

ЛУЦЬКИЙ ІНСТИТУТ РОЗВИТКУ ЛЮДИНИ
УНІВЕРСИТЕТУ «УКРАЇНА»

кафедра “Комп’ютерні технології”



СУБД Microsoft Access

Частина 2

***Методичний посібник
для самостійного вивчення***

для студентів 1 курсу
денної та заочної форм навчання
напрямів підготовки
6.010203 - «Здоров'я людини»
6.030402 - «Правознавство»
6.030507 - «Маркетинг»
6.020107 - «Туризм»

ЛУЦЬК 2011

УДК 004(075.8)
ББК 32.97я73
М 34

**СУБД Microsoft Access. Методичний посібник для
самостійного вивчення. Частина 2.** / Укладачі Л.А.
Матвійчук, І.М. Поліщук. – Луцьк: ЛІРОЛ Університету
"Україна", 2011. – 38 с.

Укладачі:

- Матвійчук Л.А., асистент кафедри комп'ютерних технологій ЛІРОЛ Університету "Україна"
- Поліщук І.М., викладач кафедри комп'ютерних технологій ЛІРОЛ Університету "Україна"

Рецензенти:

- А.П. Усійчук, доцент, зав. кафедри комп'ютерної інженерії Луцького інституту розвитку людини Університету "Україна"
- Сомов Д.О. к.т.н., доцент кафедри автоматизованого управління виробничими процесами ЛНТУ

Розглянуто кафедрою комп'ютерних технологій ЛІРОЛ Університету «Україна» (протокол №9 від 29 квітня 2011 р.)

Схвалено вченою радою факультету комп'ютерних технологій та інформаційної діяльності ЛІРОЛ Університету "Україна" (протокол № _____ від _____ р.)

Затверджено і рекомендовано до друку вченою радою ЛІРОЛ Університету "Україна" (протокол № _____ від _____ р.)

©Матвійчук Л.А., Поліщук І.М., (укладання), 2011

Вступ

Даний методичний посібник є другою продовженою частиною, на відміну від попереднього, орієнтований на отримання не тільки теоретичних відомостей з розділу СУБД Microsoft Access, але й містить практичні завдання, тести для самоконтролю, що стимулюють до кращого засвоєння знань.

В даному посібнику мова ведеться про структурування інформації, яка набуває вигляд бази даних. Рівень структурування інформації в певній предметній області залежить від рівня вивченості цієї предметної області, від рівня розвитку науки, яка займається нею. Дані є неподільною часткою інформації, вона несе у собі неподільний згусток інформації і є такою ж абстракцією, як квант енергії в ядерній фізиці. Так, як атоми складають молекули, так і дані складають записи. Запис — це є скінченна сукупність даних, який несе певний обсяг інформації про цей об'єкт. Цей обсяг інформації визначається, по-перше, предметною областю, в якій розглядається об'єкт, а по-друге, тією задачею, у якій цей об'єкт розглядається.

Система управління базами даних Microsoft Access відноситься до реляційних баз даних, в свою чергу реляційна база даних — база даних, у якій кожен запис містить інформацію, що стосується одного конкретного об'єкта. На відміну від СУБД Visual FoxPro база даних Access (фізична структура) міститься в одному файлі з розширенням MDB. Логічна структура СУБД Access складається з таких об'єктів: таблиць, запитів, форм, звітів, макросів та модулів. Доступ до цих об'єктів відбувається за допомогою відповідних вкладинок вікна Access. Таблиці, запити, форми та звіти мають таке саме призначення, як і в СУБД Visual FoxPro.

1. Модифікація структури таблиці, дії над таблицями

1.1. Зміна проекту бази даних

Зміни вносити в проект бази даних дуже просто. Проте, вносячи зміни, слід врахувати:

1) Microsoft Access не вносить автоматично зміни до запитів, форм, звітів, макросів та модулів, які використовують таблиці зі зміненою інформацією, тож ми повинні самі внести зміни у залежні об'єкти;

2) ми не зможемо змінити тип даних для поля, яке використовується у визначенні зв'язку між таблицями. Спочатку потрібно вилучити визначення зв'язку, в якому бере участь це поле;

3) перед відкриттям таблиці в режимі конструктора для внесення змін слід закрити всі об'єкти, які посилаються на цю таблицю.

До модифікації таблиць потрібно зарахувати такі дії:

- створення резервної копії; вилучення таблиць; перейменування таблиць;
- вставка полів; копіювання полів; вилучення полів; переміщення полів;
- зміна властивостей даних; зміна типів даних; зміна розміру поля;
- розбиття таблиці; об'єднання таблиці;
- зміна первинних ключів;
- стискання бази даних.

1.2. Дії з таблицями

Для виконання дій з таблицями потрібно відкрити вікно бази даних і далі виконувати дії стандартними методами на вкладці **Таблиці**.

Для зміни імені поля, типу чи опису відкриваємо потрібну таблицю в режимі **Конструктор**, далі клацаємо лівою кнопкою миші в межах потрібного поля і вносимо потрібні зміни.

Для вилучення поля спочатку виділимо його натисканням кнопки, що знаходиться ліворуч від імені поля. Потім виконуємо команду **Правка/Удалить** або натискуємо клавішу Delete.

Для вставки нового поля між двома наявними поміщаємо курсор на існуюче поле, над яким повинен з'явитися новий рядок, і виконуємо команду **Вставка/Строки** або натискуємо клавішу Insert.

Копіювання та переміщення полів виконується звичайними методами.

Для додавання даних у таблицю відкриваємо її та починаємо вводити дані в порожній рядок, позначений зірочкою. З уведенням першого символу нижче з'являється порожній рядок, позначений зірочкою. При переході від поля до поля натискуємо клавішу Tab. При натискуванні клавіші Tab в останньому полі курсор перейде на перше поле наступного запису.

Для введення даних у поле типу МЕМО ставимо курсор у поле цього типу, натискаємо комбінацію клавіш Shift + F2: відкривається вікно **Область вводу**. Після введення або коригування даних натискаємо кнопку **ОК**.

Для коригування даних у таблиці використовуються ті самі прийоми, що й під час роботи з таблицями в Microsoft Excel. В MS Access 97 не можна коригувати такі поля: **Поля счетчика, Недоступные поля, Вычисляемые поля**. Для вилучення запису помічаємо його, потім виконуємо команду **Правка/Удалить**. Слід пам'ятати, що в MS Access 97 вилучення безповоротне. Для відновлення даних їх потрібно вводити заново.

Зміна зовнішнього вигляду таблиці здійснюється тими самими прийомами, що і в MS Excel 97.

1.3. Друк таблиці

Таблицю можна надрукувати. Для друку змісту таблиці виконаємо такі дії:

1. У вікні бази даних вибираємо потрібну таблицю й відкриваємо її в режимі таблиці. Якщо потрібно надрукувати частину таблиці, то виділяємо потрібні рядки чи стовпчики.

2. Виконуємо команду **Файл/Печать** або натискаємо на кнопку **Печать** на панелі елементів керування. З вибором команди **Файл/Печать** з'явиться діалогове вікно **Печать**.

3. Якщо є виділені об'єкти, то встановлюємо перемикач **Выделенные записи**, потім натискаємо кнопку **ОК**. Для друку всіх записів просто натискаємо кнопку **ОК**.

1.4. Пошук даних

У великих таблицях виникає потреба в пошуку потрібних записів. Для цього використовуються, як правило, два інструменти. Перший — за допомогою запитів ми розглянемо пізніше, другий — за допомогою команди **Правка/Найти**.

