановлює взаємодію між сутностями. В реляційних БД сутності відповідає таблиця, а екземпляру — запис.

Головне на етапі логічного проектуванні БД полягає у виконанні таких основних дій:

- детально проаналізувати сутність предметної області, для якої необхідно розробити БД, і визначити форми й типи документів, які необхідно отримувати із БД;
- визначити кількість, назву та структуру кожної таблиці;
- визначити вміст кожної таблиці;
- установити зв'язки між таблицями.

Розглянемо реалізацію цих дій — розробимо БД для предметної області Автобаза. Вважатимемо, що на основі аналізу структури та завдань автобази, які вона виконує, у БД достатньо мати три таблиці: **АВТОМОБІЛЬ**, **ВОДІЙ**, **МЕХАНІК**.

Для кожної з цих таблиць визначимо такі їх структури:

- **АВТОМОБІЛЬ** (Номер автомобіля, Модель автомобіля, Рік виготовлення, Вантажопідйомність, Потужність, Витрати пального, Максимальна швидкість);
- **ВОДІЙ** (Посвідчення водія, Прізвище, Рік народження, Номер автомобіля, Стаж водія, Код механіка);
- **МЕХАНІК** (Код механіка, Прізвище, Оклад, Домашня адреса, Телефон, Загальний стаж).

Базу даних предметної області Автобаза на основі ER-схеми можна зобразити так, як показано на рис. 1.



Рис. 1. Схема проекту БД Автобаз

У таблиці АВТОМОБІЛЬ наведено назви полів, які однозначно визначають технічні дані автомобілів, що є на обліку автобаз, у таблиці ВОДІЙ — назви полів про дані водіїв, які працюють на цій автобазі, а в таблиці МЕХАНІК — назви полів про дані механіків, що обслуговують закріплені за ними автомобілі.

У таблиці АВТОМОБІЛЬ ключовим полем може бути тільки поле Номер автомобіля, оскільки кожен автомобіль має унікальний номер. У таблиці ВОДІЙ поле Номер автомобіля може повторюватися, оскільки на одному й тому самому автомобілі можуть працювати кілька водіїв (робота за графіком, за змінами). Тому між таблицями АВТОМОБІЛЬ і ВОДІЙ існує зв'язок **один до багатьох** (на одному кінці цієї лінії розміщено значення <1>, а на другому — символ <Б>).

У таблиці ВОДІЙ ключовим полем є Посвідчення водія, оскільки лише це поле має унікальне значення. Решта, у тому числі поле Прізвище, можуть дублюватися (на базі можуть працювати водії з однаковими прізвищами, ім'ям та по-батькові). Поле Код механіка також може дублюватися, тому що один механік обслуговує, як правило, кілька автомобілів. Але це поле є унікальним у таблиці МЕХАНІК. Тому між таблицями ВОДІЙ і МЕХАНІК існує зв'язок **багато до одного**.

Для наочності зв'язки між таблицями можна зобразити схемою, приклад якої наведено на рис. 2.



Рис. 2. Схема зв'язків між таблицями

Наступним кроком проектування БД є визначення вмісту її таблиць. Зразок вмісту табл. 1 ВОДІЙ наведено далі.

Таблиця 1. ВОДІЙ

Посвідчення водія	Прізвище	Рік народження	Номер автомобіля	Стаж водія	Код механіка
СН2374	Сокіл А.С.	1990	АА2387КИ	6	23
РК1173	Корж В.М.	1980	АА8874КИ	14	24
СН7322	Рожко П.Т	1985	АА4492КИ	11	23
КН5475	Шрамко А.І.	1970	АА3395КИ	27	23

На цьому проектування БД на папері можна вважати завершеним. Далі робота продовжується на комп'ютері за допомогою обраної СУБД.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. На яких рівнях виконується проектування БД?
2. Із якою метою виконується проектування БД на папері?
3. У якому вигляді може бути поданий проект БД?
4. Які основні дії виконуються в процесі логічного проектування БД?
5. Наведіть приклад схеми проекту найпростішої БД.

1.4

БАЗИ ДАНИХ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

База даних набуває цінності в складі інформаційної системи (ІС) та є однією з найважливіших її складових. Поняття інформаційної системи доволі об'ємне, тому однозначного її тлумачення не існує.

Далі під **інформаційною системою** будемо розуміти програмно-апаратний комплекс для збереження великих БД, виконання над ними операцій, зумовлених специфікою предметної області, забезпечення зручного інтерфейсу для доступу до необхідних даних та їх опрацювання.

- ✓ Інформаційні системи класифікуються за багатьма ознаками, із яких однією із найголовніших є спосіб доступу користувачів до бази даних. За цією ознакою їх поділяють на автономні, групові, корпоративні, глобальні.

Автономні ІС — це системи, які функціонують на одному персональному комп'ютері, найчастіше не підключеному до комп'ютерної мережі. Користуються ними один або кілька користувачів по чергову. Бази даних для таких систем розробляються для невеликих аптек, відділів кадрів підприємств, домашніх інформаційно-довідкових систем тощо. Такі системи ще називають локальними або однокористувацькими. База

даних для них створюється і підтримується, як правило, лише засобами СУБД. Але інколи для зручності роботи користувача для них розробляється прикладна програма, яка спрощує й підвищує швидкість роботи з БД. Для розроблення та супроводу БД можуть використовуватися СУБД Access, dBase, Paradox тощо.

Групові ІС створюються, як правило, на базі локальних мереж у межах однієї установи, школи, невеликих фірм. Бази даних у таких системах зберігаються централізовано на одному комп'ютері (сервері) й використовуються колективно. Бази даних для них можуть створюватися та підтримуватися за допомогою, наприклад, таких СУБД: InterBase, Sybase, Access, MS SQL Server та ін.

Корпоративні ІС створюються у великих організаціях, підрозділи яких можуть бути віддалені на значні відстані. Вони можуть охоплювати кілька локальних мереж зі спеціальними серверами. Для розроблення та підтримки БД для них можуть використовуватися СУБД Oracle, DB2, MS SQL Server тощо.

Глобальні ІС функціонують у мережі Інтернет, робота якої заснована на використанні протоколу TCP/IP. Доступ користувачів до БД здійснюється зазвичай за допомогою оглядачів (browser).

Розглянемо стисло сутність інформаційних систем, які функціонують у локальних комп'ютерних мережах.



Інформаційні системи в локальних комп'ютерних мережах будуються за варіантами:

- файл-сервер;
- клієнт-сервер.

Спрощену схему файл-серверної архітектури зображено на рис. 1.

Централізована БД у цій архітектурі зберігається на комп'ютері-сервері. З робочих станцій (із комп'ютерів-клієнтів) посилаються запити на комп'ютер-сервер. У ньому відшукуються відповідні дані й у вигляді файлів без їх опрацювання пересилаються на робочі станції.

Опрацювання цих файлів здійснюється на робочих станціях. Файл-серверна архітектура потребує пересилання мережею великих, часто надлишкових обсягів даних, що є її недоліком. Окрім того, робочі станції повинні мати підвищену потужність для опрацювання отриманих даних.

Зрозуміло, що на робочих станціях можуть зберігатися й персональні (автономні) бази даних для кожного клієнта.



Рис. 1. Файл-серверна архітектура інформаційної системи

Спрощену структуру клієнт-серверної архітектури інформаційної системи зображено на рис. 2.

У клієнт-серверній архітектурі централізована БД також зберігається на комп'ютер-сервері. Персональні бази даних, як і в попередній архітектурі, також можуть зберігатися на робочих станціях. У відповідь на запити, що надходять із робочих станцій на комп'ютер-сервер, у ньому відшукуються необхідні дані та опрацьовуються за певним алгоритмом.



Рис. 2. Клієнт-серверна архітектура інформаційної системи

Опрацьовані дані передаються на робочі станції не у вигляді файлів даних, а як окремі записи, що суттєво знижує навантаження на локальну мережу.

✓ Отже, для створення якісних БД і кваліфікованої роботи з ними необхідно володіти програмним забезпеченням, яке називається системами керування базами даних. Нині існує значна кількість таких програм. Назви деяких із них уже було наведено раніше.

БД класифікуються за багатьма ознаками. До найголовніших можуть належати призначення, модель даних, спосіб доступу.

Спрощену схему класифікації СУБД зображено на рис. 3.

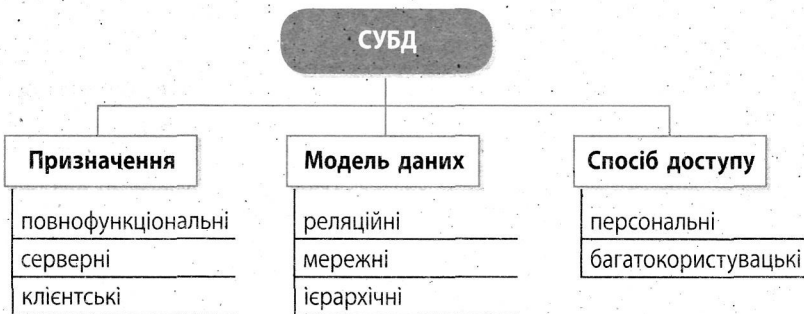


Рис. 3. Класифікація СУБД

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. ▲ Як ІС класифікуються за способом доступу до БД?
2. ▲ Поясніть сутність автономних ІС.
3. ▲ Які типи ІС використовуються в локальних мережах?
4. ✦ Наведіть означення ІС.
5. ✦ Яка різниця між клієнт-серверною і файл-серверною ІС?
6. ★ Поясніть сутність файл-серверної ІС.
7. ★ За якими основними ознаками класифікують СУБД?

ОСНОВНІ ВІДОМОСТІ ПРО СУБД ACCESS 2016

Система управління базами даних Access 2016 входить до складу пакета Microsoft Office і призначена для створення та підтримки роботи з реляційними БД. Access 2016 функціонує під керуванням ОС Windows і потребує таких системних вимог: бажано, щоб процесор мав частоту не менше 800 МГц, обсяг оперативної пам'яті був не менше 512 МБ, і вільний обсяг пам'яті на жорсткому диску — не менше 2 ГБ.

Основними об'єктами СУБД Access 2016 є:

- *таблиця* — призначена для збереження даних;
- *запит* — забезпечує пошук необхідних даних у таблицях, їх сортування та опрацювання;
- *форма* — основна функція полягає в наданні користувачеві зручних засобів для введення даних у таблицю;
- *звіти* — призначені для виведення на екран або папір даних у зручному для користувача вигляді.

Бази даних у середовищі Access 2016 можна створювати «з нуля», тобто повністю самостійно створити нову базу. Можна також скористатися шаблоном, які має Access. На основі вибраного шаблону можна швидко створити власну базу даних. Якщо наявних шаблонів не вистачає, їх можна знайти в Інтернеті. Але для якісного оволодіння методикою створення та супроводу БД початківцю доцільно спочатку починати створювати нові бази даних. Тому далі описується саме такий підхід до створення БД.

Запуск системи Access 2016 можна здійснити стандартними способами, що передбачені в ОС Windows. Після запуску системи на екрані монітора з'явиться її стартове вікно, зображене на рис. 1.

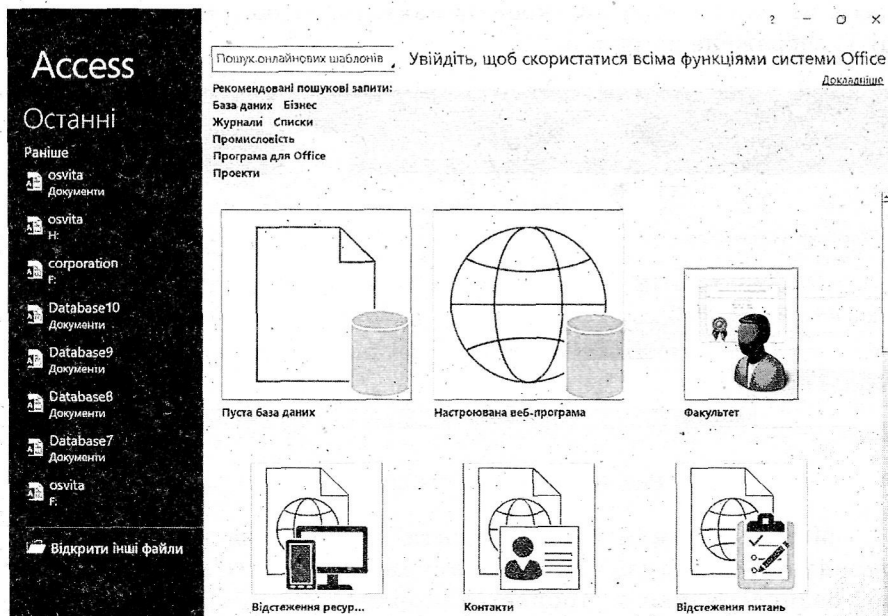


Рис. 1. Стартове вікно Access 2016

У лівій ділянці вікна відображено імена БД, із якими користувач працював останнім часом, у правій частині — піктограми шаблонів і піктограма порожньої бази даних. Поле пошуку призначене для пошуку шаблонів в Інтернеті.

Для створення нової власної БД слід клацнути піктограму Пуста база даних. Відкриється вікно, зображене на рис. 2.

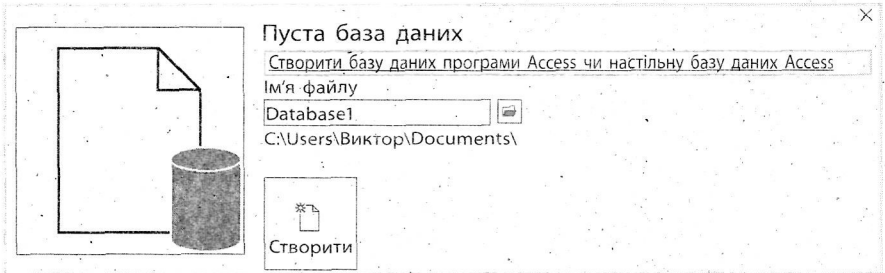


Рис. 2. Вікно для створення нової бази даних

У рядок Ім'я файлу вводимо ім'я майбутньої БД, наприклад, atb, і натиснемо кнопку (Знайти розташування для бази даних), що розташована праворуч від цього рядка. Відкриється вікно Створення бази даних, у якому вибираємо місце збереження файла БД, наприклад, диск F; далі натиснемо кнопку ОК, а потім у вікні, зображеному на рис. 2, — кнопку Створити.

У результаті цих дій файл БД буде зареєстровано в кореневому каталозі диска F; а на екрані з'явиться вікно для створення таблиці 1, зображене на рис. 3.

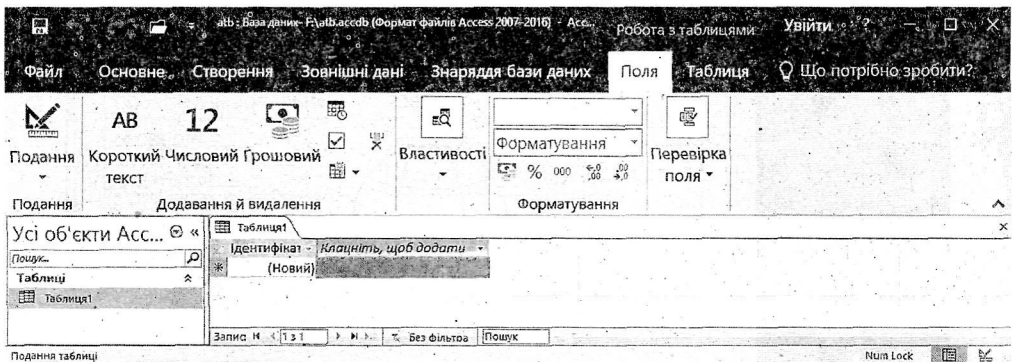


Рис. 3. Вікно для створення таблиці

Меню, команди й інструменти цього вікна розташовані на горизонтальній смужці. Їх призначення буде розглядатися поступово, безпосередньо в процесі використання. Розглянемо лише ті, які потрібні на цьому етапі. У верхній частині стрічки розміщені

вкладки Файл, Основне, Створення й інші. Зміст команд та елементів керування, які відображені на стрічці, залежить від того, яка в цей момент відкрита вкладка. У вікні, зображеному на рис. 3, відкритою є вкладка Поля. Команди й елементи керування цієї вкладки розміщені під вкладками. Вони згруповані в три розділи: Подання, Додавання й видалення, Форматування. Якщо відкрити іншу вкладку, наприклад, Основне, з'являться нові елементи й команди, які будуть згруповані в нові розділи.

Вкладка Основне містить команди й елементи керування, що найчастіше використовуються в процесі роботи з БД. Тут, зокрема, містяться команди для роботи з буфером обміну, форматування тексту, сортування та фільтрації даних тощо.

Команди вкладки Створення використовуються для створення таблиць, запитів, форм та інших об'єктів бази даних.

Команди вкладки Зовнішні дані призначенні для експортування й імпортування даних.

Вкладка Знаряддя бази даних містить команди встановлення зв'язків між таблицями, аналізу й переміщення даних між програмами й інші.

Ліва частина екрана з назвою Усі об'єкти Access — це панель переходів, де відображаються назви всіх створених об'єктів, між якими можна здійснювати перехід простим натисканням назв цих об'єктів.

Праворуч від панелі переходів розташована область редагування, у якій можуть одночасно відображатися таблиці, запити й інші об'єкти БД.

Закрити базу даних можна за допомогою команди Закрити, що міститься на вкладці Файл, а відкрити вже створену БД — за допомогою кнопки Відкрити на панелі швидкого доступу.

? ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1. ▲ Назвіть основні об'єкти Access 2016.
2. ▲ Що розміщено в області Усі об'єкти Access?
3. ▲ Як можна відкрити створену БД?
4. ★ Поясніть сутність створення бази даних «із нуля».
5. ★ Які основні дії виконуються на вкладці Створення?
6. ★ Назвіть основні системні вимоги для Access 2016.

СТВОРЕННЯ Й УВЕДЕННЯ СТРУКТУРИ ТАБЛИЦЬ

Таблиці в середовищі Access 2016 можна створювати різними способами, наприклад, шляхом імпортування зовнішніх даних, з використанням сайта SharePoint, із використанням веб-служби. Але найчастіше таблиці створюються самим розробником бази даних від їх проектування на папері до створення електронних працездатних таблиць. Саме такий варіант розглянемо далі.

Розробці таблиць передують детальний аналіз предметної області, визначення вимог, змісту та структури документів, які необхідно отримати. На основі цього слід визначити кількість таблиць, назви, структури та вміст кожної таблиці. Далі методичку розроблення та опрацювання таблиць розглядатимемо на прикладі предметної області Магазини.

Припустимо, що БД магазинів АТБ має містити дві таблиці: табл. 1 МАГАЗИНИ і табл. 2. КАДРИ. Визначимо їх структури і вміст.

Таблиця 1. МАГАЗИНИ

Номер магазину	Адреса	Директор	Телефон	Працівників
21	вул. Паркова, 33	Коцюба П.М.	234-54-63	20
31	вул. Печерська, 21	Борзов А.С.	234-22-98	13
6	вул. Річкова, 24	Середа К.М.	234-67-92	15

Таблиця 2. КАДРИ

Справа	Прізвище	Посада	Рік народження	Освіта	Стаж	Оклад	Номер магазину
105	Сокіл Т.Л.	касир	1960	середня	27	3500	6
132	Таран В.Д.	диспетчер	1973	вища	15	4000	31
120	Рябко Р.П.	експерт	1981	вища	8	4200	21
111	Семко М.М.	диспетчер	1970	середня	16	4000	21
115	Горошко Ф.Р.	диспетчер	1975	середня спеціальна	17	4000	31
116	Раков Г.П.	аналітик	1965	вища	19	4500	21
109	Шрамко Т.Л.	диспетчер	1961	середня	24	4000	6

Головною вважатимемо таблицю МАГАЗИНИ, а допоміжною — КАДРИ.

- ✓ У середовищі Access 2016 існують такі інструментарії створення таблиць, як Конструктор таблиць, Майстер таблиць, Режим таблиць.

Найпотужнішим і універсальним є Конструктор таблиць, тому далі розглядається цей спосіб. Інші способи простіші, й ними можна оволодіти самостійно.

Після визначення структури таблиць слід визначити типи полів з урахуванням тих типів, з якими може працювати Access.

У середовищі Access 2016 використовуються типи даних, перелік яких наведено на рис. 1.

Кожен із наведених типів даних має власний набір властивостей. Деякі з них є унікальними, тобто містяться лише в одному конкретному типі даних, а окремі є загальними, що містяться в різних типах даних.

Далі використовуватимемо здебільшого такі типи даних: *Короткий текст*, *Довгий текст*, *Число*, *Дата й час*.

Тип *Короткий текст* — це послідовність символів довжиною від 0 до 255, а тип *Довгий текст* — це послідовність символів до 65536.

У полі *Число* зберігаються числа. Розмір і конкретний тип чисел визначаються значенням властивості *Розмір поля*.

У полях типу *Дата й час* зберігаються дата й час різних форматів.

Для зручності уведення структури таблиць у комп'ютер доцільно для розроблених таблиць МАГАЗИНИ та КАДРИ розробити на папері додаткові таблиці з іменами полів, типом даних та їх описом. Їх вміст наведено в табл. 3 і 4.

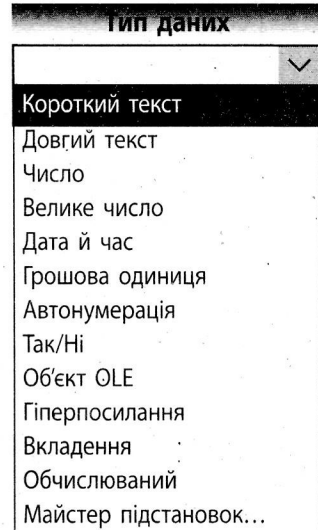


Рис. 1. Типи даних Access 2016

Таблиця 3. ТИПИ ТА ОПИС ПОЛІВ ТАБЛИЦІ «МАГАЗИНИ»

Ім'я поля	Тип даних	Опис	Властивості
Номер магазину	Число	Первинний ключ	
Адреса	Короткий текст		50
Директор	Короткий текст		40
Телефон	Короткий текст		20
Працівників	Число	Станом на 1 липня	

Таблиця 4. ТИПИ ТА ОПИС ПОЛІВ ТАБЛИЦІ «КАДРИ»

Ім'я поля	Тип даних	Опис	Властивості
Справа	Число	Первинний ключ	
Прізвище	Короткий текст		40
Посада	Короткий текст		30
Рік народження	Число		
Освіта	Короткий текст		30
Стаж	Число	Станом на 1 січня	
Оклад	Число		
Номер магазину	Число		

На цьому підготовку таблиць на папері завершено. Далі введемо структури таблиць у комп'ютер у режимі Конструктора. Для цього виконуємо такі дії.

1. Завантажимо систему Access 2016. Відкриваємо раніше створену базу даних з іменем atb. Для цього в стартовому вікні системи в області Останні натиснемо клавішу миші на імені atb.
2. У вікні, що відкриється, активуємо вкладку Створення. Відкриється вікно зі стрічкою, зображеною на рис. 2.

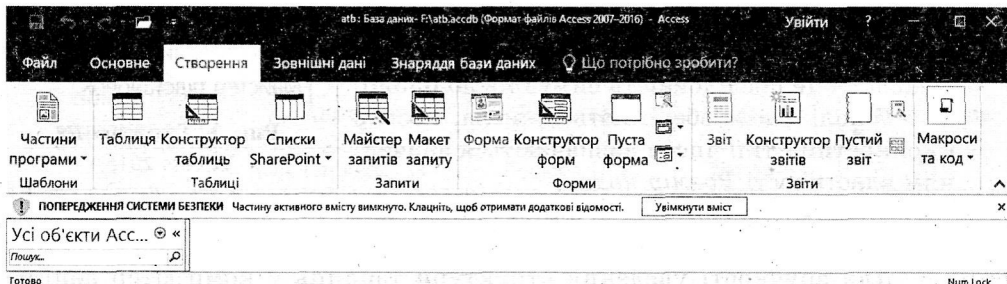


Рис. 2. Вікно Access 2016 з активованою вкладкою Створення

3. На цій стрічці натискаємо кнопку Конструктор таблиць. У результаті до бази даних додається порожня таблиця (рис. 3).

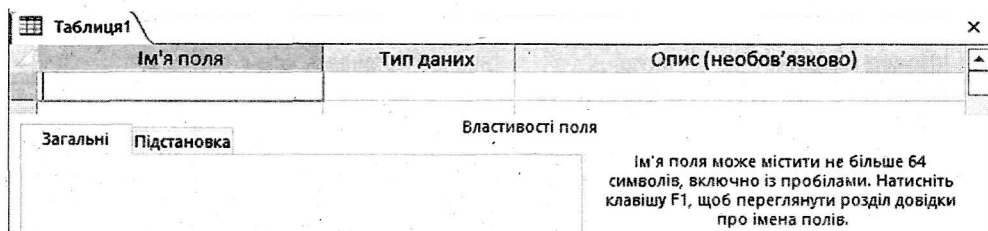


Рис. 3. Порожня таблиця в режимі Конструктора

Звернемо увагу на те, що у вікні, у якому відкрита ця таблиця, активованою має бути вкладка Конструктор (рис. 4).

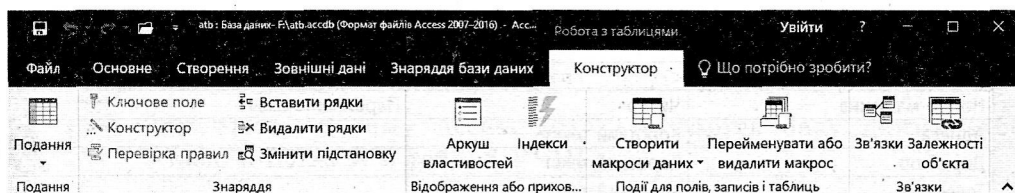


Рис. 4. Стрічка вікна Access 2016 з активованою вкладкою Конструктор

4. Уведемо ім'я поля в таблицю, зображену на рис. 3. Уведення здійснюється стандартним способом: необхідно встановити курсор миші у відповідну клітинку таблиці й за допомогою клавіатури ввести певні імена.
5. Далі слід ввести типи полів. Їх краще не вводити з клавіатури, а вибирати зі списку типів, що перелічені на рис. 1. Для відкриття цього списку необхідно в певній клітинці поля Тип даних клацнути кнопку Прапорець. Потім у цьому списку слід вибрати необхідний тип. Одразу після вибору типу даних необхідно встановити його властивість.

Перелік властивостей наведено під назвами полів і подано на рис. 5.

Опис полів не обов'язковий. Його використовують для вказування призначення поля й припустимих його значень.

Властивості поля	
Загальні	Підстановка
Розмір поля	40
Формат:	
Маска вводу	
Підпис	
Значення за замовчуванням	
Правило перевірки	
Текст перевірки	
Обов'язково	Ні
Дозволити нульову довжину	Так
Індексовано	Ні
Стискання Юнікод	Так
Режим редактора ІМЕ	Без елемента керування
Режим речення редактора	Немає
Вирівнювання тексту	Загальне

Рис. 5. Властивості полів

6. Уведемо у відкриту на екрані таблицю (рис. 3) усі дані з табл. 3 («Типи та опис полів таблиці МАГАЗИНИ»). Таблиця набуде вмісту, зображеного на рис. 6.

МАГАЗИНИ		
Ім'я поля	Тип даних	Опис (необов'язково)
Номер магазину	Число	Первинний ключ
Адреса	Короткий текст	
Директор	Короткий текст	
Телефон	Короткий текст	
Працівників	Число	станом на 1 липня

Рис. 6. Структура таблиці в режимі Конструктора

7. Збережемо таблицю з іменем МАГАЗИНИ. Для цього слід на панелі швидкого доступу натиснути кнопку Зберегти (або скористатися клавішами Ctrl+S). Відкриється віконце (рис. 7), у яке вводимо ім'я МАГАЗИНИ, й натискаємо кнопку ОК.
8. У результаті на екран буде виведено попередження (рис. 8).

