

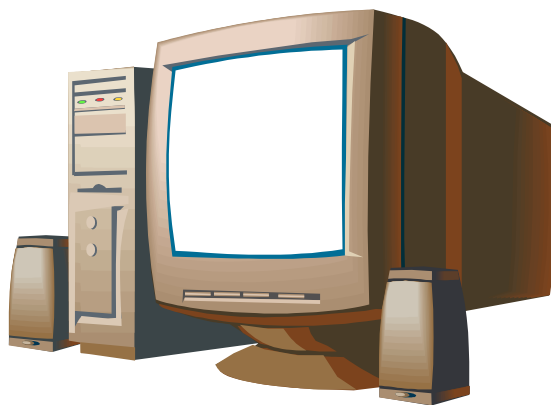
Міністерство освіти і науки України

Полтавський національний технічний університет
імені Юрія Кондратюка

Кафедра комп'ютерних та інформаційних технологій і систем

ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ

*з дисципліни „Інформатика та комп'ютерна техніка” (модуль 6)
для студентів спеціальності „Облік та аудит” всіх форм навчання*



Полтава 2006

Опорний конспект лекцій з дисципліни „Інформатика та комп’ютерна техніка” (модуль 6) для студентів спеціальності „Облік та аудит” всіх форм навчання. – Полтава: ПолтНТУ, 2006.—28с.

Укладач: Т.А. Дмитренко, ст. викладач

Відповідальний за випуск: завідувач кафедри комп’ютерних та інформаційних технологій і систем О.Л. Ляхов, доктор техн. наук, доцент

Рецензент: Т.М. Деркач, канд. техн. наук, доцент

Конспект розглянуто і схвалено на засіданні кафедри
Протокол № 7 від 8 листопада 2006 р.

ЗМІСТ

Лекції № 35 Тема: Реляційні бази даних.	4
1. Способи об'єднання таблиць у запитах	4
2. Використання полів, що обчислюються	4
Лекції № 36 Тема: Реляційні бази даних.	6
1. Автоматизація роботи додатка за допомогою макросів	6
2. Створення простого макросу	7
3. Перевірка роботи макросу	9
4. Створення складного макросу	9
5. Групи макросів	10
6. Завдання умов виконання макрокоманд	10
Лекції № 37 Тема: Реляційні бази даних.	12
1. Загальні зведення про елементи керування	12
2. Елементи керування	12
3. Загальні зведення про написи	13
4. Загальні зведення про поля	13
5. Загальні зведення про групу параметрів	13
6. Загальні зведення про вимикачі	14
7. Загальні зведення про прапорці	14
8. Загальні зведення про поля зі списком	15
9. Загальні зведення про списки	15
10. ВКЛАДКА ДАНІ	17
11. ВКЛАДКА МАКЕТ	18
Лекції № 38 Тема: Реляційні бази даних.	18
1. Создание составных форм	18
2. Убудовані елементи керування	20
3. Різновиду елементів керування по типі вмісту	20
Лекції № 39 Тема: Реляційні бази даних.	21
1. Створення і друк звітів	21
2. Автоматичне створення звіту на основі таблиці або запиту	22
3. Створення звіту за допомогою Майстра звітів	22
4. Підготовка складених звітів	26
5. Створення складених звітів	26
6. Застосування підлеглих звітів	26

Лекції № 35 Тема: Реляційні бази даних.

План лекції:

1. Способи об'єднання таблиць у запитах
2. Використання полів, що обчислюються

1. Способи об'єднання таблиць у запитах

Коли запит створюється на базі декількох таблиць, говорять, що ці таблиці поєднуються в запиті. Причому можна використовувати кілька видів об'єднань.

Внутрішнє об'єднання: У результуючий набір попадають ті записи з двох зв'язаних таблиць, у яких зв'язані поля збігаються.

Якщо дві таблиці зв'язані відношенням один-ко-многим, то в результаті запиту включаються всі записи з головної таблиці, для яких мають відповідні записи в підлеглій таблиці. Якщо запис у головній таблиці не має відповідних записів у підлеглій таблиці, цей запис у результуючу безліч не включається. Такі об'єднання створюються автоматично, коли:

З'єднання один-ко-многим у вікні Схема даних

У таблиці мають поля з однаковими іменами і погодженими типами, причому одне з полів є ключовим.

Приклад:

відкрийте таблицю Видання в режимі таблиці і внесіть у неї книгу, що не має автора - Російські биліни і перейдіть на наступний запис.

Закрити таблицю.

Відкрити запит Книги по авторах, що ми створили раніше в режимі Конструктора.

Виконати запит.

Виконати пошук по полю Назва (Росіяни биліни)

Книга не знайдена, тому що вона не має зв'язаних записів у підлеглій таблиці.

Щоб усунути помилку потрібно використовувати так назване зовнішнє об'єднання.

Ввійти в режим Конструктора.

Виділити зв'язок між таблицею Видання й Авториздание і потім двічі клацнути на ній.

З'явиться вікно Параметри об'єднання. Вибрати другий вид об'єднання.

Клацнути двічі на лінії, що з'єднує таблиці Автор і Авториздание і ввести Параметр об'єднання 3.

Виконати запит.

2. Використання полів, що обчислюються

Поля, значення яких є результатом обчислення вираження – що обчислюються. Ці поля відображають дані, розраховані на основі значень інших полів цього ж рядка запиту.

Поле автор у запиті Книги по авторах

Режим Конструктора.

У порожньому стовпці установити курсор у перший рядок нового стовпця ввести :Автор: [Прізвище]&” ”&[Ім'я]

Перша частина до : - ім'я нового поля.

Друга частина – вираження, значенням якого є злиття полів Прізвище й Ім'я. Між ними потрібно ставити пробіл, інакше при висновку на екран ім'я і прізвище автора будуть відображатися разом.

Установити сортування: по зростанню

Зберегти запит.

Створимо запит, що покаже скількох днів тримають на руках книги наші читачі і знайдемо тих, хто затримує їхній більше місяця.

Відкрити таблицю Формуляри.

Клацнути на панелі інструментів, на кнопці Новий об'єкт і вибрати Запит.

Виділити в списку елемент Конструктор і ОК.

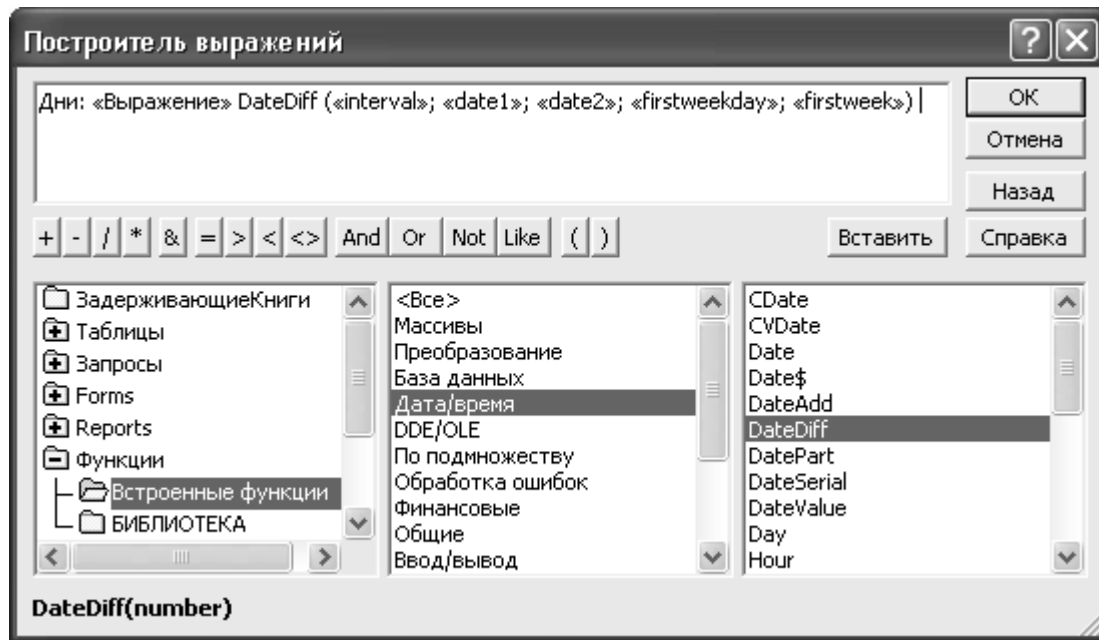
Перенести в бланк запиту поля Кодчитателя, Датаполучения, Датавозвратафакт.

Зберегти запит через панель інструментів з ім'ям Затримуючі книги.

У вільному стовпці ввести Дні: і натиснути кнопку Построитель виражень.

У першому списку відкрити Функції>Убудовані функції.

У категоріях вибрати Дата/час>DateDiff.



Натиснути мишею на елемент «interval» він буде виділений. Уведіть “d”. Це означає, що різниця між датами буде обчислюватися в днях.

Виділити аргумент: «date1».

Відкрити папку Затримуючі книги в першому списку.

Двічі клацнути на поле Датаполучения в середньому списку.

Виділивши аргумент «date2» увести Датавозвратафакт із другого списку.

Зайві слова «Вираження» видалити.

ОК.

Введіть умову добору по стовпці Дні. Для формулярів у які книга були на руках більш 30 днів.

Запустити запит.

