

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут»

Факультет зварювальний

Кафедра інженерії поверхні

**КОМП'ЮТЕРНІ БАЗИ ТИПОВИХ ТЕХНОЛОГІЙ ТА
ПРИСТРОЇВ**

Методичні вказівки

до

практичних занять

для студентів напрямку 6.050504 “Зварювання”

Затверджено вченою радою ЗФ НТУУ «КПІ»

Київ 2013

Комп'ютерні бази типових технологій та пристроїв: Методичні вказівки до практичних занять з дисципліни “Комп'ютерні бази типових технологій та пристроїв” для студентів напряму підготовки 6.050504 “Зварювання” / Уклад. Солодкий С.П. – К. : НТУУ КПІ, 2013 р. – 34 с., укр.

*Гриф надано Вченою радою ЗФ НТУУ «КПІ»
(Протокол № 5 від 28.01.13 р.)*

Навчальне видання

КОМП'ЮТЕРНІ БАЗИ ТИПОВИХ ТЕХНОЛОГІЙ ТА ПРИСТРОЇВ

Методичні вказівки

до виконання практичних занять
для студентів напряму 6.050504 “Зварювання”

Укладач: *Сергій Павлович Солодкий, канд. техн. наук, ст. викладач.*

Відповідальний редактор *В. М. Пащенко, канд. техн. наук, доцент*

Рецензент *А.Є. Пірумов, канд. техн. наук, доцент.*

ЗМІСТ

ВСТУП	4
Практичне заняття № 1 Створення бази даних з однієї таблиці	5
Практичне заняття № 2 Формування запитів та звітів для однотобличної бази даних.....	15
Практичне заняття № 3 Розробка інформаційно-логічної моделі реляційної бази даних.....	21
Практичне заняття № 4 Використання мови VBA при роботі з об'єктами бази даних.....	25
Практичне заняття № 5 Використання мови VBA для фільтрації даних в базі.....	31
Література	34

ВСТУП

Дисципліна «Комп'ютерні бази типових технологій та пристроїв» належить до дисциплін вільного вибору студентів і є однією із дисциплін у циклі підготовки бакалаврів за спеціальністю 6.050504 «Зварювання».

Мета практичних занять – закріпити знання, одержані на лекційних заняттях, комп'ютерному практикумі та в результаті самостійної роботи. Студенти опановують навички щодо проектування та створення баз даних типових технологій та пристроїв, використовуючи СУБД Access.

В даному посібнику представлені п'ять практичних занять.

На практичних заняттях 1,2 - розглядаються основи створення бази даних з однієї таблиці, формування запитів та звітів для цієї бази даних.

На практичному занятті 3 - розглядаються основи створення інформаційно-логічної моделі реляційної бази даних.

На практичних заняттях 4,5 - розглядаються основи використання мови VBA для ефективної роботи з базами даних

Заняття носять практичний характер, в кожному з них викладена їх методика виконання і завдання для самостійної роботи.

Практичне заняття № 1

СТВОРЕННЯ БАЗИ ДАНИХ З ОДНІЄЇ ТАБЛИЦІ

Мета: познайомитися з основними поняттями бази даних; навчитися створювати таблицю бази даних в СУБД Access; навчитися обробляти дані в таблиці

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

СУБД Access дозволяє зберігати великі масиви даних у певному форматі та обробляти їх, уявляючи в зручному для користувача вигляді.

Access містить набір інструментів для управління базами даних, що включає в себе конструктори таблиць, форм, запитів та звітів. Окрім того, Access можна розглядати як середовище для розробки додатків. Використовуючи макроси для автоматизації задач, можна створювати такі ж потужні, орієнтовні на користувача додатків, як й додатки, що розроблені за допомогою «повноцінних» мов програмування, доповнювати їх кнопками, меню та діалоговими вікнами.

Майстер (Wizard) – спеціальна програма для вирішення якої-небудь задачі або створення об'єкта визначеного типу. Ця програма допомагає користувачеві за декілька хвилин виконати рутинну роботу, на яку інакше потрібно декілька годин.

Програма-майстер задає питання щодо змісту, стилю та формату об'єкта без якого –небудь втручання зі сторони користувача. В Access є біля сотні майстрів, що призначені для проектування бази даних, додатків, таблиць, форм, звітів, графіків, елементів керування та властивостей.

Усі складові бази даних (таблиці, звіти, запити, форми та об'єкти) в Access зберігаються в єдиному дисковому файлі. Основним структурним компонентом бази даних є таблиця. В таблицях зберігаються усі введені користувачем дані. Зовні кожна таблиця Access 2007 схожа на звичайну таблицю та складається з стовпчиків, що мають назву поля, та рядків, що мають назву записами. Кожний

запис таблиці містить усю необхідну інформацію щодо окремих елементів бази даних. Наприклад, запис про пристрій може містити модель пристрою, комплектація, до якої установки входить та ін.

При розробці структури таблиці перш за все визначають назви полів, з яких вона повинна складатись, типи полів та їх розміри. Кожному полю таблиці привласнюється унікальне ім'я, яке не може містити більше 64 символів, при цьому за ім'ям повинна дознаватися функція поля. Далі вирішують, данні якого типу будуть міститися в кожному полі. В Access можна вибирати будь-які з основних типів даних, які відповідно привласнюються кожному полю. Значення типу поля може бути задано тільки в режимі конструктора. В табл. 1.1 представлені типи даних Access.

В Access існує чотири способи створення пустої таблиці.

1. Використання майстра баз даних для створення бази даних, що містить всі потрібні звіти, таблиці та форми, за одну операцію (майстер баз даних створює нову БД, тобто його не можна використовувати для додавання нових таблиць, форм, звітів в вже існуючу базу даних).

2. Використання майстра таблиць, що дозволяє вибрати поля для даної таблиці з безлічі визначених раніш таблиць, таких як ділові контакти, список устаткування або технології.

3. Введення даних безпосередньо в пусту таблицю в режимі таблиці. При збереженні нової таблиці в Access дані аналізуються, та кожному полю привласнюється необхідний тип даних та формат.

4. Визначення всіх параметрів макета таблиці в режимі конструктора.

Введення даних в комірки таблиці виробляється звичайним чином. Проте для деяких типів даних (числових, грошових, дата/час, логічного) Access автоматично перевіряє правильність їх введення. Наприклад, якщо ввести букву в комірку з числовим типом, тоді Access видасть повідомлення щодо невірною введеного значення та не дозволить перейти до іншої комірки, поки не будуть введені вірні дані.