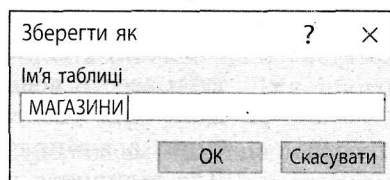


Рис. 7. Віконце для збереження таблиці

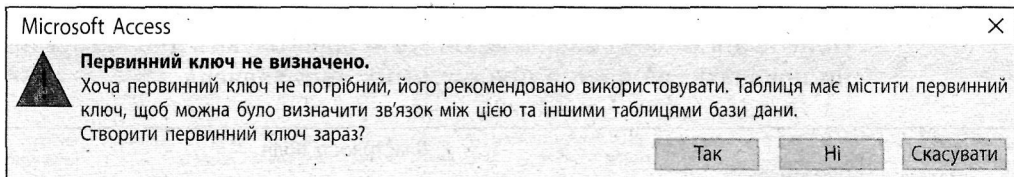


Рис. 8. Вікно попередження про невизначений первинний ключ

Натискання кнопки Так у цьому вікні зумовить створення первинного ключа типу Лічильник. На цьому етапі не обов'язково встановлювати первинний ключ, тому натиснемо кнопку Ні. Таблиця буде збережена, а її ім'я з'явиться в області переходів.

9. Закриваємо таблицю МАГАЗИНИ, для чого відкриємо її контекстне меню та виконаємо команду Закрити.
10. У такому самому порядку введемо та збережемо структуру таблиці КАДРИ. Після збереження закриваємо таблицю.

? ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. Які існують способи створення таблиць в Access 2016?
2. Із якою метою таблиці бажано створювати спочатку на папері?
3. Які дії виконуються для зберігання структури таблиці?
4. Що розуміють під терміном «структура таблиці»?
5. Чому найчастіше застосовується конструктор створення таблиць?

6. ★ Назвіть основні типи даних таблиць Access 2016.
7. ★ Поясніть порядок уведення структури таблиць в Access 2016.
8. ★ Поясніть порядок уведення типу даних поля та його властивостей.



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ



1. ★ Запустіть систему Access 2016 і створіть базу даних skola з табл. 5 КЛАСИ й табл. 6 УЧНІ.

Таблиця 5. КЛАСИ

Клас	Учні	Класний керівник	Займаються спортом
9	27	Зотов А.М.	5
10	24	Дерев'яно Н.С.	11
11	25	Терещенко Б.В.	14

Таблиця 6. УЧНІ

Прізвище	Адреса	Дата на- родження	Зріст	Улюблений предмет	Інформа- тика	Історія	Клас
Ларін Л.К.	вул. Лугова, 5	5.06.2002	163	математика	11	8	9
Лобов С.П.	вул. Сонячна, 7	9.08.2002	168	географія	9	9	9
Костенко В.С.	вул. Річкова, 4	13.09.2001	170	фізика	10	10	10
Рамко Б.В.	вул. Лугова, 9	15.11.2001	169	інформатика	11	8	10
Пека П.О.	вул. Сонячна, 7	20.12.2001	167	інформатика	12	9	10
Сергін В.В.	вул. Лісна, 5	3.03.2002	171	історія	8	11	10
Хижа Р.А.	вул. Лугова, 2	4.04.2002	172	історія	9	12	11
Собко О.К.	вул. Вишнева, 5	9.10.2001	175	інформатика	11	8	11
Настін К.Б.	вул. Лісна, 10	6.11.2001	171	географія	10	10	11



2. ★ Розробіть на папері структури таблиць КЛАСИ й УЧНІ. Типи даних полів і їх властивості виберіть самостійно.
 3. ★ Уведіть і збережіть структури таблиць КЛАСИ й УЧНІ.
- ✓ Із цією БД вам доведеться працювати протягом усього періоду їх вивчення, обережно працюйте з ними й не порушуйте їх структури.

2.2

МОДИФІКАЦІЯ СТРУКТУРИ ТАБЛИЦЬ

Створену структуру таблиць можна модифікувати, а також налаштувати її властивості, у тому числі можна визначити зовнішній вигляд і загальні характеристики таблиці. Властивості таблиць устанавлюються у вікні Аркуш властивостей, яке для відкритої таблиці

доступне на вкладці Конструктор. У цьому вікні є десятки назв властивостей таблиць. Аналогічне вікно властивостей мають й інші об'єкти бази даних (запити, форми, звіти). На цьому етапі вивчення БД будемо користуватися тими властивостями, які пропонуються за замовчуванням. Порядок установлення властивостей об'єктів Access 2016 буде розглянуто на прикладі форм і звітів.

Таблицю можна модифікувати на будь-якому етапі створення бази даних. Але найкраще це робити до встановлення зв'язків між таблицями й уведення до них даних.

У процесі модифікування структури таблиць можна виконувати деякі операції (рис. 1).



Рис. 1. Модифікування структури таблиць

- ✓ Існують різні способи **видалення полів** зі збереженої структури таблиць. Розглянемо один із найпростіших способів видалення поля на прикладі таблиці МАГАЗИНИ. В області Усі об'єкти відкриємо контекстне меню цієї таблиці, у якому виконаємо команду Конструктор. Відкриється таблиця МАГАЗИНИ в режимі Конструктор. Далі слід відкрити контекстне меню поля, яке необхідно видалити, наприклад, поля Телефон, і виконати команду Видалити рядки. Після цього модифіковану таблицю потрібно зберегти. Якщо потім відкрити цю таблицю, то поля Телефон у ній не буде.

Якщо після видалення поля одразу з'ясується, що видалення було помилкове і його необхідно відновити, то на панелі швидкого доступу слід натиснути кнопку Скасувати. Однак, якщо після видалення поля були виконані будь-які операції, скористатися цією командою для відновлення неможливо. Якщо робиться спроба видалити поле, яке містить дані, то буде видано попередження про те, що його дані будуть утрачені.

Слід пам'ятати, що в пов'язаних таблицях поля, які є первинним або зовнішнім ключем, неможливо видалити, поки не буде видалено зв'язок між таблицями.

Вставимо тепер поле Телефон у таблицю МАГАЗИНИ на його попереднє місце. Скористаємося, наприклад, таким способом. Відкриємо таблицю в режимі Конструктор. Установимо курсор на імені того поля, перед яким необхідно встановити поле (у цьому випадку на поле

Працівників), і відкриємо його контекстне меню. У цьому меню натиснемо кнопку Додати рядки. Відкриється вікно, зображене на рис. 2.

МАГАЗИНИ		
Імя поля	Тип даних	Опис (необов'язково)
Номер магазину	Число	Первинний ключ
Адреса	Короткий текст	
Директор	Короткий текст	
Працівників	Число	станом на 1 липня

Рис. 2. Вікно для вставлення нового поля

Уводимо в порожній рядок дані про поле Телефон і зберігаємо таблицю.

- ✓ **Зміна порядку розміщення полів** у таблиці здійснюється стандартними способами переміщення об'єктів у системі Windows.

Для цього необхідно виконати такі дії.

1. Відкриваємо таблицю в режимі Конструктор.
2. Установлюємо курсор на селекторі (селекторним називають перше (ліве) поле таблиці) того поля, яке потрібно перемістити, натискаємо кнопку миші та переміщаємо її у потрібну позицію.
3. Відпускаємо кнопку миші.

Перемістимо, наприклад, поле Адреса таблиці МАГАЗИНИ після поля Телефон, збережемо і закриємо таблицю. Після цього відкриємо її у режимі Таблиця. Вона буде мати вигляд, зображений на рис. 3. Аналогічно до описаного порядку відновимо попередній порядок розміщення полів у цій таблиці й збережемо її.

МАГАЗИНИ					
	Номер магазину ▼	Директор ▼	Телефон ▼	Адреса ▼	Працівників ▼
*	0				0

Рис. 3. Таблиця МАГАЗИНИ після переміщення поля

Зміну розмірів полів можна виконувати для текстових і числових типів. Для цього необхідно розмістити курсор на властивості Розмір поля і встановити нове значення.

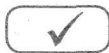
Для **зміни імені поля** достатньо встановити курсор на його імені й увести нове. Але змінювати імена полів слід лише в разі гострої необхідності, особливо для таблиць, між якими вже встановлені зв'язки. Якщо таблиці вже пов'язані, то після зміни імен полів необхідно виконати автозаміну, але вона не завжди дає позитивний результат, і деякі операції доводиться виконувати вручну.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. Які операції з модифікації таблиць можна виконати?
2. Як можна видалити поле із таблиці?

3. ✦ Якими способами можна відновити видалене з таблиці поле?
4. ✦ Як можна вставити нове поле в таблицю?
5. ✦ Як можна змінити ім'я поля?
6. ★ Поясніть порядок зміни розміщення полів.



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ



1. ✦ Відкрийте БД skola. Видаліть із таблиці УЧНІ поле Адреса. Збережіть таблицю. Відновіть поле Адреса в тому самому місці таблиці та збережіть її.

2. ✦ У таблиці УЧНІ поміняйте місцями поля Адреса й Історія. Збережіть таблицю. Переконайтеся, що позиції полів змінилися. Відновіть попередній порядок розміщення полів і збережіть таблицю.



3. ✦ Змініть у таблиці УЧНІ розмір полів Адреса та Дата народження. Переконайтеся, що зміни в таблицю внесені. Відновіть попередні розміри цих полів.

4. ✦ Замініть у таблиці УЧНІ ім'я поля Адреса на інше. Переконайтеся в правильності виконаної операції. Відновіть попереднє ім'я поля і збережіть таблицю.

2.3

КЛЮЧОВІ ПОЛЯ, ІНДЕКСИ, ЗВ'ЯЗУВАННЯ ТАБЛИЦЬ

Згадаємо, що кожна таблиця повинна мати ключове поле, тобто поле, значення якого не повторюються в жодному іншому записі. Таблиця може мати кілька ключових полів, але використовується лише одне з них, яке називають **первинним ключем**.

Найчастіше первинний ключ складається з одного поля й у ролі первинного ключа застосовується поле типу Лічильник. Якщо в ролі первинного ключа використовуються два і більше полів, його називають складним. Наприклад, у таблиці КАДРИ поле Прізвище не може бути первинним ключем, оскільки в мережі цих магазинів може бути працівник із таким самим прізвищем. А спільно поля Прізвище та Рік народження можна вважати таким ключем, тому що вони, імовірно, не дублюються. Але складні ключі бажано не використовувати як первинний ключ, оскільки в цьому випадку ускладнюється процес роботи з БД.

✓ Для створення первинного ключа слід відкрити таблицю в режимі конструктора, виділити поле, яке використовується як первинний ключ, і натиснути кнопку Ключове поле, що міститься в розділі Знаряддя вкладки Конструктор.

Первинний ключ можна створити також за допомогою контекстного меню певного поля, у якому необхідно виконати команду Ключове поле. Для цього таблиця відкривається в режимі конструктора. Скористаємося

цим способом і в таблиці МАГАЗИНИ визначимо як первинний ключ поле Номер магазину. Поряд із його назвою з'явиться малюнок ключа (рис. 1). Далі збережемо цю таблицю.

МАГАЗИНИ			
№	Ім'я поля	Тип даних	Опис (необов'язково)
1	Номер магазину	Число	Первинний ключ
	Адреса	Короткий текст	
	Директор	Короткий текст	
	Телефон	Короткий текст	
	Працівників	Число	станом на 1 липня

Рис. 1. Структура таблиці МАГАЗИНИ з ключовим полем

Якщо деяке поле в процесі створення структури таблиці оголошено типу Автономерація (рис. 1 на с. 21), тобто типу Лічильник, то воно за замовчуванням стає ключовим. Його можна також додати в таблицю, навіть у тому випадку, якщо необхідність у ньому відсутня. Поле типу Лічильник обов'язково встановлюється в тому разі, якщо ключ у таблиці взагалі визначити неможливо.

Розглянемо тепер сутність і порядок **індексування таблиць**.

Індексування — це процес створення додаткових таблиць для певного поля.

Ці таблиці зазвичай містять тільки одне поле (їх називають простими індексними таблицями), у якому зберігаються вказівники на певні записи таблиці. За допомогою вказівників визначають порядок розміщення записів, упорядкованих за значенням цього поля.

Наприклад, індексна таблиця для поля Директор таблиці МАГАЗИНИ (табл. 1 на с. 20) буде мати такі значення (рис. 2):

2
1
3

Рис. 2. Індексна таблиця для поля Директор таблиці МАГАЗИНИ

Цифра 2 у першому рядку означає, що першим за алфавітом у таблиці є друге прізвище (Борзов А. С.), цифра 1 — що другим за алфавітом є перше прізвище (Коцюба П. М.), а цифра 3 — що третім за алфавітом є третє прізвище в таблиці (Середа К. М.). Індексних таблиць для конкретної таблиці бази даних може бути кілька, наприклад, за кількістю працівників, за номерами магазинів тощо.

Головне призначення індексних таблиць — підвищення швидкості пошуку необхідних даних (інколи вона може вп'ятеро пришвидшитися). Щоб знайти деякий запис у таблиці, в Access 2016 спочатку знаходять його положення в індексі, потім вибирають із нього місце запису в таблиці, яка використовується для пошуку даних. Поля, значення

яких змінюються часто, індексувати недоцільно. Для однієї таблиці бажано мати не більше 5–6 індексних таблиць.

Зазначимо, що первинний ключ завжди індексований. За замовчуванням записи таблиці виводяться відсортованими за його значеннями. У процесі введення даних у таблицю обов'язково перевіряється значення первинного ключа на дублювання. Якщо значення дублюється, введення запису блокується.

Значення первинного ключа типу Лічильник у процесі введення даних формується автоматично.

Створимо, наприклад, просту індексну таблицю для полів Директор і Працівників таблиці МАГАЗИНИ.

Для цього відкриваємо таблицю в режимі Конструктора, вибираємо поле Директор і в розділі Властивості поля в рядку індексовано вмикаємо перемикач Так (Без повторень), оскільки мало ймовірно, що в цій мережі магазинів будь-які два директори з однаковим прізвищем. Для поля Працівників умикаємо перемикач Так (Повторення дозволені), тому що в магазинах можлива однакова кількість працівників.

- ✓ Для зв'язування двох таблиць одна з них вважається *головною*, а друга — *допоміжною*. Первинний ключ головної таблиці зв'язується з зовнішнім ключем другої. Головна вимога до ключів така: значення зовнішнього ключа мають збігатися зі значеннями первинного ключа головної таблиці. Імена цих ключів можуть бути різними, але якщо їхні імена однакові, то процес зв'язування таблиць буде простішим.

Часто первинний ключ таблиці штучно вводять у другу таблицю саме з метою їх зв'язування. Але цей ключ не є первинним ключем другої таблиці, оскільки його значення можуть повторюватися.

Наприклад, поле Магазин є первинним ключем таблиці МАГАЗИНИ, а в таблиці КАДРИ це поле є зовнішнім ключем, тому що в ній значення цього поля дублюються.

Розглянемо порядок створення зв'язку в таблицях МАГАЗИНИ і КАДРИ.

1. Завантажимо БД atb і у вікні, що відкриється, активуємо вкладку Знаряддя бази даних. Далі натиснемо кнопку Зв'язки. Відкриється вікно Відображення таблиці (рис. 3).

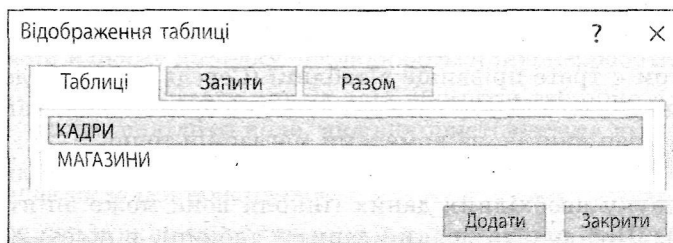


Рис. 3. Вікно з переліком таблиць БД atb

- У цьому вікні вибираємо таблиці, які необхідно зв'язати (у цьому випадку обидві таблиці), і натискаємо кнопку Додати. На екрані з'являться ці таблиці з іменами їх полів.
- Установлюємо курсор на первинному ключі таблиці МАГАЗИНИ, натискаємо кнопку миші, не відпускаючи її, переміщуємо курсор у поле зовнішнього ключа й відпускаємо кнопку. У результаті відкриється вікно Редагування зв'язків (рис. 4).

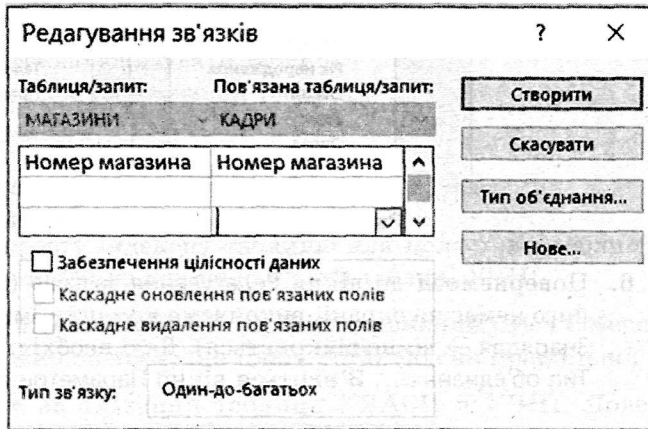


Рис. 4. Вікно Редагування зв'язків

- Умикаємо прапорець Забезпечення цілісності даних. Після цього стануть доступними прапорці Каскадне оновлення пов'язаних полів і Каскадне видалення пов'язаних полів. Умикаємо останній прапорець.

Увімкнення прапорця Забезпечення цілісності даних дає змогу зберегти цілісність даних. Якщо він вимкнений, то в таблиці можна додавати нові записи, змінювати ключові поля й видаляти пов'язані записи без попередження про порушення цілісності.

Сутність каскадного оновлення пов'язаних полів полягає в тому, що за будь-якої зміни первинного ключа в головній таблиці автоматично оновиться значення відповідного поля в усіх зв'язаних таблицях.

Сутність каскадного видалення пов'язаних полів полягає в тому, що під час видалення будь-якого запису з головної таблиці автоматично видаляються зв'язані записи в пов'язаній таблиці. Таким чином, каскадне оновлення та каскадне видалення прискорюють роботу з БД і сприяють підвищенню надійності її функціонування.

- У вікні Редагування зв'язків слід натиснути кнопку Створити, у результаті чого у вікні Зв'язки з'явиться лінія зв'язку між певними

полями таблиць (рис. 5). Далі зберігаємо базу даних. У вікні Редагування зв'язків можна ліквідувати встановлений зв'язок за допомогою кнопки Скасувати. Потім за допомогою кнопки Створити можна відкрити нове вікно й установити зв'язок заново.

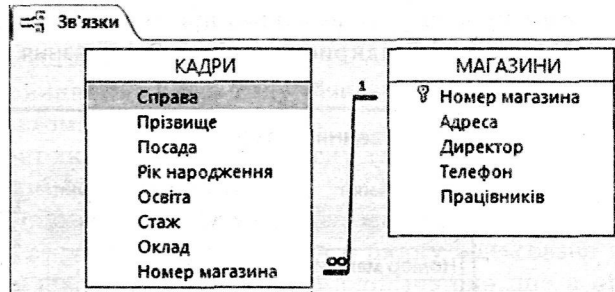


Рис. 5. Зв'язок між таблицями типу 1:Б

6. Повернемося до вікна Редагування зв'язків (якщо в цей момент його немає на екрані, виконуємо команду Змінити зв'язки в області Знаряддя — вікно відкриється). Далі необхідно натиснути кнопку Тип об'єднання.... З'явиться вікно Параметри об'єднання (рис. 6).

? X

Параметри об'єднання

1: Включити лише рядки, де збігаються об'єднані поля з обох таблиць.

2: Включити ВСІ записи з "МАГАЗИНИ" і лише ті записи з "КАДРИ", де зв'язані поля рівні.

3: Включити ВСІ записи з "КАДРИ" і лише ті записи з "МАГАЗИНИ", де зв'язані поля рівні.










Рис. 6. Вікно для встановлення параметрів об'єднання таблиць

7. За замовчуванням установлюється перший тип об'єднання (об'єднання за еквівалентністю). Як правило, розробники БД установлюють відношення за еквівалентністю. Потім слід натиснути в цьому вікні кнопку ОК і закрити вікно Редагування зв'язків.






Для перегляду всіх зв'язків між таблицями бази даних необхідно на відкритій вкладці Знаряддя бази даних натиснути кнопку Зв'язки, активізувати вкладку Конструктор і в розділі Зв'язок натиснути кнопку Усі зв'язки. Якщо схема складна, можна приховати частину таблиць і зв'язків, видаливши їх із вікна Зв'язки, для чого необхідно виділити певну таблицю й натиснути кнопку Del. У цьому разі зв'язки й таблиці видаляються лише з вікна, фізично вони залишаються, тому їх можна відновити в будь-який час.

Для скасування зв'язку між таблицями необхідно встановити курсор на лінії зв'язку, натиснути кнопку миші, а потім — клавішу Del. Можна також відкрити контекстне меню лінії зв'язку й виконати команду Видалити.

? ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1.  Що називають первинним ключем таблиці?
2.  Які первинні ключі називають простими та складними?
3.  Для чого використовується ключове поле типу Лічильник?
4.  Поясніть порядок створення ключового поля.
5.  Що називають індексуванням таблиць?
6.  Як можна скасувати зв'язок між таблицями?
7.  Із якою метою індексуються таблиці?
8.  Для чого потрібно зв'язувати таблиці?
9.  Поясніть порядок зв'язування таблиць.

✓ ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1.   Створіть просту індексну таблицю для поля Учнів таблиці КЛАСИ й індексну таблицю для поля Адреса таблиці УЧНІ.
2.   Створіть у таблиці КЛАСИ БД skola первинний ключ і збережіть таблицю. Переконайтеся, що ключ установлено правильно.
3.  Виконайте зв'язування таблиць КЛАСИ й УЧНІ. Доведіть, що зв'язок дійсно встановлений. Видаліть зв'язок між таблицями й установіть його ще раз.

2.4

УВЕДЕННЯ, ПОШУК І РЕДАГУВАННЯ ДАНИХ У ТАБЛИЦІ

Дані в таблиці можна вводити після створення їх структур. Існує два способи введення даних у таблиці: за допомогою форм і безпосереднє введення даних у таблиці.

Розглянемо порядок безпосереднього введення даних у таблицю МАГАЗИНИ. Для цього виконаємо такі дії.

1. Відкриваємо таблицю в режимі таблиці, для чого двічі клацаємо кнопкою миші на її імені в області Усі об'єкти. Можна також скористатися контекстним меню таблиці.
2. Уводяться дані першого запису таблиці МАГАЗИНИ, який позначений зірочкою. Після введення даних усього запису курсор автоматично переміститься на наступний запис, що означає його готовність до введення даних у другий запис. Необхідно пам'ятати, що вводити в поля можна лише типи даних, які збігаються з оголошеним типом поля.
3. У такому самому порядку вводимо дані решти записів таблиці та зберігаємо її.

Після завершення введення останнього запису вміст таблиці буде мати вигляд, як зображено на рис. 1.

МАГАЗИНИ						
Номер магазину	Адреса	Директор	Телефон	Працівників	Клацність, щоб додати	
+	6 вул. Річкова, 24	Середа К. М.	234-67-92	15		
+	21 вул. Паркова, 33	Коцюба П. М.	234-54-63	20		
+	31 вул. Печерська, 21	Борзов А. С.	234-22-98	13		
*	0			0		

Записи: 4 434 | Без фільтра | Пошук

Рис. 1. Вміст таблиці МАГАЗИНИ

Звернемо увагу, що записи виведені в іншому порядку, ніж вони вводилися (записи вводилися в порядку, який наведено в табл. 1 на с. 20). На рис. 1 вони впорядковані за значеннями ключового поля.

- ✓ Зазначимо, що в процесі введення даних автоматично перевіряються такі типи даних: числові, грошові, дати й часу та логічні. На вкладці Основне в групі Форматування тексту розташовані елементи, за допомогою яких можна змінити розмір шрифту тексту й інші його параметри.

Якщо на екрані не поміщаються всі записи, слід скористатися вертикальною смугою прокручування, а якщо не вміщуються поля — горизонтальною смугою прокручування. Окремі поля можна розширити або звужити звичайним порядком.

У нижній частині вікна таблиці розміщені кнопки навігації для переміщення курсора в перший, сусідній або останній запис. Навігацію в таблиці можна також здійснювати за допомогою миші, смуг прокручування, сполученням деяких клавіш.

Для додавання нового запису в таблицю необхідно натиснути кнопку Створити запис, що розташована на панелі навігації, й увести дані.

У такому самому порядку введемо дані в таблицю КАДРИ. Її вміст зображено на рис. 2.

КАДРИ							
Справа	Прізвище	Посада	Рік народження	Освіта	Стаж	Оклад	Номер магазину
	105 Сокіл Т. Л.	касир	1960	середня	27	3500	6
	109 Шрамко Т. Л.	диспетчер	1961	середня	24	4000	6
	120 Рябко Р. П.	експерт	1981	вища	8	4200	21
	111 Семко М. М.	диспетчер	1970	середня	16	4000	21
	116 Раков Г. П.	аналітик	1965	вища	19	4500	21
	132 Таран В. Д.	диспетчер	1973	вища	15	4000	31
	115 Горошко Ф. Р.	диспетчер	1975	середня спеціальна	17	4000	31
*	0		0		0	0	0

Записи: 4 838 | Без фільтра | Пошук

Рис. 2. Вміст таблиці КАДРИ

Необхідний запис у таблиці можна знайти за значенням будь-якого її поля або за фрагментом його значення. Для цього у відкритій таблиці в режимі таблиці, наприклад, таблиці КАДРИ, слід установити курсор на поле, за значенням якого необхідно шукати запис, наприклад, на поле Прізвище, і натиснути кнопку Знайти в розділі Пошук. Відкриється вікно Пошук і заміна, у поле Знайти якого введемо, наприклад, Сокіл Т. Л. (рис. 3).

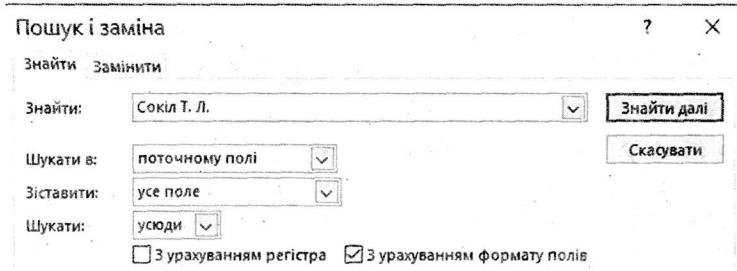


Рис. 3. Вікно для пошуку та заміни даних

Натиснемо кнопку Знайти далі. У результаті відкриється таблиця КАДРИ, і курсор буде встановлено на записі з прізвищем Сокіл Т.Л. У процесі введення даних у поле Знайти можна використовувати такі мета-символи:

- * — довільна кількість будь-яких символів;
- ? — один довільний символ;
- ≠ — одна довільна фраза.

У полі Зіставити можна вибрати одне зі значень, що зменшує необхідність використання мета-символів:

- **будь-яку частину поля** — зразок може міститися всередині значення поля;
- **усе поле** — зразок без мета-символів має збігатися з усім значенням поля;
- **початок поля** — будуть знайдені лише ті поля, які починаються зі зразка.

У вікні Пошук і заміна є два прапорці:

- **З урахуванням регістра.** Якщо прапорець установлено, то зразок має точно відповідати фрагменту, що відшукується. Якщо прапорець знятий, зразок можна вводити і великими, і малими буквами;
- **З урахуванням формату полів.** У разі зняття прапорця форматування не враховується. Наприклад, значення \$345 буде знайдено і для зразка 345. Якщо ж прапорець установлено, то зразок має бути таким самим, як і шуканий фрагмент, тобто \$345.

Після завершення пошуку на знайдений запис установлюється курсор, і поле цього запису висвітлюється іншим кольором. Оскільки

в таблиці може бути кілька значень, що відшукуються, то для продовження пошуку необхідно ще раз натиснути кнопку Знайти далі.

- ✓ Для пошуку першого входження зразка можна використовувати поле Пошук, яке розташоване в нижній частині вікна таблиці. Пошук завершується після першого визначення запису.

