

Системи управління базами даних. СУБД Microsoft Access

План.

1. Поняття бази даних. Призначення та класифікація СУБД
2. СУБД Microsoft Access. Інтерфейс програми. Етапи розробки баз даних
3. Етапи розробки бази даних
4. Створення таблиць. Типи даних.
5. Сортування, фільтрація та пошук даних у базі даних
6. Пов'язування даних баз даних
7. Створення форм. Введення і редагування даних за допомогою Форм
8. Запити, їх конструювання.
9. Створення звітів

Література

1. Редько М.М. і ін. *Інформатика і комп'ютерна техніка. Навчально-методичний посібник*
2. Глинський Я.М. *Практикум з інформатики.* - Львів.:Деол, 1998
3. *Методичний посібник "СУБД Microsoft Access"*

1. Поняття бази даних. Призначення та класифікація СУБД

Інформаційна система – комунікаційна система із збирання, передавання й обробки інформації про конкретний об'єкт для організації функцій управління.

Але постає питання як же ми можемо зберігати великі обсяги інформації в ПЕОМ? Адже вам вже відомо, що для опрацювання даних, повинні бути відповідні програми.

Нагадаємо, що інформація для обробки в комп'ютері називається *даними*.

На сьогоднішній лекції ми розглядатимемо не просто дані, а бази даних. Не кожен блок інформації можна вважати базою даних.

База даних - це сукупність даних, яким властива структурованість взаємопов'язаність, а також незалежність від прикладних програм.

Пояснимо, що означають названі властивості бази даних. Щоб користувач легко міг знаходити потрібну інформацію, остання має бути організована певним чином. Це стосується не лише інформації у комп'ютері, а й будь-якої інформації про об'єкти реального світу. Скажімо, зручно знаходити потрібну книгу в бібліотеці, користуючись каталогом. Легко відшукати у газеті оголошення, що вас цікавлять. Така легкість пошуку можлива завдяки тому, що дані у каталозі або у газеті мають структуру, або, інакше, *структуровані*. Всі книги описані однаково: автор, назва, видавництво, рік видання тощо. Всі оголошення з продажу розміщені по рубриках і також мають визначену структуру: короткий опис товару, ціна, телефон.

Структура бази даних складніша, ніж структура простого каталогу або набору газетних оголошень. Це зумовлено насамперед властивістю *взаємо-пов'язаності* даних у базі. Пояснимо це на такому прикладі: скажімо, ви хотіли б, крім каталожних карток, що описують кожну книгу, мати картки з інформацією про кожного автора (рік народження, літературний жанр, хобі тощо). Якби такі картки були створені, це був би

приклад взаємозалежних даних: відомості про окрему книгу, пов'язані з інформацією про автора. Цей зв'язок здійснюється через визначений параметр - прізвище автора.

Нарешті, остання з названих властивостей баз даних - це їхня *незалежність від прикладних програм*. Бази даних складаються таким чином, щоб із ними можна було працювати в різних програмних середовищах і на різних комп'ютерних платформах.

Щоб оперувати даними, які становлять базу, необхідна окрема програма - система управління базами даних.

Керівна програма, призначена для збереження, пошуку й обробки даних у базі, називається системою управління базами даних (скорочено СУБД).

Сучасні СУБД - це програмні додатки, які дозволяють виконувати різноманітні завдання. Всі існуючі системи задовольняють, як правило, таким вимогам.

Можливості маніпулювання даними (введення, вибір, вставка, відновлення, видалення тощо). Основні операції з даними виконуються під керуванням СУБД. Важливими показниками є продуктивність СУБД, витрати на збереження і використання даних, простота звертання до бази даних тощо.

Можливість пошуку і формування запитів. За допомогою запитів користувач може оперативнo одержувати різну інформацію, що зберігається в базі даних.

Забезпечення цілісності (узгодженості) даних. Під час використання даних багатьма користувачами важливо забезпечити коректність операцій, щоб запобігти порушенню узгодженості даних. Порушення узгодженості даних може призвести до їх невідновної втрати.

Забезпечення захисту і таємності. Крім захисту від некоректних дій користувачів, важливо забезпечити захист даних від несанкціонованого доступу і від апаратних збоїв. Проникнення в базу осіб, які не мають на це права, може спричинити руйнацію даних. Таємність бази даних дозволяє визначати коло осіб, що мають доступ до інформації, і порядок доступу.

Сьогодні існує багато СУБД, що відрізняються архітектурою, внутрішньою мовою програмування, операційною системою, якою вони керуються, а також іншими характеристиками. Найпопулярнішими СУБД, що встановлюються в невеликих організаціях і орієнтовані на роботу з кінцевими користувачами, є Access, FoxPro, Paradox. До складніших систем належать розподілені СУБД, що призначені для роботи з великими базами даних, розподіленими на кількох серверах (сервери можуть міститися в різних регіонах). Потужними СУБД такого типу є Oracle, Sybase, Informix.

Основою бази даних є *модель даних* — фіксована система понять і правил для представлення даних структури, стану і динаміки проблемної області в базі даних. У різний час послідовне застосування одержували *ієрархічна, мережева, таблицна і реляційна* моделі даних.

Реляційна модель даних

Практично всі сучасні СУБД використовують реляційну модель даних. Основою цієї моделі, запропонованої Е.Ф. Коддом у 1970 р., є поняття *відношення* (англійською мовою relation). Йдеться про відношення, подані у двовимірних (тобто звичайних) таблицях. Кодд показав, що таблиці можуть застосовуватися для збереження даних про об'єкти реального світу.

Двовимірні таблиці складаються із рядків, названих у термінології баз даних *записами*, і стовпців, названих полями.

Розглянемо приклад реляційної моделі даних. Скажімо, потрібно скласти базу

даних про підготовку до походу студентів. Назвемо цю базу «Похід». У ній насамперед повинна бути таблиця відомостей про учасників походу. Ця таблиця складатиметься з полів «Код учасника», «Прізвище», «Ім'я», «Група», «Адреса», «Телефон».

Назвемо цю таблицю «Учасники» (таке саме ім'я матиме і відношення, подане даною таблицею). Зверніть увагу, що поле «Код учасника» даної таблиці містить унікальні значення, які не повторюються в жодному із записів. Таке поле при введенні таблиці в базу даних матиме статус *ключового поля* (див. наступний параграф). Ключовим не може бути ні поле «Прізвище» (у таблиці є однофамільці), ні будь-яке інше поле.

База даних створюється на основі не однієї, а багатьох таблиць. Введемо до бази даних «Похід» ще одну таблицю - «Спорядження». До неї будуть занесені дані про спорядження, яке повинні взяти з собою учасники.

Записи у цій таблиці складені з полів «Код спорядження», «Код учасника» і «Спорядження». Перше поле («Код спорядження») є ключовим: для кожного запису воно набуває унікальних значень, на відміну від інших полів, у яких є дублюючі значення (наприклад, повторюються коди учасників 2, 4, 5 і спорядження «Казанок»).

Як можна використовувати інформацію наведених двох таблиць? Наприклад, нас цікавить, хто візьме в похід намет. З таблиці «Спорядження» дізнаємося, що намет під кодом 10 має взяти учасник із кодом 1. З таблиці «Учасники» бачимо, що код 1 належить Арбузову Костянтину. Отже, намет має взяти учасник походу Ігнатенко К. Цю інформацію ми отримали завдяки зв'язку таблиць «Учасники» і «Спорядження» через загальне поле «Код учасника».

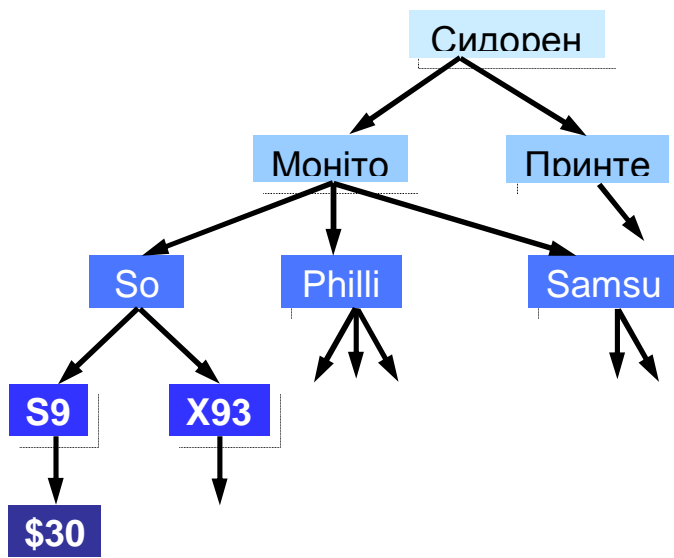
На додаток до двох таблиць, ми маємо створити й інші корисні таблиці наприклад «Продукти», у якій наведено список продуктів і визначені учасники що їх беруть. Ви можете вигадати таблиці «Графік руху», «Пам'ятні місця» «Чергування» тощо, що увійдуть у реляційну базу даних «Похід».

Ієрархічна і мережна моделі даних

При створенні баз даних, крім реляційної моделі, іноді звертаються до інших видів моделей: ієрархічної і мережної. Розглянемо коротко ці типи моделей, хоча вони мають, скоріше, історичний інтерес, оскільки основою практично всіх сучасних СУБД є реляційна модель.

Ієрархічна модель бази даних - це сукупність об'єктів різного рівня, причому об'єкти нижнього рівня підпорядковані об'єктам верхнього рівня.

Ієрархічна структура уявляється перевернутим деревом, як у прикладі на мал.

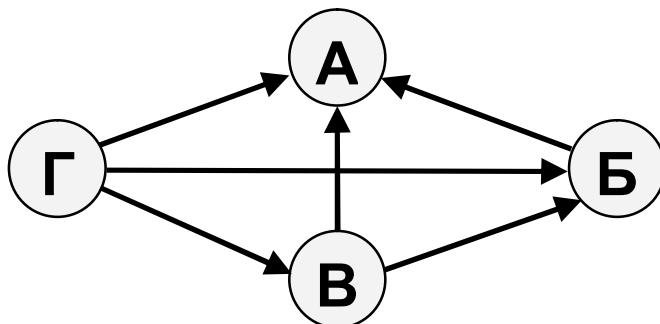


Об'єкти у цій структурі з'єднані лініями зв'язку. Зауважимо, що лінії зв'язку відповідають відношенню "один до багатьох". Тобто одному об'єкту верхнього рівня відповідає багато об'єктів нижнього рівня.

Інший тип відношень, а саме "багато до багатьох", встановлюється в мережній моделі даних.

Мережна модель бази даних, як і ієрархічна модель, уявляється сукупністю об'єктів різного рівня, однак схема зв'язків між об'єктами може бути будь-якою.

На мал. показана мережна модель, що відображає участь хлопців у різних командах.



