

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни
«Системи керування базами даних»
для студентів спеціальності
151 – «Автоматизація та комп'ютерно – інтегровані технології»
всіх форм навчання

2020

Методичні вказівки до лабораторних робіт з дисципліни «Системи керування базами даних» для студентів спеціальності 151 – «Автоматизація та комп'ютерно - інтегровані технології» всіх форм навчання. / Укл. С.І.Арсеньєва. – Запоріжжя: НУ «Запорізька політехніка», 2020. – 108 с.

Укладач :С.І.Арсеньєва, доцент, к.ф-м.н.

Рецензент:В.В. Корольков, доцент, к.т.н.

Відповідальний за випуск: зав. лаб. каф. ЕПА К.І.Пилипенко

Затверджено на засіданні кафедри
“Електропривод та автоматизація
промислових установок”
Протокол № 12 від 12.05.2020 р.
Рекомендовано до
видання НМКЕТФ
протокол № 8
від 18 червня 2020 р.

ЗМІСТ

Передмова	4
Загальні вказівки	5
Завдання до лабораторних робіт	6
Лабораторна робота № 1. Створення структури таблиць	20
Лабораторна робота №2. Введення даних в таблиці та редагування.	34
Лабораторна робота № 3. Встановлення зв'язків між таблицями та нормалізація таблиць у ACCESS.....	44
Лабораторна робота № 4. Створення простих запитів.	51
Лабораторна робота № 5.Створення складних запитів.	65
Лабораторна робота № 6. Створення і використання форм даних.	82
Лабораторна робота №7. Звіти по БД.....	103

ПЕРЕДМОВА

Методичні вказівки складаються з коротких теоретичних матеріалів і завдань із прикладами виконання до лабораторних робіт з дисципліни «Системи керування базами даних». Крім того, наприкінці кожної роботи надані контрольні питання.

Перелік літературних джерел складений з зауваженням бібліотечного фонду університету та науково-методичного забезпечення кафедри.

Для студентів спеціальності 151 – «Автоматизація та комп'ютерно - інтегровані технології» денної і заочної форм навчання.

ЗАГАЛЬНІ ВКАЗІВКИ

Розділ щодо роботи з базами даних дисципліни “ *Системи керування базами даних* ” є основним для освоєння принципів комп'ютерної обробки інформації студентами технічних спеціальностей.

Мета вивчення даного розділу дисципліни - підготовка студентів в області теорії і практики сучасних підходів до проектування, розробки, реалізації і керуванню базами даних (БД).

У результаті вивчення даного розділу студент повинний *знати*:

- сучасний рівень розвитку інформаційних систем;
- основні принципи розробки БД;
- сучасні програмні засоби для створення і керування БД;
- як розробляти інформаційні моделі;
- практично працювати із системами керування базами даних Microsoft ACCESS .

СТРУКТУРА ДИСЦИПЛІНИ

Тема 1. Створення структури таблиць.

Тема 2. Введення даних у таблицю та редагування.

Тема 3. Встановлення зв'язків між таблицями та нормалізація таблиць в ACCESS

Тема 4. Створення простих запитів.

Тема 5. Створення складних запитів.

Тема 6. Створення і використання форм даних.

Тема 7. Звіти по БД.

ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ

1. Створити базу даних зазначеної структури, занести в кожную з таблиць не менш 10 записів, використовуючи хоча б для однієї таблиці форму введення даних.

2. Модифікувати структуру бази даних відповідно до завдання (додати стовпці).

3. Вивести на екран списки з бази відповідно до завдання (у вигляді різних запитів і звітів). Створити самостійно ще 2-3 таблиці, зв'язані з вже наявними відносинами *один-ко-многим* і *многие-ко-многим*.

4. Розробити складну форму з елементами керування мітка, кнопка, прапорець, рамка, підлегла форма й інші – за узгодженням з викладачем.

5. Вставити OLE-об'єкт, створити меню чи головну кнопочкову форму для свого додатка.

Вибрати тематику бази даних можна з наведених нижче варіантів, або переглянути варіанти, що надає Access(пункт Створення бази даних у меню), або робити її за певним замовленням, скорегувавши його з викладачем.

Варіант 1.

Пацієнт: номер пацієнта; ім'я пацієнта; адреса пацієнта; дата операції; діагноз; препарат, призначений після операції; вік.

Стовпці, що додаються: стать пацієнта; дата надходження в клініку.

В таблицю **Пацієнт** додати код хірурга, що оперував, вид страхування. Створити таблицю **Хірург**: код хірурга, ім'я, дата народження, стать, стаж роботи хірургом, категорія, адреса, телефон. Вивести:

– для кожного хірурга список прооперованих пацієнтів із вказівкою діагнозу і віку;

– список пацієнтів клініки з вказівкою прізвища хірурга і його категорій.

Створити таблицю **Відділення**, яку самостійно зв'язати хоча б з однієї з двох попередніх таблиць.

Одержати список пацієнтів (із вказівкою імені хірурга, його категорії і назви відділення), прооперованих 14 травня 2010 року з

діагнозом «виразка шлунка», а також список пацієнтів чоловічої статі з усіх відділень. Одержати також повну інформацію про пацієнта по його імені й адресу пацієнта по його номеру.

Варіант 2.

Хірург: номер хірурга; ім'я хірурга; дата операції; діагноз; номер пацієнта; вік пацієнта.

Стовпці, що додаються: стать хірурга; вік хірурга.

Одержати список хірургів (із вказівкою номера хірурга, імені хірурга і номера пацієнта), що виконали операції 14 березня 2010 р. з діагнозом «апендицит».

Здійснити пошук за критеріями:

- список номерів пацієнтів, прооперованих 14 травня;
- ім'я хірурга по номеру;
- номер пацієнта по імені хірурга.

Створити таблицю **Пацієнт** : номер пацієнта, ім'я, адреса, вік, спосіб страхування. Вивести :

- списки пацієнтів із вказівкою імені, адреси, діагнозу;
- список пацієнтів по датах операції з вказівкою прізвища хірурга.

га.

Додати таблицю з видів страхування та інші, на свій погляд.

Варіант 3.

Навчальний заклад: назва навчального закладу; місто, в якому розташований навчальний заклад, кількість студентів очної форми навчання, кількість студентів заочної форми навчання, кількість викладачів.

Стовпці, що додаються: категорія навчального закладу; міністерство, в веденні якого знаходиться навчальний заклад.

Здійснити пошук за критеріями:

- назви навчальних закладів, розташованих у м. Києві;
- повна інформація про навчальний заклад (за назвою);
- кількість викладачів (по кількості студентів очної форми навчання).

В таблиці **Навчальний заклад** замінити *міністерство* на код чи скорочену назву.

Створити таблицю **Міністерства**: повна назва міністерства, місто, адреса, телефон. Вивести:

– для кожного міністерства список навчальних закладів і їхньої категорії;

– список вузів (по містах) із вказівкою повної назви міністерства, міста і телефону.

Одержати список навчальних закладів (із вказівкою назв), розташованих в Запоріжжі, що мають кількість студентів денної форми навчання більше 1000.

Підрахувати кількість заочників, що навчаються не у вузах Запоріжжя.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 4.

Розклад іспитів: предмет; дата; час; лектор; група; аудиторія; кількість студентів.

Стовпці, що додаються: кафедра, на якій працює лектор.

Здійснити пошук за критеріями:

– повна інформація про іспит (по предмету);

– кількість студентів, що здали іспит (по найменуванню);

– предмет (по даті);

– у які дні й в яких аудиторіях лектор Зайцев А.И. приймає іспити.

В таблиці **Розклад іспитів** замінити поле *кафедра* на код кафедри. Створити таблицю **Кафедра:** код кафедри, назва кафедри, аудиторія, де розташована кафедра, телефон, прізвище завідувача. Вивести:

– для кожної кафедри (у заголовку її повна назва і телефон) список груп, дати іспитів і прізвища викладачів;

– список груп (по даті іспиту) із вказівкою назви предмета і кафедри, на якій працює викладач.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 5.

Студентська група: номер групи; факультет; курс; кількість студентів в групі; прізвище старости.

Стовпці, що додаються: прізвище куратора групи; кафедра, на якій працює куратор групи.

Здійснити пошук за критеріями:

- повна інформація про групу (по номеру);
- номер групи (на прізвище старости);
- курс, номер групи (на прізвище куратора).

Одержати список кураторів (із вказівкою прізвища, кафедри, номера групи і курсу) з різних факультетів.

Створити таблицю **Старости груп**: прізвище старости, адреса, рік народження, стать, телефон. Вивести:

- список старостів всіх груп по факультетах із вказівкою групи, курсу, прізвища старости, адреси, телефону;
- список груп (по курсам) із вказівкою прізвища старости і його адресу.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 6.

Розклад руху потягів: номер потяга; пункт призначення; проміжний пункт; дата відправлення; дата прибуття; кількість годин в шляху; відстань у км.

Стовпці, що додаються: пункт відправлення.

Здійснити пошук за критеріями:

- номер потяга (по пункту відправлення);
- пункт призначення (по номеру потяга);
- дата прибуття (по даті відправлення).

Одержати список міст, в які прибувають потяга, що проходять відстань більш 1000 км.

Створити таблицю **Потяги**: номер потяга, кількість вагонів спальних, купейних, пасажирських, дата останнього техогляду, оцінка стану складу. Вивести:

- список потягів із вказівкою пункту відправлення, пункту призначення, кількості вагонів (всього) і оцінкою стану складу;
- список потягів, що проходять через заданий проміжний пункт із вказівкою кількості вагонів по категоріях.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 7.

Розклад авіарейсів: номер рейсу; пункт відправлення; аеропорт відправлення; пункт призначення; дата відправлення; дата прибуття; відстань км.

Стовпці, що додаються: проміжний пункт посадки.

Здійснити пошук за критеріями:

- повна інформація про рейс (по номеру);
- пункт призначення (по даті відправлення);
- аеропорт призначення (по пункту відправлення).

Одержати список всіх рейсів, пунктом призначення яких є Львів.

Створити таблицю **Авіалайнери:** бортовий номер, тип літака, дата останнього техогляду, оцінка стану, номер рейса, що обслуговує. Вивести:

– список рейсів із вказівкою аеропорту відправлення й аеропорту призначення, типу літаків (їх може бути трохи для даного рейса), оцінка стану літака;

– список літаків, що працюють на трасах довжиною більш ніж 2000 км.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 8.

Театральна афіша: назва театру; дата; час; назва спектаклю; прізвище режисера; автор п'єси; ціна квитка.

Стовпці, що додаються: район міста.

Здійснити пошук за критеріями:

- район міста (за назвою театру);
- повна інформація (по автору п'єси);
- назва спектаклю (по даті).

Одержати список театрів, в яких йдуть спектаклі за драматургією А.П. Чехова.

Створити таблицю **Театри міста:** назва театру, адреса, телефон, прізвище директора, кількість місць у театрі. Вивести:

- для кожного театру (із вказівкою в заголовку адреси, телефону) список спектаклів із прізвищем режисера, автором п'єси;
- список спектаклів (по датах) з назвою театру і його адреси.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 9.

Кіноафіша: назва кінотеатру; дата; час; назва фільму; кіностудія; тип фільму; ціна квитка.

Стовпці, що додаються: район міста; прізвище режисера.

Здійснити пошук за критеріями:

- кіностудія (за назвою фільму);
- назва кінотеатру і назва фільму (по даті);
- прізвище режисера і кіностудія (за назвою фільму).

Одержати список кінотеатрів (із вказівкою назви кінотеатру і району, в якому він розташований), в яких йдуть фільми Одеської кіностудії.

Створити таблицю **Кінотеатри:** назва кінотеатру, адреса, кількість місць, категорія, телефон. Вивести:

- список кінотеатрів по районах міста з зазначенням адреси, телефону і назви фільму;
- список кінофільмів із вказівкою прізвища режисера, кінотеатру, його категорії і телефону.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 10.

Картка книги: номер картки; дата обліку; назва; автор; видавництво; рік випуску; ціна книги.

Стовпці, що додаються: прізвище абонента.

Здійснити пошук за критеріями:

- назва книги й автор (по номеру картки);
- повна інформація (за назвою книги);
- номер картки (по даті обліку).

Одержати список абонентів, що читають твори Шевченка, видані до 1987 року.

Створити таблицю **Абонент:** номер читацького квитка, прізвище, адреса, телефон, місце роботи. Вивести:

- для кожного абонента прізвище, адресу, телефон і список книг, що числяться за абонентом;
- список книг, що знаходяться на руках, із вказівкою прізвища, телефону і місця роботи абонента.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 11.

Випуск партії продукції: підприємство; цех; найменування продукції; кількість одиниць у партії; дата випуску партії продукції.

Стовпці, що додаються: район, в якому розташоване підприємство; прізвище начальника цеху.

Здійснити пошук за критеріями:

- найменування продукції і цех (по даті випуску).
- цех (на прізвище начальника цеху).
- підприємство (по району).

Одержати список найменувань продукції, випущеної в 1 кварталі 2010 року.

Створити таблицю **Підприємство**: назва підприємства, міністерство, адреса, телефон. Вивести:

- список найменувань продукції, підприємства, міністерства;
- для кожного міністерства список підприємств і найменування випущеної продукції.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 12.

Викладач університету: особистий номер, ПІБ, посада, оклад, кафедра, робочий телефон.

Стовпці, що додаються: стаж роботи викладачем, факультет, на якому працює.

Здійснити пошук за критеріями:

- викладачі з окладом більше (менше) заданого;
- доценти (асистенти), їх ПІБ;
- повна інформація з заданої кафедри .

Одержати список кафедр, на яких працюють викладачі зі стажем більш 10 років.

В першій таблиці поле *кафедра* замінити кодом кафедри і видалити поле *робочий телефон*. Створити таблицю **Кафедра**: код кафедри, назва кафедри, аудиторія, телефон, факультет, кількість співробітників. Вивести:

- список викладачів для кожної кафедри з вказівкою посади й окладу;
- список кафедр університету з вказівкою кількості доцентів і асистентів на кожній кафедрі.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 13.

Бібліографічна картка: номер картки; назва книги; автор; видавництво; рік випуску; ціна книги.

Стовпці, що додаються: місто, в якому розташоване видавництво.

Здійснити пошук за критеріями:

- місто і видавництво (за назвою книги);
- назва книги, рік випуску (по номеру картки);
- автор і назва книги (по року випуску).

Одержати список видавництв, розташованих у Москві, Києві, Харкові.

З першої таблиці видалити поле *місто*, в якому розташоване видавництво. Створити таблицю **Видавництво:** назва, місто, адреса, телефон. Вивести:

- по кожному видавництву (в заголовку вказати повну адресу) список книг по року випуску з вказівкою автора і назви;
- список авторів книг із вказівкою назви книги, року видання, видавництва і міста, в якому знаходиться видавництво.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 14.

Щоденник спостереження погоди: дата; час; температура; опади.

Стовпці, що додаються: хмарність.

Здійснити пошук за критеріями:

- повна інформація (по даті);
- дата, час, температура (опади);
- температура (за часом).

Одержати список дат, коли температура повітря перевищувала +30 град. С.

В першу таблицю додати стовпець *код пункту спостереження погоди*. Створити таблицю **Пункти спостереження погоди:** код пункту, географічні координати пункту, область, в якій знаходиться пункт, назва прилеглого населеного пункту. Вивести:

– для кожного пункту спостереження по зростанню дати температури, опади, хмарність;

– щоденник спостереження погоди по датах із вказівкою температури, опадів, області і географічних координат пункту спостереження.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 15.

Картка обліку службовців: ПІБ; номер посвідчення особи; документ, що засвідчує особистість; дата вступу на посаду; дата звільнення з посади; наказ N; посада.

Стовпці, що додаються: вік.

Здійснити пошук за критеріями:

– посада (по прізвищу);

– прізвище, вік (по номеру посвідчення);

– повна інформація (за наказом).

Одержати список службовців, старше 30 років, що працюють на посаді інженера.

Створити таблицю **Службовець:** номер посвідчення особи, відділ, де працює, дата народження, адреса, телефон. Вивести:

– список співробітників, звільнених від посади, із вказівкою відділу, в якому працює, посади й адреси;

– список посад по відділах.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 16.

Зведення про учнів: ПІБ; вік; місце народження; номер школи; клас; район, де розташована школа.

Стовпці, що додаються: стать; дата надходження в школу.

Здійснити пошук за критеріями:

– вік, номер школи, клас (по прізвищу);

– район (по номеру школи);

– повна інформація (за наказом).

Одержати список школи N 24, що навчаються в 10 класі.

Створити таблицю **Школи:** номер школи, адреса, район, прізвище директора, телефон. Вивести:

– список шкіл по районах із вказівкою загальної кількості учнів школи;

– для кожної школи (із вказівкою в заголовку адреси школи і телефону) список всіх учнів по класах.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 17.

Картка учасника змагань: П.І.Б.; вік; дата змагання; вид змагання; показники в змаганні; школа; район.

Стовпці, що додаються: П.І.Б. тренера.

Здійснити пошук за критеріями:

– дата змагання (по виду змагання);

– повна інформація (по прізвищу учасника);

– П.І.Б. учасника, вік (по номеру шкали).

Одержіть список учасників змагань, що прийняли участь у плаванні 15 червня 2010 року.

Створити таблицю **Тренер:** П.І.Б. тренера, назва спортивного гуртка, в якому тренер працює, дата народження, стаж роботи тренером, категорія, вид спорту. Виведіть:

– для кожного тренера (із вказівкою місця роботи і виду спорту) список спортсменів учасників змагань цього тренера;

– список спортсменів-учасників змагань із вказівкою прізвища тренера і спортивного гуртка, в якому тренер працює.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5

Варіант 18.

Інформація про постачання: дата постачання; виріб, що постачається; кількість штук у постачанні; тип перевезення.

Стовпці, що додаються: вартість перевезення; відстань, на яке здійснюється перевезення.

Здійснити пошук за критеріями:

– повна інформація (по поставленому виробу).

– дата постачання, що постачається виріб (по кількості штук у постачанні).

Одержати всю інформацію про постачання, виконану повітряним транспортом на відстань більш 15000 кілометрів.

Видалити з першої таблиці поле *вартість перевезення*.

Створити таблицю **Транспортні перевезення**: тип перевезення, транспорт, вартість тонно-кілометра, накладні витрати. Вивести:

- список поставчань із вказівкою вартості перевезення;
- для кожного виду транспорту список усіх виробів, що поставляються.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5

Варіант 19.

Інформація про проживаючих: вулиця; номер будинку; номер квартири; прізвище, ім'я, по батькові; дата народження; стать; місце роботи; посада.

Стовпці, що додаються: площа квартири; освіта.

Здійснити пошук за критеріями:

- повна інформація (по номеру квартири);
- вулиця, номер будинку, номер квартири (по прізвищу);
- дата народження, стать, місце роботи (за посадою).

Одержати список чоловіків у віці старше 40 років, що мають середню освіту.

Створити таблицю **Житловий фонд**: вулиця, будинок, кількість поверхів, кількість квартир, відрахування на ремонт із кв.м площі квартири, дата останнього ремонту. Вивести:

- списки мешканців по квартирах із вказівкою суми відрахувань на ремонт; порахувати суму по кожному будинку і загальну суму;
- список будинків (відсортовану по вулицях) із вказівкою кількості квартир і загальної суми на ремонт.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5

Варіант 20.

Заявка магазину: код магазину, назва товару, заявлена кількість товару, відпущена кількість товару, дата видачі.

Стовпці, що додаються: одиниця виміру кількості, вартість.

Здійснити пошук за критеріями:

- список отриманих товарів (по даті видачі);
- назви заявлених товарів (по коду магазину);
- список товарів, в яких відпущена кількість менше заявленого.

Одержати список товарів, виданих 14 травня 2010 року.

Створити таблицю *Магазин*: код магазину, назва магазину, адреса магазину, прізвище директора, телефон. Вивести:

- для кожного магазину (у заголовку вказати назву магазину, адресу магазину і телефон) список заявлених товарів і їх вартість;
- по датах видачі список відпущених товарів із вказівкою назви магазину і його адреси.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5

Варіант 21.

Інформація про здані об'єкти: організація-здавач; назва об'єкта; адреса розташування об'єкта (вулиця, номер будинку); дата здачі; оцінка.

Стовпці, що додаються: район.

Здійснити пошук за критеріями:

- організація, назва об'єкта (по вулиці);
- дата здачі, оцінка (за назвою об'єкта);
- назва об'єкта (по організації).

Одержати список об'єктів (із вказівкою назви об'єкта, оцінки), зданих у РСУ N 8 в першому кварталі.

Створити таблицю *Будівельні організації*: назва організації, адреса, телефон. Вивести:

- для кожної будівельної організації список об'єктів з адресою і датою здачі;
- список зданих об'єктів із вказівкою району, оцінки здачі, назви організації-здавача і її адреси.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5

Варіант 22.

Картка обліку товарів: назва; дата; ціна; артикул; колір; розмір. Стовпці, що додаються: обсяг постачання.

Здійснити пошук за критеріями:

- повна інформація (за назвою товару);
- назва, обсяг постачання (за назвою товару);
- назва, розмір (по кольору).

Одержати список товарів, що надійшли в першому півріччі 2001 року, вартістю більш 5 млн. гривень

Створити таблицю **Постачальники**: артикул товару, найменування підприємства, його адреса, телефон, факс. Вивести:

– список товарів, поставлених підприємствами, розташованими в м. Запорожжі;

– найменування підприємств і їх телефонів, що поставили товари за ціною вище заданої.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5

Варіант 23.

Стандартні програми: ім'я програми; ім'я бібліотеки, в якій зберігається; кількість операторів; мова, на якій написана; мова, на якій може бути викликана; призначення; дата створення.

Стовпці, що додаються: автор програми.

Здійснити пошук за критеріями:

– повна інформація (за назвою програми);

– ім'я програми, мова, на якій написана (по імені програми);

– ім'я програми (по кількості операторів).

Одержати список стандартних програм, написаних мовою Асемблер, обсягом більш 150 операторів.

Створити таблицю **Програмісти**: прізвище програміста, відділ, в якому він працює, мови програмування, на яких він працює. Вивести:

– для кожного відділу список програм, створених співробітниками відділу із вказівкою мови програмування, на якій створена програма;

– для кожної бібліотеки програм її склад із вказівкою автора програми і відділу, в якому він працює.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5

Варіант 24.

Платіжна відомість: особистий номер; П.І.Б.; підрозділ; скільки виплачено; дата виплати; тип виплати.

Стовпці, що додаються: посада; стаж роботи на посаді.

Здійснити пошук за критеріями:

- повна інформація (по номеру);
- дата виплати (скільки виплачено (по прізвищу));
- прізвище, стаж роботи (по номеру).

Одержати список співробітників, що одержали премію 15 травня 2010 р.

В першій таблиці замінити поле *підрозділ* на код підрозділу. Створити таблицю **Підрозділ**: код підрозділу, назва підрозділу, начальник, телефон. Вивести:

- список підрозділів із вказівкою повної суми виплат по кожному підрозділі;
- список співробітників по стажу роботи із вказівкою назви підрозділу.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

Варіант 25.