Знайдене значення можна змінювати, вводити нове значення. Однак поле типу Лічильник, заблоковані поля та поля, що обчислюються, змінювати не можна. Знайдемо, наприклад, у таблиці КАДРИ запис, у полі Рік народження якого є значення 1981. Буде виділено запис із прізвищем Рябко Р.П.

Записи з таблиці можна вирізати й копіювати до буфера обміну, а потім за допомогою кнопки Вставити, яка розташована в розділі Буфер обміну, вставляти в іншу таблицю, а також у документи Word і Excel.

Для видалення запису з таблиці необхідно виділити його й натиснути на вкладці Основне кнопку Видалити. Відкриється меню цієї кнопки (рис. 4), у якому слід виконати команду Видалити запис.

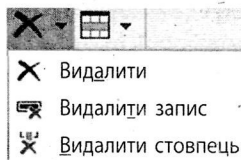


Рис. 4. Меню кнопки Видалити

У результаті відкриється вікно (рис. 5), у якому необхідно підтвердити або відмінити видалення.

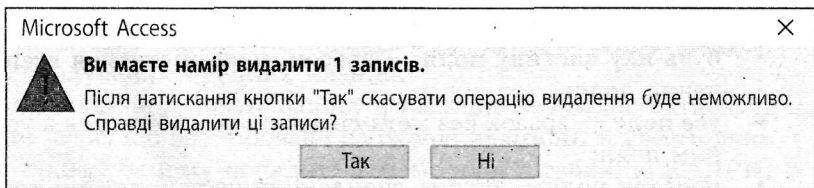


Рис. 5. Вікно підтвердження (відміни) видалення запису

Якщо необхідно видалити весь запис, то його слід виділити й так само натиснути кнопку Видалити, а потім у меню виконати команду Видалити запис.

Слід пам'ятати, що в пов'язаних таблицях зі встановленим прапорцем Забезпечення цілісності даних видалити запис не завжди вдається.

Зазначимо також, що інколи для кращого огляду таблиці виникає необхідність приховати деякі поля. Для цього необхідно виділити певні поля й виконати в групі Записи команду Додатково → Приховати поля.

Для відновлення прихованих полів необхідно в цьому самому розділі Записи виконати команду Додатково → Відобразити поля й у вікні, що відкриється, увімкнути прапорець відповідного поля.



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1. Поясніть порядок уведення записів у таблицю.
2. За значенням якого поля впорядковуються записи за замовчуванням?
3. Як можна здійснити навігацію в таблиці?
4. Як можна змінити шрифт даних у таблиці?
5. Як здійснюється пошук запису в таблиці?
6. Як видаляється запис із таблиці?
7. Які дії слід виконати для додавання нового запису в таблицю?
8. Із якою метою використовуються мета-символи в процесі пошуку записів?
9. Із якою метою і як приховуються поля?



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1. Уведіть у таблиці КЛАСИ й УЧНІ БД *skoła* наведені в них дані.
2. Виконайте навігацію по таблиці УЧНІ за допомогою кнопок навігації.
3. Додайте в таблицю УЧНІ новий запис, а потім видаліть його. Збережіть таблицю.
4. Знайдіть у таблиці УЧНІ запис за прізвищем Пека П.О.
5. Використайте мета-символи в процесі пошуку записів у таблиці УЧНІ за значенням поля Адреса.
6. Видаліть із таблиці УЧНІ будь-який запис, а потім відновіть його. Збережіть таблицю.

2.5

**СОРТУВАННЯ ТА ФІЛЬТРАЦІЯ ЗАПИСІВ.
ОПЕРАЦІЇ НАД ТАБЛИЦЯМИ**

Згадаємо, що за замовчуванням записи таблиці виводяться впорядкованими за значенням первинного ключа. Проте часто виникає необхідність отримати записи, впорядковані за значенням інших полів.

У системі Access 2016 передбачено сортування записів за значеннями одного та кількох полів.

Для сортування за значенням одного поля слід його виділити й натиснути кнопку За зростанням (А → Я) або За спаданням (Я → А).

Можна також скористатися контекстним меню поля, у якому слід виконати аналогічні команди.

Для сортування за кількома полями необхідно виділити ці поля й скористатися одним зі вказаних способів. Сортування виконується спочатку за значенням лівого виділеного поля. Якщо в ньому є поля, значення яких збігаються, то певні записи впорядковуються за значенням наступного поля. Наприклад, унаслідок сортування записів За зростанням за значенням полів Освіта й Стаж таблиці КАДРИ отримаємо розміщення записів, зображене на рис. 1.

Справа	Прізвище	Посада	Рік народження	Освіта	Стаж	Оклад	Номер магазину
120	Рябко Р. П.	експерт	1981	вища	8	4200	21
132	Таран В. Д.	диспетчер	1973	вища	15	4000	31
116	Раков Г. П.	аналітик	1965	вища	19	4500	21
111	Семко М. М.	диспетчер	1970	середня	16	4000	21
109	Шрамко Т. Л.	диспетчер	1961	середня	24	4000	6
105	Сокіл Т. Л.	касир	1960	середня	27	3500	6
115	Горошко Ф. Р.	диспетчер	1975	середня спеціальна	17	4000	31
*	0		0		0	0	0

Рис. 1. Таблиця КАДРИ, впорядкована за значенням полів Освіта й Стаж

Основне впорядкування виконане за зростанням значення поля Освіта. Записи з однаковим значенням цього поля впорядковані за зростанням значення поля Стаж.



Фільтрація — це відбір із таблиці записів, які містять задане значення в обраних полях.

Наприклад, якщо здійснити фільтрацію записів таблиці КАДРИ більше або дорівнює 24 у полі Стаж, то отримаємо записи з прізвищами Сокіл Т.Л. і Шрамко Т.Л.

В Access 2016 забезпечується фільтрація за виділенням і за формою.

Фільтрація за виділенням — це відбір записів на основі значень поточного поля.

Для її реалізації спочатку слід упорядкувати записи за значенням поля, яке використовується у фільтрації. Потім необхідно встановити курсор на тому значенні поля, за яким буде виконуватися фільтрація, і натиснути на кнопку Виділення в групі Сортування й фільтр. Далі слід вибрати необхідну умову в меню, що відкриється.

Відфільтруємо записи таблиці КАДРИ за значенням "середня" поля Освіта.

1. Упорядкуємо записи таблиці КАДРИ за значенням поля Освіта.

- Встановлюємо курсор на назві "середня" цього поля й натискаємо кнопку Виділення в групі Сортування й фільтр. Відкриється меню з умовами (рис. 2).
- Вибираємо першу умову Дорівнює "середня". Відкриється таблиця, вміст якої зображено на рис. 3.



Рис. 2. Меню умов для фільтрації записів

Справа	Прізвище	Посада	Рік народження	Освіта	Г	Стаж	Оклад	Номер магазину
109	Шрамко Т. Л.	диспетчер	1961	середня		24	4000	6
111	Семко М. М.	диспетчер	1970	середня		16	4000	21
105	Сокіл Т. Л.	касир	1960	середня		27	3500	6
*			0			0	0	0

Записи: 4 з 4 | Відфільтровано | Пошук

Рис. 3. Таблиця з відфільтрованими записами

У нижній частині таблиці висвітиться кнопка Відфільтровано (або Не відфільтровано), це означає, що записи відфільтровані (або не відфільтровані). Натискаючи цю кнопку, можна вмикати й вимикати фільтрування. До відфільтрованих записів можна застосовувати ще кілька фільтрів, наприклад, за значеннями полів Посада, Оклад тощо.

Використовуючи **фільтр за формою**, можна вводити критерії у поля таблиці умов. Щоб відкрити таблицю умов, необхідно виконати певні дії.

- У розділі Сортування і фільтр відкриваємо меню кнопки Параметри розширеного фільтра й виконуємо команду Розширений фільтр/сортування. Відкриється перелік полів таблиці, а в нижній частині вікна — таблиця конструктора.
- У таблицю конструктора, наприклад, із відфільтрованої таблиці КАДРИ, перенесемо назву поля, у яке буде вводитися критерій. Для перенесення назви поля достатньо двічі клацнути його ім'я в таблиці. Перенесемо, наприклад, ім'я поля Стаж.
- У таблиці конструктора в запис Критерії цього поля введемо умову, наприклад, >16. Далі відкриваємо меню кнопки Параметри розширеного фільтра, у якому виконуємо команду Застосувати фільтр/сортування. У результаті отримаємо записи, зображені на рис. 4.

Справа	Прізвище	Посада	Рік народження	Освіта	Г	Стаж	Оклад	Номер магазину
109	Шрамко Т. Л.	диспетчер	1961	середня		24	4000	6
105	Сокіл Т. Л.	касир	1960	середня		27	3500	6
*			0			0	0	0

Записи: 3 з 3 | Відфільтровано | Пошук

Рис. 4. Відфільтровані записи, відібрані за критерієм поля Стаж >16

У таблиці умов можна задати довільну кількість критеріїв фільтрування, які об'єднуються операторами І/АБО.

- ✓ Над таблицями в середовищі Access 2016 можна виконувати такі операції: перейменовувати, видаляти, копіювати.

Перейменувати таблицю можна за допомогою її контекстного меню. Воно відкривається клацанням імені таблиці в області Усі об'єкти. Вміст меню зображено на рис. 5.

Для перейменування таблиці слід виконати команду Перейменувати. Після цього також в області Усі об'єкти в текстове поле таблиці необхідно ввести нове ім'я і натиснути клавішу Enter. Слід пам'ятати, що після цього необхідно змінити це ім'я в усіх об'єктах бази даних (запитах, звітах та інших об'єктах).

Для **видалення таблиці** також можна скористатися її контекстним меню, у якому необхідно виконати команду Видалити.

Для **копіювання таблиці** в контекстному меню виконується команда Копіювати. Потім ще раз викликається контекстне меню й виконується команда Вставити. Відкриється вікно Вставлення таблиці, зображене на рис. 6.

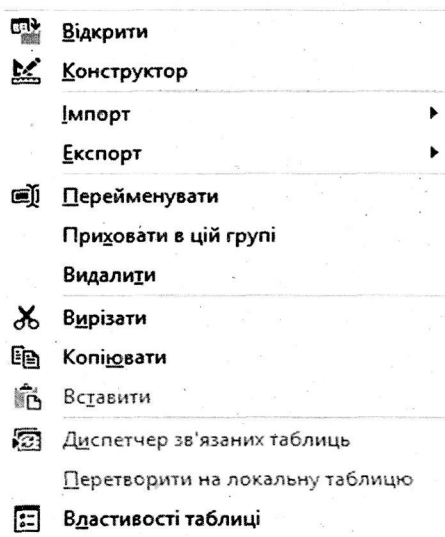


Рис. 5. Контекстне меню таблиці, відкрите в області Усі об'єкти

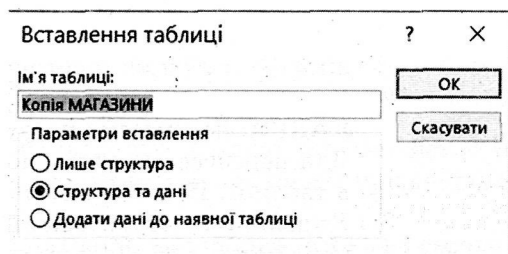



















Рис. 6. Вікно для копіювання таблиці

У поле Ім'я таблиці слід ввести нове ім'я й використати один із перемикачів, розташованих нижче. Якщо таблиця копіюється повністю, застосовується перемикач Структура та дані. Після цього натискається кнопка ОК.

? ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1.  Які існують способи впорядкування записів?
2.  Як упорядковуються записи за значенням одного поля?
3.  Що називають фільтрацією записів?
4.  Як упорядковуються записи за значеннями двох полів?
5.  Які операції можна виконувати над таблицями?
6.  Як перейменовується таблиця?
7.  Як фільтруються записи за виділенням?
8.  Поясніть сутність фільтрації записів за формою.
9.  Як копіюються й видаляються таблиці?

✓ ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1.  Упорядкуйте записи таблиці УЧНІ БД skola за значенням поля Зріст.
2.  Упорядкуйте записи таблиці УЧНІ за значеннями полів Прізвище й Історія.
3.   Відфільтруйте записи таблиці УЧНІ за одним зі значень поля Улюблений предмет.
4.  Використайте фільтр за формою для фільтрації записів таблиці УЧНІ за одним значенням поля Дата народження.
5.   Надайте таблиці КЛАСИ нове ім'я. Відновіть ім'я КЛАСИ.
6.  Виконайте копіювання таблиці УЧНІ і створіть на її основі нову таблицю з іменем ПЕРША.

Практична робота № 1

Тема: СТВОРЕННЯ СТРУКТУРИ ТАБЛИЦЬ І ВВЕДЕННЯ ВМІСТУ
Завдання: розробити структури таблиць і ввести їх вміст.
Обладнання: комп'ютер зі встановленою СУБД Access 2016.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Створіть базу даних biblioteka з двома таблицями КНИЖКИ і ЧИТАЧІ. У першій таблиці зберігаються дані про книжки бібліотеки, а у другій — дані про їх читачів.

2. Розробіть на папері структуру таблиці **КНИЖКИ**.

Таблиця 1. **КНИЖКИ**

Інвентарний номер	Назва	Автор	Рік видання	Ціна
201	Фізика	Федірко К.О.	2016	33
202	Фізика	Федірко К.О.	2016	33
203	Фізика	Мехов А.С.	2017	35
204	Фізика	Мехеда О.Н.	2017	40
205	Фізика	Мехеда О.Н.	2017	40
206	Інформатика	Соломко І.М.	2017	45
207	Інформатика	Соломко І.М.	2017	45
208	Інформатика	Соломко І.М.	2017	45
209	Інформатика	Федірко К.О.	2015	43
210	Інформатика	Мехеда О.Н.	2018	50
211	Інформатика	Соломко І.М.	2018	55
212	Інформатика	Соломко І.М.	2018	55

3. Уведіть структуру таблиці в режимі Конструктора таблиць і збережіть її.
4. Розробіть на папері структуру таблиці **ЧИТАЧІ**.

Таблиця 2. **ЧИТАЧІ**

Номер	Прізвище	Клас	Дата видачі	Дата повернення	Інвентарний номер
1	Перов С.М.	10	05.02.2018	10.03.2018	201
2	Соломаха К.К.	11	10.02.2018	22.03.2018	206
3	Басов А.С.	10	15.02.2018	26.03.2018	207
4	Перов С.М.	10	16.02.2018	27.03.2018	208
5	Пекар Н.Н.	11	16.02.2018	30.03.2018	203
6	Ахтирко М.О.	10	18.02.2018	30.03.2018	204
7	Агапова К.М.	11	03.04.2018	15.04.2018	201
8	Настіна О.С.	11	03.04.2018		210
9	Басов А.С.	10	06.04.2018		212
10	Ріпко П.С.	10	06.04.2018	16.04.2018	209
11	Соломаха К.К.	11	06.04.2018	19.04.2018	203

5. Уведіть структуру таблиці **ЧИТАЧІ** в режимі Конструктора таблиць і збережіть її.
6. У таблиці **КНИЖКИ** поле Інвентарний номер визначте як первинний ключ.
7. Установіть зв'язок між таблицями **КНИЖКИ** й **ЧИТАЧІ**.
8. Уведіть вміст таблиці **КНИЖКИ** й перевірте її правильність.
9. Уведіть вміст таблиці **ЧИТАЧІ** й перевірте її правильність.

Зробіть висновок про результати роботи.

Практична робота № 2

Тема: ПОШУК, СОРТУВАННЯ І ФІЛЬТРАЦІЯ ДАНИХ
У ТАБЛИЦЯХ

Завдання: здійснити пошук, сортування та фільтрацію даних у таблицях.

Обладнання: комп'ютер зі встановленою СУБД Access 2016 і таблицями, розробленими в попередній роботі.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. У таблиці КНИЖКИ знайдіть усі записи, у полі Автор яких є прізвище Мехеда О.Н.
2. У таблиці КНИЖКИ знайдіть книжки, видані у 2015 році.
3. У таблиці УЧНІ знайдіть записи, у полі Дата повернення яких є дата 30.03.2018.
4. У таблиці КНИЖКИ запис, у полі Інвентарний номер якого є число 205, замініть Фізика на Хімія у полі Назва. Переконайтеся, що операція виконана правильно. Відновіть значення Фізика та збережіть таблицю.
5. У таблиці ЧИТАЧІ видаліть шостий запис. Переконайтеся, що операція виконана правильно. Відновіть цей запис. Збережіть таблицю.
6. У таблиці ЧИТАЧІ приховайте поле Клас. Переконайтеся, що операція виконана успішно. Відновіть приховане поле.
7. Виконайте сортування записів таблиці КНИЖКИ в алфавітному порядку поля Автор. Відновіть упорядкування записів за значенням ключового поля:
8. Виконайте сортування записів таблиці ЧИТАЧІ в алфавітному порядку прізвищ читачів. Відновіть упорядкування записів за значенням поля Номер.
9. Виберіть із таблиці КНИЖКИ записи, у полі Автор яких є прізвище Федірko К.О.
10. Виберіть із таблиці КНИЖКИ всі записи, де ціна книжок менше 45.
11. Виберіть із таблиці ЧИТАЧІ записи, у полі Дата повернення яких відсутня дата повернення.
12. Перейменуйте таблицю КНИЖКИ на ім'я ФОНД. Відновіть попередню назву таблиці.
13. Для таблиці ЧИТАЧІ створіть копію з назвою УЧНІ.

Зробіть висновок про результати роботи.

3

ЗАПИТИ

3.1

ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЗАПИТИ

Запит — це один із основних об'єктів БД Access 2016. Головним призначенням запиту є відбір із таблиць потрібних даних, їх опрацювання й подання користувачеві в зручній формі.

Створений запит можна зберігати з певним іменем і потім неодноразово виконувати. Якщо між першим і другим запусками запиту змінилися дані в таблицях, то в процесі другого його виконання будуть використовуватися оновлені дані.

Запити не містять даних. Під час кожного нового виконання запиту формуються необхідні дані з тих таблиць, на основі яких його створено. Запити класифікують за багатьма ознаками. Класифікація запитів за основними ознаками зображена на рис. 1.

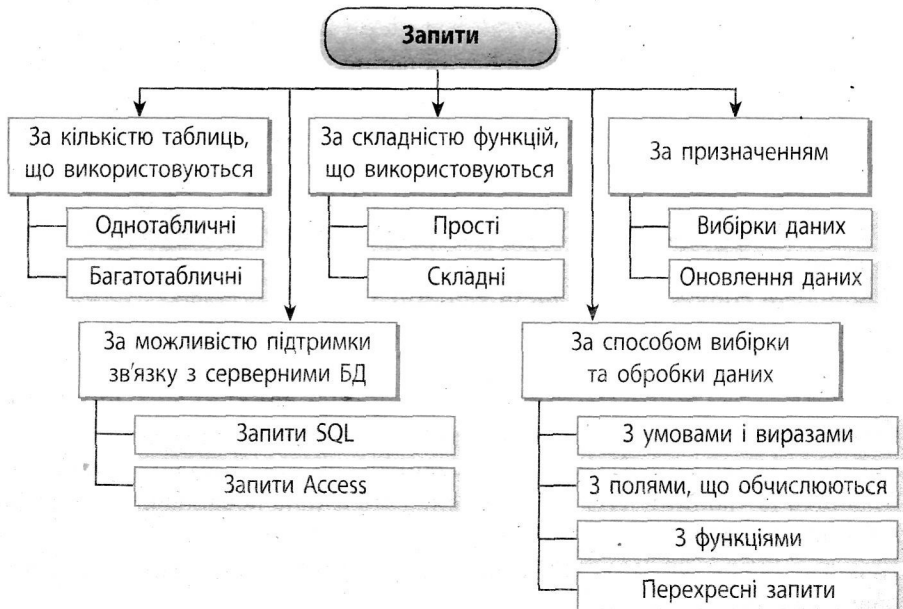


Рис. 1. Класифікація запитів

Запити, за допомогою яких вибираються дані з однієї таблиці, називають однотабличними, а з кількох — багатотабличними. До простих запитів належать такі, за якими вибираються дані за критерієм одного поля однієї таблиці. Наприклад, простими є такі запити: вибрати

з таблиці МАГАЗИНИ ті номери магазинів, де кількість працівників більше 13, вибрати з таблиці КАДРИ прізвища тих працівників, які народилися після 1979 року. До складних належать запити, за якими формуються дані з кількох таблиць за критеріями кількох полів. Наприклад, складним є запит для таблиць МАГАЗИНИ й КАДРИ, за яким формуються дані, наведені в табл. 1.

Таблиця 1. РЕЗУЛЬТАТ ВИКОНАННЯ СКЛАДНОГО ЗАПИТУ

Магазин	Адреса	Прізвище	Посада
21	вул. Паркова, 33	Семко М.М.	диспетчер
31	вул. Печерська, 21	Таран В.Д.	диспетчер
31	вул. Печерська, 21	Горошко Ф.Р.	диспетчер

Тут з таблиць МАГАЗИНИ й КАДРИ відібрані прізвища людей, які працюють у магазинах 21 і 31 диспетчерами.

До однієї БД Access 2016 може бути розроблено кілька запитів, кожен із яких можна виконати в будь-який час, і кожен із них виконує чітко визначені функції. Але запити можуть не лише виконуватися самостійно, а й використовуватися в складі форм і звітів.

- ✓ Ще раз зазначимо, що запити не містять даних, вони лише формують необхідні дані з таблиць. Наприклад, отримані за допомогою запиту дані, наведені в результуючій табл. 1, не зберігаються, а формуються в процесі виконання запиту. Щоб зберегти ці дані, необхідно створити таблицю та скопіювати до неї цей набір даних.

Описаний тип запиту називають **запитом на вибірку**. Такий тип запитів є одним із найрозповсюдженіших. Разом із тим, в Access 2016 використовуються й **запити на зміну**, за допомогою яких модифікуються структури таблиць і змінюються в них дані.

За допомогою запитів найчастіше дані вибираються на основі критеріїв. Окрім того, Access 2016 має набір убудованих функцій, за допомогою яких можна не лише вибрати з таблиць необхідні дані, а й опрацювати їх та узагальнити. Із опрацьованих даних можна створювати нові поля. Такий тип запитів називають **запитами з полями**, що обчислюються.

Перелічимо операції, які найчастіше реалізуються в запитах різних типів:

- вибір даних зі вказаних полів на основі заданих критеріїв;
- дані, отримані з таблиць, упорядковуються за значеннями вказаних полів;
- із отриманих даних може бути побудована нова таблиця або діаграма;
- вибрані з таблиць дані можуть бути опрацьовані за допомогою вбудованих функцій;

- отримані за допомогою запиту дані можуть використовуватися як джерело для інших запитів;
- дані, отримані за допомогою запитів, можуть додаватися до інших таблиць;
- можна виконувати обмін даними з іншими базами даних, а також текстовим редактором Word і електронними таблицями Excel.

Вирази в критеріях, що застосовуються в запитах, будуються на основі звичайних арифметичних операцій, операцій порівняння й логічних операцій (And, Or, Xor, Non).

Окрім того, може використовуватися низка специфічних операторів.










1. Рядкові оператори Like (які збігаються/відповідність), Not Like (які не збігаються/невідповідність), а також уже відомий оператор об'єднання рядків (&). Оператори Like та Not Like використовуються для порівняння двох рядкових виразів. Водночас перевіряється, чи збігаються ці вирази, і залежно від результату повертають значення Так, Ні або Null. Оператор Like має таку структуру: <ім'я поля> Like <зразок>. Наприклад, для таблиці КАДРИ вираз <Прізвище Like "Раков Г.П."> повертає значення Так, оскільки поле Прізвище містить значення Раков Г.П.
2. Оператори списку й діапазону: In (входження в список), Is (нааявність значення), Between... And (входження в діапазон). За допомогою оператора In перевіряється, чи збігається/відповідає значення поля з одним/одному зі значень списку. Якщо збігається/відповідає, повертається значення Так, інакше — Ні. Наприклад, для таблиці КАДРИ оператор <Прізвище In ("Семко Н.Н.; 'Горошко Ф.Р')> повертає значення Так, оскільки в цьому полі є зазначені прізвища. Оператор Is використовується лише з ключовим словом Null для з'ясування, чи містить об'єкт будь-яке значення. Повертається значення Так, якщо вираз порожній (не містить жодного значення). Оператор Between... And має таку структуру: <ім'я поля> Between <нижня межа> And <верхня межа>. Повертається значення Так, якщо значення поля містяться між значеннями <нижня межа> і <верхня межа>. Наприклад, для таблиці КАДРИ за допомогою оператора [Рік народження] Between 1965 And 1973 повертається значення Так, оскільки в цьому полі є зазначені діапазони.

Зазначимо, що критерії поділяються на прості й складні. Прості критерії, як правило, містяться лише в одному полі, а складні — у кількох полях. Наведемо приклад складного критерію: вибрати з таблиць МАГАЗИНИ й КАДРИ прізвища працівників магазинів, які народилися в період із 1961 по 1975 рік і працюють диспетчерами або аналітиками.

Далі буде стисло описано методику створення всіх типів запитів (рис. 1 на с. 44), крім запитів SQL, для розроблення яких потрібно володіти мовою SQL.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1.  Яке основне призначення мають запити?
2.  Які запити називають простими?
3.  Як класифікуються запити за призначенням?
4.  Назвіть основні класифікаційні ознаки запитів.
5.  Наведіть приклад простого запиту.
6.  Які функції виконують запити на вибірку?
7.  Назвіть операції, які найчастіше реалізуються в запитах.
8.  Які існують оператори списку й діапазону?
9.  Наведіть приклад використання операторів Like та Not Like.

3.2

ЗАПИТИ НА ВИБІРКУ ДАНИХ

Запити на вибірку даних — це запити, які забезпечують вибір необхідних даних з однієї або з кількох таблиць.

Загальний порядок створення простого запиту на вибірку (запиту для однієї таблиці) такий.

1. Відкриваємо БД, активуємо вкладку Створення й у розділі Запити натискаємо кнопку Макет запиту, який фактично є конструктором запиту. У результаті відкриваються вікно конструктора запиту (вікно Запит1) і вікно Відображення таблиці, у якому містяться імена всіх таблиць цієї бази. Звернемо увагу, що на панелі інструментів вкладки Конструктор з'явилася група кнопок Тип запиту, у якій виділеною є кнопка Вибір. Це означає, що запит на вибірку створюється за замовчуванням. Якщо будуть створюватися інші типи запитів, то необхідно вмикати відповідну кнопку в цій групі.
2. Вибираємо у вікні Відображення таблиці необхідну таблицю. Відкриється перелік її полів.
3. Створюємо запит на основі вмісту цієї таблиці.

Приклад 1

Створити простий запит із іменем Запит_1, за допомогою якого з таблиці КАДРИ виводяться дані про працівників зі стажем понад 16 років. Результуючий набір записів має містити поля: Справа, Прізвище, Рік народження, Стаж, Номер магазину.

1. У відкритій БД atb виконуємо команду Створення → Макет запиту. Відкриваються вікно конструктора запитів і вікно Відображення таблиці, у якому вибираємо таблицю КАДРИ. Для цього встановлюємо курсор на імені цієї таблиці й натискаємо кнопку Додати. Після цього вікно Відображення таблиці можна закрити.

- У записі Поле конструктора запитів послідовно розміщуємо зазначені імена полів (Справа, Прізвище, Рік народження, Стаж, Номер магазину) таблиці КАДРИ (рис. 1). Для цього достатньо двічі клацнути кнопку миші на імені певного поля цієї таблиці.

