Ніжинський державний університет
імені Миколи Гоголя

Фетісов В. С.

Робота із СУБД *Access*

Навчально-методичний посібник

2-ге видання, перероблене і доповнене

Ніжин – 2011

УДК 004.65
ББК 73
Ф 45

Рекомендовано Вченою радою
Ніжинського державного університету імені Миколи Гоголя
Протокол № 1від 06.10. 2011

Рецензенти:
к.ф.-м.н., доц. ***Лісова Т. В.***,
доц. ***Іванов В. В.***

Фетісов В. С.
Ф 45. Робота із СУБД *Access*: [навчально-методичний посібник] / Фетісов В. С. 2-ге вид., перероб. і доп. – Ніжин: НДУ ім. М. Гоголя, 2011. – с.

Посібник містить основні теоретичні відомості про бази даних, опис практичної роботи із СУБД *Access* 2007, два комплекси взаємопов'язаних лабораторних завдань тощо.

Розраховано на студентів спеціальностей "Соціальна інформатика","Педагогіка і методика середньої освіти. Математика та основи економіки", "Педагогіка і методика середньої освіти. Математика. Фізика", "Педагогіка і методика середньої освіти. Математика та інформатика". Разом із тим посібник буде корисний усім, хто прагне навчитися працювати з базою даних.

Погодження

Позначення функціональних клавіш	**<Shift>**
Комбінація функціональних клавіш. Застосовується, коли потрібно *одночасно* натиснути дві клавіші. Для цього слід натиснути одну із клавіш і, *утримуючи її натиснутою*, натиснути другу.	**<Alt>+<F4>**
Позначення кнопок у вікнах і формах.	**«ОК»**
Позначення вікон, вкладок	"*Содержание*"
Виконання команди в меню. Такий запис містить послідовність команд з меню, які відокремлюються символом ▸, і означає, що слід натиснути на першій із команд цієї послідовності, а потім рухати мишею від команди до команди, натиснувши мишею на останній команді.	**Пуск ▸ Программы ▸ Стандартные**
Найменування програми	*Access*

Загальне поняття про бази даних

Бази даних дозволяють замінити традиційні форми збереження інформації у вигляді картотек, бухгалтерських рахунків тощо на зручний вигляд. При цьому вся інформація зберігається у вигляді взаємопов'язаних файлів. Її можна змінювати, реорганізовувати, здійснювати пошук, упорядковувати за різними критеріями тощо. Як правило, бази даних містять відомості про конкретні об'єкти реального світу у деякій предметній галузі або розділі предметної галузі. Наприклад, існують бази даних у медицині, бібліографії, хімії і т.д. Прикладами баз даних можуть бути енциклопедія, картотека особистих справ робітників, складська картотека матеріальних цінностей тощо.

Призначення баз даних подвійне:

1. У зручному вигляді опрацьовувати великі обсяги даних.
2. Отримувати з цих даних потрібну інформацію.

База даних – це особливим чином організована сукупність взаємопов'язаних, збережених разом даних. Зберігання даних у базах даних дає змогу централізувати управління ними, вирішити проблеми дотримання стандартів, безпеки, цілісності, уникнути дублювання та надмірності даних. Дані у базах даних створюються і зберігаються як єдине ціле в інтересах вирішення всіх завдань даної предметної галузі, але кожна програма, що працює з нею, використовує тільки ті дані, що їй потрібні.

Залежно від характеру зв'язків між елементами бази даних розрізняють її три основні моделі:

1. *Ієрархічна модель* – це така організація даних, за якою дані одного рівня підпорядковуються даним, що знаходяться на вищому рівні.

2. *Мереживна модель* є моделлю, в якій будь-який елемент може бути пов'язаним із довільною кількістю інших.

3. *Реляційна модель*.

Реляційна модель даних

Концепція реляційної моделі даних розроблена Едгаром Коддом у 1970 р. Назва її походить від англійського слова "*relation*", що означає "відношення". В основі такої моделі є подання даних у табличному вигляді і математичне поняття відношення, згідно з яким між файлами бази даних можна встановлювати зв'язки. Це дає можливість одночасно коригувати інформацію взаємопов'язаних файлів, за рахунок чого значно зменшується час на введення і модифікацію даних, і уникнути їх надмірності, тобто дублювання одних і тих самих даних у різних файлах. Такий підхід передбачає так звану *цілісність даних*, завдяки якій забезпечується підтримка зв'язків у взаємопов'язаних файлах і захист інформації від випадкових змін взаємопов'язаних даних. Взагалі, реляційна модель передбачає наявність у складі бази даних більше одного файлу, і ефективність роботи з такою моделлю полягає саме у тому, що база даних містить кілька файлів, які поєднуються між собою зв'язками.

Останнім часом у термінології, пов'язаній з базами даних, термін "*файл*" замінюється терміном "*таблиця*". Тому надалі ми будемо також використовувати саме цей термін, хоч на практиці ці поняття тотожні.

За основу в реляційних базах даних береться ***таблиця***, тобто двовимірний масив, що складається з рядків та стовпчиків. Рядок таблиці містить інформацію *про одну* одиницю деякої множини однотипних об'єктів, наприклад, відомості про співробітника, товар, студента і т. ін., і називається ***записом***. У свою чергу, кожний рядок (запис) містить різноманітні реквізити. Наприклад, для співробітника це буде ім'я, прізвище, дата народження, адреса, стать і т. ін. Для кожного з цих реквізитів відводиться окремий стовпчик таблиці, який називають ***полем***. Таким чином, кожне поле містить дані одного типу й

одного походження. Переставити поля в одному записі, не переміщуючи аналогічним чином поля в інших записах, не можна, бо тоді втрачається сенс інформації.

Таблиця 1

Приклад таблиці бази даних – телефонний довідник

	№ з/п	**Прізвище**	**Адреса**	**Телефон**
Запис 1	1	Коваленко А.П.	вул. Шевченка, 98, кв.21	3–24–56
Запис 2	2	Лунев В.П.	вул. Перемоги, 18	2–41–35
	Поле 1	*Поле 2*	*Поле 3*	*Поле 4*

Така структура – структура прямокутної таблиці – забезпечує швидкий пошук даних, зручне використання математичного апарату, гнучкість і простоту подання даних, простоту їх введення і модифікації.

Можна поставити питання: чи можна базу даних замінити текстовими або табличними файлами, створеними текстовими або табличними процесорами, адже там також можна складати таблиці? Але у табличних і текстових процесорах не можна створювати зв'язки між файлами, і взяти інформацію одночасно з декількох файлів неможливо.

Таким чином, таблиця, запис, поле є первинними елементами організації бази даних.

У теорії баз даних використовується також ще ряд термінів. Кількість стовпчиків у таблиці називається *порядком відношення*. Значення всіх стовпчиків, з'єднаних в одному рядку, називають кортежем, а значення усіх рядків, що знаходяться в одному стовпчику, – доменом.

Системи управління базами даних (СУБД)

СУБД – це програма, призначена для створення, ведення і використання бази даних.

Функції СУБД:

1. Створення таблиць.
2. Введення і редагування у таблицях даних одночасно з контролем їх коректності.
3. Створення вихідних форм (документів), що потрібні користувачу.
4. Виконання різних видів оброблення інформації: сортування, пошуку тощо.
5. Забезпечення цілісності бази даних (при машинних збоях, вимкненні струму і т. ін.).

Для реалізації таких функцій у кожній СУБД існує мова маніпуляції даними і *транслятори*, або *інтерпретатори*, для цих мов.

Існує досить багато різноманітних за функціональними можливостями СУБД – від простих однофайлових, орієнтованих на оброблення інформації відносно невеликого обсягу, до функціонально розвинених СУБД, призначених для розв'язання складних задач. Добре відомі такі СУБД, як *Oracle, SyBase, Informix, FoxPro, Paradox, Clipper, Access, Clarion* і т. ін.

Проектування бази даних

Перш ніж створювати базу даних, потрібно ретельно її продумати і спроектувати на папері. Слід пам'ятати, що швидкодія бази даних і зручність роботи з нею залежать саме від того, наскільки добре вона спроектована. Проектування включає:

1. Визначення переліку таблиць, що увійдуть до бази даних.
2. Визначення для кожної таблиці набору полів і їх властивостей.
3. Виявлення і визначення зв'язків між таблицями.

Нормалізація бази даних

Як зазначалося раніше, основна ідея реляційної моделі полягає у тому, щоб подати довільну структуру даних у вигляді простої двовимірної таблиці, або, як кажуть, *нормалізувати*

структуру. ***Нормалізація бази даних*** – це спеціальна процедура розподілу таблиць бази даних, у процесі якого одна таблиця розподіляється на кілька з метою, у першу чергу, виключити з таблиць інформацію, що повторюється, тобто *надлишкові дані*. Сам процес нормалізації хоча зовні і простий, але має певні тонкощі. Тому для нього розроблена спеціальна теорія, яка містить спеціальні методи поділу структури даних на кілька таблиць. Операція з поділу даних виконується в три етапи. Взагалі теорія нормалізації складається з п'яти етапів, але на практиці використовують ті, що розглядаються нижче.

1. ***Створення першої нормальної форми***. Таблиця у першій нормальній формі повинна задовольняти таким вимогам: не мати однакових записів і груп полів, що повторюються. Самі записи і поля таблиці не впорядковані. Відповідно до першої вимоги, таблиці повинні мати ключові поля. Прикладом полів, що повторюються, можуть бути, наприклад, оцінки студента. В принципі, кожен студент може мати стільки оцінок, на скількох семінарах він був. Якщо таких семінарів буде, наприклад, 60, то таблиця повинна містити 60 стовпчиків – по одному для кожного семінару. Але на практиці студент ніколи не матиме такої кількості оцінок, тому більша частина значень цих полів для окремого студента буде не заповненою. Таку інформацію, різну за обсягом для кожного запису (в даному разі – студента), називають *групами, що повторюються*.

2. ***Створення другої нормальної форми***. Таблиця відповідає всім вимогам у першій нормальній формі, і будь-яке ключове поле повинно однозначно визначатися певним значенням ключа. Слід зазначити, що всі таблиці з простим ключем є таблицями у другій нормальній формі.

3. ***Створення третьої нормальної форми***. Таблиця відповідає всім вимогам у першій і другій формі і передбачається, що ключове поле є незалежним від будь-яких інших полів.

Вилучення надмірності даних

Одним із найважливіших етапів нормалізації бази даних є вилучення надлишкової інформації з її таблиць. Ця проблема не завжди очевидна, але її вирішення є дуже важливим елементом під час проектування. Наприклад, таблиця, що містить оцінки студентів, може мати такий вигляд:

Таблиця 2

Ненормалізована таблиця "Успішність студентів"

Прізвище	Група	Дисципліна	Оцінка
Гладун О.М.	МЕ21	Інформатика	4
Гладун О.М.	МЕ21	Статистика	5
Романець Ю.Д.	МЕ21	Фізика	5

Легко побачити, що для кожного студента найменування групи буде повторюватися стільки разів, скільки він буде мати оцінок. Очевидним фактом є те, що студент належить тільки до однієї групи, тому таке подання даних і є типовим прикладом *надмірності даних*, оскільки цілком достатньо *одноразове* введення інформації про належність студента до певної групи. Вирішення проблеми полягає у створенні *двох* таблиць, одна з яких буде містити відомості про оцінки, а інша – відомості про групи, до яких належать студенти.

Таблиця 3

Нормалізована таблиця "Успішність студентів"

Прізвище	Дисципліна	Оцінка
Гладун О.М.	Інформатика	4
Гладун О.М.	Статистика	5
Романець Ю.Д.	Фізика	5

Таблиця 4

Таблиця "Список студентів"

Прізвище	Група
Романець Ю.Д.	МЕ21
Гладун О.М.	МЕ22

У цьому разі для кожного студента у таблиці буде тільки один запис з найменуванням групи, до якої він належить.

Роботу з базою даних розглянемо на прикладі СУБД *Microsoft Access*, яка входить до складу офісного пакету корпорації *Microsoft*.

СУБД *Access*-2007

Об'єкти бази даних

Таблиця. Основний об'єкт бази даних. У таблицях зберігаються всі дані, а самі таблиці зберігають і структуру бази. Як правило, окрема таблиця містить дані, поєднані за змістом. Наприклад, в одну таблицю можна звести дані з особових карток студентів. Дані у такій таблиці будуть відображати кадрові відомості про студентів. В іншу таблицю – відомості про оцінки.

Запити. Служать для відбору даних із таблиць на основі заданих користувачем критеріїв і відображення їх у зручному вигляді.

Форми. Засоби для зручного введення даних.

Звіти. За своїми властивостями і структурою схожі на форми, але призначені для виведення даних, у першу чергу, на друк.

Сторінки доступу до даних. Фізично це об'єкт, виконаний у коді HTML, що розміщується на Web-сторінці і передається клієнту разом із нею. Вони здійснюють інтерфейс між клієнтом, сервером і базою даних, що міститься на сервері.

Макроси і модулі. Призначені як для автоматизації операцій, що повторюються під час роботи із СУБД, так і для створення нових функцій шляхом програмування.

Програма надає кілька засобів створення кожного з основних

об’єктів бази даних:

1. Ручні (розроблення об’єктів у режимі “Конструктор”).
2. Автоматизовані (розроблення за допомогою програм-майстрів).
3. Автоматичні – засоби прискореного розроблення найпростіших об’єктів.

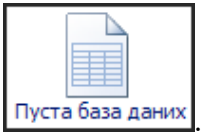
Ручні засоби найбільш трудомісткі, але забезпечують максимальну гнучкість; автоматизовані й автоматичні – найбільш продуктивні, але і найменш гнучкі. Для створення різних об’єктів доцільно користуватися різними засобами.

1. Під час розроблення таблиць, запитів і форм складної структури доцільно використовувати ручні засоби, тобто працювати в режимі “Конструктор”.

2. Під час розроблення простих форм, звітів і сторінок доступу краще використовувати автоматизовані засоби майстрів. Це пов’язано з тим, що для таких об’єктів велику роль відіграє зовнішній вигляд. Дизайн цих об’єктів дуже трудомісткий, тому його краще доручити програмі.

Створення бази даних

Першим кроком роботи з базою даних є створення самої бази даних. Для цього слід виконати такі дії.

1. У початковому вікні “*Початок роботи з Microsoft Office Access*” (“Getting Started with Microsoft Office *Access*”), що з’явиться після завантаження програми, натиснути кнопку 

. Ще один, щоправда, більш тривалий варіант створення – натиснути кнопку  “Office” і вибрати зі списку пункт “Створити” (New). Зауважимо, що ліва частина вікна містить шаблони баз даних, які користувач може використовувати як основу під час створення власної бази даних.

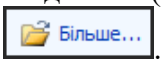
2. У правій частині головного вікна з’явиться поле “Ім’я

файлу” (“File name”), що містить назву файлу – за замовчуванням “База даних1.accdb” (“Database1.accdb”). Доцільно змінити це ім’я на змістовне.

3. Праворуч від поля з іменем знаходиться піктограма папки. Натиснувши на неї, можна змінити місце розташування бази даних і вибрати для неї потрібне місце на диску. Ім’я і місце розташування відображається під полем “Ім’я файлу”, що дає змогу здійснювати візуальний контроль розташування бази даних.

4. Натиснути кнопку **«Створити» («Create»)**.

Файл *Access 2007* створюється з розширенням .ACCDB.

Звернення до існуючої бази даних здійснюється засобами, загальноприйнятими для програм, що входять до складу MS-Office 2007. Наприклад, її можна вибрати зі списку існуючих баз даних на панелі “Відкриття останньої бази даних” (“Open Recent Database”) або натисканням кнопки 

. За другим варіантом з’являється вікно “*Відкриття файлу бази даних*” (“Open”), за допомогою якого здійснюється пошук потрібної бази даних. Такий саме варіант надає і натискання кнопки  “Office”, після чого з її меню вибирається пункт “Відкрити” (“Open”).

Закінчення роботи здійснюється натисканням на кнопці  або вибором з меню кнопки  “Office” пункту **«Вийти з *Access*»** (“Exit Access”).  При цьому система не видає запит на збереження даних, тому що дані у базах даних зберігаються автоматично під час редагування таблиць бази даних.

Інтерфейс програми

Після створення система здійснює перехід до режиму проектування бази даних і головне вікно набуває вигляду, що наведений на рис. 1.

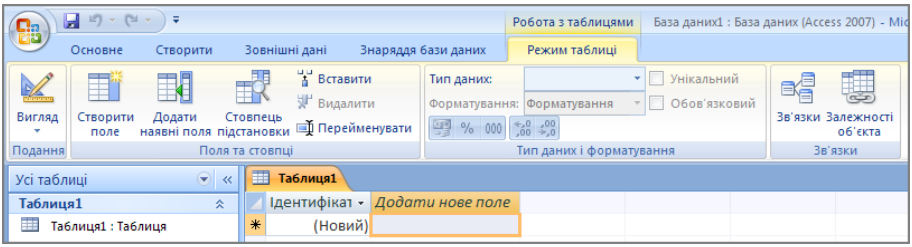


Рис. 1. Головне вікно *Access* 2007

Головне вікно складається з кількох елементів.

1. Стандартна для додатків Office 2007 користувальницька панель швидкого доступу. За замовчуванням вона має три дії: відмінити, повторити, зберегти. Але користувач має можливість настроїти її “під себе”. Для цього слід натиснути кнопку списку  праворуч від останнього елемента панелі і встановити позначку біля команд, що потрібно включити до панелі.

2. Стрічка вкладок, що прийшла на заміну головному меню попередніх версій MS-Office, на якій команди згруповані на кількох вкладках.

- Основне (Home). Робота (редагування) з основними об’єктами.
- Створити (Create). Створення таблиць, форм, звітів і запитів.
- Зовнішні дані (External Data). Наприклад, імпорт або експорт до електронної таблиці Excel.
- Знаряддя бази даних (Database Tools).

3. Вміст вкладки подається у вигляді *стрічки* команд, що містить певні команди (інструменти), згруповані за функціональним призначенням.

4. Лівий бік вікна займає *панель об’єктів* (вибору). Вона містить усі створені для бази даних основні об’єкти.  Об’єкти групуються за таблицями, для якої вони створені. Сама таблиця розташована в першому рядку списку, а всі інші об’єкти розташовані нижче. Уявлення щодо типу об’єкта надає його

піктограма, яка має вигляд  для запитів,  для форм,  для звітів. Натискання на піктограмі  , що знаходиться в рядку з іменем таблиці, приховує перелік об'єктів таблиці, після чого піктограма змінює свій вигляд на  . Зрозуміло, що натискання такої піктограми призводить до відображення переліку об'єктів таблиці. Зауважимо, що і саму панель об'єктів можна приховати, натиснувши на піктограму  . Панель об'єктів містить також ще одну групу елементів, так звані "Непов'язані об'єкти" ("Unrelated Objects"), на якій розташовуються, зокрема, макроси.

5. У центральній, робочій області вікна відбувається робота з вибраним з панелі об'єктів об'єктом.

6. У нижній правій частині головного вікна знаходиться невеличка панель, що забезпечує швидку зміну режиму роботи  .

З метою збільшення площини робочого екрану можна приховати стрічку команд. Для цього слід двічі натиснути на краю будь-якої вкладки.  *У результаті такої дії система переходить до режиму мінімізації стрічки. Візуально це відображається у тому, що біля елемента користувальницького списку "Згорнути стрічку" ("Minimize the Ribbon") встановлюється позначка. Для відновлення відображення слід натиснути на вкладці. Але це не призведе до виходу з режиму мінімізації стрічки. Для відмови від цього режиму слід натиснути кнопку користувальницького списку і зняти позначку з елемента списку "Згорнути стрічку".*

Властивості бази даних

На початку роботи із системою доцільно визначити деякі загальні властивості баз даних, що будуть за замовчуванням використовуватися при їх побудові. Для цього слід натиснути кнопку  "Office" і натиснути у нижній частині вікна-списку кнопку **«Параметри *Access*»** ("*Access* Options"). З'явиться вікно "*Параметри Access*".