Таблиця 1.1. Типи даних Access

Тип даних	Використання	Розмір	Примітки
Текстовий	Алфавітно-цифрові дані	До 255 байтів	Зазвичай вибирається для символічних даних
Поле МЕМО	Алфавітно-цифрові данні – речення, абзаци, тексти	До 64000 байтів	Слід використовувати у випадках, коли розмір тексту може перевищувати 255 символів, або коли в тексті застосовані символи форматування (наприклад табуляція)
Числовий	Числові дані	1, 2, 4, або 8 байтів	Вибір значення властивості „Розмір поля” визначатиме точність значень даних і об’єм пам’яті, необхідний для їх збереження.
Дата/Час	Дати і часу	8 байтів	Використовується для збереження календарних дат або значень часу і дозволяє виконувати вирахування в одиницях часу (хвилинали, годинах, днях, роках, тощо)
Грошовий	Данні про грошові суми, що зберігаються з чотирма знаками після коми	8 байтів	Використовується для зберігання даних про грошові суми
Лічильник	Унікальне довге ціле, яке генерується Access при створенні кожного нового запису	4 байта	Він створений для автоматичної генерації значень первинного ключа. Таблиця не може вміщати більше одного поля типу „Лічильник”
Логічний	Логічні дані	1 біт	Використовується для збереження значень „істина” і „лож”
Об’єкт OLE	Картинки, діаграми і інші об’єкти OLE з додатків Windows	До 1 гігабайта	Дозволяє зберігати такі данні, як картинки, діаграми або звукові фрагменти, які можуть мати динамічні зв’язки з іншими додатками Windows

Для усіх типів полів (окрім *Счетчик (лічильник)* та *Поле об'єкта OLE*) можна самостійно задавати обмеження для даних, що вводяться. Для цього в режимі конструктора необхідно вибрати вкладку *Общие (загальні)*, перевести курсор в поле з ім'ям *Условие на значение (умова на значення)* та ввести обмеження на данні. Обмеження можна вводити й на текстове поле. Зазвичай в цьому випадку задаються слова, що можуть бути присутні в даному полі. Вводити обмеження можна не тільки вручну, але й за допомогою будівника виразів.

Можна використовувати ще один зручний інструмент при введенні даних – параметр *Значение по умолчанию (значення за умовчанням)* (який знаходиться також на вкладці *Общие (загальні)*). Таким чином можна задати дані, які Access буде вводити за умовчанням при заповненні таблиці. Це зручно використовувати, коли більшість значень даного поля однакові. Наприклад, назва більшості пристроїв – плазмотрон. Якщо ввести це слово в рядок параметра *Значение по умолчанию (значення за умовчанням)*, тоді всі комірки поля *Пристрій* примуть значення *Плазмотрон*, та необхідно буде тільки змінити значення комірок для пристроїв з іншими назвами. Дані можна вводити та копіювати їх з однієї комірки в іншу стандартними засобами Windows.

Для кожного типу даних (окрім *Счетчик (лічильник)*) передбачено пусте значення, при цьому розрізняють порожні (Null) значення та порожні рядки. Якщо порожнє поле має порожнє значення, то це значить, що дані для нього існують, але поки не відомі. Якщо ж введений порожній рядок (два знаки прямих лапок “ ”), це означає, що дані не існують зовсім. Access дозволяє обробляти такі порожні значення.

Для зручності перегляду можна сортувати записи в таблиці у визначеній послідовності, наприклад в таблиці ПРИСТРОЇ записи можна відсортувати у порядку зменшення потужності. Кнопки сортування на панелі інструментів (або команди меню *Записи, Сортировка, Сортировка по убыванию/возрастанию*) дозволяють сортувати стовпці за збільшенням або убуванням. Перед тим як натиснути мишкою на кнопку сортування, необхідно вибрати поля, що

використовуються для сортування. В Access засоби сортування даних (а також фільтрація, пошук та заміна) реалізовані як запити, що створюються автоматично. Записи таблиці завжди розміщуються в файлі бази даних в тому порядку, в якому вони були додані в таблицю.

Відбір даних може виконуватись за допомогою фільтра, тобто набору умов, що застосовуються для відбору підмножина записів. В Access існують фільтри чотирьох типів: фільтр за виділеним фрагментом, звичайний фільтр, розширений фільтр та фільтр по введенню.

Фільтрування даних в Access виконується за допомогою двох кнопок: [Фільтр по виділенному] та [Изменить фильтр] (команди меню *Записи, Фільтр, Изменить фильтр*). Після натискання другої кнопки від таблиці залишається один запис. При цьому кожне поле становиться полем із списком (коли в ньому знаходиться курсор), в якому можна вибрати усі значення для даного поля. Після натискання мишкою на кнопку [Применить фильтр] вибираються записи, що відповідні зміненому фільтру. Ще більш складні умови фільтрації можна задати командами меню *Записи, Фільтр, Расширенный фильтр*.

Завдання на практичне заняття

1. Створіть нову базу даних.

2. Створіть структуру таблиці бази даних з інформацією згідно табл.1.2. Необхідно створити самий простий варіант бази даних, коли вся інформація зберігається в одній таблиці. Створимо базу даних технічних характеристик плазмотронів Інституту газу НАН України (табл. 1.2). Визначить поля таблиці, що створюється (у відповідності з табл.1.2). Збережіть створену таблицю.

3. Відсортуйте значення в таблиці. Необхідно відсортувати плазмотрони по зменшенню продуктивності за керамічним, а потім за металевим матеріалом. Зробити висновки.

4. Зробіть пошук та заміну даних в таблиці. Необхідно знайти в таблиці марку плазмотрону ПГ-2 та замінити його на ПГ-2М. Виконувати завдання необхідно за допомогою кнопок сортування.

5. За допомогою майстра форм створіть форму **МАРКИ ПЛАЗМОТРОНІВ** (в один стовпець). Продивитися, як буде виглядати форма на аркуші паперу.

6. Захистіть базу даних паролем.

Таблиця 2. Технічних характеристик плазмотронів Інституту газу НАН України

№	Параметри	ПГ-1	ПГ-2	П-80	П-100
1.	Струм дуги, А	120...250	150...350	150...350	150...450
2.	Напруга, В	135...150	160...220	200...250	240...280
3.	Тиск газу, МПа	0,15...0,5	0,15...0,5	0,35...0,6	0,08...0,5
4.	Витрата газів, м ³ /год				
	Повітря	2,5...4	4...6	4...12	4...20
	Природного газу	0,3...0,7	0,4...1,0	0,1...2,0	0,4...2,5
5.	Витрата води охолодження, м ³ /год	0,5	0,8	0,8	1,5
6.	Продуктивність, кг/год				
	Керамічному матеріалу	6	8	10	15
	Металевому матеріалу	10	15	25	30

Технологія виконання завдання

1. Таблицю можна створювати в режимі **Конструктор**. В ній буде 8-м полів згідно табл.1.

Перше поле - код (в табл. 1.1 - стовпець «№») – унікальний ключ запису (зазвичай використовується для зв'язку записів з різними таблицями).

1.1. Ввійдіть у програму **Access**. Для цього два рази клацніть по піктограмі Microsoft Office Access на робочому столі комп'ютера, або натисніть Пуск – Все програми - Microsoft Office - Microsoft Office Access.

1.2. Відкриється діалогове вікно управління базами даних, в якому необхідно вибрати піктограму «Новая база» та записати ім'я бази даних, наприклад «Плазмотрони інституту газа НАН України», вибрати місце для створення бази даних на комп'ютері (виберіть диск D). Далі натисніть кнопку **Создать** (Створити).