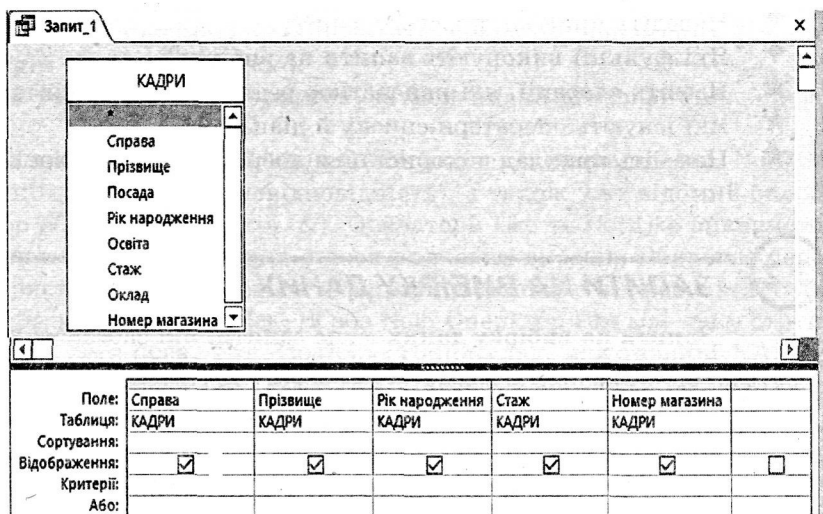


Рис. 1. Вікно конструктора запитів

- Рядок Сортування використовується для сортування даних у таблиці, яка буде отримана після виконання запити. Сортувати дані можна за значенням кількох полів. Прапорець, установлений на перетині запису Відображення та певного поля, означає, що це поле буде виведено на екран, інакше воно виводитися не буде.
- Запис Критерії призначений для запису виразу, на основі якого відбираються записи.
- Запис Або призначений для визначення додаткової умови відбору записів.
- У запис Критерії поля Стаж уводимо вираз >16 . Зберігаємо запит, для чого на панелі швидкого доступу натискаємо кнопку Зберегти й у вікні, що відкриється, уводимо ім'я Запит_1, і натискаємо кнопку ОК.
У результаті ім'я цього запити з'явиться в області переходів.
 - Виконуємо запит. Для цього на смузі в групі Результати натискаємо кнопку Запуск!. Отримаємо результат, зображений на рис. 2.
 - Для закриття запити відкриваємо його контекстне меню та виконуємо команду Закрити.

Запит_1					
Справа	Прізвище	Рік народження	Стаж	Номер магазину	
105	Сокіл Т. Л.	1960	27	6	
115	Горошко Ф. Р.	1975	17	31	
116	Раков Г. П.	1965	19	21	
109	Шрамко Т. Л.	1961	24	6	
*	р	0	0	0	

Рис. 2. Дані про працівників магазинів зі стажем роботи понад 16 років

Створений запит можна перейменувати й редагувати. Для перейменування запиту слід відкрити його контекстне меню та виконати команду *Перейменувати*. Ім'я цього запиту в області переходів буде виділено прямокутником іншого кольору. У це поле необхідно ввести нове ім'я й натиснути клавішу *Enter*.

- ✓ У процесі редагування запиту можна виконувати такі дії:
 - додавати поля з таблиці;
 - видаляти поля;
 - додавати нові поля;
 - змінювати розміри полів;
 - змінювати порядок розміщення полів;
 - змінювати критерії відбору записів, порядок їх сортування й порядок виведення (невиведення) полів;
 - перейменувати поля запиту;
 - видаляти з запиту таблиці (для багатотабличних запитів).

Для виконання перелічених операцій запит відкривається в режимі *Конструктор*. Методика їх виконання відповідає методиці виконання аналогічних операцій у таблицях і жодних складнощів не викликає.

- ✓ Методика створення запиту для кількох таблиць багато в чому схожа на методику створення запиту для однієї таблиці. Однак слід урахувати, що таблиці обов'язково повинні мати між собою зв'язок.

Приклад 2

Створити *Запит_2*, за допомогою якого в результатуючу таблицю включаються прізвища диспетчерів із полями *Номер магазину* і *Телефон* таблиці *МАГАЗИНИ*, а з таблиці *КАДРИ* — поля *Прізвище*, *Посада* й *Освіта*. Записи необхідно впорядкувати за прізвищами працівників у алфавітному порядку.

1. У відкритій БД *atb* активуємо вкладку *Створення* й виконуємо команду *Макет запиту*. Додаємо у вікно *Конструктор запиту* обидві таблиці й закриваємо вікно *Відображення таблиць*.
2. У рядок *Поле* таблиці конструктора переносимо поля *Номер магазину* і *Телефон* таблиці *МАГАЗИНИ* та поля *Прізвище*, *Посада* й *Освіта* таблиці *КАДРИ*.

3. У запис Критерії поля Посада вводимо назву професії «диспетчер», а в записі Сортунання поля Прізвище встановлюємо значення За зростанням.
4. Зберігаємо запит з іменем Запит_2 і виконуємо його. На екрані мають з'явитися записи, наведені на рис. 3.

Запит_2				
Номер магазину ▾	Телефон ▾	Прізвище ▾	Посада ▾	Освіта ▾
	31 234-22-98	Горошко Ф. Р.	диспетчер	середня спеціальна
	21 234-54-63	Семко М. М.	диспетчер	середня
	31 234-22-98	Таран В. Д.	диспетчер	вища
	6 234-67-92	Шрамко Т. Л.	диспетчер	середня
*				

Рис. 3. Дані про диспетчерів, вибрані з двох таблиць



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. ▲ Які запити називають запитами на вибірку даних?
2. ▲ Як зберігається запит?
3. ▲ У яких режимах можна відкрити запит?
4. ◆ Для чого призначено запис Критерії таблиці Конструктор запиту?
5. ◆ Як можна змінити порядок розміщення полів у запиті?
6. ◆ Поясніть різницю створення запиту для однієї і кількох таблиць.
7. ★ Поясніть алгоритм створення запиту на вибірку даних.
8. ★ Які дії можна виконувати в процесі редагування запиту?



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ



1. ◆ Створіть Запит41 на основі таблиці УЧНІ, за допомогою якого вибираються прізвища учнів, улюбленим предметом яких є історія. Записи повинні мати поля Прізвище, Клас, Улюблений предмет:
2. ◆ Створіть Запит42 на основі таблиці УЧНІ, за допомогою якого вибираються прізвища учнів, зріст яких більше 170.
3. ◆ Створіть Запит43 на основі таблиці УЧНІ, за допомогою якого вибираються прізвища учнів, улюбленими предметами яких є математика й інформатика.
4. ★ Створіть Запит44 на основі таблиць КЛАСИ й УЧНІ, за допомогою якого вибираються прізвища класних керівників учнів, що мають успішність з інформатики й історії понад 10.
5. ★ Створіть Запит45 на основі таблиці УЧНІ, за допомогою якого вибираються прізвища учнів, що народилися у 2002 році.
6. ★ Створіть Запит46 на основі таблиць КЛАСИ й УЧНІ, за допомогою якого для класів, де кількість учнів понад 24, вибираються прізвища учнів, що мають успішність з історії вище 9.

ЗАПИТИ З ФУНКЦІЯМИ І З ПОЛЯМИ, ЩО ОБЧИСЛЮЮТЬСЯ

У попередньому розділі розглядалися запити, за допомогою яких із таблиць вибираються необхідні дані за певними критеріями. Отримані дані можна також опрацювати, наприклад, обчислити середнє значення поля, знайти серед знайдених записів із мінімальним значенням певного поля тощо.

Наприклад, для таблиці КАДРИ можна обчислити середній стаж роботи працівників із вищою освітою.

Для полегшення опрацювання даних система Access 2016 має кілька вбудованих функцій, основними з яких є:

- Сума — обчислює суму значень елементів поля;
- Середнє — обчислює середнє значення поля;
- Максимум/Мінімум — повертає елемент із максимальним/мінімальним значенням поля;
- Кількість — підраховує кількість записів за значенням поля;
- Відхилення від середнього значення — показує, наскільки значення поля відхиляється від середнього.

Запити, у яких використовуються такі функції, називаються по-різному, наприклад, підсумковими запитамі, але найчастіше їх називають просто **запитами з функціями**.

В Access 2016 існують два способи використання перелічених функцій:

- до запиту, відкритого в режимі таблиці, додається запис підсумків, у якому для кожного поля може використовуватися одна з функцій;
- у режимі Конструктор створюється підсумковий запит, в якому обчислюються проміжні підсумки за групами записів.

Розглянемо перший спосіб на конкретному прикладі.

Приклад 1

Створити Запит_4, за допомогою якого з таблиці КАДРИ вибираються записи працівників, що народилися після 1961 року і мають стаж понад 15 років. Результуючі записи мають містити поля: Прізвище, Посада, Рік народження, Стаж та Оклад.

Підрахувати кількість результуючих записів за значенням поля Прізвище та обчислити загальну суму окладів цих працівників.

1. Створюємо в режимі Конструктор звичайний запит на вибірку, як зображено на рис. 1.
2. Зберігаємо запит з іменем Запит_4 і виконуємо його. Результат його виконання зображений на рис. 2.

Поле:	Прізвище	Посада	Рік народження	Стаж	Оклад
Таблиця:	КАДРИ	КАДРИ	КАДРИ	КАДРИ	КАДРИ
Підсумок:	Групування за	Групування за	Групування за	Групування за	Групування за
Сортування:					
Відображення:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Критерії:			>1961	>15	
Або:					

Рис. 1. Приклад запити на вибірку даних у режимі Конструктор

Прізвище	Посада	Рік народження	Стаж	Оклад
Семко М. М.	диспетчер	1970	16	4000
Горошко Ф. Р.	диспетчер	1975	17	4000
Раков Г. П.	аналітик	1965	19	4500
*			0	0

Рис. 2. Результат виконання запити на вибірку даних

3. На вкладці Основне в групі Записи натискаємо кнопку Підсумки (). Під останнім записом таблиці, зображеної на рис. 2; з'явиться новий запис Підсумок. У записі Підсумок клацаємо поле Прізвище й у списку, що відкривається, вибираємо функцію Кількість. Аналогічно в цьому самому записі поля Оклад вибираємо функцію Сума. Отримаємо результат, зображений на рис. 3.

Прізвище	Посада	Рік народження	Стаж	Оклад
Семко М. М.	диспетчер	1970	16	4000
Горошко Ф. Р.	диспетчер	1975	17	4000
Раков Г. П.	аналітик	1965	19	4500
*			0	0
	3			12500

Рис. 3. Запит у режимі таблиці з підсумковим записом

4. Для збереження внесених змін ще раз натискаємо кнопку Зберегти.

Розглянемо тепер другий спосіб використання функцій, тобто створення підсумкових запитів.

1. Створюємо звичайний запит на вибірку в режимі Конструктор, наприклад, запит, за допомогою якого з таблиці КАДРИ вибираються прізвища працівників з окладом вище 4000 грн і підраховується їх кількість. Результуючий перелік записів має містити поля Справа, Прізвище, Стаж, Оклад.
2. У рядок Критерії поля Оклад вводимо вираз >4000.

- На вкладці Конструктор відкриваємо меню кнопки Відображення або приховання й виконуємо команду Підсумки (Σ). У Конструкторі запити з'явиться рядок Підсумок, а в кожному полі цього рядка буде зазначено Групування за.
- У записі Підсумок клацаємо те поле, за яким потрібно виконати підрахунок кількості записів (наприклад, поле справа). У списку, що відкриється, вибираємо функцію Кількість (рис. 4).

Поле:	Справа	Прізвище	Стаж	Оклад			
Таблиця:	КАДРИ	КАДРИ	КАДРИ	КАДРИ			
Підсумок:	КІЛЬКІСТЬ	Групування за	Групування за	Групування за			
Сортування:	Групування за						
Відображення:	Сума	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Критерій:	Середнє			>4000			
Або:	Мінімум						
	КІЛЬКІСТЬ						
	StDev						
	Var						
	Перший						
	Останнє						
	Вираз						
	Розташування						

Рис. 4. Приклад підсумкового запити в режимі Конструктор

- Далі зберігаємо й виконуємо запит.



Система Access 2016 дає також змогу створювати запити з полями, що обчислюються. Таких полів у запиті може бути кілька.

Такі запити дозволяють виводити в результуючий набір записів не лише поля таблиць, а й нові поля, які створює сам користувач. У них містяться дані, отримані в процесі обчислення даних полів таблиць. Наприклад, на основі даних таблиці КАДРИ можна в результуючий набір записів увести поле Доплата, у якому обчислюється доплата до окладу залежно від стажу працівника.

Припустимо, що за кожен рік стажу понад 5 років працівники отримують надбавку в розмірі 1 % від посадового окладу. Тоді надбавку можна обчислити за формулою $\text{Доплата} = \text{Оклад} * (\text{Стаж} - 5) / 100$.

Приклад 2

На основі таблиці КАДРИ створити запит з іменем Запит_5, за допомогою якого виводяться всі записи цієї таблиці з полями Прізвище, Стаж, Оклад і Доплата, значення якої обчислюється за вже наведеною формулою.

Порядок створення запити такого типу мало відрізняється від порядку звичайних запитів.

- Відкриваємо БД atb, активуємо вкладку Створення й натискаємо кнопку Макет запити. Із таблиці КАДРИ переносимо в Конструктор запити поля Прізвище, Стаж і Оклад, а в наступне поле уводимо вираз: $\text{Доплата} : [\text{Оклад}] * ([\text{Стаж}] - 5) / 100$. Звернемо увагу, що

імена полів, які входять у вираз, беруться у квадратні дужки.

- Установлюємо в рядку Сортвання поля Прізвище значення За зростанням, щоб прізвища виводилися в алфавітному порядку. Створений запит зображено на рис. 5.

Поле:	Прізвище	Стаж	Оклад	Доплата: [Оклад]*[(Стаж-5)/100]
Таблиця:	КАДРИ	КАДРИ	КАДРИ	
Сортвання:	За зростанням			
Відображення:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Критерій:				
Або:				

Рис. 5. Запит із полем, що обчислюється

- Зберігаємо запит з іменем Запит_5 (нагадуємо, що для цього натискається кнопка Зберегти, у вікні, що відкривається, вводиться ім'я запиту й натискається кнопка ОК). Унаслідок виконання запиту має з'явитися результат, зображений на рис. 6.

Прізвище	Стаж	Оклад	Доплата
Горошко Ф. Р.	17	4000	480
Раков Г. П.	19	4500	630
Рябко Р. П.	8	4200	126
Семко М. М.	16	4000	440
Сокил Т. Л.	27	3500	770
Таран В. Д.	15	4000	400
Шрамко Т. Л.	24	4000	760
*	0	0	

Рис. 6. Результат виконання запиту з полем, що обчислюється



- Закриваємо Запит_5.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

- ▲ Назвіть основні вбудовані функції для опрацювання даних у запиті.
- ▲ Які запити називають запитом з полями, що обчислюються?
- ▲ Яке призначення має функція Кількість?
- ◆ Які існують способи використання вбудованих функцій у запитах?
- ◆ Поясніть порядок створення запиту з підсумковим записом.
- ◆ Як додається до запиту підсумковий запис у режимі таблиці?
- ★ Поясніть порядок створення підсумкового запиту.
- ★ Поясніть порядок створення запиту з полями, що обчислюються.

**ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ**

1. ✦ Створіть Запит51 на основі таблиці УЧНІ, за допомогою якого підраховується кількість учнів, улюбленими предметами яких є географія та фізика. Записи повинні мати поля Прізвище, Клас, Улюблений предмет.
2. ✦ Створіть Запит52 на основі таблиці УЧНІ, за допомогою якого обчислюється середній бал успішності окремо з інформатики й історії учнів 11 класів.
-  3. ✦ Створіть Запит53 на основі таблиці УЧНІ, за допомогою якого обчислюється кількість учнів, що мешкають на вул. Луговій.
4. ★ Створіть Запит54 на основі таблиць КЛАСИ й УЧНІ, за допомогою якого обчислюється спільний середній бал успішності з інформатики й історії учнів класу, яким керує Дерев'яно Н. С.
-  5. ★ Створіть Запит55 на основі таблиці УЧНІ, за допомогою якого обчислюється середній зріст учнів кожного класу.
6. ★ Створіть Запит56 на основі таблиць КЛАСИ й УЧНІ, за допомогою якого для учнів 10 класу, улюбленим предметом яких є інформатика, обчислюється різниця середнього бала успішності з інформатики й історії.

3.4**ЗАПИТИ З ПАРАМЕТРАМИ. ПЕРЕХРЕСНІ ЗАПИТИ**

Ми вже розглядали запити з постійними критеріями, тобто запити, у процесі повторного виконання яких критерій відбору записів не змінювався. Але на практиці виникає необхідність змінювати ці критерії. Наприклад, під час першого виконання запиту необхідно вибрати з таблиці КАДРИ прізвища диспетчерів зі стажем роботи понад 15 років, а під час другого — прізвища диспетчерів зі стажем роботи більше 10 років. Такі дії можна виконувати за допомогою запитів із параметрами.



Запит з параметрами — це запит, у процесі виконання якого пропонується ввести деякі дані, наприклад умову, яку потрібно вставити в поле. Їх ще називають запитамі зі змінними критеріями.

За запитамі з параметрами на початку їх виконання на екран видаються повідомлення про необхідність введення нового критерію (виразу). Методика створення запиту такого типу мало відрізняється від методики створення звичайного запиту.

Приклад 1

Розробити запит з іменем Запит_6, за допомогою якого з БД atb вибираються прізвища працівників за спеціальністю «диспетчер» і «експерт» із магазинів, номери яких уводяться в процесі виконання запиту. Результуючі записи мають містити поля Номер магазину й Адреса таблиці МАГАЗИНИ та поля Прізвище й Посада таблиці КАДРИ.

Порядок розроблення запиту може бути таким.

1. Відкриваємо БД atb і виконуємо команду Створення → Макет запиту. Виділяємо обидві таблиці, натискаємо кнопку Додати й закриваємо вікно Відображення таблиці.
2. Із таблиці МАГАЗИНИ переносимо в таблицю конструктора поля Номер магазину й Адреса, а з таблиці КАДРИ — поля Прізвище та Посада.
3. В умові завдання визначено, що за допомогою запиту мають відбиратися записи лише за посадами «диспетчер» і «експерт», тобто ця умова є незмінною. Тому в комірку на перетині запису Критерії і поля Посада вводимо вираз "диспетчер" Or "експерт". Кожного разу після запуску на виконання запиту користувач може вводити будь-який номер магазину. Тому на перетині запису Критерії і поля Номер магазину можна увести, наприклад, такий текст [У яких магазинах?]. Головне в тому, щоб текст містився у квадратних дужках. У результаті отримаємо запит у режимі Конструктор, зображений на рис. 1.

Поле:	Номер магазину	Адреса	Прізвище	Посада
Таблиця:	МАГАЗИНИ	МАГАЗИНИ	КАДРИ	КАДРИ
Сортування:				
Відображення:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Критерії:	[У якому магазині?]			"диспетчер" Or "експерт"
Або:				

Рис. 1. Запит із параметром

4. Зберігаємо й виконуємо Запит_6. На екрані висвітиться вікно Введення значення параметра із запитанням У якому магазині? (рис. 2).

Введення значення параметра ? ×

У якому магазині?

Рис. 2. Вікно для введення значення параметра

Уведемо, наприклад, номер магазину 21 і натиснемо кнопку ОК. Отримаємо результат, зображений на рис. 3.

Номер магазину	Адреса	Прізвище	Посада
21	вул. Паркова, 33	Рябко Р. П.	експерт
21	вул. Паркова, 33	Семко М. М.	диспетчер
*			

Рис. 3. Результат виконання запиту з параметром

5. Проаналізуємо вміст таблиці КАДРИ та переконаємося, що дійсно в магазині з номером 21 за посадою «диспетчер» і «експерт» працюють, відповідно, Семко М. М. і Рябко Р. П.

Якщо потрібно здійснювати пошук прізвищ диспетчерів і експертів у кількох магазинах, то в запис Критерії поля Номер магазину необхідно увести іншу умову, а саме діапазон номерів магазинів, наприклад, >[Більше якого?] And <[Менше якого?]. У процесі виконання запиту спочатку з'явиться повідомлення Більше якого?, а потім — Менше якого?, на які необхідно дати певну відповідь.

Зазначимо, що один запит може містити кілька параметрів у різних полях.



Перехресний запит — це також запит на вибірку даних із можливостями групування записів. Групування можна виконувати як за значеннями полів, так і за значеннями записів.

Наприклад, із таблиці КАДРИ можна отримати кількість працівників усіх посад у кожному магазині. У записах можна мати назви посад, а в полях — номери магазинів. Заголовками полів можуть бути й деякі вирази.

Для створення перехресного запиту слід використати щонайменше три поля: поле для визначення заголовка записів, поле для визначення заголовка полів і поле для вибору значень, над якими будуть виконуватися обчислення. У режимі конструктора перехресний запит спочатку створюється як звичайний на вибірку даних, а потім установлюється режим перехресного запиту. Розглянемо порядок створення перехресного запиту на конкретному прикладі.

Приклад 2

Створити перехресний запит, за допомогою якого підраховується кількість працівників із кожної професії у кожному магазині.

1. Створюємо звичайний запит на вибірку даних у режимі Конструктор, для чого виконуємо команди Створити → → Макет запиту → Додати → Закрити. Перенесемо в таблицю Конструктора запиту поля Посада та Номер магазину. Виконуємо запит і переконаємося, що він функціонує правильно.
2. Перемикаємо запит у режим Конструктор та перетворюємо його на тип перехресний. Для цього в групі Тип запиту натискаємо кнопку Перехресний. У результаті в таблиці Конструктора запиту з'являться два нових записи: Підсумок і Перехресний. У записі Підсумок обох полів не змінюємо значення Групування за. Клацаємо клітинку на перетині поля Посада й запису Перехресний і вибираємо зі списку, що розкриється, Заголовок рядка, а в полі Номер магазину цього самого запису — Заголовок стовпця.

3. У третє поле таблиці Конструктора запиту переносимо поле Номер магазину, у записі Підсумок якого встановлюємо функцію Кількість, а в записі Перехресний — Значення. У записі Сортування першого поля можна встановити необхідне сортування записів. Таблиця Конструктора запиту буде мати вміст, зображений на рис. 4.

Поле:	Посада	Номер магазину	Номер магазину
Таблиця:	КАДРИ	КАДРИ	КАДРИ
Підсумок:	Групування за	Групування за	Кількість
Перехресний:	Заголовок рядка	Заголовок стовпця	Значення
Сортування:			Заголовок рядка
Критерій:			Заголовок стовпця
Або:			Значення
			(не відображається)

Рис. 4. Таблиця конструктора перехресного запиту



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. Які запити називають запитами з параметрами?
2. Які поля обов'язково використовуються в перехресних запитах?
3. Наведіть приклад запиту з параметрами.
4. Поясніть сутність перехресного запиту.
5. Поясніть порядок створення запиту з параметрами.
6. Поясніть порядок створення перехресного запиту.



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ



1. Створіть Запит61 із параметрами на основі таблиці УЧНІ, за допомогою якого виводяться прізвища учнів із різними улюбленими предметами, назви яких уводяться в процесі виконання запиту.
2. Створіть Запит62 із параметрами на основі таблиці УЧНІ, за допомогою якого виводяться прізвища учнів різного зросту, значення яких уводяться в процесі виконання запиту.
3. Створіть перехресний Запит63, за допомогою якого підраховується кількість учнів всіх класів із кожного улюбленого предмета.
4. Створити перехресний Запит64, за допомогою якого окремо в кожному класі обчислюється кількість учнів, зріст яких більше 163.
5. Створіть Запит65 із параметрами на основі таблиці УЧНІ, за допомогою якого виводяться прізвища учнів, улюблений предмет і номер класу яких уводяться в процесі виконання запиту.
6. Створіть перехресний Запит66, за допомогою якого в кожному класі обчислюється кількість учнів із кожного улюбленого предмета.

ЗАПИТИ НА ЗМІНЕННЯ

За допомогою запитів на змінення можна не лише вибирати з таблиць необхідні дані, а й із вибраних даних створювати нову таблицю, змінювати дані, наявні в таблицях, додавати нові записи в раніше створені таблиці, видаляти з таблиць записи.

Запити на змінення в Access 2016 поділяють на такі:

- оновлення даних;
- видалення даних;
- додавання нових записів;
- створення таблиць.

Порядок створення запитів на змінення в режимі конструктора такий самий, як і порядок створення звичайних запитів на вибірку даних, а потім створений запит перетворюється на один із перелічених типів. Далі розглянуто особливості розроблення запитів для створення нової таблиці й додавання записів у таблицю.

За допомогою запитів для створення нової таблиці вибираються дані з однієї або кількох таблиць і з них формується нова таблиця. Вона може бути розміщена як у поточній, так і в іншій БД, ім'я якої вказується в процесі створення запиту цього типу. Нова таблиця не має зв'язку з тими таблицями, з яких вона створена. Тому дані в ній автоматично не оновлюються в разі зміни в таблицях-джерелах.

Приклад 1

Розробити запит, за допомогою якого на основі даних таблиць МАГАЗИНИ та КАДРИ створюється нова таблиця з іменем ДОДАТКОВА, де містяться поля Справа, Прізвища, Рік народження і Працівників тих магазинів, у яких кількість працівників більше 14.

1. У відкритій БД атб виконуємо команду Створити → → Макет запиту, виділяємо обидві таблиці й переносимо з таблиці КАДРИ поля Справа, Прізвище і Рік народження, а з таблиці МАГАЗИНИ — поле Працівників. Закриваємо вікно Відображення таблиці. У запис Критерії поля Працівників уводимо вираз >14. Виконуємо запити та переконуємося, що отримано правильний результат.
2. Перемикаємо запит у режим Конструктор й перетворюємо запит на вибірку в запит на створення таблиці. Для цього на вкладці Конструктор у групі Тип запиту натискаємо кнопку Створення таблиці. Відкривається вікно Створити таблицю (рис. 1).
3. У поле Ім'я таблиці вводимо ім'я нової таблиці, наприклад, ДОДАТКОВА, і вмикаємо перемикач Поточна база даних, оскільки цю таблицю будемо зберігати

у відкритій (поточній) БД atb. Після цього натискаємо кнопку ОК.

Рис. 1. Вікно для створення таблиці

4. Для збереження запиту з іменем Запит_7 на панелі швидкого доступу натискаємо кнопку Зберегти.
5. Виконуємо Запит_7. Відкриється вікно, зображене на рис. 2.

Рис. 2. Вікно для підтвердження збереження таблиці

6. Для збереження таблиці в поточній БД слід натиснути кнопку Так. В області переходів з'явиться ім'я таблиці ДОДАТКОВА. Далі слід закрити запит. Відкриємо створену таблицю, вміст якої наведений на рис. 3.

ДОДАТКОВА			
Справа	Прізвище	Рік народження	Працівників
120	Рябко Р. П.	1981	20
111	Семко М. М.	1970	20
116	Раков Г. П.	1965	20
105	Сокил Т. Л.	1960	15
109	Шрамко Т. Л.	1961	15
*			

Рис. 3. Вміст таблиці ДОДАТКОВА

7. Проаналізуємо вміст таблиці й переконаємося, що вона сформована правильно.

Запити на додавання призначені для додавання нових записів у таблицю на основі опрацювання за певними критеріями даних, які вже є в раніше створених таблицях.

Проте зазначимо, що інколи для додавання всіх записів до всіх полів із поточної таблиці в нову доцільніше скористатися командами Копіювати та Вставити. Записи можна додавати як у відкриту, так і в закриту таблицю. Якщо записи додаються в таблицю іншої БД, то необхідно вказати ім'я та маршрут файла цієї БД. Якщо в записі Поле таблиці Конструктора запити є символ зірочка (*), то це означає, що використовувати в запиті окремі поля цієї самої таблиці неможливо.

Загальний порядок розроблення запитів цього типу такий:

- створюється запит на вибирання, за допомогою якого формуються необхідні записи для додавання;
- запит на вибирання перетворюється в запит на додавання;
- обирається таблиця, у яку будуть додаватися записи;
- запит зберігається та виконується.

Приклад 2

Створити Запит_8, за допомогою якого до таблиці ДОДАТКОВА додаються з таблиць МАГАЗИНИ та КАДРИ прізвища працівників магазинів, у яких працює 13 робітників, що народилися у 1975 році.

1. Створюємо запит на вибірку. Із таблиці МАГАЗИНИ переносимо в таблицю Конструктора запитів поля Справа, Прізвище та Рік народження, а з таблиці КАДРИ — поле Працівників. Виконуємо цей запит. Має з'явитися результат, наведений на рис. 4.