Інвентарна відомість: інвентарний номер; назва устаткування; номер кімнати; відділ; дата одержання.

Стовпці, що додаються: вартість, прізвище матеріально відповідального.

Здійснити пошук за критеріями:

- відділ, номер кімнати (по інвентарному номеру);
- повна інформація (за назвою);
- список устаткування на прізвище матеріально відповідального.

Одержати список матеріальних цінностей, що належать лабораторії системного програмного забезпечення, отриманих у січні 2010 року.

Створити таблицю **Матеріально-відповідальні**: прізвище, відділ, посада, телефон. Вивести:

- список усіх матеріально-відповідальних із вказівкою відділу, посади, телефону і вартості устаткування, що за ним числиться;
- список устаткування, його вартість, прізвище матеріально-відповідального і його телефон.

Додати інші таблиці, на свій погляд, щоб загальна кількість таблиць дорівнювала 5.

НАЙКРАЩИМ ВАРІАНТОМ Є РЕАЛЬНА БД, ЩО СКЛАДАЄТЬСЯ ЗГІДНО ПОТРЕБАМ ПО РОБОТІ АБО ВІДПОВІДАЄ ОСОБИСТИМ ЧИ СІМЕЙНИМ ЗАХОПЛЕННЯМ ТА РОЗВАГАМ.

ЛАБОРАТОРНА РОБОТА№1

СТВОРЕННЯ СТРУКТУРИ ТАБЛИЦЬ

1.1 Мета роботи

Метою роботи є ознайомлення з інтерфейсом пакету MS ACCESS, придбання первісних навичок роботи з ним і створення структури нової бази даних.

1.2 Завдання на лабораторну роботу

1.2.1 Ознайомитися зі змістом пункту 1.3 даних методичних вказівок.

1.2.2 Відповідно до індивідуального завдання сформувати структуру бази даних, побудувати інфологічну модель, обов'язково привести її в звіт.

1.2.3 Різними засобами створити структуру таблиць, використати якомога більше типів полів, форматів та масок, обмежень на їхні значення, умов на значення та повідомлень про помилки.

1.2.4 Для компактного представлення у звіті структури ваших таблиць слід відобразити відповідно до рисунку 1.8, у вигляді таблиці Excel

1.3 Короткі теоретичні відомості про етапи створення нової БД

Щоб створити нову БД, відкрийте Access. Якщо додаток Access вже відкрито, на вкладці **Файл** виберіть пункт **Створити**. Виберіть вільну базу даних або шаблон.

Введіть *ім'я бази даних*, виберіть розташування, а потім натисніть кнопку **Створити**. ACCESS додасть до імені файлу розширення .mdb. Коли база даних відкриється, при необхідності натисніть кнопку **Увімкнути вміст** на жовтій панелі повідомлень. Файл із розширенням .mdb ACCESS використовує для збереження всіх об'єктів БД. *Об'єктами* БД називаються всі блоки, що складають БД. Це - *таблиці, запити, форми, звіти, макроси і модулі*.

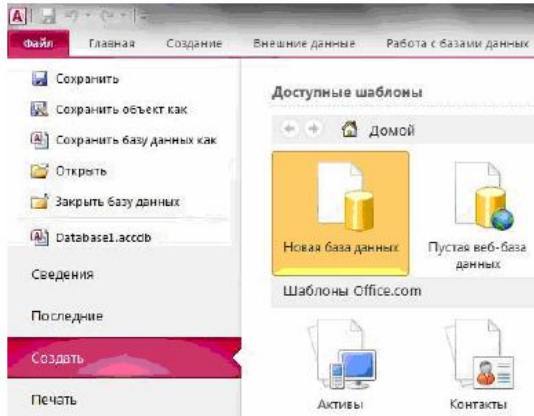


Рисунок 1.1 –Створення файлу бази даних

Можна створити нову БД, клацнувши мишею по кнопці **Створити** БД на панелі інструментів.

Після цього на екрані з'являється вікно БД. При цьому автоматично буде обраний ярличок **Таблиця**. Оскільки ви тільки створюєте нову таблицю, то список таблиць порожній.

Приклад

Створимо БД з ім'ям "MUSICWORLD" («Світ музики») у каталозі, зазначеному викладачем ACCESS.

Перед створенням БД варто обміркувати загальний проект системи, характер вихідних звітів та намалювати відповідну інфологічну модель за допомогою графічних елементів «прямокутник» (таблиця), «овал» (поле), «ромб» (питання, що відповідає зв'язку таблиць).

У БД "Світ музики" будемо зберігати інформацію:

- **про співаків** (прізвище, ім'я, дату і місце народження, музичний інструмент виконавця пісень);
- **про пісні** (назва, тривалість, дату запису, ким виконується і якому альбому належить);
- **про альбоми** (назву, дату видання, країну-видавця, час звучання).

Передбачається створення наступних звітів:

- каталог пісень за абеткою;
- каталог пісень у хронологічному порядку;
- пісні конкретного виконавця;
- склад музичної групи;
- склад альбому.

Ви можете самі спланувати, яку інформацію хочете отримати. Цей етап важливий задля планування полів у таблицях: назви пісень, імена виконавців, дати і таке далі.

Поки що наша БД буде складатися з трьох таблиць: "Співак", "Пісня", "Альбом". Проект БД представлений на рис.1.2. На рисунку показані зв'язки між однойменними полями таблиць.

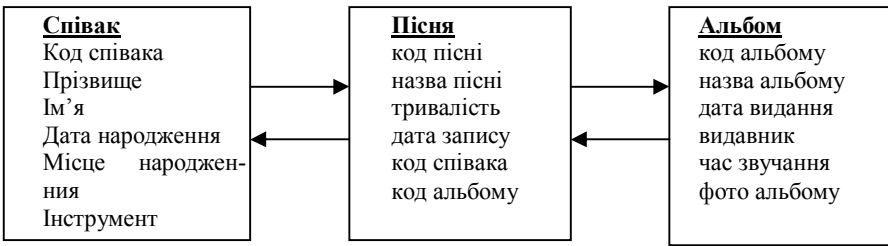


Рисунок 1.2 – Зв'язок між полями таблиць

1.4 Визначення таблиць

Коли вікно БД активне, потрібно спочатку клацнути по вкладці **Створення**, а потім – на кнопку **Таблиця** (для створення таблиці шляхом введення даних), або **Конструктор таблиць** (для створення таблиці в режимі конструктора). Ви повинні використати різні засоби створення таблиць у вашій БД.

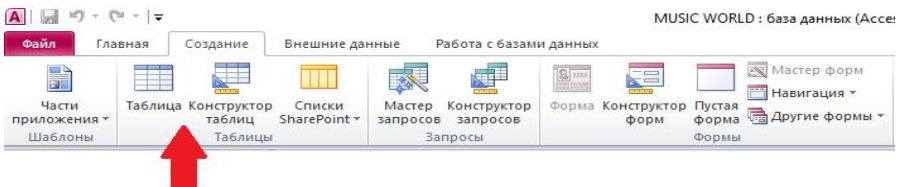


Рисунок 1.3 – Діалогове вікно створення таблиць

Якщо діяти самостійно, то краще клацнути на ярлику **Конструктор таблиць**. На екрані з'явиться вікно порожньої таблиці в режимі Конструктора (рис.1.4). У верхній частині вікна таблиці розташовані стовпці, у які варто ввести ім'я, тип даних і короткий опис кожного поля. Після того, як тип даного буде визначений, можна задати властивості цього поля. У лівому нижньому куті виводиться коротка інформація про поля і властивості, що міняються при переміщенні курсору по вікну таблиці.

У стовпці поля вказуються імена відповідних полів таблиці. Ім'я поля не повинне містити в собі більш 64 знаків. Припустимими є всі спеціальні знаки, крім "." і "!". Неприпустиме повторення імен полів.

Кожному полю зіставляється визначений тип даних. Його можна вибрати зі списку стовпця **Тип даних**, що розкривається.

ACCESS розрізняє наступні типи даних, приведених в таблиці 1.1.

Таблиця 1.1– Основні типи даних у ACCESS.

Тип да-	Опис типу даних:
<i>Текстові</i>	Розмір до 255 знаків, довжина поля визначається в графі <i>Розмір поля</i> , за замовчуванням дорівнює 50 си-
<i>МЕМО</i>	У цих полях може міститися текст, що має максимальну довжину до 64000 знаків. Ці поля не можна індексувати.
<i>Числовий</i>	Тут зберігаються числові дані, формат визначається в графі <i>Розмір поля</i> (1, 2, 4, 8 байтів).
<i>Дата/Час</i>	Завдання дати і часу, у форматі, визначеному в графі <i>Формат поля</i> (8 байтів).
<i>Грошовий</i>	Завдання грошової суми з точністю до 15 цілих і 4 дробових десяткових розрядів (8 байтів).
<i>Логічний</i>	У цих полях зберігаються логічні значення Так/Ні. Ці поля не можна індексувати.
<i>Об'єкт OLE</i>	У цих полях зберігаються об'єкти, що були оформлені за допомогою OLE-сервера в іншому додатку. Максимальний розмір поля 128 Мб; ці поля не можна індексувати.

<i>Лічильник</i>	Унікальне довге ціле, генерується ACCESS при створенні будь-якого нового запису
<i>Майстер підстановок</i>	Відображає дані, що підставляються з іншої таблиці чи списку, що задається користувачем. При цьому запускається майстер підстановок, за допомогою якого визначається тип даних і організується зв'язок з іншою таблицею

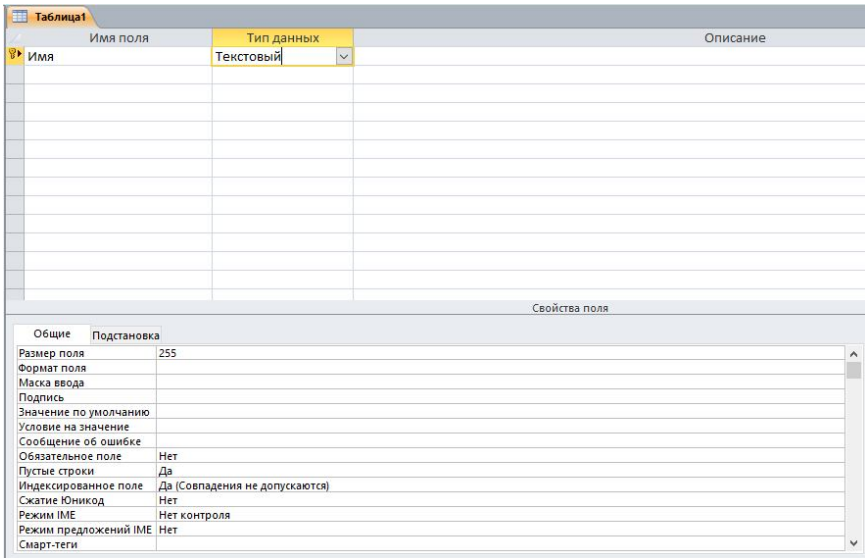


Рисунок 1.4 – Вікно конструктора таблиць.

Обов'язковою умовою нормальної обробки файлу бази даних є однозначна ідентифікація кожного запису даних, для чого кожному з них приписується власний *номер запису*.

Примітка. Номер запису називається також *первинним ключем*. За допомогою ключів встановлюється *відношення (relation)* між двома таблицями

ACCESS може призначати такий номер автоматично, приписуючи кожному знову введеному запису індивідуальний номер. Задати автоматичну нумерацію можна за допомогою вибору в якості типу поля значення *Лічильник*. Якщо при цьому властивість даного поля *Нові значення* приймає значення *Збільшення (Послідовні)*, то ACCESS збільшує вміст поля цього типу на одиницю для кожного нового запису.

За допомогою піктограми ключа (кнопка **Ключове поле**) на стандартній панелі інструментів можна оголосити поле номера запису полем первинного ключа. Таке оголошення автоматично встановлює для параметра *Індексоване поле* значення *Так (Збіги не допускаються)*. Завдяки цьому два різні записи в базі не можуть мати однакові номери. *Рекомендуємо обрати у якості ключового поля тип Текстовий, з мінімальною кількістю потрібних знаків*, відповідно вашої БД (наприклад, якщо кількість записів у таблиці не буде перебільшувати 1000, достатньо обрати формат такого поля 3, тобто максимальне значення 999.Слід пам'ятати, що у випадку Текстового поля над цими цифрами не можна виконувати арифметичні дії, бо вони не являються числами (але це у більшості випадків і не потрібно).

✎ *Примітка.* Якщо вам це потрібно, записи можна прономерувати і самостійно. У цьому випадку варто вибрати для поля номера запису тип *Числовий*, а для параметра *Індексоване поле* – значення *Так (Збіги не допускаються)*.

При введенні даних у таблицю (при заповненні бази) користувач може вносити в це поле будь-які номери. ACCESS контролює дані, що вводяться, і при спробі ввести уже використаний номер запису видасть попереджуваче повідомлення.

●**Увага!*

Варто уважно поставитися до вибору типу поля при специфікації, тому що зміна типу поля у вже заповненій базі даних може привести до втрати даних.

1.5 Обмеження для БД:

- Таблиця може містити не більш 255 полів.
- Таблиця може мати не більш 32 індексів.

– Складений індекс не може включати більш 10 полів; загальна довжина полів індексу - не більш 255 байтів.

– Загальна довжина таблиці, за винятком полів Мемо і об'єктів OLE- не більш 2 Кбайтів.

– Розмір поля Мемо– не більш 64000 байтів.

Розмір поля

Для числових полів можливі фактичні розміри полів, що співпадають з прийнятими у діючій версії Microsoft Office.

1.6 Додаткові властивості полів

У нижній частині вікна Конструктора таблиць на вкладинці Загальні (Общие) (див. рис.1.4), відображається список додаткових властивостей для кожного типу даних, які можна визначити. Це:

- розмір поля;
- формат;
- десяткові розряди;
- вхідний формат;
- стандартне значення;
- умова на значення;
- підпис поля;
- повідомлення про помилки;
- обов'язкове поле;
- індексоване поле.

Розглянемо більш докладно деякі з них.

Розмір поля. Розмір поля встановлюється за замовчуванням (50). Його можна змінити. Так, для більшості прізвищ достатньо 15 букв. Взагалі знайдіть найдовше значення та візьміть його за розмір.

Формат. Формати дозволяють вказувати спосіб відображення на екрані тексту, чисел, значень дат і часу. Одні типи даних мають стандартний формат, інші – тільки визначений користувачем формат, а треті – і те, і інше. Формати впливають тільки на відображення даних, а не на спосіб їхнього введення чи збереження в таблиці.

Для завдання користувальницьких форматів у полях даних типу Текстовий і Мемо використовуються наступні чотири символи (табл. 1.2)

Таблиця 1.2 – Символи завдання користувальницьких форматів

Символ	Опис
@	обов'язковий текстовий символ чи пробіл
&	необов'язковий текстовий символ
>	перетворить усі символи в прописні
<	перетворить усі символи в рядкові

Символи @ і & впливають на окремі символи, а > і < – на усі введені символи. Якщо, наприклад, необхідно, щоб введене ім'я відображалось прописними буквами, то варто ввести значення > для власливості **Формат поля**. При введенні, наприклад, номера телефону можна задати наступний формат (@@@)@@-@@-@@. Якщо потім ввести 612344586, то дані будуть відображені у вигляді (612)34-45-86.

Для типів даних *Числовий*, *Грошовий*, *Лічильник* існує стандартний набір форматів поля:

– **стандартний формат** - встановлюється за замовчуванням.

Відсутні роздільники тисяч і знаки валют, число десяткових знаків залежить від точності даних.

– **грошовий формат** - є присутнім символ валюти і два знаки після коми.

– **з роздільником тисяч** - два знаки після коми і роздільники тисяч.

– **процентний** - процентний стиль.

– **експонентний** - експонентний формат.

Для типів даних *Дата/Час* існує набір форматів поля:

– повний формат дати, встановлюється за замовчуванням – 13:04:96 05:30:20 PM;

– довгий формат дати – Вівторок, 4 березня 1997;

– середній формат дати – 4-бер-97;

– короткий формат дати – 04.03.97;

– довгий формат часу – 05:30:20 PM;

– середній формат часу – 05:30 PM;

– короткий формат часу – 17:30.

Для типу *Логічний* використовується наступний набір форматів:

– так/ні - за замовчуванням;

– істина/неправда;

– вкл/викл.

Десяткові розряди. Для числових і грошових типів можна задати число знаків, виведених після коми. За замовчуванням встановлюється значення Авто, при якому для форматів полів *Грошовий*, *Фіксований*, *З роздільником тисяч*, *Процентний* виводиться два десяткових знаки після коми, а для формату *Стандартний* число виведених знаків визначається поточною точністю числових значень. Можна задати фіксоване число знаків від 1 до 15.

Маска введення. Маска введення задає шаблон, що полегшує введення даних у поле. Символи, що використовуються в масці введення зведені до таблиці 1.3.

Таблиця 1.3 – Символи, що використовуються в масці введення

Символ	Опис
0	Цифра (0-9, обов'язковий символ; знаки плюс (+) і мінус (-) не дозволені)
9	Цифра чи пробіл (0-9, необов'язковий символ; знаки плюс (+) і мінус (-) не дозволені)
#	Цифра, знаки плюс (+), мінус (-) чи пробіл (необов'язковий символ; незаповнені позиції виводяться як пробіли в режимі редагування, але видаляються при збереженні даних)
L	Буква (обов'язковий символ)
?	Буква (необов'язковий символ)
A	Буква чи цифра (обов'язковий символ)
a	Буква чи цифра (необов'язковий символ)
&	Любий символ чи пробіл (обов'язковий символ)
C	Любий символ чи пробіл (необов'язковий символ)
<	Усі введені після нього символи перетворяться в рядкові
>	Усі введені після нього символи перетворяться в прописні
!	Вказує, що маска введення заповнюється спочатку праворуч, потім ліворуч. Варто використовувати, якщо в лівій частині маски знаходяться позиції, заповнювати які необов'язково. Символ можна розміщати в довільній позиції маски введення.

\	Наступний за ним символ відображається в такому ж виді, в якому він був уведений (наприклад, \{ приведе до появи символу {).
., : ; - /	Десятковий роздільник, роздільники груп розрядів, часу чи дати. (Використовувані символи роздільників визначаються у вікні <i>Мова і стандарти</i> панелі керування Windows).

Її можна задати для типів даних *Текстовий*, *Числовий*, *Грошовий*. Це, наприклад: `_. _.` роздільники для поля типу *Дата* (тобто шаблони для введення даних типу дати), чи `##-##-##` - формат телефонного номера.

Створення маски введення допоможе легко вводити інформацію в особливому, заздалегідь визначеному форматі та не втрачати час на введення кожен раз відповідних символів. Якщо, наприклад, номер телефону повинен зберігатися в виді (095) 223-33-22, то задайте маску введення, що додає дужки, пробіл, тире в потрібній позиції, і після цього просто введіть потрібні цифри.

Для цього в режимі конструктора виберіть потрібну назву поля і, клацнувши рядок *Маска введення*, натисніть кнопку, що з'явилася, із багатою кількістю крапок; вона запустить програму - Майстер, що проведе вас по всіх етапах створення маски. Якщо Майстер не завантажений, можете скористатися приведеними тут способами.

Наприклад, для поля *Телефон* можна задати наступну маску введення: `(999)900-00-00.ACCESS` автоматично додасть символ "\" для кожного проміжного символу: `\(999\)900\-00\-00`.

❖* **Увага!** Якщо визначити для тих самих даних *Маску введення* і властивість *Формат поля*, то при відображенні даних у ACCESS властивість *Формат поля* буде мати перевагу. Це означає, що навіть після збереження маски введення разом з даними, вона буде ігноруватися при форматуванні даних.

Підпис поля. Підпис поля - це більш описове ім'я поля, що ACCESS буде виводити в елементах керування; підпис формується й в заголовках звітів.

Стандартне значення. Можна визначити Значення за замовчуванням для всіх типів даних, крім Лічильника, Мемо й об'єкта OLE.

Для Числового типу значення за замовчуванням дорівнює 0, для Текстового і MEMO типів воно дорівнює NULL.

Умова на значення. Можна задати Умову на значення - вираження, що при введенні чи редагуванні значення цього поля завжди повинне бути правдивим.

Повідомлення про помилки. Це - текст, що ACCESS буде виводити на екран, при введенні значення, що не задовольняє умові на значення.

Обов'язкове поле. Воно визначає, чи обов'язково вводити дані у відповідне поле, чи воно може залишитися порожнім. Якщо ви не допускаєте, щоб у цьому полі зберігалось значення NULL, то значенням цієї властивості повинно бути Так.

Порожні рядки означають, що для Текстового і MEMO полів можна дозволити введення порожніх рядків.

Індексоване поле. Воно визначається для полів з типом даних Текстовий, Числовий, Грошовий, Дата/Час і Лічильник, щоб прискорити доступ до даних. Індекс прискорює пошук і сортування даних. Однак відновлення даних при внесенні змін сповільнюється.

Приклад

Створіть таблицю "Співак" зі структурою як на рисунку 1.2.

Ім'я поля	Тип даних	Розмір	Опис
Код співака	Текстовий	4	Код співака
Прізвище	Текстовий	15	Прізвище співака
Ім'я	Текстовий	10	Ім'я співака
Дата народження	Дата/час	8	Дата народження
Місце народження	Текстовий	20	Місце народження
Інструмент	Текстовий	15	Музичний інструмент

Рисунок 1.5 – Структура таблиці “ Співак”

Аналогічно створіть структуру таблиць "Пісня", "Альбом" за допомогою Конструктора. Таблиці можуть мати наступні структури.

Код поля	Тип даних	Розмір	Опис
Код пісні	Текстовий	4	Код пісні
Назва пісні	Текстовий	30	Назва пісні
Тривалість	Дата/час	8	Тривалість пісні
Дата запису	Дата/час	8	Перший запис пісні
Код співака	Текстовий	3	Код співака
Код альбому	Текстовий	3	Код альбому

Рисунок 1.6 – Структура таблиці “Пісня”

Ім'я поля	Типи	Розмір	Опис
	д а н и х		
Код альбому	Текстовий	4	Код альбому
Назва альбому	Текстовий	30	Назва
Дата видання	Дата/час	8	Перше видання альбому
Видавець	Текстовий	10	Країна-видавець альбому
Час звучання	Дата/час	8	Час звучання альбому

Рисунок 1.7 – Структура таблиці “Альбом”

Для компактного представлення у звіті структури ваших таблиць слід відобразити відповідно до рисунку 1.8

Ім'я таблиці-поля	Тип даних	Розмір	Формат	Маска вводу	Значення за замовченням	Умова на значення	Повідомлення про помилку	Обов'язкове поле	Індексоване поле
Співак									
Код співака	текст.	4						Так	
Прізвище	текст.	15						ні	ні
Ім'я	текст.	10						ні	ні
Дата народження	дата/час	8	короткий					ні	так
Місце народження	текст.	20			Київ		>#01.01.1980#	"Старий!"	
Інструмент	текст.	15							
Пісня									
Код пісні	текст.	4						так	так
Назва пісні	текст.	30						ні	ні
Тривалість	дата/час	8	короткий						
Дата запису	дата/час	8	короткий						
Код співака	текст.	4							
Код альбому	текст.	4							
Альбом									
Код альбому	текст.	4							
Назва альбому	текст.	30							
Дата видання	дата/час	8	короткий						
Видавець	текст.	10							
Час звучання	дата/час	8	короткий						
Обкладинка	поле об'єкта OLE								
Коментар	поле MEMO								

Рисунок 1.8 – Структури таблиць з використаними форматами, масками вводу і т.і.