Проектування бази даних

Створення бази даних - досить складний процес, тому не варто відразу після включення комп'ютера і запуску Microsoft Access починати її побудову. Спочатку потрібно скласти схему бази даних на папері.

Визначення мети створення бази даних

Насамперед спробуйте відповісти на запитання, для чого призначена створювана база даних, які її функції і яку інформацію вона має містити. Подумайте також, чому потрібна саме база даних і чому вас не влаштовує, наприклад, електронна таблиця або таблиця, набрана у текстовому процесорі.

До бази даних звертаються, коли доводиться описувати складні об'єкти з багатьма властивостями-атрибутами, коли поставлене завдання автоматизації введення даних і виводу інформації, яка вас цікавить, коли необхідно забезпечити процес введення і відновлення даних, тобто забезпечити цілісність інформації.

Наприклад ви створюєте базу даних щодо музичних груп і хочете, щоб у ній містилися такі дані:

- про твори, що виконуються групою;
- про випущені альбоми;
- про солістів групи, їхні біографії, інструменти тощо;
- про історію групи, її досягнення, успішні концерти.

Припустимо, що створювана база даних потрібна для отримання такої інформації: (

- які нові пісні були виконані цією групою в поточному році;
- якій групі належить більше альбомів;
- які групи виконують пісні даного автора;
- яка фірма випустила найбільший тираж альбому даної групи.

Можливі також багато інших запитань, відповіді на які ви зможете отримати •» даних, вміщених у базі.

Визначення таблиць, що входять до бази даних

На наступному етапі створення бази даних необхідно визначити основні теми

таблиць, а також їхню структуру. Для бази даних про музичні групи ви можете скласти таблиці «Групи», «Альбоми», «Солісти» тощо.

Надалі розглядатимемо створення бази даних для туристичної фірми, що розповсюджує путівки. У таблиці «Путівки» зберігатимуться відомості про запропоновані путівки, у таблиці «Замовлення» - відомості про замовлення путівок, а в таблиці «Клієнти» - дані про постійних клієнтів фірми.

Задання структури таблиць

Для створення ефективної бази даних важливо правильно визначити структуру таблиць, тобто склад полів. На цьому етапі слід пам'ятати:

Інформація в таблицях не повинна дублюватися. Якщо зазначені відомості зберігатимуться в одному полі однієї таблиці, то і змінювати їх у разі потреби доведеться лише в одному місці.

Бажано, щоб кожна таблиця містила інформацію лише на одну тему. Наприклад, у базі даних «Похід» (див. попередній параграф) відомості про учасників і про спорядження зберігаються в різних таблицях. Якщо видалити запис про будь-яке спорядження, таблиця учасників не постраждає.

Не рекомендується включати до таблиці дані, що утворюються внаслідок обчислень.

Інформацію про об'єкт бажано розбивати на мінімальні одиниці. Наприклад, поштові реквізити клієнта краще зазначити в окремих полях: «Індекс», «Місто» й «Адреса».

Задання ключа і визначення зв'язків між таблицями

Для того щоб пов'язати дані різних таблиць, у кожній таблиці має бути ключове поле. Значення цього поля дозволить однозначно обрати потрібний запис у таблиці, а також коректно встановити зв'язки між таблицями.

Введення даних і аналіз бази даних

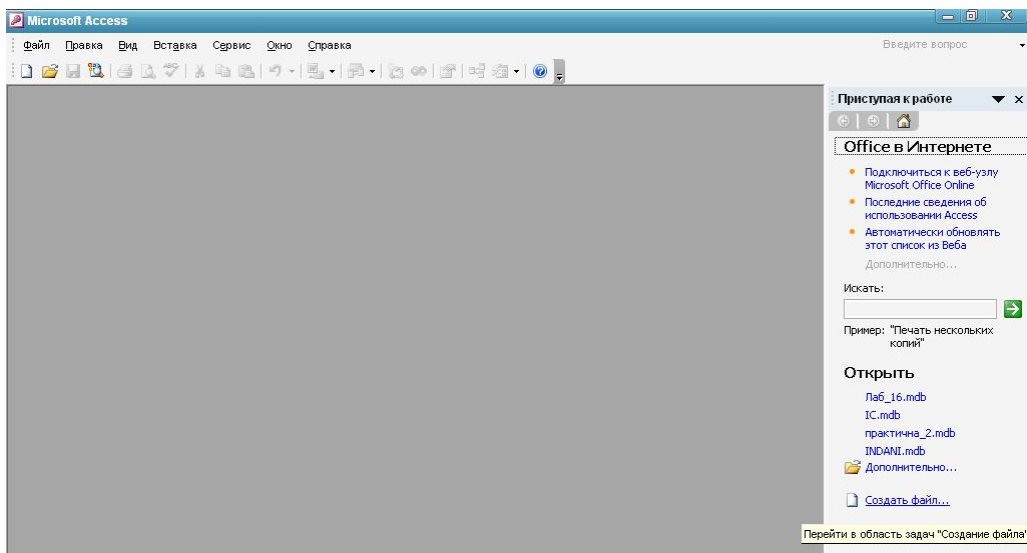
Після задання структури таблиць і встановлення зв'язків між ними слід зробити кілька записів у таблиці. На прикладі цих записів ви зможете побачити, наскільки правильно створено базу даних і чи відповідає вона поставленим вимогам. Потім можете приступати до введення даних, що залишилися у таблиці.

База даних створюється для того, щоб працювати з поміщеною у ній інформацією.

2. СУБД Microsoft Access. Інтерфейс програми.

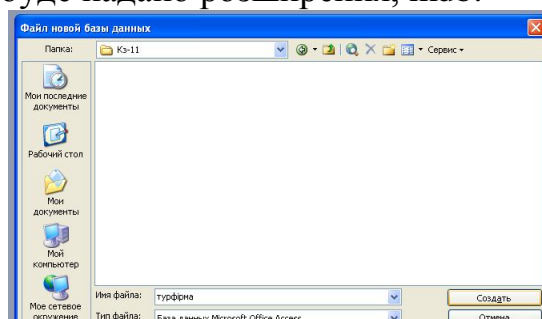
Access - це повноцінна реляційна СУБД, у якій може працювати як пересічний користувач, котрий не володіє основами програмування, так і фахівець-розробник, який створює додатки на мові Visual Basic for Applications (скоромно VBA). Популярність СУБД Access при вивченні баз даних зумовлена тим, що ця система є найпростішою для засвоєння початківцями.

Робота з базою даних починається з запуску СУБД. Для запуску Access натисніть кнопку Пуск на панелі задач і оберіть у головному меню команду Всі програми - Microsoft Access. У відповідь з'явиться діалогове вікно Microsoft Access



Як було зазначено, роботу в СУБД Access ми будемо вивчати на прикладі створення бази даних, яка описує діяльність якоїсь туристичної фірми «Подорож». Нехай ця фірма приймає заявки на туристичні путівки і продає їх.

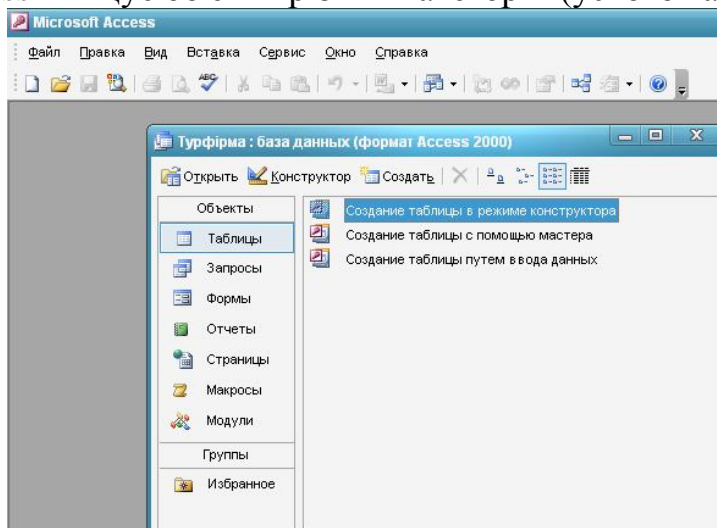
Отже, у первинному вікні ви обрали опцію створення нової БД і натиснули кнопку ОК. У наступному стандартному діалозі вам потрібно ввести ім'я файла для нової бази даних (у нашому прикладі це «Фірма «Подорож»») і позначити папку, у якій цей файл розміститься. Йому буде надано розширення, mdb.



Після задання імені і розташуван файла клацніть по кнопці Создать, і на екрані з'явиться вікно бази даних

Об'єкти бази даних

База даних в Access вміщує об'єкти різних категорій (усього таких категорій 7):



Кожній із категорій відповідає своя вкладка вікна бази даних Таблиці, Запити, Форми, Звіти, Макроси, Модулі.

Таблиці. Це основна категорія об'єктів у реляційній СУБД, оскільки вся інформація зберігається в базі даних у вигляді таблиць. Кожна таблиця складається з записів (рядків) і з полів (стовпців). Робота з таблицею виконується в двох основних режимах: у режимі конструктора й у режимі таблиці.

Запити. Об'єкти цього типу призначені для отримання даних з однієї або кількох таблиць. Відбір потрібних відомостей відбувається на основі сформульованих критеріїв. Фактично за допомогою запитів створюються нові таблиці, у яких використовуються дані з існуючих таблиць.

Форми. Цей тип об'єктів використовується в основному для зручного введення даних. Форма є ніби бланком, який потрібно заповнити. Заповнити такий бланк зможе навіть початківець. Позитивним є те, що форми запобігають безпосередньому внесенню змін у таблиці.

Звіти. Об'єкти-звіти відображають дані так, що їх зручно переглядати. На основі звіту може бути створений документ, що буде роздрукований або включений у документ іншого додатка.

Макроси. Макросами називаються «макрокоманди», що запускаються простим натисканням кількох клавіш і можуть виконувати такі дії, як відкриття таблиць і форм, виконання опцій меню, керування вікнами тощо. Користувач може створювати свої макроси для послідовностей операцій, які часто застосовуються.

Модулі. Цей тип об'єктів є програмними модулями, написаними мовою VBA. Модулі - це процедури для обробки подій або виконання обчислень. Розбиття на модулі полегшує процес створення і налаштування програми.