Справа	Прізвище	Рік народження	Працівників
115	Горошко Ф. Р.	1975	13

Рис. 4. Запит на вибірку даних із таблиць МАГАЗИНИ та КАДРИ

2. Перетворюємо створений запит на вибірку в запит для додавання, для чого переходимо в режим Конструктор і в групі Тип запити виконуємо команду Додавання.
3. Відкриється вікно Додавання, у якому вводимо ім'я таблиці ДОДАТКОВА, обираємо варіант Поточна база даних і натискаємо кнопку ОК.
4. Зберігаємо запит з іменем Запит_8 і виконуємо його. Відкриється вікно (рис. 5).

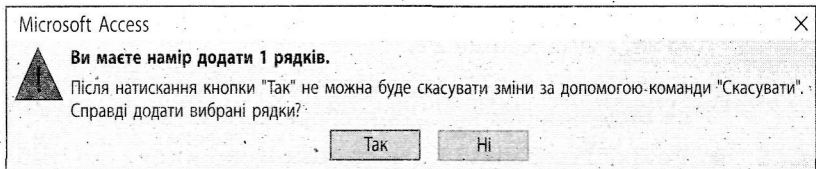


Рис. 5. Вікно для підтвердження додавання запису

5. Підтверджуємо додавання запису, для чого натискаємо кнопку Так. Закриваємо Запит 8. Відкриваємо таблицю ДОДАТКОВА. Має з'явитися таблиця, зображена на рис. 6.

Справа	Прізвище	Рік народж.	Працівників
120	Рябко Р. П.	1981	20
111	Семко М. М.	1970	20
116	Раков Г. П.	1965	20
105	Сокіл Т. Л.	1960	15
109	Шрамко Т. Л.	1961	15
115	Горошко Ф. Р.	1975	13
115	Горошко Ф. Р.	1975	13

Рис. 6. Таблиця після додавання запису



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. Які існують типи запитів на змінення?
2. Як запит на вибірку перетворюється в запит на створення таблиці?
3. Як запит на вибірку перетворюється в запит на додавання?
4. Як створюється запит для створення нової таблиці?
5. Поясніть сутність запитів на додавання нових записів.
6. Який існує загальний порядок створення запиту на додавання?
7. Наведіть приклад запиту для створення нової таблиці.
8. Наведіть приклад запиту на додавання.



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1. Розробіть Запит71, за допомогою якого на основі таблиці УЧНІ створюється нова таблиця ПЕРША1. Таблиця має містити дані про учнів 10 класу з полями Прізвище, Дата народження, Зріст, Улюблений предмет, Клас.
2. Створіть Запит72, за допомогою якого до таблиці ПЕРША1 додаються учні 9 класу з тими самими полями, що й у першому варіанті таблиці ПЕРША1. Збережіть таблицю з іменем ПЕРША2.
3. Створіть Запит73, за допомогою якого на основі даних таблиць КЛАСИ й УЧНІ створюється таблиця з іменем ПЕРША3 з полями Прізвище, Дата народження, Улюблений предмет, Клас для класів, у яких кількість учнів менше 27.
4. Розробіть Запит74, за допомогою якого до таблиці ПЕРША3 додається поле Інформатика. Дані збережіть у таблиці ПЕРША4.
5. Розробіть Запит75, за допомогою якого до таблиці ПЕРША4 додаються записи класу, де кількість учнів дорівнює 27 і які мають з історії успішність 9 балів.

Практична робота № 3

Тема: ЗАПИТИ З ФУНКЦІЯМИ ТА З ПОЛЯМИ,
ЩО ОБЧИСЛЮЮТЬСЯ

Завдання: розробити запити з функціями і з полями, що обчислюються.
Обладнання: комп'ютер зі встановленою СУБД Access 2016 і базою даних, створеною в попередніх роботах.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Створіть на основі таблиці ЧИТАЧІ Запит151, за допомогою якого вибираються дані про учнів 10 класу, що користувалися бібліотекою. Результуючий набір записів має містити поля Прізвище, Клас і Інвентарний номер.
2. Створіть на основі таблиці КНИЖКИ Запит152, за допомогою якого вибираються записи про книжки, автором яких є Мехеда О. Н. Результуючий набір записів має містити поля Автор, Назва й Рік видання.
3. Створіть на основі таблиці КНИЖКИ Запит153, за допомогою якого вибираються записи про книжки ціною в межах від 35 до 45 грн. Результуючий набір записів має містити поля Автор, Назва, Ціна.
4. Модифікуйте Запит153 так, щоб вибиралися книжки ціною від 40 до 50 грн. Збережіть його з іменем Запит154.
5. Створіть на основі таблиць КНИЖКИ та ЧИТАЧІ Запит155, за допомогою якого вибираються дані про учнів 11 класу, що користувалися в бібліотеці книжками з інформатики. Результуючий набір записів має містити поля Прізвище, Клас, Назва, Автор.
6. Створіть на основі таблиць КНИЖКИ й ЧИТАЧІ Запит156, за допомогою якого формуються записи про учнів 10 класу, що користувалися книжками авторів Соломко І. М. і Мехеда О. Н. Результуючий набір записів має містити поля Прізвище, Клас, Назва, Автор.
7. Створіть Запит157, за допомогою якого формуються записи про читачів, що не повернули в бібліотеку книжки. Результуючий набір записів має містити поля Прізвище, Назва, Автор.
8. Створіть Запит158, за допомогою якого підраховується кількість книжок за їх назвами.
9. Створіть Запит159, за допомогою якого підраховується кількість звернень до бібліотеки учнями 10 і 11 класів.
10. Створіть Запит160, за допомогою якого обчислюється середня ціна книжок із фізики й інформатики.
11. Створіть Запит161, за допомогою якого обчислюється кількість екземплярів книжок кожного автора.

Зробіть висновок про результати роботи.

Практична робота № 4

- Тема:** ЗАПИТИ З ПАРАМЕТРАМИ, ПЕРЕХРЕСНІ ЗАПИТИ І ЗАПИТИ НА ЗМІНЕННЯ ДАНИХ
- Завдання:** розробити запити з параметрами, перехресні запити та запити на змінення даних.
- Обладнання:** комп'ютер зі встановленою СУБД Access 2016 і базою даних, створеною в попередній роботі.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Розробіть Запит171 з параметрами, за допомогою якого вибираються назви й автори книжок за роком їх видання. Рік видання уводиться в процесі виконання запиту.
2. Розробіть Запит172 з параметрами, за допомогою якого вибираються прізвища читачів, що користувалися книжками з певними інвентарними номерами. Інвентарні номери вводяться у процесі виконання запиту.
3. Розробіть Запит173 з параметрами, за допомогою якого вибираються назви й ціна книжок за прізвищем автора.
4. Розробіть Запит174, за допомогою якого вибираються автори та назви книжок, якими користувалися читачі, за датами їх видачі.
5. Розробіть перехресний Запит175, за допомогою якого підраховується кількість різних назв книжок за роками їх видання.
6. Розробіть перехресний Запит176, за допомогою якого підраховується кількість звернень до бібліотеки кожного читача.
7. Розробіть Запит177, за допомогою якого створюється нова таблиця ДОДАТОК1, що містить дані про книжки з полями Назва, Автор, Прізвище, Клас.
8. Розробіть Запит178, за допомогою якого створюється таблиця ДОДАТОК2, що містить дані про книжки, видані у 2017 і 2018 роках. Таблиця має містити поля Назва, Автор, Дата видачі, Дата повернення.
9. Розробіть Запит179, за допомогою якого до таблиці ДОДАТОК2 додаються дані про книжки, видані у 2016 році.

Зробіть висновок про результати роботи.

СТВОРЕННЯ ФОРМ ЗА ДОПОМОГОЮ ПРОСТИХ ЗАСОБІВ

Форма — це об'єкт бази даних Access 2016, головне призначення якого — забезпечити користувачам зручне введення, редагування й видалення даних із бази даних.

Виникає питання: навіщо потрібні форми, якщо дані можна вводити безпосередньо в таблиці? Так, дійсно, дані зберігаються в таблицях, у яких безпосередньо можна вводити, редагувати й видаляти дані. Але не кожному користувачеві має надаватися таке право. Більше того, для забезпечення цілісності та безпечності БД як структури бази даних, так і самі таблиці не мають бути доступними для більшості користувачів. Наприклад, диспетчер аеропорту може внести до БД дані про те, що літак рейсу 1025 здійснив посадку в аеропорту о 12.00. Але він не має права змінити номер рейсу або його маршрут. Це право належить власнику компанії і адміністрації аеропорту. Так само продавець супермаркету може внести дані про реалізацію конкретного товару, але він не може змінити ціну товару, це право належить адміністрації супермаркету.

Форма — це фактично електронний бланк, який для кожної групи користувачів має свою структуру. У цій структурі містяться назви тих полів, які доступні для певної групи користувачів. Після заповнення конкретними даними цих полів дані потрапляють у відповідні поля таблиць для подальшого збереження.

Отже, форми, з одного боку, забезпечують дані від несанкціонованого доступу, а з іншого — спрощують сам процес уведення даних, оскільки користувачу не треба переглядати таблицю з десятками полів і відшукувати потрібні. Йому вже пропонується бланк, наприклад, із двома-трьома полями, у які він має право внести необхідні дані. Важливо й те, що за допомогою форми дані можуть оновлюватися одночасно в кількох пов'язаних таблицях.

Форма може складатися з багатьох елементів, які називаються елементами керування (написи, прапорці, перемикачі, графічні примітиви).

Форма має такі основні розділи.

1. **Область заголовка** — це верхня частина форми. В ній можна розмістити текст, графіку й решту елементів керування. Розділ заголовка розміщується тільки на першій сторінці.
2. **Область даних** — це основний розділ, який містить дані. Він може містити також елементи керування.
3. **Область примітки** — розміщується на останній сторінці форми.

Система Access 2016 має різні засоби створення форм, які розміщені на вкладці Створення в групі Форми (рис. 1).

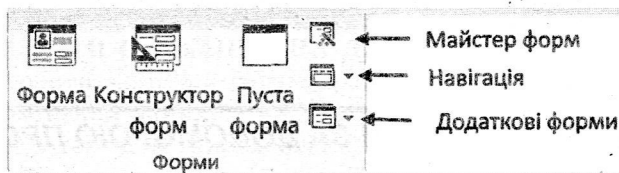


Рис. 1. Засоби створення форм

Отже, існують такі засоби створення форм: Форма, Конструктор форм, Пуста форма, Майстер форм і Додаткові форми. Додаткові форми мають такі засоби: Кілька елементів, Таблиця, Розділена форма та Модальне діалогове вікно. Усі перелічені засоби, крім Конструктора форм, є простими. Простими можна назвати засоби автоматичного створення форм, за допомогою яких можна створити форму фактично натисканням однієї кнопки. Найпотужнішим інструментарієм створення форм є Конструктор форм.

Далі розглядається порядок створення форм за допомогою засобів Форма, Розділена форма, Кілька елементів і Майстер форм.

Засіб Форма. За допомогою цього засобу створюється форма з усіма полями таблиці або запиту. У формі в кожен момент часу відображається лише один запис. Для створення форми цим засобом необхідно в області переходів виділити (або відкрити) таблицю або запит і натиснути кнопку Форма.

Приклад 1

Створити форму з іменем Форма_1 для таблиці ДОДАТКОВА. Згадаємо, що ця таблиця є складовою БД atb і не має зв'язків з іншими таблицями бази даних.

1. Відкриваємо базу даних atb і в області переходів виділяємо (або відкриваємо) таблицю ДОДАТКОВА. На вкладці Створення в групі Форми натискаємо кнопку Форма. У результаті відкриється вікно форми.

2. Зберігаємо форму, для чого на панелі швидкого доступу виконуємо команду Зберегти. У вікні, що відкриється, вводимо ім'я Форма_1 і натискаємо кнопку ОК. Форма повинна мати вміст, зображений на рис. 2.

Ім'я Форма_1 з'явиться в області переходів. У формі відображається перший запис таблиці ДОДАТКОВА. У нижній частині форми розміщено елементи навігації, за допомогою яких можна перейти до будь-якого запису таблиці й виконати необхідне редагування записів.

3. Закриваємо форму. Після цього її можна викликати в будь-який час, для чого слід установити курсор на імені форми й двічі клацнути кнопкою миші.

Форма_1

ДОДАТКОВА

Справа: 120

Прізвище: Рябко Р. П.

Рік народження: 1981

Працівників: 20

Запис: К 1 з 7 | Без фільтрів | Пошук

Рис. 2. Форма для таблиці ДОДАТКОВА

Порядок створення форми з іменем Форма_2 для таблиці МАГАЗИНИ принципово нічим не відрізняється від розглянутого раніше. Результат наведено на рис. 3.

Форма_2

МАГАЗИНИ

Номер магазину: 6

Адреса: вул. Річкова, 24

Директор: Серета К. М.

Телефон: 234-67-92

Працівників: 15

Справа	Прізвище	Посада	Рік народження	Освіта	Стаж	Оклад
105	Соціл Т. Л.	касір	1960	середня	27	3500
109	Шрамко Т. Л.	диспетчер	1961	середня	24	4000
*	0		0		0	0

Запис: К 1 з 2 | Без фільтрів | Пошук

Рис. 3. Форма для таблиці МАГАЗИНИ

До форми виведено запис магазину № 6 таблиці МАГАЗИНИ, під нею — усі записи таблиці КАДРИ, які пов'язані з виведеним записом таблиці МАГАЗИНИ. Це зумовлене тим, що між зазначеними таблицями встановлено зв'язок.

Засіб Розділена форма. Цей засіб міститься в групі Додаткові форми. Форма за допомогою цього засобу також створюється на основі таблиці або запити, виділеного (або відкритого) в області переходів. За цим засобом дані одночасно подаються у вигляді кількох записів таблиці та полів одного запису, який можна редагувати.

Приклад 2

Створити форму з іменем Форма_3 для запити з іменем Запит_2. Згадаємо, що за допомогою Запиту_2 з таблиць МАГАЗИНИ й КАДРИ відбиралися дані про диспетчерів із полями Номер магазину, Телефон, Прізвище, Посада й Освіта.

1. Виділяємо в області переходів Запит_2.
2. На вкладці Створення в групі Форми відкриваємо меню кнопки Додаткові форми, у якому виконуємо команду Розділена форма.
3. Зберігаємо форму з іменем Форма_3. Вміст цієї форми зображено на рис. 4.

Форма_3

Запит_2

Номер магазину 31

Телефон 234-22-98

Прізвище Горошко Ф. Р.

Посада диспетчер

Освіта середня спеціальна

Номер магазину	Телефон	Прізвище	Посада	Освіта
31	234-22-98	Горошко Ф. Р.	диспетчер	середня спеціальна
21	234-54-63	Семко М. М.	диспетчер	середня
31	234-22-98	Таран В. Д.	диспетчер	вища
6	234-67-92	Шрамко Т. Л.	диспетчер	середня

Записи: 1 з 4 | Без фільтра | Пошук

Рис. 4. Форма, створена на основі Запит_2

У нижній частині розташовані всі записи, створені за допомогою Запит_2, а у верхній — запис, який є виділеним у нижній частині та який у цей момент можна редагувати. Стандартними способами можна виділити будь-який інший запис.

Засіб Кілька елементів. За допомогою цього засобу створюється форма з кількома записами. Зовнішньо форма схожа на таблицю. За замовчуванням вона виводиться в режимі Форми, але її можна перевести в режим Конструктор й додати деякі елементи керування.

Приклад 3

Створити форму з іменем Форма_4 на основі Запит_1. Згадаємо, що за допомогою цього запиту з таблиці КАДРИ вибираються дані про працівників зі стажем понад 16 років із полями Справа, Прізвище, Рік народження, Стаж, Номер магазину.

1. В області переходів виділяємо Запит_1.
2. На вкладці Створити в групі Форми відкриваємо меню кнопки Додаткові форми й виконуємо команду Кілька елементів.

3. Зберігаємо форму з іменем Форма_4. У результаті форма відкривається в режимі Форми, яка зовні схожа на таблицю (рис. 5).

Справа	Прізвище	Рік народження	Стаж	Номер магазину
105	Сокіл Т. Л.	1960	27	6
115	Горошко Ф. Р.	1975	17	31
116	Раков Г. П.	1965	19	21
109	Шрамко Т. Л.	1961	24	6

Записи 1-4 | Пошук

Рис. 5. Форма, створена засобом Кілька елементів на основі Запит_1

Засіб Майстер форм. Він дає змогу вибирати з таблиць необхідні поля. Користувач у процесі створення форми відповідає на запитання від майстра, визначає властивості форми, порядок компонування форми (у стовпець, таблична й інші).

Приклад 4

За допомогою Майстра форм створити форму з іменем Форма_5 для таблиці МАГАЗИНИ. У форму включити поля Номер магазину, Адреса, Директор, Телефон.

1. В області переходів виділяємо таблицю МАГАЗИНИ, активізуємо вкладку Створення й натискаємо кнопку Майстер форм. Відкриється перше вікно майстра. У рядку Таблиці та запити вибираємо таблицю МАГАЗИНИ, на основі якої буде створено форму. З області Доступні поля переносимо в область Вибрані поля такі поля: Номер магазину, Адреса, Директор, Телефон (подвійним клацанням цих імен). Після цього вікно майстра буде мати вигляд, зображений на рис. 6.
2. Натискаємо кнопку Далі. Відкриється вікно, у якому вибираємо макет форми, наприклад, Стовпці, й натискаємо кнопку Далі.
3. У наступному вікні Майстра форм у поле Виберіть назву для форми вводимо назву Форма_5 і використовуємо перемикач Відкрити форму для перегляду й введення даних. Після цього натискаємо кнопку Готово. Форма буде автоматично збережена й відкриється в режимі Форми. Її вміст зображено на рис. 7.

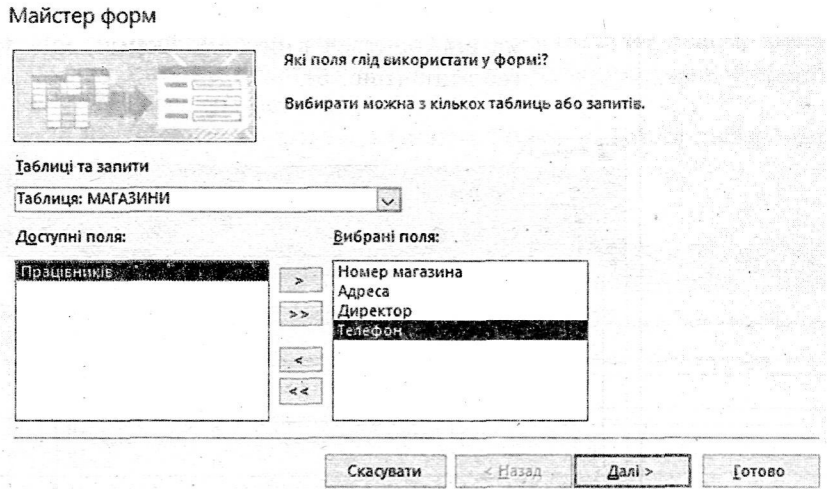


Рис. 6. Вікно Майстра форм

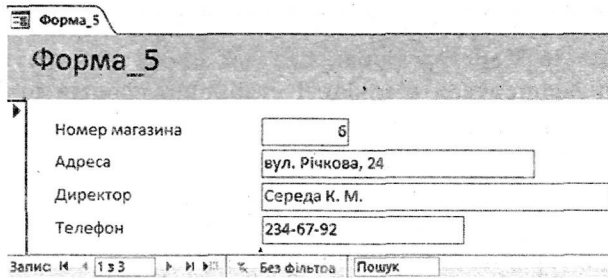


Рис. 7. Форма, створена за допомогою Майстра форм



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1. ▲ Для чого призначені форми?
2. ▲ Які елементи керування може містити форма?
3. ▲ Які основні розділи має форма?
4. ▲ Які засоби створення форм є простими?
5. ★ Поясніть порядок створення форм за допомогою засобу Форма.
6. ★ Поясніть особливість створення форм за допомогою засобу Розділена форма.
7. ★ Обґрунтуйте необхідність застосування форм.
8. ★ Які переваги має Майстер форм над іншими простими засобами?



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1. ▲ За допомогою засобу Форма створіть Форму_51 для таблиці ПЕРША1.
2. ▲ За допомогою засобу Форма створіть Форму_52 для таблиці КЛАСИ.

3. ✦ За допомогою засобу Роздільна форма створіть Форму_53 для Запиту_71.
4. ✦ За допомогою засобу Кілька елементів створіть Форму_54 для Запиту_72.
5. ★ За допомогою Майстра форм створіть Форму_55 для таблиці КЛАСИ. У форму включіть поля Клас, Учнів, Класний керівник.
6. ★ За допомогою Майстра форм створіть Форму_56 для таблиці УЧНІ. У форму включіть поля Прізвище, Адреса, Дата народження.

4.2

ЕЛЕМЕНТИ КЕРУВАННЯ ТА ВЛАСТИВОСТІ ФОРМ

Усі створені раніше форми не містили жодних елементів керування, і їх властивості не встановлювалися. Це означає, що зовнішній вигляд форм установлювався за замовчуванням. Але кожен з них можна перевести в режим Конструктор, після чого стануть доступні елементи керування, аркуш властивостей форм і аркуш властивостей елементів керування.

Елементи керування можна розмістити на формі й установити як їх властивості, так і властивості форми. Це, по-перше, сприяє полегшенню роботи користувача, а по-друге, надає формі естетичного та привабливого зовнішнього вигляду.

Перелік елементів керування можна відкрити так. Відкрити форму, наприклад, Форму_1, у режимі Конструктор, на смужці з'явиться група **Елементи керування**, у якій слід відкрити меню однойменної кнопки. Унаслідок цього відкриється вікно з елементами керування, зображене на рис. 1.

На рисунку вказано назви деяких елементів керування. Для отримання назв інших елементів достатньо встановити на них курсор. Опис найчастіше вживаних елементів керування наведено в табл. 1.



Рис. 1. Елементи керування форм

Таблиця 1. ОПИС НАЙЧАСТІШЕ ВЖИВАНИХ ЕЛЕМЕНТІВ КЕРУВАННЯ

Тип	Призначення
Текстове поле	У це поле можуть вводитися дані з клавіатури
Підпис	Поле для виведення тексту
Кнопка	Дає змогу виконати певну команду
Прямокутник	Використовується для виділення елементів керування
Лінія	Пряма лінія відповідного кольору й товщини
Прапорець	Призначений для встановлення певного режиму
Перемикач	Призначений для вибору зі списку значень параметрів або тільки одного режиму із цієї групи
Посилання	Дає змогу звернутися до веб-сторінок і файлів, а також перейти до певних місць у документі

Додати елемент керування на форму можна, наприклад, так: у режимі Конструктор встановити курсор на імені певного елемента й натиснути кнопку миші та виконати аналогічну операцію в області Форми.

Кожен елемент керування має власний набір властивостей. Цей набір є досить великим і рідко використовується в повному обсязі. У більшості випадків властивості встановлюються за замовчуванням.

Властивості — це параметри елементів керування з певними іменами.

Елемент керування може мати, наприклад, такі параметри: колір, розмір, товщина, розмір шрифту й ін.

Для виведення на екран списку значень властивостей елемента керування у відкритій формі достатньо клацнути на ньому кнопку миші. Відкриємо, наприклад, властивості елемента Лінія.

Аркуш властивостей цього елемента зображено на рис. 2.

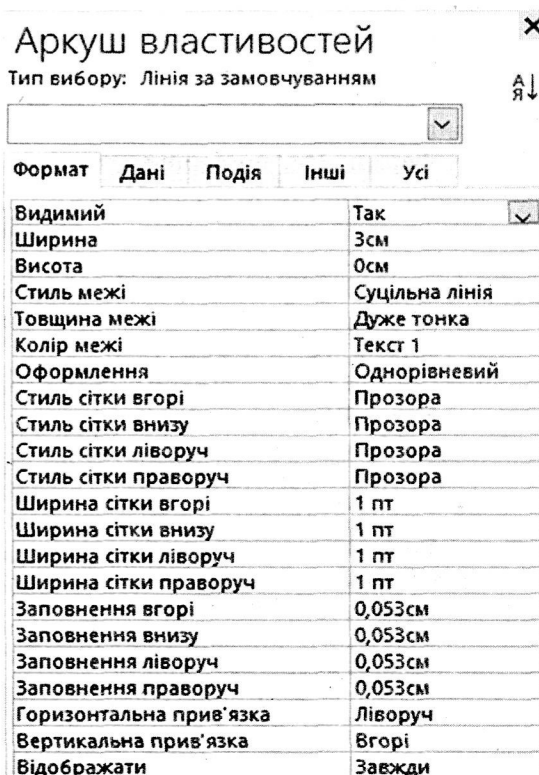


Рис. 2. Аркуш властивостей елемента Лінії

Вікно властивостей має п'ять вкладок. Вкладка Усі містить повний набір параметрів елемента керування, а інші вкладки — скорочений їх перелік.

Властивості форми часто використовують для налаштування її зовнішнього вигляду. Повний їх склад міститься у вікні Аркуш властивостей форми (рис. 3).

Для відкриття цього вікна необхідно у відкритій у режимі Конструктор форми встановити курсор на вільному місці форми (не на об'єкті форми) й натиснути кнопку миші.

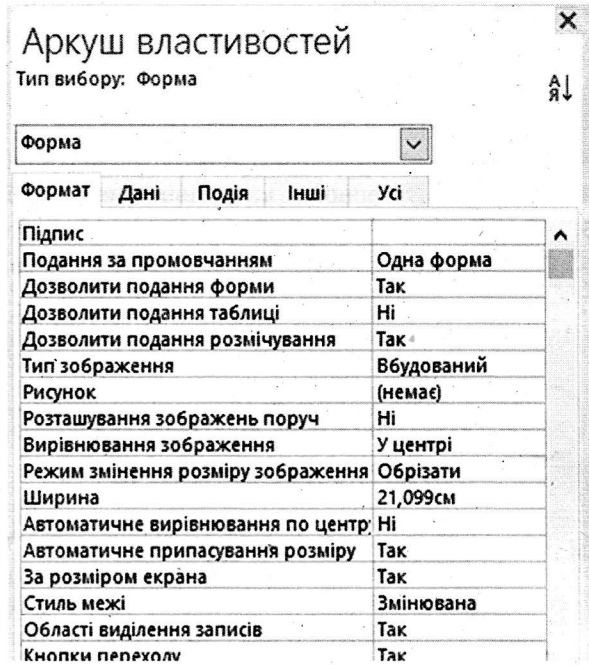


Рис. 3. Аркуш властивостей Форми

Назви та опис деяких властивостей цього вікна наведено в табл. 2.

Таблиця 2. ОПИС ЧАСТО ВЖИВАНИХ ВЛАСТИВОСТЕЙ ФОРМИ

Властивість	Опис	Значення
Підпис	Речення, яке виводиться в рядку заголовка форми в режимі форми	Текст
Подання за промовчанням	Режим виведення форми	Одна форма, Розділена форма
Дозволити подання форми	Форма виводиться в режимі форми	Так, Ні
Дозволити подання розмічування	Форма виводиться в режимі розмічування	Так, Ні
Кнопки переходу	Виведення кнопок навігації по записах	Так, Ні
Лінії-роздільники	Виведення роздільних ліній між роздільними формами	Так, Ні
Смуги прокручування	Виводиться смуга прокручування	Обидві, Немає, Лише горизонтальний, Лише вертикальний
Вирівнювання зображення	Форма автоматично центрується на екрані	Так, Ні



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. Як можна відкрити елементи керування формою?
2. Назвіть основні елементи керування формою.
3. Як можна відкрити аркуш властивостей форми?
4. Назвіть основні властивості елементів керування.
5. Для чого призначено елемент керування Текстове поле?
6. Як можна розмістити елемент керування на формі?

4.3

СТВОРЕННЯ ФОРМИ ЗА ДОПОМОГОЮ КОНСТРУКТОРА ФОРМ

Конструктор форм є найпотужнішим інструментарієм створення форм. За його допомогою можна здійснювати налаштування зовнішнього вигляду форми й елементів керування, які розташовані на ній, змінювати колір фону, тексту та стилю зображення.

