**Міністерство освіти та науки України**

**Кіровоградський національний технічний університет**

**Методичні вказівки**

**до лабораторних робіт**

**з предмету «Організація баз даних» (5 семестр)**

**для студентів денної та заочної форми навчання**

**напрямку 050102 «Комп’ютерна інженерія»**

**Розробили:**

**старший викладач Сидоренко В.В.**

**викладач Константинова Л.В.**

**Кіровоград 2016**

Методичні вказівки до лабораторних робіт з предмету «Організація баз даних» (5 семестр) для студентів денної та заочної форми навчання напряму 050102 «Комп’ютерна інженерія». Укл. Сидоренко В.В., Константинова Л.В. - Кіровоград: КНТУ, 2016.- 97 с.

Укладачі: Сидоренко В.В., Константинова Л.В.

Рецензент:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Рекомендовано кафедрою програмування та захисту інформації,

протокол № \_\_ від \_\_\_ \_\_\_\_\_\_\_\_\_\_ 2016 р.

**Вступ**

В методичних вказівках представлено теоретичний і практичний матеріал для роботи з базами даних за допомогою СКБД Microsoft Access. Дані методичні вказівки починають знайомити з основними прийомами роботи з базами даних. Приводиться багато практичних прикладів та ілюстрацій, які допомагають краще засвоїти викладений матеріал. Розглядається початок роботи з БД за допомогою СКБД Microsoft Access, ознайомлення з основними об’єктами БД, робота з таблицями, створення зв’язків між таблицями. Приводиться матеріал та приклади до поняття цілісність даних, розглядаються типи та можливості QBE-запитів. Приводиться побудова інтерфейсу користувача за допомогою форм. Розглядається отримання підсумкової інформації бази даних за допомогою звітів.

1. **Початок роботи з Access**

Організація даних є ключовим моментом під час роботи з великою кількістю за об’ємом інформації. Дуже важливо впорядкувати дані таким чином, щоб легко та швидко знаходити потрібну інформацію.

Програма Microsoft Access входить до складу пакету Microsoft Office та представляє собою грандіозну систему, яка забезпечує ефективну розробку та супровід баз даних.

Ми розглянемо русифіковану версію СКБД Access. Вона дозволяє розв'язувати широке коло завдань користувачів без програмування. Одна з основних переваг СКБД Access полягає в тому, що вона має прості та зручні засоби обробки кількох таблиць в одній базі даних. Таблиця - це місце зберігання даних. Вона є основним об'єктом бази даних.

У СКБД перших поколінь для кожної таблиці створювався свій файл бази даних. Для спільного використання таблиць у цих СКБД застосовують спеціальні програми (утиліти). Потім для спільного використання таблиць почали застосовувати спеціальні команди, які мають використовуватись перед початком роботи з базою даних для так званого зв'язування таблиць.

Пізніше спробували зберігати у одній базі даних кілька таблиць з описом зв'язків між ними. Прикладом такої СКБД є FoxPro. Але тільки у Access з'явилися зручні та наочні засоби зв'язування таблиць, які зберігаються в одному файлі.

Збереження кількох таблиць в одній базі даних має цілий ряд переваг. Основні з них такі:

1. Можливість розподілу даних за таблицями згідно зі змістом даних. Наприклад, у першій таблиці зберігаються звітні дані вчителів, у другій - їхній тижневий розклад, а в третій - відомості про позаурочну діяльність.

2. Табличне, тобто модульне, подання даних дозволяє модернізувати кожну таблицю окремо.

Редагування даних в одному місці бази призводить до автоматичного їх редагування в іншому місці. Наприклад, якщо в таблиці Список (мал. 1.) стирається запис із прізвищем "Кірін Т.В.", то відповідно запис буде стерто і в таблиці Консультація (мал. 2.).

**Список**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № п/п | Ф.И.О. | Предмет | ДатаРожд | Стаж | Нагрузка | Оклад |
| 1  2  3  4  5  6 | Костенко A.M. Носенко С.К. Борисов Н.П. Кирин Т.В. Стойко Б.С. Абрамов К.К. | Математика Физика  Химия  Физика  Литература Математика | 12.01.70 20.05.55 16.04.38 29.11.50 17.08.46 22.12.61 | 3  15  32  18  24  7 | 18  18  24  24  20  22 | 140  180  250  220  200  160 |

Малюнок 1

**Консультация**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Ф.И.О. | Аудитория | День | Время |
| 1  2  3  4  5  6 | Костенко A.M.  Носенко С. К.  Борисов Н.П.  Кирин Т. В.  Стойко Б.С.  Абрамов К. К | 215  307  112  307  230  215 | Вторник  Понедельник  Среда  Пятница  Вторник  Четверг | 15,00-17.00  16.00-18.00  15.00-17.00  16.00-18.00  15.00-18.00  15.00-17.00 |

Малюнок 2

1. **Теми лабораторних робіт, що розглядаються та оцінювання знань**

Дані методичні вказівки містять у собі 7 лабораторних робіт за наступними темами:

**1. Типи та властивості полів. Ознайомлення з режимами побудови таблиць бази даних. Робота з таблицями та їх редагування.**

**2. Створення зв'язків між таблицями. Створення схем даних. Типи відношень та цілісність даних.**

**3. Проектування та можливості QBE запитів. Запити на вибірку.**

**4. Обчислювальні поля у запитах. Додавання, поновлення, видалення інформації з таблиць БД за допомогою QBE-запитів. Параметричні запити, перехресні запити.**

**5. Побудова інтерфейсу користувача за допомогою форм. Майстер форм. Режим конструктора форм.**

**6. Робота з елементами керування в формах бази даних. Розширення можливостей інтерфейсу користувача баз даних за допомогою різних видів форм. Побудова меню роботи з БД.**

**7. Отримання підсумкової інформації бази даних за допомогою звітів. Редагування звіту за допомогою інструментів конструктора звітів.**

Форма підсумкового контролю: залік (п’ятий семестр), іспит (шостий семестр).

Максимальну кількість балів студент може одержати у випадку відвідування всіх лекцій, лабораторних занять, виконання і захисту виконаних самостійних завдань у встановлений термін, проходження контролю.

При виконанні і захисті лабораторних робіт після встановленого терміну, одержані бали перераховуються з коефіцієнтом: для самостійної роботи студента -0,3; лабораторної роботи -0,7.

В якості самостійного завдання необхідно виконати теоретичну роботу згідно обраної студентом теми.

**Шкала оцінювання: національна та ECTS**

|  |  |  |
| --- | --- | --- |
| Сума балів за всі види навчальної діяльності | ОцінкаECTS | Оцінка за національною шкалою |
| для екзамену, курсової роботи |
| 90 – 100 | **А** | відмінно |
| 82-89 | **В** | добре |
| 74-81 | **С** |
| 64-73 | **D** | задовільно |
| 60-63 | **Е** |
| 35-59 | **FX** | незадовільно з можливістю повторного складання |
| 0-34 | **F** | незадовільно з обов’язковим повторним вивченням дисципліни |

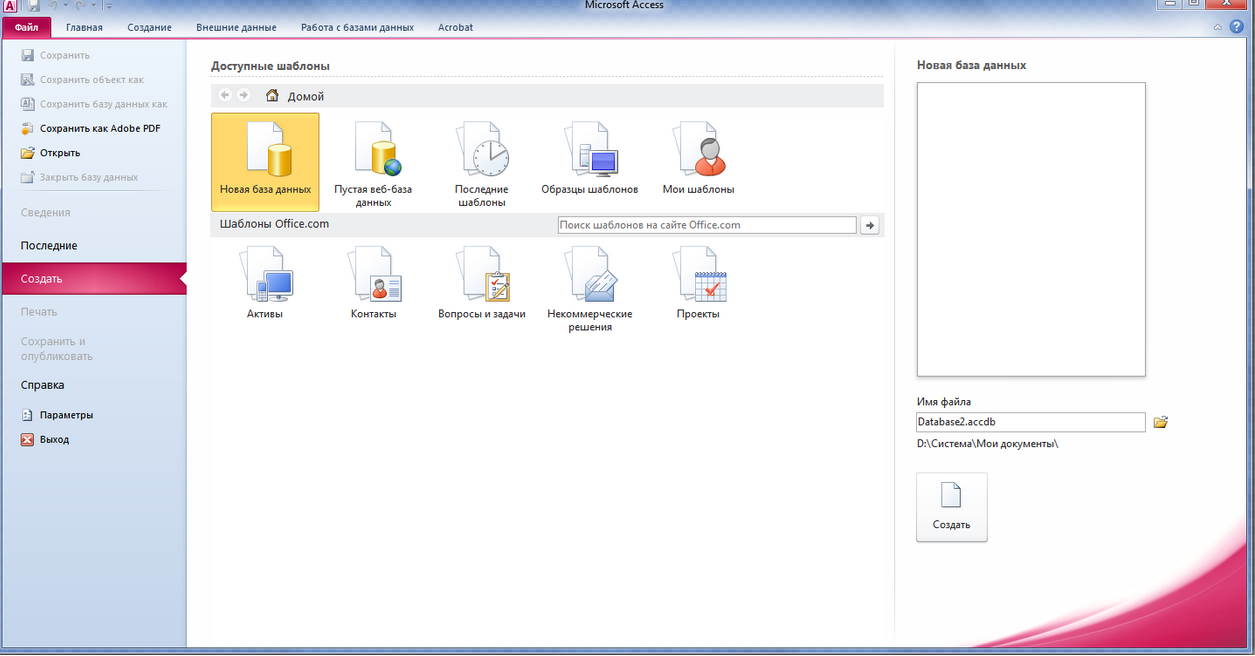
Вибравши предметну область, над якою ви будете працювати, ви повинні виконати завдання до лабораторних робіт, а також відповісти на питання вкінці кожної лабораторної роботи. Звіт повинен містити хід виконання завдань а також графічні матеріали, що підтверджують виконання цих завдань.

1. **Запуск СКБД Access**

Запуск СКБД Access можливо здійснити у кілька способів.

Ми розглядатимемо запуск через меню **Пуск (Start)** та **Проводник (Explorer)**.

Якщо для установки Access операційна система автоматично додає в меню **Пуск => Программы (Start => Programs)** пункт Microsoft Access. Найлегший спосіб запуску Access - натиснути на кнопку **Пуск (Start)**, вибрати підменю **Программы (Programs)**, а в ньому - пункт Microsoft Access. В результаті Access завантажиться, про що свідчить поява на екрані заставки програми (мал.1.).



Малюнок 1-Скриншот Microsoft Access 2010

Якщо користувач напевно не знає, який з файлів вибрати, йому потрібно звернути увагу на розширення. Бази даних Access ранніх версій мали розширення MDB (MDA і MDE). Файли MDA являють собою бібліотеки з доповненнями або майстрами, а файли MDE - це бази даних Access, що вміщують запрограмовані додатки, із яких видалені вихідні тексти модулів. Такі додатки можуть завантажувати на виконання, але модулі, які входять до них, переглядати неможливо. Деякі типи файлів, що зустрічються:

.[accdb](http://open-file.ru/types/accdb) - [файл бази даних Access 2007](http://open-file.ru/types/accdb)

.[accdc](http://open-file.ru/types/accdc) - б[аза даних з цифровим підписом Microsoft Access](http://open-file.ru/types/accdc)

.[accde](http://open-file.ru/types/accde) - [база даних Microsoft Access (тільки виконання)](http://open-file.ru/types/accde)

.[accdr](http://open-file.ru/types/accdr) - [база даних Microsoft Access (тільки читання)](http://open-file.ru/types/accdr)

.[accdt](http://open-file.ru/types/accdt) - [шаблон бази даних Microsoft Access](http://open-file.ru/types/accdt)

.[accdw](http://open-file.ru/types/accdw) - [посилка на базу даних Microsoft Access](http://open-file.ru/types/accdw)

.[accft](http://open-file.ru/types/accft) - [шаблон типів даних Microsoft Access](http://open-file.ru/types/accft)

.[adn](http://open-file.ru/types/adn) - [шаблон проекта Microsoft Access](http://open-file.ru/types/adn)

.[adp](http://open-file.ru/types/adp) - [проект Access](http://open-file.ru/types/adp)

.[mdbhtml](http://open-file.ru/types/mdbhtml) - HTML-файл бази даних Microsoft Access

**Лабораторна робота №1**

**Тема: Типи та властивості полів. Ознайомлення з режимами побудови таблиць бази даних. Робота з таблицями та їх редагування.**

**Мета: Ознайомитись з типами та властивостями полів в Access. Навчитись створювати таблиці. Навчитись працювати з таблицями, редагувати їх та оптимально використовувати Access.**

Зміст роботи за варіантом індивідуального завдання:

1. Створіть таблиці в Access за допомогою режиму таблиці.
2. Створіть таблиці в Access за допомогою конструктора таблиць.
3. Створіть таблиці в Access за допомогою майстра таблиць.
4. По заданій предметній області побудуйте таблиці (дві і більше) з полями різних типів та визначте їх ключеві поля.
5. Відредагувати таблиці своєї бази даних за допомогою перелічених контекстних меню.
6. Зробити пошук.
7. Відсортувати записи.
8. Створіть розширений фільтр.
9. Створіть ярлик таблиці та розмістіть його на робочому столі Windows.
10. Зробіть недоступною одну з таблиць вашої бази даних для інших користувачів.

**Теоретичні відомості:**

**Типи та властивості полів**

У системі Access як і в інших базах даних реляційного типу рядок таблиці ототожнюється з терміном "запис", а колонка — з терміном "поле". Кожне поле має ім'я, тип і властивості. При створенні структури таблиці обов'язково вказують імена і типи полів. Властивості полів можуть прийматися за замовчуванням. Одне або кілька полів необхідно визначити як ключеві.

В Access використовують такі типи полів:

1. Текстове поле. У текстовому полі можуть записуватись літери, цифри та інші символи. Текстові поля мають властивості, перелік яких на мал. 4.

Коротко розглянемо суть цих властивостей:

1. **Размер поля**. Він має межі від 1 до 255 символів.
2. **Формат поля**. За допомогою спеціальних символів можна задавати вигляд та розмір рядків, що вводяться. Для того щоб задати формат текстових полів, використовують такі кодові символи: @ - текстовий символ або пропуск; & — текстовий символ; < ‑ перетворення символів у нижній регістр; > — перетворення символів у верхній регістр.



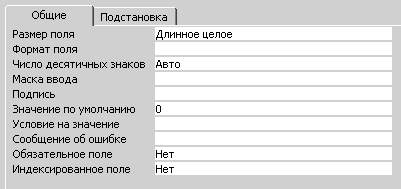
Малюнок 1

1. **Маска ввода**. За допомогою спеціальних символів можна визначити тип символу (літера, цифра, пропуск та ін.) у кожній позиції введення. У масці введення можуть використовуватись такі кодові символи: 0 — цифра від 0 до 9; 9 — цифра чи пропуск; # — цифра, пропуск, плюс або мінус; L — буква від А до Z чи від А до Я; ? - буква; А — буква або цифра; а — буква або цифра; & —будь-який символ або пропуск; С — довільний символ; .,:;-/ - роздільники, які зберігають свій вигляд у рядку даних; < — перетворення символів у нижній регістр; > —
2. перетворення символів у верхній регістр; ! — маска має заповнюватися справа наліво; \ — введення наступного після зворотної похилої лінії символу як символьної константи.
3. **Подпись**. По суті це друге ім'я поля. Воно задається рідко.
4. **Значение по умолчанию**. Дозволяє автоматично вставляти рядки, які часто повторюються, у всі новостворені поля.
5. **Условие на значение**. Дозволяє створювати фільтр, який забезпечує введення у дане поле тільки тих значень, які відповідають заданій умові.
6. **Обязательное поле**. Набуває тільки двох значень: "так" або "ні".

"Так" означає, що у дане поле обов'язково мають бути введені дані, які відповідають раніше визначеним властивостям.

**\* Пустые строки**. Набувають двох значень: "так" або "ні". "Так" означає, що порожні рядки дозволені. Тут над порожнім рядком починається рядок, який має пропуски.

**\* Индексованнoе поле**. Набуває двох значень: "так" або "ні". "Так" означає, що поле індексоване.



Малюнок 2

Зазначимо, що на початку вивчення та конструювання баз даних звичайно використовують тільки властивість "розмір поля". Інші властивості, як правило, приймаються за замовчуванням. Вони використовуються після набуття деякого досвіду роботи з базами даних.

2. Числове поле. У числове поле можуть записуватись цілі числа і числа у нормальному вигляді, тобто з плаваючою крапкою. Числові поля мають властивості, перелік яких див. на мал. 2. Аналіз цього переліку показує, що вони багато в чому збігаються з переліком властивостей текстового поля.

Розглянемо більш детально тільки властивість **Розмер поля.** Вона може набирати таких значень:

**\*** **Байт** - цілі числа у межах від 0 до 255.

**\*** **Целое** - цілі числа від - 32 768 до 32 767.

**\*** **Длинное целое** - цілі числа від -2 147 483 648 до 2 147 483 647.

**\*** **С плавающей точкой (4 байт)** - числа у межах від -3.402823 Е38 до 3.402823 Е38.

**\*** **С плавающей точкой** **(8 байт)** - числа у межах від - 1.79769313486 Е308 до 1.79769313486 Е308.

**\*** **Кодрепликации** – глобально унікальний ідентифікатор (GUID) довжиною 16 байт.

3. Поле **Дата/время.** Це поле використовують для запису дат та часу. Багато із властивостей цього поля такі самі, як і текстового поля. Властивість **Формат поля** може набирати таких значень:

**\*** **Полный формат даты** (наприклад, 12.05.97 15:20:21).

**\*** **Длинный формат даты** (наприклад, 13 серпня 1997 p.).

**\*** **Средний формат даты** (наприклад, 13 серп. 97 p.).

**\*** **Короткий формат даты** (наприклад, 13.08.97).

**\*** **Длинный формат времени** (наприклад, 15:20:21).

**\*** **Средний формат времени** (наприклад, 03:21 РМ).

**\*** **Короткий формат времени** (наприклад, 15:20).

Якщо властивість **Формат поля** набувається за замовченням, то у різні записи цього поля можна вводити різну інформацію. Наприклад в один запис цього поля може бути записано дату в такому вигляді 02/03/97, а в інший запис поля - у такому вигляді 02-03-97.

4. Логічне поле. У логічному полі може записуватись одне з таких двох значень: "так" або "ні", "хибне" чи "істинне", "ввімкнень" чи "вимкнень". Логічні поля можуть використовуватися з різною метою. Але найчастіше їх використовують в анкетних даних, де є тільки дві можливі відповіді.

5. Поле типу лічильник. Використовують як лічильник записів, його також часто використовують як ключове поле.

6. Поле типу Меmо. Це довге поле. У ньому може вміщуватися текст або комбінація тексту та чисел, яка містить 64000 символів.

7. Поле об'єкту OLE. Це поле вміщує безпосередньо не інформацію про об'єкт, а посилання на ім'я об'єкта. Як ім'я об'єкта може бути ім'я додатку, наприклад електронна таблиця Excel, редактор Word, засобами яких можуть бути створені об'єкти для впровадження їх у таблицю Access. Крім того, об'єктами можуть бути малюнки, звукозаписи та інші дані у двійковому форматі.

На початковому етапі роботи з базами даних процес їх створення доцільно починати з підготовчого етапу, а саме: з побудови на папері структури таблиць. Структуру таблиці Список (див. мал. 1. У вступі) можна побудувати так, як показано на мал. 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Размер поля** | **Другие свой-ства** | **Описание** |
| Номер 1 | Счетчик | Длинное целое |  |  |
| Ф.И.О. | Текстовий | 14 |  |  |
| Предмет | Текстовий | 12 |  |  |
| ДатаРожд | Дата/время | Кратний формат дати |  |  |
| Стаж | Числовой | Байт |  |  |
| Нагрузка | Числовой | Байт |  |  |
| Оклад | Числовой | Цалое |  |  |

Малюнок 3

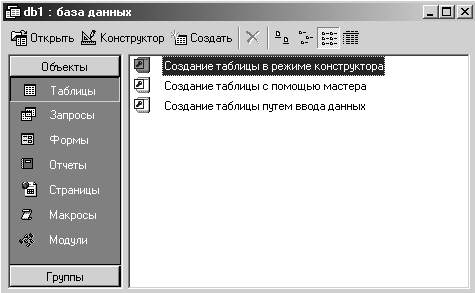
Структуру таблиці Консультация (див. мал. 2 у вступі.) можна подати так, як показано на мал. 4. Перші кроки щодо створення бази даних можуть бути різними. Ми зробимо так: запускаємо систему Access і у вікні запрошення встановлюємо перемикач у положення Новая база данних. В результаті на екрані з'явиться вікно Файл нової бази даних.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя поля** | **Тип поля** | **Размер поля** | **Другие свойства** | **Описание** |
| Номер 2 | Счетчик |  |  |  |
| Ф.И.О. | Текстовый | 14 |  |  |
| Аудитория | Числовой | Целое |  |  |
| День | Текстовый | 12 |  |  |
| Время | Текстовый | 12 |  |  |

Малюнок 4

Набираємо ім'я файла (набиремо ім'я Basa2), його тип і директорію, а потім натискаємо кнопку **Создать.** Після цього відкривається вікно **База данных,** зображене на мал. 5.

Зверніть увагу, що на екрані з'явилося головне меню системи. Воно вміщує такі меню: **Файл, Правка, Вид, Вставка, Сервис і Окно.** Багато команд головного меню системи Access дублюють ті самі операції, які можуть бути виконані іншими засобами. Оскільки більшість операцій з базами даних виконуватимемо за допомогою контекстних меню, то окремо головне меню розглядати не будемо. Після прочитання всього матеріалу ви можете повернутись до головного меню і самостійно виконати у ньому багато команд.



Малюнок 5

У вікні **База данных** натискаємо кнопку **Создать.** Реакцією системи на цю дію буде виведення на екран вікна **Новая таблица,** яке зображено на мал. 6.

У цьому вікні перераховані такі способи створення таблиць: **Режим таблицы, Конструктор, Мастер таблиц, Импорт таблиц, Связь с таблицами.**

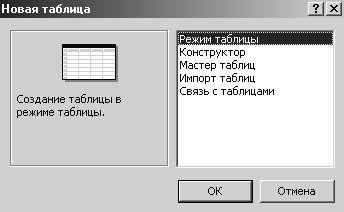
**Режим таблицы** видає бланк (форму) абстрактної таблиці. Потім вона може приймати конкретне наповнення і вміст.

Робота **Мастера таблиц** заснована на використанні заголовків таблиць, з яких користувач вибирає поле для своєї нової таблиці. Звичайно, передбачувані заготовки полегшують роботу користувача, але вони не вирішують всіх життєвих випадків. Та й навряд чи будь-яка, навіть хороша, заготовка може повністю задовольнити потреби та запити всіх користувачів.

Спосіб **Импорт таблиц** дозволяє імпортувати дані у таблицю, яка створюється, з інших файлів, у тому числі і з файлів форматів даних, які відрізняються від формату, який використовується СКБД Access.

Спосіб **Связь с таблицами** дозволяє шляхом установлення зв'язків з файлом формату інтегрувати дві інформаційні системи.

З усіх способів створення таблиць, на наш погляд, найзручнішим є спосіб (режим), який називають **Конструктор.** Він особливо зручний на етапі початкового вивчення СКБД Acсess. Саме цим способом ми і будемо користуватися.

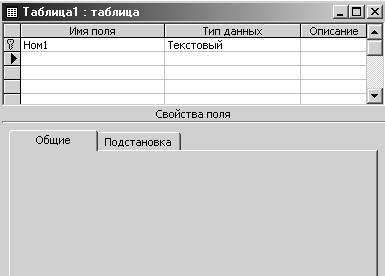


Малюнок 6

У вікні **Новая таблица** вибираємо опцію **Конструктор** і натискаємо кнопку **ОК**. На екрані з'явиться вікно таблиці 1 у режимі конструктора, зображене на мал. 7.

Заповнюємо цю таблицю тими даними, які показані на мал. 6. Звернемо увагу на те, що для установки розміру поля необхідно підвести курсор миші в рядок **Размер поля**, клацнути клавішею миші, потім установити потрібний розмір поля і натиснути клавішу **Enter**. Кожна таблиця повинна вміщувати одне або декілька полів, що однозначно визначають кожен запис в таблиці. Такі поля називають первинним ключовим полем таблиці. Якщо для таблиці визначений первинний ключ, то Microsoft Access не допускає дублювання ключа або введення значень Null в ці поля. В Microsoft Access можна виділити три типи ключових полів: лічильник, простий ключ і складний ключ.

Після заповнення всієї таблиці необхідно виділити ключове поле, яким тут буде поле **Номер 1**. Для цього клавішами переміщення курсору вказівник переміщується у відповідне поле, потім курсор миші потрібно перевести на кнопку **Ключове поле** і клацнути клавішею миші. При цьому поряд з іменем цього поля з'явиться зображення ключа.



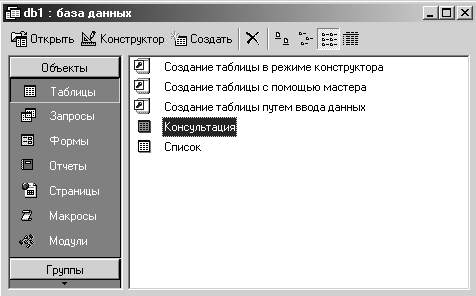
Малюнок 7

Збережемо структуру таблиці та введемо ім'я **Список**.

Аналогічно розглянутому створюється і зберігається структура таблиці Консультація.

Заповнення даними створених структур таблиць може здійснюватись різними способами.

Наприклад, у вікні запрошення системи вмикається перемикач Открыть базу даных і натискається кнопка **ОК.** Далі у вікні відкриття файла бази даних необхідно набрати ім'я файла бази даних (у нашому випадку **Basa2**) і натиснути клавішу Enter. На екрані з'явиться вікно бази даних **Basa2**, зображене на мал. 8. У цьому вікні потрібно відкрити ту таблицю, у яку будуть вводитись дані.



Малюнок 8

Далі потрібно натиснути кнопку **Открыть**. На екрані з'явиться бланк (форма) таблиці, в якій необхідно внести відповідні дані. При цьому для збереження даних, що вводяться, непотрібно застосовувати ніяких дій. Ці дані будуть збережені автоматично навіть при виході із системи.

Аналогічно вносяться дані і в таблицю **Консультация**. Якщо ж після відкриття вікна бази даних потрібно внести зміни у структуру будь-якої таблиці, то потрібно виділити відповідну таблицю і натиснути кнопку **Конструктор**. На екрані з'явиться раніше створена структура цієї таблиці. Після цього потрібно внести необхідні зміни і обов'язково, згідно з правилами, описаними раніше, зберегти змінену структуру.

Робота з таблицями

СКБД Access дозволяє виконувати різні операції над даними, які зберігаються у таблицях. При цьому одні і ті самі операції можна виконувати різними способами. Ми будемо розглядати лише один спосіб роботи з таблицями, заснований на використанні контекстного меню. На наш погляд це один з найпростіших способів роботи з таблицями. При цьому спочатку розглядатимемо операції тільки в межах однієї таблиці.