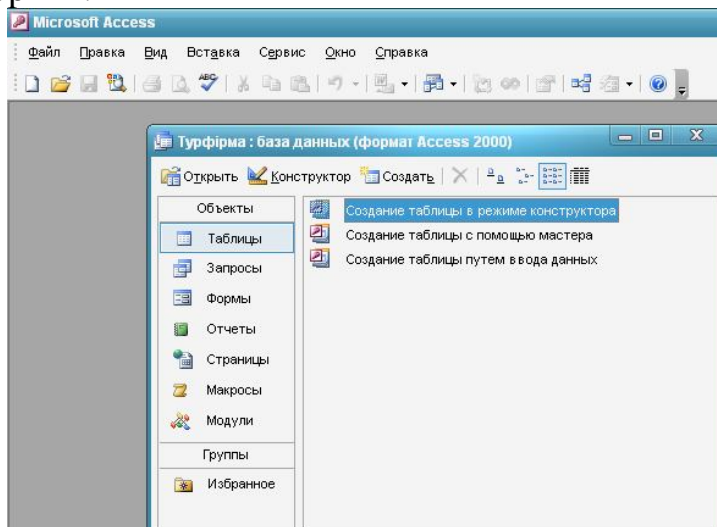
3. Створення таблиць. Типи даних.

Основою реляційної бази даних є таблиці. Тому з їх побудови доречно почати створення бази даних.

Режими створення таблиць

Таблиці бази даних описують певні теми. Однак, перед тим як почати створення таблиці, потрібно добре уявити її структуру, тобто склад полів, типи даних і властивості полів.

Для створення таблиці в додатку Access відкрийте вікно бази даних. Якщо воно ще не відкрито, натисніть клавішу F11. Перейдіть у цьому вікні на вкладку Таблиці і клацніть по кнопці Створити.

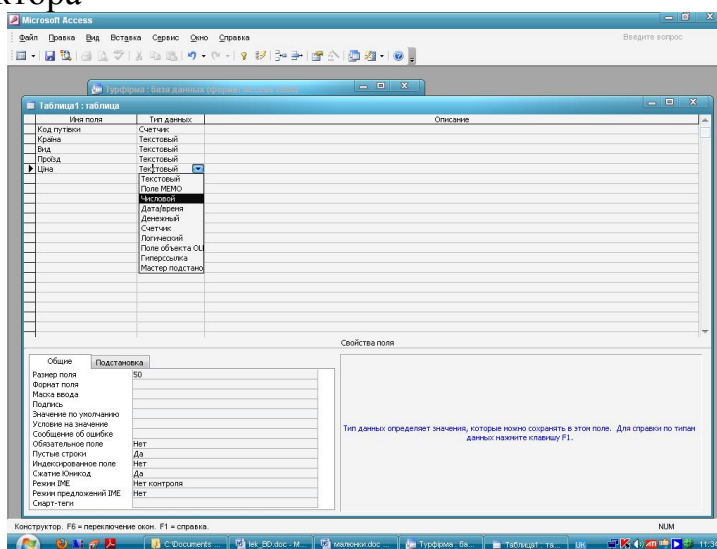


У наступному діалозі вам буде запропоновано обрати режим створення таблиці. У Access передбачено кілька таких режимів, які обираються зі списку діалогу Нова таблиця:

- Режим таблиці - застосовується для заповнення і редагування полів таблиці.
- Конструктор - режим для задання структури таблиці, тобто імен полів і типів даних.
- Майстер таблиць - створення таблиці за допомогою програми майстра.
- Імпорт таблиць - створення таблиці шляхом уведення даних із зовнішнього файла (іншої бази даних, електронних таблиць тощо).

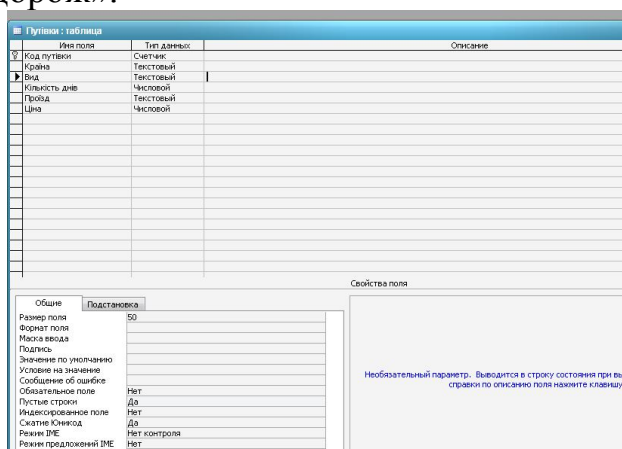
Задання структури таблиці

Припустимо, що ви запустили процедуру створення нової таблиці (тобто у вікні бази даних на вкладці Таблиці була натиснута кнопка Створити). Далі в діалозі Нова таблиця виділіть у списку позицію Конструктор і клацніть по кнопці ОК, після чого з'явиться вікно конструктора



У цьому вікні потрібно визначити склад таблиці, заповнивши для кожного поля таблиці три колонки: Ім'я поля, Тип даних і Опис. У першій колонці задається ім'я поля майбутньої таблиці, у другій - вказується його тип, а в третій - вводиться інформація про призначення поля. Кількість записів у вікні конструктора повинно збігатися з кількістю полів у створюваній таблиці.

Складемо таблицю, у якій будуть вміщені відомості про путівки, запропоновані туристичною фірмою «Подорож».



Зверніть увагу, що активне поле (у ньому в даний момент розміщений курсор) позначено зліва індикатором - трикутною стрілкою. Властивості активного поля названі в нижній частині діалогу

Якщо потрібно видалити будь-який рядок з таблиці, клацніть мишею по цьому рядку у вікні конструктора, а потім виконайте команду меню Правка -Знищити. Для вставки нового рядка потрібно активізувати поле нижче рядка, що вставляється, і виконати команду меню Вставка - Рядка.

Типи даних

Тип даних визначається значеннями, які передбачається вводити до поля (наприклад, текст або число). Якщо надалі доведеться задати інший тип даних, це можна виконати у режимі конструктора.

У Access передбачено такі типи даних.

Текстовий - для введення тексту завдовжки до 255 символів. Цей тип даних установлюється за умовчанням.

Поле МЕМО - для в'єдення заміток або довгих описів (можливе введення до 64 000 символів).

Числовий - для введення числових даних, для яких виділяється 1, 2 або 4 байти.

Дата/час - для введення дати і часу, для яких передбачено 8 байтів,,

Грошовий - використовується для роботи з грошовими одиницями. Цей тип даних займає 8 байтів і допускає до 15 символів у цілій частині числа і 4 – у дробовій. Використання грошового типу запобігає помилці округлень під час обчислень.

Лічильник - для введення числа, що автоматично збільшується на одиницю при додаванні до таблиці нового запису. Дані цього типу займають 4 байти.

Логічний - для збереження логічного значення Істина чи Хиба. Таке поле займає 1 біт.

Об'єкти OLE – для збереження в таблиці OLE об'єктів (наприклад, малюнків, звуків, документів Word тощо.). Розмір збережених об'єктів OLE обмежується лише I ємністю диска.

Гіперпосилання - служить для запису до таблиці гіперпосилань (шляху URL)

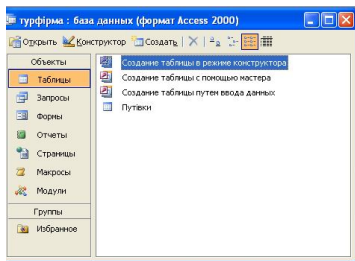
У списку **Тип даних** конструктора є також позиція **Майстер підстановок**, за допомогою якої обираються значення з іншої таблиці або зі списку значень.

Зберігання таблиці

Отже, структура таблиці задана, і ви можете закрити вікно конструктора таблиці клацанням по кнопці закриття вікна X, Якщо таблиця нова і раніше не зберігалася, з'явиться запит щодо того, чи варто зберігати структуру таблиці. Клацніть по кнопці Так і в діалозі Збереження введіть ім'я таблиці (воно може містити будь-які символи, крім крапки, знака оклику і кутових дужок), після цього натисніть кнопку ОК.



У розглянутому нами прикладі таблицю збережено під ім'ям «Путівки». Після зберігання у вікні бази даних з'явиться піктограма створеної таблиці.



4. Сортування фільтрація та пошук даних

Сортування записів

При введенні даних у таблиці або форми записи розташовуються у порядку введення. Це не завжди зручно при перегляді введеної інформації. Бажано згрупувати й упорядкувати інформацію, щоб вона була змістовною й у ній було легко орієнтуватися. Наприклад, у таблиці «Замовлення» доречно виділити товар з найбільшою купівельною спроможністю, помістивши відповідні записи на початок таблиці, а в таблиці «Клієнти» доцільно розподілити клієнтів за регіонами. Цього легко досягти сортуванням записів, яке виконується у такий спосіб.

Відкрийте таблицю і перейдіть у Режим таблиць. Активізуйте поле таблиці, за яким відбудуватиметься сортування.

Клацніть по кнопці (сортування за зростанням) або по кнопці (сортування за



спаданням).

Код путівки	Країна	Вид	Кількість днів	Провід	Ціна
20	Україна	екскурсії	3	автобус	650
19	Україна	відпочинок	5	автобус	890
4	Болгарія	екскурсії	2	авіа	1450
10	Чехія	екскурсії	2	автобус	1650
9	Чехія	екскурсії	3	авіа	1750
1	Болгарія	відпочинок	3	авіа	1900
15	Туреччина	екскурсії	3	авіа	2200
11	Чехія	відпочинок	10	авіа	2250
14	Чехія	лукування	10	автобус	2400
16	Туреччина	екскурсії	2	автобус	2700
12	Чехія	відпочинок	10	автобус	2800
18	Туреччина	відпочинок	10	автобус	2900
17	Туреччина	відпочинок	12	авіа	3300
2	Болгарія	відпочинок	10	автобус	3600
7	Греція	екскурсії	3	авіа	3670
6	Греція	відпочинок	5	авіа	3600
8	Греція	екскурсії	4	автобус	3800
5	Греція	відпочинок	10	авіа	4520
13	Чехія	лукування	14	авіа	4880
3	Болгарія	лукування	14	авіа	4950
(Счетчик)			0		0

Прості фільтри даних

Сортування даних дозволяє упорядковувати їх, однак воно не скорочує кількості записів у таблиці, які доводиться переглядати користувачу. Зручними засобами для перегляду записів є *фільтри*.

Припустимо, що вам потрібно переглянути відомості про путівки до Туреччині. Відкрийте таблицю «Путівки» у режимі таблиці. Установіть курсор *i* комірку «Ту

реччина» і натисніть кнопку



Фільтр по виділеному на панелі інструментів. На екрані з'являться записи про путівки до цієї країни


	Код путівки	Країна	Вид	Кількість днів	Проїзд	Ціна
▶ +	15	Туреччина	екскурсії	3	авіа	2200
+	16	Туреччина	екскурсії	2	автобус	2700
+	17	Туреччина	відпочинок	12	авіа	3300
+	18	Туреччина	відпочинок	10	автобус	2900
*	(Счетчик)			0		0

Цей простий фільтр дозволяє відфільтрувати записи лише за однією умовою, що накладається на поле. Для виконання фільтрації ще за однією умовою (наприклад, за полем «Проїзд») клацніть по комірці з потрібним значенням (припустимо, авіа). Далі натисніть кнопку Фільтр по виділенному - і ви одержите усього два записи, що містять слово «Туреччина» у полі «Країна» і слово «авіа» у полі «Проїзд».

