

МІЖРЕГІОНАЛЬНА
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МАУП

**МЕТОДИЧНІ МАТЕРІАЛИ
ЩОДО ЗАБЕЗПЕЧЕННЯ
САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ
з дисципліни
“ОРГАНІЗАЦІЯ БАЗ ДАНИХ
ТА БАЗ ЗНАНЬ”
(для бакалаврів)**

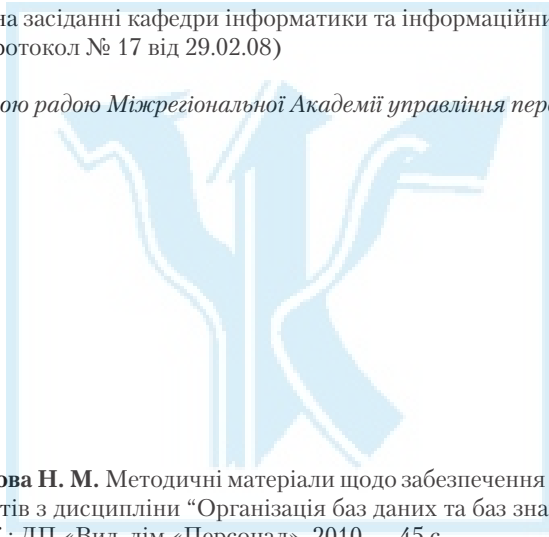
МАУП

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 “Організація баз даних та баз знань” (для бакалаврів). — К.: ДП «Вид. дім «Персонал», 2010. — 45 с.

Методична розробка містить пояснювальну записку, питання для самостійного вивчення студентами та самоконтролю, теми рефератів, практичні завдання та список літератури.

Призначена для методичного забезпечення самостійної роботи з дисципліни “Організація баз даних та баз знань” студентів денної форми навчання, які здобувають освіту за напрямком “Комп’ютерні науки” (спеціальність “Програмне забезпечення автоматизованих систем”).

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

Самостійна робота студентів є надзвичайно важливою складовою підготовки спеціалістів з напрямку “Комп’ютерні науки”. Теоретичний матеріал з програмування потребує багаторазового закріплення практичними прикладами. Студенти мають здобути навички самостійного виконання всіх етапів розробки програмного забезпечення (проектування, створення, тестування тощо). Це вимагає від студента систематичного виконання практичних завдань протягом семестру та підготовки до кожного практичного заняття.

Самостійна робота студентів є основним засобом опанування навчального матеріалу у позааудиторний час. Сучасна реформа вищої освіти — це насамперед перехід від парадигми навчання до парадигми освіти, самоосвіти. Тому при реформуванні вищої школи, уведенні кредитно-модульної технології навчання значно зростає роль самостійної роботи студентів. Студент, який хоче якомога краще оволодіти професією, має добре розуміти: на занятті викладач подає основи знань, навчає, як учити, виділяє ті ключові істини дисципліни, які пробуджують у молодій людині потяг до поглиблення й удосконалення усіх 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 студента повинна бути спланована, організаційно 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, виноситься 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. У деяких випадках ця робота проводиться відповідно до заздалегідь складеного графіка, що гарантує можливість індивідуального доступу студента до потрібних дидактичних засобів. Графік доводиться до відома студентів на початку поточного семестру. При організації самостійної роботи студентів з використанням складного обладнання

чи устаткування, складних систем доступу до інформації (наприклад, комп'ютерних баз даних, систем автоматизованого проектування тощо) передбачається можливість отримання необхідної консультації або допомоги з боку фахівця.

Самостійна навчальна діяльність студента з дисципліни “Організація баз даних та баз знань” передбачає:

- запам'ятовування певної інформації за рахунок уважного слухання і конспектування лекцій; активної роботи під час практичних занять;
- опрацювання конспектів лекцій, планів 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. Лекційний матеріал призначається для спрямування студентів у найраціональнішому напрямі щодо вивчення навчальної дисципліни і акцентуванні уваги на найскладніших, вузлових питаннях навчальної дисципліни. 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. Викладання інформатики в комп'ютерних класах або в аудиторіях, обладнаних мультимедійним обладнанням (наприклад, мультимедійним проектором або сенсорним екраном), водночас з демонстрацією студентам прийомів роботи з користувацьким інтерфейсом програми дозволяє підвищити рівень засвоєння лекційного матеріалу до 50 – 60%.

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 міститься у методичних матеріалах для практичних занять або у нав-

чально-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. Підготовку до практичного заняття краще за все здійснювати з використанням ПЕОМ зі встановленим на ньому відповідним програмним забезпеченням. За цієї можливості слід використовувати інтерактивні довідникові системи програм *MS Office* та інформаційно-пошукові системи *Internet*.

Вивчення навчального матеріалу за підручниками, навчальними посібниками, методичними вказівками, опрацювання матеріалу за першоджерелами, науковою і спеціальною літературою. Працювати із підручниками, навчальними посібниками, методичними рекомендаціями, 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 ознайомитись з їх змістом, щоб визначити, чи необхідно опрацювати 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, розділи тощо). У разі роботи з інтерактивними електронними джерелами слід використовувати можливість 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 посиланнями на інші джерела тощо) та фіксації опрацьованого матеріалу. Сучасні текстові редактори (у першу чергу *MS Word*) надають можливість створення електронного конспекту з примітками, виносками, коментарями та його роздрукування. Для самостійного поглибленого вивчення навчального матеріалу студенту слід звертатися до наукової та спеціальної літератури, яка може бути і не зазначена в навчально-методичному комплексі. Використання самостійно отриманих відомостей як у навчанні, так і на практиці є, безперечно, цінним здобутком діяльності студента на шляху формування свого професійного потенціалу.

