

Тема 4. Облік в умовах автоматизації.

Застосування СУБД MS Access у обліку

4.1. Основні поняття БД та СУБД.

База даних — це організована структура, призначена для збереження інформації. Ми знаємо, що дані й інформація — поняття взаємозалежні, але не тотожні, і повинні помітити невідповідність у цьому визначенні. Його причини чисто історичні. В ті роки, коли формувалося поняття баз даних, у них дійсно зберігалися тільки дані. Однак сьогодні більшість систем управління базами даних (СУБД) дозволяють розміщати у своїх структурах не тільки дані, але і методи (тобто програмний код – макроси, тригери і т.д.), за допомогою яких відбувається взаємодія з споживачем з іншими програмно-апаратними комплексами. **Таким чином, ми можемо говорити, що в сучасних базах даних зберігаються не тільки дані, але й інформація.**

Це твердження легко пояснити, якщо, наприклад, розглянути базу даних великого банку. У ній є всі необхідні відомості про клієнтів, про їхні адреси, кредитну історію, стан розрахункових рахунків, фінансових операціях і т.д. Доступ до цієї бази є в досить великій кількості співробітників банку, але серед них навряд чи знайдеться така особа, що має доступ до всієї бази в цілому і при цьому здатна одноосібно вносити в неї довільні зміни. Крім даних, база містить методи і засоби, що дозволяють кожному зі співробітників оперувати тільки з тими даними, що входять у його компетенцію. У результаті взаємодії даних, що містяться в базі, з методами, доступними конкретним співробітникам, утворюється інформація, що вони споживають і на підставі якої в межах власної компетенції роблять введення і редагування даних.

З поняттям бази даних тісно зв'язане поняття системи управління базою даних. Це комплекс програмних засобів, призначених для створення структури нової бази, наповнення її вмістом, редагування вмісту і візуалізації інформації. Під візуалізацією інформації бази розуміють вибірку з даних відповідно до заданого критерію, їхнє упорядкування, оформлення і наступна видача на пристрій виводу або передача по каналах зв'язку.

У світі існує безліч систем управління базами даних. Незважаючи на те що вони можуть по-різному працювати з різними об'єктами і надають користувачу різні функції і засоби, більшість СУБД спираються на єдиний сталий комплекс основних понять. Це дає нам можливість розглянути одну систему й узагальнити її поняття, прийоми і методи на весь клас СУБД. Як такий навчальний об'єкт ми виберемо СУБД Microsoft Access.

Відразу пояснимо, що якщо в базі немає ніяких даних (порожня база), те це все рівно повноцінна база даних. Цей факт має методичне значення. Хоча даних у базі і нема, але інформація в ній усе-таки є — це **структура бази**. Вона визначає методи занесення даних і збереження їх у базі. Найпростіший «некомп'ютерний» варіант бази даних - діловий щоденник, у якому кожному календарному дню виділено по сторінці. Навіть якщо в ньому не записано ні рядка, він не перестає бути щоденником, оскільки має структуру, що чітко відрізняє його від записних книжок, робочих зошитів і іншої паперової продукції.

Бази даних можуть містити різні об'єкти, але, забігаючи наперед, скажемо, що основними об'єктами будь-якої бази даних є її **таблиці**. Найпростіша база даних має хоча б одну таблицю. Відповідно, структура найпростішої бази даних тотожно дорівнює структурі її таблиці.

Ми знаємо, що структуру двовимірної таблиці утворюють стовпці і рядки. Їхніми аналогами в структурі найпростішої бази даних є **поля** і **записи**. Якщо записів у таблиці поки нема, виходить, її структура утворена тільки набором полів. Змінивши склад полів базової таблиці (чи їхні властивості), ми змінюємо структуру бази даних і, відповідно, одержимо нову базу даних.

Користувач може розробити свій інтерфейс до БД, основу якого складають **форми**. На формах розміщуються різні елементи: поля таблиць, поля із списком, кнопки, списки, що розкриваються, вимикачі, малюнки, підлеглі форми і т.д. База даних разом з інтерфейсом користувача називається **додатком БД**.

4.2. Загальна характеристика Access

Microsoft Access - реляційна система управління базами даних, яка працює під ОС Windows.

Access має стандартизований інтерфейс програм MS Office (Рис. 160). Більшість дій по роботі з елементами баз даних можна виконати з допомогою: команд основного меню, кнопок панелей інструментів, команд контекстного меню і комбінацій клавіш.

Ми вже згадали про те, що крім таблиць база даних може містити й інші типи об'єктів. Привести повну

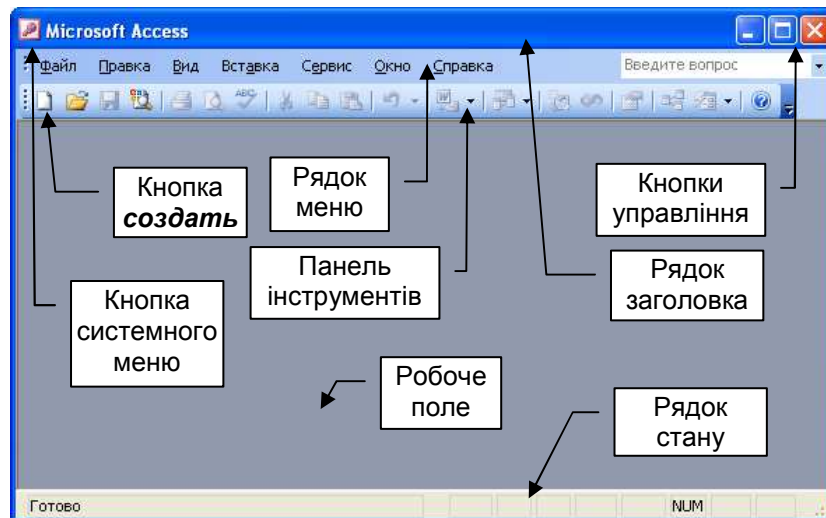


Рисунок 160

класифікацію можливих об'єктів баз даних важко, оскільки кожна система керування базами даних може реалізувати свої типи об'єктів. СУБД Microsoft Access дозволяє створювати і використовувати об'єкти семи різних типів.

Таблиця. Як ми вже говорили, це основні об'єкти будь-якої бази даних. По-перше, у таблицях зберігаються всі дані, що є в базі, а по-друге, таблиці зберігають і структуру бази (поля, їхні типи і властивості). Основні операції над таблицями: перегляд і оновлення (введення, модифікація і видалення), сортування, фільтрація і друк.

Форма – основа для конструювання графічного діалогового інтерфейсу користувача для роботи з БД. Зміст їх у тому, щоб надати користувачу засіб для заповнення тільки тих полів, що йому заповнювати дозволено. Одночасно з цим у формі можна розмістити спеціальні елементи управління (лічильники, списки що розкриваються, перемикачі, прапорці та інше) для автоматизації введення. Переваги форм розкриваються особливо тоді, коли відбувається введення даних із заповнених бланків. У цьому випадку форму роблять графічними засобами так, щоб вона повторювала оформлення бланка — це помітно спрощує роботу того хто працює з формою, знижує його стомлення і запобігає появі друкованих помилок. Форма зазвичай є засобом для введення даних, хоча з їхньою допомогою дані можна і переглядати.