	Код путівки	Країна	Вид	Кількість днів	Проїзд	Ціна
▶ +	17	Туреччина	відпочинок	12	авіа	3300
+	15	Туреччина	екскурсії	3	авіа	2200
*	(Счетчик)			0		0

Фільтрація даних стосується лише записів на екран, сама таблиця залишається без змін. Для скасування фільтра й відновлення попереднього вигляду таблиці клацніть



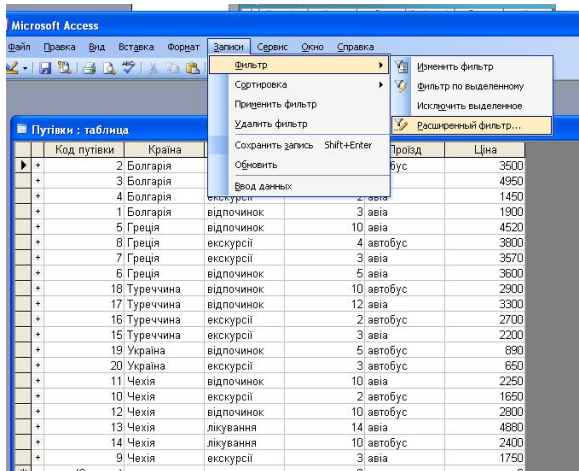
по кнопці Знищити фільтр: , розміщеній на панелі інструментів.

При установці фільтра, що містить кілька умов, зручно скористатися дещо іншим інструментом. Відкривши таблицю, у якій збираєтеся виконати фільтрацію, клацніть по кнопці Змінити фільтр на панелі інструментів. На екран буде виведено лише один рядок таблиці. У полі «Країна» з'явиться трикутна стрілка розкривного списку. За допомогою цього списку задайте одну умову. Клацніть по іншому полю й аналогічно задайте другу умову (наприклад, вид путівки). Задайте всі необхідні умови і натисніть кнопку Применение фільтра.

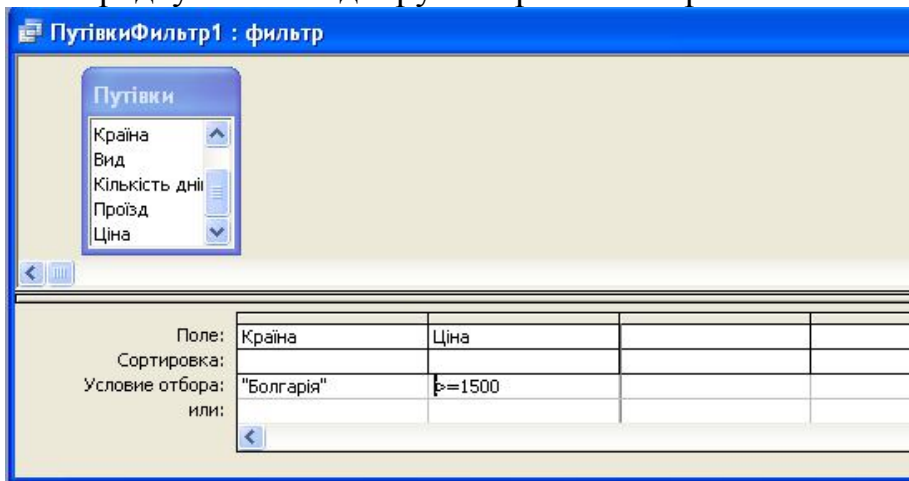
Розширений фільтр

Для задання складних умов фільтрації використовується розширений фільтр. Припустимо, ви хочете переглянути у таблиці «Путівки» записи про путівки до Болгарії вартістю не більше 1500 грн. Для цього зручно створити розширений фільтр.

Відкрийте таблицю «Путівки» бази даних «Тур фірма» «Подорожж» і перейдіть до режиму таблиці (клацання по кнопці Вид). Оберіть команду меню Записи - Фільтр – Розширений фільтр. У діалозі



Клацніть по першій комірці рядка Поле й у розкритому списку оберіть позицію «Країна». У рядку Умова відбору наберіть «Болгарія».



У сусідній комірці рядка Поле оберіть зі списку позицію «Ціна», а в умові вибору зазначте «<=1500». Клацніть по кнопці Применение фильтра (або задайте команду Фільтр -Применить фільтр). На екрані з'являться відфільтровані відомості про путівки до Болгарії в заданому інтервалі вартості.

Путівки : таблиця						
	Код путівки	Країна	Вид	Кількість днів	Проїзд	Ціна
+	3	Болгарія	лікування	14	авіа	4950
+	2	Болгарія	відпочинок	10	автобус	3500
+	1	Болгарія	відпочинок	3	авіа	1900
*	(Счетчик)			0		0

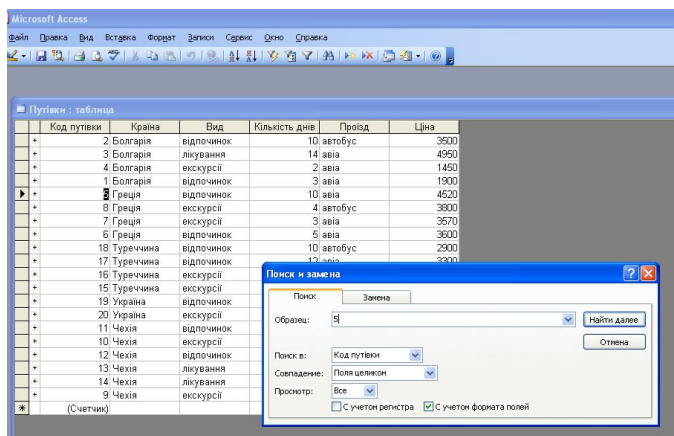
Пошук запису

Якщо таблиці баз даних великі і важко знайти той чи інший запис, можна вернутися за допомогою до засобу Пошук. Працюючи з таблицею (наприклад, «Путівки»), перейдіть у режим таблиці (якщо він не встановлений).

Клацніть мишею в тому полі, значення якого буде використано для пошуку (наприклад, «Код»). Натисніть на кнопку Знайти на панелі інструментів.

У діалозі введіть значення поля (наприклад, число 5).

Запустіть процедуру пошуку, клацнувши по кнопці Найти.



Якщо необхідно виконати пошук, враховуючи регістр слова або формат полів, установіть перед натисканням кнопки **Найти** відповідні перемикачі в діалозі пошуку. У цьому самому діалозі є розкривні списки **Просмотр** і **Совпадение**, що дозволяють задати зону пошуку (все, вверху, вниз) і ознаку збігу (поля ціликом, з будь-якою частиною поля, з початку поля).

5. Пов'язування даних баз даних Відношення між таблицями

Розглянемо можливі відношення між таблицями бази даних. Вони бувають таких типів: "один до одного", "один до багатьох", "багато до одного" і "багато до багатьох". Найпоширенішим у таблицях реляційних баз даних є відношення "один до багатьох".

Відношення "один до багатьох" означає, що одному запису таблиці відповідають кілька записів в іншій таблиці.

Розглянемо створені таблиці для бази даних «Фірма» «Подорож». Туристична фірма продає путівки, причому однакові путівки вона може продати кільком клієнтам. Тому одному запису в таблиці «Путівки» можуть відповідати декілька записів у таблиці «Замовлення»

Путівки : таблиця						
	Код путівки	Країна	Вид	Кількість днів	Проїзд	Ціна
	1	Болгарія	відпочинок	3	авіа	1900
	2	Болгарія	відпочинок	10	автобус	3500
	3	Болгарія	лікування	14	авіа	4950
	4	Болгарія	екскурсії	2	авіа	1450
	5	Греція	відпочинок	10	авіа	4520
	6	Греція	відпочинок	5	авіа	3600
	7	Греція	екскурсії	3	авіа	3570
	8	Греція	екскурсії	4	автобус	3600
	9	Чехія	екскурсії	3	авіа	1750
	10	Чехія	екскурсії	2	автобус	1650
	11	Чехія	відпочинок	10	авіа	2250
	12	Чехія	відпочинок	10	автобус	2600
	13	Чехія	лікування	14	авіа	4680
	14	Чехія	лікування	10	автобус	2400
	15	Туреччина	екскурсії	3	авіа	2200
	16	Туреччина	екскурсії	2	автобус	2700
	17	Туреччина	відпочинок	12	авіа	3300
	18	Туреччина	відпочинок	10	автобус	2900
	*	(Счетчик)		0		0

Замовлення : таблиця					
	№ замовлення	Дата	Код клієнта	Код путівки	Кількість
	1	01.01.2010	1	5	4
	2	03.03.2010	3	2	2
	3	03.03.2010	4	3	1
	4	04.03.2010	6	1	3
	5	04.03.2010	4	20	12
	6	09.03.2010	6	3	2
	7	09.03.2010	3	3	10
	8	16.03.2010	5	3	2
	*	(Счетчик)		0	0

Сторона "один" у відношенні "один до багатьох" називається *головною* таблицею. Сторона "багато" у цьому самому відношенні називається *зв'язаною* таблицею. У прикладі зв'язаних таблиць «Путівки» і «Замовлення» ми бачимо, що ключове поле головної таблиці зв'язується з відповідним полем зв'язаної таблиці.

Тип відношень "один до одного" трапляється в таблицях баз даних рідше ніж "один до багатьох".

У відношенні «один до одного» одному запису в головній таблиці відповідає один запис у зв'язаній таблиці.

У наведених нами раніше таблицях немає відношення «один до одного». Щоб проілюструвати це відношення, ми могли б додатково до таблиці «Клієнти» створити таблицю «Банківські дані», у якій містилися б банківські реквізити Клієнтів. Наприклад, у полі «Рахунок» було б зазначено номери розрахункових рахунків клієнтів. Оскільки рахунок кожного клієнта унікальний, між полем «Код клієнта» таблиці «Клієнти» і полем «Рахунок» таблиці «Банківські дані» існувало б відношення «один до одного».