1. Перейти на вкладку "*Найуживаніші*".
2. У полі "Порядок сортування нової бази даних" ("New database sort order") вибрати "Українська" (Ukrainian).
3. Доцільно також у полі "Папка баз даних за промовчанням" визначити місце (папку), до якого буде за замовчуванням звертатися програма для пошуку баз даних.

Робота з таблицями

Як зазначалося раніше, головним об'єктом баз даних є таблиці. Саме вони містять дані бази даних. Тому розроблення бази даних починається саме зі створення таблиць.

Будь-яка таблиця має свою *структуру*, яка визначається набором полів, кожне з яких характеризується *ім'ям*, *типом даних* і *набором властивостей*.

Типи даних

Дані у таблицях зберігаються у вигляді тексту, чисел, дат і т. ін. Такий вигляд пов'язаний із тим, якого роду інформацією є ці дані. Він визначається типом даних. До основних типів даних, що підтримує *Access*, належать такі:

1. **Текстовий, символьний** (**Text**) – довільна сукупність алфавітно-цифрових символів (тобто звичайний текст), не більше 255 символів. Приклади таких реквізитів такого типу: прізвище, адреса, паспортні дані.

2. **Примітка** (**Mem**o) – використовується для збереження великих обсягів тексту (до 65535 символів). Фізично текст не

зберігається у полі. Він зберігається в іншому місці бази даних, а поле містить покажчик на нього. Прикладом такого поля може бути автобіографія.

3. **Числовий** (**Number**) – число, над яким можна виконувати арифметичні операції у звичайній формі. Наприклад, оцінка.

4. **Дата й час** (**Date/Time**) – календарні дати і час. Наприклад, дата народження.

5. **Грошова одиниця** (**Currency**) – дані у вартісному вигляді. Для збереження таких даних можна скористатися і числовим типом, але грошові дані мають певні властивості, пов'язані, наприклад, із правилами округлення під час роботи з копійками. Прикладом таких даних може бути стипендія, посадовий оклад.

6. **Автонумерація** (**AutoNumber**) – спеціальний тип даних для унікальних натуральних чисел з автоматичним приростом на одиницю. Можна використовувати, наприклад, для порядкових номерів.

7. **Так/Ні** (**Логічний**, **Yes/No**) – логічні дані, що можуть мати тільки два значення: "Істинне" (TRUE) і "Хибне" (FALSE). Такий тип зручно використовувати для реквізитів, що мають два альтернативних значення (наприклад, стать).

8. **Об'єкт OLE** (**OLE Object**) – призначений для збереження, наприклад, мультимедійних об'єктів. Так само, як і поля "Примітка", у таблиці зберігається тільки посилання на об'єкт.

9. **Гіперпосилання** (**Hyperlink**) – URL адреса Web-об'єкта.

Створення таблиці

Як зазначалося раніше, відразу після створення нової бази даних система переходить до режиму проектування таблиці. У загальному випадку створення нової таблиці здійснюється так:

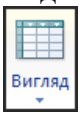
1. На стрічки вкладок вибрати пункт "Створити" ("Create").

2. На стрічці інструментів натиснути кнопку "Таблиця" ("Datasheet").

3. Перейти до режиму конструктора (проектування таблиці). Така дія ініціює появу невеличкого вікна "*Зберегти як*" ("Save

As”), у якому слід ввести ім’я таблиці. Після введення це ім’я з’являється у переліку таблиць у лівій частині вікна. Одночасно у правій частині вікна з’являється вкладка з назвою таблиці.  Можна здійснювати проектування таблиці і в режимі таблиці. Але за таким варіантом користувач позбавляється можливості одночасно зі створенням поля визначати для нього тип даних і властивості.

Отже, робота з таблицями може відбуватися у двох режимах: режимі конструктора і режимі таблиці (введення даних). Перехід від режиму до режиму здійснюється шляхом натискання на стрічці команд кнопки “Вигляд” (“View”), яка при цьому послідовно змінює вигляд: 

– робота в режимі конструктора та 

– робота в режимі таблиці.

Надалі звернення до потрібної таблиці здійснюється шляхом подвійного натискання на її назві на панелі об’єктів.

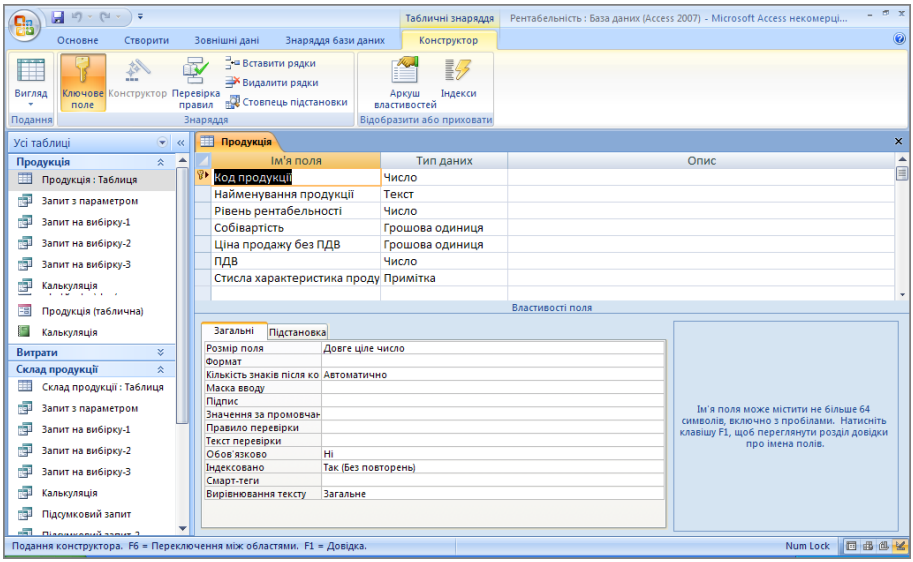


Рис. 2. Проектування таблиці в режимі “Конструктор”

Побудова структури таблиці здійснюється так:

1. Під час створення структури таблиці до неї автоматично включається поле з ім’ям “ID” з типом даних “автонумерація” (“лічильник”), завдяки чому буде відбуватися автоматична нумерація записів таблиці. За замовчуванням воно визначено ключовим і призначено для створення зв’язків між таблицями, про що мова піде пізніше. Користувачу слід прийняти рішення щодо необхідності цього поля. Як правило, таке поле або взагалі не потрібне, або повинно мати інший тип чи назву, тому його можна спокійно вилучити. Для вилучення поля зі структури слід викликати на рядку з його назвою контекстне меню і вибрати з нього пункт “Видалити рядки” (“Delete Column”).

2. Встановити курсор у полі “Ім’я поля” (“Field Name”) і ввести ім’я поля, яке повинно бути унікальним у структурі кожної таблиці.

3. У полі “Тип даних” (“Data Type”) розкрити список і

вибрати з нього потрібний.  В *Access 2007* з'явилася можливість взагалі "перекласти відповідальність" за тип даних на програму. Під час введення даних до таблиці система відслідковує характер даних, що вводиться, і пропонує вибрати відповідний тип даних. Список типів даних містить цікавий пункт "Майстер підстановок" ("Lookup Wizard"). Цей пункт дає змогу подати поле у вигляді переліку значень, з яких і тільки з яких користувач і буде вибирати потрібне. Підстановку доцільно застосовувати у двох випадках:

- Якщо поле має заздалегідь чітко визначену кількість значень. Наприклад, під час створення поля "Область мешкання" замість того, щоб вводити у поле досить довгі значення "Чернігівська", "Івано-Франківська", можна заповнити це поле за допомогою "Майстра підстановок" найменуваннями всіх областей України. Такий варіант доцільний за сталим, відносно невеликим переліком елементів, що практично ніколи не змінюється, оскільки будь-які зміни вимагатимуть модифікації структури таблиці.
- Якщо призначення поля тотожне призначенню поля іншої таблиці і значення його можна підключити з іншої таблиці. Наприклад, можна створити таблицю районів України і з цієї таблиці підключати за потреби назву району до іншої таблиці. Цей варіант більш гнучкий, ніж перший. Його доцільно використовувати, коли кількість елементів підстановки може бути великою, а їх значення постійно змінюються. За цим варіантом, не змінюючи структури таблиці, можна додавати необмежену кількість значень і редагувати ці значення.
-  У разі застосування майстра підстановок тип даних для певного поля буде визначатися типом даних поля з таблиці, з якої підставляються значення, а сам процес підстановки є нічим іншим, як створенням зв'язків між таблицями.

4. Після закінчення створення структури натиснути кнопку

 (“Зберегти”).

Під час створення таблиці доцільно – але не обов’язково – визначити *ключове (індексоване) поле*.  Це допоможе надалі під час створення зв’язків між таблицями. Крім цього, на індексоване поле можна накласти умову, яка не дозволяє вводити до нього повторювані записи, тобто виключає їх дублювання. Отже, ключове поле – це таке поле (або група полів), значення якого *однозначно* визначає кожний запис таблиці і не може мати два однакових значення. Наприклад, для студента в ролі ключового поля може бути номер залікової книжки або номер паспорта, для робітника підприємства – табельний номер, номер паспорта чи ідентифікаційний податковий номер. Як зазначалося раніше, під час створення структури таблиці до неї автоматично включається поле з ім’ям “ID”, що за замовчуванням і визначається як ключове. У найпростішому випадку з цим можна і погодися.

Для визначення ключового поля слід натиснути на його імені правою кнопкою миші й у контекстному меню вибрати пункт  “Ключове поле” (“Primary Key”). У результаті цієї дії ліворуч від поля з’явиться значок  . Піктограма у вигляді ключа вибрана не випадково, тому що, дійсно, можна сказати, що ключове поле тотожне ключу від квартири, тобто для кожного запису воно є унікальним. *Access* автоматично індексує таблицю за значенням ключа. Слід також зауважити, що програма дозволяє створювати ще й додаткові індекси за значеннями інших полів.

Щойно створена таблиця не має записів, а містить тільки назви полів-стовпчиків, що складають структуру таблиці.

 Зміна структури таблиці (складу полів або їх властивості) також здійснюється в режимі конструктора.

Властивості полів

Поля таблиці не просто визначають її структуру і тип – вони ще визначають і властивості даних, що записуються у поля. Так, наприклад, для текстового типу даних основні властивості такі:

- **Ім'я поля** (Field Name). Визначає, як слід звертатися до даних цього поля.
- **Тип даних** (Data Type). Визначає тип даних для цього поля.
- **Розмір поля** (Field Size). Максимальна довжина (у символах) даних у полі.
- **Формат** (Format). Засіб форматування даних у полі.
- **Маска вводу** (Input Mask). Форма, в якій вводяться дані у поле.
- **Підпис** (Caption). Заголовок стовпчика для поля (якщо підпис відсутній, то як заголовок стовпчика використовується властивість "Ім'я поля").
- **Значення за промовчанням** (Default Value). Значення, що вводяться у поле автоматично. Якщо під час введеня даних часто повторюється якесь значення, то доцільно "записати" його у цю властивість. Це звільнить користувача від необхідності кожного разу вводити його у поле у разі додавання нового запису у таблицю.
- **Правило перевірки** (Validation Rule). Встановлює обмеження на значення даних, що вводяться.
- **Текст перевірки** (Validation Text). Текстове повідомлення, що видається у разі невиконання правила перевірки.
- **Обов'язково** (Required). Встановлює обов'язковість введення значення для даного поля.
- **Індексовано** (Indexed). Призначене для швидкого пошуку запису.

Для різних типів даних набір властивостей відрізняється.

Знання сутності властивостей і вміле їх застосування дозволяє, зокрема, суттєво зменшити обсяг помилок під час

введення інформації. Наприклад, за допомогою масок введення розробник бази даних може примусити користувача вводити інформацію тільки у правильному форматі.

Таблиця 5

Основні маски введення

Символ (маска)	Дія маски
0	Цифра. Введення інформації обов'язкове. Знаки "+" і "-" є неприпустимими.
9	Цифра або проміжок. Введення інформації не обов'язкове. Знаки "+" і "-" є неприпустимими.
#	Цифра або проміжок. Введення інформації не обов'язкове. Відсутні символи перетворюються на проміжки. Можливе введення символів "+" і "-".
L	Буква (латиниця або кирилиця). Введення обов'язкове.
?	Буква (латиниця або кирилиця). Введення не обов'язкове.
A	Буква (латиниця або кирилиця) чи цифра. Введення обов'язкове.
a	Буква (латиниця або кирилиця) чи цифра. Введення не обов'язкове.
&	Будь-який символ або проміжок. Введення обов'язкове.
C	Будь-який символ або проміжок. Введення не обов'язкове.

Введення даних

Для введення даних слід вибрати у стрічці вкладок пункт "Основне", вибрати на панелі об'єктів потрібну таблицю і перейти до режиму таблиці.

Для заповнення таблиці даними слід встановити курсор у потрібне поле і почати вводити дані. Перехід від поля до поля

здійснюється клавішами навігації, **<Enter>** або **<Tab>**. Якщо дані візуально не повністю відображаються у полі, його можна збільшити, перетягнувши його межу або двічі натиснувши на правій межі у рядку заголовку стовпчиків. Перехід до наступного запису здійснюється автоматично після заповнення останнього поля поточного запису. Після введення у таблицю даних їх зберігати не потрібно, оскільки вони зберігаються автоматично під час їх введення у таблицю.

Вище було згадано, що в *Access 2007* з'явилася можливість узагалі "перекласти відповідальність" за тип даних на програму. Під час введення даних до таблиці система відслідковує характер даних, і за потреби запропонує змінити тип даних. Рис. 3 відображає таку ситуацію. У даному випадку до поля, що має числовий тип даних, була здійснена спроба ввести дані текстового типу. Програма попередила користувача про таку невідповідність: "Введене значення не відповідає типу даних Число у цьому стовпчику" ("The value you entered does not match the Number data type in this column") і запропонувала два варіанти:

1. ввести нове (інше) значення ("Enter new value");
2. перетворити тип даних поля на текстовий тип ("Convert the data in this column to the Text data type").

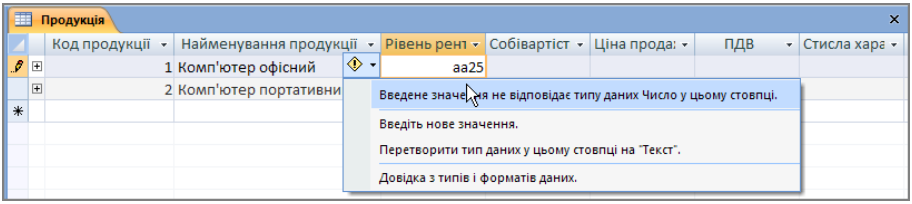


Рис. 3. Повідомлення при спробі введення некоректних даних

Введення даних у *пов'язану* таблицю зручно здійснювати з головної таблиці. Для цього після відкривання головної таблиці слід натиснути на піктограмі , що знаходиться у першому стовпчику. Після цього під рядком з'являються ("розгортаються")

записи пов’язаної таблиці, які відповідають *вибраному рядку* головної таблиці. Приклад такого введення даних подано на рис. 4.

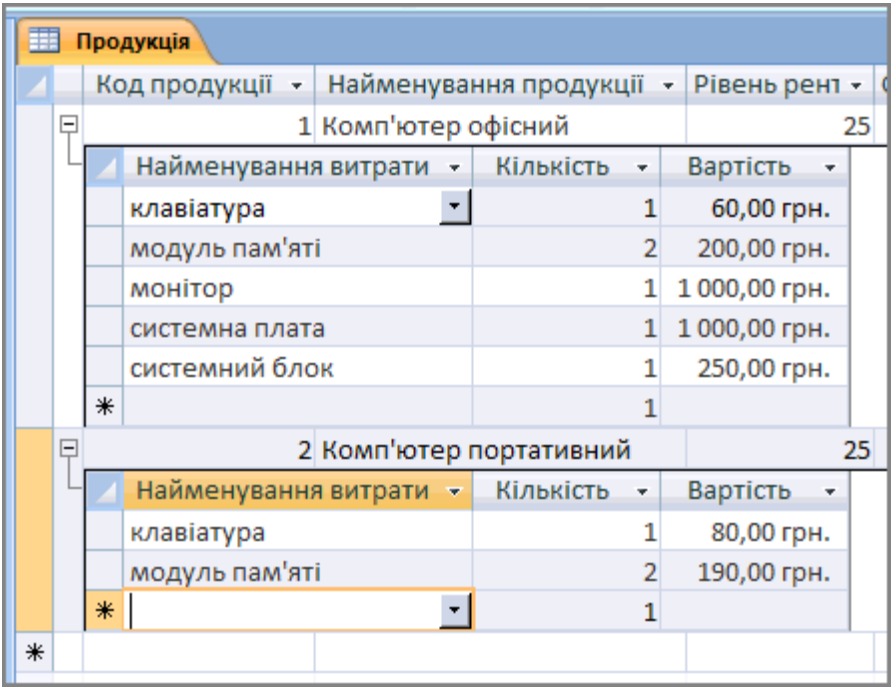


Рис. 4. Приклад введення даних через взаємопов’язану таблицю

Ще одна суттєва перевага такого способу введення полягає в тому, що під час додавання нових записів *автоматично* заповнюється поле, за яким здійснюється зв’язок із головною таблицею. Для закривання списку записів пов’язаної таблиці слід натиснути кнопку .

Переміщення за таблицею здійснюється за допомогою клавіш навігації. Швидкий перехід до першого або останнього запису таблиці забезпечує використання комбінацій функціональних клавіш **<Ctrl>+<Home>** і **<Ctrl>+<End>**. Разом із тим нижня частина вікна з таблицею містить панель навігації, якою зручно

користуватися під час переміщення по таблиці з великою кількістю записів.

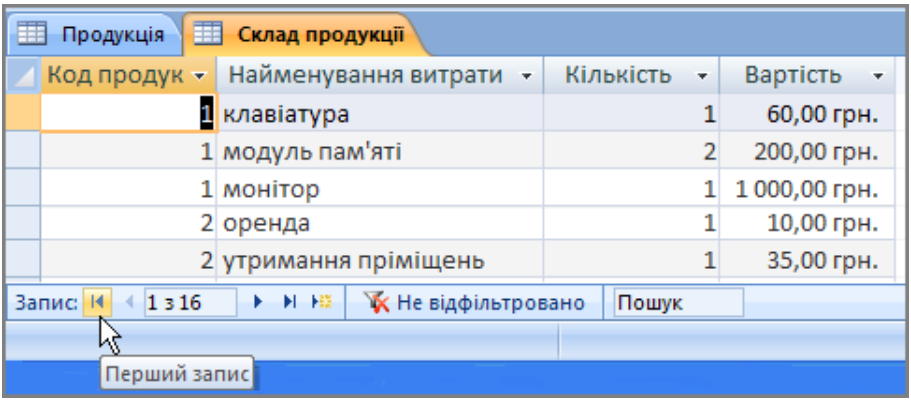


Рис. 5. Панель кнопок переходу

Інколи під час роботи з таблицею може виникнути питання відбору певної її частини або пошуку серед її записів тих, що відповідають певній умові. Взагалі для такого роду дій існують спеціальні об’єкти – запити, про створення яких мова піде нижче, але у найпростіших випадках можна обійтися і без них. У найпростішому варіанті для пошуку слід ввести контекст для пошуку до поля “Пошук” (“Search”) з панелі навігації. Більш складні дії пошуку (до речі, і заміни) надає група команд пошуку зі стрічки команд. Ще більш зручний варіант відбору даних забезпечують фільтри, які надають можливість відобразити на екрані тільки потрібні записи. Роботу з фільтрами забезпечує група команд фільтрації пункту стрічки “Основне”.

