

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ

Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до виконання лабораторних робіт з навчальної дисципліни «Бази даних»  
(1 частина) для студентів денної та заочної форми навчання за  
спеціальностями 123 «Комп'ютерна інженерія»,  
125 «Кібербезпека»

ЗАТВЕРДЖЕНО  
кафедрою кібербезпеки та  
програмного забезпечення,  
протокол від 18.08. 2020 року № 1

КРОПИВНИЦЬКИЙ  
2020

Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни «Бази даних» (1 частина) для студентів денної та заочної форми навчання за спеціальностями 123 «Комп'ютерна інженерія», 125 «Кібербезпека» / уклад. В.В. Босько, Л.В. Константинова — Кропивницький: ЦНТУ, 2020. — 77 с.

Укладач: Босько В. В., Константинова Л.В.

Рецензенти: Смірнов О. А., д-р техн. наук, професор;  
Мелешко Є.В., канд. техн. наук.

© Босько В. В., Константинова Л.В., укладання, 2020  
© Центральноукраїнський національний  
технічний університет, 2020

# Вступ

В методичних рекомендаціях представлено теоретичний і практичний матеріал для роботи з базами даних за допомогою СКБД Microsoft Access. Дані методичні рекомендації починають знайомити з основними прийомами роботи з базами даних. Приводиться багато практичних прикладів та ілюстрацій, які допомагають краще засвоїти викладений матеріал. Розглядається початок роботи з БД за допомогою СКБД Microsoft Access, ознайомлення з основними об'єктами БД, робота з таблицями, створення зв'язків між таблицями. Приводиться матеріал та приклади до поняття цілісності даних, розглядаються типи та можливості QBE-запитів. Приводиться побудова інтерфейсу користувача за допомогою форм. Розглядається отримання підсумкової інформації бази даних за допомогою звітів.

## Початок роботи

Організація даних є ключовим моментом під час роботи з великою кількістю за об'ємом інформації. Дуже важливо впорядкувати дані таким чином, щоб легко та швидко знаходити потрібну інформацію.

Програма Microsoft Access входить до складу пакету Microsoft Office та представляє собою грандіозну систему, яка забезпечує ефективну розробку та супровід баз даних.

Ми розглянемо русифіковану версію СКБД Access. Вона дозволяє розв'язувати широке коло завдань користувачів без програмування. Одна з основних переваг СКБД Access полягає в тому, що вона має прості та зручні засоби обробки кількох таблиць в одній базі даних. Таблиця - це місце зберігання даних. Вона є основним об'єктом бази даних.

У Access є зручні та наочні засоби зв'язування таблиць, які зберігаються в одному файлі.

Збереження кількох таблиць в одній базі даних має цілий ряд переваг. Основні з них такі:

1. Можливість розподілу даних за таблицями згідно зі змістом даних. Наприклад, у першій таблиці зберігаються звітні дані вчителів, у другій - їх тижневий розклад, а в третій - відомості про позаурочну діяльність.

2. Табличне, тобто модульне, подання даних дозволяє модернізувати кожну таблицю окремо.

Редагування даних в одному місці бази призводить до автоматичного їх редагування в іншому місці. Наприклад, якщо в таблиці Список (мал. 1.) стирається запис із прізвищем "Кірін Т.В.", то відповідно запис буде стерто і в таблиці Консультація (мал. 2.).

### Список

№ п/п	П.І.Б.	Предмет	ДатаНар	Стаж	Навантаження	Оклад
1	Костенко А.М.	Математика	12.01.70	3	18	140
2	Носенко С.К.	Фізика	20.05.55	15	18	180
3	Борисов Н.П.	Хімія	16.04.38	32	24	250
4	Кірін Т.В.	Фізика	29.11.50	18	24	220
5	Стойко Б.С.	Література	17.08.46	24	20	200
6	Абрамов К.К.	Математика	22.12.61	7	22	160

Малюнок 1

## Консультації

№ з/п	П.І.Б.	Аудиторія	День	Час
1	Костенко А.М.	215	Вторник	15,00-17.00
2	Носенко С. К.	307	Понедельник	16.00-18.00
3	Борисов Н.П.	112	Среда	15.00-17.00
4	Кирич Т. В.	307	Пятница	16.00-18.00
5	Стойко Б.С.	230	Вторник	15.00-18.00
6	Абрамов К. К.	215	Четверг	15.00-17.00

Малюнок 2

## Теми лабораторних робіт, що розглядаються та оцінювання знань

Дані методичні рекомендації містять у собі матеріал з 7 лабораторних робіт за наступними темами:

<b>Теми лабораторних робіт</b>
<b>1.</b> Загальні поняття ІС. Архітектура ІС. Робота з таблицями та їх редагування.
<b>2.</b> Моделювання даних. Модель «об'єкт-атрибут-зв'язок». Створення зв'язків між таблицями. Створення схем даних. Типи відношень та цілісність даних.
<b>3.</b> Теоретичні мови запитів. Проектування та можливості QBE запитів. Запити на вибірку.
<b>4.</b> Мова реляційного числення за зразком QBE. Обчислювальні поля у запитах. Маніпулювання даними. Параметричні запити, перехресні запити.
<b>5.</b> Побудова інтерфейсу користувача. Майстер форм. Режим конструктора форм.
<b>6.</b> Побудова інтерфейсу. Робота з елементами керування в формах бази даних. Різні види форм. Побудова меню роботи з БД.
<b>7.</b> Побудова інтерфейсу. Макроси. Звіти.
<b>Всього:</b>

Основним призначенням ІС є оперативне забезпечення користувача інформацією про зовнішній світ шляхом реалізації питально-відповідного відношення. Питально-відповідні відношення дозволяють виділити для ІС певний її фрагмент - предметну область (ПО), - який буде втілений в автоматизованій ІС.

Предметна область - частина реального світу, яка описується і відображається в базі даних (фірма, підприємство, школа тощо).

Вибравши предметну область, над якою ви будете працювати, ви повинні виконати завдання до лабораторних робіт, а також відповісти на питання вкінці кожної лабораторної роботи. Звіт повинен містити хід виконання завдань а також графічні матеріали, що підтверджують виконання цих завдань. Приклад звіту знаходиться у додатку 1.

Форма підсумкового контролю: залік.

Максимальну кількість балів студент може одержати у випадку відвідування всіх лекцій, лабораторних занять, виконання і захисту виконаних завдань для самостійної роботи у встановлений термін, проходження контролю.

При виконанні і захисті лабораторних робіт після встановленого терміну, одержані бали перераховуються з коефіцієнтом: для самостійної роботи студента -0,3; лабораторної роботи -0,7.

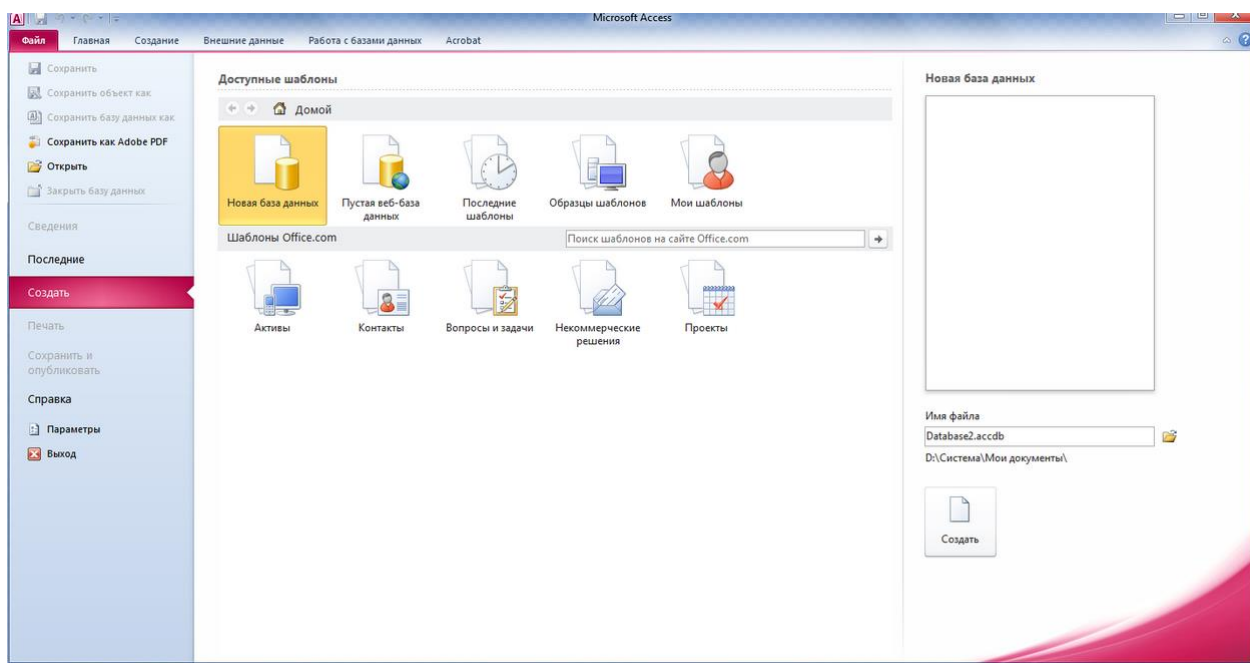
Необхідно виконати відповідні завдання за темами (предметними областями) згідно обраної студентом і узгодженої з викладачем.

### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
		для екзамену, курсової роботи
90 – 100	<b>A</b>	відмінно
82-89	<b>B</b>	добре
74-81	<b>C</b>	
64-73	<b>D</b>	задовільно
60-63	<b>E</b>	
35-59	<b>FX</b>	незадовільно з можливістю повторного складання
0-34	<b>F</b>	незадовільно з обов'язковим повторним вивченням дисципліни

Запуск СКБД Access можливо здійснити у кілька способів.

Найлегший спосіб запуску Access - натиснути на кнопку **Пуск (Start)**, вибрати підменю **Програми (Programs)**, а в ньому - пункт Microsoft Access. В результаті Access завантажиться, про що свідчить поява на екрані заставки програми (малюнок 3).



Малюнок 3-Скриншот Microsoft Access 2010

Якщо користувач напевно не знає, який з файлів вибрати, йому потрібно звернути увагу на розширення. Деякі типи файлів, що зустрічаються:

- .accdb - файл бази даних Access (починаючи з 2007 версії)
- .accdc - база даних з цифровим підписом Microsoft Access
- .accde - база даних Microsoft Access (тільки виконання)
- .accdr - база даних Microsoft Access (тільки читання)
- .accdt - шаблон бази даних Microsoft Access
- .accdw - посилка на базу даних Microsoft Access
- .accft - шаблон типів даних Microsoft Access
- .adn - шаблон проекту Microsoft Access
- .adp - проект Access
- .mdbhtml - HTML-файл бази даних Microsoft Access

# Лабораторна робота №1

**Тема: Загальні поняття ІС. Архітектура ІС. Робота з таблицями та їх редагування.**

**Мета: Ознайомитись з типами та властивостями полів в Access. Навчитись створювати таблиці, редагувати їх та оптимально використовувати Access.**

## Зміст роботи за варіантом індивідуального завдання:

1. За вибраною предметною областю побудуйте таблиці (дві і більше) з полями різних типів та визначте їх ключові поля.
2. Відредагувати таблиці своєї бази даних за допомогою перелічених контекстних меню.
3. Зробити пошук.
4. Відсортувати записи.
5. Створіть фільтр.
6. Створіть ярлик таблиці та розмістіть його на робочому столі Windows.
7. Приховайте таблицю вашої БД.

## Теоретичні відомості:

### Типи та властивості полів

У системі Access як і в інших базах даних реляційного типу рядок таблиці ототожнюється з терміном "запис", а колонка — з терміном "поле". Кожне поле має ім'я, тип і властивості. При створенні структури таблиці обов'язково вказують імена і типи полів. Властивості полів можуть прийматися за замовчуванням. Одне або кілька полів необхідно визначити як ключові.

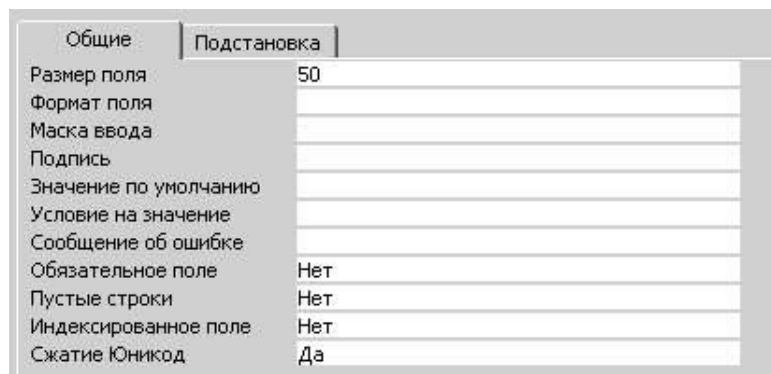
В Access використовують такі типи полів:

1. Текстове поле. У текстовому полі можуть записуватись літери, цифри та інші символи. Текстові поля мають властивості, перелік яких на мал. 4.

Коротко розглянемо суть цих властивостей:

\* **Размер поля.** Він має межі від 1 до 255 символів.

\* **Формат поля.** За допомогою спеціальних символів можна задавати вигляд та розмір рядків, що вводяться. Для того щоб задати формат текстових полів, використовують такі кодові символи: @ - текстовий символ або пропуск; & — текстовий символ; < - перетворення символів у нижній регістр; > — перетворення символів у верхній регістр.



Малюнок 1

\* **Маска ввода.** За допомогою спеціальних символів можна визначити тип символу (літера, цифра, пропуск та ін.) у кожній позиції введення. У масці введення можуть використовуватись такі кодові символи: 0 — цифра від 0 до 9; 9 — цифра чи пропуск; # —

цифра, пропуск, плюс або мінус; L — буква від А до Z чи від А до Я; ? - буква; А — буква або цифра; а — буква або цифра; & — будь-який символ або пропуск; С — довільний символ; .,:;/ - роздільники, які зберігають свій вигляд у рядку даних; < — перетворення символів у нижній регістр; > —

\* перетворення символів у верхній регістр; ! — маска має заповнюватися справа наліво; \ — введення наступного після зворотної похилої лінії символу як символної константи.

\* **Подпись.** По суті це друге ім'я поля. Воно задається рідко.

\* **Значение по умолчанию.** Дозволяє автоматично вставляти рядки, які часто повторюються, у всі новостворені поля.

\* **Условие на значение.** Дозволяє створювати фільтр, який забезпечує введення у дане поле тільки тих значень, які відповідають заданій умові.

\* **Обязательное поле.** Набуває тільки двох значень: "так" або "ні".

"Так" означає, що у дане поле обов'язково мають бути введені дані, які відповідають раніше визначеним властивостям.

\* **Пустые строки.** Набувають двох значень: "так" або "ні". "Так" означає, що порожні рядки дозволені. Тут над порожнім рядком починається рядок, який має пропуски.

\* **Индексированное поле.** Набуває двох значень: "так" або "ні". "Так" означає, що поле індексоване.

Зазначимо, що на початку вивчення та конструювання баз даних звичайно використовують тільки властивість "розмір поля". Інші властивості, як правило, приймаються за замовчуванням. Вони використовуються після набуття деякого досвіду роботи з базами даних.

2. Числове поле. У числове поле можуть записуватись цілі числа і числа у нормальному вигляді, тобто з плаваючою крапкою. Числові поля мають властивості, перелік яких див. на мал. 2. Аналіз цього переліку показує, що вони багато в чому збігаються з переліком властивостей текстового поля.

Общие	Подстановка
Размер поля	Длинное целое
Формат поля	
Число десятичных знаков	Авто
Маска ввода	
Подпись	
Значение по умолчанию	0
Условие на значение	
Сообщение об ошибке	
Обязательное поле	Нет
Индексированное поле	Нет

Малюнок 2

Розглянемо більш детально тільки властивість **Розмір поля**. Вона може набирати таких значень:

\* **Байт** - цілі числа у межах від 0 до 255.

\* **Целое** - цілі числа від - 32 768 до 32 767.

\* **Длинное целое** - цілі числа від -2 147 483 648 до 2 147 483 647.

\* **С плавающей точкой (4 байт)** - числа у межах від -3.402823 E38 до 3.402823 E38.

\* **С плавающей точкой (8 байт)** - числа у межах від - 1.79769313486 E308 до 1.79769313486 E308.

\* **Кодрепликации** – глобально унікальний ідентифікатор (GUID) довжиною 16 байт.

3. Поле **Дата/время**. Це поле використовують для запису дат та часу. Багато із властивостей цього поля такі самі, як і текстового поля. Властивість **Формат поля** може набирати таких значень:

\* **Полный формат даты** (наприклад, 12.05.97 15:20:21).

\* **Длинный формат даты** (наприклад, 13 серпня 1997 р.).

\* **Средний формат даты** (наприклад, 13 серп. 97 р.).

\* **Короткий формат даты** (наприклад, 13.08.97).

\* **Длинный формат времени** (наприклад, 15:20:21).

\* **Средний формат времени** (наприклад, 03:21 PM).

\* **Короткий формат времени** (наприклад, 15:20).