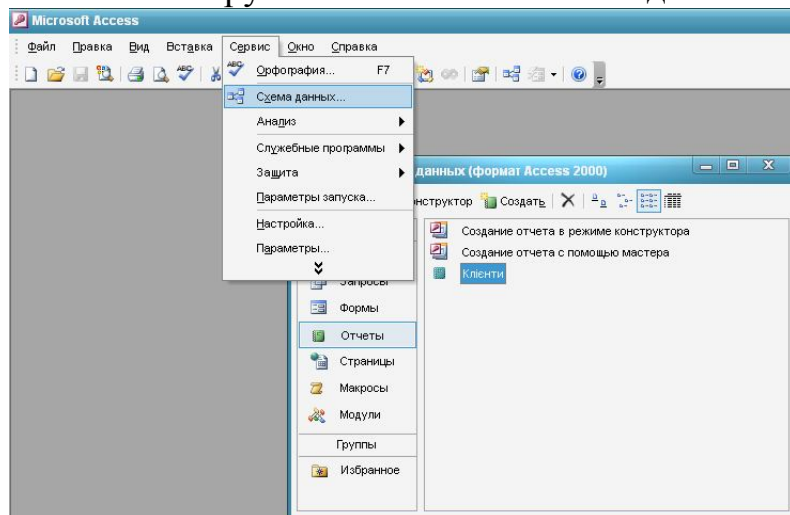
Встановлення зв'язків між таблицями

Отже, між полями таблиць можуть існувати певні відношення.

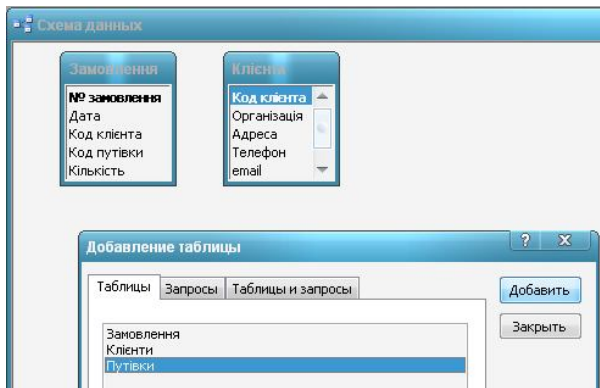
Розглянемо встановлення зв'язків на прикладі таблиць «Путівки», «Клієнти» і «Замовлення». Починаючи зв'язування таблиць, переконайтеся, що всі таблиці і форми закриті. Потім переключіться у вікно бази даних. Клацніть по кнопці *Схема даних*



на панелі інструментів вікна Access або дайте команду Сервіс – Схема даних



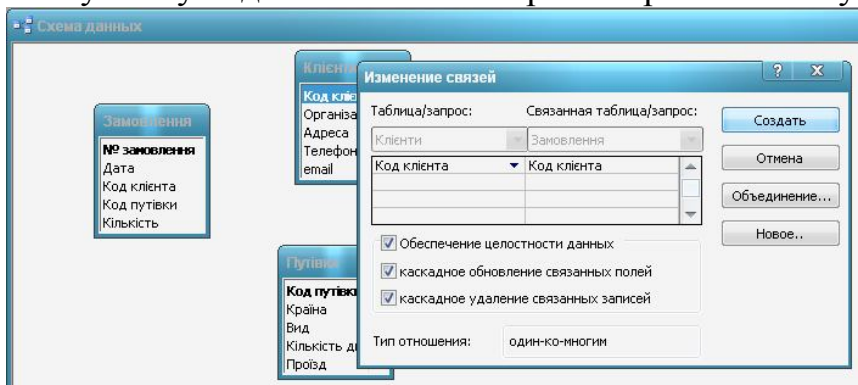
у відповідь відкриється однойменне діалогове вікно. Водночас з вікном *Схема данных* з'явиться діалог *Добавление таблицы*



Однак цей діалог може і не з'явитися, якщо раніше були створені зв'язки між таблицями (вікно Схема данных не порожнє). У цьому разі потрібно очистити вікно від зв'язків, клацнувши по кнопці Очистить макет. Потім натисніть кнопку Добавить таблицу на панелі інструментів. За умовчанням в діалозі Добавление таблицы буде відкрита вкладка Таблицы. Виділіть у списку таблицю «Замовлення» і натисніть кнопку Добавить. У вікні Схема данных з'явиться список полів таблиці «Замовлення». Аналогічно виведіть списки полів таблиць «Клієнти» і «Путівки» у вікно Схема данных, після чого закрийте діалог Добавление таблицы. Для зв'язування полів «Код путівки» оберіть це і поле у головній таблиці «Путівки» і перетягніть його мишею до зв'язаної таблиці «Замовлення».

Зазначимо, що напрямок перетягування поля завжди повинен бути «від головної таблиці до зв'язаної».

У діалозі Связи установіть перемикач Обеспечение целостности данных. Це дозволить уникнути деяких помилок при створенні експлуатації бази даних.



Клацніть по кнопці Создать, і встановлений зв'язок буде відображений у вікні Схема данных.

Зв'язок показаний лінією, позначеною цифрою 1 і символом нескінченності ∞ , що означає відношення «один до багатьох». Аналогічно створіть зв'язок між полями «Код клієнта» таблиць «Клієнти» і «Замовлення». Ви одержите схему зв'язку, показану на малюнку.



Макет зв'язків

Зручним для вас способом (наприклад, щоб лінії зв'язків не перетиналися) розмістіть списки полів у вікні Схема данных. Списки можна перетягувати, захопивши мишею заголовок списку. Розташування списків у вікні називається *макетом зв'язків*. Після цього можете закрити вікно Схема данных. Програма виведе на екран запит щодо того, чи потрібно зберігати макет зв'язків-Клацніть по кнопці Да. Якщо ви відповісте Нет, то збережуться лише створені зв'язки, а не компоновання списків.

Зверніть увагу, що в розглянутих випадках ми зв'язували ключове поле (позначене напівжирним шрифтом у списку полів) головної таблиці з відповідним полем зв'язаної таблиці. Поле зв'язаної таблиці називають *полем зовнішнього ключа*.

Для видалення будь-якого зв'язку не потрібно знову створювати макет зв'язав. Виділіть зв'язок клацанням миші й натисніть клавішу Delete. Після появи запиту на видалення зв'язку клацніть по кнопці Да.

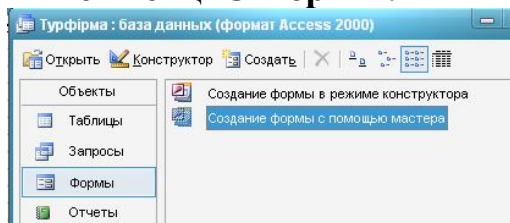
6. Створення форм. Введення даних за допомогою форм

До баз даних інформація звичайно вводиться за допомогою форм, а зберігається у вигляді таблиць.

Форма - це об'єкт бази даних, призначений для введення і відображення інформації. Форма обов'язково містить елементи (поля), до яких користувач вводить дані.

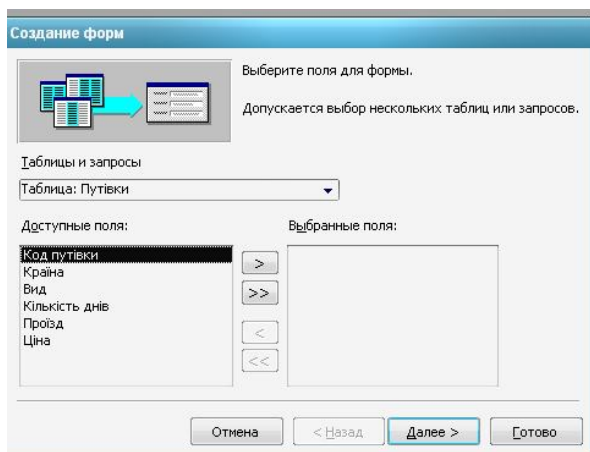
Користувач Access може створити форму самостійно, але краще звернутися до послуг програми-майстра. Це прискорить процес побудови форми, оскільки майстер виконає всю основну роботу.

Відкрийте вікно БД (натисніть клавішу F11). Якщо у вас відкрито вікно таблиці, яка буде основою форми, закрийте його. Перейдіть на вкладку **Форми** У вікні БД і клацніть по кнопці **Створити**.

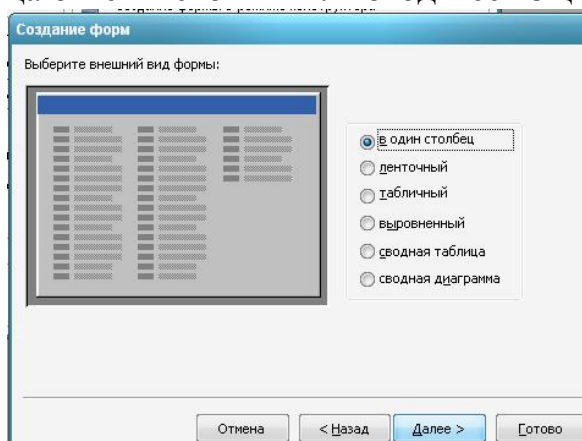


У діалозі **Нова форма** вкажіть режим Мастер форм і У розкритому списку оберіть таблицю, для якої створюватиметься форма.

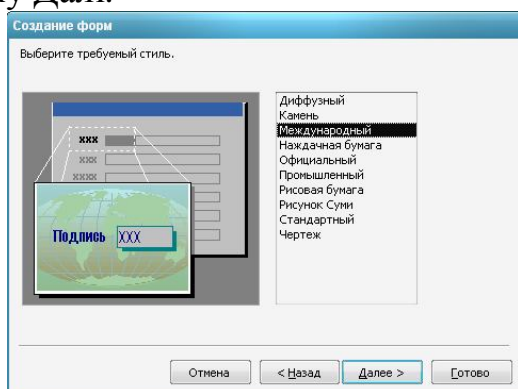
- у першому діалозі майстра створення форм, вкажіть поля, які будуть присутні у формі



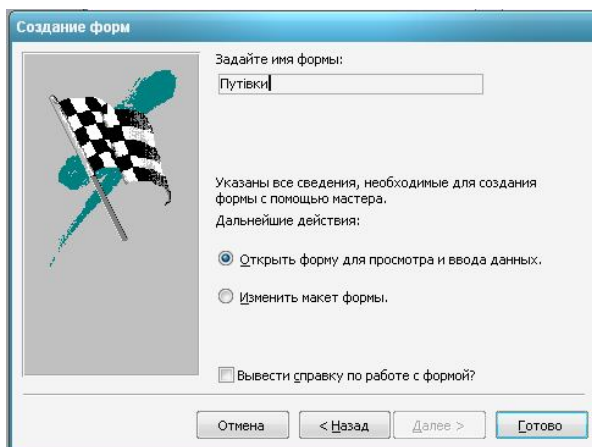
Після натискання на кнопку Далі відкриється наступний діалог для вибору вигляду форми. За умовчанням пропонується форма, в якій поля введення розміщуються в стовпчик. Погодьтеся з цією пропозицією і клацніть по кнопці Далі.



У наступному діалозі вам потрібно вибрати стиль оформлення (фон, кольори полів і написів форми). У поданому списку стилів зазначте потрібний вам і натисніть кнопку Далі.