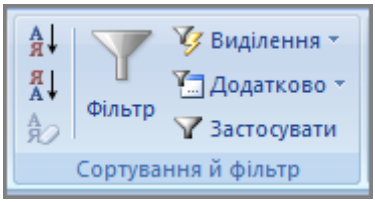


Рис. 6. Інструменти фільтрації

Дані до таблиці можуть бути також введені за допомогою *форми*, якщо вона для неї створена. Такий варіант завжди є більш зручним для користувача, оскільки він дає змогу сконцентрувати увагу на даних, які стосуються певного запису. Процедура побудови форм буде викладена нижче.

Створення зв'язків між таблицями

Дві таблиці можна пов'язати між собою на рівні їхніх полів, що мають *однакове походження*, встановивши тим самим між ними *реляційне відношення*. При цьому одна з них вважається *головною*, а інша – *пов'язаною*. Головна – це та таблиця, що бере участь у зв'язку своїм *ключовим полем*. Як правило, одному запису в головній таблиці відповідає багато записів у пов'язаній.

Зв'язки між таблицями дозволяють:

1. Підключити дані з однієї таблиці в іншу під час створення інших об'єктів.

2. Виключити можливість вилучення або зміни даних у ключовому полі головної таблиці, якщо з цим полем пов'язані поля інших таблиць.

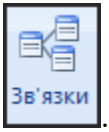
3. У разі видалення або зміни даних у ключовому полі головної таблиці автоматично і коректно відбуваються відповідні зміни даних у пов'язаних таблицях.

Таким чином, створення реляційних зв'язків між таблицями дає змогу звести до мінімуму дублювання даних, а крім того, вирішити такі важливі завдання, як захист даних і автоматизація внесення змін відразу до кількох таблиць при внесенні змін в одну

таблицю.

Під час створення зв'язків доцільно, щоб поля у таблицях, за якими встановлюється зв'язок, мали однакове найменування. Також обов'язково, щоб вони мали *однаковий тип даних* та *однакові значення властивостей*. Якщо ця умова порушена, то слід скасувати всі зв'язки, усунути розбіжності і заново їх побудувати.

Встановлення зв'язку між головною таблицею А і пов'язаною таблицею Б здійснюється таким чином:

1. Вибрати у головному меню пункт "Знаряддя бази даних" ("Database Tools") і натиснути кнопку 

. У робочій області вікна відкриється вікно з переліком таблиць, що були підключені до схеми раніше. Слід мати на увазі, що таблиця може бути створена, але не підключена до схеми зв'язків таблиць.

2. За першим зверненням до цього пункту одночасно з'явиться вікно "*Відображення таблиці*" ("Show Table"), в якому відображаються наявні таблиці. Для додавання до схеми потрібної таблиці слід встановити на неї курсор і натиснути кнопку **«Додати»** ("Add"). У вікні з'явиться список полів цієї таблиці. За наступними зверненнями до цього вікна вікно "*Відображення таблиці*" автоматично з'являтися вже не буде. Для його відображення слід викликати контекстне меню у вільному місці вікна "*Зв'язки*" і вибрати у ньому пункт "Відобразити таблицю" ("Show Table").

3. Перетягнути потрібне поле таблиці А на поле у списку таблиці Б, за яким встановлюється зв'язок. При відпусканні кнопки відкриється вікно "*Редагування зв'язків*" ("Edit Relationships").

4. Вікно "*Редагування зв'язків*" містить поле-мітку "Забезпечення цільності даних" ("Enforce Referential Integrity"). Саме встановлення прапорця біля цього поля і забезпечує

автоматизацію внесення змін відразу до кількох таблиць при зміні в одній з них. У результаті не можна буде вилучити запис із головної таблиці, якщо пов'язані з нею містять ці дані. Наприклад, якщо головна таблиця містить анкетні дані студентів, а пов'язана – оцінки студентів, то не можна буде вилучити запис з анкетними даними певного студента, доки таблиця з оцінками буде містити дані з його оцінками. Але інколи якраз і виникає потреба вилучити усі пов'язані між собою дані. Наприклад, така ситуація виникає, якщо студент залишає навчальний заклад. Для розв'язання цієї проблеми достатньо встановити прапорець біля дії "Каскадне видалення пов'язаних полів" ("Cascade Delete Related Fields").

5. Натиснути кнопку **«Створити»** ("Create"). У вікні зв'язків між двома таблицями має встановитися зв'язок. Візуально він має вигляд лінії, що з'єднує поля двох таблиць.  Під час натискання кнопки може з'явитися повідомлення "Механізм бази даних не може заблокувати таблицю, тому що вона використовується іншим користувачем або процесом" ("The database engine could not lock table … because it is already in use by another person or process"). Зазвичай це є ознакою того, що з таблицею відбувається робота, відповідно на одному з ярличків таблиці знаходиться її ім'я. Для створення зв'язку слід закінчити роботу з таблицею, викликавши контекстне меню на її ярличку і вибравши з меню пункт "Закрити" ("Close").

 За потреби розірвати (скасувати) зв'язок між таблицями потрібно у вікні зв'язків викликати контекстне меню на лінії зв'язку і з контекстного меню вибрати пункт "Видалити" ("Delete").

Якщо до схеми даних помилково двічі додана одна й та ж таблиця, слід її виділити і натиснути функціональну клавішу **<Delete>**.

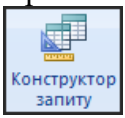
Робота із запитами

Якщо користувачеві потрібно одержати дані з бази даних, він повинен сформувати і застосувати для цього спеціальний об'єкт бази даних, який має назву *запит*. Запит є операцією відбору певної частини даних з однієї або кількох таблиць за заданими критеріями. За допомогою запитів виконують також такі операції над даними, як сортування (впорядкування), фільтрація, перетворення даних за заданим алгоритмом, створюють нові таблиці, здійснюють автоматичне наповнення таблиць даними з інших джерел, виконують обчислення за певними полями і т. ін. Запити дозволяють також набагато зручніше візуально відобразити для користувача самі поля таблиць. Сам запит – це певна *результуюча* таблиця, що створюється на основі первинних таблиць. Таким чином, первинні таблиці – найцінніша складова бази даних – залишаються *незмінними*, тобто користувач не працює з ними безпосередньо.

Створення запиту

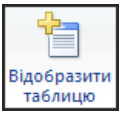
Запити до баз даних пишуться спеціальною мовою програмування, що була розроблена корпорацією IBM у 1970 р. і має назву *мова структурованих запитів – Structured Query Language* (*SQL*). У наш час вона є стандартом де-факто як мова реляційних баз даних. Але під час роботи з програмою знати цю мову зовсім не обов'язково, оскільки у ній створення запитів проводиться візуальним шляхом, із використанням так званої технології *QBE* (*Query By Example*).

Створення запиту за зразком, який у найпростішому вигляді має назву *запит на вибірку*, здійснюється так.

1. Вибрати на стрічці вкладок пункт "Створити".
2. На стрічці команд натиснути кнопку 

(Створення запиту в режимі конструктора).
3. Відкриється спеціальний бланк, що має назву *бланк запиту*

за зразком. Він складається із двох частин. У верхній відображається структура таблиць, із яких він буде отримувати дані, а нижня складається із стовпчиків, які будуть містити поля запиту.

4. Додати у запит ті таблиці, з яких слід одержати дані. Це можна зробити у початковому вікні створення запиту "Відображення таблиці" ("Show Table"), яке містить список-перелік таблиць бази даних. Для додавання таблиці слід вибрати її зі списку і натиснути кнопку **«Додати»** ("Add"). Це вікно можна у будь-який час відобразити або викликавши контекстне меню у верхній частині запиту і вибравши з нього пункт "Відображення таблиці", або натисканням кнопки 

на стрічці команд.

5. Перетягнути з верхньої частини бланка у нижню ті поля, які слід включити до запиту в потрібному порядку.

6. Сформувавши структуру запиту, його закривають і надають йому змістовне ім'я.  Оскільки запит також є *таблицею*, то його ім'я не повинно співпадати з іменами не тільки однотипних об'єктів-запитів, але і з іменами об'єктів-таблиць.

7. Ім'я запиту з'явиться на панелі об'єктів поруч з іменами таблиць, що використовуються під час побудови запиту.

Одержання даних із запиту, тобто його виконання, може бути здійснено різними шляхами:

1. Викликати контекстне меню на імені запиту на панелі об'єктів і вибрати з нього пункт "Відкрити" (Open).

2. Двічі натиснути на імені запиту на панелі об'єктів.

3.  При налагоджуванні запиту швидкий перегляд результатів його виконання, не виходячи при цьому з режиму роботи у конструкторі запитів, здійснюється шляхом натискання кнопки 

("Run") на стрічці команд конструктора запиту.

 *Побудова виразів*

Досить часто значення у полі запиту повинно містити не просто поле однієї з таблиць, а комбінацію полів з однієї або кількох таблиць. Наприклад, у запиті слід підрахувати загальну вартість товару, яка визначається як добуток загальної кількості товару і ціни за одиницю товару. Для розв'язання цієї проблеми *Access* має спеціальний інструмент – "Побудовник виразів". Побудова виразу за його допомогою здійснюється так.

1. Викликати контекстне меню на *незаповненому* стовпчику бланка запиту і вибрати у ньому пункт "Побудувати" ("Build") або натиснути на стрічці кнопку  ("Конструктор"). З'явиться вікно "*Побудовник виразів*" ("Expression Builder").

2. У нижній лівій частині вікна відображається перелік усіх об'єктів в ієрархічному вигляді. Розкрити вузол "Таблиці" ("Tables") (це може бути й інший об'єкт – форма, запит тощо) і вибрати з його складу потрібну таблицю. В середині нижньої частини вікна відобразиться перелік полів таблиці.

3. Вибрати у цьому переліку потрібне поле, двічі натиснувши на ньому, або натиснути кнопку **«Вставити»** ("Paste").

4. Вибране поле у вигляді [Ім'я таблиці 1]![Ім'я поля 1] з'явиться у верхній частині вікна, яка і призначена для формування виразу.

5. Для застосування арифметичних операцій над полями таблиць під верхньою частиною вікна розташовані кнопки із зображеннями таких дій. Наприклад, для застосування дії множення слід натиснути кнопку із зображенням символа  . Після цього верхня частина вікна буде містити вираз: [Ім'я таблиці 1]![Ім'я поля 1] *.

6. Для подальшого формування виразу знову виконуються дії, перелічені у пунктах 2–5 до повної побудови виразу. Наприклад, у нашому прикладі остаточний вираз буде мати вигляд: [Ім'я таблиці 1]![Ім'я поля 1] * [Ім'я таблиці 2]![Ім'я поля

2].

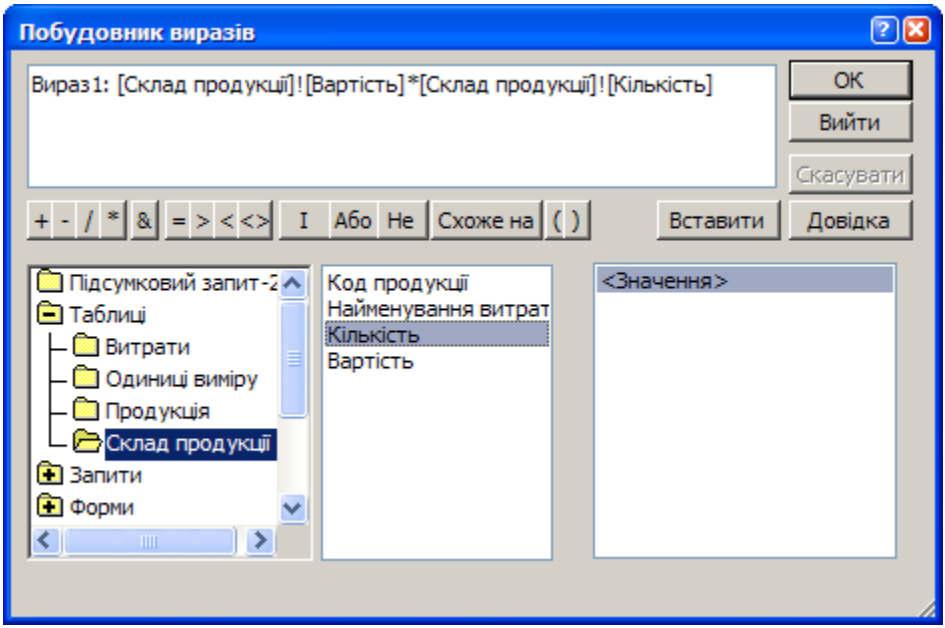


Рис. 7. Вікно побудовника виразів

За потреби включення до запиту полів кількох таблиць, звернення до поля кожної таблиці здійснюється таким чином: [Ім’я таблиці]![Ім’я поля], тобто ім’я таблиці та ім’я поля поміщаються у квадратні дужки, а між ними ставиться символ “!”. Цей символ є спеціальним оператором, що використовується для опису взаємозв’язку між об’єктами. Наприклад, *[Прихід]![Кількість]*.

За замовчуванням під час побудови виразів для стовпчика бланка запиту застосовується підпис “Вираз”. Звичайно, така назва для користувача не дає інформації про те, що саме містить цей стовпчик. Тому слід встановити курсор у полі “Вираз” і замінити цей текст на змістовну інформацію, наприклад,

“Загальна сума товару”, “Разом” і т. ін.

Використання у виразах вбудованих функцій

У виразах можна використовувати функції, які становлять окремий об’єкт. Система містить достатньо велику кількість вбудованих функцій, згрупованих за функціональним призначенням у кілька груп: масиви, дата й час, фінансові, математичні, і та ін.

Наприклад, розробник бази даних має зручні можливості під час роботи з датами, які надають функції дати і часу. Наведемо деякі з цих функцій:

1. ***Рік[Year]***(ім’я_поля). Повертає число – номер року.
2. ***Місяць[Month]***(ім’я_поля). Повертає число – порядковий номер місяця у році.
3. ***День[Date]***(ім’я_поля). Повертає число – порядковий номер дня в місяці.
4. ***Дата[Date]***(). Повертає поточну дату у вигляді цілого числа.
5. ***Now()***. Повертає поточну дату и час у вигляді числа подвійної точності.

Комбінування цих функцій дозволяє дуже просто оперувати з датами. Наприклад, для визначення кількості років слід побудувати такий вираз:

```
Рік(Дата()) - Рік(Дата_народження)
```

де “Дата_народження” – ім’я поля, що містить дані типу “дата”.

Упорядкування (сортування) записів

Якщо необхідно, щоб дані, відібрані у результаті запиту на вибірку, були впорядковані за певним полем, застосовують *сортування*. Для цього у нижній частині бланка запиту є рядок “Сортування” (“Sort”). Натискання на цьому рядку для будь-якого поля призводить до появи кнопки списку, з якого можна вибрати метод сортування: *за зростанням* або *за спаданням*. Після вибору

цього методу дані у запиті будуть упорядковані за тим полем, для якого була вибрана дія сортування.

Можливе і *багаторівневе сортування*, тобто сортування відразу за кількома полями.  У цьому разі дані, для яких слід здійснити сортування, у запиті слід розташувати в тому порядку, в якому їх потрібно впорядковувати: послідовно зліва-направо.

Використання умов відбору

Використання умов відбору забезпечує відбір даних за заданим критерієм. Відповідний рядок для відбору розташовано в нижній частині бланка запиту, він має ім'я "Критерії" ("Criteria").

Для кожного поля у цьому рядку можна задати індивідуальну умову. Наприклад, необхідно відібрати всіх студентів, що мають оцінки "4" і "5" і мешкають у місті Ніжин. Для цього у полі, що містить оцінки, задається "Критерії" " >3 " і у полі, де знаходяться адреси, вводиться текст "Ніжин".

Ситуація, за якої виникає потреба накладання умов на значення різних полів або створення подвійної умови для того самого поля, виникає досить часто. У цьому випадку застосовують *логічні оператори*. Зазначимо, що логічні оператори взагалі широко застосовуються під час створення запитів мовою SQL у разі формування різноманітних умов та обмежень (кількість яких у загальному випадку обмежується тільки синтаксисом мови) над значеннями полів таблиць баз даних.

До логічних операторів належать наступні.

1. "***Або***" ("**Or**"). ***Логічне додавання***. Здійснюється над двома значеннями і передбачає залучення до вибірки записів, для яких виконується умова на значення реквізиту ліворуч чи праворуч від логічної операції. Наприклад, якщо необхідно відібрати усі товари, вартість яких більша за 1000 грн. або менша 10 грн., то у полі, що містить вартість, слід записати умову <10 Or >1000.  Логічні оператори в *Access* записуються в англомовному варіанті та обов'язково мають ліворуч і праворуч як мінімум один

проміжок.  Для створення такої умови можна було просто записати у стовпчику бланка, що містить вартість товару, в одному рядку значення “<10”, а в другому – “>1000”. Після повторного відкривання бланка запиту введені значення перетворюються на умову <10 Or >1000. Таким чином, якщо ввести два значення в одне й те саме поле, то створюється умова з оператором “або”.

2. “***Та***” (“***And***”). ***Логічне множення***. Здійснюється над двома різними реквізитами або різними значеннями того самого реквізиту і передбачає включення до вибірки таких записів, для яких значення у полі відповідає умовам і ліворуч і праворуч від логічної операції. Наприклад, якщо необхідно відібрати усі товари, вартість яких не менше 10 грн., але і не більше 1000 грн., то у полі, що містить вартість, слід записати умову > 10 And < 1000.  При використанні умов відбору в різних полях бланка запиту між ними автоматично створюється умова з оператором “та”. Таким чином, при побудові щойно розглянутого запиту, у якому потрібно було відібрати всіх студентів, що мають оцінки “4” і “5” і мешкають у місті Ніжин, значення полів для оцінок і місця мешкання будуть пов’язані саме оператором логічного множення.

3. “***Ні***” (“***Not***”). ***Логічне заперечення***. Здійснюється тільки над одним реквізитом. Наприклад, якщо необхідно відібрати усі товари, вартість яких не дорівнює 10 грн., то у полі, що містить вартість, слід записати умову Not 10.  Замість ключового слова “Not” можна використати комбінацію символів “<>”, яка в мові SQL традиційно використовується для позначення відношення “не дорівнює”.

Ще один варіант відбору дає змогу відібрати записи за принципом подібності за допомогою ключового слова ***Like*** (“Схоже на”). ***Ключове слово*** – це найменування, що зарезервоване в системі з певною метою. Так, якщо потрібно відібрати всіх студентів, які мешкають у Києві або Кіровограді, то

у поле “Критерії” можна ввести текст “Like К*”.  Зауважимо, що такий варіант відбору застосовується до полів, що мають тип даних “текст”, “примітка” або “гіперпосилання”. Для відбору використовується й оператор ***Between...And***. Він визначає належність значення виразу *заданому діапазону*. Його синтаксис:

Between значення_1 And значення_2,

де *значення_1*, *значення_2* – вирази, що задають межі діапазону. Використання цього оператора аналогічне використанню логічного множення. Повертаючись до щойно розглянутого прикладу, коли було необхідно відібрати усі товари, вартість яких не менше 10 грн., але і не більше 1000 грн., то у полі, що містить вартість, можна було застосувати оператор ***Between***:

Between 10 And 1000

Цей варіант відбору застосовується до полів, що мають тип даних “число”, “грошова одиниця” або “автонумерація”.