Якщо властивість **Формат поля** набувається за замовченням, то у різні записи цього поля можна вводити різну інформацію. Наприклад в один запис цього поля може бути записано дату в такому вигляді 02/03/97, а в інший запис поля - у такому вигляді 02-03-97.

4. Логічне поле. У логічному полі може записуватись одне з таких двох значень: "так" або "ні", "хибне" чи "істинне", "ввімкнень" чи "вимкнень". Логічні поля можуть використовуватися з різною метою. Але найчастіше їх використовують в анкетних даних, де є тільки дві можливі відповіді.

5. Поле типу лічильник. Використовують як лічильник записів, його також часто використовують як ключове поле.

6. Поле типу Мемо. Це довге поле. У ньому може вміщуватися текст або комбінація тексту та чисел, яка містить 64000 символів.

7. Поле об'єкту OLE. Це поле вміщує безпосередньо не інформацію про об'єкт, а посилання на ім'я об'єкта. Як ім'я об'єкта може бути ім'я додатку, наприклад електронна таблиця Excel, редактор Word, засобами яких можуть бути створені об'єкти для впровадження їх у таблицю Access. Крім того, об'єктами можуть бути малюнки, звукозаписи та інші дані у двійковому форматі.

На початковому етапі роботи з базами даних процес їх створення доцільно починати з підготовчого етапу, а саме: з побудови на папері структури таблиць. Структуру таблиці Список можна побудувати так, як показано на мал. 3.

Ім'я поля	Тип поля	Розмір поля	Інші властивості	Опис
Номер 1	Счетчик	Длинное целое		
П.І.Б.	Текстовый	14		
Предмет	Текстовый	12		
ДатаНар	Дата/время	Кратный формат даты		
Стаж	Числовой	Байт		
Навантаження	Числовой	Байт		
Оклад	Числовой	Целое		

Малюнок 3

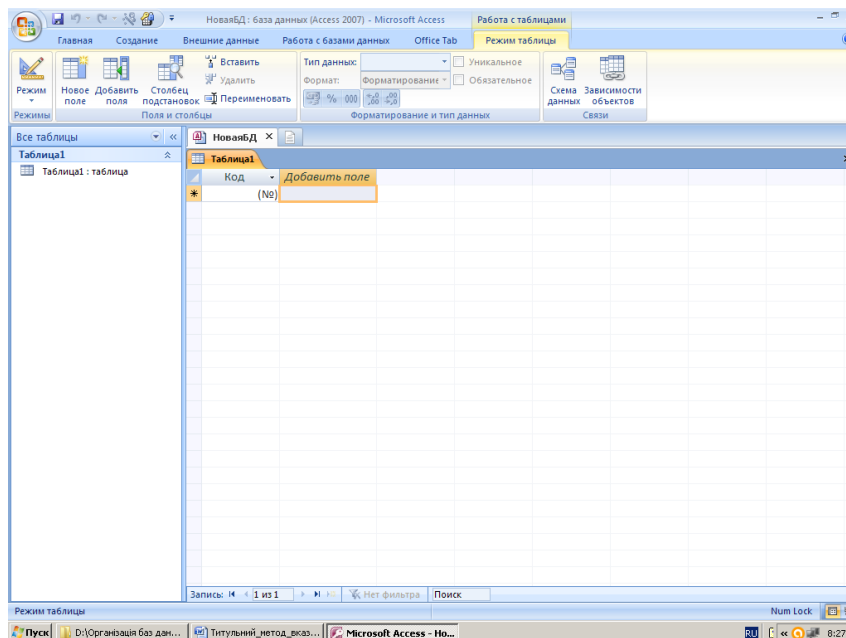
Структуру таблиці Консультації можна подати так, як показано на мал. 4. Запускаємо систему Access. Створюємо нову базу даних.

Ім'я поля	Тип поля	Ромір поля	Інші властивості	Опис
Номер 2	Счетчик			
П.І.Б.	Текстовый	14		
Аудиторія	Числовой	Целое		
День	Текстовый	12		
Час	Текстовый	12		

Малюнок 4

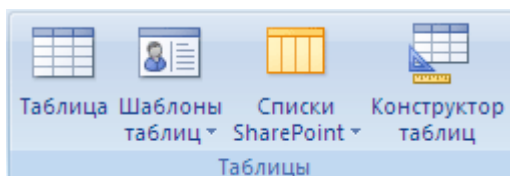
Набираємо ім'я файла, а потім натискаємо кнопку **Создать**. Після цього відкривається вікно зображене на мал. 5.

Зверніть увагу, що на екрані з'явилося головне меню системи. Воно вміщує такі вкладки: Главная, Создание, Внешние данные, Работа з базами даних, Багато команд головного меню системи Access дублюють ті самі операції, які можуть бути виконані іншими засобами. Оскільки більшість операцій з базами даних виконуватимемо за допомогою контекстних меню, то окремо головне меню розглядати не будемо.



Малюнок 5

У вікні нової бази даних вибравши вкладку **Создание** можливо з групи **Таблицы** вибрати **Таблица** мал. 6.



Малюнок 6

В базу даних буде додано таблицю, яка відкриється в режимі таблиці.

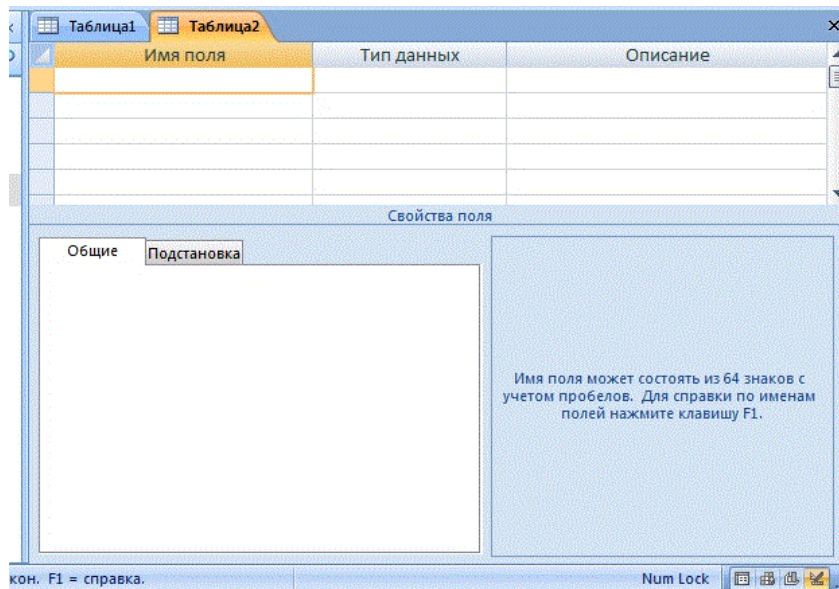
**Режим таблицы** видає бланк (форму) абстрактної таблиці. Потім вона може приймати конкретне наповнення і вміст.

**Шаблоном** називають пусту таблицю, яку можливо застосовувати як вона є або налагодити в відповідності з своїми потребами. В Access 2010 шаблони таблиць та полів замінені моделями.

Спосіб **Импорт таблиц** (вкладка **Внешние данные** група **Импорт**) дозволяє імпортувати дані у таблицю, яка створюється, з інших файлів, у тому числі і з файлів форматів даних, які відрізняються від формату, який використовується СКБД Access.

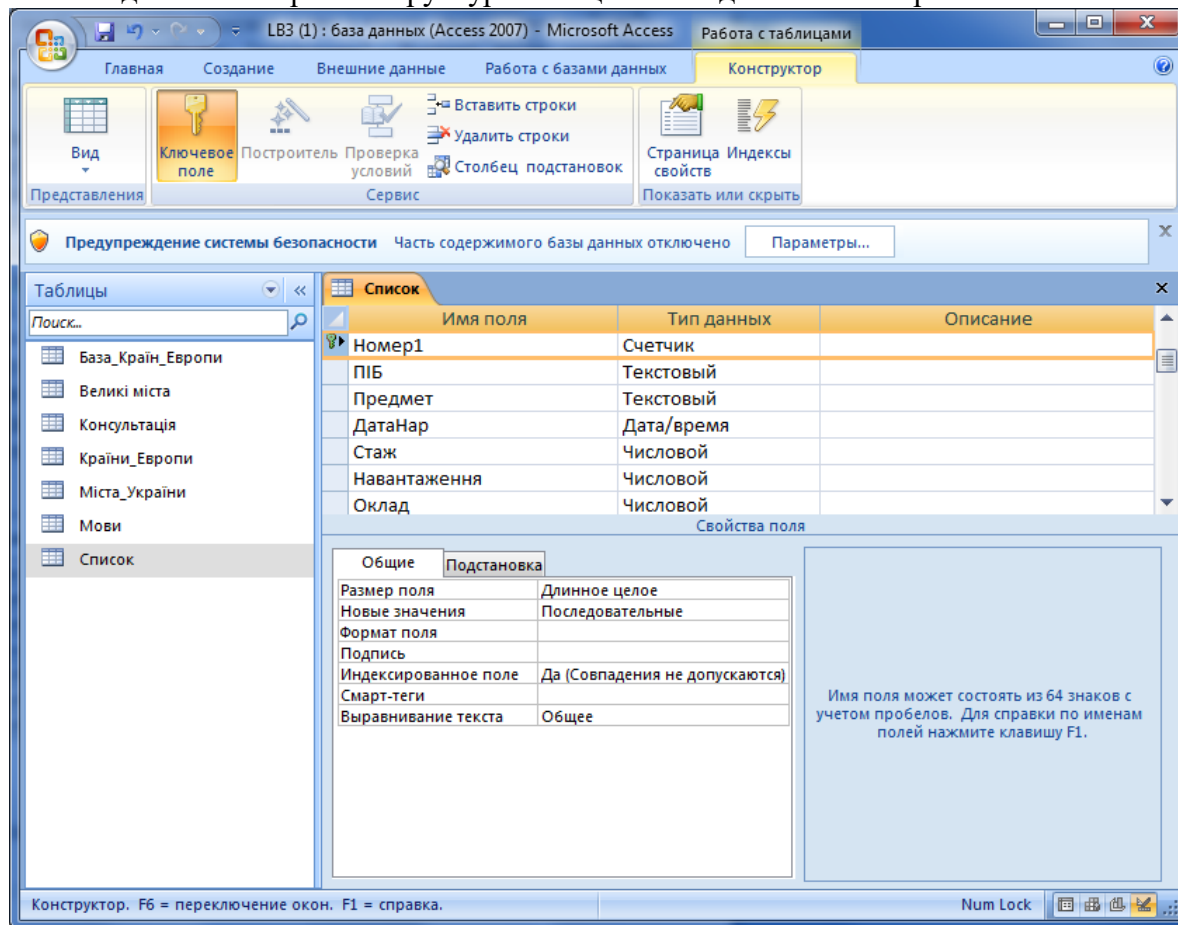
Для створення таблиці БД можливо застосовувати **импортировать список SharePoint** або створити зв'язок з ним. Крім того, можливо створити новий список SharePoint на основі готового шаблону.

Спосіб створення (режим), який називають **Конструктор таблиць**. Вибравши цей режим на екрані з'явиться вікно таблиці у режимі конструктора, зображене на мал. 7. Заповнюємо цю таблицю даними. Кожна таблиця повинна вміщувати одне або декілька полів, що однозначно визначають кожен запис в таблиці. Такі поля називають **первинним ключовим полем** таблиці. Якщо для таблиці визначений первинний ключ, то Microsoft Access не допускає дублювання ключа або введення значень Null в ці поля. Після заповнення всієї таблиці необхідно виділити ключове поле, яким тут буде поле **Номер 1**. Для цього курсор переміщується у відповідне поле, потім вибираєм кнопку **Ключевое поле**, натиснути. При цьому поряд з іменем цього поля з'явиться зображення ключа.



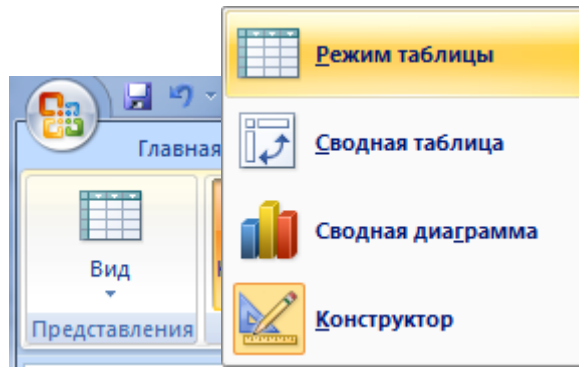
Малюнок 7

Збережемо структуру таблиці та введемо ім'я **Список**, що зображено на мал.8. Аналогічно розглянутому створюється і зберігається структура таблиці Консультація. Заповнення даними створених структур таблиць може здійснюватись різними способами.



Малюнок 8

Можливо переключатися з режиму конструктора в режим таблиці за допомогою меню Вид, що розглянуто на малюнку 9.



Малюнок 9

Аналогічно вносяться дані і в таблицю **Консультації**. Якщо ж після відкриття вікна бази даних потрібно внести зміни у структуру будь-якої таблиці, то потрібно виділити відповідну таблицю і вибрати режим **Конструктор**. На екрані з'явиться раніше створена структура цієї таблиці. Після цього потрібно внести необхідні зміни і обов'язково, згідно з правилами, описаними раніше, зберегти змінену структуру.

### Робота з таблицями

СКБД Access дозволяє виконувати різні операції над даними, які зберігаються у таблицях. При цьому одні і ті самі операції можна виконувати різними способами.

В кінець відкритої таблиці можна додати будь-який запис. Якщо потрібно змінити структуру таблиці, то для цього потрібно її відкрити у режимі конструктора. Після цього можна змінювати імена полів та типи даних, додавати нові поля та ін.

Ми будемо розглядати лише один спосіб роботи з таблицями, заснований на використанні контекстного меню. Це один з найпростіших способів роботи з таблицями. При цьому спочатку розглядатимемо операції тільки в межах однієї таблиці.

Контекстне меню має багато об'єктів бази даних. Кожна таблиця теж має декілька контекстних меню. Ми розглядатимемо такі з них:

- \* контекстне меню стовпців;
- \* контекстне меню рядків;
- \* контекстне меню таблиці;
- \* контекстне меню окремої комірки таблиці;
- \* контекстне меню назви таблиці.

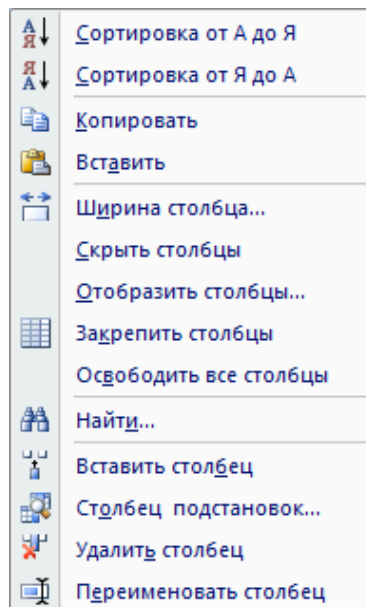
Малюнок 10

Для введення на екран будь-якого з цих меню потрібно виділити відповідний об'єкт, а потім клацнути правою клавішею миші.

Перейдемо тепер до роботи з таблицями за допомогою контекстних меню.

1. Контекстне меню стовпців. При роботі з цим меню експеримент будемо проводити з полем. Предмет у відкритій таблиці Список, яка зображена на мал. 10.

Перш за все активізуємо поле Предмет, (для виділення кількох стовпців ця операція виконується при натискуванні клавіші Shift).

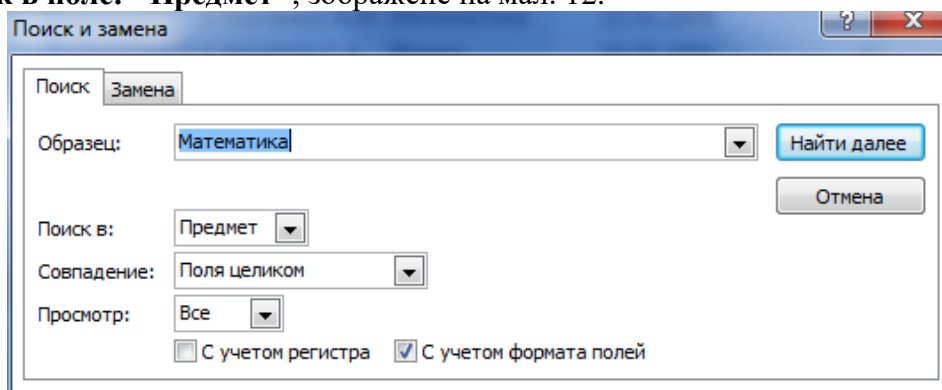


Малюнок 11

На екрані з'явиться контекстне меню поля, яке зображено на мал. 11.

Один з найбільш ефективних засобів організації даних є сортування. Сортування впорядковує дані за змістом визначених полів. Інформацію можна сортувати за алфавітом, числовими значеннями, датами та іншими параметрами. Записи сортуються або в порядку зростання (А-Я і 0-9), або у порядку спадання (Я-А і 9-0).