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 опрацьовувати 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* (в основному – *Word Wide Web, FTP та UseNet newsgroups*).

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 дисципліни “Організація баз даних та баз знань” є формування у студентів знань щодо сучасних технологій проектування баз даних і розробки систем управління базами даних, засвоєння основних понять реляційних баз даних та здобуття практичних навичок створення програмних систем з використанням системи управління базами даних *Microsoft Access*.

Завданнями курсу є оволодіння теоретичними знаннями, що необхідні для розв’язання задач автоматизації обробки інформації у різ-

них предметних областях, а також набуття практичних навичок 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 “Організація баз даних та баз знань” 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 та баз знань;
- технології збереження, пошуку та обробки інформації;
- теоретичні основи побудови та функціонування баз даних та баз знань, 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 та баз знань;
- основні поняття реляційної моделі даних;
- основи мови побудови запитів SQL;
- засоби побудови баз даних за допомогою MS Access.

Після вивчення дисципліни студент має набути таких вмінь і навичок:

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

Для розуміння тематики курсу “Організація баз даних та баз знань” 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*. Даний курс базується на дисциплінах “Основи програмування та алгоритмічні мови”, “Мови об’єктно-орієнтованого

програмування”, “Основи дискретної математики”. Для розуміння тематики дисципліни “Організація баз даних та баз знань” 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 “Організація баз даних та баз знань” має такі складові і форми їх оцінювання:

- опрацювання програмного матеріалу зі змістового модуля (за матеріалами конспекту і підручників), оцінювання знань 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 модулю, їх оцінювання під час проміжного контролю;
- виконання курсового проекту, підготовка звіту з виконаної роботи, оцінювання результатів роботи під час захисту курсового проекту;
- 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 за рішенням кафедри опубліковані.

ТЕМАТИКА САМОСТІЙНОЇ РОБОТИ З ДИСЦИПЛІНИ “ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА БАЗ ЗНАНЬ”

Змістовий модуль I. Основи організації баз даних та баз знань

Тема 1. Основні концепції побудови баз даних та баз знань

Опрацювання програмного матеріалу (за матеріалами конспекту, підручників).

Представлення даних за допомогою ЕОМ. Поняття про дані та знання. Концепція інтегрованої обробки даних: скорочення надлишковості, цілісність даних, незалежність прикладних програм від даних. Поняття предметної області. Поняття бази даних. Моделі даних: інфологічна модель даних, даталогічна модель даних, фізична модель даних. Поняття систем управління базами даних. Архітектура систем управління базами даних. Функції систем управління базами даних.

Типи знань: декларативні та процедурні, екстенціональні та інтенціональні. Інтелектуальні задачі. Різні трактовки інтелектуальної діяльності. Проблема розуміння сенсу як виявлення знань з даних та сигналів. Неформальні моделі представлення знань. Формальні моделі представлення знань. Логічна модель представлення знань. Представлення знань у семантичних мережах. Фрейми. Продукційні моделі представлення знань.

Виявлення знань з досвіду (емпіричних фактів) та інтелектуальний аналіз даних (*data mining*). Типи закономірностей *data mining*: асоціація, послідовність, класифікація, кластеризація, прогнозування. Виявлення асоціативних правил. Задача з розпізнання образів як одна із задач аналізу даних.

Використання баз знань в експертних системах. Структура знань у базі знань. Машина логічного виведення. Модуль здобуття знань. Модуль пояснення. Діалогова система взаємодії з користувачем.

Література [3–5; 7; 8; 16; 20; 24; 27]

Основні визначення

Поняття інформації, даних, знань є спорідненими поняттями. У багатьох ситуаціях часто буває достатньо інтуїтивного розуміння і інтерпретації цих категорій. Складність формального визначення

термінів “інформація”, “дані”, “знання” полягає в загальноповнотності цих термінів і в їх уявній синонімічності.

Інформація — це відомості про об’єкти та явища навколишнього середовища, що сприймають інформаційні системи в процесі життєдіяльності або роботи.

Дані — це відомості, що одержані шляхом вимірювання, спостереження, логічних або арифметичних операцій, представлені в певній знаковій системі і на певному носії у формі, придатній для постійного зберігання, передавання і (автоматизованої) обробки за допомогою електронно-обчислюваних пристроїв.

У комп’ютерній галузі проблема запам’ятовування даних загально розв’язана — запам’ятати “напам’ять” можна будь-яку кількість інформації, але проблема отримання з цієї інформації знань залишається актуальною, більш того — тільки посилюється при зростанні обсягу даних.

Знання — перевірені суспільною практикою корисні відомості, які можуть багаторазово використовуватися людьми для розв’язання тих або інших завдань. Знання — інформація, що піддається смислової інтерпретації. Знання без смислу — це інформація або дані.