Групування записів

Групування надає можливість об’єднати *однакові за якоюсь ознакою записи таблиці для певного поля* у групи і застосувати до них обчислення за допомогою кількох функцій. Потрібна функція вибирається зі списку в рядку “Підсумок” (“Total”) бланка запиту. За відсутності цього рядка слід викликати контекстне меню в нижній частині бланка і вибрати з нього пункт “Підсумки” або натиснути на стрічці команд кнопку 

.

Наприклад, функція ***Сума*** (***Sum***) надає можливість підрахувати для групи записів суму, функція ***Середнє*** (***Avg***) – обчислити середнє значення і т. ін. Так, якщо здійснити групування студентів за їх прізвищем (номером залікової книжки або іншим унікальним реквізитом студента) у таблиці, що містить дані з оцінками студентів, то за допомогою функції ***Сума*** можна

підрахувати загальну суму оцінок для кожного студента, а за допомогою функції ***Середнє*** – обчислити середній бал для кожного студента.

Інші види запитів

Вище розглядався запит на вибірку – найпростіший і найпоширеніший вид запиту. Існують і інші види запитів, деякі з них виконуються на базі попередньо складеного запиту на вибірку. До них належать:

1. *Запити з параметром*. Критерій відбору задає сам користувач, який вводить потрібне значення (*параметр*) при виклику запиту. Він формується для поля, за яким передбачається відбір даних, в умові "Критерії". Вираз параметра складається зі знака логічної умови і тексту-пояснення у квадратних дужках, що буде з'являтися під час виклику запиту. Наприклад, для відбору робітників, розмір заробітної плати яких *не перевищує* певного значення (*параметра*), необхідно у полі "Критерії" ввести вираз:

<[Введіть розмір заробітної плати]

2. *Підсумкові запити* – здійснюють математичні обчислення за заданим полем і відображають результат. Для розрахунку підсумкових значень необхідна наявність у бланку запиту рядка "Підсумок". За його відсутності у бланку запиту на стрічці вкладок слід натиснути кнопку 

«Підсумки». Якщо у бланку запиту натиснути на рядку "Підсумок", то з'явиться кнопка списку, з якого можна вибрати підсумкову функцію для розрахунку в даному полі.

3. *Запити на зміну* – дозволяють автоматизувати заповнення поля таблиці.

4. *Перехресні запити* – дозволяють створювати результуючі таблиці на основі результатів розрахунків, отриманих під час аналізу групи таблиць.

5. *Специфічні запити SQL* – запити до сервера баз даних,

написані мовою запитів *SQL*.

Робота з формами

Форма призначена для введення даних у певну таблицю. Раніше йшлося про введення даних у режимі роботи з таблицями, але застосування форм надає набагато більше можливостей як користувачеві, так і розробнику бази даних. У формі можна забезпечити більш потужний контроль введення даних, розташувати їх на екрані у зручному вигляді, максимально спростити процедуру введення даних за рахунок додавання у форму таких елементів, як списки, керуючі кнопки і т. ін. Сама форма може містити тільки частину полів таблиці, що дає змогу обмежити обсяг інформації, доступної тому чи іншому користувачеві. Для кожної категорії користувачів можна створити окрему форму, що буде мати свій набір полів. Таким чином вирішується питання прав користувачів одержувати доступ до певних даних таблиці. За допомогою форм можна не тільки вводити, але і відображати дані.

Створення форми

У загальному випадку для створення форми слід виконати такі дії:

1. Вибрати на стрічці вкладок пункт “Створити” (“Create”).

1. На панелі об’єктів встановити курсор на таблицю, для якої слід створити форму.

2. Вибрати варіант створення форми:

- в режимі конструктора (ручний варіант);
- за допомогою майстра (автоматизований варіант);
- у режимі “автоформа” (автоматичний варіант).

Перший варіант більш трудомісткий, але він дозволяє:

1. зробити вигляд форми максимально зручним для користувача;

2. додати у форму керуючі елементи (сторінки, кнопки і т. ін.);

3. автоматизувати заповнення окремих полів за допомогою списків, прапорців і т. ін.;

4. здійснити потужний контроль у разі введення даних.

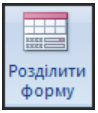
За другим і третім варіантом форма створюється просто і швидко, але при цьому вона досить часто є “примітивною”, тому ці варіанти доцільно використовувати під час створення форм для таблиць з найпростішою структурою даних.

 За будь-яким варіантом побудови програма створює форму для тієї таблиці або запиту, у списку об’єктів якої на панелі об’єктів встановлено курсор.

Автоформи

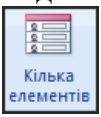
Форми для таблиць простої структури доцільно створювати за допомогою засобів автоматизації побудови форм. Форми, побудовані цілком автоматично, називаються *автоформами*. Це найбільш простий і швидкий спосіб створення форм. *Access* дає змогу створювати три види автоформ, які розрізняються за способом розташування у них полів.

Форма, що призначена для введення (відображення) тільки одного запису. Поля запису подаються послідовно “стовпчиком” один над одним. Такий спосіб подання даних у формі є найкращим. І річ навіть не в тому, що така форма відображає у зручному вигляді склад полів.  Головним є те, що за умови наявності для таблиці пов’язаної форми, дані пов’язаної таблиці, що відносяться до поточного запису таблиці, для якої створена форма, також відображаються у формі.



Форма, інформація якої розподілена на дві частини. Практично це комбінація з двох попередніх видів форм: верхня частина відображає один запис, а в нижній частині подаються усі

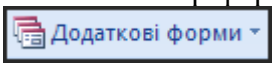
записи в табличному вигляді. Вибір запису у табличній частині призводить до її відображення у верхній частині.



Форма, що складається з багатьох записів, які подаються в табличному вигляді.

Натискання на кнопку, що символізує певний вид форми, призведе до автоматичного створення форми. Одночасно з'явиться стрічка з елементами форматування, які можна застосувати до форми.

Створення форм за допомогою майстра

Автоматизовані засоби побудови форм надає *Майстер форм* – спеціальна програма, що створює структуру форми в режимі діалогу з розробником. Майстер форм можна запустити, обравши зі списку кнопки 

("More Forms") елемент  "Майстер форм" ("Form Wizard"). Як вище було згадано, програма створює форму для тієї таблиці або запиту, на який встановлено курсор на панелі об'єктів, тому доцільно попередньо вибрати потрібну таблицю. Після завантаження майстер форм пропонує користувачеві виконати кілька кроків:

1. Вибрати зі списку "Доступні поля" ("Available Fields") поля таблиці, які слід включити до форми, тобто перемістити назви потрібних полів до списку "Вибрані поля" ("Selected Fields"). Для цього слід натиснути кнопку 

, якщо потрібно включити до звіта усі поля об'єкту, або вибирати їх одне за одним за допомогою кнопки 

.

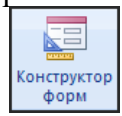
2. Натиснути кнопку **«Далі»** ("Next"). Зауважимо, що на будь-якому кроці роботу майстра можна закінчити натисканням кнопки **«Готово»** ("Finish").

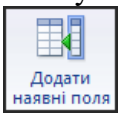
3. Вибрати вигляд форми.

4. Вибрати стиль форми.

5. Надати формі ім’я.

Створення форми в режимі конструктора

Для створення форми в режимі конструктора на стрічці команд слід натиснути кнопку 

(Конструктор форм). Ця дія ініціює появу вікна з іменем “*Форма*” (“Form”). Вікно є порожнім бланком майбутньої форми. Якщо на стрічці команд “утоплена” кнопка 

(Додати наявні поля), то одночасно праворуч від вікна “*Форма*” з’явиться вікно “*Список полів*” (“Field List”) зі списком усіх таблиць, а для вибраної таблиці буде відображений перелік полів.  Якщо вікно “*Список полів*” не з’явилося, натисніть на стрічці команд кнопку 

(Додати наявні поля).

Структура форми

Форма має три області: область заголовка, область даних і область приміток. Призначення області заголовка зрозуміло – тут, як правило, розташовують загальну назву форми. В області приміток може бути виведена деяка пояснювальна або довідкова інформація. Разом із тим у цій області часто розташовують підсумкові значення (докладно це питання буде висвітлено пізніше).  За замовчуванням області заголовка і приміток відсутні. Для відображення їх у формі слід викликати в області форми контекстне меню та вибрати пункт “Заголовок/примітка форми” (“Form Header/Footer”) або на стрічці команд натиснути кнопку  “Назва” (“Title”).

Основою форми є *область даних* – область, де безпосередньо розташовуються дані. Слід зазначити, що розміри областей можна довільно змінювати. Для цього потрібно встановити курсор на лінію, що їх розділяє (курсор при цьому повинен набути форми ), і перетягнути цю лінію у потрібне місце форми.

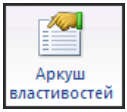
Елементи керування формою

Елементи керування надають можливість контролювати дані, наприклад, форму їх відображення, керують даними за допомогою виразів, макросів, процедур тощо. Так, дані того самого поля можна подати як текст, у вигляді списку або поля зі списком. При цьому *тип даних* цього поля не змінюється, змінюється *форма* відображення даних. Вибираючи форму подання даних, розробник у першу чергу повинен подбати про зручність введення даних користувачем. Так, наприклад, зрозуміло, що практично завжди зручніше вибрати значення зі списку, ніж ввести його вручну. Елементи керування формою містяться на вкладці "*Конструктор*" ("Design"). Ці команди є головним інструментом, за допомогою якого відбувається побудова форми. Для того щоб додати до форми елемент, слід його вибрати натисканням відповідної піктограми. Після цього курсор набуває вигляду вибраного елемента, що дає змогу визначити, з яким саме елементом зараз відбувається робота. Курсор встановлюється у потрібну позицію форми, натискається ліва кнопка миші і звичайним розтягуванням меж елемента надається потрібний розмір (рис. 8).

Рис. 8. Зміна розмірів елемента керування

Разом з елементом вставляється його приєднаний напис, що є

комбінацією імені елемента і числового значення, яке відповідає порядковому номеру цього типу елемента у формі. Наприклад, для прапорців це буде “Прапорець1”, “Прапорець2” і т.д. Зрозуміло, що такий текст не надає користувачеві інформації про призначення цього поля. Для того щоб зробити напис змістовним, слід його змінити. Для цього необхідно натиснути на ньому мишею, після чого по його периметру з’являться маркери. Надалі потрібно просто змінити текст у прямокутнику, обмеженому маркерами.

Кожен елемент керування формою має свої *властивості*, набір яких є сталим для однакових елементів, але різний для різних елементів. Для відображення їх списку слід викликати на елементі контекстне меню й вибрати з нього пункт “Властивості” (Properties”) або на стрічці команд натиснути кнопку 

(“Аркуш властивостей”).

У посібнику не ставиться за мету розгляд властивостей, застосування яких обумовлюється як практичною доцільністю, так і особистими симпатіями (і знаннями) розробника. Разом із тим автор вважає за доцільне звернути увагу на властивість “Правило перевірки”. Як зазначалося раніше, така сама властивість, за допомогою якої можна здійснити контроль над даними під час їх введення, може бути застосована для полів таблиці в режимі конструктора таблиць. Таким чином, засоби програми надають можливість встановлення контролю над введенням даних для різних об’єктів. Властивість “Правило перевірки” знаходиться на вкладці “*Дані*” вікна “*Аркуш властивостей*”. Значення умови можна сформувати двома способами. За першим варіантом здійснюється ручне формування виразу умови. Наприклад, у таблиці з відомостями про студентів є поле “Курс” (на якому навчається студент). Зрозуміло, значення цього поля повинно бути не менше ніж “1”. Якщо це поле має назву “Курс”, то як умову можна записати вираз [Курс]>0. За

другим варіантом формування умови здійснюється в автоматизованому режимі за допомогою засобу "*Побудовник виразів*". Звернення до нього здійснюється шляхом натискання кнопки  .

Таблиця 6

Основні елементи керування формою

Елемент	Призначення
Напис	Створення текстових написів.
Текстове поле	Дані текстового або числового типу.
Прапорець	Використовується для відображення даних, що можуть мати тільки два взаємовиключних значення (наприклад, стать).
Поле зі списком, список	Створення списку значень. Надалі користувач здійснює введення даних у таке поле шляхом вибору зі списку.
Вкладка	Створення у формі вкладок.
Кнопка	Широко використовується для здійснення навігації по таблиці, звернення до інших об'єктів (таблиць, форм, звітів), наприклад, за допомогою макросу, і т. ін.
Лінія	Елемент графіки, за допомогою якого можна відокремити групи однорідних полів.

 **Створення напису**

Напис – це звичайний текст з довільною інформацією. Для створення такого поля слід натиснути на піктограмі напису у панелі елементів, встановити курсор у формі і надати створеному елементу потрібний розмір.

Створення поля

Поле – це головний елемент форми. Саме поля відображають дані, що містяться у таблицях. Для побудови поля слід перетягнути його зі списку полів на потрібне місце бланка форми.

Поле може бути створене не тільки для поля таблиці. Якщо поле не прив'язується до конкретного поля, то йому надається ім'я "Без прив'язки". Як правило, такі поля створюються для відображення результатів виконання деяких операцій над полем або полями. За таким варіантом це поле буде *обчислювальним*. Наприклад, якщо таблиця з даними про абітурієнта містить усі його екзаменаційні оцінки, то можна створити підсумкове поле, в якому буде розташована загальна сума балів абітурієнта під час складання екзаменів:

=[Оцінка 1] + [Оцінка 2] + [Оцінка 3]

Обчислювальне поле починається зі знака "=" і містить назви полів, що беруться у квадратні дужки.

Створення поля зі списком

Завжди користувачеві зручніше здійснювати введення інформації шляхом вибору значення зі списку, ніж вводити дані з клавіатури. Крім того, вибір зі списку дозволяє бути впевненим, що введене значення є допустимим. Для створення елемента типу список на панелі елементів вибирається "Поле зі списком". Після цього завантажується "Майстер полів зі списком", який допомагає будувати поле зі списком. На початку роботи майстер визначає, як будуть формуватися значення списку:

1. З таблиці або звіту.
2. Шляхом введення фіксованого набору значень.

За першим варіантом вибирається таблиця або запит і визначається поле, яке і буде джерелом значень для списку. Принциповим є те, що за цим варіантом після вибору зі списку можна відразу запам'ятати значення з іншої таблиці у полі

таблиці, для якої створена форма. Для цього на одному із кроків роботи майстра створення списків слід вибрати пункт “Зберегти в поле” та вибрати, в яке саме поле потрібно зберегти вибране значення. За другим варіантом розробник сам визначає перелік значень, із яких буде складатися список. Для цього на другому етапі побудови форми заповнюємо значеннями стовпчик (як правило, такий стовпчик потрібен один, хоча й є можливість визначити кілька стовпчиків).

 **Створення поля типу “Прапорець”**

Такі поля мають тільки два значення: “Так” або “Ні”. Наявність позначки біля такого поля у формі означає “Так”, відсутність – “Ні”. Для створення поля такого типу слід вибрати на панелі елементів елемент “Прапорець” і перетягнути його на форму. Після цього слід звернутися до властивостей цього елемента і на вкладці “*Дані*” для властивості “Джерело елемента керування” вибрати зі списку полів потрібне.

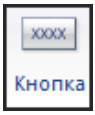
 **Створення вкладки**

Форми з кількома вкладками широко застосовуються для відображення об’єктів, що характеризуються багатьма реквізитами (наприклад, товарно-матеріальні цінності, особиста картка студента тощо), весь перелік яких дуже важко або взагалі неможливо розташувати в одній формі. Наприклад, у формі для співробітників реквізити можна згрупувати за тематичними вкладками “Загальні”, “Службові”, “Паспортні”, “Відомості для нарахування заробітної плати” і т. ін.

Для створення об’єкта на панелі елементів вибирається елемент “Вкладка”  . Створений елемент за замовчуванням має дві вкладки. При цьому користувач має можливість окремо змінювати властивості як кожної окремої сторінки (вкладки), так і усіх вкладок разом. У властивостях форми їм відповідають

об’єкти з іменами “Вкладка” з порядковим номером вкладки або “НабірВкладок”.

Після створення об’єкта за допомогою властивості “Ім’я” доцільно відразу ж надати кожній вкладці змістовну назву. Подальша робота з об’єктом нагадує загальну роботу зі створення форми. При цьому кожну вкладку слід розглядати як самостійну форму, яка має свій набір полів, написів та інших об’єктів.



Створення кнопки

Елемент “Кнопка” призначений для створення зручних умов роботи з таблицею. Наприклад, можна створити кілька кнопок навігації по таблиці, одна з яких буде відповідати за рух уперед, друга – назад, ще одна – на початок усіх записів і т. ін. Інша група кнопок може бути призначена для редагування таблиці: вилучення, додавання, копіювання записів. Під час створення кнопок зручно використовувати “Майстер кнопок”. Для цього на стрічці команд слід натиснути піктограму кнопки і виконати дії, запропоновані майстром.  Якщо після натискання піктограми кнопки у формі майстер не завантажується, тобто просто створюється кнопка, це означає, що режим “Майстер кнопок” вимкнений. Для його активації на стрічці команд натисніть кнопку  .

Завантаження форми

Завантаження форми відбувається після подвійного натискання на її назві або переходом до режиму перегляду на стрічці команд. Після цього на екрані з’являється форма, яка відображає дані таблиці, для якої вона створена. У нижній частині форми відображається панель навігації за таблицею, що містить кнопки, за допомогою яких можна переміщатися по таблиці.

Дизайн

Оскільки форми є засобами, за допомогою яких користувачі спілкуються з базою даних, то до них висуваються підвищені вимоги щодо дизайну: однотипні елементи форми, як правило, мають однакову висоту, так само вирівнюються, розташовуються на однаковій відстані і т. ін. Доступ до таких дій надають інструменти вкладки "*Упорядкувати*" ("Arrange"). Якщо, наприклад, необхідно розташувати елементи керування у формі на однаковій відстані, то необхідно виконати такі дії:

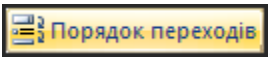
1. Виділити у формі потрібні елементи.  Виділення кількох об'єктів у формі відбувається шляхом натискання на них при одночасному утриманні натиснутою функціональної клавіші **<Shift>**.

2. На стрічці команд натиснути кнопку  "Створити інтервали по вертикалі рівними" ("Make Vertical Spacing Equal").

Розташування елементів у формі здійснюється шляхом звичайного перетягування їх у потрібне місце.

 Істотну допомогу у розробці дизайну форми надає *допоміжна сітка*. За її відсутності для її відображення на стрічці слід натиснути кнопку **«Сітка»**  .

Введення даних у поля форми відбувається по елементах керування зліва направо і зверху вниз. Проте під час проектування складних форм елементи часто міняються місцями, тоді дуже легко переплутати їх послідовність і створити тим самим незручний порядок введення даних. Фізично послідовність переходу – це порядок переходу до іншого поля після закінчення роботи з попереднім. Вона легко перевіряється за допомогою функціональної клавіші **<Tab>**. Якщо при послідовному натисканні цієї клавіші фокус введення "стрибає" по формі, то послідовність переходу нераціональна і її потрібно змінити. Для керування послідовністю переходу служить діалогове вікно "*Порядок переходу*", доступ до якого відкривається натисканням

на стрічці кнопки 

. У ньому відображається список елементів керування форми. Порядок елементів у списку відповідає поточному порядку переходу. Зміна порядку переходу виконується шляхом перетягування елемента керування на потрібне місце.