Команда "**Найти ...**" відкриває вікно **Поиска в поле**. У даному випадку відкриється вікно **Поиск в поле: "Предмет"**, зображене на мал. 12.



Малюнок 12

Після заповнення цього вікна необхідними даними натискають кнопку **Найти далее**. У результаті буде виділено у таблиці **Список** те перше значення поля **Предмет**, яке було введено у рядок **Образец**. Для продовження пошуку значень, які повторюються, необхідно натиснути кнопку **Найти далее**.

Команда **Ширина столбца...** дозволяє установити на екрані необхідну ширину вибраного стовпця.

Команда **Скрыть столбцы** ховає виділені стовпці. На екрані вони стають невидимими. Для відновлення на екрані схованих стовпців у контекстному меню заголовка таблиці необхідно виконати команду **Показать столбцы...**

Команда **Закрепить столбцы** закріплює виділені стовпці з лівого боку екрана, перешкоджаючи їм вийти за край екрана при прокручуванні.

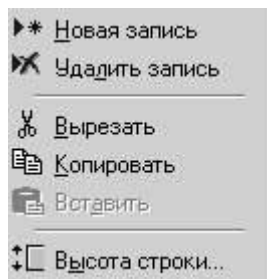
Команда **Переименовать столбец** дозволяє змінити ім'я виділеного поля безпосередньо у відкритій таблиці. Цю команду потрібно застосовувати дуже акуратно, оскільки це може вимагати зміни тих дій, які мали посилання на старе ім'я поля.

При виконанні команди **Вставить столбец** всі стовпці таблиці, починаючи з виділеного, зсуваються праворуч, а на місце виділеного вставляється порожній стовпець з іменем **Поле 1**. Потім йому можна надати конкретне ім'я і заповнити відповідними даними.

Команду **Удалить столбец** використовують для знищення в таблиці виділеного стовпця. Команда потребує підтвердження знищення.

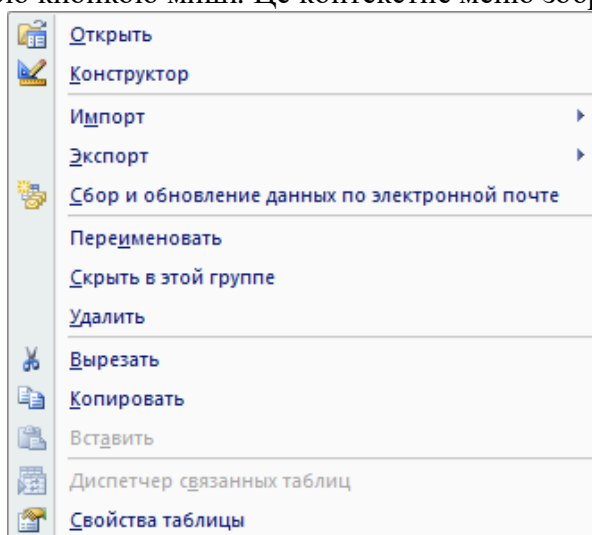
Команду **Столбец подстановка ...** використовують для виклику програми **Мастер подстановок**. Вона буде розглянута пізніше.

2. Контекстне меню рядків. Для виділення рядка курсору миші встановлюють в її крайнє ліве поле і клацають клавішею. Для виділення кількох записів клавіша миші клацається при натиснутій клавіші Shift. Після цього на екрані з'явиться контекстне меню, зображене на мал. 13.



Малюнок 13

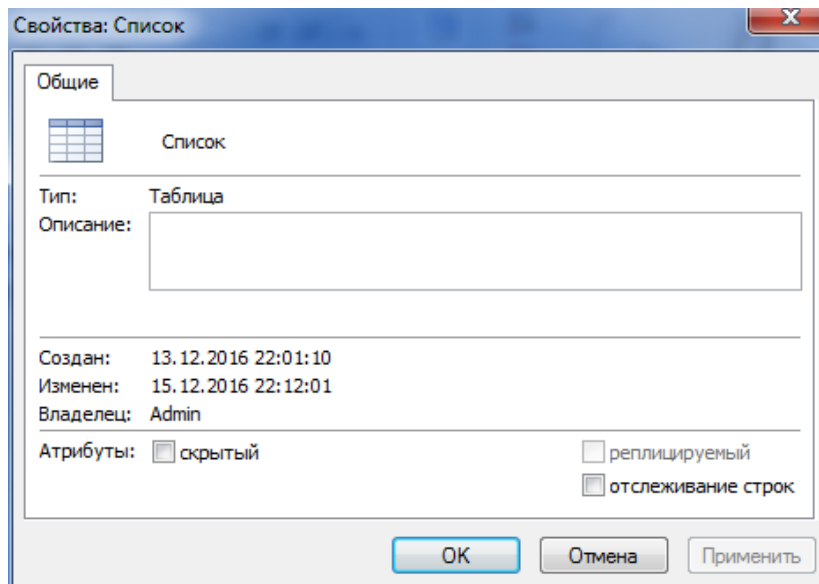
3. Контекстне меню таблиці. Для виклику його встановлюємо курсор миші на назву таблиці і клацаємо правою кнопкою миші. Це контекстне меню зображено на мал. 14.



Малюнок 14

Перетягнувши таблицю за назву можна створити та розмістити ярлик таблиці на робочому столі Windows. Так можна безпосередньо з Windows відкрити таблицю, і швидко отримати доступ до її даних.

Якщо з базою даних працює кілька чоловік, то для деяких з них користування окремими таблицями може бути заборонено. У цьому випадку таблиці можна сховати. Для цього потрібно у вікні **База даних** виділити ім'я відповідної таблиці, викликати контекстне меню таблиці і виконати команду **Свойства...** . У результаті з'явиться вікно **Свойства...**, зображене на мал. 15.



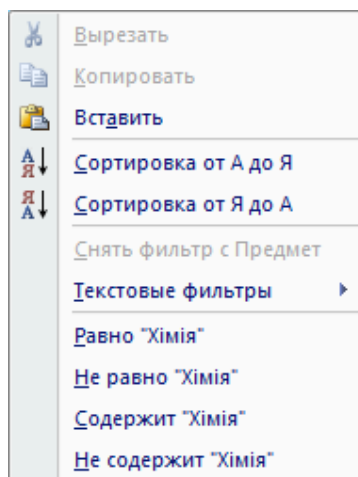
Малюнок 15

У цьому вікні необхідно встановити атрибут **Скрытый, ОК**. У результаті ім'я таблиці буде знищено з списку об'єктів БД.

Щоб відновити ім'я цієї таблиці у вікні бази даних необхідно на вільній області у списку об'єктів БД натиснути правою кнопкою миші, вибрати - **Параметры переходов** встановити прапорець у **Показывать скрытые объекты**, з'являється таблиця (вона буде сіренька). Після цього потрібно у контекстному меню цієї таблиці знищити атрибут **Скрытый**.

4. Контекстне меню окремої комірочки таблиці. Це меню зображене на мал. 16. Для цього курсор миші поміщується в комірчку таблиці і клацається права кнопка.

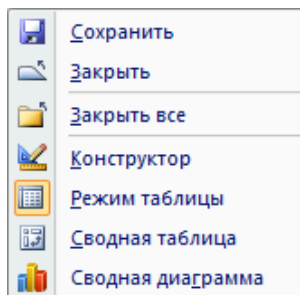
Один з найбільш ефективних засобів організації даних є фільтрація. Фільтрація дозволяє обмежити діапазон перегляду записів. Купа умов відбору називають фільтром. Сортування та фільтрація можуть виконуватись як в режимі форми так і в режимі таблиці. В результаті цих процедур відображення даних на екрані змінюється, хоча реальні дані залишаються незмінними.



Малюнок 16

5. Контекстне меню назви таблиці. Воно зображено на мал. 17.

Для виклику цього меню курсор розміщується у полі імені відкритої таблиці і клацається права клавіша.

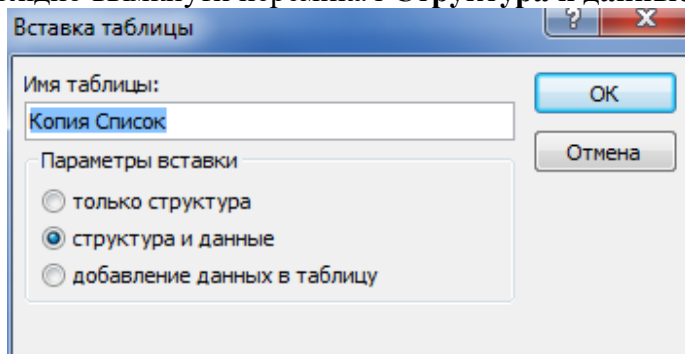


Малюнок 17

Крім розглянутих операцій з даними всередині однієї таблиці СКБД Access має широкий набір засобів для роботи з даними, які зберігаються у кількох таблицях. Можна, наприклад, копіювати рядки і стовпці однієї таблиці в іншу, переносити прямокутні ділянки з однієї таблиці до іншої, створення копії таблиць чи їх структур та ін.

Щоб зробити копію таблиці необхідно викликати контекстне меню таблиці. У цьому меню виконуємо команду **Копировать**. У вільному місці списку об'єктів бази даних і клацаємо правою клавішею. У контекстному меню, що з'явиться вікно **Вставка таблицы**, зображене на мал. 18.

Далі вводимо ім'я таблиці, в якій будемо створювати копію. Якщо потрібно копіювати всю таблицю, то необхідно ввімкнути перемикач **Структура и данные**.



Малюнок 18

### Контрольні питання до лабораторної роботи 1:

1. Що розуміють під СКБД?
2. Перелічіть головні об'єкти баз даних.
3. Які типи полів використовують в Access?
4. Коротко розкажіть про суть властивостей текстового поля.
5. Перелічіть значення які може набирати властивість Розмір поля числових полів.
6. Перелічіть значення які може набирати властивість Формат поля полів Дата/время.
7. Перелічіть способи створення таблиць.
8. Що таке ключове поле?
9. Перелічіть типи ключових полів.
10. Які операції можна виконувати над даними в таблицях?
11. В яких режимах можна працювати з таблицями?
12. Що таке фільтр?
13. В результаті фільтрації і сортування що змінюється, а що залишається незмінним?
14. За допомогою якої команди можна приховати потрібні стовпці вашої таблиці?
15. За допомогою якої команди можна відновити приховані стовпці вашої таблиці?

## Лабораторна робота № 2

**Тема: Моделювання даних. Модель «об'єкт-атрибут-зв'язок». Створення зв'язків між таблицями. Створення схем даних. Типи відношень та цілісність даних.**

**Мета: Навчитись працювати з Схемою даних. Набути навичок створення відношень у БД.**

### **Зміст роботи за варіантом індивідуального завдання:**

1. Візьміть дві таблиці своєї БД, поля яких потрібно зв'язати та створіть зв'язок між таблицями за допомогою майстра підстановок. Показати створений зв'язок на схемі даних.
2. Видалити створений зв'язок у вікні схеми даних. Звільнити макет схеми даних.
3. За допомогою майстра створіть стовпець підстановок, в якому відображається фіксований набір значень.
4. Створити відношення один-до-багатьох між таблицями своєї бази даних у вікні Схема даних.
5. Забезпечте цілісність для ваших даних у зв'язках.
6. Встановіть каскадне видалення. Встановіть каскадне відновлення.
7. Створити зв'язок між таблицями з відношенням багато-до-багатьох.
8. Створіть зв'язок між таблицями з відношенням один-до-одного.

### **Теоретичні відомості:**

#### **Зв'язування таблиць бази даних**

Дублювання даних є причиною нераціонального використання пам'яті комп'ютера.

Прагнення розв'язати цю проблему привело до розробки процедури нормалізації. Нормалізація зараз потрібна для того, щоб організувати дані таким чином, щоб їх редагування робилось тільки в одному місці бази даних. Нормалізація передбачає необхідність виконання процедури зв'язування таблиць бази даних.

Access створює реляційні бази даних, які дозволяють об'єднувати інформацію з різних таблиць.

Необхідно встановити взаємозв'язки між таблицями, записи яких логічно пов'язані. Створивши таблиці та визначивши ключі для кожної з них задаються зв'язки і на їх основі можна брати дані з кількох таблиць та вміщувати їх в одну форму, запит або звіт.

Існує два основних способи зв'язувати дані: це за допомогою полів підстановки та шляхом визначення зв'язків у діалоговому вікні Схема даних.

Між двома таблицями можна встановити два типи відношень: “один-до-одного” та “один-до-багатьох”.

Найбільш розповсюдженим є відношення “один-до-багатьох” (one-to-many), коли одному запису таблиці відповідає декілька записів в другій таблиці. У відношенні “один-до-багатьох” сторона “один” називається головною таблицею, а сторона “багато” – зв'язаною або підлеглою. Таблиця може бути зв'язаною в одному взаємозв'язку, та головною в іншому. Щоб встановити взаємозв'язок таблиць потрібно зв'язати ключ головної таблиці з співпадаючим полем у зв'язаній таблиці (зовнішнім ключем).

Відношення “один-до-одного” (one-to-one) зустрічається набагато рідше. В цьому випадку одному запису в головній таблиці відповідає один запис у зв'язаній таблиці.

1). **Мастер подстановок.** Порядок установления зв'язків будемо розглядати на прикладі двох таблиць. Вміст таблиць **Список предметів** та **Консультації** на мал. 1.

The image shows two overlapping database table windows. The top window, titled 'Список предметів', contains the following data:

Предмет	ПІБ	Кількість годин	Опис	Доб
Біологія	Кірін Т.В.	70		
Інформатика	Абрамов К.К.	70		
Література	Стойко Б.С.	35		
Математика	Костенко А.М.	140		
Фізика	Носенко С.К.	70		
Хімія	Борисов Н.П.	35		

The bottom window, titled 'Консультації', contains the following data:

Номер2	День	Час	Предмет	Аудиторія	Добавил
1	понеділок	8.00	математика	516	
2	вівторок	12.50	фізика	408	
3	середа	9.30	хімія	513	

Малюнок 1

Необхідно, наприклад поле **Предмет** з таблиці **Список предметів** підставити у таблицю **Консультації**.

Для цього відкриємо структуру таблиці **Консультації** у режимі конструктора. У цій структурі створюємо ще одне поле з будь-яким іменем. Використовуватимемо ім'я **Предмет**. Далі відкриємо значення **Тип даних**, де останнім є **Майстер підстановок** (мал. 2.).

The image shows the 'Консультації' table structure in a database management system. The 'Тип даних' (Data Type) dropdown menu is open for the 'Предмет' field. The menu options are:

- Текстовый
- Текстовый
- Текстовый
- Поле MEMO
- Числовой
- Дата/время
- Денежный
- Счетчик
- Логический
- Поле объекта OLE
- Гиперссылка
- Вложение
- Мастер подстановок.

The 'Мастер подстановок.' option is highlighted. The background shows the table structure with columns: Имя поля, Тип данных, and Описание.

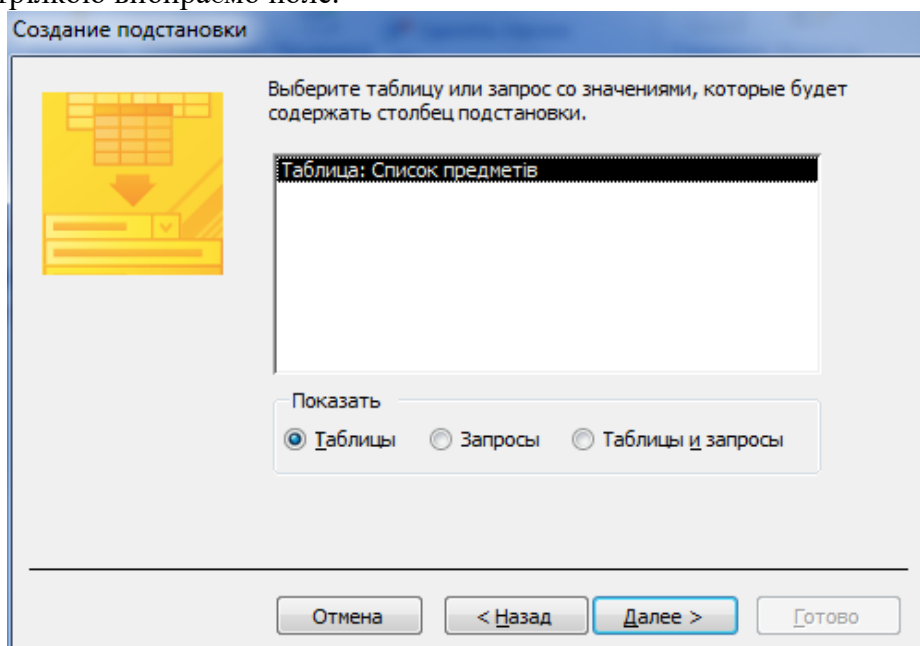
Малюнок 2

Після активізації програми **Мастер подстановок** на екрані відкривається вікно цієї програми. У ньому вибираємо спочатку **Объект «столбец подстановки»** **будет использовать значения из таблицы или запроса** і натискаємо **Далее**

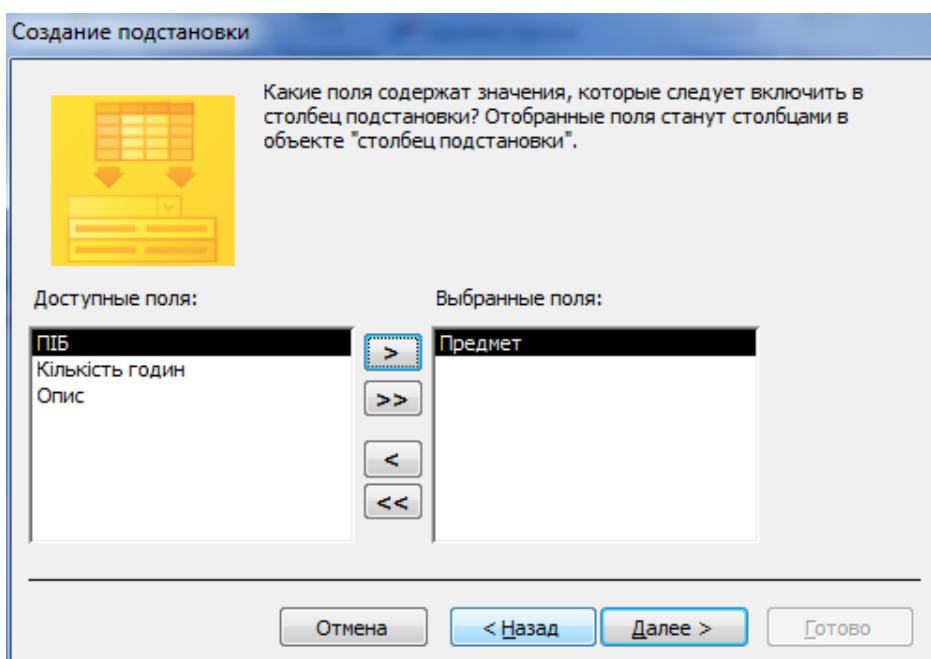
2. У наступному вікні **Мастер подстановок** з'явиться нова інформація, показано на мал. 3. Тут будуть показані всі таблиці БД за винятком тієї, в яку буде зроблена

підстановка поля (для розглянутого прикладу це **Консультації**). У нас це таблиця, **Список предметів**. Вибераємо.

