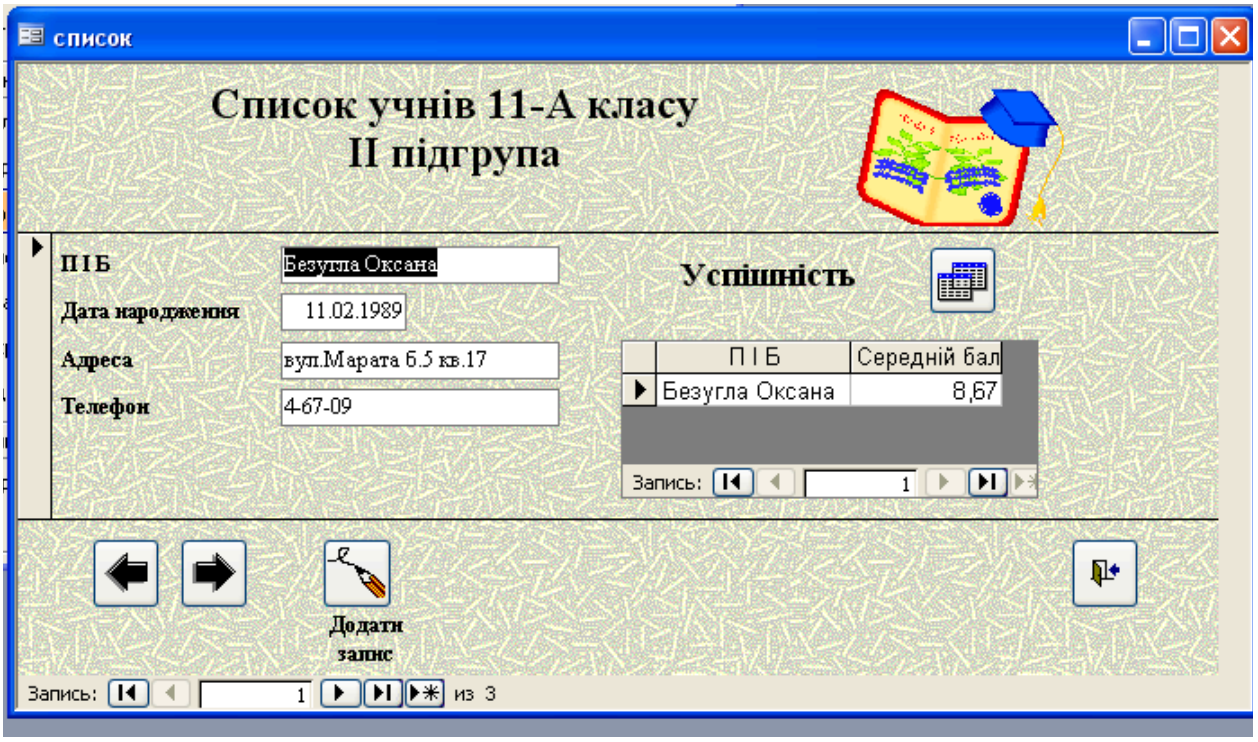
Тема: призначення форм. Сортування, пошук і фільтрація даних

Теоретичні відомості

Є два основні способи подання даних з БД для візуального огляду:

1) у вигляді таблиці;
2) у вигляді форми.

Форма подібна до бланка чи карточки (Рис.8). Прикладами форми є будь-який бланк, сторінки з паспорта, карточка з досьє чи бібліографічна карточка на книжку в бібліотеці.



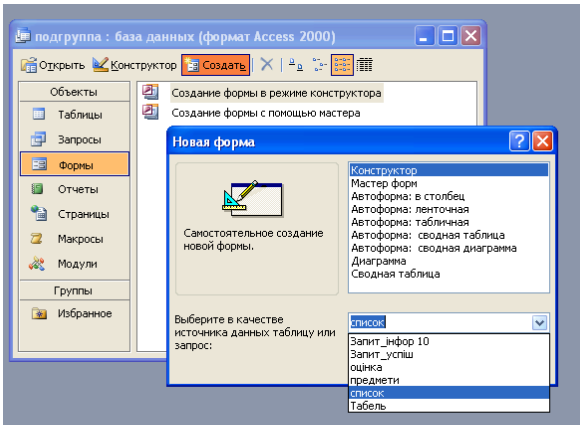
Створюючи форми, слід завжди вибирати джерело даних, тобто таблицю або запит. Це означає, що для проектування баз даних та їх створення спочатку створюється таблиця, а потім форма до неї.

Перехід до форм, окрім кращої візуалізації даних, дає низку додаткових можливостей. Виявляється, що на ***формі можна зручно розташувати:***

- поля типу OLE з картинками, фотографіями тощо;
- елементи керування: кнопки, перемикачі тощо.
- надписи: заголовки форми, рубрик (а також розрисувати форму чи задати фоновий рисунок-заставку: сутінки, глобус, хмари тощо);
- обчислювальні поля (це також елементи керування) для відображення результатів обчислень, виконаних на базі наявних полів;
- закладки (багатосторінкові форми, де поля групують за змістом на різних закладках);
- підпорядковані форми тощо.