Після розроблення макета форми її слід закрити і зберегти з потрібним ім'ям. Після відкривання форми з нею можна працювати, тобто переглядати або редагувати дані таблиці, для якої вона була створена.

Обчислення підсумкового значення

У формах можна розраховувати підсумкові значення. Для цього використовується вбудована функція ***Сума*** (***Sum***), яка має синтаксис:

=Сума(аргумент)

Вона (як, до речі, й інші вбудовані функції системи) складається з трьох елементів:

1. Знак "=" вказує на те, що наступний вираз буде обчислюватися.
2. Сума – ім'я функції, за яким потрібно до неї звертатися.
3. Аргумент: ім'я поля або вираз із арифметичними операціями над полями таблиці.

Алгоритм застосування функції складається з наступних дій:

1. Перейти в *область нижнього колонтитула* форми.  Якщо вона відсутня, слід викликати в області форми контекстне меню та вибрати пункт "Заголовок/примітка форми".
2. У цій області створити керуючий елемент "поле", який не прив'язується до жодного поля таблиці, тобто має формат "Без прив'язки".
3. Звернутися до властивостей поля.
4. Перейти на вкладку "*Дані*". Звернутися до властивості "Джерело елемента керування".
5. Ввести функцію ***Сума*** (вона знаходиться у групі Функції –

Вбудовані функції – Агрегатна функція).

Вираз для функції ***Сума***, наприклад, може мати вигляд:
=Сума([Кількість]*[Ціна])

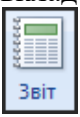
, де “Кількість“ і “Ціна” – найменування полів.

 Взагалі під час використання вбудованих функцій є можливість звернутися до даних будь-якої форми. Але цей процес значно складніший. Форми, до даних яких звертаються з основної форми, мають назву *підпорядковані*.

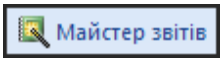
Наприклад:
[Ім’я підпорядкованої форми].[Форма]![Ім’я поля підпорядкованої форми]

Робота зі звітами

Термін “Звіт” говорить сам за себе. Звітами є документи, що призначені, у першу чергу, для виведення на друк. Це може бути екзаменаційна або платіжна відомості, *price-list* продукції і т. ін. Як правило, звіт будується на базі запиту, хоча це не є обов’язковим правилом.

Звіти, як і інші об’єкти, можна створити у “ручному” режимі за допомогою *Конструктора*, або в автоматичному – за допомогою *Майстра*. Для побудови звіту за допомогою Майстра слід на стрічці вкладок перейти на вкладку “*Створити*” і натиснути кнопку 

(“Report”). Таким чином, створюється так званий *базовий* звіт, до якого автоматично включаються *всі* поля. Більш гнучким варіантом створення звіту є використання майстра звітів, за допомогою якого, зокрема, можна включити до звіту окремі поля, він передбачає виконання таких дій:

1. На стрічці команд натиснути кнопку 

(“Report Wizard”). З’явиться вікно майстра звітів.

2. Вибрати зі списку “Доступні поля” (“Available Fields”) поля таблиці, які слід включити до звіту. Принципи відбору полів

розглянуті раніше для форм. Натиснути кнопку **«Далі»** (“Next”).

3. Якщо було вибрано *кілька* полів, то з’явиться вікно, в якому буде запропоновано виконати *групування* даних, тобто об’єднання їх за певним полем або полями. Групувати можна за кількома полями. Натиснути кнопку **«Далі»**.

4. На наступному етапі роботи майстер дає змогу визначити поля, за якими буде здійснено упорядкування (сортування) інформації.  Але ще важливішим є те, що якщо на попередньому кроці була застосована дія групування, то у цьому вікні з’являється кнопка **«Параметри зведення»** (“Summary Options …”).

5. Натискання цієї кнопки призводить до появи вікна, що відображає усі поля з *числовим* типом даних. За допомогою нескладних установок можна вибрати ті з них, за якими потрібно *обчислити* сумарне, середнє, мінімальне або максимальне значення, тобто те, що називають груповими *підсумковими* значеннями. У звіті для кожної групи буде відбуватися обчислення вибраних значень.  Якщо групування даних буде здійснюватися за кількома полями, то розрахунок групових підсумкових значень відбуватиметься у порядку розташування цих полів у списку.

6. На наступних етапах роботи майстра послідовно вибираються макет, стиль, орієнтація звіту, найменування звіту тощо.

7. Після закінчення роботи майстра звіт з вибраним найменуванням з’явиться на панелі об’єктів серед об’єктів таблиці, для якої він був побудований.

Таким чином, навіть найпростіший варіант побудови звіту – за допомогою майстра – надає можливість упорядкувати і згрупувати дані, створити проміжні і загальні підсумки. Подальша робота “шліфування” звіту здійснюється у ручному режимі. Перехід до роботи зі звітом у ручному режимі здійснюється

натисканням на стрічці команд кнопки 

(“Report Design”).

Структурно звіти складаються із двох елементів: розділів (секцій) і елементів керування. Розділами звіту є його загальний заголовок, верхній і нижній колонтитули, заголовки і примітки груп під час застосування дії групування, область даних. Розташування елемента в розділі визначає його місце у звіті і те, як часто він буде виводитися на друк. Основним елементом керування звітом є текстові поля, за допомогою яких виводиться інформація з бази даних. Важливе значення для з’ясування інформації того чи іншого показника звіту має елемент керування “текст”. У створенні звіту за допомогою майстра підписи (текст) створюються автоматично з імені поля і не завжди бувають зрозумілими. Тому іноді доводиться змінювати зміст деяких підписів у ручному режимі. Під час таких змін може з’ясуватися, що розмір (висота) секції не дає змогу розмістити в ній потрібну інформацію і її потрібно збільшити. Для цього слід встановити курсор на межу секції так, щоб він набув вигляду  , натиснути ліву кнопку миші і звичайним розтягуванням змінити висоту секції.

Звіти є кінцевим результатом роботи з базою даних. Зі звітами відбувається постійна робота користувачів, при цьому надалі вони можуть передаватися в інші організації, наприклад, різноманітні соціальні фонди, фіскальні організації і т. ін. Тому до звітів, так само як і до форм, висуваються підвищенні вимоги щодо оформлення. Дизайн звітів (вирівнювання, форматування) здійснюється за тим самим алгоритмом, що був описаний раніше у розділі “Дизайн форм”.

При налагоджуванні звіту швидко переглянути результат його виконання, *не виходячи при цьому з режиму роботи у конструкторі*, можна шляхом натискання стандартної кнопки  (“Попередній перегляд”) на панелі інструментів

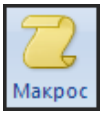
конструктора.

Робота з макросами

Функціональні можливості *Access* істотно розширюються за рахунок використання спеціальних об'єктів – макросів. ***Макроси*** – це невеличкі програми, що складаються із послідовності макрокоманд. За допомогою макросів вирішуються питання, які не можуть бути вирішені шляхом "візуального" програмування. У першу чергу, макроси призначені для створення зручного інтерфейсу бази даних. Наприклад, застосування макросів дає змогу під час завантаження *Access* відобразити не вікно "*База даних*", а якусь іншу форму; розташувати у формах кнопки, за допомогою яких зручно і просто можна звернутися до інших форм і звітів, і т. ін. Разом із тим на відміну від використання ще одного типу об'єктів *Access* – модулів, які призначені для вирішення схожих завдань, використання макросів не вимагає від користувача знання мов програмування.

Створення макросів

Для створення макросів програма має спеціальний засіб – "Конструктор макросів". За його допомогою створення макросу відбувається таким чином.

На стрічці вкладок вибирається вкладка "*Створити*". На стрічці команд натиснути кнопку 

("Macro"). Відкриється вікно-бланк для створення нового макросу. Це вікно нагадує вікно "*Конструктора таблиць*". Воно також складається із двох частин: у верхній частині знаходиться *панель опису*, у нижній – *панель аргументів*. Панель опису призначена для визначення послідовності макрокоманд, з яких буде складатися макрос. Вибір макрокоманд здійснюється в однойменному полі зі списком. Праворуч від цього поля знаходиться поле "Примітка", в якому вводиться довідкова інформація, наприклад, стислий опис дії, яку

буде виконувати ця команда.

Після вибору макрокоманди у нижній частині вікна з'являється перелік її аргументів, де користувач заповнює їх потрібними значеннями. Слід зазначити, що деякі макрокоманди взагалі не мають аргументів.

Є й інший варіант створення макросів:

1. Відкрити вікно конструктора макросів.
2. Перетягнути мишею з панелі навігації потрібний об'єкт на панель опису. Це автоматично ініціює створення макрокоманди, яка відкриває цей об'єкт. Наприклад, для звернення до форми таким макросом буде "ВідкритиФорму".

Після побудови макросу він зберігається так само, як і інші об'єкти.

Якщо кількість макросів у межах одного об'єкта досить велика, доцільно здійснювати групування макросів, наприклад, за їх функціональним призначенням. Для цього у вікні конструктора макросів слід натиснути на панелі інструментів кнопку  ("Імена макросів"), після чого у панелі опису з'явиться стовпчик з ім'ям "Ім'я макросу". Ім'я макросу можна задати тільки для першої з послідовності макрокоманд. Усі інші макрокоманди будуть належати до послідовності макросів із цим іменем, доки не зустрінеться наступне ім'я.

Макрокоманди у макросах виконуються послідовно, відповідно до порядку їх запису у бланку. Але цей порядок можна змінити шляхом задання умови, яка змінює послідовний порядок виконання макрокоманд. Умова задається у бланку конструктора макросів у стовпчику "Условие". За його відсутності на панелі інструментів слід натиснути кнопку  ("Умови"). У виконанні макросу, якщо умова виконується, то виконується і макрокоманда.

Для редагування макросу слід викликати контекстне меню на його назві на панелі об'єктів і вибрати з нього пункт "Конструктор" ("Design View").

Виконання макросів

У режимі конструктора макросів для цього на стрічці команд слід натиснути кнопку 

(“Run”). Цей варіант, як правило, застосовується під час налагодження макросу. Вже створений макрос можна завантажити, викликавши контекстне меню на його назві на панелі об’єктів і вибравши з нього пункт “Запуск”. Універсальний варіант виконання макросів здійснюється шляхом послідовного вибору зі стрічки меню команди “Знаряддя бази даних” (“Database Tools”), а потім зі стрічці команд – команди  (“Запустити макрос”). За останнім варіантом з’являється вікно, з якого вибирається потрібний макрос.

Усі макроси розміщуються на панелі об’єктів у групі “Непов’язані об’єкти”, звідки їх можна швидко виконати.

Призначення макросу для події

Найчастіше макроси застосовуються для оброблення певної події. *Подія* виникає у результаті дій користувача (натискання функціональної клавіші, кнопки миші тощо), може бути ініційована самою системою або викликана виконанням макросів. Наприклад, подією може бути введення даних у поле форми, виведення на екран форми. Подіям можна призначити макрос, який буде виконуватись у *відповідь* на цю подію. Оскільки таких подій існує досить багато, то завдання розробника бази даних часто полягає саме у створенні макросів або програм мовою *VBA* (*Visual Basic for Application*), які відповідним чином реагують на ту чи іншу подію. Саме так здійснюється в *Access* програмування. Доступ до подій об’єкта (форми, поля форми і т. ін.) відбувається так:

1. Виділити потрібний об’єкт.
2. У вікні “*Аркуш властивостей*” перейти на вкладку “*Подія*”.

Як приклад розглянемо призначення макросу для оброблення події натискання мишею кнопки.

1. Відкрити форму в режимі конструктора.

2. Звернутися до властивостей кнопки. Відкрити вкладку "*Подія*" вікна властивостей кнопки і знайти подію "Після клацання".

3. Натиснути на рядку з назвою події, що ініціює появу прихованої кнопки списку. Розкрити список, що містить назви створених макросів.

4. Вибрати потрібний макрос зі списку, після чого його ім'я стане значенням для події.

Але потрібний макрос можна створити і під час виконання викладеної вище процедури. У цьому разі взагалі не слід відкривати список для події,  тобто її значення повинно бути незаповненим (якщо значення вже було вибрано зі списку, то його слід вилучити звичайним редагуванням тексту). Подальші дії такі:

1. Натиснути на кнопку вибору  , після чого з'явиться вікно "*Вибір побудовника*".

2. Вибрати у ньому пункт "Побудовник макросів" і натиснути **«OK»**. З'явиться бланк для побудови макросу і вікно для введення назви створюваного макросу.

3. Створити макрос.

4. Після закривання бланка для побудови макросу його назва з'явиться у значенні події "Після клацання".

Лабораторні роботи

Комплекс 1

Загальні вказівки до виконання

Лабораторні роботи є єдиним комплексом взаємопов'язаних між собою завдань. Виконання кожного наступного завдання

базується на роботі, проведеній у попередніх лабораторних роботах.

Лабораторна робота №1

Тема. Створення базових таблиць

Мета. Навчитися створювати базові таблиці бази даних в СУБД *Access*, встановлювати типи полів, призначати ключові поля.

Вказівки до виконання:

1. Під час проектування баз даних широко використовуються *системи класифікації* та *кодування*. В їх основі – заміна *назви об'єкта* на його *код*. У принципі коди взагалі можна не використовувати, але тоді виникає ряд проблем. По-перше, значно збільшується обсяг інформації, що вводиться в базу даних, бо назви можуть бути задовгі, наприклад, резистор *С-23-33Н-0, 125-1300 Ом*. По-друге, одну й ту ж назву кожна людина може ввести по-різному. Це може привести до того, що база даних буде містити неправильну інформацію. А кодування не лише вдосконалює організацію інформаційного забезпечення, але і значно підвищує ефективність використання машинних ресурсів. Більше того, коди досить часто дають вичерпну характеристику про об'єкт, якщо знати принципи його побудови.

2. Якщо назва грошової одиниці для поля з типом даних "Грошова одиниця" буде "Рубли", то потрібно змінити її на "Гривні". Для цього слід виконати такі дії:

- Звернутися до засобу "*Панель управления*" і вибрати пункт "Язык и региональные стандарты". З'явиться вікно "*Язык и региональные стандарты*".
- На вкладці "*Региональные параметры*" вибрати зі списків елементи "Украинский" і "Украина".
- Якщо в полі "Сумма денег" поруч із числовим значенням не буде відображатися грошова одиниця "грн.", то слід натиснути кнопку **«Настройка»**. Відкриється вікно "*Настройка*

региональных параметров”.

- Перейти на вкладку “*Денежная единица*” та вибрати зі списку “Обозначение денежной единицы” елемент “грн.”.
- Але й цього може бути недостатньо. В такому разі слід відкрити таблицю в режимі “Конструктор” і для поля з типом даних “Грошова одиниця” вибрати потрібний формат для властивості “Формат поля”.

Завдання

1. Створіть папку Doc*Access* у своїй папці.
2. Завантажте *Access*.
3. Створіть базу даних з довільним ім’ям. Розташуйте її у створеній папці Doc*Access*.
4. Створіть базові таблиці в режимі “Конструктор” за наступною структурою:

- Таблиця “Список студентів”

Ім’я поля	Тип поля
№ залікової книжки	Текстовий
Прізвище	Текстовий
Спеціальність	Список, що містить спеціальності закладу. Створіть за допомогою “Майстера підстановок”
Курс	Числовий
Група	Текстовий
Дата народження	Дата й час
Стать	Так/Ні (Логічний)
Стипендія	Грошова одиниця
Автобіографія	Примітка

Призначте ключовим полем “№ залікової книжки”

- Таблиця “Класифікатор предметів”

Ім’я поля	Тип поля
Код предмета	Текстовий

Назва предмета	Текстовий

Призначте ключовим полем “Код предмета”

- Таблиця “Успішність студентів”

Ім’я поля	Тип поля
№ залікової книжки	Текстовий
Код предмета	Текстовий
Оцінка	Текстовий

5. У таблиці “Успішність студентів” для поля “Оцінка” змініть тип даних на числовий.

6. Закінчіть роботу з *Access*.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке “база даних” і чим вона відрізняється від електронної таблиці?
2. Які є моделі баз даних?
3. Що таке “реляційна модель”?
4. Що таке процес нормалізації бази даних?
5. За допомогою яких програм створюється база даних?
6. Яке призначення і функції СУБД?
7. З яким типом моделей баз даних працює *Access*?
8. З яких об’єктів складається файл бази даних?
9. Що таке запис?
10. Які є способи створення таблиці в базі даних?
11. Що таке структура таблиці бази даних?
12. Які об’єкти може містити файл бази даних?
13. Що таке поле?
14. Які бувають типи полів?
15. Яке призначення поля типу Примітка?
16. Що таке ключове (індексне) поле? Яке його призначення?

17. Чи обов’язкова наявність ключового поля у таблиці?
18. Як створити структуру бази даних?
19. Суть побудови таблиці в режимі конструктора?
20. Які ще є режими побудови таблиць?
21. Чому таблиці, як правило, будуються в режимі конструктора?
22. Суть модифікації структури таблиці.

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити базу даних.
2. Відкрити базу даних.
3. Створити таблицю бази даних.
4. Застосувати для поля потрібний тип даних.
5. Створити індексне (ключове) поле.
6. Змінити тип даних для побудованої раніше таблиці.

Лабораторна робота №2–3

Тема. Створення міжтабличних зв’язків. Введення даних. Робота з властивостями полів

Мета. Навчитися створювати зв’язки між таблицями, вводити дані, змінювати властивості полів.

Вказівки до виконання:

1. Якщо до схеми даних помилково двічі додана одна й та сама таблиця, слід її виділити і натиснути функціональну клавішу **«Delete»**.

2. У разі зменшення розміру поля дані, значення яких більше за новий розмір, що встановлюється, можуть бути загублені. Тому при появі повідомлення “Можливо втрачено деякі дані” переконайтеся, що цього не відбудеться, і тільки після цього натисніть кнопку **«Так»**.

3. Поле “Код предмета” заповнюються зростаючою числовою послідовністю.

4. Значення англомовних термінів, які використовуються в програмі: “Null” – порожнє значення (дані в полі відсутні); “True“ – істинне; “False” – хибне.

5. Введення даних у *пов’язану* таблицю зручно здійснювати з головної таблиці. Для цього після відкривання, наприклад, таблиці “Список студентів”, яка є головною для таблиці “Успішність студентів”, слід натиснути на піктограму  , що знаходиться у першому стовпчику. Після цього під рядком з’являються (“розгортаються”) записи пов’язаної таблиці, які відповідають *вибраному рядку* головної таблиці. Ще одна суттєва перевага такого способу введення полягає в тому, що під час додавання нових записів *автоматично* заповнюється поле, за яким здійснюється зв’язок із головною таблицею. Для “закриття” списку записів пов’язаної таблиці слід натиснути кнопку  .

Завдання

1. Модифікуйте (змінити структуру) таблицю “Список студентів”:

- для поля “№ залікової книжки” для властивості “Розмір поля” встановити значення “10” (символів), оскільки навряд чи номер буде мати більшу кількість символів;
- для поля “Курс” задайте такі властивості: “Розмір поля” – “Ціле”, “Кількість знаків після коми” – “0”.

2. Модифікуйте таблицю “Класифікатор предметів”. Для поля “Код предмета” для властивості “Розмір поля” встановити значення “3”.

3. Відповідно до здійснених змін модифікувати відповідні поля таблиці “Успішність студентів”.

4. Здійсніть аналіз на розмір поля для інших полів таблиць і за необхідністю змініть їх розміри.

5. Створіть зв’язки між таблицями із забезпеченням цілісності даних, каскадним оновленням і видаленням пов’язаних полів:

- між таблицями “Успішність студентів” та “Список студентів” за полем “№ залікової книжки”;

- між таблицями “Успішність студентів” та “Класифікатор предметів” за полем “Код предмета”.