Щоб знайти потрібний запис:

1. Відкриваємо потрібну таблицю у вікні бази даних, ставимо курсор у поле, по якому буде йти пошук.

2. Виконуємо команду **Правка/Найти**, в результаті з'явиться діалогове вікно **Поиск в поле**.

3. У рядку **Образец** вводимо частину тексту, який потрібно знайти. У списку **Совпадение** робимо вибір з пунктів **С любой частью поля**, **Поля целиком**, **С начала поля** для відповідного режиму пошуку, у списку **Просмотр** пропонується напрямок пошуку. Далі потрібно встановити чи виключити перемикачі **С учетом регистра**, **С учетом формата полей**, **Только в текущем поле**.

4. Для знаходження першого входження шуканого тексту натискаємо кнопку **Найти**. За потреби натискаємо **Найти далее**. Продовжуємо так до повного знаходження всіх входжень.

Якщо виконати пошук за унікальними даними не вдається, то в такому разі переходимо до використання запиту.

Якщо пошук відбувається за полями, які не є ключовими, то їх потрібно проіндексувати, тоді пошук відбуватиметься значно швидше.

1.5. Встановлення зв'язків між таблицями

Якщо база даних містить кілька таблиць, то під час розробки складних форм та звітів, які використовують ці таблиці, потрібно між ними (таблицями) встановити зв'язки. Для цього:

1. У вікні бази даних відкриваємо вкладку **Таблиці** і виконуємо команду **Сервис/Схема данных**. З'явиться вікно **Схема данных**. Якщо зв'язки встановлюються вперше, то воно міститиме діалогове вікно **Добавление таблицы**, як зображено на рисунку 1.

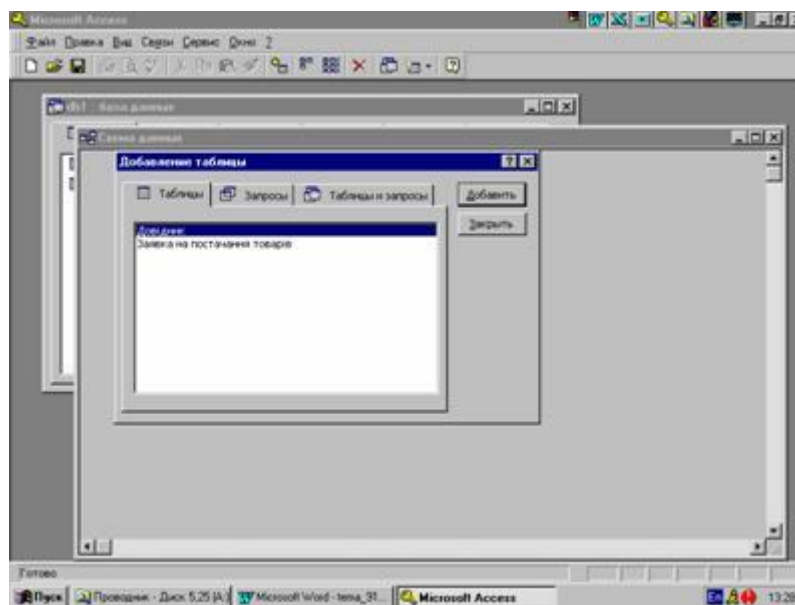


Рис. 1

2. Вибираємо таблицю, яку хочемо використати при встановленні зв'язку, потім натискаємо кнопку **Добавить**, щоб додати таблицю у вікно **Схема данных**. Після закінчення додавання таблиць натискаємо на кнопку **Закреть**.

3. Для створення зв'язку між таблицями виділяємо поле в першій таблиці, потім лівою кнопкою миші переносимо на поле, розташоване в іншій таблиці. Після відпускання лівої кнопки миші з'являється діалогове вікно **Связи**.

4. За потреби вибираємо відповідні режими. Натискаємо кнопку **Создать**, для створення зв'язку між таблицями.

Повторюємо пункти 1—4 потрібну кількість разів для створення необхідних зв'язків.

1.6. Багатотабличні бази даних. Відношення між таблицями

Реляційні бази даних складаються з декількох таблиць, зв'язок між якими встановлюється за допомогою співпадаючих полів. Кожен запис в таблицях ідентифікує один об'єкт. Відношення між об'єктами визначає відношення між таблицями. Існує 4 типи відносин:

1. Відношення «один-до-одного» означає, що кожен запис в одній таблиці відповідає тільки одного запису в іншій таблиці.

2. Відношення «один-до-багатьох» означає, що кожного запису в одній таблиці відповідає один або кілька записів в іншій таблиці.

3. Відношення «багато-до-одного» аналогічно розглянутому раніше типу. Тип відносини між об'єктами залежить від вашої точки зору.

4. Відношення «багато-до-багатьох» виникає між двома таблицями в тих випадках, коли:

- один запис із першої таблиці може бути пов'язана більш ніж з одним записом з другої таблиці;
- один запис із другої таблиці може бути пов'язана більш ніж з одним записом з першої таблиці.

У більшості випадків будь-які дві таблиці пов'язані відношенням «один-до-багатьох». Це означає, що будь-який запис у першій таблиці може бути пов'язана з декількома записами в другій, проте будь-який запис другої таблиці пов'язана тільки з одним записом у першій.

Іноді виникає потреба розбити одну таблицю на більш дрібні - проблема може полягати в тому, що деякі відомості з неї використовують не надто часто, або в тому, що якісь дані не призначаються для загального доступу. Наприклад, частину інформації про факультети потрібна тільки для рекламних цілей і використовується дуже рідко. З іншого боку, відомості про заробітну плату

повинні бути доступними лише певним співробітникам. У будь-якому з цих випадків можна створити окрему таблицю і пов'язати її з вихідною таблицею ставленням типу «один-до-одного». Це означає, що будь-який запис у першій таблиці пов'язана тільки з одним записом у другій.

Якщо ж між таблицями необхідно організувати зв'язок «багато-до-багатьох», то в Access доведеться створити додаткову таблицю перетину, за допомогою якої одна зв'язок буде зведена до двох зв'язків типу «один-до-багатьох».

2. Прості запити

Після введення даних у таблиці Access можна перейти до постановки запитів до бази даних. Ми вже раніше говорили, що найпростіша операція, яка виконується на множині даних, — це запит. Для побудови запиту в Access використовується метод QBE (Query By Example — запит за зразком). Запити забезпечують простий доступ до певної підмножини полів і записів однієї чи кількох таблиць.

Під час виконання в Access запиту на вибірку результати відображаються у формі динамічного набору даних. Динамічний набір даних має майже такий самий вигляд, як і таблиця, але він не є таблицею. Річ у тому, що коли ми створили запит, то ми створили цим програму, і в результаті виконання цієї програми маємо динамічний набір даних. Запит ми можемо зберегти, тобто ми зберігаємо не динамічний набір даних, а програму. При повторному запуску запиту запускається програма, і ми знову одержуємо динамічний набір даних. Якщо між двома запусками одного запиту зробити зміни в базі даних, то одержимо інший динамічний набір даних. Динамічний набір даних містить живі дані. Якщо їх змінити, то зміни будуть занесені до бази даних. Через це в багатокористувацьких системах цілісності баз даних потрібно приділити окрему увагу.

Якщо запит відкривається в режимі конструктора запиту, то панель інструментів налаштовується для роботи з запитами, а на час розробки запиту стає доступним меню **Запрос**.

2.1. Створення та використання запиту

Для створення запиту вручну виконаємо такі дії.

1. У вікні бази даних відкриємо вкладку **Запросы**. Даємо команду **Создать**. З'явиться діалогове вікно **Новый запрос**.