Є декілька **способів створення форм**, а саме:

- автоматично за допомогою команди **Автоформа**;
- за допомогою майстра форм;
- вручну за допомогою конструктора форм;
- **комбінованим способом.**



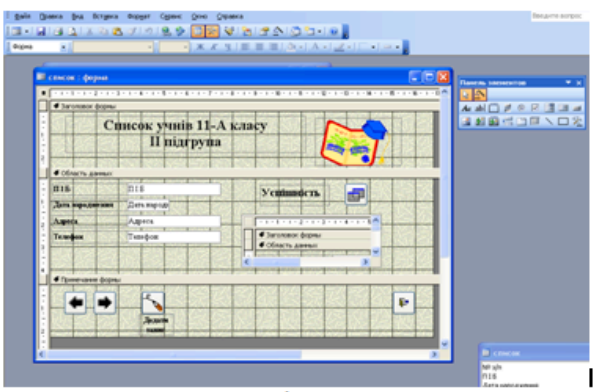


Рис.7. Форма в режимі Конструктора

Приклади Автоформ

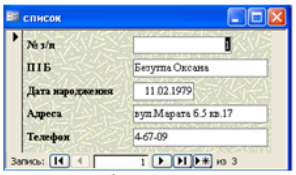


Рис.3. Автоформа: в стовпчик

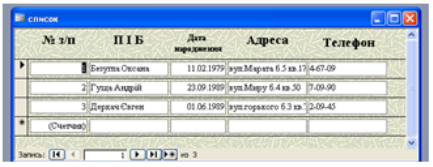


Рис.4. Автоформа: стрічкова

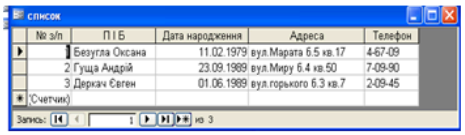


Рис.5. Автоформа: таблична

Рис.6. Автоформа: діаграма

Наприклад, виконавши команду ***Автоформа: У стовпець***, отримаємо форму, в якій усі поля з таблиці будуть розташовані в стовпець і вирівняні до лівого краю.

Використання майстра форм дає змогу швидко відібрати потрібні поля з таблиці для розміщення на формі або розташувати на одній формі поля з різних таблиць. Для створення стандартних баз даних: адресних книжок, телефонних довідників тощо — також є майстри. Достатньо виконати вказівки майстра і структура бази даних готова (часто їх недоліком є надлишкова кількість полів).

Початківцям створювати форму вручну не рекомендують. Спочатку варто створити форму командою ***Автоформа*** або ***Майстер форм***, а пізніше зробити зміни у формі за допомогою конструктора форм. Такий спосіб називається комбінованим. (Додаток 1.)

Розглянемо можливості конструктора форм. Для конструювання форми використовують ***панель елементів керування*** з кнопками (рис.1).

Щоб вставити елемент ня, його треба вибрати на панелі і клацнути на формі у точці вставляння.

За допомогою кнопки **майстра** можна в діалоговому режимі вставити у форму інші елементи: кнопки, перемикачі т

Вибрати об'єкт	Запуск майстра
Вставляти текст	Вставляти поле
Група перемикачів	Вимикач
Радіокнопка	Прапорець
Поле зі списком	Список
Кнопка	Рисунок
Вільна рамка об'єкта OLE	Рамка з посиланням на OLE
Перехід до нової сторінки	Набір закладок
Підпорядкована форма/звіт	Лінія
Прямокутник	Додаткові елементи

Рис.1

Ввести конкретні дані у форму можна *в режимі форми* за допомогою команд з меню ***Записи - Ввести дані*** або кнопки ►*

Щоб добитися естетичного вигляду, об'єкти на формі вирівнюють чи переміщують у режимі конструктора. Для вирівнювання вибраних об'єктів використовують команду **Вирівняти** з меню **Формат**.

Висновки: У діловій сфері часто приходиться працювати з даними з різних джерел, кожне з який пов'язане з певним видом діяльності Для координації всіх цих даних необхідні певні знання й організаційні навички.

Система управління базами даних (СУБД) поєднує відомості з різних джерел в одній реляційній базі даних. Створювані форми, запити і звіти дозволяють швидко й ефективно обновляти дані, отримувати відповіді на питання, здійснювати пошук потрібних даних, аналізувати дані, друкувати звіти, діаграми і поштові наклейки.

У світі існує безліч СУБД. Незважаючи на те, що вони можуть по-різному працювати з різними об'єктами і надають користувачу різні функції й засоби, більшість СУБД спираються на єдиний устояний комплекс основних понять, тому знання придбанні при роботі з даною програмою дають нам можливість працювати з різними СУБД у майбутньому.