Відкрилось вікно нової бази даних, що наведена на рис.1.1.

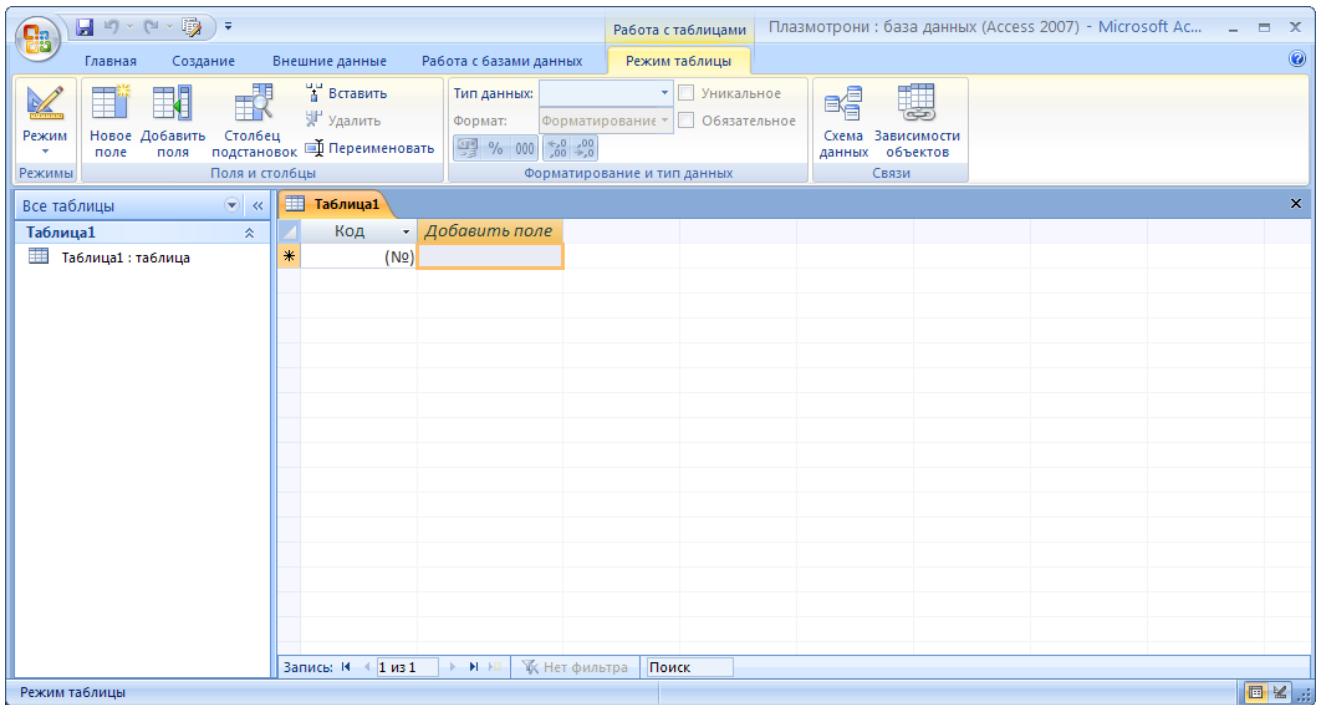


Рис.1.1. Вигляд нової бази даних в Access.

2.1. Вибираємо режим **Конструктор**. Відкрилось вікно програми, що наведено на рис.1.2.

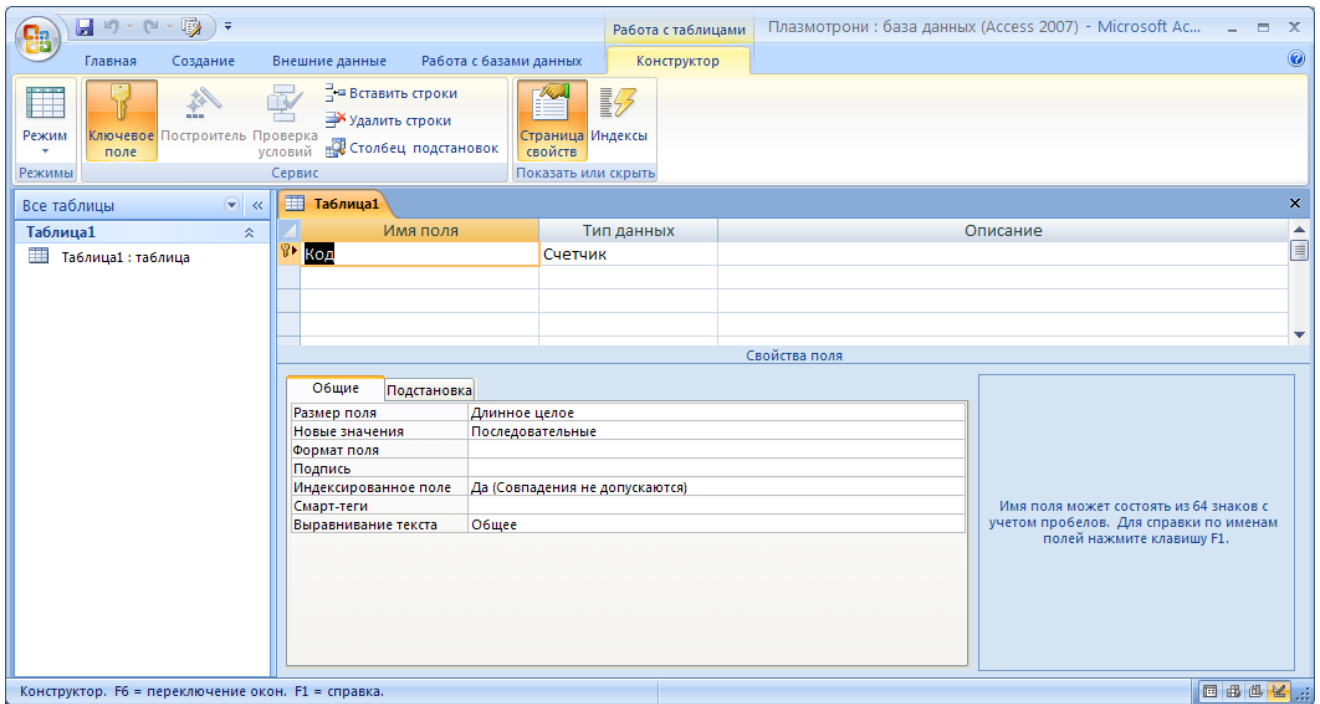


Рис.1.2. Вигляд нової бази даних в режимі Конструктор

2.2. В стовпчику «Имя поля» автоматически з`явиться *Код* та тип даних для нього – *Счетчик* (лічильник).

2.3. Необхідно продовжити заповнення полів нижче Коду введіть стовпчик Параметри, згідно табл. 1.2. Тип даних для них – Текстовый (Текстовий). Створіть усі необхідні стовпчики (табл.1.1).

Створили таблицю даних в режимі конструктор (рис. 1.3).