Знання у початковому розумінні — це результат пізнавальної діяльності людини. Але значна низка завдань здобуття знань піддається автоматизації, деякі з них вже автоматизовані. Спостерігається значний прикладний інтерес до технологій акумуляції знань та автоматизації інтелектуального аналізу інформації з метою виявлення знань. Свідченням цього є, зокрема, спроби слідом за поняттями “інформаційне суспільство”, “високотехнологічна цивілізація” тощо, увести в обіг терміни “економіка знань”, “менеджмент знань”. Знання — це уміння організувати процес та спрямувати його на досягнення поставленої мети, але одночасно знання — це люди плюс інформація, піднесені до ступеня засобів поширення цієї інформації.

Технологічно знання відрізняють від інформації такі властивості:

Структурованість. Знання повинні бути організовані з використанням зручної архітектури, яка б забезпечувала прозорість сховища знань та зручне представлення структури (змісту, рубрикаторів).

Зручність доступу і засвоєння. Для людини ця властивість означає здатність швидко зрозуміти і запам’ятати або, навпаки, пригадати; для автоматизованої обробки — засоби доступу, тобто пошук, короткі анотації до документів, індекси тощо.

Лаконічність. Лаконічність дає змогу швидко засвоювати та переробляти знання і підвищує “коефіцієнт корисного змісту”.

Несуперечність. “Хороші” дані/знання не повинні суперечити один одному, що очевидно або принаймні бажано. Проте для багатьох сфер збирання знань це спочатку не так — на вхід сховища знань може надходити суперечна інформація. Завдання інженера з знань — знайти суперечності і усунути їх на етапі збирання знань або надати різним елементам даних різну оцінку достовірності.

Оцінка достовірності. Безумовно, засвоюючи або використовуючи знання, хочеться знати, наскільки вони достовірні. Добре сховище знань (підручник або база даних) повинне мати таку оцінку для своїх елементів.

Процедури обробки. Одне з головних застосувань знань — можливість передавання знань іншим і здатність робити висновки на їх основі. Для цього повинні існувати процедури обробки знань. Здатність робити висновки означає для машини наявність процедур обробки і виведення, а також придатність структур даних для такої обробки, тобто наявність спеціальних моделей знань.

Теми рефератів

1. Порівняльний аналіз різних визначень поняття “інформація”.
2. Порівняльний аналіз різних визначень поняття “знання”.
3. Загальні характеристика концепції інтегрованої обробки даних.
4. Порівняльний аналіз моделей представлення знань.
5. Здобуття та формалізація знань у процесі розробки бази знань.

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. Дати визначення поняттям “декларативні знання”, “процедурні знання”.
5. Дати визначення поняттям “евристичні знання”, “експліцитні та імпліцитні знання”.

Тема 2. Базові поняття реляційних баз даних

Опрацювання програмного матеріалу (за матеріалами конспекту, підручників).

Реляційна модель даних. Поняття відношення. Типи даних. Поняття домену. Відношення. Атрибути. Заголовок та тіло відношення. Короткі відношення. Властивості відношень. Характеристики відношень: арність, потужність. Способи представлення відношень. Таблиці.

Поняття ключа. Умова унікальності. Умова мінімальності. Зв'язки між таблицями. Типи зв'язків. Зв'язки типу “один до багатьох”, “багато до багатьох”, “один до одного”. Схема реляційної бази даних.

Маніпулювання реляційними даними. Реляційна алгебра. Мови маніпулювання даними: SQL (Structured Query Language – структурована мова запитів) та QBE (Query-By-Example). Основні конструкції мови SQL.

Вступ до проектування баз даних. Надлишковість. Аномалії оновлення. Аномалії додавання. Аномалії видалення. Універсальне відношення проєктованої бази даних. Поняття нормалізації. Нормальні форми. Перша нормальна форма. Функціональна залежність. Повна функціональна залежність. Друга нормальна форма. Третя нормальна форма. Багатозначна залежність. Четверта нормальна форма.

Проектування бази даних. Етапи розробки бази даних. Побудова інфологічної моделі. Етапи проектування датованої моделі.

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

Основні визначення

Кожне прикладне завдання з обробки інформації пов'язане 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 завдання.

Наприклад, для автоматизації обліку співробітників деякого відділу необхідно здійснювати аналіз таких даних, як вік співробітника, його заробітна плата, стаж роботи, посада тощо. У цьому разі інформаційним об'єктом є співробітник, а дані про співробітника представляються як властивості 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 дає змогу 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 представлення 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 з обробки даних (вибір, вставка, оновлення, видалення тощо);

- забезпечує захист та цілісність даних в базі.

Користувачі бази даних працюють з її інформаційним та змістовним наповненням, їх не цікавлять подробиці фізичного збереження даних на магнітних носіях. Тому розрізняють три рівні абстракції для представлення 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 мережні, ієрархічні та реляційні моделі даних на логічному (концептуальному) рівні. Але більшість сучасних СУБД використовують реляційну модель даних. Назва реляційної моделі даних цієї моделі походить від англійського слова “relation” (відношення).

У реляційній моделі предметна область представлена у 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 окремий екземпляр об'єкта (рядок в таблиці).

Наприклад, студента можна ідентифікувати за прізвищем, але якщо зустрічаються студенти з однаковим прізвищем, то до ключа слід додати як поле Прізвище, так і поля Ім'я та По батькові. Але може статись, 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 випадку між векторами обох таблиць встановлюється взаємооднозначна 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.

Теми рефератів

1. Технологія проектування баз даних.
2. Основні концепції реляційної алгебри.
3. Нормалізація баз даних.
4. Етапи проектування баз даних.
5. Огляд конструкцій мови маніпулювання даними *SQL*.