Зверніть увагу, що тут первинні ключі мають 4 символи, зробіть, як вам зручніше.

Створюючи свою БД згідно завдання, слід використати різні засоби створення таблиць, максимальну кількість різних типів полів, для чого штучно придумати дані типу «Сімейний стан», «Стать» тощо, використавши поля типу Логічний або Підстановка, Поле Приміток про використання чогось типу Мемо, логотип товару або фірми, зразок підпису, фото (об'єкт типе OLE)і таке далі.

Контрольні питання

1. Як створюється і навіщо потрібна маска введення?
2. Що є пріоритетним при використанні формату і маски введення?
3. Яким може бути тип поля первинного ключа?
4. Назвіть способи створення таблиць, порівняйте їх.
5. Для чого використовується вкладка **Підстановки**?
6. Перелічіть типи полів і даних ACCESS, їхнє призначення, обмеження, властивості на прикладі ваших таблиць.
7. Якими є стандартні значення полів?
8. Перелічіть основні пункти меню ACCESS, їхнє призначення.

ЛАБОРАТОРНА РОБОТА №2

ВВЕДЕННЯ ДАНИХ В ТАБЛИЦЮ ТА РЕДАГУВАННЯ

2.1 Мета роботи

Придбання навичок введення та редагування даних у таблиці

2.2 Завдання на лабораторну роботу

2.2.1 Ознайомитися зі змістом пункту 2.3 даних методичних вказівок.

2.2.2 Відповідно до індивідуального завдання заповнити таблиці бази даних, створивши форми для введення для кожної таблиці

2.2.3. Змінити структуру бази даних, підготовлену в лабораторній роботі №1(додати поля).

2.2.4 Зробити пошук записів, що задовольняють заданому критерію (наприклад, по місцю народження, інструменту і т.д.).

2.2.5 Зробити сортування і фільтрацію записів за різними критеріями (тривалість запису >...,<...,прізвище П*, дата в діапазоні і т.д.), використати різні типи фільтрації.

2.2.6. Представити у звіті результати своїх дій з відповідними коментарями.

2.3 Короткі теоретичні відомості з введення даних

Щоб ввести дані в таблицю, потрібно клацнути на кнопці **Відкрити** для виділеного об'єкта-вкладки **Таблиці** (рис.2.1).

Вправа

Відкриємо БД MUSICWLD.mdb. Введемо дані в таблицю БД "Співак" як на рисунку 2.3.

2.3.1 Форми для заповнення таблиць (Докладніше форми будуть розглянуті далі).

При введенні даних у таблицю можна використовувати форми.

Форма - спосіб представлення даних, що дозволяє в зручному виді вводити результати, переглядати дані таблиць і запитів. У формі кожне поле запису виводиться у власному текстовому віконці - полі форми, що має назву відповідно імені поля таблиці даних.

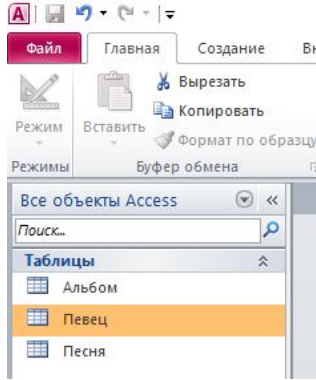


Рисунок 2.1 – Вікно проекту ACCESS

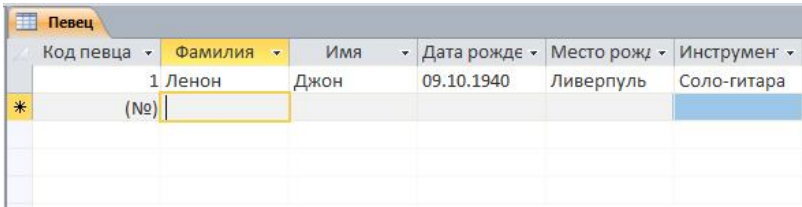


Рисунок 2.2 – Створення таблиці шляхом введення даних.

Код	Прізви- ще, ім 'я	Дата наро- дже ння	Місце на- ро- дже ння	Инстру- мен т
0001	Леннон Джон	09.10.40	Ліверпуль	Соло – гітара
0002	МакКартні Пол	18.06.42	Ліверпуль	Бас – гітара
0003	Харрісон Джордж	25.02.43	Лондон	Ритм – гітара
0004	СтаррРінго	07.07.40	Ліверпуль	Ударні

Рисунок 2.3 – Приклад заповнення таблиці "Співак".

Конструюючи форму, можна використовувати також різні елементи інтерфейсу WINDOWS: прапорці, перемикачі, списки що розкриваються, що буде здійснюватися в подальших лабораторних роботах.

Для створення форми *Програмою-Майстром* потрібно.

1. Відкрити БД, до якої додається форма.
2. Вибрати вкладку **Створення** (рис. 2.4).
3. Клацнути на кнопці **Майстер форм**.
4. У розкритому вікні-списку **Вибір таблиці/запиту** вибрати об'єкт, на основі якого створюється форма.
5. Клацнути на кнопці програми **Майстер**, і тоді всі турботи на себе візьме ACCESS.
6. Вибрати тип **Майстра**. Якщо потрібно створити форму, що виводить на екран щораз *по одному запису* з полями, розташованими в один стовпчик, виберіть тип *В один стовпець*. Якщо ж необхідно вивести на екран *відразу кілька записів* з полями, розташованими одне поруч з іншим, то варто вибрати тип *Таблична форма* і клацнути **ОК**.
7. У наступному діалоговому вікні **Наявні поля** виділити потрібні і перемістити їх у вікно форми в порядку їхньої появи в самій формі. Окремі поля перемістити кнопкою ">", а всі поля відразу кнопкою ">>".
8. Клацнути на кнопці **Далі**, перейти до наступної сторінки, у якій визначений зовнішній вигляд форми.
9. Для вибору виду форми служать перемикачі.
10. Клацнути на кнопці **Далі** для переходу до чергової сторінки, у якій потрібно ввести ім'я форми і вказати режим подальших дій:
 - почати роботу з формою;
 - повернутися до перевизначення її параметрів.
11. Закінчити, натиснувши **Готово**.

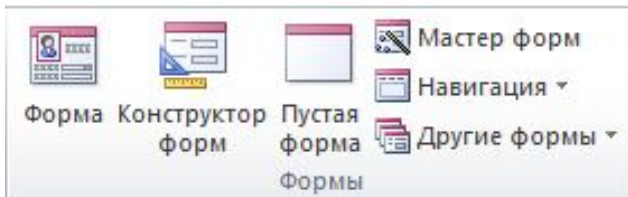
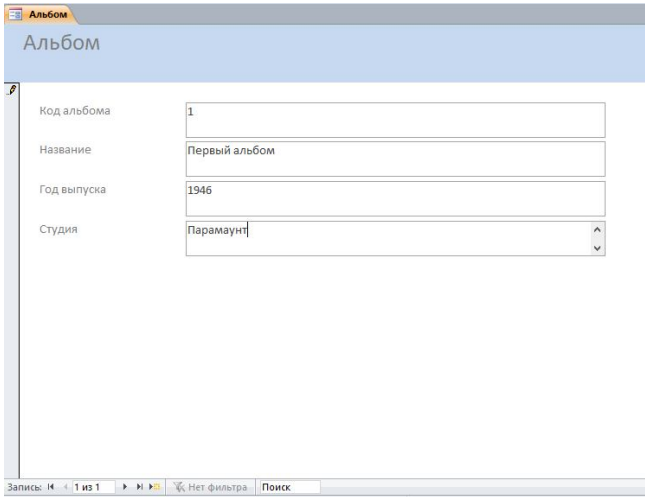


Рисунок 2.4 – Діалогове вікно створення форм

Введення і редагування за допомогою форми.

Спочатку при відкритті форми в ній міститься перший запис таблиці, зв'язаний з даною формою. Але якщо БД ще порожня, то на екрані ми побачимо порожні поля.

Переміщення у формі між полями одного запису реалізується за допомогою клавіш *Tab*, *Shift+Tab*.



Альбом	
Код альбому	1
Название	Первый альбом
Год выпуска	1946
Студия	Пармаунт

Записи: 1 из 1 | Нет фильтра | Поиск

Рисунок 2.5 – Приклад форми для введення даних

Додавання даних у форму.

Щоб додати у форму нові дані, потрібно перейти за допомогою кнопки **>** на останній запис і клацнути **>**. Також можна вибрати з меню **Записи** підменю **Перейти**, а в ньому команду **Нова**.

Вводимо дані в таблиці БД "Альбом", "Пісня", використовуючи форми.

1. Створимо форму "Опис альбому" для таблиці "Альбом". Введемо дані в таблицю "Альбом" з використанням форми "Опис альбому".

Код	Назва альбому	Дата видання	Видавець
1	Вечір важкого дня	26.04.64	США
2	Допоможіть	06.08.65	Англія
3	Білий альбом	25.11.68	Англія
4	Еббіроуд	25.11.68	Англія
5	Нехай так буде	08.05.70	Англія

Рисунок 2.6 – Приклад заповнення таблиці "Альбом".

2. Створимо форму "Опис пісні" для таблиці "Пісня". Введемо дані в таблицю "Пісня" з використанням форми "Опис пісні".

Код пісні	Назва пісні	Тривалість	Дата запису	Код	Код альбому
1	Підемо разом	04:16	29.09.69	1	3
2	Нехай буде так	04:01	08.05.70	1	5
3	Таксист	02:36	05.08.66	3	1
4	Вчора	02:04	06.08.65	2	2
5	Вечір важкого дня	02:31	26.04.64	1	1

Рисунок 2.7 – Приклад заповнення таблиці "Пісня".

2.3.2 Редагування бази даних

СУБД ACCESS надає різноманітні засоби редагування таблиць. Це – редагування *структури таблиці*, редагування *зовнішнього вигляду* таблиці, безпосереднє редагування *записів даних*. Розглянемо спочатку ситуації, коли вимагаються перші два засоби редагування. Вони зв'язані зі зміною проекту БД.

Модифікації проекту БД

Необхідність у переробці проекту БД виникає, якщо:

– Деякі таблиці стають непотрібними;

- З'явилися нові задачі, що вимагають створення не тільки нових таблиць, але і вставки в існуючі таблиці деяких сполучних полів;
- Виявилося, що деякі поля в таблицях використовувалися частіше, тому їх варто розмістити на початку таблиці;
- Деякі дані виявилися непотрібними;
- Потрібно додати деякі поля;
- Необхідно змінити тип даних в деяких полях;
- Необхідно змінити формат для збереження числових полів;
- У ключовому полі з'являються збіги. Варто змінити визначення первинного ключа;
- Потрібно створити додатковий індекс для таблиці.

Додавання і видалення полів.

Щоб вставити ще одне поле в таблицю, потрібно:

- Виділити той рядок таблиці, перед яким необхідно вставити новий рядок. Для цього потрібно клацнути по *маркеру рядка* - сірому прямокутнику в лівого краю;
- Ввійти в меню **Правка**, вибрати команду Вставити чи натиснути клавішу **Insert**. При цьому над виділеним рядком з'явиться новий порожній рядок.

Для видалення одного чи декількох полів потрібно:

- Виділити відповідний рядок чи кілька рядків при натиснутій клавіші Shift. Якщо потрібно видалити не сусідні рядки, варто виділяти їх при натиснутій клавіші Ctrl;
- Ввійшовши в меню **Правка**, вибрати команду **Видалити рядок** чи натиснути клавішу **Delete**.

Виділені поля будуть видалені з таблиці. Операцію видалення можна скасувати на рівні запиту ACCESS про підтвердження дії Видалення.

Примітка. При зміні імені поля дані не конвертуються.

***Увага!**

ACCESS забороняє конвертування будь-яких типів даних у тип *Лічильник*. Цей тип даних використовується й обробляється спеціальним чином.

Проблеми виникають у тому випадку, якщо відбувається перетворення текстового типу даних в числовий. У цьому випадку буде вилучений вміст всіх полів, що містять хоча б одну букву.

При зворотному перетворенні проблем не виникає, оскільки текстові поля можуть містити як цифри, так і букви, а числові поля - вив'язково числа.

У випадку перетворення поля з типом даних *Мето* у текстове поле, дані можуть бути загублені, якщо розмір текстового поля виявиться недостатнім, щоб прийняти весь вміст Мето-поля. У цьому випадку "зайві" символи просто відкидаються.

Додавання записів у таблицю.

Ця операція вже описана, але якщо не вдається додати запис даних, то це може бути викликано якоюсь з наступних причин:

- Дивіться установки параметрів у діалоговому вікні команди **Дозволу** підменю **Захист** меню **Сервіс**;
- Команда **Дозволити зміни** в меню **Запису** є неактивною;
- При відкритті БД у діалоговому вікні **Відкриття БД** був встановлений прапорець **Тільки читання**

Операції пошуку

Пошук інформації здійснюється за допомогою команди **Знайти** меню **Правка** головного меню чи за допомогою кнопки з зображенням бінокля в рядку піктограм. Пошук і заміна інформації здійснюється за допомогою команди **Замінити** меню **Правка**. Коло пошуку інформації можна істотно розширити, використовуючи символи шаблонів "*" і "?". За замовчуванням пошук здійснюється тільки в активному стовпчику (опція **Тільки в поточному полі** включена), що дозволяє швидше одержати результат.

Якщо необхідно зробити пошук по всій таблиці, то опцію **Тільки в поточному полі** варто відключити.

2.3.3 Сортування і фільтрація записів

Почнемо з того, що фільтри вбудовані з метою прискорити пошук і сортування даних.

При пошуку деякого значення ACCESS знаходить і відображає відповідний запис. Але якщо заданому критерію задовольняє кілька записів, то для зручності роботи можна відобразити на екрані тільки їх за допомогою кнопок сортування і фільтрації, що знаходяться на панелі інструментів.

Фільтри дозволяють переглянути тільки окремі записи в формі, звіті, запиті або таблиці або надрукувати деякі записи зі звіту, таблиці або запиту. За допомогою фільтра можна обмежити обсяг даних, що відображаються, не змінюючи макет базових об'єктів.

На панелі інструментів «Сортування і фільтр», знаходиться на вкладці «Головна».

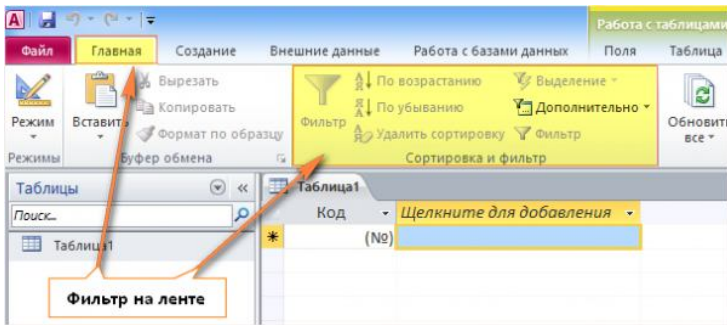


Рисунок 2.8 – Діалогове вікно для роботи з фільтрами

Швидке сортування

Для вибору поля сортування досить помістити в нього курсор на рівні будь-якого запису. Після чого потрібно клацнути на відповідній кнопці, і дані будуть миттєво відсортовані. Для сортування даних по декількох полях необхідно виділити відповідні стовпці. *Якщо стовпці несуміжні, то потрібно їх перемістити, щоб вони розташовувалися поруч.*

Для відновлення первісного порядку розташування записів потрібно використовувати команду **Видалити сортування** в меню **Фільтр**.

Фільтр по виділеному – це спосіб швидкого добору записів по виділеному зразку. Даний фільтр використовується за аналогією з пошуком. Виділяємо слово і застосовуємо фільтр, щоб система керування базами даних сама знайшла записи з цим словом.

Наприклад, виділивши слово «Такое» і застосувавши фільтр, ми отримаємо в результаті два записи, де зустрічається це слово.

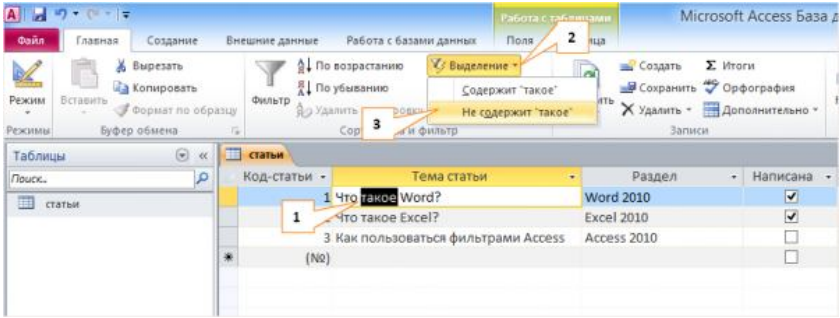


Рисунок 2.9 – Приклад застосування фільтра по виділеному

Функція *Фільтр по виділеному* має властивість нагромадження критеріїв вибору. Це означає, що, продовжуючи вибирати значення за допомогою щикликів на кнопки **Виділення**, можна додавати до фільтра нові критерії.

Розширений фільтр дозволяє відбирати записи за більш складними умовами. Він викликається командою **Головна**→**Додатково**→**Розширений фільтр** (рис.2.10).

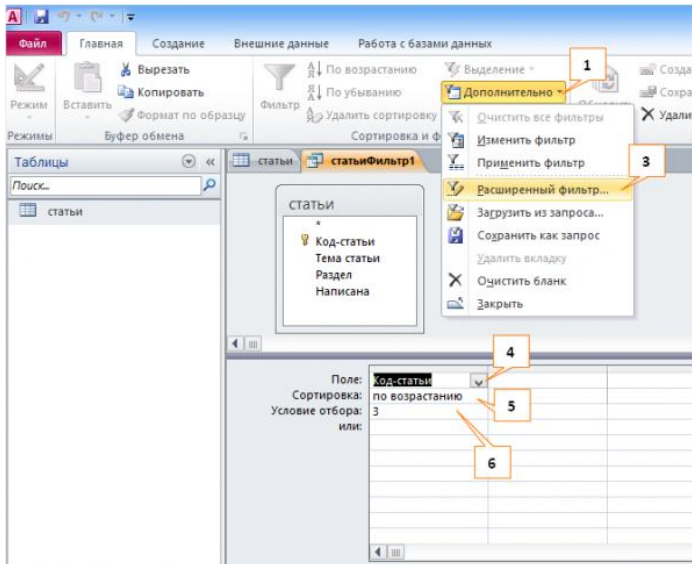


Рисунок 2.10 – Виклик функції Розширений фільтр

На рисунку 2.10 в пункті 4 ми обираємо зі списку полів, потрібне нам поле таблиці. У пункті 5 обираємо спосіб сортування. У пункті 6 обираємо умову відбору, користуючись мовою запитів.

Потім натиснувши «**Застосувати фільтр**» отримуємо з бази даних той запис, в якому значення поля «Код статті» дорівнює 3.

Контрольні питання

1. Чим форми відрізняються від таблиць?
2. Як здійснюється вибір полів з таблиці у форму?
3. Порівняйте способи введення даних у таблиці.
4. Як здійснити пошук потрібної інформації?
5. Як зробити заміну одних об'єктів іншими?
6. Як здійснити видалення записів?
7. Як переставити рядки, стовпці?
8. Як видалити рядок, стовпці?
9. Особливості процедури заміни типу поля.

ЛАБОРАТОРНА РОБОТА №3

ВСТАНОВЛЕННЯ ЗВ'ЯЗКІВ МІЖ ТАБЛИЦЯМИ ТА НОРМАЛІЗАЦІЯ ТАБЛИЦЬ В ACCESS

3.1 Мета роботи

Метою роботи є одержання практичних навичок розробки багато табличних баз даних шляхом встановлення і редагування зв'язків між таблицями, а також вивчення методології нормалізації баз даних.

3.2 Завдання на лабораторну роботу

3.2.1. Ознайомитися з теоретичним матеріалом.

3.2.2. Розробити макет схеми даних, відредагувати його при необхідності, навести його в звіті.

3.2.3. Вивчити можливості реалізації зв'язків $I:I$, $I:M$ и $M:N$.

3.2.4. За допомогою аналізатора таблиць зробити декомпозицію складної таблиці на ряд зв'язаних таблиць.

3.3 Короткі теоретичні відомості

Після того, як ви визначили кілька таблиць, варто повідомити ACCESS, як вони будуть зв'язані одна з одною. Ці зв'язки будуть використані в запитах, формах та звітах.

Для встановлення зв'язків між таблицями необхідно встановити зв'язок між тими полями, у яких міститься загальна інформація. Ці поля можуть мати різні імена, але однакові тип даних і довжину полів, а також (що особливо важливо): інформація в обох полях відповідних записів повинна бути однаковою в обох таблицях. Як правило, зв'язок встановлюється з'єднанням *ключових полів* таблиць: *первинного ключа* однієї таблиці і *зовнішнього ключа* – іншої.

3.3.1 Ключі

Первинний ключ

Кожна таблиця повинна містити *первинний ключ* – одне чи кілька полів, вміст яких унікальний для кожного запису.

Створюючи таблиці, ACCESS завжди пропонує задати первинний ключ з типом даних *Лічильник*, значення якого буде автоматично збільшуватися на одиницю при додаванні нового запису.

Зовнішні ключі. Поняття посилальної цілісності.

🦋 *Примітка.* У ACCESS поле даних типу лічильник не можна використовувати для забезпечення цілісності даних. Тому для первинного ключа важливо задати інший тип даних, наприклад Текстовий чи Числовий.

🦋 *Примітка.* Поля первинного ключа повинні бути як можна більш короткими, оскільки їхні розміри впливають на швидкість виконання операцій у базах даних.

Коли поле однієї таблиці посилається на інше поле (в іншій таблиці), воно називається *зовнішнім ключем (foreignkey)*; поле, на яке він посилається, називається його *батьківським ключем (parentkey)*. Так, у таблиці «Пісня» поле *Код співака* це зовнішній ключ, а таке саме поле у таблиці «Співак» – це батьківський ключ, хоча імена зовнішнього і батьківського ключів можуть бути неоднаковими. Одночасно поле *Код співака* є первинним ключем для таблиці «Співак», для таблиці «Пісня» – первинний ключ *Код пісні*.