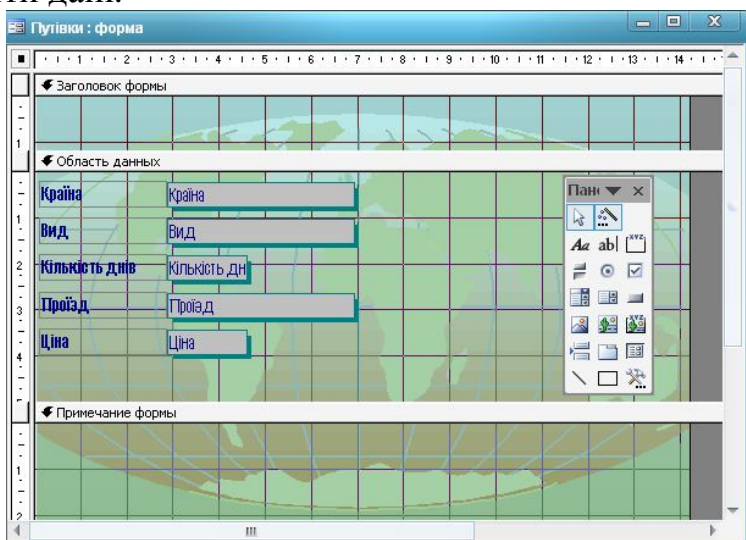


У новому діалозі введіть ім'я форми (за умовчанням пропонується ім'я таблиці-джерела).

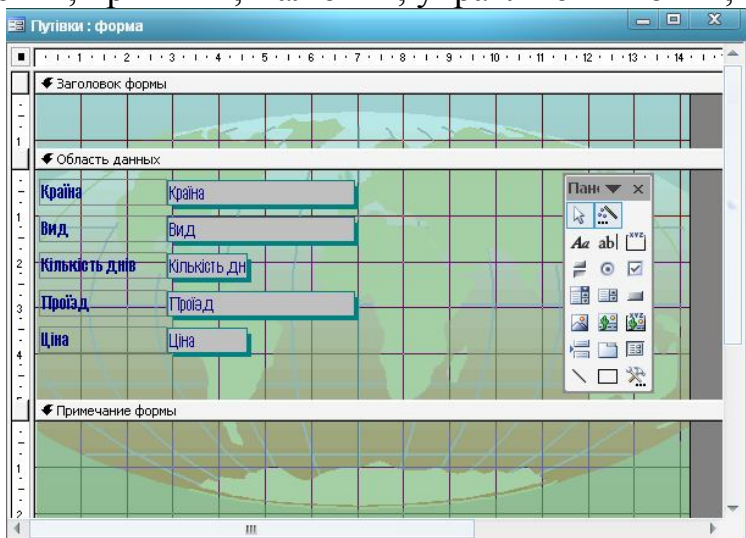


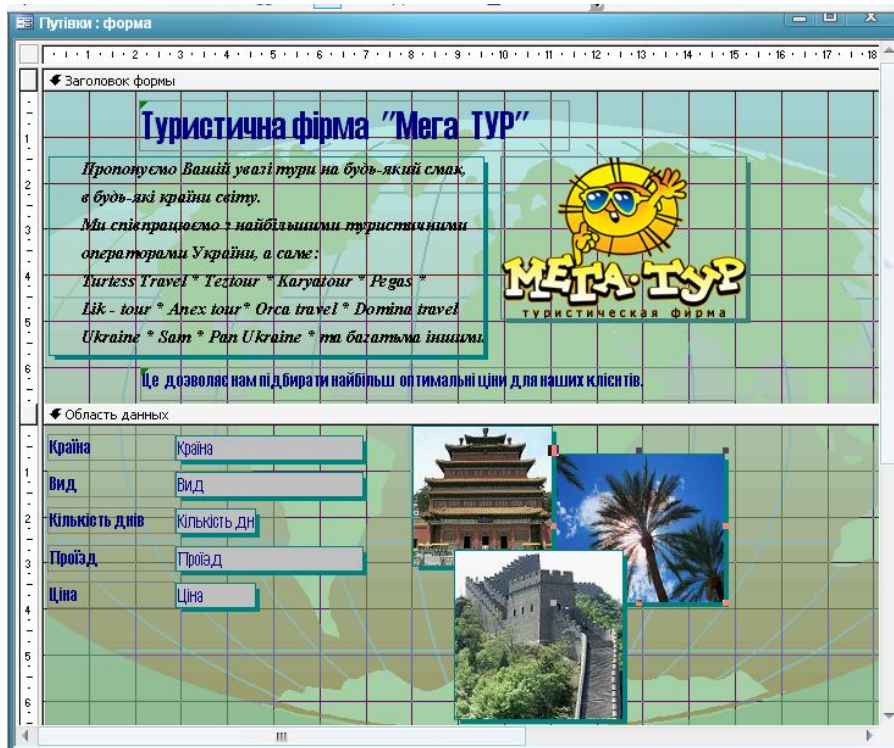
Переконайтеся також, що встановлений перемикач **Открытие формы для просмотра или ввода данных** і натисніть кнопку **Готово**.

Внаслідок виконаних дій на екрані з'явиться вікно форми, до якої можна відразу вводити дані.



За допомогою конструктора форм можна удосконалити форму, додати до неї заголовки, примітки, малюнки, управляючі кнопки, тощо

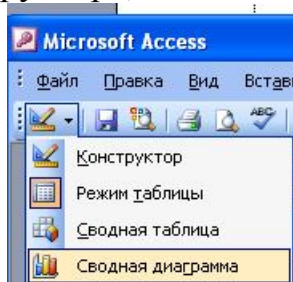




Введення даних до таблиці

Після задання структури й імені таблиці ви можете безпосередньо ввести дані до неї. Для цього потрібно перейти в Режим таблиці. Цей режим устанавлюється подвійним клацанням мишею по піктограмі таблиці (або виділенням піктограми таблиці і натисканням кнопки Відкрити після відкриття бази даних-На екрані з'явиться вікно із зображенням рядків таблиці .

Порівняйте вікно із вікном конструктора таблиць. До речі, найзручніше переключатися між режимами конструктора і таблиці за допомогою кнопки Вид на панелі інструментів. Ця кнопка має вигляд Режим таблиці, якщо включений режим конструктора, і вигляд Конструктор, якщо активним є вікно таблиці.



Коли ви заповните перший рядок таблиці, в комірці «Код путівки» з'явиться 1, а напис (Счетчик) автоматично переміститься в наступний рядок.

У процесі заповнення таблиці ви можете переміщатися між різними полями і рядками за допомогою клавіш керування курсором, а також клавіші Tab (клавіші Shift+Tab забезпечують переміщення в зворотному напрямку).

Заповнюючи таблиці (а також форми), ви можете використовувати звичайні прийоми редагування, відомі вам по роботі в програмах Блокнот і Word (прийоми вставки і видалення символів, використання буфера обміну тощо).

Введення даних за допомогою форми

Найзручнішим способом введення записів у базу даних є заповнення форм. Припустимо, що ви вже створили форму «Путівки», як було описано в пункті «Форми

і їхнє створення» попереднього параграфа. Відкрийте вікно бази даних і перейдіть на вкладку Форми. Двічі клацніть мишею по піктограмі «Путівки», після чого відкриється вікно форми.

Зверніть вагу, що у нижній частині форми розміщені кнопки панелі переходу, які дозволяють переміщатися по записах.



На панелі переходу індикатор записів відображає номер поточного запису. Кнопки на цій панелі дозволяють переходити до наступного або попереднього записів, а також у кінець або на Початок набору записів. У полі індикатора записів можна також зазначити Номер запису, до якого бажаєте перейти.

Щоб додати новий запис, потрібно клацнути по кнопці Новий запис Панелі переходу і потім ввести дані в поля форми.

Введені дані будуть розміщені в таблиці «Путівки» після закриття вікна Форми.

Таблиці «Клієнти» і «Замовлення»

Для ефективної роботи нашої бази даних необхідно побудувати ще дві таблиці бази даних «Фірма «Подорожж». В одній із них будуть відомості про постійних клієнтів фірми, які купують путівки (таблиця «Клієнти»), а в другій - відомості про зроблені замовлення (таблиця «Замовлення»).

Клієнти : таблиця		
	Имя поля	Тип данных
🔍	Код клієнта	Счетчик
	Організація	Текстовый
	Адреса	Текстовый
	Телефон	Текстовый
	email	Текстовый
▶	web сайт	Текстовый

Замовлення : таблиця		
	Имя поля	Тип данных
🔍	№ замовлення	Счетчик
	Дата	Дата/время
	Код клієнта	Числовой
	Код путівки	Числовой
	Кількість	Числовой

Доступ до інформації в базі даних забезпечується таким інструментом, як *запити*. Запити дозволяють відібрати дані, що містяться в різних таблицях бази, а

також виконати відбір відповідно до заданих умов (наприклад, список товарів, не дорожчих заданої ціни, дані про клієнтів у певному регіоні тощо). Тобто запити нагадують розглянуті раніше фільтри, однак запити - гнучкіший інструмент доступу до інформації. Так, за допомогою запитів можна не лише добути інформацію з БД, а й формувати нові поля, яких немає у первинних таблицях. У запитах можна обробляти початкові дані (знаходження середнього, максимального значення, підсумовування тощо).

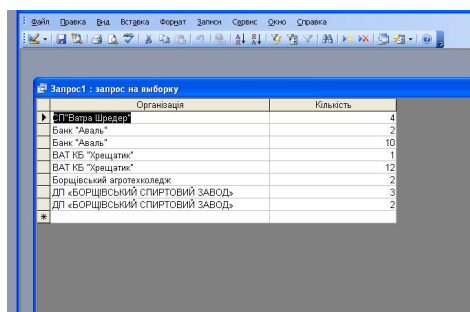
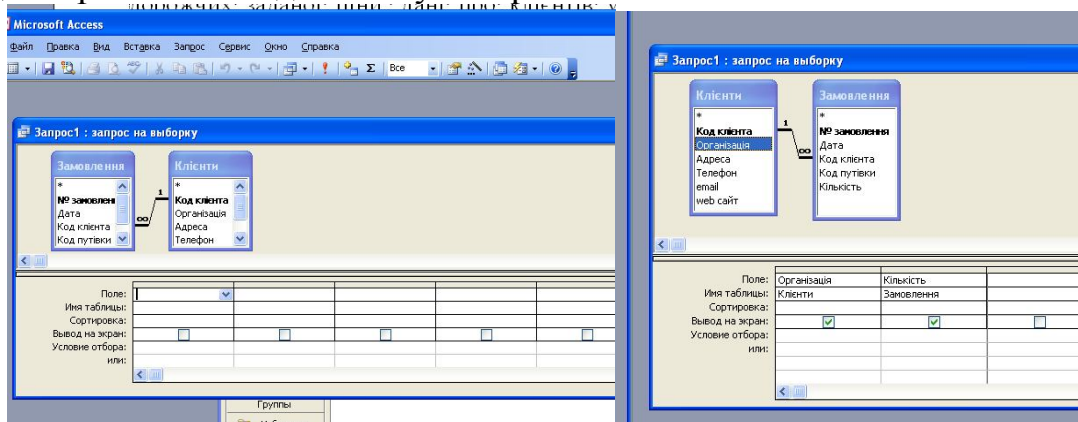
Простий запит

Створення запиту в Access (як і створення інших об'єктів) можливо здійснити за допомогою майстра або в режимі конструктора. Під час ознайомлення технологією запитів ми використовуватимемо майстер простих запитів.

