

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Технології розроблення програмного забезпечення – 2

Інформаційні системи і комплекси

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю
151 «Автоматизація та комп'ютерно-інтегровані технології»,
спеціалізацією «Комп'ютерно-інтегровані технології сталих хімічних виробництв»*

Київ
КПІ ім. Ігоря Сікорського
2020

Технології розроблення програмного забезпечення – 2. Інформаційні системи і комплекси [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», спеціалізації «Комп'ютерно-інтегровані технології сталих хімічних виробництв» / КПІ ім. Ігоря Сікорського ; уклад.: І. М. Джигирей, О. О. Квітка, Ю.А. Запорожець. – Електронні текстові данні (1 файл: 2,21 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 123 с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 3 від 05.11.2020р.) за поданням Вченої ради Інженерно-хімічного факультету (протокол № 8 від 26.10.2020 р.)

Електронне мережне навчальне видання

Технології розроблення програмного забезпечення – 2 Інформаційні системи і комплекси

Укладачі: *Джигирей Ірина Миколаївна, канд. техн. наук, доц.
Квітка Олександр Олександрович, канд. хім. наук, доц.
Запорожець Юлія Анатоліївна*

Відповідальний редактор: *Складаний Денис Миколайович, канд. техн. наук, доц.*

Рецензент: *Шахновський А.М., канд. техн. наук, доц.*

У навчальному посібнику наведено завдання, виконувані протягом лабораторних робіт та самостійної роботи, з основних принципів оброблення даних, інформаційних систем і баз даних, основних принципів створення і керування базами даних. Видання забезпечує вивчення студентами теоретичного матеріалу, підготовку до виконання лабораторних робіт та буде корисним для самостійної роботи.

© КПІ ім. Ігоря Сікорського, 2020

Зміст

Передмова	4
Лабораторна робота № 1	
<i>Вивчення прийомів роботи з СКБД MS Access.....</i>	<i>5</i>
Лабораторна робота № 2	
<i>Вивчення способів створення нової БД СКБД MS Access</i>	<i>9</i>
Лабораторна робота № 3	
<i>Робота з даними в таблиці у СКБД MS Access</i>	<i>21</i>
Лабораторна робота № 4	
<i>Створення запитів в СКБД MS Access</i>	<i>26</i>
Лабораторна робота № 5	
<i>Створення та налагодження форм в СКБД MS Access</i>	<i>47</i>
Лабораторна робота № 6	
<i>Створення та редагування звітів в СКБД MS Access.....</i>	<i>65</i>
Лабораторна робота № 7	
<i>Створення макросів в СКБД MS Access</i>	<i>75</i>
Лабораторна робота № 8	
<i>Створення сценарію JavaScript із взаємодією з веб-ресурсом.....</i>	<i>85</i>
Лабораторна робота № 9	
<i>Робота з даними за допомогою мови JavaScript з використанням форм....</i>	<i>94</i>
Лабораторна робота № 10	
<i>Створення програмного забезпечення для роботи з СКБД MS SQL Server у середовищі MS Visual Studio</i>	<i>101</i>
Лабораторна робота № 11	
<i>Оброблення результатів моделювання ХТП (ХТС) збережених у БД за допомогою створення додатку на мові С#</i>	<i>107</i>
Рекомендована література	113
Додаток А.	
<i>Заходи безпеки під час проведення практичних занять</i>	<i>114</i>
Додаток Б.	
<i>Вимоги до оформлення протоколу лабораторної роботи</i>	<i>116</i>
Додаток В.	
<i>Зразок титульного листа протоколу.....</i>	<i>117</i>
Додаток Г.	
<i>Мова сценаріїв JavaScript</i>	<i>118</i>
Додаток Д.	
<i>Індивідуальні завдання до лабораторної роботи № 8</i>	<i>120</i>

Передмова

Цей навчальний посібник є частиною інформаційно-методичного забезпечення навчальної дисципліни «Технології розроблення програмного забезпечення», зокрема кредитного модуля «Інформаційні системи і комплекси», і розроблений відповідно до програми підготовки бакалаврів спеціальності «Автоматизація та комп'ютерно-інтегровані технології». Зазначений кредитний модуль є основою формування професійних знань і умінь студентів щодо загальних питань обробляння даних, інформаційних систем і баз даних, основних принципів створення і керування базами даних. Кредитний модуль цієї дисципліни «Інформаційні системи і комплекси» відноситься до циклу загальної підготовки і є базовим у підготовці бакалаврів вказаної спеціальності. Робота студента над учбовим матеріалом з дисципліни містить такі види роботи: вивчення матеріалу за навчальними посібниками і підручниками, відвідування лекцій, виконання завдань у рамках лабораторних робіт і практичних занять, індивідуальних консультацій; здавання письмового екзамену.

Представлені матеріали мають на меті закріплення знань та набуття вміння застосовувати навички, отримані в процесі вивчення кредитного модуля «Інформаційні системи і комплекси» дисципліни «Технології розроблення програмного забезпечення». Компетентності, отримані студентами в процесі вивчення цього кредитного модуля застосовуються ними для опанування третього модуля цієї дисципліни, а також дисциплін «Комп'ютерне моделювання систем і процесів», навчальної дисципліни з моделювання процесів тепло- масообміну, дисциплін рівня підготовки «магістр», а також виконання дипломного проекту бакалавра.

Це електронне видання призначено для надання допомоги студентам денної форми навчання спеціальності «Автоматизація та комп'ютерно-інтегровані технології» у вивченні кредитного модуля «Інформаційні системи і комплекси» дисципліни «Технології розроблення програмного забезпечення». З цією метою у цьому виданні подано перелік тем, які студент повинен вивчити та, за матеріалами яких проводяться лабораторні роботи, надано методичні вказівки до виконання завдань, наведено основні теоретичні положення. Навчальний посібник містить вимоги до проведення лабораторних робіт, вимоги до оформлення протоколів і контрольні питання для самопідготовки студентів. Лабораторна робота виконується на персональному комп'ютері стандарту IBM PC під керуванням операційної системи Windows версії 7 і вище. У навчальному посібнику також наведено заходи безпеки, яких треба дотримуватись під час проведення лабораторних робіт (додаток А).

Лабораторна робота № 1

Вивчення прийомів роботи з СКБД MS Access

Метою роботи є ознайомлення з інтерфейсом, об'єктами, майстрами і засобами конструювання MS Access 2016.

Завдання. Розглянути наявні засоби роботи з реляційними базами даних у СКБД MS Access 2016, особливості сторінок, стрічки, колекцій, меню, діалогових вікон та панелей інструментів інтерфейсу MS Access 2016.

Короткі теоретичні відомості

MS Access 2016 дає змогу створювати бази даних будь-якої складності та надає засоби для роботи користувача з будь-яким рівнем підготовки. Для розв'язування стандартних задач у MS Access передбачено численні шаблони баз даних, за допомогою майстрів та графічних інструментів користувачі-початківці можуть успішно розробляти реляційні структури даних, корисні додатки баз даних, запити, форми й звіти. Користувачам-професіоналам ця СКБД дає змогу за короткий проміжок часу створювати складні комерційні продукти.

СКБД MS Access – система керування реляційними базами даних, яка охоплює інструменти створення локальних баз даних, спільних баз даних у локальній мережі або мережі Інтернет і розроблення користувацьких додатків для роботи з базами даних. Локальна база даних MS Access містить як таблиці з даними так і форми, звіти та програмний код (файл з розширенням *.accdb).

Інструменти СКБД MS Access:

- майстри створення об'єктів і перетворення баз даних;
- засоби програмування (мова структурованих запитів SQL, мова макрокоманд і мова ООП VBA);
- графічні засоби створення об'єктів баз даних і додатків, зокрема таблиць і схем баз даних, запитів, форм та звітів.

Систему доступу до даних побудовано на основі ядра бази даних Microsoft Jet Database Engine, яке формує таблиці зберігання даних, об'єкти баз даних, виконує завантаження, збереження і вилучення даних, тощо.

У СКБД MS Access процес створення бази даних передбачає створення *схеми даних* (англ. relationships), яка відображає логічну структуру бази даних (таблиць і зв'язків між ними) з встановленням цілісності пов'язаних даних. Наприклад, підтримування цілісності даних не допускає наявності запису в дочірній (підпорядкованій) таблиці, для якої немає пов'язаного запису в батьківській (головній) таблиці. Схема даних потрібна для логічного об'єднання таблиць під час розроблення форм, запитів й звітів.

База даних MS Access зберігає в одному accdb-файлі такі *об'єкти*

- об'єкти бази даних: таблиці, запити і схеми даних;
- об'єкти застосунку: форми, звіти, макроси та модулі.

Таблиця зберігає дані щодо одного об'єкту моделі даних предметної області та містить поля (стовпці) і записи (рядки). База даних MS Access може містити до 32768 об'єктів. *Запити* (англ. queries) створюють для вибірки даних з однієї або кількох пов'язаних таблиць. Результатом виконання запиту є віртуальна таблиця запиту, яку надалі можна використовувати з іншими таблицями бази даних. *Форми* необхідні для створення діалогового інтерфейсу користувацьких застосунків. *Звіти* призначено для формування результатів вирішення задач на основі даних у численних форматах.

Базу даних MS Access формату accdb можна перетворити у виконувану і незмінювану базу даних формату accde за допомогою компіляції, що значно скорочує розмір файлу. У базі даних формату accde код VBA може тільки виконуватись, переглянути і змінити його неможливо.

Стартове вікно MS Access 2016 ліворуч містить перелік останніх використаних баз даних та команду «Відкрити інші файли», праворуч, в області створення баз даних, команди створення порожньої локальної бази даних (рис. 1), веб-застосунку¹ Access та, у разі підключення до Інтернету, шаблони для створення типових баз даних.

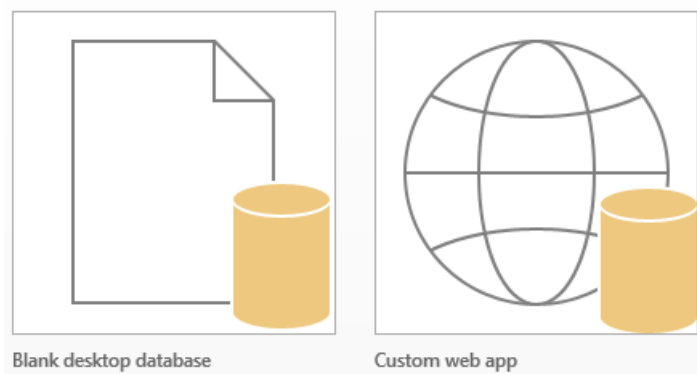


Рисунок 1 – Шаблони баз даних

Численні майстри MS Access дають змогу автоматизувати процес створення таблиць баз даних, запитів, форм, звітів, спрощують аналіз таблиць баз даних тощо:

- майстер аналізу таблиць (англ. table analyzer wizard)
- майстри створення форм і звітів;
- майстер підстановок (англ. lookup wizard);
- майстер запитів (англ. query wizard, рис. 2);
- майстер перехресних запитів (англ. crosstab query wizard);
- майстер діаграм (англ. chart wizard);
- майстер кнопок (англ. command button wizard).

Більшість майстрів можна запустити за допомогою відповідних команд вкладки «Створити» (англ. create).

¹Access app – база даних, спроектована і редагована у MS Access й використовувана у браузері. Дані та об'єкти зберігаються на сервері SQL або в базі даних Microsoft Azure SQL.

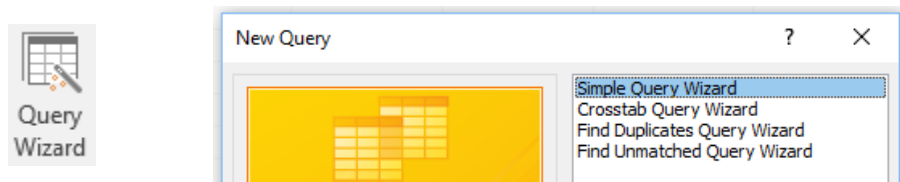


Рисунок 2 – Початкове діалогове вікно майстру запитів

Засоби конструювання об'єктів або конструктори MS Access дають змогу за допомогою наборів елементів створювати об'єкти бази даних і застосунків, зокрема, таблиці, схеми, запити тощо. У режимі макета користувач має змогу змінювати макет форми або звіту та одночасно переглядати дані у вигляді, в якому б ті відобразились у режимі форми або друку.

Під час роботи з базою даних у разі натискання команди «Файл» з'являється сторінка з командами, які відносяться безпосередньо до бази даних та налаштувань середовища. Серед іншого розділ «Відомості» міститиме попередження системи безпеки, а також команди «Стиснути і відновити» й «Зашифрувати з використанням пароллю».

Основними елементами користувацького інтерфейсу MS Access 2016 є: стрічка; область навігації; вкладки документів; колекції; діалогові вікна; контекстне меню елементів об'єктів; панель швидкого доступу; міні-панелі інструментів об'єктів; рядок стану; панель повідомлень.

Стрічка містить стандартні вкладки з групами найуживаніших команд (рис. 3), контекстні вкладки, які з'являються за необхідності, та панель швидкого доступу, на яку можна додавати команди. Наявність тих чи інших контекстних вкладок на смужці залежить від об'єкту, з яким працює користувач. Наприклад, за умови відкритої таблиці бази даних крім стандартних вкладок з'явиться група контекстних вкладок «Інструменти таблиць» з вкладками «Поля» і «Таблиця».

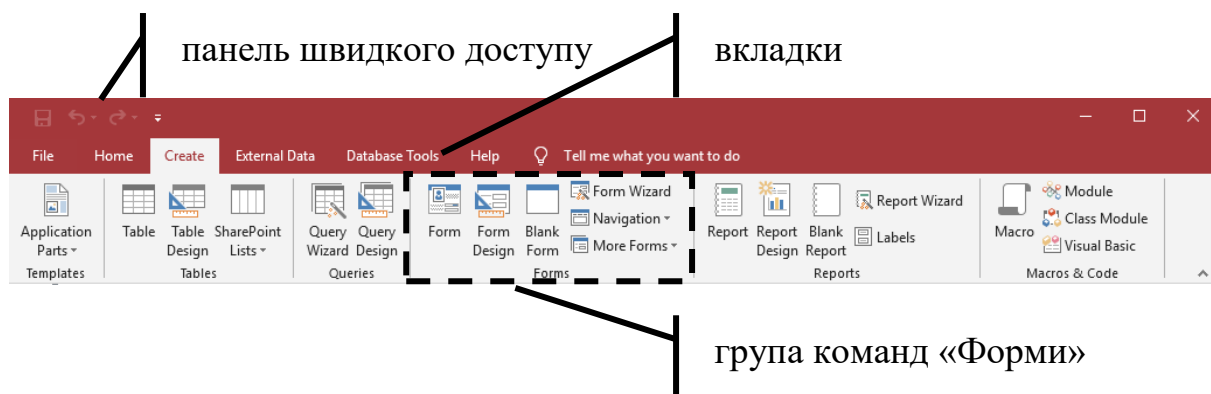


Рисунок 3 – Стрічка з активною вкладкою «Створити»

Такий елемент інтерфейсу як *колекція* або галерея відображає команди з одночасним відображенням результатів їх виконання. *Область навігації* знаходиться ліворуч у вікні MS Access й містить усі об'єкти бази даних.

Вкладки документів (таблиць, запитів, форм, звітів, макросів, тощо) відображаються у вікні документів MS Access.

Послідовність виконання роботи

1. Створити порожню локальну базу даних. Під час створення вказати як ім'я файлу власне прізвище. Внести відповідний пункт у протокол виконання роботи зі скріншотом діалогового вікна створення нової бази даних.

2. Новостворена база даних за замовчуванням міститиме одну таблицю даних з іменем «Table1». Внести² у протокол роботи перелік *стандартних* вкладок стрічки з переліком груп команд на кожній з них. Внести¹ у протокол роботи перелік *контекстних* вкладок для об'єкту-таблиці з переліком груп команд на кожній з них.

3. За допомогою команди «Показувати під стрічкою» панелі швидкого доступу перемістити цю панель нижче стрічки. Активувати на панелі швидкого доступу команди «Відкрити» та «Швидкий друк» або будь-які дві інші з переліку команд кнопки налаштування цієї панелі. Активувати на панелі швидкого доступу команду «Видалити» за допомогою виклику діалогового вікна «Параметри Access» командою «Більше команд...» з переліку команд кнопки налаштування цієї панелі. Внести відповідний пункт у протокол виконання роботи зі скріншотом панелі швидкого доступу.

4. Змінити висоту полів та ширину записів таблиці бази даних за допомогою (а) курсору і (б) відповідних команд групи «Записи» вкладки «Головна». Внести відповідний пункт у протокол виконання роботи зі скріншотом таблиці бази даних.

5. За допомогою команди «Режим конструктора» групи «Режими» вкладки «Головна» змінити відображення створеної таблиці. Внести відповідний пункт у протокол виконання роботи зі скріншотом таблиці бази даних.

6. Внести у протокол виконання роботи скріншоти принаймні чотирьох команд стрічки, які викликають майстрів MS Access. Внести у протокол скріншоти діалогових вікон, які з'являються у разі виконання цих команд.

7. Оформити протокол лабораторної роботи.

Оформлення звіту

Після оформлення звіту з лабораторної роботи до задалегідь підготовленого протоколу додаються роздруківки скріншотів з відповідними поясненнями.

Контрольні запитання

1. Назвіть три основні групи інструментів MS Access.
2. Що може містити accdb-файл?
3. Вкажіть особливості бази даних збереженої у форматі accde.

² допустимо мовою інтерфейсу додатку

4. Для чого потрібні майстри MS Access?
5. Назвіть основні елементи користувацького інтерфейсу MS Access.
6. Які п'ять вкладок з'являються на стрічці у разі створення нової бази даних?
7. У якому випадку на стрічці з'являються контекстні вкладки?
8. Яке призначення панелі швидкого доступу?
9. Чи можна змінювати набір команд на панелі швидкого доступу?
10. Чи змінюється набір вкладок стрічки у разі зміни режиму оброблення об'єкту бази даних?

Лабораторна робота № 2

Створення бази даних СКБД MS Access

Метою роботи є ознайомлення з принципами проектування баз даних, створення й редагування структури таблиць та даних у середовищі MS Access 2016.

Завдання. Розглянути наявні засоби створення реляційної бази даних у СКБД MS Access 2016.

Короткі теоретичні відомості

Реляційна база даних містить взаємопов'язані реляційні таблиці. Логічна структура бази даних визначає склад реляційних таблиць, їхню структуру і логічні зв'язки між таблицями. Для кожної таблиці визначають унікальний ключ, за допомогою якого СКБД забезпечує цілісність даних. У разі проектування бази даних, яка відповідає вимогам нормалізації, між таблицями визначено логічні зв'язки «один до багатьох», що дає змогу СКБД підтримувати зв'язну цілісність і несуперечність даних у базі.

Інформаційно-логічна модель відображає дані предметної області у вигляді сукупності інформаційних об'єктів і зв'язків між ними. Відповідність ключових і неключових властивостей предметної області встановлює функціональна залежність властивостей (реквізитів). Кожний інформаційний об'єкт у моделі даних повинен мати унікальне ім'я. Інформаційний об'єкт утворює множину реалізацій – екземплярів об'єкту, кожний з яких може бути однозначно визначено серед множини усіх екземплярів, тобто ідентифіковано за допомогою значення унікального (первинного) ключа інформаційного об'єкту.

Процес зменшення надлишковості даних називають нормалізацією. Вимоги нормалізації до реквізитів інформаційних об'єктів такі:

- інформаційний об'єкт повинен містити унікальний ідентифікатор (ключ);
- усі реквізити повинні бути незалежними один від одного;
- усі реквізити складеного ключа повинні бути незалежними один від одного;

- кожний реквізит повинен функціонально повно залежати від ключа;
- кожний реквізит повинен прямо залежати від ключа, а не через інший проміжний реквізит.

Після визначення інформаційних об'єктів необхідно визначити зв'язок між ними. Зв'язки інформаційних об'єктів можуть бути одно-однозначні («один-до-одного», 1:1), одно-багатозначні («один-до-багатьох», 1:M, «багато-до-одного», M:1) та багато-багатозначні («багато-до-багатьох», M:N). Багато-багатозначні зв'язки не може бути реалізовано безпосередньо у реляційній базі даних. Для використання таких зв'язків вводять об'єкт-зв'язку (рис. 4).

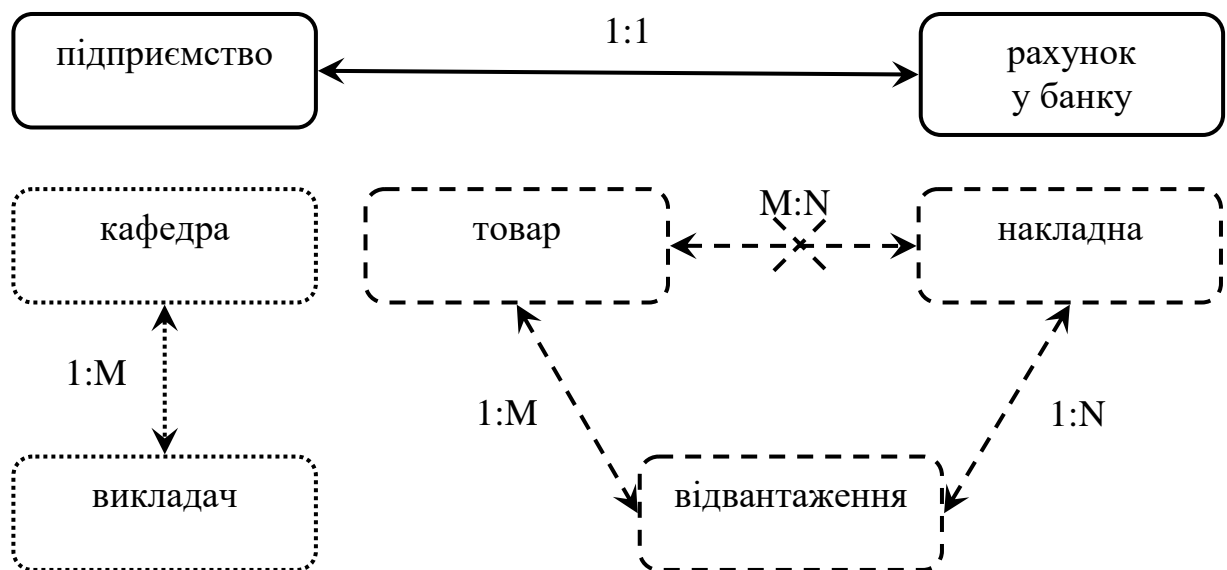


Рисунок 4 – Приклади зв'язків інформаційних об'єктів (графічне зображення)

Логічна структура реляційної бази даних є відображенням інформаційно-логічної моделі предметної області. Кожний інформаційний об'єкт моделі даних відображено відповідною реляційною таблицею. Структуру таблиці визначено складом реквізитів відповідного інформаційного об'єкту, у якій поле (стовпець) відображає один з реквізитів об'єкту. Ключові реквізити об'єкту утворюють унікальний ключ реляційної таблиці. Рядки (записи) таблиці відповідають екземплярам об'єкту. Зв'язки між об'єктами моделі даних реалізують однаковими реквізитами або ключами зв'язку у відповідних таблицях. Ключем зв'язку типу 1:M є унікальний ключ головної таблиці, ключем зв'язку в підпорядкованій таблиці може бути частина унікального ключа або поле, якого немає у складі первинного ключа. Ключ зв'язку у підпорядкованій таблиці має назву зовнішнього ключа.

Створення реляційної бази даних починається з формування структури таблиць з визначенням складу полів, їх імен, типів даних кожного поля, розміру поля, ключів, індексів таблиць та інших властивостей полів. Після визначення структури таблиць створюють схему даних, у якій встановлюють

зв'язки між таблицями. У разі створення бази даних важливо задати параметри, згідно з якими підтримуватиметься цілісність даних. Для цього потрібно вказати ключові поля таблиць, вказати обмеження на допустимі значення даних, а у разі створення схеми даних на основі нормалізованих таблиць необхідно задати параметри підтримування цілісності зв'язків бази даних. Створення баз даних завершується завантаженням тобто заповненням таблиць конкретними даними.

Послідовність виконання³

1. Створити порожню локальну базу даних. Під час створення вказати як ім'я файлу «*Прізвище*⁴ – Навчальний процес».

2. Створити таблицю «*ПІБ*⁵-Група» бази даних «*Прізвище – Навчальний процес*» у режимі конструктора. У групі «Режими» вкладки «Головна» потрібно обрати команду «Режим конструктора» (англ. «Design View»). З'явиться вікно-запит щодо збереження таблиці «Table 1», у якому необхідно, серед іншого, вказати назву таблиці як «*ПІБ-Група*». У режимі конструктора ведіть імена, тип даних, опис, а також властивості полів таблиці «*ПІБ-Група*» як показано у табл. 1 і на рис. 5. Для введення обмежувальних виразів варто скористатись будівником виразів, який можна викликати натисканням команди «...» у рядку введення відповідної властивості поля.

3. У режимі конструктора створити таблиці «*ПІБ-Студент*», «*ПІБ-Кафедра*», «*ПІБ-Викладач*», «*ПІБ-Дисципліна*», «*ПІБ-Семестр*», «*ПІБ-Успішність*» згідно умов представлених у табл. 1-7.

Таблиця 1 – Опис властивостей полів таблиці «*ПІБ-Група*»

Ім'я	Ключове	Обов'язкове	Тип даних	Число знаків після коми	Підпис поля	Умова і повідомлення про помилку
<i>pib</i> ⁶ code*	так	так	текстовий		шифр групи	–
<i>pib</i> number		ні	числовий цілий		кількість студентів	> 0 і <= 35 кількість студентів недопустима або більша прийнятної
<i>pib</i> rank		ні	числовий одинарний з плаваючою комою	2	середній бал	>= 60 і <= 100 або 0 помилка оцінювання

**тут і нижче*, унікальне поле (англ. indexed – yes, no duplicates)

³ тут і нижче – завдання подготовлено з використанням матеріалів книги Бекаревич Ю.Б., Пушкіна Н.В. *Microsoft Access 2000*. – СПб. : БВХ–Санкт-Петербург, 1999. – 480 с.

⁴ прізвище студента

⁵ тут і нижче – аббревіатура прізвища, ім'я та по батькові студента

⁶ тут і нижче – аббревіатура прізвища, ім'я та по батькові студента англійською мовою символами у нижньому регістрі

Група		
Field Name	Data Type	Description (Optional)
code	Short Text	шифр групи
number	Number	кількість студентів у групі
rank	Number	прохідний бал

General	Lookup
Field Size	5
Format	
Input Mask	
Caption	шифр групи
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	No
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

General	Lookup
Field Size	Integer
Format	General Number
Decimal Places	0
Input Mask	
Caption	кількість студентів
Default Value	0
Validation Rule	Between 1 And 35
Validation Text	кількість студентів недопустима
Required	No
Indexed	No
Text Align	General

General	Lookup
Field Size	Single
Format	General Number
Decimal Places	2
Input Mask	
Caption	середній бал
Default Value	
Validation Rule	Between 60 And 100 Or 0
Validation Text	помилка оцінювання
Required	No
Indexed	No
Text Align	General

Рисунок 5 – Властивості полів таблиці «ПІБ-Група»

Таблиця 2 – Опис властивостей полів таблиці «ПІБ-Студент»

Ім'я	Ключове	Обов'язкове	Тип даних	Число знаків після коми	Підпис поля	Умова і повідомлення про помилку
pibcode*	так	так	текстовий		шифр групи	—
pibstudent*	так	так	текстовий		номер студента	—
pibsurname		так	текстовий		ПІБ	—
pibbirthyear		ні	числовий цілий		рік народження	—
pibaddress		ні	текстовий		адреса	—
pibrank		ні	числовий одинарний з плаваючою комою	2	середній бал	≥ 60 і ≤ 100 або 0 помилка оцінювання

Таблиця 3 – Опис властивостей полів таблиці «ПІБ-Кафедра»

Ім'я	Ключове	Обов'язкове	Тип даних	Підпис поля
pibdept*	так	так	текстовий	код
pibdeptname		ні	текстовий	назва
pibphone		ні	текстовий	телефон
pibhead		ні	текстовий	ПІБ зав. каф.
pibphoto		ні	поле OLE-об'єкту	фото зав. каф.

Таблиця 4 – Опис властивостей полів таблиці «ПІБ-Викладач»

<i>Ім'я</i>	<i>Ключове</i>	<i>Обов'язкове</i>	<i>Тип даних</i>	<i>Підпис поля</i>
pibtab*	так	так	текстовий	табельний номер
pibsurname		так	текстовий	ПІБ
pibdegree		ні	текстовий	вчений ступінь
pibtitle		ні	текстовий	вчене звання
pibdept		так	текстовий	код кафедри

Таблиця 5 – Опис властивостей полів таблиці «ПІБ-Дисципліна»

<i>Ім'я</i>	<i>Ключове</i>	<i>Обов'язкове</i>	<i>Тип даних</i>	<i>Підпис поля</i>	<i>Умова і повідомлення про помилку</i>
pibdiscode*	так	так	текстовий	код	—
pibdistitle		ні	текстовий	назва	—
pibdisnumber		ні	числовий цілий	обсяг, годин	≥ 30 і ≤ 360 некоректний обсяг дисципліни
piblab		ні	числовий цілий	лабора-торні	—
pibpractice		ні	числовий цілий	практичні	—
pibsemester		ні	числовий цілий	кількість семестрів	
pibprogram		ні	приєднання (англ. attachment)	програма	

Таблиця 6 – Опис властивостей полів таблиці «ПІБ-Навчання»

<i>Ім'я</i>	<i>Ключове</i>	<i>Обов'язкове</i>	<i>Тип даних</i>	<i>Число знаків після коми</i>	<i>Підпис поля</i>
pibcode*	так	так	текстовий		шифр групи
pibdiscode*	так	так	текстовий		код дисципліни
pibtab*	так	так	текстовий		таб. номер викладача
pibtype*	так	так	текстовий		вид занять
pibhours		ні	числовий цілий		години
pibavrank		ні	числовий одинарний з плаваючою комою	2	середні бали за дисципліною

Таблиця 7 – Опис властивостей полів таблиці «ПІБ-Успішність»

Ім'я	Ключове	Обов'язкове	Тип даних	Число знаків після коми	Підпис поля
pibcode*	так	так	текстовий		шифр групи
pibstudent*	так	так	текстовий		номер студента
pibdiscode*	так	так	текстовий		код дисципліни
pibtab*	так	так	текстовий		таб. номер викладача
pibtype*	так	так	текстовий		вид занять
pibrating		ні	числовий цілий	0	оцінка

4. Створити складений первинний ключ для таблиці «ПІБ-Студент» на основі полів pibcode та pibstudent. В режимі конструктора за допомогою виділення курсором миші в області маркування таблиці полів pibcode та pibstudent (або натисканням на вказаних полях у разі натиснутої клавіші Ctrl) необхідно виділити ці два поля одночасно. Натисканням команди «Первинний ключ» у групі «Інструменти» вкладки «Конструктор» буде створено складений первинний ключ як показано на рис. 6. Подібну операцію виконати також для таблиць «ПІБ-Навчання» й «ПІБ-Успішність».

Field Name	Data Type
code	Short Text
student	Short Text
surname	Short Text
birthyear	Number
address	Short Text
rank	Number

Рисунок 6 – Результат створення складеного первинного ключа за полями pibcode та pibstudent таблиці «ПІБ-Студент»

5. Видалити одну з таблиць і створити вдруге у режимі таблиці.

6. Заповніть рядки створених таблиць власними записами за прикладами представленими на рис. 7-11.

шифр г	номер	ПІБ	рік	адреса	середній бал	Click to Add
XA-01	01	Антонюк Р.П.	1999		73	
XA-01	02	Бондаренко С.А.	1998		85	
XA-01	03	Бойко Є.І.	1999		65	
XA-01	04	Мороз Н.В.	1997		80	
XA-02	01	Бондар Н.П.	1997		62	
XA-02	02	Федорів Д.К.	1997		95	
XA-02	03	Савченко І.Р.	1997		84	
XA-03	01	Андрєєв Г.М.	1998		99	
XA-03	02	Поліщук О.К.	1999		66	
XA-04	01	Іванов К.К.	1997		78	

Рисунок 7 – Записи таблиці «ПІБ-Студент»

Група	Студент	Кафедра	Викладач	Дисципліна	Навчання	Успішність
шифр групи	кількість студент	середній бал	Click to Add			
ХА-01	15	80,13				
ХА-02	18	77,25				
ХА-03	21	85,32				
ХА-04	27	81,15				
ХА-05	17	78,45				
ЛА-01	30	79,01				
ЛА-02	34	83,65				
ЛА-03	29	77,22				
ЛА-04	11	88,9				
*	0					

Рисунок 8 – Записи таблиці «ПІБ-Група»

Група	Студент	Кафедра	Викладач	Дисципліна	Навчання	Успішність
код	назва	телефон	завідувач(к	фото	Click to Add	
ІН	Інформатики	+380442041111	Федорів А.П.	Picture		
МТ	Математики	+380442041112	Жданов К.П.			
ІС	Історії	+380442041113	Бранів Е.Н.			
ІМ	Іноземної мови	+380442041114	Лапішева С.З.			
ФК	Фізкультури	+380442041115	Шевченко Л.У			
ФЛ	Філософії	+380442041116	Берко У.Ч.			
*						

Рисунок 9 – Записи таблиці «ПІБ-Кафедра»

Група	Студент	Кафедра	Викладач	Дисципліна	Навчання	Успішність
табельний	ПІБ	вчений ступ	вчене зван	код кафедр	Click to Add	
101	Андрій А.А.	д.т.н.	професор	ІН		
102	Андрій А.А.	к.т.н.	доцент	ІН		
103	Глушак П.П.	к.т.н.	доцент	ІН		
104	Семченко П.П.	к.т.н.	доцент	ІН		
105	Черненко П.П.	к.т.н.	доцент	ІН		
201	Блишній П.П.	д.ф.-м.н.	професор	МТ		
202	Львівський П.П.		асистент	МТ		
203	Шаповалов П.П.	д.т.н.	професор	МТ		
204	Носіков П.П.		асистент	МТ		
301	Ільківський П.П.	к.ф.н.	доцент	ІС		
302	Пустовий П.П.	к.і.н.	доцент	ІС		
303	Романівський П.П.	к.і.н.	доцент	ІС		
304	Цвєтковський П.П.	к.і.н.	доцент	ІС		
401	Сорочинський П.П.	к.ф.н.	доцент	ІМ		
402	Богдановський П.П.	к.ф.н.	доцент	ІМ		
403	Лисенко П.П.	к.ф.н.	доцент	ІМ		
404	Шаповаловський П.П.	к.ф.н.	доцент	ІМ		
405	Кудрявцевський П.П.		асистент	ІМ		
501	Жирковський П.П.	к.п.н.	доцент	ФК		
502	Егор'євський П.П.		ст. викл.	ФК		
503	Ермоленко П.П.		асистент	ФК		
601	Лопатівський П.П.	к.ф.н.	доцент	ФЛ		
602	Яковлевський П.П.	к.ф.н.	доцент	ФЛ		
603	Райковський П.П.	к.ф.н.	доцент	ФЛ		
604	Соловйовський П.П.		асистент	ФЛ		
*						

Рисунок 10 – Записи таблиці «ПІБ-Викладач»

Група	Студент	Кафедра	Викладач	Дисципліна	Навчання	Успішність
код	назва	обсяг, год	лабораторн	практичні	кількість се	0
01	Інформатика	200	80	120	4	0(0)
02	Математика	200	100	100	4	0(0)
03	Історія	140	90	50	3	0(0)
04	Іноземна мова	200	0	200	4	0(0)
05	Філософія	100	40	600	2	0(0)
06	Фізкультура	100	0	100	2	0(0)
*					1	0(0)

Рисунок 11 – Записи таблиці «ПІБ-Дисципліна»

7. Змінити макет таблиці «ПІБ-Група» у режимі таблиці: порядок розташування стовпців, ширину стовпців і висоту рядків, шрифт даних, формат ліній сітки й тла тощо. Приховуйте один зі стовпців за допомогою команди «Приховати» групи «Записи» вкладки «Головна» або тієї ж команди контекстного меню стовпця. Вставте відповідний скріншот у протокол. За допомогою команди «Не приховувати» поверніть таблицю до початкового вигляду.

Група	Студент	Кафедра	Викладач	Дисципліна	Навчання	Успішність
шифр групи	код дисцип.	таб. номер	вид занять	години	середні бал	Click t
ХА-01	01	101	лаб.	40	0	
ХА-01	01	102	пр.	60	0	
ХА-01	02	201	лаб.	50	0	
ХА-01	02	202	пр.	50	0	
ХА-01	03	301	лаб.	48	0	
ХА-01	03	302	пр.	20	0	
ХА-01	04	401	пр.	50	0	
ХА-01	05	501	лаб.	50	0	
ХА-01	05	502	пр.	50	0	
ХА-01	06	601	лаб.	100	0	
ХА-02	01	101	лаб.	100	0	
ХА-02	01	103	пр.	180	0	
ХА-02	04	401	лаб.	100	0	
ХА-05	01	101	лаб.	100	0	
ЛА-01	01	102	пр.	180	0	
ЛА-01	02	201	пр.	70	0	
ЛА-02	04	403	пр.	100	0	
ЛА-03	01	101	лаб.	100	0	
ЛА-04	05	503	пр.	100	0	
*				0	0	

Рисунок 12 – Записи таблиці «ПІБ-Навчання»

Група	Студент	Кафедра	Викладач	Дисципліна	Навчання	Успішність
шифр групи	номер студ	код дисцип.	таб. номер	вид занять	оцінка	Click i
ХА-01	01	01	101	лаб.	98	
ХА-01	01	03	302	пр.	80	
ХА-01	02	01	101	лаб.	85	
ХА-01	02	03	302	пр.	60	
ХА-01	03	01	101	лаб.	75	
ХА-01	03	03	302	пр.	90	
ХА-01	04	01	101	лаб.	66	
ХА-01	04	03	302	пр.	82	
*					0	

Рисунок 12 – Записи таблиці «ПІБ-Успішність»

8. Впровадити об'єкти OLE до записів таблиці «**ПІБ-Кафедра**» у режимі таблиці, фото завідувачів кафедр⁷, скориставшись командами «Вставка» або «Додати об'єкт» контекстного меню відповідного поля таблиці. Відображення об'єктів можливе тільки у формі або звіті. Додати кілька вкладень до таблиці «**ПІБ-Дисципліна**» у поле **pibprogram**.

9. Створити схему даних бази даних «**Прізвище – Навчальний процес**» за допомогою команди «Зв'язки» групи «Зв'язки» вкладки «Інструменти бази даних». У діалоговому вікні «Таблиці» вказати усі створені таблиці.

Встановіть зв'язок між таблицями «**ПІБ-Група**» і «**ПІБ-Студент**» за простим ключем **pibcode**. Для цього у вікні схеми даних (англ. relationships) встановіть курсор миші на ключовому полі **pibcode** головної таблиці «**ПІБ-Група**» і перетягніть це поле на полі **pibcode** у підпорядкованій таблиці «**ПІБ-Студент**». У вікні редагування зв'язків (англ. edit relationships) у рядку типу відносин встановиться один-до-багатьох (англ. one-to-many). Відзначте доступний для цього типу відносин параметр забезпечення цілісності даних (англ. enforce referential integrity). Якщо таблиці «**ПІБ-Група**» і «**ПІБ-Студент**», порожні або їх вже заповнено даними, відповідають вимогам цілісності, між таблицями буде встановлено зв'язок. Встановіть прапорці каскадне оновлення пов'язаних полів (англ. cascade update related fields) і каскадне видалення пов'язаних записів (англ. cascade delete related records), тоді буде забезпечено автоматичне коректування даних для збереження цілісності у взаємозалежних таблицях. Встановіть зв'язок для інших пар таблиць: «**ПІБ-Кафедра**» - «**ПІБ-Викладач**» за ключем **pibdept**, «**ПІБ-Дисципліна**» - «**ПІБ-Навчання**» за ключем **pibdiscode**, «**ПІБ-Викладач**» - «**ПІБ-Навчання**» за ключем **pibtab**, «**ПІБ-Група**» - «**ПІБ-Навчання**» за ключем **pibcode** (рис. 13).