Имя поля	Тип данных
Код	Счетчик
Тип плазмотрону	Текстовый
Струм дуги, А	Текстовый
Напруга робоча, В	Текстовый
Тиск газу, МПа	Текстовый
Витрата повітря, м3/год	Текстовый
Витрати природного газу, м3/год	Текстовый
Витрата води охолодження, м3/год	Текстовый
Продуктивність за керамічним матеріалом, кг/год	Текстовый
Продуктивність за металевим матеріалом, кг/год	Текстовый

Рис.1.3. Таблица даних в режимі конструктора

2.4. Перейдіть в режим таблиці.

2.5. Відредагуйте стовпчики таблиці.

Натисніть на назву кожного стовпчика та виберіть **Ширина столбца...** та натисніть кнопку *«По ширине данных»*.

2.6. Заповніть таблицю згідно табл.1.1.

Створили базу даних, що має одну таблицю (рис. 1.4).



Тип плазмотрону	Струм дуги, А	Напряга робоча, В	Тиск газу, МПа	Витрата повітря, м3/год	Витрати природного газу, м3/год
ПГ-1	120...250	135...150	0,15...0,5	2,5...4	0,3...0,7
ПГ-2	150...350	160...220	0,15...0,5	4...6	0,4...1,0
П-80	150...350	200...250	0,35...0,6	4...12	0,1...2,0
П-100	150...450	240...280	0,08...0,5	4...20	0,4...2,5
*					

Рис.1.4. Частина таблиці нової бази даних

3. Сортування даних в таблиці.

3.1. Відсортируйте плазмотрони за продуктивністю від максимуму до мінімуму (за керамічним та металевими порошками).

3.2. Виберіть стовпчик *Продуктивність за керамічним матеріалом/Продуктивність за металевим матеріалом*, натисніть праву кнопку миші та виберіть **Сортировать от максимального к минимальному**.

4. Коли в таблиці дуже багато даних, буде корисним команда **Найти** (знайти).

4.1. Знайдіть плазмотрон типу ПГ-1 та замініть його назву на ПГ-1М за допомогою команди **Найти** (Знайти). Для цього виберіть стовпчик «Тип плазмотрону», натисніть правою кнопкою миші та виберіть **Найти** (Знайти).

4.2. В діалоговому вікні (закладка **Поиск** (Пошук) за умовчанням), в полі «Образец» наберіть ПГ-1. В закладці **Заменить** наберіть ПГ-1М та натисніть кнопку «Найти далее». В таблиці виділиться комірка з назвою ПГ-1 та заміниться її назва на ПГ-1М.

5. Для створення форми **МАРКИ ПЛАЗМОТРОНІВ** необхідно:

- відкрити вкладку *Формы* в вікні бази даних;
- клацнути по кнопці [Создать];
- в діалоговому вікні, що з'явилося, вибрати пункт *Мастер форм*;

- вибрати в списку таблицю, що створили;
- вибрати поле марка плазмотрону;
- вибрати вигляд форми *Форма в один стовець*;
- вибрати стиль оформлення;
- задати ім`я форми.

6. Захистіть базу даних паролем.

5.1. Відкрийте в монопольному режимі створену базу даних.

Натисніть кнопку [Microsoft Office], а потім виберіть команду «Открыть» (Відкрити). В діалоговому вікні відкрити знайдіть файл, який потрібно відкрити, і виділіть його. Клацніть стрілку поряд з кнопкою «Открыть» (Відкрити) і виберіть команду «Монопольно».

5.2. На вкладці «Средства базы данных» (засоби бази даних) в групі «Средства базы данных» клацніть **Зашифровать с помощью пароля** (Зашифрувати за допомогою пароля).

5.3. Введіть надійний пароль та натисніть кнопку ОК.

Контрольні питання

1. Дати коротку характеристику СУБД Access.
2. Перелічить сервісні можливості Access.
3. Перелічить типи даних, що допускаються для використання в Access.
5. Що уявляє собою та як здійснюється сортування та фільтрація даних?
6. Коротко описати технологію створення бази даних.
7. Якими способами здійснюється заповнення бази даних?
8. Описати технологію введення та перегляду даних за допомогою форми.

Практичне заняття № 2

ФОРМУВАННЯ ЗАПИТІВ ТА ЗВІТІВ ДЛЯ ОДНОТАБЛИЧНОЇ БАЗИ ДАНИХ

Мета: придбання навичків роботи в СУБД Access щодо формуванням запитів та звітів для однотобличної бази даних

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1 Формування запитів на вибірку

Запити є потужним засобом обробки даних, що зберігаються в Access.

За допомогою запитів можна продивлятися, аналізувати та змінювати дані з декількох таблиць. Вони також використовуються в якості джерела даних для форм та звітів. Запити дозволяють розраховувати підсумкові значення та виводити їх в компактному форматі, що подібний формату електронної таблиці, а також виконувати розрахунки над групами записів.

Запити можна створювати самостійно і за допомогою майстрів. Майстри запитів автоматично виконують основні дії в залежності від відповідей користувача на поставлені питання. Самостійно розробляти запити можна в режимі конструктора.

В Access можна створювати наступні типи запитів:

- запит на вибірку;
- запит з параметрами;
- перехресний запит;
- запит на зміну (видалення, оновлення та додавання записів, створення таблиць);
- запити SQL (запити на об'єднання, запити до серверу, керуючі та підлеглі запити).

Запит на вибірку використовується найчастіше. При його виконанні дані, що задовольняються вимогам відбору, вибираються з однієї або декількох таблиць та виводяться у визначеному порядку.

Також можливе використання запитів на вибірку, щоб згрупувати запис для розрахунку суми, середнього значення, перерахунку та інших дій.

Запит з параметрами – це запит, при виконанні якого у відповідному діалоговому вікні користувачеві видається запрошення ввести дані, на основі яких він буде виконуватися. Наприклад, часто потрібні дані про те, які пристрої входять в ту чи іншу установку. В цьому випадку замість створення окремих запитів по кожній установці можна створити один запит з параметрами, в якому у якості параметра буде використовуватися назва установки. При кожному визові цього запиту пропонується ввести назву установки, після чого на екрані виводяться всі поля, що вказані в запиті, наприклад, назва установки та пристрої, що входять до її складу.

Для створення нового запиту необхідно в вікні бази даних вибрати вкладку *Запросы* та клацнути мишею по кнопці «Создать». В вікні, що відкрилось *Новый запрос* необхідно вибрати один з п'яти пунктів: *Конструктор*, *Простой запрос*, *Перекрестный запрос*, *Повторяющиеся запросы*, *Записи без подчиненных*.

Конструктор дозволяє самостійно створювати любий тип запиту, але цей режим рекомендується користувачам, що вже мають деякий досвід створення запитів. Простий запит дозволяє створювати за допомогою майстра запитів на вибірку з визначених полів таблиць або інших запитів.

При виконанні запитів на вибірку Access витягує записи з таблиці та формує результуючий набір даних, який виглядає, як таблиця, не будучи нею. Результуючий набір даних – це динамічний (або віртуальний) набір записів, що не зберігається в базі даних, тобто після закриття запиту результуючий набір даних цього запиту припиняє своє існування.