За допомогою конструктора форм найчастіше створюють форму із такими розділами:

- заголовок форми;
- верхній колонтитул;
- область даних;
- нижній колонтитул;
- примітка до форми.

Розглянемо порядок створення форми за допомогою конструктора форми на конкретному прикладі.

Приклад

Створити для бази даних atb на основі таблиці МАГАЗИНИ форму з іменем Форма_6 із полями Номер магазину, Адреса, Директор, Телефон. Перед першими двома полями розмістити текст Координати, а перед двома іншими — Контакти.

1. Виділяємо в області переходів таблицю МАГАЗИНИ, активізуємо вкладку Створити й у групі Форми виконуємо команду Конструктор форм. На екрані з'явиться порожня форма з розділом Подробиці й буде активована вкладка Конструктор. У групі Колонтитули натискаємо кнопку Назва. У результаті відкриється вікно (рис. 1), у якому розташовано підрозділ Верхній колонтитул форми, підрозділ Нижній колонтитул форми і текстове поле з назвою Форма1.
2. Установлюємо курсор у поле Форма1 і вводимо текст заголовка, наприклад, Магазины нашего району, та натискаємо кнопку Enter.

3. Уводимо у верхню частину області Подробиці текст Координати, який, відповідно до умови завдання, має розташовуватися перед першими двома полями. Для цього натискаємо кнопку Напис, встановлюємо курсор у необхідну позицію області Подробиці, уводимо текст Координати й натискаємо кнопку Enter.

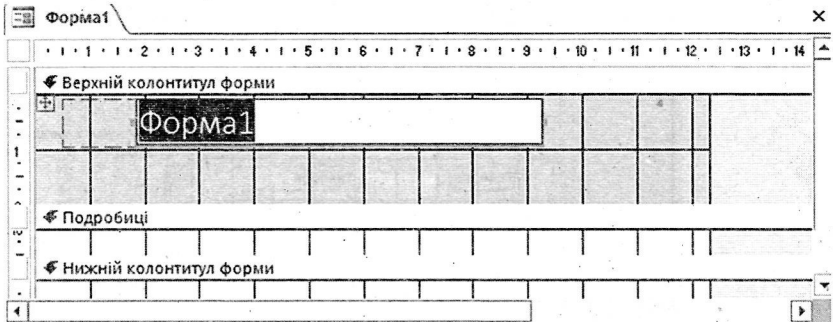


Рис. 1. Порожня форма в режимі Конструктор

4. У правій частині вікна має бути розташоване поле Список полів. Якщо воно відсутнє, необхідно натиснути кнопку Додавання наявних полів, і тоді це поле з'явиться. У Списку полів натискаємо кнопку Відобразити всі таблиці. Ліворуч від назви таблиці МАГАЗИНИ встановлюємо перемикач у положення (-). Під назвою цієї таблиці відкриється список її полів. Перемістимо поле Номер магазину в те місце розділу Подробиці, куди необхідно його помістити. У результаті на формі буде розташовано два пов'язані елементи: власне поле введення та підпис для нього (підпис збігається з іменем цього поля). Так само перемістимо в бланк форми поле Адреса.
5. Після двох перших полів розміщуємо назву двох останніх полів. Для цього за аналогією з попереднім у групі Елементи керування натискаємо кнопку Напис, встановлюємо курсор у необхідну позицію області Подробиці, вводим назву Контакти й натискаємо кнопку Enter. Перемістимо в розділ Подробиці поля Директор і Телефон. Розміщуємо елементи форми так, щоб вона мала гарний вигляд, наприклад, як на рис. 2.
Після цього можна закрити вікно Список полів. Якщо у форму необхідно додати поля з кількох таблиць, слід розкрити їх поля так само, як і поля таблиці МАГАЗИНИ.
6. Збережемо форму з іменем Форма_6 так само, як зберегалися форми, створені іншими засобами.



Рис. 2. Форма, створена в режимі Конструктор

Після цього форму можна переглянути в таких режимах: Режим форм, Подання таблиці, Режим розмітки. На рис. 3 створена форма відкрита в режимі Форми.

За допомогою кнопок навігації можна перейти до будь-якого запису. Зовнішній вигляд форми можна налаштувати відповідно до потреб користувача. Для цього слід відкрити контекстне меню форми (згадаємо, що для цього слід установити курсор на вільному місці форми, відкритої у режимі Конструктор, і натиснути праву кнопку миші). Вміст контекстного меню форми зображено на рис. 4.

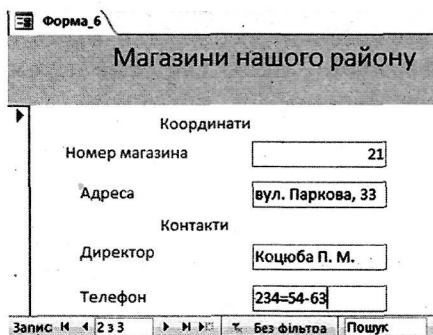


Рис. 3. Форма, відкрита в режимі Форми

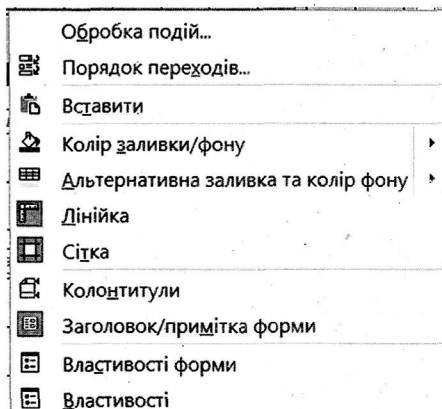


Рис. 4. Контекстне меню форми

Для зміни кольору фону форми слід установити курсор миші на команді Колір заливки/фону. Відкриється набір кольорів, у якому

вибираємо необхідний. У результаті колір фону буде інший. Аналогічно можна відкрити контекстне меню Заголовок форми й інших її компонентів і встановити потрібні кольори. Можна також встановити або зняти на формі лінійку, сітку та виконати інші налаштування. Один із варіантів налаштування зовнішнього вигляду форми зображено на рис. 5.

Рис. 5. Варіант оформлення зовнішнього вигляду форми

Зазначимо, що порядок створення форм у режимі Конструктор на основі запитів принципово не відрізняється від порядку її створення на основі таблиць.

Форму, як і інші об'єкти бази даних, бажано проектувати так, щоб потім не потрібно було вносити зміни до її структури. І все ж із часом виникає необхідність внесення певних змін.

Access 2016 створює можливість виконувати такі дії з модифікування форм:

- змінювати параметри відображення форми;
- додавати нові елементи керування;
- змінювати властивості форми;
- видаляти елементи керування;
- змінювати властивості елементів керування.



Наведені операції, як правило, виконуються в режимі Конструктора форм на основі описаних правил їх створення.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. ▲ Які розділи містить форма, створена за допомогою Конструктора форм?
2. ▲ Як можна відкрити список полів таблиць?
3. ▲ Як відкривається контекстне меню форми?
4. ★ Як можна змінити колір фону форми?
5. ★ У яких режимах можна переглянути створену форму?
6. ★ Поясніть порядок уведення заголовка форми.
7. ★ Які операції можна виконувати з модифікації форми?

**ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ**

1. ✦ Створіть за допомогою Конструктора форм для таблиці КЛАСИ Форму_61 із полями Клас, Учні, Займаються спортом, Класний керівник. Перед першими трьома полями розмістіть текст Класи і учні, а перед останнім — Керівник. Заголовок форми — Класи школи. Збережіть форму.
2. ✦ Перегляньте Форму_61 у різних режимах. Здійсніть навігацію по записах форми. Використайте Аркуш властивостей форми для надання їй гарного зовнішнього вигляду.
-  3. ✦ Створіть за допомогою Конструктора форм для таблиці УЧНІ Форму_62 із полями Прізвище, Клас, Улюблений предмет, Інформатика, Історія. Перед першими двома полями розмістіть назву Учні, а перед останніми трьома — Предмети. Заголовок форми — Успіхи учнів.
4. ✦ Для Форми_62 самостійно налаштуйте її зовнішній вигляд.
-  5. ✦ Створіть за допомогою Конструктора форм для таблиць КЛАСИ й УЧНІ Форму_63 із полями Клас, Класний керівник, Прізвище, Адреса, Дата народження. Перед першими двома полями розмістіть текст Класи, а перед трьома останніми — Учні. Заголовок форми — Наша школа.
6. ✦ Перегляньте форму в різних режимах. Самостійно виберіть колір кожного компонента форми.

4.4**ВИКОРИСТАННЯ ФОРМ ДЛЯ ВВЕДЕННЯ Й РЕДАГУВАННЯ ДАНИХ**

Форма — це гарний засіб відображення, уведення й редагування даних, що зберігаються в таблицях або формуються запитамі. Але форма не призначена для створення таблиць і наповнення їх даними. Таблиці створюються розробниками БД, а потім у процесі роботи з БД різним групам користувачів надається право вносити зміни в окремі поля таблиці.

Саме для цього й призначені форми. Наприклад, провізори аптек можуть вносити дані про купівлю певних ліків. Завідувач аптеки може вносити зміни про кількість ліків, що надійшли до аптеки, а її керуючий може вносити зміни на ціни ліків.

Далі стисло розглядається порядок додавання, редагування й видалення записів із таблиці за допомогою форм на конкретних прикладах.

1. Додавання нових записів.

Наприкінці кожної форми є порожній запис. До нього можна перейти, натиснувши кнопку Створити запис, що розташована на панелі навігації, або іншим шляхом, наприклад, послідовним натисканням на кнопку Наступний запис. Для введення нового

запису необхідно перейти на порожній запис, заповнити його поля необхідними даними та натиснути клавішу Enter. Після цього можна перейти до введення наступного запису.

Після завершення введення всіх записів необхідно зберегти форму. Уведені у форму записи будуть автоматично додані до таблиць, на базі яких створена форма. Однак, перш ніж вводити нові записи, необхідно переконатися, що у вікні Аркуш властивостей параметра Дозволити додавання надано значення Так. Інакше записи вводитися в таблицю не будуть.

Розглянемо порядок введення (додавання) нового запису в таблицю на прикладі Форми_6. Згадаємо, що Форма_6 створювалася на основі таблиці МАГАЗИНИ. Відкриємо Форму_6 у режимі Форми. Перейдемо на порожній запис, для чого натиснемо, наприклад, кнопку Створити запис, розташовану на панелі навігації. Уведемо в цю форму в порожній рядок, наприклад, такі дані: Номер магазину — 45, Адреса — вул. Гончара, 3, Директор — Костюк В. А., Телефон — 234-11-11 (рис. 1).

Рис. 1. Запис, доданий у таблицю

Після цього збережемо форму й закриємо її. Таблиця МАГАЗИНИ, відкрита в режимі подання таблиці, набуде змісту, зображеного на рис. 2.

МАГАЗИНИ						
	Номер магазину	Адреса	Директор	Телефон	Працівників	Клацніть, щоб додати
	6	вул. Річкова, 24	Середа К. М.	234-67-92	15	
	21	вул. Паркова, 33	Коцюба П. М.	234-54-63	20	
	31	вул. Печерська, 21	Борзов А. С.	234-50-50	13	
	45	вул. Гончара, 3	Костюк В. А.	234-11-11	0	
*	0				0	

Рис. 2. Зміст таблиці МАГАЗИНИ після доданого запису

Звернемо увагу, що поле Працівників для введеного запису є порожнім, оскільки у Формі_6 такого поля створено не було й дані до нього не вводилися.

2. Редагування записів.

Для редагування запису в таблиці за допомогою форми необхідно спочатку знайти цей запис у формі, відкритій у режимі Форми, а потім внести необхідні зміни у відповідні поля. Пошук запису можна здійснити за допомогою кнопок панелі навігації. Якщо відомий номер запису, до якого необхідно внести зміни, то цей номер вводиться в поле поточного запису й натискається кнопка Enter.

Можна також скористатися кнопкою Пошук на панелі навігації. Наприклад, якщо увести прізвище Костюк В. А., то автоматично висвітяться дані запису з цим прізвищем. Після внесення відповідних змін у запис форму необхідно зберегти. Уведемо в поле Адреса цього запису нову адресу: вул. Вільна, 7. Зберігаємо та закриваємо форму. Після цього відкриємо таблицю МАГАЗИНИ й переконаємося, що зміни внесені правильно.

3. Видалення записів.

Видалити запис із таблиці за допомогою форми можна, наприклад, так: знаходимо у формі запис, який необхідно видалити, у групі Запис вкладки Основне відкриваємо меню кнопки Видалити й виконуємо команду Видалити запис.

Видалимо, наприклад, за допомогою Форми_6 із таблиці МАГАЗИНИ запис, де є прізвище Костюк В. А. Для цього відкриваємо Форму_6 у режимі Форми. Знаходимо запис із прізвищем Костюк В. А. У групі Записи відкриваємо меню кнопки Видалити (Del) і виконуємо команду Видалити запис. У результаті на екрані з'явиться вікно, де необхідно підтвердити видалення цього запису. Після цього слід зберегти форму. Відкриємо таблицю МАГАЗИНИ й переконаємося, що запис із прізвищем Костюк В. А. дійсно видалений.

Слід урахувати, що для зв'язаних таблиць видалити запис не завжди вдається. Якщо запис не має підлеглих записів у зв'язаних таблицях або дозволене каскадне видалення записів, то видалення можливе. У такому випадку відкривається вікно з проханням підтвердити видалення. Для підтвердження видалення слід натиснути кнопку Так, інакше буде виведено повідомлення про неможливість видалення. Якщо ж запис має підлегли записи у зв'язаних таблицях, то необхідно видалити з інших таблиць усі записи, які з ним пов'язані.

Зазначимо, що форми створюють можливість переглядати й прослуховувати об'єкти типу OLE. Поля з даними цього типу можуть бути малюнками, звуковими файлами, файлами Word або Excel. Вставити в поле форми об'єкт OLE можна, наприклад, так: скопіювати об'єкт OLE у буфер обміну, а потім натиснути кнопку Вставити, яка розташована в розділі Буфер обміну.



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1. Як здійснюється перехід на порожній запис у формі?
2. Як можна змінити необхідний запис за допомогою форми?
3. Поясніть порядок додавання записів у таблиці за допомогою форми.
4. Поясніть порядок видалення запису за допомогою форми.
5. Для чого призначене поле типу OLE?



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1. Відкрийте Форму_61 у режимі Форми. Додайте новий запис: Клас — 8, Учнів — 25, Займаються спортом — 6. Збережіть форму. Відкрийте таблицю КЛАСИ та переконайтеся, що запис доданий.
2. Відкрийте Форму_61 у режимі Форми. Знайдіть запис, який ви вводили у п. 1. Внесіть зміни в поле Учнів. Збережіть Форму_61. Переконайтеся, що зміни внесені. Видаліть із форми останній уведений запис. Збережіть форму. Переконайтеся, що запис видалений.
3. Відкрийте Форму_62 у режимі Форми. Додайте новий запис: Прізвище — Рожко Р. П., Клас — 10, Улюблений предмет — інформатика, Інформатика — 11, Історія — 10. Збережіть форму. Відкрийте таблицю УЧНІ та переконайтеся, що запис доданий.
4. Відкрийте Форму_62 у режимі Форми. Знайдіть запис, який ви вводили у п. 3. Внесіть зміни в поле Прізвище. Збережіть Форму_62. Переконайтеся, що зміни внесені. Видаліть із форми останній уведений запис. Збережіть форму. Переконайтеся, що запис видалений.
5. Відкрийте Форму_63 у режимі Форми. Додайте новий запис: Клас — 7, Класний керівник — Пугач О. В., Прізвище — Косів К. О., Адреса — Лісна, 9, Дата народження — 05.04.2002. Збережіть форму. Відкрийте таблиці КЛАСИ й УЧНІ та переконайтеся, що запис у відповідні поля додано.
6. Відкрийте Форму_63 у режимі Форми. Знайдіть запис, який ви вводили у п. 5. Внесіть зміни в поля Класний керівник і Прізвище. Збережіть Форму_63. Переконайтеся, що зміни в таблиці КЛАСИ й УЧНІ внесені. Видаліть із форми останній уведений запис. Збережіть форму. Переконайтеся, що запис видалений.

Практична робота № 5

Тема: **СТВОРЕННЯ ФОРМ**

Завдання: розробити найпростішу форму на основі таблиць і запитів, створених у попередніх роботах.

Обладнання: комп'ютер зі встановленою СУБД Access 2016 і базою даних, створеною в попередній роботі.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Створіть форму з іменем Форма_71 для таблиці **КНИЖКИ** за допомогою засобу Форма. За допомогою кнопок на панелі навігації перегляньте всі записи таблиці.
2. Створіть форму з іменем Форма_72 для таблиці **ЧИТАЧІ** за допомогою засобу Розділена форма. Проаналізуйте переваги й вади засобів Форма та Розділена форма.
3. Створіть форму з іменем Форма_73 для Запит151 за допомогою засобу Кілька форм (Запит151 було створено в Практичній роботі № 3. За допомогою цього запиту з таблиці **ЧИТАЧІ** вибиралися дані про учнів 10 класу, які користувалися бібліотекою. Результуючий набір записів містив поля Прізвище, Клас і Інвентарний номер).
4. Створіть форму з іменем Форма_74 для таблиці **КНИЖКИ** за допомогою Майстра форм. У форму включіть поля Назва, Автор, Ціна. Використайте властивості форми й елементів керування для надання формі гарного вигляду.
5. Створіть форму з іменем Форма_75 для Запит151 за допомогою Майстра форм. Проаналізуйте переваги Майстра форм над засобом Кілька форм.
6. Створіть форму з іменем Форма_76 для таблиць **КНИЖКИ** й **ЧИТАЧІ** за допомогою Конструктора форм. У форму включіть поля Назва, Автор, Ціна, Прізвище, Клас.
7. Надайте формі заголовок **Наша бібліотека**. Перед першими трьома полями введіть назву Дані про книжки, а перед останніми двома — Дані про читачів. Додайте до форми поле Ціна.
8. Для різних елементів форми самостійно виберіть назву й тип шрифту, колір фону та колір окремих елементів.

Зробіть висновок про результати роботи.

ПОНЯТТЯ ЗВІТУ. АВТОМАТИЧНЕ СТВОРЕННЯ ЗВІТУ



Звіт — це об'єкт бази даних, призначений для вибирання із БД необхідних даних і виведення їх на екран або принтер у зручному для користувача вигляді.

Звіт дає змогу отримати дані у вигляді документа, із яким працює користувач певної професії. Наприклад, для бухгалтера дані можуть бути оформлені у вигляді одного документа, а для працівника відділу кадрів — іншого. Фактично звіт створює можливість отримати документ у такій формі, з якою працює користувач певної професії на паперових бланках.

Звіт дає змогу подати дані практично в будь-якому форматі. Дані можуть бути згруповані за певними ознаками, відфільтровані й упорядковані. До звіту можна додавати малюнки, діаграми, таблиці статистичних даних. У звіті можна обчислювати відповідні значення, наприклад, за професією, за стажем роботи тощо. Дані у звіті можуть подаватися в текстовому, числовому, табличному або графічному вигляді.

Access 2016 створює можливість налаштувати звіт, надати йому гарного естетичного вигляду. Зокрема можна налаштовувати фон і колір елементів звіту, змінювати шрифт, уставляти у звіт рисунки, використовувати діалогову графіку, вставляти об'єкти інших застосунків.

Джерелом даних для звіту є таблиці або підсумковий набір записів, отриманих унаслідок виконання запиту. Підсумковий набір записів на основі кількох таблиць сприймається у звіті як одна таблиця.

У процесі створення звіту необхідно чітко визначити структуру й зовнішній вигляд документа, який необхідно отримати. Доцільно на папері розробити кілька варіантів звіту та обрати найраціональніший.

Для кожної бази даних може бути створено кілька звітів різної форми з різними даними. Наприклад, один звіт може бути створено для директора школи, а другий — для вчителя. Засоби Access 2016 для створення звітів зображені на рис. 1.

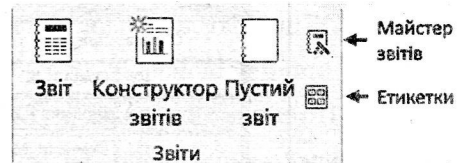


Рис. 1. Засоби створення звіту

- ✓ Основними засобами створення звітів є Звіт, Конструктор звітів і Майстер звітів, із яких найпростішим є Звіт.

За допомогою засобу Звіт створюється звіт для однієї таблиці. Якщо потрібно за допомогою цього засобу створити звіт для кількох таблиць,

то необхідно створити для цих таблиць відповідний запит. Потім на основі нового запиту створюється звіт. Джерелом даних засобу Майстер звітів може бути як запит, так і кілька таблиць. Найпотужнішим є засіб Конструктор звітів. Порожній звіт застосовується для створення порожнього бланка звіту, який потім можна в режимі Конструктор наповнити необхідними даними. Засіб Етикетки призначений для створення етикеток.

Звіт може бути виведений в одному з чотирьох режимів: Конструктора, Подання звіту, Режимі розмічування, Попереднього перегляду. Назви цих режимів збігаються з назвами команд підменю кнопки Вигляд. Порядок створення звіту багато в чому схожий з порядком створення форм. Властивості звіту мають ті самі значення, що й властивості форми, тому вони тут не описуються.

Розглянемо порядок створення звіту за допомогою засобу Звіт. Згадаємо, що джерелом даних для цього засобу може бути одна таблиця або запит, а сам звіт створюється фактично автоматично, простим натисканням однієї кнопки. До звіту, створеного за допомогою цього засобу, включаються всі поля таблиці або підсумковий набір записів, отриманих унаслідок виконання запиту. Такий звіт можна потім доповнити деякими елементами й налагодити в режимі Конструктор. Його також можна відправити електронною поштою та роздрукувати.

Приклад

Розглянемо порядок створення звіту на прикладі таблиці КАДРИ.

1. Відкриваємо базу даних atb і відмічаємо (або відкриваємо) таблицю КАДРИ.
2. Виконуємо команду Створити → Звіт. У результаті на екрані з'явиться звіт у режимі Подання звіту.
3. Зберігаємо звіт. Для цього на панелі швидкого доступу натискаємо кнопку Зберегти, у вікні, що відкриється, уведемо ім'я Звіт_1 і натискаємо кнопку ОК. Ім'я звіту з'явиться в області переходів. Зміст створеного звіту в режимі Розмітки зображено на рис. 2.

КАДРИ					26 вересня 2017 р.
					10:08:44
Справа	Прізвище	Посада	Рік народження	Освіта	
105	Сокил Т. Л.	касир	1960	середня	
109	Шрамко Т. Л.	диспетчер	1961	середня	
111	Семко М. М.	диспетчер	1970	середня	
115	Горошко Ф. Р.	диспетчер	1975	середня спеціальна	
116	Раков Г. П.	аналітик	1965	вища	
120	Рябко Р. П.	експерт	1981	вища	
132	Таран В. Д.	диспетчер	1973	вища	

Рис. 2. Звіт у режимі Розмітки

Створений звіт можна переглянути і в інших режимах. Згадаємо, що для цього в розділі Подання необхідно відкрити меню кнопки Вигляд і обрати необхідний режим. Редагування звіту найкраще робити в режимі Конструктор. Зокрема в цьому режимі можна змінити форматування звіту, змінити розмір полів, додати нові елементи керування з необхідними властивостями тощо.

Порядок створення звіту на основі запиту фактично нічим не відрізняється від порядку його створення на основі таблиці. На рис. 3 наведено Звіт_2 у режимі Розмітки, створений на основі Запит_1.

Справа	Прізвище	Рік народження	Стаж	Номер магазину
105	Сокил Т. Л.	1960	27	6
115	Горошко Ф.Р.	1975	17	31
116	Раков Г. П.	1965	19	21
109	Шрамко Т. Л.	1961	24	6
4				

Рис. 3. Звіт, створений на основі Запит_1

? ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. Для чого призначено звіт?
2. Що є джерелом даних для звіту?
3. За допомогою якого засобу звіт створюється автоматично?
4. Назвіть основні засоби створення звіту.
5. У яких режимах можна відкрити звіт?
6. Як можна перейти від одного режиму перегляду звіту до іншого?
7. Поясніть порядок створення звіту за допомогою засобу Звіт.

✓ ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1. За допомогою засобу Звіт створіть Звіт_51 на основі таблиці КЛАСИ.
2. За допомогою засобу Звіт створіть Звіт_52 на основі таблиці УЧНІ.
3. За допомогою засобу Звіт створіть Звіт_53 на основі таблиці ПЕРША1.
4. За допомогою засобу Звіт створіть звіт на основі запиту Запит41 (див. стор. 50). Згадаємо, що Запит41 було створено для таблиці УЧНІ. Результуючий набір записів містив поля Прізвище, Клас, Улюблений предмет.
5. За допомогою Майстра звітів створіть Звіт_55 на основі таблиці КЛАСИ.
6. За допомогою Майстра звітів створіть Звіт_56 на основі таблиці ПЕРШАЗ.

5.2

СТВОРЕННЯ ЗВІТУ ЗА ДОПОМОГОЮ КОНСТРУКТОРА ЗВІТІВ

Розглянемо загальний порядок створення звіту за допомогою Конструктора звітів.

1. Вибираємо імена таблиць (або запити), на основі яких буде створюватися звіт. Створюємо порожній звіт і вводимо текст його заголовка.
2. Вибираємо необхідні поля з таблиць (або запити) і групуємо записи за призначенням певного поля. За необхідності вставляємо ознаки сортування записів у групах.
3. Розміщуємо у звіті додаткові елементи керування. Переглядаємо звіт, надаємо йому необхідного вигляду й зберігаємо зі вказаним іменем.

Розглянемо порядок створення звіту на прикладі.

Приклад

Створити звіт на основі таблиць МАГАЗИНИ та КАДРИ. Из таблиці МАГАЗИНИ у звіт включити поля Номер магазину та Директор, а з таблиці КАДРИ — поля Прізвище, Освіта та Стаж. Згрупувати записи за значенням поля Посада й подати їх у порядку зменшення стажу. Звіт оформити приблизно так, як зображено на рис. 1.

Звіт_3				
Працівники магазинів				
Номер магазину	Директор	Прізвище	Освіта	Стаж
Посада працівників аналітик				
21	Коцюба П. М.	Раков Г. П.	вища	19
Посада працівників диспетчер				
31	Борзов А. С.	Таран В. Д.	вища	15
21	Коцюба П. М.	Семко М. М.	середня	16
31	Борзов А. С.	Горошко Ф. Р.	середня спеціальна	17
6	Середа К. М.	Шрамко Т. Л.	середня	24
Посада працівників експерт				
21	Коцюба П. М.	Рябко Р. П.	вища	8
Посада працівників касир				
6	Середа К. М.	Сокіл Т. Л.	середня	27

Рис. 1. Зразок оформлення звіту

1. Відкриваємо базу даних atb, виділяємо таблиці МАГАЗИНИ та КАДРИ й виконуємо команду Створити → Конструктор звітів. На екрані з'явиться порожній бланк звіту, структура якого за замовчуванням зображена на рис. 2.

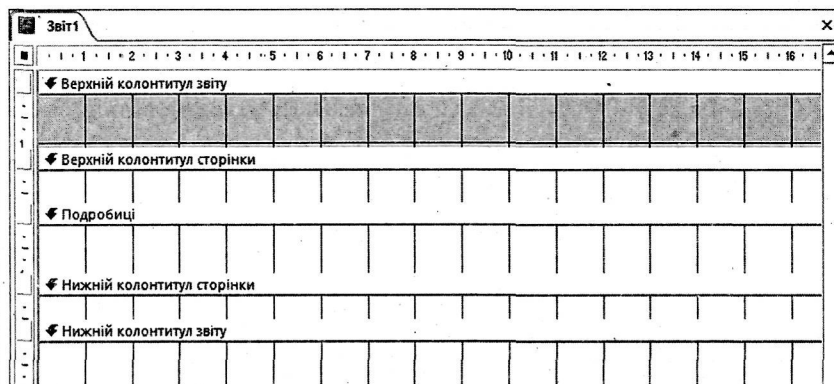


Рис. 2. Порожній бланк звіту

Якщо бланк звіту на вашому комп'ютері має іншу структуру, відкрийте аркуш властивостей звіту й установіть необхідні властивості. Можна також відкрити контекстне меню звіту й виконати в ньому необхідні команди. Наприклад, якщо відсутній розділ Верхній колонтитул звіту, то в контекстному меню звіту слід виконати команду Заголовок/примітка звіту.