Контекстне меню має багато об'єктів бази даних. Кожна таблиця теж має декілька контекстних меню. Ми розглядатимемо такі з них:

\* контекстне меню стовпців;

\* контекстне меню рядків;

\* контекстне меню таблиці;

\* контекстне меню окремої комірки таблиці;

\* контекстне меню заголовка таблиці.



Малюнок 9

Для введення на екран будь-якого з цих меню потрібно виділити відповідний об'єкт, а потім клацнути правою клавішею миші. Наприклад, для виведення контекстного меню стовпця потрібно його виділити і після цього клацнути правою клавішею миші.

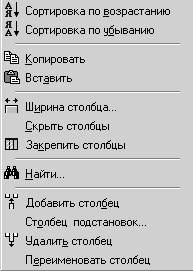
Редагування даних у будь-якій відкритій таблиці здійснюється відповідно до загальноприйнятих правил роботи в системі Windows. Нагадаємо, що для цього необхідно підвести курсор миші у відповідну комірку і клацнути її лівою клавішею. Далі використовують звичайні засоби редагування.

В кінець відкритої таблиці можна додати будь-який запис. Якщо потрібно змінити структуру таблиці, то для цього потрібно її відкрити у режимі конструктора. Після цього можна змінювати імена полів та типи даних, додавати нові поля та ін.

Перейдемо тепер до роботи з таблицями за допомогою контекстних меню.

1. Контекстне меню стовпців. При роботі з цим меню експеримент будемо проводити з полем. Предмет у відкритій таблиці Список, яка зображена на мал. 9.

Перш за все активізуємо поле Предмет, для чого установимо курсор миші на його ім'я і клацнемо її лівою клавішею (для виділення кількох стовпців ця операція виконується при натискуванні клавіші Shift).



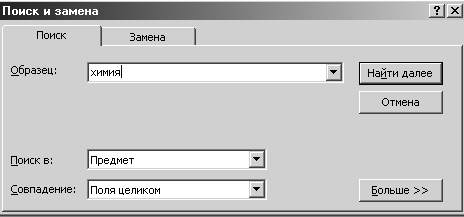
Малюнок 10

Потім клацнемо правою клавішею миші і на екрані з'явиться контекстне меню поля, яке зображено на мал. 10.

Один з найбільш ефективних засобів організації даних є сортування. Сортування впорядковує дані за змістом визначених полів. Інформацію можна сортувати за алфавітом, числовими значеннями, датами та іншими параметрами. Записи сортуються або в порядку зростання (А-Я и 0-9), або у порядку зменшення (Я-А и 9-0).

Команди **Сортировка по возростанию** і **Сортировка по убыванию** використовуються для сортування записів у таблиці відповідно до зростання чи зменшення тих даних, які вміщує вибране поле.

Команда "**Найти ...**" відкриває вікно **Поиска в поле**. У даному випадку відкриється вікно **Поиск в поле: "Предмет"**, зображене на мал. 11.



Малюнок 11

Після заповнення цього вікна необхідними даними натискають кнопку **Найти далее**. У результаті буде виділено у таблиці **Список** те перше значення поля **Предмет**, яке було введене у рядок **Образец**. Для продовження пошуку значень, які повторюються, необхідно натиснути кнопку **Найти далее**.

Команда **Ширина столбца...** дозволяє установити на екрані необхідну ширину вибраного стовпця.

Команда **Скрыть столбцы** ховає виділені стовпці. На екрані вони стають невидимими. Для відновлення на екрані схованих стовпців у контекстному меню заголовка таблиці необхідно виконати команду **Показать столбцы...**

Команда **Закрепить столбцы** закріпляє виділені стовпці з лівого боку екрана, перешкоджаючи їм вийти за край екрана при прокручуванні.

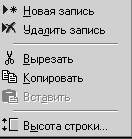
Команда **Переименовать столбец** дозволяє змінити ім'я виділеного поля безпосередньо у відкритій таблиці. Цю команду потрібно застосовувати дуже акуратно, оскільки це може вимагати зміни тих дій, які мали посилання на старе ім'я поля.

При виконанні команди **Вставить столбец** всі стовпці таблиці, починаючи з виділеного, зсуваються праворуч, а на місце виділеного вставляється порожній стовпець з іменем **Поле 1**. Потім йому можна надати конкретне ім'я і заповнити відповідними даними.

Команду **Удалить столбец** використовують для знищення в таблиці виділеного стовпця. Команда потребує підтвердження знищення.

Команду **Вставить столбец подстановка ...** використовують для виклику програми **Мастер подстановок**. Вона буде розглянута пізніше.

1. Контекстне меню рядків. Для виділення рядка курсору миші встановлюють в її крайнє ліве поле і клацають клавішею. Для виділення кількох записів клавіша миші клацається при натиснутій клавіші Shift. Після цього клацається права клавіша миші, і на екрані з'явиться контекстне меню, зображене на мал. 12.



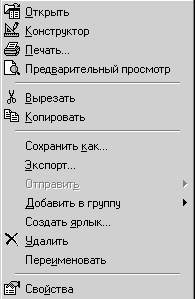
Малюнок 12

Команда **Высота строки ...** дозволяє встановити необхідну висоту не тільки вибраних, але й усіх рядків таблиці.

Команду **Вырезать** використовують для розміщення у спеціальній області пам'яті, яку називають буфером обміну, виділених рядків. Команда потребує підтвердження знищення рядків в таблиці.

За допомогою команди **Копировать** виділені записи розміщуються в буфері обміну. При цьому у самій таблиці ці записи залишаються.

За допомогою команди **Вставить** вставляється зміст буфера обміну в активну область таблиці.



Малюнок 13

3. Контекстне меню таблиці. Для виклику його на екран у відкритому вікні бази даних (див. мал. 9) установлюємо курсор миші на значок таблиці (він розташований ліворуч від імені таблиці) і клацаємо правою кнопкою миші. Це контекстне меню зображено на мал. 13.

Команда **Открыть** і **Конструктор** мають такі самі призначення, як і однойменні кнопки у вікні бази даних.

Команда **Вырезать** вміщує виділену таблицю в буфер обміну. При цьому з екрана знищується ім'я таблиці. Будьте уважні при виконанні цієї команди, тому що можна легко загубити таблицю.

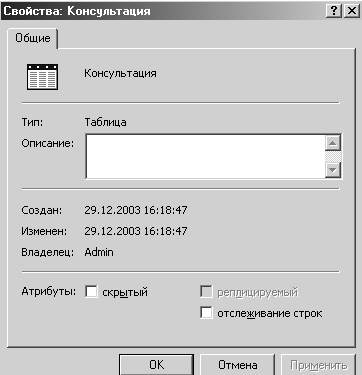
За командою **Копировать** уміщується виділена таблиця в буфер обміну, залишаючи її на екрані, а за командою **Удалить** - знищується без розміщення її в буфер обміну.

За командою **Печать** - друкується виділена таблиця.

За допомогою команди **Переименовать** можна змінити ім'я виділеної таблиці.

Команда **Создать ярлык ...** дозволяє створити та розмістити ярлик таблиці (піктограму) на робочому столі Windows. Так можна безпосередньо з Windows відкрити таблицю, і швидко отримати доступ до її даних.

Якщо з базою даних працює кілька чоловік, то для деяких з них користування окремими таблицями може бути заборонено. У цьому випадку таблиці можна сховати. Для цього потрібно у вікні **База даних** виділити ім'я відповідної таблиці, викликати контекстне меню таблиці і виконати команду **Свойства...** . У результаті з'явиться вікно **Свойства...,** забражене на мал. 14.



Малюнок 14

У цьому вікні необхідно встановити атрибут **Скрытый** і натиснути кнопку **ОК,** після чого значок цієї таблиці стане блідим. Тепер розкриємо пункт меню **Сервис** і виконаємо в ньому команду **Параметри** .... У відкритому вікні **Параметри** виберемо прапорець **Скрытые объекты** і натиснути кнопку ОК. У результаті ім'я таблиці буде знищено з вікна **База данных.**

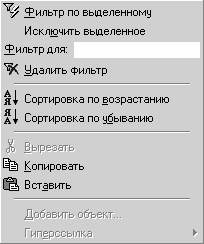
Щоб відновити ім'я цієї таблиці у вікні бази даних необхідно знову відкрити меню команди **Сервис,** виконати команду **Параметри** ..., у відкритому вікні **Параметры** встановити прапорець **Скрытые объекти і** натиснути кнопку **ОК.** З'явиться ім'я таблиці з блідим прапорцем. Після цього потрібно у контекстному меню цієї таблиці знищити атрибут **Скрытый.**

4. Контекстне меню окремої комірки таблиці. Це меню зображене на мал. 15. Для виклику цього меню курсор миші помішується в комірку таблиці і клацається права кнопка.

Один з найбільш ефективних засобів організації даних є фільтрація. Фільтрація дозволяє обмежити діапазон перегляду записів. Купа умов відбору називають фільтром. Сортування та фільтрація можуть виконуватись як в режимі форми так і в режимі таблиці. В результаті цих процедур відображення даних на екрані змінюється, хоча реальні дані залишаються незмінними.

Команда **Фильтр по выделенному** відбирає записи на основі поточного виділеного блоку. Наприклад, якщо у полі **Предмет** виділено значення - **Физика** та **Химия** - і після цього буде виконана команда **Фильтр по выделенному,** то в таблиці залишаться лише ті записи, які вміщують ці зміни.

Команда **Исключить выделенное** використовує фільтр стосовно всіх записів, за винятком виділених. Наприклад, якщо в полі **Предмет** виділено - **Математика** - і після цього виконана команда **Исключить выделенное,** то у таблиці залишаться всі записи, за винятком тих, які у полі **Предмет** мають - **Математика.**



Малюнок 15

Виконання команди **Удалить фильтр** дозволяє відновити вміст всієї таблиці. Наприклад, якщо ця команда виконується після попередньої, то у таблиці відновляться записи, які вміщують у полі **Предмет - Математика.**

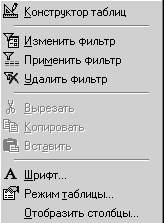
Суть інших команд цього меню зрозуміла уже з їхніх назв. Вони зустрічались у попередніх контекстних меню і не потребували особливого пояснення.

5. Контекстне меню заголовка таблиці. Воно зображено на мал. 16.

Для виклику цього меню курсор розміщується у поле імені відкритої таблиці і клацається права клавіша. Команда **Фильтр** по выделенному, **Исключить выделенное** і **Удалить фильтр** за своїм змістом принципово не відрізняються від однойменних команд попереднього меню.

Команди **Изменить фильтр** і **Применить фильтр** використовують разом. За першою командою змінюються умови фільтрації записів, а за другою - здійснюється власне фільтрація записів таблиці.

Якщо ми виберемо команду **Изменить фильтр** ми маємо можливість змінити умови вибору. На панелі вибираємо меню **Фильтр** та команду **Розширений фильтр.** Додаємо у бланк фільтру критерії для інших полів нашої таблиці, а також задаємо для них принципи сортування та вводимо потрібні вирази напроти комірки умови відбору. Натискаємо кнопку **Применить фильтра**. Ваш Розширений фільтр збережеться для подальшого використання та відобразиться у діалоговому вікні під час подальшого вибору команди Розширений фільтр. Ви маєте змогу видалити його, змінити або доповнити.



Малюнок 16

За командою **Шрифт ...** змінюється шрифт у таблиці. За замовчуванням використовується шрифт Агіаl Суr з атрибутом **Обычный** та розміром 10 пунктів.

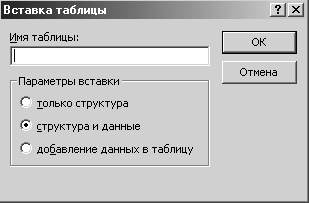
Використання команди **Отобразить столбцы ...** дозволяє відновити у таблиці сховані стовпці.

Команда **Отобразить** столбцы звільняє всі закріплені стовпці у режимі таблиці. Звільнені стовпці залишаються у тому самому порядку, в якому вони знаходились при закріпленні.

Крім розглянутих операцій з даними всередині однієї таблиці СКБД Access має широкий набір засобів для роботи з даними, які зберігаються у кількох таблицях. Можна, наприклад, копіювати рядки і стовпці однієї таблиці в іншу, переносити прямокутні ділянки з однієї таблиці до іншої, створення копії таблиць чи їх структур та ін.

Розглянемо порядок створення копій таблиць. У відкритому вікні **База данных** викликаємо контекстне меню таблиці. У цьому меню виконуємо команду **Копировать**. Підводимо курсор миші в область вікна бази даних і клацаємо правою клавішею. У контекстному меню, що з'явиться вікно **Вставка** **таблицы**, зображене на мал. 17.

Далі вводимо ім'я таблиці, в якій будемо створювати копію. Якщо потрібно копіювати всю таблицю, то необхідно ввімкнути перемикач **Структура и данные**. Якщо необхідно копіювати тільки структуру таблиці, то вмикається перемикач **Только структура**. Після цього слід натиснути кнопку **ОК**.



Малюнок 17

**Контрольні питання до лабораторної роботи 1:**

**1. Що таке СКБД?**

**2. Перелічіть головні об’єкти баз даних.**

**3. Які типи полів використовують в Access?**

**4. Коротко розкажіть про суть властивостей текстового поля.**

**5. Перелічіть значення які може набирати властивість Розмір поля числових полів.**

**6. Перелічіть значення які може набирати властивість Формат поля полів Дата/время.**

**7. Перелічіть способи створення таблиць.**

**8. Що таке ключове поле.**

**9. Перелічіть типи ключових полів.**

**10.Які операції можна виконувати над даними в таблицях?**

**11.Перелічіть об’єкти в таблицях які можна редагувати за допомогою контекстних меню.**

**12.В яких режимах можна працювати з таблицями?**

**13.Що таке фільтр?**

**14.В результаті фільтрації і сортування що змінюється, а що залишається незмінним?**

**15.За допомогою якої команди можна приховати потрібні стовпці вашої таблиці?**

**16.За допомогою якої команди можна відновити приховані стовпці вашої таблиці?**

**Лабораторна робота № 2**

**Тема: Створення зв'язків між таблицями. Створення схем даних. Типи відношень та цілісність даних.**

**Мета: Навчитись працювати з Схемою даних. Набути навичок створення відношень у БД.**

Зміст роботи за варіантом індивідуального завдання:

1. Візьміть дві таблиці, поля яких потрібно зв’язати та створіть зв'язок між таблицями за допомогою майстра підстановок в своїй базі даних.
2. Показати створений зв’язок на схемі даних.
3. Видалити створений зв’язок.
4. За допомогою майстра створіть стовпець підстановок, в якому відображається фіксований набір значень.
5. Відкрийте схему даних та додайте всі таблиці вашої бази даних до порожнього макету (якщо макет був не порожній звільніть його перед цим).
6. Створити відношення один-до-багатьох між таблицями своєї бази даних за допомогою Схеми даних.
7. Забезпечте цілісність для ваших даних у зв’язках.
8. Встановіть каскадне видалення
9. Встановіть каскадне відновлення.
10. Змінити існуючий зв’язок. Навчитися змінювати ключ.
11. Створити зв’язок між таблицями з відношенням багато-до-багатьох.
12. Вивести на екран всі існуючі зв’язки. Вивести на екран всі зв’язки тільки для конкретної таблиці.
13. Створіть зв’язок між таблицями з відношенням один-до-одного.

**Теоретичні відомості:**

**Зв'язування таблиць бази даних**

Коли розроблялися СКБД перших поколінь, комп'ютери мали ще малі об'єми пам'яті. Питання раціонального використання пам'яті було тоді дуже актуальним. Особливо гострою була ця проблема для збереження баз даних великих об'ємів, у яких дані часто дублювалися. Дублювання даних є причиною нераціонального використання пам'яті комп'ютера.

Прагнення розв'язати цю проблему привело до розробки процедури нормалізації. Нормалізація дозволила використовувати дублювання даних.

Хоча питання пам'яті і не стоїть так гостро, як раніше, все-таки процедура нормалізації є необхідною. Нормалізація зараз потрібна для того, щоб організувати дані таким чином, щоб їх редагування робилось тільки в одному місці бази даних. Нормалізація передбачає необхідність виконання процедури зв'язування таблиць бази даних.

Access створює реляційні бази даних, які дозволяють об’єднувати інформацію з різних таблиць.

Необхідно встановити взаємозв’язки між таблицями, записи яких логічно пов’язані. Створивши таблиці та визначивши ключі для кожної з них задаються зв’язки і на їх основі можна брати дані з кількох таблиць та вміщувати їх в одну форму, запит або звіт.

Існує два основних способи зв’язувати дані: це за допомогою полів підстановки та шляхом визначення зв’язків у діалоговому вікні Схема даних.

Між двома таблицями можна встановити два типи відношень: “один-до-одного” та “один-до-багатьох”.

Найбільш розповсюдженим є відношення “один-до-багатьох” (one-to-many), коли одному запису таблиці відповідає декілька записів в другій таблиці. У відношенні “один-до-багатьох” сторона “один” називається головною таблицею, а сторона “багато” – зв’язаною або підлеглою. Таблиця може бути зв’язаною в одному взаємозв’язку, та головною в іншому. Щоб встановити взаємозв’язок таблиць потрібно зв’язати ключ головної таблиці з співпадаючим полем у зв’язаній таблиці (зовнішнім ключем).

Відношення “один-до-одного” (one-to-one) зустрічається набагато рідше. В цьому випадку одному запису в головній таблиці відповідає один запис у зв’язаній таблиці.

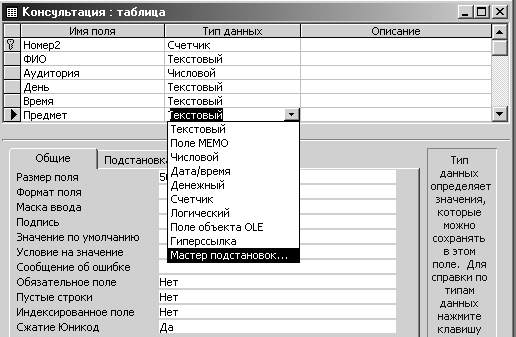
Для створення зв'язку між таблицями використовуємо **Мастер подстановок**. Порядок установлення зв'язків будемо розглядати на прикладі двох таблиць, які зберігаються у файлі **Ваsа2**. Вміст таблиці **Список** цього файла розглядали у Л.р.№1.На мал. 1- таблиця **Консультація.**



Малюнок 1

Розв'яжемо таку задачу: поле **Предмет** з таблиці **Список** підставимо у таблицю **Консультация**. Ця задача може бути вирішена так.

У вікні бази даних файла **Basa2** відкриємо структуру таблиці **Консультация** у режимі конструктора. У цій структурі створюємо ще одне поле з будь-яким іменем. Використовуватимемо ім'я **Предмет**. Далі відкриємо значення **Тип данных**, де останнім є рядок Майстер підстановок (мал. 2.).



Малюнок 2

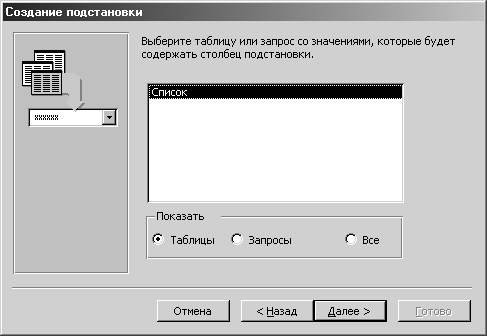
Після активізації програми **Мастер подстановок** на екрані відкривається вікно цієї програми. У цьому вікні перемикач необхідно встановити у положення **Таблица или запрос ...** і після цього натиснути кнопку **Далее**.

2. У результаті останньої дії у вікні **Мастер подстановок** з'явиться нова інформація, показано на мал. 3.

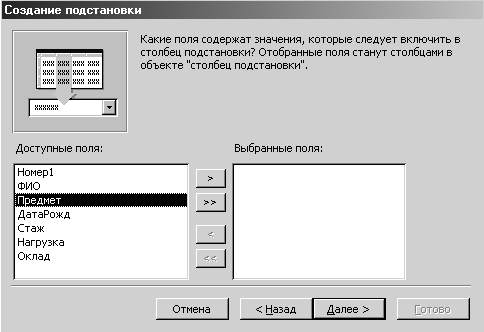
У цьому вікні будуть показані всі таблиці бази даних за винятком тієї, в яку буде зроблена підстановка поля (для розглянутого прикладу це **Консультации**). Але для файла **Basa2** це тільки одна таблиця, а саме таблиця **Список**. Виділяємо рядок, у який записано **Список**, і натискаємо кнопку Далее.

3. Вміст вікна **Мастер** подстановок зміниться і набере вигляду, який показано на мал. 4. У цьому вікні ліворуч розташовані доступні поля таблиці **Список**. Необхідно виділити те поле, над яким здійснюється операція підстановки. У даному випадку це поле - **Предмет**.

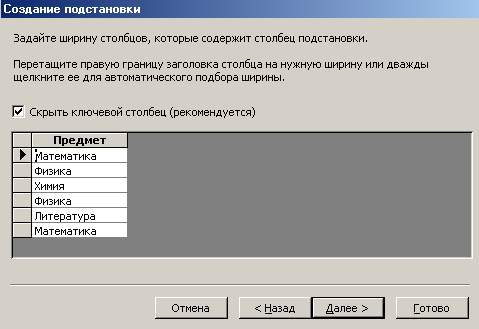
Кнопкою одинарна стрілка ліворуч поле повертається у лівий список. Така необхідність виникає, наприклад, у тому випадку, коли була допущена помилка. Кнопкою подвійна стрілка праворуч переносяться всi поля.



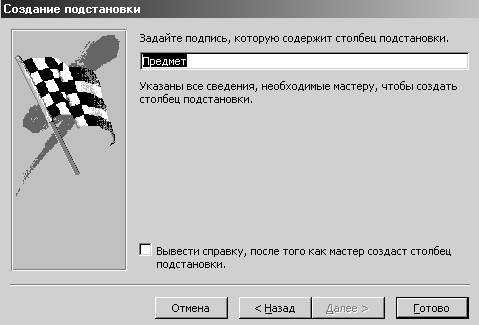
Малюнок 3



Малюнок 4



Малюнок 5



Малюнок 6

Після перенесення поля **Предмет** необхідно натиснути кнопку **Далее**.

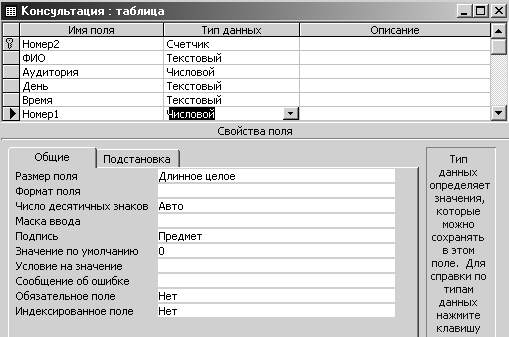
4. Вміст вікна **Мастер подстановок** знову змінюється і набирає вигляду, показаного на малюнку 5.У цьому вікні можна відрегулювати ширину стовпців (у даному випадку стовпця **Предмет**). Для цього мишу встановлюють в область заголовка поля так, щоб курсор набрав такого вигляду, який він має на мал. 5. Після цього натискують клавішу миші, і миша переміщується по поверхні столу до досягнення необхідної ширини поля. Потім клавішу відпускають і натискають кнопку **Далее**.

5. Вікно майстра підстановок набирає тепер вигляду, який зображено на мал. 6.

У цьому вікні потрібно задати підпис для стовпця підстановки. Залишимо те саме слово - Предмет і натиснемо кнопку **Готово**.

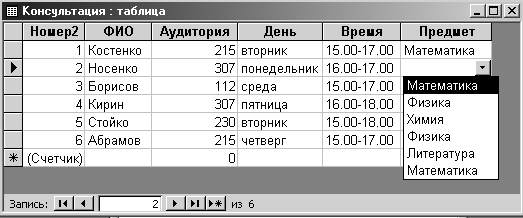
Після цього на екрані з'явиться вікно **Мастер подстановок**, у якому необхідно підтвердити необхідність збереження таблиці, тобто натиснути кнопку **Да**. У результаті виконаних дій з'явиться вікно, зображене на мал. 7.

Зверніть увагу, що у результаті виконаних операцій у структурі таблиці **Консультация** з'явиться не ім'я поля Предмет, а ім'я ключового поля таблиці **Список** (у даному випадку ім'я **Номер** 1). Це і означає, що таблиця **Список** і таблиця **Консультация** зв'язані по цьому полю. Слід звернути увагу на те, що змінився тип поля, а на вкладці **Подстановка** наведені характеристики того поля, яке приєднується до **Консультации.** Закрийте вікно структури таблиці **Консультация** і відкрийте її для перегляду. Зазначимо, що у цій таблиці з’явиться поле **Предмет.** Клацніть кнопкою миші в позиції першого рядка цього поля і розкрийте список його значень. Вікно набере вигляду, зображеного на мал. 8.



Малюнок 7

Переконайтесь у тому, що у даному списку зберігаються всі значення поля **Предмет** зв'язаної таблиці **Список.** Тепер кожне з цих значень може бути перенесено у поточну комірку поля **Предмет.**

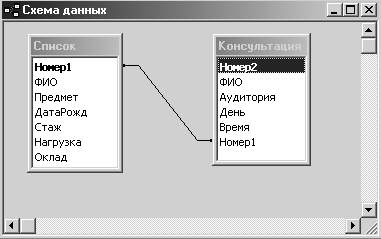


Малюнок 8

Таким чином, поставлене завдання виконано. Поле **Предмет** приєднано до таблиці **Консультация.**

Щоб переконатися в наявності зв'язку між таблицями, відкриємо меню **Сервис,** у якому виконаємо команду **Схема данных.** З'явиться вікно **Схема данных** зі зв'язками між таблицею **Список і** таблицею **Консультация,** зображене на мал. 9.

Зв'язок між таблицями у цьому вікні показано лінією, яка з'єднує ключеві поля таблиць.



Малюнок 9

Надалі робота з таблицями буде здійснена з урахуванням встановлених зв'язків.

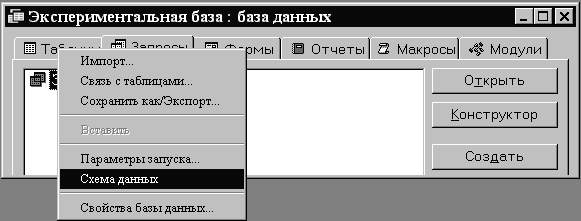
Щоб прибрати зв'язок між таблицями, необхідно клацнути лінію, яка з'єднує таблиці (див. мал. 9) так, щоб вона стала жирною, потім потрібно клацнути правою клавішею миші. З'явиться контекстне меню, в якому необхідно виконати команду **Удалить связь.** Далі потрібно підтвердити необхідність знищення зв'язку, закрити вікно **Схема данных** і продовжити роботу з базою даних.

Система Access дозволяє приєднати до іншої таблиці не тільки одне поле, але й декілька. Методика виконання такої операції аналогічна розглянутій вище.