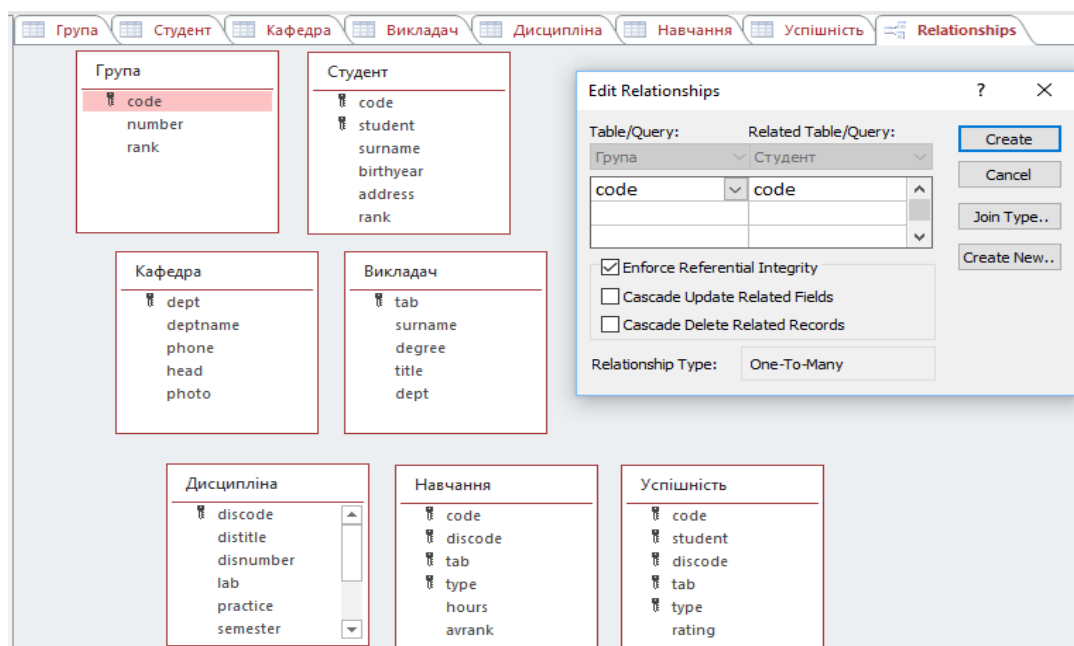


Рисунок 13 – Створення зв'язку «один-до-багатьох»

⁷ довільні растрові зображення розміром до 200 точок за шириною й висотою

Встановіть зв'язок по складеному ключу **pibcode** + **pibstudent** між таблицями «**ПІБ-Студент**» і «**ПІБ-Успішність**». Для цього в головній таблиці «**ПІБ-Студент**» виділіть обидва цих поля, утримуючи клавішу Ctrl, і перетягніть обидва поля на поле **pibcode** у підпорядкованій таблиці «**ПІБ-Успішність**». У вікні редагування зв'язків для поля **pibstudent** головної таблиці виберіть відповідне поле підпорядкованої таблиці «Зв'язана таблиця / запит» (англ. related table / query). У цьому ж вікні встановіть режим забезпечення цілісності даних та інші параметри зв'язку.

Аналогічно за складеним ключем **pibcode** + **pibdiscode** + **pibtab** + **pibtype** встановити зв'язок між парою таблиць «**ПІБ-Навчання**» та «**ПІБ-Успішність**». Скріншот схеми даних додати до протоколу.

10. Перевірка підтримки зв'язної цілісності. Перевіримо, як впливають встановлені раніше параметри підтримки цілісності у разі внесення змін до таблиці «**ПІБ-Група**» та «**ПІБ-Студент**», пов'язані одно-багатозначними відносинами. Відкрийте таблиці «**ПІБ-Група**» та «**ПІБ-Студент**» у режимі таблиці. Змініть значення ключового поля **pibcode** (номер групи) у запису таблиці «**ПІБ-Група**», яка має підлеглі записи. Для завершення режиму редагування перейдіть до іншого запису. Переконайтеся, що для підпорядкованої таблиці «**ПІБ-Студент**» автоматично змінилося значення поля **pibcode** для студентів, що навчаються в цій групі. Відбувається зміна, оскільки встановлено параметр, що задає каскадне оновлення пов'язаних полів. Відкриваючи інші підлеглі таблиці переконайтеся, що каскадне оновлення поширюється також на підпорядковані таблиці таких рівнів як «**ПІБ-Навчання**» і «**ПІБ-Успішність**».

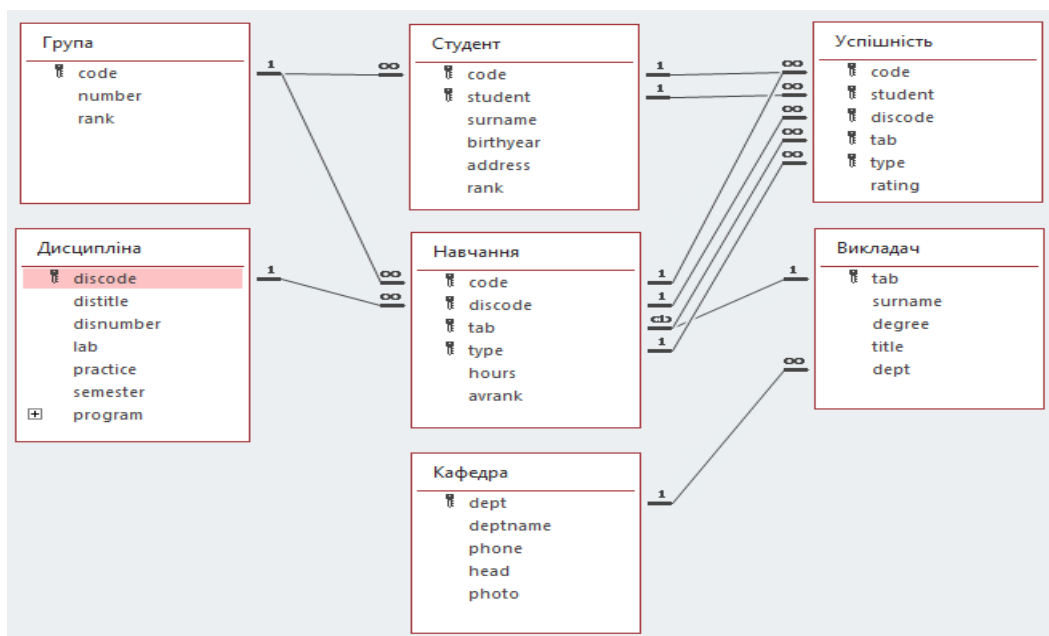


Рисунок 14 – Схема бази даних «**Прізвище – Навчальний процес**»

Підготуйте новий запис у підпорядкованій таблиці «**ПІБ-Студент**» зі значенням ключа зв'язку **pibcode**, не представленим у головній таблиці «**ПІБ-Група**». Переконайтеся, що включити такий запис у підпорядковану таблицю неможливо. У головній таблиці «**ПІБ-Група**» виділіть запис, у якого

є підлегли, і виконайте команду «Редагування | Видалити». Переконайтеся, що разом з видаленням запису в головній таблиці «**ПІБ-Група**» видаляються всі підлегли записи, так як було встановлено параметр каскадного видалення пов'язаних записів.

Якщо для одно-багатозначних зв'язків, встановлених в схемі даних, не задано параметр забезпечення цілісності даних, то користувач у разі додавання, видалення записів і змінюванні значень ключових полів повинен сам відслідковувати несуперечність даних і цілісність зв'язків. Якщо встановлено тільки прапорець забезпечення цілісності даних, то Access дає змогу користувачеві:

- додати запис в таблицю, яка не підпорядкована ніякій іншій таблиці в схемі даних (знаходиться на верхньому рівні ієрархії);
- додати запис у підпорядковану таблицю, коли в головній є запис зі значенням ключа зв'язку;
- видалити запис у таблиці, якщо немає пов'язаних з нею записів у підпорядкованих таблицях.

Якщо поряд з прапорцем «*Забезпечення цілісності даних*» встановлено прапорець «*Каскадне видалення пов'язаних записів*», то користувач може видалити запис. При цьому автоматично буде видалено і всі підпорядковані записи. Якщо поряд з параметром «*Забезпечення цілісності даних*» задано параметр каскадного оновлення пов'язаних полів, то користувач може змінювати значення ключового поля. При цьому автоматично буде оновлено значення ключів і у всіх пов'язаних підпорядкованих записах. Таким чином, у разі завантаження таблиць за встановлених параметрів цілісності система відстежує коректність зв'язків під час введення записів. MS Access дає можливість під час перегляду таблиці відображати записи підпорядкованих таблиць. Тому користувач може контролювати коректність зв'язків у видимому ланцюжку таблиць у разі додавання, видалення та редагування записів.

11. Відкрийте таблицю «**ПІБ-Кафедра**». Якщо таблиця має стовпець з плюсами, натисніть значок «+» у рядку запису, щоб з'явилися записи підпорядкованої таблиці, пов'язані з поточним записом. При цьому «+» на значку перетвориться на «-». Натисканням на значку «-» підпорядковані записи закриваються. Таким чином може бути відкрито підпорядковані записи кожного запису головної таблиці.

Відкрийте всі підпорядковані записи, скориставшись командою «Розгорнути усі» меню «Дочірні таблиці» (англ. subdatasheets) групи «Записи» вкладки «Головна». За допомогою команди «Згорнути усі» поверніть таблицю до початкового вигляду.

Перейдіть у режим конструктора та натисніть на вкладці «Конструктор» команду «Властивості» (англ. property sheet) групи «Відобразити/Приховати». У властивості таблиці *ім'я підпорядкованої таблиці* (англ. subdatasheet name) за замовчуванням встановлено параметр *Авто*, що і визначає єдиний зв'язок таблиці і виведення стовпця зі значками плюс для відкриття записів за зв'язком.

Видалення або додавання підпорядкованих таблиць можливе за допомогою команд «Дочірня таблиця» і «Видалити» меню «Дочірні таблиці» (англ. subdatasheets) групи «Записи» вкладки «Головна».

12. Введення і редагування даних взаємозалежних таблиць. Відкрийте таблицю «*ПІБ-Навчання*». Виберіть запис про заняття, для якого потрібно ввести оцінки, отримані студентами. Відкрийте підпорядковану таблицю успішності, натиснувши значок «+» у рядку обраного запису. Створіть кілька записів успішності студентів, пов'язаних з записом таблиці «*ПІБ-Навчання*». При цьому значення ключа зв'язку **pibcode**, **pibdiscode**, **pibtab**, **pibtype** вводиться в поле підпорядкованого запису автоматично. Таким чином, для створення підпорядкованого запису досить ввести тільки частину ключа **pibstudent** і отриману ним оцінку.

Для контролю правильності введення номера прізвища студента відобразіть його прізвище з головної таблиці «*ПІБ-Студент*». Встановіть зв'язок таблиці успішності з головною таблицею «*ПІБ-Студент*» за ключем зв'язку **pibcode** і **pibstudent**, виконавши команду «Дочірня таблиця», і натисніть на значку «+» в записі про успішність.

13. Оформіть протокол лабораторної роботи.

Оформлення протоколу

Після оформлення протоколу лабораторної роботи до заздалегідь підготовленого складників додаються роздруківки скріншотів створених таблиць у режимі конструктора й властивостей кожного поля цих таблиць з відповідними поясненнями, так і роздруківки скріншотів ходу виконання усіх інших пунктів завдання.

Контрольні запитання

1. Що охоплюють вимоги нормалізації?
2. Назвіть види зв'язків інформаційних об'єктів.
3. У якому режимі відбувається введення даних у таблицю?
4. Чи можуть повторюватись у таблиці значення в одному з полів складеного ключа?
5. У якому режимі відбувається візуальне налаштування макету таблиці?
6. Для чого створюється схема даних бази Access?
7. У яких відносинах повинні бути таблиці, щоб для них можна було встановити параметри забезпечення цілісності даних?
8. Чи можна в підпорядковану таблицю «*Студент*» додати запис про студента групи, яку не представлено в таблиці «*Група*», якщо для цих таблиць забезпечується цілісність даних?
9. Що станеться у разі змінювання значення ключового поля у головній таблиці, якщо для її зв'язку з підпорядкованою встановлено прапорець «Каскадне оновлення пов'язаних полів»?
10. Для чого призначено знак «+» у лівому стовпці відкритої таблиці?

Лабораторна робота № 3

Робота з даними в таблиці у СКБД MS Access

Мета роботи. Набути умінь використання прийомів роботи з даними у середовищі MS Access 2016.

Завдання. Розглянути засоби заміни, сортування й фільтрування даних у СКБД MS Access 2016.

Короткі теоретичні відомості

Під час оброблення даних у режимі таблиці і форми крім перегляду, оновлення, додавання, видалення записів і значень полів часто необхідно знайти задані значення в полях таблиць великого обсягу, замінити їх, відсортувати записи відповідно до значень певного поля або навіть декількох полів. Найважливішим і часто розв'язуваним завданням є відбір записів, що відображаються в таблиці або формі, поля яких містять значення, що задовольняють певним умовам. Це завдання вирішується за допомогою фільтрації записів у режимі таблиці або форми. Відбір записів може бути виконано і за допомогою запитів, призначених для виконання складніших завдань оброблення даних і охоплюють засоби відбирання записів згідно з умовами.

Діалогове вікно «Пошук і заміна» використовують у разі потреби знайти і за необхідності замінити невеликі обсяги даних, а також коли запит для пошуку або заміни значень використовувати незручно. Можна виконувати пошук за форматом даних, а також шукати часткові або повні збіги значень у полі. Діалогове вікно обробляє рядок пошуку як шаблон і повертає всі записи, які відповідають шаблону. Діалогове вікно веде пошук тільки в одній таблиці, а не у всій базі даних. Якщо потрібно використовувати діалогове вікно для пошуку в кількох таблицях, можна створити запит вибірки, щоб відібрати потрібні дані, а потім виконати пошук за результатами цього запиту.

Можна відсортувати записи в таблиці, запиті, формі або звіті в одному або декількох полях. З метою налаштування результатів, можна виконати сортування записів за декількома полями. У разі сортування за декількома полями важливо визначити, що називають зовнішнім і внутрішнім полями сортування. Щоб отримати потрібні результати, необхідно вказати відповідні поля в якості внутрішніх і зовнішніх полів сортування.

Для перегляду і змінювання записів бази даних, що задовольняють зазначеним користувачем умовам відбору, передбачено фільтрацію таблиці. Фільтр – це набір умов, застосовуваний для відбору підмножини записів у таблиці, формі або запиті. Фільтрація – це зручний спосіб перегляду тільки тих даних, які повинні відображатися в базах даних MS Access. Фільтри використовують для відображення певних записів у формі, звіті, запиті або таблиці. Застосовуючи фільтр, можна обмежити дані у відображенні, не

змінюючи структуру основного об'єкта. Так як в отриманому відображенні після застосування фільтра містяться тільки записи з вибраними значеннями, інші дані залишаються прихованими, поки не буде очищено фільтр. Доступність команд фільтра залежить від типу і значень поля. MS Access дає змогу використовувати такі типи фільтрів: загальний фільтр, фільтр за виділенням, фільтрація за формою і розширений фільтр.

Звичайний фільтр дає змогу зробити відбір записів шляхом введення значень у порожню таблицю або форму за командою «Змінити фільтр» (англ. filter by form) з меню «Розширені» групи «Сортування і фільтрування» головної вкладки. У вікні звичайного фільтра можна створити новий фільтр або переглянути і відкоригувати раніше створений фільтр. На вкладці «Знайти» (англ. look for) у поля фільтра вводять значення, за якими будуть відбирати записи. Значення можна вводити або обрати зі списку. Введення значень у кілька полів одного рядка фільтра визначає відбір записів, у яких присутні всі зазначені значення. Задані умови об'єднують логічною операцією «і». Для того щоб вказати альтернативний варіант відбору записів, що міститимуться у наборі відображуваних записів, вибирається вкладка «Або» (англ. or) у лівому нижньому куті вікна фільтру. Набір умов, введених у різні рядки, зв'язується логічною операцією «або». Наступний альтернативний варіант може бути задано на наступній вкладці «Або». Фільтр буде відбирати записи, що містять в полях усі значення, вказані на вкладці «Знайти» і на вкладках «Або».

Розширений фільтр надає вікно для вказування умов відбору, що відображаються у записах таблиці або форми. У верхній частині вікна виводиться список полів активної таблиці, а в нижній частині вікна – бланк запиту. У рядок бланка запиту зі списку перетягуються поля, за якими потрібно задати умови відбору записів (рис. 15). Умови відбору записів вводять в комірки ряду «Умова відбору» (англ. criteria) для одного або кількох полів. Крім того, в бланку запиту у рядку «Сортування» можна обрати сортування для одного або кількох полів.

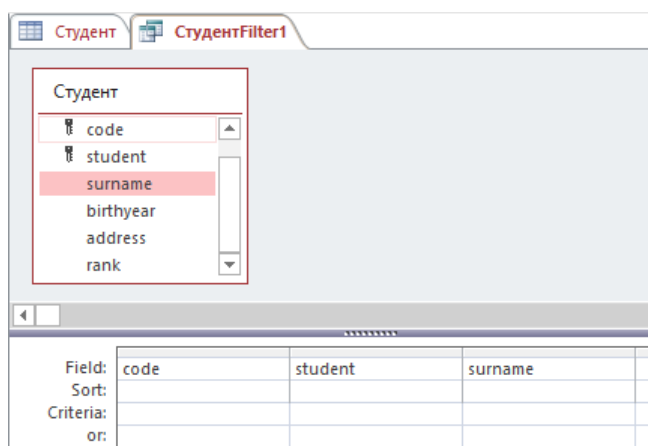


Рисунок 15 – Розширений фільтр таблиці «*ЛІБ-Студент*»

У рядку «Умови відбору» записують вираз, результатом якого є конкретне значення. Вираз складається з операндів і операторів. У якості

операндів виразу можуть використовувати літерали, константи та ідентифікатори (посилання). Літералами є конкретні значення, які MS Access сприймає так як їх записано, числа, текстові рядки, дати. Текстові рядки записують у подвійних лапках, дати – між знаками #. Константами є значення, які не змінюються й визначено в Access (наприклад, True, False, Так, Ні, Null). Ідентифікатор – це посилання на значення поля, елементу керування або властивості. Ідентифікаторами можуть бути імена полів, таблиць, форм, звітів тощо, які записують у квадратні дужки. Зазвичай MS Access проводить автоматичну підстановку дужок.

У багатьох випадках посилання на конкретне значення повинно вказувати точне його місце розташування в ієрархії об'єктів бази даних, починаючи з об'єкта верхнього рівня. Якщо необхідно вказати посилання на поле в конкретній таблиці, формі, звіті, то перед іменем поля потрібно поставити ім'я таблиці, форми або звіту, записане в квадратні дужки і відокремлене від імені поля знаком оклику. Наприклад, посилання на поле в таблиці прийме вигляд [Ім'я таблиці]! [Ім'я поля].

Операторами порівняння та логічними операторами, використання яких дозволено у виразі умови відбору є: =, <, >, <>, <=, >=, Between, In, Like, And, Or, Not. Вони визначають операцію над одним або кількома операндами. Якщо вираз в умові відбору не містить оператора, то за замовчуванням використовується оператор =. Якщо текстові значення у виразі містять пробіли або знаки пунктуації, то їх вводять у подвійних лапках. В іншому випадку лапки можна не вводити, їх буде додано автоматично. Можливе використання операторів шаблону: зірочка (*) і знак питання (?). Оператор Between дає змогу задати інтервал для числового значення. Оператор Like дає змогу використовувати зразки, що містять символи шаблону в разі пошуку в текстових полях. Наприклад, Like "Гончаренко *". Умови відбору, які задано в одному рядку, зв'язують за замовчуванням за допомогою логічної операції «і», задані в різних рядках – за допомогою логічної операції «або». Ці операції також може бути встановлено явно у виразі умови відбору за допомогою операторів «and» і «or», відповідно.

Сформувавши умову відбору можна за допомогою будівника виразів (англ. expression builder). Перейти у вікно будівника виразів можна за допомогою вибору команди «Побудувати» з контекстного меню поля. Після введення виразу в бланк MS Access виконує синтаксичний аналіз виразу і відображає його відповідно до результатів цього аналізу.

Послідовність виконання

1. *Пошук і заміна значень у полях таблиць.* Відкрити базу даних «Прізвище – Навчальний процес», відкрити таблицю «ПІБ-Студент». Необхідно замінити ПІБ одного зі студентів на власні ПІБ. Для цього скористайтесь командою «Знайти» або «Замінити» у групі «Знайти» вкладки «Головна». У діалоговому вікні «Знайти і замінити» перейти на вкладку «Знайти» та у полі «Знайти» вкажіть ПІБ студента повністю або

частково. У другому випадку варто змінити значення параметру співпадіння (англ. match) на «будь-яка частина поля». На вкладці «Замінити» цього діалогового вікна у полі «Замінити» вказати власні ПІБ і виконати команду заміни. У протокол внести відповідні скріншоти і пояснення.

2. Сорткування записів. Для сортування відображуваних записів таблиці «ПІБ-Студент» в порядку зростання або спадання значень поля **pibsurname** встановіть курсор у це поле і натисніть відповідну кнопку у групі «Сортування і фільтрування» головної вкладки стрічки MS Access:

- сортування за зростанням (англ. ascending);
- сортування за спаданням (англ. descending);
- розширене сортування (англ. advanced).

Перед сортуванням записів таблиці за кількома полями введіть в таблицю записи про студентів з однаковими прізвищами, але різними роками народження. Виділіть стовпці **pibsurname** і **pibbirthyear** та виконайте команду сортування. У протокол внести відповідні скріншоти і пояснення.

3. Фільтрування за виділенням фрагментом. Відкрити таблицю «ПІБ-Студент» в режимі таблиці. Для відбору всіх записів про студентів, прізвища яких починаються на певну літеру (бажано, щоб така літера першою у прізвищах зустрічалась два рази і більше), виділити цю літеру в полі **pib-surname** одного із записів, де прізвище починається з цієї літери.

Виконати команду «Починається зі **значення**» меню «Вибір» (англ. selection) групи «Сортування і фільтрування» головної вкладки стрічки MS Access або контекстного меню виділення (рис. 16). У результаті фільтрації в таблиці буде відображено тільки ті записи, в яких прізвище студента починається на вказану літеру.

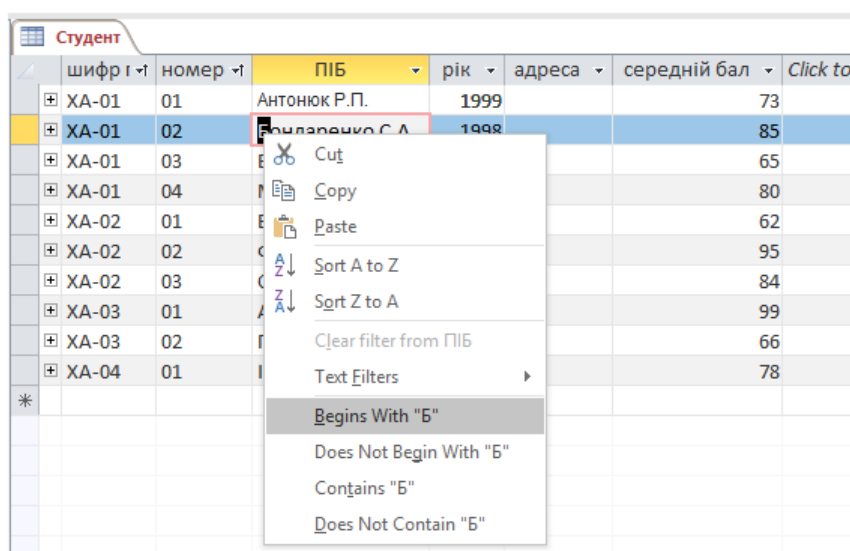


Рисунок 16 – Фільтрування за виділенням

Продовжити фільтрацію отриманої підмножини записів за додатковою умовою відбору. Серед знайдених студентів обрати тих, хто народився у певному році. Виділити в полі **pibbirthyear** одного із записів потрібне значення. Після виконання команди «Дорівнює значення» з усіх записів

підмножини у таблиці залишаються тільки ті, в яких поле **pibbirthyear** має задане значення.

У протокол внести відповідні скріншоти і пояснення. Відмінити фільтрування відповідною командою з групи «Сортування і фільтрування».

Для відбору записів, що не містять виділеного значення у полі, після виділення значення необхідно виконати команди «Не містить **значення**», «Не починається зі **значення**» або «Не дорівнює **значення**» для числових полів. Для відбору всіх записів про студентів, прізвища яких не починаються на певну літеру, виділити цю літеру в полі **pibsurname** одного із записів, де прізвище починається з цієї літери. Виконати команду «Не починається з **значення**» меню «Вибір» (англ. selection) групи «Сортування і фільтрування» головної вкладки стрічки MS Access або контекстного меню виділення. У результаті фільтрації в таблиці буде відображено тільки ті записи, в яких прізвище студента не починається на вказану літеру.

У протокол внести відповідні скріншоти і пояснення.

Для повторного використання фільтра виконати команду «Застосувати фільтр/сортування» з меню «Розширені» групи «Сортування і фільтрування» головної вкладки. Фільтр, створений у разі виконання фільтрування за виділенням, зберігається автоматично у разі збереження таблиці, запиту або форми. Після наступного відкриття таблиці цей фільтр можна застосувати знову.

4. Звичайний фільтр. Відкрити таблицю «**ЛІБ-Студент**». Виконати команду «Змінити фільтр», вказати для поля **pibrank** значення «>=**значення**» на вкладці «Знайти». Після виконання команди «Застосувати фільтр» буде відображено таблицю тільки із записами для яких значення у полі **pibrank** >=**значення**. У протокол внести відповідні скріншоти і пояснення.

Видалити фільтр за допомогою команди «Очистити» (англ. clear grid) з меню «Розширені» групи «Сортування і фільтрування» головної вкладки або скористайтесь контекстним меню відповідного поля з активним значком фільтрування.

5. Розширений фільтр. Відкрити таблицю «**ЛІБ-Студент**». Оберіть команду «Розширений фільтр» з меню «Розширені» групи «Сортування і фільтрування» головної вкладки. Перетягніть поле **pibrank** у вікно фільтру та створіть фільтр «**pibrank** < **значення1** або **pibrank** > **значення2**», де **значення1** < **значення2** з одночасним сортуванням результатів фільтрування (рис. 17).

Field:	rank
Sort:	Ascending
Criteria:	<70 Or >90
or:	

Рисунок 17 – Приклад застосування розширеного фільтру

Для збереження підготовленого фільтру, виконайте команду «Зберегти як запит» (англ. save as query) з меню «Розширені» групи «Сортування і

фільтрування» головної вкладки або скористайтесь контекстним меню поля запиту розширеного фільтру. Створений запит з'явиться серед об'єктів бази даних «Навчальний процес» і його можна буде використати повторно.

У протокол внести відповідні скріншоти і пояснення.

7. Оформити протокол лабораторної роботи.

Оформлення протоколу

Після оформлення протоколу лабораторної роботи до заздалегідь підготовлених складників додаються роздруковки скріншотів ходу виконання завдань.

Контрольні запитання

1. У якому діалоговому вікні виконують операцію заміни значення поля запису?
2. Чи обов'язково для пошуку задавати повне значення поля запису?
3. Чи можливо виконання сортування записів таблиці за кількома полями?
4. Які типи фільтрів може бути використано для відбору записів у таблицях і формах?
5. Яку команду застосовують для видалення збереженого фільтра?
6. З яких частин складається вікно розширеного фільтра?
7. Як задати в рядку «Умови відбору» текстовий рядок, що міститиме пробіли?
8. За допомогою якої команди зберігають розширений фільтр?
9. Що дають змогу задати оператори Between та Like?
10. Що таке будівник виразів і з якою метою його використовують?

Лабораторна робота № 4

Створення запитів в СКБД MS Access

Метою роботи є набуття вмінь та досвіду роботи із запитами у середовищі MS Access 2016.

Завдання. Розглянути засоби створення, виконання і змінювання запитів до бази даних у СКБД MS Access 2016.

Короткі теоретичні відомості

Запити є основним інструментом вибирання, оновлювання й обробляння даних у таблицях бази даних. MS Access відповідно з концепцією реляційних баз даних для виконання запитів використовує мову структурованих запитів SQL, але дає змогу створювати запити за допомогою майстра без використання інструкцій мови SQL. Основним видом запиту є запит на вибірку, результатом виконання якого є нова таблиця, яка існує до закриття

запиту. Ненормалізована, з повторюваними даними, таблиця запиту не зберігається у базі даних, зберігається лише запит.

MS Access надає зручний для користувача діалоговий графічний засіб формування запиту за зразком QBE (англ. query by example), за допомогою якого легко може бути побудований складний запит. Запит у режимі конструктора містить схему даних і бланк запиту, у якому формують структуру таблиці запиту і умови вибірки записів (рис. 18). Нижня панель є бланком запиту за зразком (бланком QBE), який і потрібно заповнити. У разі конструювання запиту досить додати таблиці в схему запиту, які містять необхідні дані, виділити і перетягнути необхідні поля з цих таблиць до бланку запиту та ввести умови відбору записів.

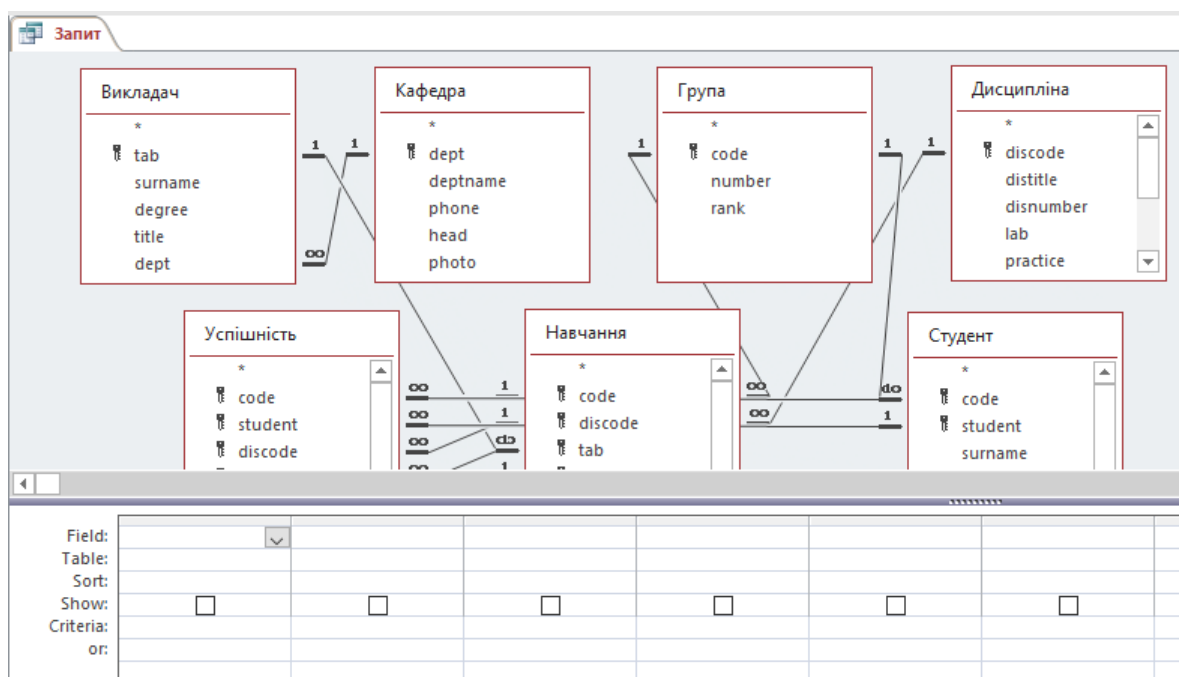


Рисунок 18 – Вікно конструктора запитів

Запит дає змогу:

- охопити у таблиці запиту необхідні поля таблиці;
- відібрати записи за умовами пошуку;
- створити нову таблицю запиту на основі об'єднання записів взаємопов'язаних таблиць;
- створити нову таблицю на основі даних існуючих таблиць;
- виконати розрахунки в отриманих записах;
- згрупувати записи;
- оновити поля для вибраної сукупності записів;
- видалити необхідні записи;
- додати необхідні записи.

У схемі даних запиту відображаються використовувані в запиті таблиці зі списком полів, а також зв'язки між ними, встановлені в схемі бази даних. Перший рядок у списку полів, зазначений зірочкою (*), позначає всі можливі значення полів таблиці. Крім зв'язків, внесених до запиту відповідно зі

схемою даних бази, MS Access автоматично встановлює зв'язки для об'єднання таблиць, якщо таблиці мають поля, які містять дані однакових типів з однаковими іменами. Зв'язки, не встановлені програмою MS Access автоматично, може встановити користувач, перетягнувши задіяні у зв'язку поля з одного списку полів в іншій.

Обраний тип об'єднання таблиць задається під час встановлення зв'язку між таблицями і визначає спосіб формування записів запиту. За замовчуванням зв'язки встановлюються з параметром об'єднання першого типу, тобто об'єднання тільки тих записів, в яких значення пов'язаних полів обох таблиць збігаються.

Бланк запиту за зразком представлений у вигляді таблиці, яка призначена для конструювання структури таблиці результату запиту та умов вибірки даних з вихідних таблиць. Спочатку ця таблиця порожня. Кожен стовпець бланка відноситься до одного поля, з яким потрібно працювати в запиті. Поля можуть використовуватися для включення їх у таблицю (результат виконання запиту), вказування сортування за ним, визначення умов відбору записів, а також для виконання обчислень у записах таблиці.

Однотабличні запити на вибірку даних. Однотабличний запит реалізує вибірку і перегляд результатів з однієї таблиці за умовами відбору. У запиті, як і в таблиці, для кожного запису можна виконати обчислення з використанням даних полів і створенням нового обчислюваного поля. Для створення обчислюваного поля потрібно увести вираз у комірку рядка «Поле» порожнього стовпця бланка запиту. У разі виконання запиту на основі значень полів, обчислюють нові значення такого поля.

У разі розв'язування практичних задач зручно вводити вираз до умови відбору в процесі виконання запиту без переходу в режим конструктора за допомогою параметру запиту. Ім'я параметру запиту задають у рядку «Умова відбору» у квадратних дужках. Під час виконання запиту це ім'я з'явиться у діалоговому вікні введення значення параметру (рис. 19). У запиті можна визначити кілька параметрів. Параметри можна використовувати не тільки окремо, але і як операнди виразів в умовах відбору або обчислюваних полях. Варто вказувати тип даних для параметру запиту для мінімізування помилок.

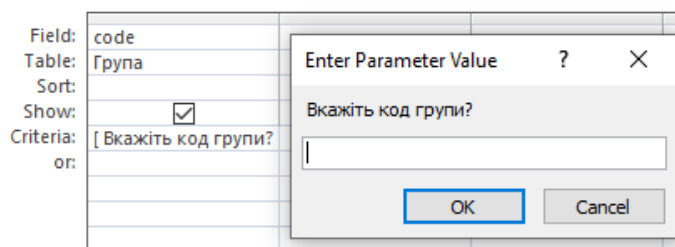


Рисунок 19 – Діалогове вікно введення значення параметру

Групові операції дають змогу виділити групи записів з однаковими значеннями у зазначених полях і використовувати для деяких полів цих груп одну зі статистичних функцій. У MS Access передбачено дев'ять статистичних функцій:

Sum	- сума значень деякого поля для групи.
Avg	- середнє значень поля у групі.
Max, Min	- максимальне і мінімальне значення поля у групі.
Count	- число значень поля в групі без урахування порожніх значень.
StDev	- порівняння відхилення від середнього поля в групі.
Var	- дисперсія значень поля в групі.
First, Last	- значення поля з першого чи останнього запису в групі.

Будь-який запит у MS Access реалізується за допомогою інструкцій мови SQL, які передаються на виконання в ядро бази даних Microsoft Jet. Більшість запитів можна створити в графічному режимі QBE, але і в цьому випадку запит зберігається у вигляді інструкцій SQL. Під час створення запиту в режимі конструктора програма MS Access автоматично створює еквівалентну інструкцію SQL, яку можна побачити, переключившись у режим SQL (командою «Режим SQL» групи «Результати» вкладки «Конструювання»). Зміни інструкції SQL, в свою чергу, відображаються в бланку запиту.

У разі створення форми, звіту або сторінки доступу до даних на основі кількох таблиць для визначення джерела записів створюється інструкція SQL. Якщо під час створення використано майстер, Access автоматично формує інструкцію SQL. Якщо майстер не використовувався, інструкція SQL формується користувачем у рядках *Джерело записів* (англ. Record source) або *Джерело рядків* (англ. row source). Також може використовуватися *Бланк запиту*, який викликають натисканням кнопки будівника у відповідному рядку.

Багатотабличні запити на вибірку даних. Багатотабличний запит дає змогу сформулювати запис результату шляхом об'єднання взаємопов'язаних записів з таблиць бази даних і охоплення потрібних полів з них. Багатотабличний запит здійснює об'єднання даних, які на етапі проектування були розділені (фрагментовані) на безліч об'єктів, що відповідають вимогам нормалізації. У результаті виконання запиту формується таблиця, яка містить повторювані дані, причому забезпечується зручне сприйняття їх користувачем.

Наприклад, к разі об'єднання двох нормалізованих таблиць, пов'язаних одно-багатозначними відносинами, результуючий запис утворюється на основі запису підлеглої таблиці, в яку додано поля з пов'язаного запису в головній таблиці. Зауважимо, що подібне об'єднання формує нову таблицю, в якій не виконується вимога третьої нормальної форми реляційної моделі даних. При цьому дані головної таблиці дублюються в різних записах.

При проектуванні і конструюванні запиту найважливішою умовою є правильне уявлення про джерело формування записів результату. Записи в таблиці багатотабличного запиту формуються на основі записів підпорядкованої таблиці нижнього рівня в схемі даних запиту.

Під час об'єднання записів пов'язаних таблиць між будь-якими двома таблицями бази даних може бути встановлено зв'язок, якщо вони мають

спільне поле або спільні поля. Зв'язок встановлюється незалежно від типу відносин між таблицями. Ці відносини можуть бути і невизначеними. Зв'язок встановлюється для об'єднання записів таблиць за спільним полем для їх оброблення. Крім того, є можливість задати спосіб об'єднання записів, який доповнить результуючий набір записів непов'язаними записами однієї або іншої таблиці. '

Для будь-якої пари пов'язаних таблиць можна обрати один з трьох способів об'єднання записів:

- об'єднання тільки тих записів, в яких пов'язані поля обох таблиць співпадають (за умовчанням);
- об'єднання тих записів, в яких пов'язані поля обох таблиць співпадають, а також об'єднання всіх записів з першої таблиці, для яких немає пов'язаних у другій, з порожнім записом другої таблиці;
- об'єднання тих записів, в яких пов'язані поля обох таблиць співпадають, а також об'єднання всіх записів з другої таблиці, для яких немає пов'язаних в першій, з порожнім записом першої таблиці.

Для встановлення в схемі бази даних зв'язків між таблицями за замовчуванням обрано перший спосіб об'єднання. У разі оброблення таблиць виконується цей спосіб об'єднання, якщо його спеціально не перевизначено.

Якщо між таблицями в запиті не встановлено зв'язок, то програма MS Access не знає, які записи пов'язані з якимись, і в таблиці запиту відображаються всі комбінації записів таблиць, тобто будуть об'єднуватися всі з усіма. Таким чином, якщо одна таблиця містить 10 записів, а інша 4, то в таблиці запиту буде 40 записів. Це називається повне об'єднання або Декартове множення.