При збереженні запиту залишається тільки його структура – перелік таблиць, список полів, порядок сортування, обмеження на записи, тип запиту та ін. Запит при збереженні в базі даних в порівнянні з результуючим набором даних має ряд переваг:

- на фізичному носії інформації (зазвичай це жорсткий диск) для нього потрібно менше місця;

- в ньому можуть використовуватися оновлені версії будь-яких записів, що змінені із часом останнього запуску.

При кожному виконанні запит звертається до базових таблиць та знову створює результуючий набір даних. Оскільки сам по собі результуючий набір даних не зберігається, запит автоматично відображає будь-які зміни, що трапились в базових таблицях з моменту його останнього запуску.

Для збереження запиту необхідно зробити наступні дії. Виконати команду *Файл, Сохранить* або клацнути мишкою по кнопці «Сохранить» на панелі інструментів. Якщо цей запит зберігається вперше, необхідно ввести його нове ім'я в діалоговому вікні *Сохранение*.

2.2 Формування звітів

Звіт – це гнучкий та ефективний засіб для організації перегляду та друку стогової інформації. В звіті можна отримати результати складних розрахунків, статистичних порівнянь, а також помістити в нього рисунки та діаграми.

Користувач має змогу розробити звіт самостійно або створити його за допомогою майстра. Майстер з розробки звітів швидко виконає усю рутинну роботу. Після виклику майстра виводяться діалогові вікна із запрошенням ввести необхідні дані, і звіт створюється на основі відповідей користувача.

При роботі з майстром в залежності від того, який звіт необхідно створити, Access пропонує різноманітні варіанти макетів.

Основна відміна між звітами та формами полягає в їх призначенні. Якщо форми призначені переважно для вводу даних, тоді як звіти – для їх перегляду (на екрані або на папері). В форматі використовуються поля, що розраховуються (зазвичай за допомогою розрахунків на основі полів у поточному запису). В звітах поля, що розраховуються формуються на основі загальної групи записів, сторінки записів або усіх записів звіту. Все, що можна створити з формою (за випадком введення даних), можна зробити і із звітом.

Для створення звіту необхідно відкрити вкладку *Отчеты* та клацнути мишкою по кнопці «Создать», після чого відкриється вікно *Новый отчет*, в якому приведені шість способів створення звітів: *Конструктор*, *Мастер отчетов*, *Автоотчет в столбец*, *Автоотчет ленточный*, *Мастер диаграмм*, *Почтовые наклейки*.

Конструктор дозволяє самостійно створити звіт.

Майстер звітів автоматично створює звіт на основі вибраних полів таблиць (записів) та макетів звітів. Цей спосіб створення звітів є найбільш зручним.

Автоотчет в столбец та *Автоотчет ленточный* – найпростіші способи створення звітів, тобто тут достатньо вказати тільки ім'я таблиці (запиту), на основі якої буде створюватися звіт, все інше зробить майстер звітів.

Майстер діаграм допоможе створити звіт у вигляді діаграми.

Завдання на практичне заняття

1. На основі таблиці УСТАНОВКИ ДЛЯ ПЛАЗМОВОГО НАПИЛЕННЯ (табл. 2.1) створіть простий запит на вибірку, в якому повинні відображатися тип установки, тип плазмотрону, потужність. Збережіть запит.

2. На основі таблиці УСТАНОВКИ ДЛЯ ПЛАЗМОВОГО НАПИЛЕННЯ створіть звіт з групуванням даних по типу установки. Збережіть звіт.

Технологія виконання завдання


Створіть нову базу даних. Задайте базі даних ім'я УСТАНОВКИ ДЛЯ ПЛАЗМОВОГО НАПИЛЕННЯ.

Введіть дані в базу згідно табл.2.1.


1. Для створення простого запиту необхідно:

- в вікні бази даних відкрити закладку *Запросы*;
- в вікні, що відкрилося натиснути на кнопку [Создать];
- вибрати з пункту вікна *Новый запрос – Простой запрос*;
- вибрати необхідну таблицю - УСТАНОВКИ ДЛЯ ПЛАЗМОВОГО

НАПИЛЕННЯ;

- в вікні *Доступные поля* перевести виділення на параметр *Тип установки*;
- клацнути мишкою по кнопці  , в результаті чого словосполучення *Тип установки* перейде у вікно *Выбранные поля*;
- аналогічно в вікно *Выбранные поля* випередити поля *Тип плазмотрону* та *Потужність*;
- задайте ім'я запиту;
- клацніть мишкою по кнопці [Готово].

Для збереження запиту необхідно:



- клацнути мишкою по кнопці  або виконати команду *Файл, Сохранить*;
- закрити вікно запиту.

Таблиця 2.1. Технічні характеристики обладнання для плазмового напилення

Тип установки	Тип плазмотрону	Робочий газ	Витрата плазмоутворювального газу, м ³ /год	Продуктивність за порошком, кг/год, не більше	Потужність, кВт, не більше
Київ-7	ПУН-1	Повітря, суміш повітря і вуглеводневого газу	Повітря – 3,9...12 Природного газу – 0,1...2	Керамічним – 10 Металевим - 25	Установки – 100 Розпилювача - 80
УМП-6	ПГП-1	азот	4	Керамічним – 4,5 Металевим – 7	Установки – 50 Розпилювача - 28
УМП-7	УМП-7	азот (аргон)	3	Керамічним – 5 Металевим – 12	Установки – 50 Розпилювача - 30
УПУ-3Д	ПП-25	аргон (азот)	0,5...6	Керамічним – 2 Металевим – 6	Розпилювача - 16
УПУ-8	УПУ-8М	Аргон (азот)	1,8...2,9	Керамічним – 5 Металевим – 20	Установки – 120 Розпилювача -40

2. Для створення звіту необхідно:

- відкрити вкладку *Отчеты* та клацнути мишкою по кнопці [Создать];

- в вікні, що відкрилося, вибрати пункт *Мастер отчетов*;
- клацнути мишкою по знаку списку, що розкривається в нижній частині вікна;
- вибрати із списку, що з'явився, таблицю **УСТАНОВКИ ДЛЯ ПЛАЗМОВОГО НАПИЛЕННЯ**;
- клацнути мишкою по кнопці [ОК]. В вікні, що з'явилося, вибрати поля, які будуть присутні в формі. Так як в даній таблиці будуть присутні всі поля з таблиці, необхідно клацнути по кнопці ;
- клацнути мишкою по кнопці [Далее];
- в вікні, що з'явилося, наведений список полів. Переведіть виділення на поле *Тип установки*;
- клацніть мишкою по кнопці , задаючи таким чином групування даних за типом установки;
- клацнути мишкою по кнопці [Далее];
- виберіть макет та стиль для звіту;
- клацніть мишкою по кнопці [Готово], після чого на екрані з'явиться сформований звіт.

