

МІЖРЕГІОНАЛЬНА  
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МАУП

**МЕТОДИЧНІ МАТЕРІАЛИ  
ЩОДО ЗАБЕЗПЕЧЕННЯ  
САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ  
з дисципліни  
“БАЗИ ДАНИХ  
ТА ІНФОРМАЦІЙНІ СИСТЕМИ”  
(для бакалаврів)**

МАУП

Київ  
ДП «Видавничий дім «Персонал»  
2010

Підготовлено доцентом кафедри прикладної математики та програмування  
*Н. М. Москальковою*

Затверджено на засіданні кафедри інформатики та інформаційних  
технологій (протокол № 17 від 29.02.08)

*Схвалено Вченою радою Міжрегіональної Академії управління персоналом*

**Москалькова Н. М.** Методичні матеріали щодо забезпечення самостійної роботи студентів з дисципліни “Бази даних та інформаційні системи” (для бакалаврів). — К.: ДП «Вид. дім «Персонал», 2010. — 45 с.

Методична розробка містить пояснювальну записку, тематику самостійної роботи з дисципліни, практичні завдання, питання для самоконтролю та список літератури.

Призначена для методичного забезпечення самостійної роботи з дисципліни “Бази даних та інформаційні системи” студентів денної форми навчання, які здобувають освіту за напрямком “Прикладна математика”.

© Міжрегіональна Академія  
управління персоналом (МАУП), 2010  
© ДП «Видавничий дім «Персонал», 2010

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

Сучасні інформаційні технології передбачають використання систем управління базами даних (СУБД), якщо обсяг інформації, яку необхідно опрацювати, достатньо великий. Необхідність використання СУБД посилюється за умови виконання дій над інформацією (експорт, обробка, імпорт тощо) кількома спеціалістами. Вибір серед достатньо великої кількості сучасних СУБД програмного продукту, що забезпечить ефективне його використання у сфері діяльності підприємства, а також не суперечить інтересам програміста, спеціаліста є складним завданням, особливо для початківця. Існують багато спільних структур і об'єктів різних СУБД. Це дає можливість вивчити сучасні системи програмування баз даних на прикладі таких СУБД, які, по-перше, найпоширеніші у використанні, а, по-друге, мають риси інших СУБД. Особливі акценти зроблено на організації запитів, використанні мови SQL (Structured Query Language). Актуальною на сьогоднішній день є підготовка спеціалістів, які вміють ефективно організовувати бази даних, володіють сучасними засобами програмування в середовищі СУБД, використовують бази даних в інформаційних системах. Саме для підготовки таких спеціалістів й призначена дисципліна “Бази даних та інформаційні системи”.

Сучасна реформа вищої освіти — це насамперед перехід від парадигми навчання до парадигми освіти, самоосвіти. Тому при реформуванні вищої школи, введенні кредитно-модульної технології навчання значно зростає роль самостійної роботи студентів. Самостійна робота студентів є основним засобом опанування навчального матеріалу у позааудиторний час. Студент, який хоче якнайкраще оволодіти професією, має добре розуміти: на занятті викладач дає основи знань, навчає, як учити, підкреслює ті ключові істини дисципліни, які пробуджують у молодій людини потяг до поглиблення й удосконалення своїх знань. Лише постійне самостійне навчання дає можливість якомога ближче підійти до вершини знань певної галузі, оволодіти такою сумою знань і вмінь, які б дали змогу заявити про себе як про професіонала.

Самостійна робота студентів є надзвичайно важливою складовою підготовки спеціалістів за напрямком “Прикладна математика”. Теоретичний матеріал з програмування потребує багаторазового підкріплення практичними прикладами. Студенти мають набути навичок самостійного виконання усіх етапів розробки програмного забезпе-

чення (проекування, створення, тестування тощо). Це вимагає від студента систематичного виконання практичних завдань протягом семестру та підготовки до кожного практичного заняття.

Самостійна робота студента повинна бути спланована, організовано і методично спрямована як особиста творча праця без безпосередньої взаємодії з викладачем. Згідно із державними стандартами навчальний матеріал навчальної дисципліни, передбачений робочим навчальним планом для засвоєння студентом у процесі самостійної роботи, вноситься на підсумковий контроль поряд з навчальним матеріалом, який опрацьовувався при проведенні навчальних занять. Навчальний час, відведений для самостійної роботи, регламентується робочим навчальним планом і повинен, згідно із Болонською декларацією, становити не менше 50 % загального обсягу навчального часу студента, відведеного для вивчення конкретної дисципліни.

Самостійна навчальна діяльність студента з дисципліни “Бази даних та інформаційні системи” передбачає:

- запам'ятовування певної інформації під час конспектування лекцій; активної роботи під час практичних занять;
- роботу над конспектами лекцій, планами практичних занять;
- опрацювання літературних джерел (конспектування самостійно вивченого матеріалу, реферування);
- роботу з каталогами звичайних і електронних бібліотек, інформаційно-пошуковими сервісами Internet;
- вивчення навчального матеріалу за паперовими та електронними підручниками, навчальними посібниками, практикумами тощо;
- опрацювання матеріалу за першоджерелами, науковою і спеціальною літературою;
- підготовку доповідей, рефератів, написання курсових робіт; пошукову і науково-дослідну діяльність.

*Самостійна робота студента під час лекції.* Належне ведення конспекту під час лекції сприяє збереженню необхідної інформації та дає студенту змогу в подальшому проаналізувати її. За умови подання лекційного матеріалу в усній формі одночасно засвоюється до 20 % інформації.

*Робота над конспектами лекцій, планами практичних занять.* При підготовці до практичних занять студент має спиратися на складений ним конспект лекції. При опрацюванні матеріалу лекції слід зіставити законспектований матеріал з планом практичного заняття, що

міститися у методичних матеріалах для практичних занять або у навчально-методичному комплексі. Якщо у конспекті бракує матеріалу з окремих питань лекції або недостатньо розкриті деякі питання практичного заняття, або вони винесені на самостійне опрацювання, студент повинен звернутися до рекомендованих підручників, навчальних посібників і відповідних методичних матеріалів.

*Вивчення навчального матеріалу за підручниками, навчальними посібниками, методичними вказівками, опрацювання матеріалу за першоджерелами, науковою і спеціальною літературою.* Працювати із підручниками, навчальними посібниками, методичними вказівками, практикумами, науковою і спеціальною літературою незалежно від типу їх носія (паперового чи електронного) необхідно таким чином, щоб отримати максимум теоретичних знань і навичок. При роботі з цими джерелами студент насамперед повинен ознайомитися з їх змістом, щоб визначити, чи необхідно опрацьовувати це джерело і чи стосується воно навчального курсу, що вивчається, і тільки після цього визначити послідовність його опрацювання і відібрати необхідний для вивчення матеріал з цього джерела (глави, розділу тощо). У разі роботи з інтерактивними електронними джерелами слід використовувати можливості навігації за документом, що надаються сучасними програмами, призначеними для читання електронних документів відповідних форматів (*MS Word, Adobe Reader, Adobe Acrobat* тощо), і особливо переваги гіпертекстової технології подання навчального матеріалу, а саме — за допомогою гіперпосилань знаходити відповіді на поставлені питання. При опрацюванні матеріалу необхідно з'ясувати суть питання, що вивчається, не уникаючи при цьому визначення суті незрозумілих чи незнайомих слів, термінів. Саме інтерактивні гіпертекстові електронні джерела (довідки у складі програмних продуктів, електронні посібники та словники) дозволяють конкретизувати терміни та визначення якнайшвидше. При вивченні матеріалу необхідно аналізувати прочитане, порівнюючи з прослуханою та законспектованою лекцією, робити логічні висновки, позначати незрозумілі положення з метою їх подальшого з'ясування на практичному занятті. Бажано відпрацювати зручну для себе певну систему позначень (позначки на полях конспекту, підкреслення маркерами різних кольорів, доповнення конспекту альтернативними формулюваннями та посиленнями на інші джерела тощо) та фіксації опрацьованого матеріалу.

*Робота з бібліотечними фондами та дистанційними джерелами з метою пошуку необхідної інформації.* З позицій випереджаючої освіти навчання тільки за конспектом лекцій і основною літературою, вказаною у навчальній програмі, є недостатнім. У більшості випадків належна підготовка вимагає вміння швидко знаходити та опрацювати необхідний матеріал за першоджерелами, науковою і спеціальною літературою та коректно цитувати знайдене. Перелік такої літератури, як правило, наводиться у навчально-методичному комплексі навчальної дисципліни. Тому завдання студента полягає у самостійному пошуку цих матеріалів у паперових або електронних фондах бібліотек, а також у різноманітних файлових архівах, базах даних та базах знань, доступ до яких здійснюється за допомогою відповідних сервісів *Internet*.

Для пошуку документу використовуються різні його ознаки. Насамперед це — реквізити документа (УДК. Автор(и). Заголовок опису. Основний заголовок: відомості, що належать до заголовка/Відомості про відповідальність. — Відомості про видання (у тому числі URL — адреса Web — документа або Ftp — файлу). — Місце видання, дата видання. — Обсяг.). УДК — це універсальна десяткова класифікація будь-яких офіційних видань у всьому світі. Відповідні довідники видаються багатьма мовами і постійно оновлюються. Коли код УДК невідомий, то необхідно звернутися до алфавітного каталогу бібліотеки і за назвою джерела або за прізвищем та ініціалами автора знайти відповідний бібліотечний шифр джерела.

Матеріали щодо методів підвищення ефективності пошуку інформації в *Internet* містяться у статтях: <http://www.yandex.ru/info/search.html>, <http://www.searchengines.ru/>, <http://www.zodchiy.ru/links/search/>, <http://www.citforum.ru/internet/search/index.shtml>, <http://websearch.report.ru/>, <http://www.kokoc.com/search-engines/index.shtml>, <http://www.zhurnal.ru/search-r.shtml>.

*Метою вивчення дисципліни* “Бази даних та інформаційні системи” є здобуття знань щодо сучасних технологій проектування баз даних і розробки систем управління базами даних, засвоєння основних понять реляційних баз даних та набуття практичних навичок створення програмних систем з використанням системи управління базами даних Microsoft Access.

*Завдання курсу* — дати студентам теоретичні знання необхідні для розв’язання завдань автоматизації обробки інформації у різних

предметних сферах, а також сформувати практичні навички використання та проектування систем управління базами даних, розробки програмних засобів збереження та маніпулювання даними. Під час вивчення дисципліни передбачається систематична практична робота студентів як під керівництвом викладача, так і самостійно. Предметом вивчення курсу “Бази даних та інформаційні системи” є реляційна модель даних, засоби маніпулювання реляційними базами даних, об’єкти системи управління базами даних *Microsoft Access* та засоби їх створення.

Після вивчення дисципліни студент повинен *знати*:

- тенденції та перспективи розвитку інформаційних систем, систем управління базами даних;
- технології збереження, пошуку та обробки інформації;
- теоретичні основи побудови та функціонування баз даних, характеристики сучасних СУБД, сучасні технології організації БД;
- правила розробки структури баз даних та створення прикладного програмного забезпечення з використанням систем управління базами даних;
- принципи побудови та технологію проектування баз даних;
- основні поняття реляційної моделі даних;
- основи мови побудови запитів SQL;
- засоби побудови баз даних за допомогою MS Access.