Запити на змінювання. Запит на створення таблиці використовується для збереження результатів запиту. Цей вид запиту базується на запиті на вибірку, але на відміну від нього зберігає таблицю, що містить результати запиту. Необхідність у збереженні результатів запиту виникає, наприклад, в тому випадку, коли один і той самий запит виконується неодноразово, а вихідні дані для нього не змінюються. Використовуючи замість запиту таблицю, можна значно прискорити доступ до даних, особливо якщо запит побудовано на кількох таблицях великого обсягу. Збереження таблиці запиту корисно і тоді, коли необхідно накопичувати підсумкові дані протягом тривалого часу, а дані, на яких базувався запит, можна видалити.

Запит на оновлення (англ. update query) використовують для оновлення інформації в полях таблиці бази даних. Зміни вносяться до поля цілої групи записів, що відбираються з допомогою вказаних користувачем умов відбору, а не окремого запису. У результаті одним запитом можна виконати необхідні зміни в багатьох записах.

За допомогою запиту на додавання (англ. append query) проводиться додавання записів з таблиці з результатами запиту до таблиці бази даних. Тому в запиті повинно бути сформовано записи з полями, які відповідають

доповнюваним записам таблиці бази даних. Структура запису таблиці запиту може не повністю співпадати зі структурою-запису таблиці, в яку додаються записи. У записі запиту бути менше полів, якщо на поля в таблиці, до якої додають записи, не накладено вимогу щодо обов'язковості їх заповнення. Допускається невідповідність типів полів, якщо можливо перетворення типу даних поля в тип даних іншого поля. Крім того, до бланку запиту можна включати поля, за якими задаються умови відбору.

Запит на видалення (англ. delete query) дає змогу видалити записи з однієї таблиці або з кількох взаємозалежних таблиць, для яких встановлено параметр забезпечення цілісності даних «Каскадне видалення пов'язаних записів». У запиті вказують таблицю, з якої необхідно вилучити записи, і задають умови відбору записів.

Перехресний запит є спеціальним запитом підсумкового типу. Такий запит дає змогу побачити обчислювані значення у вигляді перехресної таблиці, яка нагадує електронну таблицю. У перехресній таблиці лівий стовпець утворює заголовки рядків зі значень одного поля, верхній рядок утворює заголовки стовпців із значень іншого поля, а на перетині рядків і стовпців розміщуються підсумкові значення, обчислені за значенням третього поля. Для отримання підсумкових значень записи групують за полями, використовуваними в якості заголовків рядків і стовпців, і для значень третього поля в отриманих групах записів застосовують одну з вибраних статистичних функцій. Можна передбачити виконання статистичної функції та отримання загального підсумкового значення для кожного рядка в цілому.

Послідовність виконання

1. Однотабличні запити на вибірку

1.1. Створення запиту на вибірку з логічними операціями в умові відбору. Розглянемо процес конструювання однотабличного запиту на вибірку на прикладі отримання інформації з таблиці «**ПІБ-Дисципліна**» бази даних «**Прізвище – Навчальний процес**». Нехай необхідно вибрати предмети, на вивчення яких відводиться не більше 100 годин, крім того передбачено лабораторні роботи, а також обрати предмети, за якими заняття займають більше 150 годин і число семестрів вивчення цього предмету не більше двох. Результат повинен містити найменування предмета (**pibdistitle**), загальне число годин з предмета (**pibdisnumber**), кількість годин лабораторних занять (**piblab**) і число семестрів (**pibsemester**).

Для створення запиту у вікні бази даних оберіть вкладку стрічки «Створити» і в групі «Запити» натисніть команду «Конструктор запитів». Відкриється порожнє вікно запиту на вибірку в режимі конструктора та діалогове вікно додавання таблиць. Додайте відповідний скріншот та пояснення у протокол. У діалоговому вікні оберіть таблицю «**ПІБ-Дисципліна**» і натисніть команду «Додати» та закрийте діалогове вікно. Обрану таблицю буде відображено в області схеми даних запиту.

Для видалення таблиці зі схеми даних запиту можна встановити на неї курсор і натиснути клавішу «Delete» на клавіатурі або скористатись командами вкладки «Конструктор» або контекстного меню таблиці. Для додавання таблиці можна скористатись командою «Відобразити таблицю» вкладки «Конструктор» або контекстного меню бланку запиту.

У вікні конструктора перетягніть зі списку полів таблиці «*ПІБ-Дисципліна*» поля **pibdistitle**, **pibdisnumber**, **piblab** і **pibsemester** у стовпці бланку запиту в рядок «Поле». У рядку «Виведення» (англ. show) відзначити поля, інакше їх не буде відображено в таблиці запиту.

Якщо ви помилково перетягнули в бланк запиту непотрібне поле, видаліть його. Для цього перемістіть курсор в область маркування стовпця зверху, де він набуде вигляду чорної стрілки, спрямованої вниз, і клацніть кнопку миші. Стовпець виділиться. Натисніть клавішу «Delete» або виконайте відповідні команди вкладки «Конструктор» або контекстного меню стовпця.

Сформульовані в задачі умови вимагають формування такого логічного виразу:

(**pibdisnumber** <= 100 and **piblab** <> 0) or
(**pibdisnumber** > 150 and **pibsemester** <3)

Запишіть умови з перших дужок у відповідних полях **pibdisnumber** і **piblab** першого рядка «Умови відбору» (рис. 20). Між умовами в різних полях одного рядка виконується логічна операція and. Умови з других дужок запишіть у відповідних полях **pibdisnumber** та **pibsemester** другого рядка «Умови відбору». Між умовами, записаними в різних рядках, виконується логічна операція or. Додати відповідний скріншот та пояснення у протокол.

Field:	distitle	disnumber	lab	semester
Table:	Дисципліна	Дисципліна	Дисципліна	Дисципліна
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		<= 100	<> 0	
or:		> 150		< 3

Рисунок 20 – Умови відбору записів таблиці «*ПІБ-Дисципліна*»

Виконайте запит, натиснувши команду «Виконати» або «Режим таблиці» у групі «Результати» вкладки «Конструктор». На екрані з'явиться вікно запиту в режимі таблиці з записами з таблиці «*ПІБ-Дисципліна*», що відповідають заданим умовам відбору. Додайте відповідний скріншот та пояснення у протокол.

Збережіть запит з іменем «*Прізвище-Запит01*». Закрийте запит. Виконайте збережений запит за допомогою команди «Відкрити».

1.2. Використання в умові відбору виразів з іменами полів. У п. 1.2 в умові відбору як операнди використано конкретні значення для відбору за полями. Необхідно створимо запит, умови відбору якого порівнюють значення у різних полях.

Нехай необхідно перевірити правильність вказаного обсягу дисциплін у таблиці «*ПІБ-Дисципліна*». За запитом потрібно відібрати тільки ті записи, у яких значення в полі **pibdisnumber** не дорівнює значенню, одержуваному в разі додавання значень полів **piblab** і **pibpractice**, для цього примусово змініть значення поля **pibdisnumber** кількох записів на більші або менші збережених.

Для створення запиту у вікні бази даних оберіть вкладку стрічки «Створити» і в групі «Запити» натиснути команду «Конструктор запитів». Відкриється порожнє вікно запиту на вибірку в режимі конструктора та діалогове вікно додавання таблиць. У вікні конструктора перетягніть зі списку полів таблиці «*ПІБ-Дисципліна*» поля **pibdistitle**, **pibdisnumber**, **piblab** і **pibpractice** у стовпці бланка запиту в рядок «Поле». Вкажіть у бланку запиту для стовпця **pibdisnumber** у рядку «Умови відбору» «<>[**piblab**] + [**pibpractice**]». Виконайте запит. Додайте відповідний скріншот та пояснення у протокол. Збережіть запит з іменем «*Прізвище-Запит02*». Закрийте запит.

1.3. Створення запиту з обчислюваним полем. Нехай необхідно знайти записи про дисципліни, в яких загальна кількість годин не збігається з сумою годин лабораторних і практичних занять. Варто розрахувати різницю між загальним числом годин з дисципліни (поле **pibdisnumber**) і сумою годин лабораторних (поле **piblab**) та практичних (поле **pibpractice**) занять. У відповіді потрібно відобразити тільки ті записи, для яких ця різниця не дорівнюватиме нулю.

Створіть у режимі конструктора запит на вибірку для таблиці «*ПІБ-Дисципліна*». Перетягніть до бланку запиту поля **pibdistitle**, **piblab**, **pibpractice**, **pibdisnumber**. Для підрахунку різниці створіть обчислюване поле в порожній клітинці рядка «Поле» із записом «[**pibdisnumber**] - [**piblab**] - [**pibpractice**]». Для відбору записів з ненульовим значенням різниці в обчислюваному полі в рядок «Умова відбору» вкажіть «<> 0».

Після введення виразу система за замовчуванням формує ім'я обчислюваного поля «вираз 1:» (або англ. «Expr1:»). Це ім'я з'явиться перед виразом «[**pibdisnumber**] - [**piblab**] - [**pibpractice**]». Ім'я поля може бути виправлено безпосередньо в бланку запиту. Введіть нове ім'я поля – «обсяг не дорівнює л.р.+п.з.».

Для формування складного виразу в обчислюваному полі доцільно використовувати будівник виразів. Будівник дає змогу вибрати необхідні у виразі імена полів з таблиць, запитів, форм, знаки операцій, функції. Видаліть вираз в обчислюваному полі і використайте будівник для його формування.

Викличте будівник виразів командою «Побудувати» в контекстному меню рядка «Поле» обчислюваного поля. У лівій частині вікна будівника

виразів оберіть таблицю «**ПІБ-Дисципліна**». Праворуч відобразиться список її полів. Послідовно вибирайте потрібні поля подвійним натисканням і встановіть знаки операцій. У такому разі у верхній частині вікна сформується вираз.

Збережіть запит з іменем «**Прізвище-Запит03**». Закрийте запит. Наведіть у протоколі заповнений бланк запиту, вікно будівника виразів під час формування обчислюваного поля та результат виконання запиту.

1.4. Конструювання запиту з функцією Count. Розглянемо технологію конструювання однотобличного запиту з груповою операцією на прикладі таблиці «**ПІБ-Студент**». Визначимо фактичне число студентів у групах.

Створіть у режимі конструктора запит на вибірку для таблиці «**ПІБ-Студент**». З таблиці «**ПІБ-Студент**» перетягніть до бланку запиту поле **pibstudent**. За цим полем буде проводитися групування. Виконайте команду «Групові операції» (англ. totals) з групи «Відобразити/Приховати» вкладки «Конструктор». У бланку запиту з'явиться рядок «Групова операція», у якому для всіх полів вказано групування. Замініть опцію групування у стовпці **pibstudent** на «Count». Виконайте запит.

Замініть підпис отриманого поля «CountOf**pibcode**» на «Фактичне число студентів». Для цього в бланку запиту встановіть значення властивості caption «Фактичне число студентів». Повторно виконайте запит. Представте результати запиту у протоколі. Збережіть запит з іменем «**Прізвище-Запит04**». Закрийте запит.

1.5. Запит з функцією Avg. Підрахуємо середній бал у групі. Сформуйте запит на вибірку для таблиці «Студент» з функцією підрахунку середнього значення для поля **pibrank**. Для обмеження точності відображення результату двома знаками виберіть у вікні властивостей поля у рядку формату поля – фіксований. Відобразіть у протоколі результат виконання запиту. Збережіть запит з іменем «**Прізвище-Запит05**». Закрийте запит.

1.6. Запит з кількома статистичними функціями. Виконайте підрахунок кількості студентів та середнього прохідного балу в групі в одному запиті. Відобразіть у протоколі бланк запиту і результат його виконання. Збережіть запит з іменем «**Прізвище-Запит06**». Закрийте запит.

1.7. Завдання умов відбору в запитах з груповими операціями. У запит з груповими операціями можна включати поля з вказуванням умов відбору записів з таблиць. У такому разі спочатку проводиться відбір записів, а потім групування.

Підрахуйте в число студентів, що одержали бал більше певного балу, наприклад, «80», та їхній середній бал. У запит «**Прізвище-Запит6**» додайте ще одне поле **pibrank**. У цьому полі-копії вкажіть у рядку групових операцій «Where», зніміть галочку у рядку «Show», а у рядку умови відбору задайте «> 80».

Відобразіть у протоколі бланк запиту і результат його виконання. Збережіть запит з новим іменем «**Прізвище-Запит07**». Закрийте запит.

1.8. Перегляд існуючого запиту у режимі SQL. Відкрийте збережений запит «**Прізвище-Запит7**» у режимі конструктора. Для переходу в режим SQL

виберіть у групі «Результати» вкладки «Конструктор» команду «Режим SQL».

```
SELECT Count(ПІБ-Студент. pibstudent) AS CountOfpibstudent,  
        Avg(ПІБ-Студент. pibrank) AS AvgOfpibrank  
FROM ПІБ-Студент  
WHERE (((ПІБ-Студент. pibrank)>80));
```

Інструкцію **select** використано для вибору полів і записів з таблиць бази даних і формування таблиці запиту - набору записів. Вираз **from** визначає таблиці або запити, які є джерелом даних для даного запиту. Вираз **where** задає умови відбору записів. Зарезервоване слово **as** дає змогу визначити для обчислюваного поля псевдонім.

Перейдіть у режим конструктора, внесіть зміни до бланку запиту (наприклад, змініть умову, додайте поле, тощо). Поверніться в режим SQL і перегляньте як змінилася інструкція SQL. Щоб переконатися, що запити можна редагувати в режимі SQL, а не тільки в режимі конструктора, внесіть зміни до інструкції SQL, потім перейдіть в режим конструктора і переконайтеся, що до бланку запиту внесено зміни.

Додайте у протоколі змінений бланк запиту і його відображення у режимі SQL. Збережіть запит з новим іменем «**Прізвище-Запит08**». Закрийте запит.

2. Багатотабличні запити на вибірку даних

2.1. Об'єднання записів у багатотабличних запитах.

(А) Розглянемо, як впливає вибраний для зв'язаних таблиць спосіб об'єднання на результат об'єднання записів цих таблиць, в запиті на вибірку з таблиць «**ПІБ-Студент**» і «**ПІБ-Успішність**», які знаходяться в однобагатозначних відносинах. Між цими таблицями встановлено зв'язок за полями **pibcode** та **pibstudent**.

У таблицю успішності запис з номером групи ХА-01 (див. тексти завдань до попередніх лабораторних робіт) і номером студента 05 може бути внесено тільки у випадку, якщо для зв'язку між таблицями не вказано параметр забезпечення цілісності.

Відкрийте схему даних. Між таблицями «**ПІБ-Студент**» і «**ПІБ-Успішність**» раніше встановлено зв'язки, встановіть курсор на лінію зв'язку і натисніть двічі. У вікні «Редагування зв'язків» вказано параметр забезпечення цілісності даних, відмовтеся від нього. Це дасть змогу додати в таблицю «**ПІБ-Успішність**» підлеглі записи, для яких немає головного. За замовчуванням для зв'язку встановлюється перший спосіб об'єднання. Переконайтеся в цьому, натиснувши у діалоговому вікні «Редагування зв'язків» кнопку «Тип об'єднання», яка відкриває вікно параметрів об'єднання. Закрийте вікно «Властивості об'єднання» та «Редагування

зв'язків». Зверніть увагу на вигляд лінії зв'язку, для якої не встановлено параметр забезпечення цілісності та обрано перший спосіб об'єднання. Закрийте схему даних.

Створіть за допомогою майстра простий запит на вибірку на основі цих двох таблиць. На вкладці «Створити» у групі «Запити» натисніть команду «Майстер запитів». У діалоговому вікні «Новий запит» вкажіть простий запит. У діалоговому вікні створення простих запитів оберіть таблицю «ПІБ-Студент» та її поля **pibcode**, **pibstudent** і **pibsurname** та таблицю «Успішність» і її поля **pibdiscode**, **pibtab** і **pibrating**. Задайте ім'я запиту «Прізвище-Запит09». Відобразіть у протоколі заповнене діалогове вікно майстру запитів і результат виконання запиту. Вкажіть записи якої таблиці є джерелом записів результату, з якої таблиці додаються записи та у якому відношенні знаходяться ці таблиці.

(Б) Для зміни способу об'єднання таблиць відкрийте запит «Прізвище-Запит09» в режимі конструктора.

Параметри об'єднання потрібно змінювати в запиті. Зміни, внесені в схему даних після створення запиту, не вносяться до запиту.

Натисніть правою кнопкою миші на лінії зв'язку таблиць і виконайте команду «Параметри об'єднання» контекстного меню. У вікні «Параметри об'єднання» виберіть другий спосіб об'єднання записів таблиць. Виконайте ті самі дії для другої лінії зв'язку. Представте у протоколі змінену схему даних й відобразіть результат виконання запиту. Проаналізуйте отриманий результат.

(В) Повторіть виконання пункту (Б), але виберіть для зв'язку третій спосіб об'єднання. Представте у протоколі змінену схему даних й відобразіть результат виконання запиту. Проаналізуйте отриманий результат.

2.2. Параметри об'єднання в інструкціях SQL. Відкрийте у режимі конструктора створений запит «Прізвище-Запит09», де для вихідних таблиць обрано другий спосіб об'єднання. Для переходу в режим SQL виберіть у групі «Результати» вкладки «Конструктор» команду «Режим SQL».

Інструкція SQL для цього запиту матиме вигляд:

```
SELECT ПІБ-Студент. pibcode, ПІБ-Студент. pibstudent,  
ПІБ-Студент. pibsurname, ПІБ-Успішність. pibdiscode,  
ПІБ-Успішність. pibtab, ПІБ-Успішність. pibrating  
FROM ПІБ-Студент LEFT JOIN ПІБ-Успішність ON  
(ПІБ-Студент.[ pibstudent] = ПІБ-Успішність.[ pibstudent])  
AND (ПІБ-Студент.[ pibcode] = ПІБ-Успішність.[ pibcode]);
```

Складник FROM цієї інструкції SELECT визначає не тільки таблиці, які є джерелом даних для цього запиту, але також способи об'єднання LEFT JOIN і поля, за якими зв'язуються таблиці. Операцію LEFT JOIN використано для створення лівого зовнішнього об'єднання. Ліве зовнішнє

об'єднання охоплює усі записи з першої (лівої) таблиці, навіть якщо немає співпадаючих значень для записів у другій (правій) таблиці.

Поверніться в режим конструктора і змініть спосіб об'єднання таблиць запиту, вибравши замість другого параметру об'єднання третій. Інструкція SQL для такого запиту матиме вигляд:

```
SELECT ПІБ-Студент. pibcode, ПІБ-Студент. pibstudent,  
        ПІБ-Студент. pibsurname, ПІБ-Успішність. pibdiscode,  
        ПІБ-Успішність. pibtab, ПІБ-Успішність. pibrating  
FROM ПІБ-Студент RIGHT JOIN ПІБ-Успішність ON  
        (ПІБ-Студент.[ pibstudent] = ПІБ-Успішність.[ pibstudent])  
        AND (ПІБ-Студент.[ pibcode] = ПІБ-Успішність.[ pibcode]);
```

Операцію RIGHT JOIN використовують для створення правого зовнішнього об'єднання, яке охоплює всі записи з другої (правої) таблиці, навіть якщо немає співпадаючих значень із записами з першої (лівої) таблиці. Якщо переглянути запит, в якому обрано першого спосіб об'єднання таблиць, то у складнику FROM буде зазначено відповідну операцію INNER JOIN.

Представте у протоколі усі три варіанти інструкції SQL.

2.3. Конструювання запиту на основі кількох взаємозв'язаних таблиць.

Розглянемо технологію конструювання багато-табличних запитів на вибірку на прикладі отримання інформації про успішність студентів з бази даних «Навчальний процес». Нехай необхідно отримати інформацію про оцінки, отримані студентами з усіх дисциплін. Результат повинен містити прізвище студента, назви дисциплін і оцінки.

Натисніть команду «Конструктор запитів» групи «Запити» вкладки «Створити». Додайте таблицю «ПІБ-Студент» для вибірки прізвища з поля pibsurname, таблицю «ПІБ-Успішність» для визначення кодів дисциплін з поля pibdiscode, за якими студент здавав екзамени, й оцінок з дисципліни з поля pibrating та таблицю «Дисципліна» для вибірки назв дисциплін з поля pibdistitle. Представте у протоколі отриману у вікні конструктора запитів схему даних. Проаналізуйте види встановлених зв'язків. Упевніться, що поля pibdiscode таблиць «Дисципліна» та «Успішність» пов'язані. Збережіть запит з іменем «Прізвище-Запит10»

У вікні конструктора перетягніть зі списку полів таблиці «ПІБ-Студент» поле pibsurname, зі списку полів таблиці «ПІБ-Дисципліна» поле pibdistitle й зі списку полів таблиці «ПІБ-Успішність» поле pibrating у стовпці бланка запиту в рядок «Поле». Необхідно отримати інформацію про двох конкретних студентів, введіть певні значення в рядок «Умова відбору». Записуйте прізвища студентів у поле pibsurname у різних рядках бланка запиту, оскільки необхідно вибрати записи з різними значеннями. Якщо у базі даних прізвища студентів записано з ініціалами, то використовуйте шаблон у рядку «Умова відбору», наприклад, «Прізвище*». Після введення прізвища з символом шаблону (*) СКБД додає оператор Like, що визначає пошук за

зразком. Представте у протоколі заповнений бланк запиту та результат його виконання.

2.4. Введення параметрів до запиту. У запиті «*Прізвище-Запит10*» для вказування прізвища студента необхідно щоразу коригувати бланк запиту. Щоб уникнути цього, доцільно використовувати параметри в запиті, тобто перед виконанням запиту діалогове вікно MS Access буде запитувати у користувача конкретні значення параметрів і використовувати їх як умови відбору.

Нехай необхідна інформація щодо оцінки, отриманої студентом за певну дисципліну. Відкрийте запит «*Прізвище-Запит10*» і збережіть як «*Прізвище-Запит11*». Введіть в умову відбору поля **pibsurname** замість конкретного прізвища назву параметра, згідно з яким користувач вказуватиме прізвище під час виконання запиту. Назву параметру введіть як текст у квадратних дужках, наприклад, [Прізвище та ініціали студента]. Цей текст СКБД сприймає як ім'я параметра. До умови відбору поля **pibdistitle** введіть другий параметр запиту: [Назва предмету].

Виконайте запит. Уведіть потрібні значення параметрів у діалогових вікнах. Якщо вказаний користувачем студент є в списку студентів і відмітка про успішне здавання екзамену з дисципліни є в таблиці успішності, то в таблиці результатів запиту буде відображено запис про це. Представте у протоколі заповнений бланк запиту та результат його виконання.

Перегляньте запит в режимі SQL. Запишіть у протокол інструкцію **SELECT** цього запиту. Вкажіть у якому виразі у якості умов відбору використано параметр.

2.5. Використання імен полів різних таблиць в умові відбору. Нехай необхідно вибрати записи з таблиці «Навчання», у яких години практичних занять з інформатики не відповідають рівномірному розподілу за семестрами всіх годин практики. Для вирішення цього завдання необхідно використати таблицю «Навчання», в якій містяться відомості про планові заняття у групах у поточному семестрі, в тому числі про тривалість (поле **pibhours**) кожного виду заняття (поле **pibtype**), й таблицю «Дисципліна», в якій містяться відомості про назву дисципліни (поле **pibdistitle**), загальну тривалість вивчення дисципліни (поле **pibdisnumber**), обсяг годин практики (**pibpractice**) і кількість семестрів вивчення дисципліни (**pibsemester**). Підготуйте запит «*Прізвище-Запит11*» у режимі конструктора, який використовує вказані таблиці і поля.

Для відбору з таблиці «*ПІБ-Навчання*» записів про практичні заняття з певної дисципліни, наприклад, інформатики, у рядку Умова відбору для поля **pibdistitle** таблиці «*ПІБ-Дисципліна*» вкажіть значення "Інформатика", а для поля **pibtype** таблиці «*ПІБ-Навчання*» - значення "пр." (практичне заняття).

За умови рівномірного розподілу практики за семестрами загальне число годин практичних занять з предмета (**pibpractice**) має дорівнювати виробленню годин практики (**pibhours**) з таблиці «Навчання» помножене на число семестрів (**pibsemester**) з таблиці «Дисципліна». Щоб охопити тільки

ті записи, для яких число годин не відповідає цьому твердженню, запишіть у рядку Умова відбору поля **pibpractice** таблиці «**ПІБ-Дисципліна**» вираз:

<>[ПІБ-Навчання.hours]*[pibsemester]

Вказівка таблиці «**ПІБ-Навчання**» для поля **pibhours** є обов'язковим, тому що поле з таким же ім'ям є і в таблиці «**ПІБ-Дисципліна**». Якщо результат виконання запиту не містить записів, це означає, що для вказаної дисципліни в кожній із студентських груп години практичних занять відповідають рівномірному розподілу за семестрами всіх годин практики.

2.6. Групові операції в багато-табличному запиті. Нехай необхідно виконати аналіз отриманих студентами оцінок з різних дисциплін. Для формування відомостей створіть багатотабличний запит на вибірку на основі таблиць «**ПІБ-Студент**», «**ПІБ-Успішність**» та «**ПІБ-Дисципліна**». До бланку запиту додайте поля **pibdistitle**, **pibrating** та **pibsurname**.

Основу створення записів цього запиту становитиме таблиця успішності, а підлеглі дані обиратимуться з таблиць «**ПІБ-Студент**» та «**ПІБ-Дисципліна**». Таким чином, кожен рядок результату міститиме інформацію про одну оцінку, отриману студентом за вказаним в рядку предметом. Кількість рядків у таблиці запиту дорівнюватиме числу рядків у таблиці успішності.

Для групування записів з однаковою назвою дисципліни і оцінкою, а також підрахунку кількості записів у групі натисніть команду «Групові операції» у групі «Показати / Приховати» вкладки «Конструктор» і у відповідному рядку в полях **pibdistitle** та **pibrating** оберіть групування (англ. Group by), а в полі **pibsurname** оберіть підрахування (англ. count). Якщо у результаті виконання запиту в полі **Countofpibsurname** усі записи містять значення «1», змініть у таблиці «**ПІБ-Успішність**» значення кількох записів так, щоб кілька студентів мали однакові оцінки за одну й ту саму дисципліну. Повторіть виконання запиту. Представте у протоколі заповнений бланк запиту та результат його виконання. Збережіть запит з іменем «**Прізвище-Запит12**».

Перегляньте запит в режимі SQL. Запишіть у протокол інструкцію **SELECT** цього запиту.

2.7. Вирішення завдання, що вимагає виконання кількох запитів. Нехай необхідно визначити середнє навантаження викладача кафедри в поточному семестрі. Для цього потрібно підрахувати число викладачів кафедри, потім загальну кількість годин занять, що забезпечуються кафедрою, і завершити вирішення завдання розрахунком середнього навантаження викладача.

Створіть запит на вибірку, в якому за таблицею «**ПІБ-Викладач**» підраховується число викладачів на кожній кафедрі. Збережіть запит під ім'ям «**Прізвище-Запит131**».

Для підрахунку загального числа годин занять, що забезпечуються кожною кафедрою, створіть запит на вибірку на базі таблиць «**ПІБ-Викладач**» і «**ПІБ-Навчання**». Таблиця «**ПІБ-Навчання**» потрібна для підрахунку числа

годин занять, що проводяться викладачами кожної кафедри. Оскільки ця таблиця не містить коду кафедри, до якої відноситься той чи інший викладач, необхідно додати до запиту таблицю «*ПІБ-Викладач*», що визначає на якій кафедрі працює викладач. Записи цього запиту формуються на основі записів таблиці «Навчання», причому число записів до проведення групування дорівнює числу записів у цій таблиці. У разі виконання запиту до кожного запису додається код кафедри *pibdept*, за яким і проводиться групування. Кількість годин кафедри підраховується за допомогою функції *pibsum* за полем *pibhours*. Число записів у таблиці результату запиту дорівнює числу кафедр, для яких є рядки в таблиці «*ПІБ-Навчання*». Збережіть запит під ім'ям «*Прізвище-Запит132*».

Для остаточного вирішення задачі розрахунку середнього навантаження викладача кафедри підготуйте на базі двох попередніх запитів третій запит на вибірку з обчислюваним полем. Для додавання у схему нового запиту запитів «*Прізвище-Запит131*» та «*Прізвище-Запит132*» під час конструювання запиту перейдіть на вкладку «Запити» діалогового вікна «Таблиці», яке з'являється після натискання команди «Конструктор запитів» вкладки «Створити». Варто пересвідчитись, що запит «*Прізвище-Запит131*» у схемі нового запиту містить поле «*Countofpibtab*», а запит «*Прізвище-Запит131*» - поле «*Sumofpibhours*». Для створення обчислюваного поля, який розраховує середнє навантаження викладача, в рядок Поле порожнього стовпця введіть вираз $[Sumofpibhours] / [Countofpibtab]$. Таблицю «*ПІБ-Кафедра*» додайте до запиту для доповнення коду кафедри назвою. Для зміни в таблиці результату напису Вираз 1 (англ. *expr1*), що формується за замовчуванням для обчислюваного поля, та його формату (для поля результату з округленням до цілого), додайте в напис поля (англ. *caption*) значення «Середнє навантаження викладача», у формат поля – фіксований і в число десяткових знаків – 0. Збережіть запит під ім'ям «*Прізвище-Запит133*» (рис. 21). Представте у протоколі заповнені бланки запитів та результати їх виконання.

Назва кафедри	Середнє навантаження викладача
Інформатики	152
Математики	43
Історії	17
Іноземної мови	50
Фізкультури	67
Філософії	25

Рисунок 21 – Результат виконання запиту «*Прізвище-Запит133*»

Немає необхідності попередньо виконувати перший і другий запити, на основі яких виконується третій запит. Виконання вкладеного запиту ініціюється системою під час виконання побудованого на ньому запиту.

Створіть новий запит, додайте до схеми запиту відповідні таблиці та поля, вкажіть необхідні підрахунки в рядку групових операцій. Додайте обчислюване підсумкове поле $[Sumofpibhours] / [Countofpibtab]$. Оберіть у рядку Групова

операція обчислюваного поля значення «Вираз» (англ. expression). Виконайте запит. Збережіть запит під ім'ям «*Прізвище-Запит134*». Представте у протоколі заповнений бланк запиту та результати його виконання.

Перегляньте запит в режимі SQL. Запишіть у протокол інструкцію SELECT запиту «*Прізвище-Запит134*».

3. Запити на змінювання

3.1. Конструювання запиту створення таблиці. Створіть запит підрахунку кількості студентів у кожній групі на основі таблиці «*ПІБ-Студент*» та її полів **pibcode** та **pibstudent**. Збережіть запит як «*Прізвище-Запит14*».

У вікні бази даних перейдіть у вказаний запит і оберіть вкладку «Конструктор» стрічки. Перетворіть цей запит на вибірку в запит на створення таблиці. Оберіть команду «Створити таблицю» (англ. make table). У вікні «Створити таблицю» уведіть ім'я створюваної таблиці – «*ПІБ-Кількість студентів*». Щоб переглянути, які записи буде представлено у новій таблиці, натисніть команду «Вигляд». Виконайте запит, щоб таблицю «*ПІБ-Кількість студентів*» було збережено в базі даних. Тепер цю таблицю можна побачити у списку таблиць вікна бази даних. Відобразити у протоколі результат виконання запиту.

Перейдіть в режим SQL. Еквівалентну запиту на створення таблиці інструкцію SELECT ... INTO запишіть у протокол.

3.2. Конструювання запиту на оновлення. Розглянемо технологію створення запиту на оновлення на прикладі оновлення поля **pibnumber** (кількість студентів групи в таблиці «*ПІБ-Група*»). Кількість студентів у групах раніше було підраховано за допомогою статистичної функції count в запиті на вибірку «*Прізвище-Запит14*». Запит на оновлення безпосередньо на такому запиті побудувати не можна. Тому використовуйте для оновлення не сам запит, а таблицю «*ПІБ-Кількість студентів*», отриману за запитом на створення таблиці в п. 3.1.

Для формування запиту на оновлення спочатку створіть запит на вибірку на основі двох таблиць: оновлюваної таблиці «*ПІБ-Група*» і таблиці «*ПІБ-Кількість студентів*», що містить дані для оновлення. У схемі даних запиту автоматично встановиться зв'язок цих таблиць за полем **pibcode**. Перетворіть запит на вибірку в запит на оновлення. Оберіть команду «Оновлення» (англ. update) на вкладці «Конструктор». Після виконання цієї команди в бланку запиту з'явиться рядок «Оновлення».

Заповніть бланк запиту. Додайте оновлюване поле **pibnumber** зі списку таблиці «*ПІБ-Група*» в рядок «Поле». У рядок «Оновлення» (англ. update to) введіть ім'я поля CountOf**pibstudent** таблиці «*ПІБ-Кількість студентів*», з якого отримуватимуть значення для оновлення. Ім'я поля вкажіть у квадратних дужках.

Натисніть команду «Вигляд» перед виконанням запиту для перегляду. Виконайте запит. Представте у протоколі заповнений бланк запиту та результат його виконання. Перегляньте оновлюване поле **pibnumber** в

таблиці «**ПІБ-Група**» після виконання запиту. Збережіть запит під ім'ям «**Прізвище-Запит15**».

3.3. Оновлення поля таблиці обчислюваним значенням. Розглянемо формування запиту на оновлення з використанням виразу на прикладі заповнення поля **pibhours** для лабораторних занять таблиці «**ПІБ-Навчання**». Нехай оновлення поля **pibhours** має відбуватись за допомогою даних, обчислюваних на основі полів **piblab** та **pibsemester** з таблиці «**ПІБ-Дисципліна**». Розрахункову кількість годин полів визначимо за формулою $\text{piblab} / \text{pibsemester}$. Відповідно до завдання у записях лабораторних занять таблиці «**ПІБ-Навчання**» необхідно оновити поле **pibhours** розрахунковим числом годин. Вид занять можна визначити за значенням поля **pibtype** таблиці «**ПІБ-Навчання**». Дані для розрахунку годин містяться в таблиці «**ПІБ-Дисципліна**». Таким чином, запит повинен будуватись на основі таблиць «**ПІБ-Навчання**» і «**ПІБ-Дисципліна**».

Створіть запит на вибірку на основі таблиць «**ПІБ-Навчання**» і «**ПІБ-Дисципліна**». Перетворіть його в запит на оновлення, натиснувши відповідну команду. Додайте в бланк запиту оновлюване поле **pibhours** таблиці «**ПІБ-Навчання**». У рядку оновлення для цього поля введіть вираз $[\text{piblab}] / [\text{pibsemester}]$. Імена полів обов'язково вкажіть у квадратних дужках. Для відбору в таблиці «**ПІБ-Навчання**» оновлюваних записів про лабораторні заняття в бланк запиту додайте поле **pibtype** і вкажіть у рядку умови відбору значення «лаб.». Перегляньте вміст поля **pibhours** таблиці «**ПІБ-Навчання**» до оновлення. Виконайте запит. У діалоговому вікні з'явиться повідомлення про число оновлюваних записів. Перегляньте вміст поля **hours** після оновлення. Представте у протоколі заповнений бланк запиту та результат його виконання. Збережіть запит під ім'ям «**Прізвище-Запит16**».

3.4. Створення запиту на додавання. Нехай в деякому підрозділі створюється таблиця про успішність студентів «**ПІБ-Успішність**» і необхідно її доповнювати даними з таблиці успішності, що зберігається в централізованій базі даних. Структура централізованої таблиці успішності аналогічна структурі внутрішньої таблиці «**ПІБ-Успішність**», але має деякі додаткові поля. Створіть запит на вибірку на основі таблиці «**ПІБ-Успішність**». Перетворіть його на запит на додавання командою «Додати» (англ. *append*) з вкладки «Конструктор». У діалоговому вікні «Додавання» у полі імені таблиці вкажіть таблицю «**ПІБ-Успішність**», в яку потрібно додати записи.

Якщо таблицю «**ПІБ-Успішність**» розміщено в одній відкритій базі даних з таблицею-приймачем додаваних записів, відзначте «в поточній базі даних» (англ. *current database*). Для таблиці, що знаходиться в іншій базі даних, треба відзначити «в іншій базі даних» (англ. *another database*) і ввести ім'я файлу бази даних.

Після перетворення запиту на запит на додавання в його бланку з'являється рядок Додавання (англ. *append to*). Для формування нових записів потрібно додати до бланку запиту поля-складові ключа таблиці, в яку буде

проводитися додавання, і поле **pibrating**. Оскільки в цьому завданні імена полів обох таблиць співпадають, MS Access автоматично сформує в рядку додавання імена полів таблиці-одержувача, в які буде додано значення з відповідних полів таблиці запиту. Якщо б ці імена не співпадали, для вказівки імен полів таблиці-одержувача, відповідних полям таблиці запиту, можна відкрити в кожному полі рядку додавання список полів і обрати потрібне ім'я.

Для попереднього перегляду записів, які буде додано в таблицю, натисніть команду «Вигляд». Для додавання записів виконайте запит.

Якщо таблиця, в яку додають записи, містить ключове поле, додавані записи повинні містити такі ж поля. Ті записи, додавання яких призведе до появи **співпадінь** чи порожніх значень у ключовому полі, не буде додано. Записи не додаються і у випадку, якщо неможливо перетворення типу даних у додаваних полях або не виконуються обмеження щодо значення. Якщо в схемі даних встановлено параметр забезпечення цілісності даних, то додавання підлеглих записів можливо тільки в тому випадку, коли в головних таблицях вже є пов'язані записи.

Збережіть запит під ім'ям «**Прізвище-Запит17**». Представте у протоколі заповнений бланк запиту та результат його виконання. Перейдіть в режим SQL. Еквівалентну запиту інструкцію **INSERT ... INTO** запишіть у протокол.

3.5. Створення запиту на видалення. Розглянемо формування запиту на видалення записів про групи, де кількість студентів менше певного значення. Таблиці «**ПІБ-Група**» і «**ПІБ-Студент**» перебувають у відношенні типу один-до-багатьох і між ними встановлено зв'язок за полем **pibcode** і параметр забезпечення цілісності даних, але не встановлено параметр каскадного видалення пов'язаних записів. Для зв'язку таблиць «**ПІБ-Студент**» і «**ПІБ-Успішність**» за полями **pibcode** і **pibstudent** параметр каскадного видалення пов'язаних записів встановлено.

Якщо параметр каскадного видалення пов'язаних записів не встановлено, видалення записів обох таблиць за допомогою одного запиту неможливо. Якщо параметр каскадного видалення пов'язаних записів встановлено, досить одного запиту на видалення записів головної таблиці.

Для того щоб перевірити встановлені між таблицями у схемі даних зв'язки та її параметри, відкрийте схему даних. За необхідності приведіть їх у відповідність з умовами, сформульованими у завданні.

Створіть запит на вибірку, в схему даних якого додайте таблиці «**ПІБ-Група**» і «**ПІБ-Студент**». У режимі конструктора перетворіть запит на запит на видалення командою «Видалити» (англ. delete) з групи «Тип запиту» вкладки «Конструктор». Після перетворення запиту в запит на видалення в його бланку з'являється рядок «Видалення». Додайте символ зірочки (*) зі списку полів таблиці «**ПІБ-Студент**» до бланку запиту. У рядку видалення в стовпці цього поля з'являється значення «3» (англ. from).

Якщо в запиті об'єднано дві або більше таблиць, що у відношенні 1:M, то видалити можна тільки записи однієї з таблиць на стороні багатьох.

Щоб задати умови відбору видалення записів додайте поле number з таблиці «**ПІБ-Група**». У рядку видалення під ім'ям цього поля з'явиться значення «Умова» (англ. where)/ Заповніть рядок умови відбору для цього поля умовою <значення. Для попереднього перегляду записів, які буде видалено, натисніть команду «Вигляд». Для видалення записів виконайте запит. Збережіть запит під ім'ям «**Прізвище-Запит181**». Представте у протоколі заповнений бланк запиту та результат його виконання.