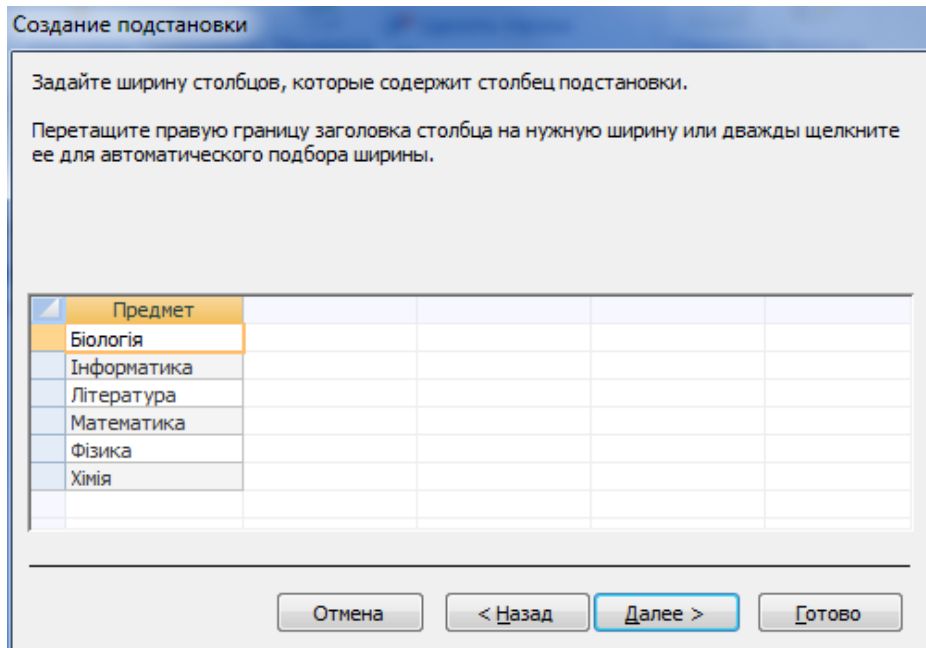
3. У наступному вікні **Мастер** подстановок зміниться (мал. 4). Необхідно виділити те поле, над яким здійснюється операція підстановки. У даному випадку це поле – **Предмет** і стрілкою вибираємо поле.



Малюнок 3



Малюнок 4

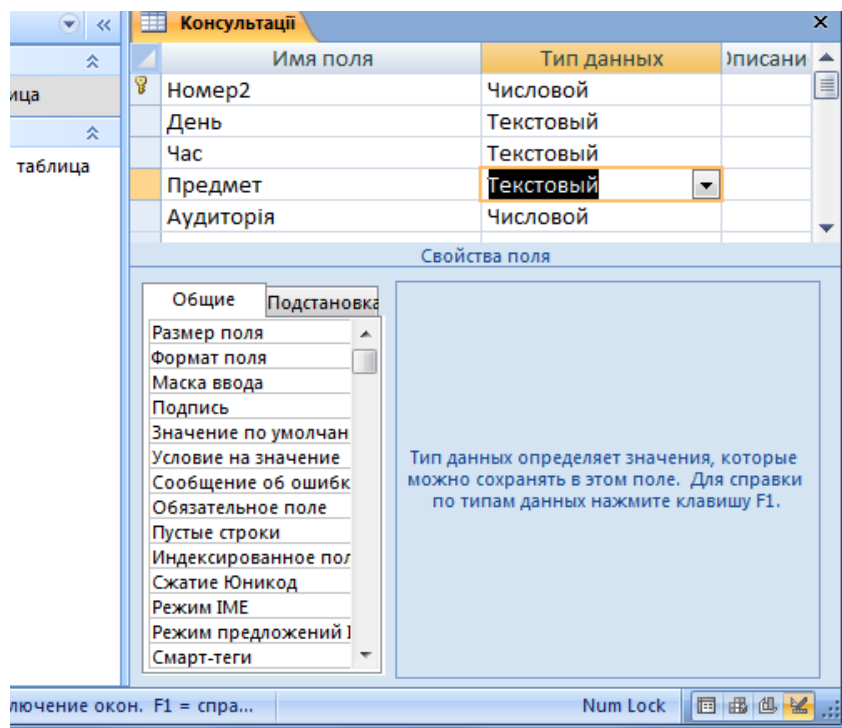


Малюнок 5

1. У наступному вікні є можливість вибрати порядок сортування.
2. У цьому вікні редагуємо ширину стовпця (мал. 5), тиснемо **Далее**.
3. Задаємо підпис для стовпця (мал. 6), залишимо - Предмет. Тиснемо **Готово**.

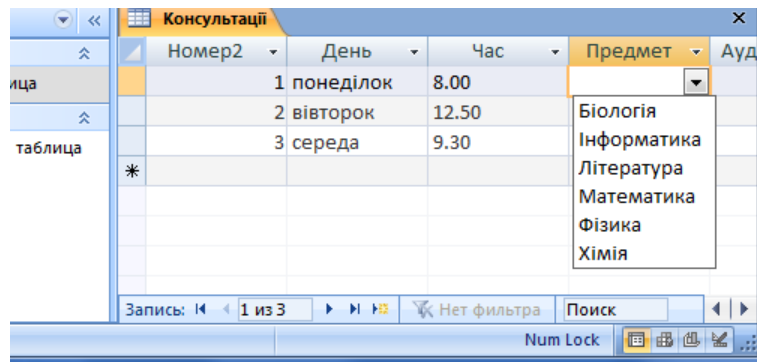
Після цього необхідно підтвердити збереження таблиці, тобто натиснути кнопку **Да**. Як результат - мал. 7.

Закрийте вікно структури таблиці **Консультації** і відкрийте її для перегляду. Клацніть кнопкою миші в позиції першого рядка поля **Предмет** і розкрийте список його значень. Вікно набере вигляду, зображеного на мал. 8.



Малюнок 7

Переконайтесь у тому, що у даному списку зберігаються всі значення поля **Предмет** зв'язаної таблиці **Список предметів**. Тепер кожне з цих значень може бути перенесено у поточну комірку поля **Предмет**.

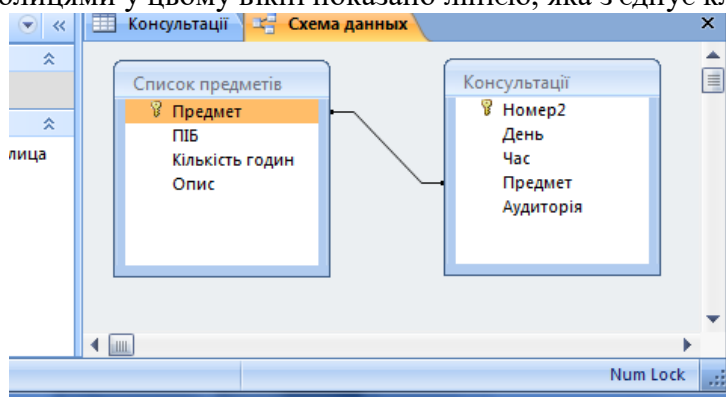


Малюнок 8

Таким чином, поставлене завдання виконано. Поле **Предмет** приєднано до таблиці **Консультації**.

Щоб переконатися в наявності зв'язку між таблицями, відкриємо **Схему даних** з вкладки **Робота с базами даних**. З'явиться вікно **Схема даних** зі зв'язком між таблицею **Список предметів і таблицю Консультації**, зображене на мал. 9.

Зв'язок між таблицями у цьому вікні показано лінією, яка з'єднує ключеві поля таблиць.



Малюнок 9

Надалі робота з таблицями буде здійснена з урахуванням встановлених зв'язків.

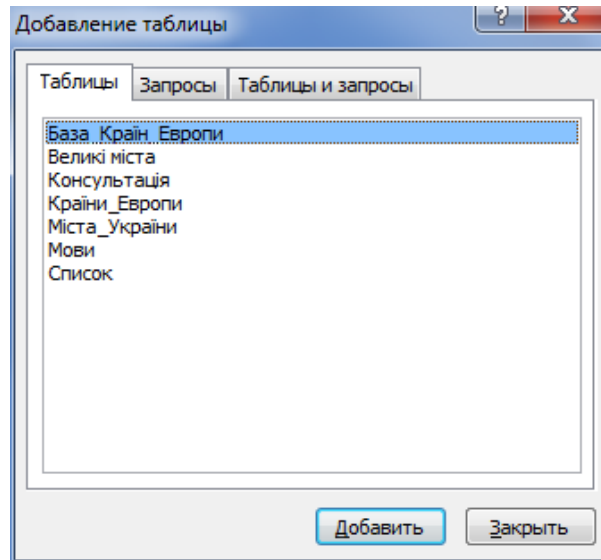
Щоб прибрати зв'язок між таблицями, необхідно клацнути лінію, яка з'єднує таблиці (див. мал. 9) так, щоб вона стала жирною, потім потрібно клацнути правою клавішею миші. З'явиться контекстне меню, в якому необхідно виконати команду **Удалить связь**.

Система Access дозволяє приєднати до іншої таблиці не тільки одне поле, але й декілька. Методика виконання такої операції аналогічна розглянутій вище.

## Створення схем даних

2). Вікно **Схема даних** (Relationships), що розглядалось вище, дозволяє зв'язувати поля з різними іменами та бачити загальну картину відношень між таблицями бази даних.

У відкритому вікні **Схема даних**, якщо є збережений макет схеми даних, цей макет буде виведений на екран. Якщо збереженого макету схеми даних немає. Можливо вибрати з вкладки конструктор або з контекстного меню **Добавленные таблицы** (мал. 10), що дозволяє додати таблиці або запити в вікно **Схема даних**.



Малюнок 10

### Визначення зв'язків між таблицями

Необхідно закрити всі відкриті таблиці. Не дозволяється створення або вилучення зв'язків між відкритими таблицями.

Для зв'язування полів треба вибрати поле в одній таблиці і перемістити його за допомогою миші на відповідне поле в іншій таблиці.

В більшості випадків зв'язують ключове поле (наведене в списку полів напівжирним шрифтом) однієї таблиці з відповідним йому полем (яке часто має те ж саме ім'я), що називають полем зовнішнього ключа в іншій таблиці. Зв'язані поля не обов'язково повинні мати однакові імена, однак, вони повинні мати однакові типи даних і мати зміст одного типу. Крім того, поля, які зв'язують, типу **Числової** повинні мати однакові значення властивості **Размер поля**.

Після цього відкривається вікно діалогу **Изменение Связей**. Тут потрібно перевірити правильність імен зв'язуємих полів, які виводяться в двох стовпцях бланку в вікні зв'язків. При необхідності, вибирати в осередках бланку інші імена полів. Створюється зв'язок натиском кнопки **Создать**. Це повторюється для кожної пари зв'язуємих таблиць.

В вікні **Схема данных** користувач має можливість виконати наступні операції:

Змінити макет таблиці.

Змінити існуючий зв'язок.

Усунути зв'язок.

Усунути таблицю з вікна **Схема данных**. Додати таблицю.

Вивести на екран всі існуючі зв'язки або зв'язки тільки для конкретної таблиці.

При спробі закрити вікно **Схема данных** виводиться запрошення зберегти поточний макет схеми даних. Незалежно від збереження поточного макету створені зв'язки будуть збережені в базі даних.

### Зміна існуючого зв'язку

Встановимо вказівник на лінію зв'язку, яку вимагається змінити, і двічі натиснемо кнопку миші. Виберемо параметри зв'язку.

### Вилучення зв'язку

Встановимо вказівник на лінію зв'язку, яку вимагається усунути, і натиснемо кнопку миші (вибрана лінія зв'язку стане жирною), після цього натиснемо клавішу **DEL** (або у контекстному меню - **Удалить**).

### Перегляд існуючих зв'язків

Викликаємо вікно **Схема данных**.

Для перегляду зв'язків, певних для конкретної таблиці, виберемо таблицю і натиснемо кнопку **Прямые связи** на вкладке **Конструктор**.

Для того щоб на екрані були зв'язки тільки конкретної таблиці, слід заздалегідь виконати команду **Очистить макет**. Після цього додайте в вікно потрібну таблицю і натисніть кнопку **Прямые связи**. За необхідністю натиснувши **Все связи** можна отобразить все существующие в БД связи.

Приклад зв'язку трьох таблиць по ключовому полю «№» з першої таблиці.



Малюнок 11

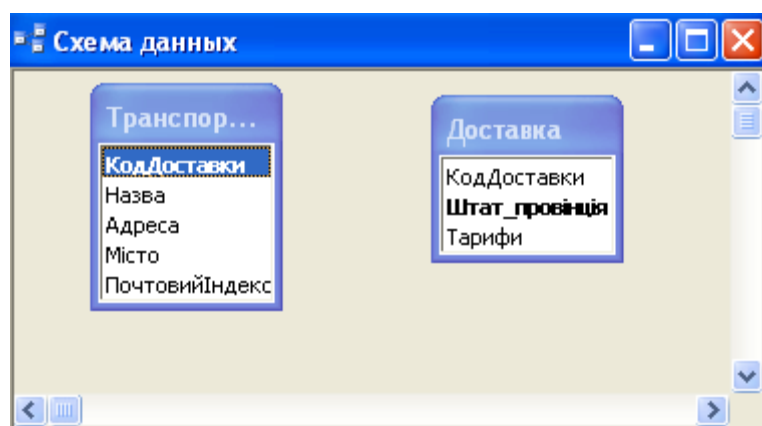
За допомогою вікна **Схема данных** можна задавати як прості відношення, так і складні взаємозв'язки особливо в таких випадках:

- Якщо треба встановити цілісність даних;
- Якщо ключ вміщує більше ніж одне поле;
- Якщо немає прямого зв'язку між двома таблицями.

### Приклад встановлення взаємозв'язків

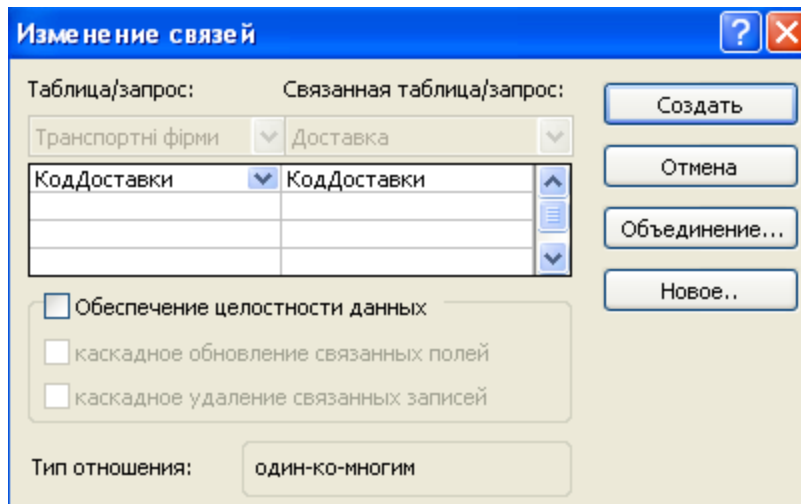
Для приклада будемо працювати з таблицями **Доставка**, яка містить дані про тарифи на перевезку товарів та **Транспортні фірми**, яка містить дані про фірми, які займаються перевезками. Вартість перевезки залежить від транспортної фірми та пункту перевезення. Створимо наші таблиці будь-яким способом та заповнимо їх даними. Оскільки кожна фірма перевозить у різні регіони, то одній фірмі відповідають декілька тарифів, тобто між таблицями зв'язок “один-до-багатьох”.