А також студент має набути навичок:

- використання інформаційних систем у різних предметних сферах;
- проектування інформаційних систем та баз даних;
- створення програмного забезпечення для доступу до баз даних (засобами *MS Access*);
- здійснення аналізу даних засобами сучасних систем управління базами даних.

Для розуміння тематики курсу “Бази даних та інформаційні системи” студенти повинні мати базові знання і навички роботи на персональному комп’ютері в операційній системі *Microsoft Windows*, із стандартними програмами *Microsoft Windows* і додатками з пакету *Microsoft Office*. Пропонований курс базується на дисциплінах “Основи програмування та алгоритмічні мови”, “Мови об’єктно-орієнтованого програмування”, “Основи дискретної математики”. Для розуміння тематики дисципліни “Бази даних та інформаційні системи”

студенти повинні мати знання з основ програмування та алгоритмічних мов, володіти мовами об'єктно-орієнтованого програмування, мати досвід використання систем та інструментальних засобів програмування.

Підсумкова перевірка знань студентів передбачена у вигляді іспиту.

Самостійна робота з дисципліни “Бази даних та інформаційні системи” передбачає такі види робіт і форми їх оцінювання:

- опрацювання програмного матеріалу за змістовим модулем (за матеріалами конспекту і підручників) та оцінювання під час проміжного контролю;
- підготовка та власне аудиторна робота під час практичних і лабораторних занять. Результати оцінюються під час поточного контролю;
- виконання самостійних робіт у формі есе, рефератів з конкретних проблем та складання письмових звітів на електронних або паперових носіях або усних доповідей;
- виконання практичних завдань до змістовного модуля та оцінка їх результатів під час проміжного контролю;
- виконання письмової контрольної роботи або тестування;
- підготовка до складання іспиту;
- звіт про науково-дослідну роботу, результати якої можуть бути використані при написанні випускної роботи і опубліковані за рішенням кафедри.

**ТЕМАТИКА САМОСТІЙНОЇ РОБОТИ**  
*з дисципліни*  
**“БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ”**

**Змістовний модуль I. Основні концепції створення інформаційних систем**

**Тема 1. Проектування інформаційних систем**

**Опрацювання програмного матеріалу** (за матеріалами конспекту, підручників).

Наведення даних за допомогою ЕОМ. Концепція інтегрованої обробки даних: скорочення надлишковості, цілісність даних, незалеж-



ність прикладних програм від даних. Поняття предметної області. Поняття бази даних. Моделі даних: інфологічна модель даних, дата-логічна модель даних, фізична модель даних. Поняття систем управління базами даних. Архітектура систем управління базами даних. Функції систем управління базами даних.

Реляційні інформаційні структури та системи управління базами даних. Загальні властивості відношень у реляційній базі даних. Ефективність використання пам'яті. Принципи нормалізації. Нормальні форми та їх види. Ключі та індекси. Цілісність даних.

Проектування інформаційних систем засобами **MS Access**. Створення заголовка відношення. Типи даних **MS Access**. Призначення властивостей полів таблиці. Підстановка значень до поля таблиці. Використання властивостей вкладки *Подстановка* для визначення способу підстановки значень. Використання засобу *Мастер подстановки*. Визначення зв'язків між таблицями у **MS Access**. Засіб *Схема даних*. Зміна типу зв'язку між таблицями.

*Література* [2; 6; 10–15; 21–23; 29; 30]

### Основні визначення

Кожна прикладна задача обробки інформації пов'язана з тією чи іншою частиною реального світу, яку називають *предметною областю*. Предметна область зазвичай розглядається як деяка сукупність реальних об'єктів (сутностей), кожний з яких має певний набір *властивостей*. Для відображення предметної області будують *інформаційну модель*, яка містить описи інформаційних сутностей (об'єктів) та тих їх властивостей (атрибутів), що є значущими для розв'язання задачі.

Наприклад, для автоматизації обліку співробітників деякого відділу необхідно проаналізувати такі дані, як вік співробітника, його заробітна плата, стаж роботи, посада тощо. У цьому разі інформаційним об'єктом є співробітник, а дані про співробітника є властивостями цього об'єкта. При цьому окремого співробітника розглядають як *екземпляр* інформаційного об'єкта “Співробітник”.

У повній інформаційній моделі кожний екземпляр інформаційного об'єкта є унікальним, тобто набір його властивостей відрізняється принаймні одним значенням від усіх інших екземплярів цього об'єкта. Тому кожному екземпляру надають унікальний ідентифікатор, який дозволяє здійснювати посилання на нього. Наприклад, податковий ідентифікаційний код є однозначним ідентифікатором екземпляра

об'єкта “платник податків”, табельний номер співробітника — однозначним ідентифікатором екземпляра об'єкта “співробітник фірми”, інвентарний номер обладнання — однозначним ідентифікатором екземпляра об'єкта “обладнання” тощо. У складних предметних областях інформаційні об'єкти є взаємопов'язаними. Наприклад, співробітник фірми є платником податків, обладнання є власністю деякої фірми тощо. Тому для комплексного аналізу предметної області необхідно мати засоби наведення та аналізу цих зв'язків.

*Загальний опис усіх інформаційних об'єктів та їх взаємозв'язків називається зовнішньою (інфологічною) моделлю предметної області.*

Створення інфологічної моделі предметної області є важливим етапом, який передує проектуванню довільної інформаційної системи. Вся інформація про предметну область, яка специфікована у її інформаційній моделі, має бути збережена у пам'яті комп'ютера за допомогою деяких універсальних механізмів.

*База даних — це сукупність даних, які організовані згідно із деякими загальними правилами опису, збереження та маніпулювання даними, які забезпечують незалежність даних від прикладних програм.*

Сучасна технологія роботи з базами даних передбачає, що доступ користувачів та прикладних програм до бази даних здійснюється за допомогою спеціальних засобів, які називають системою управління базою даних (СУБД).

*Системою управління базою даних називається комплекс програм, що призначені для створення, обробки та управління даними в базі, забезпечення ефективного доступу до неї з боку всіх користувачів, а також для підтримки даних в актуальному стані.*

СУБД є складним програмним комплексом, який забезпечує виконання функцій, пов'язаних зі створенням та експлуатацією бази даних, а саме:

- підтримує внутрішню (логічну) модель даних та надає користувачам мову опису та маніпулювання даними в цій моделі;
- виконує операції обробки даних (вибір, вставка, оновлення, видалення тощо);
- забезпечує захист та цілісність даних у базі.

Користувачі бази даних працюють з її інформаційним та змістовим наповненням, їх не цікавлять подробиці фізичного збереження даних на магнітних носіях. Тому розрізняють три рівня абстракції для наведення даних, які зберігаються у базі даних:

- зовнішня інформаційна модель даних;

- логічна модель даних (логічна схема збереження даних);
- фізична модель даних (фізична схема збереження даних).

Протягом років становлення та розвитку інформаційних систем використовувались мережневі, ієрархічні та реляційні моделі даних на логічному (концептуальному) рівні. Але більшість сучасних СУБД використовують реляційну модель даних. Назва реляційної моделі даних походить від англійського слова “relation” (відношення).

*У реляційній моделі предметна область наводиться в вигляді сукупності взаємопов'язаних відношень (таблиць), кожне з яких описує деякий клас однотипних об'єктів предметної області.*

У реляційній таблиці, стовпці називаються *атрибутами (полями)*, оскільки вони характеризують одну із властивостей (аспектів) об'єктів. Список назв усіх стовпців (атрибутів) називається *схемою відношення*. Наприклад, відношення СТУДЕНТЫ може описуватись такою схемою:

СТУДЕНТ = (Номер залікової книжки, Прізвище, Ім'я, По батькові, Курс, Група, Дата народження).

Кожний вектор (кортеж) даних у відношенні, що відповідає одному рядку таблиці, називається *записом*. Кожний запис (рядок) описує один екземпляр інформаційного об'єкта. Для однозначної ідентифікації рядків таблиці можна використовувати деяку підмножину її атрибутів, яка називається *ключем*.

*Ключ — це мінімальний набір атрибутів, за значенням яких можна однозначно ідентифікувати окремих екземплярів об'єкта (рядок у таблиці).*

Наприклад, студента можна ідентифікувати за прізвищем, але якщо трапляються студенти з однаковим прізвищем, то до ключа слід додати як поле *Прізвище*, так і поля *Ім'я* та *По батькові*. Але може статися, що відомі студенти з однаковим прізвищем, ім'ям, по батькові. Тоді ключ має містити ще і дату народження.

Підмножина полів таблиці є ключем тоді і тільки тоді, коли виконуються такі умови:

1. Рядки таблиці мають відрізнятися значенням хоча б одного ключового атрибута (унікальність значень ключових атрибутів).
2. Жоден з атрибутів не може бути виключений з підмножини ключових атрибутів без порушення умови унікальності (мінімальність).

У реляційній моделі даних між таблицями встановлюють зв'язки, які відображають взаємозв'язки між відповідними інформаційними об'єктами предметної області. Зв'язок “первинний ключ – зовнішній ключ” породжує зв'язок типу “один до багатьох” між таблицями. Якщо зовнішній ключ є одночасно внутрішнім ключем, то зв'язок має тип “один до одного”. У цьому разі між векторами обох таблиць встановлюється взаємоднозначна відповідність.

*Реляційною базою даних називається набір взаємопов'язаних таблиць, зв'язки між якими реалізуються у вигляді посилань між однотипними атрибутами цих таблиць. Сукупність зв'язків між таблицями утворюють схему реляційної бази даних.*

У *MS Access* таблиці використовуються для збереження даних. В комірках таблиці зберігається лише фактична інформація. Способи створення реляційних таблиць в СУБД *MS Access* наведені у вигляді таблиці.

Метод	Опис
Конструктор	<ul style="list-style-type: none"> <li>• задайте імена атрибутів</li> <li>• вкажіть тип даних кожного з них</li> <li>• змініть при необхідності властивості полів</li> <li>• визначте ключ</li> <li>• збережіть таблицю</li> </ul>
Майстер	<ul style="list-style-type: none"> <li>• оберіть атрибути із запропонованих таблиць (типи даних та їх властивості будуть визначені автоматично)</li> <li>• визначте ключ</li> <li>• визначте зв'язки з існуючими таблицями</li> </ul>
Введення даних	<ul style="list-style-type: none"> <li>• розпочніть введення даних в таблицю (тип та розміри полів будуть визначені автоматично відповідно до введених значень)</li> <li>• при збереженні визначте ключ таблиці або автоматично буде додано ключове поле типу <i>Счетчик</i></li> </ul>

При визначенні значень атрибутів можна використовувати такі типи даних:

1. *Текстовий* – рядок до 255 символів;
2. *Числової* – числові значення;
3. *Дата/время* – значення дати та часу;
4. *Денежний* – грошові значення;
5. *Счетчик* – унікальні автоматично зростаючі цілі значення;

6. *Логический* — два можливих значення (True/False, Да/Нет, 0/1);
7. *Поле MEMO* — великі текстові фрагменти (до 64 Кбайт);
8. *Поле OLE* — об'єкти з інших застосувань (малюнки, таблиці тощо);
9. *Гиперссылка* — гіперпосилання на ресурси у мережі Інтернет;
10. *Мастер подстановок* — дозволяє здійснити підстановку списку значень.