Якщо поле є зовнішнім ключем, то воно зв'язано з таблицею, на яку посилається, тобто кожне значення в цьому полі (зовнішньому ключі) безпосередньо зв'язано зі значенням в іншому полі (батьківському ключі). Це значить, що *таблиця знаходиться в стані посилальної цілісності* (російською – *Ссылочная целостность*).

Зовнішній ключ, як і первинний, може бути визначений на будь-якій кількості полів, що усі разом розглядаються як єдине ціле (складений ключ – наприклад, складений ключ з трьох полів-Прізвища, Ім'я та по Батькові). Зовнішній ключ і батьківський, на який він посилається, повинні бути визначені на **однаковій** безлічі полів

Таким чином, *основна ідея посилальної цілісності* полягає в тому, що всі значення зовнішніх ключів відсилають до визначеного рядка батьківського ключа. Це означає, що кожне значення зовнішнього ключа повинне бути представлене в батьківському ключі один і тільки один раз. Як тільки значення розміщається в зовнішньому ключі, батьківський ключ перевіряється для того, щоб переконатися в тім, що таке значення в ньому є присутнім; у протилежному випадку команда відкидається (так, не можна посилатися з таблиці «Пісня» на співака, код якого відсутній у «рідній» таблиці «Співак»).

Поле ключа, первинного чи зовнішнього, перевіряється незалежно від того, додається воно, змінюється чи видаляється. Якщо зміна ключа порушує зв'язок, виходить, вона порушує цілісність даних.

3.3.2 Встановлення зв'язків

Конструктор зв'язків

Конструктор зв'язків викликається командою **Робота з базами даних** → **Схема даних** (рис.3.1).

Спочатку вікно **Схема даних** не містить таблиць. Додавати таблиці у вікно можна такими способами:

- використавши діалогове вікно **Додавання таблиці**, що відкривається автоматично, якщо вікно **Схема даних** для бази — даних відкривається вперше;
- за допомогою кнопки **Додати таблицю** панелі інструментів;
- за допомогою команди **Зв'язки** → **Додати таблицю**;
- у вікні **Схема даних** викликати контекстне меню і вибрати команду **Додати таблицю**.

Після вибору потрібних таблиць треба клацнути на кнопці **Закрити** вікна **Додавання таблиці**. Екран буде виглядати як на рисунку 3.1.

Встановлення зв'язків між таблицями

Для встановлення зв'язків між таблицями потрібно вибрати загальне поле в одній таблиці і перетягнути його мишкою в загальне поле тієї таблиці, яку необхідно зв'язати з першою. При цьому викликається діалогове вікно **Зв'язки**, показане на рисунку 3.2.

У вікні **Зв'язки** міститься інформація про поля зв'язку головної і підлеглої таблиць, параметрах забезпечення цілісності даних і типі встановлюваного зв'язку.

У верхній частині вікна виводяться імена таблиць. Таблиця, що знаходиться ліворуч вважається первинною в цьому відношенні. Нижче містяться поля зв'язку обох таблиць. Після встановлення зв'язку можна включити підтримку цілісності даних.

» На замітку

Цілісність даних – це набір правил, що охороняють зв'язані таб-

лиці від внесення некоректних значень.

Каскадне відновлення зв'язаних полів

Якщо включена опція **Забезпечення цілісності даних** у вікні **Зв'язки**, ACCESS дозволяє включити опцію **Каскадне відновлення зв'язаних полів**. Ця опція дає можливість змінювати вміст поля зв'язку (ключове поле в первинній таблиці, наприклад, Код товару в таблиці Товари на рисунку 3.3).

●* *Увага!*

Якщо опція **Каскадне відновлення зв'язаних полів** не включена, змінити значення ключового поля первинної таблиці не можна.

Каскадне видалення зв'язаних записів

При включеній опції **Забезпечення цілісності даних** у вікні **Зв'язки**, ACCESS дозволяє включити також опцію **Каскадне видалення зв'язаних записів**. Ця опція дає можливість видаляти записи в зв'язаних таблицях, а потім – запис у первинній таблиці.

●* *Увага!*

Будьте обережні з опцією видалення зв'язаних записів! ACCESS не попереджає про каскадне видалення.

Для видалення існуючого зв'язку потрібно зайти у вікно **Схема даних**, виділити зв'язок, що видаляється, і видалити його за допомогою кнопки на панелі інструментів, чи **команди** Видалити в контекстному меню правої кнопки миші.

Остаточний вид схеми бази даних показаний на рисунку 3.3

3.3.3 Використання індексів

Індекс виконує в БД приблизно ту ж функцію, що і картки в бібліотечному каталозі. Індекс БД містить список усіх записів таблиці з вказівкою їхніх адрес. ACCESS виконує пошук по індексу, що знаходиться в оперативній пам'яті комп'ютера. Якщо в таблиці створений первинний ключ, то він і стає індексом.

Вправа

Створимо складний індекс для таблиці "Пісня".

Виберіть таблицю “Пісня” і клацніть кнопкою **Конструктор**. У меню **Конструктор** виберіть команду **Індекси**. Задайте в другій ячейці стовпця *Поле* ім'я "Код альбому". Закрийте діалогове вікно **Індекс**.

У таблиці виберіть поле "Код альбому". Перейдіть у бланку властивостей до поля *Індексоване поле*. Розкривши список властивостей виберіть «Так».

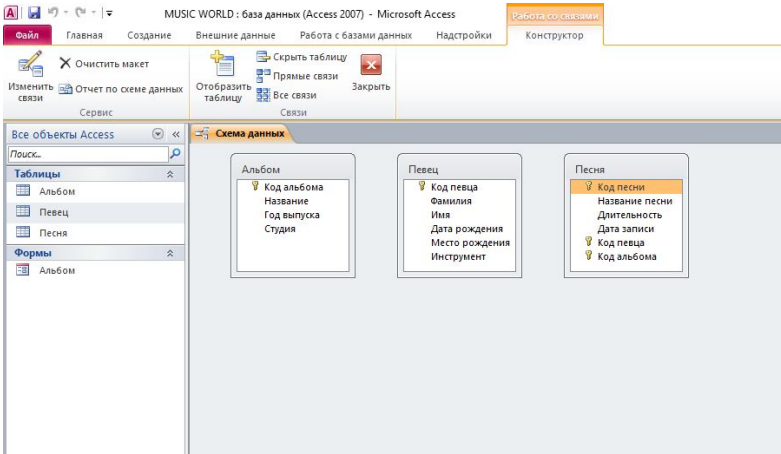


Рисунок 3.1 – Вихідне вікно Схема даних.

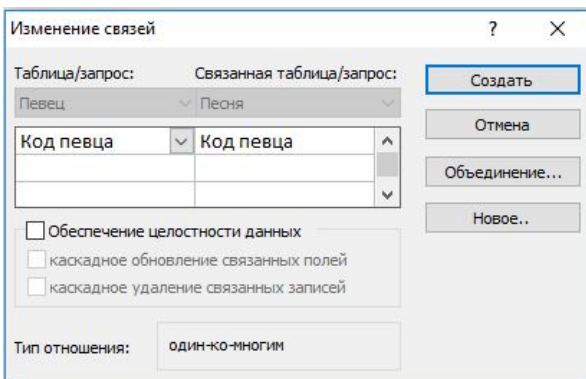


Рисунок 3.2 – Діалогове вікно Зв'язки

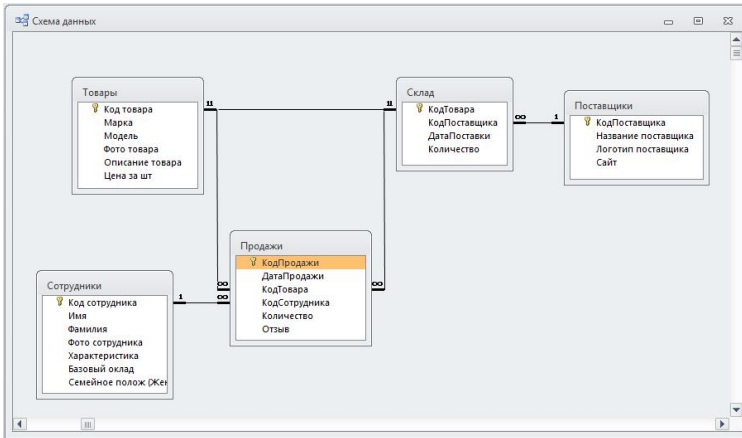


Рисунок 3.3.— Відображення зв'язків між таблицями.

Таким чином, створений складний індекс для таблиці "Пісня". Він буде необхідний при створенні звітів і запитів виду "Пісня з альбому".

3.3.4 Нормалізація БД

Нормалізація є одним з основоположних принципів проектування моделі даних. Це дозволяє усунути надмірності інформації, що зберігається та уникнути аномалій в її організації. Результатом нормалізації є легко підтримувана модель, що не вміщує невизначеностей та повторення даних. Суть її – створення багатьох якомога простих таблиць (що вміщують мало полів) за декількома простими правилами – так званими формами нормалізації.

Існує декілька нормальних форм (НФ). Кожна наступна форма визначає більш сильну ступінь нормалізації, бо вимагає виконання всіх умов, що накладалися у попередніх НФ.

Перша нормальна форма (1НФ). Таблиця знаходиться у 1НФ, коли вона не вміщує полів, що повторюються і складових часток полів (тобто кожне поле вміщує одне значення, а не їхню комбінацію). Якщо значення повторюються, потрібно виділити їх у окремі таблиці з відповідними первинними ключами і встановленням відносин 1:М від

нових таблиць до старих. У не ключових полях не повинні бути переліки значень через кому. У первинних ключах значення унікальні.

Друга нормальна форма (2НФ). Модель даних знаходиться у 2НФ, якщо вона, по-перше, задовольняє вимогам 1НФ, та, по-друге, у таблицях, що мають складений первинний ключ, не ключові поля не залежать від частини первинного ключа, а залежать від усього ключа цілком. Якщо це не так, повинно робити окремі таблиці. Якщо первинний ключ – одне поле, ця форма виконується автоматично.

Третя нормальна форма (3НФ). Модель даних знаходиться у 3НФ, якщо вона, по-перше, задовольняє вимогам 2НФ та, по-друге, якщо у таблицях відсутній взаємозв'язок між не ключовими полями.

Цього можна досягти шляхом виділення у окрему таблицю полів з однією самою залежністю від не ключового поля, використання полів, що визначають цю залежність, у якості первинного ключа нової таблиці та встановлення відносин 1:М від нової таблиці до старої.

Інші форми нормалізації у ACCESS не використовуються.

Якщо ви самостійно при створенні структури таблиць не виконали всі вимоги з нормалізації (першу, другу та третю нормальні форми), вам допоможе в одержанні «правильної» БД аналіз таблиць, що запускається за допомогою команди **Робота з базами даних→ Аналіз таблиць**. Результати відповідності ваших таблиць 1НФ, 2НФ та 3НФ та самі визначення з поясненнями потрібно навести у звіті.

Контрольні питання

1. Що таке первинні і зовнішні ключі?
2. Яким образом встановлюються і видаляються зв'язки між таблицями?
3. Поняття цілісності даних; як це підтримується в ACCESS?
4. Яким чином установлюється Каскадне видалення зв'язаних полів?
5. Як здійснити Каскадне відновлення зв'язаних полів?
6. Для чого використовується функція аналізу таблиць у ACCESS?
7. Що таке індекс? Його призначення і види.
8. Що таке форми нормалізації? Дайте визначення та покажіть, як для ваших таблиць виконуються ці правила.

Лабораторна робота № 4 СТВОРЕННЯ ПРОСТИХ ЗАПИТІВ

4.1 Мета роботи

Метою роботи є ознайомлення з основними прийомами складання і використання запитів.

4.2 Завдання на лабораторну роботу

4.2.1 Ознайомитися зі змістом пункту 4.3 методичних вказівок.

4.2.2 Для бази даних, створеної в попередніх лабораторних роботах, за узгодженням з викладачем розробити різні типи запитів, використовуючи можливі варіанти їхнього створення.

Наприклад:

– побудувати запит по одній таблиці, для чого скласти каталог пісень, у якому пісні розташовані в хронологічному порядку. Вибрати для запиту поля “ Назва пісні ”, “ Тривалість ”, “ Дата запису ” у зростаючому порядку. Для виконання запиту натиснути на панелі інструментів кнопку з зображенням знаку оклику (!). Дати запиту ім'я “ Каталог пісень ”;

– побудувати запит по двох таблицях, наприклад, зробити перелік пісень по альбомах на основі таблиць “ Альбом ” і “ Пісня “. Зв'язок між таблицями установити по полю “ Код пісні “. Назвати запит “ Пісня з альбому “, включити в нього поля “ Назву альбому ”, “ Дата видання ”, “ Назва пісні ”, відсортувавши по назвах альбомів.

Для БД, створених згідно свого варіанту, зробити аналогічні запити по узгодженню з викладачем, у звіті показати відповідний **Конструктор запиту** та результат його виконання.

4.3 Короткі теоретичні відомості

4.3.1 Динамічний набір даних

За допомогою запитів користувач може вибрати з таблиці необхідні дані і представити їх на екрані. Результат обробки запиту являє собою таблицю і називається *динамічним, тимчасовим набором* даних.

При збереженні запиту зберігається тільки *структура запиту* – перелік таблиць, список полів, порядок сортування, обмеження на записи, тип запиту і т.д. *Динамічний набір даних при цьому не зберіга-*

ється. При кожному виконанні запиту він будується знову на основі "свіжих" табличних даних.

4.3.2 Типи запитів

ACCESS підтримує наступні основні типи запитів:

Таблиця 4.1 – Основні типи запитів

Запит на вибірку	Найпоширеніший тип запиту. Витягає дані з однієї чи декількох таблиць і результати відображає в об'єкті в режимі таблиці, у якому допускається зміна записів (при деяких обмеженнях)
Груповий запит	Являє собою спеціальну версію запиту на вибірку. Дозволяє обчислювати суми, підраховувати кількість записів і виконувати розрахунки підсумкових значень. При виборі цього типу запиту ACCESS додає в бланк запиту рядок Групова операція
SQL	Це запит, створюваний за допомогою інструкції SQL. Прикладами SQL-запитів є: <i>запит на об'єднання, запит до сервера, керуючий запит.</i>
Запит на зміну	Дозволяє створювати нові таблиці (команда Створення таблиці) чи змінювати дані в існуючих таблицях (команди Видалення, Відновлення і Додавання). Якщо в процесі виконання запиту на вибірку можна змінити тільки один запис, то запит-дія дозволяє вносити зміни в кілька записів відразу при виконанні однієї операції.
Перехресний запит	Відображає результати статистичних розрахунків (такі, як суми, кількість записів і середнє значення), виконаних за даними з одного поля. Ці результати групуються по двох наборах даних у форматі перехресної таблиці. Перший набір виводиться в стовпці ліворуч і утворює заголовки рядків, а другий виводиться у верхньому рядку й утворює заголовки стовпців.

4.3.3 Створення запитів на вибірку

Запит на вибірку називають також *запитом за зразком (QBE - QueryByExample)*. Він складається шляхом специфікації окремих параметрів у вікні проектування з використанням підказок (зразків).

QBE – запит є стандартним типом запиту. Для його проектування потрібно у вікні бази даних вибрати розділ **Створення**, а потім клацнути по кнопці **Майстер запитів** або **Конструктор запитів**.

Щоб створити запит без допомоги майстра запитів варто вибрати пункт **Конструктор запитів**, після чого ACCESS відобразить порожній проект запиту, а також діалогове багатосторінкове вікно **Додавання таблиці**, за допомогою якого можна вибрати таблиці чи інші запити, що будуть використовуватися при створенні даного запиту.

Для вибору таблиць чи запитів потрібно виділити відповідний елемент і натиснути кнопку **Додати**. По закінченні вибору варто натиснути кнопку **Закрити**.

Вікно Конструктора запитів складається з двох частин (рис.4.1):

- області таблиць запиту;
- бланка запитів.

Область таблиць запиту – це місце, де розміщуються таблиці чи запити і встановлюються зв'язки між ними.

Бланк запиту призначений для визначення полів і умов, що будуть використані для витягнення динамічного набору даних. У кожному стовпці бланка запиту міститься інформація про одне поле з таблиці чи запиту у верхній частині екрана.

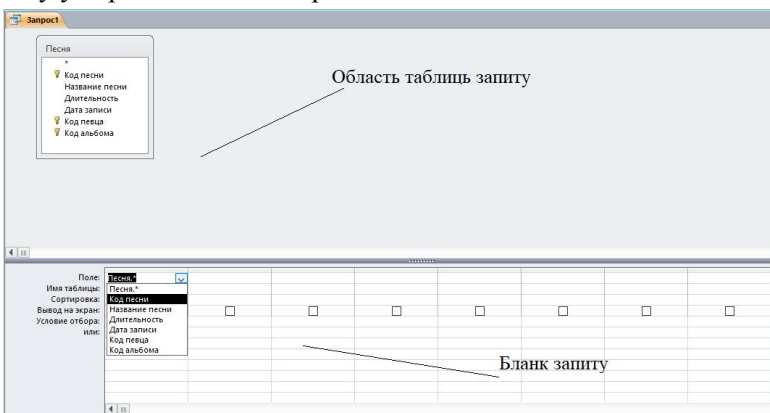


Рисунок 4.1 – Вікно конструктора запитів

У бланку запиту містяться шість рядків:

Таблиця 4.2 – Бланк запиту

Поле	Ім'я поля
Ім'я таблиці	Ім'я таблиці
Сортування	Місце введення інструкцій сортування
Виведення на екран	Визначає, чи буде присутнім поле в динамічному наборі даних
Умова добору	Містить першу умову, що обмежує набір записів
Або	Інші умови обмеження набору записів

Вставка і видалення полів

Ім'я поля вводиться в бланк запиту чи перетаскуванням його з відповідної таблиці, чи зі списку, що розкривається, у потрібному стовпці бланка запиту. Щоб додати в бланк запиту всі поля таблиці, потрібно виділити поле (*) у таблиці, а потім перетягнути піктограму *Поле* в перший осередок бланка запиту.

Для видалення стовпців із бланка запиту можна скористатися командами *Правка* → *Видалити стовпці* чи *Правка* → *Очистити бланк*, а також шляхом виділення стовпця і натискання клавіші .

Для кращого сприйняття результатів виконання запиту можна перейменувати поля в динамічному наборі даних. Для цього в рядку *Поле* бланка запиту потрібно перед старим ім'ям вставити нове ім'я, розділивши їх символом (:).

➤ **Примітка.** Зміна відображуваного імені поля приводить до зміни тільки заголовка цього поля в таблиці. Ім'я поля в базовій таблиці залишається незмінним.

Відображення поля

При виборі поля для нього автоматично включається опція **Виведення на екран**. Скасування виведення здійснюється вручну (рисунок 4.2).

Зміна порядку сортування

Сортування даних по окремих полях може виконуватися як по зростанню, так і по убутанню. Сортування по декількох полях залежить від порядку розміщення полів у бланку запиту (зліва на право).

Примітка. Поля типу *Мето* і *OLE* не сортуються

Добір записів

Умови добору записів – це набір визначених у ACCESS чи задаваних користувачем правил. Умови задаються за допомогою вираження. Вираження може задаватися за зразком чи використовувати складні функції вибору.

Наприклад, можна вибрати список товарів, поставлених у поточному кварталі від постачальників, розташованих у визначеному регіоні.

	Поле виводиться на екран	Поле не виводиться на екран	
Поле:	Название песни	Название Альбом	Фамилия
Имя таблицы:	Песня	Альбом	Певец
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:			
или:			

Рисунок 4.2 – Включення і відключення виводу на екран

Зв'язування таблиць у запиті

При створенні запиту на вибірку з декількох таблиць необхідно враховувати обмеження, що накладаються на можливість редагування полів. В принципі, дані можна змінювати в отриманому після виконання запиту динамічному наборі даних, і зміни ці будуть збережені у використуваних таблицях. Однак якщо встановлений прапорець **Забезпечення цілісності даних** і ключове поле є частиною зв'язку, то редагувати дані в цьому полі не можна (якщо не буде встановлений прапорець **Каскадне відновлення зв'язків**).

Зв'язок між таблицями можна створити наступними трьома способами:

- при створенні макета бази даних.
- вибрати для запиту дві таблиці з однаковими полями одного типу, одне з яких також є **ключовим** повним в одній з цих таблиць.
- створити зв'язок у вікні конструктора запитів.

Перші два способи створюють зв'язок автоматично. Однак іноді необхідно додати в запит незв'язані таблиці, як у наступних прикладах:

- дві таблиці мають загальне поле, назване різними іменами.
- таблиця не зв'язана (і не може бути зв'язана) з іншою таблицею.

Зв'язати дві таблиці, що не були автоматично зв'язані, можна у вікні конструктора запитів. Однак таке об'єднання не приводить до створення постійного зв'язку між цими таблицями. Такий зв'язок буде функціонувати тільки в запиті, над яким ви працюєте.

❖* Увага!

Усі таблиці в запиті повинні бути зв'язані, принаймні, з однією таблицею.

Типи об'єднань між таблицями

Між таблицями і запитом можна установити наступні типи зв'язків:

- один-до-одного;
- один-до-багатьох;
- багато-до-одного;
- багато-до-багатьох.

При завданні зв'язку між таблицями встановлюються тільки правила зв'язування, але не спосіб перегляду даних на основі цього зв'язку.

Для перегляду даних у двох таблицях їх варто об'єднати загальним полем (чи групою полів) в обох таблицях. Такий метод з'єднання таблиць називається **об'єднанням (joining)**. Існують різні типи об'єднань:

- рівні об'єднання (внутрішні об'єднання);
- зовнішні об'єднання;
- самооб'єднання;
- перехресні (декартові) об'єднання.

Внутрішнє об'єднання (рівне об'єднання)

Прийнято в ACCESS за замовчуванням. Означає, що з таблиць вибираються тільки записи, що мають однакове значення в обох таблицях.

ACCESS може також знайти втрачені записи, тобто записи в одній таблиці, що не мають підлеглих записів в іншій таблиці. У цьому випадку при створенні запиту у вікні **Новий запит** варто вибрати опцію **Запису без підлеглих**.

При об'єднанні таблиці здобувають деякі нові риси (чи властивості) зв'язку. **Властивість зв'язку** – це правило, що керує відображенням усіх записів (для заданих полів). і яке впливає на ACCESS при інтерпретації будь-яких виключень (а можливо, і помилок) у таблицях (наприклад, чи повинні відображатися записи без підлеглих записів).

ACCESS дозволяє змінити тип об'єднання шляхом зміни його властивостей. Для зміни властивостей об'єднання варто вибрати лінію зв'язку і двічі клацнути на ній лівою кнопкою миші. На екрані з'явиться діалогове вікно **Параметри об'єднання** (рис.4.3).

У вікні є три перемикачі. Перший варіант звичайно називають **внутрішнім (INNER JOIN** у термінах мови **SQL**) об'єднанням, а два наступних – **зовнішнім**. Ці об'єднання керують роботою ACCESS при побудові динамічного набору даних.