Створення запиту з параметрами

Щоб переглянути книги до приклада будь-якого автора потрібно створити запит у якому прізвище автора буде параметром, запитуваним при виконанні.

Відкрити запит Книги по авторах у режимі Конструктора.

У рядок Умови добору для стовпця Прізвище вводиться [Прізвище автора:]

У результаті при запуску запиту буде з'являтися діалогове вікно Прізвище автора:

Аналогічно створити запит на вибірку на основі таблиць Автори-АвторИздание-Издание). Поля: Ім'я, Прізвище, Ціна, ТипОбложки, ДатаПокупки. У поле Датапокупки ввести умова добору Between[Початок періоду:] And[Кінець періоду:]

Використання статистичних функцій у запиті

У запиті може виконуватися підрахунок підсумкових значень. Наприклад, для числового поля можна обчислити середнє значення або суму значень для всього або відібраного записів, можна підрахувати кількість записів, що повертаються запитом і т.д. У цих запитах використовуються статистичні функції.

Для просмотра фвойств формы нужно выделить всю форму. Правка → Выделить форму. Или щелкнув левой кнопкой мыши на квадратике в левом верхнем углу окна Конструктора формы, где сходятся вертикальная и горизонтальная линейки. Когда форма выделена, в области выделения появляется маленький черный квадратик.

Лекції № 36 Тема: Реляційні бази даних.

План лекції:

1. Автоматизація роботи додатка за допомогою макросів
2. Створення простого макросу
3. Перевірка роботи макросу
4. Створення складного макросу
5. Групи макросів
6. Завдання умов виконання макрокоманд

1. Автоматизація роботи додатка за допомогою макросів

У Microsoft Access можна визначити **макрос**, що виконує практично тієї ж дії, який можна реалізувати натисканням клавіш на клавіатурі або за допомогою миші. Основною перевагою макросів у Access є те, що вони можуть виконуватися у відповідь на багато видів подій.

Подією може бути зміна даних, відкриття і закриття форми або звіту, а також передача фокуса від одного елемента керування, іншому. У макрос можна включити кілька макрокоманд і задати перевірку умов таким чином, що в залежності від значень даних у формах або звітах будуть виконуватися різні дії.

Microsoft Access надає різні **типи макрокоманд**, що дозволяють автоматизувати роботу додатка. Розглянемо основні дії, виконувані за допомогою макрокоманд:

- Відкриття будь-якої таблиці, запиту, форми або звіту в будь-якому доступному режимі виконується макрокомандами: *ОткрытьТаблицу*, *ОткрытьЗапрос*,

ОткрытьФорму, ОткрытьОтчет. За допомогою макрокоманди *Закрити* можна закрити будь-яку відкриту таблицю, запит, форму або звіт.



- Для запуску інших макросів використовується макрокоманда *ЗапускМакроса*. Виконання процедури або функції VBA здійснюється макрокомандою *ЗапускПрограми*. Щоб перервати виконання поточного макросу або всіх макросів можна використовувати *ОстановитьМакрос* або *ОстановитьВсеМакросы*.
- Для визначення спеціального рядка меню використовується макрокоманда *ДобавитьМеню*. Можна зробити доступним або не доступним, зняти галочку або позначити пункт спеціального меню, у тому числі і контекстного за допомогою макрокоманди *ЗадатьКомандуМеню*. Виконати будь-як команду одного зі стандартних команд меню Access дозволяє макрокоманда *КомандаМеню*.
- Для переміщення, зміни розмірів, згортання або відновлення будь-якого вікна усередині робочої області Access використовуються макрокоманди *СдвигРазмер, Згорнути, Розгорнути, Відновити*. Щоб передавати будь-як елементів керування у вікні використовується *КЭлементуУправления*.
- Для висновку на екран інформаційних повідомлень і подачі звукових сигналів для залучення уваги до ваших повідомлень використовуються макрокоманди *Повідомлення і Сигнал*.
- Для перейменування будь-якого об'єкта бази даних, копіювання обраного об'єкта в поточну або іншу базу даних Access, збереження або видалення об'єктів з вашої бази даних використовуються макрокоманди *Перейменувати, КопироватьОбъект, Зберегти, УдалитьОбъект*. Крім того, макрокоманди *ПреобразоватьБазуДанных, ПреобразоватьЭлектроннуюТаблицу, ПреобразоватьТекст* дозволяють імпортувати, експортувати або зв'язувати таблиці з інших баз даних, імпортувати або експортувати електронні таблиці і текстові файли.

Розглянемо інші можливості макросів. Командні кнопки, що відкривають і розміщають форми і встановлюють значення полів, допоможуть переключатися з однієї задачі на іншу. Ви можете створити дуже складні процедури редагування, що перевіряють правильність даних, що вводяться у форму, включаючи перевірку даних в інших таблицях. Наприклад, можна організувати перевірку імені клієнта, що вводиться у форму замовлення, і відкривати іншу форму, щоб користувач уводив додаткові зведення, якщо в базі не існує запису з інформацією про клієнта.

Макроси особливо корисні для побудови невеликих персональних додатків або створення прототипів великих додатків. Для складних додатків або додатків, призначених для роботи в мережі в режимі колективного користування, вам швидше за все прийдесться використовувати VBA. Але навіть якщо ви вважаєте, що готово відразу перейти до VBA, краще спочатку вивчити макрокоманди. Ви побачите, що майже усі з них прийдесться використовувати в VBA. Тому вивчення макросів – прекрасне введення в програмування в Access у цілому.

2. Створення простого макросу

У вікні бази даних перейдіть на вкладку *Макроси* і натисніть кнопку *Створити*, щоб відкрити вікно нового макросу, подібне тому, що показано на мал. 6.14. Верхня частина вікна використовується для визначення нового макросу, а нижня призначена для введення значень аргументів макрокоманд, включених у макрос. У верхній частині

присутні принаймні два стовпці з заголовками *Макрокоманда* і *Примітка*. Ви можете побачити всі чотири стовпці, показані на мал. 14, натиснувши кнопки *Імена макросів* () і *Умови* () на панелі інструментів.

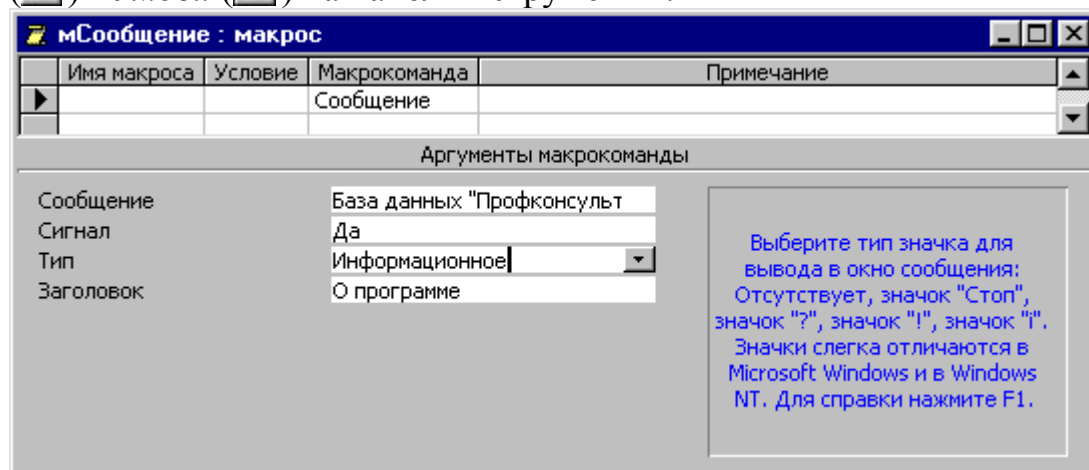


Рис. 6.14. Вікно макросу *мСообщение*

У правій нижній частині вікна макросу виводиться коротка довідка. Зміст повідомлення міняється в залежності від положення курсору у верхній частині вікна макросу (помніть: для одержання контекстної довідки завжди можна натиснути клавішу F1).

У стовпці *Макрокоманда* задається одна з більш ніж 49 макрокоманд, наданих Access. Якщо ви клацніть у будь-якому осередку стовпця *Макрокоманда*, то в правому кінці осередку з'явиться кнопка зі стрілкою вниз. Натискання цієї кнопки відкриває список макрокоманд.

Щоб познайомитися з роботою вікна макросу, виберіть у списку макрокоманду *Повідомлення*. Вона використовується для відкриття монопольного вікна діалогу з повідомленням. Це прекрасний спосіб для висновку в додатку різного роду попереджуючих або інформаційних повідомлень без створення для цього спеціальної форми.

Стовпець *Примітка* особлива корисний для документування складних макросів, що містять велике число макрокоманд. У стовпці *Примітка* можна вводити додаткові коментарі, розміщаючи них на порожніх рядках, не утримуючих макрокоманд.

Після вибору макрокоманди, такий як *Повідомлення*, Access виведе в нижній частині вікна макросу її аргументи.

Як і у випадку роботи з вікнами таблиці і запиту в режимі конструктора, для переходу з верхньої частини вікна макро в нижню і назад ви можете скористатися клавішею F6.