Відкриємо Схему даних. Додаємо таблиці **Транспортні фірми** та **Доставка** (мал.12).



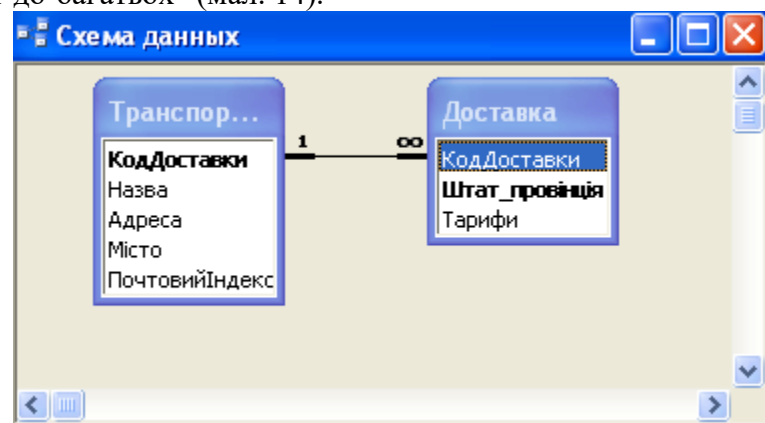
Малюнок 12

Встановимо взаємозв'язок між таблицями **Транспортні фірми** та **Доставка**, щоб зв'язати інформацію про транспортні фірми з даними про тарифи на перевезення. В вікні **Схема данных** перетягуємо поле **Код доставки** з таблиці **Транспортні фірми** в таблицю **Доставка**. (Напрямок перетягування ключа визначає, яка таблиця є головною, а яка зв'язаною. Таблиця, звідки переміщується ключове поле, є головною. Таблиця куди воно переміщується, - зв'язаною). З'являється діалогове вікно **Изменение связей** із значенням “один-до-багатьох” в полі **Тип отношения** (мал. 13).



Малюнок 13

В діалоговому вікні **Изменение связей** встановіть прапорець в поле **Обеспечение целостности данных**. Цілісність не допускає появи незв'язаних записів. Тепер натискаємо **Создать**. Таблиця **Транспортні фірми** тепер зв'язана з таблицею **Доставка**. Зв'язуючі поля обох таблиць з'єднані лінією, поміченою цифрою 1 і символом нескінченності, що вказує про відношення "один-до-багатьох" (мал. 14).



Малюнок 14

### Контроль інформації шляхом забезпечення цілісності даних

Якщо таблиці зв'язані, то зміни у одній з таблиць впливають на дані в іншій. Це забезпечується цілісністю даних, яка представляє собою систему правил, які виключають вільну зміну зв'язаних записів.

Цілісність даних накладає наступні обмеження на введення та корективи інформації.

- Не можна вводити значення у зовнішній ключ зв'язаної таблиці, якщо не існує співпадаючого значення у ключовому полі головної таблиці. Наприклад, ви не зможете додати запис у таблицю **Закази**, вказавши **Код замовника**, якого немає в таблиці **Замовники** (не можна ввести заказ для неіснуючого замовника).
- Забороняється видаляти записи в головній таблиці, якщо існують відповідні записи в зв'язаній таблиці. Не можна, наприклад, видаляти запис в таблиці **Замовники**, якщо їй відповідають записи в таблиці **Закази** (не можна видалити замовника, у якого є закази).
- Не можна змінити значення ключового поля в головній таблиці, якщо в зв'язаній таблиці є записи, які посилаються на це значення. Так, наприклад, вам не вдасться змінити **Код замовника** в таблиці **Замовники**, якщо в таблиці **Закази** існують записи для цього замовника.

Цілісність даних можна забезпечити з виконанням наступних умов.

- Співпадаюче поле в головній таблиці повинно бути ключовим.
- Всі значення зовнішнього ключа зв'язаної таблиці повинні бути присутніми в ключовому полі головної таблиці.
- Зв'язані поля в обох таблицях повинні мати однаковий розмір і тип даних.
- Обидві таблиці повинні належати одній базі даних Access. Якщо зв'язані таблиці знаходяться в різних файлах, вони повинні мати формат Microsoft Access (.mdb), а база даних, у якій зберігаються зв'язані таблиці, повинна бути відкритою.

### Каскадне відновлення та видалення даних

В Access існує можливість обійти обмеження на зміну ключових полів і видалення даних у зв'язаних таблицях не порушуючи цілісність даних.

Каскадне відновлення забезпечує розповсюдження змін в головній таблиці на відповідні записи в зв'язаній таблиці. Так, наприклад, зі зміною коду замовника в таблиці **Замовники** відбудеться відновлення співпадаючого поля в таблиці **Закази**, і зв'язок між таблицями збережеться.

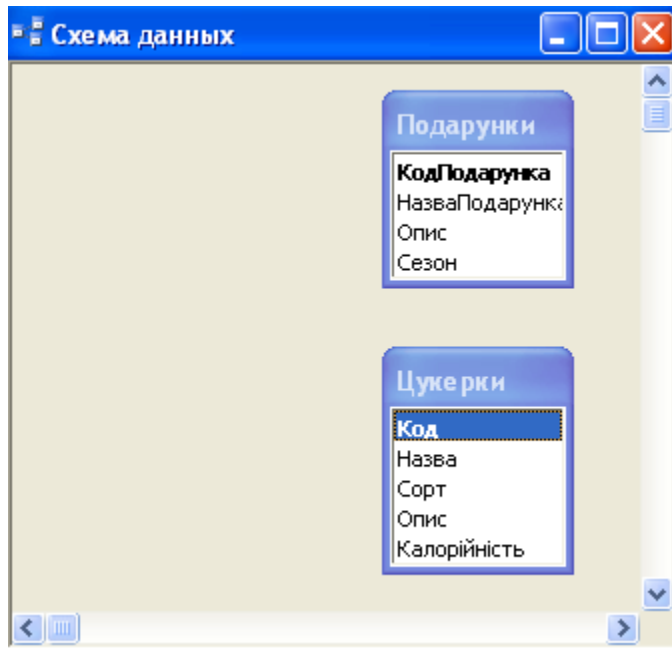
Якщо задати каскадне видалення, з видаленням запису в головній таблиці видаляються відповідні йому записи в зв'язаних таблицях. Наприклад, видалення замовника з таблиці **Замовники** включає видалення всіх його заказів з таблиці **Закази**, а також видалення даних про склад заказів в таблиці **Подробиці заказів**. Тому що в цьому випадку видаляться і не виконані закази, каскадне видалення слід використовувати з обережністю.

### Відношення багато-до-багатьох

Відношення багато-до-багатьох передбачає, що кожному запису в одній таблиці відповідає декілька записів в іншій. При цьому кожна сторона відношень виглядає як відношення один-до-багатьох. Але якщо розглядати взаємозв'язок таблиць з обох сторін, стає очевидним, що ні одна з таблиць не може бути головною і для їх зв'язування необхідна третя таблиця.

Зв'язуюча таблиця представляє собою проміжну таблицю, яка служить мостом між двома таблицями у відношеннях багато-до-багатьох. Її ключ складається з ключових полів цих таблиць, з якою з яких вона зв'язана відношенням багато-до-багатьох. Окрім ключових полів, зв'язуюча таблиця повинна мати хоча б одне поле, якого немає у зв'язуємих таблицях, але яке має значення для кожної з них. Таким чином, відношення багато-до-багатьох складається з відношень багато-до-одного та один-до-багатьох.

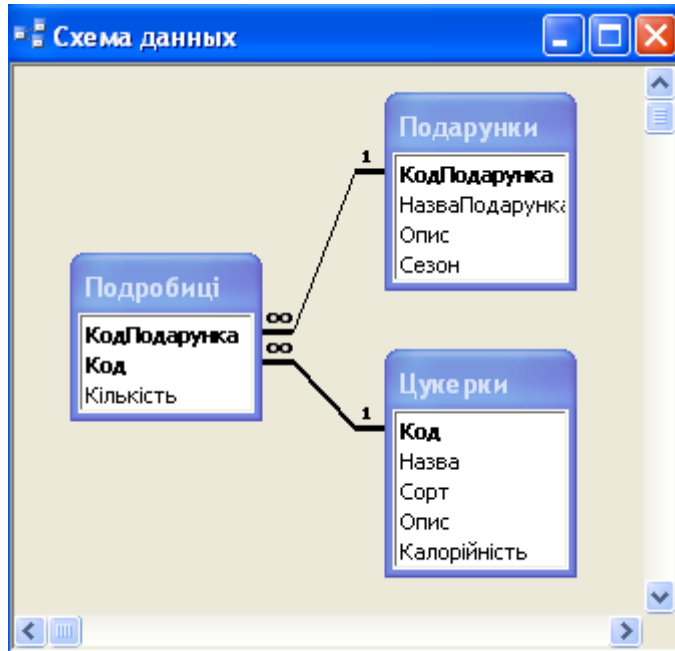
На прикладі розглянемо, як за допомогою таблиці **Подробиці** створити відношення багато-до-багатьох для таблиць **Подарунки** та **Цукерки** (мал. 15).



Малюнок 15

Між таблицями Подарунки та Подарунки існує відношення один-до-багатьох через ключове поле Код подарунка. Крім цього таблиця Подарунки з таблицею Цукерки теж зв'язані відношенням багато-до-одного через складний ключ, який складається з полів Код подарунка та Код цукерки. Натискаємо на кнопку **Прямые связи**. З'являться лінії зв'язку, які вказують типи відношення між цими таблицями. Перетягнемо таблиці так, щоб схема краще продивлялась, щоб таблиці і лінії зв'язку не накладались.

Таблиця Подарунки виступає зв'язуючою між таблицями Подарунки та Цукерки (мал. 16).



Малюнок 16

## Контрольні питання до лабораторної роботи 2:

1. Для чого потрібна нормалізація?
2. Для чого потрібно створювати міжтабличні зв'язки?
3. Перелічіть способи зв'язувати дані між двома таблицями в Access.
4. Які типи відношень можна встановити між двома таблицями?
5. Які можливості розкриває майстер підстановок?

- 6. Які існують два основні способи зв'язувати дані між таблицями в Access?**
- 7. Як визначити зв'язки між таблицями?**
- 8. Як виглядають зв'язані таблиці на Схемі даних?**
- 9. Як змінити зв'язок між таблицями?**
- 10. Які типи відношень існують?**
- 11. Як створити відношення один-до-багатьох?**
- 12. Як створити відношення один-до-одного?**
- 13. Як створити відношення багато-до-багатьох?**
- 14. Якщо існує взаємозв'язок між таблицями, яку таблицю називають головною, а яку підлеглою?**
- 15. Що таке цілісність даних?**
- 16. Які обмеження на введення та корективи інформації накладає цілісність?**
- 17. Що таке каскадне відновлення?**
- 18. Що таке каскадне видалення?**

## Лабораторна робота № 3.

**Тема: Теоретичні мови запитів. Проектування та можливості QBE запитів. Запити на вибірку.**

**Мета: Навчитися проектувати прості запити та застосовувати можливості QBE запитів.**

### Зміст роботи за варіантом індивідуального завдання:

1. Створити запит за допомогою майстра простих запитів. Виберіть «подробний» звіт.
2. Створити запит за допомогою майстра простих запитів. Виберіть «ітоговий» звіт.
3. Внесіть корективи у свої запити за допомогою. Конструктора запитів.
4. Побудувати QBE-запит в режимі конструктора, який виводить записи кількох полів вашої таблиці:
  - а) вилучіть останнє поле з бланку запиту, додайте інше поле до бланку запиту;
  - б) зробіть, щоб не всі поля, які входять до бланку запиту, відображались у динамічній таблиці.
5. Відсортуйте:
  - а) текстове поле з таблиці за зростанням та спаданням за допомогою QBE-запиту;
  - б) числове поле з таблиці за зростанням та спаданням за допомогою QBE-запиту.
6. За допомогою запитів, створених для відповідних таблиць БД відобразіть записи:
  - а) (наприклад, прізвища працівників чи клієнтів, чи виконавців) які починаються з певної літери;
  - б) наприклад, людей, які не проживають, не працюють або не були у певному місті;
  - в) події, які здійснювались в інтервалі часу з певного періоду до іншої дати;
  - г) (наприклад, ціна товару або інше числове поле) що більше 10, але не 100.  
(Задайте умову відбору для поля вашої таблиці, використайте Построитель виражений для своїх виразів у запитах вашої бази даних, використайте такі операції: >, <, =, <>, And, Or, Not, Like).
7. За допомогою QBE-запиту опрацюйте зміст числового поля нашої таблиці, використовуючи статистичні функції. Наприклад, вивести суму (або середнє значення, або мінімальне значення тощо), яку отримав кожен працівник за весь час роботи з таблиці Журнал видачі.

№ вид.	ПІБ	Отримав	Дата
1	Іванов В.А.	100	1.01.2010
2	Іванов В.А.	200	3.03.2010
3	Петров А.Б.	500	5.03.2010

Таб.ном.	ПІБ	Отримав
		SUM

### Теоретичні відомості:

#### Створення та використання запитів

Система Access має значно ширші можливості з пошуку різних відомостей з бази даних окрім пошуку у режимі таблиці.

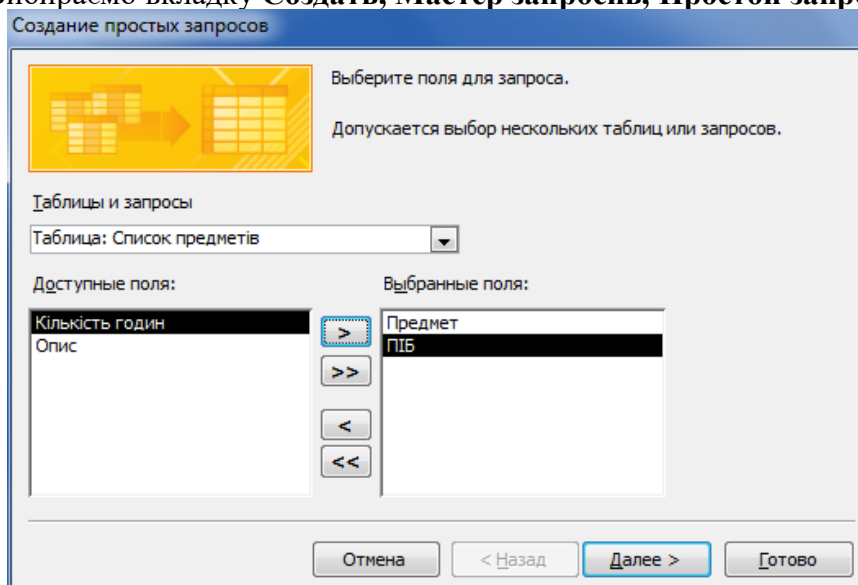
Запит – засіб відбору даних в БД з одної або декількох таблиць за критеріями визначеними користувачем. Бувають запити на вибірку та запити дії. Запити бувають QBE ( Query By Example-запит за зразком), параметри яких встановлюються у вікні конструктора запитів, і SQL-запити (Structured Query Language–структурована мова запитів), при створенні яких використовуються оператори і функції мови SQL. Найпоширеніші це запити на вибірку, розглянемо QBE.

**Простий запрос.** Цей спосіб реалізується майстром **Простий запрос.**

Майстер **Простий запрос** дозволяє із таблиць БД отримати нову тимчасову динамічну таблицю, в яку включені окремі поля різних таблиць. Наприклад, із зв'язаних таблиць

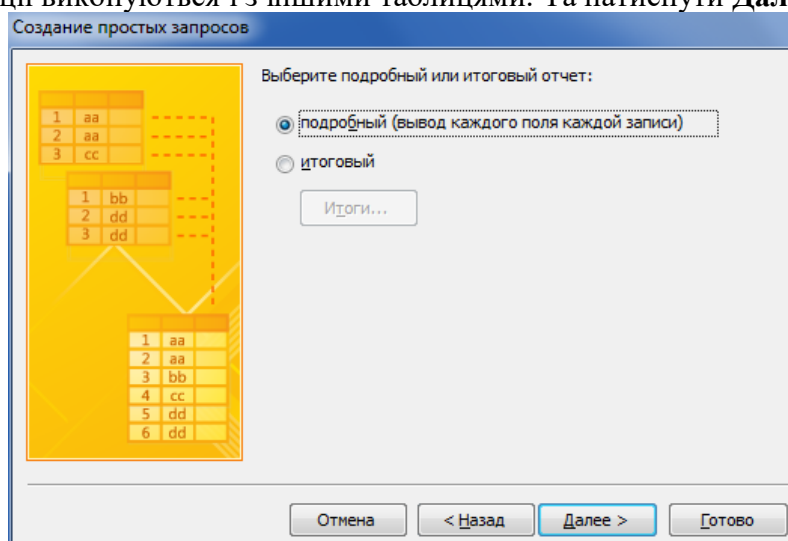
**Список предметів і Консультації** можна отримати запит у вигляді таблиці з іменем **Сведения**, в якій будуть знаходитися поля **ПІБ**, **Предмет** таблиці **Список предметів** і **День** і **Аудиторія** таблиці **Консультації**. Після цього із запиту **Сведения** можна отримати конкретні дані.