Зовнішні об'єднання відображають усі записи, що мають відповідні один одному дані в обох полях. Вони також показують записи з однієї таблиці, що не мають підлеглих їм даних в іншій таблиці. Існує два види зовнішніх об'єднань: **ліві (LEFT JOIN)** і **праві (RIGHT JOIN)**. Для вказівки виду об'єднання в ACCESS у верхній частині конструктора запитів використовується стрілка.

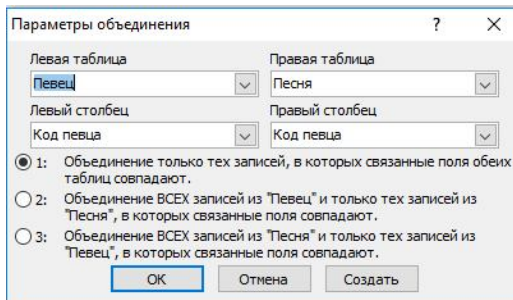


Рисунок 4.3 – Діалогове вікно Параметри об'єднання

4.3.4 Вибір по складних умовах добору записів

Введення умов добору записів по одному полю

Оператор Like і символи підстановки

У поле **Умови добору** бланка запиту можна задавати маску добору за допомогою оператора **Like**, у якому використовуються символи підстановки (Таблиця 4.3 і Таблиця 4.4).

Таблиця 4.3 – Символи підстановки, використовувані оператором Like

Спеціальний символ	Призначення
?	Любий символ (0-9, Aa-Zz, Aa-Яя)
*	Будь-яка кількість символів (0-n)
#	Будь-яка цифра

Таблиця 4.4 – Приклади використання символів підстановки разом з оператором Like

Вираження	У яких полях використане	Результат
Like "34*"	Клієнти. Телефон	Знаходить усі записи про клієнтів, телефони яких починаються на 34.
Like "И*"	Клієнти. Прізвище	Знаходить усі записи про клієнтів, прізвище яких починаються на букву И.
Like "*ПЗУ*"	Товар. Опис	Знаходить усі записи, що містять слово ПЗУ в будь-якій місці поля Опис.
Like "*.01.99"	Замовлення. Дата	Знаходить усі записи за січень 1999 року.

●* **Увага!**

Оператор *Like* і символи підстановки можуть використовуватися тільки для трьох типів полів: текстового, *Мето* і дати. При роботі з іншими типами полів це може привести до помилки.

Завдання незбіжних значень

Для завдання незбіжних значень використовуються оператори *NOT* і оператор "не дорівнює" (<>). Вони записуються перед тим вираженням, у якому записаний тип розбіжності.

Наприклад, *NOT* Ріо-де-Жанейро чи <> фонтан.

Введення умови добору записів для об'єкта OLE

Для об'єкта OLE також можна ввести умови добору, а саме – умови наявності не порожнього запису *IS NOT NULL*. Наприклад, вибрати тільки ті записи, у яких значення в полі малюнка *NOT NULL*.

Введення декількох умов добору записів в одне поле

Для створення складних умов добору записів можна використовувати різні оператори. В основному такі умови добору записів складаються з декількох операторів *AND* і *OR*. Для пошуку записів, що не задовольняють деякому значенню, можна використовувати оператор *NOT* з цим значенням.

При роботі з запитом ім'я поля варто взяти в квадратні дужки ([]), як при створенні полів, що обчислюються, чи вказівці умов добору записів, наприклад, [Дата Візиту]+30.

●* **Увага!**

Якщо при введенні імені поля в умови добору записів не включити його в квадратні дужки, ACCESS автоматично візьме це ім'я в лапки і буде сприймати його, як текст, а не як ім'я поля.

Функції оператора OR

Оператор *OR* можна використовувати в одному осередку поля бланка запити (рисунк 4.4), чи використовуючи *рядок або* (рисунк 4.5).

Поле:	Инструмент	Фамилия	Название песни
Имя таблицы:	Певец	Певец	Песня
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:	"Гитара" Or "Ударные" Or "Скрипка"		

Рисунок 4.4 – Використання оператора OR в умовах добору записів

Поле:	Инструмент	Фамилия	Название песни
Имя таблицы:	Певец	Певец	Песня
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:	"Гитара"		
	"Ударные"		
	"Скрипка"		

Рисунок 4.5 – Використання рядка або (OR) бланка запиту

Крім рядків *Умова добору* і *або (OR)* у ACCESS передбачено ще п'ять додаткових *рядків або (OR)* для введення умов добору записів. При необхідності введення більшої кількості умов, можна використувати оператор *OR* між окремими екземплярами даних.

Використання списку значень і оператора IN

Для створення умов добору для одного поля з використанням декількох екземплярів даних можна скористатися оператором *IN*. При виконанні запиту будуть відображені записи, що містять у зазначеному полі значення, перераховані в списку оператора *IN*. Наприклад, *IN*("гитара";"ударные";"скрипка") для поля *Инструмент*(рис 4.6).

Поле:	Инструмент	Фамилия	Название песни
Имя таблицы:	Певец	Певец	Песня
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:	In ("гитара"; "ударные"; "скрипка")		

Рисунок 4.6 - Використання оператора IN в умовах добору записів

Завдання діапазону значень

Діапазон значень можна задати за допомогою операторів **AND** чи **BETWEEN...AND...**

Оператор **AND** використовується в тих запитах, у яких значення поля повинне одночасно задовольняти декільком умовам.

Наприклад, $>=100$ And $<=200$ чи Between 100 And 200.

Введення умов добору записів по декількох полях

Для використання операторів **AND** і **OR** у запиті по декількох полях варто погоджувати розміщення екземплярів даних (чи масок) в осередках **Умова добору і або (OR)** одного поля з їх розміщенням в іншому полі. Якщо ви хочете зв'язати кілька полів за допомогою оператора **AND**, розміщайте екземпляри даних (чи маски) в одному і тому ж рядку. Для зв'язку умов оператором **OR** їх варто розміщати в різних рядках бланка запиту (рисунок 4.7).

У даному прикладі умова задається формулою:

("Гитара" And >1942 And "Olimpic") Or

("Ударные" And <1970 And "Sunset Sound") Or

("Скрипка" And <1950 And "Record Plant")

Поле:	Инструмент	Дата рождения	Студия
Имя таблицы:	Певец	Певец	Альбом
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:	"Гитара"	> "1942"	"Olympic"
	"Ударные"	< "1970"	"Sunset Sound"
	"Скрипка"	> "1950"	"Record Plant"

Рисунок 4.7 – Приклад використання операторів AND і OR у запиті

4.3.5 Створення в запиті обчислюваних полів

При створенні запиту можна використовувати не тільки поля з таблиць, але і обчислювані поля. Наприклад, можна створити поле Премія, що обчислюється, у якому буде відображатися результат множення значення поля Оклад на значення поля Кв_Премія. Для цього у відповідний осередок бланка запиту потрібно ввести оператор Премія: [Оклад] * [Кв_Премія]. Як мінімум, можна дати таким чином нову назву поля.

🦋 **Примітка.** Ім'я для поля що обчислюється рекомендується задавати по двох основних причинах: по-перше, для заголовка стовпця таблиці запиту, і, по-друге, для звертання до цього поля у формі, звіті чи іншому запиті.

Якщо умови для створення обчислюваного поля більш складні, потрібно використати **Побудовник виразів** (російською **Построитель выражений**), який викликається через праву кнопку миші у пункті **Побудувати...(Построить...)** та вміщує всі таблиці і їхні поля, оператори, функції тощо. Робота з ним очевидна і не потребує додаткових пояснень.

Обчислюване поле можна задати у групових запитах (дивись далі), тоді потрібно обрати пункт **Выражение** у команді **Группировка**.

Приклади:

Довести до відома: Iif([Сума затрат]between 0 and 250; “ Погасити негайно! ”;” Погасити через 20 днів “)

Пенсіонери: Iif (Year(Date()-[Дата народження])>60;” Пенсіонер “;” Не пенсіонер “)

Докладніше можна розібратися з цими питаннями, вивчаючи Help пакету або літературу та через пошук у Інтернеті [1,2].

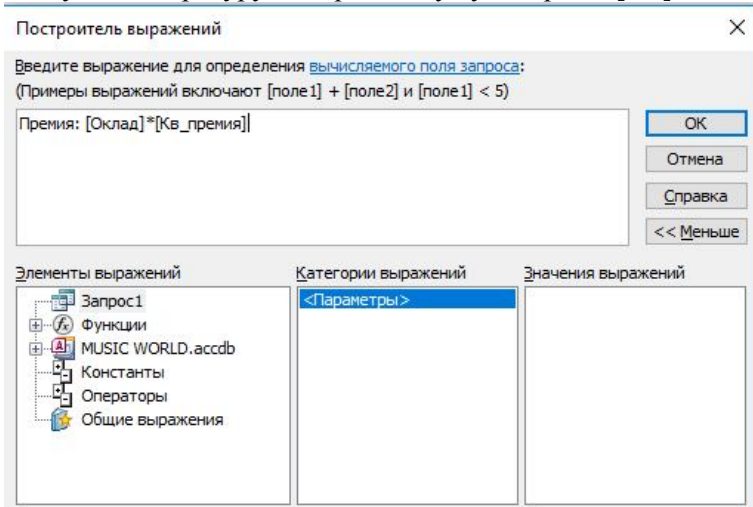


Рисунок 4.8 – Вікно Побудовника виразів

Властивості полів у запиті.

У загальному випадку поля, виведені в наборі записів запиту, успадковують властивості, задані для відповідних полів таблиці. Можна визначити інші значення властивостей.

Щоб задати властивості якогось поля, потрібно клацнути по кожному осередку відповідного стовпця в бланку QBE, а потім - по кнопці Властивості на панелі інструментів чи виконати, команду Вид → Властивості.

Якщо виконати деякі зміни і перейти в режим таблиці, то ці зміни торкнуться і її.

Контрольні питання

1. Що таке "запит"?
2. Які типи запитів підтримує ACCESS?
3. Призначення і склад бланка запитів?
4. Вставка і видалення полів із бланка запиту.
5. Як відсортувати таблицю по окремих полях?
6. Яким образом задаються критерії добору записів?
7. Зв'язування таблиць у запиті?
8. Які існують типи зв'язків між таблицями
9. Які існують типи об'єднань між таблицями?
10. Що називається внутрішнім об'єднанням?
11. Що називається зовнішнім об'єднанням?
12. Використання функцій **OR** і **AND** при завданні умов добору записів?
13. Призначення оператора **Like** ?
14. Призначення оператора **IN**?
15. Завдання діапазону значень?
16. Створення в запиті поля, що обчислюється. Призначення **побудовника виразів**?

Лабораторна робота № 5 СТВОРЕННЯ СКЛАДНИХ ЗАПИТІВ

5.1 Мета роботи

Метою роботи є ознайомлення з основними прийомами складання і використання складних запитів.

5.2 Завдання на лабораторну роботу

5.2.1 Ретельно ознайомитися з теоретичним матеріалом даної теми .

5.2.2 Для бази даних, створеної в попередніх лабораторних роботах, за узгодженням з викладачем розробити різні типи запитів, використовуючи можливі варіанти їхнього створення:

- групові;
- перехресні;
- параметричні;
- запити на зміну.

5.2.3. Оформити звіт з лабораторної роботи, який повинен містити 2-3 групових запити, в тому числі з обчислюваним полем (вираженням), умовами Where та Having, перехресний, з одним та декількома параметрами та один запит на зміну.

Усі запити повинні бути представлені і у Конструкторі запитів, і з результатами виборки. Відповіді на контрольні запитання можна дати не письмово.

5.3 Короткі теоретичні відомості і вказівки по виконанню роботи

До складних запитів звичайно відносять групові, перехресні, запити з параметром та запити на змінення(видалення, додавання, змінення , створення нової таблиці і т.д.).

Створення групових запитів

Для виконання групового запиту необхідно в режимі конструктора запиту активізувати рядок Групова операція одним із двох способів:

- вибрати команду Вид \rightarrow Групові операції \sum
- клацнути на кнопці Групові операції (Σ) панелі інструментів.

У ACCESS можна виконувати групові запити для всіх записів відразу чи для групи записів в одній чи декількох таблицях. Для виконання обчислень необхідно для кожного поля, включеного в запит, вибрати відповідну опцію зі списку, що розкривається, у **рядку Групова операція** (рисунк5.1). Усі 12 функцій цього списку можна розділити на чотири категорії:

- групування;
- вираження;
- умова;
- підсумкові функції.

Групування

Ця категорія складається тільки з функції **Групування** (**GROUPBY** у термінах **SQL**), що служить для вказівки поля, що буде використовуватися в якості об'єднуючого для подальших підсумкових обчислень по кожній групі. Ця функція вибирається за замовчуванням для всіх осередків рядка **Групова операція**.

Вираження

Дана функція вказує ACCESS, що необхідно створити поле, значення якого буде обчислюватися. Наприклад, можна створити запит, що буде, ґрунтуючись на знижці для клієнта, показувати зекономлену їм суму.

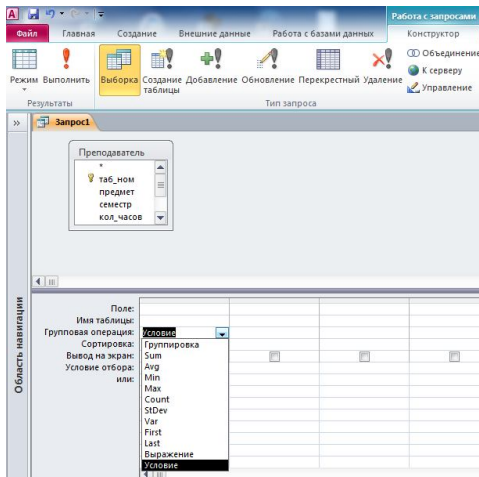


Рисунок 5.1 - Список рядка, де розкривається **Групова операція**

Умова

Дана функція (у термінах *SQL - Where*) повідомляє ACCESS критерії добору полів для обчислень. Наприклад, потрібно створити запит для перерахування всіх типів товарів вартістю нижче 500\$. Оскільки поле вартості не використовується для групування і не може застосовуватися для підсумкових обчислень, користувач визначає параметр умови, повідомляючи ACCESS, що використовувати це поле потрібно як поле обмежуючого критерію перед виконанням обчислень, тобто умова визначає, над якими записами будуть здійснюватися обчислення.

Підсумкові функції

Підсумкова (чи агрегатна) функція означає функцію, що одержує в якості аргументу групи записів і виконує деякі математичні дії над ними (таблиця 5.1).

Таблиця 5.1 – Підсумкові (агрегатні) функції

Функція	Опис	Підтримувані типи полів
Count	Кількість непорожніх полів	AutoNumber, Number, Currency, Date/Time, Yes/No, Text, Memo, OLE
Sum	Сума значень у полі	AutoNumber, Number, Currency, Date/Time, Yes/No
Avg	Середнє значення в полі	AutoNumber, Number, Currency, Date/Time, Yes/No
Max	Максимальне значення в полі	AutoNumber, Number, Currency, Date/Time, Yes/No, Text
Min	Мінімальне значення в полі	AutoNumber, Number, Currency, Date/Time, Yes/No, Text
StDev	Стандартне відхилення значень у полі	AutoNumber, Number, Currency, Date/Time, Yes/No
Var	Дисперсія розподілу значень у полі	AutoNumber, Number, Currency, Date/Time, Yes/No

First	Значення поля в першому записі чи таблиці запиту	AutoNumber, Number, Currency, Date/Time, Yes/No, Text, Memo, OLE
Last	Значення поля в останньому записі чи таблиці запиту	AutoNumber, Number, Currency, Date/Time, Yes/No, Text, Memo, OLE

Виконання групових операцій для групи записів

При необхідності одержання інформації з окремих груп записів потрібно згрупувати записи по відповідному полю, а потім задати тип потрібної агрегатної функції.

ACCESS дозволяє виконувати групові операції для багатьох полів і багатьох таблиць так само, як і для одного поля однієї таблиці. Наприклад, можна згрупувати записи по постачальниках і по типу товару для визначення суми поставлених товарів кожного типу для окремих постачальників.

Визначення критерію для групового запиту

Крім групування записів для запитів, можна вказати критерій (умова добору) для обмеження кількості записів, що будуть розглядатися при обчисленнях і відображенні результату. Іншими словами, можна задати умову вибірки записів **до** обчислень і **після** обчислень.

У першому випадку умова задається в стовпці, у якому в рядку **Групова операція** стоїть опція **Умова** (рис. 5.2).

◆*Увага!

Якщо в рядку **Групова операція** ви визначаєте опцію Умова, то побачити це поле не зможете: ACCESS використовує його винятково для перевірки умови до обчислень. При необхідності висновку поля на екран його необхідно продублювати у бланку з опцією **Групування**.

При необхідності фільтрації підсумкових даних критерії добору задаються у відповідних стовпцях бланка запиту, у яких задані опції Групування чи конкретні підсумкові функції (рисунок 5.3). У термінах **SQL** така умова задається оператором *Having*.

Запрос3 - Microsoft Access

Результаты

Тип запроса

Область навигации

Поле:	Ванития	Цена за шт	Цена за шт	Цена за шт	Цена за шт	Количество	Количество
Имя таблицы:	Сотрудники	Товары	Товары	Товары	Товары	Продажи	Продажи
Групповая операция:	Группировка	Sum	Max	Min	Avg	Count	Условие
Сортировка:							
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:							
или:							-6

Рисунок 5.2 – Визначення критерію добору даних до обчислень

Отбор по нескольким параметрам - Microsoft Access

Результаты

Тип запроса

Область навигации

Поле:	Код товара	Марка	Описание товара	Цена за шт	Количество	Название поставщи	Сайт
Имя таблицы:	Товары	Товары	Товары	Товары	Склад	Поставщики	Поставщики
Сортировка:							
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			Like *IPS*Qualcomm	<7000			
или:							

Рисунок 5.3 – Визначення критерію добору даних після обчислення

☞ *На замітку*

Якщо в запиті використовуються агрегуючі функції, то оператор **Where** виконується перед ними, тоді як оператор **Having** застосовується після розбивки на групи й обчислення значень функцій для кожної групи (наприклад, виводить тільки ті дані після агрегації, у яких значення більше заданого).

●* *Увага!*

Результати виконання операторів **Where** і **Having** можуть відрізнитися. Це зв'язано з тим, що **Where** відсіває рядки перед групуванням даних, а **Having** – після.

Створення вираження для групових операцій

У ACCESS користувач має можливість створювати власні вираження. Для цього в порожньому осередку рядка **Поле** потрібно записати вираження, а в рядку **Групова операція** вибрати опцію **Вираження**. На рисунку 5.4 показаний приклад складання запиту, що визначає уцінку товару на складі (ASUS) як 20% від добутку ціни товару на кількість. При цьому такі обчислення здійснюються тільки для тих товарів, що надійшли на склад до 30.03.2018.

5.4 Створення перехресних запитів

Для відображення вибірових даних у компактному й наочному виді ACCESS може формувати спеціальний тип запиту – перехресний.

Перехресний запит – це вибірка даних, записаних у вигляді двовимірної матриці, що створена з таблиці ACCESS. Цей запит представляє визначені дані з обраних полів у форматі, схожому на формат електронної таблиці Excel.

Для створення перехресного запиту необхідно вказати, як мінімум, три параметри:

- поле заголовків рядків;
- поле заголовків стовпців;
- поле для вибору значень.

Дані параметри вказуються в призначеному для цього рядку

Допустимо, необхідно створити запит, що відображав би в результатуючій таблиці марки товарів як заголовки рядків, назви клієнтів як заголовки стовпців, а в кожному осередку таблиці містилися би дані про кількість товару у кожного клієнта. На рисунку 5.5 приведений

приклад завдання параметрів такого запиту, а результат виконання запиту показаний на рисунку 5.6.

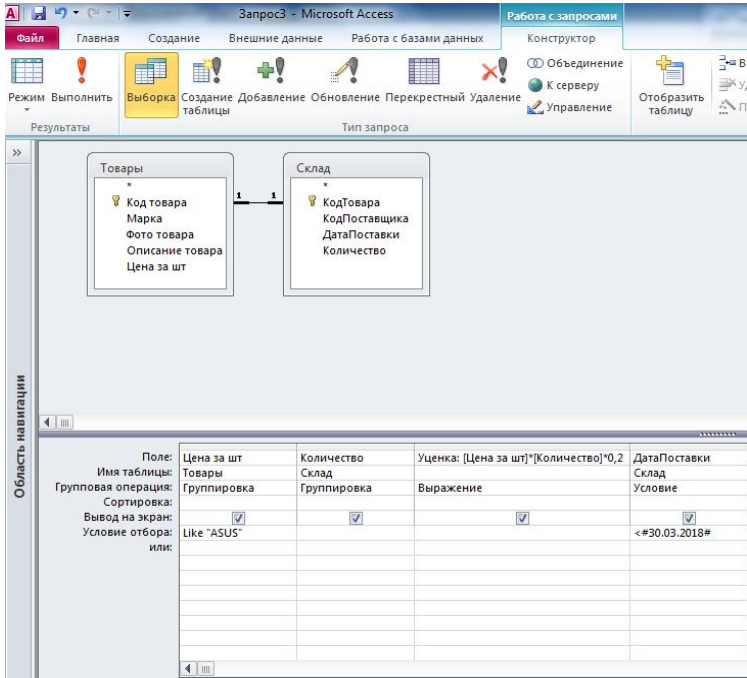


Рисунок 5.4 – Запит з використанням вираження в груповій операції

Майстер перехресного запиту

Майстер перехресного запиту викликається у вікні **Новий запит** і дозволяє створити перехресний запит на підставі отриманої інформації про:

- вихідну таблицю чи ім'я запиту;
- поля для заголовків рядків;
- поля для заголовків стовпців;
- поля для значень;
- заголовок.

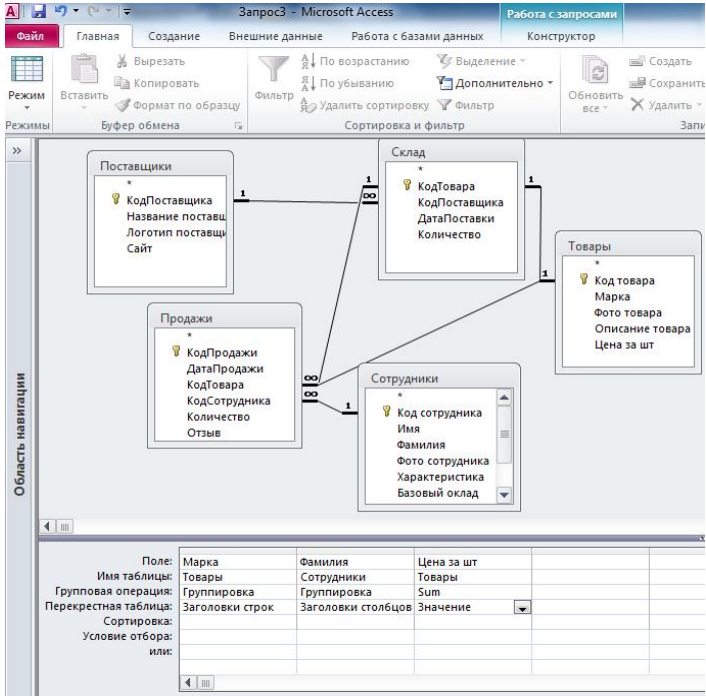


Рисунок 5.5 – Створення перехресного запиту

5.5 Створення параметричних запитів

Параметричний запит – це запит, при виконанні якого користувачу пропонується ввести значення якогось параметра, з обліком якого і виконується запит. Іншими словами, це завдання одного чи декількох критеріїв добору, але не в режимі конструктора таблиць, а *динамічно* в процесі виконання запиту.