6. Скасуйте зв’язок між таблицями “Успішність студентів” та “Список студентів” та знову поновить його.

7. Відкрийте таблицю “Список студентів” у режимі конструктора. Ознайомтеся з властивостями текстового, числового і грошового типів даних та даних типу “Дата й час”.

8. Для таблиці “Список студентів” виконайте такі дії:

- введіть один запис;
- спробуйте ввести ще один запис із тим самим значенням для ключового поля “№ залікової книжки”, що було надано першому запису. Що відбудеться?
- встановіть для полів “Прізвище” і “Номер групи” для властивості “Обов’язково” значення “Так”;
- спробуйте ввести запис без заповнення даними цих полів. Що відбудеться?
- встановіть для поля “Номер групи” властивість “Значення за промовчуванням” – номер вашої групи;
- введіть запис. Як змінився характер даних при введенні?
- введіть ще записи, так щоб загальна кількість студентів дорівнювала шести. Передбачте, щоб таблиця містила записи про студентів як мінімум двох спеціальностей, двох груп і двох курсів.

9. Введіть у таблицю “Класифікатор предметів” п’ять записів (для п’яті дисциплін).

10. Відкрийте таблицю “Успішність студентів”. Виконайте для неї наступні дії:

- ознайомтеся з властивостями даних числового типу;
- для поля “Оцінка” встановіть такі властивості:

  10.1. “Розмір поля” – “Ціле”, “Кількість знаків після коми” – “0”.

  10.2. Властивості “Обов’язково” надайте значення “Так”.

- Введіть *одну* оцінку для першого студента. Введення даних зручно здійснювати з таблиці “Список студентів” (див. вказівки

до виконання).

- Для поля “Оцінка” встановіть такі властивості:

10.1. для властивості “Правило перевірки” визначте правило, згідно якому оцінка має знаходиться в межах від “2” до “5” (для створення інтервалу використовуйте логічний оператор “And”).

10.2. властивість “Текст первірки” заповніть текстом на зразок “Значення оцінки має бути в інтервалі від 2 до 5”.

- Введіть запис, у якому поле “Оцінка” не буде відповідати застосованому правилу перевірки, наприклад, дорівнювати “6”. Переконайтеся у появі відповідного повідомлення.

- Для кожного студента введіть не менш ніж три оцінки з дисциплін.

11. Відкрийте таблицю “Список студентів”. Вилучить запис для останнього, 6-го студента. Перегляньте таблицю “Успішність студентів”. З неї повинні бути вилучені усі дані, що стосуються 6-го студента.

12. Закінчить роботу з *Access*.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Як ввести дані у базу даних?
2. Яке призначення індексованого (ключового) поля?
3. Навіщо потрібні зв’язки між таблицями?
4. Яке призначення “Майстра підстановок”?
5. Як створити зв’язок між таблицями?
6. Як скасувати зв’язок між таблицями?
7. Визначення поняття “властивості поля”?
8. Яке призначення властивості “Розмір поля”?
9. Яке призначення властивості “Правило перевірки”?
10. Яке призначення властивості “Текст перевірки”?
11. Для чого призначена властивість “Значення за

промовчанням”?

12. Яке призначення властивості “Обов’язково”?

13. Як організувати введення даних у поле однієї таблиці з іншої таблиці?

14. Чому введення даних до пов’язаної таблиці зручно здійснювати з головної?

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити і скасувати зв’язок між таблицями.
2. Змінити властивості полів таблиці.
3. За допомогою “Майстра підстановок” організувати введення даних у поле з іншої таблиці.
4. Ввести, змінити, вилучити дані з таблиць.

Лабораторна робота №4

Тема. Побудова запитів

Мета. Засвоїти призначення запитів; навчитися створювати запити за параметром і на вибірку; формувати обчислювальні поля.

Вказівки до виконання:

1. Програма дає змогу створювати копії будь-яких об’єктів бази даних. Для цього виконуються стандартні дії копіювання та вставлення об’єкта. Так, для копіювання об’єкта його слід виділити і, наприклад, натиснути кнопку копіювання на стрічці команд. Відповідно для вставлення можна, наприклад, на стрічці команд натиснути кнопку вставлення.

2. Для надання значень логічним полям використовуйте такі: “Істина” (можна також застосувати англомовний аналог “True”); “Хибність” або “False” – англійською. Надавати значення можна обома мовами, але значення, що вводиться англійською мовою, автоматично перетворюються на українську.

3. Розрахункове поле будується за допомогою “Побудовника

виразів". Для звернення до нього слід встановити курсор у порожню клітинку потрібного стовпчика бланка запиту, викликати контекстне меню й вибрати в ньому пункт "Побудувати…".

Завдання

1. Створіть бланк запиту в режимі "Конструктор" за наступним зразком:

Прізвище	**Назва предмета**	**Оцінка**	**Група**	**Стать**

2. Запам'ятати запит з ім'ям "Запит на вибірку-1". Як після збереження змінилася панель об'єктів?

3. Виконайте запит.

4. Створити із запиту "Запит на вибірку-1" копію з ім'ям "Запит на вибірку-2".

5. Сформувати в ньому умову, що дозволить відібрати студентів жіночої статі.

6. Створити із запиту "Запит на вибірку-1" копію з ім'ям "Запит на вибірку-3".

7. Упорядкуйте в ньому записи в такому порядку: за групою, прізвищем і назвою предмета. Зверніть увагу, що для цього доведеться змінити порядок розташування полів таблиць.

8. Створіть із запиту "Запит на вибірку-1" копію з ім'ям "Запит з параметром".

9. Встановіть в ньому параметр, за яким можна відібрати студентів за будь-якою оцінкою.

10. Створіть у режимі "Конструктор" запит, до якого включіть поля "Група", "Прізвище", "Оцінка". Впорядкуйте інформацію за групою і прізвищем.

11. Запам'ятайте створений запит з ім'ям "Підсумковий запит" і виконайте його.

12. Для поля "Оцінка" розрахуйте середній бал, для чого у рядку "Підсумок" виберіть значення "Середнє".

13. Виконайте запит. Як змінилася форма подання інформації

у запиті?

14. Створіть запит у режимі “Конструктор”, включивши до нього поля “Код предмета” та “Назва предмета”.

15. Упорядкуйте інформацію за назвою.

16. Додайте до запиту поле “Оцінка”. Розрахуйте для нього:

- загальну суму (оцінок);
- загальну кількість (оцінок);
- середню оцінку.

17. Запам’ятайте запит з ім’ям “Середній бал за групами і оцінками”.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке запит?
2. Які є типи запитів?
3. Що таке “простий запит на вибірку”?
4. Яке призначення конструктора запитів?
5. Яке призначення бланка запиту в режимі “Конструктор”?
6. Що таке “запит з параметром”?
7. Що таке “групова операція”?
8. Як додати до бланка запиту рядок “групова операція”, якщо він відсутній?
9. Найпростіші функції, що можуть бути застосовані в групових операціях.

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити запит у режимі “Конструктор”.
2. Додати або вилучити таблицю до запиту в конструкторі запитів.
3. Додати поля з таблиці до запиту в конструкторі запитів.
4. Вилучити поля з запиту.
5. Упорядкувати інформацію запиту в конструкторі запитів.

6. Створити умову для відбору інформації з таблиці.
7. Застосувати параметр для відбору інформації з таблиці.
8. Модифікувати запит.
9. Застосувати стандартні функції в групових операціях.

Лабораторна робота №5

Тема. Побудова форм

Мета. Навчитися створювати форми засобами СУБД *Access*.

Вказівки до виконання:

1. Інструменти, за допомогою яких відбувається робота над дизайном форми (вирівнювання, розташування полів, керуючих елементів і та ін.), містить вкладка "*Упорядкувати*". Наприклад, вирівнювання тексту підписів здійснюється за допомогою інструмента  "Вирівняти зліва" або властивості "Ліворуч" (відстань у см від лівого краю форми).

2. Розміри елемента керування можна довільно змінювати. Для цього потрібно його виділити, встановити курсор на один із маркерів, що оточують елемент по периметру, і перетягнути його в інше місце.

3. Виділення кількох об'єктів у формі проводиться шляхом натискання на них при одночасному утриманні натиснутою функціональної клавіші **<Shift>**.

4. *Майстер кнопок* повинен автоматично з'являтися при перенесенні (перетягуванні) на форму елемента "Кнопка". Якщо цього не відбувається, на вкладці "*Конструктор*" натисніть ("утопіть") на панелі елементів кнопку  ("Застосувати майстри елементів керування").

5. Під час роботи з майстром кнопок для вибору дій операцій із записами вам буде потрібна категорія "Перехід між записами".

6. Для підрахунку кількості рядків у таблиці застосовується функція ***Кількість*** (***COUNT***), аргументом якої є ім'я поля; для розрахунку середньої величини – функція ***Середнє***.

7. Для відображення у формі області приміток слід викликати на ній контекстне меню та вибрати з нього пункт “Заголовок/примітка форми”.

8. Для надання імені будь-якому об’єкту форми застосовуйте властивість “Підпис”.

9. Майстер форм знаходиться у групі “Додаткові форми”.

Завдання

1. Створіть для таблиці “Список студентів” за допомогою засобу “Автоформа” три варіанти форми: для введення одного запису, для відображення кількох елементів у табличному вигляді та “розділену форму”.

2. Створіть для таблиці “Успішність студентів” за допомогою засобу “Автоформа” форму для відображення кількох елементів.

3. У формі для таблиці “Успішність студентів”у полі приміток створіть три поля, в одному з яких виведіть текст “Підсумкові дані:”, інші два – це обчислювальні поля, в яких слід розрахувати загальну кількість оцінок і середній бал з оцінок.

4. У приєднаних до обчислювальних полів написах введіть текст “Всього оцінок” і “Середній бал”.

5. Застосуйте до текстового поля “Підсумкові дані:” такі елементи форматування “Товщина шрифта” – жирний шрифт і курсивний шрифт.

6. Додайте до форми в області “Подробиці” поля “Прізвище” і “Назва предмета” з відповідних таблиць.

7. За допомогою “Конструктора” створіть форму для таблиці “Список студентів”. Надайте формі ім’я “Список студентів (конструктор)”. До форми включати всі поля. Звернути особливу увагу на дизайн форми (текст і поля повинні бути вирівняні, розташовані на однаковій відстані і т. ін.).

8. Виконати такі дії з дизайну форми:

- встановіть однакову ширину для приєднаних написів, вирівняйте їх по лівому краю форми, а текст в написах – по правому краю;
- розташуйте на однаковій відстані по вертикалі приєднані

написи і поля;

- встановіть однакову висоту для приєднаних написів і полів.

9. За допомогою *Майстра кнопок* створіть у формі кнопки для навігації таблицею, за допомогою яких можна буде переходити від запису до запису в обох напрямках, до останнього і до першого запису.

10. Створіть копію форми “Список студентів (конструктор)”.

11. Розташуйте у формі кнопку пошуку, за допомогою якої можна знайти будь-якого студента за номером залікової книжки.

12. За допомогою майстра форм створіть форму, яка буде містити всі відомості для студента з таблиці “Список студентів”, а також його оцінки і назви предметів, за якими вони одержані.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке форма? Яке її призначення?
2. З яких елементів складається форма?
3. Які є способи створення форми?
4. Які види форм можна створити в автоматичному режимі? Що вони собою являють?
5. Які види форм можна створити за допомогою майстра форм?
6. Яке призначення конструктора форм?
7. Що таке елемент керування? Які є елементи керування?
8. Чи має перевагу спосіб введення даних до таблиць за допомогою форм над введенням за допомогою таблиць, а якщо це дійсно так, то в чому визначаються такі переваги?
9. Чи обов’язково під час створення форми включати до неї всі поля таблиці?
10. Чи можна у разі створення форми за допомогою майстра включати до неї тільки окремі поля таблиці, а якщо це так, то як це можна зробити на практиці?

11. Чи можна додати до форми поля з інших таблиць?
12. Чи можна за допомогою форми додати до таблиці новий запис?
13. Що спільного між таблицями і формами?
14. Для чого у форму вставляють елементи керування?
15. Як у формі створити текстове поле?
16. Як у формі створити обчислювальне поле?
17. Яке призначення “Побудовника виразів”?
18. Як вирівняти елементи форми?
19. Як розташувати елементи форми на однаковій відстані?
20. Як змінити відстань між елементами форми?
21. Що таке “функція” і як застосувати функцію у формі?

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити форму в автоматичному режимі.
2. Додати до форми наявні поля.
3. Створити у формі текстове поле.
4. Застосувати “Побудовник виразів”.
5. Створити у формі обчислювальні поля.
6. Створити форму в режимі “Конструктор”.
7. Вирівняти елементи форми по лівому (правому) краю форми.
8. Розташувати елементи форми на однаковій відстані. Змінити цю відстань одночасно для усіх елементів.
9. Створити елемент керування “Кнопка”.
10. Надати користувачеві можливість відбору інформації за ключовим полем.
11. Застосувати майстра форм.

Лабораторна робота №6

Тема. Побудова звітів

Мета. Навчитися створювати звіти засобами СУБД *Access*.

Вказівки до виконання:

1. Для редагування текстового елемента звіту слід натиснути на ньому мишею, після чого він оточується по периметру рамкою з маркерами. Надалі виконується звичайне редагування тексту у рамці.

2. Розміри текстового поля можна довільно змінювати. Для цього слід його виділити, встановити курсор на один із маркерів, що оточують поле по периметру, і перетягнути його в інше місце.

3. Якщо в полі відображається значення на зразок "########", то це є ознакою того, що значення у полі не поміщається і розміри цього поля потрібно збільшити.

4. Розташування полів звіту в межах однієї сторінки можна досягти за рахунок зміни місця розташування полів, зменшення їх ширини, збільшення висоти. Наприклад, назви заголовків, що складаються з двох слів ("Назва предмета"), в області верхнього колонтитулу можна розташувати в два рядки.

5. Для зміни розміру секції (даних, колонтитулу і т. ін.) слід встановити курсор на межу між секціями так, щоб він набув вигляду  , і звичайним перетягуванням перемістити її в іншу позицію.

6.  Якщо при побудові звіту відображаються не всі поля або не вся інформація, то це є ознакою того, що інформація більше за розміри (ширину) сторінки. Розв'язується ця проблема двома шляхами:

- зміною властивості "Розмір поля" для полів таблиці шляхом його зменшення;
- зміною розмірів полів звіту шляхом його збільшення.

Завдання

1. Створіть запит, до якого включить поля "Спеціальність", "Курс", "Група", "Стипендія".

2. Упорядкуйте інформацію запиту за полями "Спеціальність", "Курс" і "Група".

3. За допомогою Майстра звітів створіть звіт з назвою

“Звіт 1” для цього запиту. Під час створення запиту врахуйте такі вимоги:

- згрупуйте інформацію за полями “Спеціальність”, “Курс” і “Група”;
- обчисліть суму й середнє значення за полем “Стипендія”.

4. Створіть запит, до якого включить поля “Спеціальність”, “Курс”, “Група”, “Прізвище”, “Назва предмета”, “Оцінка”.

5. Упорядкуйте інформацію запиту за полями “Спеціальність”, “Курс”, “Група”, “Прізвище”.

6. За допомогою Майстра звітів створіть звіт з назвою “Звіт 2” для цього запиту. Під час створення запиту врахуйте такі вимоги:

- згрупуйте інформацію за полями “Спеціальність”, “Курс”, “Група”, “Прізвище”;
- упорядкуйте інформацію за полем “Назва предмета”;
- обчисліть середній бал за полем “Оцінка”, а також визначте мінімальне та максимальне значення оцінки;
- застосуйте альбомну форму орієнтації звіту.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке звіт? Яке його призначення?
2. Для чого призначений у звіті “рівень групування”?
3. В яких режимах можна створити звіт?
4. Чи можна у звіті замінити текстову інформацію?

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити звіт у режимі майстра звітів.
2. Змінити набір полів (додати нові або вилучити існуючі).
3. Створити рівні групування.
4. Упорядкувати інформацію за потрібними полями.
5. Обчислити підсумкові значення.

6. Змінити орієнтацію звіту.
7. Редагувати створений звіт у режимі “Конструктор”.

Лабораторна робота №7

Тема. Створення макросів

Мета. Навчитися створювати найпростіші макроси.

Вказівки до виконання:

1. У створенні макросу “Макрос-повідомлення” використовуються макрокоманди “Повідомлення” (MsgBox) і “Відкритиформу” (OpenForm).

2. У створенні макросу “Успішність” врахуйте наступне:

- Для відкривання запиту використовується макрокоманда “Відкритизапит” (OpenQuery).
- Для виведення складу продукції, що відображається у формі, застосовується макрокоманда “Застосуватифільтр” (ApplyFilter). Як аргумент у нашому випадку слід вибрати “Умова відбору” (Where Condition). Його значення сформуйте за допомогою *побудовника виразів*. Це значення у вигляді виразу повинно пов’язати між собою однойменні поля “№ залікової книжки” у формі “Список студентів” і запиті “Запит-успішність”.

3. Під час створення кнопки “Успішність” визначають дію, що буде виконуватися при натисканні кнопки. Така дія є макросом, що відкриває запит “Запит-успішність”. Тому для властивості кнопки “Після клацання” (On Click) на вкладці “Подія” (Event) слід розкрити список і вибрати з переліку потрібний макрос.

4. Під час редагування властивостей об’єкта слід розрізняти призначення властивостей “Підпис” (Caption) на вкладці “Формат” (Format) та “Ім’я” (Name) на вкладці “Інші” (Other). Перша властивість визначає найменування (підпис), що бачить користувач на екрані, друга властивість визначає ім’я, за яким відбувається звернення до об’єкта.

Завдання

1. Створіть макрос з ім’ям “Макрос-повідомлення”, за допомогою якого буде відкриватися форма “Список студентів” (конструктор). Передбачте таке:

- відображенню підлягають студенти тільки для одного курсу, наприклад, другого;
- за допомогою макрокоманди “Повідомлення” сформувати повідомлення, що форма відображає дані про студентів певного курсу в режимі перегляду;
- заборонити редагування даних під час їх перегляду.

2. Створіть із запиту “Запит на вибірку-1” запит з ім’ям “Запит-успішність”.

3. Модифікуйте створений звіт:

- додайте поле “№ залікової книжки”;
- введіть параметр для відбору даних для заданого номера залікової книжки.

4. Створіть макрос “Успішність” для відкриття запиту “Запит-успішність”.

5. Створіть із запиту “Запит-успішність” запит з ім’ям “Запит-успішність-1”.

6. Модифікуйте створений звіт:

- вилучіть поле “Прізвище”;
- вилучіть параметр для відбору даних для заданого номера залікової книжки.

7. Створіть з макросу “Макрос-успішність” запит з ім’ям “Макрос-успішність-1”.

- Модифікуйте макрос, замінивши в ньому все, що пов’язано із “Запит-успішність” на “Запит-успішність-1”.

8. У формі “Список студентів” (конструктор) створіть кнопку для виклику даних про успішність студента із запиту “Запит-успішність-1”, картка якого відображається на екрані. Назвіть кнопку “Успішність”. Натискання цієї кнопки повинно призводити до відкривання запиту “Запит-успішність-1”.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке макрос?
2. Яке призначення має макрос?
3. З яких елементів складається вікно конструктора макросів?
4. Як відбувається побудова макросу за допомогою конструктора макросів?
5. Що таке аргумент макросу?
6. Чи є обов'язковим наявність аргументів у макросі?
7. Як надати значення аргументу?
8. Які існують варіанти виконання макросів?
9. Що таке подія? У результаті чого виникають події?
10. Як призначити макрос для події?
11. Яка макрокоманда використовується для виведення повідомлення користувачу?
12. Які макрокоманди використовуються для відкривання форми, запиту?
13. Яке призначення має макрокоманда "ЗастосуватиФільтр"? Які вона має аргументи і як визначаються їхні значення?
14. Які аргументи і з якими значеннями застосовують для заборони зміни даних під час роботи користувача з такими об'єктами, як форми та запити?
15. Які елементи управління використовуються для створення форми з кількох сторінок?
16. Як створити групу макросів?
17. Який макрос використовується для навігації за сторінками форми? Які аргументи він має?
18. Що таке подія? У результаті чого виникають події?