1. Вибираємо вкладку **Создать, Мастер запросив, Простой запрос.** мал. 1.



Малюнок 1

2. Необхідно відкрити список таблиць у **Таблицы и Запросы** і виділити ту з них, у якій необхідно вибирати відповідні поля. Потім ці поля потрібно перенести в поле вибраних. Аналогічні операції виконуються і з іншими таблицями. Та натиснути **Далее**.

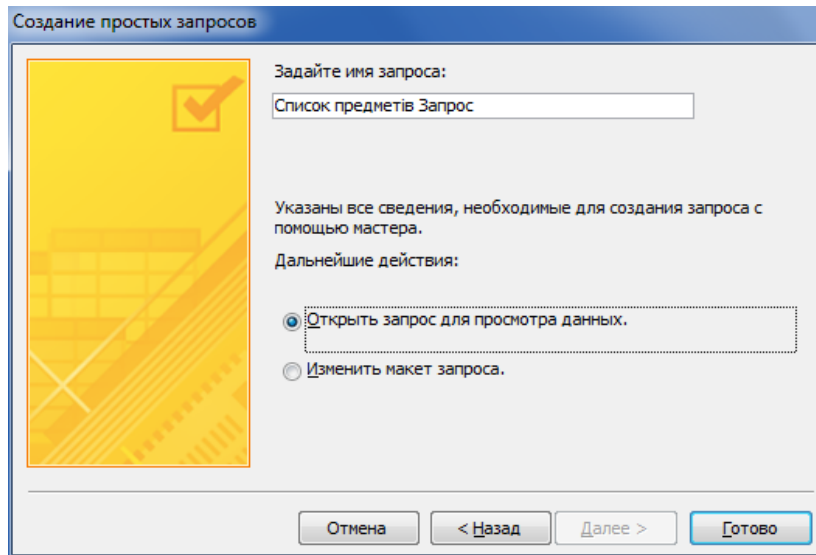


Малюнок 2

3. Наступне вікно майстра – вибір з двох перемикачів (мал. 2) **Подробный** чи **Итоговый** включаємо **Подробный** і натискаємо **Далее**. З'явиться вікно, показане на мал. 3.

4. Введемо ім'я запиту **Сведения** і натиснемо кнопку **Готово**. У результаті на екрані з'явиться запит з іменем **Сведения**, показаний на мал. 4.

Зверніть увагу, що в цю таблицю введені саме ті поля двох перших таблиць, які ми зазначили вище.

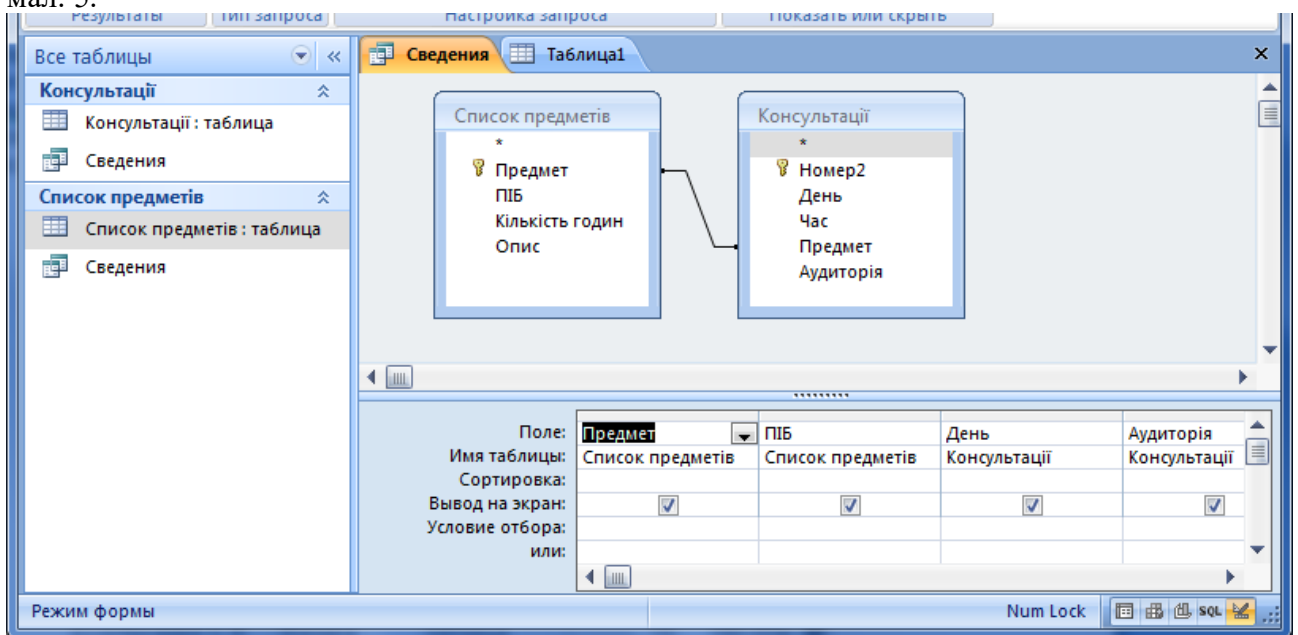


Малюнок 3

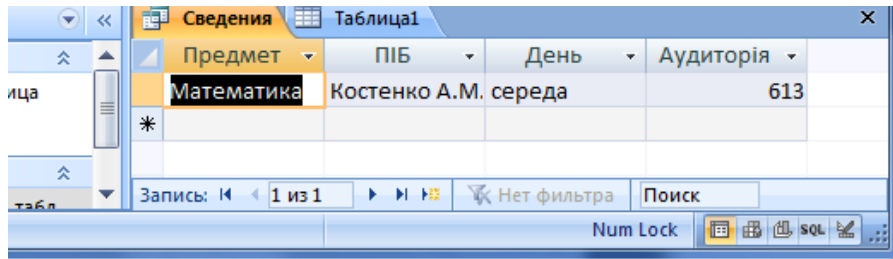
Предмет	ПІБ	День	Аудиторія
Біологія	Кірін Т.В.	понеділок	516
Інформатика	Абрамов К.К.	вівторок	408
Математика	Костенко А.М.	середа	613
Фізика	Носенко С.К.	четверг	
*			

Малюнок 4

Тепер з отриманої таблиці зробимо конкретний вибір, а саме - запит про вибір математики. Для цього перейдемо у режим конструктора. Чи натиснемо кнопку Конструктор на вкладці главная, чи за допомогою контекстного меню. На екрані з'явиться вікно запити, зображене на мал. 5.



У рядку **Условие отбора** у полі **Предмет** введемо - **Математика**. Після цього на панелі інструментів натиснемо кнопку зі знаком оклику. На екрані з'явиться таблиця, показана на мал. 6.



Малюнок 6

Нагадаємо, що розглянута методика виконання простого запиту здійснювалась з урахуванням того, що був ввімкнений перемикач **Подробний** (мал.2). Розглянемо тепер порядок створення запиту при увімкненому перемикачі **Итоговый ...**.

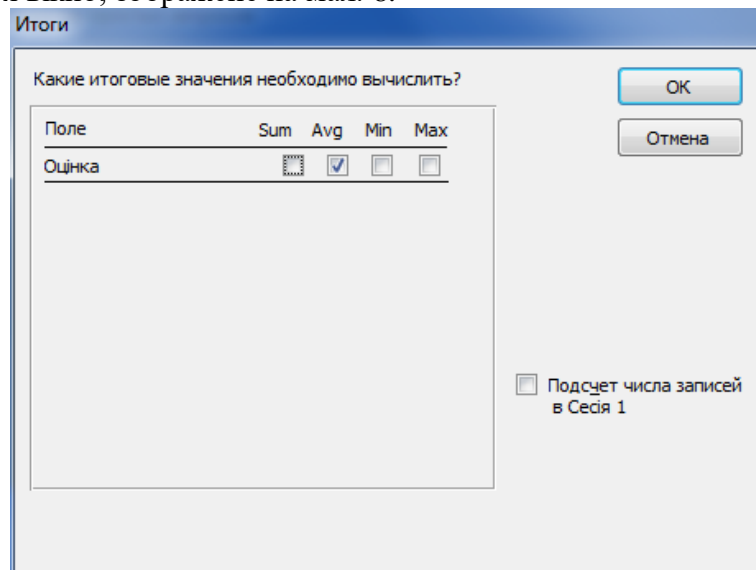
Створення підсумкового запиту буде розглядатися на прикладі таблиці з іменем **Сесія**:



Малюнок 7

Створимо запит **Сесія 1**, у який включимо поля, **ПІБ**, **Оцінка**.

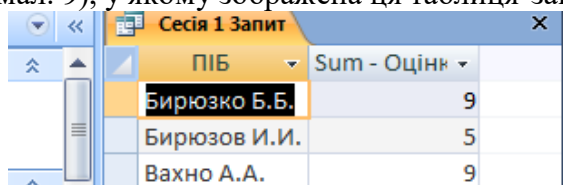
Процес створення запиту починаємо так само, як це робили раніше. Коли з'явиться вікно, зображене на мал. 2, вмикаємо перемикач **Итоговый**, потім натискаємо кнопку **Итоги ...**. У результаті з'явиться вікно, зображене на мал. 8.



Малюнок 8

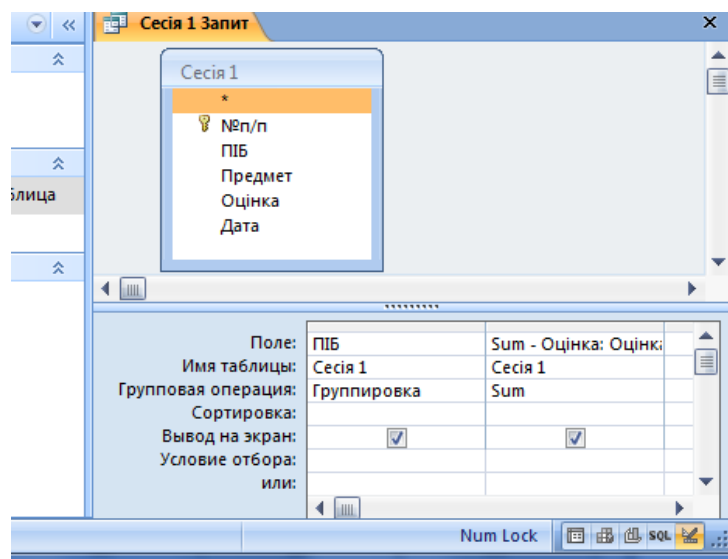
Зверніть увагу, що в цьому вікні показано тільки числове поле. Необхідно ввімкнути прапорці для тих полів, для яких необхідно отримати підсумкові значення. Вмикаємо, наприклад, для поля Оцінка пропарець у колонці Sum. Після цього натискаємо кнопку **ОК**. З'явиться знову вікно, зображене на мал. 2, у якому необхідно натиснути кнопку **Далее**.

У вікні, яке відкриється, введемо ім'я запиту **Сесія 1 Запит** і натиснемо кнопку **Готово**. З'явиться вікно (мал. 9), у якому зображена ця таблиця-запит.



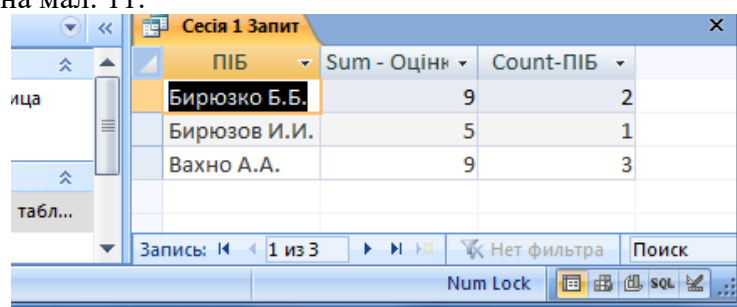
Малюнок 9

Перейдемо у режим **Конструктор**. З'явиться вікно, зображене на мал. 10. Це вікно дозволяє сформуванати нові запити. Там є ще рядок **Групповая операция**. Під груповою операцією розуміємо обробку значень даного поля за допомогою певної операції. Цей рядок використовують для виклику списка доступних операцій для кожного поля і для вибору з нього необхідного.



Малюнок 10

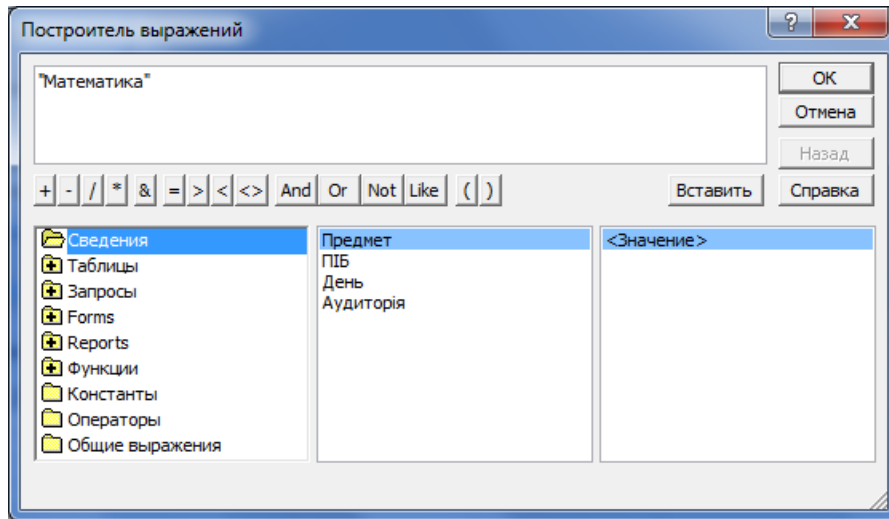
Припустимо, що необхідно підрахувати - скільки разів зустрічається кожне прізвище у таблиці **Сесія 1**. Для цього потрібно створити у вікні, зображеному на мал. 10, ще одне поле з іменем ПІБ та вибрати для нього групову операцію Count (підрахунок). Після цього на панелі інструментів потрібно натиснути кнопку зі знаком оклику і на екрані з'явиться бланк-запит, зображений на мал. 11.



Малюнок 11

У системі Access є також можливість пошуку інформації, використовуючи логічні вирази. Створення цих виразів здійснюється за допомогою програми, яку називають **Построитель выражений**. Ця програма створює нову таблицю, зміст якої відповідає заданому виразу.

Роботу програми **Построитель выражений** будемо розглядати на прикладі таблиці-запиту **Сведения**. Для запуску програми необхідно викликати контекстне меню рядка **Условия выбора** У цьому меню потрібно виконати команду **Построить...** . У результаті з'явиться вікно **Построитель выражений**, зображене на мал. 12.



Малюнок 12

Це вікно складається з чотирьох частин. Верхню частину використовують для введення логічного виразу.

Троє нижніх вікон використовують для вибору елементів, які у них знаходяться. Ліворуч розташовані об'єкти бази даних: таблиці, запити та ін., а також функції, константи і оператори, які можуть бути використані у виразі.

У середній частині перераховані елементи вибраного об'єкта ліворуч.

Праву нижню частину вікна використовують для вибору конкретних значень тих об'єктів, які вказані в середній частині.

У разі конструювання логічних виразів потрібно дотримуватися певних правил. Основні з них: ім'я об'єкта бази даних (наприклад ім'я таблиці) відокремлюється від імені поля знаком оклику; імена беруться у квадратні дужки; дані символічного типу беруться у лапки. При цьому, якщо ви не самі набираєте вказані об'єкти та значення, а вибираєте їх з тих, які показані у вікні, то Access сама дотримується правил синтаксису.

Приступимо тепер до конструювання виразів. Введемо такий вираз:

**([Список предметів]![Предмет]="Математика") And ([Консультації]![День]="середа")**  
Коли набрали весь вираз, **ОК**. З'явиться вікно запити. Натискаємо **Выполнить**. У результаті з'явиться потрібна інформація.

Приклад іншого виразу для іншого запити:

**([Сесія 1]![Дата]<#20.07.2016#) And ([Сесія 1]![Оцінка]=5)**

Звертаємо вашу увагу на те, що після математичних знаків (у даному випадку після знаків більше чи менше) не повинно бути пропусків. Інакше числові константи будуть сприйматися як текстові.