2. У діалоговому вікні **Новый запрос** виберемо пункт **Конструктор** і натиснемо кнопку **ОК**. У результаті відкриється вікно конструктора запиту з діалоговим вікном **Добавление таблицы**.

3. На вкладці **Таблицы** або **Таблицы и запросы** вибираємо потрібні таблиці та натискаємо кнопку **Добавить**.

4. Після закриття діалогового вікна **Добавление таблицы** на весь екран розкриється вікно конструктора запиту, яке містить вікно таблиць і запитів, що використовуються у створюваному запиті.

Вікно конструктора запиту поділено на дві частини. У верхній частині міститься вікно таблиць і запитів, які використовуються у створюваному запиті, а нижня частина містить бланк запиту QBE, в якому задаються поля, ім'я таблиці (чи таблиць), порядок сортування, умови відбору.

Для додавання полів у запит переміщуємо їх з вікна таблиць у рядок **Поле**. Потім у рядку **Условия отбора** задаємо умови відбору у вигляді виразів. У полях **Сортировка** задаємо порядок сортування **По возрастанию** або **По убыванию**. В рядку **Вывод на экран** можна управляти виводом на екран полів.

5. Після створення запиту й установки всіх його параметрів запит можна виконати. Для цього натискаємо на кнопку **Запуск** панелі інструментів або використовуємо команду **Запрос/Запуск**. Запит відбирає дані і відображає результати в динамічному наборі.

Запит можна зберегти командою **Файл/Сохранить**. Запит можна надрукувати, давши команду **Файл/Печать**. Запит можна модифікувати в режимі SQL командою **Вид/Режим SQL**. Запит можна також модифікувати в режимі конструктора.

2.2. Завдання умов у запитах

Користувачеві баз даних рідко коли доводиться будувати запити, в яких указувалися б лише потрібні поля та порядок сортування. Найчастіше доводиться мати справу з *запитами на вибірку за умовами*. Це найпоширеніший вид запитів, які дають змогу вибрати потрібні дані та дістати відповідь, яка задовольняє умови запиту. В Access є ще й інші типи запитів — *запити на змінювання* та *перехресні запити*. Запити на змінювання змінюють дані у вхідних таблицях згідно з умовами, які визначені в самому запиті. Ці запити використовуються для внесення великого числа змін до бази даних. Перехресні запити дозволяють показати дані у форматі, що нагадує електронну таблицю. За допомогою перехресного запиту можна згрупувати значний обсяг інформації і показати його в зручному для сприйняття вигляді.

Розглянемо види умов, які можна вводити в запити. Такими можуть бути:

- текстові умови;
- числові умови;
- умови для дат;
- умови з логічною операцією **И**;
- умови з логічною операцією **ИЛИ**;
- умови з комбінацією логічних операцій **И** та **ИЛИ**.

Текстові умови. Допускається використання різних способів введення текстових виразів: Київ, «Київ», =Київ, =«Київ». Access автоматично візьме текстовий фрагмент у лапки і поставить знак рівності. При заданні текстових умов можна використовувати оператор «Like» разом із символами «*» і «?». Нагадаємо, що символ «*» використовується для заміни невизначеної кількості

символів, а символ «?» — для заміни одного символу. Наприклад: Like «K*» — будуть вибрані всі записи з полями з відповідною умовою, в яких значення починаються з літери **K**, Like «??p*» — будуть вибрані всі записи з відповідними умові полями, в яких третя літера буде **p** (наприклад, Харків, Херсон, Переяслав-Хмельницький, Пирново і т. д.).

Числові умови. Допускається використання операторів =, <, >, <=, >=, <>, between ... and ..., наприклад =25,00, <=30, >=45, between 17 and 23.

Умови для дат. Дати можна вводити в таких форматах: 28/5/2001, #28/5/2001#, Май 28 2001, 28-Май-2001. Можна також вживати оператор between ... and Наприклад: between 1/1/2001 and 1/6/2001, between date() and date()+7.

Логічні умови. Якщо потрібно використати оператор **AND(И)**, то не менше двох умов перебувають у різних стовпчиках. Якщо потрібно використати оператор **OR(ИЛИ)**, то не менше двох умов буде для одного поля.

У результаті виконання запиту дані будуть подані у вигляді динамічного набору. В динамічному наборі можна переставляти поля, вилучати та додавати поля звичайними методами.

Щоб побачити запит у режимі SQL, потрібно вибрати команду **Вид/Режим SQL**.

2.3. Створення запитів для проведення статистичних розрахунків

Досить часто виникає потреба пошуку інформації, якої немає в явному вигляді в базі даних. Наприклад, фірма надає різні послуги, нас можуть цікавити такі дані:

1. Яке максимальне за вартістю замовлення виконала фірма?
2. Яка середня вартість виконаних замовлень?
3. Яка середня тривалість у часі виконання замовлень?
4. Яка найменша тривалість виконання замовлень?
5. Яка кількість замовлень виконується не більше двох днів?

Для виконання статистичних розрахунків потрібно створити запит на вибірку, а потім у вікні конструктора активізувати рядок **Групповая операция**. Для цього досить натиснути правою кнопкою миші на рядок **Сортировка** і в контекстному меню вибрати пункт **Групповые операции**. Рядок цей стане над рядком **Сортировка**. Для вилучення рядка **Групповые операции** виконуємо команду **Вид/Групповые операции**. Рядок містить список групових операцій.

Группировка — вибираються певні поля, які надалі оброблятимуться як група.

Sum — підсумок по даному полю.

Avg — середнє по даному полю.

Min — мінімум по даному полю.

Max — максимум по даному полю.

Count — кількість ненульових полів.

StDev — стандартне відхилення.

Var — середнє відхилення.

First — значення поля з першого запису.

Last — значення поля з останнього запису.

Выражение — починається зі знаку =.

Условие — умова на обмеження.

Узагальнюючі запити можна використовувати для виконання розрахунків по всіх записах у таблиці чи запиту. Наприклад, знайти кількість культур, мінімальне, максимальне та середнє значення по врожайностях культур. Для побудови такого запиту:

1. Вибираємо таблицю.
2. Активізуємо рядок **Групповые операции**.
3. Для будь-якого поля в рядку **Групповые операции** включаємо відповідну опцію. Отже, поле може бути з відповідною опцією чи без опції.
4. Виконуємо команду **Запрос/Запуск**.

5. Результат буде однорядковим. Для всіх полів, до яких були застосовані опції рядка **Групповые операции**, Access створює заголовки, що складаються з імені опції та імені поля. Ці назви за бажання можна замінити, перейменувавши поля в режимі конструктора. Для цього поміщаємо курсор на початок комірки поля (ліворуч від його імені), потім набираємо потрібне ім'я, а в кінці імені ставимо дві крапки.

2.4. Узагальнюючі запити для груп записів

Розглянемо узагальнюючі запити для груп записів. З групами записів доводиться мати справу досить часто, наприклад, обліковуючи надоєне молоко за кожною дояркою. Кожна доярка тричі на день доїть корів, закріплених за нею. По закінченні місяця потрібно підрахувати надій по кожній доярці. Ще один приклад. Бригада складається з робітників, які працюють за нарядною системою. За кожну зроблену роботу закривається наряд і передається до бухгалтерії підприємства. По закінченні місяця потрібно підрахувати заробіток по кожному робітнику. Нехай таблиця бази даних має такий вигляд:

Прізвище та ініціали	Дата	Номер наряду	Нараховано
Василюк І. І.	2/3/2011	1	15,76
Петров П. П.	3/5/2011	2	13,45
Міщанчук М. М.	3/5/2011	4	17,26
Павлов І. І.	4/5/2011	7	16,80
Мельник М. М.	4/5/2011	8	15,70
.....			
Гересимчук І. І.	31/5/2001	247	23,15

Проведемо розрахунки по робітникові Герасимчук І. І.