Звіт - по своїх властивостях і структурі звіти багато в чому схожі на форми, але призначені тільки для виводу даних, причому для виводу не на екран, а на друкувальний пристрій (наприклад принтер). У зв'язку з цим звіти відрізняються тим, що в них прийняті спеціальні прийоми для групування виведених даних і для виводу спеціальних елементів оформлення, характерних для друкованих документів (верхній і нижній колонтитули, номери сторінок, службова інформація про час створення звіту, логотип компанії і т.п.) .

Запити. Ці об'єкти служать для витягу даних з таблиць і надання їх користувачу в зручному виді. За допомогою запитів виконують такі операції, як вибірку даних, їхнє сортування і фільтрацію. За допомогою запитів можна виконувати перетворення даних по заданому алгоритму, створювати нові таблиці, виконувати автоматичне наповнення таблиць даними, імпортованими з інших джерел, виконувати найпростіші обчислення в таблицях і багато чого іншого.

Починаючи користувачі не відразу розуміють роль запитів, оскільки всі ті ж операції можна робити і з таблицями. Так, дійсно, це так, але є поняття зручності (у першу чергу швидкодії) і поняття безпеки.

З міркувань безпеки, чим менше доступу до базових таблиць мають кінцеві користувачі, тим краще. По-перше, знижується ризик того, що необережними діями вони ушкодять дані в таблицях. По-друге, надавши різним користувачам різні запити, можна ефективно розмежувати їхній доступ до даних у суворій відповідності з колом персональних обов'язків. У банках, наприклад, одні співробітники мають доступ до таблиць даних про клієнтів, інші — до їх розрахункових рахунків, треті — до таблиць активів банку. Якщо і є спеціальні служби, що мають доступ до всіх інформаційних ресурсів банку (з метою контролю й аналізу), то вони позбавлені засобів для внесення змін — усе зроблено так, щоб одна людина не могла зробити фіктивну операцію, незалежно від того, яку посаду він займає. У базі даних, що має правильно організовану структуру, для здійснення протиправних дій необхідна змова декількох учасників, а такі дії припиняються не програмними, а традиційними засобами забезпечення безпеки.

Особливість запитів полягає в тому, що вони використовують дані з базових таблиць і створюють на їхній основі тимчасову *результуючу таблицю*. Якщо хочуть підкреслити факт «тимчасовості» цієї таблиці, то її ще

називають *моментальним знімком*. Коли ми працюємо з *основними таблицями* бази, ми фізично маємо справу з жорстким диском, тобто з дуже повільним пристроєм (нагадаємо, що це зв'язано з особливістю збереження даних). Коли ж на підставі запиту ми одержуємо *результуючу таблицю*, те маємо справу з електронною таблицею, що не має аналога на жорсткому диску, — це тільки *образ* відібраних записів з БД. Так як цей «образ» формується і зберігається у оперативній пам'яті комп'ютера, зрозуміло, робота з ним відбувається набагато швидше й ефективніше — це ще одна підстава для того, щоб широко використовувати запити.

Макроси і модулі. Ці категорії об'єктів призначені як для автоматизації повторюваних операцій при роботі із системою управління базами даних, так і для створення нових функцій шляхом програмування. У СУБД Microsoft Access *макроси* складаються з послідовності внутрішніх команд СУБД і є одним із засобів автоматизації роботи з базою. *Модулі* створюються засобами зовнішньої мови програмування, у даному випадку мови Visual Basic for Applications. Це один із засобів, за допомогою яких розроблювач бази може закласти в неї нестандартні функціональні можливості, задовольнити специфічні вимоги замовника, підвищити швидкодію системи управління, а також рівень її захищеності.

БД Access може включати до 32 368 об'єктів (таблиці, форми, звіти і т.д.). Одночасно можна відкрити до 2048 таблиць. Таблиці модно імпортувати з інших БД Access, а також з БД FoxPro, Paradox, Microsoft SQL Server або з електронних таблиць Excel і ін.

Учебні бази даних: «Борей», «Замовлення» і «Рішення» дозволяють краще зрозуміти, як слід організувати і розробляти власні БД і як використовувати Access.

Питання для самоконтролю

1. Перерахуйте і дайте характеристику об'єктам, з яких складається база даних Access.

4.3. Створення бази даних

У області завдань в групі **Создание** (Рис. 161) наявні декілька можливостей створення бази даних.

Новая база данных - створює нову БД з порожнього файлу. Спосіб відрізняється гнучкістю і трудомісткістю - вимагає окремого визначення кожного елементу БД.

Создание из имеющегося файла - вибирає існуючий файл як основу для нової БД.

Создание с помощью шаблона - створює БД по одному з шаблонів за допомогою Майстра. Спосіб прискорює процес створення БД і дозволяє одержати базу із зразками інформації в таблицях. Він застосовується, коли користувачу підходить одна із запропонованих типових БД.

Незалежно від способу створення БД її можна завжди змінити і розширити.

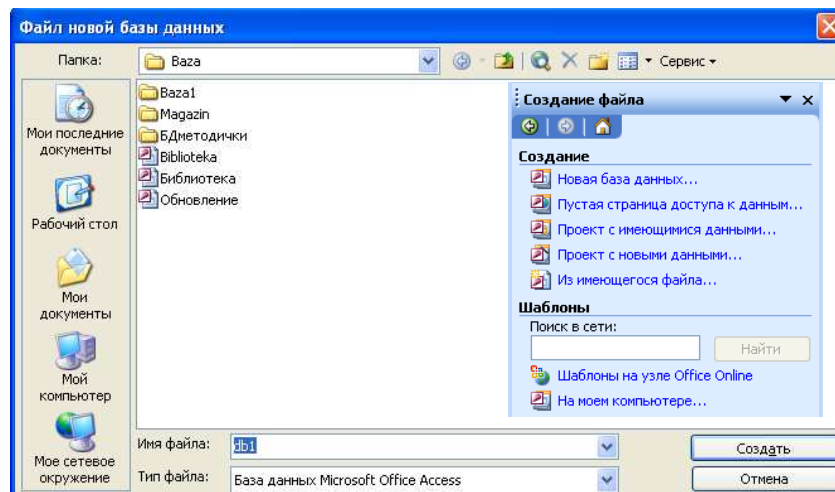


Рисунок 161

Access зберігає всі об'єкти БД в одному файлі. Це спрощує технологію ведення БД і додатку користувача. Забезпечується компактність розміщення всіх об'єктів БД на диску і ефективність обробки даних. Файл порожньої БД можна створити, вибравши рядок - **Новая база данных**. Відкриється діалогове вікно (Рис. 161), в якому треба вибрати папку і задати ім'я нової БД. Ім'я файлу - не більше 215 символів не повинне містити символи: \ / ? : * " < > |. За умовчанням файли зберігаються в папці **Мои документы** мають ім'я db1 і розширення .mdb. Додаток БД може бути скопійований і збережений у файлі формату MDE. При цьому початковий текст програм на VBA віддаляється, БД стискається (що значно скорочує розмір файлу) після чого її неможливо модифікувати.