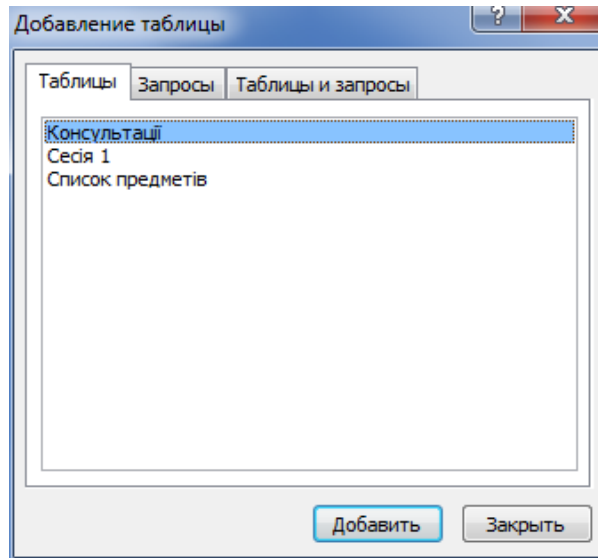
Окрім розглянутих способів пошуку інформації в базі даних може використовуватися **Перекресный запрос** для побудови таблиць з підрахованими підсумковими значеннями і **Повторяющиеся записи** для групування записів з однаковими значеннями полів та **Записи без подчиненных**.

Результатом виконання запити на вибірку являється набір записів, зібраних у таблиці, що називається Recordset (динамічний, тимчасовий набір даних). У об'єкті Recordset допускається додавання, зміна і видалення записів. У ньому відображаються також записи, які можна додавати, видаляти або змінювати у вихідних таблицях.

### Створення QBE-запитів у режимі конструктора

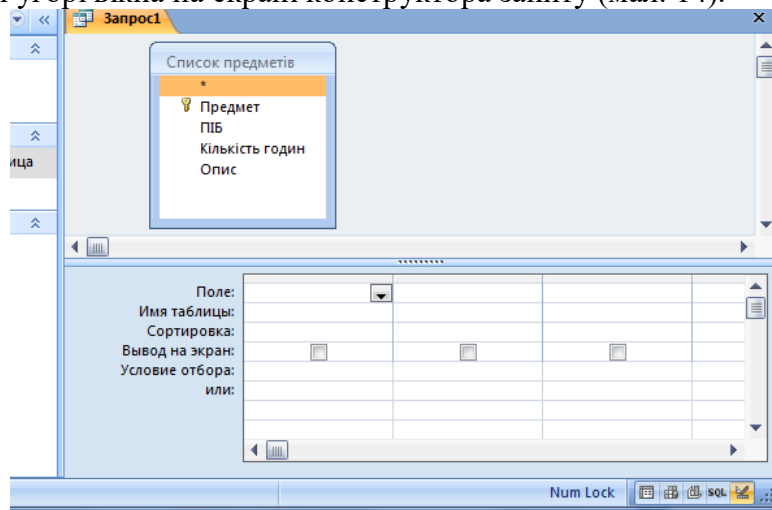
Для створення запити найкраще почати роботу в режимі конструктора запити.

1. На вкладці **Создать** вибираємо **Конструктор запросов**.
2. У діалогове вікно **Добавление таблицы** вибираємо потрібні (мал. 13).



Малюнок 13

Виберіть таблицю з вашої бази даних, а потім клацніть на кнопці **Добавить**. Назва вашої таблиці з'явиться угорі вікна на екрані конструктора запитів (мал. 14).



Малюнок 14

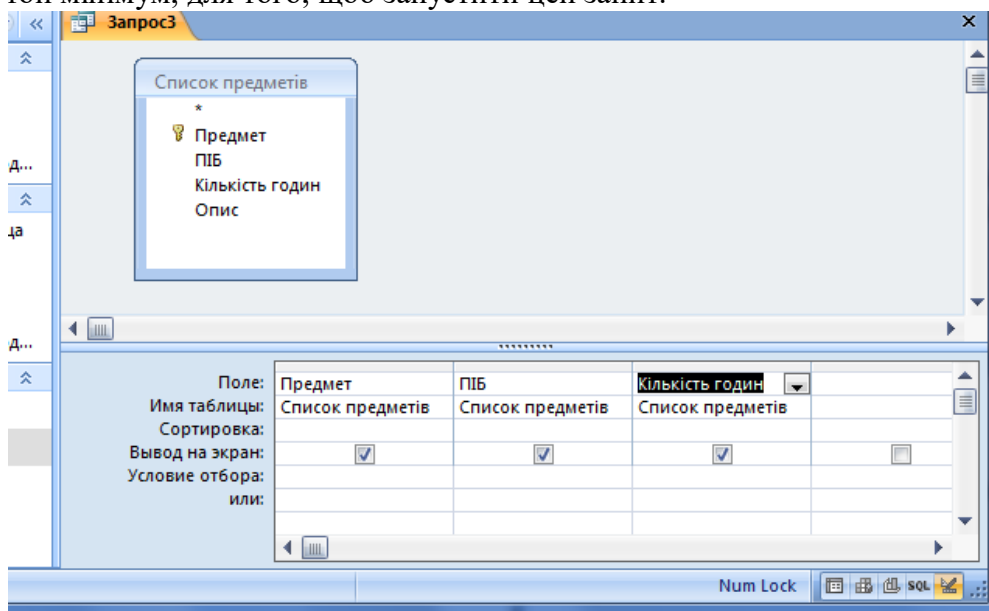
Ви будете використовувати поля, які видно в списку полів, представлених у бланку запитів, який знаходиться нижче області конструктора запитів. Стовпці вміщують поля вибраної таблиці, якою в даному випадку є Список предметів. Ось список тих задач, які вирішують рядки цього бланка:

- **Поле.** Цей рядок містить поле (з списку полів), які ви бажаєте включити в запит. Він може також уміщувати й вирази (тобто створене вами тимчасове поле).
- **Имя таблицы.** Показує ім'я таблиці, з якої узяті поля. Це корисно в таких випадках, коли запит заснований на декількох таблицях, особливо якщо назви полів з декількох таблиць співпадають.
- **Сортировка.** Дозволяє вказувати на підставі яких полів буде проводитися сортування записів.
- **Вывод на экран.** Визначає, чи буде дане поле виводитися на екран. Використовуйте цю опцію тоді, коли ви використовуєте деяке поле для умови відбору, але не бажаєте, щоб воно з'явилося на екрані.
- **Условие отбора.** Умови відбору дозволяють вам включати або не включати записи в результативну таблицю, ґрунтуючись на їх значеннях або комбінаціях значень.

У залежності від типу створеного запити, рядків може бути і більше.

Щоб заповнити поля запити, потрібно двічі натиснути на полі в списку полів таблиці. Access перенесе це поле в перший стовпчик бланка запити (мал.15). У іншому рядку

з'явиться назва таблиці, з якої це поле отримали. Включивши хоча б одне поле в запит, ви вже маєте той мінімум, для того, щоб запустити цей запит.



Малюнок 15

### Видалення полів і таблиць із запиту

Для видалення поля з бланка запиту варто виділити колонку, у якій воно знаходиться, і натиснути клавішу [ Del ] або активізувати команду **Удалить столбцы**, або **Вырезать**.

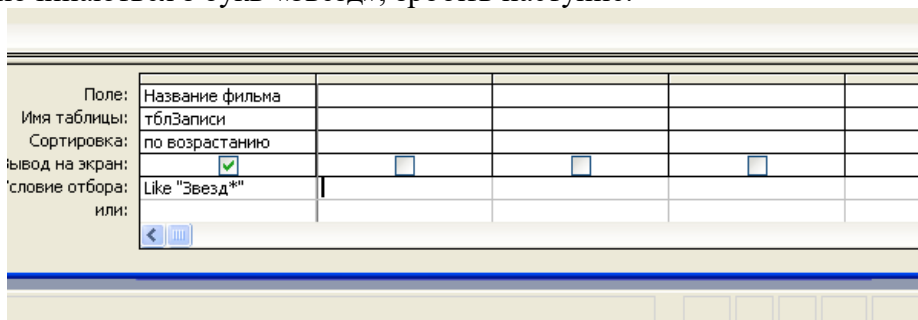
### Сортування даних у запиті

Для того щоб встановити порядок сортування в полі, виконайте наступне:

1. У бланку запиту помістіть курсор у рядок **Сортировка** того поля, за яким потрібно провести сортування. Коли ви це зробите, то побачите, що поле сортування пропонує випадючий список.
2. Клацніть на кнопці списку, що випадає, і ви побачите, що він пропонує вам три можливості: за зростанням, за спаданням і відсутнє.

### Встановлення критеріїв відбору записів

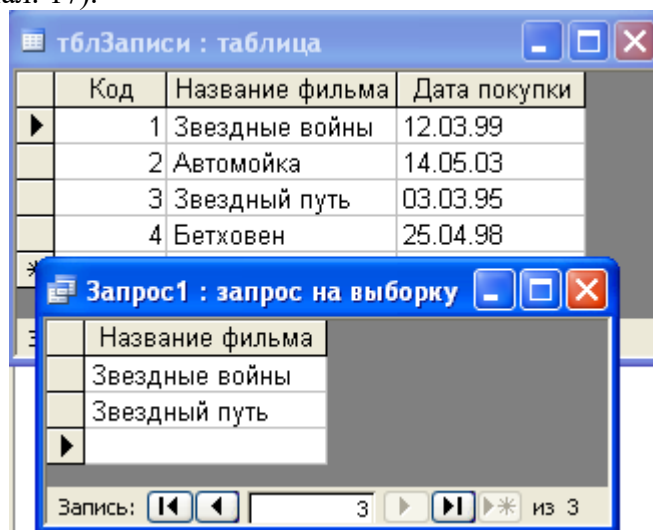
За допомогою критеріїв відбору користувач вказує Access, які записи потрібно вибирати в таблицях і відобразити в таблиці результатів виконання запиту. Критерії відбору вказуються для одного або декількох полів. Наприклад, для того щоб побачити список фільмів, що починаються з букв «Звезд», зробіть наступне:



Малюнок 16

В режимі конструктора запиту у першому рядку умов відбору для поля **НазваниеФильма** надрукуйте Like Звезд\* і натисніть <Enter>. Access перетворить це в Like “Звезд\*”, поставивши Звезд\* у лапки. Ця умова відбору показана на мал.16.

Службове слово Like і підстановочний знак зірочка (\*) говорять програмі Access, що ви хотіли б побачити всі назви фільмів, що починаються з “Звезд”. Клацніть на кнопці **Вид**, щоб побачити результат (мал. 17).



Малюнок 17

IS NULL – для виявлення незавершених чи пустих записів.

IS NOT NULL – для відображення лише тих записів, де є дані у необхідному полі.

Часто в предикатах потребується не тільки оцінювати рівність оператора як істинного чи хибного, але й здійснювати інші види зв'язків. Це реалізується за допомогою булевих операторів і знаків відношення, причому предикат може утримувати необмежену кількість умов.

В цілому, реляційний оператор – це математичний символ, котрий вказує на визначений тип порівняння між двома значеннями, при цьому SQL має в своєму розпорядженні наступний їх набір:

- = рівний чому-небудь;
- > більше чим;
- < менше чим;
- >= більше чим або дорівнює;
- <= менше чим або дорівнює;
- <> не дорівнює.

Ці оператори мають стандартні значення для числових даних, а для символічних їх визначення залежить від кодів ASCII символів – вони слідуєть в алфавітному порядку, причому заголовні літери мають менший код, ніж рядкові, тому, наприклад, “Z”<”a”.

Стандартними булевими операторами, які використовуються в SQL, є AND, OR і NOT.

Приклад: >= #01.01.97 # And < # 01.01.98 # або

Between # 01.01.97 # And #01.01.98 #

### Скасування відображення полів у запиті

Поле відображається в таблиці результатів запиту, якщо у відповідній колонці бланка запиту, у рядку **Вивод на екран** включений контрольний індикатор. Для скасування відображення поля достатньо виконати натиск на індикаторі .

### Обчислення в запиті

Запит можна використовувати для виконання розрахунків і підведення результатів, узагальнивши дані з вихідних таблиць. Для цих цілей у Access передбачені статистичні

функції SQL. Статистичну функцію задають у рядку **Групповая операция**, що з'являється після натискання кнопки з грецькою літерою сигма, розміщеної на панелі інструментів. За допомогою статистичної функції можна опрацювати зміст кожного поля запиту. Результат опрацювання з'являється в результуючому наборі записів запиту.

<b>Функція</b>	<b>Дія</b>
Sum	Підсумовування значень визначеного поля в записах, що відібрані запитом.
Avg	Обчислення середнього значення в записах визначеного поля, що відібрані запитом.
Min	Обчислення мінімального значення в записах визначеного поля, відібраних запитом.
Max	Обчислення максимального значення в записах визначеного поля, відібраних запитом.
Count	Обчислення кількості записів, відібраних запитом, у визначеному полі.
First	Визначається перше значення в зазначеному полі записів, відібраних запитом.
Last	Визначається останнє значення в зазначеному полі записів, відібраних запитом.
StDev	Обчислюється стандартне відхилення значень даного поля для усіх записів, відібраних запитом.
Var	Обчислюється варіація значень даного поля для всіх записів, відібраних запитом.

### **Контрольні питання до лабораторної роботи 3:**

- 1. Що являють собою запити і для чого їх використовують?**
- 2. Якими способами можна створити запит?**
- 3. Чим відрізняється «ітоговий» запит від «подробного» запиту?**
- 4. В якому режимі можна вносити корективи у готовий запит?**
- 5. За допомогою якого засобу можна легко створювати вирази у QBE?**
- 6. Що утворюється в результаті виконання запиту на вибірку?**
- 7. Які запити називають QBE-запитами?**
- 8. Які можливості QBE-запитів?**
- 9. Як видаляти та додавати поля і таблиці у бланку QBE?**
- 10. Як відсортувати дані за допомогою QBE?**
- 11. Яким чином можна встановити критерії відбору записів в своїй БД?**
- 12. Перелічіть статистичні функції SQL, які ми можемо застосовувати у QBE-запитах.**
- 13. Для чого служить рядок Условие отбора у вікні конструктора запиту?**
- 14. В яких випадках використовують оператор BETWEEN?**
- 15. Який оператор застосовується для відбору записів, які починаються з певного символу?**

## Лабораторна робота №4

**Тема:** Мова реляційного числення за зразком QBE. Обчислювальні поля у запитах. Маніпулювання даними. Параметричні запити, перехресні запити.

**Мета:** Навчитися створювати запити з обчислювальними полями. Вивчити правила побудови виразів. Навчитися будувати запити на створення таблиці, на додавання інформації з іншої таблиці, на вилучення записів, на поновлення інформації в БД, перехресні запити та запити з параметрами за допомогою QBE.

### Зміст роботи за варіантом індивідуального завдання:

1. Створіть QBE-запит до таблиці вашої БД з обчислювальним полем, яке містить інформацію з інших двох полів, розділену розділовими знаками та допоміжним текстом. Наприклад: Альфа-Львів, тел.: 77-89-54.
2. Створіть запит, за допомогою якого можна було б вираховувати дані таким чином, значення першого поля додаються до другого і множаться на зня третього. 25% від отриманого розмістити в одну з колонок динамічної таблиці.
3. Створіть запит з обчислювальним полем Повідомлення. Наприклад, повідомлення: «Закінчується термін дії», якщо значення у полі Термін дії від 10 до 0 і повідомлення: «Термін дії закінчено», якщо термін дії – 0.
4. Створіть вираз у запиті, який замінює порожні записи у таблиці на 0 або на «Пусто».
5. За допомогою QBE-запиту додайте записи з одної таблиці БД до записів іншої таблиці БД.
6. Видаліть з таблиці БД записи, які відповідають деяким критеріям за допомогою QBE.
7. Замінити в таблиці БД значення, відібрані на основі визначених критеріїв на нові, які обчислюються виразом.
8. За допомогою QBE створіть таблицю з копією записів іншої таблиці БД.
9. Створіть запит, який запитує значення критерію, за яким буде здійснюватися вибір інформації.
10. Створіть перехресний запит.

### Теоретичні відомості:

Обчислювальні поля дозволяють створювати і показувати вирази на основі наявних полів. Вираз представляє комбінацію символів-ідентифікаторів, операторів та значень, яка дає певний результат.

Access може утворювати обчислювальні поля з інших полів, буквальних значень і функцій.

#### Синтаксис обчислювальних полів

Для обчислювальних полів вимагається використовувати певний синтаксис. Наприклад, для комбінації полів Прізвище, Ім'я і кома, синтаксис буде таким: **Виразення1**: [Прізвище] &" ," & [Ім'я].

Синтаксис визначає структуру команди, яку ви використовуєте. В обчислювальних полях визначаються стандартні оператори і їх коректне застосування до відповідних полів.

Перша частина наведеного обчислювального поля **Виразення1**, є ім'ям нового поля. Access запропонує вам його, якщо цього не зробите ви самі. Саме на це ім'я вам потрібно буде посилатися в подальшому при використанні цього поля в формі, звіті або іншому виразі.

Використовуючи в запитах поля з таблиць, ви повинні укласти ці поля в квадратні дужки. Повне ім'я поля складається з назви таблиці і назви поля, розділених знаком «!»

### Використання операторів в обчислювальних полях

Є цілий ряд операторів, що можна використати в обчислювальних полях. Деякі з них призначені для математичних обчислень, як, наприклад, наведені в таблиці.