Втримуючись його інструкцій, ви зможете обрати потрібну таблицю і поля цих, переглянути результати відбору на екрані.

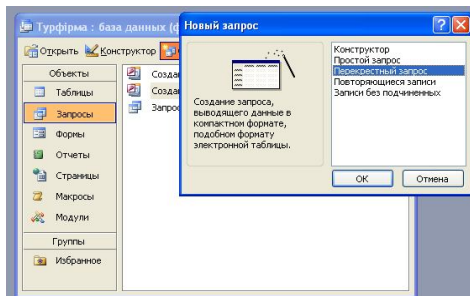
Наприклад, вас цікавить формація щодо того, які організації й у якій кількості замовляли путівки. Очевидно, що такий запит має бути зроблений на основі таблиць «Замовлення» і «Клієнти». Вважатимемо, що відповідні таблиці побудовані і що між ними встановлено зв'язок.

Відкрийте вікно бази даних, перейдіть на вкладку Запити і натисніть кнопку Створити. У діалозі Новий запит. У першому вікні майстра простих запитів зазначте, що ви створюєте запит на основі таблиці «Замовлення». Для цього в списку Таблиці/запити виділіть опцію «Таблиця: Замовлення». Потім у списку Доступні поля клацніть по позиції «Код клієнта» і натисніть кнопку із символом . Зазначена вами позиція переміститься до списку Выбранные поля.



Перехресний запит

Ефективним засобом аналізу даних є перехресний запит. Цей запит дозволяє згрупувати дані рядків або стовпців і виводити підсумкові значення до окремого стовпця.



Замовлення Запрос : запрос на выборку

	Код клиента	Организация	Кількість
▶		1 СП"Ватра Шре,	4
	3	Банк "Аваль"	2
	3	Банк "Аваль"	10
	4	ВАТ КБ "Хрець	1
	4	ВАТ КБ "Хрець	12
	5	Борщівський а	2
	6	ДП «БОРЩІВС	3
	6	ДП «БОРЩІВС	2
*			

Создание перекрестных таблиц

Выберите поля, значения которых будут использованы в качестве заголовков строк.
Допускается выбор не более трех полей.

Выберите поля по порядку сортировки данных. Например, можно сначала выполнить сортировку значений по странам, а затем по городам.

Доступные поля:

- Код клиента
- Кількість

Выбранные поля:

- Организация

Образец:

Организация	Заголовок1	Заголовок2	Заголовок3
Организация1	ИТОГИ		
Организация2			
Организация3			
Организация4			

Отмена < Назад Далее > Готово

Создание перекрестных таблиц

Выберите таблицу или запрос, поля которых необходимо вывести в перекрестном запросе.

Для включения полей из нескольких таблиц сначала создайте обычный запрос, содержащий все необходимые поля.

Показать: Таблицы Запросы Таблицы и запросы

Образец:

	Заголовок1	Заголовок2	Заголовок3
	ИТОГИ		

Отмена < Назад Далее > Готово

Создание перекрестных таблиц

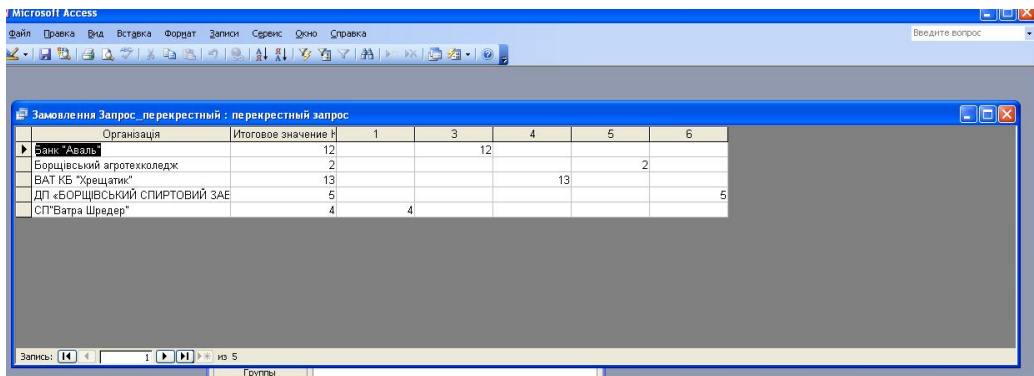
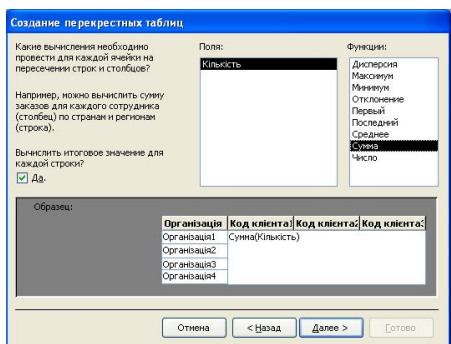
Выберите поля для использования их значений в качестве заголовков столбцов.

Например, чтобы использовать имя каждого сотрудника в качестве заголовка столбца, выберите поле ИмяСотрудника.

Образец:

Организация	Код клиента	Код клиента	Код клиента
Организация1	Дисперсия(Кількість)		
Организация2			
Организация3			
Организация4			

Отмена < Назад Далее > Готово



Використання виразів у запитах

При формуванні запиту ви можете задати обробку даних, наприклад знайти суму або середнє значення для будь-якого поля. Для цього використовуються *вирази*, за якими виконуються обчислення, а результати обчислень заносяться в окреме поле. У виразах можна вживати знаки арифметичних операцій +, -, *, /, оператори порівняння =, <, >, <=, >=, а також імена полів, які взяті у квадратні дужки. Наприклад, вираз [Ціна]*1,25 означає що вміст поля Цена збільшується у 1,25 разу.

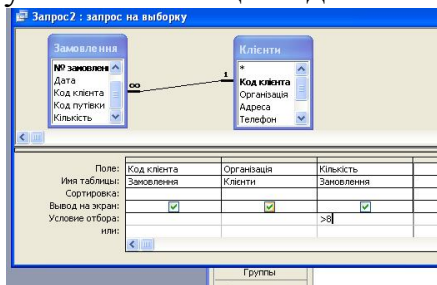
Надалі ми використовуватимемо вирази при заданні умов відбору і під час формування поля, що обчислюється в запиті.

Умови відбору

Запит, сформований згідно з вказівками пункту «Створення запиту», містить всі замовлення путівок. Якщо ж вас цікавлять лише великі замовлення (кількість путівок перевищує певне число), краще сформувавши запит із заданням умови відбору у такий спосіб:

У вікні бази даних перейдіть на вкладку Запросы і клацніть двічі по піктограмі «Замовлення: Запрос».

У відповідь відкриється вікно запиту. Перейдіть у режим конструктора запитів, клацнувши по кнопці Вид на панелі інструментів.



У діалозі наведено схему даних для розглянутих таблиць, нижче - бланк запиту. Клацніть по комірці, розташованій на перехресті рядка Условие отбора і стовпця «Кількість». Введіть із клавіатури вираз «>8» і натисніть Enter.

Розрахунки в запиті

Проілюструємо виконання розрахунків на прикладі запиту, сформованого на основі таблиць «Замовлення» і «Путівки». Нас цікавитиме сума кожного замовлення, що обчислюється як добуток ціни путівки та кількості путівок: $[Ціна] * [Кількість]$. Виконується подібний запит таким чином.

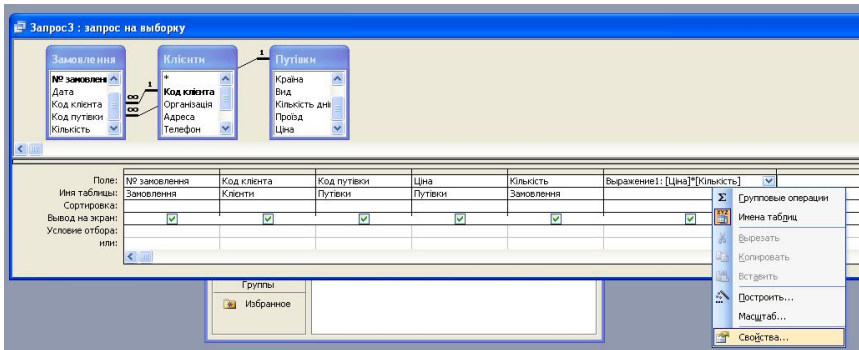
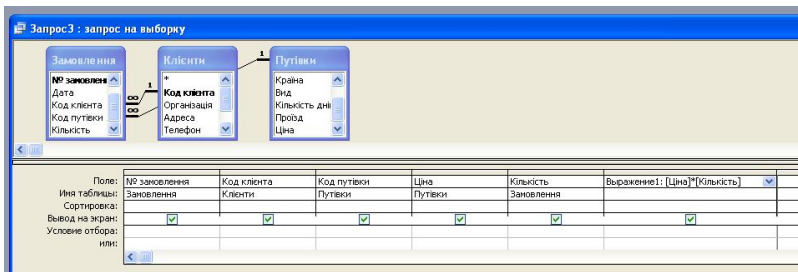
Спочатку сформуємо запит, показаний на мал.



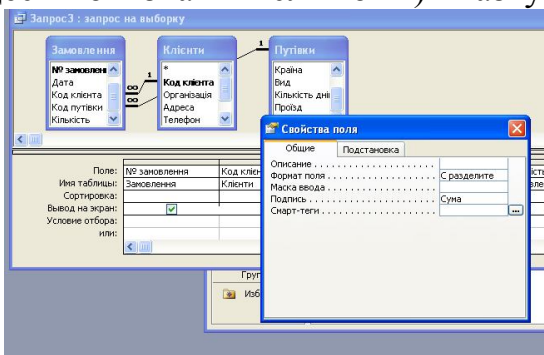
№ замовлення	Код клієнта	Код путівки	Ціна	Кількість
1	1	5	4520	4
2	3	2	3600	2
3	4	3	4950	1
4	6	1	1900	3
5	4	20	650	12
6	6	3	4950	2
7	3	3	4950	10
8	5	3	4950	2
*	(Счетчик)	(Счетчик)	(Счетчик)	

Передемо в режим конструктора запитів, клацнувши на кнопці Вид на панелі інструментів.