1. Протягом місяця або в кінці, якщо обсяг роботи не дуже великий, вводимо дані.

2. Відкриваємо базу даних, активізуємо вкладку **Запросы**. Далі — **Создать/Новый запрос**, вибираємо режим конструктора, додаємо потрібну таблицю.

3. Зносимо поля таблиці у запит, активізуємо рядок **Групповые операции**. По полю «Прізвище та ініціали» вибираємо операцію **Группировка**, по полю «номер наряда» — Count, по полю «нараховано» — Sum, у рядку умов по полю «Прізвище та ініціали» набираємо «Герасимчук І. І.», по полю дата — between 1/5/2001 and 31/5/2001.

4. Запрос/Запуск.

Результат матимемо однорядковий, по Герасимчук І.І., буде виведено кількість нарядів, по яких проведено нарахування, суму нарахувань за місяць.

2.5. Обчислювальні (розрахункові) поля в запиті

Після того, як одержано результати по запиту, їх потрібно надрукувати у вигляді звіту. У звіті роблять виведення розрахункових полів. Попервах побудова розрахункових полів викликає певні труднощі. Тому автори рекомендують такий підхід: усі потрібні поля вставити у запит, щоб при побудові звіту скористатися **Автоотчетом** або **Мастером отчетов**, потім, за потреби, в режимі конструктора слід зробити модифікацію звіту. Такий підхід дає змогу уникнути побудови розрахункових полів у звіті.

Нехай у запиті вставлено поля «Кількість» та «Ціна». Ставимо курсор у вільну комірку в рядку ПОЛЕ. Набираємо: Сума: [Кількість]*[Ціна]. Розрахункове поле матиме ім'я «Сума».

3. Реляційна модель даних

3.1. Відношення

Визначення. Дана сукупність множин $D_1, D_2, \dots, D_n$ (не обов'язково різних). Відношення R , яке визначене на цих множинах, є множина впорядкованих n -ок або кортежів $\langle d_1, d_2, \dots, d_n \rangle$ таких, що d_1 належить D_1, d_2

належить $D_2, \dots, d_n$ належить D_n . Множини $D_1, D_2, \dots, D_n$ називаються доменами відношення R . Величина n називається степенем відношення R .

Розглянемо відношення P степені 4, визначене на доменах $P\#$ (номер деталі), $RNAME$ (найменування деталі), $COLOR$ (колір деталі), $WEIGHT$ (вага деталі).

P

P#	PNAME	COLOR	WEIGHT
P1	Nut	Red	12
P2	Bolt	Green	17
P3	Screw	Blue	17
P4	Screw	Red	14
P5	Cam	Blue	12
P6	Cog	Red	19

Кожний рядок у таблиці являє собою одну n -ку (або просто кортеж) відношення. Число кортежів у відношенні називається кардинальним числом Відношення. Наприклад, кардинальне число відношення P дорівнює 6. Відношення першого степеня називають унарним, степеня 2 — бінарним, степеня 3 — тернарним, степеня n — n -арним.

Якщо ввести поняття декартового добутку множин $D_1, D_2, \dots, D_n$ як $D_1 \times D_2 \times \dots \times D_n$, то кортеж $\langle d_1, d_2, \dots, d_n \rangle$, є елементом множини всіх можливих упорядкованих кортежів $\langle d_1, d_2, \dots, d_n \rangle$, таких, що $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$. Вважатимемо, що R є відношення на множинах $D_1, D_2, \dots, D_n$, якщо R є підмножиною декартового добутку $D_1 \times D_2 \times \dots \times D_n$. Як правило, вважається, що кортежі у відношенні впорядковані.

3.2. Домени та атрибути

Важливо зрозуміти, що домен являє собою множину значень, яку може мати атрибут. Отже, атрибут — це конкретне використання доменна всередині відношення. Скажімо, в попередньому прикладі відношення визначаються

п'ятьма атрибутами. Значення цих атрибутів утворюються з множини значень відповідного доменна. Взагалі, треба було б розрізняти імена доменів та атрибутів. Але, якщо з кожного доменна утворюється лише один атрибут, то такого розрізнення не проводять. Якщо ж з одного доменна в одному відношенні утворюють два атрибути, то тоді їм дають різні імена.

Відношення бувають нормалізовані та ненормалізовані. В реляційній моделі єдиними відношеннями, які допустимі, є ті, які відповідають такій умові: кожне значення у відношенні, тобто значення кожного атрибута в кожному кортежі, є неподільним. Або інакше: на перетині будь-яких рядків та стовпчиків має бути лише одне значення. Таке відношення є нормалізованим.

3.3. Ключі

Якщо всередині даного відношення R існує атрибут, значення якого однозначно ідентифікують кортежі цього відношення, то такий атрибут називають первинним ключем даного відношення. Може бути так, що первинний ключ буде неможливо утворити з одного атрибута, але завжди знайдеться така комбінація атрибутів, яку можна ідентифікувати як первинний ключ. Існування такої комбінації гарантується тим, що множини не мають однакових елементів, отже, кортежі як елементи множини повинні бути різні. Звичайно накладають обмеження, що жоден з компонентів первинного ключа не має бути нульовим.

Дамо тепер визначення реляційної бази даних. Реляційною моделлю бази даних є представлення користувачем цієї бази даних у вигляді сукупності змінюваних у часі нормалізованих відношень різних степенів.

У СУБД під кортежем розуміють запис, а під атрибутом — поле.

3.4. Реляційна модель даних «Постачальники-деталі»

Визначимо реляційну модель даних таким чином [12]:

DOMAI	S#	CHARA
N		CTER (5)

	DOMAI	SNAME	CHARA
N			CTER (20)
	DOMAI	STATUS	NUMERI
N			C (3)
	DOMAI	CITY	CHARA
N			CTER (15)
	DOMAI	P#	CHARA
N			CTER (6)
	DOMAI	PNAME	CHARA
N			CTER (20)
	DOMAI	COLOR	CHARA
N			CTER (6)
	DOMAI	WEIGHT	NUMERI
N			C (4)
	DOMAI	QTY	NUMERI
N			C (5)

RELATION S (S#, SNAME, STATUS, CITY) KEY (S#)

RELATION P (P#, PNAME, COLOR, WEIGHT, CITY) KEY (P#)

RELATION SP (S#, P#, QTY) KEY (S#, P#)

Припускається, що в цих відношеннях є невідомий користувачу порядок.

Суть кортежа відношення SP така: постачальник (S#) поставляє задану деталь (P#) у заданій кількості.

Для подальшої роботи наведемо деякі значення.

S

#	SNAME	STATUS	CITY
1	Smith	20	London

2	Jones	10	Paris
3	Blake	30	Paris
4	Clark	20	London
5	Adams	30	Athens

P

#	PNAME	COLOR	WEIGHT
1	Nut	Red	12
2	Bolt	Green	17
3	Screw	Blue	17
4	Screw	Red	14
5	Cam	Blue	12
6	Cog	Red	19

SP

S#	P#	QTY
S1	P1	300
S1	P2	200

S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

4. Застосування форм

Форми є основним засобом організації інтерфейсу користувача в додатках Microsoft Access. Можна створювати форми з різноманітними цілями:

1. *Виведення та редагування даних.* Це найпоширеніший засіб використання форм. Форми забезпечують виведення на екран даних в обраному нами представленні. З їх допомогою значно спрощується внесення змін, додавання та видалення даних із бази. У формі можна зробити доступними тільки для читання всі дані або певну їх частину, автоматично вибирати інформацію з інших пов'язаних таблиць і вміщувати її у форму, обчислювати у формі значення, що виводяться, приховувати (робити невидимими) або, навпаки, виводити деякі дані залежно від значень інших полів запису або від обраних користувачем параметрів.