Після створення файлу з'являється вікно відкритої БД (Рис. 162), в якому виконується робота із створення елементів бази даних.

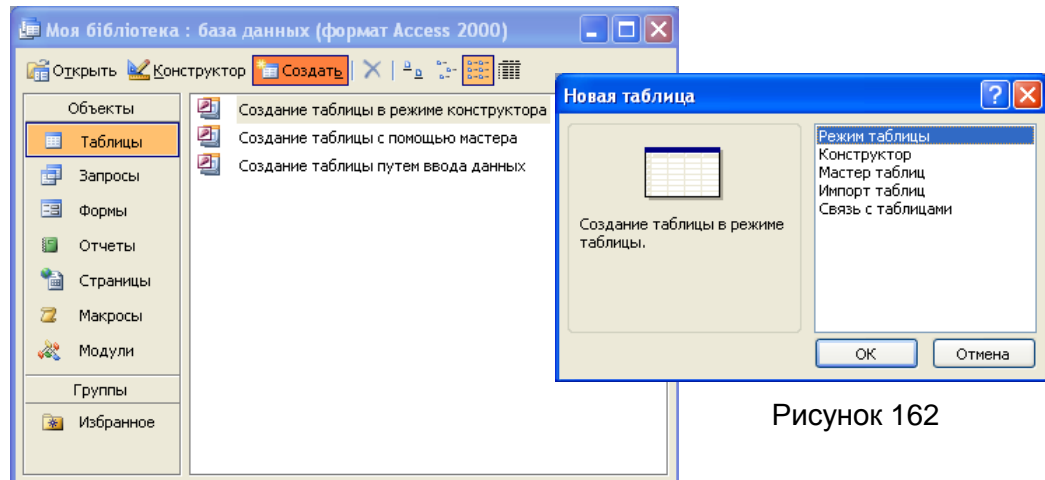


Рисунок 162

У розділі **Объекты** представлені всі об'єкти, які можуть бути створені в БД. При створенні нової БД список для будь-якого вибраного типу об'єктів відсутній.

Завдання для самостійної роботи

1. Запустіть програму **Access** і створіть нову базу даних **Моя бібліотека**, яку збережіть у папці **Навчання** → **Навчання 1** → **Таблиці**.

4.4. Створення таблиць

Створення таблиці складається з двох етапів. Спочатку визначається її структура: склад і розміщення полів, їх імена, тип даних і розмір кожного поля, ключі, індекси і т.п. На другому етапі створюються записи таблиці, які заповнюються даними.

У вікні БД (Мал. 162) вибрати об'єкт **Таблицы** і натиснути кнопку **Создать**. Відкриється вікно **Новая таблица** (Рис. 162)

Варіанти створення таблиць:

- введення даних в порожню таблицю - **Режим таблицы**;
- за допомогою Конструктора - **Конструктор**;
- за допомогою Майстра - **Мастер таблиц**;
- з таблиць, що імпортуються, - **Импорт таблиц**;
- шляхом створення таблиць, пов'язаних з таблицями, що знаходяться в зовнішньому файлі - **Связь с таблицами**.

Незалежно від способу створення, зміну структури таблиці можна виконати в режимі **Конструктор**. Для користувачів тих, що мають досвід розробки БД саме цей спосіб рекомендується для створення таблиць.

Для переходу в режим **Конструктор** (виділити рядок **Конструктор** і клацнути по кнопці **OK**) з'явиться вікно (Рис. 103) для введення характеристик полів (стовпців таблиці). У верхній частині форми - таблиця із списком полів. Кожному полю - відповідає один запис (рядок).

Имя поля – ім'я, по якому програма звертається до поля. Наприклад, ПІБ, НазвКаф, Ном_дом. Ім'я повинне бути унікальним і не містити символи: ! . ' [] .

Тип даних визначається значеннями даних, які вводяться в поле і операціями з ними. Всього дев'ять типів даних, які можна вибрати з випадного списку.

- **Текстовый** – тип даних за умовчанням. Текст або цифри, що не беруть участь в розрахунках. Кількість символів (максимально 255) указується у властивості **Размер поля**.
- **Поле MEMO** – довгий текст до 65 535 символів.
- **Числовой** – дані, які використовуються в обчисленнях. Розмір поля 1, 2, 4 або 8 байт.
- **Денежный** – грошові і числові дані, використовувани в розрахунках з точністю до 15 знаків в цілій частині і до 4 знаків в дробі без округлення.
- **Дата/время** – значення дати і часу від 100 до 9999 року.

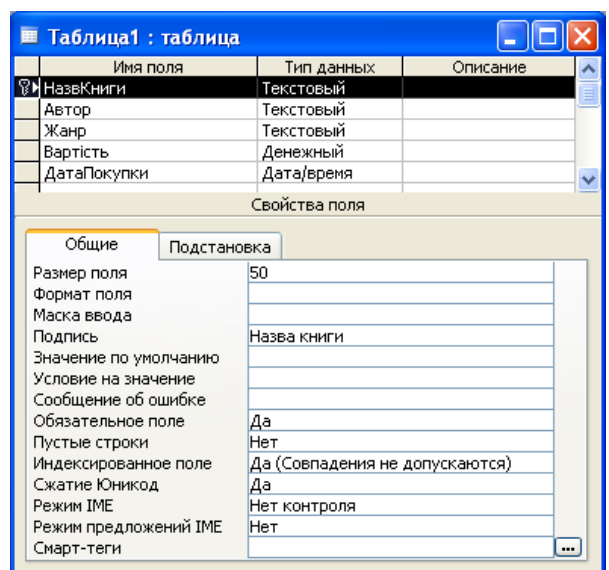


Рисунок 163

- **Счётчик** – використовується для створення унікального ключа таблиці. Вводиться автоматично для кожного нового запису виді зростаючого (на 1) або випадкового значення. Значення поля не можна змінити або видалити.
- **Логический** – логічні дані, які можуть мати одне з двох можливих значень: Да/Нет, Істина/Брехня, Вкл./Выкл.
- **Поле объекта OLE** – таблиця Excel, документ Word, малюнок, звукозапис або інші дані зв'язані або упроваджені в таблицю Access.
- **Гиперссылка** – шлях до файлу на диску або адреса в мережі Інтернет.
- **Мастер подстановок** – будує для поля список на основі значень полів з іншої таблиці.