Значення аргументу *Повідомлення* являє собою текст, що Access буде виводити у вікні діалогу. Аргумент *Сигнал* задає, чи буде звуковий сигнал супроводжувати появу вікна діалогу. Аргумент *Тип* дозволяє вставити у вікно повідомлення значок, наприклад червоне коло з білим перекрестием. В осередок аргументу *Заголовок* уводиться текст, що з'явиться в заголовку вікна діалогу. Використовуйте в цьому макросі установки, приведені на мал. 6.14. (Повний текст повідомлення: *База даних «Профконсультант середньої школи»*).

Перед запуском макрос необхідно зберегти. Виберіть команду *Файл-> Зберегти* (або *Файл->Зберегти як/Експорт*). Access відкриє вікно діалогу. Введіть ім'я *МакросСообщение* і натисніть кнопку ОК, щоб зберегти макрос.

3. Перевірка роботи макросу

Перед запуском макросу непогано перевірити його роботу, виконавши макрокоманди в покроковому режимі. Щоб почати покрокову перевірку, перейдіть у вікно бази даних, на вкладці *Макроси* виділіть ім'я макросу, що ви хочете протестувати, і натисніть кнопку *Конструктор*. Після відкриття вікна макросу натисніть кнопку *По кроках* на панелі інструментів або виберіть команду *Запуск->По кроках*.

Тепер після запуску макросу Access буде відкривати вікно діалогу *Покрокове виконання макросу* перед виконанням кожного кроку. У цьому виводиться ім'я макросу, назва макрокоманди, умова її виконання й аргументи макрокоманди.

Якщо в макросі визначено більш однієї макрокоманди, то після першого кроку ви знову повернетесь до вікна діалогу *Покрокове виконання макросу*, що покаже наступну готову до виконання макрокоманду.

Якщо під час виконання макросу в якій-небудь макрокоманді зустрінеться помилка, то Access спочатку виведе вікно діалогу, що пояснює неї. Потім з'явиться вікно діалогу *Помилка макрокоманди* з інформацією про макрокоманду, що викликала помилку. У цей момент можна натиснути тільки кнопку *Перервати* і відредагувати макрос, щоб усунути причину помилки.

4. Створення складного макросу

У Microsoft Access усередині одного макросу ви можете задати кілька макрокоманд і порядок їхнього виконання. Розглянемо створення макросу з ім'ям *Autoexec*. Microsoft Access буде запускати цей макрос при відкритті бази даних, у якій він зберігається. Щоб уникнути запуску макросу *Autoexec*, при відкритті бази даних можна натиснути клавішу shift. У цьому макросі визначені вісім макрокоманд, автоматично виконуваних Access при відкритті цієї бази даних. Перша макрокоманда *ПесочныеЧасы* виводить покажчик у виді піскових годин, щоб показати користувачеві, що наступні кроки можуть зайняти кілька секунд. Аргумент *Уключити висновок* наступної макрокоманди *ВыводНаЭкран* встановлений у значення *Ні*, так що під час виконання макросу ви не побачите на екрані зайвих зображень. Третя макрокоманда *ЗапускМакроса* запускає інший макрос *мСообщение*. Наступна макрокоманда *Открытьформу* відкриває форму *фПрофконсультант*. Як показано на мал. 6.15, роботу цієї макрокоманди визначають чотири аргументи: *Ім'я форми* задає форму, що відкривається, *Режим* встановлює режим, у якому форма буде відкрита. (Цей аргумент може приймати чотири значення: *Форма*, *Конструктор*, *Перегляд і Таблиця*). За замовчуванням для аргументу *Режим даних* встановлюється значення *Зміна*, що дозволяє користувачеві додавання, видалення і зміна записів під час роботи з формою. (Цей аргумент може приймати наступні значення: *Додавання*, *Зміна* і *Тільки читання*.) Для аргументу *Режим вікна* за замовчуванням встановлюється значення *Звичайне*. У цьому випадку форма відкривається в режимі, заданому її властивостями.

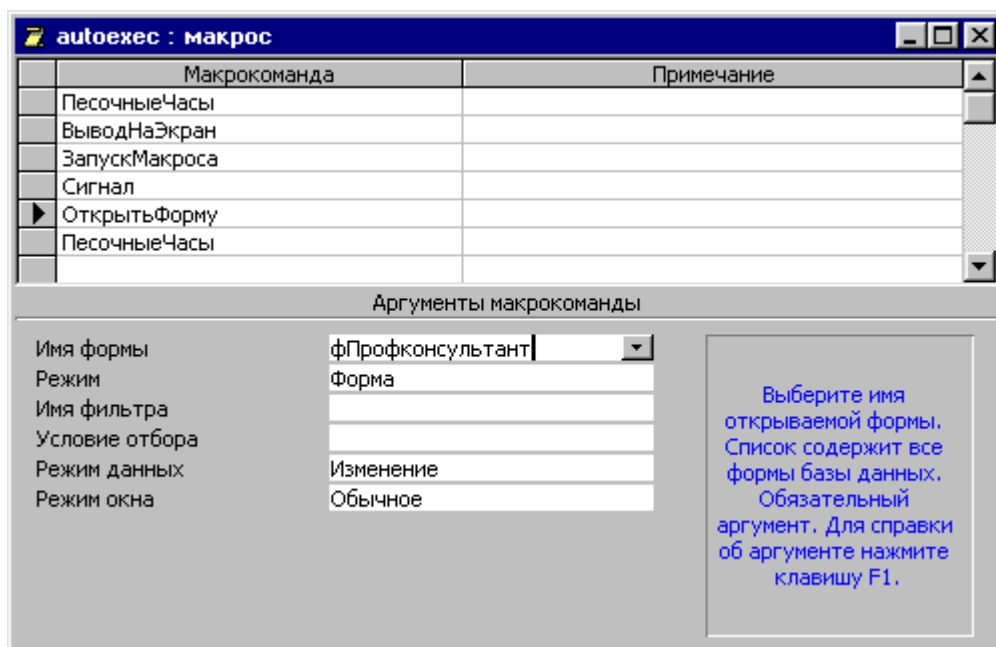


Рис. 6.15. Макрос *Autoexec*, що запускається автоматично при відкритті бази даних. Потім макрокоманда *Розгорнути* збільшує активне вікно до розмірів робочої області Access. Остання макрокоманда (це знову *ПесочныеЧасы*) відновлює колишній вид покажчика миші, щоб користувач знав про закінчення роботи макросу. Отже, використання в макросі декількох макрокоманд дозволяє автоматизувати задачі, що приходиться виконувати регулярно.

5. Групи макросів

У більшості форм, створюваних для додатка потрібне застосування значного числа макрокоманд. Одні використовуються для редагування полів, інші відкривають звіти, треті реагують на натискання командних кнопок. Можна створити макроси для кожної окремої операції, але в цьому випадку в додатку нагромадяться сотні різних макросів. Існує більш простий спосіб, при якому для кожної форми або звіту створюється своя група макросів. Можливий також і інший підхід, при якому групи макросів створюються для кожного типу операцій. Наприклад, можна створити групу макросів, що містить усі макрокоманди *ОткрытьФорму*, використовувані в базі даних. У групу можна включити кілька окремих макросів. Кожному макросові в групі дається ім'я, що заноситься в стовпець *Ім'я макросу*. Як ви вже бачили на прикладі макросу *Autoexec*, за допомогою макрокоманди *ЗапускМакроса* можна виконати інший макрос. Точно так само Access виконає конкретний макрос із групи, якщо в макрокоманді *ЗапускМакроса* вказати ім'я групи макросів, крапку й ім'я макросу. На початку групи макросів можна помістити кілька макрокоманд, що будуть виконуватися при виклику групи без вказівки імені макросу. Ім'я, зазначене при збереженні групи, з'явиться в списку макросів у вікні бази даних.

6. Завдання умов виконання макрокоманд

Іноді необхідно, щоб макрокоманди в макросі виконувалися тільки за певних умов. Наприклад, ви хочете оновлювати запис лише в тому випадку, якщо нові дані в елементах керування форми задовольняють умовам на значення або хочете виводити або ховати визначені елементи керування в залежності від значень інших елементів керування.

Розглянемо макрос *мПросмотр*, що містить умови для перевірки необхідності виконання визначених макрокоманд (див. мал. 6.16). Цей макрос запускається за допомогою командної кнопки *Перегляд* у формі *фПечать*. Ця форма дозволяє користувачеві переглянути і роздрукувати визначений звіт, установлюючи відповідний перемикач і натискаючи кнопки *Перегляд* і *Печатка* (див. мал. 6.17). Перемикачі знаходяться усередині одного елемента керування, що називається групою. Кожен перемикач установлює визначене числове значення (перший установлює 1, другий – 2, третій – 3, четвертий – 4) для цього елемента керування, що перевіряється в стовпці *Умова*.