Контрольні запитання

1. Що таке запити? Якими можливостями вони володіють?
2. Перерахуйте та охарактеризуйте основні типи запитів, що використовуються в СУБД Access.
3. Що таке звіти? Якими можливостями вони володіють?
4. Коротко опишіть технологію створення запитів на вибірку.
5. Опишіть технологію створення звітів із групуванням даних (на прикладі створення звіту по таблиці **УСТАНОВКИ ДЛЯ ПЛАЗМОВОГО НАПИЛЕННЯ** з групуванням даних по типу установки).

Практичне заняття № 3

РОЗРОБКА ІНФОРМАЦІЙНО-ЛОГІЧНОЇ МОДЕЛІ РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ

Мета: придбання навичків роботи в СУБД Access щодо розробки інформаційно-логічної моделі та створенню структури реляційної бази даних

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Слово «реляційна» походить від англійського relation – відношення. Відношення – математичне поняття, але в термінології моделей даних відношення зручно зображати у вигляді таблиць, рядків які відповідають кортежам відношень, а стовпці – атрибутам. Ключем називають будь-яку функцію від атрибутів кортежу, яка може бути використана для його ідентифікації. Така функція може бути значенням одного з атрибутів (простий ключ) або задаватися алгебраїчним виразом, що включає в себе значення декількох атрибутів (складений ключ). Не дивлячись на те, що дані в рядках кожного з стовпців складеного ключа можуть і повторюватися, комбінація даних кожного рядка цих стовпців є унікальною.

В Access можна виділити три типи ключем полів: простий ключ, складений ключ та зовнішній ключ.

Однією з важливих переваг реляційних баз даних складається в тому, що можна зберігати логічні згруповані дані в різних таблицях та задавати зв'язки між ними, об'єднуючи ці таблиці в єдину базу.

Для завдання зв'язку таблиці повинні мати поля з однаковими іменами або хоча б з однаковими форматами даних. Зв'язок між таблицями встановлює відношення між значеннями в цих полях, що співпадають. Така організація дозволяє зменшити надмірність даних, що зберігаються, спрощує їх введення та організацію запитів та звітів. Пояснимо це на деякому прикладі.

В Access можна задати три типи зв'язків між таблицями: *один до багатьох*, *багато до багатьох*, *один до одного*.

Найбільш частіше використовується тип зв'язку між таблицями *один до багатьох*. При такому зв'язку кожному запису в таблиці *A* може відповідати

декілька записів в таблиці **B** (поля з цими записами називають зовнішніми ключами), а запис в таблиці **B** не може мати більш одного відповідного його запису в таблиці **A**.

При зв'язку типу *багато до багатьох* одного запису в таблиці **A** може відповідати декілька записів в таблиці **B**, а одного запису в таблиці **B** – декілька записів таблиці **A**. Така схема реалізується тільки за допомогою третьої (єднальної) таблиці, ключ якої складається хоча б з двох полів, одне з яких є загальним з таблицею **A**, а друге – загальним з таблицею **B**.

При зв'язку типу один до одного запис в таблиці **A** може мати не більше одного зв'язаного запису в таблиці **B**, та навпаки. Цей тип зв'язку використовується не дуже часто, оскільки такі дані можна помістити в одну таблицю. Зв'язок типу *один до одного* застосовують для розділу дуже широких таблиць, відділення частини таблиці у цілях її захисту, а також для збереження відомостей, що відносяться до підмножини записів в головній таблиці.

Тип зв'язку, що створюється залежить від полів, до яких визначався цей зв'язок:

- зв'язок типу *один до багатьох* задається в тому випадку, коли тільки одне з полів є ключовим або має унікальний індекс, тобто значення в цьому полі не повторюються;
- зв'язок типу *один до одного* задається в тому випадку, коли обидва поля, що зв'язані є ключовими або мають унікальні індекси;
- зв'язок типу *багато до багатьох* фактично уявляють собою два зв'язки типу *один до багатьох* через третю таблицю, ключ якої складається хоча б з двох полів, що є загальними для двох інших таблиць.

Цілісність даних визначає систему правил, що використовуються в СУБД Access для підтримки зв'язків між записами в зв'язаних таблицях, а також забезпечує захист від випадкового видалення або зміни даних, що зв'язані. Контролювати цілісність даних можна при виконанні наступних вимог:

- поле, що зв'язане однією таблицею є ключовим полем або має унікальний індекс;

- поля, що зв'язані мають один тип даних. (Тут існує виключення: поле лічильника може бути зв'язане з числовим полем, якщо воно має тип *Длинное целое*);
- обидві таблиці належать одній базі даних Access.

Для установки цілісності даних база даних, в якій знаходяться таблиці, повинна бути відкрита. Для таблиць, що зв'язані з баз даних інших форматів встановити цілісність даних неможливо.

Завдання на практичне заняття

1. Створити базу даних Деканат у відповідності з логічною схемою, що наведена на рис. 1.1.
2. Створити структуру таблиці СТУДЕНТИ.
3. Створити структуру таблиці ДИСЦИПЛІНИ.
4. Створити структуру таблиці ВИКЛАДАЧІ.
5. Створити структуру таблиці ОЦІНКИ.
6. Розробити схему даних, тобто задати зв'язки між таблицями, що створені.
7. Створити форму *Студенти*.
8. Заповнити даними таблицю СТУДЕНТИ, використовуючи форму *Студенти*.
9. Створити форму *Дисципліни*.
10. Заповнити даними таблицю ДИСЦИПЛІНИ використовуючи форму *Дисципліни*.
11. Створити форму *Оцінки*.
12. Заповнити даними таблицю ОЦІНКИ, використовуючи форму *Оцінки*.