Після видалення записів підпорядкованої таблиці видалимо записи головної таблиці. Зробіть копію запиту «**Прізвище-Запит18**». Модифікуйте запит. Видаліть зі схеми запиту таблицю «**ПІБ-Студент**». Для видалення записів виконайте запит. Представте у протоколі заповнений бланк запиту та результат його виконання. Збережіть запит під ім'ям «**Прізвище-Запит182**».

Результати роботи запиту на видалення залежать від встановлених в базі даних відносин між таблицями і параметрів цілісності. Так як задано параметр забезпечення цілісності даних і параметр каскадного видалення пов'язаних записів не встановлено, для видалення записів груп і пов'язаних з ними записів про студентів необхідно в запиті спочатку видалити записи підпорядкованої таблиці «**ПІБ-Студент**» і тільки після цього головної таблиці «**ПІБ-Група**». Якби параметр каскадного видалення було задано, то достатньо було б запису головної таблиці «Група», а пов'язані записи підлеглої таблиці «**ПІБ-Студент**» було б видалено автоматично. Якщо встановлено саме цей параметр, то у разі видалення записів таблиці «Студент» буде видалено і всі записи підпорядкованої таблиці успішності.

Перейдіть в режим SQL. Еквівалентну запиту інструкцію DELETE ... FROM запишіть у протокол.

3.6. Створення перехресної таблиці. Розглянемо створення перехресної таблиці на прикладі таблиці успішності. Нехай необхідно для кожного викладача визначити число студентів, знання яких за типом заняття оцінено. Скористайтесь майстром запитів. У вікні «Новий запит» оберіть «Перехресний запит» (англ. Crosstab Query Wizard). У вікні майстра створення перехресних запитів виберіть таблицю успішності, поля якої буде виведено у перехресному запиті. У наступному вікні майстра виберіть для заголовків рядків поле **pibtab**. Для заголовків стовпців виберіть **pibtype**. Для підрахунку кількості студентів, які екзаменувались у певного викладача за певним видом заняття, і розміщення цього значення у клітинці на перетині рядків і стовпців таблиці оберіть поле **pibstudent** і функцію Count. Щоб підрахувати загальну кількість студентів (підсумкове значення для кожного рядка), знання яких оцінював викладач, встановіть прапорець «Обчислити підсумкові значення для кожного рядка?». Збережіть запит під ім'ям «**Прізвище-Запит19**». Представте у протоколі результат його виконання.

3.7. Створення перехресної таблиці на основі даних з кількох таблиць. Очевидно, в результаті виконання перехресного запиту, створеного в

попередній вправі, зручніше бачити в таблиці не табельні номери викладачів, а їх прізвища. Прізвища можна отримати з таблиці «*ПІБ-Викладач*», яка перебуває у відношенні 1:М з таблицею успішності та пов'язана з нею за полем *pibtab*. Оскільки майстер не дає змогу будувати перехресний запит на полях кількох таблиць, перед тим як створювати такий запит, потрібно створити запит на вибірку для таблиць «*ПІБ-Успішність*» і «*ПІБ-Викладач*», в якому обрати необхідні поля.

Створіть за допомогою майстра простий запит. У цьому запиті виберіть з таблиці «*ПІБ-Викладач*» поле *pibsurname*, а з таблиці успішності - поля *pibtype* і *pibstudent*. Перевірте результат виконання запиту. У режимі конструктора варто упевнитись, що схема цього запиту містить саме таблиці «*ПІБ-Викладач*» і «*ПІБ-Студент*» пов'язані за полем *pibtab*. Збережіть запит під ім'ям «*Прізвище-Запит201*».

Створіть перехресний запит, джерелом даних для якого буде збережений запит на вибірку «*Прізвище-Запит201*». Виберіть *pibsurname* в якості поля для заголовків рядків. Для заголовків стовпців виберіть *pibtype*. Для обчислення значень клітинок виберіть *pibstudent* і функцію Count. Збережіть запит під ім'ям «*Прізвище-Запит202*».

Представте у протоколі заповнені бланки запитів та результати їх виконання.

Створіть за допомогою майстра простий запит. У цьому запиті виберіть з таблиці «*ПІБ-Викладач*» поле *pibsurname*, а з таблиці успішності - поля *pibtype* і *pibrating*. Перевірте результат виконання запиту. Збережіть запит під ім'ям «*Прізвище-Запит203*».

Створіть перехресний запит, джерелом даних для якого буде збережений запит на вибірку «*Прізвище-Запит203*». Виберіть *pibsurname* в якості поля для заголовків рядків. Для заголовків стовпців виберіть *pibtype*. Для обчислення значень клітинок виберіть *pibrating* і функцію Avg для розрахунку середньої оцінки, яку виставив викладач. Збережіть запит під ім'ям «*Прізвище-Запит204*».

Представте у протоколі заповнені бланки запитів та результати їх виконання.

3.8. Створення перехресної таблиці режимі конструктора. Розглянемо, як створити перехресну таблицю на основі однієї таблиці «*ПІБ-Навчання*», в якій заголовками рядків будуть табельний номер викладача (*pibtab*), стовпці – коди дисциплін (*pibdiscode*), а на перетині рядків і стовпців - сума годин за полем *pibhours*.

Почніть конструювання запиту додаванням таблиці «*ПІБ-Навчання*» в схему даних запиту. У бланк запиту додайте поля *pibtab* й *pibdiscode*, за якими буде здійснюватися групування, і які будуть використовуватися в рядках і стовпцях перехресної таблиці відповідно, і поле *pibhours*, за яким буде підраховуватися сумарне число годин для отриманих груп. У вікні конструктора перетворіть запит на вибірку в перехресний запит за допомогою команди «Перехресний» групи «Тип запиту» вкладки «Конструктор». У бланку запиту з'явиться два додаткові рядки («Групова

операція» і «Перехресна таблиця»). В рядку «Перехресна таблиця» для поля **pibtab** виберіть зі списку значення «Заголовки рядків», для поля **pibdiscode** – «Заголовки стовпців», для поля **pibhours** – «Значення». В рядку «Групова операція» для поля **pibhours** виберіть зі списку значення «Sum», залишивши в інших рядках значення «Групування». Для додавання в перехресну таблицю підсумкових значень для кожного рядка доповніть бланк запиту ще одним полем **pibhours**, замініть Значення «Групування» на «Sum», а в рядку «Перехресна таблиця» виберіть «Заголовки рядків». Виконайте запит. Представте у протоколі заповнений бланк запиту та результат його виконання.

Для більшої інформативності отриманої перехресної таблиці замініть в ній табельні номери прізвищами викладачів, а коди предметів їх найменуванням. До поля, яке містить результат підсумовування за рядками, додайте напис «Разом, годин» за допомогою вікна властивостей. Поверніться в режим конструктора. Поле з прізвищем викладача розміщено в таблиці «**ПІБ-Викладач**», а назви дисциплін у таблиці «**ПІБ-Дисципліна**», тому додайте в схему запиту ці таблиці. Замініть у бланку запиту поле **pibdiscode** на поле **pibdistitle** таблиці «Дисципліна», поле **pibtab** на поле **pibsurname** таблиці «**ПІБ-Викладач**». Внесіть потрібні зміни у назву поля, яке містить суму за рядками. Виконайте запит. Представте у протоколі заповнений бланк запиту та результат його виконання. Збережіть запит під ім'ям «**Прізвище-Запит21**».

Перейдіть в режим SQL. Еквівалентну запиту інструкцію TRANSFORM запишіть у протокол.

Оформити протокол лабораторної роботи.

Оформлення протоколу

Після оформлення протоколу лабораторної роботи до заздалегідь підготовлених складників додаються роздруківки скріншотів ходу виконання завдань.

Контрольні запитання

1. Яка команда дає змогу додати необхідні таблиці в запит?
2. Як у запиті відобразити потрібні поля таблиці?
3. Яка логічна операція застосовується до умов відбору, записаних для різних полів у одному рядку?
4. Які оператори порівняння та логічні оператори можна використати для запису умови відбору?
5. Чи можна використовувати у виразі умови відбору ім'я поля?
6. Для яких значень виконуються операції у виразі обчислюваного поля?
7. З яких записів утворюється група у разі використання групової операції?

8. Яке значення має бути обране в рядку Групова операція (англ. total) для поля, за яким задається умова відбору?
9. У якому стовпці бланка запиту записується умова відбору груп?
10. Як переглянути для запиту, створеного в режимі конструктора, еквівалентну інструкцію SQL?
11. Яка інструкція SQL відповідає запиту на вибірку?
12. Після якого ключового слова в компоненті FROM вказують поля зв'язку таблиць?
13. Який спосіб об'єднання записів за умовчанням встановлюється для зв'язку таблиць?
14. Які поля представлено в списку, що відкривається в рядку «Поле» бланку запиту?
15. Яким компонентом SQL визначається необхідність у сортуванні?
16. Яка інструкція мови SQL відповідає запиту на оновлення записів?
17. Яка інструкція мови SQL відповідає запиту на створення таблиці?
18. Яка інструкція мови SQL відповідає запиту на додавання записів?
19. Яка інструкція мови SQL відповідає запиту на видалення записів?
20. Як за допомогою майстра побудувати багатотабличний перехресний запит?

Лабораторна робота № 5

Створення та налагодження форм в СКБД MS Access

Мета роботи. Набути умінь використання прийомів роботи з формами у середовищі MS Access 2016.

Завдання. Розглянути засоби створення і редагування форм у СКБД MS Access 2016.

Короткі теоретичні відомості

Форми – це основа розробляння діалогових додатків користувача для роботи з базою даних. Форми забезпечують роботу з даними однієї або кількох взаємопов'язаних таблиць. Дані з таблиць виводять на екран за допомогою макету форми. Користувач може додавати, видаляти, змінювати записи таблиць, отримувати результати обчислень. У процесі роботи з даними можливий контроль даних, перевірка обмеження доступу до даних, виведення необхідних додаткових даних. Форма містить елементи керування, які відображають поля таблиць і графічні елементи, не пов'язані з полями таблиць. Елементи керування призначено для розробки макету форми, а саме розміщення полів таблиць і запитів, написів, впроваджуваних об'єктів, обчислюваних полів, кнопок, тощо.

Форма у цілому та її елементи мають багато властивостей, змінювання яких налаштовує зовнішній вигляд, розмір, положення елементів у формі, визначає джерело даних форми, режим введення/виведення, зв'язує елементи

і вирази, макроси або програми. Набір властивостей доступний у відповідному вікні, де їх розбито на категорії (вкладки). Основні вкладки у вікні властивостей:

- **макет** – властивості зовнішнього вигляду форми та її елементів;
- **дані** – властивості, які визначають джерело даних форми та її елементів, режим використання форми (введення, дозвіл на змінювання, додавання, видалення, тощо);
- **події**, які виникають у роботі з конкретним об'єктом або елементом.

Для швидкого створення форм призначено майстри. Формування макету форми за вимогами, додавання процедур оброблення подій забезпечено засобами конструювання. Конструктор форм можна використовувати як для створення нової форми, так і для редагування форми, створеної майстром. Для налаштування зовнішнього вигляду форми MS Access надає режим макету.



Під час створення форми визначають джерело даних форми (таблиці, запити). З джерел даних обирають поля, які необхідно відобразити у формі, створюють обчислювані поля. Для оформлення форми створюють графічні елементи: кнопки, перемикачі, елементи, пояснювальний текст і рисунки. Для налаштування різних елементів форми використовують їхні властивості.

Форми можна відобразити у чотирьох режимах:

- режим **форми** призначено для введення, перегляду і корегування даних таблиць, на яких базується форма;
- режим **макету** забезпечує перегляд даних майже у тому ж вигляді, у якому ті відображено у режимі форми, але дає змогу змінити зовнішній вигляд форми;
- режим **конструктора** призначено для розробки форми за допомогою набору інструментів, який забезпечує детальніше опрацювання структури форми, використання усіх елементів керування;
- режим **таблиці** забезпечує перегляд даних, відібраних для форми з джерела даних у табличному вигляді.

Однотабличну форму призначено для завантаження, переглядання і модифікування даних однієї таблиці. Створити однотабличну форму можна за допомогою команди «Майстер форм» групи «Форми» вкладки «Створити». Якщо у базі даних створено запит, у якому обрано таблиці, необхідні для отримання результату, визначено поля, які необхідно представити у результаті, параметри, виконано обчислення, то доцільно створювати форму на основі такого запиту. Вибір запиту з параметром як джерела записів дає змогу створити форму, яка забезпечує діалог з користувачем і виведення тих записів, які відповідають вказаним у діалозі умовам селекції. Розділена форма дає змогу одночасно відображати дані одного джерела у двох представленнях: у режимі форми та у режимі таблиці. У будь-якій частині форми можна виконувати змінювання даних, які одразу відображаються в іншому представленні.

Форму на основі однієї таблиці може бути побудовано як самостійну для завантаження, перегляду і коригування таблиці, а також як допоміжну для додавання у якусь форму. Для створення форми може бути використано майстри Access. Точне формування макета форми відповідно до вимог, наданих користувачем в процесі її проектування, забезпечується засобами конструювання форм. Майстер форм може створювати форму для однієї таблиці або для кількох взаємозалежних таблиць. Майстер дає змогу користувачеві задати, які поля додати у форму, а також вибрати стиль її оформлення.

Команда «Форма» дає змогу створити форму, яка одночасно відображає лише один запис. Поля запису відображено всередині одного розділу в стовпчик з написами ліворуч. Команда «Кілька елементів»  **Multiple Items** створює форму, яка відображає одразу кілька записів і усі поля у такій формі розміщено в одному рядку. Команда «Розділена форма»  **Split Form** дає змогу отримати форму у двох відображеннях, в одному розділі представлено таблицю, а в другому – виділений запис таблиці.

Режим конструктора форм можна використовувати як для створення нової форми, так і для редагування вже створеної форми, у тому числі за допомогою майстра. Форма в режимі конструктора має три області: область даних, заголовок форми і примітку форми. Графічні об'єкти, пов'язані з записами таблиць і призначені для відображення даних деякого поля, називаються елементами керування. Основними типами елементів керування є поле, поле зі списком і список. Області форми наповнюються різними графічними елементами. Тип елемента керування, що обирається для поля, за замовчуванням визначається у властивостях поля таблиці бази даних, з яким пов'язано поля форми. Графічні об'єкти, не пов'язані з таблицями або запитамі, призначено для створення макета форми і містять написи полів, впроваджені об'єкти, їх надписи, заголовки. Інформація про ці елементи зберігається в макеті форми. У формі може бути передбачено кнопки керування для різних цілей, наприклад: переходу до інших записів таблиці; роботи із записами (додати, дублювати, відновити, видалити, зберегти); роботи з формою (відкрити, закрити, фільтрувати, оновити). Як форма в цілому, так і кожен з її елементів має свої властивості, які можна переглядати і змінювати. Властивості елемента дають змогу визначити його зовнішній вигляд, розмір, місце розташування у формі, режим вводу / виводу, прив'язати до елемента вираз, макрос чи програму.

Конструювання нової форми засобами конструктора форм починається з порожньої форми і повністю покладається на користувача. У разі конструювання форми можна додавати нові полів і написи, поля зі списками, створювати кнопки, впроваджувати об'єкти з інших додатків, наприклад, рисунки, діаграми тощо, змінювати розташування окремих полів, їх відображення і написи. Всі елементи, що додаються до форми, є елементами керування. Прикладами різних елементів керування слугують поля, написи,

списки, перемикачі, кнопки і лінії. Спосіб створення елемента керування залежить від того, який елемент створюється: приєднувальний, вільний або обчислюваний. У разі створення нової одно-табличної форми визначається таблиця бази даних, на основі якої створюється форма, вибираються поля таблиці, які повинні бути представлені у формі, здійснюється їх розміщення в макеті форми, створюються обчислювані поля та інші графічні елементи (кнопки, перемикачі, елементи оформлення, пояснювальний текст, рисунки). Для налаштування різних елементів форми використовується типовий набір їх властивостей. Для редагування раніше створеної форми необхідно перейти в режим конструктора форм. У цьому режимі можна виконувати всі зазначені вище дії для створення нових елементів, а також видалення елементів форми та їх змінювання. Зокрема, можна змінити текст написів, місце розташування, розмір, шрифт і інші параметри відображення елементів форми. Редагувати форму та її елементи можна графічними засобами, а також шляхом зміни їх властивостей.

Багатотабличну форму створюють для роботи з даними кількох взаємопов'язаних таблиць. Вона може бути однією, побудованою на багатотабличному запиті, або складеною, яка містить головну форму і підпорядковані форми.

Багатотабличну форму створюють для роботи з даними кількох зв'язаних таблиць. Причому форму може бути побудовано як безпосередньо на таблицях, так і на запитах. Багатотаблична форма може складатися з основної частини і однієї або кількох підлеглих форм. Підпорядковану форму, як правило, будують на базі підпорядкованої таблиці відносно таблиці-джерела записів основної частини форми. У цьому випадку при роботі з формою у підпорядкованій частині відображаються підлеглі записи щодо поточного запису головної таблиці, яка відображена в основній частині форми. За наявної схеми даних, що складається із нормалізованих таблиць, зв'язаних одно-багатозначними відносинами, можуть бути створені екранні форми, які забезпечують коректне введення взаємопов'язаних даних. Макет таких форм доцільно робити адекватним формам первинних документів-джерел даних. При цьому забезпечується комфортна робота користувача і забезпечується найважливіший аспект технології роботи з базою даних – одноразове введення даних. Такі форми дають змогу в будь-який момент проглянути вміст раніше введених документів. У процесі проектування форми доцільно виконати таку послідовність дій.

- визначення таблиць – об'єктів завантаження та відповідного документа-джерела;
- визначення схеми даних (фрагменту схеми даних) – основи для побудови форми;
- визначення загальної структури екранної форми, тобто її макета відповідно зі структурою вхідного документа і схеми даних;
- визначення складу реквізитів для кожної з частин складової форми.

Ключові поля таблиці-джерела записів основної частини складеної форми треба вводити в основну частину форми, а в підлеглий формі передбачити ключові поля таблиці-джерела записів підпорядкованої форми, яких немає в таблиці-джерелі записів основної частини форми. Після виконання перерахованих дій здійснюється конструювання екранної форми. Багатотаблична форма може бути створена в режимі конструктора або за допомогою майстра форм. Найтехнологічнішим є початкове створення форм за допомогою майстра з подальшою її доробкою в конструкторі.

Майстер форм дає можливість будувати багатотабличні форми на основі зв'язаних таблиць. Форма, побудована майстром, буде містити підпорядковану форму, якщо в якості джерела основної частини вибирається головна таблиця. У такому разі може бути обрано варіант, що передбачає явне включення підпорядкованої форми, або кнопку для виклику підпорядкованої форми. Підлегла форма відображає дані з усіх записів підпорядкованої таблиці, які пов'язані із записом головної таблиці, яка відображається в основній частині форми. За допомогою майстра можна створити складену форму, що охоплюватиме одну або декілька підлеглих форм стандартного вигляду з вибраними полями. Якщо обрано варіант з кнопкою виклику пов'язаної форми, то за необхідності перегляду підлеглих записів пов'язана підлегла форма викликається на екран натисканням цієї кнопки. Вміст зв'язаної форми синхронізовано з поточним записом форми. Цей спосіб побудови багатотабличних форм зручний у складних формах, перевантажених елементами керування, а також у разі, якщо користувачеві немає необхідності постійно бачити пов'язані дані. Складену форму, отриману за допомогою майстра, можна відредагувати в режимі конструктора, зокрема доповнити іншими формами.

Якщо обрано поля з таблиць, для яких не визначено зв'язок у схемі даних, з'явиться повідомлення про неможливість створити форму. У такому разі майстер пропонує або змінити склад полів, або виводить схему даних для можливого визначення потрібних зв'язків. Після зміни зв'язків майстер потрібно запустити ще раз.

Складена багатотаблична форма, побудована майстром, може не містити підлеглих і зв'язаних форм. Така форма створюється, якщо необхідно відображати записи підпорядкованої таблиці, доповнені полями з однієї або декількох головних таблиць. У такому випадку джерелом даних, що виводяться у формі, є запис підлеглої таблиці, але до полів цієї таблиці додаються поля з головної таблиці. Така багатотаблична форма, яка не має підлеглих форм, називається одиночною формою. В одному записі, що відображається у формі, містяться значення полів із запису підлеглої таблиці і полів з єдиним, пов'язаним з нею, записом головної таблиці. Записи в одиночній формі є результатом операцій з'єднання кожного запису підлеглої таблиці з відповідним записом головної таблиці. Так утворюється структура даних, в якій вже не виконуються вимоги третьої нормальної форми реляційної моделі, а довідкові дані повторюються.

Обчислення у формі можна виконувати як у кожному записі форми, так і для групи записів у разі формування підсумкових величин. Обчислювані величини відображаються у полі форми, але не зберігаються у таблиці. Щоб зробити обчислення на основі даних одного запису у формі, необхідно створити обчислюваний елемент керування, в який записується вираз. Для створення обчислюваного елемента керування потрібно відкрити форму в режимі конструктора і, натиснувши на панелі елементів кнопку «Поле», розмістити елемент керування на формі. Потім в елемент керування вводиться вираз. Вираз повинен починатися зі знака дорівнює. Як операнди-вирази використовуються імена полів, константи, а в якості операторів - знаки арифметичних операцій. Обчислення підсумкових значень для записів форми виконується за допомогою вбудованих статистичних функцій, записуваних в якості виразу в обчислюваних елементах керування. Обчислюваний елемент керування, який створюється для розрахунку підсумкового значення, потрібно розміщувати в області приміток форми. В статистичних функціях не можна використовувати імена інших обчислюваних елементів керування та імена елементів керування.

Основна форма та її підлегла форма, як правило, будуються на основі таблиць, між якими встановлено зв'язок типу з один-до-багатьох. При цьому в основній формі виводиться один запис, а в підпорядкованій формі - кілька записів, які підпорядковані їй. При розрахунку підсумкового значення для групи записів підпорядкованої форми обчислюване поле може бути відображено в записі основної форми.

Засобами конструктора форм можна створювати багатотабличні форми. Конструктор є гнучкішим засобом в порівнянні з майстром, оскільки дає змогу отримувати не тільки типові рішення, передбачені майстром форм. У разі створення форми можна створювати нові елементи, кнопки, додавати підлеглі форми. У процесі роботи конструктора можуть бути виконані будь-які зміни наявних елементів, їх властивостей, видалення елементів. У разі створення форми за допомогою конструктора можна також підключати майстер створення деяких елементів керування.

У будь-яку форму, в тому числі однотобличну, можна додати підпорядковану форму, за допомогою конструктора. В якості підпорядкованої можна використовувати створену форму, або сконструювати її в процесі побудови багатотабличних форм. Можна перетягнути з вікна бази даних в область даних створену форму. У режимі конструктора в якості підпорядкованої форми можна використати форму, побудовану для підлеглої таблиці щодо джерела основних форми так, як у разі використання майстра. Крім того, якщо застосовується конструктор, в якості джерела підпорядкованої форми може бути використана головна таблиця щодо джерела основної частини. У цьому випадку для одного запису основної форми будуть відображатися дані тільки з одного запису головної таблиці.

Послідовність виконання

1. *Створення однотабличної форми в режимі форми.* Розглянемо створення однотабличної форми на прикладі таблиці «**ПІБ-Дисципліна**» бази даних «**Прізвище – Навчальний процес**». Оберіть серед переліку об'єктів бази даних таблицю «**ПІБ-Дисципліна**» і на вкладці «Створити» оберіть команду «Форма». Коротко опишіть виконувані дії. Отриманий результат відобразити в протоколі.

2. *Конструювання нової форми та додавання до неї пов'язаних полів базового джерела.* Розглянемо процес створення за допомогою конструктора нової однотабличної форми на основі таблиці «**ПІБ-Кафедра**». Оберіть серед переліку об'єктів бази даних таблицю «**ПІБ-Кафедра**» і на вкладці «Створити» оберіть команду «Конструктор форм».

Для додавання у форму полів, пов'язаних з базовим джерелом записів «**ПІБ-Кафедра**», відобразіть на екрані список полів цього джерела. Для цього виконайте команду «Додати поля» групи «Інструменти» вкладки «Конструктор». Зі списку полів додайте потрібні поля в область даних форми (розміщується поле і пов'язаний з ним напис). У самому полі буде відображено ім'я поля таблиці бази даних, а в якості напису буде використано значення з властивості поля «Напис», яке було визначено під час конструювання таблиці.

Для присвоєння формі імені «**Прізвище-Форма00**» її можна закрити і змінити ім'я за замовчуванням «Форма1». Отриманий результат, а саме форму у режимі перегляду, відобразити в протоколі. Коротко опишіть у протоколі виконувані дії.

3. *Створення незв'язаного текстового поля.* Створення у формі текстового поля може бути виконано за допомогою кнопки групи елементів керування, а саме «Текстове поле» (англ. text box).

Накресліть курсором незв'язане (англ. unbound) текстове поле на формі. Для встановлення зв'язку створюваного поля форми з полем таблиці «**ПІБ-Кафедра**», джерела записів форми, виділіть створюване поле у формі й відкрийте вікно його властивостей. У цьому вікні на вкладці «Дані» у рядку «Джерело» (англ. control source) виберіть поле зі списку доступних полів джерела і закрийте вікно властивостей. Отриманий результат відобразіть в протоколі. Коротко опишіть у протоколі виконувані дії.

4. *Редагування однотабличної форми, створеної майстром, в режимі конструктора.* Розглянемо принципи редагування на прикладі раніше створеної однотабличної користувацької форми на основі таблиці «**ПІБ-Дисципліна**» бази даних «**Прізвище – Навчальний процес**». Всі поля таблиці «**ПІБ-Дисципліна**» повинні розміщуватися в області даних форми відповідно до вимог користувача.

Відкрийте форму «**ПІБ-Дисципліна**», раніше створену в режимі форми (див. п. 1 послідовності виконання цієї роботи). Перейдіть у режим конструктора, форма відкриється у вікні конструктора форм. Варто пересвідчитись, що у заголовку форми (англ. form header) автоматично


створено графічний елемент «Напис» (англ. label). Змініть шрифт і розмір напису. Уведіть в область заголовку ще один напис – «КПІ ім. Ігоря Сікорського».

Встановіть однакові параметри форматування написів полів: шрифт Arial, 12, напівжирний. Для полів оберіть напівжирний шрифт Arial, 10.

Перегляньте властивості елементів форми. Для цього за виділеного елемента відкрийте діалогове вікно властивостей. Вікно властивостей елемента «Текстове поле» відобразить у протоколі. Збережіть форму з іменем «*Прізвище-Форма01*».

Отриманий результат відобразити в протоколі. Коротко опишіть у протоколі виконувані дії.

5. Введення нових записів в таблицю за допомогою форми. Після створення форми «*Прізвище-Форма01*» можна почати роботу з таблицею: перегляд, введення записів в таблицю, їх видалення та змінювання значень в полях записів. Розглянемо процес завантаження, перегляду і змінювання даних в таблиці «*ПІБ-Дисципліна*». Значення, що вводяться в поля форми, повинні відповідати типам даних та їх властивостям, зазначеним визначеній структурі таблиці. Перейдіть з режиму конструктора у режим форми за допомогою відповідної команди вкладки «Головна» або контекстного меню форми.

Введення і перегляд даних таблиці через форму здійснюється за записами таблиці «*ПІБ-Дисципліна*» з використанням стандартних кнопок переходу за записами, розташованими внизу форми. Для створення нового запису в таблиці «*ПІБ-Дисципліна*» створіть поточний новий запис, скориставшись відповідною кнопкою  «Новий запис» внизу форми. Уведіть значення в поля форми для нового запису. Створіть кілька нових записів.

Завершення створення або редагування запису таблиці «*ПІБ-Дисципліна*» відбувається під час переходу до будь-якого іншого запису таблиці. Запис може бути збережено лише за умови введення значень ключових полів таблиці джерела форми, а також тих полів, які було визначено у властивостях поля як обов'язкові.

Відобразить у протоколі форму, в якій представлено дані з одного рядка таблиці «*ПІБ-Дисципліна*». Коротко опишіть у протоколі виконувані дії.

6. Створення за допомогою майстра двохтабличної форми з включенням підпорядкованої форми. Створіть форму для введення, перегляду і редагування даних у таблицях «*ПІБ-Група*» і «*ПІБ-Студент*». Документом-джерелом такої форми є список студентів групи. З цього документу мають завантажуватися одночасно таблиці «*ПІБ-Група*» і «*ПІБ-Студент*», які в сукупності утворюватимуть об'єкт завантаження.

Скористайтесь майстром форм. Оберіть такі поля для відображення: з таблиці «*ПІБ-Група*» - усі поля, з таблиці «*ПІБ-Студент*» - усі поля, крім **pibcode**. Далі оберіть відображення на основі таблиці «*ПІБ-Група*» з опцією форми з підлеглою формою. Для вигляду підлеглої форми оберіть опцію

«рядковий». Вкажіть заголовок головної форми «*Прізвище-Група-головна*» і підлеглої форми «*Прізвище-Студент-підпорядкована*».

Збережіть форми з іменами «*Прізвище-Форма02-main*» та «*Прізвище-Форма02-subform*». Отриманий результат відобразіть у протоколі.

Відредагуйте головну і підлеглу форми у режимі конструктора (шрифт, ширина полів, тощо). Отриманий результат відобразіть в протоколі. Коротко опишіть у протоколі виконувані дії.

7. Проектування форми для роботи з даними двох зв'язаних таблиць. Спроектуйте форму для введення, перегляду і редагування даних у таблицях «*ПІБ-Група*» і «*ПІБ-Студент*». Документом-джерелом такої форми є список студентів групи. З цього документу мають завантажуватися одночасно таблиці «*ПІБ-Група*» і «*ПІБ-Студент*», які в сукупності утворюватимуть об'єкт завантаження.

Створіть порожню форму. У режимі конструктора або перегляду з списку полів праворуч перетягніть на форму поля **pibcode**, **pibnumber** і **pibrank** таблиці «*ПІБ-Група*». Додайте заголовок до форми за допомогою кнопки «Заголовок» групи «Заголовок/Примітка» вкладки «Конструктор». Отриманий результат відобразіть в протоколі. Збережіть форму під іменем «*Прізвище-Форма03-main*»

Для додавання підлеглої форми скористайтесь елементом керування «Підлегла форма/звіт» на вкладці «Конструктор». За допомогою маркера створіть елемент керування «Підлегла форма» потрібного розміру. Після цього з'явиться діалогове вікно майстра підпорядкованих форм за умови, що опцію використання майстрів елементів керування (англ. Use control wizards) увімкнено у групі «Елементи керування» вкладки «Конструктор». Для відображення полів таблиці «*ПІБ-Студент*» у підпорядкованій формі можна скористатись цим майстром або відмовитись і виконати додавання без використання майстра.

Наповніть підпорядковану форму без використання майстра. Для визначення джерела даних форми на вкладці «Дані» властивостей підпорядкованої форми у рядку «Джерело» оберіть «*Прізвище-Форма02-subform*», вкажіть поле **pibcode** як основне і підпорядковане поля зв'язку (англ. link master fields та link child fields). Відредагуйте вигляд головної та підпорядкованої форм. Отриманий результат відобразіть в протоколі. Коротко опишіть у протоколі виконувані дії.

8. Проектування форми для роботи з даними таблиці, підпорядкованої кільком таблицям. Здійснимо проектування форми, яка дасть змогу проводити будь-які дії з даними про заняття у підпорядкованій таблиці «*ПІБ-Навчання*» і при цьому відображати у формі довідкову інформацію з головних таблиць «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*».

Введення і редагування записів про заняття поточного семестру в таблиці «*ПІБ-Навчання*» повинні проводитися після завантаження таблиць з довідковими даними, що забезпечить встановлення зв'язків змінюваних записів з відповідними записами цих таблиць. Форма повинна забезпечувати роботу з даними про заняття у кожній групі студентів окремо, тобто записи

занять групують відповідно до їх підпорядкованості записів таблиці «**ПІБ-Група**»

Необхідно додати до форми таблицю «**ПІБ-Навчання**», як об'єкт завантаження, оскільки через форму будуть вводитися лише дані цієї таблиці. Додати таблицю «**ПІБ-Група**» як джерело основної частини складеної форми, оскільки роботу з даними таблиці «**ПІБ-Навчання**» варто виконувати окремо для кожної групи. Також потрібно додати таблиці «**ПІБ-Дисципліна**» і «**ПІБ-Викладач**» для відображення описових даних за курсами і викладачами.

Створіть нову форму й назвіть «**Прізвище-Форма04-План**». Цю головну форму створіть на основі таблиці «**ПІБ-Група**» та усіх її полів. Створіть нову форму й назвіть «**Прізвище-Форма04-Навчання**». Цю підпорядковану форму створіть на основі таблиці «**ПІБ-Навчання**» та усіх її полів крім ключового.

Для відображення назви дисципліни та прізвища викладача додайте в підлеглу форму описові поля з головних таблиць, а саме з таблиці «**ПІБ-Дисципліна**» поля **pibdistitle**, **pibdisnumber**, **piblab**, **pibpractice**, з таблиці «**ПІБ-Викладач**» - **pibsurname**, **pibtitle** та **pibdegree**.

Додайте підпорядковану форму до головної. За допомогою керування властивостями полів форми, які відносяться до таблиць «**ПІБ-Група**», «**ПІБ-Дисципліна**» та «**ПІБ-Викладач**», унеможливіть їх редагування, варто обмежитись тільки читанням.

Відредагуйте вигляд головної та підпорядкованої форм. Отриманий результат у режимі перегляду відобразить в протоколі. Коротко опишіть у протоколі виконувані дії.

9. Розробка одиночної багатотабличної форми за допомогою майстра. Розглянемо створення одиночної багатотабличної форми без підлеглих і зв'язаних форм, призначених для роботи з даними таблиці «**ПІБ-Навчання**». Відповідно до схеми бази даних «**Прізвище – Навчальний процес**» таблиця «**ПІБ-Навчання**», що містить перелік усіх занять, підпорядкована таблицям «**ПІБ-Дисципліна**» і «**ПІБ-Викладач**», які містять довідкові дані. Для відображення в одному записі форми даних про окреме заняття з таблиці «**ПІБ-Навчання**», а також додаткової довідкової інформації (предмет і викладач) можна побудувати багатотабличну форму без підлеглих форм, тобто одиночну форму. У процесі створення одиночної форми майстром автоматично формується запит на з'єднання підпорядкованої таблиці з головною. Цей запит стає джерелом записів одиночної форми. Запит можна переглянути в загальних властивостях форми.

Схема даних для форми буде складатися з трьох таблиць: «**ПІБ-Навчання**», «**ПІБ-Дисципліна**» і «**ПІБ-Викладач**». Джерелом записів форми повинна бути таблиця «**ПІБ-Навчання**». Інші дві таблиці будуть джерелами додаткових полів для розшифровки ідентифікаторів заняття.

У діалоговому вікні майстра форм оберіть поля таблиці «**ПІБ-Навчання**», що додаються у форму, а потім - поля таблиць «**ПІБ-Дисципліна**» і «**ПІБ-Викладач**».

У наступному сеансі цього діалогового вікна виділено таблицю «**ПІБ-Навчання**» як основне джерело для створення форми. Оскільки ця

таблиця є підпорядкованою в схемі даних щодо інших таблиць, використовуваних у формі, майстер пропонує побудувати форму типу «одиначна форма». Майстер будуватиме форму без підлеглих форм з полями з усіх таблиць. Оберіть вигляд форми «в один стовпчик» (англ. columnar), за якого у формі буде відображатися тільки один запис, побудований шляхом додавання полів з таблиць «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*» до запису таблиці «*ПІБ-Навчання*». Відобразіть у протоколі одиначну багатотабличну форму, побудовану майстром.

Перегляньте інструкцію SQL, записану майстром як джерело записів отриманої форми. Для цього виділіть форму в режимі конструктора, відкрийте вікно властивостей і вкладку «Дані». Перегляньте відображення інструкції SQL у вигляді запиту за зразком Для цього натисніть кнопку виклику будівника в кінці рядка джерела записів у вікні властивостей форми. Побудовану інструкцію SQL відобразіть в протоколі.

Збережіть форму з іменем «*Прізвище-Форма05*». Коротко опишіть у протоколі виконувані дії.

10. Зміна написів і розміщення полів. Відкрийте форму «*Прізвище-Форма02-main*» в режимі конструктора. У цій формі представлено поля, розміщені в області даних, з написами основної частини форми, а також поля і написи підпорядкованої форми «*Прізвище-Форма02-subform*» всередині рамки. Збережіть ці форми з іменами «*Прізвище-Форма06-main*» та «*Прізвище-Форма06-subform*».

Змініть в області заголовка назву форми на «Список студентів групи №», що буде виводитися у разі роздрукування форми. Змініть розміщення й розмір полів головної й підпорядкованої форм з метою зручнішого перегляду. Уточніть підписи полів, шрифт і розміри полів і підписів, заголовку форми. Вирівняйте елементи за допомогою команд групи «Розмір і порядок» вкладки «Впорядкування». Відобразіть у протоколі головну форму в режимі перегляду. Коротко опишіть у протоколі виконувані дії.

11. Створення кнопок управління за допомогою майстра. Для того щоб у формі переходити до наступного і попереднього запису таблиці «*ПІБ-Група*», створимо відповідні кнопки керування в основній частині форми.

Відкрийте форму «*Прізвище-Форма06-main*» в режимі конструктора. Збережіть цю форму з іменем «*Прізвище-Форма07-main*». Вставте з вкладки «Конструктор» у головну форму інструмент керування «Кнопка». Запуститься майстер кнопок (англ. command button wizard). У вікні майстра кнопок оберіть дію, яку потрібно виконувати у разі натискання кнопки. В області категорій оберіть «Перехід за записами», в області дій – «Попередній запис» (англ. goto previous record). У наступному сеансі діалогового вікна можна вибрати вид кнопки (текст або малюнок). Таким чином кнопку переходу буде вбудовано у форму. Дії повторіть для вбудовування кнопки переходу до наступного запису таблиці.

Створіть кнопку для закриття форми. В області категорій оберіть роботу з формою, а в області дій – закриття форми. Відредагуйте розмір і підказки для цих кнопок за допомогою властивості «Підказка» (англ. controlltip text).

Відобразіть у протоколі головну форму в режимі перегляду. Коротко опишіть у протоколі виконувани дії.

Редагування підпорядкованої форми. Аналогічні дії з доопрацювання виконайте для підлеглої форми «*Прізвище-Форма06-subform*». Змініть написи полів-стовпців в заголовку форми, а також шрифт, розміри полів і написів. Відобразіть у протоколі підпорядковану форму в режимі перегляду. Коротко опишіть у протоколі виконувани дії.

12. *Введення даних у взаємозалежні таблиці через форму.* Введіть нові дані в таблиці «*ПІБ-Група*» і «*ПІБ-Студент*». Створена багатотаблична форма з заголовком «*Прізвище-Форма07-main*» забезпечує одночасну роботу з даними двох таблиць.

Для введення даних щодо групи попередньо здійсніть перехід до нового (порожнього) запису групи за допомогою, наприклад, створеної у формі кнопки «Наступний». Введіть основну частину форми значення реквізитів групи – шифр групи, кількість студентів і середній бал групи.

Введення ключового поля з шифром групи обов'язковий для створення запису в таблиці «*ПІБ-Група*». Причому цей запис створюється незалежно від того, чи буде введено список студентів групи. Інші два описових реквізити групи можна ввести і пізніше, якщо не було додаткових обмежень на їх значення у властивостях під час конструювання таблиці.

Введіть, не змінюючи поточного запису групи, реквізити студентів у область підпорядкованої форми. Введення номера студента у групі завжди обов'язковий для створення запису в таблиці «*ПІБ-Студент*», який разом з введеним в основну частину форми номером групи утворює унікальний ключ в цій таблиці. Запис зберігається після переходу до чергового рядка в підлеглій формі. Здійсніть аналогічно введення даних для кількох наявних і новостворених груп студентів.