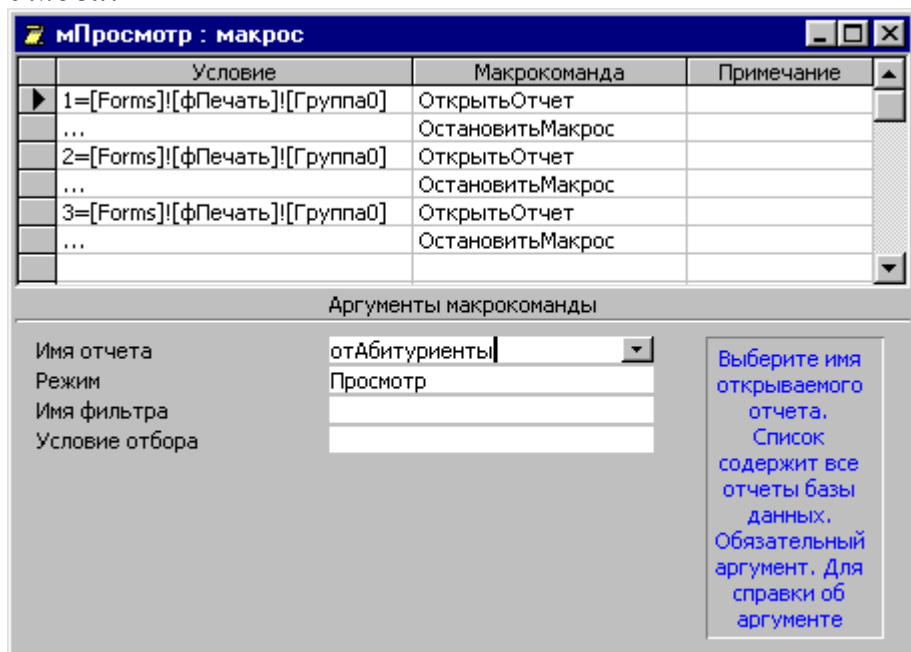


Рис. 6.16. Умови для макросу *мПросмотр*

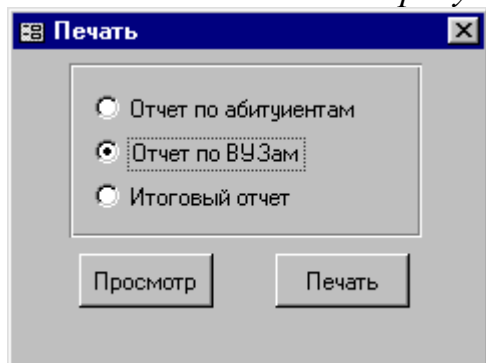


Рис. 6.17. Форма *фПечать*

При включенні умови в макрос Access виконує макрокоманду в цьому рядку тільки в тому випадку, якщо умовне вираження має значення *Истина*. Уведення многоточия (...) у стовпці *Умова* в наступних рядках дозволяє поширити дія умови на ряд наступних макрокоманд. У цьому випадку Access обчислює умовне вираження тільки один раз і виконує групу макрокоманд, якщо умова, задана для першої макрокоманди в цій групі, є щирим.

У даному випадку умови використовуються для перевірки значення групи перемикачів *Группа0* у формі.

Використання груп макросів дозволяє легко знаходити потрібні макроси. Ви можете створювати групи макросів не тільки по задачах, але і по функціях. Припустимо, що мається форма з поруч командних кнопок, кожної з яких призначений макрос, що виконує визначену задачу. Ви можете зібрати ці макроси в одну групу і дати їй ім'я, що

вказує на зв'язок з формою. Так, якщо форма *Клієнти* містить кілька командних кнопок, то можна створити групу макросів з ім'ям *МакросыФормыКлиенты*. У цю групу варто включити макроси, призначені командним кнопкам у цій формі.

Лекції № 37 Тема: Реляційні бази даних.

План лекції:

1. Загальні зведення про елементи керування
2. Елементи керування
3. Загальні зведення про написи
4. Загальні зведення про поля
5. Загальні зведення про групу параметрів
6. Загальні зведення про вимикачі
7. Загальні зведення про прапорці
8. Загальні зведення про поля зі списком
9. Загальні зведення про списки
10. ВКЛАДКА ДАНІ
11. ВКЛАДКА МАКЕТ

1. Загальні зведення про елементи керування

Усі зведення у формі або звіті утримуються в елементах керування. Елементи керування — це об'єкти форми або звіту, що служать для висновку даних на екран, виконання макрокоманд або оформлення форми або звіту. Наприклад, поле можна використовувати для висновку даних на екран у формі або звіті, кнопку — для відкриття іншої форми або звіту, а лінію або прямокутник — для поділу й угруповання елементів керування для того, щоб вони краще сприймалися користувачем.

У Microsoft Access існують наступні типи елементів керування, що утримуються на панелі елементів у режимі конструктора форми або режимі конструктора запиту: поле, напис, група, перемикач, прапорець, вимикач, поле зі списком, список, кнопка, малюнок, приєднана рамка об'єкта, вільна рамка об'єкта, набір вкладок, підлегла форма/звіт, розривши сторінки, лінія, прямокутник і додаткові елементи Active.

Елементи керування можуть бути зв'язаними, вільними або що обчислюються. Зв'язаний елемент керування приєднаний до поля базової таблиці або запиту. Такі елементи керування використовуються для відображення, введення або відновлення значень з полів бази даних. Для елемента керування, що обчислюється, як джерело даних використовується вираження. У вираженні можуть бути використані дані з поля базової таблиці або запиту для форми або звіту, а також дані іншого елемента керування форми або звіту. Для вільного елемента керування джерела даних не існує. Вільні елементи керування використовуються для висновку на екран даних, ліній, прямокутників і малюнків. На приведеному нижче малюнку показані приклади зв'язаного, вільного й елементів керування, що обчислюється.

2. Елементи керування

Напису

Поля

Групи

Вимикачі

Перемикачі

Прапорці

Поля зі списком

Списки

Кнопки

Малюнки, вільні і зв'язані рамки об'єкта

Підлеглі форми

Підлеглі звіти

Розриви сторінок

Лінії

Прямокутники

Набори вкладок

Елементи HTML

Елементи Active

3. Загальні зведення про написи

Написи призначені для відображення у формі або звіті описових текстів, таких як заголовки, підписи або короткі інструкції. У написах не виводяться значення полів або виражень; вони завжди є вільними і не міняються при переході від запису до запису.

Напис може бути приєднана до іншого елементові керування (такий напис називають підписом). Наприклад, поле створюється з приєднаним написом, що містить підпис цього поля. Цей напис з'являється як заголовок стовпця у формі в режимі таблиці.

Напис, створений за допомогою інструмента «Напис», розміщається окремо і не приєднується ні до якого елемента керування. Такі написи використовуються для відображення різних зведень, наприклад, заголовків форми або звіту, а також для висновку тексту, що пояснює. Напису, не приєднані до елементів керування, не відображаються в режимі таблиці.

4. Загальні зведення про поля

Поля використовуються у формі або звіті для відображення даних з таблиці, запиту або інструкції SQL. Поле такого типу називають приєднаним, тому що воно зв'язано з даними в поле в джерелі даних. Крім того, існують вільні поля. Наприклад, можна створити вільне поле для відображення результатів обчислень або для прийому даних, що вводяться користувачем. Уміст вільного поля ніде не зберігається.

5. Загальні зведення про групу параметрів

Група використовується у формі або звіті для висновку обмеженого набору параметрів. Група робить вибір параметра простим і наочним. У кожен момент часу в групі може бути обраний тільки один параметр.

Група складається з рамки групи і набору прапорців, перемикачів або вимикачів. При приєднанні групи до поля до нього приєднується тільки рамка групи, а не знаходяться в ній прапорці, вимикачі або перемикачі. Користувач не повинний визначати властивість Дані (ControlSource) для кожного елемента керування в групі. Замість цього варто задати у властивості Значення параметра (OptionValue) кожного прапорця, вимикача або перемикача число, що є припустимим для поля, до якого приєднана рамка групи. При виборі параметра в групі Microsoft Access вводиться у поле значення, рівне значенню властивості Значення параметра (OptionValue) обраного елемента.

Примітка. У властивості Значення параметра (OptionValue) потрібно задавати число, тому що значенням групи може бути тільки числове, а не текстове значення. Microsoft Access зберігає це число в базовій таблиці. У даному прикладі, щоб замість числа відображати в таблиці «Замовлення» ім'я одержувача, варто створити окрему таблицю «Доставка», що містить назви фірм, що виконують доставку, а потім зробити поле «Доставка» таблиці «Замовлення» подстановочним полем, що приймає дані з таблиці «Доставка».

Група може бути також зв'язана з вираженням або бути вільною. Вільні групи застосовуються в спеціальних діалогових вікнах для прийняття даних, що вводяться користувачем, і для виконання дій, заснованих на цих даних.

6. Загальні зведення про вимикачі

У формі або звіті вимикач може бути використаний як окремий елемент керування, у якому відображаються значення логічного поля з базової таблиці, запиту або інструкції SQL. Наприклад, у наступному прикладі вимикач приєднаний до поля «Поставкипрекращены» таблиці «Товари». Поле «Поставкипрекращены» містить поле логічного типу. Якщо кнопка вимикача натиснута, поле в таблиці «Товари» має значення «Так»; якщо кнопка вимикача не натиснута, поле має значення «Ні».

Коли користувач натискає кнопку вимикача, приєданого до логічного поля, Microsoft Access відображає значення в базовій таблиці у форматі, що визначається значенням властивості поля Формат поля (Format) («Так»/«Ні», «Істина»/«Неправда» або «Вкл»/«Выкл»).

Вимикачі особливо зручні при використанні в групах. У такій групі легко бачити, який з вимикачів натиснутий. Замість підпису на вимикач можна помістити малюнок. Наприклад, у приведеному вище прикладі замість підпису «Постачання припинені» підійде малюнок кошика для паперів.