У вікні клацніть по полю праворуч від поля «Кількість». Введіть вираз $[Ціна] * [Кількість]$ і натисніть клавішу Enter. Перед введеним виразом з'явиться текст **Виразеніе1:**



Клацніть правою кнопкою миші в зоні поля з виразом і оберіть у контекстному меню команду Свойства. У діалозі Свойства поля задайте формат поля С разделителем (два десяткові знаки після коми) і назву поля «Сума».



Натисніть кнопку Вид і перейдіть у Режим таблиці. Ви отримаєте запит, в

останньому стовпці якого буде зазначена сума кожного замовлення.

Не замовлення	Код клієнта	Код путівки	Ціна	Кількість	Сума
1	1	5	4520	4	18 080.00
2	3	2	3600	2	7 200.00
3	4	3	4950	1	4 950.00
4	6	1	1900	3	5 700.00
5	4	20	650	12	7 800.00
6	6	3	4950	2	9 900.00
7	3	3	4950	10	49 500.00
8	5	3	4950	2	9 900.00

Отже, на основі таблиць бази даних ви отримали запит, у якому було виведено обчислюване поле - сума всіх зроблених замовлень на путівки. Розрахунки виконуються безпосередньо при виведенні запиту. Результати обчислень у таблицях не зберігаються. Тому результати запиту завжди представляють поточний вміст бази даних.

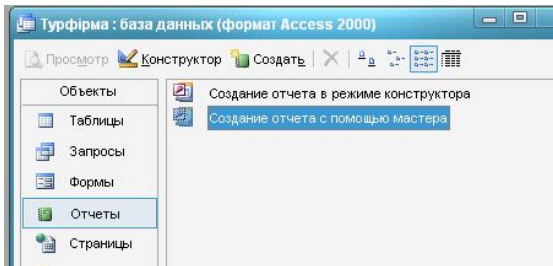
7. Створення звіту

Для представлення даних бази в звичайному вигляді потрібно створити *звіт*. Для цього в Access передбачено декілька засобів. Ми розглянемо майстер звітів, конструктор і майстер діаграм.

Найпростіше скласти звіт за допомогою майстра звітів. Складемо звіт про клієнтів фірми «Подорож».

Відкрийте базу даних. Перейдіть на вкладку Звіти.

Клацніть по кнопці Створити і у діалозі Новий звіт оберіть позицію Майстер звітів.

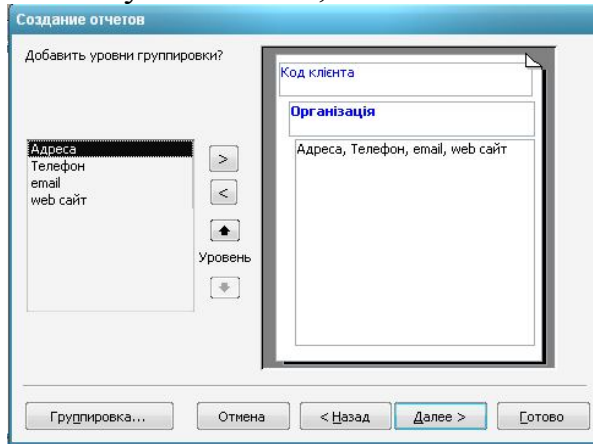


У розкритому списку вкажіть таблицю «Клієнти», на основі якої буде створюватися звіт. Клацніть по кнопці ОК.

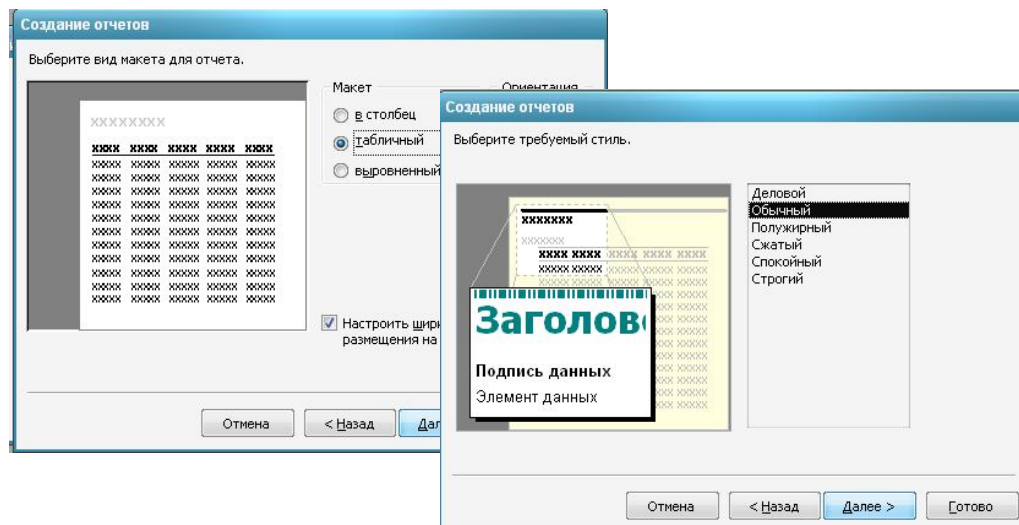


У діалозі Создание отчетов перемістіть усі поля зі списку Доступные поля до списку Выбранные поля, після чого натисніть кнопку Далее. Наступний діалог призначений для задання рівнів групування для полів звіту. Можете прийняти

установки за умовчанням, як показано на мал. Натисніть кнопку Далее.



Дотримуйтесь вказівок майстра, задаючи в наступних діалогах порядок сортування полів, вид макета, стиль оформлення звіту, а також ім'я звіту.



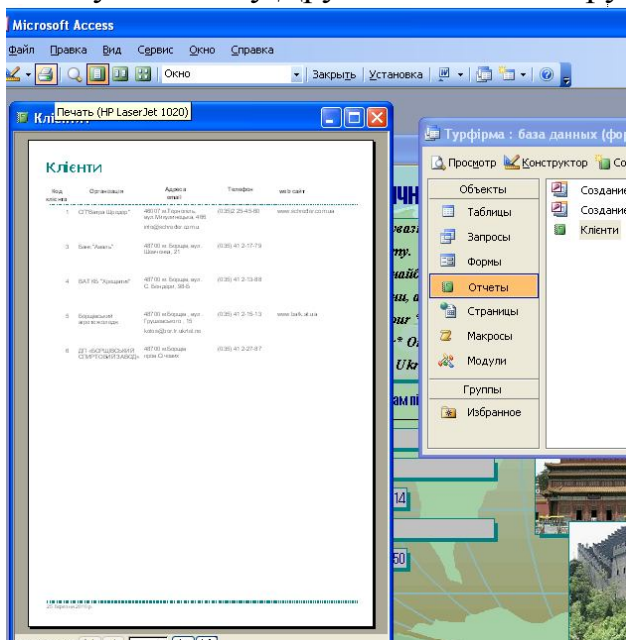
На завершення клацніть по кнопці Готово. Ви отримаєте звіт, виведений у вікно перегляду.

Клієнти

Код клієнта	Організація	Адреса email	Телефон	web сайт
1	СП "Ватра Шредер"	46007 м.Тернопіль, вул. Микулинецька, 46Б info@schreder.com.u	(035)2 25-45-60	www.schreder.com.ua
3	Банк "Аваль"	48700 м. Борщів, вул. Шевченка, 21	(035) 41 2-17-79	
4	ВАТ КБ "Хрещатик"	48700 м. Борщів, вул. С. Бендери, 98-Б	(035) 41 2-13-88	
5	Борщівський агротехколедж	48700 м.Борщів, вул. Грушевського, 15 kolos@bor.tr.ukrtel.n	(035) 41 2-15-13	www.batk.at.ua
6	ДП «БОРЩІВСЬКИЙ СПИРТОВИЙ ЗАВОД»	48700 м.Борщів, пров. Січових Стрільців,	(035) 41 2-27-87	

Можна відрегулювати розташування написів і їхній формат. Для цього потрібно переключитися за допомогою кнопки Вид у режим конструктора звітів Ці операції, що виконуються в окремому вікні конструктора звітів, прості і зрозумілі.

Зазначимо, що звіти можна створювати не лише на основі таблиць, а й використовуючи дані запитів. Для формування такого звіту потрібно в діалозі Новий отчет зазначити відповідний запит. Створений звіт можна надіслати на друк, для чого слід натиснути кнопку Друк на панелі інструментів.



Питання для самоконтролю

1. Що таке база даних?
2. Назвіть основні властивості бази даних.
3. Що таке СУБД? Яким вимогам повинні задовольняти СУБД?
4. Що таке реляційна модель даних? Наведіть приклад реляційної моделі даних.
5. Як здійснюється зв'язок таблиць у реляційній базі даних?
6. Що таке ієрархічна структура даних?
7. Чим відрізняється мережна модель даних від ієрархічної?
8. Наведіть приклади ієрархічної і мережної моделей даних.
9. З яких етапів складається проектування бази даних?
10. Навіщо визначати структуру таблиць перед створенням бази даних?
11. Якими принципами потрібно керуватися при створенні таблиць?
12. Як запустити Microsoft Access:
13. Що таке вікно бази даних?
14. Як створити файл бази даних?
15. Які об'єкти бази даних передбачені в Access?
16. Назвіть режими Access, призначені для роботи з таблицями.
17. Як викликається режим конструктора таблиць?
18. Опишіть операції задання структури таблиці за допомогою конструктора.
19. Які типи даних передбачені в Microsoft Access?
20. У чому полягає особливість поля типу Счетчик?
21. Як і навіщо вводиться до таблиці ключове поле?
22. Яким способом здійснюється зберігання таблиці поза базою даних?
23. Що таке форми і для чого вони потрібні?
24. Які відношення між таблицями вам відомі?
25. Що таке головна і зв'язана таблиці у відношенні «один до багатьох»?
26. Опишіть процедуру створення зв'язків між таблицями.
27. Як виглядатиме лінія зв'язку у вікні Схема даних, якщо не установити перемикач Обеспечение целостности данных?
28. Що таке макет зв'язків?
29. Як видалити зв'язок між таблицями?
30. Що таке ключове поле?
31. Що таке поле зовнішнього ключа?
32. Як здійснити сортування записів за зростанням (за спаданням)?
33. У чому полягає перевага фільтрації перед сортуванням?
34. Як задати простий фільтр?
35. Що таке розширений фільтр?
36. Як задається розширений фільтр?
37. Як виконується пошук потрібної інформації в базі даних?
38. Для чого призначені запити?
39. Чим запити відрізняються від фільтрів?
40. Опишіть процедуру створення простого запиту в базі даних.
41. Що таке перехресний запит і як його створити?
42. Для чого застосовуються вирази в запитах?
43. Як задати умову відбору в запиті?
44. Як створити поле, що обчислюється в запиті?