Після вибору типу даних в нижній частині форми стають доступними **Свойства поля**, які відрізняються для різних типів даних. Деякі властивості указуються за умовчанням, інші необхідно задати. Основні властивості полів:

Размер поля – встановлює граничну довжину даних (у символах), які можуть розміщуватися в даному полі. Важливо правильно встановити розмір поля: якщо розмір поля перевищуватиме необхідний, то пам'ять використовуватиметься неефективно; якщо розмір поля буде менший за необхідне, то частина символів буде відкинута;

Формат поля – визначає спосіб форматування даних в елементах таблиці;

Маска ввода – визначає форму введення даних в полі. Наприклад, номер телефону 35-45-55;

Подпись – визначає заголовок поля, яке буде виведено в заголовку стовпців таблиці. Наприклад, Прізвище, ім'я і по батькові, Назва кафедри, Номер будинку;

Значение по умолчанию – те значення, яке вводиться в поле автоматично;

Условие на значение – обмеження, використовуване для перевірки правильності введення даних. Наприклад, дата покупки товару не може перевищувати поточну дату;

Сообщение об ошибке – повідомлення, яке виводиться автоматично при введенні помилкових даних;

Обязательное поле – може приймати значення Да/Нет. Якщо Так, то потрібне обов'язкове введення даних в полі. Якщо Ні, то дані в поле вводити необов'язково. Наприклад, поля Національність або Віросповідання;

Особливо слід звернути увагу на властивість - **Индексированное поле**. Воно може приймати значення: *Нет - неиндексированное, Да (Допускаются совпадения) и Да (Совпадения не допускаются)*. Індексация поля ще не означає, що поле є ключовим. Щоб зробити поле ключовим, потрібно спочатку задати властивості поля *Да (Совпадения не допускаются)*, потім виділити рядок опису поля і натиснути кнопку із зображенням ключа на панелі інструментів. Якщо ключ складений, то в цій властивості для кожного з вхідних в нього полів повинно бути задано - *Да (Допускаются совпадения)*.

Звичайно таблиця повинна мати хоч би одне ключове поле. Тому при завершенні створення таблиці за відсутності в ній ключа Access попереджає про це і пропонує створити ключ типу *Счетчик*. Це можна прийняти або ігнорувати і створити ключі пізніше.

Після створення таблиць потрібно виконати їх зв'язування, що дозволяє контролювати цілісність і достовірність інформації в БД. Крім того, чим раніше утворені зв'язки між таблицями, тим зручніше створювати багатотабличні запити, форми і звіти. За наявності зв'язків між таблицями розробка перерахованих об'єктів БД ведеться з урахуванням встановлених зв'язків. Тому важливою є послідовність створення таблиць. Першими слід створювати основні (батьківські) таблиці (на стороні зв'язку 1). Потім - залежні (дочірні) таблиці і таблиці які мають поля зі списками що розкриваються (на стороні зв'язку ∞), котрі використовують дані з основних таблиць.

Завдання для самостійної роботи

1. В базі даних **Моя бібліотека** створіть наступні таблиці. Для полів **Автор** і **Жанр Табл_Книги** за допомогою Мастера подстановок дані мають заноситися з відповідних полів **Табл_Автор** і **Табл_Жанр**

Ім'я таблиці	Ім'я поля	Тип даних	Розмір поля
Табл_Жанр	Жанр	Текстовий	50
Табл_Автор	Автор	Текстовий	50
Табл_Книги	Назва	Текстовий	50
	Автор	Текстовий	50
	Жанр	Текстовий	50

4.5. Зв'язування таблиць

Для зв'язування таблиць БД потрібно викликати вікно схеми даних. Зразковий вид вікна приведений на Рис. 164.

Після відкриття основного вікна БД вікно схеми бази даних можна викликати по команді **Сервис** → **Схема даних** основного меню або за допомогою однойменної команди контекстного меню.

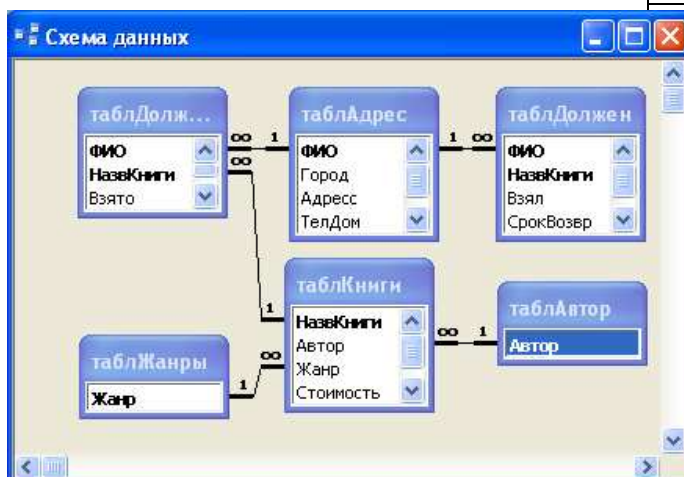


Рисунок 164

Побудова схеми даних полягає в додаванні в неї таблиць і встановленні зв'язків між ними. Додавання таблиць звичайно проводиться за допомогою відповідної кнопки на панелі інструментів..



У разі збігу імен і типів полів у декількох таблицях Access утворює зв'язки автоматично. Якщо встановлені зв'язки не влаштовують користувача, то їх можна змінити.

Якщо Access побудувала не всі потрібні зв'язки (імена полів що зв'язуються можуть не співпадати), або не будувала їх зовсім, то для утворення нового зв'язку необхідно виділити поле однієї таблиці, при натиснутій лівій кнопці миші протягнути покажчик до поля зв'язку іншої таблиці і відпустити кнопку. При цьому Access пропонує визначити вигляд і параметри зв'язку в діалоговому вікні **Изменение связей** (Рис. 165).

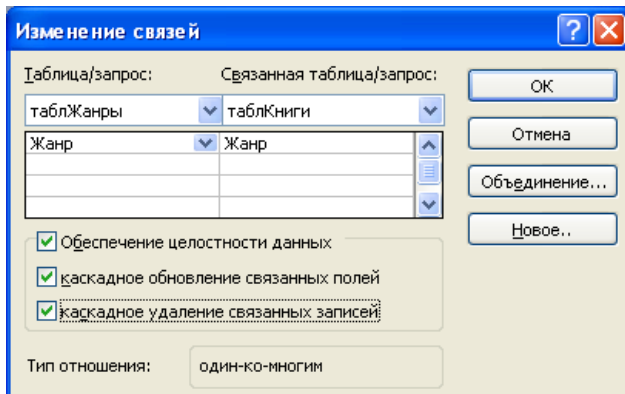


Рисунок 165

таблиць співпадають;

- об'єднання всіх записів з додаткової таблиці і тих записів основної таблиці, в яких зв'язані поля співпадають;
- об'єднання всіх записів з основної таблиці і тих записів додаткової таблиці, в яких зв'язані поля співпадають.

Якщо встановлення зв'язків проводиться для непорожніх таблиць, то Access піддає аналізу всю наявну інформацію і за наявності порушень цілісності повідомляє про це.

На Рал. 164 написи характеризують вид зв'язку (1:M). Лінія указується при об'єднанні записів, в яких зв'язані поля співпадають. Ознака контролю цілісності - потовщення закінчень зв'язуючих ліній.

Для зміни або видалення зв'язків треба підвести покажчик миші до лінії потрібного зв'язку, виділити її клацанням і натиснути праву кнопку. З'явиться контекстне меню, що пропонує змінити або видалити зв'язок. При виборі першого пункту з'являється вікно, показане на Рис. 164. Видалити зв'язок можна також, виділивши його мишею, а потім натиснувши клавішу **Del**.