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити макрос за допомогою конструктора макросів.

2. Змінити природний порядок виконання макросів у групі макросів.

3. Виконати макрос.

4. Застосувати макроси для оброблення подій.

5. Застосувати макроси для відкривання форм, запитів, виведення повідомлень користувачеві, відбору частини даних таблиці (здійснення фільтрації даних).

6. Створити макрос для оброблення події “Після клацання”.

7. Створити групу макросів.

8. Створити макрос для навігації сторінками форми.

9. Визначати розташування кнопок у формі за допомогою її властивостей.

10. Встановити потрібні розміри кнопки у формі за допомогою її властивостей.

Комплекс 2

Загальні вказівки до виконання

1. Лабораторні роботи є єдиним комплексом взаємопов’язаних між собою завдань. Виконання кожного наступного завдання базується на роботі, проведеній у попередніх лабораторних роботах.

2. “*Індивідуальний номер*” – номер у навчальному журналі групи.

3. Наявність у пункті завдання символів “***” вказує на його підвищену складність. Виконання такого пункту не є обов’язковим, але за його виконання нараховуються додаткові бали.

Постановка задачі

Підприємство є виробником продукції. Для визначення ціни продукції необхідно розрахувати її ***собівартість***, тобто врахувати всі ***витрати на її виробництво***. ***Витрати*** бувають двох видів:

1. ***Матеріальні***. Витрати, що мають предметний, речовий характер, наприклад, у разі збирання комп'ютера потрібні монітор, клавіатура, миша.

2. ***Нематеріальні***. Витрати, що безпосередньо не входять до складу продукції. Це витрати на заробітну плату, електроенергію, опалення й освітлення приміщень, їх оренду та інше.

Крім витрат на виготовлення продукції, ціна включає також ***прибуток***, який повинен мати виробник у результаті реалізації продукції.

Сутність економічних показників, що зустрічаються в лабораторних завданнях

1. ***Податок на додану вартість*** (***ПДВ***). Податок, що справляється з підприємства на суму приросту вартості на цьому підприємстві, яка розраховується як різниця між виручкою від реалізації товарів та послуг і сумою на сировину, напівфабрикати, отримані від інших виробників.

2. Ефективність діяльності підприємства визначає показник, що характеризує рівень віддачі від витрат і ступінь використання ресурсів, який називається ***рентабельність*** (англ. *Profitability* – прибутковість, доходність). Рівень рентабельності є відношенням прибутку до витрат.

3. ***Собівартість***. Сукупні витрати на виробництво продукції.

4. ***Ціна*** – це грошове вираження вартості товару, яке є формою вираження споживчої вартості одиниці товару або послуги. Компонентами основної ціни є собівартість і валовий прибуток.

Лабораторна робота №1

Тема. Створення базових таблиць

Мета. Навчитися створювати базові таблиці бази даних в СУБД *Access*, встановлювати типи полів, призначати ключові поля.

Вказівки до виконання:

1. Під час проектування баз даних широко використовуються ***системи класифікації та кодування***. В їх основі – заміна назви об'єкта на його ***код***. У принципі коди взагалі можна не використовувати, але тоді виникає ряд проблем. По-перше, значно збільшується обсяг інформації, що вводиться в базу даних, бо назви можуть бути задовгі, наприклад, резистор *С-23-33Н-0,125-1300 Ом*. По-друге, одну й ту ж назву кожна людина може ввести по-різному. Це може привести до того, що база даних буде містити неправильну інформацію. А кодування не лише вдосконалює організацію інформаційного забезпечення, але і значно підвищує ефективність використання машинних ресурсів. Більше того, коди досить часто дають вичерпну характеристику про об'єкт, якщо знати принципи його побудови.

2. Якщо назва грошової одиниці для поля з типом даних "Грошова одиниця" буде "Рубли", то потрібно змінити її на "Гривні". Для цього слід виконати такі дії:

- Звернутися до засобу "*Панель управления*" і вибрати пункт "Язык и региональные стандарты". З'явиться вікно "*Язык и региональные стандарты*".
- На вкладці "*Региональные параметры*" вибрати зі списків елементи "Украинский" і "Украина".
- Якщо в полі "Сумма денег" поруч із числовим значенням не буде відображатися грошова одиниця "грн.", то слід натиснути кнопку **«Настройка»**. Відкриється вікно "*Настройка региональных параметров*".
- Перейти на вкладку "*Денежная единица*" та вибрати зі списку "Обозначение денежной единицы" елемент "грн.".
- Але й цього може бути недостатньо. В такому разі слід відкрити таблицю в режимі "Конструктор" і для поля з типом даних "Грошова одиниця" вибрати потрібний формат для властивості "Формат поля".

Завдання

1. Створити у своїй папці папку Doc*Access*.

2. Завантажити *Access*.

3. Створити базу даних із довільним ім'ям. Розташовувати її у створеній папці Doc*Access*.

4. Створити базові таблиці в режимі "Конструктор" за наступною структурою:

• Таблиця "Продукція". Містить список продукції, яку виготовляє підприємство.

Ім'я поля	**Тип поля. Примітка**
Код продукції	Числовий. Ключове поле
Найменування продукції	Текстовий
Рівень рентабельності	Текстовий
Собівартість	Грошова одиниця
Ціна продажу без ПДВ	Грошова одиниця
ПДВ	Числовий
Стисла характеристика продукції	Примітка

• Таблиця "Витрати". Містить перелік усіх можливих витрат на всі види продукції.

Ім'я поля	**Тип поля. Примітка**
Найменування витрати	Текстовий. Ключове поле
Ознака витрати	Так/Ні (Логічний). Ознака, що визначає, чи є витрата матеріальною або нематеріальною
Одиниця виміру	Текстовий

• Таблиця "Склад продукції". Містить перелік конкретних витрат на конкретний вид продукції. Ключове поле відсутнє.

Ім'я поля	**Тип поля. Примітка**
Код продукції	Числовий
Найменування витрати	Текстовий
Кількість	Числовий Для матеріальних витрат – це кількість одиниць, що потрібна на

	виготовлення продукції, для нематеріальних – завжди дорівнює “1”
Вартість	Грошовий. Для матеріальних витрат це ціна їх придбання за одиницю, для нематеріальних – загальна сума

- Таблиця “Одиниці виміру”.

Ім’я поля	Тип поля
Скорочена назва	Текстовий. Ключове поле
Повна назва	Текстовий

5. У таблиці “Продукція” для поля “Рівень рентабельності” змінити тип даних на “числовий”.

6. Завершити роботу з *Access*.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке “база даних” і чим вона відрізняється від електронної таблиці?
2. Які є моделі баз даних?
3. Що таке “реляційна модель”?
4. Що таке процес нормалізації бази даних?
5. За допомогою яких програм створюється база даних?
6. Яке призначення і функції СУБД?
7. З яким типом моделей баз даних працює *Access*?
8. З яких об’єктів складається файл бази даних?
9. Що таке запис?
10. Які є способи створення таблиці в базі даних?
11. Що таке структура таблиці бази даних?
12. Які об’єкти може містити файл бази даних?
13. Що таке поле?
14. Які бувають типи полів?

15. Яке призначення поля типу Примітка?
16. Що таке ключове (індексне) поле? Яке його призначення?
17. Чи обов'язкова наявність ключового поля у таблиці?
18. Як створити структуру бази даних?
19. Суть побудови таблиці в режимі конструктора?
20. Які ще є режими побудови таблиць?
21. Чому таблиці, як правило, будуються в режимі конструктора?
22. Суть модифікації структури таблиці.

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити базу даних.
2. Відкрити базу даних.
3. Створити таблицю бази даних.
4. Застосувати для поля потрібний тип даних.
5. Створити індексне (ключове) поле.
6. Змінити тип даних для побудованої раніше таблиці.

Лабораторна робота №2-3

Тема. Створення міжтабличних зв'язків. Введення даних. Робота із властивостями полів

Мета. Навчитися створювати зв'язки між таблицями, змінювати властивості полів, вводити дані.

Вказівки до виконання:

1. Якщо до схеми даних помилково двічі додана одна й та сама таблиця, слід її виділити і натиснути функціональну клавішу **<Delete>**.

2. У разі зменшення розміру поля дані, значення яких більше за встановлений новий розмір, можуть бути загублені. Тому за появи повідомлення "Можливо втрачено деякі дані" переконайтеся, що цього не відбудеться, після чого натисніть кнопку **«Так»**.

3. Поле “Код продукції” заповнюються зростаючою числовою послідовністю.

4. Значення англомовних термінів, які використовуються в програмі: “Null” – порожнє значення (дані в полі відсутні); “True“ – істинне; “False” – хибне.

5. Не слід надавати *однакові* імена ключовим полям *різних* таблиць, у яких міститься *різна* інформація: це може призвести до хибних результатів у побудові запитів. Наприклад, не слід надавати однакове ім’я “Найменування” для полів, у яких містяться дані про найменування продукції в одній таблиці і про найменування витрат – в іншій таблиці.

6. Введення даних у *пов’язану* таблицю зручно здійснювати з головної таблиці. Для цього після відкривання, наприклад, таблиці “Продукції”, яка є головною для таблиці “Склад продукції”, слід натиснути на піктограму ⊞, що знаходиться у першому стовпчику. Після цього під рядком з’являються (“розгортаються”) записи пов’язаної таблиці, які відповідають *вибраному рядку* головної таблиці. Ще одна суттєва перевага такого способу введення полягає в тому, що під час додавання нових записів *автоматично* заповнюється поле, за яким здійснюється зв’язок із головною таблицею. Для “закриття” списку записів пов’язаної таблиці слід натиснути кнопку ⊟.

Завдання

1. Створити зв’язки між таблицями із забезпеченням можливості вилучення пов’язаних записів:

- між однойменними полями “Код продукції” у таблицях “Продукція” та “Склад продукції”;
- між однойменними полями “Найменування витрати” у таблицях “Витрати” та “Склад продукції”;
- між полем “Скорочена назва” таблиці “Одиниці виміру” та полем “Одиниця виміру” таблиці “Витрати”.

2. Скасувати зв’язок між таблицями “Витрати” та “Одиниці виміру”.

3. Модифікувати (змінити структуру) таблиці “Одиниці

виміру”. Для поля “Скорочена назва” для властивості “Розмір поля” встановити значення “5” (символів), оскільки навряд чи скорочена назва буде мати більшу кількість символів.

4. Аналогічним чином модифікувати структуру для поля “Одиниця виміру” таблиці “Витрати”.

5. Відновити зв’язок між таблицями “Одиниці виміру” та “Витрати”.

6. Відкрити таблицю “Продукція” в режимі конструктора. Ознайомитися з властивостями текстового, числового і грошового типів даних.

7. Ввести в таблицю “Продукція” один вид продукції. Заповненню підлягають поля “Код продукції”, “Найменування продукції” і “Рівень рентабельності”. Значення останнього розрахувати як суму “індивідуального номера” і числа “10”.

8. Ввести в таблицю “Одиниці виміру” потрібні для витрат одиниці виміру (наприклад, “штуки“, “метри”, “кілограми”), а також “грн.”.

9. Відкрити таблицю “Витрати”. Виконати для неї наступні дії:

- Ввести одну матеріальну витрату. Для матеріальних витрат встановлювати (прапорець) для поля “Ознака витрати”.
- Спробувати ввести ще один запис із тим самим значенням для ключового поля “Найменування витрати”, яке було надано першому запису. Що відбудеться?
- Змінити структуру таблиці. Встановити для поля “Одиниця виміру” для властивості “Обов’язково” значення “Так”, а для властивості “Дозволити нульову довжину” значення “Ні”. Що відбудеться під час збереження змін таблиці?
- Спробувати ввести в таблицю запис без заповнення поля “Одиниця виміру”. Що відбудеться?
- Ввести усі витрати, потрібні на виготовлення продукції. Для нематеріальних витрат у полі “Одиниця виміру” вводити значення “грн.”. Кількість матеріальних витрат не менше 5-ти, нематеріальних – не менше 3-х.

10. Скасувати зв’язок між таблицями “Витрати” і “Склад продукції”.

11. Змінити структуру таблиці “Склад продукції”. Для поля “Найменування витрати” у стовпчику “Тип даних” вибрати значення “Майстер підстановок” і визначити, що “Стовпець підстановки” буде використовувати поле “Найменування витрати” таблиці “Витрати”. Передбачте, щоб значення цього поля (при виборі) були впорядковані за алфавітом.

12. Відкрити схему даних і переконатися у наявності зв’язку за полем “Найменування витрати” таблиць “Витрати” та “Склад продукції”. Як змінився характер зв’язку?

13. Ввести в таблицю “Продукція” ще один вид продукції. Передбачити, щоб обидва види продукції були однотипними, тобто складалися з однакових витрат, наприклад: “комп’ютер офісний”, “комп’ютер портативний”. Значення у полі “Рівень рентабельності” таке саме, як і для першого виду продукції.

14. Виконати для таблиці “Склад продукції” наступні дії:

- Встановити за допомогою відповідних властивостей обов’язкове заповнення “Код продукції” і “Найменування витрати”.
- Ввести одну витрату для першого виду продукції. Для матеріальних витрат у поле “Вартість” вводиться ціна їх придбання за одиницю (штуку), а для нематеріальних – сума (витрат). Для нематеріальних витрат у поле “Кількість” вводити значення “1”. Введення даних зручно здійснювати з таблиці “Продукція” (див. вказівки до виконання).
- Для поля “Кількість” встановити такі значення властивостей:

1.1. “Розмір поля” – “Ціле”, “Кількість знаків після коми” – “0”.

1.2. Властивості “Значення за промовчанням” надати значення “1”.

- Ввести ще одну витрату для першого виду продукції. Як змінився характер даних при їх введенні?

- Для поля “Кількість” встановити такі значення властивостей:

1.1. “Правило перевірки”: більше “0” (записується “>0”).

1.2. “Текст перевірки” заповнити текстом на зразок “У цьому полі обов’язково повинні бути дані”.

- Ввести ще одну витрату для першого виду продукції. У полі “Кількість” навмисно змінити значення з “1” на “0”. Що відбудеться?

15. Ввести всі витрати для обох видів продукції. Передбачити, що серед матеріальних витрат є такі, що потрібні для виготовлення продукції у кількості більше, ніж одна штука.

16. Завершити роботу з *Access*.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Як ввести дані у базу даних?
2. Яке призначення індексованого (ключового) поля?
3. Навіщо потрібні зв’язки між таблицями?
4. Яке призначення “Майстра підстановок”?
5. Як створити зв’язок між таблицями?
6. Як скасувати зв’язок між таблицями?
7. Визначення поняття “властивості поля”?
8. Чим є для властивості “Індексовано” значення “Повторення дозволяються”?
9. Яке призначення властивості “Розмір поля”?
10. Яке призначення властивості “Правило перевірки”?
11. Яке призначення властивості “Текст перевірки”?
12. Для чого призначена властивість “Значення за промовчанням”?
13. Яке призначення властивості “Обов’язково”?
14. Як організувати введення даних у поле однієї таблиці з іншої таблиці?
15. Чому введення даних до пов’язаної таблиці зручно

здійснювати з головної?

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити і скасувати зв'язок між таблицями.
2. Змінити властивості полів таблиці.
3. За допомогою "Майстра підстановок" організувати введення даних у поле з іншої таблиці.
4. Ввести, змінити, вилучити дані з таблиць.

Лабораторна робота №4

Тема. Побудова запитів

Мета. Засвоїти призначення запитів; навчитися створювати запити з параметром і на вибірку; формувати обчислювальні поля.

Вказівки до виконання:

1. Програма дає змогу створювати копії будь-яких об'єктів бази даних. Для цього виконуються стандартні дії копіювання та вставлення об'єкта. Так, для копіювання об'єкта його слід виділити і, наприклад, натиснути кнопку  на стрічці команд. Відповідно для вставлення можна, наприклад, на стрічці команд натиснути кнопку  .

2. Для надання значень логічним полям використовуйте такі: "Істина" (можна також застосувати англомовний аналог "True"); "Хибність" або "False" – англійською. Надавати значення можна обома мовами, але значення, що вводиться англійською мовою, автоматично перетворюються на українську.

3. Розрахункове поле будується за допомогою "Побудовника виразів". Для звернення до нього слід встановити курсор у порожню клітинку потрібного стовпчика бланка запиту, викликати контекстне меню й вибрати в ньому пункт "Побудувати…".

Завдання

1. Створити запит у режимі “Конструктор” за наступним зразком:

Назва продукції	Назва витрати	Кількість	Повна назва одиниці виміру	Вартість	Ознака витрати

2. Запам’ятати запит з ім’ям “Запит на вибірку-1”.

3. Створити із запиту “Запит на вибірку-1” копію з ім’ям “Запит на вибірку-2”.

4. Сформувати в ньому умову, що дозволить відібрати тільки матеріальні витрати.

5. Створити із запиту “Запит на вибірку-1” копію з ім’ям “Запит на вибірку-3”.

6. Упорядкувати у ньому записи в такому порядку: за найменуванням продукції, ознакою витрати, найменуванням витрати. Зверніть увагу, що для цього доведеться змінити порядок розташування полів таблиць. Передбачте, щоб нематеріальні витрати у запиті були розташовані перед матеріальними.

7. Створити із запиту “Запит на вибірку-1” копію з ім’ям “Запит з параметром”.

8. Встановити в ньому параметр, за яким можна відібрати витрати, вартість яких буде не менше значення, що вводиться користувачем з клавіатури.

9. Створити запит у режимі “Конструктор”, включивши до нього поля “Найменування продукції” та “Ознака витрати”.

10. Впорядкувати інформацію за найменуванням продукції.

11. Додати до запиту поле “Вартість”. Розрахувати для цього поля:

- загальну суму витрат;
- загальну кількість витрат;
- середнє значення (вартість однієї витрати).

12. Запам’ятати запит з ім’ям “Підсумковий запит”.

13. Створити запит у режимі “Конструктор”, включивши до нього поля “Найменування продукції”, “Ознака витрати”,

“Найменування витрати”, “Повна назва (одиниці виміру)”, “Вартість” і “Кількість”.

14. Створити в запиті розрахункове поле з ім’ям “Сума”, що буде розраховуватися як добуток полів “Вартість” і “Кількість”.

15. Ввести в запит параметр (вводиться користувачем з клавіатури), за яким можна здійснити відбір за певним видом продукції.

16. Запам’ятати створений запит з ім’ям “Підсумковий запит-2”.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке запит?
2. Які є типи запитів?
3. Що таке “простий запит на вибірку”?
4. Яке призначення конструктора запитів?
5. Яке призначення бланка запиту в режимі “Конструктор”?
6. Що таке “запит з параметром”?
7. Що таке “групова операція”?
8. Як додати до бланка запиту рядок “групова операція”, якщо він відсутній?
9. Найпростіші функції, що можуть бути застосовані в групових операціях.

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити запит у режимі “Конструктор”.
2. Додати або вилучити таблицю до запиту в конструкторі запитів.
3. Додати поля з таблиці до запиту в конструкторі запитів.
4. Вилучити поля з запиту.
5. Впорядкувати інформацію запиту в конструкторі запитів.
6. Створити умову для відбору інформації з таблиці.