Питання для самоконтролю та співбесіди

1. Що таке реляційна база даних?
2. Поясніть відмінність понять “домен” та “тип даних”.
3. Що таке відношення?
4. Що таке ключ?
5. Для чого використовуються зв'язки між таблицями?
6. Для чого виконують нормалізацію бази даних?
7. Дайте визначення першої нормальної форми.
8. Дайте визначення другої нормальної форми.
9. Дайте визначення третьої нормальної форми.
10. Дайте визначення четвертої нормальної форми.

Типові практичні завдання до змістового модуля I

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 з щомісячною виплатою”.
5. Спроекувати реляційну схему даних “Кредитування на придбання квартири з 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 та 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. Розробка баз даних у середовищі *MS Access*

Тема 3. Основи роботи у середовищі *MS Access*

Опрацювання програмного матеріалу (за матеріалами конспекту, підручників).

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*.

Налаштування середовища *MS Access*. Визначення параметрів запуску програми у *MS Access*. Адміністрування бази даних. Стиснення та відновлення файлів *MS Access*.

Загальні відомості про засоби захисту *MS Access*. Визначення пароля для бази даних. Створення *mde*-файлів. Використання майстра захисту.

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* належить до програмного забезпечення класу RAD (Rapid Application Development), що можна перекласти як “швидка розробка застосувань”. У файлі *MS Access* зберігаються дані разом з процедурами їх обробки у вигляді одного файла 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 у 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 декількох 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 і 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.

Теми рефератів

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”.

Питання для самоконтролю та співбесіди

1. Охарактеризуйте призначення таблиць у *MS Access*.
2. Охарактеризуйте призначення форм у *MS Access*.
3. Охарактеризуйте призначення запитів у *MS Access*.
4. Охарактеризуйте призначення звітів у *MS Access*.
5. Як визначити пароль для бази даних у *MS Access*?
6. Охарактеризуйте об'єкти *MS Access*.
7. Як здійснюється іменування об'єктів у *MS Access*?
8. Охарактеризуйте призначення модулів у *MS Access*.
9. Які об'єкти в *MS Access* використовуються для збереження даних?
10. Які об'єкти в *MS Access* використовуються для аналізу даних?

Тема 4. Робота з таблицями у середовищі MS Access

Опрацювання програмного матеріалу (за матеріалами конспекту, підручників).

Створення таблиць засобами MS Access. Створення заголовка відношення. Типи даних MS Access. Призначення властивостей полів таблиці.

Підстановка значень до поля таблиці. Використання властивостей вкладинки *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 [1; 3; 4; 8; 9; 17–19; 25; 26; 28]

Основні визначення

У MS Access таблиці використовуються для збереження даних. Способи створення реляційних таблиць у СУБД MS Access наведено у вигляді таблиці

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

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) *Денежный* — 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 представлення значень поля. Властивість *Маска ввода* дозволяє відображати дані у загальноприйнятій формі (наприклад, номер телефону відображати 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 встановлено 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 даних”, для відкриття якого необхідно виконати команду *Сервіс* – *Схема даних*. У цьому вікні зв’язки між таблицями відображаються у вигляді ліній, які проведено між полями таблиці.

Практичні завдання


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 у місяцях, 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 переведення, 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, КількістьМісць, ДобоваВартістьУтриманняПацієнта).


Таблиця “Лікар” ( Код, Прізвище, Ім’я, По батькові, Посада, Відділення, КількістьПлановихХворих).


Таблиця “Пацієнт” ( Код, Прізвище, Ім’я, По батькові, НомерКартки, Діагноз, Відділення, ДатаГоспіталізації, ДатаВиписки, Лікар).


Таблиця “Діагноз” ( КодДіагнозу, Діагноз).

4. Засобами *MS Access* створити схему даних для предметної області “Облік аналізів крові пацієнтів” та створити таблиці “КлінічнийАналізКрові”, “БіохімічнийАналізКрові”, “ПаспортПацієнта”, “Діагноз” з відповідними полями:

Таблиця “Діагноз” ( Код, Найменування).

Таблиця “КлінічнийАналізКрові” ( Код, КодПацієнта, Дата, Еритроцити, Нб, Лейкоцити, ЛейкПалЯдрові, ЛейкСегмЯдрові, Лімфоцитів, Моноцитів, СОЕ, Висновок).

Таблиця “БіохімічнийАналізКрові” ( Код, КодПацієнта, Дата, ЗагальнийБілок, Альбуміни, ГлобулінАльфа1, ГлобулінАльфа2, ГлобулінБета, ГлобулінГамма, ТімоловаПроба, СулемоваПроба, АлАТ, БілірубінЗагальний, ПротромбіновийІндекс, Фібріноген, Висновок).

Таблиця “ПаспортПацієнта” ( НомерКартки, Прізвище, Ім’я, По батькові, ДатаНародження, Стать, Адреса, МісцеРоботи, ДатаГоспіталізації, ДатаВиписки, ДіагнозПриГоспіталізації, КлінічнийДіагноз, СупутнійДіагноз).