2. *Керування ходом виконання додатка.* Щоб автоматизувати виведення певних даних або виконання певної послідовності дій, можна створити форми для роботи з макросами або функціями VBA. Для запуску макросів або процедур VBA можна створити спеціальні елементи керування, які називаються командними кнопками. За допомогою макросів і процедур можна відкривати інші форми, виконувати запити чи команди меню, фільтрувати

виведені на екран дані, встановлювати значення в записах і формах, виводити меню на екран, роздруковувати звіти та виконувати ряд інших дій. Можна спроектувати форму таким чином, щоб макроси або функції VBA запускались у відповідь на певні події — наприклад, відкриття форми, активізацію певного елемента керування або зміну даних у формі.

3. *Введення даних.* Можна створити форми, призначені тільки для введення у базу даних нових даних або значень, що допомагають автоматизувати виконання додатка.

4. *Виведення повідомлень.* Форми можуть надавати допоміжну інформацію про роботу додатка або про виконувані дії. За допомогою макрокоманди **Сообщение** або функції VBA можна вивести на екран інформацію, попередження чи повідомлення про помилки.

5. *Друк інформації.* Незважаючи на те, що для друку інформації найчастіше доводиться використовувати звіти, можна також роздруковувати інформацію, що міститься у формі. Оскільки можна визначити один набір опцій для виведення форми на екран, а інший — для її друку, форма може відігравати подвійну роль. Наприклад, можна задати два заголовки і дві примітки для форми — одну пару (заголовок і примітка) для введення замовлення, а другу — для друку за даним замовленням рахунку-фактури.

4.1. Стандартна панель інструментів вікна режиму форми

Корисно приділити певний час і увагу панелі інструментів, що Access виводить на екран при відкритті форми. Щоб відкрити форму, клацаємо на корінці **Форма** у вікні бази даних, вибираємо потрібну форму зі списку й натискаємо кнопку **Открыть**. Кнопки стандартної панелі інструментів режиму форми перелічені нижче в порядку їх розташування зліва направо:

Кнопка **Конструктор** відкриває форму в режимі конструктора.

Кнопка **Режим форми**. Ця кнопка зображена натиснутою, мовби вказуючи на те, що ми працюємо в режимі форми.

Кнопка **Режим таблицы** відкриває форму в режимі таблиці.

Кнопка **Печать**. Натискаємо цю кнопку, щоб надрукувати виведені у формі дані.

Кнопка **Просмотр**. Маємо можливість роздрукувати більшість форм із даними. Натискаємо цю кнопку, щоб переглянути сторінку перед її друком.

Кнопка **Новая запись**. Користуємося цією кнопкою, щоб для введення нового запису перейти на порожній рядок, розташований наприкінці таблиці.

Кнопка **Вырезать**. Слугує для того, щоб скопіювати виділене в буфер обміну і вилучити виділене.

Кнопка **Копировать**. Натискаємо цю кнопку для того, щоб скопіювати виділене в буфер обміну без вилучення виділеного.

Кнопка **Найти**. Натискаємо цю кнопку, щоб знайти певні значення деяких полів.

Кнопка **Вставить**. Натискаємо цю кнопку, щоб вставити дані, які перебувають у буфері обміну, у виділений рядок або елемент керування.

Кнопка **По возрастанию**. Натискаємо в будь-якому місці елемента керування, пов'язаного з полем набору записів, по якому хочемо здійснити сортування, а потім натискаємо цю кнопку. Після цього рядки у формі виводитимуться в порядку зростання значень обраного поля. Ми не можемо сортувати дані в підпорядкованій формі.

Кнопка **По убыванию**. Клацаємо в будь-якому місці елемента керування, пов'язаного з полем набору записів, по якому ми хочемо здійснити сортування, а потім натискаємо цю кнопку. Після цього рядки у формі виводитимуться в порядку убуття значень обраного поля. Ми не можемо сортувати дані в підпорядкованій формі.

Кнопка **Изменить фильтр**. Ми можемо відбирати і сортувати виведені у формі дані точно так само, як це робили під час роботи з запитом або таблицею. Натискаємо цю кнопку, щоб створити чи змінити фільтр або порядок сортування.

Кнопка **Применить фильтр**. Натискаємо її, щоб застосувати фільтр, який ми визначили.

Кнопка **Показать все записи**. Користуємося нею, щоб скасувати дію фільтра і вивести у формі всі записи базової таблиці чи запиту.

Кнопка **Окно базы данных** активізує вікно бази даних, роблячи його видимим, якщо до цього воно було прибране, і розгортає, якщо воно було згорнуте в значок.

Кнопка **Восстановить поле/запись**. Натискаємо цю кнопку, щоб скасувати всі зміни полів поточного запису. За відсутності змін ця кнопка недоступна (блякла кнопка).

Кнопка **Отменить**. Нею можна скасувати останню виконану дію. Функція цієї кнопки аналогічна дії команди **Правка/Отмена** або комбінації клавіш Ctrl + Z. Ця кнопка недоступна, якщо немає змін, які можна було б скасувати.

Кнопка **Карточка подсказки** відкриває основне меню карток підказувань, що дадуть нам докладні покрокові інструкції для виконання типових дій.

Кнопка **Справка**. Натискаємо цю кнопку, щоб біля покажчика з'явився знак питання. Після цього встановлюємо цей покажчик на будь-якому об'єкті, клацаємо кнопкою миші — і одержимо контекстну довідку щодо цього об'єкта.

4.2. Заголовок, ділянка даних і ділянка приміток

Звичайно інформація, яку ми хочемо вивести з базової таблиці або запиту, розміщується в ділянці даних, розташованій у центрі вікна форми. Щоб вивести на екран інформацію чи елементи керування, що не змінюватимуться від запису до запису, ми можемо додати у верхню частину вікна форми заголовок або в нижню частину — ділянку приміток.

4.3. Створення простої форми для введення даних

Access дає змогу швидко створювати форми за допомогою майстра форм. Можна також розробляти форми вручну, відкривши порожню форму та перемістивши мишкою об'єкти (наприклад, поля, текст, графіку тощо) в

потрібне місце форми. Можна також поєднати автоматизовану та ручну розробку.

1. Виберемо таблицю, а потім натиснемо кнопку **Новый объект** панелі інструментів або відкриємо меню кнопки **Новый объект** і виконаємо команду **Автоформа** — тоді для вибраної таблиці буде створена форма за замовчуванням. У цій формі всі поля вирівняні по лівому краю форми, а їх імена використані як написи полів.

2. Якщо ми хочемо керувати створенням форм, то чинимо так: у вікні бази даних відкриваємо вкладку **Формы**, потім — **Создать**. З'явиться діалогове вікно **Новая форма**. Вибираємо **Мастер форм**, а у списку внизу вікна вибираємо таблицю чи запит, на основі яких створюватиметься форма. Далі виконуємо вказівки **Мастера форм**.

4.4. Розробка форм вручну

Форми можна розробляти вручну, відкривши порожню форму та додавши в неї потрібні об'єкти.