# Створення схем даних

Як уже йшлося раніше існує два основних способи зв’язувати дані: це за допомогою полів підстановки, як було розглянуто вище, та шляхом визначення зв’язків у діалоговому вікні Схема даних.

В вікні бази даних натиснемо кнопку **Схема данных** на панелі інструментів. Відкривається вікно **Схема даних**. Якщо є збережений макет схеми даних, цей макет буде виведений на екран. Якщо збереженого макету схеми даних немає, відкривається вікно діалогу **Добавленные таблицы**, що дозволяє додати таблиці або запити в вікно **Схема данных**.



Малюнок 10

# Визначення зв'язків між таблицями

Необхідно закрити всі відкриті таблиці. Не дозволяється створення або вилучення зв'язків між відкритими таблицями. В вікні бази даних слід натиснути кнопку на панелі інструментів.

Якщо в базі даних будуть відсутні певні міжтабличні зв'язки, автоматично відкривається вікно діалогу **Добавление таблицы**. Якщо вимагається додати таблицю в вікно **Схема данных**, а вікно діалогу **Добавление таблицы** не відкрите, то потрібно натиснути кнопку на панелі інструментів.

Далі треба вибрати імена таблиць, що додаються і натиснути кнопку **Добавить** і закрити вікно діалогу.

Для зв'язування полів треба вибрати поле в одній таблиці і перемістити його за допомогою миші на відповідне поле в другій таблиці.

В більшості випадків зв'язують ключове поле (наведене в списку полів напівжирним шрифтом) однієї таблиці з відповідним йому полем (яке часто має те ж саме ім'я), що називають полем зовнішнього ключа в другій таблиці. Зв'язані поля не обов'язково повинні мати однакові імена, однак, вони повинні мати однакові типи даних і мати зміст одного типу. Крім того, поля, які зв’язують, типу **Числовой**  повинні мати однакові значення властивості **Размер поля**.

Після цього відкривається вікно діалогу **Связи**. Тут потрібно перевірити правильність імен зв'язуємих полів, які виводяться в двох стовпцях бланку в вікні зв'язків. При необхідності, вибирати в осередках бланку інші імена полів. Створюється зв'язок натиском кнопки **Создать**. Це повторюється для кожної пари зв'язуємих таблиць.

В вікні **Схема данных** користувач має можливість виконати наступні операції:

Змінити макет таблиці.

Змінити існуючий зв'язок.

Усунути зв'язок.

Усунути таблицю з вікна Схема даних.

Вивести на екран всі існуючі зв'язки або зв'язки тільки для конкретної таблиці.

При спробі закрити вікно **Схема данных** виводиться запрошення зберегти поточний макет схеми даних. Незалежно від збереження поточного макету створені зв'язки будуть збережені в базі даних.

Для зв'язування поля таблиці з іншим полем в тій же таблиці слідує двічі додати таблицю в вікно **Схема данных**.

**Зміна існуючого зв'язку**

Треба закрити всі відкриті таблиці. В вікні бази даних натиснути кнопку на панелі інструментів. Якщо таблиці, зв'язку між якими вимагається змінити, не виводяться в вікні **Схема данных**, треба натиснути кнопку на панелі інструментів, встановити вказівник на ім'я потрібної таблиці і двічі натиснути кнопку миші. Після цього натиснути кнопку **Закрыть**.

Встановимо вказівник на лінію зв'язку, яку вимагається змінити, і двічі натиснемо кнопку миші. Виберемо параметри зв'язку.

**Вилучення зв'язку**

Закриємо всі відкриті таблиці. В вікні бази даних натиснемо кнопку на панелі інструментів. Встановимо вказівник на лінію зв'язку, яку вимагається усунути, і натиснемо кнопку миші (вибрана лінія зв'язку стане жирною), після цього натиснемо клавішу **DEL.**

**Перегляд існуючих зв'язків**

Якщо вікно **Схема данных** не відкрите, натиснемо кнопку на панелі інструментів. Для перегляду всіх існуючих в базі даних міжтабличних зв'язків натискаємо кнопку на панелі інструментів.

Для перегляду зв'язків, певних для конкретної таблиці, виберемо таблицю і натиснемо кнопку на панелі інструментів.

Якщо всі таблиці, зв'язані з виділеною таблицею, вже виведені в вікно Схема данных, натиск кнопки **Показать прямые связи** не має ефекту, оскільки ця кнопка здатна тільки виводити на екран ще не виведені таблиці. Для того щоб на екрані були уявлені зв'язки тільки конкретної таблиці, слід заздалегідь натиснути кнопку на панелі інструментів або вибрати в меню **Правка** команду **Очистить** **макет**. Після цього додайте в вікно потрібну таблицю і натисніть кнопку **Показать прямые связи**. Для додавання таблиці натисніть кнопку на панелі інструментів, встановіть вказівник на ім'я таблиці, двічі натисніть кнопку миші і натисніть кнопку **Закрыть**.

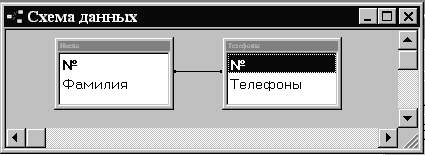
**Створення або зміна ключа**

В режимі конструктора таблиці виділимо поле або поля, що визначаються як ключові. Натиснемо кнопку **Ключ** на панелі інструментів.

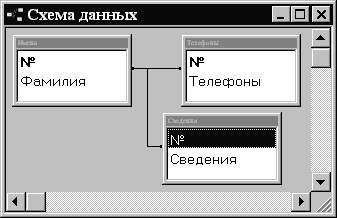
Допускається призначення ключовим полем поля, що вже містить дані, однак, якщо в цьому полі є значення що повторюються або порожні значення, буде виведене повідомлення про помилку.

Отримавши таке повідомлення користувач має три можливості: виконати запит на пошук записів, які повторюються для пошуку записів, що містять порожні значення або повторюються в полі, і змінити ці значення; вибрати інше поле або додати поле лічильника і визначити його як ключове.

Нижче наведені приклади зв'язку двох і трьох таблиць по ключовому полю «№» з першої таблиці.



Малюнок 11а



Малюнок 11б

За допомогою вікна **Схема данных** можна задавати як прості відношення, так і складні взаємозв’язки особливо в таких випадках:

-Якщо треба встановити цілісність даних;

-Якщо ключ вміщує більше ніж одне поле;

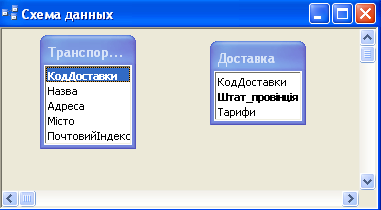
-Якщо немає прямого зв’язку між двома таблицями.

**Встановлення взаємозв’язків**

Вікно **Схема данных** (Relationships), що розглядалось вище , дозволяє зв’язувати поля з різними іменами та бачити загальну картину відношень між таблицями бази даних.

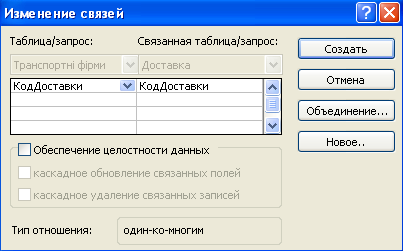
Для приклада будемо працювати з таблицями Доставка, яка містить дані про тарифи на перевозку товарів та Транспортні фірми, яка містить дані про фірми, які займаються перевозками. Вартість перевозки залежить від транспортної фірми та пункту перевезення. Створимо наші таблиці будь-яким способом та заповнимо їх даними. Оскільки кожна фірма перевозить у різні регіони, то одній фірмі відповідають декілька тарифів, тобто між таблицями зв’язок “один-до-багатьох”.

Відкриємо Схему даних. Якщо вікно схеми даних не пусте, натискаємо на кнопці **Очистить макет.** На панелі інструментів натискаємо на кнопці **Отобразить таблицу** та додаємо таблиці Транспортні фірми та Доставка (мал.12).



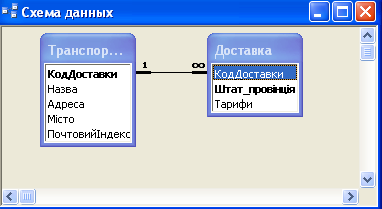
Малюнок 12

Встановимо взаємозв’язок між таблицями Транспортні фірми та Доставка, щоб зв’язати інформацію про транспортні фірми з даними про тарифи на перевезення. Для цього в вікні **Схема данных** перетягуємо поле Код доставки з таблиці Транспортні фірми в таблицю Доставка. (Напрямок перетягування ключа визначає, яка таблиця є головною, а яка зв’язаною. Таблиця, звідки переміщується ключове поле, є головною. Таблиця куди воно переміщується, - зв’язаною.) З’являється діалогове вікно **Изменение связей** із значенням “один-до-багатьох” в полі **Тип отношения** (мал. 13).



Малюнок 13

В діалоговому вікні **Изменение связей** встановіть прапорець в поле **Обеспечение целостности данных**. Цілісність не допускає появи незв’язаних записів. Тепер натискаємо **Создать**. Таблиця Транспорті фірми тепер зв’язана з таблицею Доставка. Зв’язуючі поля обох таблиць з’єднані лінією, поміченою цифрою1 і символом нескінченності, що вказує про відношення “один-до-багатьох” (мал. 14).



Малюнок 14

**Контроль інформації шляхом забезпечення цілісності даних**

Якщо таблиці зв’язані, то зміни у одній з таблиць впливають на дані в іншій. Це забезпечується цілісністю даних, яка представляє собою систему правил, які виключають вільну зміну зв’язаних записів.

Цілісність даних накладає наступні обмеження на введення та корективи інформації.

* Не можна вводити значення у зовнішній ключ зв’язаної таблиці, якщо не існує співпадаючого значення у ключовому полі головної таблиці. Наприклад, ви не зможете додати запис у таблицю **Закази**, вказавши **Код замовника**, якого немає в таблиці **Замовники** (не можна ввести заказ для неіснуючого замовника).
* Забороняється видаляти записи в головній таблиці, якщо існують відповідні записи в зв’язаній таблиці. Не можна, наприклад, видаляти запис в таблиці **Замовники**, якщо їй відповідають записи в таблиці **Закази** (не можна видалити замовника, у якого є закази).
* Не можна змінити значення ключового поля в головній таблиці, якщо в зв’язаній таблиці є записи, які посилаються на це значення. Так, наприклад, вам не вдасться змінити **Код замовника** в таблиці **Замовники**, якщо в таблиці **Закази** існують записи для цього замовника.

Цілісність даних можна забезпечити з виконанням наступних умов.

* Співпадаюче поле в головній таблиці повинно бути ключовим.
* Всі значення зовнішнього ключа зв’язаної таблиці повинні бути присутніми в ключовому полі головної таблиці.
* Зв‘язані поля в обох таблицях повинні мати однаковий розмір і тип даних.
* Обидві таблиці повинні належати одній базі даних Аccess. Якщо зв’язані таблиці знаходяться в різних файлах, вони повинні мати формат Microsoft Access (.mdb), а база даних, у якій зберігаються зв’язані таблиці, повинна бути відкритою.

**Каскадне відновлення та видалення даних**

В Ассess існує можливість обійти обмеження на зміну ключових полів і видалення даних у зв’язаних таблицях не порушуючи цілісність даних.

Каскадне відновлення забезпечує розповсюдження змін в головній таблиці на відповідні записи в зв’язаній таблиці. Так, наприклад, зі зміною коду замовника в таблиці **Замовники** відбудеться відновлення співпадаючого поля в таблиці **Закази**, і зв’язок між таблицями збережеться.

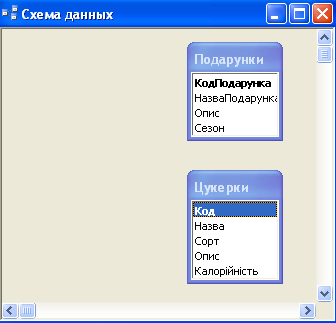
Якщо задати каскадне видалення, з видаленням запису в головній таблиці видаляються відповідні йому записи в зв’язаних таблицях. Наприклад, видалення замовника з таблиці **Замовники** включає видалення всіх його заказів з таблиці **Закази**, а також видалення даних про склад заказів в таблиці **Подробиці заказів**. Тому що в цьому випадку видаляться і не виконані закази, каскадне видалення слід використовувати з обережністю.

Відношення багато-до-багатьох

Відношення багато-до-багатьох передбачає, що кожному запису в одній таблиці відповідає декілька записів в іншій. При цьому кожна сторона відношень виглядає як відношення один-до-багатьох. Але якщо розглядати взаємозв’язок таблиць з обох сторін, стає очевидним, що ні одна з таблиць не може бути головною і для їх зв’язування необхідна третя таблиця.

Зв’зуюча таблиця представляє собою проміжну таблицю, яка служить мостом між двома таблицями у відношеннях багато-до-багатьох. Її ключ складається з ключових полів цих таблиць, з кожною з яких вона зв’язана відношенням багато-до-багатьох. Окрім ключових полів, зв’язуюча таблиця повинна мати хоча б одне поле, якого немає у зв’язуємих таблицях, але яке має значення для кожної з них. Таким чином, відношення багато-до-багатьох складається з відношень багато-до-одного та один-до-багатьох.

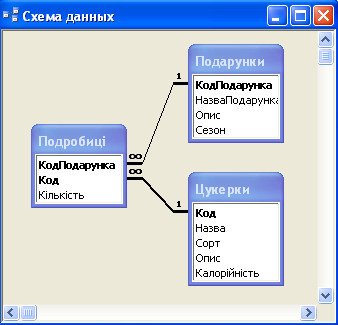
Скористуємось Схемою даних та на прикладі розглянемо, як за допомогою таблиці Подробиці створити відношення багато-до-багатьох для таблиць Подарунки та Цукерки (мал. 15).



Малюнок 15

На панелі інструментів Схема даних натискаємо на кнопку **Отобразить таблицу** щоб додати наші таблиці. Між таблицями Подарунки та Подробиці існує відношення один-до-багатьох через ключове поле Код подарунка. Крім цього таблиця Подробиці з таблицею Цукерки теж зв’язані відношенням багато-до-одого через складний ключ , який складається з полей Код подарунку та Код цукерки. На панелі інструментів Схема даних натискаємо на кнопку **Отобразить прямые связи**. З’являться лінії зв’язку, які вказують типи відношення між цими таблицями. Перетягнемо таблиці так, щоб схема краще продивлялась, щоб таблиці і лінії зв’язку не накладались.

Таблиця Подробиці виступає зв’язуючою між таблицями Подарунки та Цукерки (мал. 16).



Малюнок 16

**Контрольні питання до лабораторної роботи 2:**

**1.Для чого потрібна нормалізація?**

**2.Перелічіть всі форми нормалізації.**

**3.Для чого потрібно створювати міжтабличні зв’язки?**

**4.Перелічіть способи зв’язувати дані між двома таблицями в Access.**

**5.Які типи відношень можна встановити між двома таблицями?**

**6.Які можливості розкриває майстер підстановок?**

**7.Які існують два основні способи зв’язувати дані між таблицями в Access?**

**8.Як виглядає вікно Схема даних?**

**9.Як визначити зв’язки між таблицями?**

**10.Як виглядають зв’язані таблиці на Схемі даних?**

**11.Як змінити зв’язок між таблицями?**

**12.Які операції користувач має можливість виконати в вікні Схема даних?**

**13.В яких випадках задаються взаємозв’язки за допомогою вікна Схема даних?**

**14. Які типи відношень існують?**

**15. Як створити відношення один-до-багатьох?**

**16. Як створити відношення один-до-одного?**

**17. Як створити відношення багато-до-багатьох?**

**18. Якщо існує взаємозв’язок між таблицями, яку таблицю називають головною, а яку підлеглою?**

**19. Що таке цілісність даних?**

**20. Які обмеження на введення та корективи інформації накладає цілісність?**

**21. Що таке каскадне відновлення?**

**22. Що таке каскадне видалення?**

**Лабораторна робота № 3.**

**Тема: Проектування та можливості QBE запитів. Запити на вибірку.**

**Мета: Навчитися проектувати прості запити та застосовувати можливості QBE запитів.**

Зміст роботи за варіантом індивідуального завдання.:

1. Створити запит за допомогою майстра простих запитів. Виберіть «подробний» звіт.
2. Створити запит за допомогою майстра простих запитів. Виберіть «ітоговий» звіт.
3. Внесіть корективи у свої запити за допомогою. Конструктора запитів.
4. Побудувати QBE-запит в режимі конструктора, який виводить записи кількох полів вашої таблиці:

а) вилучіть останнє поле з бланку запиту, додайте інше поле до бланку запиту;

б) зробіть, щоб не всі поля, які входять до бланку запиту, відображались у динамічній таблиці.

1. Відсортуйте:

а) текстове поле з таблиці за зростанням та спаданням за допомогою QBE-запиту;

б) числове поле з таблиці за зростанням та спаданням за допомогою QBE-запиту.

1. За допомогою запитів, створених для відповідних таблиць БД відобразіть записи:

а) прізвища працівників чи клієнтів, чи виконавців, які починаються з певної літери;

б) людей, які не проживають, не працюють або не були у певному місті;

в) події, які здійснювались в інтервалі часу з певного періоду до іншої дати;

г) ціна товару або інше число більше 10, але не 100.

(Задайте умову відбору для поля вашої таблиці, використайте Построитель виражений для своїх виразів у запитах вашої бази даних, використайте такі операції: >, <, =, <>, And, Or, Not, Like).

1. За допомогою QBE-запиту опрацюйте зміст числового поля нашої таблиці, використовуючи статистичні функції. Наприклад, вивести суму (або середнє значення, або мінімальне значення тощо), яку отримав кожен працівник за весь час роботи з таблиці Журнал видачі.

|  |  |  |  |
| --- | --- | --- | --- |
| № вид. | ПІБ | Отримав | Дата |
| 1 | Іванов В.А. | 100 | 1.01.2010 |
| 2 | Іванов В.А. | 200 | 3.03.2010 |
| 3 | Петров А.Б. | 500 | 5.03.2010 |

|  |  |  |
| --- | --- | --- |
| Таб.ном. | ПІБ | Отримав |
|  |  | SUM |
|  |  |  |

**Теоретичні відомості:**

**Створення та використання запитів**

Запит у Access використовують для пошуку з однієї чи кількох таблиць та видачі на екран даних, які відповідають умові, визначеній користувачем.

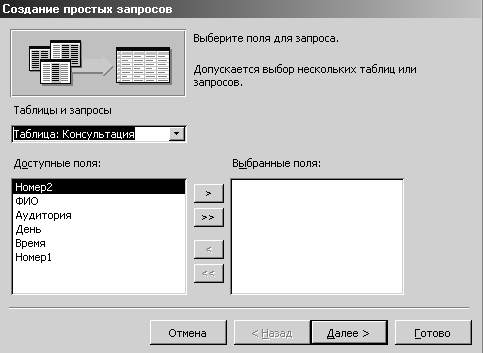
Найпростіший спосіб пошуку деяких даних у таблиці ми вже розглядали, коли описували контекстне меню стовпця.

Однак система Access має значно ширші можливості з пошуку різних відомостей з бази даних. Розглянемо спочатку спосіб реалізації запитів, який в даній системі називається **Простой запрос.** Хоч цей спосіб і має назву простого, він, з одного боку, найуживаніший, а з другого знаючи принципи його реалізації, можна досить легко самостійно розібратися і в інших способах. Цей спосіб реалізується майстром **Простой запрос.**

Майстер **Простой запрос** дозволяє із зв'язаних таблиць бази даних отримати нову таблицю, в яку включені окремі поля різних таблиць. Наприклад, із зв'язаних таблиць **Список і Консультация** можна отримати запит у вигляді таблиці з іменем **Сведения,** в якій будуть знаходитися поля **Фио, Предмет, Стаж і Оклад** таблиці **Список** і поле **День** таблиці **Консультация.** Після цього із запиту **Сведения** можна отримати конкретні дані.

Для більшої наочності викладення роботу програми будемо розглядати в такій послідовності.

1. Запускаємо майстра простих запитів. Для цього у вікні **База данных** необхідно натиснути на пункт Запити **Запросы** на панелі об’єктів, потім на панелі інструментів натиснути кнопку **Создать,** вибрати **Простой запрос** і клацнути кнопку **ОК.** На екрані з'явиться вікно **Создание простых запросов,** зображене на мал. 1.



Малюнок 1

2. Далі необхідно відкрити список таблиць у вікні таблиці **Запросы** і виділити ту з них, у якій необхідно вибирати відповідні поля. Потім ці поля потрібно перенести в таблицю вибраних. Аналогічні операції виконуються і з іншими таблицями.

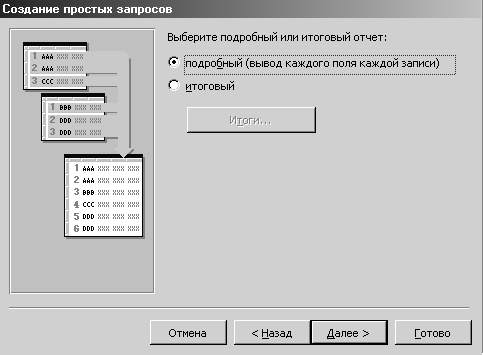
Для прикладу, що розглядається, в область вибраних з таблиці Список переносяться поля **Фио**, **Предмет**, **Стаж** і **Оклад**, а з таблиці **Консультация** - поле **День**.

3. Потім необхідно натиснути кнопку Далее. У результаті вміст вікна Создание простых запросов зміниться і набере вигляду, показаного на мал. 2.

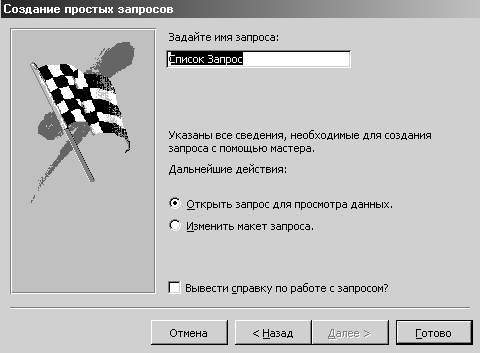
4. З двох перемикачів **Подробный** і **Итоговый** включаємо **Подробный** і натискаємо кнопку **Далее**. З'явиться вікно, показане на мал. 3.

5. Введемо ім'я запиту **Сведения** і натиснемо кнопку **Готово**. У результаті на екрані з'явиться запит з іменем **Сведения**, показаний на мал. 4.

Зверніть увагу, що в цю таблицю введені саме ті поля двох перших таблиць, які ми зазначили вище.

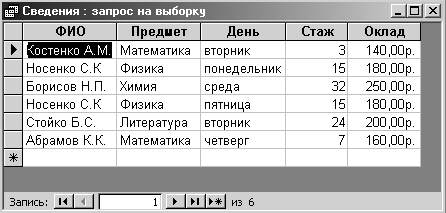


Малюнок 2

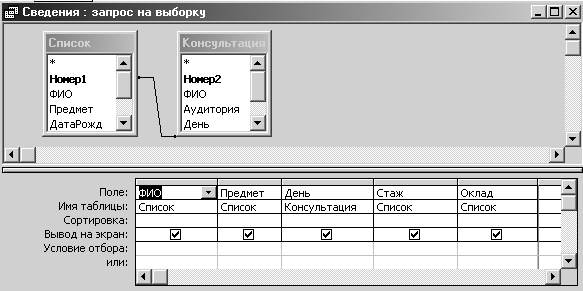


Малюнок 3

Тепер з отриманої таблиці зробимо конкретний вибір, а саме - запит про вибір математиків, які консультують у вівторок.



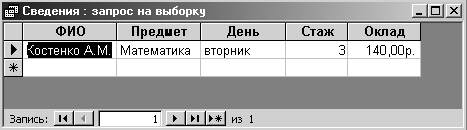
Малюнок 4



Малюнок 5

Для цього закриємо таблицю. З'явиться вікно База даних. У ньому виділимо Сведения і натиснемо кнопку Конструктор. На екрані з'явиться вікно Запрос на выборку, зображене на мал. 5.

У рядку **Условие отбора** у полі **Предмет** введемо - **Математика,** а у цьому самому рядку в полі **День - вторник.** Після цього на панелі інструментів натиснемо кнопку зі знаком оклику. На екрані з'явиться таблиця, показана на мал. 6.



Малюнок 6



Малюнок 7

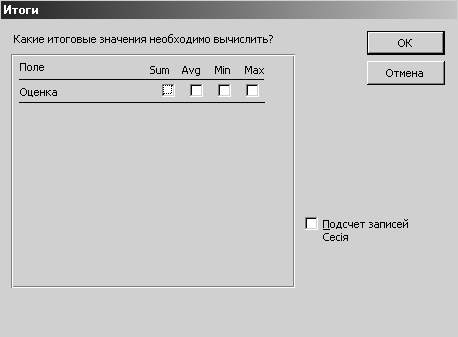
Нагадаємо, що розглянута методика виконання простого запиту здійснювалась з урахуванням того, що був ввімкнений перемикач **Подробний** (мал.2).Розглянемо тепер порядок створення запиту при увімкненому перемикачі **Итоговый ... .**

Створення підсумкового запиту буде розглядатися на прикладі бази даних з іменем **Basa3,** у якому зберігається наступна таблиця з іменем **Сессия:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Фио | Предмет | Оценка | Дата |
| 1  2  3  4  5  6 | Вахно А.А.  Бирюзов И.И.  Вахно А.А.  Бирюзко Б.Б.  Бирюзко Б.Б.  Вахно А.А. | Математика  Математика  Физика  Математика  Химия  Химия | 3  5  3  5  4  3 | 14.07.97  14.07.97  20.07.97  14.07.97  25.07.97  25.07.97 |

Створимо запит **Сессия 1,** у який включимо поля, **Фио**, **Оценка.**

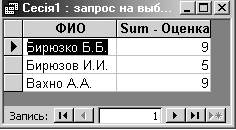
Процес створення запиту починаємо так само, як це робили раніше. Коли з'явиться вікно, зображене на мал. 2, вмикаємо перемикач **Итоговый**, потім натискаємо кнопку **Итоги ...** . У результаті з'явиться вікно, зображене на мал. 8.



Малюнок 8

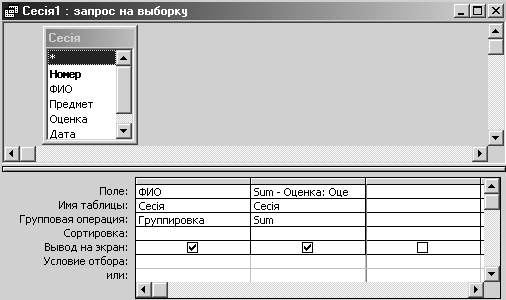
Зверніть увагу, що в цьому вікні показано тільки числове поле. Необхідно ввімкнути прапорці для тих полів, для яких необхідно отримати підсумкові значення. Вмикаємо, наприклад, для поля Оценка пропарець у колонці Sum. Після цього натискаємо кнопку **ОК.** З'явиться знову вікно, зображене на мал. 2, у якому необхідно натиснути кнопку **Далее**.

У вікні, яке відкриється, введемо ім'я запиту **Сессия 1** і натиснемо кнопку **Готово**. З'явиться вікно (мал. 9), у якому зображена ця таблиця-запит.