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, Перснадбавка, 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 “ЛікарськіЗасоби”, “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. Як здійснюється зв’язування таблиць?
2. Для чого використовується майстер підстановки?
3. Що визначає властивість поля “*Присоединенный столбец*”?
4. Що визначає властивість поля “*Число столбцов*”?
5. Що визначає властивість поля “*Ширина столбцов*”?
6. Як визначити для підстановки поле зі списком з виведенням на екран двох стовпців?
7. Як приховати виведення стовпця ключа на екран при використанні підстановки?
8. Як визначити тип зв’язку між таблицями у *MS Access*?
9. Як визначити зв’язок між таблицями у *MS Access*?
10. Що таке цілісність даних в таблицях *MS Access*?
11. Що таке каскадне відновлення та каскадне видалення даних у таблицях *MS Access*?
12. Що таке каскадне відновлення та віддалення даних у таблицях *MS Access*?
13. Опишіть процедуру створення таблиці у *MS Access*.
14. Назвіть типи даних, які можна використовувати при визначенні таблиць у *MS Access*.
15. Як визначити маску вводу даних для поля?
16. Як визначити значення за замовчуванням для поля в таблиці?
17. Як здійснити перевірку правильності значень, що користувач уводить у таблицю?
18. Як визначити повідомлення при уведенні неправильного значення, що користувач уводить у таблицю?
19. Що таке індекс? Як визначити індексоване поле?
20. Назвіть способи об’єднання таблиць у запитах.

21. Дайте визначення внутрішнього об'єднання таблиць.
22. Дайте визначення лівого об'єднання таблиць.
23. Дайте визначення правого об'єднання таблиць.
24. Наведіть приклади таблиць, для яких результат виконання лівого об'єднання не співпадатиме з результатом виконання внутрішнього об'єднання.
25. Наведіть приклади таблиць, для яких результат виконання правого об'єднання не співпадатиме з результатом виконання внутрішнього об'єднання.

Тема 5. Робота з формами у середовищі MS Access

Опрацювання програмного матеріалу (за матеріалами конспекту, підручників).

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*.

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 макету 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. Представлення пов'язаних таблиць за допомогою підпорядкованих форм. Майстер побудови підпорядкованих форм. Створення підпорядкованої форми на основі декількох таблиць або запитів. Додавання підпорядкованої форми.

Майстри побудови форм: *Автоформа в столбец*, *Автоформа ленточная*, *Автоформа табличная*. Використання обчислювальних елементів управління у формах. Розрахунок підсумкових значень за групою записів підпорядкованої форми.

Створення кнопочкових форм. Використання засобу *Диспетчер кнопочных форм*.

Побудова діалогових форм. Уведення параметрів запитів за допомогою діалогових форм.

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 представлення даних таблиць та запитів, які пов'язані відношенням “один до багатьох”, причому головна форма представляє бік “один”, а підпорядкована форма — бік “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.




Структурно форма 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 даних* — використовується для відображення даних.

Зв'язок між формою та джерелом даних реалізується за допомогою *елементів управління*, які розміщують в області даних. Для додавання елементів до форми використовується *Панель елементів*, яка містить такі кнопки:

Кнопка панелі елементів	Призначення
Выбор объектов 	Використовується для виділення елемента керування або групи елементів, розділу, форми або звіту.
Мастера 	Включає та відключає майстри зі створення елементів керування.
Надпись 	Використовується для виведення пояснюючого тексту, наприклад заголовок, підпис або інструкція.

Поле 	Призначене для перегляду даних у джерелі даних (зв'язані поля), для уведення та зміни даних у таблицях, для виводу результатів обчислень (вільні та обчислювані поля).
Група переключачей 	Використовується для розміщення набору прапорців, перемикачів або вимикачів, що представляють набір альтернативних значень.
Выключатель 	Елементи керування, які приєднуються до логічних полів бази даних, або використовуються у діалогових вікнах для представлення вибору користувача.
Переключатель 	
Флажок 	
Поле со списком 	Складений елемент керування, який об'єднує поле та список.
Список 	Призначений для перегляду даних у джерелі даних (зв'язані поля), для уводу значень з фіксованого списку значень (список може бути пов'язаний з деяким полем таблиці або запиту або вільний).
Кнопка 	Використовується для виконання дій.
Рисунок 	Використовується для відображення незмінного малюнка у формі або звіті.
Свободная рамка объекта 	Використовується для включення у форму об'єктів OLE. При цьому об'єкт OLE стає часткою форми, але не зберігається в БД (малюнок, звук, діаграма, слайд).
Присоединенная рамка объекта 	Використовується для відображення в формі або звіті об'єктів OLE. При переході від запису до запису у форму або звіті виводяться різні об'єкти.
Разрыв страницы 	Використовується для позначення початку нового екрану у формі, нової сторінки у звіті.
Набор вкладок 	Використовується для створення форми або діалогового вікна з кількома вкладками, на які можна додати інші елементи керування.

Подчиненная форма/отчет 	Використовується для виводу у формі або звіті даних з декількох таблиць.
Линия 	Використовуються для відокремлення даних, а також для відокремлення різних розділів форми або сторінки.
Прямоугольник 	
Дополнительные элементы 	Додавання додаткових елементів.