1. **Формы/Создать/Новая форма**.

2. У діалоговому вікні **Новая форма** натискаємо кнопку **Конструктор**. Потім у списку **Выберите в качестве источника данных таблицу или запрос** робимо вибір.

3. Натискаємо **ОК**, форма відобразиться в режимі конструктора форми.

4. Далі виконуємо команду **Вид/Панели элементов**.

5. За допомогою **Панели элементов** поміщаємо на форму потрібні об'єкти (поля, тексти, графіки, кнопки).

6. Виконуємо команду **Файл/Сохранить** для збереження форми.

5. Створення звітів

Звіт є кінцевим наслідком багатьох задач управління базами даних. Можна створювати різні звіти з різними рівнями деталювання.

Звіти можна створювати вручну, за допомогою засобу **Автоотчет** або за допомогою майстра звітів.

5.1. Автоматичне створення звіту

Якщо у нас є вибрана таблиця чи запит, відкриваємо меню кнопки **Новый объект** панелі інструментів і вибираємо команду **Автоотчет**. Буде створено звіт у стовпчик.

5.2. Створення звітів за допомогою майстра звітів

Майстри звітів використовуються для створення звітів у стовпчик, стрічкових звітів з групами та без, поштових наклейок та підсумкових звітів. Для створення звіту за допомогою майстра звітів:

1. У вікні бази даних відкриваємо вкладку **Отчет** і натискаємо кнопку **Создать** — з'явиться діалогове вікно **Новый отчет**.
2. Вибираємо у списку праворуч пункт **Мастер отчетов**. У списку внизу виберемо таблицю (чи запит), дані якої будуть використані у звіті.
3. Натискаємо кнопку **ОК** — з'явиться діалогове вікно **Создание отчетов**. Виконавши певні дії в цьому вікні і натиснувши кнопку **Далее**, переходимо до наступного діалогового вікна і т. д.
4. Після виходу з шостого діалогового вікна звіт потрібно надрукувати. Наприклад, **Файл/Печать**.

ПРАКТИЧНЕ ЗАВДАННЯ

Завдання 1. База даних "Абітурієнт"

Постановка завдання. У цьому варіанті завдання необхідно створити базу даних, в якій вирішувалися наступні завдання:

введення, зберігання і пошук інформації про абітурієнтів;

- реєстрація основних даних про внз, що віддаються перевага абітурієнтами;
- ведення обліку оплати абітурієнтами підготовчих курсів;
- відстежування результатів вступу і відвідування занять після вступу.

Порядок виконання завдання.

1. Створення бази даних. У меню Файл виберіть команду Створити і із списку шаблонів на закладці Загальні виберіть шаблон "База даних". Збережете створювану БД під ім'ям Абітурієнт.mdb.

2. За допомогою конструктора створіть наступні таблиці.

Таблиця: тАбитуриенты

<i>Ім'я</i>	<i>Тип</i>	<i>Розмір</i>
<i>КодАб</i>	<i>Лічильник</i>	<i>4</i>
<i>Прізвище</i>	<i>Текстовий</i>	<i>15</i>
<i>Ім'я</i>	<i>Текстовий</i>	<i>15</i>
<i>По батькові</i>	<i>Текстовий</i>	<i>1</i>
<i>Паспорт</i>	<i>Текстовий</i>	<i>15</i>
<i>Індекс</i>	<i>Довге ціле</i>	<i>4</i>
<i>Місто</i>	<i>Текстове</i>	<i>15</i>
<i>Адреса</i>	<i>Текстова</i>	<i>20</i>
<i>Середній бал</i>	<i>Одинарне з плаваючою 4</i>	
<i>Фото</i>	<i>Поле об'єкту OLE</i>	<i>-</i>
<i>Примітки</i>	<i>Поле MEMO</i>	<i>-</i>

Таблиця: тАттестат

<i>Ім'я</i>	<i>Тип</i>	<i>Розмір</i>
<i>КодАб</i>	<i>Довге ціле</i>	<i>4</i>
<i>Російський</i>	<i>Довге ціле</i>	<i>4</i>
<i>Математики</i>	<i>Довге ціле</i>	<i>4</i>
<i>Фізики</i>	<i>Довге ціле</i>	<i>4</i>

Таблиця: тВУЗЫ

<i>Ім'я</i>	<i>Тип</i>	<i>Розмір</i>
<i>КодВУЗа</i>	<i>Лічильник</i>	<i>4</i>
<i>НазваниеВУЗа</i>	<i>Текстовий</i>	<i>50</i>
<i>Індекс</i>	<i>Довге ціле</i>	<i>4</i>
<i>Місто</i>	<i>Текстове</i>	<i>20</i>
<i>Вулиця</i>	<i>Текстовий</i>	<i>15</i>
<i>Телефон</i>	<i>Текстовий</i>	<i>15</i>
<i>Рейтинг</i>	<i>Довге ціле</i>	<i>4</i>
<i>Твір</i>	<i>Одинарний з плаваючою</i>	<i>4</i>
<i>Математики</i>	<i>Довге ціле</i>	<i>4</i>
<i>Фізики</i>	<i>Довге ціле</i>	<i>4</i>
<i>ПроцПринятых</i>	<i>Ціле</i>	<i>2</i>
<i>Перевірка</i>	<i>Логічний</i>	<i>1</i>
<i>Дата</i>	<i>Дата/час</i>	<i>8</i>

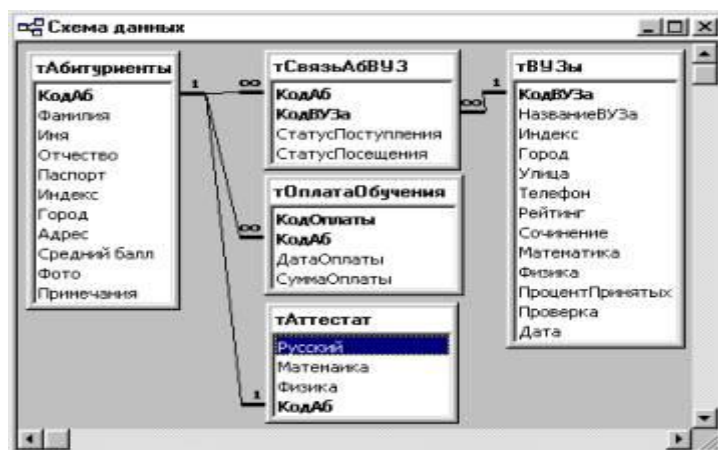
Таблиця: тОплатаОбучения

<i>Ім'я</i>	<i>Тип</i>	<i>Розмір</i>
<i>КодОплаты</i>	<i>Лічильник</i>	<i>4</i>
<i>КодАб</i>	<i>Довге ціле</i>	<i>4</i>
<i>ДатаОплаты</i>	<i>Дата/час</i>	<i>8</i>
<i>СуммаОплаты</i>	<i>Грошовий</i>	<i>8</i>

Таблиця: тСвязьАбВУЗ

Ім'я	Тип	Розмір
КодАб	Довге ціле	4
КодВУЗа	Довге ціле	4
СтатусПоступления	Логічний	1
СтатусПосещения	Логічний	1

3. Створіть наступну схему даних :



4. Виконаєте підстановку поля КодАб з таблиці тАбитуриент в таблиці тСвязьАбВУЗ, тОплатаОбучения, тАттестат. Наприклад, відкрийте таблицю тАттестат в режимі конструктора, виділите поле КодАб, перейдіть на вкладку Підстановка і вкажіть властивості поля, показані на малюнку, :

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Таблица или запрос
Источник строк	тАбитуриенты
Присоединенный столбец	1
Число столбцов	2
Заглавия столбцов	Нет
Ширина столбцов	0см;3см
Число строк списка	8
Ширина списка	3см
Ограничиться списком	Да

Аналогічно виконаєте підстановку поля КодВУЗа з таблиці тВУЗы в таблицю тСвязьАбВУЗы.