Кожне поле таблиці має набір властивостей, які впливають на спосіб уведення та відображення даних у цьому полі. Перелік властивостей поля залежить від типу поля та наводиться на вкладці *Общие* у вікні конструктора таблиць. Властивість *Формат поля* слугує для визначення формату наведення значень поля. Властивість *Маска ввода* дозволяє відобразити дані у загальноприйнятій формі (наприклад, номер телефону відобразити з символами тире). Властивість *Подпись* слугує для підпису, який буде відобразитись у заголовку поля (за замовчуванням підпис поля співпадає з його назвою). Властивість *Значение по умолчанию* дозволяє задати значення, яке буде встановлене за замовчуванням для кожного нового запису таблиці. Властивість *Условие на значение* використовується для перевірки належності значення, що вводиться у поле, заданому діапазону. Якщо значення є помилковим, то видається повідомлення, текст якого задається у властивості *Сообщение об ошибке*. Поле можна визначити як обов'язкове (властивість *Обязательное поле*) або дозволити залишати його значення порожнім (властивість *Пустые строки*). За допомогою властивості *Индексированное поле* можна визначити індексоване поле, тобто поле, за яким система здійснює прискорений пошук записів. Можна також вимагати унікальність значень індексованого поля, встановивши цій властивості значення *Да (Совпадения не допускаются)*. На вкладці *Подстановка* можна визначити підстановку для заданого поля, яка дозволяє вводити дані в поле за допомогою вибору їх зі списку або поля зі списком.

Окремі таблиці не дають змоги представити структуру та взаємозв'язки об'єктів предметної області. Тому наступним етапом створення схеми даних є встановлення зв'язків. Зв'язок — це асоціація (посилання), що встановлюється між таблицями та дає змогу запобігти надлишковості даних. Зв'язок між таблицями встановлюється за полями, які мають однаковий тип даних. Встановлення зв'язків у *Access* здійснюється у вікні *“Схема данных”*, для відкриття якого необхідно виконати команду *Сервис Схема данных*. У цьому вікні

зв'язки між таблицями відображаються у вигляді ліній, які проведено між полями таблиці.

### Тематика практичних завдань

1. Засобами *MS Access* створити схему даних для предметної області “Кредитування на придбання автомобіля” з таблицями “Автомобілі”, “Позичальник”, “Кредити” та відповідними полями:

*Автомобілі* ( 🔑 Код автомобіля, Назва автомобіля, Дата випуску, Об'єм двигуна, Дата продажу автомобіля)

*Позичальник* ( 🔑 Ідентифікаційний номер, Прізвище, Адреса, Серія та номер паспорта, Місце роботи, Код автомобіля, Сума кредиту, Дата видачі кредиту, Дата погашення кредиту)

*Кредити* ( 🔑 Код кредиту, Ідентифікаційний номер, Сума боргу, Термін виплати за місяцями, Процентна ставка, Щомісячний внесок)

2. Засобами *MS Access* створити схему даних для предметної області “Відправлення грошових переказів у гривнях” з таблицями “Відправник”, “Отримувач”, “Пункт переказу” з відповідними полями:

Таблиця “*Відправник*” ( 🔑 Код відправника, Прізвище відправника, Країна, Місто, Адреса, Сума переказу, Плата за переказ, Дата відправлення)

Таблиця “*Отримувач*” ( 🔑 Код отримувача, Код відправника, Прізвище отримувача, Країна, Місто, Адреса, Серія та номер паспорта, Дата отримання переказу)

Таблиця “*Пункт переказу*” ( 🔑 Код пункту, Назва пункту, Контрольний номер грошового переказу, Прізвище оператора, Податок на переказ, Код відправника, Код отримувача)

3. Засобами *MS Access* створити схему даних для предметної області “Облік використання фондів стаціонару медичної установи” з таблицями “Відділення”, “Лікар”, “Пацієнт”, “Діагноз” з відповідними полями:

Таблиця “*Відділення*” ( 🔑 Код, Найменування, Кількість Місць, Добова Вартість Утримання Пацієнта)

Таблиця “*Лікар*” ( 🔑 Код, Прізвище, Ім'я, По батькові, Посада, Відділення, Кількість Планових Хворих)


Таблиця “*Пацієнт*” ( 🔑 Код, Прізвище, Ім'я, По батькові, Номер Картки, Діагноз, Відділення, Дата Госпіталізації, Дата Виписки, Лікар)


Таблиця “*Діагноз*” ( 🔑 Код Діагнозу, Діагноз)

4. Засобами *MS Access* створити схему даних для предметної області “Облік аналізів крові пацієнтів” та створити таблиці “Клінічний Аналіз Крові”, “Біохімічний Аналіз Крові”, “Паспорт Пацієнта”, “Діагноз” з відповідними полями:

Таблиця “*Діагноз*” (  Код, Найменування)

Таблиця “*Клінічний Аналіз Крові*” ( Код, Код Пацієнта, Дата, Еритроцити, Нб, Лейкоцити, ЛейкПалЯдрові, ЛейкСегмЯдрові, Лімфоцитів, Моноцитів, СОЕ, Висновок)

Таблиця “*Біохімічний Аналіз Крові*” (  Код, Код Пацієнта, Дата, Загальний Білок, Альбуміни, Глобулін Альфа1, Глобулін Альфа2, ГлобулінБета, Глобулін Гамма, Тімолова Проба, Сулемова Проба, АлАТ, Білірубін Загальний, Протромбіновий Індекс, Фібріноген, Висновок)

Таблиця “*Паспорт Пацієнта*” (  НомерКартки, Прізвище, Ім’я, По батькові, Дата Народження, Стать, Адреса, Місце Роботи, Дата Госпиталізації, Дата Виписки, Діагноз При Госпиталізації, Клінічний Діагноз, Супутній Діагноз)

5. Засобами *MS Access* створити схему даних для предметної області “Облік бюджету медичної установи” та створити таблиці “Відділення”, “Співробітники”, “Посади” з відповідними полями:

Таблиця “Відділення” (  Код, Найменування, КількістьМісць, ДобоваВартістьУтриманняПацієнта)

Таблиця “Співробітники” (  Код, Прізвище, Ім’я, По батькові, Посада, Перс надбавка, Відділення, Стать)

Таблиця “Посади” (  Код, Найменування, Оклад)

6. Засобами *MS Access* створити схему даних для предметної області “Облік переліку та вартості процедур, що відпускаються у медичних закладах санаторно-курортного типу” та створити таблиці “Лікарські Засоби”, “ЛікиНаПроцедуру”, “Процедура” з відповідними полями.

7. Засобами *MS Access* створити схему даних для предметної області “Облік замовлень на використання виробничих ліній для таблетування різних лікарських форм” та створити таблиці “Упаковка”, “Замовлення”, “Склад”, “Субстанції” з відповідними полями.

8. Засобами *MS Access* створити схему даних для предметної області “Облік замовлень на отримання лікарських розчинів лікарнею” та створити таблиці “Розчини”, “Замовлення”, “Пацієнт”, “Діагноз” з відповідними полями.

9. Засобами *MS Access* створити схему даних для предметної області “Облік роботи аптечної бази” та створити таблиці “Склад”, “Фармацевт”, “Замовлення”, “Товар” з відповідними полями.

10. Засобами *MS Access* створити схему даних для предметної області “Облік роботи дистриб’юторської мережі іноземної фармацевтичної компанії (NSP), яка розробляє власні біологічні добавки” та створити таблиці “Склад”, “Валюта”, “Дистриб’ютор”, “Спонсор”, “Товар” з відповідними полями.

### Питання для самоконтролю та співбесіди

1. У чому полягає концепція інтегрованої обробки даних?
2. Що таке база даних?
3. Назвіть функції систем управління базами даних.
4. Що таке реляційна база даних?
5. Поясніть відмінність понять “домен” та “тип даних”.
6. Що таке відношення?
7. Що таке ключ?
8. Для чого використовуються зв’язки між таблицями?
9. Для чого виконують нормалізацію бази даних?
10. Дайте визначення першої нормальної форми.
11. Дайте визначення другої нормальної форми.
12. Дайте визначення третьої нормальної форми.
13. Дайте визначення четвертої нормальної форми.
14. Як здійснюється зв’язування таблиць?
15. Для чого використовується майстер підстановки?
16. Що визначає властивість поля *Присоединенный столбец*?
17. Що визначає властивість поля *Число столбцов*?
18. Що визначає властивість поля *Ширина столбцов*?
19. Як визначити для підстановки поле зі списком з виведенням на екран двох стовпців?
20. Як приховати виведення стовпця ключа на екран при використанні підстановки?
21. Як визначити тип зв’язку між таблицями у *MS Access*?
22. Як визначити зв’язок між таблицями у *MS Access*?
23. Що таке цілісність даних у таблицях *MS Access*?
24. Що таке каскадне відновлення та каскадне видалення даних в таблицях *MS Access*?
25. Що таке каскадне відновлення та видалення даних в таблицях *MS Access*?



26. Опишіть процедуру створення таблиці у *MS Access*.
27. Назвіть типи даних, які можна використовувати при визначенні таблиць у *MS Access*.
28. Як визначити маску вводу даних для поля?
29. Як визначити значення по замовчуванню для поля в таблиці?
30. Як здійснити перевірку правильності значень, що користувач вводить у таблицю?
31. Як визначити повідомлення при введенні неправильного значення, що користувач вводить у таблицю?
32. Що таке індекс? Як визначити індексоване поле?
33. Назвіть способи об'єднання таблиць у запитах.
34. Дайте визначення внутрішнього об'єднання таблиць.
35. Дайте визначення лівого об'єднання таблиць.
36. Дайте визначення правого об'єднання таблиць.
37. Наведіть приклади таблиць, для яких результат виконання лівого об'єднання не співпадатиме з результатом виконання внутрішнього об'єднання.
38. Наведіть приклади таблиць, для яких результат виконання правого об'єднання не співпадатиме з результатом виконання внутрішнього об'єднання.

## **Тема 2. Візуальне об'єктно-орієнтоване програмування**

Опрацювання програмного матеріалу (за матеріалами конспекту, підручників).

Об'єкти та класи. Базові класи. Вкладеність об'єктів. Характеристики об'єктно-орієнтованого програмування: інкапсуляція, наслідування і підкласи, поліморфізм. Об'єктно-орієнтоване програмування і OLE-об'єкти.

Типи даних і дії над ними. Прості та складні змінні, їх визначення та область дії. Функції та процедури. Види передавання аргументів функцій і процедур: за значенням, за адресою. Вбудовані функції та процедури. Побудова функцій та процедур користувача в середовищі *MS Access*. Вирази та їх побудова. Основні структури програмування та їх кодування в середовищі *MS Access*: ланцюг, розгалуження, цикл. Використання *Visual Basic* для прикладного програмування: об'єкти, властивості та методи.