Усі властивості основних об'єктів бази даних та елементів керування групують за їх призначенням на групи, кожній з яких відповідає певна вкладинка у вікні властивостей: *Макет* — властивості, які встановлюють загальний вигляд елемента або об'єкта, *Данные* — властивості, які стосують 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* розташовані усі властивості

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

Практичні завдання

1. Засобами *MS Access* створити засоби уведення даних для бази даних “Кредитування на придбання автомобіля” та кнопку форму.
2. Засобами *MS Access* створити засоби уведення даних для бази даних “Відправлення грошових переказів у гривнях” та кнопку форму.
3. Засобами *MS Access* створити засоби уведення даних для бази даних “Облік використання фондів стаціонару медичної установи” та кнопку форму.
4. Засобами *MS Access* створити засоби уведення даних для бази даних “Облік аналізів крові пацієнтів” та кнопку форму.
5. Засобами *MS Access* створити засоби уведення даних для бази даних “Облік бюджету медичної установи” та кнопку форму.
6. Засобами *MS Access* створити засоби уведення даних для бази даних “Облік переліку та вартості процедур, що відпускаються

- у медичних закладах санаторно-курортного типу” та кнопкову форму.
7. Засобами *MS Access* створити засоби введення даних для бази даних “Облік замовлень на використання виробничих ліній для таблетування різних лікарських форм” та кнопкову форму.
 8. Засобами *MS Access* створити засоби введення даних для бази даних “Облік замовлень на отримання лікарських розчинів лікарнею” та кнопкову форму.
 9. Засобами *MS Access* створити засоби введення даних для бази даних “Облік роботи аптечної бази” та кнопкову форму.
 10. Засобами *MS Access* створити засоби введення даних для бази даних “Облік роботи дистрибуторської мережі іноземної фармацевтичної компанії (NSP), яка розробляє власні біологічні добавки” та кнопкову 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 форма?
14. Що таке підпорядкована форма?
15. Опишіть способи створення підпорядкованих форм.
16. Для чого призначені елементи управління?
17. Які елементи управління використовуються при створенні форм у *MS Access*?
18. Що таке кнопкові форми? Як створити кнопку форму у *MS Access*?
19. Як використовується “Диспетчер кнопкових форм”?
20. Що таке діалогові форми? Як створити діалогову форму у *MS Access*?

Тема 6. Використання запитів для аналізу та маніпулювання даними

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.

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 стилі SQL*. Створення запитів за допомогою майстрів побудови запитів.

Використання виразів при побудові запитів. Правила побудови виразів. Побудова виразів за допомогою *Построителя выражений*. 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 *Повторяющиеся записи*. Побудова запиту *Повторяющиеся записи* за допомогою майстра. Засоби побудови запиту *Повторяющиеся записи* у режимі *Конструктор*.

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

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

Література [1; 3; 4; 8; 9; 17–19; 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 запити 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 та 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 даних дозволяє особливим чином представити дані, що зберігаються у таблицях. Результат виконання запити виглядає як таблиця та називається динамічним набором записів. Динамічний набір записів не зберігається в базі, але формується кожного разі при запуску запити 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. Логічна умова відбору даних за кількома полями складається з кількох простих умов, які об'єднані за допомогою операторів *And* (і) або *Or* (або), 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 для кожної групи 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 запитів*.

Практичні завдання

Описати процес створення таких запитів у навчальній базі даних “*Борей*”.

Запит 1. Визначити, на яку суму (з урахуванням знижки) оформив замовлення товарів типу “*молочные продукты*” кожний із співробітників, що проробив на фірмі хоча б 10 років.

Запит 2. Визначити категорії товарів, що поставлялись у банках з США або 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 “*Рыбпродукты*”, які постачаються у пакетах.

Запит 7. 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 почтою товарів на найбільшу суму (з урахуванням знижки та вартості доставки).

Запит 9. Знайти постачальників та співробітників з однакових країн.

Запит 10. Яку кількість замовлень обслужив кожний із співробітників, який прийшов 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 оформили замовлення товарів на суму (з урахуванням знижки) більше середнього.

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

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

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

Запит 19. Знайти постачальників і клієнтів з однакових міст.

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

Запит 21. 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 *Таблиця 1* 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 у 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 текстових функцій у майстрі побудови виразів.
14. Як формуються прості умови відбору записів?
15. Як відібрати 15 записів, що мають найбільші значення за деяким полем?
16. Як формуються складені умови відбору записів?
17. Як знайти унікальні значення деякого поля таблиці?
18. Як здійснюється сортування записів у запиті?
19. Які агрегатні функції можна використовувати в запитах?
20. Для чого використовується конструкція *SELECT*?
21. Для чого використовується службове слово *AS* ?
22. Для чого використовується службове слово *FROM* ?
23. Для чого використовується конструкція *UNION* ?
24. Для чого використовується конструкція *ORDER BY* ?
25. Для чого використовується конструкція *GROUP BY* ?
26. Як мовою *SQL* визначається поле, за яким здійснюється об'єднання таблиць?
27. Як мовою *SQL* визначається умова відбору записів?

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

Тема 7. Проектування звітів у середовищі *MS Access*

Опрацювання програмного матеріалу (за матеріалами конспекту, підручників).

Призначення звітів. Розділи звітів. Елементи управління, що використовуються у звітах.

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

Створення звітів за допомогою майстрів побудови звітів. Редагування звіту у режимі *Конструктор*. Обчислення у звітах. Обчислю-

вальні 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

Звіт — це об'єкт, призначений для представлення 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 та 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;
- *Верхний колонтитул* — вміст розділу виводиться угорі кожної сторінки звіту;
- *Заголовок группы* — 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 данных* — у цьому розділі розміщують елементи управління, які забезпечують зв'язок звіту з джерелом даних;
- *Нижний 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 “Колонтитулы” та “Заголовок/примечание отчета”.