Відобразіть у протоколі принаймні одне введення нового запису в таблицю «*ПІБ-Студент*» через форму для новоствореної групи, а також таблиці «*ПІБ-Студент*» і «*ПІБ-Група*» з новими записами. Коротко опишіть у протоколі виконувани дії.

14. *Обчислення підсумкового значення для записів підпорядкованої форми і виведення його в основній формі.* Нехай у формі «*Прізвище-Форма07-main*» потрібно на основі записів підпорядкованої форми «*Прізвище-Форма06-subform*» підрахувати кількість студентів. Збережіть ці форми з іменами «*Прізвище-Форма08-main*» та «*Прізвище-Форма08-subform*»

У підпорядкованій формі в області приміток створіть обчислюваний елемент керування з ім'ям «zvd14» і запишіть туди вираз, що містить статистичну функцію підрахунку кількості записів = Count ([surname]). Причому в якості аргументу функції потрібно вказувати не ім'я елемента керування у формі, а ім'я поля, оскільки джерелом даних є таблиця.

Елементи керування з області приміток не відображаються при перегляді підпорядкованої форми, якщо в її властивості «Режим за замовчуванням» (англ. default view) встановлено значення «Таблиця» (англ. datasheet).

Для відображення результату обчислення в основній частині форми створіть обчислюваний елемент керування і запишіть вираз


=[Forms]![ім'я основної форми]![ім'я елемента керування основної форми, який містить підпорядковану форму].[Form]![ім'я елемента керування у підпорядкованій формі]

Наприклад,

=[Forms]![Прізвище-Форма08-main]![Прізвище-Форма08-subform].
[Form]![zvd14]

Відобразіть у протоколі вираз уведений в обчислюваний елемент керування і форму у режимі перегляду. Коротко опишіть у протоколі виконувані дії.

15. Використання запиту як джерела записів багатотабличної форми. Розглянемо додавання в нову форму під час її конструювання поля **pibsurname** з таблиці «**ПІБ-Викладач**» і поля **pibdeptname** з таблиці «**ПІБ-Кафедра**».

Створіть нову форму без елементів керування (англ. blank form) і в режимі конструктора у списку властивостей форми, яку можна відобразити натисканням команди «Властивості» у групі «Інструменти» вкладки «Конструктор», на вкладці «Дані» відкрийте будівник  з рядка «Джерело записів». У діалоговому вікні «Показати таблицю» оберіть таблиці «**ПІБ-Кафедра**» і «**ПІБ-Викладач**», які з'являться у схемі даних запиту. У бланк запиту додайте поле **pibsurname** з таблиці «**ПІБ-Викладач**» і поля **pibdeptname** з таблиці «**ПІБ-Кафедра**». Під час закриття цього запиту з'явиться вікно підтвердження його збереження у вигляді інструкції SQL, яка стане значенням властивості «Джерело записів» нової форми. Відобразіть у протоколі створену інструкцію SQL.

У списку полів форми, який можна відобразити натисканням команди «Додати поле» у групі «Інструменти» вкладки «Конструктор», з'явиться два доступні елементи **pibsurname** і **pibdeptname**. Додайте ці поля на форму. Збережіть цю форму з іменем «**Прізвище-Форма09**». Створену форму в режимі перегляду представте у протоколі. Коротко опишіть у протоколі виконувані дії.

16. Вбудовування підпорядкованої форми засобами конструктора. Створіть однотабличну форму «**Прізвище-Форма10**», джерелом записів якої є таблиця «**ПІБ-Викладач**».

Для вбудовування підпорядкованої форми в режимі конструктора оберіть на вкладці «Конструктор» у групі «Елементи керування» елемент керування «Підпорядкована форма / звіт». Зауважте, що кнопка майстра елементів керування не повинна бути активною. Додайте рамку підпорядкованої форми. Встановіть зв'язок між формою «**Прізвище-**

Форма00», створеною у результаті виконання п. 2 завдання, і елементом керування «Підпорядкована форма» з іменем ChildN. У вікні властивостей елементу керування ChildN на вкладці «Дані» у рядку «Об'єкт-джерело» оберіть форму «**Прізвище-Форма00**». У рамці підпорядкованої форми відобразиться обрана форма в режимі конструктора.

Зв'язок між основною і підпорядкованою формами встановлюється автоматично, якщо в схемі бази даних визначено зв'язок таблиць, відповідних форм. Якщо зв'язки між таблицями не визначено, то необхідно виконати дії з встановлення зв'язку. У вікні властивостей елементу керування ChildN у рядках «Підлеглі поля» (англ. link child fields) та «Основні поля» (англ. link master fields) введіть імена поля(ів), за якими зв'язуються таблиці. Для цього завдання таким полем є **pibdept**. Створену форму в режимі перегляду представте у протоколі. Коротко опишіть у протоколі виконувані дії.

17. Використання додаткового майстра для додавання підпорядкованої форми. Переконаємося в тому, що джерелом підпорядкованої форми може бути головна таблиця «**ПІБ-Кафедра**». Створіть однотабличну форму «**Прізвище-Форма11**», джерелом записів якої є таблиця «**ПІБ-Викладач**».

Для вбудовування підпорядкованої форми в режимі конструктора оберіть на вкладці «Конструктор» у групі «Елементи керування» елемент керування «Підпорядкована форма / звіт». Зауважте, що кнопка майстра елементів керування повинна бути активною.

У діалоговому вікні майстра підпорядкованих форм оберіть існуючу форму «**Прізвище-Форма00**» або таблицю «**ПІБ-Кафедра**», на основі якої повинна бути створена підлегла форма. Після завершення роботи додаткового майстра підлеглу форму буде вбудовано в форму «**Прізвище-Форма11**», тобто буде виводитися довідка про кафедру, на якій працює викладач.

Коротко опишіть у протоколі виконувані дії.

18. Вбудовування підпорядкованої форми перетягуванням готової форми. Створіть однотабличну форму «**Прізвище-Форма12**», джерелом записів якої є таблиця «**ПІБ-Викладач**». У вікні бази даних виділіть форму «**Прізвище-Форма00**» і перетягніть її в область даних форми «**Прізвище-Форма12**». Відразу після перетягування готова форма залишається ще незв'язаною. Для того щоб зв'язати вбудовану форму з основною, необхідно виділити вбудовану форму, викликати вікно властивостей і записати у рядки «Підлеглі поля» і «Основні поля» ім'я поля зв'язку таблиць-джерел основної та вбудованої форми. Для виділеної вбудованої форми «**Прізвище-Форма00**» відкрийте вікно властивостей і запишіть ім'я поля зв'язку **dept** в рядки «Підлеглі поля» та «Основні поля».

Коротко опишіть у протоколі виконувані дії.

19. Вбудовування підпорядкованої форми перетягуванням таблиці-джерела підпорядкованої форми. Якщо перетягнути таблицю для створення підпорядкованої форми, активується додатковий майстер. У результаті вбудовується рамка підпорядкованої форми і запускається майстер. Майстер

пропонує в якості поля для зв'язку форм зовнішній ключ підпорядкованої таблиці і автоматично створює багатозаписну форму.

Створіть однотобличну форму «*Прізвище-Форма13*», джерелом записів якої є таблиця «*ПІБ-Кафедра*». Виділіть у вікні бази даних таблицю «Викладач» і перетягніть цей об'єкт в область даних відкритої форми. Автоматично активується додатковий майстер. Відобразіть у протоколі отриману форму в режимі конструктора з підпорядкованою формою, яку створив майстер після перетягування у форму таблиці-джерела. Коротко опишіть у протоколі виконувані дії.

20. Підготовка майстром форми для роботи з даними підпорядкованої таблиці. Створіть відповідно до результатів проектування складену форму плану занять, що міститиме підлеглу форму для роботи з даними таблиці «*ПІБ-Навчання*», з використанням майстра і конструктора форм. Підлегла форма повинна бути побудована як одиночна з основним джерелом записів – таблицею «*ПІБ-Навчання*» – з приєднанням полів з головних таблиць «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*».

Створіть нову форму за допомогою майстра форм, а в якості таблиці, яка буде джерелом записів основної частини складеної форми вкажіть таблицю «*ПІБ-Група*». В діалоговому вікні створення форм крім полів таблиці «*ПІБ-Група*» потрібно додати поля таблиць «*ПІБ-Навчання*», «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*». У наступному сеансі діалогового вікна виділіть таблицю «*ПІБ-Група*», яка стане головною для створення основної частини форми. Далі оберіть тип форми «Підлеглі форми» для відображення записів підпорядкованої таблиці «*ПІБ-Навчання*».

У відображенні розміщення полів у вікні майстра видно, що поля підлеглої таблиці «*ПІБ-Навчання*» і головних щодо неї таблиць «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*» буде розміщено всередині рамки підпорядкованої форми. Виберіть в наступному сеансі вікна майстра вигляд форми «Рядковий».

В останньому сеансі вікна створення форм задайте ім'я складеної форми «*Прізвище-Форма14*», а також ім'я підпорядкованої форми «*Прізвище-Форма14-Навчання*». Виберіть для доопрацювання форми як подальшу дію майстра «Змінити макет форми».

В отриманій підлеглій формі повинні виводитися всі записи про заняття для однієї групи з таблиці «*ПІБ-Навчання*» з додатковими полями довідкових даних з головних таблиць. Оберіть значення властивості «Вигляд за замовчуванням» з групи «Формат» властивостей підлеглої форми – «Суцільні форми»(англ. continuous forms).

Відобразіть у протоколі створену форму «*Прізвище-Форма14*» в режимі перегляду. Коротко опишіть у протоколі виконувані дії.

21. Доопрацювання форми конструктором для отримання зручного інтерфейсу користувача. У підлеглій формі «*Прізвище-Форма14-Навчання*» форми «*Прізвище-Форма14*» запис зайняття має вигляд стрічки, що містить всі поля з таблиць «*ПІБ-Навчання*», «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*». Лише поля з таблиці «*ПІБ-Навчання*» служать для введення даних. Інші поля

повинні відображатись із завантажених таблиць «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*», але не повинні бути доступні користувачу для редагування, додавання і видалення записів.

Отримане розміщення полів незручне для користувача, тому форму доцільно доопрацювати засобами конструктора. У процесі редагування форми в режимі конструктора:

- забезпечить захист від змін довідкових даних у таблицях «*ПІБ-Група*», «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*»;
- передбачте для зручності користувача кнопки переходу до перегляду занять для іншої групи і кнопку закриття форми;
- для візуальної перевірки правильності введення ідентифікаторів викладача і предмета створіть поля зі списками, у яких відображатимуться відповідні довідкові дані (п. 22).

В основній частині форми відредагуйте поля таблиці «*ПІБ-Група*». Уточніть текст підписів полів, шрифт і розміри полів і написів, введіть текст у заголовки форми. Видаліть елемент з підписом підпорядкованої форми. Змініть кілька властивостей форми «*Прізвище-Форма14*».

Створіть дві кнопки для переходу до наступної або попередньої групи, а також кнопку для закриття форми. Захистіть дані записів таблиці «*ПІБ-Група*» від змінювання під час роботи з формою, оскільки вони повинні використовуватися тільки для відображення. Для захисту певного поля виділіть рамку поля і у вікні властивостей на вкладці «Дані» у рядку «Блокування» (англ. locked) оберіть значення «Так». Після вказування такого значення цієї властивості поле буде доступно лише для читання. Збережіть форму з іменем «*Прізвище-Форма15*».

За допомогою майстра отримано підпорядковану рядкову форму «*Прізвище-Форма14-Навчання*» в режимі конструктора. Для змінювання вигляду підпорядкованої форми активуйте її властивості. На вкладці «Макет» замініть у рядку «Режим за замовчуванням» значення «Суцільні форми» на значення «Проста форма» (англ. single form). Це дасть змогу відображати в підлеглій формі лише один запис про заняття. Вид форми у разі зміни режиму за умовчанням у конструкторі залишиться тим самим. У підлеглій формі розмістіть поля так як зручніше, на Вашу думку, користувачеві. Варто поля таблиці «*ПІБ-Навчання*», в які потрібно буде вводити дані, розмістити у верхній частині області даних. У нижній частині області даних розмістіть поля, в які будуть автоматично виводитися довідкові дані з таблиць «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*». Щоб, за необхідності, створити рамки, використовуємо команду «Прямокутник» з групи елементів керування.

Уточніть текст підписів полів, шрифт і розміри полів і написів. Виконайте відносне вирівнювання написів і полів за допомогою команд вкладки «Впорядкування» групи «Розмір і порядок». В область приміток форми введіть інструкцію, яка вимагає від користувача обов'язково ввести дані в поля, що ідентифікують заняття: **pibcode**, **pibtab** і **pibtype**. Без цього не можливо створити новий запис у таблиці «Навчання». Захистіть від

випадкових змін довідкові дані полів **pibdistitle**, **pibdiscode**, **pibhours**, **piblab**, **pibpractice**. Збережіть форму з іменем «**Прізвище-Форма15-Навчання**».

Відобразіть у протоколі відредаговані форми в режимі перегляду. Коротко опишіть у протоколі виконувані дії.

22. Створення поля зі списком за допомогою майстра. Для підлеглої форми «**Прізвище-Форма15-Навчання**» розгляньте можливість візуального контролю правильності введення ідентифікаторів заняття **pibdiscode** і **pibtab**, якщо для них створити поле зі списком (англ. combobox).

Створіть поле зі списком для введення значень коду предмета **pibdiscode** з таблиці «**ПІБ-Навчання**». Це дасть змогу переглядати і вводити значення, які вже є в головній таблиці «**ПІБ-Дисципліна**», а також перевіряти відповідність коду і найменування предмета, наявні в документі-джерелі. Оберіть на вкладці «Конструктор» у групі «Елементи керування» елемент керування «Поле зі списком», додайте елемент на форму. Активується майстер елемента керування «Поле зі списком».

Визначіть спосіб, яким список поля отримуватиме значення. Для формування списку зі зв'язаної таблиці оберіть варіант з використанням значень з таблиці або запиту (англ. I want the combo box to get the values from another table or query).

У наступному вікні оберіть таблицю «**ПІБ-Дисципліна**», яка буде джерелом значень в списку поля. Оберіть поля **pibdiscode** та **pibdistitle** для відображення найменування предмета. Ці поля утворять записи списку, для яких можна обрати порядок сортування та ширину стовпця у списку.

Оберіть поле **pibdiscode** джерелом значень для поля форми, що є ключем зв'язаної таблиці «**ПІБ-Дисципліна**». З цього поля буде обиратись значення для введення в поле форми.

Оберіть перемикач «Зберегти в полі» (англ. store that value in this field) і поле форми **pibdiscode**, як поле таблиці «**ПІБ-Навчання**», в яке будуть вводитися значення зі списку.

Введіть підпис поля зі списком «Код дисципліни». У результаті одержано поле зі списком. Варто пересвідчитись, що у властивостях макету списку у рядку кількості стовпців вказано значення «2», а у рядку ширини стовпців вказано два значення відмінні від нуля і розділені крапкою з комою. Відобразіть у протоколі розкритий список в режимі перегляду. Коротко опишіть у протоколі виконувані дії.

23. Створення поля зі списком без використання майстра. Створіть без використання майстра поле зі списком зі значеннями ідентифікатора викладача (**pibtab**). Це дасть змогу переглядати і вводити значення, які вже є в головній таблиці «**ПІБ-Викладач**», а також перевіряти відповідність номера та прізвища викладача, що містяться в документі-джерелі завантаження. Зауважте, що кнопка майстра елементів керування повинна бути неактивною.

Розмістіть поле зі списком на формі. У рядку «Дані» (англ. control source) властивостей поля зі списком оберіть поле **pibtab** таблиці «**ПІБ-Навчання**». У рядку типу джерела рядків оберіть «Таблиця / запит», у рядку джерела рядків – таблицю «**ПІБ-Викладач**». Поля, що міститимуться в

списку, та їх порядок визначаються будівником. Активуйте будівника у рядку джерела рядків (англ. row source) і додайте до бланку запитів поля **pibtab** та **pibsurname** з таблиці «**ПІБ-Викладач**». Для того щоб у списку виводилися два поля, на вкладці «Макет» у рядку кількості стовпців вкажіть значення «2». Для налаштування ширини стовпців списку в рядках «Ширина списку» і «Ширина стовпців» задайте відповідні значення. У результаті створено поле зі списком, яке розміщено на формі.

Якщо необхідно перетворити звичайне поле в поле зі списком, потрібно в контекстному меню поля обрати пункт «Перетворити елемент в» і обрати рядок «Поле зі списком». У разі активної кнопки Майстра елементів елемент буде перетворено майстром. Якщо цю команду відключено, необхідно для отримання поля зі списком встановити властивості цього поля так, як описано вище.

Відобразіть у протоколі розкритий список в режимі перегляду. Коротко опишіть у протоколі виконувані дії. Збережіть форму з іменем «**Прізвище-Форма16-Навчання**».

24. Перегляд і коригування облікових даних. Відкрийте форму «**Прізвище-Форма15**». Змініть режим на режим форми.

Для перегляду записів занять у групі зробіть спершу поточним номер потрібної групи. Для послідовного перегляду занять за групами використовуйте кнопки «Попередня група» і «Наступна група».

У разі великої кількості груп доцільно використовувати функцію пошуку потрібного запису. Щоб здійснити пошук, курсор потрібно встановити в поле номера групи, командою «Знайти» вкладки «Головна» і у вікні пошуку задати зразок пошуку. Після натискання в цьому вікні кнопки «Знайти далі» (англ. find next) у формі відобразяться дані про заняття вказаної групи. Здійсніть перегляд всіх занять у вказаній групі. Для цього можна використовувати наявні в нижньому лівому куті підпорядкованої форми в рядку «Запис» стандартні кнопки переходу за записами.

Змініть для одного із записів значення у полі **pibhours**. Зміна значень у полях **pibdiscode**, **pibtab** і **pibtype** очевидно може проводитися тільки в разі помилкового введення. При цьому слід мати на увазі, що зміна цих значень призводить до зміни зв'язків редагованого запису з відповідною головною таблицею у зв'язку з підтримкою цілісності. Коротко опишіть у протоколі виконувані дії. Оформити протокол лабораторної роботи.

Оформлення протоколу

Після оформлення протоколу лабораторної роботи до заздалегідь підготовлених складників додаються роздруковки скріншотів ходу виконання завдань.

Контрольні запитання

1. Чи може форма, створена майстром, містити дані з кількох таблиць?
2. Як у формі встановити захист від змінювання даних поля?
3. Який елемент керування дає змогу створити обчислюване поле у формі?
4. З якого символу починається вираз, який записано в обчислюване поле?
5. Як викликати Будівник для формування виразу в обчислюваному полі?
6. Як відобразити у головній формі підсумкове значення, обчислене у підпорядкованій формі?
7. Який формат має посилання на обчислюване поле у підпорядкованій формі?
8. Що потрібно виконати, якщо не працює зв'язок основної та підпорядкованої форми?
9. Чи може зв'язок між основною та підпорядкованою формами встановлюватись за кількома полями?
10. Чи можна шляхом перетягування таблиці з вікна бази даних у форму створити у ній підпорядковану форму?
11. Чи може майстер побудувати форму на основі непов'язаних таблиць?
12. Чи може майстер побудувати форму на основі раніше створеного запиту?
13. У якому елементі керування розміщено підпорядковану форму?
14. У яких властивостях вказують поля зв'язку основної та підпорядкованої форм?
15. Яка команда дає змогу створювати форму на основі вибраної таблиці?
16. Для чого призначено групу елементів керування конструктора форм?
17. Як у режимі конструктора переглянути доступні у формі поля?
18. За допомогою якої команди переглядають властивості елемента керування?
19. Яка кнопка дає змогу переключитись з режиму конструктора у режим форми?
20. Чи можна у формі за допомогою кнопок переходу за записами зробити поточним новий запис?

Лабораторна робота № 6

Створення та редагування звітів в СКБД MS Access

Мета роботи. Набути умінь використання прийомів роботи зі звітами у середовищі MS Access 2016.

Завдання. Розглянути засоби створення звітів у СКБД MS Access 2016.

Короткі теоретичні відомості

Засоби розроблення звітів у MS Access призначено для створення макету звіту, за яким може бути здійснено виведення даних у вигляді вихідного

друкованого документа. Ці засоби дають змогу створювати звіт складної структури, що забезпечує виведення взаємозв'язаних даних з багатьох таблиць, їх групування, обчислення, фільтрування даних.

Перед початком створення звіту користувач повинен спроектувати макет звіту, а саме визначити склад і зміст розділів звіту, розміщення в ньому значень, що виводяться з полів таблиць бази даних, і обчислюваних реквізитів, а також визначити поля, за якими потрібно групувати дані. Для кожного рівня групування визначають заголовки і примітки, обчислювані підсумкові значення. Крім того, оформлюють заголовки і підписи реквізитів звіту і визначають порядок виведення даних у звіт.

Звіт можна створити за допомогою майстра або в режимі конструктора звітів. Звіт, створений майстром, можна доопрацювати в режимі конструктора.

За необхідності виведення результатів виконання задачі, в якості основи для звіту може бути використано багатотабличний запит. На запит може бути покладено складні види вибірки та попередньої обробки даних. Конструктор звітів дає змогу легко структурувати і оформлювати дані, отримані в запиті.

Поряд з однотобличними звітами MS Access дає змогу створювати складніші звіти, що забезпечують виведення даних з кількох взаємопов'язаних таблиць бази даних. Для виведення взаємозв'язаних даних у багатотабличний звіт автоматично використовуються зв'язки таблиць, встановлені в схемі даних БД. Багатотабличні звіти можуть містити основну частину і допоміжну частину, тобто підлеглий звіт. Для кожної з цих частин як джерела даних вибирають свою таблицю або декілька таблиць. Багатотабличні звіти можуть охоплювати поля з різних таблиць і при цьому мати підлеглі звіти. Найпростішим багатотабличним звітом є звіт для двох таблиць, що знаходяться в одно-багатозначних відносинах. У цих відносинах одна таблиця є головною, а інша – підлеглою. Для створення звіту з висновком записів деякої підпорядкованої таблиці відображенням довідкових даних з головних таблиць потрібно в якості основного джерела вибрати підпорядковану таблицю. Цей багатотабличний звіт не буде містити підлеглої частини. Для створення такого звіту можна використовувати майстер звітів, який дає змогу вибрати з взаємозалежних таблиць потрібні поля в заданій послідовності та вказати, яка з таблиць у звіті буде записостворювальною, тобто основним джерелом даних. Крім того, майстер дає можливість групувати та сортувати записи звіту за різними полями, підрахувати підсумкові значення.

Послідовність виконання

1. *Створення однотобличного звіту за допомогою конструктору.* Розглянемо створення однотобличного звіту на прикладі одержання списків студентів за групами. Вимоги до макету звіту «*Прізвище–Звіт01*»:

- списки студентів кожної групи у звіті повинні виводитися послідовно разом із заголовками;

- необхідно розрахувати середній прохідний бал для кожної групи і відобразити його в звіті;
- записи списку групи повинні виводитися в порядку зростання номера студента в групі;
- на кожній сторінці звіту необхідно вивести його назву.

В області навігації оберіть таблицю «**ПІБ-Студент**», яка буде джерелом даних для звіту, і виконайте команду «Звіт» групи «Звіти» з вкладки «Створити». У відкритому вікні конструктора макет створюваного звіту буде містити розділи. Результат відобразить у протоколі.

Пересвідчіться, що розділ заголовку звіту (англ. report header) відображено. Для виконання вимог щодо групування та сортування даних, представлених у звіті, натисніть команду «Групування і сортування» вкладки «Конструктор», якщо область групування, сортування і підсумків не відображено. У режимі макету виконайте групування за полем **pibcode**, оскільки загальний список необхідно розбити за групами. Виконайте команду «Додати групування» і в полі групування, сортування і підсумків внизу вікна з'явиться новий рядок і буде показано список доступних полів. Якщо перемкнутись у режим конструктора, то можна побачити новий розділ («Заголовок групи '**pibcode**'»).

Задайте сортування за полем **pibstudent** для виведення відсортованого списку студентів у кожній групі. Для встановлення у групі сортування за номерами студентів виконайте команду «Додати сортування», а у списку полів **pibstudent**. Рядки звіту одразу буде відсортовано, а в області групування, сортування і підсумків з'явиться новий рядок, який вказуватиме на сортування за полем **pibstudent**.

Розмістіть поле **pibcode** у розділі «Заголовок групи '**pibcode**'» для того щоб значення шифру групи було представлено один раз в заголовку групи. Відкоригуйте підпис поля, змінивши його на «Список студентів групи». Встановіть прийнятний шрифт в елементах. Встановіть розмір рамки за розміром тексту підпису.

Пересвідчіться, що поля **pibstudent**, **pibsurname**, **pibbirthyear** та **pibrank** розміщено в області даних, яка визначає вміст рядків табличної частини. Поле розміщується разом з підписом, який система визначає за властивостями полів таблиці «**ПІБ-Студент**».

Відобразить у протоколі результат у режимі перегляду і в режимі конструктора та коротко опишіть виконувані дії.

2. Додавання до звіту обчислюваного поля, заголовку, поточної дати, номера сторінки. Для додавання до звіту розрахункового параметру «Середній прохідний бал групи» натисніть кнопку «Поле» (англ. text box) на панелі елементів і розмістіть елемент «Вільний» (англ. unbound) у розділі «Примітка» групи **pibcode** (**pibcode** footer). Вкажіть у властивостях елементу «Вільний» вираз для розрахунку середнього значення. Для цього уведіть на вкладці «Дані» у рядок «Дані» (англ. control source) функцію $\text{=Avg}([\text{pibrank}])$, в рядок «Число десяткових знаків» (англ. decimal places) – 2,

на вкладці «Макет» у рядок «Формат поля» (англ. format) – «Фіксований» (англ. fixed). Відредагуйте підпис поля. Для цього виділіть підпис та у властивостях на вкладці «Макет» у рядку «Підпис» (англ. caption) вкажіть «Середній прохідний бал групи».

Такі дії, як зміна підпису або введення виразу в полі, можна виконати і не звертаючись до властивостей елементів.

Додайте до звіту поточну дату за допомогою вбудованої функції NOW(). Для цього створіть у заголовку звіту вільний елемент, натиснувши кнопку «Поле», і задайте у вікні його властивостей на вкладці «Дані» у рядку «Дані» функцію =NOW(). На вкладці «Макет» у рядку «Формат поля» виберіть значення «Повний формат дати» (англ. general date). Підпис цього поля виділіть і видаліть.

Додайте до звіту номер сторінки в нижньому колонтитулі. Створіть вільний елемент і вкажіть у його властивостях на вкладці «Дані» у рядку «Дані» вираз =[Page]. Відредагуйте підпис цього поля, вказавши «Сторінка» в його властивостях на вкладці «Макет» у рядку «Підпис».

Додайте назву звіту «*Прізвище – СПИСКИ СТУДЕНТІВ*» у розділ «Тема звіту» (англ. report header). Для цього скористайтесь кнопкою панелі елементів «Напис» (англ. label). Встановіть інший шрифт напису за допомогою інструментів форматування. Для виведення назви звіту на наступних сторінках додайте його також у рядок «Верхній колонтитул» (англ. page header). Для цього можна скопіювати назву з розділу «Тема звіту» і вставити в рядок «Верхній колонтитул» та обрати потрібний шрифт. Також варто вказати «Без заголовка» у властивостях звіту на вкладці «Макет» у рядку «Верхній колонтитул».

Отриманий результат відобразить у протоколі. Збережіть звіт з іменем «*Прізвище-Звіт02*».

3. Перегляд і друк звіту. Перейдіть з режиму конструктора звітів у режим попереднього перегляду, натиснувши кнопку «Попередній перегляд» (англ. report view). Для перегляду створеного звіту потрібно у вікні бази даних вибрати потрібний звіт на вкладці «Звіти» (англ. reports) і натиснути кнопку «Перегляд» (англ. preview).

Звіт під час перегляду відобразиться на екрані таким, яким його буде надруковано. У режимі попереднього перегляду є свої групи інструментів. Для перегляду потрібних сторінок звіту можна використовувати стандартне поле номера сторінки в нижньому лівому кутку вікна звіту.

Виведіть звіт на друк за допомогою команди «Друк» (англ. print) у pdf-файл. Відобразить у звіті першу сторінку розробленого звіту і списки студентів у режимі попереднього перегляду.

4. Створення основної частини звіту на базі головної таблиці. Розгляньте прийоми створення багатотабличного звіту на основі таблиць

«**ПІБ-Кафедра**» і «**ПІБ-Викладач**», який міститиме відомості про кожну кафедру і списки її викладачів. Необхідно вивести у звіт дані для кожної кафедри, зокрема, назву, код та телефон, а також прізвище і фотографію завідувача. Ці дані містяться в таблиці «**ПІБ-Кафедра**». У табличній частині для кожної кафедри потрібно вивести дані про викладачів кафедри, які містяться в таблиці «**ПІБ-Викладач**».

Почніть створення звіту в режимі конструктора. У вікні бази даних виберіть об'єкт «Звіти» і натиснемо команду «Створити». У вікні «Новий звіт» оберіть пункт «Конструктор» для побудови звіту в режимі конструктора. Оберіть у тому ж вікні відповідно до макету звіту таблицю «**ПІБ-Кафедра**», яка буде джерелом даних для основної частини звіту. У вікні конструктора звітів додайте розділ «Тема звіту» (англ. report header) за допомогою команди «Тема / примітка звіту» (англ. report header / footer) конструктора звітів. Додайте текст «**Прізвище – КАФЕДРА**», скориставшись командою «Напис». У область даних перетягніть поля **pibdeptname**, **pibdept**, **pibphone**, **pibhead** і **pibphoto** таблиці «**ПІБ-Кафедра**» зі списку, який відкрийте командою «Список полів» конструктора звітів. Розмістіть поля і підписи, відредагуйте їх. Додайте в рядок «Тема звіту» поточну дату, скориставшись командою «Вставка дати і часу», а у рядку «Нижній колонтитул» вкажіть номер сторінки.

Збережіть звіт з іменем «**Прізвище–Звіт03**». Відобразіть у протоколі хід заповнення основної частини звіту в режимі конструктора.

5. Створення підлеглого звіту в режимі Автозвіт. Для виведення у звіт «**Прізвище–Звіт03**» з таблиці «**ПІБ-Викладач**» списку викладачів відповідної кафедри підготуйте окремий звіт, який буде додано в основну частину звіту в якості підлеглого. Для автоматичного створення багатозаписного підлеглого звіту вікні «Новий звіт» оберіть режим «Автозвіт: рядковий» (англ. autoreport: tabular). У якості джерела даних цього звіту оберіть таблицю «**ПІБ-Викладач**». Завершіть створення автозвіта виведенням на екран звіту, в якому в якості заголовка вказано ім'я таблиці. У звіт включено всі поля таблиці і в якості заголовків стовпців обрано підписи полів, задані у властивостях таблиці. Результат відобразіть в протоколі.

6. Доопрацювання підлеглого звіту в режимі конструктора. Для того щоб підлеглий звіт можна було додати до основної частини звіту в потрібному вигляді, допрацюйте його в режимі конструктора.

Верхній колонтитул, в якому після роботи майстра виявляються розміщеними написи полів, і нижній колонтитул, де розміщуються дата і номер сторінки, не відображаються у разі вбудовування звіту як підлеглого.

Для відображення в повному звіті заголовків стовпців табличної частини перенесіть назви стовпців з верхнього колонтитулу в заголовок підлеглого звіту. Для цього розширте розділ заголовка звіту, виділіть всі підписи полів у верхньому колонтитулі і перетягніть їх. Видаліть обидва колонтитула, натиснувши команду «Колонтитули», оскільки звіт буде використовуватися

тільки як підлеглий. Видаліть елемент поля **pibdept** і його напис у підлеглому звіті, оскільки значення цього поля буде повторюватися в усіх рядках викладачів однієї кафедри, а одноразове відображення коду кафедри передбачено в основній частині звіту.

Збережіть відредагований підлеглий звіт з іменем «**Прізвище**–Звіт04». Остаточний сконструйований підлеглий звіт представте у протоколі.

7. Додавання підлеглого звіту та доопрацювання двохтабличного звіту. Додайте підлеглий звіт в основний перетягуванням. Розмістіть вікно бази даних поряд з вікном звіту «**Прізвище**–Звіт03» у режимі конструктора. Перейдіть у вікно бази даних. Оберіть з переліку імен звітів підлеглий звіт «**Прізвище**–Звіт04» і перетягнемо його в область даних звіту «**Прізвище**–Звіт03». Видаліть елемент напису підлеглого звіту. Відобразіть в протоколі звіт «**Прізвище**–Звіт03» після впровадження підлеглого звіту в режимі конструктора. Збережіть цей звіт з іменем «**Прізвище**–Звіт05». Для того щоб підлеглий звіт не було обрамлено, виділіть підлеглий звіт, відкрийте його властивості і на вкладці «Макет» у рядку «Тип границі» оберіть значення «Прозорий».

СКБД автоматично встановить зв'язок між основним і підлеглим звітом за полем **pibdept**, тому що так визначено в схемі даних. У такому разі немає значення, чи включено поле зв'язку в підлеглий звіт. MS Access працює з усіма полями таблиці, на основі якої побудовано звіт. Побачити цей зв'язок можна у властивостях підлеглого звіту на вкладці «Дані» у рядках «Підлегли поля» та «Основні поля».

Для додавання підлеглого звіту іншим способом можна скористатись Командою «Підпорядкована форма / звіт» у разі вимкненої команди майстра елементів. У такому разі буде створено елемент «Вільний» з написом «Дочірній». У властивостях елементу «Вільний» на вкладці «Дані» у рядку «Об'єкт-джерело» потрібно обрати звіт «**Прізвище**–Звіт04».

Задайте критерій сортування записів для їх виведення у звіті в потрібному порядку. Для цього в режимі конструктора натисніть команду в області виділення звіту і викличте контекстно-залежне меню. У меню оберіть рядок «Сортування і групування» та в діалоговому вікні вкажіть поле сортування.

Перегляньте вміст двохтабличного звіту «**Прізвище**–Звіт05», натиснувши команду «Попередній перегляд». Виведіть звіт на друк, додайте до протоколу.

8. Побудова майстром багатотабличного звіту без підлеглої частини. У звіті повинні виводитися у вигляді списку дані про заняття, що проводяться в кожній групі з деталізуванням найменування дисциплін та прізвищ викладачів. Рядки звіту повинні бути впорядковані за кодом предмета. У разі виведення даних у звіті повинні також формуватися розрахункові сумарні години занять у кожній групі.

Основні відомості про заняття, що проводяться в групах, містить таблиця «**ПІБ-Навчання**». З таблиці «**ПІБ-Навчання**» можна отримати перелік

ідентифікаторів занять, що проводяться в кожній групі, із зазначенням тривалості занять. Ця таблиця є основним джерелом записів для створюваного звіту. Назви дисциплін містить таблиця «**ПІБ-Дисципліна**», а прізвища викладачів – таблиця «**ПІБ-Викладач**».

Оберіть у вікні «Новий звіт» режим створення «Майстер звітів» і підпорядковану таблицю «**ПІБ-Навчання**» в якості записостворювальної таблиці для звіту. У вікні «Створення звітів» оберіть з таблиці «**ПІБ-Навчання**» поля, які міститимуться у звіті – **pibcode**, **pibtype**, **pibdisnumber**. З таблиці «**ПІБ-Дисципліна**» оберіть поле з назвами дисциплін **pibdistitle**, а з таблиці «**ПІБ-Викладач**» поле **pibsurname**. Відобразіть в протоколі вікно «Створення звіту» після вибору всіх полів, використуваних у звіті.

Для того щоб додати поля в потрібній послідовності, слід мати на увазі, що нове поле буде додано за виділеним у списку полів, вже включених до звіту. Наприклад, для включення поля **pibdistitle** після **pibcode** останнє повинно бути попередньо виділено.

Далі у вікні майстра в рядку «Виберіть тип представлення даних» оберіть таблицю «**ПІБ-Навчання**», оскільки її було обрано в якості джерела записів. Вкажіть рівень групування за полем **pibcode**. У вікні майстра відобразиться загальна структура макета звіту. Далі у діалоговому вікні майстра створення звіту оберіть поле **pibdistitle**, за яким потрібно виконати сортування за зростанням. Майстер дає змогу проводити сортування за чотирма полями в порядку зростання або спадання значень.

Для обчислення підсумкових значень у полі звіту «Разом за групою» у підсумковому рядку групи, натисніть команду «Підсумки» (англ. summary options). Оберіть функцію Sum для поля **pibdisnumber**, щоб розрахувати сумарну кількість годин.

У вікні майстра оберіть вид макета звіту ступінчастий (англ. stepped) з книжковою орієнтацією. Встановіть прапорець «Налаштувати ширину полів для розміщення на одній сторінці». Далі оберіть один з можливих стилів оформлення, наприклад, «Стиснутий» (англ. compact). Вкажіть назву звіту «**Прізвище–Вивчення предметів у групах**», яке також відобразиться в заголовку звіту. Під цим ім'ям майстер автоматично зберігає звіт в базі даних. Для того щоб відразу доопрацювати звіт у режимі конструктора оберіть подальшу дію – «Змінити макет звіту». Збережіть звіт з іменем «**Прізвище–Звіт06**» та представте у протоколі.

9. Перегляд запиту, який формує майстер для створення багатотабличного звіту. На основі інформації, що зберігається в схемі даних, і заданих користувачем полів у разі створення макета звіту майстер будує необхідний запит. За цим запитом формуються записи з полів кількох взаємозалежних таблиць.

Відкрийте вікно властивостей звіту «**Прізвище–Звіт06**». У рядку «Джерело записів» відображається створена майстром інструкція SQL, що

реалізує запит вибірки заданих полів з різних таблиць для формування звіту. Перегляньте запит, створений майстром у вигляді запиту QBE. Для цього в діалоговому вікні властивостей звіту в рядку «Джерело записів» оберіть будівник, у вікні якого відображається схема даних та бланк запиту. Відобразіть список полів, доступних у звіті, якщо у властивостях звіту як джерело записів записана інструкція SQL. Для цього натисніть на панелі конструктора звітів команду «Список полів». Отриманий результат представте у протоколі.

10. Редагування макета звіту в режимі конструктора. Відредагуйте написи полів у звіті «*Прізвище–Звіт06*», побудованому майстром: для поля номера групи **pibcode** в заголовку групи; для поля підрахунку кількості записів і поля підрахунку суми годин групам. Перемістіть поле, яке визначає поточну дату з виразом = Now (), поле номера сторінок звіту з нижнього колонтитулу в заголовок звіту. Для того щоб заголовки друкувалися не на кожній сторінці звіту, а для кожної групи, перемістіть їх з верхнього колонтитулу в заголовок групи. Щоб відомості про кожну групу друкувалися на окремій сторінці вставте в примітки для групи розрив сторінки, скориставшись командою «Розрив сторінки» конструктора звітів. Відредагуйте текст заголовків відповідно до проекту макета. Після доопрацювання звіту в режимі конструктора перейдіть в режим перегляду. Отриманий результат представте у протоколі.

У звітах можна виводити значення деякого поля запису або підсумкового поля групуванням підсумком з накопиченням. Наприклад, можна накопичувати суму годин від групи до групи. Властивість «Сума з накопиченням» (англ. running sum) розміщено на вкладці «Дані».

11. Створення запиту для звіту. Розглянемо створення звіту з використанням запиту на прикладі формування бланку «Екзаменаційна відомість» для певної групи студентів із вказаної дисципліни. Розглянемо підготовку запиту, що забезпечує вибірку інформації, необхідної для формування бланку екзаменаційної відомості.