7. Застосувати параметр для відбору інформації з таблиці.
8. Модифікувати запит.
9. Застосувати стандартні функції в групових операціях.

Лабораторна робота №5

Тема. Побудова форм

Мета. Навчитися створювати форми засобами СУБД *Access*.

Вказівки до виконання:

1. Інструменти, за допомогою яких відбувається робота над дизайном форми (вирівнювання, розташування полів, керуючих елементів і та ін.), містить вкладка "*Упорядкувати*". Наприклад, вирівнювання тексту підписів здійснюється за допомогою інструмента  "Вирівняти зліва" або властивості "Ліворуч" (відстань у см від лівого краю форми).

2. Розміри елемента керування можна довільно змінювати. Для цього потрібно його виділити, встановити курсор на один із маркерів, що оточують елемент по периметру, і перетягнути його в інше місце.

3. Виділення кількох об'єктів у формі проводиться шляхом натискання на них при одночасному утриманні натиснутою функціональної клавіші **<Shift>**.

4. *Майстер кнопок* повинен автоматично з'являтися у разі перенесення (перетягування) на форму елемента "Кнопка". Якщо цього не відбувається, на вкладці "Конструктор" натисніть ("утопіть") на панелі елементів кнопку  ("Застосувати майстри елементів керування").

5. Підсумкове значення для поля розраховується за допомогою функції ***Сума***. Для її застосування в області приміток створюється керуючий елемент "Текстове поле", що не прив'язується до жодного поля таблиці, тобто має формат "Без прив'язки". Функція застосовується у властивості "Джерело елемента керування" вкладки "*Дані*" аркуша властивостей.

6. Для відображення у формі області приміток слід викликати на ній контекстне меню та вибрати з нього пункт “Заголовок/примітка форми”.

7. Створення сторінок (вкладок) здійснюється за допомогою елемента керування “Вкладка”.

Завдання

1. Створити для таблиці “Склад продукції” за допомогою засобу “Автоформа” три варіанти форми: для введення одного запису, для відображення багатьох записів у табличному вигляді та “розділену форму”. До форми включати всі поля.

2. За допомогою “Конструктора” створити форму для таблиці “Склад продукції”. Включити у форму всі поля.

3. Виконати такі дії для дизайну форми:

- Встановити однакову ширину для приєднаних написів, вирівняти їх по лівому краю форми, а текст в написах – по правому краю.
- Розташувати на однаковій відстані по вертикалі приєднані написи і поля.
- Встановити однакову висоту для приєднаних написів і полів.

4. За допомогою Майстра кнопок створити у формі кнопки для навігації по таблиці, за допомогою яких можна буде переходити від запису до запису в обох напрямках, а також до останнього і до першого записів.

5. *** Створити поле, в якому розрахувати загальну суму витрат (підсумкове значення для поля “Вартість”).

6. За допомогою “Конструктора” створити форму “Продукція” для однойменної таблиці. Включити до неї поля “Код продукції”, “Найменування продукції” і “Рівень рентабельності”. Під час побудови виконати такі ж дії для дизайну форми, як і ті, що були застосовані для таблиці “Склад продукції”.

7. Модифікувати таблицю “Продукція”, включивши до її складу поля “Артикул” і “Вид продукції” (основна продукція,

запчастина). Тип і властивості полів визначити самостійно.

8. У режимі конструктора для таблиці “Продукція” створити форму “Відомості про продукцію”. Форма повинна відповідати таким вимогам:

- мати дві вкладки з іменами “Загальні відомості” й “Економічні показники”;
- поля “Найменування продукції” і “Код продукції” повинні бути розташовані безпосередньо у формі, поза вкладками;
- на вкладці “Загальні відомості” розмістити реквізити: “Артикул”, “Вид продукції”, “Стисла характеристика продукції”;
- на вкладці “Економічні показники” розмістити реквізити: “Рівень рентабельності”, “Собівартість”, “Ціна продажу без ПДВ”, “ПДВ”.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке форма? Яке її призначення?
2. З яких елементів складається форма?
3. Які є способи створення форми?
4. Які види форм можна створити в автоматичному режимі? Що вони собою являють?
5. Які види форм можна створити за допомогою майстра форм?
6. Яке призначення конструктора форм?
7. Що таке елемент керування? Які є елементи керування?
8. Чи має перевагу спосіб введення даних до таблиць за допомогою форм над введенням за допомогою таблиць, а якщо це дійсно так, то в чому визначаються такі переваги?
9. Чи обов’язково під час створення форми включати до неї всі поля таблиці?
10. Чи можна під час створення форми за допомогою майстра включати до неї тільки окремі поля таблиці, а якщо це так, то як це можна зробити на практиці?

11. Чи можна за допомогою форми додати до таблиці новий запис?
12. Що спільного між таблицями і формами?
13. Для чого у форму вставляють елементи керування?
14. Як вирівняти елементи форми?
15. Як розташувати елементи форми на однаковій відстані?
16. Як змінити відстань між елементами форми?
17. *** Що таке “функція” і як застосувати функцію у формі?

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити форму в режимі “Конструктор”.
2. Ввести у форму текстову інформацію.
3. Вирівняти елементи форми по лівому (правому) краю форми.
4. Вирівняти текст елементів для приєднаних надписів.
5. Розташувати елементи форми на однаковій відстані. Змінити цю відстань одночасно для усіх елементів.
6. Створити кнопку.
7. Надати користувачеві можливість відбору інформації за ключовим полем.
8. Створити форму, що має кілька сторінок.
9. *** Застосувати у формі функцію.

Лабораторна робота №6

Тема. Побудова звітів

Мета. Навчитися створювати звіти засобами СУБД *Access*.

Вказівки до виконання:

1. Для редагування текстового елемента звіту слід натиснути на ньому мишею, після чого він оточується по периметру рамкою з маркерами. Надалі виконується звичайне редагування тексту у рамці.
2. Розміри текстового поля можна довільно змінювати. Для

цього слід його виділити, встановити курсор на один із маркерів, що оточують поле по периметру, і перетягнути його в інше місце.

3. Розташування полів звіту в межах однієї сторінки можна досягти за рахунок зміни місця розташування полів, зменшення їх ширини, збільшення висоти. Наприклад, назви заголовків, що складаються з двох слів (“Найменування витрати”), в області верхнього колонтитулу можна розташувати в два рядки.

4. Для зміни розміру секції (даних, колонтитулу і т. ін.) слід встановити курсор на межу між секціями так, щоб він набув вигляду  , і звичайним перетягуванням перемістити її в іншу позицію.

Завдання

1. Створити із запиту “Підсумковий запит-2” копію з ім’ям “Калькуляція”.

2. Модифікувати запит:

- вилучити параметр, за яким здійснюється відбір за назвою продукції;
- упорядкувати дані за полями “Ознака витрати”, “Найменування витрати”;
- замість поля “Повна назва” (одиниці виміру) включити поле “Скорочена назва” (одиниці виміру).

3. За допомогою *Майстра звітів* створити звіт для цього запиту. При створенні звіту врахувати наступні вимоги:

- включити у звіт усі поля із запиту;
- згрупувати інформацію за полями “Найменування продукції”, “Ознака витрати”;
- впорядкувати записи за полем “Найменування витрати”;
- встановити підсумкове значення у вигляді суми (Сум) за полем “Сума”.

4. Відкрити створений запит у режимі “Конструктор” і виконати його редагування;

- у верхньому колонтитулі змінити текст “Скорочена назва” (одиниці виміру) на “Од. вим.”;

- розташувати усі поля звіту в межах однієї сторінки;
- у верхньому колонтитулі вирівняти текст заголовків по центру;
- в області даних вирівняти поля відносно заголовків цих полів в області колонтитулу;
- замінити всі англомовні написи на відповідні українські. Наприклад, текст “Summary” – на “Підсумок”, текст “Page” – на “Сторінка” тощо.

5. Модифікувати запит “Калькуляція”. Ввести в запит параметр (вводиться користувачем з клавіатури), за яким можна здійснити відбір за певним видом продукції.

 *Контроль знань та навичок*

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке звіт? Яке його призначення?
2. Для чого призначений у звіті “рівень групування”?
3. У яких режимах можна створити звіт?
4. Чи можна у звіті замінити текстову інформацію?

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити звіт у режимі майстра звітів.
2. Змінити набір полів (додати нові або вилучити існуючі).
3. Створити рівні групування.
4. Упорядкувати інформацію за потрібними полями.
5. Обчислити підсумкові значення.
6. Змінити орієнтацію звіту.
7. Редагувати створений звіт у режимі “Конструктор”.

Лабораторна робота №7

Тема. Створення макросів

Мета. Навчитися створювати найпростіші макроси і розташовувати їх на власних панелях інструментів і меню.

Вказівки до виконання:

1. У створенні макросу "Макрос-повідомлення" використовуються макрокоманди "Повідомлення" (MsgBox) і "Відкрити форму" (OpenForm).

2. У створенні макросу "Склад продукції" врахуйте наступне:

- для відкривання запиту використовується макрокоманда "Відкритизапит" (OpenQuery);
- для виведення складу продукції, що відображається у формі, застосовується макрокоманда "Застосуватифільтр" (ApplyFilter). Як аргумент у нашому випадку слід вибрати "Умова відбору" (Where Condition). Його значення сформуйте за допомогою *побудовника виразів*. Це значення у вигляді виразу повинно пов'язати між собою однойменні поля "Найменування продукції" у формі "Продукція" і запиті "Запит на вибірку-1".

3. При створенні кнопки "Склад продукції" визначають дію, що буде виконуватися у разі натискання кнопки. Така дія є макросом, що відкриває запит "Запит на вибірку-1". Тому для властивості кнопки "Після клацання" (On Click) на вкладці "Подія" (Event) слід розкрити список і вибрати з переліку потрібний макрос.

4. Під час редагування властивостей об'єкта слід розрізняти призначення властивостей "Підпис" (Caption) на вкладці "Формат" (Format) та "Ім'я" (Name) на вкладці "Інші" (Other). Перша властивість визначає найменування (підпис), що бачить користувач на екрані, друга властивість визначає ім'я, за яким відбувається звернення до об'єкта.

5. Під час створення макросів урахуйте наступне:

- Для задання імен макросів "Загальні відомості" і "Економічні показники" у складі макросу "ПерехідНаСторінку" в бланку конструктора макросів необхідно відобразити стовпчик

“Ім’я макросу”.

- Під час створення макросів “Загальні відомості” та “Економічні показники” для переходу із сторінки на сторінку застосуйте макрокоманду “ПерейтиДоЕлементаКерування”. Вона має обов’язковий аргумент “Ім’я елемента керування”, значення якого відповідає імені вкладки, на яку слід здійснити перехід.

6. Ім’я об’єкта, в тому числі і сторінки (вкладки), надається за допомогою властивості “Підпис”. Властивість “Ім’я” залишається незмінним – “Сторінка1”. Саме це ім’я (“Сторінка”) використовується в аргументі макрокоманди для переходу на потрібну вкладку.

Завдання

1. Створити макрос з ім’ям “Макрос-повідомлення”, за допомогою якого буде відкриватися форма “Склад продукції”. Передбачити таке:

- Відображенню підлягають дані тільки для одного виду продукції.
- За допомогою макрокоманди “Повідомлення” сформувати повідомлення, що форма відображає дані в режимі перегляду для конкретного виду продукції.
- Заборонити редагування даних під час їх перегляду.

2. Створити макрос “Склад продукції” для відкриття запиту “Запит на вибірку-1”. Передбачити таке:

- Задати умову, яка дозволить переглядати склад тільки тієї продукції, картка якої відображається на екрані.
- Заборонити редагування даних під час їх перегляду.

3. Модифікувати таблицю “Продукція”, включивши до її складу поля “Артикул” і “Вид продукції” (основна продукція, запчастина). Тип і властивості полів визначити самостійно.

4. У режимі конструктора для таблиці “Продукція” створити форму “Відомості про продукцію”. Форма повинна відповідати таким вимогам:

- Форма повинна мати дві сторінки з іменами “Загальні відомості” й “Економічні показники”.

- Розмістити безпосередньо у формі (над елементом “вкладка”) поля “Найменування продукції” і “Код продукції”.
- Розмістити на першій сторінці реквізити: “Артикул”, “Вид продукції”, “Стисла характеристика продукції”.
- Розмістити на другій сторінці реквізити: “Рівень рентабельності”, “Собівартість”, “Ціна продажу без ПДВ”, “ПДВ”.

5. Створити макрос “Перехід”. Створити в його складі два макроси, які нададуть можливість переходу до певної сторінки форми, з іменами “Загальні відомості” й “Економічні показники”.

6. Розташувати на вкладці “Загальні відомості” форми “Відомості про продукцію” кнопку, за допомогою якої можна здійснити перехід на другу вкладку, і надати їй ім’я “Економічні показники”, а на іншій вкладці – кнопку, за допомогою якої можна здійснити перехід на першу вкладку, і надати їй ім’я “Загальні відомості”. Під час створення кнопки визначити дію, яка буде виконуватися при натисканні кнопки. Оскільки така дія є макросом, то для цього слід вибрати категорію “Інше”, з неї дію “Запустити макрос” і вибрати потрібний макрос.

7. Використовуючи властивості кнопок, досягнути однакових розмірів і розташування обох кнопок на вкладках.

8. *** У формі “Продукція” створити кнопку для виклику даних про склад продукції (“Запит на вибірку-1”), картка якої відображається на екрані. Назвати кнопку “Склад продукції”. Натискання цієї кнопки повинно призводити до відкривання запиту “Запит на вибірку-1”.

 Контроль знань та навичок

Після виконання лабораторної роботи студент повинен ***знати***:

1. Що таке макрос?
2. Яке призначення має макрос?
3. З яких елементів складається вікно конструктора макросів?
4. Як відбувається побудова макросу за допомогою

конструктора макросів?

5. Що таке аргумент макросу?
6. Чи є обов’язковим наявність аргументів у макросі?
7. Як надати значення аргументу?
8. Які існують варіанти виконання макросів?
9. Що таке подія? У результаті чого виникають події?
10. Як призначити макрос для події?
11. Яка макрокоманда використовується для виведення повідомлення користувачу?
12. Які макрокоманди використовуються для відкривання форми, запиту?
13. Яке призначення має макрокоманда “ЗастосуватиФільтр”? Які вона має аргументи і як визначаються їхні значення?
14. Які аргументи і з якими значеннями застосовують для заборони зміни даних під час роботи користувача з такими об’єктами, як форми та запити?
15. Які елементи управління використовуються для створення форми з кількох сторінок?
16. Як створити групу макросів?
17. Який макрос використовується для навігації за сторінками форми? Які аргументи він має?
18. Що таке подія? У результаті чого виникають події?

Після виконання лабораторної роботи студент повинен ***уміти***:

1. Створити макрос за допомогою конструктора макросів.
2. Змінити природний порядок виконання макросів у групі макросів.
3. Виконати макрос.
4. Застосувати макроси для оброблення подій.
5. Застосувати макроси для відкривання форм, запитів, виведення повідомлень користувачеві, відбору частини даних таблиці (здійснення фільтрації даних).

6. Створити макрос для оброблення події “Після клацання”.
7. Створити групу макросів.
8. Створити макрос для навігації по сторінках форми.
9. Визначати розташування кнопок у формі за допомогою її властивостей.
10. Встановити потрібні розміри у формі за допомогою її властивостей.

Термінологічний словник

Автентифікація (***Authentication***). Процес підтвердження автентичності користувача за допомогою пароля.

База даних. Особливим чином організована сукупність взаємопов'язаних, збережених разом даних.

Домен. Усі значення для поля таблиці.

Головна таблиця. Таблиця, що бере участь у зв'язку з іншою таблицею своїм ключовим полем.

Запит. Об'єкт бази даних. Призначений для відбору даних із таблиць на основі заданих користувачем критеріїв і відображення їх у зручному вигляді.

Захист інформації. Сукупність методів і засобів, що забезпечують конфіденційність, достовірність, автентичність та доступність інформації в умовах загрози доступу до неї з боку осіб, які не мають права на користування цією інформацією.

Звіт. Об'єкт бази даних, призначений для виведення даних, передусім на друк.

Ієрархічна модель – така організація даних у СУБД, за якою дані одного рівня підпорядковуються даним, що знаходяться на вищому рівні.

Інкапсуляція. Властивість зберігати інформацію (властивості, методи, події) про об'єкт.

Ключове поле. Поле (або група полів) таблиці, значення якого однозначно визначає кожний запис.

Ключове слово. Текстове найменування у системі, що зарезервоване з певною метою, наприклад, для індексування даних таблиці.

Кортеж. Значення всіх стовпчиків, з'єднаних в одному рядку таблиці.

Макроси. Невеличкі програми, що складаються із послідовності макрокоманд. За допомогою макросів вирішуються питання, що не можуть бути вирішені шляхом "візуального" програмування.

Мережна модель організації даних в СУБД. Модель, за якою будь-який елемент може бути пов'язаний із довільною кількістю інших.

Монопольний режим. Режим роботи, коли із системою може працювати тільки один користувач.

Надлишкові дані. Дані, що повторюються в базі даних.

Нормалізація бази даних. Спеціальна процедура розподілу таблиць бази даних, у процесі якого одна таблиця розподіляється на кілька з метою, насамперед, вилучення з таблиць інформації, що повторюється.

Обліковий запис. Інформація про користувача: його ім'я і особистий код, що призначена для визначення його прав під час роботи з базою даних.

Подія. Деяка дія, що активізує стандартну реакцію об'єкта (натискання кнопки миші, функціональної клавіші, вибір пункту меню і т. ін.)

Порядок відношення. Кількість стовпчиків у таблиці.

Права доступу. Набір правил, що регламентують дозвіл для конкретного користувача або однотипних груп користувачів працювати з тією чи іншою частиною інформації і здійснювати цілком визначені дії над певними об'єктами.

Синтаксис. Набір правил, що встановлює структуру висловлювань мови (програмування).

Сторінки доступу до даних. Об'єкт бази даних, виконаний у коді HTML, що розміщується на Web-сторінці і передається клієнту разом з нею. За його допомогою здійснюють інтерфейс між клієнтом, сервером і базою даних, що міститься на сервері.

Структура бази даних. Принцип організації записів у базі даних і зв'язків між ними.

Таблиця. Основний об'єкт бази даних. У таблицях зберігаються всі дані, а самі таблиці зберігають і структуру бази.

Тригер. Обмеження для таблиці: правила контролю даних, що застосовуються на рівні запису і визначаються до таблиці в цілому.

Форма. Об’єкт бази даних, призначений для зручного введення даних.

Structured Query Language (***SQL***). Мова, що була спеціально розроблена для створення запитів, а також поновлення та керування реляційними базами даних.

SQL-запит. Запит, що створюється за допомогою інструкцій SQL.

QBE (**Query By Example**). Технологія побудови запиту “за прикладом”.

VBA (***Visual Basic for Application***). Visual Basic для додатків. Мова програмування, що постачається разом із офісним пакетом Microsoft Office.

Література

1. Вилларіал Б. Программирование Access 2002 в примерах. / Б.Вилларіал. – М.: КУДИЦ-ОБРАЗ, 2003. – 496 с.

2. Киммел Пол. Освой самостоятельно программирование для Microsoft Access 2002 за 24 часа. – М.: Издательский дом “Вильямс”, 2003. – 480 с.

3. Фетісов В.С. СУБД Access-2007: навчально-методичний посібник / В.С. Фетісов. – Ніжин: Видавництво НДУ ім. М. Гоголя, 2009. – 95 с.