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

Практичні завдання

1. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Кредитування на придбання автомобіля”. Додати кнопки виклику звітів на кнопкову форму.
2. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Відправлення грошових переказів у гривнях”. Додати кнопки виклику звітів на кнопкову форму.
3. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Облік використання фондів стаціонару медичної установи”. Додати кнопки виклику звітів на кнопкову форму.
4. Засобами *MS Access* створити засоби введення даних для бази даних “Облік аналізів крові пацієнтів”. Додати кнопки виклику звітів на кнопкову форму.
5. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Облік бюджету медичної установи”. Додати кнопки виклику звітів на кнопкову форму.
6. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Облік переліку та вартості процедур, що відпускаються у медичних закладах санаторно-курортного типу”. Додати кнопки виклику звітів на кнопкову форму.
7. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Облік замовлень на використання виробничих ліній для таблетування різних лікарських форм”. Додати кнопки виклику звітів на кнопкову форму.
8. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Облік замовлень на отримання лікарських розчинів лікарнею”. Додати кнопки виклику звітів на кнопкову форму.
9. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Облік роботи аптечної бази”. Додати кнопки виклику звітів на кнопкову форму.
10. Засобами *MS Access* створити засоби підготовки звітів для бази даних “Облік роботи дистриб’юторської мережі іноземної фармацевтичної компанії (NSP), яка розробляє власні біологічні добавки”. Додати кнопки виклику звітів на кнопкову форму.

Питання для самоконтролю та співбесіди

1. Назвіть розділи звіту та опишіть їх призначення.
2. Чим відрізняються розділи звіту: колонтитул та заголовок групи?

3. Чим відрізняються розділи звіту: колонтитул та примітка групи?
4. Чим відрізняються розділи звіту: верхній колонтитул та заголовок звіту?
5. Чим відрізняються розділи звіту: нижній колонтитул, примітка та заголовок звіту?
6. Які властивості можна встановити для області даних звіту?
7. Як створити обчислювальне поле у звіті?
8. Як обчислити значення за групою записів у звіті?
9. Як вивести номер сторінки у звіті?
10. Як обчислити кількість записів у кожній групі звіту?
11. Як здійснити нумерацію записів у групі звіту?
12. Як додати групування у звіті?
13. Як на першій сторінці звіту вивести дату друку звіту?
14. Як визначити назву стовпця для групи у звіті?
15. Як обчислити середнє значення для деякого поля за всіма записами групи?
16. Як обчислити середнє значення для деякого поля по всіма записами звіту?
17. Як обчислити для кожного запису звіту процент, який становить значення поля до загальної суми за всіма записами звіту?
18. Назвіть етапи створення звітів за допомогою майстра?
19. Назвіть основні прийоми редагування звітів у режимі Конструктор.
20. Як здійснюються обчислення у звітах?
21. Як здійснюється групування записів у звітах?

Тема 8. Автоматизація дій користувача за допомогою макросів та процедур VBA (Visual Basic for Applications)

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.

Поняття макросу. Призначення макросів. Виконання макросів.

Створення макросів. Режим *Конструктор*. Макрокоманди. Призначення основних макрокоманд. Визначення параметрів макро-

команди. 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

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) фільтрації даних;
- 4) виконання будь-якої команди будь-якого меню *MS Access*;
- 5) переходу 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) імпорту (експорту) таблиць баз даних, електронних таблиць, текстових файлів;
- 9) запуску програмного забезпечення і передавання 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 у стовпці макрокоманд та всі наступні макрокоманди, для яких стовпець *Имя макроса* є порожнім.

Питання для самоконтролю та співбесіди

1. Що таке об'єкт, властивість, подія?
2. Як в *Access* можна визначити процедуру обробки події?
3. Наведіть приклад процедури обробки події для форми?
4. Наведіть приклад процедури обробки події для звіту?

5. Що таке макрос?
6. Для чого використовуються макроси?
7. Що таке макрокоманда?
8. Назвіть основні макрокоманди.
9. Яка макрокоманда використовується для встановлення значення властивості об'єктів?
10. Як вивести повідомлення за допомогою макрокоманди?
11. Як визначити умову виконання макрокоманди?
12. Як створити макрос для перевірки коректності введених значень?
13. Як у формі створити кнопки для виконання фільтрації записів за умовою?
14. Як змінити фокус у поточній формі за допомогою макрокоманди?
15. Як створити групу макросів?

Типові практичні завдання до змістового модуля II

I. Створити базу даних обліку інформації про співробітників фірми.

1. У базі даних має бути представлена наступна інформація про співробітника:
 - прізвище, ім'я, по батькові;
 - посада, посадовий оклад, персональна надбавка для працівника, дата найму співробітника, робочий телефон;
 - дані 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 представлення інформації про співробітників фірми (поля розмістити на трьох вкладках *Дані про співробітника*, *Особисті дані*, *Фото*);
 - форму *Філіали* для представлення інформації про філіали фірми;
 - у формі *Філіали* розмістити підпорядковану форму *підпорядкована форма Співробітник* для представлення службової інформації про співробітників філіалу (за допомогою кнопки *Додат-*

кова інформація реалізувати відкриття форми *Співробітник* для обраного співробітника);

- форму *Посади* для представлення інформації про посади фірми,
- форму *Міста* для представлення переліку міст.