Малюнок 9

Закриємо тепер таблицю-запит **Сессия 1**, виділимо її у вікні бази даних і натиснемо кнопку Конструктор. З'явиться вікно, зображене на мал. 10. Це вікно дозволяє сформувати нові запити. У цьому вікні є ще рядок **Групповая операция**. Під груповою операцією розуміємо обробку значень даного поля за допомогою певної операції. Цей рядок використовують для виклику списка доступних операцій для кожного поля і для вибору з нього необхідного. Для цього курсор миші встановлюють у рядок **Групповая операция** в колонку відповідного поля і клацають клавішею миші.



Малюнок 10

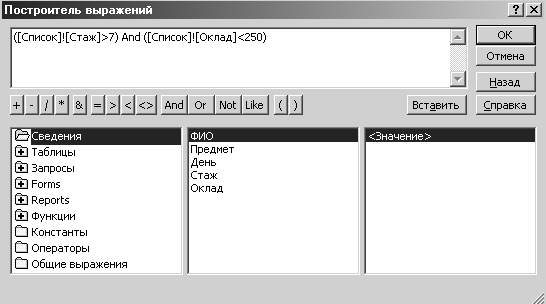
Припустимо, що необхідно підрахувати - скільки разів зустрічається кожне прізвище у таблиці **Сессия 1**. Для цього потрібно створити у вікні, зображеному на мал. 10, ще одне поле з іменем **Фио** та вибрати для нього групову операцію Count (підрахунок). Після цього на панелі інструментів потрібно натиснути кнопку зі знаком оклику і на екрані з'явиться бланк-запит, зображений на мал. 11.



Малюнок 11

У системі Access є також можливість пошуку інформації, використовуючи логічні вирази. Створення цих виразів здійснюється за допомогою програми, яку називають **Построитель выражений.** Цяпрограма створює нову таблицю, зміст якої відповідає заданому виразу.

Роботу програми **Построитель выражений** будемо розглядати на прикладі таблиці-запиту **Сведения.** Для запуску програми необхідно викликати контекстне меню рядка **Условия выбора**  У цьому меню потрібно виконати команду **Построить...** . У результаті з'явиться вікно **Построитель выражений,** зображене на мал. 12.



Малюнок 12

Це вікно складається з чотирьох частин. Верхню частину використовують для введення логічного виразу. Тут розташовані також кнопки управління вікном, кнопки зі знаком математичних операцій та логічних операцій. Ці знаки переносяться у вираз, який створюється, клацанням на них клавішею миші.

Троє нижніх вікон використовують для вибору елементів, які у них знаходяться. Ліворуч розташовані об'єкти бази даних: таблиці, запити та ін., а також функції, константи і оператори, які можуть бути використані у виразі.

У середній частині перераховані елементи вибраного об'єкта ліворуч. Наприклад, якщо ліворуч виділено ім'я таблиці-запиту, то у середній частині буде перелічено імена її полів. Для переносу їх в вираз необхідно два рази клацнути клавішею миші на відповідному імені.

Праву нижню частину вікна використовують для вибору конкретних значень тих об'єктів, які вказані в середній частині.

У разі конструювання логічних виразів потрібно дотримуватися певних правил. Основні з них: ім'я об'єкта бази даних (наприклад ім'я таблиці) відокремлюється від імені поля знаком оклику; імена беруться у квадратні дужки; дані символьного типу беруться у лапки. При цьому, якщо ви не самі набираєте вказані об'єкти та значення, а вибираєте їх з тих, які показані у вікні, то Access сама дотримується правил синтаксису.

Приступимо тепер до конструювання виразів (див. мал. 12). Введемо такий вираз:

**[Предмет] = Математика And [День] = Вторник**

Коли набрали весь вираз, натискаємо кнопку **ОК**. З'явиться вікно запиту. Його необхідно закрити і підтвердити необхідність збереження структури. У результаті з'явиться вікно бази даних. Відкрийте таблицю запит **Сведения** і ви переконаєтесь, що на екрані буде та сама інформація.

Створіть тепер ще раз таблицю-запит Сведения в її початковому вигляді і викличте вікно Построитель выражений. Після цього введіть такий логічний вираз:

**[Оклад]<250 And [Стаж]>7**

Звертаємо вашу увагу на те, що після математичних знаків (у даному випадку після знаків більше чи менше) не повинно бути пропусків. Інакше числові константи будуть сприйматися як текстові.

Окрім розглянутих способів пошуку інформації в базі даних може використовуватися «**Мастер перекресных запросив»** для побудови таблиць з підрахованими підсумковими значеннями і «**Мастер повторяющихся** **записей»** для групування записів з однаковими значеннями полів.

Запити призначені для відбору даних, які задовольняють заданим критеріям (наприклад, дані про клієнтів, які проживають у визначеному регіоні, дані про торгівельні операції за останній квартал). Результатом виконання запиту являється набір записів, зібраних у таблиці, що називається Recordset (динамічний, тимчасовий набір даних). У об'єкті Recordset допускається додавання, зміна і видалення записів. У ньому відображаються також записи, які можна додавати, видаляти або змінювати у вихідних таблицях.

У Access запити поділяються на QBE-запити ( Query By Example-запит по зразку), параметри яких встановлюються у вікні конструктора запитів, і SQL-запити (Structured Query Language–структурована мова запитів), при створені яких використовуються оператори і функції мови SQL.

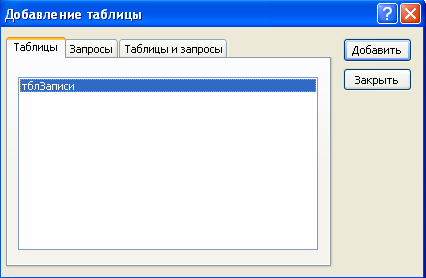
**QBE-запити**

Один із найбільш поширених типів QBE-запитів - запит на вибірку. Запити використовують таблиці, що містять ваші дані. Тому коли ви хочете створити запит, ви повинні вирішити, які таблиці потрібно включити в запит. Також, ви можете створювати запити на основі декількох таблиць (або запитів). Для початку давайте складемо запит, що використовує тільки одну таблицю, як-от таблицю з вашого приклада. Процес створення запиту можна розбити на ряд кроків, що визначають, зокрема:

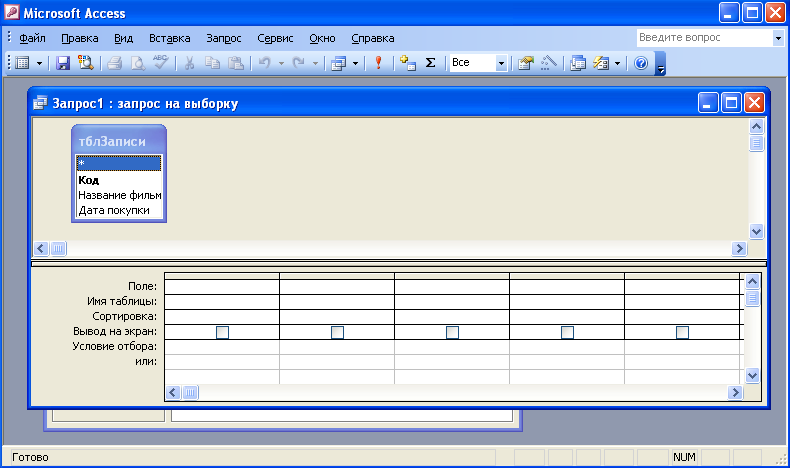
1. поля, які повинні бути включені в запит
2. яким чином повинні бути відсортовані дані
3. умови відбору, що повинні бути використані в запиті.

Для створення запиту найкраще почати роботу в режимі конструктора запиту.

1. Клацніть на вкладці **Запрос**. На даний момент список запитів порожній, оскільки ми ще не створювали жодного запиту.
2. Перед тим, як перейти в режим конструктора запиту, ви побачите діалогове вікно **Добавление таблицы** (мал. 13). Тут здійснюється вибір таблиць і запитів, які використовуються при розробці запиту.

****

Малюнок 13

1. Виберіть таблицю з вашої бази даних, а потім клацніть на кнопці **Добавить**. Назва вашої таблиці з'явиться угорі вікна на екрані конструктора запиту (мал. 14).
2. 

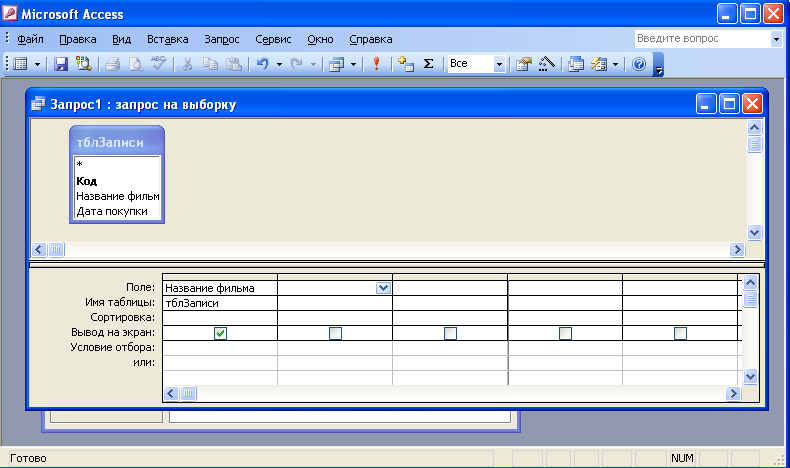
Малюнок 14

Ви будете використовувати поля, які видні в списку полів, представлених у бланку запитів, який знаходиться нижче області конструктора запитів. Більша частина роботи по створенню запитів виконується саме тут. Стовпці вміщують поля вибраної таблиці, якою в даному випадку є **тблЗаписи**. Ось список тих задач, які вирішують рядки цього бланка:

1. **Поле.** Цей рядок містить поле (з списку полів), які ви бажаєте включити в запит. Він може також уміщувати й вирази (тобто створене вами тимчасове поле).
2. **Имя таблицы.** Показує ім'я таблиці, з якої узяте поле. Це корисно в таких випадках, коли запит заснований на декількох таблицях, особливо якщо назви полів з декількох таблиць співпадають.
3. **Сортировка.** Дозволяє вказувати на підставі яких полів буде проводитися сортування записів.
4. **Вывод на экран.** Визначає, чи буде дане поле виводитися на екран. Використовуйте цю опцію тоді, коли ви використовуєте деяке поле для умови відбору, але не бажаєте, щоб воно з’явилося на екрані.
5. **Условие отбора.** Умови відбору дозволяють вам включати або не включати записи в разультативну таблицю, грунтуючись на їх значеннях або комбінаціях значень. Отут також, як і у фільтрах, ви можете комбінувати умови відбору за допомогою операторів AND і OR.

У залежності від типу створеного запиту, рядків може бути і більше.

Для того, щоб заповнити поля запиту, потрібно двічі натиснути на полі (у нашому випадку) **Название фильма** в списку полів таблиці **тблЗаписи**. Access перенесе це поле в перший стовпчик бланка запита (мал.15). У іншому рядку з’явиться назва таблиці, з якої це поле отримали. Включивши хоча б одне поле в запит, ви вже маєте той мінімум, для того, щоб запустити цей запит.



Малюнок 15

**Видалення полів і таблиць із запиту**

Для видалення поля з бланка запиту варто виділити колонку, у якій воно знаходиться, і натиснути клавішу [ Del ] або активізувати в меню **Правка** команду **Удалить**.

Щоб видалити з запиту таблицю, слід спочатку виділити її у верхній частині вікна конструктора запиту, натиснувши на імені будь-якого з її полів, а потім натиснути клавішу [ Del ] або викликати команду **Удалить** із меню **Правка**. Поля видаленої з запиту таблиці, що були поміщені в бланк запиту, видаляться автоматично.

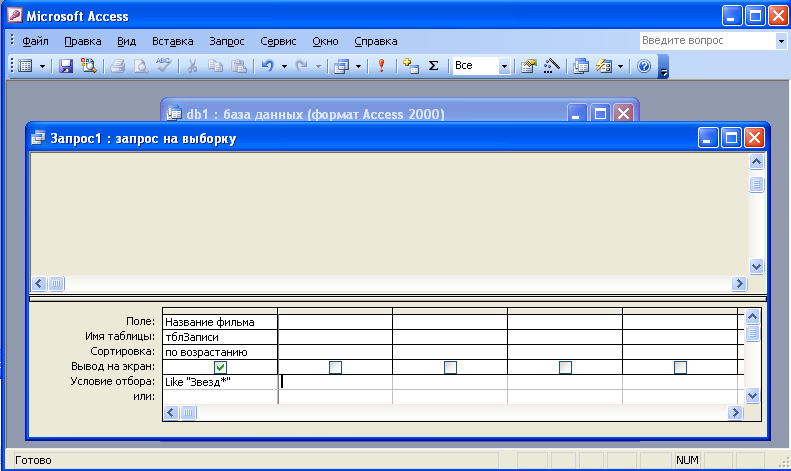
**Сортування даних у запиті**

Для того щоб встановити порядок сортування в полі, виконайте наступне:

1. У бланку запиту помістіть курсор у рядок **Сортировка** того поля, по якому потрібно провести сортування. Коли ви це зробите, то побачите, що поле сортування пропонує випадаючий список.
2. Клацніть на кнопці списку, що випадає, і ви побачите, що він пропонує вам три можливості: за зростанням, за спаданням і відсутнє.

**Встановлення критеріїв відбору записів**

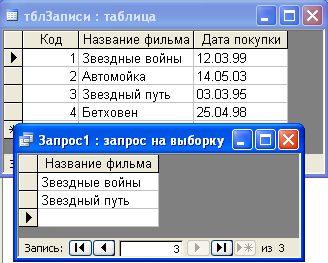
За допомогою критеріїв відбору користувач вказує Access, які записи потрібно вибирати в таблицях і відображати в таблиці результатів виконання запиту. Критерії відбору вказуються для одного або декількох полів. Наприклад, для того щоб побачити список фільмів, що починаються з букв «Звезд», зробіть наступне:



# Малюнок 16

1. В режимі конструктора запиту у першому рядку умов відбору для поля **НазваниеФильма** надрукуйте Like Звезд\* і натисніть <Enter>. Access перетворить це в Like Звезд\*, поставивши Звезд\* у лапки. Ця умова відбору показана на мал.16.

Службове слово Like і підстановочний знак зірочка (\*) говорять програмі Access, що ви хотіли б побачити всі назви фільмів, що починаються з Звезд. Клацніть на кнопці **Вид**, щоб побачити результат (мал. 17).

****

Малюнок 17

IS NULL – для виявлення незавершених чи пустих записів.

IS NOT NULL – для відображення лише тих записів, де є дані у необхідному полі.

Часто в предикатах потребується не тільки оцінювати рівність оператора як істинного чи хибного, але й здійснювати інші види зв’язків. Це реалізується за допомогою булевих операторів і знаків відношення, причому предикат може утримувати необмежену кількість умов.

В цілому, реляційний оператор – це математичний символ, котрий вказує на визначений тип порівняння між двома значеннями, при цьому SQL має в своєму розпорядженні наступний їх набір:

= рівний чому-небудь;

> більше чим;

< менше чим;

>= більше чим або дорівнює;

<= менше чим або дорівнює;

<> не дорівнює.

Ці оператори мають стандартні значення для числових даних, а для символьних їх визначення залежить від кодів ASCII символів – вони слідують в алфавітному порядку, причому заголовні літери мають менший код, ніж рядкові, тому, наприклад, “Z”<”a”.

Стандартними булевими операторами, які використовуються в SQL, є AND, OR і NOT.

Приклад: > = #01.01.97 # And < # 01.01.98 # або

Between # 01.01.97 # And #01.01.98 #

**Скасування відображення полів у запиті**

У разі потреби можна скасувати відображення поля в результуючому наборі записів (навіть того, по якому проводиться сортування). Поле відображається в таблиці результатів запиту, якщо у відповідній колонці бланка запиту, у рядку **Вывод на экран** включений контрольний індикатор. Для скасування відображення поля достатньо виконати натиск на індикаторі  .

**Обчислення в запиті**

Запит можна використовувати для виконання розрахунків і підведення результатів, узагальнивши дані з вихідних таблиць. Для цих цілей у Access передбачені статистичні функції SQL. Статистичну функцію задають у рядку **Груповая операция,** що з'являється після натискання кнопки з грецькою літерою сигма, розміщеної на панелі інструментів. За допомогою статистичної функції можна опрацювати зміст кожного поля запиту. Результат опрацювання з'являється в результуючому наборі записів запиту.

**Функція Дія**

Sum Підсумовування значень визначеного поля, що знаходяться в записах, що відібрані запитом.

Avg Обчислення середнього значення в тих записах визначеного поля, що відібрані запитом.

Min Обчислення мінімального значення в записах визначеного поля, відібраних запитом.

Max Обчислення максимального значення в записах визначеного поля, відібраних запитом.

Count Обчислення кількості записів, відібраних запитом, у визначеному полі.

First Визначається перше значення в зазначеному полі записів, відібраних запитом.

Last Визначається останнє значення в зазначеному полі записів, відібраних запитом.

StDev Обчислюється стандартне відхилення значень даного поля для усіх записів, відібраних запитом.

Var Обчислюється варіація значень даного поля для всіх записів, відібраних запитом.

**Редагування запиту**

Редагування запиту здійснюється в режимі конструктора, що встановлюється при натисканні кнопки **Вставка** у вікні бази даних.

Щоб додати в запит поле, потрібно перетягнути його із списку поля у верхній частині вікна конструктора в ту колонку області специфікації, у котру його необхідно вставити. Поле , що знаходиться в даній колонці, а також усі поля, розташовані справа від нього, будуть зрушені праворуч на одну колонку.

Для видалення поля варто маркірувати його і натиснути клавішу [Del] або вибрати в меню **Правка** команду **Удалить столбцы**.

Поле, позицію котрого потрібно змінити, також варто попередньо маркірувати. При переміщенні поля зміщується праворуч поле, спочатку розташоване в обраній позиції, а також усі поля справа від нього.

**Контрольні питання до лабораторної роботи 3:**

1. **Що являють собою запити і для чого їх використовують?**
2. **Якими способами можна створити запит?**
3. **Які два типи звітів можна створити за допомогою майстра простих запитів?**
4. **Чим відрізняється «стоговий» запит від «подобного» запиту?**
5. **В якому режимі можна вносити корективи у готовий запит?**
6. **За допомогою якого засобу можна легко створювати вирази?**
7. **Як працює Построитель выражений?**
8. **Що утворюється в результаті виконання запиту на вибірку?**
9. **Які запити називають QBE-запитами?**
10. **Які можливості QBE-запитів?**
11. **Як видаляти та додавати поля і таблиці у бланку запиту?**
12. **Як відсортувати дані за допомогою запитів?**
13. **Яким чином можна встановити критерії відбору записів в своїй БД?**
14. **Перелічіть статистичні функції SQL, які ми можемо застосовувати у QBE-запитах.**
15. **Для чого служить рядок Условие отбора у вікні конструктора запиту?**
16. **Які булеві оператори застосовуються в умовах вибору запиту?**
17. **В яких випадках використовують оператор BETWEEN?**
18. **Який оператор застосовується для відбору записів, які починаються з певного символу?**

**Лабораторна робота №4**

**Тема: Обчислювальні поля у запитах. Додавання, поновлення, видалення інформації з таблиць БД за допомогою QBE-запитів. Параметричні запити, перехресні запити.**

**Мета: Навчитися створювати запити з обчислювальними полями. Вивчити правила побудови виразів. Навчитися будувати запити на створення таблиці, на додавання інформації з іншої таблиці, на вилучення записів, на поновлення інформації в БД, перехресні запити та запити з параметрами за допомогою QBE-запитів.**

Зміст роботи за варіантом індивідуального завдання:

1. Створіть запит до таблиці вашої БД, який в останньому стовбці динамічної таблиці містив би інформацію з попередніх двох полів, розділену розділовими знаками та допоміжним текстом. Наприклад: Альфа-Львів, тел.: 77-89-54.
2. Створіть запит, за допомогою якого можна було б вираховувати дані таким чином, перший рядок додавався до другого і множився на третій. 25% від отриманого розмістити в одну з колонок динамічної таблиці.
3. Створіть запит з обчислювальним полем Повідомлення. Це поле буде вміщувати повідомлення: «Закінчується термін дії» якщо значення у полі Термін дії від 10 до 0 і повідомлення «Термін дії закінчено», якщо термін дії – 0.
4. Створіть вираз у запиті, який замінює порожні записи у таблиці на 0 або на «Пусто».
5. За допомогою QBE-запиту додайте записи з одної таблиці БД до записів іншої таблиці БД.
6. Видаліть з таблиці БД записи, які відповідають деяким критеріям за допомогою QBE-запитів.
7. Замінити в таблиці БД значення, відібрані на основі визначених критеріїв на нові, які будуть результатом виконання виразу.
8. За допомогою QBE-запиту створіть «архівну» копію записів таблиці БД.
9. Створіть запит, який запитує значення критерію, за яким буде здійснюватися вибір інформації.
10. Створіть перехресний запит.

##### Теоретичні відомості:

Обчислювальні поля дозволяють створювати і показувати вирази на основі наявних полів. Вирази є головним засобом виконання багатьох операцій Microsoft Access. Вираз представляє комбінацію символів-ідентифікаторів, операторів та значень, яка дає певний результат.

Ось декілька прикладів, в яких зручно використати вирази.

* Комбінування полів Прізвище і Ім'я, поділених комами, в вираз з метою показу їх на екрані, наприклад, в такому вигляді: *Баркер, Диана.*
* Додавання днів, тижнів або місяців до поточної дати з метою виводу на екран майбутньої дати.
* Перемноження значень полів, таких як Кількість і Ціна, щоб вивести на екран загальну вартість.

Access може утворювати обчислювальні поля з інших полів, буквальних значень і функцій. Перед тим як зайнятися побудовою виразів, давайте ознайомимося з їхнім синтаксисом, прийнятим в Access.

**Синтаксис обчислювальних полів**

Для обчислювальних полів вимагається використовувати певний синтаксис. Наприклад, для комбінації полів Прізвище, Ім'я і кома, синтаксис буде таким: Выражение1: [Прізвище] &"," & [Ім'я].

Синтаксис визначає структуру команди, яку ви використовуєте. В обчислювальних полях визначаються стандартні оператори і їх коректне застосування до відповідних полів.

Перша частина наведеного обчислювального поля **Выражение1**, є ім'ям нового поля. Access запропонує вам його, якщо цього не зробите ви самі. Саме на це ім'я вам потрібно буде посилатися в подальшому при використанні цього поля в формі, звіті або іншому виразі.

**Заключення полів в дужки**

Використовуючи в запитах поля з таблиць, ви повинні укладати ці поля в квадратні дужки. Якщо в назві поля не буде проблем, Access поставить дужки за вас. В цьому відношенні [Прізвище] і [Ім'я] будуть гарними прикладами.

# Використання операторів в обчислювальних полях

Є цілий ряд операторів, що можна використати в обчислювальних полях. Деякі з них призначенні для математичних обчислень, як, наприклад, наведені в таблиці.

**Таблиця: Математичні оператори для обчислювальних полів**

|  |  |
| --- | --- |
| **Операторы** | Описание |
| **+** | Сложение |
| **−** | Вычитание |
| **\*** | Умножение |
| **/** | Деление |
| **^** | n-я степень |

Прикладом застосування математичного оператора може служити створення нового поля для загальної вартості шляхом множення полів Кількість і Ціна. Це може виглядати так: ЗагальнаВартість:

[Кількість]\* [Ціна].

Ви вже бачили ще один оператор — амперсанд (&). Він був використаний для того, щоб з'єднати два рядки з буквальною константою в один новий рядок:

Выражение1: [Прізвище] &"," & [Ім'я].

Інше популярне обчислювальне поле з'єднує поля Місто, Область і zip - код з комою в одне поле. Синтаксис для нього може бути таким:

Адреса: [Місто] &","& [Область] & " " & [ZlP-код]

а результат, наприклад, таким: Олександрія, КР 0246207.

В цьому прикладі з адресою ми додали пробіли і коми між назвами міста, областю і zip - кодом, помістивши їх у лапки і, використовуючи їх як буквальні константи.

# Приклади обчислювальних полів

Наступний список містить ще декілька обчислювальних полів, що часто можуть бути корисними:

###### Вартість: [Ціна]\*[Кількість]

ЗагальнаВартість: [Вартість]+([Вартість]\*[ПодатковийКоефіцієнт])

З0днів: [ДатаЗамовлення]+30

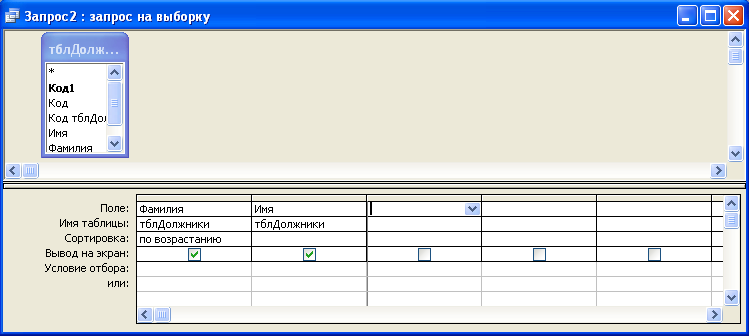
КодМіста:Left([Телефон];3)

Не всі поля, що використовуються в обчислювальному полі, повинні бути використані в бланку запиту. Якщо поле уявлене в списку полів області конструктора запиту, ви вже можете використовувати його в бланку запиту. Не має значення і те, в якому порядку знаходяться поля або обчислювальні поля.

**Створення простого обчислювального поля**

Для того щоб показати, як створювати обчислювальні поля, нам потрібно створити запит, із яким ми будемо працювати.

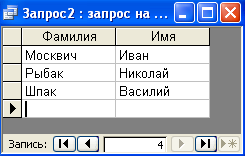
Створіть новий запит,вибравши поля **Фамилия** та **Имя** із списку полів таблиці **тблДолжники** (мал.1).



#### Малюнок 1

Визначте порядок сортування **по возростанию** для поля **Фамилия**.

Тепер клацніть на кнопці **Вид**, щоб побачити результат у вигляді таблиці (мал. 2).

****

Малюнок 2

Тепер ми можемо продовжити роботу з обчислювальними полями.

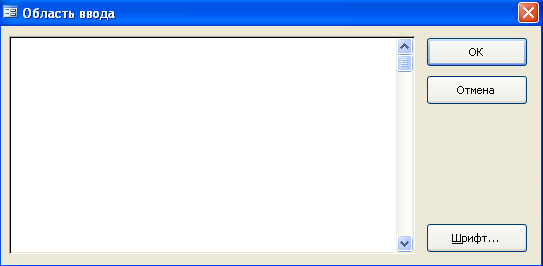
Одним із найпростіших і найбільш корисних прикладів обчислювального поля є злиття двох текстових полів. Корисних тому, що часто, наприклад, вам потрібно взяти прізвища й імена і зв'язати їх якимось чином на поштових ярликах, у звітах або формах.