5. Заповніте таблиці даними, ввівши в кожен з них не менше п'яти рядків. Наприклад, перші два рядки таблиці тАбитуриенты :

Код Аб	Прізвище	Ім'я	По батькові	Паспорт	Індекс	Місто	Адреса	Середній бал	Фото	Примітки
1	Василук	Вася	В	V –АЮ	328010	Луцьк		4,60		

2	Гнатюк	Петро	Г	VI -АС	459020	Ковель		4,30		
...	...	...	...	...	...	...	...	...	...	...

6. Побудуйте запит на вибірку імен і прізвищ тих абітурієнтів, які мають середній бал від 4 до 5:

Поле:	Фамилия	Имя	Средний балл
Имя таблицы:	тАбитуриенты	тАбитуриенты	тАбитуриенты
Сортировка:			
Вывод на экран:	☒	☒	☐
Условие отбора:			>=4 And <=5
или:			

7. За допомогою наступного запиту виведіть прізвища тих абітурієнтів, які проживають або в Рівне, або в Сарни :

Поле:	Фамилия	Город
Имя таблицы:	тАбитуриенты	тАбитуриенты
Сортировка:		
Вывод на экран:	☒	☐
Условие отбора:		"Белгород" Or "Москва"
или:		

8. Побудуйте багатотабличний запит для виведення списку прізвищ абітурієнтів і назв тих ВНЗ, в які вони збираються поступати, :

Поле:	Фамилия	НазваниеВУЗа
Имя таблицы:	тАбитуриенты	тВУЗы
Сортировка:		
Вывод на экран:	☒	☒
Условие отбора:		
или:		

9. За допомогою наступного запиту виведіть прізвища тих абітурієнтів, які доки не визначилися з вибором ВНЗ (без переваг), :

Поле:	Фамилия	КодАб
Имя таблицы:	тАбитуриенты	тСвязьАбВУЗ
Сортировка:		
Вывод на экран:	☒	☐
Условие отбора:		Is Null
или:		

10. Прикладом створення обчислюваного поля в запиті може служити наступний запит, в якому підрахуйте середній бал оцінок атестатів абітурієнтів:

Поле:	Фамилия	Средний балл: ((Русский)+[Математика]+[Физика])/3
Имя таблицы:	тАбитуриенты	
Сортировка:	по возрастанию	
Вывод на экран:	☒	☒
Условие отбора:		
или:		

11. Побудуйте підсумковий запит для визначення суми оплати кожним абітурієнтом за підготовчі курси в 2001 році:

Поле:	Фамилия	Год: Year([ДатаОплаты])	Итого: СуммаОплаты
Иная таблицы:	тАбитуриенты		тОплатаОбучения
Групповая операция:	Группировка	Группировка	Sum
Сортировка:			
Вывод на экран:	☒	☒	☒
Условие отбора:		2001	
или:			

12. За допомогою майстра побудови перехресних запитів побудуйте перехресний запит, що виводить наступну таблицю, :

Прізвище	Разом	Січень	Лютий	Березень	Квітень
Василук	500 грн.	200грн.	200грн.		
Гнатюк	500 грн.		200грн.	100грн.	
Сидоров	500 грн.				300грн.

13. Створіть форму для введення даних в таблицю тАбитуриенты ..

14. Створіть форму для введення інформації по внз:

15. Створіть форму, що складається з двох пов'язаних форм, для ведення обліку оплати абітурієнтами занять на підготовчих курсах:

16. Створіть стрічкову форму для зміни статусу абітурієнтів :

17. Створіть форму для перегляду переваг абітурієнтів :

18. Створіть звіти, що включають списки абітурієнтів і ВНЗ.

19. Створіть підсумковий звіт по оплаті абітурієнтами підготовчих курсів :

20. Створіть макрос для висновку інформаційного повідомлення про призначення програми:

21. Створіть головну форму з набором командних кнопок і призначте кожній кнопці макрос, що виконує відкриття потрібної форми або звіту:



ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

Варіант 1.

База даних Бібліотека.

1. Створити параметричний запит для відображення прізвищ і телефонів учнів, які повинні здати книги до дати, визначеної параметром, і назв цих книг.
2. Створити запит для відображення числа учнів кожного класу, що взяли книги.
3. Створити звіт по обох таблиць.

Варіант 2.

База даних Туризм.

1. Створити запит для відображення всієї інформації про фірми, що пропонують путівки на відпочинок в гірськолижних базах.
2. Створити запит для відображення інформації про вартість путівок до Єгипту з урахуванням наданих знижок.
3. Створити звіт по обох таблиць.

Варіант 3.

База даних Банк.

1. Дати запит на створення таблиці, що відображає інформацію про курси продажу і купівлі доларів США і Канади.
2. Створити запит для відображення інформації про суму продажу по всім відділенням.
3. Створити звіт по продажу та купівлі валюти по всім відділенням з зазначенням загальної суми.

Варіант 4.

База даних Магазин.

1. Створити параметричний запит для відображення всієї інформації про фірми, що постачають певний значенням параметра товар.
2. Створити запит для відображення загальної суми поставок кожного товару.
3. Створити звіт по другій таблиці, включивши всі поля, вказати загальну суму поставок кожного товару.

Варіант 5.

База даних Квитки.

1. Створити перехресний запит для відображення кількості проданих квитків кожним кіоском.
2. Створити запит для відображення вартості певного виду проїзного квитка (параметричний запит).
3. Створити звіт по продажах кожного виду проїзних квитків всіма кіосками з зазначенням загальної суми по кожному виду транспорту.

Варіант 6.

База даних Товари.

1. Створіть параметричний запит для відображення в алфавітному порядку назв фірм, що постачають телевізори вартістю до певної суми, назви і опису цього товару.
2. Створити запит для відображення інформації про всі телевізорах.
3. Створити звіт за даними про музичних центрах.

Варіант 7.

База даних Бібліотека.

1. Створити запит для відображення всієї інформації про користувачів бібліотеки.
2. Створити запит, який відображає назви книг і їх авторів, взятих у бібліотеці.
3. Створити звіт по обох таблиць.

ТЕСТ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ

Вибрати і записати правильну відповідь(і):

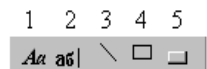
1. Клавіша, яка використовується для позначення всіх потрібних об'єктів у звіті — це:

- а) Alt;
- б) Ctrl;
- в) Shift;
- г) Ins;

2. Зона, в якій розташовується підсумковий рядок, називається:

- а) зона заголовка;
- б) зона даних;
- в) зона приміток;
- г) зона нижнього колонтитула;
- д) зона верхнього колонтитула.

3. Записати по порядку номерів, що виконують зазначені



кнопки:

- а) «Надпись»;
- б) «Линия»;
- в) «Поле»;
- г) «Кнопка»;
- д) «Прямоугольник»;
- е) «Рисунок»;
- є) «Переключатель»;
- ж) «Поле со списком».

4. Чи можна створити звіт з пов'язаних файлів?

- а) так;
- б) ні (спочатку створюється об'єднаний файл, за яким і будується звіт).