2. Уводимо заголовок звіту. Для цього активуємо вкладку Конструктор, відкриваємо меню кнопки Елементи керування та натискаємо кнопку Підпис. Установлюємо курсор в область верхнього колонтитула звіту, натискаємо кнопку миші й уводимо заголовок звіту, наприклад, Працівники магазинів, і натискаємо кнопку Enter.
3. Відкриваємо розділ Список полів (якщо він не відкритий у цей момент). Для цього на вкладці Конструктор у групі Знаряддя виконуємо команду Додавання наявних полів. У розділі списку полів натискаємо кнопку Відобразити таблиці (якщо вони відсутні в списку). Відкриваємо список полів таблиць МАГАЗИНИ й КАДРИ, для чого перед іменем цих таблиць установлюємо перемикач у положення (-).
4. Перетягуємо в розділ Подробиці з таблиці МАГАЗИНИ поля Номер магазину та Директор, а з таблиці КАДРИ — поля Прізвище, Посада, Освіта та Стаж. Звернемо увагу, що в процесі перенесення полів з'являється

група з двох зв'язаних полів: ліворуч — назва поля, праворуч — поле для відображення даних (рис. 3).



Рис. 3. Приклад зв'язаних полів

Ці поля в області Подробиці можна за допомогою маркерів переміщувати як завгодно. Можна також стандартними засобами змінювати їх розміри. Поле відображення даних можна видалити з області Подробиці тільки разом із полем назви (лівим), а поле назви можна видалити окремо. Для видалення поля слід установити на ньому курсор миші й натиснути кнопку Delete. Але для поставленого завдання поля назв необхідні, тому видаляти їх не потрібно.

5. Переміщуємо з розділу Подробиці в розділ Верхній колонтитул сторінки всі поля назв (ліві поля), окрім поля Посада. Для цього можна скористатися буфером обміну: відкриваємо контекстне меню поля, яке слід перемістити, виконуємо команду Вирізати, потім відкриваємо контекстне меню Верхнього колонтитула сторінки й виконуємо команду Вставити.
6. Згрупуємо записи, які будуть включені до звіту, за посадою, тобто за значенням поля Посада. Для цього на вкладці Конструктор натискаємо кнопку Групування та сортування, яка міститься в розділі Групування та підсумки. У результаті на екрані в нижній частині звіту з'явиться розділ Групування, сортування й підсумок, у якому виконуємо команду Додати групу. Висвітяться всі імена полів, що перенесені в розділ Подробиці. Обираємо поле Посада, за яким виконується групування записів. В області звіту з'явиться новий розділ Групувати за Посада. Установлюємо сортування згрупованих записів за значенням поля Стаж, для чого спочатку натискаємо кнопку Додати сортування, а потім — поле Стаж.

Виконані налаштування відобразяться так, як зображено на рис. 4. Закриваємо розділ Групування, сортування й підсумок.

7. Переміщуємо обидва поля Посада в розділ Верхній колонтитул Посада. У поле назви Посада вводимо текст Посада працівників.

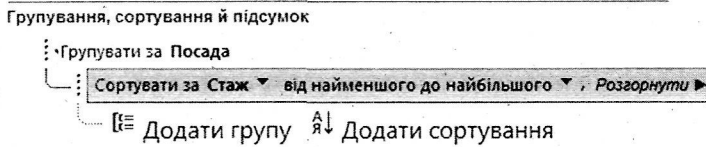


Рис. 4. Групування за полем Посада й упорядкування за полем Стаж

- У розділах Верхній колонтитул сторінки й Подробиці вирівнюємо в одну лінію всі поля орієнтовно так, як зображено на рис. 1 на с. 86. Остаточний зміст звіту, створеного в режимі Конструктор, який відповідає завданню, зображено на рис. 5.

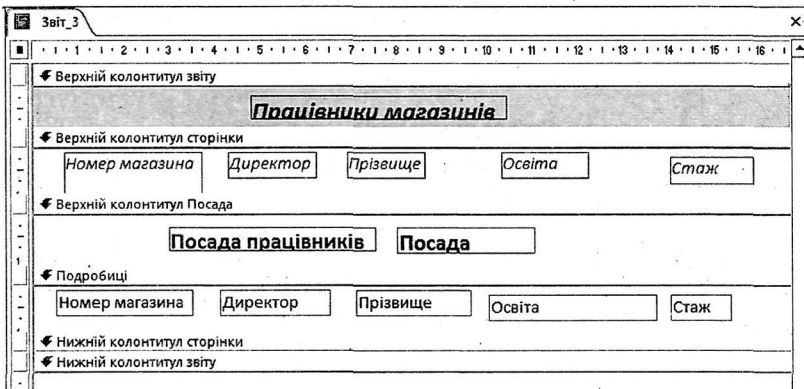


Рис. 5. Приклад оформлення звіту в режимі Конструктора звітів

- Зберігаємо Звіт_3 і відкриваємо його в режимі Подання звіту. Переконаємося, що його зміст відповідає змісту, зображеному на рис. 5.

До звіту можна додати необхідні елементи керування, розмістити їх у певному порядку, виділити окремі елементи кольором, товщиною ліній, змінити гарнітуру й розмір шрифту тощо. Усі ці операції принципово не відрізняються від аналогічних операцій, які виконувалися в процесі розробки форм.

Розглянемо такий варіант налаштування звіту. Відкриваємо звіт у режимі Розмічування. Виділимо будь-яке поле й на вкладці Форма відкриємо меню кнопки Форматування елементів керування. Потім відкриємо меню кнопки Заливка фігури. Відкриється вікно кольорів, у якому слід вибрати необхідний. Цим кольором буде виділено все поле.

Можна також підрахувати кількість записів у кожній групі. Для цього на вкладці Конструктор у групі Групування та підсумки відкриємо меню кнопки Підсумки й виконаємо команду Кількість значень. У результаті під кожним записом з'являться відповідні числа.

Приклад оформлення Звіту_3 зображено на рис. 6.

Працівники магазинів				
Номер магазину	Директор	Прізвище	Освіта	Стаж
Посада працівників <input type="text" value="аналітик"/>				
<input type="text" value="21"/>	<input type="text" value="Коцюба П. М."/>	<input type="text" value="Раков Г. П."/>	<input type="text" value="вища"/>	<input type="text" value="19"/>
		<input type="text" value="1"/>		
Посада працівників <input type="text" value="диспетчер"/>				
<input type="text" value="31"/>	<input type="text" value="Борзов А. С."/>	<input type="text" value="Таран В. Д."/>	<input type="text" value="вища"/>	<input type="text" value="15"/>
<input type="text" value="21"/>	<input type="text" value="Коцюба П. М."/>	<input type="text" value="Семко М. М."/>	<input type="text" value="середня"/>	<input type="text" value="16"/>
<input type="text" value="31"/>	<input type="text" value="Борзов А. С."/>	<input type="text" value="Горошко Ф. Р."/>	<input type="text" value="середня спеціальна"/>	<input type="text" value="17"/>
<input type="text" value="6"/>	<input type="text" value="Середа К. М."/>	<input type="text" value="Шрамко Т. Л."/>	<input type="text" value="середня"/>	<input type="text" value="24"/>
		<input type="text" value="4"/>		
Посада працівників <input type="text" value="експерт"/>				
<input type="text" value="21"/>	<input type="text" value="Коцюба П. М."/>	<input type="text" value="Рябко Р. П."/>	<input type="text" value="вища"/>	<input type="text" value="8"/>
		<input type="text" value="1"/>		
Посада працівників <input type="text" value="касир"/>				
<input type="text" value="6"/>	<input type="text" value="Середа К. М."/>	<input type="text" value="Сокіл Т. Л."/>	<input type="text" value="середня"/>	<input type="text" value="27"/>
		<input type="text" value="1"/>		
		<input type="text" value="7"/>		

Рис. 6. Приклад оформлення Звіту_3

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. ▲ Що розміщується у верхньому колонтитулі звіту?
2. ▲ Як вводиться заголовок звіту?
3. ▲ Поясніть порядок відкриття списку полів.
4. ✦ Сформулюйте загальний порядок створення звіту за допомогою Конструктора звіту.
5. ✦ Поясніть порядок групування звіту.
6. ★ Як переміщуються поля назв із розділу Подробиці у верхній колонтитул сторінки?
7. ★ Як можна підрахувати кількість записів у групах?

✓

ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1. ✦ За допомогою Конструктора звітів створіть Звіт_61 на основі таблиці КЛАСИ. Проаналізуйте звіт у різних режимах.
2. ✦ За допомогою Конструктора звітів створіть Звіт_62 на основі таблиці УЧНІ. Проаналізуйте звіт у різних режимах.



3. ✦ За допомогою Конструктора звітів створіть Звіт_63 на основі таблиці ПЕРША1. Проаналізуйте звіт у різних режимах.

4. ✦ За допомогою Конструктора звітів створіть Звіт_64 на основі запиту Запит41 (див. стор. 50). Згадаємо, що Запит41 створювався для таблиці УЧНІ, результуючі записи якого містили поля Прізвище, Клас, Улюблений предмет.



5. ★ За допомогою Майстра звітів створіть Звіт_65 на основі таблиць КЛАСИ й УЧНІ. Звіт має містити поля Класний керівник, Прізвище, Адреса, Інформатика. Проаналізуйте звіт у різних режимах.

6. ★ За допомогою Конструктора звітів створіть Звіт_66 на основі таблиць КЛАСИ й УЧНІ. Звіт має містити поля Учні, Класний керівник, Прізвище, Дата народження. Групування записів виконайте за значенням поля Улюблений предмет.

Практична робота № 6

Тема: СТВОРЕННЯ ЗВІТІВ

Завдання: розробити найпростіший звіт на основі таблиць і запитів, створених у попередніх роботах.

Обладнання: комп'ютер зі встановленою СУБД Access 2016 і базою даних, створеною в попередній роботі.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Створіть Звіт_71 для таблиці КНИЖКИ за допомогою засобу Звіт. Перегляньте та проаналізуйте звіт у різних режимах.
2. Створіть Звіт_72 для таблиці ЧИТАЧІ за допомогою засобу Звіт. Проаналізуйте звіт у різних режимах.
3. Створіть Звіт_73 для Запит151 за допомогою засобу Звіт (Запит151 було створено в Практичній роботі № 3. За його допомогою вибиралися дані про учнів 10 класу з таблиці ЧИТАЧІ. Запит містив поля Прізвище, Клас, Інвентарний номер).
4. Створіть Звіт_74 на основі таблиці ЧИТАЧІ за допомогою Майстра звітів. Тип звіту — книжковий. Згрупуйте записи звіту за значенням поля Клас.
5. Створіть Звіт_75 на основі Запит151 за допомогою Майстра звітів. Проаналізуйте звіт і переконайтеся в правильному результаті.
6. За допомогою Конструктора звітів створіть Звіт_76 для таблиць КНИЖКИ й ЧИТАЧІ. У звіт включіть поля Назва, Автор, Рік видання, Прізвище, Клас. Згрупуйте записи за значенням поля Автор. Уведіть назву звіту. Установіть необхідні властивості звіту й елементів керування.

Зробіть висновок про результати роботи.

ПРИЗНАЧЕННЯ, ОСНОВНІ ПОНЯТТЯ ТА ТЕРМІНИ МОВИ SQL

Сучасні системи управління реляційними базами даних, у тому числі Access 2016, містять потужні візуальні засоби, зрозумілий та зручний графічний інтерфейс, що забезпечує ефективну роботу з БД.

Разом із тим, існують спеціальні мови для створення й супроводу БД. Однією з них є мова SQL (Structured Query Language — структурована мова запитів), яка фактично стала стандартом мови реляційних БД. Вона постійно розвивається й удосконалюється, змінюються й міжнародні стандарти цієї мови.

Міжнародним стандартом передбачено, що повний склад мови SQL складається з таких окремих частин:

- DDL — мова визначення даних. Команди цієї групи використовуються для створення та зміни структури об'єктів бази даних, наприклад, таблиць;
- DML — мова маніпулювання даними. Команди використовуються для маніпулювання даними, які містяться в об'єктах, наприклад, для зміни або видалення даних у таблицях;
- DCL — мова керування даними. Команди цієї групи призначені для керування доступом до даних;
- DQL — мова формування запитів.

Окрім того, часто як складові SQL виділяють мову адміністрування БД і мову керування транзакціями, а також мову SQL/XML, що дає змогу інтегрувати дві популярні технології роботи з даними.

Мова SQL, що містить усі перелічені складові, створює можливість виконувати повний комплекс робіт із БД. Хоча для багатьох практичних задач достатньо й візуальних засобів СУБД, інколи для професійної роботи з базами даних без мови SQL обійтися просто неможливо. Тому у візуальних засобах СУБД реалізовані й окремі частини мови SQL. У системі Access 2016 реалізована частина SQL, що забезпечує формування запитів і роботу з ними.

Далі стисло описуються основні відомості саме про мову запитів SQL.

Як і інші мови програмування, вона має власний синтаксис.

У синтаксичних конструкціях використовуються такі поняття, як ключові слова, оператори, інструкції, речення й ін.

- ✓ Найчастіше в мові SQL використовуються такі оператори:
 - SELECT — визначаються поля, із яких необхідно вибрати дані;
 - FROM — визначається таблиця, поля якої вказані в реченні SELECT. Ключові слова SELECT і FROM завжди використовуються разом;
 - WHERE — визначається умова відбору полів, за якою вибираються дані;
 - ORDER BY — визначається порядок сортування отриманих результатів;
 - GROUP BY — визначається порядок групування записів.



Інструкція — це логічно завершена конструкція, яка може інтерпретуватися самостійно. Вона складається з речень і закінчується крапкою з комою.

Інструкції можуть містити коментарі, які не впливають на їх виконання. Найчастіше застосовуються однорядкові коментарі, які починаються двома символами --.

Конструкції у квадратних дужках є обов'язковими, а у фігурних дужках — обов'язковими.



Речення — це частина інструкції, що обов'язково містить ключове слово, яке й визначає його назву. Наприклад, SELECT Прізвище називають реченням SELECT, а WHERE Посада='вчитель' — реченням WHERE.

Далі наведено структуру найуживаніших речень мови SQL:

```
SELECT <список полів>           -- речення SELECT
FROM <ім'я таблиці>           -- речення FROM
WHERE <ім'я поля>=<умова>;    -- речення WHERE
```

Інструкція може складатися з кількох речень і завершується комою з крапкою. Наприклад, за допомогою інструкції:

```
SELECT Прізвище, Адреса, Телефон
FROM Школа
WHERE Посада='вчитель';
```

із таблиці Школа вибираються всі записи, у полі Посада яких є значення вчитель. Результуючий набір записів містить поля Прізвище, Адреса, Телефон.

Багато термінів, понять, синтаксичних правил і властивостей мови SQL збігаються з відповідними назвами класичних мов програмування. Усі конструкції мови (ключові слова, оператори тощо) однаково сприймаються великими та малими літерами. За стандартом SQL для імен об'єктів (таблиць, полів та інших) використовується латинський

алфавіт, але в багатьох випадках допускається використання й національного алфавіту.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. ▲ Яка частина мови SQL реалізована в Access 2016?
2. ▲ Які оператори містить найпростіша інструкція мовою SQL?
3. ★ Из яких частин складається мова SQL?
4. ★ Для чого призначена мова SQL?
5. ★ Поясніть загальний формат найпростішої інструкції мовою SQL.
6. ★ Наведіть приклад найпростішої інструкції мовою SQL.

6.2

НАЙПРОСТІШІ ЗАПИТИ МОВОЮ SQL У СИСТЕМІ ACCESS 2016

Розглянемо порядок створення найпростішого запиту на основі однієї таблиці КАДРИ за допомогою мови SQL.

Для цього в системі Access 2016 виконуємо команду Створити → → Макет запиту. Закриваємо вікно Відображення таблиці і на вкладці Конструктор у групі Результати натискаємо кнопку SQL. Відкриється вікно Запит1, у робочому полі якого висвітлиться оператор SELECT. Він обов'язково використовується з оператором FROM і має таку мінімальну загальну структуру:

```
SELECT <список імен полів>  
FROM <ім'я таблиці>;  
Наприклад, інструкція:  
SELECT *  
FROM <ім'я таблиці>;
```

забезпечує виведення всіх полів таблиці. А після введення інструкції:

```
SELECT *  
FROM КАДРИ;
```

і натискання кнопки Запуск на екран будуть виведені записи всіх полів таблиці КАДРИ.

Після виконання інструкції:

```
SELECT Справа, Прізвище, Освіта, Оклад  
FROM КАДРИ;
```

отримаємо результат, зображений на рис. 1.

Як бачимо, виводяться записи з полями, що вказані в операторі SELECT. Зазначимо, що порядок розміщення полів у фразі SELECT може бути довільним і не збігатися з порядком їх розміщення в початковій таблиці. Поля виводяться в порядку їх розміщення в реченні SELECT.

Інколи доцільно після оператора SELECT указувати не лише ім'я поля, а й ім'я таблиці, якій воно належить. У такому випадку використовується така форма: <ім'я таблиці.ім'я поля>, тобто ім'я поля відокремлюється від імені таблиці крапкою.

Відомо, що записи в таблиці не повторюються. Проте коли виводяться не всі поля таблиці, зміст деяких записів може збігатися.

Наприклад, після виконання інструкції:

```
SELECT Посада
FROM КАДРИ;
```

будуть виведені назви посад з усіх записів таблиці.

Справа	Прізвище	Освіта	Стаж
105	Сокіл Т. Л.	середня	27
109	Шрамко Т. Л.	середня	24
120	Рябко Р. П.	вища	8
111	Семко М. М.	середня	16
116	Раков Г. П.	вища	19
132	Таран В. Д.	вища	15
115	Горошко Ф.Р.	середня спеціальна	17
*	0		0

Рис. 1. Записи таблиці КАДРИ з окремими полями

Для виведення значень полів, що не повторюються, у реченні SELECT використовується ключове слово DISTINCT із такою структурою:

```
SELECT DISTINCT <список імен полів>
FROM <ім'я таблиці>;
```

Наприклад, після виконання інструкції:

```
SELECT DISTINCT Посада
FROM КАДРИ;
```

отримаємо результат, зображений на рис. 2.

Ключове слово ALL у реченні SELECT, навпаки, дає змогу отримати записи, що повторюються. Воно встановлюється за замовчуванням.

Посада
аналітик
диспетчер
експерт
касир

Рис. 2. Назви посад у таблиці КАДРИ

- ✓ У реченні SELECT можуть бути не лише імена наявних у таблиці полів, а й вирази. Кожен такий вираз є новим полем у запиті.

Вирази можуть містити арифметичні операції, деякі математичні функції (LOG — натуральний алгоритм, EXP — експонента й інші), а як змінні використовуються наявні імена полів. Наприклад, для таблиці КАДРИ створимо запит, за допомогою якого виводяться прізвища працівників, стаж, оклад і нове поле, значення якого обчислюється за виразом: $\text{Оклад} * (\text{Стаж} - 5) / 100$. Для надання імені цього поля після виразу вказується ключове слово AS і назва поля.

Запит може мати такий вміст:

```
SELECT Прізвище, Стаж, Оклад, (Оклад*(Стаж-5)/100) AS Доплата
FROM КАДРИ;
```

У результаті виконання цієї інструкції отримаємо результат, зображений на рис. 3.

Прізвище	Стаж	Оклад	Доплата
Сокіл Т. Л.	27	3500	770
Шрамко Т. Л.	24	4000	760
Рябко Р. П.	8	4200	126
Семко М. М.	16	4000	440
Раков Г. П.	19	4500	630
Таран В. Д.	15	4000	400
Горошко Ф.Р.	17	4000	480

Рис. 3. Записи таблиці КАДРИ з полем, що обчислюється

Принципова відмінність у роботі з базами даних, що містять кілька таблиць, від БД з однією таблицею полягає в організації та врахуванні зв'язків між таблицями. Наприклад, за допомогою інструкції:

```
SELECT *
FROM УЧНІ, КЛАСИ;
```

кожен запис таблиці УЧНІ буде з'єднаний із кожним записом таблиці КЛАСИ, до того ж будуть включені всі поля таблиць обох таблиць.

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. Який оператор обов'язково використовується з оператором SELECT?
2. За допомогою якої інструкції можна вивести всі поля таблиці?
3. Яке призначення має ключове слово DISTINCT?
4. Для чого застосовується ключове слово ALL у реченні SELECT?
5. Наведіть приклад інструкції SELECT для виведення окремих полів таблиці.
6. Поясніть порядок створення полів із виразами.

6.3

ЗАПИТИ З УМОВОЮ. ГРУПУВАННЯ ЗАПИТІВ

За допомогою запитів з умовою вибираються ті записи, значення яких відповідають певному критерію. Для цього в інструкції SELECT після оператора FROM указується речення WHERE з необхідною умовою.

Умова набирає значення TRUE або FALSE. Відбираються ті записи, які відповідають умові, значення виразу в яких має TRUE.

Приклад

Вибрати з таблиці КАДРИ працівників, стаж роботи яких понад 16 років, із полями Справа, Прізвище, Рік народження, Стаж, Номер магазина.

1. Для реалізації цього завдання створимо такий запит:

```
SELECT Справа, Прізвище,
Стаж
FROM КАДРИ
WHERE Стаж>15;
```

2. У результаті виконання запиту отримаємо результат, зображений на рис. 1.

Справа	Прізвище	Стаж
105	Сокіл Т. Л.	27
115	Горошко Ф.Р.	17
116	Раков Г. П.	19
109	Шрамко Т. Л.	24

Рис. 1. Записи таблиці КАДРИ, отримані за умовою

Для групування записів використовується оператор GROUP BY. Він розміщується в інструкції SELECT після оператора WHERE (якщо він застосовується).

Використовується така загальна структура інструкції:

```
SELECT <список полів>
FROM <ім'я таблиці>
[WHERE <умова>] GROUP BY <список полів>;
```

У список полів речення GROUP BY можуть включатися тільки ті поля, які є в реченні SELECT.

```
SELECT Посада, COUNT (*) AS 'Кількість посад'
FROM КАДРИ
GROUP BY Посада
```

У результаті виконання цього запиту отримаємо результат, зображений на рис. 2.

Як бачимо, підраховується кількість різних посад, що містяться в таблиці КАДРИ.

```
SELECT Посада, SUM (Оклад) AS 'Сума окладів'
FROM КАДРИ
GROUP BY Посада;
```

У результаті виконання цього запиту отримаємо результат, зображений на рис. 3.

Посада	'Кількість посад'
аналітик	1
диспетчер	4
експерт	1
касир	1

Рис. 2. Кількість посад у таблиці КАДРИ

Посада	'Сума окладів'
аналітик	4500
диспетчер	16000
експерт	4200
касир	3500

Рис. 3. Сума окладів працівників різних професій

У цьому прикладі обчислюється сума окладів працівників для кожної назви посади.

- ✓ Групувати рядки можна також із використанням оператора умови.

Зазначимо, що оператор WHERE використовується для записів початкової таблиці. Його не можна використовувати для згрупованих записів. Для вибору записів серед згрупованих використовується оператор HAVING, за яким у групі виконуються ті самі дії, як і в операторі WHERE для всієї початкової таблиці.

Таким чином, оператор HAVING може використовуватися лише за наявності оператора GROUP BY (за винятком окремих випадків, які трапляються доволі рідко).

В інструкції SELECT оператори WHERE, GROUP BY і HAVING виконуються в такому порядку.

1. Із початкової таблиці насамперед відбираються ті записи, які відповідають умові в операторі WHERE.
2. Відібрані записи за допомогою оператора GROUP BY групуються у групи.
3. Наприкінці за допомогою оператора HAVING відбираються записи, які відповідають зазначеній умові.

```
SELECT Посада
FROM КАДРИ
GROUP BY Посада
HAVING SUM (Оклад)>4200
```

У результаті виконання запиту отримаємо результат, зображений на рис. 4.

У цьому прикладі обчислюється сума окладів для кожної групи посад і виводяться назви тих посад, сума окладів яких більше 4200.

```
SELECT Посада, COUNT (Посада) AS (Кількість)
FROM КАДРИ
WHERE Стаж>15
GROUP BY Посада
HAVING Посада IN ('касир', 'диспетчер', 'експерт');
```

Результат виконання запиту зображений на рис. 5.

Посада
аналітик
диспетчер

Рис. 4. Назви посад, сума окладів яких вище 4200

Посада	Кількість
диспетчер	3
касир	1

Рис. 5. Кількість посад у таблиці КАДРИ працівників, стаж яких більше 15 років

За допомогою оператора IN із таблиці вибираються всі ті записи, у яких значення вказаного поля збігається хоча б із одним значенням

у списку значень оператора IN. Якщо значення поля збігається хоча б із одним значенням у списку оператора IN, виробляється значення TRUE.

У цьому прикладі підраховується кількість працівників професії «касир», «диспетчер» і «експерт», стаж роботи яких більше 15 років. Звернемо увагу, що назва посади «експерт» не виведена через те, що стаж роботи працівника цієї професії не відповідає умові.



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1. Який оператор використовується в запитах з умовою?
2. Які поля таблиці можуть застосовуватися в операторі GROUP BY?
3. Який оператор використовується для групування записів?
4. Поясніть призначення оператора HAVING.
5. Наведіть приклад інструкції з оператором WHERE.
6. Наведіть приклад інструкції з оператором GROUP BY.



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

1. Складіть інструкцію вибірки із таблиці УЧНІ записів із полями Прізвище, Адреса, Інформатика.
2. Складіть інструкцію вибірки з таблиці УЧНІ записів із полями Прізвище, Зріст і Середній бал, який обчислюється так: (Інформатика+Історія)/2.
3. Складіть інструкцію вибірки з таблиці УЧНІ записів із полями Прізвище, Клас, Адреса, у полі Історія яких є значення більше 9.
4. Складіть інструкцію для таблиці УЧНІ, за допомогою якої обчислюється кількість різних балів у полі Інформатика.
5. Складіть інструкцію обчислення кількості учнів у кожному класі в таблиці УЧНІ.

СУТНІСТЬ ІМПОРТУ ТА ЕКСПОРТУ ОБ'ЄКТІВ

На практиці часто доводиться одночасно працювати не з одним файлом бази даних, а з кількома, у тому числі розміщеними в Інтернеті. У таких випадках виникає потреба в обміні даними (об'єктами) між базами даних, створених за допомогою однієї СУБД, наприклад, Access 2016, а також різних СУБД, наприклад, Access 2016 і Paradox. Окрім того, інколи виникає потреба в обміні даними між Access 2016 та іншими Windows-програмами, наприклад, Word, Excel, Outlook, HTML і XML-документами.

Найпростішим способом обміну об'єктами між Windows-програмами є їх копіювання і вставлення за допомогою буфера обміну. Наприклад, у документ, створений у текстовому редакторі Word, можна вставити рисунок, створений у графічному редакторі Paint.

У системі Access 2016 для обміну об'єктами є спеціальні операції, які отримали назву імпорту й експорту. Крім того, для отримання даних з інших зовнішніх джерел у системі Access 2016 підтримується технологія зв'язування, яка тут не розглядається.

Отже, для отримання об'єктів у базу даних із зовнішніх джерел у системі Access 2016 використовується технологія імпорту, а для їх передавання в інші застосунки — технологія експорту. Імпорт і експорт можуть відбуватися між різними базами даних, у тому числі між базами даних різних форматів (наприклад, між Access 2016 і Paradox, FoxPro), а також між документами табличних і текстових процесорів, Outlook, документами форматів HTML і XML.

У процесі імпорту об'єкти перетворюються у формат Access 2016 і розміщуються в новому об'єкті, а в об'єкті-джерелі залишаються без змін. Наприклад, нова таблиця й початкова існують незалежно одна від одної.