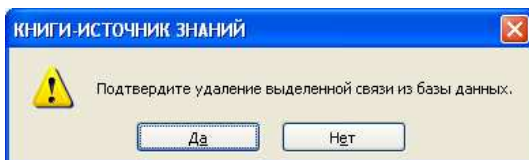


Рисунок 166

Якщо у вікні **Изменение связей** встановлені прапорці **Обеспечение целостности данных**, **каскадное обновление связанных полей**, **каскадное удаление связанных полей**, то зміни, що вносяться в головну таблицю, будуть автоматично вноситься в залежні таблиці. Наприклад, в головній таблиці

прізвище Петрова змінене на Петренко, а прізвище Сидоров вилучене. У всіх залежних таблицях відбудеться автоматична заміна Петрова на Петренко, а записи що містять прізвище Сидоров будуть автоматично вилучені. При спробі внести зміни в залежні таблиці раніше головних буде виведено повідомлення про неможливість внесення таких змін.

Завдання для самостійної роботи

1. Відкрийте базу даних **Моя бібліотека** і наповніть її таблиці наступними даними

Табл_Жанр
Жанр
Поэма

Табл_Автор
Автор
Тарас Шевченко

Табл_Книги		
Назва	Автор	Жанр
Гайдамаки	Тарас Шевченко	Поэма
Катерина	Тарас Шевченко	Поэма

2. Відкрийте



Схему даних. Вона має виглядати так як показано на рисунку.

Якщо це не так – встановить відповідні зв'язки між таблицями.

4.4. Створення запитів

За допомогою запиту можна:

- вибрати записи, які задовольняють умові відбору;
- включити в результуючу таблицю задані користувачем поля;
- провести обчислення для кожного запису;
- згрупувати записи з однаковими значеннями в один запис з одночасним виконанням над ними групових функцій;
- провести оновлення, додавання або видалення записів в таблицях;

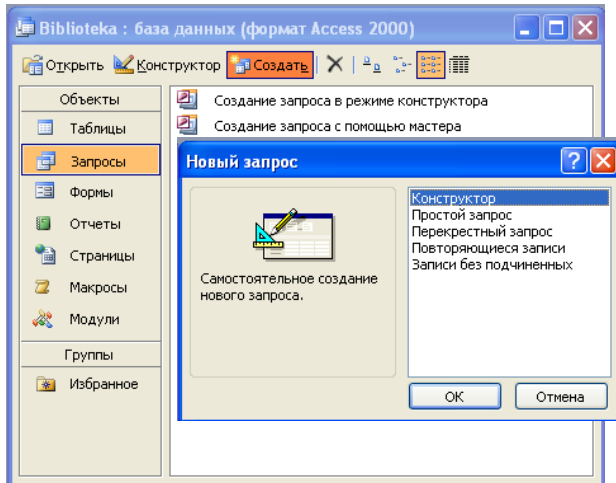


Рисунок 167

- створити нову таблицю, використовуючи дані з існуючих таблиць.
- Для створення запиту потрібно вибрати вкладку **Запросы** і натиснути кнопку **Создать**. Або вибрати в пункті **Вставка** головного меню підпункт **Запрос**. В результаті з'являється вікно (Рис. 167), в якому пропонується вибір варіанту створення запиту:
- уручну за допомогою конструктора - **Конструктор**;
 - автоматичне створення простого запиту на вибірку - **Простой запрос**;
 - автоматичне створення перехресного запиту для компактного представлення даних у вигляді зведеної (перехресної) таблиці - **Перекрестный запрос**;
 - автоматичне створення запиту на пошук записів із значеннями полів, що

повторюються, - **Повторяющиеся записи**;

- автоматичне створення запиту на пошук записів в одній таблиці, які не мають підлеглих записів в іншій таблиці - **Записи без подчиненных**.

Запити засновані на полях таблиць і/або запитів з БД. Всі способи, окрім першого, реалізуються за допомогою Майстрів, що спрощують розробку запиту. Якщо створений запит не задовольняє вимогам, то можна скористатися Конструктором, або створити заготовку запиту за допомогою Майстра, яку потім підправити в режимі Конструктора. Рекомендується створювати запити в режимі Конструктора. В цьому випадку при створенні запиту буде використана мова QBE, при необхідності цей запит можна уточнити на мові SQL. Access автоматично генерує код SQL при створенні запиту на QBE і навпаки.

Виклик **Конструктора** проводиться при створенні запиту або відкритті існуючого запиту і переводі його в режим Конструктора. У першому випадку перейти в режим Конструктора можна у вікні БД з вкладки **Запросы** двома способами: натисканням кнопки **Создать** і натисканням кнопки **OK** у вікні (Рис. 167), що з'явилося, або натисканням кнопки **Конструктор**. При виклику Конструктора в головному меню з'являється додатковий пункт **Запрос**, який має підпункти, що дозволяють виконувати різні операції в процесі створення запиту: виконання запиту, додавання таблиці в модель запиту, зміна виду запиту і т.д.

Складання запиту в режимі Конструктора включає визначення:

- таблиць і полів таблиць;
- виду запиту (вибірка, додавання, видалення, перехресний запит, SQL-запит);
- умов відбору записів;
- параметрів відображення результатів виконання запиту (показ полів, сортування значень).

Всі ці дії виконуються в запитальній формі (Рис. 168), яка відноситься до форми запиту на мові QBE.

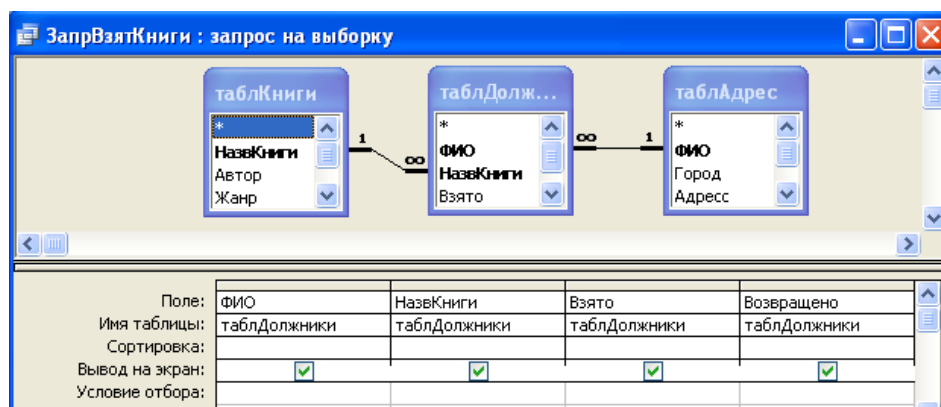


Рисунок 168

Запитальна форма включає три основні елементи: заголовок (ім'я і тип запиту); область таблиць, їх полів і зв'язків між таблицями; бланк запиту за зразком.