Для виведення в заголовок бланку екзаменаційної відомості значень параметрів виду та групи необхідні дані з полів **pibtype** та **pibcode** таблиці «*ПІБ-Навчання*», для виведення значень параметрів з загальним написом «Викладач» (прізвище, вчене звання) необхідні дані з полів **pibsurname** та **pibtitle** таблиці «*ПІБ-Викладач*», для виведення параметру «Назва дисципліни» - **pibdistitle** таблиці «*ПІБ-Дисципліна*», для додавання в табличну частину бланку значень реквізитів № п/п, Прізвища – полів **pibstudent** та **pibsurname** таблиці «*ПІБ-Студент*». Таким чином, запит для вибірки цих взаємопов'язаних даних необхідно побужувати на основі таблиць «*ПІБ-Навчання*», «*ПІБ-Викладач*», «*ПІБ-Дисципліна*» та «*ПІБ-Студент*».

Створіть запит на вибірку і додайте в схему даних запиту вказані таблиці. При цьому зв'язки типу 1:М між таблицями встановляться автоматично відповідно до схеми даних БД. Між таблицями «*ПІБ-Студент*» і

«ПІБ-Навчання» автоматично встановиться зв'язок-об'єднання за однойменним полем **pibcode**. При цьому записи з цих таблиць об'єднуються і додаються в результат тільки в тому випадку, якщо пов'язані поля містять однакові значення.

У бланк запиту додайте необхідні для звіту поля з таблиць, представлені в схемі даних запиту. У рядку бланка запиту «Умова відбору» вкажіть параметрами запиту номер групи і назву дисципліни для діалогового введення їх значень у разі виконання запиту. Це дасть змогу отримати дані для конкретної відомості.

Для виведення у звіті значень параметру виду підсумкової атестації (наприклад, екзамен або залік), яких немає безпосередньо в таблицях БД, передбачте обчислюване поле, в якому будуть формуватися вказані значення відповідно можливих значень поля **pibtype** (наприклад, лабораторні та практичні) таблиці «ПІБ-Навчання». Запишіть в обчислюване поле вираз, у якому сформуєте вбудовану функцію керування

Вираз 1: `iif ([pibtype] J = "лекції"; "екзамен"; "залік")`

Відповідно до формату функції, якщо виконається умова **[pibtype] = "лабораторні"** (тобто в полі **pibtype** знаходиться значення «лабораторні»), то результатом функції буде «екзамен». В іншому випадку, тобто при **[pibtype] = "практичні"**, результатом функції буде «залік». Збережіть запит з іменем «Прізвище-Запит22». Виконаємо запит. Результати виконання запиту з даними для формування звіту представте в протоколі.

15. Конструювання звіту на основі запиту. Розглянемо створення звіту на основі запиту «Прізвище-Запит22». Оберіть у вікні бази даних об'єкт «Звіти» і натиснемо команду «Створити». У вікні «Новий звіт» оберіть запит «Прізвище-Запит22», який буде джерелом даних для звіту, і рядок «Конструктор». Щоб забезпечити виведення окремої відомості для кожного виду занять передбачте групування за полем **pibtype**. Для цього натисніть команду «Сортування й групування» конструктора звітів і заповніть поля. Після визначення групування у вікні конструктора звіту з'являються розділи «Тема групи **pibtype**» (англ. header) і «Примітка групи **pibtype**» (англ. footer). Щоб кожне групування могло бути оформлене як самостійна екзаменаційна відомість, потрібно формувати заголовок макету відомості в заголовку групи **pibtype**, а рядки для підведення підсумків здавання екзамену або заліку в примітці групи **pibtype**. Створіть в заголовку та примітці групи **pibtype** необхідні текстові елементи, за винятком назв параметрів полів. Для цього скористайтесь командою «Напис».

Виконайте розміщення полів в табличній частині звіту на основі запиту-джерела для формування даних звіту. Викличте вікно списку полів запиту «Прізвище-Запит22» натиснувши команду «Список полів». В області даних розмістіть поля **pibstudent** та **pibsurname** студента, на основі вмісту яких формуватимуться рядки табличної частини звіту відповідно до проекту (для цього зі списку полів перетягніть поля **pibstudent** і **pibsurname**). Встановіть

інший шрифт в елементах. Для формування заголовка стовпців перемістіть підпис кожного поля з табличної частини в розділ «Тема групи **pibtype**», додайте підпис до цього розділу. Оформіть інші елементи, наприклад, замініть підпис поля «Номер студента», щоб у бланку відомості з'явилися стовпці для оцінок і підпису, додайте у рядок із заголовками стовпців текстові елементи «Відмітка про здавання» і «Підпис викладача».

Розмістіть поля в заголовку. Розмістіть поля **pibcode**, **pibdistitle**, **pibsurname** викладача, **pibtitle** та Вираз 1 в заголовку групи, щоб їхні значення повторювалися у всіх записах групування. Для цього перетягніть поля зі списку полів запиту розділ звіту «Тема групи **pibtype**».

Додайте до звіту поточну дату за допомогою вбудованої функції Now (). Для цього створіть незв'язаний елемент, натиснувши кнопку «Текст» і розмістіть його. Задайте у вікні властивостей на вкладці «Дані» у рядку «Дані» вираз = Now (), а на вкладці «Макет» у рядку «Формат поля» оберіть значення «Середній формат дати» (англ. medium date).

Додайте номер сторінки в розділ нижнього колонтитула. Для цього створимо незв'язаний елемент і заповнимо в його властивостях рядок «Дані» іразом = "Сторінка" & [Page].

Створіть горизонтальні і вертикальні лінії, скориставшись командою «Лінії». Встановимо потрібну товщину ліній у властивостях цього елемента.

Збережіть звіт під ім'ям «**Прізвище–Звіт07**».

16. Попередній перегляд звіту з введенням параметрів. Під час підготовки до перегляду звіту, побудованого на базі запиту з параметрами, MS Access попередньо виконує запит і виводить діалогові вікна введення параметрів звіту.

Для відображення звіту «**Прізвище–Звіт07**» в тому вигляді, в якому він буде надрукований, перейдіть з режиму конструктора звіту, натиснувши команду Вигляд. Введіть значення для першого параметра запиту номеру групи в діалоговому вікні, наприклад, «ХА-01». Введіть значення «Інформатика» другого параметра запиту назви предмету в іншому діалоговому вікні.

Результати представте в протоколі.

Оформіть протокол лабораторної роботи.

Оформлення протоколу

Після оформлення протоколу лабораторної роботи до задалегідь підготовлених складників додаються роздруковки скріншотів ходу виконання завдань.

Контрольні запитання

1. З яких розділів складається звіт?
2. Як переглянути список доступних в конструкторі звітів полів і вставити потрібне поле в звіт?

3. Де доцільно розміщувати значення поля, за яким проводиться групування?
4. Чи можна за допомогою перетягування включити у звіт підлеглий звіт?
5. Чи потрібно встановлювати зв'язок між головним і підлеглим звітом, якщо його визначено в схемі даних?
6. У якому рядку властивостей звіту вказується джерело даних?
7. На основі яких об'єктів майстер будує звіт?
8. Чи можливий вибір полів з різних таблиць і запитів у разі побудови звіту майстром?
9. Скільки рівнів групування може визначити майстер у звіті?
10. Як переглянути створену майстром інструкцію SQL у вигляді запиту QBE?
11. Яка властивість поля запису або підсумкового поля потрібно використовувати, щоб виводити його значення наростаючим підсумком?
12. За рахунок чого повторювані значення в таблиці запиту можуть бути відображені у звіті лише один раз?
13. Чи буде при перегляді звіту виводитися діалогове вікно введення параметру, якщо звіт створено на основі запиту з параметром?
14. Як переглянути список полів, доступних в режимі конструктора звіту?
15. Які поля містить цей список, якщо звіт побудований на запиті?

Лабораторна робота № 7

Створення макросів в СКБД MS Access

Метою роботи є опанування прийомів створення макросів для послідовного виконання запитів, синхронного оброблення даних форм, переходу від діалогової роботи з базою даних до створення звітів, а також організування діалогу в макросі та керування представленням елемента в формі.

Завдання. Розглянути засоби створення і редагування макросів у СКБД MS Access 2016.

Короткі теоретичні відомості

Використання макросів для автоматизації керування реакцією застосунку на дії користувача у формах і звітах дає змогу створювати повноцінні інтерактивні застосунки без написання коду на VBA.

Макроси забезпечують виконання частини команд, доступних у VBA, а створення макросу для користувача є простішим ніж написання коду VBA. Конструктор макросів володіє зрозумілим інтерфейсом і дає змогу істотно спростити процес додавання функціональних можливостей у застосунок бази даних.

Макрос – це програма, яку утворено з послідовності макрокоманд. Макрокоманда – це інструкція для виконання певної дії на об'єктами та

їхніми елементами. У MS Access є близько 50 макрокоманд, які реалізують практично будь-які дії, необхідні для вирішення задач: відкривання форми або звіту, друк звіту, запуск запиту на виконання, застосування фільтру, присвоєння значення, створення меню, тощо.

Макрос може бути представлено як окремий об'єкт, відображуваний в області навігації у групі «Макроси». Такий макрос має назву ізолюваного. Також макрос може бути пов'язано з подією форми, звіту або елементу керування і тоді його впроваджують у форму або звіт. Такий макрос має назву впровадженого. Такий макрос не відображається як об'єкт у групі «Макроси», а стає складником форми або звіту.

Ізолювані макроси можна запускати на виконання з області навігації. Можливе вирішення задач за допомогою ланцюга взаємопов'язаних макросів, перший з яких запускається на виконання з області навігації. Користувач запускає головний макрос на виконання, а далі усе керування виконанням задачі здійснюється зсередини макросу: макрос відкриває потрібні об'єкти, вибирає і обробляє дані, викликає інші макроси, тощо. За необхідності з макроса можна ініціювати діалог з користувачем. Для переходу за різними гілками макросу використовують блок керування «Якщо». Ізолюваний макрос може виконуватись у відповідь на численні види подій, що виникають у формах, звітах та їхніх елементах керування.

Впроваджений макрос завжди пов'язується з подією і зберігається у формі або звіті. Зв'язок макросів з подіями дає змогу автоматизувати застосунки за допомогою використання макросів для відкриття форм, друку звітів, виконання послідовності запитів, для виконання дій, які залежать від значення певного поля у базі даних, для виведення користувацьких повідомлень або відключення попереджувальних повідомлень під час виконання запитів дії, тощо. Збереження впроваджених макросів разом з формами і звітами спрощує керування об'єктами застосунку.

Створення макросів здійснюється у діалоговому режимі і зводиться до запису у вікні конструктора послідовності макрокоманд, для яких задають аргументи. Кожному макросу присвоюється ім'я. У разі виконання макросу команди виконується послідовно у порядку їх розташування, обробляються дані та об'єкти, вказані в аргументах макрокоманд.

Макроси даних призначено для додавання логіки до базових таблиць. Макроси даних зв'язуються з подіями у базовій таблиці і виконуються у разі змінювання, додавання або видалення запису. Такі макроси дають змогу перевіряти дані і виконувати обчислення. Прив'язуючи логіку до даних і зосереджуючи її у базових таблицях, макроси даних забезпечують доступ до неї з об'єктів застосунку. Так, виконання операцій їх записами у формах, побудованих на базових таблицях, ініціює виконання макросів даних.

Макроси даних дають змогу реалізувати у базі даних функції, які у серверних базах даних реалізовано за допомогою особливого виду збережуваних процедур – тригерів. Тригер також зв'язують з певною операцією у базовій таблиці та викликають щоразу у випадку виконання цієї операції.

За допомогою макросів даних, як і тригерів, можуть виконуватись дії із забезпечення реляційної цілісності даних, які не підтримують стандартні засоби. Макрос даних немає ні параметрів, ні значення, яке повертається.

У разі програмування у застосунку отримати доступ до даних бази можна через форми і звіти. У макросах даних, на відміну від макросів, прив'язаних до подій в об'єктах застосунку, для доступу до даних використовуються посилання на поля таблиць. Звернення до даних через поля об'єктів застосунку не допускається.

Крім вбудованих макросів даних можна створювати іменовані макроси даних. Ці макроси даних можуть виконуватись тільки у разі виклику з інших макросів даних. В іменованому макросі можна вказувати параметри. Значення параметрам присвоюється у макроси, що викликає.

Послідовність виконання роботи

1. Макрос для послідовного виконання запитів. Розглянемо основні прийоми створення макросу на прикладі послідовного виконання запитів. Макрос запускатиметься з вікна бази даних.

Нехай необхідно підрахувати кількість студентів у групах і внести ці дані в поле **pibnumber** таблиці «**ПІБ-Група**». Підрахунок кількості студентів реалізовано запитом «**Прізвище-Запит14**» (див. п. 3.1 послідовності виконання лабораторної роботи №4). Оновлення поля **pibnumber** на основі отриманих в цьому запиті даних виконує запит оновлення «**Прізвище-Запит15**» (див. п. 3.2 послідовності виконання лабораторної роботи №4). Поставлену задачу можна вирішити шляхом послідовного виконання таких запитів. Створіть макрос, який послідовно виконує ці запити. Додайте у макрос макрокоманди, які автоматизують дії користувача з керування об'єктами під час розв'язання задачі та, зокрема, дії, що забезпечують відображення результатів вирішення завдання у зручному вигляді.

Для створення макросу у вікні бази даних, оберіть команду «Макрос» на вкладці «Головна», відкриється вікно макросу. Дії, що утворюють макрос, необхідно вводити у стовпець макрокоманд. Для введення макрокоманди натисніть кнопку розкриття списку макрокоманд в цьому стовпці і оберіть «Відкрити запит», щоб виконати перший запит. У розділі аргументів макрокоманди оберіть зі списку ім'я запиту «**Прізвище-Запит14**» в нижній частині вікна макросу. У стовпець приміток введіть текст, що описує виконання макрокоманди, наприклад, «Створення таблиці з результатами підрахунку кількості студентів в групі».

Для виконання запиту оновлення «**Прізвище-Запит15**» введіть макрокоманду «Відкрити запит». Порядок розміщення макрокоманд у бланку визначає послідовність їх виконання. Для додавання в макрос цієї макрокоманди скористайтесь перетягуванням об'єктів за допомогою миші. У вікні бази даних оберіть запит «**Прізвище-Запит15**» і перетягніть його в рядок макрокоманди. Сформується макрокоманда відкривання запиту, в

аргументах якої автоматично з'явиться ім'я запиту, режим відображення запиту і режим роботи з його даними.

Збережіть запит під ім'ям «*Прізвище-Макрос01*». Запустіть макрос за допомогою команди «Виконати» вкладки «Конструктор». Під час виконання макросу на екран виводяться повідомлення, пов'язані з виконуваними запитом. Для тимчасового відключення виведення вікон цих повідомлень доповніть макрос макрокомандою «Встановити повідомлення». Пересвідчіться, що команда «Показати усі дії» у групі «Показати або приховати» вкладки «Конструктор» є активною. У рядку аргументу макрокоманди «Встановити повідомлення» оберіть значення «Ні». Для відновлення виведення повідомлень після виконання запитів можна задати значення «Так». Однак цю макрокоманду можна не виконувати, оскільки після припинення роботи макросу виведення попереджень відновлюється автоматично.

Відкрийте таблицю «*ПІБ-Група*», щоб зробити наочним процес роботи макросу і оновлення таблиці. Закрийте макрос і запустіть його з вікна бази даних, скориставшись командою «Виконати». Хід і результати виконання запиту з оновлення поля **pibnumber** відображаються у вигляді повідомлень і таблиць.

Створіть на будь-якій формі кнопку запуску макросу. Для цього у вікні бази даних виберіть макрос «*Прізвище-Макрос01*», який необхідно запускати за допомогою натискання кнопки. Перетягніть його в форму, відкриту в режимі конструктора. На форму буде додано кнопку, пов'язану з цим макросом, що має підпис який відповідає імені макросу.

Відобразіть у протоколі макрос в режимі конструктора. Коротко опишіть у протоколі виконувані дії. Закрийте макрос. За допомогою команди «Перетворити макроси» отримайте створений макрос у вигляді коду VBA і представте у протоколі. Примітки і обробники помилок під час перетворення макросу можна не додавати.

2. Макрос синхронного оброблення даних двох форм. Розглянемо створення групи макросів, призначених для фільтрування записів в одній формі, які будуть запускатися у раз ініціювання користувачем події в іншій формі.

Нехай під час переглядання даних про дисципліну, що вивчається студентами, необхідно відображати інформацію про всіх студентів, які отримали «незадовільно» з цієї дисципліни. Для вирішення задачі використовуємо форми «*Прізвище-Форма01*» і «*Прізвище-Аутсайтери*». Однотабличну однозаписову форму «*Прізвище-Форма01*» вже створено і вона дає змогу переглядати інформацію щодо дисциплін. Форму «*Прізвище-Аутсайтери*» створіть на основі запиту, який з таблиць «*ПІБ-Успішність*» і «*ПІБ-Студент*» відбирає поля **pibdiscode**, **pibsurname**, **pibrating**, у якому для поля **pibrating** в рядку умови відбору записано значення менше 75 балів (або інше). Форма відобразить список всіх записів про низькі рейтинги, які мають студенти з усіх предметів. Обидві форми мають поле **pibdiscode**, що містить назву дисципліни. Для вирішення задачі необхідно створити групу

макросів з ім'ям «Прізвище-Аутсайтери», яка міститиме макроси «Прізвище-Макрос02» і «Прізвище-Макрос03». Нехай макрос «Прізвище-Макрос02» виконується у разі відкриття форми «Прізвище-Форма01», відкриває форму «Прізвище-Аутсайтери» і визначає її місце розташування на екрані. Нехай макрос «Прізвище-Макрос03» фільтрує записи про студентів за значенням поля **pibdiscode**, взятому з поточного запису форми. Макрос повинен виконуватися тоді, коли користувач, працюючи в формі «Прізвище-Форма01», ініціює подію активації поля. Створення груп макросів дає змогу об'єднувати макроси, призначені для вирішення однієї задачі, і спростити супровід програми.

Для створення макросу у вікні бази даних, оберіть команду «Макрос» на вкладці «Головна», відкриється вікно макросу. Щоб доповнити вікно макросу стовпцем, який дає змогу створити групу макросів, активуйте опцію «Імена макросів» вкладки «Конструктор».

Запишіть імена, макрокоманди і аргументи⁸ макросів «Прізвище-Макрос02», «Прізвище-Макрос03» у макрокоманди і збережіть групу макросів під ім'ям «Прізвище-Аутсайтери». Це ім'я буде виведено в списку макросів у вікні бази даних.

Для запуску макросу «Прізвище-Макрос02» встановіть зв'язок події «Відкриття» форми «Прізвище-Форма01» з цим макросом. Відкрийте властивості форми в режимі конструктора і на вкладці «Події» для властивості «Відкриття» оберіть значення «Прізвище-Аутсайтери. Прізвище-Макрос02».

Для запуску макросу «Прізвище-Макрос03» встановіть в якості значення властивості «Подвійне натискання кнопки» елемента керування **pibdiscode** форми «Прізвище-Форма01» значення «Прізвище-Аутсайтери. Прізвище-Макрос03». Подія «Подвійне натискання кнопки» для елемента керування **pibdiscode** виникає, якщо користувач двічі швидко натискає ліву кнопку миші в той момент, коли курсор миші встановлено на поле **pibdiscode** або приєднаного до нього підпису.

Щоб перевірити роботу макросів, відкрийте форму «Прізвище-Форма01». На екрані буде відображено результат роботи макросу «Прізвище-Аутсайтери. Прізвище-Макрос02» – відкриті форм «Прізвище-Аутсайтери» і «Прізвище-Форма01». Причому в формі «Прізвище-Аутсайтери» відобразяться записи про невдачі з усіх предметів.

Виконайте подвійне натискання на поле з кодом предмета в формі «Прізвище-Форма01». Результат виконання макросу «Прізвище-Макрос03» відобразиться на екрані. До форми «Прізвище-Аутсайтери» було застосовано фільтр, в якому записано умову відбору [**pibdiscode**] = [Формы]! [Прізвище-Форма01]! [**pibdiscode**], і тепер в ній відображаються тільки записи зі значенням коду дисципліни з форми «Прізвище-Форма01».

⁸ Наприклад, ім'я макросу – «ПрізвищеМакрос02»; макрокоманда – «Відкрити форму»; аргумент – «Прізвище-Форма01» або ім'я макросу – «ПрізвищеМакрос03»; макрокоманда – «Застосувати фільтр»; аргумент – «[discode]=[Формы]! [Прізвище-Форма01]! [discode]» тощо.

Щоб макрос «*Прізвище-Макрос03*» виконувався у разі якщо користувач, працюючи в формі «*Прізвище-Форма01*», активує поле **pibdiscode**, видаліть зв'язок макросу «*Прізвище-Макрос03*» з подією «Подвійне натискання кнопки» для елемента управління **pibdiscode**. Встановіть його зв'язок з подією «Вхід» цього елемента керування.

Перегляньте послідовність полів в формі «*Прізвище-Форма01*» за допомогою команди «Послідовність переходу» в контекстному меню елемента керування форми. Якщо поле **pibdiscode** розташовано на першому місці, розмістіть його нижче. Відкрийте форму «*Прізвище-Форма01*» і переконайтеся, що при переході від запису до запису синхронно відображаються відповідні записи форми «*Прізвище-Аутсайтери*».

Коротко опишіть у протоколі виконувані дії. Закрийте групу макросів. За допомогою команди «Перетворити макроси» отримайте створені макроси у вигляді коду VBA і додайте у протокол.

3. Макрос переходу від діалогової роботи з базою даних до звітування. Нехай необхідно в сеансі введення даних про успішність підготувати звіти про отримані студентами бали більше 94 і менше 60. Для накопичення даних створіть таблиці «*ПІБ-Не зараховано*» і «*ПІБ-Відмінно*», які містять поля **pibcode**, **pibstudent**, **pibdiscode** і **pibsurname**. Ці таблиці повинні мати складений ключ **pibcode** + **pibstudent**. У цих таблицях неможливо повторно ввести записи з однаковим значенням ключа.

Для введення даних про результати іспитів створіть форму «*Прізвище-Відомість*». Основна частина форми відображатиме записи з таблиці «*ПІБ-Навчання*», доповнені пов'язаними даними з головних по відношенню до неї таблиць «*ПІБ-Дисципліна*» і «*ПІБ-Викладач*» (див. п. 9 послідовності виконання лабораторної роботи №5, форма «*Прізвище-Форма05*»). Створіть підпорядковану форму «*Прізвище-Успішність*», яка відображатиме пов'язані з таблицею «*ПІБ-Навчання*» записи з таблиці «*ПІБ-Успішність*», і додайте до форми «*Прізвище-Відомість*» за допомогою команди «Підпорядкована форма/звіт» з групи «Елементи керування» вкладки «Конструктор». Під час підпорядкування форми вкажіть поля зв'язку головної і підпорядкованої форм. Підсхему даних, на якій будується форма, можна визначити із загальної схеми даних.

Для вирішення завдання створіть групу макросів «*Прізвище-Списки успішності*», яка міститиме макроси:

- «*Прізвище-Заповнення*», яким заповнюють таблиці «*ПІБ-Не зараховано*» і «*ПІБ-Відмінно*» через відповідні однозаписеві форми;
- «*Прізвище-Звітування*», який роздруковуватиме відповідні звіти після завершення введення даних про успішність;
- «*Прізвище-Сеанс*», який відкриватиме форми «*Прізвище-Не зараховано*» і «*Прізвище-Відмінно*», а також очищатиме таблиці перед формуванням нових звітів. Це дасть змогу в черговому сеансі введення даних в таблицю «*ПІБ-Успішність*» для однієї або кількох груп студентів для одного або кількох предметів сформулювати звіти

для цих груп, а також використовувати макрос для коригування таблиці. Звіти будуть містити дані тільки за результатами одного сеансу введення.

У перший макрос з ім'ям «Прізвище-Сеанс» введіть макрокоманди відкривання форм, які використовуватимуться для заповнення таблиць «Прізвище-Не зараховано» і «Прізвище-Відмінно».

Безпосередньо за певною макрокомандою «ВідкритиФорму» введіть макрокоманду «ВиконатиКоманду» («ЗапускКомандиМеню»). За цією макрокомандою для поточного об'єкта виконуватиметься команда зі стрічки, задана аргументом макрокоманди, тобто виконання цієї макрокоманди еквівалентно вибору користувачем зазначеної команди на вкладках стрічки. Оберіть як аргумент «ВиділитиУсіЗаписи». Як аргумент другої макрокоманди «ВиконатиКоманду» («ЗапускКомандиМеню») виберіть «ВидалитиЗапис». Дії цих макрокоманд для відкритої форми рівносильні виконанню відповідних команд вкладки «Головна». Таким способом здійснюється видалення усіх записів в таблицях «ПІБ-Не зараховано» і «ПІБ-Відмінно».

Якщо форми не повинні відображатись на екрані введіть за макрокомандами «ВиконатиКоманду» («ЗапускКомандиМеню») макрокоманду «Згорнути», яка застосовується до вікна поточного об'єкта. Якщо є потреба відображати одночасно процес введення даних і заповнення таблиць для звіту, можна не згортати форми, а доповнити макрос макрокомандами, які визначатимуть розміщення об'єктів на екрані, щоб отримати зручне відображення всіх форм.

Для виконання цього макросу в момент, коли користувач відкриває форму «Прізвище-Відомість», у властивості форми в рядок події «Відкриття» введіть ім'я макросу «Прізвище-Списки успішності. Прізвище-Сеанс». У результаті цього макрос буде виконуватися один раз на початку сеансу введення даних.

Макрос «Прізвище-Заповнення» повинен виконуватися у разі оновлення поля «Оцінка» в записах форми «Прізвище-Відомість». Зв'яжіть цей макрос з властивістю події «ПісляОновлення» поля **pibrating**, увівши в рядок властивості ім'я цього макросу «Прізвище-Списки успішності. Прізвище-Заповнення».

Щоб почати формування записів про студентів, які отримали менше 60 балів, введіть макрокоманду «НаЗапис», яка дасть змогу зробити поточним новий запис у формі «Прізвище-Не зараховано». Вкажіть в аргументах макрокоманди: тип об'єкту – «форма», ім'я об'єкту – «Не зараховано», запис – «новий». Наступні макрокоманди повинні заповнити поля нового запису. Це дає змогу зробити макрокоманда «ЗадатиЗначення», яка задає полям нового запису форми «Прізвище-Не зараховано» значення з полів поточного запису форми «Прізвище-Відомість».

Аргумент «Елемент» макрокоманди визначає поле, якому повинно присвоюватися значення. А «Вираз» визначає поле, з якого обирають значення. Вираз не повинен починатися зі знака рівності. Для посилання на

поле або елемент керування у формі, з якої викликано макрос, досить вказати тільки ім'я елемента. Для посилання на інші об'єкти необхідно використовувати повний синтаксис. У аргументі «Елемент» вкажіть [Форми]![Не зараховано]![поле], де у першій макрокоманді вкажіть поле **pibcode**, в наступних **pibstudent**, **pibdiscode** і **pibsurname**. Скористайтесь будівником для формування складного посилання на елемент. Будівник можна викликати натисканням кнопки в правій частині рядка аргументу.

Оскільки нові значення вибирають з полів поточного запису підлеглої форми «**Прізвище-Успішність**», в аргументі «Вираз» макрокоманди «ЗадатиЗначення» можна вказати тільки імена цих полів (**pibcode**, **pibstudent**, **pibdiscode** і **pibsurname**, відповідно).

Запишіть аналогічний набір макрокоманд для формування запису про високу оцінку (форма «**Прізвище-Відмінно**»).

Дії формування записів в таблицях «**ПІБ-Не зараховано**» і «**ПІБ-Відмінно**» повинні виконуватись тільки в тому випадку, якщо в поле **pibrating** введено значення менше 60, а макрокоманди формування запису про відмінну оцінку – якщо в поле **pibrating** введено значення більше 94. Для аналізу оцінки, введеної у форму «**Прізвище-Відомість**», і в залежності від результату формувати новий запис в одній з таблиць «**ПІБ-Не зараховано**» або «**ПІБ-Відмінно**», організуйте виконання різних макрокоманд в залежності від результату перевірки умови.

Додайте у вікно макросу стовпець умов за допомогою відповідної команди вкладки «Конструктор». У рядок другої макрокоманди «НаЗапис» в стовпець умов додайте логічний вираз [**pibrating**]>94, а в інших макрокомандах цієї групи проставте крапки (...). Введені в стовпець умови дають змогу пропустити виконання цих макрокоманд, якщо введена оцінка відрізняється від вказаної. У рядок першої макрокоманди «НаЗапис» в стовпець умов введіть логічний вираз [**pibrating**]<60. Тепер якщо вираз прийме значення «істина», тобто у разі введення значення в поле **pibrating**<60 буде виконуватися ця макрокоманда і всі наступні до макрокоманди з новою умовою. Щоб у разі значення виразу «хиба» не виконувалася група макрокоманд, що формує запис таблиці «**ПІБ-Не зараховано**», вказуємо у стовпці умов цих макрокоманд три крапки (...).

Макрос «**Прізвище-Звітування**» повинен виконуватись у разі закриття користувачем форми «**Прізвище-Відомість**». Введіть у властивостях форми в рядку події «Закриття» ім'я цього макросу «**Прізвище-Списки успішності. Прізвище-Звітування**». Введіть в цей макрос макрокоманди «Закрити» для форм «**Прізвище-Не зараховано**» і «**Прізвище-Відмінно**», обравши в ньому для аргументу «Збереження» значення «Так». Виконання цих макрокоманд дасть змогу зберегти в таблиці останній сформований запис. Для виведення звітів на екран в режимі попереднього перегляду використовуйте макрокоманду «Відкрити звіт». Створіть звіти «**Прізвище-Не зараховано**» і «**Прізвище-Відмінно**». Імена звітів та режим їх відкриття вкажіть в аргументах макрокоманд (якщо немає необхідності переглядати звіт, можна відразу отримати друкований документ встановивши в аргументі режим

друку). Відкритий для перегляду звіт буде відображатися до тих пір, поки користувач не закриє його.

Для перевірки макросу відкрийте форму «*Прізвище-Відомість*»⁹, введіть нові записи, закрийте форму. На екрані відобразяться звіти «*Прізвище-Не зараховано*» і «*Прізвище-Відмінно*».

Коротко опишіть у протоколі виконувані дії, додайте відповідні скріншоти. Закрийте групу макросів. За допомогою команди «Перетворити макроси» отримайте створені макроси у вигляді коду VBA і додайте у протокол.

4. Організування діалогу в макросі. У попередньому завданні під час закриття форми «*Прізвище-Відомість*», через яку вводилися дані про оцінки за зданими студентами дисциплінами, було передбачено виконання макросу для безумовного представлення звітів про студентів-відмінників та студентів, яким дисципліну не зараховано. Доповніть макрос макрокомандами, які уможливають в діалозі з користувачем визначити необхідність виведення звітів. Використайте в якості діалогового вікна допоміжну форму, у якій створіть поле зі списком значень. Заповніть список іменами звітів, виконання яких може запросити користувач, а також передбачте відмову від виведення звітів.

У проколі представте форму «*Прізвище-Діалог*», в якій поле зі списком повинно мати назву «*ПрізвищеПолеЗіСписком*». Виділення макрокоманд відкриття звітів в окремий макрос «*Прізвище-Звіт1*» дає змогу користувачеві організувати роботу з формою «*Прізвище-Діалог*». У формі «*Прізвище-Діалог*» для елемента «*ПрізвищеПолеСоСписком*» виберіть в рядку властивості «Після оновлення» значення «*Прізвище-Списки успішності. Прізвище-Звіт1*». Відкрийте форму «*Прізвище-Відомість*», введіть кілька записів про оцінки, отримані студентами, зокрема, >90 та <60. Закрийте форму. З'явиться вікно форми «*Прізвище-Діалог*». Для перевірки працездатності макросу «*Прізвище-Звіт1*» виберіть в формі «*Прізвище-Діалог*» одне зі значень.

Для роздруківки звіту відкрийте форму «*Прізвище-Діалог*», оберіть потрібний звіт у списку, макрос «*Прізвище-Звіт1*» повинен вивести обраний звіт. Для виведення питання про необхідність друкувати звіти можна використати функцію MsgBox(). Функція MsgBox () виведе вікно, що містить повідомлення, дочекається поки користувач натисне кнопку і поверне значення, яке вкаже на яку кнопку він натиснув. Текст повідомлення вказують першим аргументом. Функція MsgBox() повертає значення «1» у разі натискання у вікні повідомлення кнопки «Так» і значення «2» у разі натискання кнопки «Ні».

5. Керування представленням елемента в формі. Нехай у формі «*Прізвище-Відомість*» під час перегляду даних про оцінки отримані студентами, якщо в поле **pibrating** уведено значення <60, повинен з'являтися текст «Не зараховано».

Створіть в примітці підпорядкованої форми елемент керування

⁹ Таблиці «*Прізвище-Не зараховано*» і «*Прізвище-Відмінно*» повинні містити принаймні по одному запису до виконання макросу.

«Напис ...» із значенням «Не допущено». Розміщення напису в примітці стрічкової форми відобразить одноразово. Причому буде забезпечено її зв'язок з поточним записом стрічкової форми. Властивості напису «Виведення на екран» (або «Видимий», англ. visible) задайте значення «Ні».

Створіть макрос, в якому властивості «Виведення на екран» буде присвоюватися значення «Так» у разі перегляду запису з оцінкою. Для цього додайте в макрос стовпець «Умова». В цей стовпець введіть логічний вираз **pibrating=50**. У стовпці Макрокоманда виберіть макрокоманду «ЗадатиЗначення». У рядок аргументу «Елемент» введіть посилання на властивість «Виведення на екран» елемента керування «Напис ...». У рядок аргументу «Вираз» введіть значення цієї властивості – «Так».

Щоб у разі переходу до запису з іншою оцінкою напис «не зараховано» ставав невидимим, введіть другу макрокоманду «ЗадатиЗначення». В умови цієї макрокоманди введіть $< > 2$, а в аргументі «Вираз» вкажіть «Ні» в якості значення властивості «Виведення на екран». Збережіть макрос під ім'ям «**Прізвище-Відображення**». Для зв'язку макросу з підлеглою формою «**Прізвище-Успішність**», у підпорядкованій формі в її властивості на вкладці «Події» в рядок «Поточна запис» запишіть ім'я макросу «**Прізвище-Відображення**», вибравши його зі списку. Перевірте працездатність макросу. Відкрийте форму «**Прізвище-Відомість**» і, переглядаючи записи, переконайтеся, що якщо значення поля оцінки менше 60, то з'являється напис «Не зараховано».

Коротко опишіть у протоколі виконувані дії. Закрийте макроси. За допомогою команди «Перетворити макроси» отримайте створені макроси у вигляді коду VBA і додайте у протокол. Оформіть протокол лабораторної роботи.

Оформлення протоколу

Після оформлення протоколу лабораторної роботи до заздалегідь підготовлених складників додаються роздруківки скріншотів ходу виконання завдань.

Контрольні запитання

1. Чим визначається послідовність виконання макрокоманд?
2. Яка макрокоманда дає змогу змінити поточний запис в об'єкті?
3. Яка макрокоманда дає змогу змінити значення поля запису?
4. Яка макрокоманда дає змогу виконати фільтрування записів?
5. Як записати посилання на елемент керування в формі?
6. Як організувати послідовне виконання кількох запитів дії?
7. Які аргументи має макрокоманда «ВідкритиЗапит»?
8. Яка команда дає змогу почати виконання макросу з вікна бази даних?
9. Як організувати виконання макросу у разі відкриття форми?
10. Як записати посилання на макрос, який включено в групу?

Лабораторна робота № 8

Створення сценарію JavaScript із взаємодією з веб-ресурсом

Мета роботи – набути уміння створення сценаріїв мовою JavaScript та застосовувати їх до веб-документів.

Завдання. Розглянути принципи і прийоми створення та застосування сценаріїв мовою JavaScript.

Короткі теоретичні відомості

JavaScript – високорівнева скриптова мова. Поруч з HTML та CSS, JavaScript – одна з ключових веб-технологій. Інтеграція HTML, CSS та JavaScript утворює динамічну мову розмітки гіпертексту, DHTML. Ці три складники DHTML пов'язані між собою об'єктною моделлю документу, DOM. DOM визначає логічну структуру документа і спосіб доступу та керування елементами, які містить документ. В об'єктній моделі будь-який документ можна представити у вигляді логічно деревовидної структури, елементами якої є об'єкти. У вузлах цієї структури знаходяться об'єкти, а не дані, зі всіма властивими об'єктам параметрами і поведінкою. Основне призначення реалізування DOM – уможливлення доступу і маніпулювання елементами документу за допомогою об'єктів і забезпечення взаємодії цих об'єктів. Реалізація DOM охоплює керування подіями, які також представлено у вигляді об'єктів. DOM надає прямий доступ до усіх елементів документу, а сумісно з моделлю подій, такий підхід дає змогу веб-браузеру обробляти дії користувача, виконувати вбудовані, впроваджені й зв'язані сценарії та динамічно змінювати вміст документу без перезавантаження.

Сценарій, скрипт (англ. script) – це одна або більше інструкцій JavaScript.

Обробник події (англ. event handler) – це атрибути html-дескрипторів вбудовані у документ, які визначають команду або функцію, що виконується у разі виникнення події. Наприклад, атрибут обробки події Click буде мати назву onClick.

Об'єкт – конструкція із властивостями, які є змінними JavaScript або іншими об'єктами.

Метод – це функція, що відноситься до об'єкту.

Синтаксис: Ім'я_об'єкту.Метод(список параметрів).

Властивість використовується для опису об'єкту. Властивість визначають шляхом присвоєння їй значення.

Синтаксис: Ім'я об'єкту.Властивість=значення.

Змінна. JavaScript підтримує чотири простих типи даних: цілий, дійсний, рядковий і логічний.

Літерал – абсолютне, «буквальне» значення даних відповідного типу.

Оператор виконує операцію над одним чи декількома операндами і змінними. Розрізняють унарні (або складені) і бінарні оператори.

Функція – вбудований або заданий користувачем набір операторів, що виконують певну задачу, тобто деяку сукупність інструкцій, і може повертати деяке значення.

Вираз – це комбінація змінних, літералів і операторів, у результаті обчислення якої отримуємо одне значення.

Інструкція – поодинокі дії, що завершується крапкою з комою. Інструкції розрізняють вбудовані та користувача. До вбудованих відносять оператори вибору `if...else` та `switch`, оператори циклу `for` та `while`, оператори роботи з об'єктами `new`, `for...in`, `with` та `this`.

Cookie – особливий об'єкт, що містить інформацію про статус клієнта, яка буде передана на сервер.

Ієрархія – кожен об'єкт, породжений вищим об'єктом. Наприклад, форма є об'єктом (`orderForm`) і одночасно властивістю такого об'єкту як документ: `document.orderForm`.

Сценарій, розміщений у дескрипторі `<script>...</script>`, може бути розташований у будь-якому місці документу, але за можливості слід розташовувати його у секції `<head>`. У такому випадку сценарій буде завантажено одразу після виклику документа браузером. Дескриптор `<script>` дає змогу також зв'язати сценарій із зовнішнім файлом, що містить програмний код на мові JavaScript і має розширення `*.js`, за допомогою атрибуту `src`. У якості значення параметру задається адреса ресурсу.

Розрізняють три групи об'єктів JavaScript: вбудовані об'єкти, об'єкти браузера і об'єкти пов'язані із дескрипторами мови HTML. До вбудованих об'єктів відносять такі:

DATE	– Містить інформацію про дату та час. Дає змогу встановити чи отримати день, місяць, рік або час.
MATH	– Забезпечує виконання математичних операцій. Властивостями об'єкту є математичні константи, а методами – математичні функції.
STRING	– Послідовність символів у подвійних чи одинарних лапках.
ARRAY	– Забезпечує роботу з масивами.
Об'єкти користувача	– Створення таких об'єктів здійснюється за допомогою функцій, що визначають властивості та методи цих об'єктів.

До об'єктів браузеру належить об'єкт **document**, що відповідає усьому HTML-документу. Також до них відносяться такі об'єкти як **navigator**, який містить загальну інформацію про браузер клієнта, об'єкт **history**, який містить інформацію про документи, що відкривались протягом певного сеансу роботи, та об'єкт **window**, що містить інформацію про вікно браузера. Ієрархію об'єктів JavaScript на стороні клієнта показано на рис. 22.