Вільні вимикачі використовуються також у спеціальних діалогових вікнах для прийому даних, що вводяться користувачем.

7. Загальні зведення про прапорці

У формі або звіті прапорець може бути використаний як окремий елемент керування, у якому відображаються значення логічного поля з базової таблиці, запиту або інструкції SQL. Наприклад, у наступному прикладі прапорець приєднаний до поля «Поставкипрекращены» таблиці «Товари». Поле «Поставкипрекращены» є полем

логічного типу. Якщо прапорець установлений, поле в таблиці «Товари» має значення «Так»; якщо прапорець знятий, поле має значення «Ні».

При установці або знятті прапорця, приєднаного до логічного поля, Microsoft Access відображає значення в базовій таблиці у форматі, що визначається значенням властивості поля **Формат поля (Format)** («Так»/«Ні», «Істина»/«Неправда» або «Вкл»/«Викл»).

Крім того, прапорці включаються в групу для відображення набору обраних значень. Для одержання додаткових зведень про групи параметрів натисніть кнопку .

Вільні прапорці використовуються в спеціальних діалогових вікнах для прийому даних, що вводяться користувачем.

8. Загальні зведення про поля зі списком

У багатьох випадках зручніше вибрати значення зі списку, чим уводити конкретне значення з клавіатури по пам'яті. Поле зі списком дозволяє вибрати кожної з цих способів уведення значення, не вимагаючи при цьому значного місця у формі. Поле зі списком є комбінацією двох елементів: поля і списку, що розкривається.

Значення, обране або введене в приєднане поле зі списком, вставляється в поле, до якого приєднане поле зі списком.

У поле зі списком список складається з рядків з даними. Рядка містять один або кілька стовпців, із заголовками або без заголовків. Якщо поле зі списком, що містить декількох стовпців, є приєднаним, то зберігається значення одного зі стовпців.

Вільне поле зі списком дозволяє зберігати значення, використовуване в іншому елементі керування. Наприклад, за допомогою вільного поля зі списком можна обмежити значення, що відбираються в іншому полі зі списком або в спеціальному діалоговому вікні. Вільне поле застосовується також для пошуку запису за допомогою значення, обраного або введеного в поле зі списком.

Поля зі списком мають властивість **Обмежитися списком (LimitToList)**, що визначає, чи допускається введення в поле будь-яких значень або тільки співпадаючих з одним зі значень списку.

Якщо у формі досить вільного місця і потрібно, щоб список постійно знаходився на екрані, а також якщо потрібно обмежити дані, що вводяться, наявним списком, замість поля зі списком можна використовувати список.

9. Загальні зведення про списки

У багатьох випадках зручніше вибрати значення зі списку, чим уводити конкретне значення по пам'яті. Крім того, вибір зі списку дозволяє бути упевненим, що введене значення є припустимим.

Список складається з рядків з даними. Рядка містять один або кілька стовпців, що можуть бути постачені заголовками. Якщо список з декількох стовпців є приєднаним, то зберігається значення одного зі стовпців.

Вільний список дозволяє зберігати значення, використовуване в іншому елементі керування. Наприклад, за допомогою вільного списку можна обмежити значення, що

відбираються в іншому списку або в спеціальному діалоговому вікні. Вільний список застосовується також для пошуку запису за допомогою значення, обраного в списку. У тих випадках, коли у формі недостатньо місця для відображення списку, або якщо поряд з вибором зі списку потрібно вводити нові значення з клавіатури, замість списку варто використовувати поле зі списком.

Що таке елемент керування «Набір вкладок»?

Елемент керування «Набір вкладок» використовується для представлення декількох сторінок даних в одному наборі. Це особливо зручно при роботі з багатьма елементами керування, що можуть бути розподілені на двох або більш категорій. Наприклад, елемент керування «Набір вкладок» може бути використаний у формі «Співробітники», щоб відокремити загальні зведення від особистих.

Для одержання додаткових зведень про створення елемента керування «Набір вкладок» натисніть кнопку .

Додавання елемента керування «Набір вкладок» для створення форми, що складає з декількох сторінок

- 1 Відкрийте форму в режимі конструктора.
- 2 На панелі елементів виберіть елемент керування Набір вкладок і укажіть форму, у яку необхідно помістити елемент керування.

При додаванні елемента керування «Набір вкладок» у ньому створюються дві сторінки. Першою сторінкою є верхня.

3 Для додавання елементів керування в елемент керування «Набір вкладок» виберіть вкладку сторінки, на яку їх необхідно додати. Потім, використовуючи кожної з перерахованих нижче способів, додайте елемент керування.

- Для висновку на екран списку полів натисніть кнопку Список полів на панелі інструментів, а потім перемістите на вкладку сторінки одне або кілька полів.
- Виберіть елемент керування на панелі елементів і натисніть кнопку миші на сторінці вкладки. Додавати можна будь-який тип елементів керування, за винятком елемента керування «Набір вкладок».
- Скопіюйте елементи керування з іншої частини форми або з іншої сторінки. (Переміщати елементи керування їхньої іншої частини форми або з іншої сторінки не можна.)

4 Також можна виконати наступні дії.

- Для зміни імені вкладки двічі клацніть кнопку миші на сторінці, яку необхідно змінити, щоб відкрити для неї вікно властивостей, а потім в осередку властивості Підпис укажіть нове ім'я. Якщо в осередку властивості Підпис ім'я не зазначене, то використовуються дані з осередку властивості Ім'я.
- Для додавання, видалення або зміни порядку вкладок натисніть праву кнопку миші на границі елемента керування «Набір вкладок» і виберіть команду Уставити вкладку, Видалити вкладку або Послідовність вкладок.
- Для зміни порядку вкладок натисніть на сторінці праву кнопку миші і виберіть команду Порядок переходу.
- Для зміни шрифту, його розміру, стилю і т.д. двічі клацніть кнопку миші на границі елемента керування «Набір вкладок», на екрані буде відкрите вікно властивостей, установите в ньому необхідні властивості. Настроювання для різних сторінок можуть відрізнятися.

5 Установите необхідний розмір елемента керування «Набір вкладок». Переглянете кожну вкладку, щоб переконатися, що всі елементи керування добре до них підходять.

Примітка. При зміні розміру елемента керування «Набір вкладок» зображення елементів керування обрізано не буде. Для зменшення розміру елемента керування «Набір вкладок» може знадобитися переміщення елементів керування.

6 Для перевірки елемента керування переключіться в режим форми.

Примітки

- Для одержання додаткових зведень про інші властивості, які можна установити для елемента керування «Набір вкладок», натисніть кнопку

- Щоб переглянути приклад елемента керування «Набір вкладок», у навчальній базі даних «Борею» відкрийте форму «Співробітники». Для одержання додаткових зведень про відкриття бази даних «Борею» натисніть кнопку .

Для перегляду фвойств форми потрібно виділити усю форму. Виправлення > Виділити форму. Або клацнувши лівою кнопкою миші на квадратику в лівому верхньому куті вікна Конструктора форми, де сходяться вертикальна і горизонтальна лінійки. Коли форма виділена, в області виділення з'являється маленький чорний квадратик.

10.ВКЛАДКА ДАНІ

На цій вкладці зосереджені найважливіші властивості, що визначають, як користувач буде працювати з даними.

Кожна форма має властивість ДЖЕРЕЛО ЗАПИСІВ. Значенням цієї властивості може бути ім'я таблиці, запиту. У формі відображаються записи з однієї або декількох таблиць. Якщо джерелом даних для форми є просто таблиця, то в ній виводяться всі записи цієї таблиці. Можна отфільтровать запису, що будуть виводиться у формі, якщо установити властивість ФІЛЬТР.

Властивість ПОРЯДОК СОРТУВАННЯ дозволяє відсортувати запису у формі по зазначеному полю або декільком полям. Змінити порядок сортування можна при перегляді записів у формі.

Властивості ДОЗВОЛИТИ ЗМІНИ, ДОЗВОЛИТИ ВИДАЛЕННЯ, ДОЗВОЛИТИ ДОДАВАННЯ , УВЕДЕННЯ ДАНИХ визначають режим роботи з даними. Форми можуть використовуватися для перегляду, редагування або введення даних. Змінюючи ці властивості, ви можете заборонити або дозволити редагувати дані у формі, заборонити або дозволити видалення або додавання записів.

Властивість УВЕДЕННЯ ДАНИХ установлюється рівним НЕМАЄ за замовчуванням.

Якщо установити значення цієї властивості ТАК, форма відкривається в режимі введення нових записів і не дозволяє переглядати існуючі записи.

Заборонити або дозволити зміна даних у формі можна також за допомогою властивостей ТИП НАБОРУ ЗАПИСІВ. За замовчуванням, якщо форма базується на таблиці, ця властивість має значення ДИНАМІЧНИЙ НАБІР. Припускаючи використовувати форму для перегляду записів установите значення поля СТАТИСТИЧНИЙ НАБІР.

11.ВКЛАДКА МАКЕТ

Властивості зібрані на цій вкладці, визначають зовнішній вигляд форми.

ПІДПИС визначає текст який виводиться в заголовку вікна форми.

РЕЖИМ ЗА ЗАМОВЧУВАННЯМ і ПРИПУСТИМИ РЕЖИМИ дозволяють указати вид форми: проста форма (у якій одночасно видна тільки один запис), стрічкова форма і таблиця.