Форми. Класифікація форм. Призначення форм. Режими відображення форм. Використання форм для введення даних у режимі










*Режим форми.* Проектування форм у режимі *Конструктор*. Відображення джерела даних за допомогою режиму *Режим таблиць*. Призначення звітів. Розділи звітів. Елементи управління, що використовуються у звітах.

*Література* [1; 3; 4; 8; 9; 17–19; 25; 26; 28]

### Основні визначення

Візуальне об'єктно-орієнтоване програмування в *Access* використовується для створення форм та звітів. Форми є основою інтерфейсу користувача. Звіт — це об'єкт, який призначений для наведення даних з бази у вигляді зручному для їх друку. Зв'язок між формами та звітами і джерелом даних реалізується за допомогою *елементів управління*, які розміщують в області даних. Для додавання елементів до форми використовується *Панель елементів*, яка містить кнопки, перелік і призначення яких наведено у таблиці.

Кнопка панелі елементів	Призначення
<i>Вибір об'єктів</i> 	використовується для виділення елемента керування або групи елементів, розділу, форми або звіту
<i>Мастера</i>  	включає та відключає майстри зі створення елементів керування
<i>Надпись</i> 	використовується для виведення пояснюючого тексту, наприклад заголовок, підпис або інструкція
<i>Поле</i> 	призначене для перегляду даних у джерелі даних (зв'язані поля), для введення та зміни даних у таблицях, для виведення результатів обчислень (вільні та обчислювані поля)
<i>Група переключателів</i> 	використовується для розміщення набору прапорців, перемикачів або вимикачів, що представляють набір альтернативних значень
<i>Выключатель</i> 	елементи керування, які приєднуються до логічних полів бази даних, або використовуються у діалогових вікнах для представлення вибору користувача
<i>Переключатель</i> 	
<i>Флажок</i> 	
<i>Поле со списком</i> 	складений елемент керування, який об'єднує поле та список

<i>Список</i> 	призначений для перегляду даних у джерелі даних (зв'язані поля), для введення значень з фіксованого списку значень (список може бути пов'язаний з деяким полем таблиці або запиту або вільний)
<i>Кнопка</i> 	використовується для виконання дій
<i>Рисунок</i> 	використовується для відображення незмінного малюнка в формі або звіті
<i>Свободная рамка объекта</i> 	використовується для включення в форму об'єктів OLE. При цьому об'єкт OLE стає частиною форми, але не зберігається в БД (малюнок, звук, діаграма, слайд)
<i>Присоединенная рамка объекта</i> 	використовується для відображення у формі або звіті об'єктів OLE. При переході від запису до запису в форму або звіті виводяться різні об'єкти
<i>Набор вкладок</i> 	використовується для створення форми або діалогового вікна з кількома вкладками, на які можна додати інші елементи керування
<i>Подчиненная форма/отчет</i> 	використовується для виведення в формі або звіті даних з кількох таблиць
<i>Линия</i> 	використовуються для відокремлення даних, а також для відокремлення різних розділів форми або сторінки
<i>Прямоугольник</i> 	

Усі властивості основних об'єктів бази даних та елементів керування групують згідно з їх призначенням на групи, кожній з яких відповідає певна вкладка у вікні властивостей: *Макет* — властивості, які встановлюють загальний вигляд елемента або об'єкта; *Данные* — властивості, які стосуються даних, що пов'язані з елементом або об'єктом; *События* — властивості, які дозволяють визначити процедури, що будуть виконані при виникненні певної події; *Другие* — додаткові властивості; на вкладці *Все* розміщено всі властивості.

Обчислювані поля у звітах можуть використовуватись для обчислення значень за окремими записами (поле розміщують в області даних) або за групою записів (поле розміщують у розділах *Залоговок* або *Примечение*).

Для більшості елементів управління визначені наступні властивості:

1. *Имя* — використовується для ідентифікації елемента.

2. *Вывод на экран* — визначає, чи буде елемент відображено на екрані у режимі форми.
3. *Формат поля* — визначає спосіб відображення елемента у режимі форми.
4. *Данные* — визначає, які дані будуть відображені за допомогою елемента управління.
5. *Значение по умолчанию* — значення, яке буде автоматично присвоюватись елементу при додаванні нового запису.
6. *Условие на значение* — визначає умову, якій має задовольняти уводжуване значення.
7. *Сообщение об ошибке* — текст повідомлення, яке буде відображено у вікні повідомлення при порушенні умови щодо уводжуваного значення тощо.

Але окрім наведених властивостей кожний елемент керування має деякі притаманні лише йому властивості. Наприклад, поле зі списком має наступні важливі властивості:

1. *Присоединенный столбец* — вказує, який стовпець приєднано до базового поля. Дані цього стовпця зберігаються у полі при виборі рядка у списку.
2. *Число столбцов* — задає кількість стовпців.
3. *Ширина столбцов* — визначає ширину кожного стовпця.
4. *Источник строк* — визначає дані, які будуть використані як елементи списку.

### **Тематика практичних завдань**

1. Описати візуальні елементи управління та їх властивостей форми *“Клиенты”* бази даних *“Борей”*.
2. Описати візуальні елементи управління та їх властивостей форми *“Типы”* бази даних *“Борей”*.
3. Описати візуальні елементи управління та їх властивостей форми *“Сотрудники”* бази даних *“Борей”*.
4. Описати візуальні елементи управління та їх властивостей форми *“Заказы”* бази даних *“Борей”*.
5. Описати візуальні елементи управління та їх властивостей форми *“Отчеты о продажах”* бази даних *“Борей”*.
6. Описати візуальні елементи управління та їх властивостей звіту *“Список товаров”* бази даних *“Борей”*.
7. Описати візуальні елементи управління та їх властивостей звіту *“Продажи по сотрудникам и странам”* бази даних *“Борей”*.

8. Описати візуальні елементи управління та їх властивостей звіту *“Продажи по типам”* бази даних *“Борей”*.
9. Описати візуальні елементи управління та їх властивостей звіту *“Суммы продаж по годам”* бази даних *“Борей”*.
10. Описати візуальні елементи управління та їх властивостей звіту *“Счет”* бази даних *“Борей”*.

### Питання для самоконтролю та співбесіди

1. Для чого призначені елементи управління?
2. Які елементи управління використовуються при створенні форм в *MS Access*?
3. Яка властивість елемента управління використовується для ідентифікації елемента?
4. Яка властивість елемента управління визначає, чи буде елемент відображено на екрані у режимі форми?
5. Для чого призначена властивість *Вывод на экран*?
6. Як використовується властивість *Формат поля*?
7. Які формати полів можна задавати за допомогою властивості *Формат поля*?
8. Для чого призначена властивість *Данные*?
9. Як використовується властивість *Значение по умолчанию*?
10. Як використовується властивість *Условие на значение*?
11. Що відбудеться при порушенні умови на значення, яка визначена у полі *Условие на значение*?
12. Для чого призначена властивість *Сообщение об ошибке*?
13. Як використовується властивість *Присоединенный столбец*?
14. Для якого елемента управління можна визначити властивість *Число столбцов* та як вона використовується?
15. Як використовується властивість *Ширина столбцов*?

### Тема 3. Об'єкти та їх побудова

**Опрацювання програмного матеріалу** (за матеріалами конспекту, підручників)

Принципи об'єктно-орієнтованого програмування. Основні поняття об'єктно-орієнтованого програмування: об'єкт, властивість, метод, подія. Використання процедур та функцій. Події та процедури обробки подій.

Об'єкти *MS Access*. Правила іменування об'єктів. Властивості об'єктів. Головне вікно *MS Access*. Режими перегляду об'єктів. Використання основних об'єктів *MS Access* для обробки даних. Проектування об'єктів у режимі *Конструктор*.

Поняття макросу. Призначення макросів. Виконання макросів. Створення макросів. Режим *Конструктор*. Макрокоманди. Призначення основних макрокоманд. Визначення параметрів макрокоманди. Умовні макрокоманди. Правила визначення умов для макрокоманд. Використання груп макросів. Правила визначення імен макросів та груп макросів. Використання модулів. Модулі класу. Стандартні модулі.

*Література* [1; 3; 4; 8; 9; 17–19; 25; 26; 28]

### **Основні визначення**

*Microsoft Access* — це система управління базами даних реляційного типу, за допомогою якої можна швидко розробити програмне забезпечення для збереження та аналізу даних. *MS Access* належать до програмного забезпечення класу *RAD* (*Rapid Application Development*), що можна перевести як “швидка розробка застосувань”. У файлі *MS Access* зберігаються дані разом з процедурами їх обробки у вигляді одного файлу з розширенням \*.mdb.

Основними об'єктами бази даних *MS Access* є:

- *Таблиці*, які використовуються для збереження даних;
- *Запити*, які використовуються для вибірки даних з таблиць та їх обробки;
- *Форми*, які використовуються для перегляду, введення та зміни даних в таблицях, а також для запуску процедур обробки даних;
- *Звіти*, які призначені для виведення даних на друк;
- *Макроси та модулі* призначені для виконання операцій, що визначені розробником. Макросом називають набір з однієї або кількох макрокоманд, кожна з яких виконує деяку операцію. Використання макросів є одним із способів побудови програмного забезпечення. Зазвичай запуск макросу здійснюється у відповідь на виникнення деякої події.

Кожний об'єкт бази даних *Access* у межах свого класу має унікальне ім'я. Робота з усіма об'єктами бази даних здійснюється у двох режимах:

1. У режимі *даних* — використання об'єктів для обробки даних користувачем бази даних. У цьому режимі вводяться дані в таблиці та форми, виконуються запити, звіти тощо. Перехід у цей режим здійснюється при відкритті довільного об'єкта.

2. У режимі *структури* — створення та зміна об'єктів (таблиць, форм, запитів, звітів, модулів, макросів). Перехід у цей режим здійснюється за допомогою команди *Конструктор*. Зміна властивостей об'єкта в режимі *Конструктор* здійснюється за допомогою діалогового вікна *Свойства*.

Розробка застосування у *MS Access* починається зі створення таблиць та схеми даних. Після того за потребою створюють запити, форми та звіти.

Макросом називають набір із однієї або більше макрокоманд, що виконують певні операції. Макрокомандою називають замкнену інструкцію, яка визначає виконання деякої операції. Макроси використовуються для виконання таких дій:

- 1) відкриття (закриття) об'єктів *MS Access* — таблиць, запитів, форм, звітів, макросів;
- 2) обчислення та встановлення значення елементів управління форм, звітів;
- 3) фільтрація даних;
- 4) виконання будь-якої команди будь-якого меню *MS Access*;
- 5) переходи між вікнами та стандартні дії з ними (переміщення, зміна розмірів тощо);
- 6) виведення на екран інформаційних повідомлень;
- 7) копіювання, перейменування, видалення об'єктів *MS Access*;
- 8) імпорт (експорт) таблиць баз даних, електронних таблиць, текстових файлів;
- 9) запуск програмного забезпечення і передача йому параметрів.