Наприклад, створений запит, що відображає інформацію про товари, що є на складі. Якщо цей запит використовується часто, то можна створити параметричний запит, що запитував би користувача про назву товару. Для створення такого запиту необхідно в наявний запит на вибірку в стовпці *Назва товару* (наприклад) в осередок *Умова добору* ввести потрібну фразу. Наприклад, *[Введіть назву товару]*.

Марка	Константин	Любомирог	Фидотов
ASUS VivoBook			13000
Asus ZenFone 3 Max	7000		
Asus Zenfone 4		14000	
Asus ZenFone Go	3700		
Dell Inspiron 3552	6800		
Dell Inspiron 3576		17700	
Dell Vostro 3568	46000		
HTC 10 Lifestyle		14000	14000
HTC Desire 630 Dual Sim	3600		
HTC Desire 830 Dual Sim		6600	
Lenovo IdeaPad 320		7300	
Lenovo IdeaPad 330	10500		
Lenovo IdeaPad 720S		37000	
Meizu M5			3500
Meizu M5 Note 32GB	4400		
Meizu M5C			2700
Samsung Galaxy A5 2017			12000
Samsung Galaxy J5 (2016)	5000		5000
Samsung Galaxy S9		29000	29000
Samsung UE75NU			89000
Sony KDL			15315
Sony VAIO Pro	13000		
Sony Xperia L1 Dual	5000		5000
Sony Xperia X Compact		17200	
Sony Xperia XZ Premium			20600
Xiaomi Mi Gaming			34100

Рисунок 5.6 – Результат виконання перехресного запиту

Порада

Ви можете створити параметричний запит з використанням будь-якого припустимого оператора, включаючи оператор *Like* із шаблонами. Наприклад, параметр *Like [ввести назву товару]&** дозволяє користувачу запустити запит для одного товару чи для всіх (*) (див. Лабораторну роботу №1).

Аналогічно створюються багато параметричні запити. При цьому варто мати на увазі, що вводити потрібно коректні значення, інакше в результуючій таблиці не виявиться ні одного запису.

За замовчуванням параметричний запит виконується зліва направо. Однак цей порядок можна змінити в діалоговому вікні *Параметри запиту*, що викликається командою *Запит* → *Параметри* (рис. 5.7).

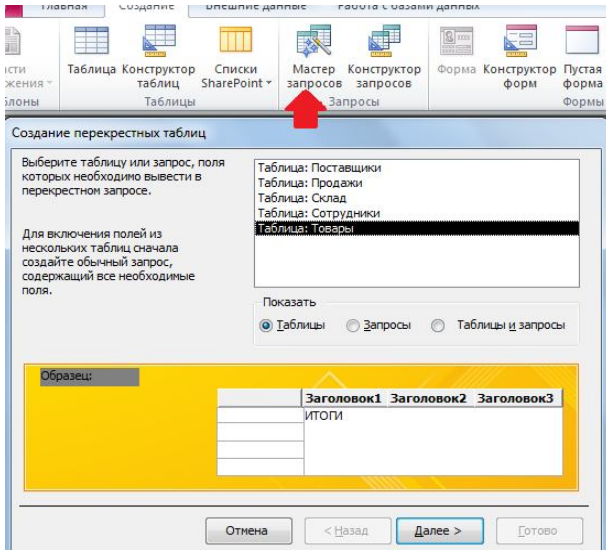


Рисунок 5.7 – Діалогове вікно параметрів запиту

❖*Увага!

Тексти запитів введення повинні збігатися з аналогічними в рядку *Умова добору* у вікні конструктора запитів, інакше запит може не працювати.

5.6 Створення запитів на зміну

Запити на зміну (*action query*) дозволяють таким чином змінювати значення полів записів:

- видалити запис з таблиці чи групи таблиць;
- додати запис з однієї таблиці в іншу;
- оновити інформацію в групі записів;
- створити нові таблиці.

Порада

Завжди зберігайте резервну копію даних перед виконанням таких запитів

Для запуску запиту на зміну використовується кнопка **Запуск** панелі інструментів (на ній зображений знак оклику). У вікні бази даних ці запити позначені знаком оклику.

Створення запитів на зміну схоже на створення запитів на вибірку – точно так само необхідно вибрати поля для запиту і вказати умови добору. Крім того, необхідно вказати визначені властивості цих запитів – **Додавання, Створення таблиці, Відновлення** чи **Видалення**.

Створення запитів на відновлення

Для створення запиту на відновлення виконаєте такі дії.

1. Створіть запит на вибірку. Перегляньте необхідні дані в режимі таблиці.

2. Перетворіть запит на вибірку в запит на відновлення. Для цього потрібно вибрати пункт меню Запит → Відновлення. При цьому ACCESS додасть рядок властивостей Відновлення у вікні конструктора запитів. У цьому рядку потрібно вказати необхідні зміни.

Примітка. Зміни можна задавати для декількох полів відразу, включаючи поля, використані для завдання умов добору.

Побудова запиту на створення таблиці

Запит на зміну можна застосувати для створення нових таблиць (на основі наявних), використовуючи умову добору записів. Для цього необхідно створити запит **на створення таблиці**, виконавши наступні дії:

- створіть новий запит, використовуючи необхідні таблиці;
- виберіть пункт **Запит → Створення таблиці** та у діалоговому вікні, що з'явилося, задайте назву сформованої таблиці;
- виберіть необхідні поля з наявних таблиць;
- вкажіть умови добору;
- перейдіть у режим таблиці і перевірте правильність добору записів;
- запустіть запит на виконання (кнопка Запуск чи пункт **Запит → Запуск**).

☞ На замітку

При створенні таблиці за допомогою запиту на створення формати даних і розміри полів зберігаються такими ж, якими вони були в базовій таблиці запиту. Однак ніякі інші властивості полів і таблиці не успадковуються! Для завдання ключового поля чи інших властивостей варто відредагувати макет нової таблиці.

Побудова запиту на додавання записів

Даний тип запиту дозволяє додавати записи в зазначену (обов'язково існуючу) таблицю, причому як у таблицю в поточній базі даних, так і в будь-яку іншу базу даних ACCESS. Використання такого запиту можливо, наприклад, коли необхідно з наявної базової таблиці перенести інформацію в архівну з наступним їхнім видаленням з базової таблиці. (наприклад, оновити список співробітників фірми).

При роботі з запитом на додавання необхідно пам'ятати наступні моменти.

1. Якщо таблиця, у яку додаються записи, має ключове поле, то записи, що додаються, не можуть мати значення Null чи повторювані значення в цьому полі.

2. При додаванні записів до таблиці, розташованої в іншій базі даних, необхідно знати місце розташування і назву цієї бази даних.

3. Якщо ви використовуєте символ * у рядку конструктора запитів, використовувати окремі поля з цієї таблиці не можна – ACCESS сприйме це як спробу додавання одного і того ж запису двічі і блокує її.

Побудова запиту на додавання аналогічна побудові запиту на створення таблиці – спочатку необхідно створити запит на вибірку і переконатися, що вибираються потрібні записи, а потім перетворити його в запит на додавання. Після виконання запиту на додавання необхідно відкрити таблицю (таблиці), у яку додані записи, і перевірити результат.

Побудова запиту на видалення записів

З усіх запитів на зміну запит на видалення найбільш небезпечний. На відміну від інших запитів він видаляє записи раз і назавжди. Запит на видалення може одночасно видаляти записи з декількох таб-

лиць. Однак для видалення зв'язаних записів з декількох таблиць необхідно виконати наступні дії:

- визначити зв'язки між таблицями у вікні *Схема даних*;
- установити опцію *Забезпечення цілісності даних*;
- установити опцію *Каскадне видалення зв'язаних записів*.

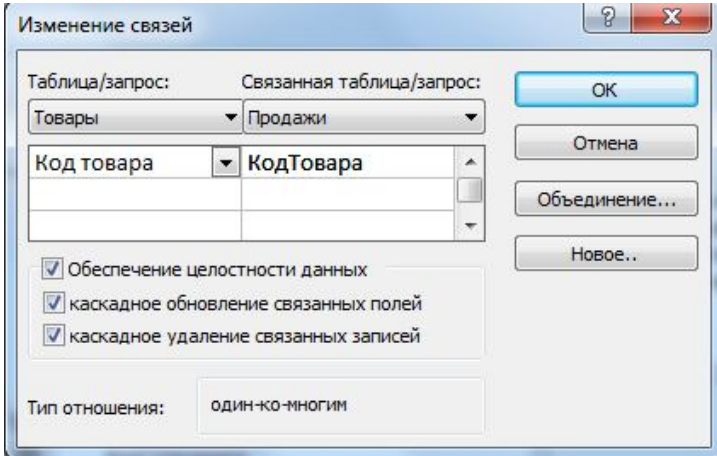


Рисунок 5.8 – Діалогове вікно Зв'язки

На рисунку 5.8 приведене діалогове вікно *Зв'язки*. Зверніть увагу на встановлені опції!

При роботі зі зв'язками *один-до-багатьох* без завдання зв'язків і без включення параметра *Каскадне видалення зв'язаних записів* ACCESS видалить записи тільки з однієї таблиці. Порядок такий, що ACCESS спочатку видаляє записи з боку *багато* цих зв'язків, а потім вам належить видалити записи з боку *один*.

❖*Увага!

Оскільки запит на видалення робить необоротне руйнування на таблиці, перед початком його роботи завжди варто виконувати архівне копіювання таблиць.

🔗На замітку

Запит на видалення *цілком* видаляє записи, а не тільки зазначені поля даних. Для видалення даних з окремих полів необ-

хідно використовувати запит на відновлення, щоб замінити значення цих полів на порожні значення (*Null*).

Майстер запитів на пошук повторюваних записів

Цей майстер допомагає створити запит, що вказує на однакові записи в таблиці, на підставі значень поля в цій таблиці. ACCESS запитує, по яких полях потрібно виконувати пошук дублікатів і які поля повинні бути присутніми у результуючій таблиці.

Майстер запитів на пошук записів без підлеглих записів

Цей майстер допомагає створити запит, що дозволяє знайти в двох зв'язаних таблицях *запис-сироту* і *запис-вдову*.

Запис - сирота – це запис у таблиці з боку *багатьох*, котрий не має відповідного запису в таблиці з боку *один*.

Запис – вдова – це запис у таблиці з боку *один* зв'язку *один-до-одного* чи *один-до-багатьох*, що не має відповідного запису в іншій таблиці.

При запуску цього майстра ACCESS запитує імена таблиць, для яких необхідно виконати порівняння, ім'я поля, по якому створений зв'язок між таблицями, імена полів першої таблиці, що повинні виводитися в результуючій таблиці, і назву результуючої таблиці. Потім створюється запит, що дозволяє відшукати записи, що не мають відповідних записів в іншій таблиці.

Якщо ви створюєте зв'язки таблицями і намагаєтеся встановити опцію підтримки цілісності даних, а ACCESS не в змозі зробити це, ймовірно, виникли проблеми, зв'язані з записами, що допоможе вирішити майстер запитів на пошук записів без підлеглих записів

Збереження запитів на зміну

Збереження запиту на зміну нічим не відрізняється від збереження інших типів запитів. Для цього в режимі конструктора запиту необхідно клацнути на кнопці *Зберегти* панелі інструментів (чи вибрати з меню пункт *Файл →Зберегти*). Якщо збереження запиту виконується вперше, ACCESS запропонує ввести ім'я для нового запиту.

Зберегти запит можна також при виході з поточного режиму. Для цього необхідно вибрати з меню пункт *Файл →Закрити* і на питання про збереження змін відповісти **Так**.

Проблеми, що виникають при роботі з запитами на зміну

При роботі з запитами на зміну необхідно пам'ятати про деякі можливі проблеми. При виконанні запиту можуть з'явитися повідомлення про те, що, наприклад, загублено кілька записів через руйнування ключа чи що записи заблоковані на час виконання запиту. Нижче розповідається про деякі проблеми і про те, як їх уникнути.

Помилки даних при додаванні і відновленні

ACCESS не дозволяє вводити значення, що не підходять для існуючого типу поля. При цьому просто ігнорується те, що вводиться і привласнюється полю значення *Null*. При роботі з запитами на додавання це означає, що записи будуть додані, але вони можуть виявитися порожніми!

Руйнування ключа при роботі з запитами на зміну

Додати записи з ключовим полем в іншу базу даних, у якій також зазначене ключове поле, неможливо: ACCESS не допускає відновлення запису і зміни ключового поля, якщо воно вже задано. Значення ключового поля можна змінити тільки в наступних випадках:

- значення ключового поля ще не зазначено;
- значення змінюваного поля не зв'язано з полями інших таблиць.

ACCESS не дозволяє видалити поле з боку *один* у зв'язку *один-до-багатьох* раніше видалення записів з боку *багатьох*.

ACCESS не дозволяє додати чи оновити значення поля, якщо в результаті з'явиться повторний запис в *індексованому полі*, збіги в якому не допускаються.

Блокування полів у багато користувальницькому середовищі

ACCESS не буде виконувати запити на зміну записів, що заблоковані іншими користувачами. У цьому випадку при виконанні запитів на відновлення чи на додавання можна продовжити роботу над іншими записами; однак при цьому складно буде визначити, які записи були змінені, а які ні.

Текстові поля

При додаванні чи відновленні текстових полів, довжина яких менше поточної довжини поля, відбудеться усікання текстових даних, про що ACCESS *не повідомить*.

Резюме

– Перед виконанням запитів на зміну необхідно зробити резервні копії таблиць.

– Перед створенням запитів на зміну необхідно створити запит на вибірку для перевірки правильності добору даних.

– Запит на додавання може працювати одночасно тільки з однією таблицею.

– Для виконання запиту на додавання необхідна таблиця, у яку будуть додані записи (запит на додавання такої таблиці не створює).

– Для додавання всіх записів однієї таблиці в іншу краще скопіювати таблицю в буфер обміну, а потім вставити її в іншу таблицю, чим використовувати запит на додавання.

– Запити на створення таблиці можуть використовувати поля з однієї чи декількох таблиць для створення нової.

– Запити на видалення дозволяють видаляти записи з декількох таблиць, зв'язаних відношенням *один-до-одного*.

– При використанні запитів на видалення для таблиць, зв'язаних відношенням *один-до-багатьох*, спочатку варто видалити записи з боку *багатьох*, а потім – з боку *один*.

– Не рекомендується зберігати запит на зміну, якщо ви не збираєтесь використовувати його багаторазово.

– При виконанні запитів на зміну дотримуються умов цілісності даних. Якщо при виконанні запиту на зміну відбудеться порушення умов цілісності даних, ACCESS припинить виконання даної операції.

Контрольні питання

1. Поняття групового запиту.
2. Категорії групових запитів.
3. Що називається підсумковими функціями, для чого вони використовуються?
4. Склад і призначення підсумкових функцій.
5. Визначення критеріїв для групового запиту.
6. Призначення операторів *Where* та *Having*.
7. Створення вираження для групових операцій.
8. Поняття Створення запитів на відновлення перехресного запиту.
9. Послідовність створення перехресного запиту.
10. Як задаються критерії для перехресного запиту.

11. Робота з Майстром перехресного запиту.
12. Призначення і технологія створення параметричного запиту.
13. Створення запитів на відновлення.
14. Побудова запиту на створення таблиці.
15. Побудова запиту на додавання записів.
16. Побудова запиту на видалення записів.
17. Майстер запитів на пошук повторюваних записів.
18. Майстер запитів на пошук записів без підлеглих записів.
19. Проблеми, що виникають при роботі з запитами на зміну, і можливі шляхи їхнього рішення.

Лабораторна робота №6

СТВОРЕННЯ І ВИКОРИСТАННЯ ФОРМ ДАНИХ

6.1 Мета роботи

Метою роботи є ознайомлення і придбання практичних навичок створення і використання форм даних.

6.2 Завдання на лабораторну роботу

6.2.1 Ознайомитися з теоретичним матеріалом з пункту 6.3.

6.2.2 Для таблиць і запитів, створених у попередніх лабораторних роботах, за узгодженням з викладачем розробити різні типи форм введення і відображення даних, у тому числі з використанням макросів.

6.2.3 Оформити звіт з лабораторної роботи, відповіді на контрольні питання.

6.3 Короткі теоретичні відомості і вказівки по виконанню роботи


Форми зручні для перегляду, введення, зміни чи видалення даних, що форматовані. Режим форми забезпечує максимальну гнучкість перегляду даних, дозволяючи відображати одночасно всі поля однієї чи декількох записів.

У ACCESS використовуються шість основних типів форм:

- у стовпець (називається також повно екранною формою);
- стрічкова;
- таблицна;
- головна / підлегла;
- зведена таблиця;
- діаграма.

Існує три методи створення форми, які можна використати, знаходячись у вікні бази даних.

1. За допомогою команди меню **Вставка** → **Форма**.
2. З використанням вкладки **Форми**, далі клацніть на кнопці

Створити.  **Мастер форм**

3. У режимі таблиці чи запиту клацніть на кнопці **Новий об'єкт** виберіть опцію **Нова форма**.

В усіх випадках з'явиться діалогове вікно **Нова форма** (рис.6.1).

*Після вибору опції **Майстер форм** потрібно вибрати поля, зовнішній вигляд форми і стиль форми.*

6.3.1 Елементи керування та їх властивості

У ACCESS елементом керування називається будь-який елемент форми чи звіту.

Елементи керування можна створити за допомогою спеціальної панелі (рисунок 6.2), що викликається командою **Вид → Панель елементів** чи за допомогою кнопки **Панель елементів**.

Приєднані, вільні й елементи керування, що обчислюються

Існують три основних типи елементів керування:

- приєднані;
- вільні;
- обчислювані

Приєднаними елементами керування називають елементи, що зв'язані з полем таблиці. При введенні значення в приєднаний елемент керування поле таблиці в поточній записі автоматично оновлюється.

Вільні елементи керування (змінні, чи *змінні пам'яті*) зберігають введену величину без відновлення поля таблиці. Їх можна використовувати для відображення тексту, значень, що повинні бути передані макросам, ліній і прямокутників. Крім того, вони призначені для збереження об'єктів OLE (наприклад, малюнків), що містяться не в таблиці, а в самій формі.

Елементи керування, що обчислюються, створюють на основі таких виражень, як функції і формули. Оскільки вони не приєднані до полів таблиці, вони не оновляють вміст полів таблиці 6.1.

Вирівнювання елементів керування

Для вирівнювання елементів керування по одній лінії потрібно спочатку виділити їх, утримуючи натиснутою клавішу **Shift** чи перетаскуючи покажчик миші через потрібні елементи, вкладаючи їх в рам-

ку, а потім використовувати команду **Формат** → **Вирівняти опцію** чи **Вирівняти** в контекстному меню.

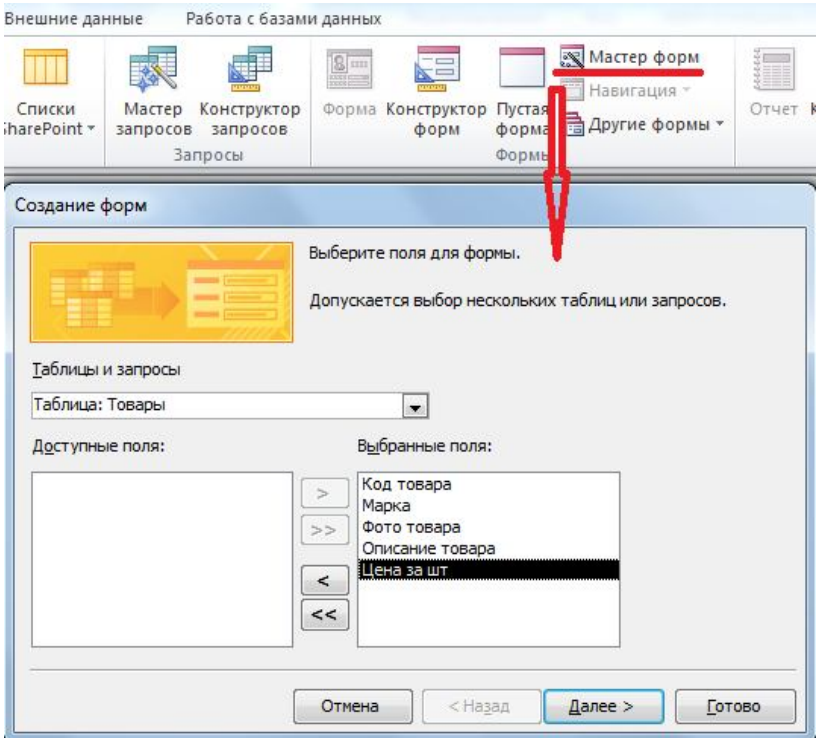


Рисунок 6.1 – Діалогове вікно *Нова форма*

Вирівнювати елементи керування можна також, прив'язавши їх до сітки чи задавши інтервали по горизонталі чи вертикалі за допомогою команди **Формат** → **Інтервал по вертикалі** чи **Формат** → **Інтервал по горизонталі**.

Порада

При вставці об'єкта OLE (наприклад, малюнка) він може не вміститися цілком у рамку. Необхідно вибрати значення властивості **Установка розмірів** для даного елемента рівним **По розміру рамки**.

Властивості елементів керування

Властивостями називають параметри елементів керування, полів і об'єктів баз даних, що визначають характеристики об'єктів, наприклад, розмір, колір, зовнішній вигляд і ім'я. За допомогою властивостей можна також змінювати тип поводження елементів керування, визначаючи, наприклад, чи буде елемент керування схований, чи можливо в ньому редагування.

Деякі властивості, наприклад, формат поля, маска введення, усипаковуються з поля таблиці, зв'язаного з елементом керування. При цьому зміна властивостей елемента керування не викликає зміни властивостей поля в таблиці-джерелі.

Для заданого типу елемента керування можна створити набір стандартних властивостей (див. нижче), клацнувши на кнопці елемента обраного типу панелі елементів, а потім на кнопці **Властивості** панелі інструментів. Змінюючи набір властивостей, прийнятих за замовчуванням, можна створювати форми набагато швидше, ніж змінюючи їх для кожного елемента керування по окремі.