СМУГИ ПРОКРУЧУВАННЯ дозволяє вказати ,які потрібні смуги прокручування у формі.

ОБЛАСТЬ ВИДІЛЕННЯ – це область, розташована біля лівої границі форми, використовується у формах , у яких одночасно виводиться кілька записів для того, щоб вибрати який-небудь запис.

Лекції № 38 Тема: Реляційні бази даних.

План лекції:

1. Создание составных форм
2. Убудовані елементи керування
3. Різновиду елементів керування по типі вмісту

1. Создание составных форм

Форми є основним об'єктом, що забезпечує користувачеві зручний доступ даних, тому них приділене велика увага в дійсній книзі,-У першій частині ми показали кілька способів створення форм і детально описали основний інструмент розробки форм — Конструктор. Ви переконалися, яке різноманіття можливостей надає цей інструмент

для того, щоб форми вийшли зручні і красиві. Однак у *гл. 5 "Створення форм для введення даних"* ми розглянули лише самі основні питання, що відносяться до роботи з формами й елементами керування. Тут ми поговоримо про це більш докладно й обговоримо більш складні питання, що стосуються створення й оформлення складених форм — многотабличних і многостраничних, і розглянемо застосування елементів керування всіх типів. Матеріал цієї глави є продовженням *гл. 5*, тому рекомендується ознайомитися з нею перш, ніж переходити до подальшого вивчення. Частина матеріалу про впровадження малюнків і інших об'єктів у форму винесена в *гл. 15*. Ми будемо ще раз повертатися до форм у *гл. 11 і 13* при розгляді програмування у формах і звітах за допомогою макросів і процедур VBA.

Отже, у даній главі будуть розглядатися наступні питання:

- Огляд основних елементів керування у формах і звітах
- Приклади використання основних елементів керування
- Створення складених форм, побудованих на базі зв'язаних таблиць і підлеглих форм, що включає
- Способи створення многостраничних форм із застосуванням роздільника сторінок і набору вкладок
- Використання спливаючих форм і діалогових вікон
- Створення полів, що обчислюються, і застосування виражень для визначення властивостей форми і її елементів керування
- Рекомендації з оформлення форм

Відображення панелі елементів на екрані виконується одним із трьох способів:

- виберіть команду **Вид, Панель елементів** (View, Toolbox);
- натисніть кнопку **Панель елементів** (Toolbox) на панелі інструментів **Конструктор форм** (Form Design);
- клацніть правою кнопкою миші по будь-якій панелі інструментів і виберіть у контекстному меню команду **Панель елементів** (Toolbox).

Найбільше часто застосовувані елементи керування розташовані прямо на панелі. Інші елементи відображаються у виді додаткового списку, якщо натиснути кнопку **Інші елементи** (More Controls) у правому нижньому куті панелі елементів.

У верхній частині панелі елементів розташовані дві спеціальні кнопки: **Вибір об'єктів** (Select Objects) і **Майстри** (Control Wizards). Кнопка **Вибір об'єктів** у натиснутому стані означає, що в даний момент включений режим виділення елементів, тобто реакцією на щиглика миші у формі буде виділення відповідного елемента керування або іншого об'єкта, що потрапив у зазначену область. Цей режим включений за замовчуванням. Однак, якщо натиснути на іншу кнопку на панелі елементів, призначену для створення елемента керування, кнопка вибору об'єктів автоматично віджимається. Її потрібно знову натиснути, якщо ви передумали створювати елемент і хочете повернутися в режим виділення об'єктів. Натиснута кнопка **Майстра** означає, що при розміщенні елемента керування на формі запуститься відповідний майстер. При віджатій кнопці майстра не запускаються й елементи потрібно набудовувати вручну. Майстра елементів допомагають створювати елементи керування, що вимагають

визначеного настроювання, такі як поля зі списком, командні кнопки, групи елементів вибору. Практично для кожного такого елемента керування існує свій майстер. Майстри елементів зручно використовувати як на початковому етапі роботи з Access для навчання, так і в подальшій роботі, оскільки вони автоматизують рутинну операцію стандартного настроювання елемента, представляючи неї у виді зручного діалогу.

Крім майстрів елементів, роботу з настроювання елементів керування полегшують спеціальні майстри, що допомагають у створенні нетривіальних значень властивостей елемента, них називають побудовцями. Наприклад, Побудувач запитів (Query Builder) допомагає створити інструкцію SQL, що визначає джерело рядків для списків і полів зі списками. Побудувач виражень (Expression Builder) допомагає створювати синтаксично правильні вираження для елементів керування, що обчислюються.

Основні кнопки, розташовані на панелі елементів, призначені для створення елементів керування різних типів. Назва типу елемента керування відображається у виді спливаючої підказки при наведенні покажчика миші на відповідну кнопку на панелі керування.

Панель елементів звичайно відображається як "плаваюча" (див. мал. 9.1), так що її можна вільно переміщати по екрані, знаходячи для неї зручне місце. Переміщаючи за допомогою миші рамку панелі елементів, можна змінити її розміри або закріпити в границі вікна, наприклад так, як це показано на мал. 9.2.

2. Убудовані елементи керування

У виді кнопок безпосередньо на панелі елементів розміщуються лише убудовані елементи керування, на відміну від впроваджуваних об'єктів і елементів керування Active, що **можуть бути** додані в додаток

3. Різновиди елементів керування по типі вмісту

У Access існує три різновиди елементів керування, у залежності від типу вмісту в них, тобто від способу заповнення їх даними:

- приєднані
- вільні
- вычисляемые

До якого з перерахованих видів відноситься той або інший елемент керування, визначається тим, як заданий для нього джерело даних.

Приєднані елементи керування зв'язані з полями базової таблиці, тобто тієї таблиці, що є джерелом даних для форми. Якщо джерелом даних є запит, то приєднані елементи керування можуть зв'язуватися з полями в різних таблицях. У приєднаному елементі відображаються дані, що утримуються в зв'язаному з ним полі таблиці, і при зміні цих даних відповідним чином обновляється і значення в полі таблиці. У приєднаних елементах можна відображати всі доступні в Access типи даних, у тому числі об'єкти OLE і гіперпосилання.

Вільні елементи керування не зв'язані з таблицями. Вони призначені або для введення інформації, що використовується не для безпосереднього редагування даних у джерелі, а в інших цілях (звичайно макросами або програмами VBA), або для відображення об'єктів OLE, що зберігаються в самих формах. Вільними елементами є також всі елементи, не зв'язані з якими-небудь даними, а призначені лише для поліпшення візуального сприйняття форм, такі як лінії, прямокутники, малюнки.

Елементи керування, що обчислюються - це такі елементи, значення яких обчислюються на основі значень інших елементів. Як джерело даних для цих елементів використовуються вираження і функції.

Лекції № 39 Тема: Реляційні бази даних.

План лекції:

1. Створення і печатка звітів
2. Автоматическое создание отчета на основе таблицы или запроса
3. Создание отчета с помощью Мастера отчетов
4. Подготовка составных отчетов
5. Создание составных отчетов
6. Применение подчиненных отчетов

1. Створення і друк звітів

Однієї з основних задач створення і використання баз даних є надання користувачам необхідної інформації на основі існуючих даних. У Access 2002 для цих цілей призначені форми і звіти. Звіти дозволяють вибрати з бази даних необхідну користувачем інформацію й оформити неї у виді документів, які можна переглянути і надрукувати. Джерелом даних для звіту може бути таблиця або запит. Крім даних, отриманих з таблиць, у звіті можуть відображатися обчислені по вихідним даним значення, наприклад підсумкові суми.

Звіти і форми Access мають багато загального. Однак, на відміну від форм, звіти не призначені для введення і виправлення даних у таблицях. Вони дозволяють лише переглядати і друкувати дані. У звіті неможливо змінити вихідні дані за допомогою елементів керування, як це мож-но зробити за допомогою форм. Хоча в звітах можна використовувати такі ж елементи керування для вказівки стану перемикачів, прапорців і списків.

Звіт, як і форма, може бути створений за допомогою майстра. Розділи звіту подібні розділам форми і включають заголовок і примітка звіту, область даних, а також верхній і нижній колонтитули. У примітку звіту часто поміщають поля з підсумковими значеннями. Елементи керування можуть бути додані в звіт за допомогою панелі інструментів **Панель елементів** (Toolbox), ідентичної тієї, що використовується в режимі Конструктора форм. Форматування й угруповання елементів керування в звіті виконуються аналогічно форматуванню й угрупованню елементів керування у формі. Форми можуть містити підлеглі форми, а звіти можуть містити підлеглі звіти.

2. Автоматичне створення звіту на основі таблиці або запиту

Access пропонує кілька способів створення звітів. Найбільш простим з них є використання засобів автоматичного створення звіту. Автоматично створюваний на основі таблиці або запиту звіт називається *автозвіт*. Access дозволяє автоматично створювати звіти двох форматів: у стовпець і стрічковий.