Макроси можуть бути об'єднані у групи. Ім'я макроса задається у стовпці *Имя макроса*, який виводиться на екран за допомогою команди *Вид, Имя макроса*. Для запуску макроса з групи вказується ім'я групи, а потім після точки — ім'я макроса. При виконанні макроса у групі макросів виконується команда в стовпці макрокоманд та всі наступні макрокоманди, для яких стовпець *Имя макроса* є порожнім.

### Тематика практичних завдань

1. Загальна характеристика схеми даних навчальної бази даних “*Борей*”.

2. Загальна характеристика таблиць навчальної бази даних “*Борей*”.
3. Загальна характеристика форм навчальної бази даних “*Борей*”.
4. Загальна характеристика запитів навчальної бази даних “*Борей*”.
5. Загальна характеристика звітів навчальної бази даних “*Борей*”.
6. Автоматизація дій користувача за допомогою групи макросів “*Итоги продаж*” бази даних “*Борей*”.
7. Автоматизація дій користувача за допомогою групи макросів “*Клиенты*” бази даних “*Борей*”.
8. Автоматизація дій користувача за допомогою групи макросів “*Наклейки для клиентов*” бази даних “*Борей*”.
9. Автоматизація дій користувача за допомогою групи макросів “*Поставщики*” бази даних “*Борей*”.
10. Автоматизація дій користувача за допомогою групи макросів “*Телефоны клиентов*” бази даних “*Борей*”.

### **Питання для самоконтролю та співбесіди**

1. Охарактеризуйте призначення таблиць в *MS Access*.
2. Охарактеризуйте призначення форм в *MS Access*.
3. Охарактеризуйте призначення запитів в *MS Access*.
4. Охарактеризуйте призначення звітів в *MS Access*.
5. Як визначити пароль для бази даних в *MS Access*?
6. Дайте характеристику об'єктам *MS Access*.
7. Як здійснюється іменування об'єктів у *MS Access*?
8. Охарактеризуйте призначення модулів в *MS Access*.
9. Які об'єкти в *MS Access* використовуються для збереження даних?
10. Які об'єкти в *MS Access* використовуються для аналізу даних?
11. Що таке об'єкт, властивість, подія?
12. Як в *Access* можна визначити процедуру обробки події?
13. Наведіть приклад процедури обробки події для форми?
14. Наведіть приклад процедури обробки події для звіту?
15. Що таке макрос?
16. Для чого використовуються макроси?
17. Що таке макрокоманда?
18. Назвіть основні макрокоманди.
19. Яка макрокоманда використовується для встановлення значення властивості об'єктів?
20. Як вивести повідомлення за допомогою макрокоманди?



21. Як визначити умову виконання макрокоманди?
22. Як створити макрос для перевірки коректності уведених значень?
23. Як у формі створити кнопки для виконання фільтрації записів за умовою?
24. Як змінити фокус у поточній формі за допомогою макрокоманди?
25. Як створити групу макросів?

#### ***Тема 4. Технології побудови інтерфейсу інформаційної системи***

**Опрацювання програмного матеріалу** (за матеріалами конспекту, підручників).

Планування спеціальних програмних засобів на основі СУБД. Технології побудови інтерфейсу в *MS Access*: побудова спеціальної панелі інструментів для форм; побудова спеціального рядка меню для форм; визначення початкової кнопкової форми.

Використання форм для введення і редагування даних. Панель інструментів *Режим форми*. Пошук, сортування та фільтрація даних у формі. Запис фільтра у вигляді запити. Експорт результатів виконання фільтрації.

Створення макета форми. Розділи форм. Елементи управління. Властивості елементів управління. Майстри побудови елементів управління. Майстер побудови списків, полів зі списками, майстер створення групи перемикачів, майстер створення кнопок.

Підпорядковані форми. Зв'язані форми. Наведення пов'язаних таблиць за допомогою підпорядкованих форм. Майстер побудови підпорядкованих форм. Створення підпорядкованої форми на основі кількох таблиць або запитів. Додавання підпорядкованої форми.

Майстри побудови форм: *Автоформа в столбці*, *Автоформа ленточная*, *Автоформа табличная*. Використання обчислювальних елементів управління у формах. Розрахунок підсумкових значень за групою записів підпорядкованої форми. Створення кнопчних форм. Використання засобу *Диспетчер кнопчних форм*. Побудова діалогових форм. Уведення параметрів запитів за допомогою діалогових форм.

Сортування та групування даних у звітах. Зміна параметрів сортування та групування у діалоговому вікні *Сортування і групування*.

Створення звітів за допомогою майстрів побудови звітів. Редагування звіту у режимі *Конструктор*. Обчислення у звітах. Обчислювальні поля за одним записом. Обчислювальні поля для груп записів. Використання властивості *Сумма с накоплением* для групи та для всіх записів звіту.

*Література* [1; 3; 4; 8; 9; 17–19; 25; 26; 28]

### Основні визначення

Форми є основою інтерфейсу користувача. Форми виконують дві важливі функції в базах даних:

- 1) дають змогу вводити та редагувати дані в таблицях та запитах (форми даних);
- 2) дають змогу здійснювати управління роботою програми (кнопкові та діалогові форми).

Підпорядкована форма — це форма, яка пов'язана з іншою (головною) формою за деякою сукупністю полів. Підпорядковані форми використовуються для наведення даних таблиць та запитів, які пов'язані відношенням “один до багатьох”, причому головна форма представляє сторону “один”, а підпорядкована форма — сторону “багато”. Відображення даних у підпорядкованій формі синхронізується з головною формою таким чином, що у підпорядкованій формі відображаються тільки ті записи, які пов'язані з поточним записом головної форми.

Структура форми складається з таких розділів:

- *заголовок форми* — відображається у верхній частині форми, зазвичай використовується для розміщення тексту заголовка форми та інструкцій щодо роботи з формою;
- *примечание форми* — відображається у нижній частині форми, призначення аналогічне до заголовку форми;
- *область данных* — використовується для відображення даних.

Кнопкова форма — це форма, яка містить кнопки, при натисканні яких здійснюється виконання визначених на етапі розробки дій, таких як виконання запити, відкриття звіту, виконання макросу тощо. Для створення кнопкових форм використовується *Диспетчер кнопкових форм (Сервис, Службные программы, Диспетчер кнопковых форм)*.

Звіт — це об'єкт, який призначений для наведення даних з бази у вигляді, зручному для їх друку. Роздрукувати дані без обробки можна безпосередньо з таблиці (кнопка “*Печать*” на панелі інструментів). Звіти використовуються, коли перед друком необхідно згрупувати

дані за деякими полями, обчислити проміжні та підсумкові значення, коли необхідно оформити колонтитули сторінок тощо. Джерелом даних для звіту є таблиці та запити. Зв'язок між звітом і джерелом даних, подібно до форм, здійснюється за допомогою елементів управління. Звіти зручніше створювати за допомогою майстра з подальшим редагуванням у режимі *Конструктор*.

Зміна структури звіту здійснюється у режимі *Конструктор*. У режимі конструктора звіт містить такі розділи:

- *Заголовок отчета* — вміст розділу виводиться лише на першій сторінці звіту;
- *Верхній колонтитул* — вміст розділу виводиться угорі кожної сторінки звіту;
- *Заголовки групи* — необов'язковий розділ, який використовується при групуванні даних, виводиться угорі кожної групи;
- *Примечание группы* — необов'язковий розділ, який використовується при групуванні даних, виводиться унизу кожної групи;
- *Область даних* — у цьому розділі розміщують елементи управління, які забезпечують зв'язок звіту з джерелом даних;
- *Нижній колонтитул* — вміст розділу виводиться унизу кожної сторінки звіту;
- *Примечание отчета* — вміст розділу виводиться лише наприкінці звіту.

Для відображення заголовка та приміток звіту, колонтитулів необхідно у вікні конструктора у меню “Вид” увімкнути перемикачі “Колонтитули” та “Заголовок/примечание отчета”.

### Тематика практичних завдань

1. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Кредитування на придбання автомобіля” та кнопку форму.
2. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Відправлення грошових переказів у гривнях” та кнопку форму.
3. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Облік використання фондів стаціонару медичної установи” та кнопку форму.
4. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Облік аналізів крові пацієнтів” та кнопку форму.

5. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Облік бюджету медичної установи” та кнопку форму.
6. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Облік переліку та вартості процедур, що відпускаються у медичних закладах санаторно-курортного типу” та кнопку форму.
7. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Облік замовлень на використання виробничих ліній для таблетування різних лікарських форм” та кнопку форму.
8. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Облік замовлень на отримання лікарських розчинів лікарнею” та кнопку форму.
9. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Облік роботи аптечної бази” та кнопку форму.
10. Засобами *MS Access* створити засоби введення даних та підготовки звітів для бази даних “Облік роботи дистриб’юторської мережі іноземної фармацевтичної компанії, яка розробляє власні біологічні добавки” та кнопку форму.

### **Питання для самоконтролю та співбесіди**

1. Для чого призначені форми?
2. Опишіть способи створення форм у *MS Access*.
3. Як виконується пошук у формах *MS Access*?
4. Як здійснити фільтрацію у формах *MS Access*?
5. Які розділи містить форма?
6. Для чого використовується заголовок форми?
7. Для чого використовується примітка форми?
8. Для чого використовується область даних у формах?
9. Чим відрізняються “*табличная форма*” та “*ленточная форма*”?
10. Чим відрізняються “*форма: в один столбец*” та “*ленточная форма*”?
11. Як створити форму з використанням полів кількох таблиць?
12. Як змінити джерело записів для форми?
13. Як зв’язуються головна та підпорядкована форми?
14. Що таке підпорядкована форма?
15. Опишіть способи створення підпорядкованих форм.

16. Що таке кнопкові форми? Як створити кнопку форму у *MS Access*?
17. Як використовується “Диспетчер кнопочных форм”?
18. Що таке діалогові форми? Як створити діалогову форму у *MS Access*?
19. Назвіть розділи звіту та опишіть їх призначення.
20. Чим відрізняються розділи звіту: колонтитул та заголовок групи?
21. Чим відрізняються розділи звіту: колонтитул та примітка групи?
22. Чим відрізняються розділи звіту: верхній колонтитул та заголовок звіту?
23. Чим відрізняються розділи звіту: нижній колонтитул та заголовок звіту?
24. Які властивості можна встановити для області даних звіту?
25. Як створити обчислюване поле в звіті?
26. Як обчислити значення за групою записів у звіті?
27. Як вивести номер сторінки у звіті?
28. Як обчислити кількість записів у кожній групі звіту?
29. Як здійснити нумерацію записів у групі звіту?
30. Як додати групування у звіті?
31. Як на першій сторінці звіту вивести дату друку звіту?
32. Як визначити назву стовпця для групи зі звіту?
33. Як обчислити середнє значення для деякого поля за всіма записами групи?
34. Як обчислити середнє значення для деякого поля за всіма записами звіту?
35. Як обчислити для кожного запису звіту процент, який складає значення поля до загальної суми за всіма записами звіту?
36. Назвіть етапи створення звітів за допомогою майстра.
37. Назвіть основні прийоми редагування звітів у режимі *Конструктор*.
38. Як здійснюються обчислення у звітах?
39. Як здійснюється групування записів у звітах?