Перейдіть у режим конструктора запиту. Потім у бланку запиту розмістіть курсор у порожній стовпчик поруч із полем **Фамилия**, справа від нього.

Натисніть <Shift+F2>. З'явиться діалогове вікно **Область ввода**, як на мал.3.

Також для створення виразів у запитах використовують **Построитель выражений**. Для цього, перебуваючи в комірці, де треба створити вираз, викличемо **Построитель** за допомогою кнопки **Построить** панелі інструментів.

Засіб **Область ввода** є відмінною можливістю для запровадження інформації, коли вам хотілося б бачити усе поле.

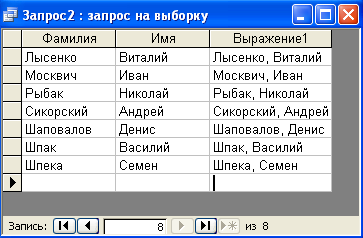
****

Малюнок 3

Наберіть **Фамилия & ", " & Имя** , а потім клацніть на ОК. Відразу ж після цього Access перетворить введене вами у **Выражение 1: [Фамилия] & ", " & [Имя]**.

Access створить ім'я **Выражение1** (мал. 4).

В результаті отримаємо динамічну таблицю, де з’явилось нове поле **Выражение1**.

****

Малюнок 4

Ім’я обчислювальних полів можна змінювати. Наприклад:

ПолноеИмя: **[**Фамилия**]** & ", " & [Имя].

### Визначення властивості Подпись обчислювального поля

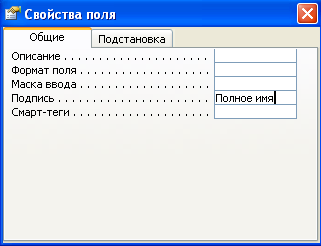
Як і у випадку з полями таблиці, ви можете встановити властивість **Подпись** для будь-якого поля в запиті, щоб на екран виводився текст, відмінний від назви поля. Це дуже зручно для обчислювальних полів, коли ви можете виводити на екран слова з прогалинами між ними і не мати прогалин у назві поля.

Щоб встановити властивість **Подпись** обчислювального поля **ПолноеИмя**, що було вами створене, виконайте наступне:

Помістіть курсор у стовпчик **ПолноеИмя** запиту.

Виберіть **Свойства** з меню **Вид**. Ви побачите вікно властивостей, подібне вікну властивостей текстового поля.

У рядку властивості **Подпись** надрукуйте **Полное имя** (мал.5).

****

Малюнок 5

Використання вбудованих функцій Access в обчислювальних полях

Функції Access значно розширюють ваші можливості по відображенню даних. За допомогою цих вбудованих функцій ви можете наступне:

* Виводити на екран не все, а тільки частину даних.
* Комбінувати дані з новою інформацією, отриманої за допомогою функції Access.
* Комбінувати функції Access і виводити цілком нову інформацію.

Серед прикладів обчислювальних полів, що приводилися вище, було поле, яке називається **КодГорода**. Для нього було використано вираз Left ([Телефон];3). Воно потрапляє в першу категорію типів інформації, наданих функціями Access.

Функція Left() є вбудованою функцією Access, подібно функції Date(). Функція Left() разом із двома своїми аргументами, [Телефон] і 3, говорить Access, що потрібно взяти три крайніх зліва символи поля Телефон і повернути їх у якості результату.

Іноді функції потребують аргументів. У даному прикладі поточними аргументами є поле [Телефон] і 3.

Ще одним прикладом використання функції Access є одержання ініціалів з імені і прізвища. Це теж може бути виконане за допомогою функції Left(). От відповідне обчислювальне поле:

Инициалы: Left([Имя];1) & Left ([Фамилия];1)

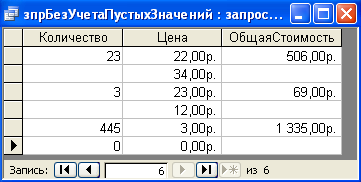
**Обробка порожніх значень у запитах**

Одна з проблем, що виникає при використанні обчислювального поля, породжується тим фактом, що поля можуть містити порожні значення. Порожнє значення буде поширюватися на всі наступні результати запиту.

Порожнє значення (значення Null**)** є присутнім у полі тоді, коли в поле не вводилося ніяких значень і не було ніякого значення за замовчуванням. Коли поле містить порожнє значення, це означає, що його значення невідоме, і Access не знає, що в цьому випадку робити.

Щоб розібратися з порожніми значеннями, створимо тимчасову таблицю, що містить необхідні поля. У даному випадку це поля **Количество** і **Цена**. Подивіться на мал. 7, де представлений робочий бланк запиту з полями Количество і Цена, а також з обчислювальним полем ОбшаяСтоимость. Цей запит ніяк не опрацьовує порожні значення. На мал. 6 поданий результат у випадку, коли порожні значення не обробляються.

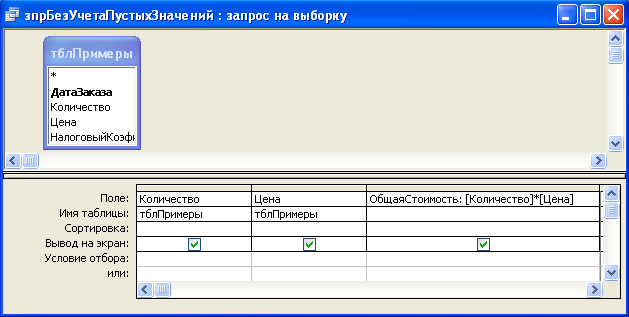
Щоб бути готовим до порожніх значень, у Access є функції, такі як IsNull() або Nz(). Синтаксис функції IsNull() такий: IsNull (вираз)



Малюнок 6

Вираз може бути полем або обчислювальним значенням. Функція IsNull() поверне або Істина, або Брехня, в залежності від того, є значення виразу порожнім або ні. Передбачається, що це повинно використовуватися разом із функцією IIF(). Синтаксис функції IIF() такий: **IIF (критерий;значение1;значение2)**

При виклику функції ІIF(), якщо критерій дає Істина, то функція IIF() повертає значення1, якщо критерій дає Брехня, то функція IIF() повертає значення2.

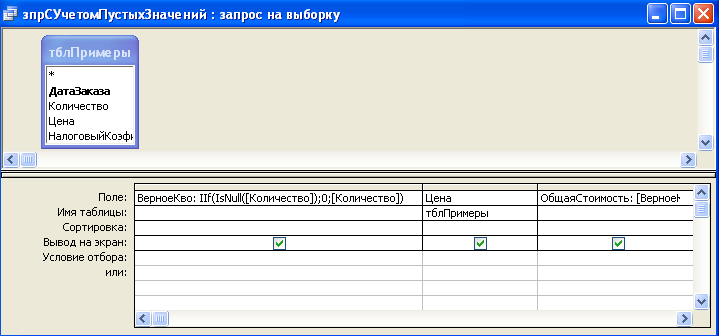
****

Малюнок 7

Щоб виправити запит, показаний на мал.6, ми створимо обчислювальне поле за назвою **ВерноеКво**. Буде використана функція IIF() із значенням IsNull(Количество) у якості критерію ( значення, що приймає , або Істина, або Брехня) Ось потрібний вираз:

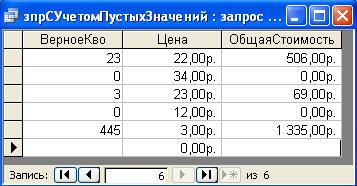
ВерноеКво: IIF (IsNull ([Количество]);0;[Количество])

Якщо функція IsNull() *(критерій)* повертає Істина, то замість Количество використовується 0 *(значення1).* Якщо IsNull() повертає Брехня, то використовується поле Количество *(значення2).* Ви можете побачити це обчислювальне поле на мал.8, де запит поданий у режимі конструктора запиту; результуючий лист з даними показаний на мал.9.

****

Малюнок 8

###### Порожні значення тепер замінені нулями.

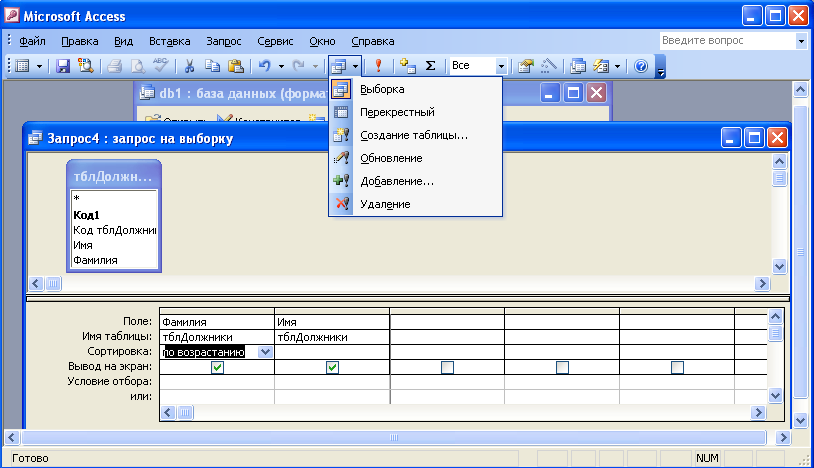
****

Малюнок 9

Зверніть увагу, що обчислювальне поле **ОбщаяСтоимость** тепер використовує обчислювальне поле ВерноеКво замість поля Количество.

**Запити іншит типів**

Ми вже розглянули запит на вибірку. У Access існують і запити інших типів: запити дії (на зміну), параметричні запити і перехресні запити.



Малюнок 10

**Запити на зміну**

Запити на зміну використовують для зміни і переносу даних у таблицях, для поновлення, додавання й видалення груп записів, а також для створення нової таблиці на основі однієї або декількох таблиць.

Розрізняють чотири типи запитів на зміну: запит на додавання, запит на видалення, запит на поновлення і запит на створення таблиці.

**Запит на додавання**

За допомогою запиту на додавання записи з однієї таблиці (усі або відібрані запитом) можна помістити в кінець іншої таблиці. Обидві таблиці можуть знаходитися як в одній, так і в різних базах даних.

Якщо необхідно додати записи до таблиці з іншої бази даних, спочатку варто приєднати таблицю-джерело до бази, що містить цільову таблицю, за допомогою команди **Связать таблицы** з підміню **Получить внешние данные** меню **Файл**. Для відбору записів, що додають потрібно скласти запит на вибірку. Потім слід виконати складений запит і оцінити результат, переключившись у режим таблиці за допомогою команди **Datasheet View** із меню **Вид**. Після цього необхідно повернутися в режим конструктора й активізувати команду **Запрос добавления** з меню **Запрос**.

У діалоговому вікні, яке при цьому відкриється **Append** користувач повинен задати в полі **Имя таблицы** ім'я таблиці, до якої він хоче приєднати дані з результуючого набору записів запита. Закриття діалогового вікна за допомогою кнопки **ОК** призводить до того, що Access добавляє в бланк запиту рядок **Append To.** У цей рядок автоматично (або вручну) вставляються імена тих полів цільової таблиці, що збігаються з іменами полів запиту. Далі варто виконати запит, натиснувши кнопку **Run** на панелі інструментів. У спеціальному діалоговому вікні Access зазначить, скільки записів буде додано до цільової таблиці, і запропонує підтвердити виконання цієї операції.

**Запит на видалення**

Запити цього типу служать для видалення з таблиці груп записів, що відповідають деякому критерію відбору. Оскільки записи видалені за допомогою запиту, не можна відновити, варто старанно аналізувати критерії відбору.

Отже, насамперед необхідно розробити запит на вибірку видаляємих записів, указавши критерії в рядку **Критерии**, виконати цей запит, щоб перевірити коректність заданих критеріїв, і повернутися в режим конструктора.

Після цього слід активізувати в меню **Запрос** команду **Удаление** **запроса**. Access додасть у бланк запиту рядок **Удалить** і введе в її комірки значення **Условие**. Це означає, що користувач може встановити додаткові критерії відбору. Далі варто виконати запит, натиснувши кнопку з зображенням знака оклику на панелі інструментів. У спеціальному діалоговому вікні Access зазначить, скільки записів видалиться з таблиці, і запропонує підтвердити видалення.

**Запит на поновлення**

Використовуючи запит на поновлення, користувач може змінити групу записів, відібраних на основі визначених критеріїв. При розробці запиту на поновлення насамперед створюється і перевіряється запит на вибірку. Після цього в режимі конструктора активізується команда **Запрос на обновление** з меню **Запрос**, унаслідок чого Access додає в бланк запиту рядок **Update To**, що призначений для вказівки нових значень полів таблиці. У якості типових можуть виступати і обчислювальні вирази. Після натискання кнопки **Run** Access зазначить у спеціальному діалоговому вікні, скільки записів зміниться в таблиці, і спробує підтвердити зміни.

**Запит на створення таблиці**

На основі результуючого набору записів запиту можна побудувати нову таблицю за допомогою запиту на створення таблиці. Такі запити звичайно застосовують для архівації старих записів або для зберігання резервних копій таблиць.

Спочатку необхідно підготувати запит на вибірку й одержати динамічний набір даних Recordset, заодно перевіривши коректність запиту. Якщо результуючий набір записів відповідає вашим вимогам, поверніться у режим конструктора й у меню **Запрос** виберіть команду **Запрос создания таблицы.** Access відчинить діалогове вікно **Создание таблицы**, в яке варто ввести ім'я нової таблиці. Далі потрібно виконати запит, натиснувши кнопку зі знаком окликуна панелі інструментів. У спеціальному діалоговому вікні Access зазначить, скільки записів добавляється в нову таблицю, і запропонує підтвердити виконання цієї операції.

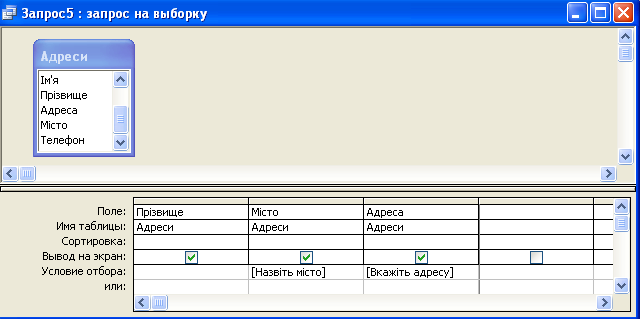
**Параметричні запити**

Запити, що являють собою варіанти базового запиту і незначно відрізняються один від одного, називаються параметричними.

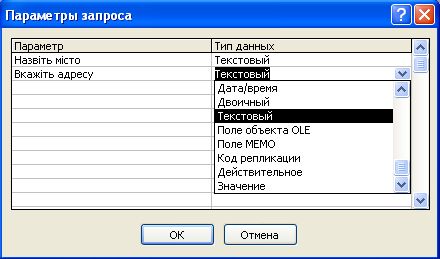
Розглянемо створення параметричного запиту для відбору з таблиці Адресипрізвища осіб, які проживають в зазначеному місті (поле Населений пункт ) і по визначеній адресі (поле Адреса ).

Насамперед розробимо запит на вибірку даних. Включимо в нього поля Прізвище, Населений пункт і Адреса. Поля рядка **Критерии** заповніть не конкретними критеріями відбору, а звертанням до користувача для запровадження критерію. На основі цього критерію (з урахуванням інших критеріїв, заданих у специфікації запиту) буде здійснюватися добір записів. Звертання необхідно узяти в квадратні дужки, наприклад: [Назвіть місто] , [Зазначте адресу].

Активізуйте команду **Параметры** меню **Запрос.** У поле **Параметры** діалогового вікна **Параметрические запросы** введіть текст, що знаходиться в полі **Критерии** специфікації запиту. У даному випадку дужки не потрібні. Після визначення типу даних у полі **Тип данных** і натискання кнопки **ОК** запит буде готовий. Тепер можете приступити до вводу параметрів.

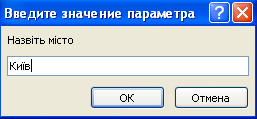
****

Малюнок 11



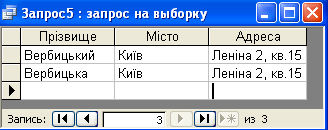
Малюнок 12

Для перевірки працездатності запиту перейдіть у режим таблиці, установивши опцію **Datasheet View** у меню **Вид**. При цьому Access звернеться до вас із проханням задати значення критерію у вікні **Введите значение параметра**.



Малюнок 13

Привласніть першому параметру значення *Київ*, а другому- *Леніна 2, кв.15*. Після запровадження параметрів буде отриманий результат виконання параметричного запиту.

****

Малюнок 14

**Перехресні запити**

Перехресні запити дозволяють компактно відображати відібрані дані й об'єднувати одночасну інформацію. Таблиці, у яких відображаються результати виконання перехресного запиту, зручні для створення діаграм й графіків за допомогою майстра.

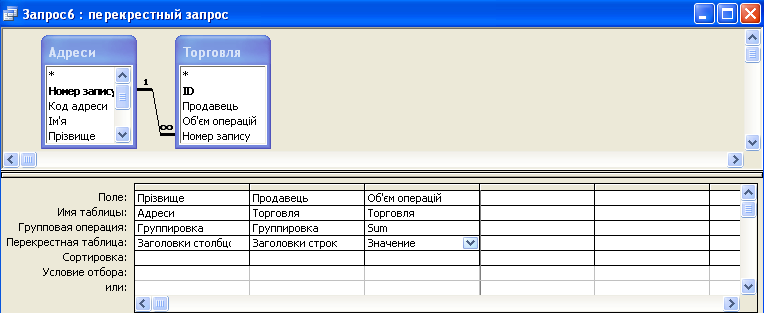
Визначимо об’єм виторгу від продажу різноманітних товарів за умови, що відповідна інформація зберігається в базі даних окремо для кожного місяця. При використанні перехресного запиту найменування товару з'являється в результуючому наборі даних тільки один раз, Access об’єднує дванадцять записів по місяцях. Цим перехресний запит відрізняється від запиту на вибірку, при виконанні якого в результуючий набір даних включається окремо кожен з дванадцяти місячних записів по даному товару.

Створимо перехресний запит *Продавець - Покупець* на основі таблиці *Торгівля*, використовуючи для вибору прізвищ покупців таблицю *Адреси*.

У меню **Запрос** активізуйте команду **Перекрестный Запрос.** Access відобразить у бланку запиту рядки **Total** і **Перекрестная таблица**. Включіть в бланк запиту поля *Продавець* і *Обсяг операцій* із таблиці *Торгівля* і поле *Прізвище* з таблиці *Адреси*. Щоб значення поля (наприклад, *Продавець* ) з'явилося в заголовках рядків, натисніть на рядку **Перекрестная таблица** і виберіть у списку рядок **Заголовок строк**. У рядку **Total** для цього поля повинна зберегтися стандартна установка **Группировать по.**

Щоб значення поля *Прізвище* з'явилися в заголовку колонки, натисніть на рядку **Перекрестная таблица** і виберіть рядок **Заголовок столбцов**. У цьому випадку в рядку **Total** повинна залишитися стандартна установка **Группировать по.**

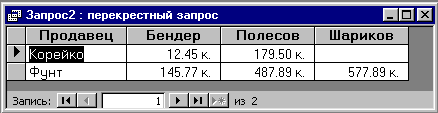
Для відображення змісту поля в перехресній таблиці в якості значення (Обсяг операцій) клацніть на рядку **Перекрестная таблица**  і виберіть рядок **Значение**. У рядку **Total** для цього поля введіть необхідну функцію- Sum.

****

Малюнок 15

При визначенні критеріїв відбору в рядку **Total** для бажаного поля слід вибрати значення **Условие** і ввести вираз в рядок **Критерии.** Для цього поля рядок **Перекрестная таблица** залиште порожнім. Критерії відбору ми вводити не будемо.

Щоб відобразити на екрані результуючий набір записів створеного запиту, варто перейти в режим таблиці.



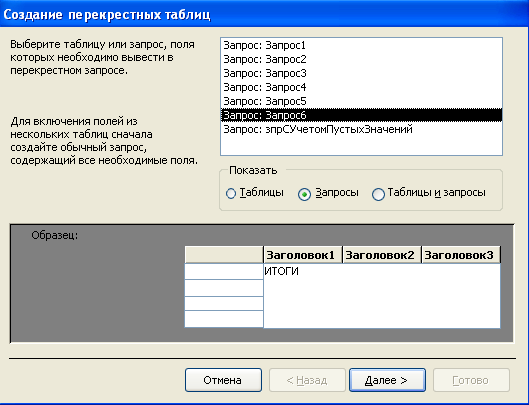
Малюнок 16

**Створення перехресного запиту за допомогою майстра**

Розглянемо процес розробкиперехресного запиту з попереднього приклада за допомогою спеціального майстра -Crosstab Query Wizard. Цей майстер дозволяє створити перехресний запит на основі тільки однієї таблиці чи одного запиту.

У вікні бази даних перейдіть на вкладку **Запрос** і клацніть на кнопці **Новый**. У діалоговому вікні **Новый** **запрос**, яке при цьому з’явиться виберіть значення  *Crosstab Query* Wizard і натисніть кнопку **ОК .**

У першому діалоговому вікні виберіть таблицю або простий запит на основі якого майстер створить перехресний запит. Оскільки у нашому прикладі використовується дві таблиці, *Торгівля* й *Адреси,* спочатку варто відобразити з обох таблиць дані, необхідні для перехресного запиту. Цю роботу ми вже проробили створивши запит *Операції з клієнтами* за допомогою майстра простого запиту. Виберіть його в якості базового і натисніть кнопку **Далі**.

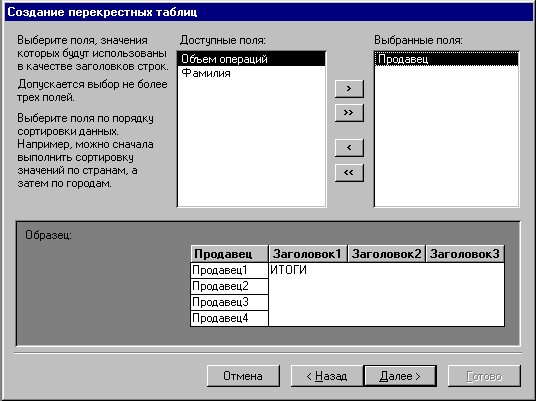
****

Малюнок 17

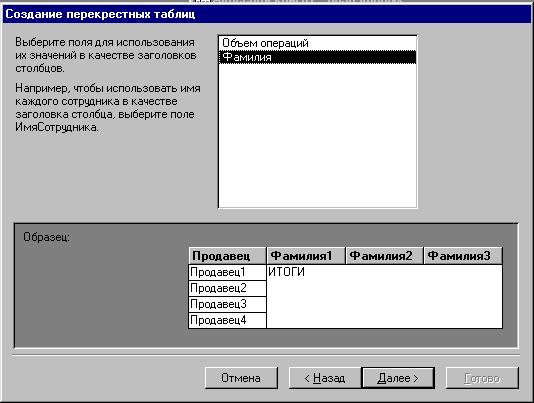
Щоб зазначити, імена полів, які повинні використовуватися в якості заголовків рядків перехресної таблиці, із запропонованих полів **Доступные поля** такого вікна майстра виберіть поле *Продавець* і помістіть його в список **Выбранные поля** за допомогою кнопки > .

Перейдіть в наступне вікно майстра. У цьому вікні потрібно вибрати поле, значення якого будуть фігурувати в заголовках колонок таблиці з результатами виконання перехресного запиту. Виберіть поле *Прізвище* і натисніть кнопку **Далі.**

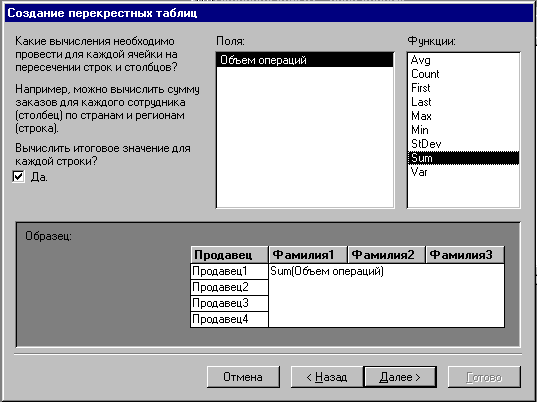
Малюнок 18



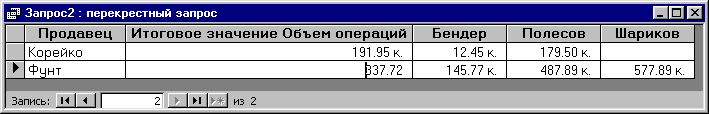
В наступному вікні майстра зазначте, значення якого поля повинні знаходитися в комірках результуючої таблиці, і виберіть операцію для обробки цих значень- **Sum**. Крім цього в таблиці з результатами виконання перехресного запиту потрібно визначити суму значень у кожному рядку. Для цього установіть опцію **Да, включите суммирование по строкам**. У результаті перехресного запиту буде добавлятися колонка зі значеннями обсягу продажів у кожного продавця



Малюнок 19



Малюнок 20



Малюнок 21

**Контрольні питання до лабораторної роботи 4:**

1. **Яке поле називають обчислювальним полем?**
2. **Що являє собою вираз?**
3. **Перелічіть правила побудови виразів.**
4. **Перелічіть оператори, які використовують в виразах.**
5. **Які функції називають вбудованими функціями?**
6. **Яким чином можна обробляти порожні записи у запитах?**
7. **Для чого призначений запит на вибірку?**
8. **Що змінюється при визначенні властивості «Подпись» обчислювального поля?**
9. **Які засоби існують в Access для побудови виразів?**
10. **В яких випадках застосовують функцію IsNull()?**
11. **Як застосувати функцію IIF в обчислювальних полях?**
12. **Запити яких типів ви знаєте?**
13. **Які дії можна робити за допомогою запитів на зміну?**
14. **Чим відрізняються запити на зміну від запитів на вибірку?**
15. **Що представляють собою параметричні запити в чому їх зручність?**
16. **Яким чином будують перехресні запити?**
17. **Чи можна додавати інформацію в таблиці БД з інших БД?**
18. **Чи створює запит на зміну динамічний набір записів Recordset в результаті?**
19. **Як перевірити дію запитів на зміну?**
20. **Які запити використовують для зберігання резервних копій таблиць БД?**
21. **Які можливості розкриває запит на поновлення?**
22. **Запити якого типу служать для видалення з таблиць груп записів, що відповідають деякому критерію відбору?**

**Лабораторна робота № 5.**

**Тема: Побудова інтерфейсу користувача за допомогою форм. Майстер форм. Режим конструктора форм.**

**Мета: Навчитися створювати та редагувати форми для бази даних. Вивчити всі засоби роботи з формою у режимі конструктора.**

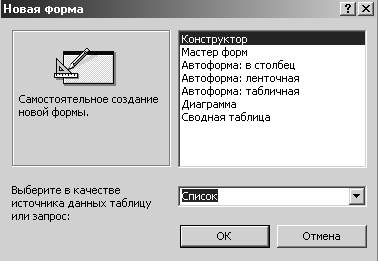
Зміст роботи за варіантом індивідуального завдання:

1. Створити автоформи для своєї бази даних трьох типів.
2. Створити форму за допомогою майстера форм.
3. Зробити різними зовнішні вигляди та стилі ваших форм.
4. Побудувати форму на основі полів з декількох таблиць.
5. Побудувати форму на основі полів з декількох запитів.
6. Створити «Сводную таблицу».
7. Побудувати діаграму.
8. Відкрийте створену за допомогою майстра форму у режимі конструктора, застосовуючи основні засоби для роботи з формою. Додайте елементи за допомогою панелі елементів, списку полів.
9. Додавши до своєї форми області Заголовка, Приміток, верхнього та нижнього колонтитулів, заповніть всі області необхідними даними.
10. Змініть вигляд форми за допомогою панелі властивостей форми (застосувати властивість Подпись, Рисунок, Ширина та ін.).
11. Змініть джерело даних для своєї форми за допомогою панелі властивостей форми.
12. Створіть форму, в якій забороніть користувачу змінювати інформацію.