5. Пункт меню сортує зміст таблиці за зростанням:

- а) «Формат/Сортировка/По возрастанию»;
- б) «Записи/Сортировка/По возрастанию»;

- в) «Записи/Упорядочить/По возрастанию»;
- г) «Вид/Сортировка/По возрастанию».

6. Пункт меню дає можливість переглянути і відредагувати індекси, що вже існують:

- а) «Вид/Индексы»;
- б) «Сервис/Индексы»;
- в) «Формат/Индексы»;
- г) «Индексы/Формат».

7. PrimaryKey — це:

- а) індексоване поле;
- б) ключове поле;
- в) відсортоване поле.

8. «Мастер создания запросив» викликається за допомогою кнопки:

- а) «Конструктор»;
- б) «Простой запрос»;
- в) «Автозапрос — табличный»;
- г) «Запрос — сводная таблица»;
- д) «Мастер запросов».

9. «Простая форма» — це:

- а) вікно для вибірки потрібної інформації;
- б) вікно для перегляду потрібної інформації;
- в) вікно для обчислення потрібної інформації;
- г) вікно для виклику іншої форми.

10.«Построитель выражений» застосовується у

- а) простих формах;
- б) запитах;
- в) звітах;
- г) таблицях;
- д) макросах;
- е) кнопочних формах.

Основні терміни та визначення

Атрибут — частина інформації про певний об'єкт, найчастіше міститься у полі.

База даних — це множина даних з певною множиною операцій над даними, які зберігаються у структурованому вигляді.

Запис — скінчена сукупність даних, яка несе певний обсяг інформації про описуваний об'єкт.

Запит — об'єкт, за допомогою якого можна отримати потрібні дані з однієї чи кількох таблиць.

Зв'язок — спосіб, за допомогою якого інформація з однієї таблиці пов'язується з інформацією іншої таблиці.

Звіт — об'єкт, призначений для форматування, обчислення підсумків та інших значень і виведення потрібних даних на друкуючий пристрій.

Макрос — структурований опис однієї чи кількох стандартних дій, які має виконати СУБД у відповідь на певну подію (натиснення кнопки, вибір пункту меню тощо).

Модуль — об'єкт, який містить програми, написані мовою Visual Basic. За допомогою програм можна виконувати нестандартні дії, що потрібні користувачеві для роботи.

Поєднання — сукупність інформації, яка міститься в кількох джерелах на основі атрибутів, які збігаються між собою в кожному з джерел.

Реляційна база даних — база даних, у якій кожен запис містить інформацію, що стосується одного конкретного об'єкта.

Система управління базами даних — прикладна програма або комплекс програм, які дають можливість користувачеві виконувати дії над даними, що зберігаються.

Таблиця — об'єкт СУБД, де зберігається інформація. Здебільшого складається з рядків (*записів*) та стовпчиків (*полів*).

Форма — об'єкт, призначений для введення нових даних, перегляду існуючих, обчислень над даними та управління роботою бази.

Література

1. Диго С.М. Проектирование и использование баз данных. - М.; Финансы и статистика, 1995. - 200 с.
2. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник. 3-є вид./ За ред. Г.А. Шинкаренка, О.В. Шишова. – К.: Каравела, 2008. – 640 с.
3. Информатика: Практикум по технологии работы на компьютере/Под ред. Н.В. Макаровой. - 3-е изд., перераб. - М.: Финансы и статистика, 2005. - 256 с.
4. Злобін Г.Г. Основи інформатики, комп'ютерної техніки і комп'ютерних технологій (для студентів економічних спеціальностей) /Г.Г. Злобін.- К.: Каравела, 2008.– 240с.
5. Злобін Г.Г. Персональний комп'ютер: Навч. Курс. 2-е вид /Г.Г. Злобін.- Львів.: Простір М, 2007.- 294с.
6. Каратыгин С. А. ACCESS 2000. Руководство пользователя с примерами.– М.: Лаборатория Базовых Знаний, 2000 – 276 с.
7. Карпенко С. Г., Попов В. В., Тарнавський Ю. А., Шпортюк Г. А. Інформаційні системи і технології: Навч. посіб. для студ. вищ. навч. закл. — К.: МАУП, 2004.
8. Карпов Б. Microsoft Access 2002: справочник. – Питер, 2001.
9. Наумова Н.М. Інформатика і комп'ютерна техніка: Навч. посіб. для дист. навчання. / Наумова Н.М.- К.: Університет «Україна», 2004.-404 с.
- 10.Рзаєв Д. О., Шарапов О. Д., Ігнатенко В. М., Дибкова Л. М. Інформатика та комп'ютерна техніка: Навч.-метод. посібник для самост. вивч. дисц. — К.: КНЕУ, 2004. — 486
- 11.Рзаєв Д.О., Шарапов О.Д., Ігнатенко В.М., Дибкова Л.М.Інформатика та комп'ютерна техніка: Навч.-метод. посіб-ник для самост. вивч. дисц. - К.: КНЕУ, 2002. - 486 с. (частина 1), (частина 2).
- 12.Шинкаренко Г.А. Інформатика. Комп'ютерна техніка. Комп'ютерні технології / Г.А. Шинкаренко, О.В. Шишов. – К.: Каравела, 2007. – 640 с.
- 13.Хомоненко А.Д., Цыганков В.М., Мальцев М.Г., Базы данных : Учебник для высших учебных заведений /Под ред проф. А.Д. Хомоненко – Спб: Корона, 2000. - 416с.

Зміст

Вступ.....	3
1. Модифікація структури таблиці, дії над таблицями	6
1.1. Зміна проекту бази даних.....	6
1.2. Дії з таблицями.....	6
1.3. Друк таблиці	7
1.4. Пошук даних.....	8
1.5. Встановлення зв'язків між таблицями.....	9
1.6. Багатотабличні бази даних. Відношення між таблицями.....	10
2. Прості запити.....	11
2.1. Створення та використання запиту	12
2.2. Завдання умов у запитах	13
2.3. Створення запитів для проведення статистичних розрахунків	14
2.4. Узагальнюючі запити для груп записів	16
2.5. Обчислювальні (розрахункові) поля в запиті	17
3. Реляційна модель даних	17
3.1. Відношення.....	17
3.2. Домени та атрибути	18
3.3. Ключі	19
3.4. Реляційна модель даних «Постачальники-деталі»	19
4. Застосування форм.....	22
4.1. Стандартна панель інструментів вікна режиму форми.....	23
4.2. Заголовок, ділянка даних і ділянка приміток.....	25
4.3. Створення простої форми для введення даних.....	25
4.4. Розробка форм вручну	26
5. Створення звітів	26
5.1. Автоматичне створення звіту	27
5.2. Створення звітів за допомогою майстра звітів	27
ПРАКТИЧНЕ ЗАВДАННЯ	28
ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ.....	34
ТЕСТ ДЛЯ САМОПЕРЕВІРКИ ЗНАНЬ.....	36
Основні терміни та визначення	38
Література	39

НАВЧАЛЬНО-МЕТОДИЧНЕ ВИДАННЯ

Методичний посібник для самостійного вивчення з курсу " СУБД Microsoft Access " для студентів 1 курсів напрямків «Здоров'я людини», «Правознавство», «Маркетинг», «Туризм»

денної та заочної форм навчання

Комп'ютерний набір та верстка:

Л.А. Матвійчук, І.М. Поліщук

Підп. до друку _____ 2010. Формат 60×84/16. Папір офс.

Гарн. Таймс. Ум друк. арк. 3

Обл.-вид.арк. 2,5. Тираж 10 прим. Зам.3852

Луцький інститут розвитку людини

Університету «Україна»

43023, м. Луцьк, вул. Карбишева, 2