### **Типові практичні завдання до змістового модуля І**

1. Спроекувати реляційну схему даних “Кредитування на придбання автомобіля”.
2. Спроекувати реляційну схему даних “Кредитування на навчання”.

3. Спроекувати реляційну схему даних “Кредитування на придбання квартири”.
4. Спроекувати реляційну схему даних “Кредитування на придбання квартири із щомісячною виплатою”.
5. Спроекувати реляційну схему даних “Кредитування на придбання квартири із щоквартальною виплатою”.
6. Спроекувати реляційну схему даних “Кредитування стоматологічних послуг”.
7. Спроекувати реляційну схему даних “Короткострокове кредитування”.
8. Спроекувати реляційну схему даних “Відправлення грошових переказів у гривнях”.
9. Спроекувати реляційну схему даних “Відправлення грошових переказів у валюті”.
10. Спроекувати реляційну схему даних “Облік використання фондів стаціонару медичної установи”.
11. Спроекувати реляційну схему даних “Облік клінічних і біохімічних аналізів крові пацієнтів”.
12. Спроекувати реляційну схему даних “Облік бюджету медичної установи”.
13. Спроекувати реляційну схему даних “Облік призначень для лікування пацієнта та вартості лікування у медичних закладах санаторно-курортного типу”.
14. Спроекувати реляційну схему даних “Облік процедур, що відпускаються у медичних закладах санаторно-курортного типу”.
15. Спроекувати реляційну схему даних “Облік замовлень на використання виробничих ліній для таблетування різних лікарських форм”.
16. Спроекувати реляційну схему даних “Облік замовлень на отримання лікарських розчинів лікарнею”.
17. Спроекувати реляційну схему даних “Аптечна база”.
18. Спроекувати реляційну схему даних “Дистриб’юторська мережа іноземної фармацевтичної компанії”.
19. Спроекувати реляційну схему даних “Нарахування заробітної плати дистриб’ютора іноземної мережі”.

## Змістовий модуль II. Засоби аналізу даних в інформаційних системах

### Тема 5. Мова запитів SQL

**Опрацювання програмного матеріалу (за матеріалами конспекту, підручників).**

Запити. Класифікація запитів. Запити на вибірку. Вибірка даних із кількох таблиць. Способи об'єднання запитів з кількох таблиць. Повне об'єднання. Внутрішнє та зовнішнє об'єднання. Ліве зовнішнє об'єднання та праве зовнішнє об'єднання. Зміна типу об'єднання таблиць.

SQL і управління реляційними базами даних. Використання SQL для побудови і наповнення бази даних. Оператор SELECT. Сортування, видалення інформації, що повторюється, використання спеціальних функцій для обчислень. Групування даних і побудова звітів. Обробка невизначених значень. Об'єднання таблиць, аналіз даних. Запити SQL, структуровані запити.

Режими перегляду запитів. Правила конструювання запиту за допомогою режиму *Конструктор*. Перегляд результатів виконання запиту у режимі *Режим таблиць*. Перегляд запиту у вигляді SQL-конструкції – режим *в виді SQL*. Створення запитів за допомогою майстрів побудови запитів.

Використання виразів при побудові запитів. Правила побудови виразів. Побудова виразів за допомогою *Построителя виражений*. Правила визначення у виразах об'єктів та їх властивостей. Використання стандартних функцій у виразах.

Визначення умов відбору записів за допомогою логічних виразів. Логічні оператори AND, OR, BETWEEN, IN, LIKE. Правила побудови складних логічних умов за кількома полями запиту.

Обчислення у запитах. Обчислювальні поля. Правила побудови обчислювальних полів. Визначення імен обчислювальних полів. Побудова запитів з параметрами. Використання властивостей запиту.

Групування даних у запитах та використання агрегатних функцій. Використання підзапитів. Використання майстрів для побудови запитів. Побудова простого запиту. Обчислення підсумкових значень за допомогою майстра побудови простих запитів.

Перехресні запити. Побудова перехресного запиту за допомогою майстра. Формування та редагування перехресного запиту у режимі *Конструктор*.

Запит *Повторяючіся записи*. Побудова запиту *Повторяючіся записи* за допомогою майстра. Засоби побудови запиту *Повторяючіся записи* у режимі *Конструктор*.

Пошук записів без підпорядкованих. Побудова запиту *Записи без підчинених* за допомогою майстра. Засоби побудови запиту *Записи без підчинених* у режимі *Конструктор*.

Запити-дії. Запити на оновлення даних. Запити на знищення записів. Запити на додавання записів. Запити на створення нової таблиці.

*Література* [1–4; 6; 8; 9; 13; 15; 18; 19; 25; 26; 28]

### **Основні визначення**

*Запит* — це об'єкт, за допомогою якого здійснюється перегляд, зміна та аналіз бази даних так, як це визначено користувачем. За допомогою запиту можна вибрати, змінити або згрупувати дані, які містяться в одній або кількох таблицях. За допомогою запитів здійснюється проведення обчислень та підбиття підсумків. Запит можна використати як основу для інших запитів або як джерело записів для форм і звітів.

Використовують кілька основних типів запитів:

- 1) запити на вибірку даних;
- 2) перехресні запити;
- 3) запити на зміну: створення таблиці, знищення, оновлення, додавання записів;

Запит на вибірку дає змогу особливим чином навести дані, що зберігаються у таблицях. Результат виконання запиту має вигляд таблиці та називається динамічним набором записів. Динамічний набір записів не зберігається в базі, але формується кожного разу при запуску запиту на виконання.

Запити на вибірку можуть бути переглянуті у таких режимах:

- 1) у режимі конструктора (*Конструктор*);
- 2) у режимі таблиці (*Режим таблиць*);
- 3) у вигляді SQL — Structured Query Language (*Режим SQL*).

У запитах часто використовуються арифметичні та логічні вирази для обчислення значень та визначення умов відбору записів. Для побудови виразів використовується засіб *Построитель выражений*. У рядку "*Условие отбора*" визначають деякий логічний вираз відбору записів. Якщо для деякого запису значення виразу — істина, то запис додається до результуючого набору, у протилежному разі — не додається.



ся. Логічна умова відбору даних за кількома полями складається з кількох простих умов, які об'єднані за допомогою операторів *And* (и) або *Or* (или), причому

- якщо умови задані в одному рядку, то буде використано оператор *And*;
- якщо умови задані в різних рядках, то буде використано оператор *Or*.

Логічні оператори можуть використовуватись також для об'єднання логічних виразів для одного поля. Оператор *BETWEEN* використовується для визначення належності значення виразу вказаному діапазону. За допомогою оператора *In* здійснюється перевірка, чи співпадає значення виразу з одним із елементів вказаного списку. За допомогою оператора *Like* можна задати шаблон, якому має відповідати значення текстового виразу.

Відібрані у запиті записи можуть бути згруповані з метою обчислення агрегатних функцій над кожною групою записів. Наприклад, співробітників можна згрупувати за посадою та для кожної групи записів обчислити сумарну заробітну плату.

Так звані запити-дії використовуються для внесення зміни в базу даних. За допомогою таких запитів можуть бути оновлені або видалені групи записів, додані дані в таблицю або створена нова таблиця. У режимі *Таблиця* з меню *"Вид"* можна переглянути записи, над якими будуть виконані дії. Для виконання запиту на зміну даних необхідно запустити його за допомогою кнопки *Запуск* панелі інструментів *Конструктор запитов*.

### Тематика практичних завдань

Запит 1. Визначити, на яку суму (з урахуванням знижки) зробив замовлення товарів типу *"молочные продукты"* кожний із співробітників, хто пропрацював на фірмі хоча б 10 років.

Запит 2. Визначити категорії товарів, що поставлялись у банках з США або з Канади.

Запит 3. Визначити десять найдорожчих замовлень, що були зроблені у 1998 році.

Запит 4. Визначити середню суму замовлень товарів (з урахуванням знижки та вартості доставки) для кожної країни замовника та типу товару (перехресний запит).

Запит 5. Визначити перелік товарів типу *"Кондитерские изделия"*, які повторно були поставлені клієнтам з Бразилії.

Запит 6. Знайти марки товарів типу “*Молочные продукты*” або “*Рыбопродукты*”, які постачаються у пакетах.

Запит 7. Визначити, яку кількість замовлень обслужив кожний із співробітників, вік якого більше 35 років, та на яку суму (з урахуванням знижки).

Запит 8. Визначити 10 країн, до яких було поставлено поштою товарів на найбільшу суму (з урахуванням знижки та вартості доставки).

Запит 9. Знайти постачальників та співробітників з однакових країн.

Запит 10. Визначити, яку кількість замовлень обслужив кожний із співробітників, який влаштувався на роботу у другому півріччі 1993 року, та на яку суму (з урахуванням знижки).

Запит 11. Знайти марки товарів типу “*Кондитерские изделия*” або “*Мясо/птица*”, які постачаються у коробках.

Запит 12. Для кожного типу товарів визначити розмір максимальної знижки.

Запит 13. Визначити перелік товарів типу “*Молочные продукты*”, які повторно були поставлені клієнтам з *Австрії*.

Запит 14. Визначити кількість та суму замовлень, які були розміщені у різні роки.

Запит 15. Знайти співробітників, які замовили товарів на суму (з урахуванням знижки) більше середнього.

Запит 16. Визначити кількість та суму замовлень, які доставлені кожним із видів доставки.

Запит 17. Сформуванати перелік клієнтів та категорій товарів, які вони не замовляли.

Запит 18. Сформуванати перелік марок товарів, які не доставляли до США та Канади.

Запит 19. Знайти постачальників та клієнтів з однакових міст.

Запит 20. Визнати, на яку загальну суму замовлено товарів кожного типу до кожної країни клієнта (перехресний запит).

Запит 21. Визначити, яка кількість товарів зберігається на складі для кожного типу товару та кожної країни постачальника (перехресний запит).

Запит 22. Створити таблицю *Таблиця 1*, до якої додати записи про замовлення товарів (код замовлення, марка товару, категорія товару, назва постачальника та ціна товару), тип яких “*Кондитерские изделия*”, ціна перевищує 2000 р., та замовлення на які були розміщені у 1997 році.

Запит 23. У всіх записах таблиці *Таблиця1* змінити назву постачальника *Forots d'Erables* на *Forots Erables*.

Запит 24. Створити таблицю *Таблиця2*, до якої розмістити інформацію про товари (марка товару, категорія товару, назва постачальника та ціна товару), тип яких "*Фрукты*", ціна перевищує 2000 р, та замовлення на які були розміщені у 1998 році.

Запит 25. З таблиці *Таблиця2* видалити всі записи про товари, ціна яких більше 2300 р.