Технології імпорту з різних систем (Word, Excel та ін.) відрізняються одна від одної.

Загальний порядок імпортування однаковий для всіх систем і складається з кількох етапів.

1. Відкривається БД-приймач та обирається система, із якої буде здійснюватися імпорт (наприклад, Access, Excel).
2. Обираються об'єкти, які необхідно імпортувати, і налаштовуються параметри їх імпортування (у разі необхідності).
3. Виконується імпорт.

За своєю суттю експорт є операцією, зворотною до імпорту. У процесі експорту здійснюється перенесення даних із об'єктів системи Access 2016 в іншу базу даних цієї самої системи, а також у зовнішні файли різних форматів. Під час експорту дані перетворюються в новий формат, а об'єкти-джерела залишаються незмінними.

Оскільки загальні принципи імпортування й експортування всіх типів об'єктів бази даних ідентичні, далі розглядається лише технологія та правила імпортування й експортування об'єктів із однієї бази даних Access 2016 в іншу БД Access 2016.



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАТЬ

1. Який існує найпростіший спосіб обміну даними між Windows-програмами?
2. Між якими програмами можна здійснювати обмін даними з БД Access 2016?
3. У чому полягає сутність імпортування даних у БД Access 2016?
4. Чим принципово відрізняється експорт від імпорту даних?
5. Поясніть загальний порядок імпортування даних у БД Access 2016.

7.2

ІМПОРТ ОБ'ЄКТІВ З ОДНІЄЇ БАЗИ ДАНИХ В ІНШУ

Імпортувати можна будь-які об'єкти бази даних. Одночасно можна імпортувати як кілька об'єктів одного типу (наприклад, кілька запитів), так і кілька об'єктів різного типу (наприклад, таблиць і запитів).

Розглянемо технологію імпорту на прикладі імпорту таблиці КАДРИ й запиту Запит_1 із бази даних atb у базу даних skola. Отже, джерелом є база atb, а приймачем — база skola.

1. Відкриваємо базу даних — приймач skola. Активуємо вкладку Зовнішні дані. Панель інструментів набуде вигляду, зображеного на рис. 1.



Рис. 1. Панель інструментів вкладки Зовнішні дані

- У групі Імпорт і зв'язування відкриваємо меню кнопки Нове джерело даних, у якому встановлюємо курсор на назві Із бази даних. Відкриється перелік джерел, зображений на рис. 2.

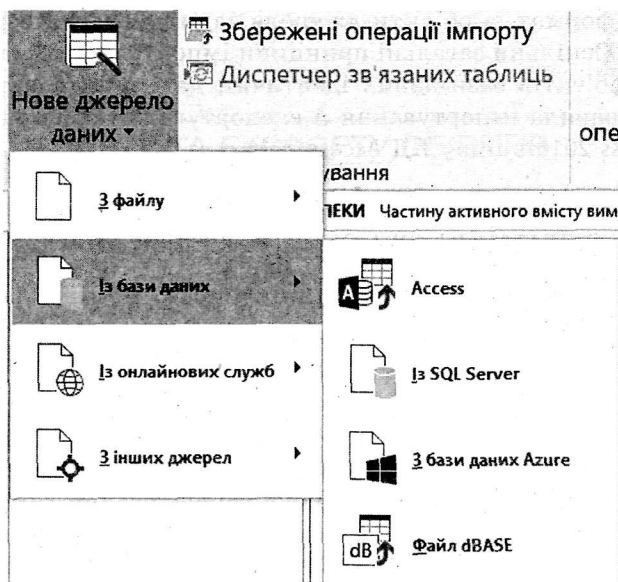


Рис. 2. Перелік джерел даних для імпортування

- Оскільки виконується імпорт об'єктів із однієї БД Access 2016 в іншу БД Access 2016, натискаємо кнопку Access. Відкриється перше вікно Майстра імпорту, зображене на рис. 3.

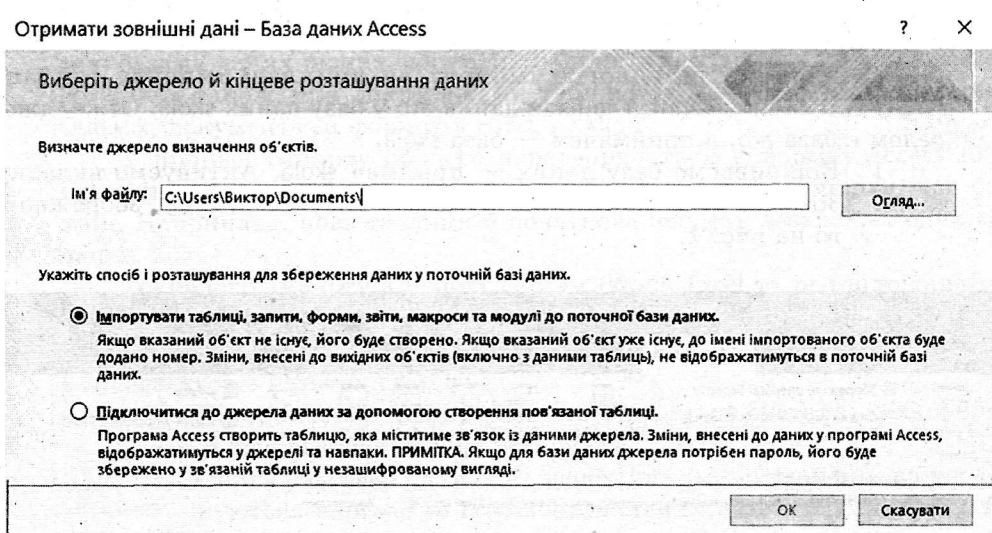


Рис. 3. Перше вікно Майстра імпорту

У поле Ім'я файлу цього вікна можна ввести повне ім'я файлу бази-джерела. Можна також скористатися кнопкою Огляд... і вибрати файл atb.

Скористаємося кнопкою Огляд... і знайдемо ім'я atb, вмикаємо перемикач Імпортувати таблиці, запити... і натискаємо кнопку ОК. Відкриється вікно Імпортувати об'єкти, зображене на рис. 4.

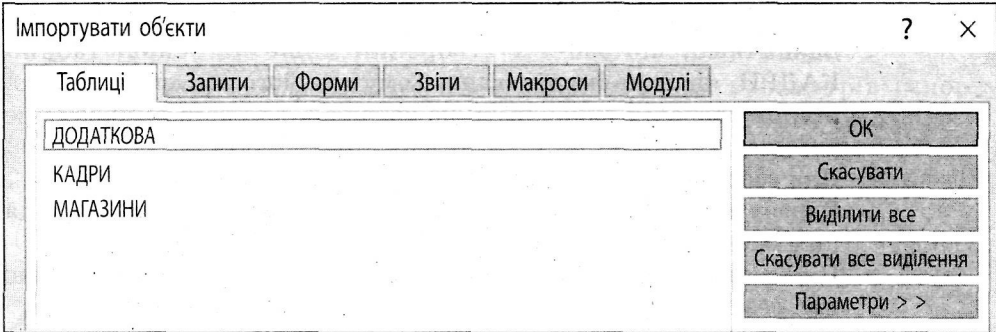


Рис. 4. Вікно для імпортування об'єктів

У цьому вікні розміщені вкладки об'єктів бази-джерела atb. На вкладці Таблиці обираємо таблицю КАДРИ, для чого встановлюємо курсор на її імені й натискаємо кнопку миші. Потім на вкладці Запити обираємо Запит_1.

- Можна налаштувати деякі параметри імпортування обраних об'єктів. Для цього натиснемо кнопку Параметри>>.

У результаті у вікні відобразиться область налаштування параметрів імпорту, зображена на рис. 5.

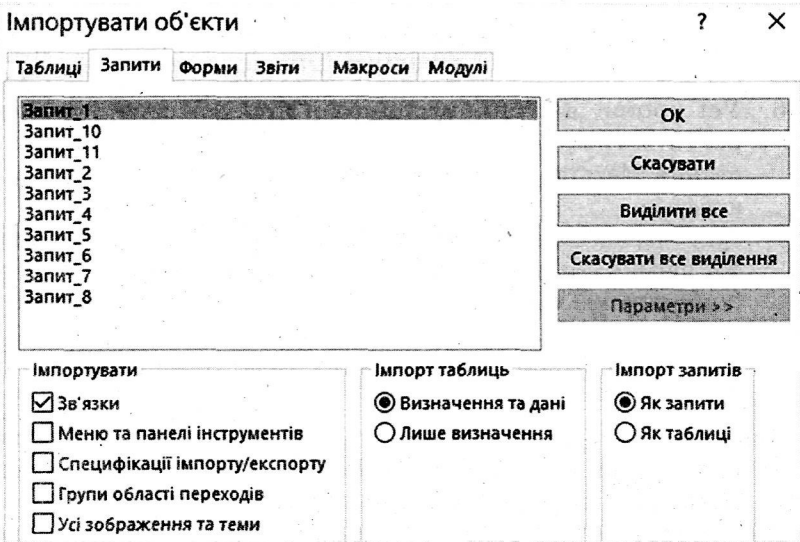


Рис. 5. Вікно з областю налаштування параметрів імпорту

Розглянемо призначення параметрів групи Імпорт таблиць.

Перемикач Визначення та дані встановлюється в тому випадку, коли з бази-джерела необхідно імпортувати і структури, і дані всіх тих таблиць, які обрані користувачем у допоміжному вікні.

Якщо обрати перемикач Лише визначення, то будуть імпортуватися лише структури таблиць, дані не імпортуються. Вмикаємо перший перемикач.

Зазначимо, що Запит_1 створений лише на основі таблиці КАДРИ, яка вже імпортована. Тому для його запуску в БД skola ніяких операцій виконувати не потрібно, для чого вмикаємо перемикач Як запити.

- Для збереження всіх налаштувань натискаємо кнопку ОК. Одразу починається імпортування обраних об'єктів у базу даних skola. Після завершення імпорту на екран буде видано повідомлення про результат імпорту, зображене на рис. 6.

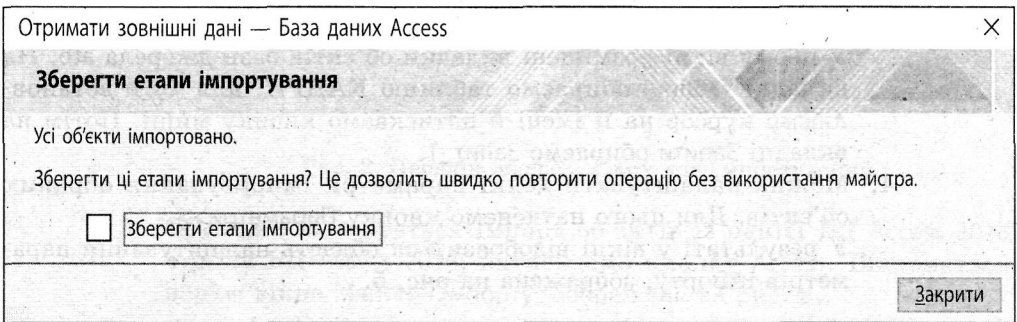


Рис. 6. Вікно для збереження етапів імпортування

- Усі кроки, які виконувалися в процесі імпорту, можна зберегти з тим, щоб за необхідності повторити їх без використання Майстра імпорту (за замовчуванням ці кроки не зберігаються).

Для збереження виконаних кроків необхідно у вікні, що відкрито, увімкнути прапорець Зберегти етапи імпортування, але тут виконувати цю операцію немає потреби. Тому просто натискаємо кнопку Закрити.

Якщо тепер відкрити базу даних skola, то в ній побачимо таблицю КАДРИ і Запит_1. Їх вміст такий самий, як у БД atb.

Для імпортування електронної таблиці із книги Excel спочатку відкривається БД-приймач, потім на вкладці Зовнішні дані натискається кнопка Excel. Далі виконуються команди Майстра імпорту.

Для імпортування текстового файлу його попередньо необхідно структурувати так, щоб кожний рядок файлу був записом, а кожен запис

ділився на окремі поля. Для розмежування полів часто застосовується кома з крапкою. Приклад структурованого текстового файла:

Микола; 050-400-22-33; 21 січня

Олена; 063-333-11-55; 5 травня

?

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1. ▲ Які об'єкти БД можна імпортувати в іншу БД?
2. ▲ На якій вкладці Access 2016 здійснюється імпортування?
3. ★ Із якою метою здійснюється налаштування параметрів імпортування?
4. ★ Поясніть загальний порядок імпортування об'єктів із однієї БД в іншу.

7.3

ЕКСПОРТ ОБ'ЄКТІВ З ОДНІЄЇ БАЗИ ДАНИХ В ІНШУ

Експортувати з однієї бази даних Access 2016 в іншу БД Access 2016 можна всі об'єкти: таблиці, запити, форми, звіти. У процесі експортування в БД, у яку здійснюється експорт, створюється копія відповідних об'єктів. Правила експортування мало відрізняються від правил імпорту. Однак, на відміну від імпортування, експортувати можна лише об'єкти одного типу.

Порядок і правила експортування розглянемо на прикладі експорту таблиці УЧНІ бази даних skola в базу даних atb. Отже, БД skola є джерелом.

1. Відкриваємо БД-джерело (skola) та обираємо таблицю УЧНІ натисканням клавіші миші на її імені в області переходів. На вкладці Зовнішні дані в групі Експорт натискаємо кнопку Access. Відкриється вікно, зображене на рис. 1.

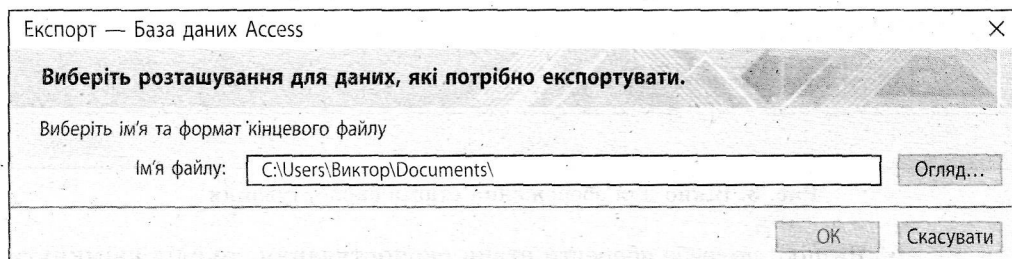


Рис. 1. Вікно для відкриття бази даних-приймача

2. Натискаємо кнопку Огляд... і у вікні, що відкриється, знаходимо файл atb.accdb, вибираємо його й натискаємо кнопку Зберегти.

Після цього у вікні (рис. 1) натискаємо кнопку ОК. Відкриється вікно Експорт, зображене на рис. 2.

- У текстове поле цього вікна вводимо нову назву експортованої таблиці, наприклад, МОЇ УЧНІ (нову назву можна і не вводити, однак, щоб далі не було плутанини, краще її ввести). Для експортування структури таблиці і її даних використовуємо перемикач Визначення та дані (якщо потрібно експортувати лише структуру таблиці, необхідно увімкнути перемикач Лише визначення).

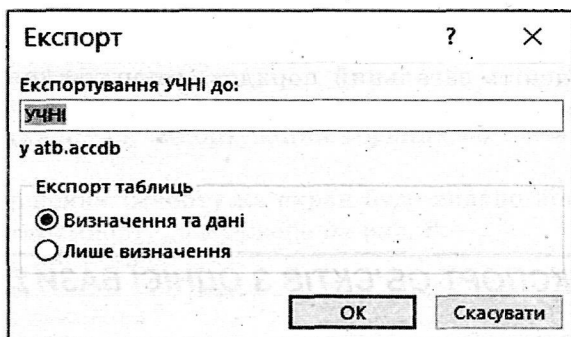


Рис. 2. Вікно для встановлення режиму експортування

- Для початку експортування натискаємо кнопку ОК. Якщо під час експорту виникають деякі проблеми, відповідні повідомлення виводяться на екран. Наприклад, якщо виявиться, що в базі-приймачі вже є об'єкт із таким самим іменем, буде виведена рекомендація ввести нове ім'я або змінити об'єкт. Після успішного експортування відкриється вікно, зображене на рис. 3.

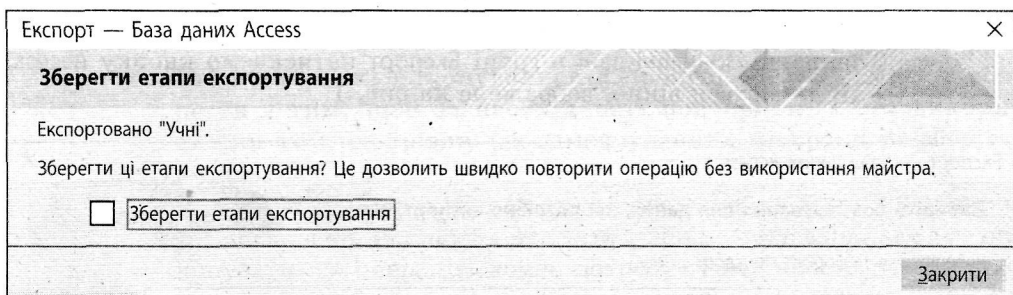


Рис. 3. Вікно для збереження етапів експортування

- ✓ Якщо є потреба зберегти етапи експортування, то слід увімкнути прапорець Зберегти етапи експортування, інакше просто натискається кнопка Закрити.

Ім'я таблиці МОЇ УЧНІ з'явиться в області переходів бази даних atb. Якщо відкрити цю таблицю, то можна переконатися, що її вміст повністю збігається зі вмістом таблиці УЧНІ бази даних skola.



ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

1. Скільки типів об'єктів можна одночасно експортувати в Access 2016?
2. Поясніть призначення перемикача Лише визначення.
3. Поясніть порядок експортування таблиць із однієї БД в іншу.



ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

Для виконання наведених завдань створіть на жорсткому диску дві бази даних: persha і druga.

1. Імпортуйте в БД persha форму з іменем Форма_1 із БД atb. Перевірте правильність імпортування.
2. Експортуйте із бази даних atb Запит_2 в БД druga. Переконайтеся, що експортування виконане правильно.
3. Імпортуйте в базу даних persha Звіт_1 із БД atb. Перевірте імпортований звіт у БД persha.
4. Експортуйте з бази даних atb таблицю ДОДАТКОВА в БД druga. Переконайтеся в правильності виконаної операції.
5. Імпортуйте в базу даних druga всі об'єкти БД persha. Перевірте правильність імпортування.
6. Експортуйте з бази даних druga Запит_2 і таблицю ДРУГА у БД persha. Переконайтеся в правильному виконання операції.

Атрибут — це властивість сутності.

База даних — це сховище даних різного типу про об'єкти та взаємозв'язки між ними певної предметної області, що зберігаються, оновлюються та опрацьовуються за встановленими правилами.

Запит — це один із основних об'єктів БД Access 2016. Головним призначенням запиту є відбір із таблиць потрібних даних, їх опрацювання й подання користувачеві в зручній формі.

Запит з параметрами — це запит, у процесі виконання якого пропонується ввести деякі дані, наприклад умову, яку потрібно вставити в поле. Їх ще називають запитом із змінними критеріями.

Запит на вибірку даних — це запит, який забезпечує вибір необхідних даних з однієї або з кількох таблиць.

Запит на зміну — це запит, за допомогою якого модифікуються структури таблиць і змінюються в них дані.

Запит перехресний — це запит на вибірку даних із можливостями групування записів. Групування можна виконувати як за значеннями полів, так і за значеннями записів.

Звіт — це об'єкт бази даних, призначений для вибирання із БД необхідних даних і виведення їх на екран або принтер у зручному для користувача вигляді.

Інструкція — це логічно завершена конструкція, яка може інтерпретуватися самостійно. Вона складається з речень і закінчується крапкою з комою.

Ключ — це унікальне поле або набір кількох полів у таблиці, які однозначно ідентифікують записи.

Мова SQL (Structured Query Language — структурована мова запитів) — це спеціальна мова для створення й супроводу БД, яка фактично стала стандартом мови реляційних баз даних.

Предметна область — сфера застосування конкретної бази даних.

Речення — це частина інструкції, яка обов'язково містить ключове слово, яке й визначає його назву.

Система управління базами даних (СУБД) — це інструмент, призначений насамперед для створення структури бази даних, введення й оновлення даних, пошуку необхідних даних та їх опрацювання за певним алгоритмом.

Сутність — це деякий об'єкт реального світу. Вона має екземпляри, які відрізняються один від одного значеннями атрибутів.

Транзакція — це послідовність операцій над даними, яка сприймається СУБД як єдине ціле.

Фільтрація за виділенням — це відбір записів на основі значень поточно-го поля.

Форма — це об'єкт бази даних Access 2016, головне призначення якого — забезпечити користувачам зручне введення, редагування й видалення даних із бази даних.

АЛФАВІТНИЙ ПОКАЖЧИК

А

Атрибут 12

Б

База даних 4

— документальна 4

— фактографічна 5

Ж

Журналізація 6

З

Запит 44

— для створення нової таблиці 59

— з параметрами 55

— з полями, що обчислюються 53

— з функціями 51

— на вибірку даних 47

— на додавання 60

— перехресний 57

Звіт 83

І

Індексування таблиць 29

Інструкція 93

Інформаційна система 14

— автономна 14

— глобальна 15

— групова 15

— корпоративна 15

К

Ключ 8

— первинний 28

Конструктор форм 74

М

Мова SQL 92

О

Об'єкт предметної області 4

П

Предметна область 4

Р

Реляційна модель даних 8

Речення 93

С

Система управління
базами даних (СУБД) 5

Сутність 12

Т

Транзакція 6

Ф

Фільтрація 38

— за виділенням 38

— за формою 39

Форма 65, 78

ЛІТЕРАТУРА

1. Андон Ф., Резниченко В. Язык запросов SQL. Учебный курс. — СПб: Питер; К.: Изд. группа BHV, 2006.
2. Биков В. Ю., Руденко В. Д. Системы управления информацией базами даних в освіті. — К.: ІЗМН, 1966. — 288 с.
3. Петров В. Н. Информационные системы. — СПб: Питер, 2002. — 688 с.
4. Руденко В. Д., Речич Н. В., Потієнко В. О. Інформатика (рівень стандарту): Підручник для 10 (11) класу закладів загальної середньої освіти. — Харків: Вид-во «Ранок», 2018. — 160 с.
5. Руденко В. Д. Бази даних в інформаційних системах: Навч. посіб. для студ. пед. ун-тів / За ред. В. Ю. Бикова. — К.: Фенікс, 2010. — 240 с.
6. Хоменко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных. Учебник для высших учебных заведений / Под ред. проф. А. Д. Хоменко. — 2-е изд., доп. и перераб. — СПб: КОРОНА принт, 2002. — 672 с.

ЗМІСТ

Передмова	3
1. Загальні відомості про бази даних	
1.1. Поняття бази даних	4
1.2. Реляційна модель даних	7
1.3. Проектування баз даних	11
1.4. Бази даних в інформаційних системах	14
1.5. Основні відомості про СУБД Access 2016	17
2. Таблиці	
2.1. Створення й уведення структури таблиць	20
2.2. Модифікація структури таблиць	25
2.3. Ключові поля, індекси, зв'язування таблиць	28
2.4. Уведення, пошук і редагування даних у таблиці	33
2.5. Сортування та фільтрація записів. Операції над таблицями	37
Практична робота № 1	41
Практична робота № 2	43
3. Запити	
3.1. Загальні відомості про запити	44
3.2. Запити на вибірку даних	47
3.3. Запити з функціями і з полями, що обчислюються	51
3.4. Запити з параметрами. Перехресні запити	55
3.5. Запити на змінення	59
Практична робота № 3	63
Практична робота № 4	64
4. Форми	
4.1. Створення форм за допомогою простих засобів	65
4.2. Елементи керування та властивості форм	71
4.3. Створення форми за допомогою Конструктора форм	74
4.4. Використання форм для введення й редагування даних	78
Практична робота № 5	82

5. Звіти	
5.1. Поняття звіту. Автоматичне створення звіту	83
5.2. Створення звіту за допомогою Конструктора звітів	86
Практична робота № 6	91
6. Основи мови запитів SQL	
6.1. Призначення, основні поняття та терміни мови SQL	92
6.2. Найпростіші запити мовою SQL у системі Access 2016	94
6.3. Запити з умовою. Групування запитів	96
7. Імпорт і експорт об'єктів баз даних	
7.1. Сутність імпорту та експорту об'єктів	100
7.2. Імпорт об'єктів з однієї бази даних в іншу	101
7.3. Експорт об'єктів з однієї бази даних в іншу	105
Комп'ютерний словник	108
Алфавітний покажчик	109
Література	109

Навчальне видання
РУДЕНКО Віктор Дмитрович

Т901500У. Підписано до друку 27.08.2018.
Формат 70×100/16. Папір офсетний.
Гарнітура Шкільна. Друк офсетний.
Ум. друк. арк. 9,1.

**ІНФОРМАТИКА
БАЗИ ДАНИХ**
модуль для учнів 10–11 класів
рівень стандарту

ТОВ Видавництво «Ранок»,
вул. Кібальчича, 27, к. 135, Харків, 61071.
Свідоцтво суб'єкта видавничої справи
ДК № 5215 від 22.09.2016.
Для листів: вул. Космічна, 21а, Харків, 61145.

Редактор *Л. А. Каюда*
Технічний редактор *С. В. Яшиш*

E-mail: office@ranok.com.ua
Тел. (057) 719-48-65,
тел./факс (057) 719-58-67.

Регіональні представництва
видавництва «Ранок»:

Київ – тел. (044) 229-84-01,
e-mail: office.kyiv@ranok.com.ua,
Львів – тел. (067) 269-00-61,
e-mail: office.lviv@ranok.com.ua.

З питань придбання продукції
видавництва «Ранок» звертатися за тел.:

Житомирі – (067) 122-63-60;

у Харкові – (057) 727-70-80;

Львові – (032) 244-14-36;

Києві – (044) 599-14-53, 377-73-23;

Миколаєві та Одесі – (067) 551-10-79;

Вінниці – (0432) 55-61-10;

Черкасах – (0472) 51-22-51;

Дніпрі – (056) 785-01-74, 789-06-24;

Чернігові – (0462) 93-14-30.

E-mail: commerce@ranok.com.ua.

«Книга поштою»: вул. Котельниківська, 5, Харків, 61051.

Тел. (057) 727-70-90, (067) 546-53-73.

E-mail: pochta@ranok.com.ua

www.ranok.com.ua

Надруковано у друкарні ТОВ «ТРИАДА-ПАК»

м. Харків, пров. Сімферопольський, 6. Тел. +38(057)703-12-21

www.triada-pack.com, email: sale@triada.kharkov.ua

ISO 9001:2015 № UA228351, FAMO TRIADA LLC (065445)

Папір, на якому надрукована ця книга,



безпечний для здоров'я
та повністю
переробляється



з оптимальною білізною,
рекомендованою
офтальмологами



виблювався
без застосування
хлору

Разом дбаємо про екологію та здоров'я

ВИДАВНИЦТВО
РАНОК

Бази даних

Навчальний посібник містить:

- основні теоретичні відомості про бази даних
- опис структури та порядок роботи в середовищі Access
- алгоритми і приклади створення різних об'єктів за допомогою СУБД Access
- запитання для самоперевірки знань
- завдання для самостійного виконання
- практичні роботи з покроковим описом виконання

Посібники на підтримку вибіркових модулів:

- Бази даних. Модуль для учнів 10–11 класів. Рівень стандарту (автор В. Д. Руденко)
- Основи алгоритмізації та програмування мовою Python (автори В. Д. Руденко, О. О. Жугастров)
- Вивчаємо Java у школі: Навчальний посібник у 2 частинах (автори В. Д. Руденко, О. О. Жугастров)
- Перші кроки програмування мовою Java (автори В. Д. Руденко, О. О. Жугастров)



ISBN 978-617-09-4776-5



9 786170 947765

ВИДАВНИЦТВО
РАНОКНАВЧАЛЬНО-МЕТОДИЧНА ЛІТЕРАТУРА
УСІ КНИГИ ТУТ!

- ranok.com.ua
- e-ranok.com.ua
- pochta@ranok.com.ua
- (057) 727-70-90