**Теоретичні відомості:**

**Створення та використання форм.**

Розглянуті раніше способи подачі даних - табличні. Для деяких користувачів табличне подання даних може бути незручним. Часто дані необхідно розташовувати у такому вигляді, в якому зберігаються відповідні документи. Система Access дозволяє розташовувати на екрані дані так, як того бажає користувач, тобто у вигляді різноманітних форм. Форма може використовуватись також для введення, редагування, пошуку даних, управління ходом виконання програми, виведення різних повідомлень, друку інформації.



Малюнок 1

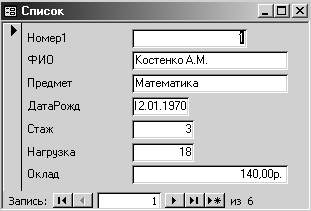
Система Access може створювати три види автоформ: стовпцеву, стрічкову та табличну. Методику створення цих форм розглянемо на прикладі бази даних **Basa2.**

У відкритому вікні бази даних виберемо об’єкт **Формы** і натискаємо кнопку **Создать.** У результаті відкривається вікно **Новая форма.** В цьому вікні відкриваємо список таблиць і вибираємо одну з них, а саме ту, з якої будемо створювати форму. Вибираємо, наприклад, **Список**

Створюємо першу стандартну форму - у стовпчик. Для цього виконуємо команду **Автоформа: в столбец.** У результаті буде створена форма, зображена на мал. 2.

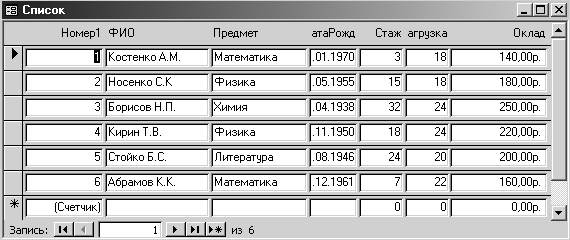
У цьому вікні всі поля розташовані у стовпець і виведено тільки один запис. Перегляд послідовних записів таблиці **Список** здійснюється за допомогою кнопок прокрутки.

При спробі закрити створену форму система робить запит про необхідність її збереження. Якщо форма зберігається, то видається ще один запит про збереження форми з означеним іменем. Можна ввести нове ім'я (введемо **Спис1)** чи згодитися з тим, яке пропонує система, і для цього натиснути кнопку **ОК.**



Малюнок 2

Створюємо тепер другу стандартну форму (стрічкову з тієї самої таблиці **Список).** Порядок її створення аналогічний попередньому, з тією різницею, що замість команди **Автоформа: в столбец** виконується команда **Автоформа: ленточная.** Стрічкова форма даних зображена на мал. 3.



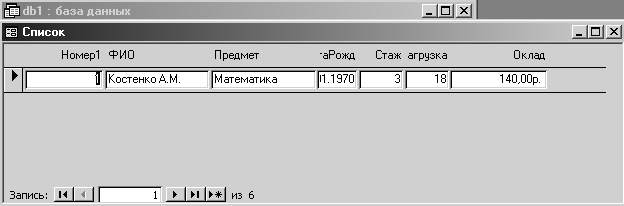
Малюнок 3

У цій формі кожен запис займає окремий рядок-стрічку. Збережемо цю форму з іменем **Спис2.**

Розміри форм стрічкової і в стовпчик можна змінювати. Для цього спочатку потрібно встановити приблизний розмір вікна (так як це звичайно робиться у Windows), а потім розкрити меню **Окно** і виконати команду **По размеру формы.**

Третя стандартна форма звіту - таблична. Вона створюється аналогічно тому, як і дві попередні. У цьому випадку у вікні **Новая форма** виконується команда **Автоформа: табличная.**

Створіть табличну форму для таблиці **Список.** Проаналізуйте її зовнішній вигляд і переконайтесь, що вона мало чим відрізняється від самої таблиці. Але її можна привести до звичайного вигляду табличної форми.



Малюнок 4

Для цього потрібно відкрити список **Представление формы, значок** якого знаходиться на панелі інструментів поряд з однойменною кнопкою (кнопка має позначення трикутника з олівцем). У списку, що відкривається, необхідно вибрати **Режим форми.** У результаті таблиця набере вигляду, показаного на мал. 4. До форми, яка зображена на мал. 4, може бути застосована описана раніше команда **По розмеру формы.**

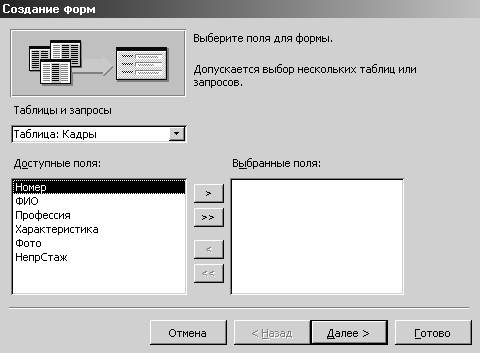
Крім розглянутих трьох видів стандартних форм, користувач може сам створювати довільну форму. Для цього використовують програму **Мастер форм.**

Для аналізу роботи майстра форм створимо базу даних **Basa4,** яка буде утримувати таблицю з іменем **Кадры.**

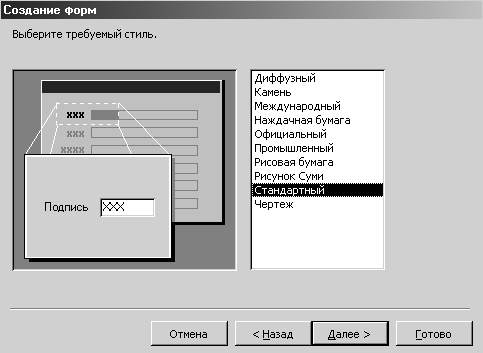
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Фио | Профессия | Характеристика | Фото | Непр-Стаж |
| 1 | Ушаков В. А. | Электрик | Работает на заводе 10 лет. Исполнителен, специалист высшего класса. |  | 12 |
| 2 | Бойко С.М. | Плотник | Специалист средней ква-лификаиии. |  | 3 |
| 3 | Сидоров А. А. | Токар | Работает старательно, имеет изобретательные способности |  | 8 |

Особливість цієї таблиці полягає у тому, що поле **Характеристика** має тип **Мемо,** а поле **Фото** - тип **OLE.** Для розуміння суті роботи майстра форм не будемо заповнювати поле **Фото** інформацією, залишимо його порожнім.

У вікні **Новая форма** (див. мал. 1) виберемо **Мастер форм,** відкриваємо список таблиць і запитів, з якого вибираємо таблицю **Кадры** і натискаємо кнопку **ОК.** З'явиться вікно **Создание форм,** яке зображено на мал. 5.



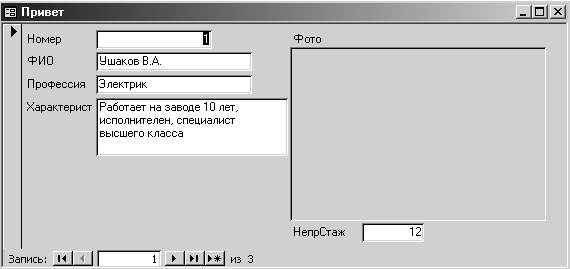
Малюнок 5



Малюнок 6

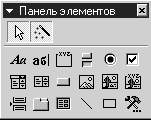
Перенесемо в область вибраних всі доступні поля і натиснемо кнопку **Далее,** У вікні, що з'явилося, вибираємо зовнішній вигляд форми. У даному випадку виберемо вигляд **В один столбец** і натиснемо кнопку Далее. З'явиться вікно, зображене на мал. 6.

Дане вікно використовують для вибору стилю форми. Список можливих стилів перерахований праворуч. Виділяйте один за одним різні імена стилів і ви побачите, що кожному імені відповідає відповідне оформлення звіту. Виберемо стандартний стиль і натиснемо кнопку **Далее.** З'явиться вікно, у яке введіть ім'я форми **Привет** і після цього **Готово.** З'явиться вікно, зображене на мал. 7.



Малюнок 7

Кнопками прокрутки можна викликати інші записи, які включені у форму **Привет.**



Малюнок 8

Кожну Access-форму можна представити на екрані в одному з трьох режимів:

1. у режимі конструктора;
2. у режимі форми (робочому режимі);
3. у режимі таблиці .

Переключення між перерахованими режимами здійснюється або за допомогою команд меню **Вид**, або за допомогою кнопки **Вид** на панелі інструментів ( перша кнопка зліва), що відкриває список із назвами режимів.

Форму можна редагувати у режимі **Конструктор.** При цьому є значна кількість елементів управління для її редагування. Відкрийте, наприклад, форму **Привет** у режимі конструктора. Зверніть увагу, що на екрані в панелі інструментів з'явиться кнопка **Панель элементов** (на ній зображено ключ з молотком). Установіть курсор миші на цю кнопку, клацніть по кнопці миші. Ви побачите, що ліворуч на екрані з'явиться сама панель (див. мал.8), на якій вміщуються різні кнопки для створення елементів управління. Права верхня кнопка - це **Мастер элементов.** Він визначає, яким способом розібратися з елементами управління форми.

Будь-яка форма містить Заголовок, Область даних, і Розділ приміток.

Ми розглянули найпростіші типи форм. Microsoft Access дозволяє створювати досить складні форми. Існують так звані підлеглі, зв’язані, багато сторінкові, смугові та інші форми розробка яких не набагато складніша за ті форми, які ми розглядали.

У формі, як і в запиті, також можна здійснити пошук необхідних даних. При цьому методика роботи щодо пошуку даних близька до аналогічної методики у запитах.

Проектування екранних форм – це захоплююча творча робота, яка є подякою від СКБД, за те що ви нею користуєтесь.

Вибір засобу запровадження даних залежить від частоти їх зміни.

Режим таблиці рекомендується використовувати тоді, коли необхідно одержати найбільш повний огляд даних.

Але якщо дані будуть змінюватися часто, поміщайте їх у форму, оскільки в режимі форми можна сконцентрувати увагу на даних, які відносяться до певного запису, наприклад на інформації, що стосується конкретного клієнта.

Будь-яка форма будується на основі Access-таблиці або запиту. Імена полів беруться зі специфікації таблиці, а поля у формі можна розташувати за своїм розсудом. При цьому варто визначити послідовність розташування полів у формі, об'єднавши логічно взаємозалежні поля в групи, а також внести у форму елементи керування. На основі однієї таблиці користувач може побудувати декілька форм : першу - для керівника, другу - для бухгалтера, третю - для робітника складу і т.д.

Вся інформація форми міститься в її елементах керування, деякі з них безпосередньо пов'язані з полями базової таблиці. У таких елементах можна показати зміст відповідних полів таблиці і внести в них зміни. Інші елементи форми, наприклад назви полів, служать для її оформлення.

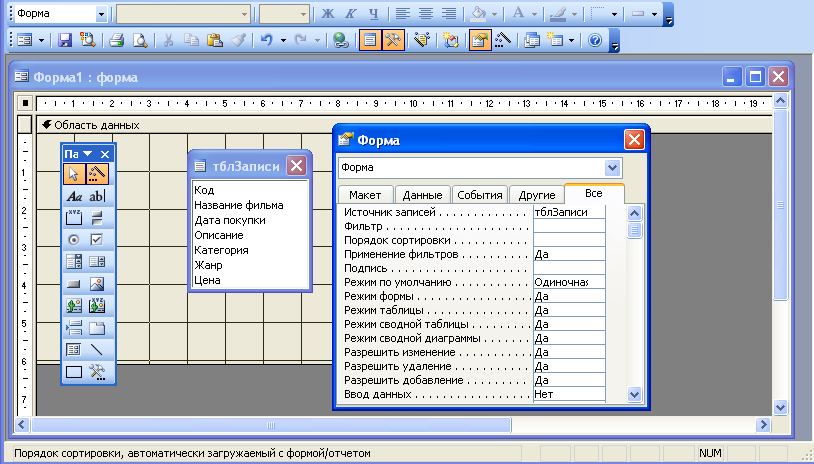
За допомогою конструктора форм зручно будувати будь-яку форму, також редагувати форми, які були створені іншими способами.

**Екран конструктора форм**

Приклади різноманітних вікон, що будуть з'являтися на вашому екрані при роботі в режимі конструктора, показані на мал.9.

Ось список різноманітних засобів, що використовуються при роботі з формою і їх стислий опис.

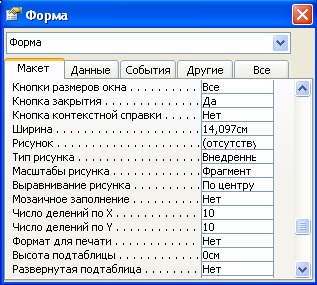
1. **Панель элементов**. Містить всі елементи керування, що використовуються в формах для введення і відображення інформації (в залежності від типу інформації).
2. **Список полей**. Подібний списку полів для роботи в режимі конструктора запитів. Цей список залежить від джерела (таблиці або запиту), на якому заснована форма. За допомогою миші ви можете перемістити поля з списку полів в форму, подібно тому, як це робиться в запитах.
3. **Панель свойств**. На панелях властивостей ви знайдете властивості об'єкту. Властивості описують той об'єкт, якому вони належать (в даному випадку форму). Прикладами властивостей є **Подпись, Источник записей** і т.д.

****

Малюнок 9

**Категорії властивостей**

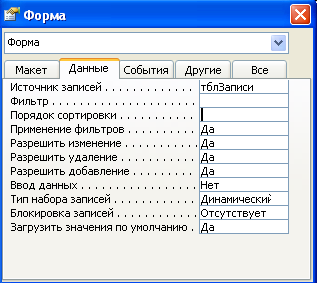
Подвійним натиском миші на області виділення форми викличте появу на екрані панелі властивостей форми (мал. 10).



Малюнок 10

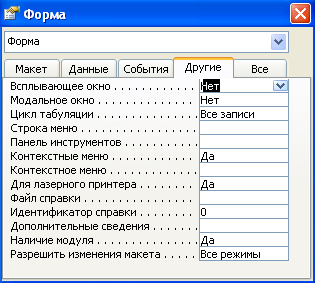
На панелі властивостей є вкладки, що призначені для того, щоб поділити властивості на категорії. Для різноманітних об'єктів Access властивості будуть однакові, а число властивостей в кожній категорії буде змінюватись. Ось вид цих категорій:

1. **Макет.** Визначає, як повинні виглядати об'єкти. Прикладами є властивості **Ширина**, **Кнопки размеров окна, Кнопки размеров** і **Рисунок**.
2. **Данные.** Визначає, яким чином Access використовує з об'єктом дані. Властивість **Источник записей** форми говорить про те, що таблиця або запит використовуються в формі, тут же ви бачите властивості **Фильтр**, **Порядок сортировки** і т. і. (мал. 11).

****

Малюнок 11

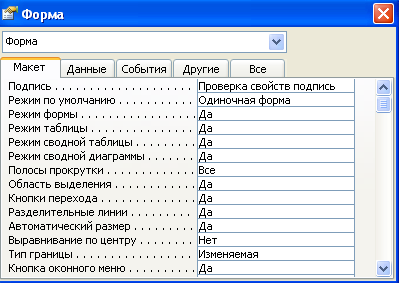
1. **События.** Дозволяють вам приписати певні команди різноманітним подіям, що відбуваються з об'єктом. Прикладом такої події може служити подія On-Click для будь-якої кнопки керування. Це означає, що коли ви натиснете на цій кнопці керування, Access виконає ту задачу, яку ви забажали виконати.
2. **Другие.** Всі інші властивості, які не потрапили в раніше перераховані категорії (мал. 12).

****

Малюнок 12

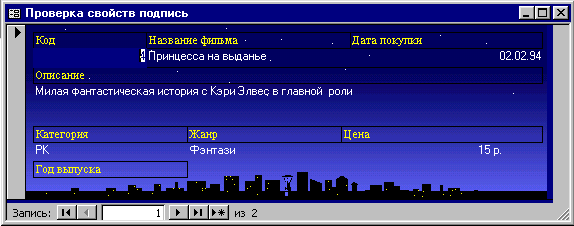
**Зміна властивості Подпись**

Щоб змінити значення властивості Подпись, натисніть на вкладці **Макет** панелі властивостей. Ви побачите рядок властивості **Подпись** вгорі на вкладці (мал.13).

****

Малюнок 13

Змініть значення властивості **Подпись,** що знаходиться зараз на **Проверка свойства подпись** (мал. 13). В режимі форми, в рядку заголовку форми ви побачите те, що надрукували в полі властивості **Подпись** (мал.14).



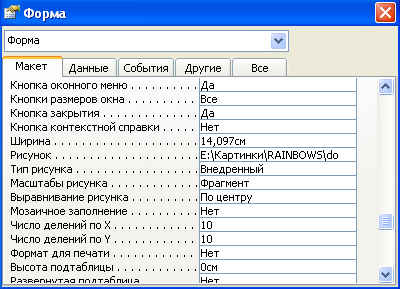
Малюнок14

**Зміна властивості Рисунок**

Щоб змінити властивість **Рисунок** треба зробити наступне:

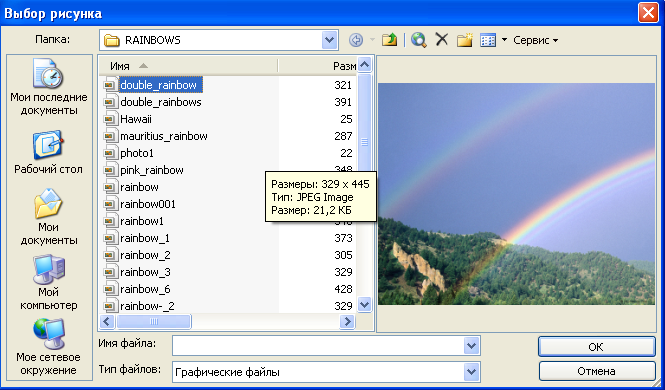
1 Якщо панель властивостей ще не відкрита, відкрийте її подвійним натиском на області виділення форми. Після цього клацніть на вкладці **Макет** на панелі властивостей.

2 Знайдіть властивість **Рисунок**; встановіть там курсор. В кінці рядка властивості з'явиться кнопка будівника. На мал. 15 ви бачите початок шляху до того малюнку, що зараз використовується у властивості **Рисунок**.

****

Малюнок 15

3 Натисніть на кнопці будівника. Access відкриє діалогове вікно **Выбор рисунка**, що дозволяє вам вибрати потрібний графічний файл для фона своєї форми.

****

Малюнок 16

Є і інші властивості, що впливають на зовнішній вигляд графіки, це **Тип рисунка**, **Масштабы** **рисунка**, **Выравнивание рисунка** і **Мозаичное заполнение.**

**Властивості «Данные»**

Деякі з перерахованих в таблиці властивостей **Данные** допомагають при роботі з формами.

|  |  |
| --- | --- |
| Властивість | **Опис** |
| **Источник записей** | Властивість, яка визначає те, на чому грунтується форма |
| **Разрешить изменение** | Виберіть **Да** або **Нет**, щоб дозволити або заборонити користувачу змінювати інформацію в формі |
| **Разрешить удаление** | Виберіть **Да** або **Нет**, щоб дозволити або заборонити користувачу видалять записи |
| **Разрешить добавление** | Виберіть **Да** або **Нет**, щоб дозволити або заборонити користувачу додавати нові записи |

**Контрольні питання до лабораторної роботи 5:**

1. **Що таке форми? Де їх застосовують?**
2. **На основі яких даних створюють форми?**
3. **Які способи створення форм існують?**
4. **Який з способів створення найлегший? Чому?**
5. **Чим він відрізняється від інших способів?**
6. **Переконайте в зручності Автоформ?**
7. **Перелічіть режими відображення форм. Як переключатися між режимами?**
8. **Як зробити різними зовнішній вигляд та стилі ваших форм?**
9. **Для чого потрібні діаграми?**
10. **Що представляє собою «Сводная таблица»?**
11. **Якими способами можна вводити та редагувати інформацію у базі даних?**
12. **У якому випадку використовують форми для роботи з даними?**
13. **Який спосіб створення форм дозволяє будувати змістовні, привабливі, досить складні за структурою та зручні форми?**
14. **Які засоби для роботи з формою існують?**
15. **Які властивості впливають на зовнішній вигляд форми?**
16. **Що змінилось у вашій формі після зміни властивості Підпис форми?**
17. **На основі яких даних зазвичай будується форма?**
18. **Який засіб для роботи з формою містить всі елементи керування?**
19. **Яким чином можна викликати панель властивостей форми?**
20. **Які елементи можна створити у формі за допомогою засобу Список полів?**

**Лабораторна робота №6**

**Тема: Робота з елементами керування в формах бази даних. Розширення можливостей інтерфейсу користувача баз даних за допомогою різних видів форм. Побудова меню роботи з БД.**

**Мета: Навчитися використовувати елементи керування в формах, та застосовувати їх для розширення можливостей форм. Навчитися працювати з багатосторінковими формами, кнопковими формами, головною з підлеглою формами та ін.**

Зміст роботи за варіантом індивідуального завдання:

**При побудові нової форми:**

1. **Додайте змістовний заголовок до області заголовку форми.**
2. **Додайте поля та надписи до області даних вашої форми.**
3. **Додайте елементи до області приміток вашої форми.**
4. **Створіть кнопки для виходу та відкриття іншої форми.**
5. **Створіть прапорці або перемикачі.**
6. **Додайте малюнок (фірмовий знак, логотип).**
7. **За допомогою панелі властивостей елементів змініть їх розмір та розташування.**
8. **Для кожного елементу форми за допомогою панелі властивостей визначте можливості: ім’я, малюнок, прозорість, відображення на екрані та ін.**
9. **Створіть форму з підлеглою формою за допомогою майстра форм.**
10. **Змініть вигляд підлеглої форми (складна структура, режим відображення).**
11. **Створіть багатосторінкову форму.**
12. **Створіть початкову кнопкову форму, яка повинна відкривати інші форми вашої бази даних, у тому числі і вашу головну з підлеглою.**
13. **Створіть головну кнопкову форму за допомогою «Диспетчера кнопочных форм».**
14. **Зробіть так, щоб ваша програма розпочиналась з запуску головної кнопкової форми з робочого стола комп’ютера.**

**Теоретичні відомості:**

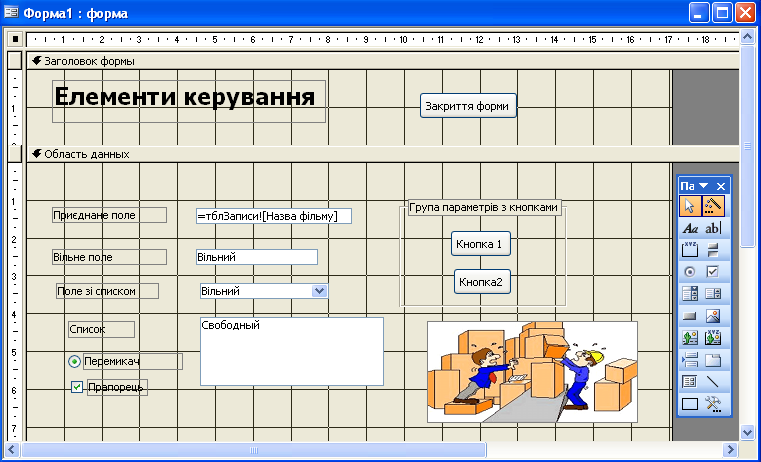
Відображати інформацію в формах, звітах, і вводити дані в форми допомагають елементи керування.

**Елементи керування Access**

Елементи керування представляють собою об’єкти в формах, звітах або на сторінках доступу до даних, які використовують для відображення даних, виконання дій або покращення інтерфейсу. Наприклад, можна використати поле у формі для відображення даних, кнопку в формі для відкриття іншої форми, лінію або прямокутник для розділу та групування елементів керування, які роблять більш зрозумілим їх призначення.

Елементи керування використовуються, щоб полегшити роботу користувача під час пошуку інформації. Крім того, вони підвищують швидкість введення даних. Прикладами елементів керування є кнопки команд, перемикачі, прапорці опцій, поля зі списками і текстові поля (мал. 1).

Елементи керування бувають зв’язаними, вільними або такими, які обчислюються. Зв’язаний елемент керування прив’язаний до поля базової таблиці або запита. Такі елементи керування використовуються для відображення, введення або відновлення значень з полів бази даних. Для вільного елемента керування джерела даних не існує. Вільні елементи керування використовуються для виводу на екран даних, ліній, прямокутників та малюнків. Для елемента керування, що обчислюється в якості джерела даних використовуються вирази. В виразах можливо використовувати дані з полів базової таблиці або запиту для форми, а також дані іншого елемента керування форми.

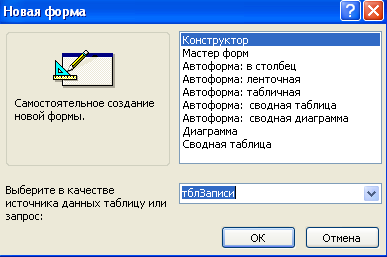
****

Малюнок 1

**Створення форми, що містить елементи керування**

Щоб створити форму не за допомогою майстра форм, зробіть наступне:

1. Відкрийте базу даних (з свого прикладу) і натисніть на вкладці **Формы**.
2. Натисніть на кнопці **Создать**. З'явиться діалогове вікно **Новая форма**.
3. В нижній частині вікна виберіть в якості джерела записів будь-яку таблицю (з вашого прикладу).

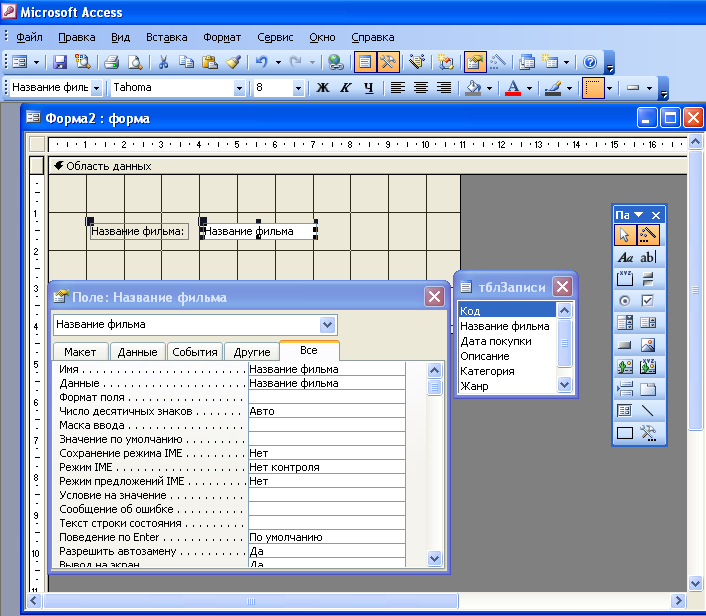


Малюнок 2

1. Виберіть Конструктор і клацніть на кнопці **ОК**. Після цього Access створить порожню форму і відкриє її в режимі конструктора.