Типи елементів керування

Таблиця 6.1 – Елементи керування, використовувані у формах і звітах

Елемент керування	Призначення
<i>Основні елементи керування</i>	
Напис	Служить для відображення тексту
Поле	Дозволяє вводити дані
<i>Розширені елементи керування введення даних</i>	
Група перемикачів	Містить кілька вимикачів, перемикачів чи прапорців
Вимикач	Може знаходитися в двох станах: натиснутому чи віджатому, вгору чи вниз. Звичайно на нього поміщають картинки чи піктограми
Перемикач	Відображається у виді кружка з крапкою всередині

	ні, що позначає активізацію цього елемента. Його також називають <i>радіокнопкою</i> .
Прапорець	Може знаходитися в двох станах. Відображається у вигляді квадрата з галочкою усередині, що позначає активізацію цього елемента
Поле зі списком	Список значень, що розкривається, в який можна вводити й інші значення
Список	Список значень, що відображаються в формі чи звіті
Кнопка	Звичайно використовується для виклику макросу чи запуску програми на мові Basic для виконання якої-небудь дії
Підлегла форма	Призначена для відображення ще однієї форми чи звіту у вихідній формі чи звіті
Набір вкладок	Дозволяє відображати кілька сторінок у виді папки з вкладками
<i>Елементи керування графікою</i>	
Малюнок	Відображає на екрані растрове зображення
Вільна рамка об'єкта	Містить об'єкт OLE чи малюнок, що не зв'язаний з полем таблиці. Крім того, може містити діаграми, звук і відеозображення
Приєднана рамка об'єкта	Містить об'єкт OLE чи малюнок, зв'язаний з полем таблиці.
Лінія	Лінія, товщину і колір якої можна змінювати. Використовується для поділу елементів керування
Прямокутник	Прямокутник, розміри і колір якого можна змінювати. Може бути порожнім чи заповненим і використовуватися для виділення елементів форм
Розрив сторінки	Звичайно використовується в звітах і позначає початок нової сторінки

6.3.2 Створення і зміна форм введення даних

Створення форми

Для створення форми і зв'язування її з таблицею бази даних (як уже розглядали при вивченні способів уведення даних у таблиці) чи запитом необхідно виконати наступні дії.

В вікні бази даних клацніть на кнопці **Форми і Створити**.

В вікні, що з'явилося, **Нова форма** виберіть **Конструктор**, а в списку, що розкривається нижче - ім'я відповідного запиту чи таблиці.

Розгорніть вікно форми до максимального значення.

Розміщення полів у формі

Для розміщення елементів керування у формі потрібно виконати три основних етапи.

1. Відобразити вікно списку полів за допомогою щиглика на кнопці **Список полів** панелі інструментів.

2. Визначити тип створюваного елемента керування за допомогою щиглика на потрібній кнопці панелі елементів.

3. Вибрати потрібні поля і перетягнути їх у вікно конструктора форм (аналогічно побудові запиту). При цьому перетаскувати відразу можна *групу полів*, попередньо виділивши їх звичайним чином.

Зміна послідовності переходу між елементами

За замовчуванням послідовність переходу між елементами форми завжди відповідає порядку, у якому вони додавалися в цю форму. Для зміни послідовності переходу у вікні конструктора форм виберіть команду **Вид →Послідовність переходу**. Кнопка **Авто** дозволяє розмістити поля в порядку зліва праворуч і зверху вниз, відповідно до їхнього розташування у формі.

Додавання фонового малюнка

Для додавання фонового малюнка потрібно виконати наступні дії.

1. Вибрати форму, клацаючи в її верхньому лівому куті (у місці перетинання двох лінійок).

2. Викликати вікно властивостей і клацнути на властивості **Малюнок**.

3. Ввести шлях до файлу з малюнком.

4. Клацнути на властивості **Масштаби малюнка** і задати нове значення.

6.3.3 Термінологія конструювання форм

Дослідження режиму Конструктора форми

Сітка в центрі вікна режиму конструктора форми (рис.6.2) представляє розмір форми при її запуску. Щоб змінити розмір форми, виділіть край сітки і розтягніть її до потрібного розміру. Зверніть увагу на лінійки уздовж верхнього і лівого країв сітки: вони допомагають задати остаточний розмір форми.

Щоб змінити властивості самої форми, клацніть правою кнопкою миші на області виділення форми чи звіту (прямокутник на перетинанні лінійок ліворуч угорі). У контекстному меню, що з'явилося, виберіть команду **Властивості**. Тоді відкриється вікно **Форма**, у якому можна задати усі властивості форми (рис. 6.3).

Щоб змінити властивості форми у вікні Форма, виконайте наступні дії.

1. Відкрийте вікно **Форма**, якщо воно ще не відкрито.
2. Клацніть на вкладці, що відноситься до змінюваної властивості. Нижче проводиться список і короткий опис вкладок у вікні властивостей форми (див. також рис.6.3 – 6.4).
3. Виберіть текстове поле біля змінюваної властивості і введіть нове значення. Деякі властивості дозволяють запустити *Побудовник виразів (Построитель выражений)*, робота з яким була вже розглянута раніше.
4. Закрийте вікно властивостей форми. При цьому внесені зміни зберігаються.

Сітка, на якій розміщуються елементи керування, за замовчуванням служить для фіксації елементів керування. Це допомагає вибудовувати елементи керування на форму в лінію. Іноді таке захоплення сіткою елементів керування заважає роботі. Елементи продовжують зберігати невірне розташування, крім того, буває важко змінити їхній розмір. Щоб блокувати захоплення сіткою елементів керування, скиньте прапорець *Прив'язка до сітки* в меню **Формат**. Щоб тимчасово деактивізувати опцію *Прив'язати до сітки*, утримуйте натиснутої клавішу <Ctrl> при переносі чи зміні розміру елемента керування на сітці.

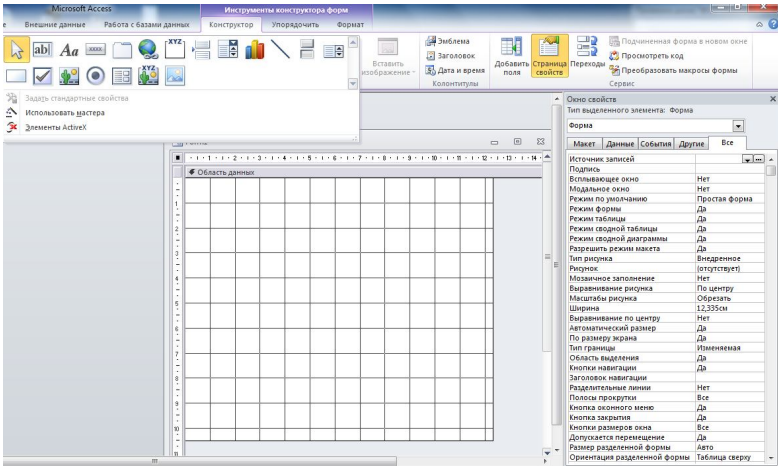


Рисунок 6.2 – Вікно Конструктора форм.

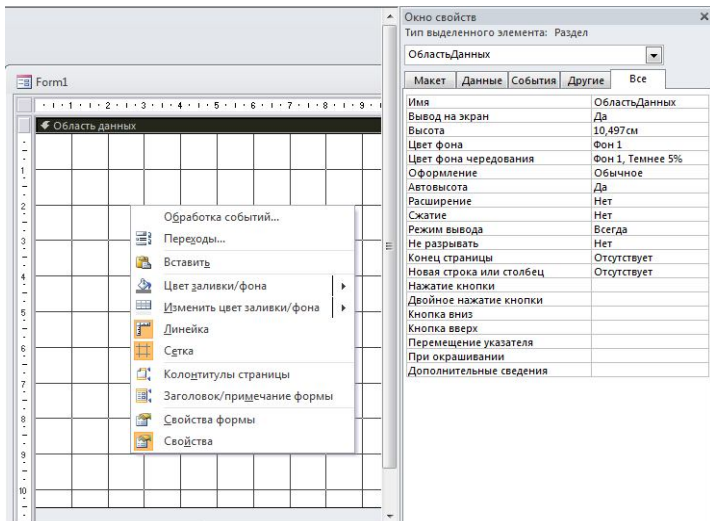


Рисунок 6.3– Вікно властивостей форми.

Додавання до форми елементів керування

До числа елементів керування відносяться, кнопки, прапорці і написи, розміщені на формі (див. панель елементів на рис.6.2). Розміщення елементів керування на формі (чи в сітці) полягає в їхньому перетаскуванні й опусканні шляхом щиглика в потрібному місці. Щигликом миші елементи керування вибирають з вікна шляхом виконання щиглика в потрібному місці. При вибірці елемента керування з панелі покажчик миші перетворюється в перехрест'я. Воно дозволяє перетягнути елемент керування в потрібне місце на сітці і наступному протяганні вказати два протилежних кути місця розташування елемента керування. Коли елемент керування розміщується на сітці, покажчик миші відновлює свою звичайну форму. Панель інструментів, застосовувана в режимі Конструктора, представлена вище на рис.6.2.

Ліва верхня кнопка з зображенням стрілки покажчика миші натиснута чи віджата в залежності від поточного фокуса вікна. Коли ця кнопка натиснута, можна вибирати об'єкти за допомогою покажчика миші. Права верхня кнопка з зображенням «чарівної палички» (позначка “ / “) запускає чи скасовує майстра створення елементів керування. Деякі елементи керування, такі як **Кнопки**, мають (як уже відзначалося вище) асоційовані з ними майстри. За допомогою цієї кнопки можна скасовувати і дозволяти запуск Майстра.

У таб.6.1 описувалися деякі стандартні елементи панелі інструментів.

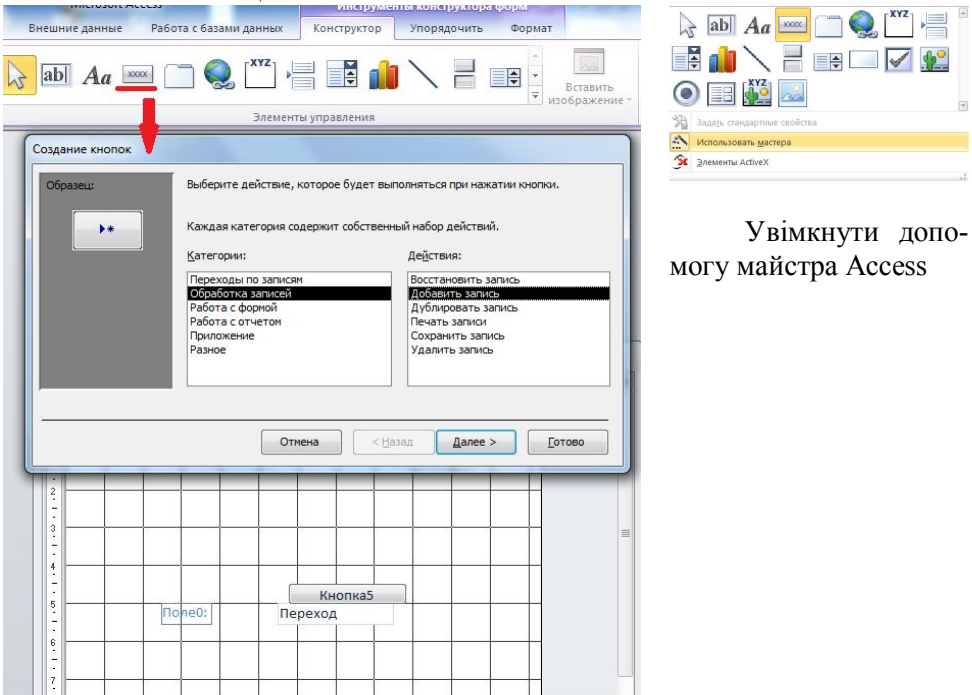
Елемент керування **Кнопка** широко застосовують також для виклику інших форм, тому нижче описане додавання до форми кнопки і застосування відповідного Майстра.

1. Виберіть на панелі елемент керування **Кнопка**
2. Розташуйте елемент керування на формі, розміщуючи два протилежних кути. Запускається Майстер кнопок.
3. В лівому списку виберіть категорію дії, що повинна виконувати кнопка.
4. В правому списку виберіть необхідну дію і клацніть на кнопці **Далі**.
5. Введіть необхідний текст напису на кнопці чи виберіть для неї придатний малюнок.
6. Введіть ім'я кнопки і клацніть на кнопці **Готово**.

Як створювати властивості елементів керування.

Для кожного елемента керування є таблиця властивостей, подібно до таблиці властивостей форми. На рис.6.5 показано вікно властивостей елемента керування **Кнопка**.

Вікна властивостей різних елементів злегка відрізняються одне від іншого, але в цілому їхня структура подібна та схожа на вже вивчені елементи керування таблиць Excel. Це полегшує недосвідченим користувачам заміну установок властивостей. Так, для елементів **Лінія** та **Кнопка** розходження обмежуються тим, що з елементом керування **Кнопка** асоційовано більше властивостей.



Увімкнуті допомогу майстра Access

Рисунок 6.4 – Діалогове вікно Майстра Кнопок.

Щоб змінити властивість у вікні властивостей, виконайте наступні дії.

1. Відкрийте вікно властивостей, клацнувши правою кнопкою миші на елементі керування і виберіть в контекстному меню команду **Властивості (Свойства)**.

2. Клацніть на вкладці, що містить змінювану властивість.

3. Виберіть текстове поле біля змінюваної властивості і введіть його значення. Деякі властивості дають можливість відкрити вікно макросу чи запустити будівника вираження.

4. Закрийте вікно властивостей. При цьому внесені зміни зберігаються.

6.3.4 Робота з підлеглими формами

Підлеглими формами називають форми, поміщені всередину інших форм; це дає можливість переглядати на одному екрані значно більше інформації. Форму, що містить всередині себе підлеглу форму, називають головною.

Підлеглу форму встановлюють окремо від головної. Вона приводиться в списку на вкладці форми вікна бази даних. Зв'язок між таблицями, на які створені як головна, так і підлегла форми, забезпечує синхронізацію цих форм. Приведемо деякі поняття, корисні при роботі з підлеглими формами:

– Таблиці і запити, що служать джерелом даних форми, зазвичай містять зв'язки типу відносини «один-до-багатьох». Таблиця чи запит, що відноситься до підлеглої форми, є на боці «до багатьох» даного зв'язку. Таблиця ж і запит стосовно до головної форми, знаходиться на боці «один».

– Підлегла форма повинна бути створена в першу чергу. Перед переміщенням її в головну форму вона повинна бути збережена.

– У форму можна помістити кілька підлеглих форм.

– Підлегли форми можуть бути вкладеними. Інакше кажучи, в підлеглий формі також може знаходитися підлегла форма.

– У головній формі може знаходитися не більш двох підлеглих форм, що містять у собі підлегли форми.

Щоб створити підлеглу форму, виконайте наступне.

1. Відкрийте головну форму в режимі **Конструктор**, клацнувши на вкладці **Форми** у вікні бази даних, а потім клацнувши на кнопці **Конструктор**.

The image shows a screenshot of a software interface for creating forms. The window title is "Сотрудники" (Employees). Below the title bar, there is a header "Область данных" (Data area). The main area contains a grid of fields for data entry. The fields are arranged in two columns:

- Left column: "Код сотрудника" (Employee code), "Имя" (Name), "Фамилия" (Surname), "Фото сотрудника" (Employee photo), "Характеристика" (Characteristic), "Базовый оклад" (Basic salary).
- Right column: "Код сотру" (Employee code), "Имя" (Name), "Фамилия" (Surname), "Характеристика" (Characteristic), "Базовый окл" (Basic salary).

Рисунок 6.5 – Вікно Конструктора при створенні підлеглих форм

2. Виберіть в меню **Вікно** команду **Ліворуч праворуч**. При цьому вікно бази даних і вікно **Режим конструктора форми** розташуються по горизонталі без перекриття

3. Перетягніть з вкладки **Форми** у вікні бази даних і опустіть в потрібному місці головної форми ту форму, що повинна стати підлеглою

4. Збережіть і закрийте головну форму.

6.3.5 Додавання елементів керування з використанням макросів

Переглядати всі об'єкти (у нашому прикладі – пісні) у табличній формі може бути незручно, якщо записів багато. Потрібна якась форма представлення списку пісень, що дозволила б знайти потрібну. Крім того, таблична форма не дозволяє включати ніякі елементи керування (кнопки, напису і т.д.), крім власне таблиці, тому її зазвичай застосовують для підлеглих форм.

Виберемо для роботи стрічкову форму і створимо її для таблиці "Пісня" за допомогою Майстра форм, як описано вище. Однак у та-

кій формі будуть відображатися всі записи з таблиці, що незручно, якщо записів багато. Тому створимо кнопки, що дозволяють фільтрувати записи у формі. Нехай це будуть кнопки з буквами алфавіту. Спочатку збільшимо висоту області заголовка форми, щоб у нього помістилися всі кнопки. Потім відключимо кнопку **Майстер елементів** (ControlWizard) на панелі інструментів, тому що в даному випадку Майстер нам не допоможе. Тепер виберіть на панелі елементів елемент керування **Кнопка** (CommandButton) і потім помістіть покажчик миші у верхній лівий кут області заголовка і клацніть лівою кнопкою. З'явиться досить довга кнопка з стандартним заголовком *Кнопка 17* (17-це порядковий номер елемента, у вас він може бути інший). Якщо у вас відкрите вікно **Властивості** (Properties), розкрийте вкладку **Макет** (Format) і введіть значення властивості **Підпису** (Caption)-**A** (це може бути будь-як буква, з якої починаються обрані вами назви об'єктів у фільтруємому полі).

За допомогою кнопки на панелі інструментів **Формат (форма/звіт)** зробіть цей напис напівжирним і потім підведіть покажчик миші до будь-якої границі кнопки. Коли він перетвориться в двонаправлену стрілку, двічі клацніть лівою кнопкою миші. Кнопка стане маленької і прийме квадратну форму. Можете перемістити її в потрібне місце. Тепер, оскільки ми поки обходимося без програмування, нам доведеться створити макрос.

Клацніть на ярлику **Макроси** (Macros) у вікні **База даних** (Database) і натисніть кнопку **Створити** (New) на панелі інструментів вікна бази даних. Відкриється вікно **Конструктора макросів**. Нам доведеться зберегти кілька макросів в одному об'єкті, кожен макрос буде мати своє ім'я. Тому потрібно, щоб у вікні **Конструктора макросів** відображався стовпець **Ім'я макросу** (MacroName). Якщо ви його не бачите, натисніть кнопку **Імена макросів** (MacroName) на інструментальній панелі. Введіть ім'я першого макросу: **A**. У стовпці **Макрокоманда** (Action) потрібно вибрати зі списку, що розкривається, потрібну макрокоманду: **Застосувати Фільтр** (ApplyFilter). У нижній частині вікна **Конструктора** з'являються поля, у яких можна задати умови фільтрації записів. Введіть в поле **Умови добору** (WhereConditions) рядок *Ucase ([Прізвище]) Like 'A*'*. Це вираження є частиною пропозиції *Where* інструкції SQL.

Тепер можна написати аналогічні макроси для інших букв алфавіту. Кожен макрос буде складатися з однієї макрокоманди **Застосува-**

ти Фільтр (ApplyFilter), тільки у вираженні в умовах добору будуть присутні різні букви. Після того, як будуть створені всі необхідні макроси, потрібно зберегти їх в об'єкті з ім'ям "Фільтр пісень". Для цього можна виконати команду меню **Файл →Зберегти** (File→Save) чи натиснути клавішу <F12>. У діалоговому вікні **Збереження** (Save) введіть ім'я макросу *Фільтр Пісень*.

Нарешті, залишилися підключити цей макрос до кнопки у формі. Виділіть елемент керування **Кнопка** і розкрийте вкладку **Події** (Events) у діалоговому вікні **Властивості** (Properties). Ви побачите список подій, зв'язаних із кнопкою. Кожна з цих подій може бути оброблена за допомогою макросу чи процедури VBA. Клацніть лівою кнопкою миші в поле значення властивості **Натискання кнопки** (Click) і розкрийте список. Виберіть зі списку елемент *Фільтр пісень А*. Тепер при натисканні кнопки "А" у формі буде виконуватися фільтрація записів.

Інші кнопки ми створимо за допомогою копіювання кнопки "А". Для цього виділіть цю кнопку і натисніть клавіші <Ctrl>+<Ins>, щоб скопіювати її в буфер. Натисніть клавіші <Shift>+<Ins>. З'явиться ще одна кнопка з написом *А*. Перемістіть її на потрібне місце, змініть напис на *Б*, а значення властивості **Натискання кнопки** (Click) для нового елемента керування установіть рівним *Фільтр пісень Б*. Натисніть ще раз <Shift>+<Ins> і одержите наступну кнопку. Проробіть з нею аналогічні операції і підключіть до неї макрос *Фільтр пісень В*. Продовжуйте ці дії доти, поки не одержите потрібну кількість кнопок.

Наступна операція, що ми будемо робити, теж зв'язана зі створенням кнопки. Цю кнопку ми створимо в розділі приміток форми, і при її натисканні буде відкриватися, наприклад, інформація про пісню, назва якої виділена в списку (у нашому випадку це може бути інформація про альбом пісень). Тепер потрібно знову натиснути кнопку **Майстер елементів** (ControlWizard) на панелі елементів. Знову виберіть на панелі елементів елемент керування **Кнопка** (CommandButton) і клацніть лівою кнопкою миші в області приміток форми. Запуститься Майстер кнопок, що спочатку запропонує вам вибрати ту дію, що повинне виконуватися при натисканні кнопки. Виберіть спочатку в списку **Категорії** (Categories) елемент **Робота з формою** (FormOperation), а потім у списку **Дії - Відкрити форму** (OpenForm) (рис.6.6)

Потім вам потрібно вказати ім'я форми, що відкривається - "Каталог пісень", вибрати перемикач **Відкрити форму для відібраних записів** (Open the form and find specific data to display) і вказати відповідні поля в обох формах. Для цього виберіть у списках "Альбом" і "Пісня" поле "Код" і натисніть кнопку з двонаправленою стрілкою між списками.