3. У формі *Співробітник* та *підпорядкована форма Співробітник* розмістити обчислюване поле *заробітна плата*, що складається з посадового окладу, надбавки за вислугу років — 50 грн. за кожні 5 років служби та персональної надбавки. У формі *Філіали* розмістити обчислювальне поле для представлення сукупного фонду заробітної плати філіалу. Створити головну кнопочну 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 за кожним філіалом (Назва та ідентифікаційний код філіалу, прізвище, ім'я, по батькові співробітника, посада, вік та стаж роботи, впорядкувати за прізвищем та зростанням) з нумерацією співробітників для кожного філіалу. Визначити середній вік та стаж співробітників у кожному з філіалів та загалом у фірмі.

5. Створити звіт *Зарплата* (параметром звіту є дата видачі), в якому навести перелік співробітників у 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 прізвищами та збільшенням), з нумерацією співробітників у кожному із філіалів. Визначити загальний фонд заробітної плати у кожному філіалі та загалом у фірмі.

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 кнопочну форму. Реалізувати виклик форми *Звіти* із головної кнопочку форми.

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

1. У базі даних має бути представлена 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 представлення інформації про товари, що надійшли згідно із визначеною поставкою (за допомогою кнопки *Інформація про постачальників* реалізувати відкриття форми *Постачальник* для всіх постачальників обраної поставки);
- форму *Категорії* для представлення 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 представлення суми поставки з урахуванням знижки та вартості доставки товару. Створити головну кнопочку форми, 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 та збільшенням) з нумерацією товарів для кожної категорії. Визначити сумарну вартість поставлених товарів за кожним із постачальників та частку (у процентах) сумарної 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 та збільшенням) з нумерацією товарів для кожної із категорій. Визначити

загальну сплачену суму за кожною із категорій та за всіма товарами загалом.

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

III. Створити базу даних обліку інформації про клієнтів фірми.

1. У базі даних має бути представлена 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 видом замовлених робіт (назва клієнта, ідентифікаційний код, номер рахунка, робочий телефон, сума замовлення, сума, яку вже сплачено, сума боргу, впорядкувати за назвою та зростанням), з нумерацією клієнтів для кожного із видів замовлених робіт. Визначити середній розмір замовлення за кожним видом 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 та тижнями), з нумерацією оплат для кожного із замовлень. Визначити загальну суму оплати за кожним тижнем 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. — 560 с.
2. *Боуман Д. и др.* Практическое руководство по SQL / Д. Боуман, С. Эмерсон, М. Дарновски. — К.: Диалектика, 1997. — 336 с.
3. *Методичні* вказівки до виконання лабораторних робіт “Система управління базами даних Microsoft Access: лабораторний практикум” / О. В. Вітюк, А. В. Кузьмін, Н. М. Москалькова та ін. — К.: МАУП, 2003. — (Ч. 1). — 166 с.
4. *Методичні* вказівки до виконання лабораторних робіт “Система управління базами даних Microsoft Access: лабораторний практикум” / 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. — М., 2000. — 384 с.
6. *Грабер М.* Введение в SQL. — М.: Лори, 1996. — 379 с.
7. *Коннолли Т.* 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: Методичні вказівки до виконання контрольних і самостійних 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. завед. / Под ред. проф. А. Д. Хомоненко. — СПб.: КОРОНА принт, 2000. — 416 с.

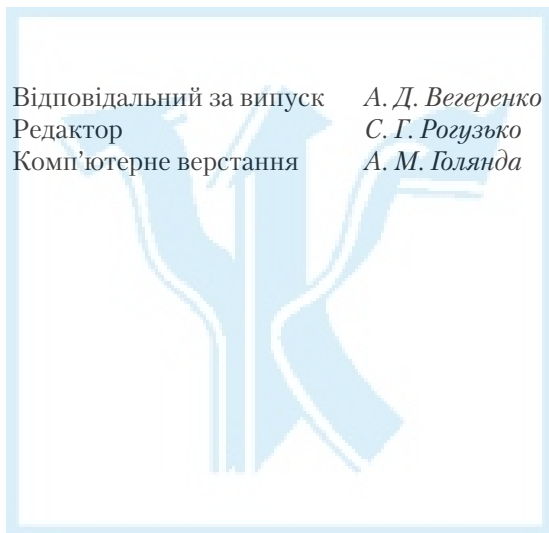
Додаткова

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-во ЛБЗ — ЮМС. — 376 с.
18. *Карпов Б.* Microsoft Office 2000. — СПб.: Питер, 2000. — 448 с.
19. *Коупстейк 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. *Осипов Г. С.* Приобретение знаний интеллектуальными системами. — М.: Наука, 1997. — 310 с.
25. *Парг К. и др.* Секреты Access. — К.: Диалектика, 1998. — 210 с.
26. *Пасько В.* Access 97 (русифицированная версия). — 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.
30. *Хомоненко А. Д. и др.* Базы данных: Учебник для высш. учеб. завед. / Под ред. проф. А. Д. Хомоненко. — СПб.: КОРОНА принт, 2000. — 416 с.

ЗМІСТ

Пояснювальна записка.....	3
Тематика самостійної роботи з дисципліни “Організація баз даних та баз знань”.....	9
Список літератури.....	43



Зам. № ВКЦ-3799

Формат 60×84/₁₆. Папір офсетний.

Друк ротаційний 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. ХХ

*Свідоцтво про внесення до Державного реєстру
суб'єктів видавничої справи ДК № 3262 від 26.08.2008*