Спочатку необхідно додати поле з списку полів.

1. Натисніть на кнопці **Список полей** панелі інструментів. З'явиться список полів.
2. Перемістіть поле з вибраної вами таблиці (в нашому випадку **Название фильма**) в форму і розмістіть його в необхідній позиції. Відпустіть кнопку миші. Поле **Название фильма** з'явиться в формі, як показано на мал. 3**.**

****

Малюнок 3

Зверніть увагу, що в вікні властивостей в якості джерела даних для елементу керування (властивість **Данные**) вказане поле **Название фильма**. Це те, з чим пов'язаний елемент керування. Тому в режимі форми в полі елемента керування з'явиться інформація з поля **Название фильма**. Якщо для властивості **Данные** значення не вказане, елемент керування є *вільним.*

Зверніть увагу, що хоча б цей елемент керування і виглядає як два окремих елемента керування, це дві частини одного цілого: зв'язане поле разом з приєднаним до нього підписом. Підписи приєднуються до більшості елементів керування, призначених для введення даних.

**Переміщення і зміна розмірів елементів керування**

Існує два засоби переміщення елементів керування. Щоб пересунути елемент керування першим засобом, зробіть наступне:

1. Помістіть покажчик миші на елементі керування або на самому елементі, або на приєднаному до нього підпису. Вказівник миші прийме вигляд руки.
2. Побачивши такий вказівник, натисніть і не відпускайте ліву кнопку миші.
3. Пересуньте елемент керування на нове місце в формі.
4. Пересунувши елемент керування в потрібну вам позицію, відпустіть кнопку миші.

А якщо утримувати клавішу <Shift >, то за допомогою клавіш зі стрілками можна змінювати розміри елементу керування.

**Переміщення елементів керування незалежно один від одного**

Щоб пересунути елемент керування незалежно від приєднаного до нього підпису або навпаки пересунути окремо підпис, дійте так:

1. Розмістіть вказівник миші на маркері переміщення тієї частини складеного елементу керування, яку вимагається перемістити (або на підписі, або на текстовому полі). Покажчик миші прийме вигляд руки ,що вказує вверх.
2. Побачивши цей вказівник, натисніть і не відпускайте ліву кнопку миші.
3. Пересуньте елемент керування в нову позицію в формі.
4. Пересунувши елемент керування в потрібну вам позицію, відпустити кнопку миші.

**Одночасне переміщення декількох елементів керування**

Для того щоб ви могли перемістити декілька елементів керування водночас, спочатку ви повинні помістити в форму пару нових полів. Помістіть в форму поля **Описание** і **Категория**.

Щоб захопити більше одного поля відразу, натисніть в списку полів на першому з них (поле **Описание**). Після цього, утримуючи клавішу <Ctrl> натисніть на наступному полі (поле **Категория**). Пересуньте обидва поля в форму, помістивши їх під полем НазваниеФильма, і відпустіть кнопку миші. В результаті поля виявляться в формі.

Щоб водночас пересунути декілька полів в формі, виконайте наступне:

1. Помістіть вказівник миші в формі, під нижнім правим кутом останнього з елементів керування, що необхідно пересунути.
2. Натисніть і не відпускайте ліву кнопку миші.
3. Пересуньте вказівник миші вгору і ліворуч над всіма елементами керування, що необхідно виділити. Ви повинні побачити рамку, захоплюючу ці елементи керування.
4. Захопивши всі елементи керування, що ви хотіли б пересунути, відпустіть кнопку миші.
5. Помістіть вказівник миші на будь-який з елементів керування, що потрібно перемістити. Коли покажчик прийме вигляд руки, натисніть ліву кнопку миші і пересуньте елементи керування туди, куди вам потрібно.

Цей же засіб може бути використаний для вибору декількох елементів керування при зміні їх розмірів і вирівнювання.

**Вирівнювання елементів керування**

В більшості випадків при розміщенні елементів керування в формі бажано їх чітко вирівнювати. В Access є можливість легко зробити це.

1. Виділити елементи керування, що необхідно вирівняти. При виборі декількох елементів керування для вирівнювання або зміни розмірів вибирайте тільки ту частину складених елементів керування, яку ви хотіли б вирівняти (або розміри якої ви хотіли б змінити).
2. Виберіть пункт **Формат**, **Выровнят**ь з меню конструктора форми. В підміню, що з’явиться ви побачите наступні варіанти вирівнювання елементів керування: **по левому краю, по правому краю, по верхнему краю, по нижнему краю** або **по узлам сетки.**
3. Виберіть пункт по правому краю з меню **Выровнять**. Ви побачите, як текстові вікна вирівняються за самим правим елементом керування.
4. Повторивши пункти 1-3, вирівняйте підписи елементів керування, виділивши і підрівнявши їх по лівому краю.

**Зміна розмірів елементів керування**

Існує два засоби зміни розмірів елементів керування, в залежності від їхнього числа.

Щоб змінити розмір одного елементу керування, зробіть наступне:

1. Натисніть на елементі керування, щоб виділити його.
2. Протягніть покажчик миші через маркер розміру. Коли покажчик миші опиниться над маркером розміру, він перетвориться в пару направлених в протилежні сторони стрілок, що вказують в різноманітних напрямках.
3. Утримуючи ліву кнопку миші, пересуньте маркер розміру елементу керування. Ви побачите, як він звужується або розширюється, в залежності від того, в якому напрямку ви його рухаєте.

При зміні розміру декількох елементів керування можна виділити елементи керування, а потім:

-використовувати засіб, описаний вище,

або

-вибрати пункт **Размер** з меню **Формат**. Після цього в підменю, яке з’явилось вибрати один з можливих варіантів: **по размеру данных, по узлам сетки, по самому высокому, по самому низкому, по самому широкому** і **по самому узкому.**

**Робота з властивостями елементу керування**

Можна змінити властивості елементу керування так само як і змінити властивості форми.

Щоб побачити, як змінюються властивості елементу керування, давайте змінимо властивість **Текст строки состояния** в вікні властивостей. Для цього треба виконати наступне:

-Помістить курсор в рядок властивості **Текст строки состояния** в вікні властивостей.

-Замінити поточний текст в рядку цієї властивості на *“Это название фильма”*.

-Переключитися в режим форми.

Тепер, при переході в поле **Название фильма**, надрукований вами текст *“Это название фильма”* буде з'являтися в рядку стану в нижній частині вікна Access.

Панель інструментів форматування дозволяє змінювати зовнішній вигляд елементів керування.

**Інші типи форм**

Ми розглянули найпростіші типи форм. Microsoft Access дозволяє створювати досить складні форми. Як вже відомо існують так звані підлеглі, зв’язані, багатосторінкові, смугові та інші форми, розробка яких не набагато складніша за ті форми, які ми розглядали.

Багатосторінкова форма – це форма, яка використовуютьсяпри необхідності вивести на екран записи, кожен з яких містить великий об'єм інформації.

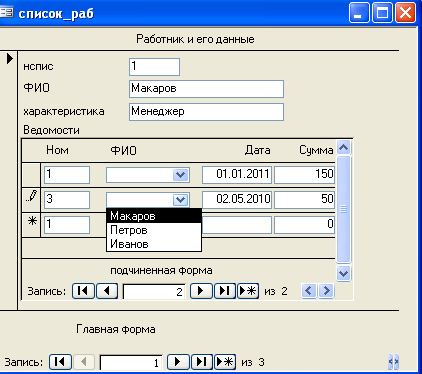
Підлегла форма - це форма, яка знаходиться всередині іншої форми. Первинна форма називається головною формою, а форма всередині форми називається підлеглою формою.

# Зв’язок головної та підлеглої форм в Microsoft Access

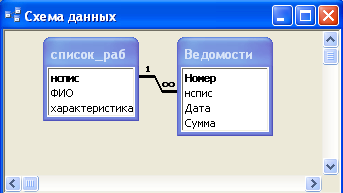
Підлегла форма звичайно використовується для виведення даних з таблиць або [запитів](javascript:HelpPopup('actip9.hlp','defQuery');), з відношенням “один-до-багатьох”. Наприклад, таблиця «Типи» та таблиця «Товари» зв’язані відношенням «один-до-багатьох». Кожен тип товарів складається з декількох товарів. Грунтуючись на цих таблицях можна створити підлеглу форму “Товари” всередині головної форми “Типи”. Ми розглянемо, наприклад, форму “Список\_раб” (мал. 4), яка вміщує підлеглу “Ведомости”.

Зі створенням форми та підлеглої форми, які грунтуються на таблицях, зв’язаних відношенням «один-до-багатьох», головна форма представляє сторону «один» відношення, а підлегла форма представляє сторону «багато» (мал.5).

Головна форма синхронізується з підлеглою формою таким чином, що в підлеглій формі виводяться тільки записи, зв’язані з записом в головній формі. Якщо підлегла форма створюється за допомогою майстра або шляхом перетягування форми, таблиці або запиту з вікна бази даних в іншу форму, Microsoft Access автоматично синхронізує головну форму з підлеглою формою із виконанням двох наступних умов.



Малюнок 4

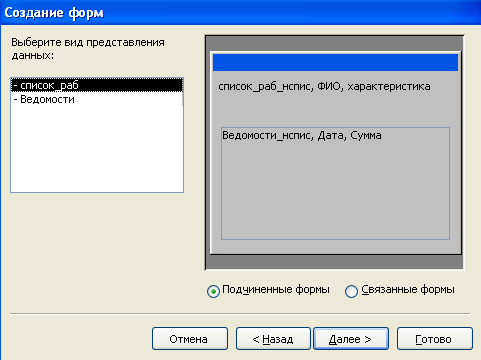


Малюнок 5

* Визначені відношення для вибраних таблиць або для базових таблиць вибраних запитів.
* Головна форма грунтується на таблиці з первинним ключем, а підлегла форма грунтується на таблиці, яка зберігає поле з таким же іменем, як і в цього первинного ключа, і з таким же або сумісним типом даних. Наприклад, якщо первинний ключ базової таблиці головної форми має тип даних «Счетчик», а його влативість **Размер поля (FieldSize)** має значення **Длинное целое**, то відповідне поле в базовій таблиці підлеглої форми повинно мати тип даних «Числовой» з властивістю **Размер поля (FieldSize)**, яка має значення **Длинное целое**. З вибором запиту або запитів базові таблиці цих запитів повинні також відповідати цим умовам.

Створимо за допомогою майстра форм форму “Список\_раб”, яка буде грунтуватись на таблиці “Список\_раб” та таблиці “Ведомости”.

1. В списку діалогового вікна **Новая форма** виберіть **Мастер форм**.
2. Виберіть з списку таблицю або запит. Наприклад, щоб створити форму “Список\_раб”, яка демонструє в підлеглій формі список виданих сум, які відносяться до конкретного працівника, виберіть таблицю “Список\_раб” (сторона “один” відношення “один-до-багатьох”).
3. Виберіть поля таблиці або запиту, які потрібно включати в форму.
4. В тому ж діалоговому вікні майстра виберіть з списку іншу таблицю або запит. В розглянутому прикладі виберіть таблицю «Ведомости» (сторона «багато» відношення «один-до-багатьох»). Послідовність вибору таблиць або запитів не має значення.
5. Виберіть поля таблиці або запита, які потрібно включити в форму.
6. Якщо до запуску майстра зв’язки були встановлені коректно, то після натискання кнопки **Далее** пропонується відібрати таблиці або запити, які будуть використовуватись для представлення даних. Використовуючи той же приклад форми “Список\_раб”, виберіть рядок –**Список\_раб**.
7. В тому ж діалоговому вікні виберіть перемикач **Подчиненные формы** (див. мал.6).



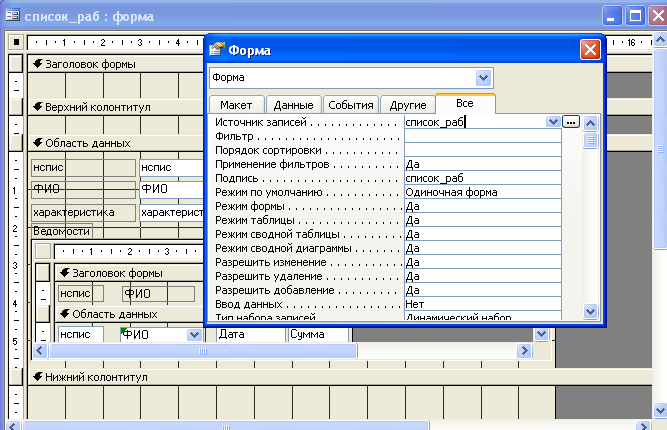
Малюнок 6

1. Слідуючи інструкціям інших діалогових вікон майстер Microsoft Access, у результаті створить дві форми: одну для форми та елемента керування **“Подчиненная форма”**, а іншу для підлеглої форми.

З використанням форми з підлеглою формою для вводу нових записів поточний запис в головній формі зберігається з переходом до підлеглої форми. Це гарантує, що записи з таблиці на стороні «багато» будуть мати зв’язаний запис в таблиці на стороні «один». Це також автоматично зберігає кожен запис, який додається в підлеглу форму. Підлегла форма може відображатись у вигляді таблиці, або у вигляді простої чи стрічкової форми. Головна форма може відображатися тільки як проста форма.

Головна форма може мати будь-яке число підлеглих форм, якщо кожна підлегла форма поміщається в головну форму. Є також можливість створювати підлеглі форми до сьомого рівня вкладення. Це позначає, що можна мати підлеглу форму в головній формі, а іншу підлеглу форму в цій підлеглій формі и т.д. Наприклад, можна створити головну форму, яка відображає інформацію про клієнтів, підлеглу форму, яка відображає інформацію про замовлення, та ще одну підлеглу форму, яка відображає інформацію про замовлення товарів.

Під час створення підлеглої форми користувач має можливість вказати її відображення в наступних режимах: тільки в [режимі таблиці](javascript:HelpPopup('actip9.hlp','defDatasheetView');), тільки як проста або стрічкова форма в режимі форми або в будь-якому вигляді. Підлеглі форми, відображені як прості або стрічкові форми, мають бути простими або мати достатньо складну структуру, включаючи заголовки, області приміток і т.д.



Малюнок 7

# Визначимо режим відображення підлеглої форми

Під час створення підлеглої форми за допомогою майстра форм або майстра підлеглих форм, Microsoft Access встановлює значення властивості **Режим по умолчанию (DefaultView)** підлеглої форми в відповідності з тим, що вибрав користувач в діалогових вікнах майстра. Наступні дії можуть бути корисними у створенні підлеглої форми без допомоги майстра, або для зміни режиму, вибраного в майстрі.

1. Відкрийте головну форму в режимі конструктора. Підлегла форма буде виведена у режимі конструктора в елементі керування **Подчиненная форма**.
2. Виберіть елемент керування «підлегла форма». Тепер можна натиснути його ще раз та встановити властивість форми як і у випадку відкриття підлеглої форми в режимі конструктора з вікна бази даних.
3. Двічі натисніть область виділення форми, щоб відкрити вікно властивостей підлеглої форми.
4. В комірці властивостей **Режим по умолчанию (DefaultView)** виберіть один з режимів: **Простая форма**, **Ленточная форма** або **Таблица**.
5. В комірці властивостей **Допустимые режимы (ViewsAllowed)** виберіть **Форма**, якщо перемикатися в режим таблиці не потрібно. Виберіть значення **Таблица**, якщо не потрібно перемикатися в режим форми. Якщо потрібно зберегти можливість перемикання між режимами таблиці і форми, виберіть значення **Все**.

Багатосторінкові форми також часто користуються попитом. Для таких форм використовуємо у структурі форми області верхнього та нижнього колонтитулу.

Для того, щоб вставити в форму номера сторінок треба вибрати в меню **Вставка** **Номера страниц.** В області нижнього або верхнього колонтитула з’явиться елемент поле, в якому буде такий вираз: ="Страница " & [Page].

# Створення кнопкової форми

Якщо створювати базу даних за допомогою майстра автоматично створюється кнопкова форма, яка допомагає пересуватись базою даних. На кнопковій панелі містяться кнопки, при натисканні яких відкриваються форми або звіти (або відкриваються інші кнопкові форми, за допомогою яких відкриваються додаткові форми та звіти), виконується вихід з Microsoft Access або настроюється сама кнопкова форма. Диспетчер кнопкових форм дозволяє створювати кнопкові форми, які подібні тим, що створює майстер баз даних.

# Створення кнопкової форми за допомогою «Диспетчера кнопочних форм»

Створюючи базу даних за допомогою майстра автоматично створюється кнопкова форма, яка дозволяє легко переміщуватись формами та звітами бази даних. В базі даних, що вже існує створити кнопкову форму можна за допомогою «Диспетчера кнопочних форм»:

1. Виберіть в меню **Сервис** команду **Служебные программы** та підкоманду **Диспетчер кнопочных форм**.
2. В діалоговому вікні **Диспетчер кнопочных форм** натисніть кнопку **Изменить**.
3. В діалоговому вікні **Изменение страницы кнопочной формы** наберіть ім’я кнопочної форми в полі **Название кнопочной формы** та натисніть кнопку **Создать**.
4. В діалоговому вікні **Изменение элемента кнопочной формы** наберіть у полі **Текст** текст для першого елемента кнопкової форми та виберіть команду в полі із списком **Команда**.
5. Для більшості команд під полем **Команда** відчиняється нове поле зі списком. За необхідністю, виберіть потрібний елемент в цьому полі.
6. Таким чином створюють інші кнопки кнопкової форми. Щоб змінити або видалити яку-небудь з створених кнопок, виберіть її ім’я з списку **Элементы данной кнопочной формы** та натисніть кнопку **Изменить** або **Удалить**. Якщо потрібно змінити порядок елементів кнопочної форми, виберіть елемент з списку та скористайтесь кнопками **Вверх** або **Вниз**.
7. Натисніть кнопку **Закрыть**.

Диспетчер кнопкових форм дозволяє створювати кнопкові форми, які відкривають інші кнопкові форми, щоб назначити кнопкову форму стартовою (тобто, яка автоматично відкривається з відкриттям бази даних), виберіть ім’я кнопкової форми в діалоговому вікні **Диспетчер кнопочных форм** та натисніть кнопку **По умолчанию**. Щоб із натиском кнопки цієї кнопкової форми відчинялась інша кнопкова форма, на кроці 5 описаної вище процедури виберіть команду **Перейти к кнопочной форме** у полі **Команда** та вкажіть кнопкову форму, на яку потрібно перейти.

При створенні кнопкової форми за допомогою диспетчера кнопкових форм Microsoft Access створює таблицю «Switchboard Items», яка описує текст і дії кнопок форми. Якщо потім змінити кнопкову форму в режимі конструктора форми, програма може перестати працювати. Якщо потрібно буде значно змінити кнопкову форму, краще всього створити її заново і назначити стартовою.

Якщо створити ярлик для будь якої форми і помістити його на робочий стіл вашого комп’ютера, то користувачу буде зручніше працювати з базою даних. Щоб помістити форму **“**Список**”** на робочий стіл викликаємо контекстне меню форми та вибираємо **Создать ярлык**. У пункті **Размещение** обираємо потрібний шлях.

**Контрольні питання до лабораторної роботи 6:**

1. **Що таке елемент керування?**
2. **Яких типів елементи керування бувають?**
3. **Які елементи називають вільними?**
4. **Які елементи називають зв’язаними?**
5. **Які елементи називають обчислювальними?**
6. **Перелічіть елементи керування, які присутні на панелі елементів.**
7. **Якими способами можливо змінювати розташування та розмір елементів керування?**
8. **Перелічіть типи форм, які ви знаєте.**
9. **Яку форму називають головною формою?**
10. **Яку форму називають підлеглою?**
11. **В яких випадках використовують підлеглі форми і яким чином повинні бути зв’язані таблиці на основі яких створюють такі форми?**
12. **Зі створенням форми та підлеглої форми, які грунтуються на таблицях, зв’язаних відношенням «один-до-багатьох», яку сторону відношення представляє підлегла форма? Продемонструйте на схемі даних.**
13. **Якими двома способами можна створити підлеглу форму?**
14. **Із виконанням яких двох умов Microsoft Access автоматично синхронізує головну форму з підлеглою формою?**
15. **У якому вигляді може відображатись підлегла форма?**
16. **У якому вигляді може відображатись головна форма?**
17. **Яке число підлеглих форм може мати головна форма, якщо кожна підлегла форма поміщається в головну форму?**
18. **До якого рівня вкладення є можливість створювати підлеглі форми?**
19. **Чим характеризуються багатосторінкові форми?**
20. **Що представляє собою головна кнопкова форма?**
21. **Які є методи створення головної кнопкової форми?**
22. **Яким чином можна зробити, щоб база даних запускалась з головної кнопкової форми з робочого стола комп’ютера?**

**Лабораторна робота №7**

Тема: Отримання підсумкової інформації бази даних за допомогою звітів. Редагування звіту за допомогою інструментів конструктора звітів.

**Мета: Навчитися швидко будувати звіти за допомогою майстра звітів. Ознайомитися з можливостями конструктора звітів. Навчитися створювати групи та сортувати дані у звітах.**

Зміст роботи за варіантом індивідуального завдання:

1. **Побудуйте звіти за допомогою автозвітів.**
2. **Створити звіт за допомогою майстра звітів.**
3. **Розглянути звіт у режимі конструктора. Виконайте вказівки удосконалення звітів з пункту “Покращення звіту”(дивись теоретичні відомості).**
4. **Різними способами зробіть групування й сортування інформації. Роздрукуйте.**
5. **Виберіть різні види макетів і стилів для усіх своїх звітів. Роздрукуйте.**
6. **Побудуйте поштову наклейку за допомогою майстра поштових наклейок.**
7. **Заповніть побудовані вами наклейки інформацією у режимі конструктора. Роздрукуйте наклейки.**
8. **Використайте майстра діаграм для зображення інформації у своєму звіті.**
9. **Створіть обчислювальні поля.**
10. **Додайте в звіт номер сторінки, дату й час. Роздрукуйте отриманий вдосконалений звіт.**

**Теоретичні відомості:**

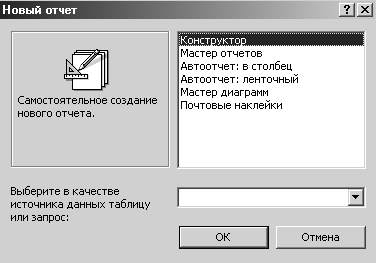
**Використання майстра звітів**

Звіт – це такий компонент бази даних, що використовується для того, щоб вибрати дані з однієї чи кількох таблиць, здійснити над ними обчислення, підвести підсумки і вивести їх на екран чи на принтер. Звіти є близькими до форми. Структури у тих і інших багато в чому схожі, але формати **звітів** відповідають стандартам паперу. Багато з того, що викладено для форм, справедливе і для звітів. Для роботи зі звітами найчастіше використовують режим конструктора.

Можливості звітів дуже великі. З їх допомогою можна групувати інформацію, що виводиться, здійснювати проміжні обчислення. Для кожної групи можна задати окремі заголовки і примітки. Більша частина даних у звіті отримується з базової таблиці, запросу або інструкції SQL, яка є джерелом даних для звіта. Інша інформація про звіт зберігається у його структурі.

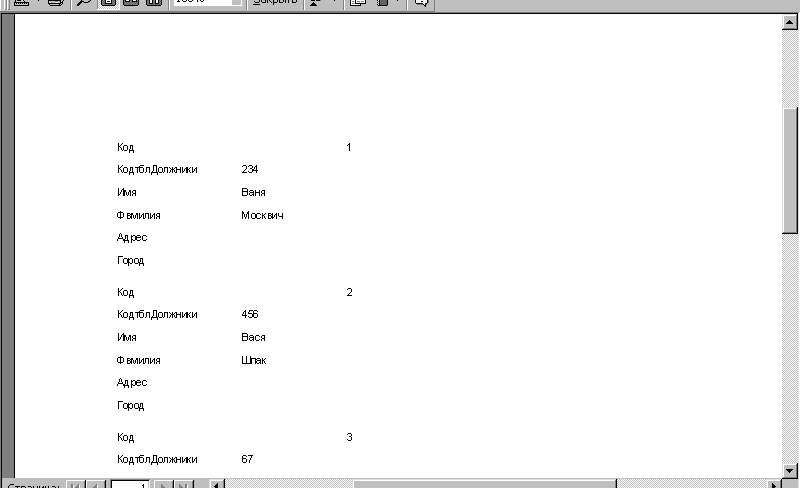
Access пропонує різні засоби для створення звітів, до яких відносяться ряд майстрів та режим конструктора. Крім того існує можливість швидко створити найпростіший звіт за допомогою функції **Автоотчет**. Щоб побачити ці майстри, у вікні бази даних натисніть пункт звіти на панелі об’єктів, а потім на кнопку Создать. Відкриється діалогове вікно Новый отчет .

Діалогове вікно Новый отчет схоже на вікно Новая форма і майже також дозволяє визначити джерело записів (таблицю або запит) для звіту, який ви бажаєте створити. Виберіть потрібну вам таблицю (мал. 1).



Малюнок 1

Два існуючих у списку автозвіту, в стовпчик і стрічковий, пропонують швидкі й зручні способи складання звітів. Складені з їх допомогою звіти легше читати, і виглядають вони більш професійно. Можна вибрати **Автоотчет:** в стовпець з діалогового вікна **Новый отчет** або той самий звіт, але на цей раз складений в результаті вибору **Автоотчета** : стрічковий в діалоговому вікні **Новый отчет.** На малюнку 2 зображений швидкий звіт типу **Автоотчет: в столбец** в режимі попереднього перегляду. Автозвіт годиться для побудови швидких звітів, але, такі звіти бажано змінювати та покращувати.

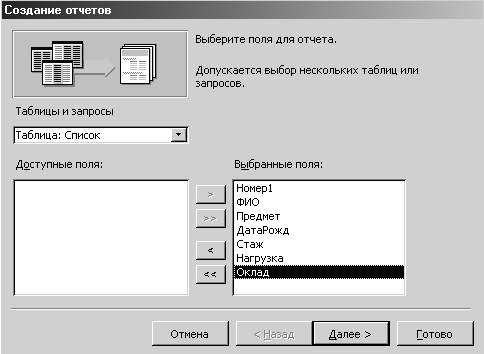


Малюнок 2

**Побудова звітів за допомогою майстра звітів**

Вище було показано, як створювати звіт, який має в змісті інформацію з таблиці , у стислому вигляді, котрий ви не могли б змінити. Зараз ви навчитесь складати аналогічний вид звіту і формувати його.