Щоб створити автозвіт:

1. На панелі об'єктів вікна **База даних** (Database) клацніть по ярлику **Отчеты** (Reports) і натисніть кнопку **Создать** (New). З'явиться діалогове вікно **Новый отчет**
2. У списку діалогового вікна **Новый отчет** (New Report) виділіть один з елементів: **Автоотчет: в столбец** (AutoReport: Columnar) или **Автоотчет: ленточный** (AutoReport: Tabular).
3. У поле зі списком, що знаходиться в нижній частині діалогового вікна **Новый отчет** (New Report), утримуються імена всіх таблиць і запитів бази даних, що можуть бути використані як джерело даних для звіту. Клацніть лівою кнопкою миші по кнопці зі стрілкою для відкриття списку, а потім виділіть в списку необхідний елемент.
4. Натисніть кнопку **ОК**.

У результаті Access автоматично створить звіт на основі обраного джерела даних, використовуючи стрічковий формат або формат у стовпець. Стрічковий формат розташовує поля виведених записів у рядок. Формат у стовпець розташовує поля виведених записів у стовпець. Автоотчет, створений з використанням кожного з цих двох форматів, буде включати всі поля і записи, що маються в обраному джерелі даних.

Щоб створений звіт можна було використовувати надалі, його необхідно зберегти. Для цього виберіть команду **Файл, Сохранить** (File, Save) или нажміть кнопку **Сохранить** (Save) на панелі інструментів. Затем, в текстовое поле появившегося діалогового окна **Сохранение** (Save As) введіть назву нового звіту (наприклад: Мой отчет) и нажміть кнопку **ОК**.

3. Створення звіту за допомогою Майстра звітів

Ще одним простим і швидким способом створення звітів є використання Майстра звітів. З його допомогою можна створювати звіти (від найпростіших до підлеглі звіти, що включає), що містять поля з однієї або більш таблиць або запитів. Використання Майстра звітів дозволяє дуже швидко познайомитися з багатьма основними засобами розробки звітів, наданими Access 2002. Процес створення звітів, більш складних, чим дозволяє майстер, описаний.

Щоб більш наочно представити процес створення звіту за допомогою майстра, опишемо процедуру створення звіту на основі відразу декількох таблиць: "Товари" (Products), "Постачальники" (Suppliers) і "Типи" (Categories) бази даних "Борею" (Northwind). У звіті буде представлена поточна кількість товарів на складі по категоріях.

Щоб створити простий звіт:

1. На панелі об'єктів вікна бази даних клацніть по ярлику **Отчеты** (Reports) і натисніть кнопку **Создать** (New). З'явиться діалогове вікно **Новый отчет** (New Report) (см. рис. 6.1).
2. В списку діалогового окна **Новый отчет** (New Report) виділіть елемент **Мастер отчетов** (Report Wizard). У поле зі списком, що знаходиться в нижній частині діалогового вікна **Новый отчет** (New Report), утримуються імена всіх таблиць і запитів бази даних, що можуть бути використані як джерело даних для звіту. Розкрийте цей список і виділіть в ньому таблицю "Товари". Натисніть кнопку **ОК**. З'явиться перше діалогове вікно Майстра звітів.
3. Насамперед потрібно визначити, які поля включити в звіт. З таблиці "Товари" у звіті будуть використовуватися наступні поля: "КодТовара" (ProductID), "Марка" (ProductName), "Ціна" (UnitPrice), "НаСкладі" (UnitsInStock). За допомогою кнопки ">" (або двічі клацнувши по елементі списку) по черзі перемістите ці поля зі списку **Доступні поля** (Available Fields) у список **Обрані поля** (Selected Fields). У звіті поля з'являються ліворуч праворуч, відповідно до послідовності їхнього розташування в списку **Обрані поля**.

Зауваження

Якщо на якому-небудь кроці створення звіту за допомогою Майстра звітів була допущена помилка або виникла необхідність зміни яких-небудь параметрів, то для повернення до попереднього кроку натисніть кнопку **Назад** (Back). Крім того, у будь-який момент можна натиснути кнопку **Скасування** (Cancel) для відмовлення від створення звіту і повернення до вікна бази даних.

4. Крім того, щоб бачити постачальників товару і категорії, у створюваний звіт необхідно додати поле "Назва" (CompanyName) з таблиці "Постачальники" (Suppliers) і "Категорія" (CategoryName) з таблиці "Типи" (Categories). Для додавання першого відсутнього поля розкрийте список **Таблиці і запити** (Tables/Queries) і виберіть елемент **Таблиця: Постачальники** (Table: Suppliers). Ім'я постачальника не повинне відображатися наприкінці звіту. Щоб стовпець "Назва" виводився відразу за стовпцем "Марка", у списку **Обрані поля** (Selected Fields) виділіть поле "Марка", потім у списку **Доступні поля** (Available Fields) виділіть поле "Назва" і натисніть кнопку ">". Access перемістить поле "Назва" зі списку **Доступні поля** (Available Fields) і вставить його в список **Обрані поля** (Selected Fields) після поля "Марка", перед полем "Ціна".
5. Аналогічно додайте в кінець списку поле "Категорія" з таблиці "Типи". Щоб перейти до другого діалогового вікна Майстра звітів, натисніть кнопку **Далі**.

Зауваження

Можна таким способом додавати в створюваний звіт поля з різних таблиць, що зв'язані між собою в схемі даних.

6. В другому діалоговому вікні необхідно вказати, яким образом будуть структуровані дані в звіті. Майстер перевіряє зв'язки між таблицями й у залежності від обраного представлення пропонує кілька варіантів угруповання запису в звіті. Щоб здійснити угруповання по товарах, виділите в списку елемент **Товари** (by Products).
7. Натисніть на кнопку **Вивести додаткові зведення** (Show Me More Information) у лівій частині діалогового вікна для відображення одного з допоміжних діалогових вікон Майстра звітів. Якщо в цьому діалоговому вікні натиснути кнопку **Показати приклади** (Show Me Examples), то Access виведе додаткові допоміжні діалогові вікна, у яких використовуються "приклади з таблиць "Продажу" (Sales Reps), "Клієнти" (Customers) і "Замовлення" (Orders), що дозволяють продемонструвати різні способи угруповання, що Майстер звітів може автоматично застосувати до створюваного звіту. Закрийте всі допоміжні діалогові вікна і поверніться до діалогового вікна Майстра звітів, зображеному на мал. 6.4. Натисніть кнопку Далі (Next) для переходу до третього діалогового вікна Майстра звітів.
8. У третім діалоговому вікні Майстра звітів необхідно визначити додаткові рівні угруповання в звіті. Виділите в списку поле "Категорія" (Category) і натисніть кнопку ">" (або просто двічі клацніть по цьому полю в списку), щоб установити угруповання по категорії товарів.

Зауваження

Якщо буде потрібно скасувати додавання поля угруповання, натисніть кнопку "<" або двічі клацніть за назвою цього поля (виділеному жирним шрифтом) в області діалогового вікна з ескізом звіту.

9. Майстер звітів дозволяє задати інтервали значень, по яких групуються дані. Щоб змінити них у цьому ж (третьому) діалоговому вікні Майстра звітів, натисніть кнопку **Угруповання** (Grouping Options). На екрані з'явиться діалогове вікно **Інтервали угруповання** (Grouping Intervals), зображене на мал. 6.7. Змінюючи інтервал угруповання, можна впливати на спосіб угруповання даних у звіті. Для числових полів можна задати угруповання по десятках, сотням і іншим значенням. Для текстових полів можливе угруповання по першій букві, першим трьом буквам і т.д.
10. Для створюваного звіту виберіть зі списку **Інтервали угруповання** (Grouping Intervals) варіант **Звичайний** (Normal) і натисніть кнопку ОК. Для переходу до четвертого діалогового вікна Майстра звітів натисніть кнопку **Далі** (Next).
11. Access 2002 дозволяє сортувати запису в межах групи за значенням довільного поля, при цьому сортування проводиться не більш ніж по чотирьох полях (мал. 6.8). Виберіть у першому списку, що розкривається, поле "Кодтовара" (ProductID). За замовчуванням прийнятий порядок сортування по зростанню; якщо ви хочете вибрати порядок сортування по убутанню, то натисніть на кнопку праворуч від списку, що розкривається. Для повернення до сортування по зростанню знову натисніть на цю кнопку.