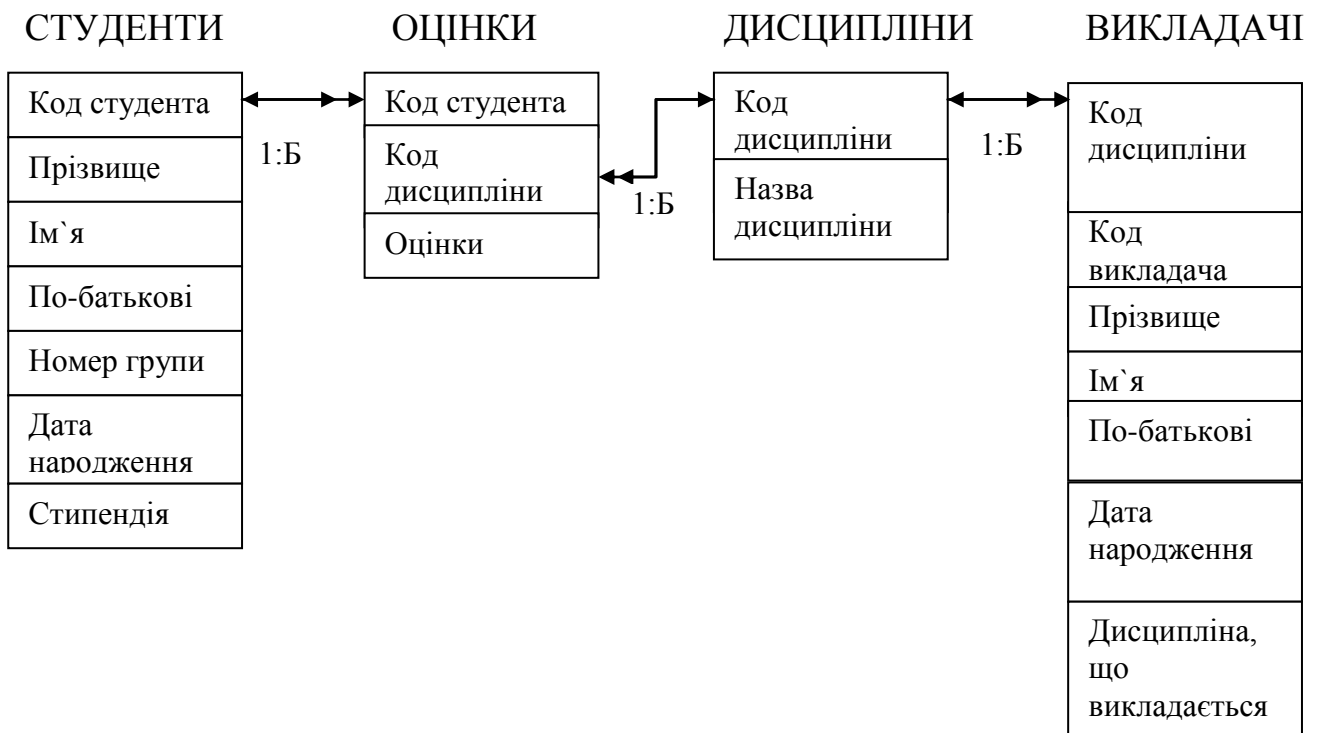


Рис.2.1. Логічна схема бази даних: 1:Б – зв'язок типу *один до багатьох*

Контрольні питання

1. Що таке реляційна база даних?
2. Що таке ключ? Якими бувають ключі?
3. Охарактеризувати існуючі типи зв'язків між таблицями.
4. Поясніть, що таке цілісність даних.
5. Поясніть побудову інформаційно-логічної моделі бази даних на прикладі

БД Деканат.

Практичне заняття № 4

ВИКОРИСТАННЯ МОВИ VBA ПРИ РОБОТІ З ОБ'ЄКТАМИ БАЗИ ДАНИХ

Мета: ознайомлення з можливостями мови Visual Basic for Applications (VBA) при створенні та роботі з базами даних, а також з основними об'єктами бази даних – таблицями, полями, формами

СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

VBA є повнофункціональним об'єктно-орієнтовною мовою програмування, що дозволяє створювати додатки користувача в середовищі СУБД Access. VBA – це загальний засіб програмування для всього сімейству Microsoft Office.

VBA доречно використовувати для створення нестандартних процедур обробки функцій, що виконують складні розрахунки, які неможливо записати у вигляді виразів.

Основою програм на VBA є процедури, що складаються з інструкцій, які виконують необхідні операції та розрахунки. Процедури зберігаються в модулях, з яких вони запитуються на виконання. Модуль не виконується, але служить для об'єднання процедур за їх функціональному призначенню або прив'язування до форми або звіту.

Розрізняють процедури-підпрограми та процедури-функції.

Процедури-підпрограми (Sub) не повертають значення процедури, що визивається.

Процедура-підпрограма може виконувати будь-яку дію, наприклад корегувати дані бази, виконувати розрахунки в полях, відчиняти форми, друкувати звіти.

Формат процедури-підпрограми має наступний вигляд:

[Private! Public] [Static] Sub <ім`я процедури> [(список аргументів)]

[<опис змінної >]

[< інструкція>]

[Exit Sub]

[< інструкція >]

End Sub

Тут Sub, End – відмічають відповідно початок та кінець тіла процедури;

Public – вказує, що процедура Sub є загальною, тобто доступною для всіх інших процедур у всіх модулях; Private – вказує, що процедура Sub доступна тільки для процедур того модуля, в якому вона описана; Static – вказує, що значення локальних змінних процедур зберігається між викликами цієї процедури; Exit Sub – приводить до негайного завершення процедури Sub.

Процедура – функція (Function) повертає значення, яке привласнюється її імені усередині процедури.

Формат процедури-функції має наступний вигляд:

[Private! Public] [Static] Function <ім`я процедури> [(список аргументів)]

[<опис змінної >]

[< інструкція>]

[< ім`я процедури>=< вираз>]

[Exit Function]

[< інструкція >]

[< ім`я процедури>=< вираз>]

End Function

В тілі процедури-функції (на відміну від процедури-підпрограми) присутня інструкція привласнення імені процедури значення, що розраховується виразом. Ця інструкція дозволяє повернути значення з процедури-функції в місце її виклику.

При опису змінної зазвичай використовується інструкція привласнення Dim, яка привласнює вираз змінної або константі. Інструкція привласнення

завжди включає в себе знак рівності (=). Для привласнення значення змінної, що описана як об'єкт, використовується інструкція Set.

Завдання на практичне заняття

1. Написати процедуру VBA, що створює нову базу даних.
2. Написати процедуру, що створює в новій базі даних таблицю СТУДЕНТИ з полями *Номер студента*, *ПІБ*, *Дисципліна 1*, *Дисципліна 2*, *Дисципліна 3*, *Дисципліна 4*, *Середній бал*.
3. Внести в створену таблицю п'ять записів в усі поля (окрім *Середній бал*).
4. Створити процедуру, що підраховує середній бал всіх студентів та занести отримані значення в поле *Середній бал*.
5. Створити форму, що відображає дані таблиці СТУДЕНТИ

Технологія виконання завдання

1. Відкрити нову базу даних та вкладку *Модули*, написати процедуру, що створює нову базу даних із збереженням її на диску D:

Створення нової бази даних

Sub CreateDatabaseX()

Опис змінних

Dim myWs As Workspace

Dim myDb As Database

Визначаємо стандартний об'єкт Workspace (робочий простір)

Set myWs = DBEngine.Workspaces(0)

Створюємо нову базу даних з вказаним порядком символів, що вказується

dbLangGeneral

**Set myDb = myWs.CreateDatabase ("D:\NewDB.mdb",
dbLangGeneral)**

myDb.Close

End Sub

В результаті виконання процедури на диску D повинна з'явитися нова база даних з назвою NewDB.mdb.

2. Створення нової таблиці СТУДЕНТИ в створеній базі даних

Sub CreateTableDefX ()

Визначаємо змінні

Dim myDb As Database

Dim myTab As TableDef

Dim myF As Field

Set myDb = CurrentDb ()

Створюємо новий об'єкт TableDef – таблицю СТУДЕНТИ

Set myTAB = myDb.CreateTableDef ("Студенти")

Створюємо новий об'єкт Field – текстове поле *Номер студента* та додаємо його до сімейства полів об'єкта таблиці СТУДЕНТИ

Set myF = myTab.CreateField ("Номер студента", dbInteger)

myTab.Fields.Append myF

Створюємо новий об'єкт Field – текстове поле *ІІІБ* та додаємо його до сімейства полів об'єкта таблиці СТУДЕНТИ

Set myF = myTab.CreateField ("ІІІБ", dbText)

myTab.Fields.Append myF

Створюємо новий об'єкт Field – поле *Дисципліна 1* та додаємо його до сімейства полів об'єкта таблиці СТУДЕНТИ

Set myF = myTab.CreateField ("Дисципліна 1", dbInteger)

myTab.Fields.Append myF

Аналогічно робимо з іншим полями таблиці.