### Питання для самоконтролю та співбесіди

1. Що таке запит?
2. Які види запитів існують в *MS Access*?
3. Як створити запит на основі фільтра?
4. Як створити запит на вибірку даних за допомогою майстра побудови простих запитів?
5. Як створити запит в режимі *Конструктор*?
6. Що таке обчислюване поле?
7. Правила побудови виразів у *MS Access*.
8. Побудова виразів засобами *Построителя выразений*.
9. Використання вбудованих функцій у майстрі побудови виразів.
10. Використання фінансових функцій у майстрі побудови виразів.
11. Використання функцій *Дата/Время* у майстрі побудови виразів.
12. Використання статистичних функцій у майстрі побудови виразів.
13. Використання статистичних функцій у майстрі побудови виразів.
14. Використання текстових функцій у майстрі побудови виразів.
15. Як формуються прості умови відбору записів?
16. Як відібрати 15 записів, що мають найбільші значення за деяким полем?
17. Як формуються складені умови відбору записів?
18. Як знайти унікальні значення деякого поля таблиці?
19. Як здійснюється сортування записів у запиті?
20. Які агрегатні функції можна використовувати в запитах?
21. Для чого використовується конструкція *SELECT*?
22. Для чого використовується службове слово *AS*?
23. Для чого використовується службове слово *FROM*?
24. Для чого використовується конструкція *UNION*?
25. Для чого використовується конструкція *ORDER BY*?

26. Для чого використовується конструкція GROUP BY?
27. Як мовою SQL визначається поле, за яким здійснюється об'єднання таблиць?
28. Як мовою SQL визначається умова відбору записів?
29. Як мовою SQL визначається групування записів?
30. Що таке запит з параметром?
31. Як здійснюється побудова обчислювальних полів у запитах?
32. Наведіть приклади запитів з використанням логічного оператора AND.
33. Наведіть приклади запитів з використанням логічного оператора OR.
34. Наведіть приклади запитів з використанням логічного оператора BETWEEN.
35. Наведіть приклади запитів з використанням логічного оператора IN.
36. Наведіть приклади запитів з використанням логічного оператора LIKE.
37. Як у запитах здійснюється групування даних?
38. Опишіть створення простого запиту за допомогою майстра.
39. Опишіть створення підсумкового запиту за допомогою майстра.
40. Для чого використовуються перехресні запити?
41. Опишіть створення перехресного запиту за допомогою майстра.
42. Як створити перехресний запит у режимі конструктора?
43. Опишіть створення запиту на пошук записів, що повторюються, за допомогою майстра.
44. Опишіть створення запиту на пошук записів без підлеглих за допомогою майстра.
45. Для чого використовуються запити-дії?
46. Назвіть типи запитів-дій.
47. Наведіть приклад запиту на створення таблиці
48. Наведіть приклад запиту на додавання записів в таблицю
49. Наведіть приклад запиту на видалення записів з таблиці
50. Наведіть приклад запиту на оновлення записів у таблиці

## ***Тема 6. Організація експорту та імпорту даних***

Стандарт ODBC. Порівняння зв'язування та імпорту. Імпорт з баз даних файлів певної структури (dBase, Paradox, SQL і таке інше). Імпорт з електронних таблиць. Імпорт текстових файлів. Зв'язування

файлів. Експорт даних в таблиці інших баз даних (dBase, Paradox, SQL і таке інше). Експорт в електронну таблицю. Використання даних таблиць у документах MS Word та MS Excel.

*Література* [1; 2; 6; 9; 17; 19; 25; 26]

### **Основні визначення**

Для створення та заповнення даними таблиць можна також використовувати функцію імпортування даних *MS Access*. Імпортування даних передбачає створення у поточній базі даних об'єкта (таблиці, запиту, звіту тощо) на основі деякого об'єкта, що зберігається в іншій базі даних чи в електронній книзі *MS Excel*. При цьому якщо таблиця зберігається у книзі *MS Excel*, то перший рядок таблиці *Excel* повинен містити назви полів таблиці.

Імпортування таблиці із зовнішнього файлу *MS Excel*.

1. Активізувати послугу головного меню *MS Access* Файл *Внешние данные* Імпорт...
2. У діалоговому вікні "*Импорт*", що з'явиться, вибрати тип файлів *Microsoft Excel (\*.xls)*, та вибрати файл зовнішньої електронної книги, що містить дані для таблиці. У результаті буде активовано майстер імпортування даних, причому на екрані з'явиться діалогове вікно майстра.
3. На першому кроці майстра (у першому вікні) потрібно вибрати з переліку сторінку електронної книги *Excel*, на якій зберігаються дані для імпорту, та натиснути кнопку *Далее*.
4. На другому кроці роботи майстра потрібно встановити відмітку біля пункту *Первая строка содержит заголовки столбцов* та натиснути кнопку *Далее*.
5. На третьому кроці роботи майстра потрібно вибрати з переліку таблицю поточної бази даних *Access*, у якій потрібно зберегти імпортовані дані, та натиснути кнопку *Готово*. При цьому дані з електронної таблиці будуть скопійовані у вибрану таблицю бази даних.

### **Тематика практичних завдань**

1. Виконати експорт даних у *MS Excel* запиту "*Список товаров*" навчальної бази даних "*Борей*". Виконати аналіз даних засобами *MS Excel*: відобразити на діаграмі для кожної категорії товарів загальну кількість товарів, що зберігаються на складі.

2. Виконати експорт даних у *MS Excel* запиту “Список товарів” навчальної бази даних “Борей”. Виконати аналіз даних засобами *MS Excel*: відобразити на діаграмі максимальну ціну товарів для кожної категорії товарів.
3. Виконати експорт даних у *MS Excel* запиту “Продажи товарів в 1997” навчальної бази даних “Борей”. Виконати аналіз даних засобами *MS Excel*: відобразити на діаграмі загальну суму продажів товарів у кожному кварталі.
4. Виконати експорт даних у *MS Excel* запиту “Продажи товарів в 1997” навчальної бази даних “Борей”. Виконати аналіз даних засобами *MS Excel*: відобразити на діаграмі загальну суму продажу товарів кожної марки.
5. Виконати експорт даних в *MS Excel* запиту “Продажи товарів в 1997” навчальної бази даних “Борей”. Виконати фільтрацію даних засобами *MS Excel*: відобразити товари, сума продажу яких більше 30000 р., або замовлення виконано у I або II кварталі.

#### **Питання для самоконтролю та співбесіди**

1. Дайте загальну характеристику стандарту ODBC.
2. Назвіть найпоширеніші формати баз даних.
3. Як здійснити імпорт таблиць в *MS Access*?
4. Як здійснити аналіз даних в *MS Excel*?
5. Як використовують дані таблиць у документах *MS Word*?

#### **Тема 7. Забезпечення роботи інформаційної системи в мережі**

Планування роботи. Типи блокування. Встановлення таблиць для блокування. Блокування записів. Відміна блокування таблиць та записів. Використання сеансів роботи. Використання буферів при редагуванні даних. Використання транзакцій. Обробка мережних помилок.

Загальні відомості про засоби захисту *MS Access*. Визначення пароля для бази даних. Створення *mde*-файлів. Використання майстра захисту. Налаштування середовища *MS Access*. Визначення параметрів запуску програми у *MS Access*. Адміністрування бази даних. Стиснення та відновлення файлів *MS Access*.

*Література* [1; 2; 6; 9; 17; 19; 25; 26]

## Основні визначення

У *MS Access* реалізовано такі механізми керування доступом до бази даних:

- Кодування та декодування — файл бази даних стискається та стає недоступним для читання за допомогою службових програм та текстових редакторів.
- Відображення та скриття об'єктів у вікні бази даних.
- Використання параметрів запуску — визначають настройки запуску застосування бази даних.
- Встановлення пароля для відкриття бази даних.
- Використання захисту на рівні користувачів.

Захист на рівні користувачів дозволяє встановити різні рівні доступу до важливих даних та об'єктів у базі даних. Інформація щодо користувачів зберігається у файлі робочої групи.

### Питання для самоконтролю та співбесіди

1. Які засоби захисту даних реалізовано у *MS Access*?
2. Як визначити пароль бази даних у *MS Access*?
3. Як створити *mde*-файл?
4. Які параметри запуску програми можна визначити засобами *MS Access*?
5. Як здійснити стиснення даних у *MS Access*?

### Типові практичні завдання до змістового модулю II

- I. Створити базу даних обліку інформації про співробітників фірми.
1. У базі даних має бути наведена наступна інформація про співробітника:
    - прізвище, ім'я, по батькові;
    - посада, оклад для визначеної посади, персональна надбавка для цього працівника, дата найма співробітника, робочий телефон;
    - номер паспорта, дата народження, місто проживання, адреса проживання, домашній телефон, фотографія;
    - назва філіалу, в якому працює співробітник, ідентифікаційний код філіалу, місто розташування філіалу, адреса філіалу, телефон офісу.
  2. Створити форми даних:
    - форму *Співробітник* для подання інформації про співробітників фірми (поля розмістити на трьох вкладках *Дані про співробітника*, *Особисті дані*, *Фото*);

- форму *Філіали* для інформації про філіали фірми;
  - у формі *Філіали* розмістити підпорядковану форму *підпорядкована форма Співробітник* для наведення службової інформації про співробітників філіалу (за допомогою кнопки *Додаткова інформація* реалізувати відкриття форми *Співробітник* для обраного співробітника);
  - форму *Посади* для наведення інформації про посади фірми;
  - форму *Міста* для представлення переліку міст.
3. У формі *Співробітник* та *підпорядкована форма Співробітник* розмістити обчислювальне поле *заробітна плата*, яка складається з окладу посади, надбавки за вислугу років — 50 грн. за кожні 5 років служби та персональної надбавки. У формі *Філіали* розмістити обчислювальне поле для наведення сукупного фонду заробітної плати філіалу. Створити головну кнопочку форму, з якої здійснювати виклик форм *Філіали*, *Співробітник*, *Посади*, *Міста*. Заповнити базу даних інформацією про 7–10 філіалів з різних міст, у кожному із яких працює 7–10 співробітників (на 5–7 посадах).
  4. Створити звіт *Кадри*, в якому навести перелік співробітників у кожному із філіалів (назва та ідентифікаційний код філіалу, прізвище, ім'я, по батькові співробітника, посада, вік та стаж роботи, впорядкувати за прізвищем за зростанням) з нумерацією співробітників для кожного із філіалів. Визначити середній вік та стаж співробітників у кожному із філіалів та в цілому по фірмі.
  5. Створити звіт *Зарплата* (параметром звіту є дата видачі), в якому навести перелік співробітників по кожному із філіалів (прізвище, ім'я, по батькові, номер паспорта, заробітна плата (з урахуванням надбавок), податок (13 % на всю суму), сума на руки, впорядкувати за прізвищем за зростанням) з нумерацією співробітників для кожного із філіалів. Визначити загальний фонд заробітної плати у кожному філіалі та по фірмі в цілому.
  6. Створити діалогову форму *Звіти* для друку звітів, в якій користувач обирає тип звіту *Кадри* або *Зарплата*, а також може обрати назву філіалу, для якого буде створено звіт (якщо назву філіалу не обрано, то звіт формується для всіх філіалів). Якщо обрано тип звіту *Зарплата*, то користувач має визначити місяць та рік видачі. Форма також містить кнопки *Перегляд* для



перегляду звіту, *Друк* для друку звіту та *Отмена* для виходу у головну кнопкову форму. Реалізувати виклик форми *Звіти* із головної кнопкової форми.