Для вказівки таблиць, які використовуються в запиті, потрібно помістити їх в запитальну форму і вказати зв'язки між ними. Включення об'єктів в запит проводиться у вікні **Добавление таблицы**, яке викликається автоматично (при створенні запиту) або примусово при роботі із запитом шляхом натиснення кнопки панелі інструментів.



При створенні запиту за умовчанням надається заготовка запиту на вибірку. Змінити вид запиту можна за допомогою пункту **Запрос** основного меню, де можливі види запитів перераховані як підпункти меню.

Умови відбору записів задаються в бланку у вигляді виразу як:

- *литералы* – конкретні значення, які сприймає Access так, як вони записані. Текстові рядки беруться в лапки: 565, "Інформатика";
- *константи* – незмінні значення: True, False, Null;
- *ідентифікатор* – посилання на значення поля, укладається в квадратні дужки: [ФІО];
- *оператори порівняння і логічні оператори* - =, <, >, <>, <=, >=, Between 10 And 100 – задає інтервал для числового значення або дати, In ("Математика"; "Фізика") виконує перевірку на рівність будь-якому значенню із списку, Like "*Математика" – дозволяє використовувати зразки, що використовують символи шаблону, при пошуку в текстових полях, Or, Not;

Приклад побудови запиту в режимі конструктора з умовою відбору приведений на Рис. 169.

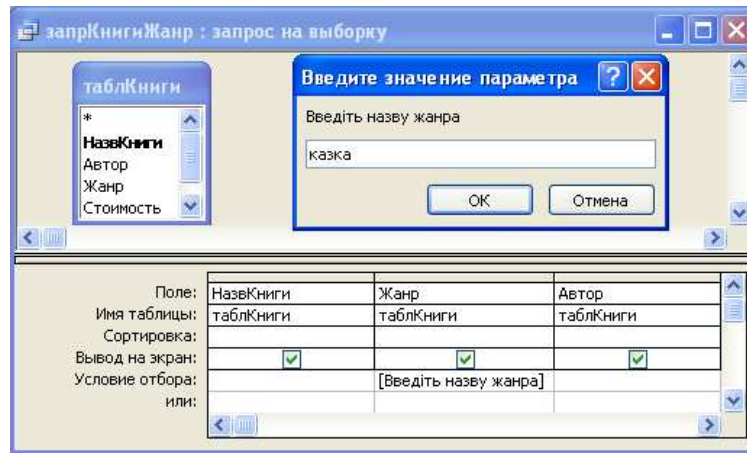


Рисунок 169

В результаті виконання цього запиту з'явиться діалогове вікно для введення жанру книги. Наприклад, при введенні "Казка" запит відобразить таблицю, в якій будуть тільки книги, що відносяться до жанру "Казка".

Для створення запиту у вигляді інструкції мови SQL можна за наявності на екрані запитальної форми скористатися пунктом меню **Запрос** → **Запрос SQL**, в якому вибрати потрібний вид SQL-запиту (на об'єднання, до сервера або управляючий). Якщо запит вже описаний в запитальній формі (QBE), вибір згаданого вище пункту приведе до перетворення запиту, якщо можливо, в речення мови SQL (Рис. 170).

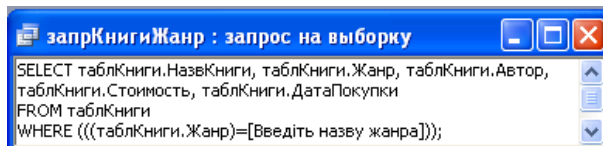


Рисунок 170

Завершити створення запиту можна:

- вибором пункту меню **Файл** → **Сохранить**;
- вибором пункту меню **Файл** → **Сохранить как/Экспорт**;
- натисненням кнопки **Сохранить** на панелі інструментів;

- шляхом закриття вікна створення запиту.

Групові операції дозволяють виділити в таблиці групи записів з однаковими значеннями у вказаних полях і використовувати для цих груп статистичну функцію:

- **Sum** - сума значень поля для групи;
- **Avг** – середнє від всіх значень поля в групі;
- **Max, Min** - максимальне і мінімальне значення поля в групі;
- **Count** - число значень поля в групі без урахування порожніх значень.

Для створення запиту з використанням групових операцій необхідно при відкритому в режимі Конструктора запиті вибрати пункт меню **Групповые операции** або натиснути однойменну кнопку на панелі інструментів.

Вікно конструктора запитів для складання запиту до *таблКниги* з угрупованням по полю *Жанр* і підсумовуванню вартості книг для кожної групи, і результат виконання цього запиту приведені на рис 171.

У запиті над полями можуть проводитися обчислення. Результат утворює обчислюване поле в таблиці

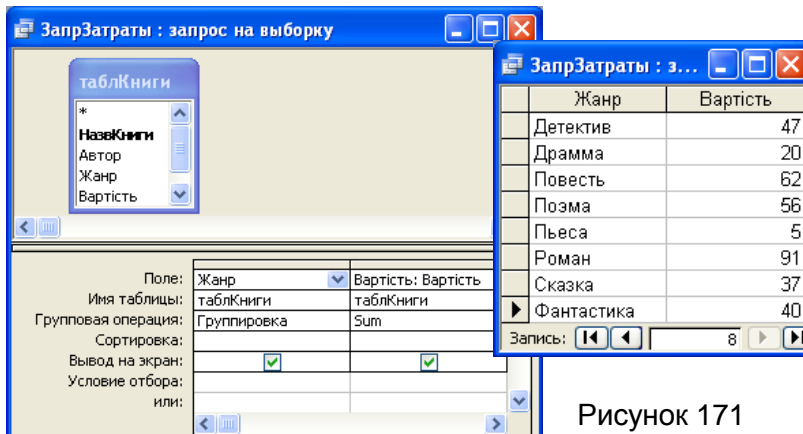


Рисунок 171

запиту. При кожному запуску запити проводяться обчислення з поточними значеннями полів.

Вираз вводиться в бланк запиту (рис 172) в порожнє поле рядка *Поле*. Після натиснення *Enter* перед введеним виразом додається ім'я поля. Наприклад, **Загальна вартість: [Кількість]*[ЦінаОд]**.

В Access визначено близько 150 вбудованих функцій, які можна використовувати в обчисленнях і умовах відбору. За призначенням їх можна згрупувати:

- **Функції дати и времени** – використовуються при обробці дат і часу. Повертають дату і час повністю або частково. Наприклад, функція Date формує поточну дату. Month – виділяє місяць із значення поля, що містить дату.
- **Функції обробки текста** – використовуються при роботі з символьними рядками.
- **Функції преобразования типа данных** – дозволяють задавати тип даних, щоб уникнути автоматичного підбору даних системою.
- **Математические и тригонометрические функции** – виконують операції над числами, які неможливо виконати за допомогою стандартних арифметичних операторів.
- **Финансовые функции** – служать для розрахунку відсотка повернення по інвестиціях, амортизаційних відрахувань і т.п.
- **Статистические функции** – використовуються для обчислення середнього, мінімального і максимального значень і т.д.