Під час завантаження документу браузер автоматично створює ряд об'єктів. Зокрема, створюються об'єкти, що зберігають інформацію про усі форми документу (масив **forms**), усі зображення (масив **images**), усі фрейми (масив **frames**), усі гіперпосилання (масив **links**) і усі мітки (масив **anchors**). Усі ці масиви об'єктів є властивостями об'єкту **document**, який відноситься до об'єктів браузеру.

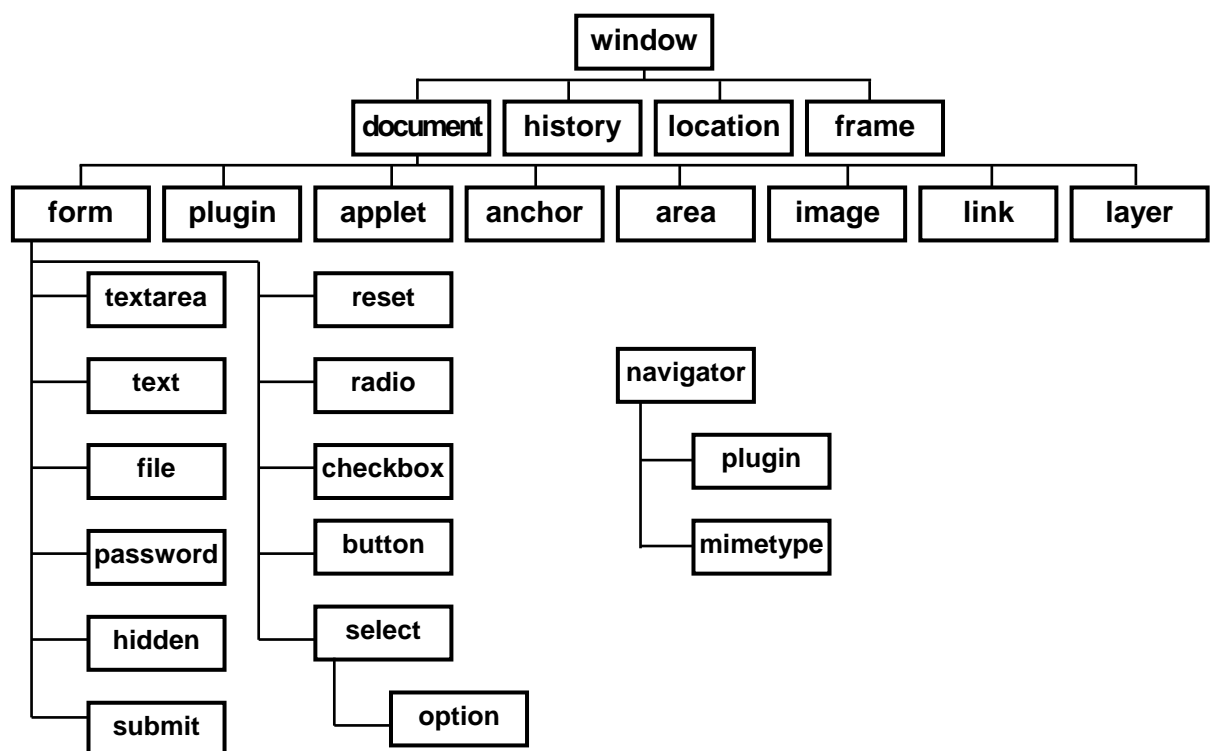


Рис. 22. Ієрархія об'єктів JavaScript

Деякі масиви і відповідні їм HTML-елементи об'єкту **document** такі:

all	усі елементи
anchors	
applets	<applet>, <object>
images	
forms	<form>
frames	<frame>, <iframe>
links	, <area href= >
scripts	<script>
stylesheets	стильові специфікації

Як і усі масиви у JavaScript, ці масиви об'єктів нумеруються починаючи з нуля, для них встановлені властивість **length**, тобто кількість елементів у масиві. Звернутись до елементу масиву можна за номером, наприклад

```
document.images[0];
```

або за іменем, якщо відповідний дескриптор має атрибут `name` або `id`, наприклад

```
document.images['malyunok'] або document.malyunok.
```

Властивостями об'єктів, пов'язаних із дескрипторами є атрибути відповідних дескрипторів. Код:

```
document.links[0].href='новий URL';
```

перепризначає значення першого гіперпосилання документа.

Іменувати можна не лише дескриптори `<form>`, `<frame>`, ``, `<link>` і `<a>`, але і інші дескриптори, наприклад: `<table>`, `<td>`, `<div>`, ``, тощо.

Як вже згадувалось усі дескрипторні об'єкти є властивостями об'єкту `document`, починаючи від адреси документа і закінчуючи кольором фону. Серед методів цього об'єкту інтерес представляє застарілий, але зручний і корисний метод `write`. За допомогою цього методу сторінки можна створювати динамічно. Синтаксис:

```
document.write(HTML-код);
```

Наприклад, такий сценарій дає змогу створити нове вікно із заданим текстом певного розміру:

```
<script language="JavaScript">
  <!-- hide
    function openWindow()
    {
      newWin= open("", "displayWindow",
        "width=500, height=400, status=yes, toolbar=yes, menubar=yes");
      newWin.document.open();
      newWin.document.write("<html><head><title>Нове вікно");
      newWin.document.write("</title></head><body>");
      newWin.document.write("<center><font size="+3>");
      newWin.document.write("Це новий HTML-документ, який було");
      newWin.document.write("створено за допомогою JavaScript!");
      newWin.document.write("</font></center>");
      newWin.document.write("</body></html>");
      newWin.document.close();
    }
  // -->
</script>
```


Для того, щоб сценарій було виконано, необхідно створити активатор цієї події, наприклад, кнопку з написом “Відкрити нове вікно”, у разі натискання на яку виконуватиметься сценарій:

```
<input type="button" value="Відкрити нове вікно" onClick="openWindow()">
```

де `onClick` – обробник події `openWindow`.

Обробники подій – це потужний інструмент мови JavaScript. Реагуючи на певні дії користувача, сценарій може перевіряти введення користувача і негайно реагувати на нього без зайвих звернень до сервера. Розрізняють оброблювачі подій миші, оброблювачі клавіатурних подій, оброблювачі фокусних подій та оброблювачі подій властивих окремим елементам документа. Обробка подій може виконуватись двома способами, один з яких – створення сценарію, що виконуватиметься у разі виникнення події. Інший – вказування у відповідному дескрипторі

Обробник події="перелік дій чи виклик функції".

Наприклад, як тільки документ буде завантажено почне виконуватись функція `cnange_bg`, а у разі вивантаження за допомогою методу `alert` з'явиться системне вікно із текстом «Сеанс завершено»:

```
<BODY onload="cnange_bg();" onunload="alert ('Сеанс завершено');"> ... </BODY>
```

Розглянемо найуживаніші оброблювачі подій.

Події миші

<code>onclick</code>	Натискання лівої кнопки миші
<code>ondblclick</code>	Подвійне натискання лівої кнопки миші
<code>onmouseover</code>	Наведення курсору миші на певну область
<code>onmouseout</code>	Виведення курсору миші із певної області
<code>onmousemove</code>	Переміщення курсору миші у межах певної області
<code>onmousedown</code>	Натискання будь-якої кнопки миші
<code>onmouseup</code>	Відпускання кнопки миші

Клавіатурні події

<code>onkeypress</code>	Натискання будь-якої клавіші крім системних
<code>onkeydown</code>	Натискання будь-якої клавіші
<code>onkeyup</code>	Відпускання клавіші
<code>onhelp</code>	Виклик підказки (F1)

Фокусні події

<code>onfocus</code>	Надбання фокусу
<code>onblur</code>	Втрата фокусу

Інші події

<code>onload</code>	Завантаження документа чи рисунку
<code>onunload</code>	Вивантаження документа
<code>onerror</code>	Виникнення помилки під час завантаження
<code>onreset</code>	Скидання значень введених користувачем за допомогою форми

onchange	Зміна значень поля форми
onsubmit	Передача даних введених через форму
onselectstart	Виділення фрагменту документу
ondragstart	Переміщення фрагменту
onfilterchange	Зміна фільтру
onscroll	“Прокрутка” документу у вікні
onresize	Зміна розмірів вікна браузера

Деякі уживані для створення сценаріїв JavaScript об'єкти, властивості та методи представлено у Додатку Г.

Послідовність виконання роботи

Створіть веб-сторінку змістовно наповнену з огляду на тему індивідуального завдання до курсової роботи з дисципліни обсягом від 3 до 5 тисяч знаків.

1. Вбудовування сценарію JavaScript. Додайте до одного з дескрипторів веб-сторінки вбудований сценарій JavaScript виклику вікна повідомлення, пов'язаний з подією миші. Текст у вікні повідомлення повинен містити Ваші ПІБ. Додайте до протоколу текст дескриптору та скріншот відображення результату виконання сценарію.

2. Впровадження сценарію JavaScript. Впровадьте у веб-сторінку сценарій JavaScript, який міститиме функцію **pib02()**, у результаті виклику якої з'являтиметься деяке користувацьке повідомлення. Виклик функції організуйте за допомогою події клавіатури. Текст у вікні повідомлення повинен містити Ваші ПІБ. Додайте до протоколу текст сценарію та скріншот відображення результату його виконання.

3. Динамічне додавання фрейму. До створеного впровадженого або зв'язаного сценарію додайте функцію **pib03()** генерування фрейму (iframe), який буде додано у поточний документ під час завантаження або у результаті дій користувача. Вміст нового елементу може бути завантажено як зовнішній веб-ресурс або створено за допомогою методу **write**.

Приклад 8.1. Фрейм

```
// вказування додавання елементу під час завантаження документу
window.onload = function() {
    // створення нового елементу – дескриптора iframe
    var miyframe = document.createElement("iframe");
    // вказування розмірів елементу
    miyframe.width = "200";
    miyframe.height = "200";
    // присвоєння властивості src дескриптора iframe значення (url)
    miyframe.setAttribute("src", "http://m.kxtp.kpi.ua/");
    // додавання елементу в кінці документу
    document.body.appendChild(miyframe);
}
```

У випадку, якщо необхідно додати власний вміст об'єкту-фрейму, можна скористатись методом `write`.

```
var miytext("<body>я – фрейм</body>");
miyframe.contentWindow.document.open();
miyframe.contentWindow.document.write(miytext);
miyframe.contentWindow.document.close();
```

Можна додати об'єкту-фрейм перед іншим елементом батьківського документа. Для цього варто скористатись методом `insertBefore`. Наприклад,

```
var elmnt = document.getElementById('imya');
elmnt.parentNode.insertBefore(miyframe, elmnt);
```

Додайте до протоколу лістинг функції та скріншот відображення результату її виконання.

4. Використання вікна введення `prompt()`. До створеного впровадженого або зв'язаного сценарію додайте функцію `pib04()` введення користувачем значення певної змінної, яке буде використано для змінювання веб-сторінки. Під час виконання завдання використовуйте свої ПІБ у будь-якому контексті.

Приклад 8.2. Змінювання веб-документа

```
<html>
<head>
<script type="text/javascript">
  function prompt82()
  {
    // виклик вікна введення
    var name=prompt("Уведіть ПІБ користувача веб-ресурсу",
      "Прізвище Ім'я По батькові")
    // перевірка вмісту вікна введення
    if (name!=null && name!="") {
      // створення нового елемента документа – дескриптора div
      let pib = document.createElement("div");
      // створення вмісту нового елемента документа
      pib.innerHTML = "<strong>Вітаємо</strong> "+name;
      // додавання нового елемента після елемента,
      // вказаного за допомогою атрибуту id у тілі документа
      korystuvach.after(pib);
    }
  }
</script>
</head>
<body>
  <a id="korystuvach" onclick="prompt82()">Користувач?</a>
</body>
</html>
```

Додайте до протоколу лістинг функції та скріншот відображення результату її виконання.

5. Об'єкт Date. За допомогою використання властивостей об'єкту `Date` до створеного впровадженого або зв'язаного сценарію додайте функцію `pib05()` додавання до веб-сторінки або представлення у вікні повідомлення поточних дати і часу під час завантаження документа або у результаті дій користувача. Додайте до протоколу лістинг функції та скріншот відображення результату її виконання.

6. Об'єкт Math. З використанням властивостей об'єкту `Math` до створеного впровадженого або зв'язаного сценарію додайте функцію `pib06()` розрахунку значення

$$x = [\cos(a) + \sin(b)] / [a^n] \text{ або іншого,}$$

де a та b – значення введені користувачем за допомогою вікон введення, n – ваш порядковий номер у списку студентів академічної групи.

Потрібно передбачити обробку не введення користувачем значень a та b . До отриманого значення x додайте випадкове число у діапазоні від 0 до n за допомогою методу `Math.random()`, який представлено математичною функцією генерування псевдовипадкового числа від 0 до 1. Результат розрахунків потрібно вивести в окрему вкладку браузера.

Приклад 8.3. Метод `open` об'єкту `window`

```
var vikno = window.open("about:blank", "Нове вікно");  
// створення нового елемента документа (дескриптора div)  
var ryadok = vikno.document.createElement('div'),  
// створення змінної-об'єкту (тіла документа)  
tilo = vikno.document.body;  
// створення вмісту нового елемента документа  
ryadok.innerHTML = "текст";  
// вказування додати рядок першим елементом тіла нового вікна  
tilo.insertBefore(ryadok, tilo.firstChild);
```

Додайте до протоколу лістинг функції та скріншот відображення результату її виконання.

7. Оператор `if..else`. Відповідно до варіанту індивідуального завдання (Додаток Д) розробити та налагодити розгалужений алгоритм і функцію без аргументів `pib07()` з використанням умовного оператора. Додати функцію до створеного впровадженого або зв'язаного сценарію. Організувати виклик функції за вказівкою користувача. Значення змінних a , b та t можна вказати як константи або передбачити введення їх користувачем. Результат можна надати користувачу будь-яким способом. Додайте до протоколу лістинг функції та скріншот відображення результату її виконання.

8. Оператор `for`. Відповідно до варіанту індивідуального завдання (див. Додаток Д) розробити і реалізувати циклічний алгоритм розрахунку суми або добутку. Передбачити введення користувачем початкового і кінцевого

значень лічильника циклу. Під час організування циклу передбачити виведення у нове вікно браузера значення лічильника циклу, значення доданка (співмножника) і поточного значення суми (добутку) на кожному кроці. Якщо знаменник доданка (для співмножника – або чисельник) дорівнює нулю, такий доданок (співмножник) не враховувати та вивести у комірку відповідне повідомлення. Додайте до протоколу лістинг функції **pib08()** з та скріншот відображення результату її виконання.

9. Використання методу confirm(). За допомогою використання методу **confirm()** до створеної функції **pib05()** з додайте оброблення вибору користувача пов'язаного зі способом виведення отримуваних результатів, «Ok» – у нове вікно браузеру, «Відміна» – у вікно повідомлень.

Приклад 8.4. Метод confirm() об'єкту window

```
function okcancel() {  
var vybir=confirm("Зробіть вибір")  
    if (vybir==true)  
    {  
        document.write("Ви обрали «Ok»!")  
    }  
    else  
    {  
        document.write("Ви обрали «Відміна»!")  
    }  
}
```

Додайте до протоколу лістинг зміненої функції **pib09()** та скріншот відображення результату її виконання.

10. Функція з аргументами, яка повертає значення. Змініть функцію **pib07()** таким чином, щоб аргументи функції містили формальні параметри *a*, *b* та *t*, які набувають введених користувачем значень. Змінену функцію зберегти під ім'ям **pib10()** і розмістити у сценарії в заголовку веб-документу. Фактичні значення параметрів увести за допомогою сценарію, розміщеного у тілі документу.

Приклад 8.5. Функція з аргументами

```
<html>  
<head>  
    <script type="text/javascript">  
        function delta(a,b) {  
            return Math.abs(a-b);  
        }  
    </script>  
</head>  
<body>  
    <script type="text/javascript">  
        document.write(delta(243,315))  
    </script>  
</body>  
</html>
```

Додайте до протоколу лістинг зміненої функції `rib10()` та скріншот відображення результату її виконання. Оформіть протокол лабораторної роботи.

Оформлення протоколу

Після оформлення протоколу лабораторної роботи до заздалегідь підготовлених складників додаються роздруківки скріншотів ходу виконання завдань.

Контрольні запитання

1. Розкрийте поняття «подія», «оброблювач події», «об'єкт», «властивість», «змінна», «літерал», «оператор», «функція», «вираз», «інструкція».
2. У чому відмінність впровадження і зв'язування сценарію?
3. Назвіть групи об'єктів JavaScript.
4. Які елементи об'єкту `document` Ви можете назвати?
5. Як нумеруються масиви об'єктів JavaScript?
6. Перелічіть методи об'єкту `window`. У чому полягає відмінність методів `prompt()` та `confirm()`?
7. Як додати плаваючий фрейм до веб-сторінки в динамічному режимі?
8. Що дає змогу здійснити метод `write()` об'єкту `document`?
9. Керування потоком обчислень. Запишіть синтаксис операторів `for` та `while`.
10. Керування потоком обчислень. Запишіть синтаксис оператору `if` та `switch`.

Лабораторна робота № 9

Робота з даними за допомогою мови JavaScript з використанням форм

Мета роботи – набути уміння створювати і обробляти html-форми.

Завдання. Розглянути принципи і прийоми створення та застосування форм JavaScript.

Короткі теоретичні відомості

Вихідною точкою для пошуку підходів розуміння DHTML є HTML, структуру якого використовує DHTML, щоб додавати свої динамічні можливості. *Об'єктна модель документу (DOM)* дає змогу сценаріям керувати кожним html-елементом. Розширення зв'язків між HTML і DHTML проявляється у таблицях стилів, де з'являються динамічні стилі. Будь-який css-селектор може стати динамічним з використанням DHTML. DOM представляє собою прикладний програмний інтерфейс (API) та визначає логічну структуру документу і спосіб доступу та маніпулювання елементами

документу. Об'єктна модель DHTML надає розробникам веб-ресурсів прямий програмний доступ до усіх елементів документу, що дає змогу браузеру обробляти введення користувача, виконувати сценарії і динамічно змінювати вміст документу, не перезавантажуючи його.

Ієрархічна структура об'єктів моделі DHTML схожа на ієрархічну структуру об'єктів JavaScript, але додатково дає змогу звертатись і встановлювати властивості каскадних таблиць стилів. Ім'я властивості у CSS не відповідає назві властивості у JavaScript – у більшості випадків необхідно відкинути дефіс і приєднати наступну частину до попередньої з великої літери. Наприклад, на властивість CSS `background-color` у сценарії JavaScript необхідно посилатись як на `backgroundColor`.

Більшість властивостей об'єктів відповідають атрибутам елементів документу і мають такі ж само імена, що й імена атрибутів. У сценарії можна отримати значення атрибутів елемента і, навпаки, змінити певне значення. Динамічна зміна властивостей об'єктів і, відповідно, елементів що їх представляють, є основною концепцією динамічного HTML. Об'єктна модель дає змогу здійснювати доступ і змінювати не тільки значення параметрів елементів, але і їх вміст. Це можливо за допомогою таких властивостей об'єктів: `innerText`, `innerHTML`, `outerHTML` і `outerText`.

У об'єктній моделі DHTML з кожним елементом сторінки можна пов'язати певну дію користувача: натискання кнопки миші, натискання клавіші, переміщення у області елементу курсору і т.д. Ця технологія базується на фундаментальному понятті «події» у системах із графічним інтерфейсом користувача. У разі виникнення будь-якої події динамічно створюється властивість `event` об'єкта `window`, що входить у об'єктну модель. Ця властивість і є об'єктом, що відповідає генерованій події. Цикл життя будь-якої події охоплює такі етапи:

1. Відбувається дія користувача чи виникає умова, що викликає подію.
2. Корегується об'єкт `event`, щоб відобразити параметри події, що виникла.
3. Генерується подія.
4. Викликається оброблювач подій елемента-джерела події, який виконує визначені програмістом дії і завершує свою роботу.
5. Подія передається наверх ієрархією об'єктів і викликає оброблювач події об'єкта, що є батьківським по відношенню до об'єкта-джерела події. Це «спливання» наверх ієрархією об'єктів продовжується, доки не буде досягнуто самий верхній об'єкт ієрархії - об'єкт `window`, або оброблювач події якогось із об'єктів не анулює подію.
6. Виконуються заключні дії за замовчуванням, якщо такі вказано, але за умови, що подію не буде скасовано.

DHTML дає змогу створювати мультимедійні ефекти, що реалізуються за допомогою застосування фільтрів до елементів сторінки і організації переходів від одного візуального стану до іншого. *Фільтр* – це деякий алгоритм, що перетворює візуальне відображення елемента у вікні браузера. *Статичний фільтр* перетворює елемент і після цього він відображається. Застосувати фільтр легко: досить задати значення його властивості `filter`,

згідно правил завдання властивостей CSS. *Динамічний фільтр* діє у часі на візуальне відображення, що приводить до ефекту анімації. Динамічний фільтр ще називають *переходом* із одного стану в інший.

Динамічні фільтри або ефекти переходів, як і статичні задають з допомогою властивості **filter** стилю, відмінність складається у необхідності створення сценарію з оброблення події. Оскільки такі ефекти часто супроводжуються зміною зображень, то необхідно задати у тій області, до якої будуть застосовані фільтри, не одне зображення, а кілька. При цьому вони повинні займати одне і те саме місце на сторінці, що досягається за допомогою абсолютного позиціонування і властивості **visibility** обраного об'єкту, значення якої змінюється у вказаному сценарії.

За допомогою DHTML можливе створення інтерактивних сторінок, що будуть містити списки, що розкриваються; дані, що можна сховати чи відобразити; елементи, що рухаються; анімацію тексту тощо. Наприклад, елемент, що рухається, створюють шляхом застосування абсолютного позиціонування та присвоєння властивостям **top** і **left** даного об'єкту нових значень. Список, що розкривається, створюють використовуючи таку візуальну властивість CSS як **display**.

Зміна стилю елемента відповідно до налаштувань таймеру дає змогу створювати текстові анімації. Сценарії можуть змінювати один чи більшу кількість стилів для кожного сигналу таймера. Зміна зовнішнього вигляду документу за допомогою таймера корисна для привертання уваги до змісту документу. Наприклад, наступний код демонструє зміну властивості CSS **letter-spacing** визначеного елемента у відповідь на сигнали таймера, таким чином створюється так званий пружний текст:

Приклад 9.1. Пружний текст

```
<html><head>
<title> Пружний текст </title>
<script type="text/javascript">
    // Масив значень розмірів
    var sizes = new Array("0px", "1px", "2px", "4px", "8px");
    sizes.pos = 0;
    function rubberBand() {
        var el = document.all.elastic;
        if (null == el.direction)
            el.direction = 1;
        else if ((sizes.pos > sizes.length - 2) ||
            (0 == sizes.pos))
            el.direction *= -1;
        el.style.letterSpacing = sizes[sizes.pos += el.direction];
    }
</script>
</head>
<body onload="window.tm = setInterval('rubberBand()', 100);"
onunload="clearInterval(window.tm);">
<h1 id="elastic" align="center">Це пружний текст</h1>
</body></html>
```

Використовуючи вищезгадані властивості `innerText`, `innerHTML`, `outerHTML` і `outerText` можна змінювати вміст веб-сторінки без обміну із сервером.

Приклад 9.2. Властивість `innerText`

```
<html>
<head>
  <script type="text/javascript">
    function ChangeText()
    {
      test.innerText="Динамічно побудований текст";
      window.alert ("Новий текст");
    }
  </script>
</head>
  <p id=test> Пробний текст </p>
  <form>
    <input id=Click type=button value="Змінити зміст документа"
      onClick="ChangeText();">
  </form>
</html>
```

HTML-форми призначено для отримання даних від віддаленого користувача. За їх допомогою можна організувати найпростіший діалог між користувачем і сервером, наприклад, реєстрацію користувача на сервері або вибір необхідного документу з представленого списку.

Існує кілька ключових елементів форм і ряд пов'язаних із ними атрибутів.

- `<form>...</form>` – це дескриптори, що відкривають і закривають форму. Форма може мати кілька атрибутів основний з яких – `action`, який у якості значення приймає адресу для передавання на вузол інформації з форми, і `method`, що приймає значення `get` та `post`. Ці значення визначають спосіб передавання даних форми. У JavaScript форму представлено об'єктом `HtmlFormElement`.
- дескриптор `<input>` відповідає за створення керувальних елементів, які розміщено у формі. З вказаним дескриптором можуть використовуватись атрибути:
 - `type= "x"` визначає елемент керування;
 - `name= "x"` – ім'я елемента керування;
 - `value= "x"` – значення елемента керування;
 - `size= "x"` – ширина поля елемента керування у пікселях (для елементів `text` та `password` – у символах);
 - `maxlength= "x"` – гранична кількість символів, яку приймає елемент керування;
 - `checked= "x"` – відмічений прапорець (перемикач);
 - `src= "x"` – покажчик на малюнок, що використовується у формі як кнопка;

- елемент `<textarea>...</textarea>` створює текстову область з розширеними можливостями.
- елемент `<select>...</select>` створює меню. Меню може бути випадаючим або у вигляді списку.
- дескриптор `<option>...</option>` визначає кожен елемент списку меню окремо.

Екранні керувальні елементи визначають спосіб введення інформації. Синтаксично елементи представляють собою значення атрибуту `type` дескриптора `<input>`.

- **text** – поле введення тексту, складається з одного рядка; ширину поля визначає значення атрибуту `size`;
- **password** – поле подібне до поля введення тексту; відмінність полягає у тому, що символи відображаються у вигляді `"*"`;
- **checkbox** – прапорець, що приймає два логічних значення. Можна створити групу з декількох прапорців; при цьому усі вони можуть бути відмічені;
- **radio** – перемикач; на відміну від прапорця, з декількох перемикачів, що утворюють групу, може бути обраний лише один;
- **submit** – кнопка, натискання якої означає підтвердження вибору і відправку інформації на веб-вузол; напис на кнопці визначається встановленням її значення `value`;
- **reset** – кнопка, натискання якої означає скидання поточних значень елементів керування і встановлення значень за замовчуванням;
- **file** дає змогу створити кнопку виклику діалогового вікна вибору файлів;
- **hidden** – прихований елемент керування, який не представлено зображенням на екрані;
- **image** – елемент, який дає змогу вставити зображення і використовувати його для підтвердження вибору чи скидання замість кнопок **submit** або **reset**;
- **button** – кнопка; для роботи кнопки необхідні пов'язані із нею сценарії.

Розглянемо детальніше деякі із керувальних елементів. Створимо форму для передавання інформації про клієнта на вузол:

Приклад 9.3. Форма

```
<form method="post" action="/cgi/bin/mailscript">
  Прізвище: <input type="text" name="surname" size="25" maxlength="100"><br>
  E-mail: <input type="text" name="email" size="25" maxlength="100">
<p> Вік <br>
  20-25 <input type="checkbox" name="20" checked>
<br>
  25-30 <input type="checkbox" name="25">
<p> Стать: <br>
  <input type="radio" name="button" value="female" checked> Чоловіча
<br>
  <input type="radio" name="button" value="female"> Жіноча
```

```

<p> Місто:
  <select>
    <option name="kiev"> Київ
    <option name="lviv"> Львів
    <option name="other"> Інше
  </select>
<p>Додаткові відомості
<p>
<textarea rows="5" cols="25"></textarea>
  <input type="submit" value="Ok">
  <input type="reset" value="Скидання"><br>
</form>

```

Окрім кнопок **submit** та **reset** використовуються і прості кнопки, з якими звичайно зв'язана якась дія. Проста кнопка створюється за допомогою коду:

```

<input type="button" name="ім'я кнопки" value="напис на кнопці"
  onclick="дія пов'язана із кнопкою">

```

Наприклад, представлений нижче код викличе появу повідомлення з інформацією про дату та час на цей момент:

Приклад 9.4. Кнопка

```

<input type="button" value="Показати дату та час" onclick="alert(Date());">

```

На кнопці можна розмістити будь-який фрагмент HTML-коду. Загальний вигляд нестандартних кнопок такий:

```

<button name="ім'я кнопки" onclick="дія пов'язана із кнопкою">
  HTML-код фрагменту, який розміщено на кнопці
</button>

```

Наприклад, можна зменшити малюнок у розмірі та помістити його у рядок, що бігає:

Приклад 9.5. Нестандартна кнопка

```

<button name=btn onclick="дія пов'язана із кнопкою">
  <marquee behavior=alternate bgColor=#ffb6c1 width=300 align="baseline">
    
  </marquee>
</button>

```

Оформіть протокол лабораторної роботи.

Послідовність виконання роботи

Створіть веб-сторінку, яка міститиме форму опитування за індивідуальною тематикою. Опитувальник щонайменше повинен охоплювати:

1. Три елементи `input` для введення ПІБ опитуваного.
2. Елемент `checkbox` для відповіді на певне запитання.

Приклад 9.6. Елемент `checkbox`

```
<form name="formochka">
  <input type="checkbox" name="halochka" checked>
  <span>Правда чи ні?</span>
</form>
<div id="ryadochok"></div>
<script>
  var halochkaBox = document.formochka.halochka;
  function klats(vklvykl){
    var ryadochok = document.getElementById("ryadochok");
    var halochka = vklvykl.target.checked;
    ryadochok.textContent = halochka;
  }
  halochkaBox.addEventListener("click", klats);
</script>
```

3. Елементи `radio` для відповіді на певне запитання.

Приклад 9.7. Елементи `radio`

```
<form name="formochka">
  <input type="radio" name="discipline" checked="checked" value="OITCP" />
  <span>Основи інженерії та технології сталого розвитку</span></br>
  <input type="radio" name="discipline" value="I33" />
  <span>Інклюзивне зелене зростання</span></br>
  <input type="radio" name="discipline" value="CIP" />
  <span>Сталий інноваційний розвиток</span>
</form>
<div id="ryadochok"></div>
<script>
  function klats(vybir){
    var ryadochok = document.getElementById("ryadochok");
    var discipline = vybir.target.value;
    ryadochok.textContent = "Ви обрали: " + discipline;
  }
  for (var i = 0; i < formochka.discipline.length; i++) {
    formochka.discipline[i].addEventListener("click", klats);
  }
</script>
```

4. Список `select` для відповіді на певне запитання.
5. Елемент `textarea` для розгорнутої відповіді на певне питання.
6. Елемент `password` для введення користувачем паролю.

7. Елемент-кнопку для виклику функцій оброблення відповідей користувача.

У результаті виконання сценарію JavaScript, який міститиме функцію(ї) оброблення відповідей користувача, користувач повинен отримати розгорнутий звіт щодо перевірки правильності його відповідей. Звітування користувачу можна організувати у цьому ж веб-документі або іншому веб-документі, плаваючому фреймі чи текстовому файлі. Необхідно передбачити перевірку правильності введення паролю до початку оброблення полів форми. Для усіх елементів форми передбачити наявність значень за замочуванням.

Оформіть протокол лабораторної роботи.

Оформлення протоколу

Після оформлення протоколу лабораторної роботи до задалегідь підготовлених складників додаються роздруковки скріншотів ходу виконання завдань.

Контрольні запитання

1. Що означає аббревіатура DOM?
2. Розкрийте етапи життєвого циклу події.
3. Як додати кнопку до веб-документу і зв'язати з обробником події?
4. Як створити текстову область? Які атрибути має елемент «текстова область»?
5. Як додати меню та його складники? Які атрибути має елемент «меню»? Які атрибути мають складники елемента «меню»?
6. Як додати форму-прапорець? Які атрибути має елемент «прапорець»?
7. Як додати форму-перемикач? Які атрибути має елемент «перемикач»?
8. Як створити кнопку виклику діалогового вікна вибору файлів?
9. Як створити поле введення паролю? Які атрибути має елемент «пароль»?
10. Чи можна обробити форму, в якій немає жодної кнопки з типом submit?

Лабораторна робота № 10

Створення програмного забезпечення для роботи з СКБД MS SQL Server у середовищі MS VisualStudio

Метою роботи є опанування методів та принципів створення застосунків для забезпечення роботи з СКБД у середовищі MS VisualStudio.

Завдання. Розглянути засоби розроблення програмних комплексів з використанням можливостей роботи з СКБД у середовищі MS VisualStudio.

Короткі теоретичні відомості

Базою даних називають об'єднаний набір інформації. Існує велика кількість різноманітних методів організації інформації (даних) в бази, з метою спрощення роботи з ними. Програмна система, що забезпечує роботу з базою даних, називається системою керування базами даних (СКБД). Система керування базами даних забезпечує доступ до даних та дає змогу працювати з даними. З допомогою СКБД можна наповнити базу інформацією і, за необхідності, знайти необхідну інформацію, видалити чи змінити її.

Найбільшого поширення набули реляційні бази даних. Практично в усіх реляційних СКБД використовується мова структурованих запитів (англ. Structured Query Language) для виконання запитів, тобто звернень користувача до бази даних, які відповідають заданим параметрам, і маніпуляцій з даними. Найпоширенішими є такі реляційні СКБД як MS SQL Server, Oracle, Sybase, MS Access тощо. Реляційна модель даних представляє собою набір відношень, що змінюються в часі. У разі створення бази даних, основою моделі є відношення.

Середовище програмування підключається до БД і взаємодіє з нею через програмні засоби, які спрощують зв'язок між СКБД та програмним додатком. У разі використання C# зв'язок програми з СКБД може бути реалізовано через Microsoft ActiveX Data Objects (ADO.NET).

Microsoft SQL Server – це система керування клієнт-серверними реляційними базами даних, орієнтована на роботу під керуванням операційних систем Microsoft Windows.

Компоненти сервера. Компонент SQL Server Database Engine містить компонент Database Engine, основну службу для зберігання, обробки та забезпечення безпеки даних, реплікації, повнотекстового пошуку, засобів управління реляційними і XML-даними, інтеграції аналітики з базами даних і інтеграції PolyBase для доступу до Hadoop і іншим різноманітним джерелам даних, а також сервер Data Quality Services (DQS). Служби Analysis Services містять засоби створення додатків оперативної аналітичної обробки (OLAP) і додатків інтелектуального аналізу даних, а також засоби керування ними. Служби Reporting Services охоплюють серверні і клієнтські компоненти для створення, керування і розгортання табличних, матричних і графічних звітів, а також звітів у вільній формі. Служби Integration Services набір графічних засобів і програмованих об'єктів для переміщення, копіювання і перетворення даних. Служби Master Data Services - це рішення SQL Server з керування основними даними. MDS можна налаштувати для керування будь-якою структурою (товари, замовники, рахунки). Підтримуються ієрархії, детальне налаштування безпеки, транзакції, керування версіями даних і бізнес-правила, а також використання надбудови керування даними для Excel.

Служби машинного навчання (у базах даних) підтримують розподілені і масштабовані рішення машинного навчання, які використовують корпоративні джерела даних. У SQL Server 2016 підтримувалась мова R. SQL Server 2017 підтримує R і Python. Сервер машинного навчання (автономний)

підтримує розгортання розподілених масштабованих рішень машинного навчання на безлічі платформ і використання різних корпоративних джерел даних, включаючи Linux і Hadoop. У SQL Server 2016 підтримувалась мова R. SQL Server 2017 підтримує R і Python.

Клієнтська частина (засоби керування). SQL Server Management Studio - це інтегроване середовище для доступу, налаштування, керування, адміністрування і розробки всіх компонентів SQL Server. Середовище Management Studio дає змогу розробникам і адміністраторам, що володіють різними рівнями навичок, використовувати SQL Server.

SQL Server Configuration Manager забезпечує базові можливості керування конфігураціями для служб, серверних протоколів, клієнтських протоколів і псевдонімів клієнтів SQL Server.

SQL Server Profiler надає графічний користувацький інтерфейс для спостереження за екземпляром компонента Database Engine або служб Analysis Services; дає змогу в реальному часі відстежувати виконання всіх команд.

Помічник з налаштування Database Engine ядра компонента Database Engine допомагає створювати оптимальні набори індексів, індексованих представлень і секцій.

Клієнт Data Quality надає простий і зрозумілий графічний користувацький інтерфейс для підключення до сервера DQS і виконання операцій очищення даних. дає змогу централізовано відстежувати різні дії, що виконуються під час операції очищення даних.

SQL Server Data Tools (SSDT) містить інтегроване середовище розробки, призначене для створення рішень для таких компонентів бізнес-аналітики: Analysis Services, Reporting Services і Integration Services (раніше - середовище Business Intelligence Development Studio).

Компоненти зв'язку – компоненти для зв'язку між клієнтами та серверами і мережеві бібліотеки для DB, ODBC і OLE DB.

Конфігурація екземпляра сервера охоплює:

- параметри запуску служб сервера;
- виділення пам'яті для MS SQL Server;
- прив'язку до процесорів;
- тип автентифікації користувачів;
- параметри аудиту доступу до сервера;
- конфігурацію клієнтських підключень до сервера;
- налаштування новостворюваних баз даних;
- мову за замовчуванням для повідомлень сервера;
- керування іменами входу, ролями і правами.

Послідовність виконання роботи

1. Створення бази даних. Відкрийте Microsoft SQL Server Management Studio. Пройдіть авторизацію.

У оглядачі об'єктів оберіть об'єкт «Бази даних» і відкрийте контекстне меню. Оберіть команду «Створити базу даних...». Задайте ім'я нової бази даних (**pib**).

У переліку складників бази даних **pib**, оберіть гілку «Таблиці». У контекстному меню оберіть пункт «Таблиця...» з підменю «Створити». Створіть таблицю **pibaddress** з полями **city** (nvarchar(50)), **street** (nvarchar(50)), **house** (nvarchar(50)), **flat** (nvarchar(10)), **passport** (nvarchar(50)) і таблицю **pibpassport** з полями **passport** (nvarchar(50)), **surname** (nvarchar(50)), **name** (nvarchar(50)), **lastname** (nvarchar(10)).

Необхідну для виконання роботи БД створено. Оскільки програма матиме можливості для додавання та редагування записів таблиці БД **pib**, то необхідності заповнювати БД інформацією через MS SQL Server наразі немає.

2. Створення програми для роботи з БД. Відкрийте MS Visual Studio. Під час вибору нового проекту оберіть «Windows Form Application». Оберіть елемент керування DataGridView і перетягніть його на створену форму. Оберіть елемент керування Button і перетягніть його на форму. Оберіть об'єкт DataGridView і в його меню оберіть пункт «Choose Data Source». Оберіть підпункт «Add Project Data Source». Оберіть джерелом даних БД. Для вибору моделі БД оберіть варіант DataSet. У відкритому вікні вкажіть зв'язок з БД, натиснувши «New Connection». Вкажіть СУБД, з якою буде пов'язаний об'єкт DataGridView – MS SQL Server. Оберіть активну службу MS SQL Server и базу даних (**pib**). Перевірте зв'язок між програмою та сервером (англ. test connection). Перегляньте рядок зв'язку з БД (рядок «Connection string»). У відкритому вікні оберіть об'єкти БД, які потрібно відобразити. На форму додайте об'єкти button, label, textbox (рис. 23).

За допомогою контейнера TabControl розмістіть на формі вкладки для додавання, видалення, редагування та відображення інформації БД **pib**. Пошук інформації в БД реалізуйте за допомогою накладення фільтрів на кожне поле таблиці **dbo.pibaddress**.