У діалоговому вікні **Новый отчет** виберіть в якості джерела записів таблицю, опцію **Мастер отчетов** і натисніть кнопку ОК. В результаті Access запустить майстер звітів. В першому екрані майстра звітів оберіть поля, які потрібні для звіту. Виберіть всі поля (кнопка » ), а потім клацніть на кнопці **Далее** (мал. 3).

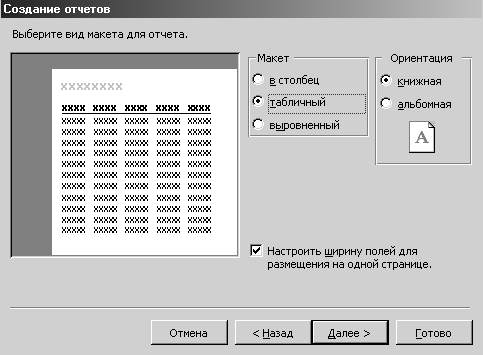


Малюнок 3

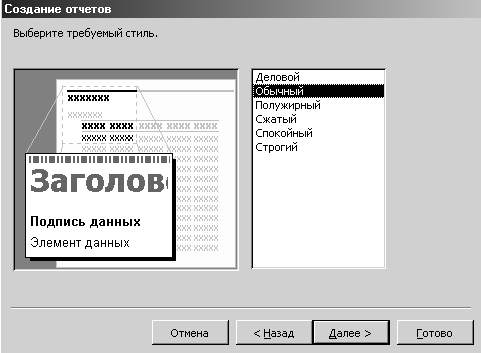
Обравши інформацію, яку ви хотіли б бачити у звіті, ви здатні організувати її з врахуванням її різних аспектів. Це можна зробити на наступному екрані майстра звітів.

Цей екран дозволяє вам вибрати засіб угрупування даних. В результаті цього ви здатні відображати поля в різній послідовності . Угруповуючи дані, ви надаєте вказівку Access, які поля необхідно показати "як групу" і як підраховувати підсумки по даним. За допомогою сортування можна розташувати дані у звіті у потрібному порядку.

Щоб змінити групування, використовуйте кнопки вибору, які знаходяться на екрані майстра звітів. Для зразка, погодьтесь з групуванням, яке запропонує Access, і клацніть на кноці **Далее**. На екрані вас запитають, як сортувати записи у звіті. Виберіть потрібне поле і натисніть **Далее**.



Малюнок 4



Малюнок 5

**Вибір виду макету для звіту**

Цей екран дозволяє вибрати вид макету , який буде використовуватися для звіту, - ви здатні вибирати з цілого ряду запропонованих макетів . При цьому , ліворуч на екрані , обраний макет буде відображено у графічному вигляді. Окрім того можна вибрати орієнтацію аркушів звіту на екрані - книжкову або альбомну (мал. 4 ).

На цьому ж екрані можна зробити так, щоб всі поля були на одній сторінці. Якщо ви не бажаєте, щоб всі поля були на одній сторінці, Access надрукує їх на двох сторінках. Обравши потрібний вид макету, натисніть на кнопці **Далее**.

**Вибір стиля звіту**

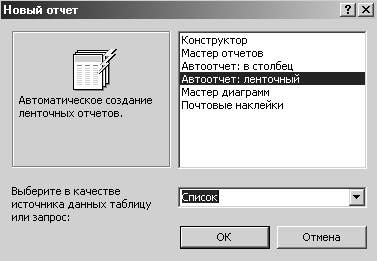
На цьому екрані ви можете вибрати стиль зі списку різних стилів. **Стиль** звіту надає інформації потрібний зовнішній вигляд, в залежності від аудиторії, на яку він розрахований (Мал. 5).

Щоб перейти до наступного екрану, клацніть на кнопці **Далее**. В поле для імені звіту надрукуйте заголовок звіту. Цей текст буде використано у режимі попереднього перегляду у рядку заголовку і в назві звіту.

Клацніть на кнопці **Готово**. Access побудує звіт і відчинить його у режимі попереднього перегляду. Закрийте звіт.

**Використання автозвітів та майстра звітів**

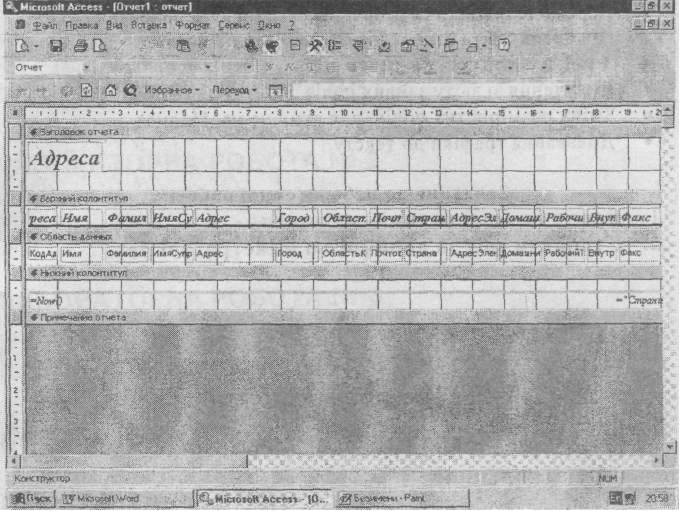
Може трапитись, що вам потрібно звіт, для якого не підходе автозвіт або майстер звітів. Тоді можна створити звіт за допомогою конструктора звітів.



Малюнок 6

В Access мається деякі інструменти для роботи зі звітами. Для початку створимо автозвіт Виберіть таблицю як джерело записів і команду **Автоотчет ленточный** (мал. 6). Створимо Автозвіт.

Клацніть на кнопці Вид, щоб перейти у режим конструктора звітів. Перед тим, як продовжити роботу, ознайомимось з деякими інструментами, які використовуються в режимі конструктора звітів (мал. 7).

Малюнок 7

Дані у звіті можуть бути розбиті на розділи. Кожен розділ має конкретне призначення та друкується на сторінці та у звіті у заданому порядку.

Нижче приведено список деяких інструментів:

* **Панель элементов.** Тут ви знайдете усі елементи управління, які використовуються у звіті (мал.8).
* **Список полів.** Складається з усіх полів джерела записів для звітів.
* **Вікно властивостей.** Показує всі властивості усіх об’єктів, які використовуються для звіту, такі як елементи управління, розділи звіту і сам звіт.

та розділів звіту:

* **Область заголовка звіту.** Схожа з областю заголовка форми. Взагалі використовується при будуванні титульного листа звіту. Друкується перед верхнім колонтитулом на першій сторінці звіту.
* **Область верхнього колонтитула.** Тут взагалі розташовуються назви стовпців, або назва звіту, якщо треба, щоб воно повторювалось на кожній сторінці.
* **Область нижнего колонтитула.** Використовується для вказівки номера сторінки.
* **Область даних.** Призначена для відображення даних окремих записів з джерела записів та звіту.
* **Область примечаний**. Призначається для приміток, інструкцій користувача а також ітогових значень та ін.



Малюнок 8 (Панель елементів)

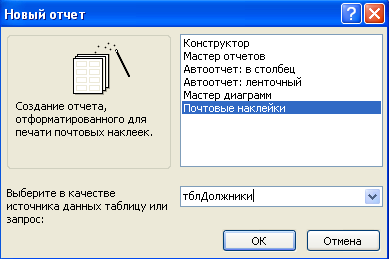
Access має ряд майстрів за допомогою яких будуються звіти. У списку діалогового вікна **Новый отчет** є можливість створювати поштові наклейки.

**Створення поштових наклейок за допомогою майстра**

Майстер по створенню поштових наклейок дуже простий у використанні та при цьому має великі можливості. Цей майстер підтримує стандарт Avery, хоч з його допомогою ви можете будувати й свої власні наклейки.

За допомогою майстра по створенню поштових наклейок можна створити наклейки для розсилки нагадування людям, які взяли на деякий час фільм з вашої відеотеки. Для цього, знаходячись на в вкладці **Отчеты** в вікні БД, зробіть наступні кроки:

1. Натисніть на кнопці **Создать**.
2. Виберіть в якості джерела записів таблицю, потім виділить команду **Почтовые наклейки** (мал. 9) і клацніть на кнопці ОК, Access запустить майстра по створенню поштових наклейок.

****

Малюнок 9

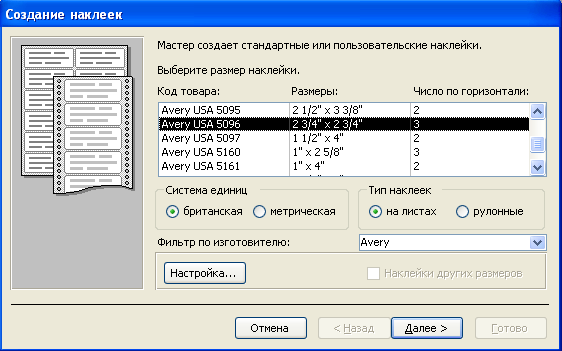
**Вибір номера Avery**

На першому екрані майстра вам потрібно вказати номер Avery, який буде використаний при створенні поштових наклейок. Avery являється стандартом при роздрукуванні поштових наклейок на принтерах.

Із списку виберіть число 5096, при якому на одному листі друкуються три наклейки (мал. 10). Якщо ви не знайшли числа 5096 в списку, клацніть на кнопці того перемикача в групі перемикачів **Система единиц**, біля якого написано **британская**.

Завжди краще ще раз впевнитися у вірному виборі номера Avery. Після того як ви натиснете на кнопку майстра **Готово**, для зміни розмірів наклейки вам прийдеться починати все з самого початку. Взагалі, це стосується всіх майстрів: після закінчення роботи з майстром вже неможливо повернутися назад.

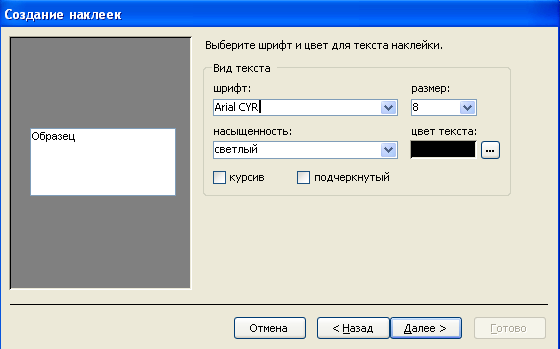
Щоб перейти до наступного діалогового вікна, натисніть на кнопці **Далее**.



Малюнок 10

**Вибір шрифтів та їх розмірів для наклейок**

Наступне вікно дозволяє вибирати шрифти та їх розміри для наклейок (мал.11).

****

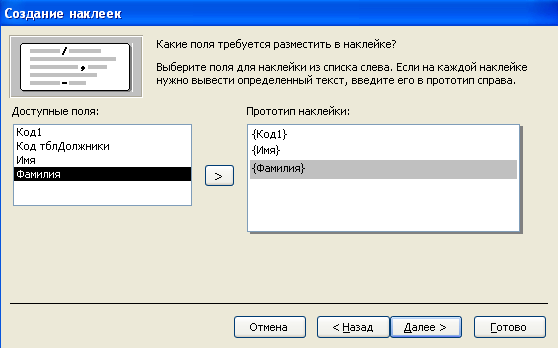
Малюнок 11

Обравши підходящий вид й розміри шрифту, клацніть на кнопці **Далее**, щоб перейти до наступної сторінки. Саме тут здійснюється основна робота по створенню наклейок.

**Заповнення наклейок інформацією**

Тут ви можете оформити наклейку саме так, як вам потрібно, обираючи поля й вводячи відповідний текст.

Створіть прототип наклейки (мал. 12), двічі клацнувши на потрібних полях або виділив їх й клацнувши на кнопці вибору (>).

****

Малюнок 12

Щоб побудувати наклейку, показану на мал.12, виконайте наступне:

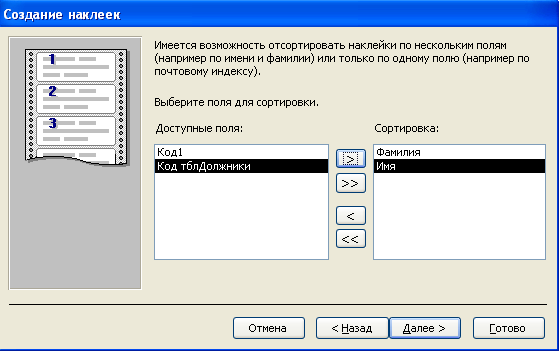
1. Двічі клацніть на полі **Фамилия**. В прототипі наклейки з’явиться текст {Фамилия}.
2. Поставте після тексту {Фамилия} кому й прогалину.
3. Двічі клацніть на полі **Имя**.
4. Нажміть клавішу <Enter>. Лінія підсвітки передвинеться на рядок нижче, розмістившись під тільки що введеним текстом.
5. Повторіть тільки що виконані кроки, щоб ввести решту полів й відповідний текст в наклейку. Натисніть на кнопці **Далее** й перейдіть до наступного вікна.

**Вибір полів для сортування**

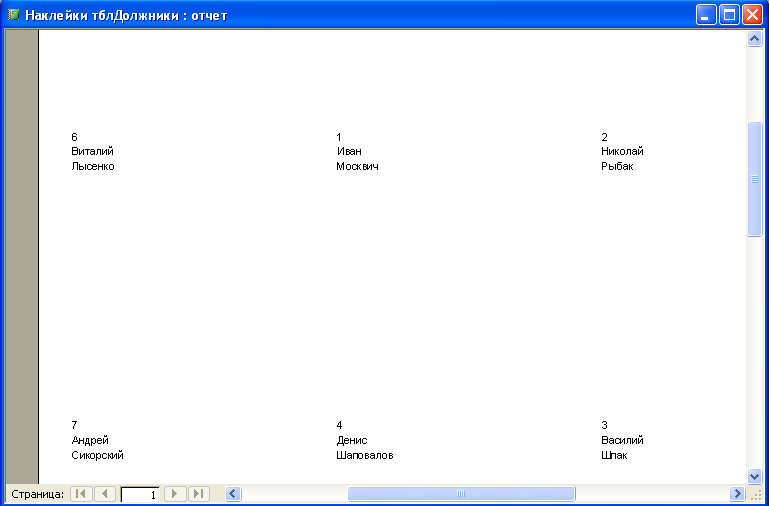
Тут ви можете вибрати поля для сортування поштових наклейок. Виберіть поля **Фамилия** та **Имя**. Виділіть їх, натисніть на кнопці переміщення вправо (>) (мал. 13). Клацніть на кнопці **Готово**, щоб завершити роботу з майстром.

В результаті Access створить наклейки, показані на мал. 14.

Виключно легко створювати наклейки для різноманітних адресатів з однієї таблиці.



Малюнок 13



Малюнок 14

**Друкування звітів**

В Access підтримується стандарт Windows, який дозволяє або попередньо продивитися звіт перед друкуванням за допомогою спеціальних опцій, або відразу друкувати звіт на принтері. Також підтримується діалогове вікно, яке дозволяє друкувати звіти на різних принтерах. Перш ніж перейти до розглядання діалогових вікон **Печать** та **Параметры страницы**, давайте розглянемо попередній перегляд.

**Попередній перегляд**

Ніколи не зайвий перегляд звіту, перед його друкуванням на принтері,- це дозволяє зекономити багато паперу й часу. Особливо це корисно при частих змінах формату.

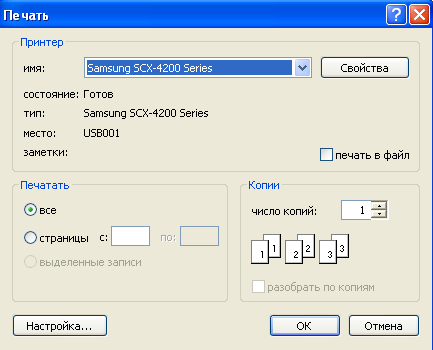
Знаходячись в режимі попереднього перегляду, можна використовувати цілий ряд команд з панелі інструментів.

Таблиця :

|  |  |
| --- | --- |
| **Команда** | **Описання** |
| Печать | Дозволяє друкувати звіт з режиму попереднього перегляду. |
| Масштаб | Дозволяє переключатися між режимом **Окно**, в якому звіт підганяється під розміри вікна, й розміром звіта у відсотках . |
| Одна страница | Переключає в режим односторінкового перегляду. |
| Две страницы | Переключає в режим двосторінкового перегляду. Він зручний, якщо ви бажаєте подивитися, який вигляд буде мати звіт з обох сторін листа. Особливо він зручний при форматуванні, хоча ви можете використовувати вказівник миші, щоб побачити частину сторінки в збільшеному масштабі. |
| Несколько страниц | Ця кнопка дозволяє переглядати на екрані одночасно від 1 до 6 сторінок звіту. |
| Связи с Office | Дозволяє зв’язувати свій звіт з продуктами Word та Excel. |
| Окно БД | Переносе вас у вікно БД, в якому можливо вибрати інші об’єкти з БД. |
| Новый объект | Якщо з’явилась необхідність в створені нового об’єкту для БД, натисніть на цій кнопці, і з випадаючого списку виберіть об’єкт. Якщо ви передумали, натисніть на кнопці **Отменить** в діалоговому вікні **Новый объект**, щоб повернутися назад. |
| Помощник | Якщо ви відключили **Помощника**, натисніть на цій кнопці, щоб увімкнути його знову, готового допомогти вам вирішити любу проблему. |

**Параметри діалогового вікна Печать**

Якщо натиснути на кнопці **Печать** панелі інструментів в режимі попереднього перегляду або в режимі конструктора, звіт буде одразу ж надруковано на принтері. Це дуже зручно. Однак, якщо виникає питання, на який принтер піде звіт, або про настройку цього принтера, краще вибрати **Файл**, **Печать** із меню. В результаті відкриється діалогове вікно **Печать** (мал.15).

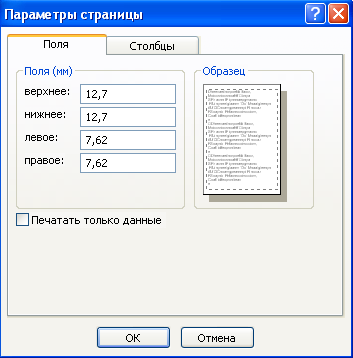
****

Малюнок 15

Основні параметри цього діалогового вікна:

* **Принтер.** Використовуючи випадаючий список принтерів, доступних з вашого комп’ютера, можна вибрати той, на який буде відправлений звіт. Спочатку показується принтер, що використовується за замовчуванням. Можна змінити окремі властивості даного принтера, натиснув на кнопці **Свойства.**
* **Печатать.** Дозволяє визначити число сторінок для друку. Це можна визначити, або вибрав усе, або вказавши сторінки **с** і **по**. Остання опція, **выделенные записи** , дозволяє друкувати конкретні записи, виділенні вами в режимі таблиці.
* **Копии.** Дозволяють визначити, скільки копій звіту потрібно надрукувати. Можна також вибрати опцію **разобрать**, що дозволяє друкувати копії або по екземплярам, або друкувати по декілька копій кожної сторінки.

Ще однією корисною кнопкою на цьому екрані є кнопка **Настройка.** Коли ви натиснете на ній, Access відкриє діалогове вікно **Параметры страницы** (мал. 16), дозволяючи вам змінити розміри полів сторінок звіту.



Малюнок 16

**Огляд інструментів конструктора звітів**

## Access має ряд інструментів для роботи з звітами, які дуже схожі на інструменти конструктора форм. Відкрийте звіт, побудований за допомогою автозвіту. Потім натисніть на кнопці Вид, щоб перейти в режим конструктора звітів.

## Деякі інструменти, які використовуються в режимі конструктора звітів:

## **Панель элементов**. Тут ви знайдете всі елементи керування, які використовуються в звітах.

## **Список полей**. Містить всі поля з джерела записів для звіту.

## **Окно свойств**. Показує всі властивості різних об’єктів, які використовуються в звіті, таких як елементи керування, розділи звіту й сам звіт.

## **Область заголовка отчета**. Використовується при побудові титульної сторінки звіту. Друкується з самого початку звіту.

## **Область верхнего колонтитула**. Тут розміщуються заголовки стовпців й назва звіту, якщо потрібно, щоб він повторювався на кожній сторінці.

## **Область нижнего колонтитула**. Використовується для вказування номерів сторінок, дати та інших приміток. Деяким подобається вказувати внизу кожної сторінки загальну кількість сторінок.

## **Область данных**. Використовується для відображення даних окремих записів з джерела записів для звіту. Це головна частина вашого звіту.

## **Примечание отчета**. Використовується для приміток, інструкцій, виразів, кінцевих значень.

**Покращення звіту**

Ось перелік способів зміни й удосконалення звітів.

* Переміщення надписів з області заголовка звіту в область верхнього колонтитулу й створення кращої назви.
* Зміна властивості **Подпись отчета** на таку, яка краще відображає зміст звіту.
* Видалення із звіту зайвих полів.
* Зміна розмірів тих полів, для яких це необхідно (щоб всі назви, прізвища вміщувались в відповідні поля).
* Додавання графіки до звіту.

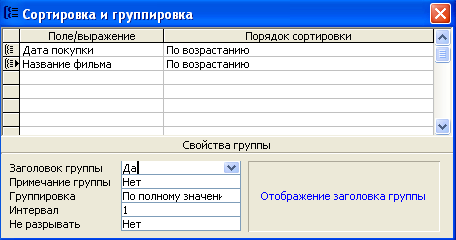
**Зміна висоти розділів звіту й переміщення елементів керування**

Бувають звіти, для яких потрібно щоб назва звіту з’являлась на кожній сторінці. Для цього треба перемістить назву з області заголовка звіту в область верхнього колонтитула. Для цього потрібно спочатку збільшити висоту колонтитула, а потім зсунути униз поля, які вже знаходяться в цьому розділі звіту й звільнити місце для назви звіту.

**Створення груп й сортування**

Access дозволяє вам згруповувати данні в звіт. При цьому ви можете створювати верхні й нижні колонтитули, які відображають сумарні дані по кожній групі.

1. Відкрийте звіт в режимі конструктора.
2. Натисніть на кнопці **Сортировка и Группировка**. З’явиться діалогове вікно **Сортировка и Группировка** .
3. В першому стовпці діалогового вікна надрукуйте ім’я поля з вашої таблиці. Ви одразу ж побачите, що в нижній частині з’являться властивості групи.
4. Встановіть значення **Да** для властивості **Заголовок группы** (мал. 17). В звіті з’явиться новий розділ – область заголовка групи “ім’я вашого поля”.
5. Клацніть на властивості **Примечание группы** й встановіть для нього значення **Да**. Закрийте діалогове вікно.

****

Малюнок 17

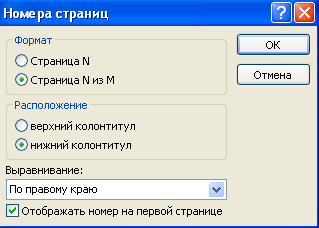
**Створення обчислювальних полів**

Обчислювальне поле складається з функцій Access й полів. Щоб добавити обчислювальне поле, зробіть наступні кроки:

1. Помістіть елемент керування **поле** в область примітки групи.
2. Двічі клацніть на приєднаному до нього підпису й надрукуйте в ньому текст **Число записей**.
3. Щоб встановити властивість **Данные поля**, двічі клацніть в цьому полі й надрукуйте **=Count(\*)**.

Крім функції **Count()**, маються й інші корисні функції. Ви можете, наприклад, використовувати функції Sum(ім’я поля) (дає в результаті суму значень поля) й **Avg**(ім’я поля) (середнє арифметичне значення).

### Додавання в звіт номера сторінки або дата/час.

****

Малюнок18

Щоб додати до звіту номер сторінки, зробіть наступне:

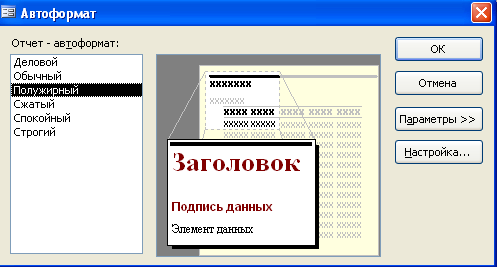
1. В режимі конструктора звіту виберіть **Вставка, Номера страниц** із меню.
2. З’явиться діалогове вікно **Номера страниц** (мал. 18), в якому ви можете вибрати тип й положення номеру на сторінці. Виберіть наступні установки:

* **Формат Страница Ν из М**
* **Расположение нижний колонтитул**
* **Выравнивание По правому краю**

1. Натисніть на кнопці **ОК**. Access розмістить відповідне поле в область нижнього колонтитула у нижньому правому куті звіту.

Те ж саме можна зробити з часом й датою, вибравши з меню **Вставка** команду **Дата и время**.

Обираючи **Формат, Автоформат** з меню, можна легко змінити стиль свого звіту та в подальшому використовувати саме цей новий стиль (мал.19).

****

Малюнок 19

**Контрольні питання до лабораторної роботи 7:**

1. **Що таке звіти? Для чого їх застосовують?**
2. **На основі яких даних створюють звіти?**
3. **Які способи створення звітів існують?**
4. **Який з способів створення найлегший? Чому?**
5. **Перелічіть недоліки та переваги автозвітів?**
6. **Переконайте в зручності майстра звітів?**
7. **Як зробити різними зовнішній вигляд та стилі ваших звітів?**
8. **Що таке наклейки?**
9. **Що виконує майстер діаграм? Як їх застосовують?**
10. **Для чого призначені поштові наклейки?**
11. **Які можливості дає попередній перегляд звіту?**
12. **Перелічіть головні інструменти конструктора звітів?**
13. **Для яких даних призначена область заголовку звіту?**
14. **Для яких даних призначена область верхнього колонтитулу звіту?**
15. **Для яких даних призначена область нижнього колонтитулу звіту?**
16. **Для яких даних призначена область даних звіту?**
17. **Перелічіть засоби, за допомогою яких покращують звіти.**
18. **Як додавати до звіту обчислювальні поля?**
19. **Яким чином можна додати дату і час, номер сторінок у звіт?**

**Використана література**

1. Пасічник В.В., Резніченко В.А. Організація баз даних та знань. – К..:Видавнича група ВХV, 2006, - 348 c.:іл.ISBN 966-552-156-X
2. Гайна Г.А. Г12 Основи проектування баз даних: Навчальний посібник. − К.: КНУБА, 2005. – 204 с. ISBN 966-627-117-6
3. Access 2000 Шаг за шагом издательство Эком Москва, 2002.
4. С.В. Глушаков Д.В. Ломотько Базы данных Учебный курс Харьков «Фолио» Москва «АСТ» 2001.
5. А. Горев, С. Макашарипов, Р. Ахаян. Эффективная работа с СУБД. Питер, 1997
6. І.Т. Зарецька Б.Г. Колодяжний А.М. Гуржій О.Ю Соколов Інформатика, Київ, “Навчальна книга”2002.
7. Боб Виллариал Программирование Access 2002 в примерах перевод с английского Москва 2003. - 496
8. А.Ю. Берко О.Н. Верес Організація баз даних Практичний курс. Навчальний посібник. Львів, 2003.