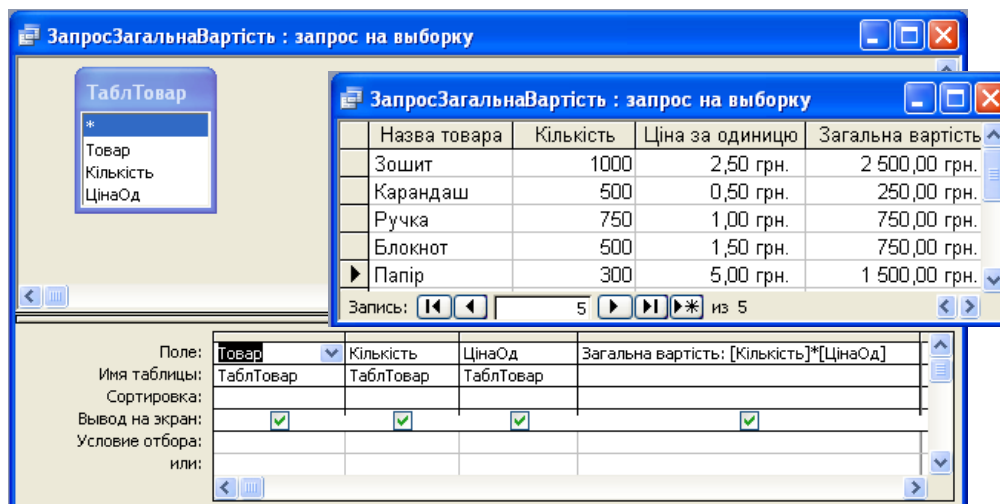


Рисунок 172

Для запису виразів можна використовувати **Построитель выражений** (Рис. 173), який викликається кнопкою **Построить**.

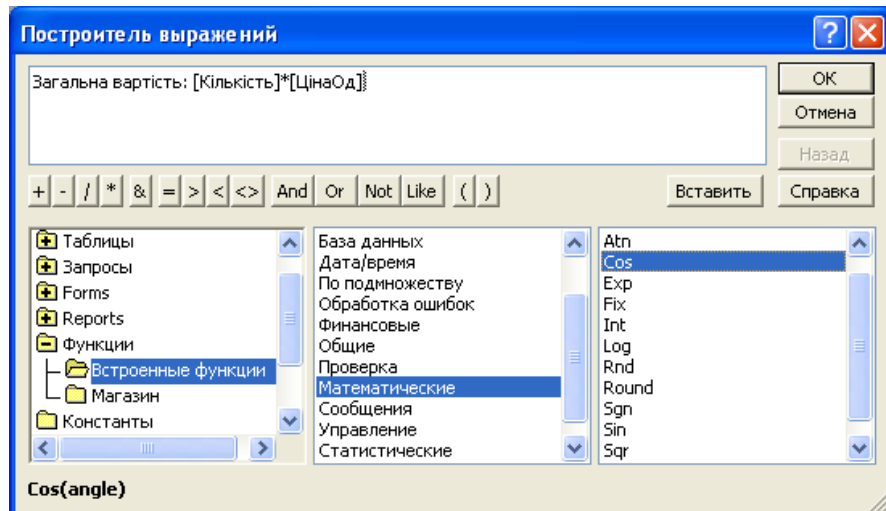


Рисунок 173

Завдання для самостійної роботи

1. В базі даних **Моя бібліотека** створіть запити на вибірку з **Табл_Книги** книг певного **автора** – **Запит_Автор** і книг певного **жанру** – **Запит_Жанр**.

4.7. Створення форм

Для створення форми - на вкладці **Формы** натиснути кнопку **Создать**. Почати створення форми можна також, вибравши, в пункті **Вставка** головного меню, підпункт **Форма**. З'являється вікно **Новая форма** (Рис. 174).

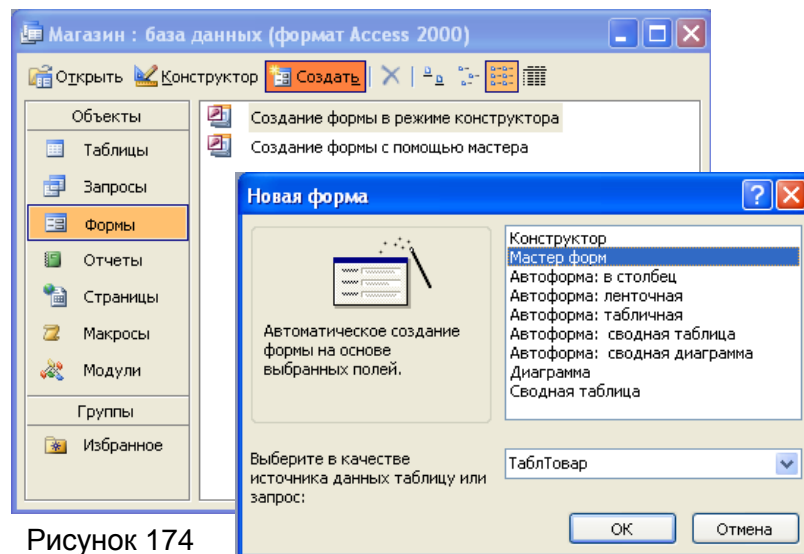


Рисунок 174

Варіанти створення форми:

1. за допомогою Конструктора - **Конструктор**;
2. за допомогою Майстра - **Мастер форм**;
3. автоматичне створення форми стандартного вигляду, в якій по-різному розміщуються поля
 - **Автоформа: в столбец**;
 - **Автоформа: в ленточная**;
 - **Автоформа: табличная**;
4. автоматичне створення форми стандартного вигляду із зведеною таблицею;
5. створення форми із зведеною таблицею Microsoft Excel - Сводная таблица;
6. створення форми з діаграмою - Диаграмма;
7. створення форми із зведеною таблицею.

Рекомендується створювати форми з використанням **Майстра форм** і потім удосконалити їх в режимі **Конструктора**.

Найпростіше створити форму по варіанту 3-4. У цих випадках автоматично без діалогу з користувачем

виходять нескладні форми, що включають всі поля джерела даних (таблиці або запиту). Після створення форма доступна для переглядання/редагування даних.

Запустивши, Майстер (варіант 2), користувач може створити форми таких же видів, як і при виборі варіантів 3-4. Але тут можна в діалозі вибрати як джерела даних довільне число запитів і/або таблиць, включивши у форму потрібні поля. Крім того, можна змінити стиль фонового зображення. При необхідності провести інші зміни макету форми, після завершення роботи з Майстром, можна перейти в режим Конструктора.

Режим Конструктора є найбільш потужним, але трудомістким засобом розробки форм. Щоб скоротити загальний час розробки, доцільно перед викликом Конструктора скористатися одним з інших способів створення форми, а Конструктор використовувати для приведення форми до потрібного вигляду.

Виклик Конструктора форм проводиться при створенні форми або відкритті існуючої форми і переключенні її в режим Конструктора.