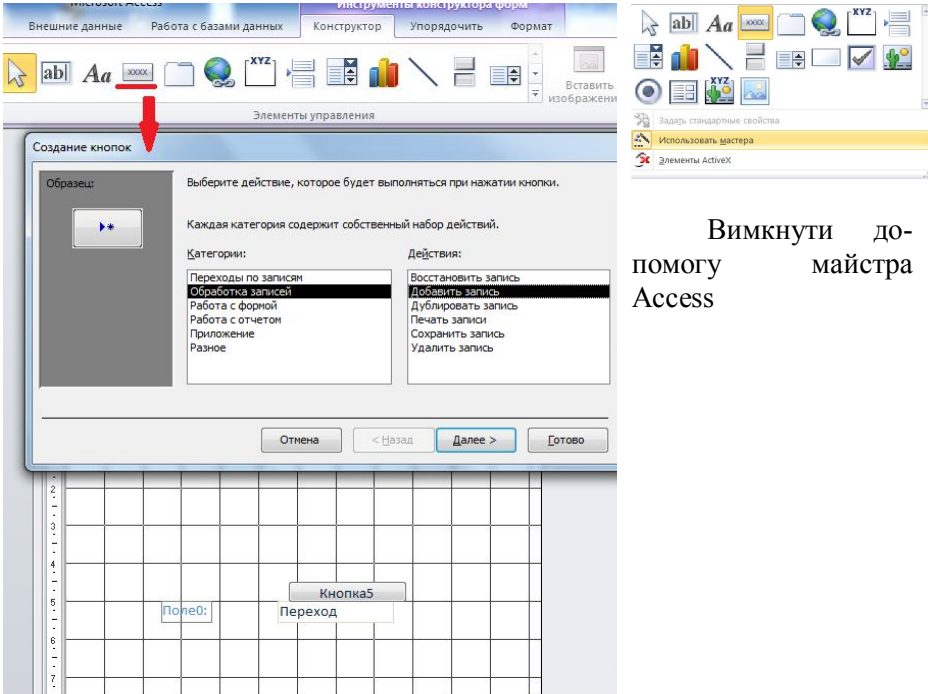
В інших вікнах **Майстра кнопки** потрібно вказати напис на кнопці - *Перегляді* ім'я кнопки - cmdView. Коли Майстер закінчить роботу, ви побачите у формі кнопку. Властивість **Натискання кнопки** (Click) при цьому буде мати значення **[Процедура обробки подій]** (EventProcedure). Клацніть лівою кнопкою миші в області значення цієї властивості. Праворуч від цієї області з'явиться кнопка **Будівника** (із трьома крапками). Натисніть цю кнопку, і відкриється вікно редактора програм VBA. У цьому вікні відображається процедура VBA, що відкриває форму "Каталог пісень" відображаючи в ній тільки один запис. Ця процедура була створена Майстром, ми не будемо поки її пояснювати, тому що програмування на VBA у цих лабораторних роботах не розглядається.

Впровадження у форми OLE - об'єктів:

У форми чи звіті СУБД ACCESS можна впроваджувати OLE - об'єкти (згідно першим буквам виразу Object Linking and Embedding) з інших додатків, а також установлювати з форм чи звітів бази даних зв'язок з OLE-об'єктами.

OLE - об'єкти можуть бути зв'язані зі значеннями в записах якогось поля (приєднані об'єкти) чи із самою формою чи об'єктом (вільні об'єкти).

У залежності від типу обробки OLE - об'єкт можна або розмістити (*впровадити*) у формі, або вказати програмі, що потрібно знайти зазначений об'єкт і помістити (*зв'язати*) його в рамку приєданого чи вільного об'єкта форми чи звіту.



Вимкнути до-
помогу
Access
маїстра

Рисунок 6.6 – Діалогове вікно створення кнопок

Приєднаний об'єкт – це об'єкт, що міститься в таблиці і який можна відобразити в формі чи надрукувати в звіті. Приєднаний об'єкт зв'язаний з полем, що містить дані типу OLE. Приєднані об'єкти (малюнки чи Документи) можна додавати чи редагувати запис за записом, так само , як і при роботі зі звичайними змінними. Для відображення приєданого об'єкта OLE використовується приєднана рамка об'єкта.

Вільний об'єкт міститься не в таблиці, а безпосередньо в формі чи в звіті. Він є графічним еквівалентом елемента керування типу "Напис". Він не відноситься до жодного з полів запису і не змінюється від запису до запису

Малюнок є ще одним прикладом вільного об'єкта. Якщо у вільній рамці об'єкта OLE допускається редагування об'єкта після подвій-

ного щиглика на ньому, то малюнок тільки відображається у своєму форматі і редагувати його не можна.

Зв'язування і впровадження об'єктів.

При *впровадженні* об'єкта ACCESS зберігає його копію у вашій таблиці. Подвійним щигликом по цьому об'єкті ви можете запустити програму, в якій він був сформований. Після запуску відповідної програми можна робити його зміни.

Завершивши редагування, ви можете вийти з викликаної програми, і копія зміненого об'єкта знову буде збережена в таблиці.

Має сенс *впроваджувати* об'єкт, якщо:

– ви хочете редагувати об'єкт у ACCESS;

– об'єкт повинний бути складовою частиною БД;

– ви не можете сформувавши такий об'єкт безпосередньо в ACCESS (наприклад, малюнок PaintBrush чи звуковий файл).

Ви можете зв'язати об'єкт, створений іншою прикладною програмою, з таблицею ACCESS. У цьому випадку об'єкт повинний бути збережений не в таблиці, а окремо, на диску, у форматі файлу прикладної програми, у якій він був сформований. У разі потреби ACCESS буде витягати поточний варіант зв'язаного об'єкта з цього файлу.

Зв'язок об'єкта рекомендується в тому випадку, коли в ACCESS ви хочете мати доступ до змінюваних іншими програмами об'єктів.

Одне з переваг *зв'язування* перед *впровадженням* складається в можливості зв'язування одного вихідного об'єкта з декількома таблицями ACCESS без збереження його копій у кожній з цих таблиць.

В такий спосіб ви заощадите місце на диску. При зміні об'єкта його не треба буде редагувати багато разів, тому що всі таблиці будуть посилатися на відредагований варіант.

Зв'язку варто віддати перевагу над *впровадженням*, коли, наприклад, ви хочете, щоб таблиця ACCESS посилалася на постійно оновлюваний іншою прикладною програмою об'єкт: на робочий лист чи лист діаграми EXCEL (де знаходяться, наприклад, посадкові оклади співробітників).

У випадку зв'язування ви щораз будете одержувати доступ до останнього варіанта об'єкта, збереженого на диску.

Технологія впровадження об'єкта

Припустимо, що ви хочете впровадити в таблицю *OLE -об'єкт*. Для цього в таблиці повинно бути *OLE - поле*, яке було задане на етапі створення структури таблиці.

1. Активізуйте це поле (встановіть на нього курсор).
2. Скористайтеся командою **Виправлення** → **Вставити об'єкт**, і на екрані з'явиться відповідне діалогове вікно.
3. За допомогою вікна можна сформувати новий об'єкт безпосередньо під час вставки. Ним може бути:
 - зображення *PaintBrush*;
 - звуковий файл;
 - слайд *PowerPoint*;
 - діаграма *MSGGRAPH* чи *MSEXCEL*;
 - інші, раніше сформовані на диску об'єкти.
4. При виборі перемикача **Створити** у вікні розташовується список **Тип** об'єкта.
5. В ньому вказуються всі встановлені на вашому комп'ютері OLE- програми, об'єкти яких можуть бути впроваджені.
6. Виберіть необхідний тип об'єкта з цього списку, натисніть **ОК**, і потрібна програма буде запущена
7. Сформуєте об'єкт і поверніться в ACCESS. Копія об'єкта буде поміщена в таблицю.
8. Якщо на диску вже є об'єкт, то виберіть перемикач *З файлу* і натисніть кнопку **Пошук**. На екрані з'явиться діалогове вікно **Пошук**. Вкажіть диск, каталог, ім'я файлу і натисніть клавішу **ОК**. Обраний файл буде впроваджений у таблицю.

Вправа

Вставимо в форму "Опис альбому" малюнок з тематичної бібліотеки Microsoft Office.

Відкрийте форму "Опис альбому" у режимі **Конструктора форми**. Клацніть на панелі елементів на кнопці **Малюнок**. Створіть рамку малюнка, перетягнувши інструмент **Малюнок**. При створенні рамки малюнка з'явиться діалогове вікно Вибір малюнка, в якому буде представлений перелік файлів з малюнками, що містяться в поточному каталозі. Виберіть придатний по тематиці файл. Після появи цього файлу в діалоговому вікні Вибір малюнка клацніть на кнопці **ОК**.

У результаті малюнок буде впроваджений і відображений у зазначеному місці форми.

Зв'язок з існуючим об'єктом

1. Виберіть в меню **Виправлення** команду **Вставка об'єкта**. Відкриється діалогове вікно.

2. Виберіть в вікні перемикач *З файлу* і натисніть кнопку **Пошук**.

3. Вкажіть диск, каталог, ім'я файлу.

4. Встановіть прапорець **Зв'язок**, і обраний файл буде *зв'язаний* з таблицею, а не *впроваджений* в неї.

Редагування OLE- зв'язків

Якщо ви хочете змінити зв'язки, то зробіть наступні дії.:

– Ввійдіть в меню **Правка**, виберіть команду **Зв'язки**. На екрані з'явиться відповідне діалогове вікно.

– В області *Група* необхідно вказати вид зв'язку OLE.

– В списку **Зв'язки** зазначені всі файли, що в даний момент зв'язані з активною таблицею ACCESS. Зазначено ім'я файлу, каталог, в якому він збережений, а також тип прикладної програми, за допомогою якої він був сформований.

Якщо ви вставили зв'язок і хочете автоматично просліджувати зміну вихідного файлу зв'язаного об'єкта, виберіть у групі *Відновлення* перемикач *Автоматичне*. Якщо вибрати перемикач *По питанню*, то поточні зміни не будуть негайно відслідковані програмою.

В діалоговому вікні **Зв'язки** можна:

– розірвати зв'язок;

– відкрити вихідну програму для редагування об'єкта;

– переміняти джерело, тобто зв'язати таблицю з іншим файлом.

Якщо ви перемістите об'єкт в інший каталог, то цей каталог необхідно вказати, інакше при встановленню зв'язку відбудеться помилка. Якщо ви відкриєте таблицю, що містить зв'язки, ACCESS буде шукати відповідні файли даних.

Якщо об'єкт не буде знайдений, зв'язок не буде встановлено. В цьому випадку треба переміняти джерело чи розірвати зв'язок.

Коротко резюмуємо вищесказане:

У ACCESS можна вставляти малюнки, відеокліпи, файли зі звуком, ділові діаграми, електронні таблиці Excel, а також тексти Word. Крім того, будь-який об'єкт **OLE** може бути зв'язаний з формами і звітами. При цьому їх можна не тільки використовувати в ACCESS, але і редагувати безпосередньо у формі.

Приєднаний об'єкт – це об'єкт, що міститься в таблиці бази даних і який можна відобразити в формі чи надрукувати в звіті. Він зв'яз-

заний з полем, що містить дані типу OLE. Для відображення приєднаного об'єкта OLE використовується *приєднана рамка об'єктів*.

Вільний об'єкт міститься не в таблиці, а безпосередньо в формі чи звіті. Він є графічним еквівалентом елемента керування *написі* зазвичай використовується для об'єктів OLE форми чи звіту. Ці об'єкти не відносяться до жодного з полів запису і не змінюються від одного запису до іншого.

Малюнок є ще одним прикладом вільного об'єкта. Якщо у вільній рамці об'єкта OLE допускається редагування об'єкта за допомогою подвійного щиглика на ньому і запуску додатка-джерела (наприклад, Paintbrush, Word, Excel, програвача файлу зі звуком чи відеозображення), то малюнок тільки відображається (зазвичай у форматі BMP, PCX чи WMF) і редагувати його не можна.

Основна відмінність між зв'язуванням і впровадженням об'єктів у форму чи звіт полягає в тому, що *впровадження* об'єкта приводить до збереження його копії в базі даних. *Зв'язування* об'єкта, навпаки, приводить до збереження в базі даних не об'єкта, а лише адреси його розташування за межами бази даних.

Зв'язування має дві переваги.

- 1.Зміни можна вносити, використовуючи зовнішній додаток (навіть без запуску ACCESS).
- 2.Зовнішній файл не входить до складу файлу бази даних (MDB - файл), що сприяє економії ресурсів дискової пам'яті.

●*Увага!

Якщо зовнішній файл буде переміщений в іншу папку або ім'я файлу, що містить об'єкт, буде змінено, зв'язок цього об'єкта з ACCESS буде порушено.

Впровадити вільний об'єкт у форму чи звіт можна двома способами.

- 1.Вставити об'єкт у форму чи звіт (при цьому буде створений об'єкт типу малюнок чи вільна рамка об'єкта).
- 2.Спочатку створити об'єкт типу малюнок чи вільну рамку об'єкта, а потім вставити об'єкт чи малюнок в цю рамку.

Якщо об'єкт не є об'єктом OLE, його варто вставляти у форму безпосередньо в виді малюнка чи вільної рамки об'єкта за допомогою команд **Копіювати** і **Вставити**.

Для цього необхідно виконати наступні дії.

1. Створіть чи відобразите на екрані об'єкт, використовуючи зовнішній додаток.

2. Скопіюйте чи виріжіть цей об'єкт в буфер обміну.

3. Перейдіть в вікно форми чи звіту і вставте об'єкт.

В результаті буде автоматично створена вільна рамка об'єкта.

Електронні таблиці, текстові документи й інші об'єкти можна зберігати як дані в таблиці. Приєднані об'єкти, наприклад, малюнок з редактора PaintBrush, електронну таблицю або інший об'єкт, створений в будь-якому іншому OLE додатку, зберігаються в таблиці в полі типу OLE. Відразу після створення приєднану рамку об'єкта можна зв'язати з полем OLE - таблиці за допомогою властивостей **Дані**, а потім використовувати її для впровадження об'єктів в записі таблиці.

Контрольні питання

1. Способи створення форм і їхнє призначення.
2. Елементи керування форм.
3. Використання **Майстра форм**.
4. Використання **Конструктора** при створенні форм.
5. Підлеглі форми.
6. Способи створення елемента **Кнопка**, використання для цієї мети макросів
7. Типи OLE- об'єктів.
8. Впровадження OLE- об'єктів у форму.

Лабораторна робота №7

ЗВІТИ ПО БД

7.1. Мета роботи:

Створити звіти по базі даних

7.2. Завдання до лабораторної роботи

7.2.1 Ознайомитися з теоретичним матеріалом у пункті 7.3.

7.2.2. Створити не менше трьох звітів: з групуванням за допомогою Майстра, з використанням підсумкових значень (тобто даних по числовим полям) та сортування, за виглядом виданої Справки та з даними про час складання звіту (використання функцій Access типу Now(), Date() Time(), Format()).

7.3 Короткі теоретичні відомості

7.3.1 Типи звітів

Звіт - це засіб оформлення інформації, що зберігається в таблиці чи в запиті. Звіт є однією з трьох можливостей перегляду даних. Інші дві можливості – це форма і безпосередньо таблиця даних. У звіті зазвичай виводяться відразу кілька записів, причому можна вказати, у якому вигляді потрібно одержати і вивести дані.

У ACCESS можна створювати наступні типи звітів (дивись. рисунок 7.1).

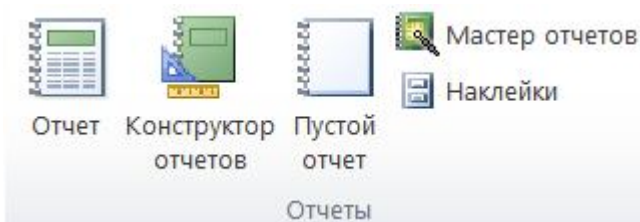


Рисунок 7.1 – Вікно створення звіту.

1. Стрічкові звіти. Дані роздруковуються у виді рядків і стовпців із групуванням і підсумками. Їх різновиди включають звіти з підсумковим і груповим підсумовуванням.

2. *Звіти в стовпець*. Дані роздруковуються у виді форми з можливим включенням підсумків і графіків.

3. *Звіти для розсилання*. Створюються як шаблони листів.

4. *Поштові наклейки*. Створюються наклейки для конвертів.

Стрічковий звіт.

Він схожий на таблицю. Для поділу даних у стрічкових звітах можна використовувати номери сторінок, дати складання звітів, лінії і прямокутники. У них можуть використовуватися обрамлення і заливання, малюнки, ділова графіка і поля МЕМО.

Звіти в стовпець.

Вони відображають одну чи кілька записів на сторінці по вертикалі. В нього можуть входити розділ інформації з записів однієї таблиці і розділ з декількома зв'язаними записами з іншої таблиці.

У ACCESS Майстер звітів допомагає створити практично будь-який тип звіту.

Щоб створити за допомогою Майстра звіт в один стовпець, потрібно таке.

1. Відкрити базу даних.

2. Вибрати вкладку **Звіт**.

3. Вибрати команду **Створити** і з'явиться діалогове вікно для створення звіту.

4. Клацнувши на стрілці, розкрити список *Вибір таблиці/запиту* і вибрати таблицю чи запит, по якому створюється звіт.

5. Клацнути по кнопці **Майстер**, і на екран буде виведене діалогове вікно **Майстра по розробці звітів**, де можна вибрати тип звіту.

6. Вибрати тип *в один стовпець*, потім натиснути на кнопці **ОК**.

7. На екрані з'явиться перша сторінка діалогового вікна програми **Майстра звіту в один стовпець**.

8. Вибрати необхідні для звіту поля зі списку *Наявні поля*, перемістити їх у список полів звіту кнопкою ">", а всі поля відразу – кнопкою ">>". Забрати окреме поле можна кнопкою "<", а всі поля - кнопкою "<<".

9. Клацнувши на кнопці **Далі**, реалізується перехід до другої сторінки.

10. Вибрати перше поле, по якому ACCESS повинна робити сортування записів, наприклад, за абеткою. Перемістити, його кнопкою ">" у розділ *Порядок сортування*.

Якщо для записів, що повторюються по першому полю сортування, потрібно виконувати сортування по другому і наступним полям, то варто проробити те ж саме. Тут уже відомими кнопками можна переносити поля.

11. Клацнути на кнопці **Далі**, і з'явиться третя сторінка, де зображується, як буде виглядати звіт.

12. Вибрати придатний тип звіту й орієнтацію його сторінок.

13. Задати ім'я, клацнути на кнопці **Готово**, і ACCESS продемонструє звіт у вікні попереднього перегляду.

7.3.2 Створення більш складних звітів

З групуванням за допомогою Майстра.

1. Відкрити базу даних.

2. Вибрати вкладку **Звіт**

3. Вибрати команду **Створити**, з'явиться діалогове вікно для створення звіту.

4. Клацнувши на стрілці, розкрити список *Вибір таблиці /запиту* і вибрати таблицю чи запит, по якому створюється звіт.

5. Клацнути на кнопці **Майстер**, і на екран буде виведене діалогове вікно **Майстра** по розробці звітів, де можна вибрати тип звіту.

6. Вибрати звіт *Групування даних і обчислення підсумків*. База даних вимагає розбивки даних на групи й обчислення підсумків по групах даних.

7. Клацнути **ОК**, і з'явиться перша сторінка діалогового вікна.

8. Вибрати необхідні для звіту поля зі списку за допомогою кнопок.

9. Клацнути на кнопці **Далі**, і з'явиться друга сторінка.

10. Вибрати поля, по яких ACCESS повинна робити групування записів: перше поле – першим по черзі, друге поле – другим і так далі. Вибір декількох полів приведе до збереження підгруп у межах груп.

11. Клацнути на кнопці **Далі**, і реалізується перехід до третьої сторінки. Тут можна вказати, яким образом варто поєднувати дані по полях групування.

12. У вікні, що розкрилося, вибрати тип групування, якщо вас не задовольнили попередні можливості.

13. Клацнути на кнопці **Далі**.

14. На наступних сторінках вибрати поля, по яких ACCESS повинна сортувати записи.

15. Клацнути на кнопці **Далі** та вказати, як буде виглядати звіт.

16. Вибрати тип звіту і потрібну орієнтацію сторінок.

17. Клацнувши **Далі**, ви будете на останній сторінці. Тут варто назвати звіт, потім клацнути на кнопці **Готово**, і звіт з'явиться у вікні перегляду.

Конструктор звітів

Звіт можна утворити у вікні **Конструктора звітів**. Панель інструментів **Конструктора звітів** має вид таблиці 7.1

Таблиця 7.1 – Панель інструментів конструктора звітів

Кнопка панелі інструментів	Призначення
Представлення звіту	Відображає в списку, що розкривається, три типи представлення звіту
Зберегти	Зберігає поточний макет звіту
Друк	Друкує форму, таблицю, звіт, питання
Попередній перегляд	Включає режим попереднього перегляду
Орфографія	Перевіряє орфографію обраної області чи документа
Видалити в буфер	Видаляє фрагмент документа в буфер
Копіювати в буфер	Копіює обраний об'єкт у буфер обміну
Вставити з буфера	Копіює вміст буфера обміну в документ
Копіювати формат	Копіює формат одного елемента керування в інший
Скасувати	Скасовує попередню дію
Список полів	Відкриває чи закриває вікно списку полів
Панель елементів	Відкриває чи закриває панель елементів
Сортування й групування	Відкриває чи закриває вікно Сортування й Групування
Автоформат	Застосовує готові стилі до форми чи звіту

Програма	Відкриває чи закриває вікно модуля
Властивості	Відображає вікно властивостей для обраного елемента
Побудувати	Викликає Майстра чи Будівника для обраного об'єкта
Вікно бази даних	Викликає вікно бази даних
Новий об'єкт	Створює новий об'єкт
Довідка	Надає довідку

Записи в звіті зазвичай обробляються послідовно. У залежності від макета звіту записи обробляються по-різному. Звіти поділяються на розділи. *Заголовок звіту* друкується на початку звіту, використовується на титульній сторінці.

Таблиця 7.2 – Послідовність обробки записів в звіті

<i>Верхній колонтитул.</i>	Друкується вгорі кожної сторінки.
<i>Заголовок групи.</i>	Друкується перед обробкою кожної групи.
<i>Область даних</i>	Друкується кожен запис таблиці чи динамічного набору даних запиту.
<i>Примітка групи.</i>	Друкується після обробки останнього запису групи
<i>Нижній колонтитул.</i>	Друкується у нижній частині кожної сторінки.
<i>Примітка звіту</i>	Друкується наприкінці звіту після обробки всіх записів.

Для того, щоб зв'язати звіт із запитом потрібні такі дії.

1. Натисніть клавішу **F1** для відображення вікна БД, якщо воно ще не відображено.
2. Клацніть на вкладці **Звіти**
3. У вікні БД клацніть на кнопці **Створити**. При цьому з'явиться діалогове вікно **Новий звіт**.
4. У діалоговому вікні Новий звіт виберіть опцію **Конструктор**.

5. Клацніть на стрілці списку, що розкривається, з написом *Виберіть як джерело даних таблицю чи запит*. При цьому на екрані з'явиться список всіх таблиць і запитів поточної БД.

6. Виберіть необхідний запит і клацніть на кнопці **ОК**.

7. Розгорніть до максимального розміру вікно конструктора звітів.

Після цього з'явиться порожнє вікно конструктора звітів. Зверніть увагу на три розділи: *Верхній колонтитул*, *Область даних*, *Нижній колонтитул*. Від того, що цей звіт зв'язаний із запитом, дані з запиту будуть використані при перегляді і роздруківці звіту. Поля з запиту будуть доступні для використання в макеті звіту і з'являться у вікні списку полів.

Контрольні питання

1. Чи надає звіт ще ОДИН спосіб перегляду записів з таблиць БД?
2. З якими об'єктами можна зв'язати звіт?
3. Яким образом можна згрупувати дані в звіті?
4. Чи можна в звіті вивести підсумкові значення? Як це зробити?
5. Які можливості для висновку даних забезпечує сортування?