II. Створити базу даних обліку інформації про поставки товарів на склад.

1. У базі даних має бути наведена наступна інформація про товар:
  - назва, категорія товару, ціна товару, одиниця виміру товару, назва постачальника, країна, місто та адреса постачальника, телефон постачальника та адреса *Web*-сторінки;
  - номер поставки, дата поставки товару, кількість товару у поставці, знижка при поставці, вартість доставки.
2. Створити форми даних:
  - форму *Товари* для обліку інформації про товари;
  - форму *Постачальник* для обліку інформації про постачальників;
  - форму *Поставки* для обліку інформації про поставки та підпорядковану форму *Поставлено* для подання інформації про товари, що надійшли згідно із визначеною поставою (за допомогою кнопки *Інформація про постачальників* реалізувати відкриття форми *Постачальник* для всіх постачальників обраної поставки);
  - форму *Категорії* для наведення інформації про категорії товарів.
3. У формі *Поставлено* розмістити обчислювальне поле, в якому обчислювати ціну продажу товару з урахуванням знижки. У формі *Поставки* розмістити обчислювальне поле для наведення суми поставки з урахуванням знижки та вартості доставки товару. Створити головну кнопкову форму, з якої здійснювати виклик форм *Товари*, *Постачальник*, *Поставки*, *Категорії*. Заповнити базу даних інформацією про 7–10 поставок, кожна з яких включає поставку 7–10 товарів (7–10 різних категорій). Поставки здійснюються 7–10 постачальниками.
4. Створити звіт *Постачальники* (параметром звіту є дата), в якому навести перелік постачальників та перелік товарів, що поставлені кожним із постачальників (назва постачальника, адреса, телефон, номер замовлення, дата, назва товару, вартість, впорядкувати за назвою за зростанням) з нумерацією товарів для кожної категорії. Визначити сумарну вартість поставлених товарів кожним постачальників та частку (у процентах) сумар-

- ної вартості поставок кожним постачальником від загальної вартості поставлених товарів.
5. Створити звіт *Розрахунки* (параметром звіту є дата), в якому навести перелік поставок товарів за категоріями (назва, ціна, номер замовлення, дата, кількість, вартість, вартість з урахуванням знижки, вартість з урахуванням доставки, впорядкувати за вартістю за зростанням) з нумерацією товарів для кожної із категорій. Визначити загальну сплачену суму за кожною категорією та за весь товар в цілому.
  6. Створити діалогову форму *Звіти* для друку звітів, в якій користувач обирає тип звіту *Постачальники* або *Розрахунки* та визначає дату. Якщо обрано тип звіту *Розрахунки*, то користувач має визначити категорію товару, для якого формується звіт. Форма також містить кнопки *Перегляд* для перегляду звіту, *Друк* для друку звіту та *Отмена* для виходу у головну кнопочкову форму. Реалізувати виклик форми *Звіти* із головної кнопочкової форми.
- III. Створити базу даних обліку інформації про клієнтів фірми.
1. У базі даних має бути наведення наступна інформація про клієнта:
    - назва клієнта, прізвище, ім'я, по батькові директора, контактний телефон, номер рахунка клієнта;
    - країна, місто та юридична адреса клієнта, ідентифікаційний код клієнта;
    - вид робіт, виконання яких замовив клієнт, опис замовлення, сума замовлення, знижка, дата замовлення, дата виконання замовлення, номер акта прийняття робіт;
    - сума оплати за замовлення (оплата може здійснюватись за кількома рахунками), дата оплати, метод оплати, номер платіжного рахунка.
  2. Самостійно спроектувати та створити засоби введення інформації до бази даних (форми). Створити звіт *Клієнти*, в якому навести перелік клієнтів за кожним із видів замовлених робіт (назва клієнта, ідентифікаційний код, номер рахунка, робочий телефон, сума замовлення, сума, яку вже сплачено, сума боргу, впорядкувати за назвою за зростанням) з нумерацією клієнтів для кожного із видів замовлених робіт. Визначити середній роз-

мір замовлення за кожним із видів робіт та загальний розмір боргу.

3. Створити звіт *Оплата* про кошти, що були сплачені за звітний період (параметром звіту є період, протягом якого здійснювалась оплата — початкова та кінцева дата). Вказати назву організації платника, дату замовлення, суму замовлення, суму оплати, дату оплати, метод оплати, номер платіжного рахунка, групування за датою оплати за тижнями з нумерацією оплат для кожного із замовлень. Визначити загальну суму оплати за кожний тиждень та за вказаний період, а також процентне співвідношення сплачених сум протягом кожного тижня.

## **СПИСОК ЛІТЕРАТУРИ**

### *Основна*

1. *Боровиков В. В.* Access 2002. — М.: Солон-Р. — 560с.
2. *Боуман Д. и др.* Практическое руководство по SQL / Д. Боуман, С. Эмерсон, М. Дарновски. — К.: Диалектика, 1997. — 336 с.
3. Система управління базами даних Microsoft Access: лабораторний практикум. Метод. вказівки до виконання лаб. робіт / О. В. Вітюк, А. В. Кузьмін, Н. М. Москалькова, В. В. Попов, М. Є. Сіницький, Ю. А. Тарнавський. — К.: МАУП, 2003. — Ч. 1. — 166 с.
4. Система управління базами даних Microsoft Access: лабораторний практикум. Метод. вказівки до виконання лаб. робіт / О. В. Вітюк, А. В. Кузьмін, Н. М. Москалькова, В. В. Попов, М. Є. Сіницький, Ю. А. Тарнавський. — К.: МАУП, 2004. — Ч. 2. — 168 с.
5. *Гаврилова Т. А.* Базы знаний интеллектуальных систем. — 2000. — 384 с.
6. *Грабер М.* Введение в SQL. — М.: Лори, 1996. — 379 с.
7. *Коннолли Т. и др.* Базы данных. Проектирование, реализация и сопровождение. Теория и практика. — М.: Вильямс, 2000. — 1120 с.
8. *Попов В. В. та ін.* Практикум і контрольні роботи з MS Access: Метод. вказівки до виконання контрол. і самост. робіт / В. В. Попов, Л. О. Левченко, Н. М. Москалькова. — К.: МАУП, 2006. — 136 с.
9. *Послед Б. С.* Access 2002. Приложения баз данных: Лекции и упражнения. — М.: DiaSoft UP. — 656с.

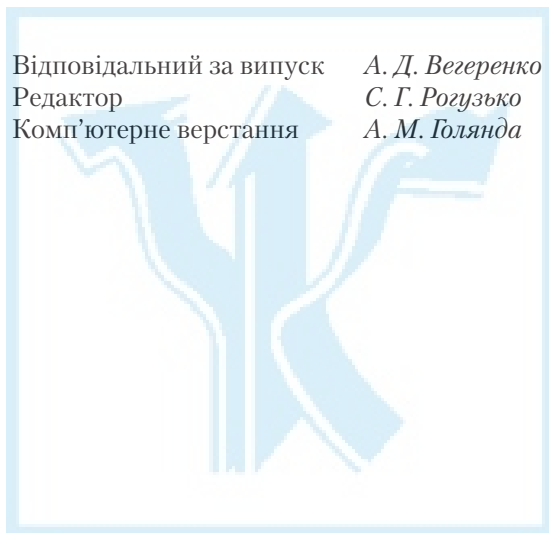
10. *Хомоненко А. Д. и др.* Базы данных: Учебник для высш. учеб. заведений / Под ред. проф. А. Д. Хомоненко. – СПб.: КОРОНА принт, 2000. – 416 с.

*Додаткова*

11. *Бекаревич Ю., Пушкина Н.* Самоучитель Microsoft Access 2000. – М., 1999. – 480 с.
12. *Бойко В. В., Савинков В. М.* Проектирование баз данных информационных систем. – М.: Финансы и статистика, 1989. – 351 с.
13. *Дейт К.* Введение в системы баз данных. – 7-е изд. – М.: Изд. дом “Вильямс”, 2001. – 1072 с.
14. *Диго С. М.* Проектирование и использование баз данных. – М.: Финансы и статистика, 1995. – 208 с.
15. *Дрибас В. П.* Реляционные модели данных. – М.: Мир, 1992. – 192 с.
16. *Загоруйко Н. Г.* Прикладные методы анализа данных знаний. – Новосибирск: Изд-во Ин-та математики, 1999. – 270 с.
17. *Каратыгин А.* Access 2000. Руководство пользователя с примерами. – Изд. ЛБЗ – ЮМС. – 376 с.
18. *Картов Б.* Microsoft Office 2000. – СПб.: Питер, 2000. – 448 с.
19. *Коунстейк С.* Access 97 шаг за шагом. – М.: Бином, 1998. – 208 с.
20. *Люгер Джордж Ф.* Искусственный интеллект: стратегии и методы решения сложных проблем. – М.: Изд. дом “Вильямс”, 2003. – 865 с.
21. *Мартин Дж.* Организация баз данных в вычислительных системах. – М.: Мир, 1980. – 662 с.
22. *Мейер М.* Теория реляционных баз данных. – М.: Мир, 1987. – 608 с.
23. *Нагао М. и др.* Структуры и базы данных / М. Нагао, Т. Катаяма, С. Уэмура. – М.: Мир, 1986. – 197 с.
24. *Оситов Г. С.* Приобретение знаний интеллектуальными системами. – М.: Наука, 1997. – 310 с.
25. *Парз К. и др.* Секреты Access. – К.: Диалектика, 1998. – 210 с.
26. *Пасько В.* Access 97 (русифицированная версия). – М., 2000. – 368 с.
27. *Соколов Е. Н., Вейткявичус Г. Г.* Нейроинтеллект. От нейрона к нейрокомпьютеру. – М.: Наука, 1989.
28. *Стоицкий Ю.* Office 2000. – СПб.: Питер, 2000.
29. *Ульман Д.* Основы систем баз данных. – М.: Финансы и статистика, 1983. – 334 с.

## **ЗМІСТ**

Пояснювальна записка.....	3
Тематика самостійної роботи з дисципліни “Бази даних та інформаційні системи”.....	8
Список літератури.....	43



Відповідальний за випуск *А. Д. Вегеренко*  
Редактор *С. Г. Рогузько*  
Комп'ютерне верстання *А. М. Голянда*

Зам. № ВКЦ-3794

Формат 60x84/<sub>16</sub>. Папір офсетний.  
Друк ротатійний трафаретний. Наклад 30 пр.  
Міжрегіональна Академія управління персоналом (МАУП)  
03039 Київ-39, вул. Фрометівська, 2, МАУП  
ДП «Видавничий дім «Персонал»  
03039 Київ-39, просп. Червонозоряний, 119, літ. XX  
*Свідоцтво про внесення до Державного реєстру  
суб'єктів видавничої справи ДК № 3262 від 26.08.2008*