Форма в режимі Конструктора (Рис. 175) в загальному випадку містить області: *заголовок* і *примечание форми*, *верхній і нижній колонтитул* і *область даних* (у якій відображаються дані джерел). Обов'язковою є область даних, інші - необов'язкові. У області даних звичайно розміщуються поля таблиць.

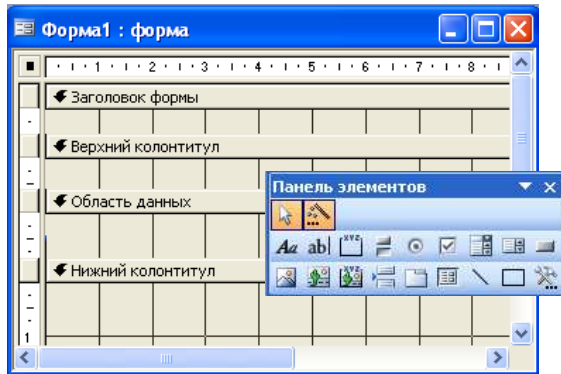


Рисунок 175

Управляти відображенням необов'язкових областей можна за допомогою пункту меню **Вид**, знімаючи і встановлюючи відповідні прапорці. Можливості Конструктора визначаються складом інструментів **Панель елементов** (Рис. 175). За допомогою кожного з наведених інструментів можна побудувати форму яка б найбільше відповідала поставленим вимогам: наявність підписів, пояснень, списків, кнопок навігації та ін. елементів, що разом роблять введення та редагування даних у БД більш зручним.

При розробці звітів в режимі Конструктора використовується аналогічна панель елементів. Звідси - уніфікована назва окремих інструментів (наприклад, Подчиненная форма/отчет), а також текст довідки, що описує роботу з формами і із звітами. Більш того, Access дозволяє форму перетворити в звіт.

На Рис. 174. приведений приклад форми для заповнення полів таблиці.

Якщо форми використовуються для відображення інформації, а не для введення даних, то відповідно поля форми необхідно блокувати. У вікні властивостей на вкладці дані для властивості **Блокировка** встановлюється значення **Да**. Аналогічно можна заборонити або вирішити додавання, зміну або видалення даних.

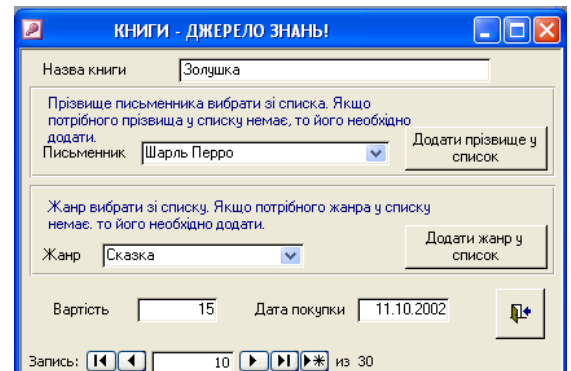


Рисунок 176

Завдання для самостійної роботи

1. В базі даних **Моя бібліотека** створіть наступні форми:

- для заповнення Табл_Автор;
- для заповнення Табл_Жанр;
- для заповнення Табл_Книги;
- для виведення результатів запитів **Запит_Автор** і **Запит_Жанр**.

4.7. Створення звітів

Створення звіту проводиться шляхом натиснення кнопки **Создать** у вікні БД на вкладці **Отчеты**, або задавши в пункті **Вставка** головного меню команду **Отчет**. З'являється вікно Рис. 177.

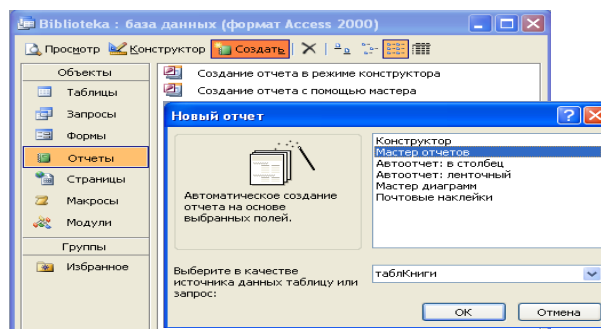


Рисунок 177

Варіанти створення звіту:

1. За допомогою Конструктора - **Конструктор**;
2. За допомогою Майстра - **Мастер отчетов**;
3. Автоматичне створення звіту стандартного вигляду, в якому кожен запис базового запиту або таблиці

- представлений у вигляді назви і значення поля, - **Автоотчет: в столбец**;
4. Автоматичне створення стандартного звіту, в якому дані записи базового запиту або таблиці виводяться в одному рядку, - **Автоотчет: ленточный**;
 5. Створення звіту з діаграмою - **Диаграмма**;
 6. Створення звіту для друку поштових наклейок - **Почтовые наклейки**.

Простіше всього створити звіт по варіантах 3 і 4. У цих випадках виходять прості звіти, які включають всі поля джерела даних (таблиці або запиту). Після створення звіт доступний для перегляду або друку. У режимі Конструктора надаються потужніші засоби, які вимагають більше знань і часу для розробки звіту. Але звіт виходить якіснішим.

Робота з Конструктором звітів мало відрізняється від роботи з Конструктором форм. Зокрема - використовується така ж панель елементів (мал. 175).

У звіті використовуються ті ж області, що і у формах: області заголовка і примітки, області верхнього і нижнього колонтитулів, а також область даних. Крім того, в звіті можна включати області угруповання записів.

Заголовок і примітка виводяться один раз: на початку і кінці звіту відповідно.

Верхній/нижній колонтитули поміщаються в початок/кінець кожної сторінки. Вміст області даних виводиться один раз для кожного запису початкової таблиці або запиту. Якщо задане угруповання записів звіту, то по кожному полю, по якому проводиться угруповання даних, Access формує заголовок і примітку групи.

Для створення в звіті області угруповання необхідно при відкритому в режимі Конструктора звіті вибрати пункт меню **Вид/Сортировка и группировка** або натиснути однойменну кнопку на панелі інструментів.

Істотна відмінність між звітом і формою в тому, що звіти призначені виключно для виведення даних на друк. Тому в них можна відмовитися від використання (доступних для включення в звіт) елементів, що управляють, для введення даних: списків, полів із списком, перемикачів і т.п.

Звіти можуть знаходитися в двох режимах: Конструктора або Перегляду. Проглянути готовий звіт можна після виділення потрібного звіту при натисненні кнопки **Просмотр** у вікні БД, виборі пункту меню **Файл**.

Завдання для самостійної роботи

1. В базі даних **Моя бібліотека** створіть звіт для виводу на друк даних з **Табл_Книги**.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Ващук Ф. Г. Нелюбов В. О. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРА Навчальний посібник, ЗақДУ, Ужгород – 2007
2. Эффективная работа: Windows XP / Э. Ботт, К. Зигерт. – СПб.: Питер, 2004. – 1069 с.: ил.
3. Омельченко Л.Н., Федоров А.Ф. Самоучитель Microsoft Windows XP. СПб.: БХВ-Петербург, 2002. – 560 с.: ил.
4. Бекаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2002. СПб.: БХВ-Петербург, 2002. – 720 с.: ил.