Set myF = myTab.CreateField ("Дисципліна 2", dbInteger)

myTab.Fields.Append myF

.....

Set myF = myTab.CreateField ("Середній бал", dbDouble)

myTab.Fields.Append myF

Додаємо об'єкт таблицю СТУДЕНТИ до сімейства таблиць бази даних

myDb.TableDefs.Append myTab

End Sub

3. Відкриваємо створену таблицю та вносимо п'ять записів в усі поля (окрім поля *Середній бал*).
4. Створюємо процедуру, що підраховує середній бал всіх студентів та заносимо отримані значення в поле *Середній бал*.

Private Sub SB ()

Dim myDb As Database `об'єктна змінна типу бази даних

Dim myRec As Recordset `об'єктна змінна типу набору записів

Dim sb As Double `змінна для розрахунку середнього балу

Dim i As Integer `змінна циклу

Dim max As Integer `змінна для зберігання числа записів в таблиці

Set myDb = CurrentDb () `Працюємо з поточною базою даних

`Відкриваємо набір записів таблиці СТУДЕНТИ та привласнюємо послання на нього об'єктній змінній myRec

Set my Rec = myDb.OpenRecordset ("Студенти")

i= 0

myRec.MoveLast `Йдемо до останнього запису таблиці

max = myRec.RecordCount `При цьому RecordCount містить число записів в таблиці, яке необхідно нам для розрахунку середнього бала в кожному рядку таблиці

myRec.MoveFirst `Переходимо до першого запису таблиці та розраховуємо середній бал

Do While i < max

**sb = (myRec! Дисципліна1+ myRec! Дисципліна2+ myRec!
Дисципліна3+ myRec! Дисципліна4)/4**

myRec.Edit `Заносимо значення середнього бала в
однойменне поле

myRec! [Середній бал] = sb

myRec.Update `Для внесення даних в поля таблиці
обов`язково використовуються команди Edit та Update.

myRec.MoveNext `Переходимо до наступного запису таблиці
та повторюємо все поки не досягнемо останнього запису.

i = i+1

Loop

`Зачиняємо набір записів.

myRec.Close

End Sub

5. Створюємо форму, що буде відображати дані таблиці СТУДЕНТИ .

Контрольні запитання

1. Дати визначення макросу. Якими можливостями володіють макроси?
2. Дати визначення модулю. Якими можливостями володіють модулі?
3. Описати технологію створення процедур на VBA.
4. Описати технологію створення баз даних на VBA.
5. Описати технологію створення таблиць на VBA.

Практичне заняття № 5

ВИКОРИСТАННЯ МОВИ VBA ДЛЯ ФІЛЬТРАЦІЇ ДАНИХ В БАЗІ

Мета: ознайомлення з можливостями фільтрації даних в базі за допомогою мови Visual Basic for Applications (VBA)

Завдання на практичне заняття

1. Створити таблиці бази даних та встановити зв'язки між ними.
2. Створити простий запит до таблиць бази даних.
3. Створити форми, що утворюють складну форму з підлеглою.
4. Розмістити в формі елемент управління *Набір вкладок* та написати процедуру обробки події, що виконує відбір (фільтрацію) даних в формі у відповідності з вибором вкладки.

Технологія виконання

1. Відкрити нову базу даних (можна використовувати вже існуючу), для чого виконати наступне:
 - Створити таблицю КАТЕГОРІЯ ДЖЕРЕЛА з поля *Код категорії* (ключове поле) та *Найменування категорії* (книжки, журнали, статті, довідники та ін.);
 - Створити таблицю КАТАЛОГ ДЖЕРЕЛ з полями *Код джерела* (ключове поле), *Код категорії*, *Автор*, *Найменування*, *Рік видання*, *Число сторінок* та ін.;
 - Встановити зв'язок між таблицями типу один до багатьох від таблиці КАТЕГОРІЯ ДЖЕРЕЛА до таблиці КАТАЛОГ ДЖЕРЕЛ;
 - Занести в таблиці дані.
2. Створити простий запит до таблиць КАТАЛОГ ДЖЕРЕЛ та КАТЕГОРІЯ ДЖЕРЕЛА, що відбирає усі поля та усі записи.

3. Створити форму на основі запиту, що створений в п. 2.
4. Створити пусту форму та розмістити в ній підлеглу форму, що буде створена в п.3. В цій же формі розмістити елемент управління *Набір вкладок* з найменуванням *Категорія* та вкладками, імення кожної з яких відповідає категоріям, що використовуються (книжки, журнали та ін.). Натискання на відповідну вкладку набору повинне визивати відбір (фільтрацію) даних в підлеглий формі.
5. В вікні властивостей елемента управління *Набір вкладок* в графі *Изменение* вибрати опцію *Процедура обработки события*, яка буде реагувати на вибір відповідної категорії.

В тілі процедури повинен міститися набір записів наступного типу:

```

If Me! [Категория] = 1 Then
Me! [Каталог джерел]. Form.RecordSource = "SELECT*FROM
[Каталог джерел Запрос]. [Наименование категории] Like 'Книжки';"
End If

      If Me! [Категория] = 2Then
.....
End If

.....

```

Тут вираз `Me![Категория] = 1` вказує, що якщо в поточній формі в елементі управління *Набор вкладок* з іменем *Категории* вибрана вкладка з індексом 1 (в даному випадку вкладка *Книжки* має індекс 1, вкладка *Журнали* – 2 та таке інше), тоді джерелом рядків (`RecordSource`) для підлеглий форми з іменем *Каталог джерел* є запит з іменем *Каталог джерел Запрос* з умовами відбору записів по найменуванню категорії *Книжки*.

6. Перевірити правильність роботи форми з відбором записів по категорії.

Контрольні питання

1. Описати технологію створення процедури обробки події на VBA.
2. Написати процедуру створення форми на VBA.
3. Написати процедуру створення звіту на VBA.
4. Описати основні властивості форм, що доступні при програмуванні на VBA.

ЛІТЕРАТУРА

1. Базы данных: учеб. Пособие для студ. высш. учеб. заведений / А.В. Кузин, С.В. Левонисова. – 2 –е изд., стер. - М.: Издательский центр «Академия», 2008. – 320 с.
2. Кошелев В.Е. Access 2007. Эффективное использование. – М.: ООО «Бином-Пресс», 2008. – 592 с.
3. Пащенко В. М. Обладнання для газотермічного нанесення покриттів.– К.: ІВЦ “Політехніка”, 2001. – 416 с.