12. Щоб указати для стовпця звіту підсумкову інформацію, натисніть" на кнопку **Підсумки** (Summary Options). У діалоговому вікні, що **з'явилося**, Підсумки (Summary Options), зображеному на мал. 6.9, можна задати параметри для обраного стовпця. Вибрати можна серед усіх числових полів звіту, що не є лічильниками, що Майстер звітів перелічить у цьому діалоговому вікні і дозволить установити для кожного з них прапорці Sum, Avg, Min і Max. У залежності від установлених прапорців Майстер звітів додасть відповідні підсумкові поля в кінець звіту, що обчислюють підсумкову суму, середнє значення, мінімальне або максимальне значення даного поля по всіх записах. Група перемикачів **Показати** (Show) дозволяє вибрати, чи впливає в звіті вказувати тільки підсумки або необхідно видавати повний звіт, додаючи поля з проміжними підсумками наприкінці кожної групи і поля з повними підсумками наприкінці звіту. Для створюваного звіту установите прапорці **Sum** для поля "НаСкладі" і Avg для поля "Ціна", перемикач **дані і підсумки** (Detail and Summary) і прапорець **Обчислити відсотки** (Calculate Percent of Total for Sum), що дозволяє виводити частку групи (у відсотках) у загальному підсумку для всіх груп. Для повернення до діалогового вікна Майстра звітів натисніть кнопку **ОК**.
13. Натисніть кнопку **Далі** (Next), щоб перейти до п'ятого діалогового вікна Майстра звітів, зображеному на мал. 6.10. Майстер звітів запропонує на вибір шість варіантів оформлення згрупованих даних (варіантів макета звіту). У лівій частині діалогового вікна Майстра відображається ескіз обраного макета. Виберіть у групі **Макет** (Layout) перемикач **східчастий** (Stepped). За замовчуванням Майстер звітів установлює прапорець **Настроїти ширину полів для розміщення на одній сторінці** (Adjust the field width so all fields fit on a page), що дозволяє заощадити папір і зробити звіт більш зручним для читання. У групі **Орієнтація** (Orientation) можна вибрати розташування сторінки звіту при печатці. Широкиим звітам, у яких відображається багато полів в одному рядку, зручно задавати орієнтацію **альбомна** (Landscape). Для звіту ж з нашого приклада залишіть обраним перемикач **книжкова** (Portrait). Щоб перейти до шостого діалогового вікна Майстра звітів, натисніть кнопку **Далі** (Next).
14. У шостому діалоговому вікні Майстра звітів виберіть один із запропонованих стилів оформлення звіту. У лівій частині діалогового вікна Майстра відображається ескіз звіту обраного стилю. Для створюваного звіту виберіть стиль **Стиснутий** (Compact), а потім натисніть кнопку **Далі** (Next) для переходу до останнього діалогового вікна Майстра звітів.

Зауваження

У шостому вікні Майстра звітів відображається список усіх доступних у даний момент стилів. Крім шести заздалегідь визначених стандартних стилів, Access дозволяє настроїти будь-який стандартний стиль або створити свій власний для використання при оформленні форм і звітів, у тому числі за допомогою майстра. Набір варіантів оформлення для звітів можна поповнити і застосувати аналогічно тому, як це робиться для форм.

15. В останньому діалоговому вікні Майстра звітів як заголовок звіту введіть: Товари на складі. Майстер звітів збереже створений звіт під цим же ім'ям. Виберіть перемикач **Переглянути звіт** (Preview the Report) і натисніть кнопку **Готове** (Finish) для завершення створення звіту. Майстер звітів створить звіт і відобразить його в режимі Попереднього перегляду. Якщо при цьому необхідно одержати довідку по роботі зі звітом, попередньо установите прапорець **Вивести довідку по роботі зі звітом?** (Display Help on Working with the Report?).

На мал. 6.13 зображений звіт, створений майстром. Для перегляду всіх частин звіту скористайтеся вертикальною і горизонтальною смугами прокручування. Цей звіт можна використовувати безпосередньо в тім виді, як його створив майстер, або застосувати його як основу для розробки більш складного звіту.

4. Підготовка складених звітів

Однієї з найважливіших функцій, що звичайно надають настільні СУБД, є можливість створення різних звітів — документів, що містять дані з таблиць бази даних. Відмінною рисою Microsoft Access є наявність дуже могутнього генератора звітів, що дозволяє створювати звіти високої якості. Наявність майстрів, що генерують автоматично або за допомогою діалогу з користувачем найрізноманітніші звіти, значно полегшує трудомісткий процес їхнього створення. Microsoft Access 2000 і 2002 мають істотні відмінності і переваги в порівнянні з іншими найбільш відомими СУБД: це головна риса продуктів Microsoft — зручність використання й одночасно потужність продукту — у сполученні з можливістю побудови комплексних рішень на базі сучасних технологій, а саме: інтеграція баз даних з SQL Server, з іншими додатками Office і публікація інтерактивних даних у Web. У Access 2002 засобу публікації в Web значно розширилися. Це стосується і звітів: тепер вони можуть бути перетворені у формат XML і в інтерактивні сторінки доступу до даних.

5. Створення складених звітів

Під складеними звітами в даному випадку розуміються звіти, що мають складну структуру: підлеглі звіти, що включають, побудовані на базі перехресних запитів, виведені на печатку у виді декількох стовпчиків, як це робиться, наприклад, у газетах. Нижче будуть описані такі звіти, їхнє призначення і способи створення

6. Застосування підлеглих звітів

За аналогією зі складеними формами, що включають підлеглі форми, можна припустити, що складені звіти використовуються також і для печатки даних з таблиць, зв'язаних відношенням "один-ко-многим". Однак це не зовсім так. У більшості випадків для відображення даних зі зв'язаних таблиць не потрібно впроваджувати підлеглі звіти.

Наприклад, якщо потрібно створити звіт по товарах з розбивкою по категоріях товарів, це можна зробити двома способами:

- в основному звіті виводяться дані про категорії товарів, а в підлеглому — про товари, що входять у категорії. І це буде аналогічно створенню форми "Категорії товарів", у яку включається підлегла форма, що відображає дані з таблиці товарів;
- те ж саме може бути виконане без використання підлеглого звіту за допомогою угруповання товарів по категоріях. Звіт у цьому випадку повинний будуватися на базі запиту, що включає дані з таблиць "Категорії" і "Товари".

Проте бувають ситуації, коли використання підлеглого звіту необхідно або переважно. Наприклад:

- Головний звіт являє собою контейнер, у який включаються один або більш підлеглих звітів з незв'язаних таблиць. У цьому випадку головний звіт може взагалі не бути зв'язаним з таблицею або запитом і містити тільки напису й елементи, що обчислюються.
- Іноді підсумкові дані потрібно представити в документі у виді окремої таблиці поряд з детальними даними. У цьому випадку просто угрупованням записів з підсумковими рядками не обійтися, потрібно ввести підлеглий звіт, що виведе підсумкові дані в необхідному виді.
- Угруповання в звітах Access є дуже могутнім механізмом, що дозволяє одержати самі різні способи обробки даних звіту. Однак тільки при використанні підлеглих звітів можна створити групи, що містять дві і більш підгрупи даних.
- Головний звіт може містити, по-перше, кілька підлеглих звітів і, по-друге, два рівні вкладеності підлеглих звітів. І, крім того, на будь-якому рівні вкладеності поряд з підлеглим звітом можна вставити в головний звіт підлеглу форму.

Основне достоїнство підлеглих звітів полягає в тому, що з їхньою допомогою можна розмістити на сторінці звіту найрізноманітнішу інформацію в зручному структурованому виді, чого неможливо домогтися, якщо просто створити складний запит, включивши в нього дані з багатьох таблиць.

Приклади використання підлеглих звітів можна знайти в списку звітів навчальної бази даних "Борею" (Northwind).

Відкрийте звіт "Продажу по типах" (Sales by Category) у режимі Конструктора. Ви побачите, що підлеглий звіт розміщений у розділі заголовка групи "Категорія" (CategoryName). Обидва звіти, і головний і підлеглий, базуються на тому самому запиті "Продажу по типах" (Sales by Category), але в головному звіті виводяться дані про категорії — ім'я категорії, а в підлеглому — про товари, що входять у цю категорію (найменування товару і його ціна).

Виділіть елемент керування **Підлегла форма/звіт** (Subform/siibreport) і подивіться властивості цього елемента. Ви побачите, що головний і підлеглий звіти зв'язуються по полю "Категорія" (CategoryName). Оскільки й у формах, і в звітах використовується той самий елемент керування **Підлегла форма/звіт**, властивості для зв'язаних полів ті ж, що й у формі.

Зверніть увагу, що в макеті підлеглого звіту немає елемента керування, зв'язаного з полем "Категорія", однак, як і при зв'язуванні форм, як поля для зв'язку можна використовувати поля базового запиту.

Звіт "Продажу по роках" (Sales by Year) являє приклад звіту, у якому підлеглий звіт використовується для висновку підсумкових даних про продажі по кварталах року (мал. 10.3).

Обидва звіти, основний і підлеглий, базуються на запиті "Продажу по роках" (Sales by Year), що дозволяє вибрати всі замовлення за зазначений період часу (перевіряється поле "Датаисполнения" (ShippedDate). Запит включає поле, що обчислюється, "Рік" (Year), що розраховується на основі дати замовлення і використовується при угрупованні в звіті, і поле, що містить підсумкову суму кожного замовлення "Промежуточнаясумма" (Subtotal).

Підлеглий звіт "Підлеглий для продажів по роках" (Sales by Year Subreport) містить кілька вичислюваних елементів:

- "Годнадпись" (YearLabel), що відображається у верхньому лівому куті, обчислюється за допомогою вираження:

= "Підсумок за " & [Рік] & " м." (або=[Year] & " Summary");

- "Квартал" (Quarter), що як джерело даних містить вираження:

=Format([Датаисполнения];"q") (АБО =Format([ShippedDate];"q"));

- "ПодсчетПромежуточнойСуммы" (CountSubtotal) — кількість виконаних замовлень, що обчислюється за допомогою статистичної функції Count:

=Count([КодЗаказа]) (або =Count([OrderI]));

- "Квпромежуточнаясумма" (QtrSubtotal) — підсумкова сума за квартал, що обчислюється за допомогою вираження:

=Sum([ПромежуточнаяСумма]) (або =Sum([Subtotal])).

Угруповання записів виконується по році і по кварталі. Зверніть увагу, що роздягнув **Область даних** (Detail) звіту порожній, тобто в звіт виводяться тільки підсумкові значення.

Детальні дані, тобто всі записи з базового запиту, включені в головний звіт.