Приклад 10.1. Лістинг фільтрів

```
private void button1_Click(object sender, EventArgs e)
{
    //місто;
    addressBindingSource1.Filter = "city =" + textBox1.Text + "";
}

private void button2_Click(object sender, EventArgs e)
{
    addressBindingSource1.Filter = "street =" + textBox2.Text + "";
}

private void button6_Click(object sender, EventArgs e)
{
    addressBindingSource1.Filter = "";
}
```

Скидання фільтрів

Програма pib					
Інформація	Особисті дані		Додати запис	Редагувати запис	Видалити запис
	Місто	Вулиця	Будинок	Квартира	Паспорт
	Київ	Кудрі	2А	36	СН829534
	Київ	Теліги	55	142	СН124895
	Житомир	Рамси	6	8	СЕ171218
	Буча	Зелена	12	1	РН154622

Вкажіть місто

вулицю

будинку

квартиру

паспорт

Скинути

Знайти

Знайти

Знайти

Знайти

Знайти

Рисунок 23 – Розміщення складників на формі

Додавання інформації реалізуйте шляхом створення SQL-запиту до БД (використайте оператор **INSERT**). Необхідна інформація вводиться користувачем у відповідні поля, потім програма автоматично формує і виконує відповідний запит.

Програма pib				
Інформація	Особисті дані	Додати запис	Редагувати запис	Видалити запис
<div>Додати інформацію щодо квартири</div> <div>Додати інформацію щодо особи</div>				
Вкажіть місто		<input type="text" value="Вінниця"/>		
Вкажіть вулицю		<input type="text" value="Шевченко"/>		
Вкажіть номер будинку		<input type="text" value="1"/>		
Вкажіть номер квартири		<input type="text" value="5"/>		
Вкажіть серію і номер паспорта		<input type="text" value="СН654541"/>		
		<div>Додати запис</div>		

Рисунок 24 – Додавання даних

Видалення інформації реалізуйте шляхом формування SQL-запиту до БД (оператор **DELETE**). Необхідна інформація вводиться користувачем, програма автоматично формує і виконує відповідний запит.

Приклад 10.2. Лістинг додавання запису

```
private void AddressUpdate_Click(object sender, EventArgs e)
{
    using (System.Data.SqlClient.SqlConnection connection =
        new System.Data.SqlClient.SqlConnection("DataSource=SERVER;InitialCatalog=pib;
        IntegratedSecurity=True"))
    {
        connection.Open();
        string queryString = "INSERT INTO dbo.pibaddress (city,street,house,flat,passport)
        VALUES (";
        queryString = queryString + "\"" + NewCity.Text + "\", " + NewStreet.Text + "\", " +
        NewHouse.Text + "\", " + NewFlat.Text + "\", " + NewPassport.Text + "\"";
        try
        {
            System.Data.SqlClient.SqlCommand command =
            new System.Data.SqlClient.SqlCommand(queryString, connection);
            command.ExecuteNonQuery();
        }
        catch
        {
            MessageBox.Show("Помилка./nМожливо неправильно введено дані у поля
            введення, спробуйте ще раз");
        }
        finally
        {
            this.adressTableAdapter1.Fill(this.pibDataSet1.adress);
            connection.Close();
        }
    }
}
```

Приклад 10.3. Лістинг видалення запису

```
private void Delete_Click(object sender, EventArgs e)
{
    using (System.Data.SqlClient.SqlConnection connection =
        new System.Data.SqlClient.SqlConnection("DataSource= SERVER;InitialCatalog=
        pib;IntegratedSecurity=True"))
    {
        connection.Open();
        string queryString = "DELETE FROM dbo.pibaddress WHERE city=";
        queryString = queryString + "\"" + DeleteCity.Text + "\" AND street=\"" + DeleteStreet.Text
        + "\" AND house=\"" + DeleteHouse.Text + "\" AND flat=\"" + DeleteFlat.Text + "\" AND
        passport=\"" + DeletePassport.Text + "\"";
        try
        {
            System.Data.SqlClient.SqlCommand command =
            new System.Data.SqlClient.SqlCommand(queryString, connection);
            command.ExecuteNonQuery();
        }
        catch
        {
            MessageBox.Show("Помилка./nМожливо неправильно введено дані у поля
            введення, спробуйте ще раз");
        }
    }
}
```



```

    }
finally
{
this.pibadressTableAdapter1.Fill(this.pibDataSet1.adress);
connection.Close();
}
}
}

```

Організуйте також редагування записів БД. Оформіть протокол лабораторної роботи.

Оформлення звіту

Після оформлення звіту з лабораторної роботи до заздалегідь підготовленого протоколу додаються роздруківки скріншотів і листингів ходу виконання завдань.

Контрольні запитання

1. Які складники Microsoft SQL Server є компонентами серверу?
2. Які складники Microsoft SQL Server є засобами керування?
3. Які функції покладено на служби Reporting Services?
4. Які функції покладено на SQL Server Management Studio?
5. Які параметри охоплює конфігурація екземпляра сервера?
6. Який оператор SQL використовується для додавання даних?
7. Який оператор SQL використовується для видалення даних?

Лабораторна робота № 11

Оброблення результатів моделювання ХТП (ХТС) збережених у БД за допомогою створення додатку на мові С#

Метою роботи є опанування методів та принципів створення застосунків для забезпечення взаємодії з СКБД для вибірки, впорядкування й оброблення даних.

Завдання. Розглянути засоби розроблення програмних комплексів з використанням можливостей роботи з СКБД у середовищі MS VisualStudio.

Короткі теоретичні відомості

Для простої вибірки даних використовується скорочений синтаксис оператора **SELECT** діалекту T-SQL:

```

SELECT [ALL | DISTINCT] [TOP n [PERCENT]] СпісокВибірки
FROM Ім'яТаблиці
WHERE УмоваВідбору

```

СписокВибірки визначає поля, які буде охоплено в підсумковому наборі даних, **Ім'яТаблиці** вказує таблицю БД, з якої повертаються записи, а **УмоваВідбору** обмежує кількість записів, що повертаються, за допомогою логічних операторів. За замовчуванням команда **SELECT** повертає всі записи, зокрема і дублікати, що визначається ключовим словом **ALL**, яке можна опустити. Для отримання набору унікальних записів необхідно вказувати ключове слово **DISTINCT**. Використання ключового слова **TOP** дає змогу вивести не всі записи підсумкового набору, а тільки *n* перших. Можна вибирати не фіксовану кількість записів, а певний відсоток від усіх рядків – для цього вказується ключове слово **PERCENT**.

СписокВибірки може містити такі елементи:

* | **Ім'яПоля** | **Вираз** [**AS** **Псевдонім**], [... *n*]

Для вибірки всіх полів з таблиці в списку вибірки необхідно вказати зірочку (*). Ключове слово **AS** дає змогу замінити в підсумковому наборі даних звичайні імена полів псевдонімами. Ім'я псевдоніма має задовольняти стандартним правилам іменування об'єктів. Якщо необхідно включити неприпустимі символи, пробіли або, наприклад, кириличні знаки, *ім'я псевдоніма* потрібно взяти у квадратні дужки.

Елемент **Вираз** задає вираз, який включається в підсумковий набір даних. **Вираз** може містити константи, імена полів, функції та їх комбінації. За замовчуванням ім'я колонки з виразом не визначено, тому можна вказати псевдонім.

Одержуваний набір даних можна сортувати в зростаючому або спадному порядку. Сортування можливе за іменем поля (навіть якщо воно і не вказано в списку вибірки), за псевдонімом або за позицією у списку вибірки, які вказуються в розділі

ORDER BY **Ім'яПоля** [... *n*] [**ASC** | **DESC**]

За замовчуванням сортування здійснюється за зростанням, що відповідає зарезервовану слову **ASC**, яке може опускатися, для сортування у порядку спадання вказується – **DESC**.

УмоваВідбору визначає критерій відбору записів, що включаються в підсумковий набір. У результат буде включено тільки ті рядки, які відповідають накладеним умовам. Умова може включати вирази, утворені за допомогою операторів порівняння або логічних операторів. Умови можуть також об'єднуватися і за допомогою логічних операндів **AND**, **OR** і **NOT**. За допомогою логічного оператора **BETWEEN** можна отримати відповідь на питання, чи лежить величина в зазначеному діапазоні.

Для пошуку за шаблоном символічних рядків використовується логічний оператор **LIKE**. У шаблоні можна використовувати такі універсальні символи:

% - будь-який рядок, що складається з 0 і більше символів;

_ - один символ;

[] - будь-який символ із заданої множини або діапазону,

[^] - будь-який символ, який не потрапляє в заданий діапазон або безліч.

Для визначення відповідності **Виразу** одному з перерахованих в заданому списку значень застосовується логічний оператор **IN**. Цей оператор можна записати у вигляді групи умов, об'єднаних операндом **OR**. Однак в список значень не можна включати невизначене значення **NULL**, для роботи з такими значеннями використовується функція вибірки **IS NULL**.

Вибірка даних з кількох таблиць передбачає з'єднання декількох таблиць для отримання єдиного набору результатів, що включають записи і поля кожної таблиці. Існує три види з'єднань: внутрішнє, зовнішнє і перехресне. Для об'єднання трьох і більше таблиць можна застосовувати послідовність з'єднань. Для з'єднання таблиць необхідно розділ **FROM** доповнити ключовими словами **JOIN**, яке визначає з'єднувані таблиці і метод з'єднання, і **ON**, яке вказує загальні для таблиць поля.

За внутрішнього з'єднання порівнюються значення спільних полів двох таблиць, повертаються тільки записи, що задовольняють критерію зв'язування в обох таблицях. Зовнішнє з'єднання також повертає об'єднані записи, відповідні умові об'єднання. Однак ліві і праві з'єднання повертають записи і не відповідні вказаній умові з'єднання. Такі з'єднання застосовуються для отримання повного набору записів однієї з таблиць. За лівого з'єднання в результат буде включено всі записи лівої таблиці (ім'я якої розташовано зліва від **JOIN**), незалежно від того, чи є для них відповідний запис в правій таблиці (ім'я таблиці розташоване праворуч від **JOIN**) чи ні. У разі правого з'єднання (ключове слово **RIGHT JOIN**) у результат включаються всі записи правої таблиці, незалежно від того, чи є для них відповідний рядок в лівій таблиці. У разі перехресного з'єднання виводяться всі комбінації записів таблиць, при цьому не потрібно вказувати співпадаючих значень полів, тому умова **ON** опускається. У нормалізованих базах даних перехресні з'єднання найчастіше використовуються для отримання списку всіх можливих комбінацій записів двох таблиць, тобто число записів підсумкового набору дорівнюватиме добутку числа записів першої таблиці на число записів другої.

Об'єднання декількох наборів результатів. Оператор **UNION** об'єднує результати двох і більше операторів **SELECT** і застосовується в разі, коли дані не можна отримати за допомогою одного запиту. Для отримання єдиного підсумкового набору даних необхідно написати окремі оператори **SELECT** і об'єднати їх за допомогою оператора **UNION**, при цьому, на відміну від з'єднання, записи в підсумковий набір додаються один за одним. За замовчуванням повторювані записи видаляються, для отримання всіх записів необхідно вказати ключове слово **ALL**. Необхідно також враховувати, що список полів, порядок і всі їхні властивості повинні бути однакові у всіх використовуваних запитах. Імена полів підсумкового набору беруться з першого запиту, тому створення псевдонімів полів виконується в ньому. Для

отримання відсортованого набору даних розділ **ORDER BY** вказується після останнього оператора **SELECT**.

Аналітична вибірка даних з бази даних спирається на можливості T-SQL створювати запити, нерозривно пов'язані з *агрегатними функціями*: Avg, Count, Sum, Max, Min. У разі виконання агрегатної функції здійснюється об'єднання значень окремого поля таблиці або частини записів, після чого виконується вказане агрегування. Агрегатна функція повертає одне єдине значення, тому використання інших імен полів в списку вибірки заборонено.

Підзапит – це оператор **SELECT**, включений в інші запити. Підзапити застосовуються для розбивки складного запиту на серію логічних етапів. Їх застосування ефективно, якщо запит використовує записи, повернуті іншим запитом. Вкладені підзапити - повертають єдине значення або список значень. *Вкладений запит* виконується один раз, а потім результуюче значення використовується в зовнішньому запиті. *Пов'язані підзапити* використовують дані зовнішнього запиту, причому пов'язаний запит виконується один раз для кожного запису зовнішнього запиту. У пов'язаних підзапитах, щоб розрізняти імена таблиць необхідно застосовувати їх псевдоніми.

Для групування записів за полями або виразів застосовується розділ **GROUP BY** оператора **SELECT**, що дає змогу застосовувати для кожної групи функції агрегування. У разі використання **GROUP BY** для кожної визначеної групи значень виводиться тільки один запис в підсумковому наборі даних. Для групування записів допускається також використання розділу **WHERE**, в цьому випадку групуються записи, що задовольняють певній умові. Розділ **WHERE** дає змогу визначити, які записи повинні піддатися угрупованню, а розділ **HAVING** - які групи повинні бути виведені в підсумковий набір даних. Ключове слово **HAVING** можна використовувати тільки в розділі **GROUP BY**.

```
[ WITH { [ XMLNAMESPACES ,]  
    [ <СпільнийТабличнийВираз> ] } ]  
SELECT СписокВідбору [ INTO НоваТаблиця ]  
[ FROM Ім'яТаблиці ] [ WHERE УмоваВідбору ]  
[ GROUP BY Ім'я ]  
[ HAVING УмоваВідбору ]  
[ ORDER BY Ім'я [ ASC | DESC ] ]
```

Послідовність виконання роботи

Для наданого викладачем файлу бази даних з результатами моделювання деякого хіміко-технологічного процесу створіть програму для роботи з БД.

1. Створення форми. Відкрийте MS Visual Studio. Під час вибору нового проекту оберіть «Windows Form Application». Оберіть елемент керування DataGridView і перетягніть його на створену форму. Оберіть елементи

керування **Button** і перетягніть їх на форму. Оберіть об'єкт **DataGridView** і в його меню оберіть пункт «Choose Data Source». Оберіть підпункт «Add Project Data Source». Оберіть джерелом даних надану БД. Для вибору моделі БД оберіть варіант **DataSet**. У відкритому вікні вкажіть зв'язок з БД, натиснувши «New Connection». Вкажіть СУБД, з якою буде пов'язаний об'єкт **DataGridView** – MS SQL Server. Оберіть активну службу MS SQL Server и базу даних (**piбctп**). Перевірте зв'язок між програмою та сервером (англ. test connection). Перегляньте рядок зв'язку з БД (рядок «Connection string»). У відкритому вікні оберіть об'єкти БД, які потрібно відобразити. На форму додайте необхідні об'єкти **button**, **label**, **textbox** та інші. За необхідності за допомогою контейнера **TabControl** розмістіть на формі вкладки.

2. Створення програми взаємодії користувача з БД. Реалізуйте сортування інформації за обраним користувачем полем БД та обраним порядком сортування. Необхідна інформація вводиться користувачем у відповідне поле форми, потім програма автоматично формує і виконує потрібний запит. Окремо реалізуйте виведення програмою на форму перших десяти записів БД після сортування.

Реалізуйте *відбір* записів, значення одного з полів яких належать певному діапазону. Необхідна інформація вводиться користувачем у відповідне поле форми, потім програма автоматично формує і виконує потрібний запит.

Реалізуйте внутрішнє з'єднання двох таблиць за вказівкою викладача.

Реалізуйте визначення середнього, максимального і мінімального значень для числових полів однієї з таблиць наданої БД у разі натискання відповідної кнопки на формі користувачем.

Реалізуйте групування частини записів за умовою (розділ **WHERE**) за допомогою розділу **GROUP BY** оператора **SELECT**. Необхідна інформація вводиться користувачем у відповідні поля форми, потім програма автоматично формує і виконує потрібний запит.

Оформлення звіту

Після оформлення звіту з лабораторної роботи до заздалегідь підготовленого протоколу додаються роздруковки скріншотів та лістингів ходу виконання завдань.

Контрольні запитання

1. Який оператор SQL використовується для простої вибірки даних?
2. Яке ключове слово SQL використовується для отримання набору унікальних записів?
3. Яке ключове слово SQL дає змогу замінити в підсумковому наборі даних звичайні імена полів псевдонімами?
4. Який оператор SQL використовується для сортування даних?

5. Яке ключове слово SQL дає змогу сортувати дані у порядку спадання?
6. Який логічний оператор SQL дає змогу отримати відповідь на питання, чи лежить величина в зазначеному діапазоні?
7. Який оператор SQL дає змогу здійснити пошук за шаблоном символічних рядків?
8. Що означає «праве з'єднання», «ліве з'єднання»?
9. Назвіть агрегатні функції T-SQL.
10. Як організувати групування записів за допомогою оператора SELECT?

Рекомендована література

1. *Бази даних. Створення об'єктів MS Access – 2003 та 2007 : метод. вказівки до вик. комп. практ. для студентів спец. «Автоматизоване управління технологічними процесами» / КПП ім. Ігоря Сікорського; укладач Л.Д. Ярошук. Київ : НТУУ «КПІ», 2012. 106 с.*
2. *Бекаревич, Самоучитель Microsoft Access 2016 / Ю.Б. Бекаревич, Н.В. Пушкіна. СПб., 2017. 480 с.*
3. *Берко, А.Ю. Системи баз даних / А.Ю. Берко, О.М. Верес, В.В. Пасічник. Львів, 2019. Книга 1. Організація баз даних та знань. 438 с. Книга 2. Системи управління базами даних та знань. 584 с.*
4. *Гайна, Г.А. Основи проектування баз даних / Г.А. Гайна. Київ, 2018. 204 с.*
5. *Грофф, Д. Р. SQL. Полное руководство / Д. Грофф, П. Вайнберг, Э. Оппель. Москва, 2017. 957 с.*
6. *Пасічник, В. В. Організація баз даних та знань / В. В. Пасічник, В. А. Резніченко. Київ : BHV, 2006. 384 с.*
7. *Троелсен, Э. Язык программирования C# 6.0 и платформа .NET 4.6 / Э. Троелсен, Ф. Джепикс. Москва [и др.] : Вильямс, 2018. 1438 с.*
8. *Фримен, Э. Изучаем программирование на JavaScript / Э. Фримен, Э. Робсон . Москва : Питер, 2015. 637 с.*
9. *Цеслів, О.В. Технологія проектування та адміністрування баз даних і сховищ даних / О.В. Цеслів, А.С. Коломієць ; КПП ім. Ігоря Сікорського. Київ, 2017. 281 с.*
10. *Шпортко, О. В. Розробка баз даних в СКБД Microsoft Access: практикум для студентів вищих та учнів професійно-технічних навчальних закладів / О.В. Шпортко, Л.В. Шпортко. Київ, 2018. 184 с.*

Додаток А.

Заходи безпеки під час проведення практичних занять

Цикл лабораторних робіт виконується в комп'ютерному класі кафедри технічних і програмних засобів автоматизації інженерно-хімічного факультету, де розміщено персональні комп'ютери. Обладнання живиться електричним струмом напругою 220 В. Тому під час виконання лабораторних робіт слід дотримуватися заходів безпеки наступних інструкцій.

ІНСТРУКЦІЯ

з техніки безпеки при навчанні студентів на ПЕОМ в учбових лабораторіях кафедри технічних і програмних засобів автоматизації інженерно-хімічного факультету

1. Знання і суворе дотримання цих правил є обов'язковим для всіх осіб, допущених до роботи на ПЕОМ. Доведення їх до кожного зі студентів підтверджується особистим підписом кожного з них у контрольному листі з техніки безпеки. Особи, які не одержали такого інструктажу та не поставили підпис у контрольному листі з техніки безпеки, до роботи на ПЕОМ не допускаються.

2. Усі роботи в учбових лабораторіях кафедри кібернетики ТПЗА проводяться лише з дозволу викладача або співробітника кафедри.

3. Під час проведення занять в учбовій лабораторії не повинні знаходитися сторонні особи, в тому числі студенти інших груп. Студенти не повинні самовільно залишати учбову лабораторію під час занять.

4. Під час роботи на ПЕОМ слід пам'ятати, що в них використовується напруга, небезпечна для життя.

5. Всі особи, працюючі в учбових лабораторіях кафедри ТПЗА, повинні бути ознайомлені з правилами надання першої медичної допомоги при ураженні електричним струмом.

6. Перед вмиканням ПЕОМ кожен з працюючих повинен отримати дозвіл викладача або співробітника кафедри.

7. У випадках виникнення короткого замикання, горіння, диму, вогню в апаратурі, пристрій необхідно негайно вимкнути з мережі та доповісти викладачеві або співробітникові кафедри. Самостійні дії по усуненню пошкодження забороняються.

8. У випадку виходу з ладу обладнання або програмного забезпечення, що зумовлені іншими причинами, доповісти викладачеві або співробітникові кафедри. Вимикати апаратуру при цьому не дозволяється. Самостійні дії по усуненню пошкодження забороняються.

9. Працюючі в учбових лабораторіях кафедри ТПЗА несуть майнову та адміністративну відповідальність за збереження та використання обладнання, наданого для їх праці.

10. Категорично забороняється:

- самостійно вмикати та вимикати тумблери на щитку електроживлення;
- несанкціоновано вмикати електрообладнання;
- приносити та вмикати своє обладнання та пристрої, встановлювати власне

- програмне забезпечення;
- залишати без нагляду увімкнені пристрої та лабораторію;
- пересувати обладнання та комплектуючі;
- підключати та відключати інформаційні кабелі та кабелі живлення;
- використовувати власні носії інформації без дозволу викладачів або співробітників кафедри;
- знаходитись в учбовій лабораторії у верхньому одязі.
- після закінчення занять обладнання не вимикається, робоче місце має бути прибрано працюючим та перевірено викладачем або співробітником кафедри.

ІНСТРУКЦІЯ

про міри пожежної безпеки у лабораторіях, учбових та робочих приміщеннях кафедри технічних і програмних засобів автоматизації інженерно-хімічного факультету

1. Всі студенти повинні знати та ретельно виконувати «Загальні правила пожежної безпеки в КПП ім. Ігоря Сікорського».
2. Завідуючий кафедрою та завідуючий лабораторією відповідають за забезпечення пожежної безпеки всіх приміщень кафедри та за справність протипожежного обладнання та сигналізації.
3. Усе електричне обладнання, яке знаходиться в лабораторіях та приміщеннях кафедри, повинно мати заземлення.
4. У всіх приміщеннях необхідно дотримуватись чистоти, не займати приміщення непотрібними меблями, обладнанням та матеріалами.
5. Усі двері основних та додаткових виходів утримувати у стані швидкого відкривання.
6. Зберігання та використання горючих та легкоспалахуючих рідин у приміщеннях кафедри забороняється.
7. Ремонт електричного обладнання проводити у строгій відповідності з правилами пожежної безпеки.
8. Усі електрозахисти повинні знаходитися у закритому положенні, не займаними сторонніми предметами.
9. Коридори, проходи, тамбури, евакуаційні виходи та підходи до першочергових засобів пожежогасіння, а також комунікаційні ніші повинні бути постійно вільними, чистими та нічим не зайнятими.
10. Відповідальні особи перед закриттям приміщень повинні ретельно оглянути їх, забезпечити прибирання виробничих відходів, перевірити якість перекриття води, газу, відключити напругу електромережі, перевірити стан пожежної сигналізації та засобів пожежогасіння.
11. Від усіх приміщень мати два комплекти ключів. Один комплект здавати черговому, а інший – зберігати в певному місці, яке відомо обслуговуючому персоналу.

Студенти повинні знати та ретельно виконувати «Загальні правила техніки безпеки в КПП ім. Ігоря Сікорського», про що вони ставлять свій підпис у відповідному контрольному листі з техніки безпеки перед початком проведення циклу лабораторних робіт. Студенти, які не пройшли інструктаж і не поставили підпис у контрольному листі, до роботи не допускаються.

Додаток Б.

Вимоги до оформлення протоколу за результатами виконання лабораторної роботи

Протокол повинен містити титульний аркуш, копію загального завдання (мета, завдання і послідовність виконання роботи), короткі теоретичні відомості, хід виконання лабораторної роботи, висновки.

Протокол оформлюється на аркушах формату А4 шрифтом Times New Roman 14 розміру (інтервал абзаців – полуторний (1,5), вирівнювання – за шириною, поля сторінок: ліве, верхнє, нижнє – 20 мм, праве – 10 мм).

Кожну структурну частину звіту потрібно починати з нової сторінки. Першою сторінкою протоколу є титульний аркуш (Додаток В), який включають до загальної нумерації сторінок, але номер сторінки не ставлять. На подальших сторінках номер проставляють у верхньому правому куті.

Додаток В.

Зразок титульного листа протоколу

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНЖЕНЕРНО-ХІМІЧНИЙ ФАКУЛЬТЕТ

Кафедра технічних та програмних засобів автоматизації

Протокол
лабораторної роботи № ____

« _____ *Назва роботи* _____ »

з кредитного модуля «Технології розроблення програмного забезпечення –
2. Інформаційні системи і комплекси»

Протокол склав(ла)
студент(ка) гр. ____–____
Прізвище І. Пб.

Прийняв(ла): *Прізвище І. Пб.*

Кількість балів: _____

Дата: _____

Підпис: _____

Київ 2020

Додаток Г. Мова сценаріїв JavaScript

Таблиця Г.1 – Деякі об'єкти JavaScript

Назва	Опис
ANCHORS	Елемент прив'язки – це позиція у тексті, яка може бути елементом гіперпосилання. Встановлюється дескриптором <a>. Властивість об'єкту document.
BUTTON	Елемент форми, який вказують у дескрипторі <from>. Властивість об'єкту form.
CHECKBOX	Елемент форми, який користувач натисканням миші може встановлювати у стан вкл./викл. Властивість об'єкту form.
DATE	Заміняє звичайний тип дати. Властивостей не має. Методи отримання і встановлення інформації про дату та час поділяються на чотири групи: set, get, to та parse.
DOCUMENT	Об'єкт містить інформацію про поточний документ. Властивості відповідають атрибутам дескриптора <body>, а методи дають змогу керувати відображенням тексту.
ELEMENTS	Масив елементів об'єкту form, охоплює кнопки, перемикачі, селекторні кнопки, текст і текстові області.
FORM	Масив, що містить усі форми документу. Властивість об'єкту document.
FRAME	Вікно, що містить незалежний документ. Об'єкт типу window.
HIDDEN	Текстовий об'єкт, що не відображається. Властивість об'єкту document.
HISTORY	Об'єкт браузеру, містить посилання на URL відкритих сторінок.
LINK	Об'єкт типу location. Містить інформацію про гіперпосилання у документі. Властивість об'єкту document.
LOCATION	Містить повну інформацію про URL документу. Властивість об'єкту document.
MATH	Містить значення математичних констант і методи функцій.
NAVIGATOR	Містить інформацію про браузер клієнту.
OPTION	Селекторна кнопка. Властивість об'єкту form.
PASSWORD	Створюється текстовим полем пароллю. Властивість об'єкту form.
RESET	Пов'язаний із кнопкою скидання. Властивість об'єкту form.
SELECT	Список у html-формі. Властивість об'єкту form.
STRING	Сукупність символів. Значенням об'єкту можна керувати за допомогою методів.
SUBMIT	Передає дані, якими користувач заповнив форму, сценарію вказаному у властивості action. Властивість об'єкту form.
TEXT	Однорядкове поле введення. Властивість об'єкту form.
TEXTAREA	Багаторядкове поле введення. Властивість об'єкту form.
WINDOW	Об'єкт верхнього рівня для document, location та history. Створюється під час завантаження сторінки.

Таблиця Г.2 – Деякі властивості об'єктів JavaScript

<i>Назва</i>	<i>Опис</i>
action	Відображає атрибут action дескриптора <form>.
bgColor	Колір фону документа. Властивість об'єкту document.
checked	Булеве значення, що показує чи обрано перемикач чи селекторну кнопку.
cookie	Рядок з інформацією про клієнта, яку зберігає браузер на машині клієнта. Властивість об'єкту document.
href	Повертає рядок із повним URL. Властивість об'єктів link та location.
index	Повертає порядковий номер опції у елементі select. Властивість об'єкту option.
length	Властивість об'єкту пов'язана з його довжиною чи розміром.
method	Містить атрибут method дескриптора <form>.
name	Повертає рядок із атрибутом name об'єкту.
status	Повідомлення, що з'являється на панелі стану. Властивість об'єкту window.
title	Повертає значення, встановлене у дескрипторі <title>.
value	Значення властивості об'єкту. Залежить від типу об'єкту.

Таблиця Г.3 – Деякі методи об'єктів JavaScript

<i>Назва</i>	<i>Опис</i>
alert	Відображає діалогове вікно попередження з повідомленням заданим користувачем.
clear	Очищає вікно незалежно від його вмісту.
click	Імітує натискання миші на елементі form.
close	Для об'єкту document закриває потік виведення і виводить його вміст на екран
confirm	Відображає підтверджувальне діалогове вікно з повідомленням та кнопками Ok та Cancel.
open	Відкриває потік куди записується виведення функцій write та writeln.
write	Записує у вікно документу один чи декілька рядків.

Додаток Д.
Індивідуальні завдання до лабораторної роботи № 8

Завдання до п. 7.

1. Розробити алгоритм і програму для визначення залежності:
де $x = \frac{tg(t)}{\sin(t) + \cos(t)}$; $t = 1.2$.

$$Y = \begin{cases} \sin^4(x-5) & \text{ї дè } x < 0 \\ x^3 - \sqrt{x} & \text{ї дè } 0 \leq x < 10 \\ e^{-x} + |x| & \text{ї дè } x \geq 10 \end{cases}$$

2. Розробити алгоритм і програму для визначення залежності:
де $x = a \cdot \cos(t^2) - b \cdot t$;
 $a = 1$; $b = 0.2$; $t = 0.4$.

$$Y = \begin{cases} \frac{1}{x} + \frac{1}{3x^2} & \text{ї дè } x < 5 \\ 7x^{\ln(x)} & \text{ї дè } 5 \leq x < 7 \\ 2\sin x^4 + \cos x & \text{ї дè } x \geq 7 \end{cases}$$

3. Розробити алгоритм і програму для визначення залежності:
де $x = b \cdot t^2 + a$;
 $a = -3$; $b = 4$; $t = 3.5$.

$$Y = \begin{cases} e^{-2x} + x & \text{ї дè } x < 0 \\ x^4 + 5x & \text{ї дè } x = 0 \\ |\sin x + \operatorname{tg} x| & \text{ї дè } x > 0 \end{cases}$$

4. Розробити алгоритм і програму для визначення залежності:
де $x = 2.5 \cdot t^2 - a/t$;
 $a = 5$; $t = 7.8$.

$$Y = \begin{cases} 5^{\operatorname{tg} x} - 4 & \text{ї дè } x < 2 \\ \frac{1}{2\sin x + 3} & \text{ї дè } 2 \leq x < 3 \\ (2\sin x)^x & \text{ї дè } x \geq 3 \end{cases}$$

5. Розробити алгоритм і програму для визначення залежності:
де $x = a \cdot \sin t^b - a$;
 $a = 0.2$; $b = 0.4$; $t = 0.3$.

$$Y = \begin{cases} \cos 5x - x^{2.3} & \text{ї дè } x < -2 \\ |\ln x^4 - 5| & \text{ї дè } -2 \leq x < 2 \\ \frac{1}{\operatorname{tg} x + \operatorname{ctg} 2x} & \text{ї дè } x \geq 2 \end{cases}$$

6. Розробити алгоритм і програму для визначення залежності:
де $x = \ln t^a + b$;
 $a = 3$; $b = 0.3$; $t = 2.7$.

$$Y = \begin{cases} 10^{\sin x - \cos 2x} & \text{ї дè } x < 3 \\ 5 & \text{ї дè } 3 \leq x < 9 \\ |x^6 - 7\sqrt{x}| & \text{ї дè } x \geq 9 \end{cases}$$

7. Розробити алгоритм і програму для визначення залежності:
де $x = t^a + b \cdot t^b$;
 $a = 0.3$; $b = 0.4$; $t = 0.2$.

$$Y = \begin{cases} x^2 + 4 & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x < -2 \\ |\sin x - 2^{-x}| & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } -2 \leq x < 10 \\ \log_5 x^2 & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x \geq 10 \end{cases}$$

8. Розробити алгоритм і програму для визначення залежності:
де $x = t^{a \cdot b - b}$;
 $a = 0.3$; $b = 1.1$; $t = 0.7$.

$$Y = \begin{cases} \lg(3x - x^2) & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } 0 < x < 2 \\ \sin(3x^3 + 4) & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } 2 \leq x < 3 \\ \frac{1}{x^4 + 5} & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x \geq 3 \end{cases}$$

9. Розробити алгоритм і програму для визначення залежності:
де $x = a \cdot \sin^2 t + b \cdot \sin t^3$;
 $a = 1.3$; $b = 1.5$; $t = 1.8$.

$$Y = \begin{cases} |\sin x^2 - b \cdot \cos^2 x^3| & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x < 0 \\ \lg|x^2 - 5.1| + e^{-x^2 + 1} & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } 0 \leq x < 1 \\ tg^2(x - 2) - 2.5 & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x \geq 1 \end{cases}$$

10. Розробити алгоритм і програму для визначення залежності:
де $x = a \cdot \ln t$;
 $a = 1.2$; $b = 2.1$; $t = 2.8$.

$$Y = \begin{cases} \ln|(x - 1.5) - \sin x| + 2 \cdot a \cdot x^2 & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x < 1 \\ tgx - b \cdot \cos^2 x^3 & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } 1 \leq x \leq 2.1 \\ t^{-x^2 + 1.5x} + \ln x & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x > 2.1 \end{cases}$$

11. Розробити алгоритм і програму для визначення залежності:
де $x = a \cdot \sin t + b \cdot \cos 2t$;
 $a = 2.5$; $b = 0.8$; $t = 2.93$.

$$Y = \begin{cases} \ln|x - 3| + 1.4 & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x < 0 \\ tg(x - 1) - a \cdot \cos^2 x^3 & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } 0 \leq x < 1 \\ \ln|x^2 + 1| - e^{-2x} & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x \geq 1 \end{cases}$$

12. Розробити алгоритм і програму для визначення залежності:
де $x = e^{-\sin^2 t} + \cos t$;
 $t = 1.37$.

$$Y = \begin{cases} 3 & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x = 1.5 \\ \ln x & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x > 2 \\ \sqrt{|x|} & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } x \leq -3 \\ |x + tgx^2| & \text{ї} \text{ } \text{д} \text{ } \text{е} \text{ } -3 < x \leq 2 \end{cases}$$

13. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = a \cdot \sin^3 t + 1.5; \\ a = 1.2; \quad b = 2.3; \quad t = 0.8.$$

$$Y = \begin{cases} \cos^2 x + b \cdot \cos^3 x & \text{ї } \text{д} \text{е } x < 2 \\ e^{-x} + \operatorname{tg} x & \text{ї } \text{д} \text{е } 2 \leq x \leq 2.5 \\ \ln x^2 + b \cdot \ln x^4 & \text{ї } \text{д} \text{е } x > 2 \end{cases}$$

14. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = a \cdot \sin t - b \cdot \cos t^2; \\ a = 1.2; \quad b = 2.1; \quad t = 8.2.$$

$$Y = \begin{cases} \ln |\sin^3 x^3| - 1 & \text{ї } \text{д} \text{е } x \leq 2 \\ e^{-2 \cdot x^2 + \sin x} & \text{ї } \text{д} \text{е } 2 < x \leq 2.3 \\ \cos(a \cdot x) + b \cdot \sin x^2 & \text{ї } \text{д} \text{е } x > 2.3 \end{cases}$$

15. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = t^a + b \cdot t^b; \\ a = 0.3; \quad b = 0.4; \quad t = 0.2.$$

$$Y = \begin{cases} x^2 + 4 & \text{ї } \text{д} \text{е } x < -2 \\ |\sin x - 2^{-x}| & \text{ї } \text{д} \text{е } -2 \leq x < 10 \\ \log_5 x^2 & \text{ї } \text{д} \text{е } x \geq 10 \end{cases}$$

16. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = b \cdot t^2 + a; \\ a = -3; \quad b = 4; \quad t = 3.5.$$

$$Y = \begin{cases} e^{-2x} + x & \text{ї } \text{д} \text{е } x < 0 \\ x^4 + 5x & \text{ї } \text{д} \text{е } x = 0 \\ |\sin x + \operatorname{tg} x| & \text{ї } \text{д} \text{е } x > 0 \end{cases}$$

17. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = \frac{\operatorname{tg}(t)}{\sin(t) + \cos(t)}; \quad t = 1.2.$$

$$Y = \begin{cases} \sin^4(x - 5) & \text{ї } \text{д} \text{е } x < 0 \\ x^3 - \sqrt{x} & \text{ї } \text{д} \text{е } 0 \leq x < 10 \\ e^{-x} + |x| & \text{ї } \text{д} \text{е } x \geq 10 \end{cases}$$

18. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = a \cdot \cos(t^2) - b \cdot t; \\ a = 1; \quad b = 0.2; \quad t = 0.4.$$

$$Y = \begin{cases} \frac{1}{x} + \frac{1}{3x^2} & \text{ї } \text{д} \text{е } x < 5 \\ 7x^{\ln(x)} & \text{ї } \text{д} \text{е } 5 \leq x < 7 \\ 2\sin x^4 + \cos x & \text{ї } \text{д} \text{е } x \geq 7 \end{cases}$$

19. Розробити алгоритм і програму для визначення залежності:

$$\text{де } x = 2.5 \cdot t^2 - a/t; \\ a = 5; \quad t = 7.8.$$

$$Y = \begin{cases} 5^{\operatorname{tg} x} - 4 & \text{ї } \text{д} \text{е } x < 2 \\ \frac{1}{2\sin x + 3} & \text{ї } \text{д} \text{е } 2 \leq x < 3 \\ (2\sin x)^x & \text{ї } \text{д} \text{е } x \geq 3 \end{cases}$$

20. Розробити алгоритм і програму для визначення залежності:

де $x = a \cdot \sin t^b - a$;
 $a = 0.2$; $b = 0.4$; $t = 0.3$.

$$Y = \begin{cases} \cos 5x - x^{2.3} & \text{ї дè } x < -2 \\ |\ln x^4 - 5| & \text{ї дè } -2 \leq x < 2 \\ \frac{1}{\operatorname{tg} x + \operatorname{ctg} 2x} & \text{ї дè } x \geq 2 \end{cases}$$

Завдання до п. 8.

Варіант	Сума	Варіант	Добуток
1	$S = \sum_{n=i}^K \frac{n}{n^2 - 5n + 6}$	2	$Y = \prod_{i=M}^N \frac{i^2 + 2i + 3}{i - 3}$
3	$S = \sum_{n=L}^K \frac{n - 2}{n^2 - 16}$	4	$P = \prod_{k=M}^N \frac{k + 3}{(k - 5)(k - 6)}$
5	$H = \sum_{n=L}^k \frac{n^2 - n}{n^2 + n + 6}$	6	$Y = \prod_{k=N}^M \frac{k + 1}{(k - 5)(k - 7)}$
7	$S = \sum_{i=M}^k \frac{i^2 - 4i + 7}{i^2 - 6i + 13}$	8	$R = \prod_{n=M}^K \frac{n}{(n - 3)(n - 8)}$
9	$S = \sum_{i=K}^N \frac{i^2 - i + 3}{i - 5}$	10	$Z = \prod_{n=L}^m \frac{(n + 3)^3}{(n - 5)^2}$
11	$D = \sum_{i=K}^J \frac{i(i + 1)}{(i + 5)(i - 3)}$	12	$Y = \prod_{l=i}^j \frac{(l + 3)^2}{(l - 7)(l - 9)}$
13	$B = \sum_{i=K}^M \frac{(i + 1)^2}{2i^3 - 3i - 11}$	14	$Q = \prod_{k=M}^L \frac{k + 1}{(k - 8)(k - 6)}$
15	$J = \sum_{n=K}^L \frac{(n + 4)^3}{n^3 - n^2 + 2n - 3}$	16	$X = \prod_{n=z}^K \frac{n}{(n - 2)(n - 5)}$
17	$S = \sum_{j=Z_1}^{Z_2} \frac{(j + 2)^3}{(j - 8)(j - k)}$	18	$Y = \prod_{k=i}^j \frac{k^2 + 4k + 7}{k - 5}$
19	$Z = \sum_{i=K}^M \frac{(i + 1)^3}{i^2 - i - 7}$	20	$P = \prod_{l=M}^N \frac{l^2 + 3l}{(l - 4)(l + 5)}$