**Таблиця: Математичні оператори для обчислювальних полів**

Оператор	Опис
+	Додавання
-	Віднімання
*	Множення
/	Ділення
^	n-а ступінь

Прикладом застосування математичного оператора може служити створення нового поля для загальної вартості шляхом множення полів Кількість і Ціна. Це може виглядати так:  
ЗагальнаВартість: [Кількість]\*[Ціна].

Оператор амперсанд (&) використовується для того, щоб з'єднати два рядки з буквальною константою в один новий рядок:

Вираження1: [Прізвище] & ", " & [Ім'я].

Інше популярне обчислювальне поле з'єднує поля Місто, Область і ZIP - код з комою в одне поле. Синтаксис для нього може бути таким:

Адреса: [Місто] & ", "& [Область] & " " & [ZIP-код]

а результат, наприклад, таким: Олександрія, КР 0246207.

В цьому прикладі з адресою ми додали пробіли і коми між назвами міста, областю і ZIP - кодом, помістивши їх у лапки і, використовуючи їх як буквальні константи.

Наступний список містить ще декілька обчислювальних полів, що часто можуть бути корисними:

Вартість: [Ціна]\*[Кількість]

ЗагальнаВартість: [Вартість]+([Вартість]\*[ПодатковийКоефіцієнт])

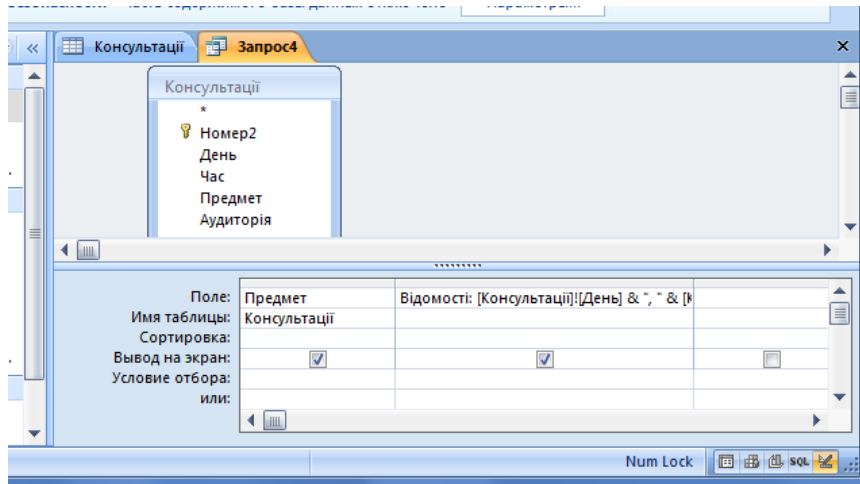
30днів: [ДатаЗамовлення]+30

КодМіста:Left([Телефон];3)

Не всі поля, що використовуються в обчислювальному полі, повинні бути використані в бланку запиту. Якщо поле уявлене в списку полів області конструктора запиту, ви вже можете використовувати його в бланку запиту. Не має значення і те, в якому порядку знаходяться поля або обчислювальні поля.

### Створення простого обчислювального поля

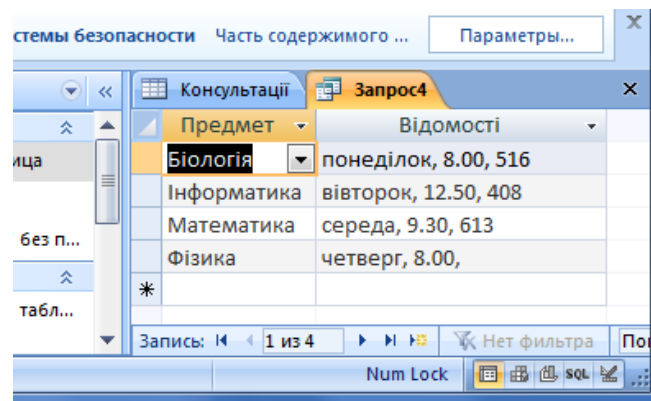
1. В бланку запиту додаються необхідні поля для запиту з необхідної таблиці.
2. У вільному місці рядка Поле викликаємо Построитель виражений або Область вводу (Shift F2) і будуємо вираз. Найпростіший варіант є злиття двох текстових полів (мал.1).



Малюнок 1

3. Для злиття полів День, Час, Аудиторія будемо вираз такий:  
Відомості: [Консультации]![День]&"&"&[Консультации]![Час]&"&"&[Консультации]![Аудиторія] та тиснемо ОК.

4. Запускаємо запит. В результаті отримуємо динамічну таблицю, де з'явилося нове поле (мал. 2).



Малюнок 2

Як і у випадку з полями таблиці, ви можете встановити властивість **Подпись** для будь-якого поля в запиті, щоб на екран виводився текст, відмінний від назви поля. Це дуже зручно для обчислювальних полів, коли ви можете виводити на екран слова з прогалинами між ними і не мати прогалин у назві поля.

### Використання вбудованих функцій Access в обчислювальних полях

Функції Access значно розширюють ваші можливості по відображенню даних і дають наступне:

- Виводити на екран не все, а тільки частину даних.
- Комбінувати дані з новою інформацією, отриманою за допомогою функції Access.
- Комбінувати функції Access і виводити цілком нову інформацію.

Серед прикладів обчислювальних полів, що приводилися вище, було поле, яке називається **КодГорода**. Для нього було використано вираз Left ([Телефон];3). Воно потрапляє в першу категорію типів інформації, наданих функціями Access.

Функція Left() є вбудованою функцією Access, подібно функції Date(). Функція Left() разом із двома своїми аргументами, [Телефон] і 3, говорить Access, що потрібно взяти три крайніх зліва символи поля Телефон і повернути їх у якості результату.

Іноді функції потребують аргументів. У даному прикладі поточними аргументами є поле [Телефон] і 3.

Ще одним прикладом використання функції Access є одержання ініціалів з імені і прізвища. Це теж може бути виконане за допомогою функції Left(). Ось відповідне обчислювальне поле: Инициалы: Left([Имя];1) & Left ([Фамилия];1)

### Обробка порожніх значень у запитах

Одна з проблем, що виникає при використанні обчислювального поля, породжується тим фактом, що поля можуть містити порожні значення. Порожнє значення буде поширюватися на всі наступні результати запиту.

Порожнє значення (значення Null) є присутнім у полі тоді, коли в поле не вводилося ніяких значень і не було ніякого значення за замовчуванням. Коли поле містить порожнє значення, це означає, що його значення невідоме, і Access не знає, що в цьому випадку робити (мал. 3).

Щоб бути готовим до порожніх значень, у Access є функції, такі як IsNull() або Nz(). Синтаксис функції такий: IsNull (вираз).

Количество	Цена	ОбщаяСтоимость
23	22,00р.	506,00р.
	34,00р.	
3	23,00р.	69,00р.
	12,00р.	
445	3,00р.	1 335,00р.
0	0,00р.	

Малюнок 3

Вираз може бути полем або обчислювальним значенням. Функція IsNull() поверне або Істина, або Брехня, в залежності від того, є значення виразу порожнім або ні. Передбачається, що це повинно використовуватися разом із функцією IIF(). Синтаксис функції IIF() такий: **IIF (критерій;значення1;значення2)**

При виклику функції IIF(), якщо критерій дає Істина, то функція IIF() повертає значення1, якщо критерій дає Брехня, то функція IIF() повертає значення2.

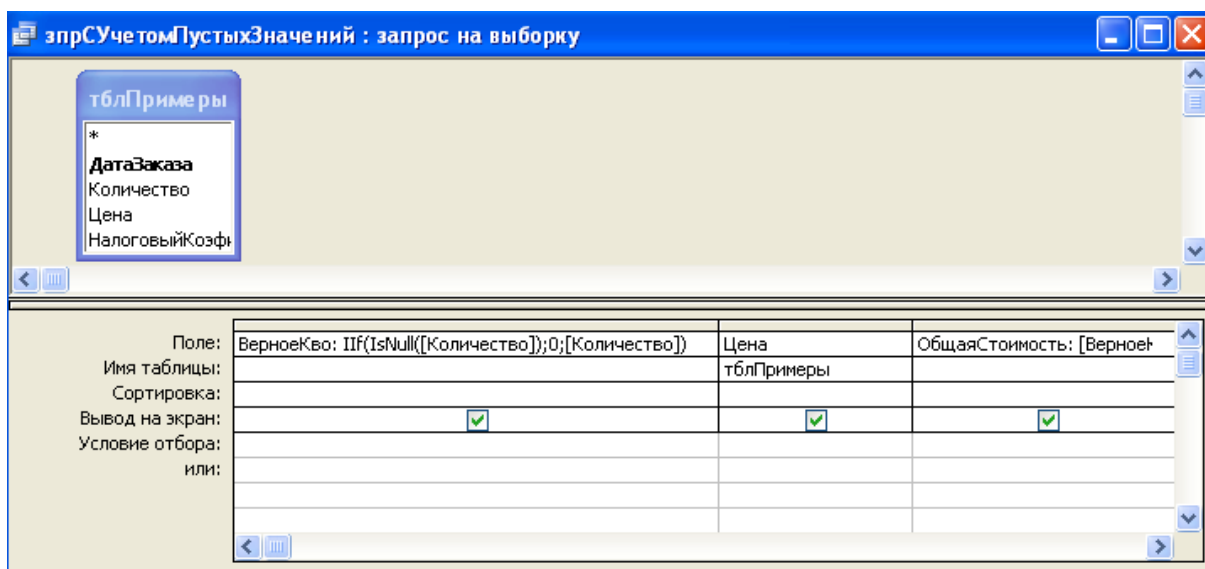
Поле:	Количество	Цена	ОбщаяСтоимость: [Количество]*[Цена]
Имя таблицы:	tblПримеры	tblПримеры	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			
или:			

Малюнок 4

Щоб виправити запит, показаний на мал.3, ми створимо обчислювальне поле за назвою **ВерноеКво**. Буде використана функція IIF() із значенням IsNull(Количество) у якості критерію (значення, що приймає , або Істина, або Брехня) Ось потрібний вираз:

ВерноеКво: IIF (IsNull ([Количество]);0;[Количество])

Якщо функція IsNull() (*критерій*) повертає Істина, то замість Количество використовується 0 (*значення1*). Якщо IsNull() повертає Брехня, то використовується поле Количество (*значення2*). Ви можете побачити це обчислювальне поле на мал.5, де запит поданий у режимі конструктора запиту; результуючий лист з даними показаний на мал.6.



Малюнок 5

Порожні значення тепер замінені нулями.

ВерноеКво	Цена	ОбщаяСтоимость
23	22,00р.	506,00р.
0	34,00р.	0,00р.
3	23,00р.	69,00р.
0	12,00р.	0,00р.
445	3,00р.	1 335,00р.
	0,00р.	

Малюнок 6

Зверніть увагу, що обчислювальне поле **ОбщаяСтоимость** тепер використовує обчислювальне поле ВерноеКво замість поля Количество.

### Запити інших типів

Ми вже розглянули запит на вибірку. У Access існують і запити інших типів: запити дії (на зміну), параметричні запити і перехресні запити.

### Запити на зміну

Запити на зміну використовують для зміни і переносу даних у таблицях, для поновлення, додавання й видалення груп записів, а також для створення нової таблиці на основі однієї або декількох таблиць.

Розрізняють чотири типи запитів на зміну: запит на додавання, запит на видалення, запит на поновлення і запит на створення таблиці.

### Запит на додавання

За допомогою запиту на додавання записи з однієї таблиці (усі або відібрані запитом) можна помістити в кінець іншої таблиці. Обидві таблиці можуть знаходитися як в одній, так і в різних базах даних.

Для відбору записів, що додають потрібно скласти запит на вибірку. Потім слід виконати запит і оцінити результат, переключившись у режим таблиці за допомогою команди із меню **Вид**. Після цього необхідно повернутися в режим конструктора й активізувати команду **Добавление** з вкладки Конструктор.

У діалоговому вікні, яке при цьому відкриється **Добавление** користувач повинен задати ім'я таблиці, до якої він хоче приєднати дані з результуючого набору записів запита. Закриття діалогового вікна за допомогою кнопки **ОК** призводить до того, що Access додає в бланк запиту рядок **Append To**. У цей рядок автоматично (або вручну) вставляються імена тих полів цільової таблиці, що збігаються з іменами полів запиту. Далі варто виконати запит, натиснувши кнопку **Run (!)** на панелі інструментів. У спеціальному діалоговому вікні Access зазначить, скільки записів буде додано до цільової таблиці, і запропонує підтвердити виконання цієї операції.

### Запит на видалення

Запити цього типу служать для видалення з таблиці груп записів, що відповідають деякому критерію відбору. Оскільки записи видалені за допомогою запиту, не можна відновити, варто старанно аналізувати критерії відбору.

Отже, насамперед необхідно розробити запит на вибірку видаляємих записів, указавши критерії в рядку **Критерии**, виконати цей запит, щоб перевірити коректність заданих критеріїв, і повернутися в режим конструктора.

Після цього слід активізувати на вкладці Конструктор команду **Удаление**. Access додасть у бланк запиту рядок **Удалить** і введе в її комірки значення **Условие**. Це означає, що користувач може встановити додаткові критерії відбору. Далі, натиснути **!** на пенелі, запускаємо. У спеціальному діалоговому вікні Access зазначить, скільки записів видалиться з таблиці, і запропонує підтвердити видалення.

### Запит на поновлення

Використовуючи запит на поновлення, користувач може змінити групу записів, відібраних на основі визначених критеріїв. При розробці запиту на поновлення насамперед створюється і перевіряється запит на вибірку. Після цього в режимі конструктора активізується команда **Обновление з вкладки Конструктор**, унаслідок чого Access додає в бланк запиту рядок **Update To**, що призначений для вказівки нових значень полів таблиці. У якості типових можуть виступати і обчислювальні вирази. Після натискання кнопки **!**, запуск, Access зазначить у спеціальному діалоговому вікні, скільки записів зміниться в таблиці, і спробує підтвердити зміни.

### Запит на створення таблиці

На основі результуючого набору записів запиту можна побудувати нову таблицю за допомогою запиту на створення таблиці. Такі запити звичайно застосовують для архівації старих записів або для зберігання резервних копій таблиць.

Спочатку необхідно підготувати запит на вибірку й одержати динамічний набір даних Recordset, заодно перевіривши коректність запиту. Якщо результуючий набір записів

відповідає вашим вимогам, поверніться у режим конструктора й виберіть команду **Создание таблицы**. Access відчинить діалогове вікно **Создание таблицы**, в яке варто ввести ім'я нової таблиці. Далі потрібно виконати запит, натиснувши кнопку зі знаком оклику на панелі інструментів. У спеціальному діалоговому вікні Access зазначить, скільки записів добавляється в нову таблицю, і запропонує підтвердити виконання цієї операції.

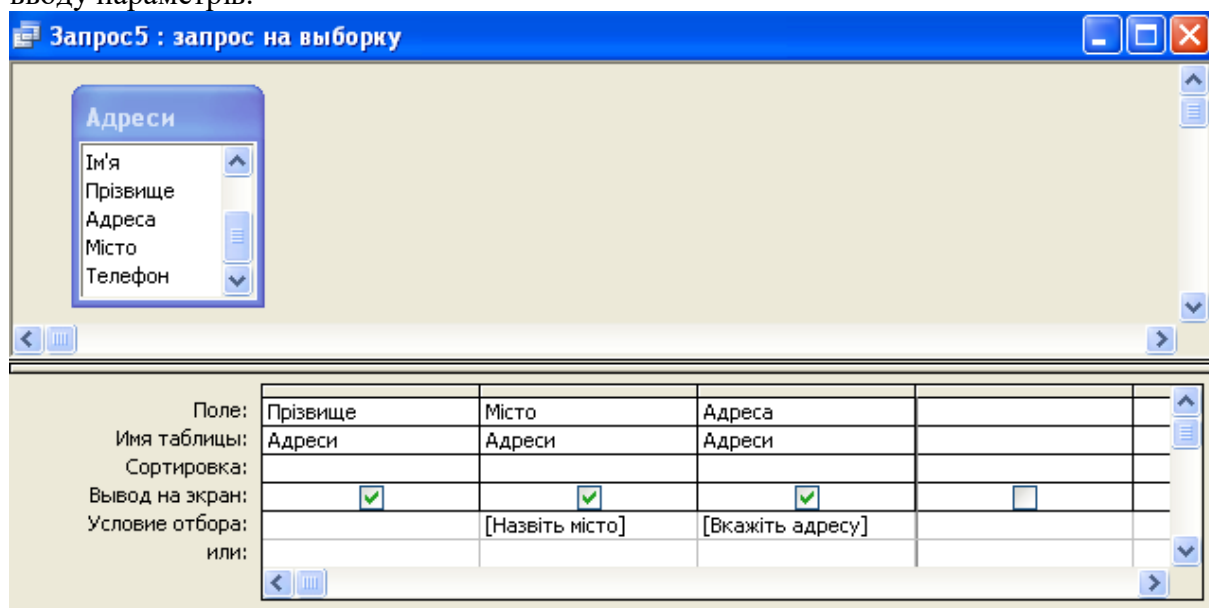
### Параметричні запити

Запити, що являють собою варіанти базового запиту і незначно відрізняються один від одного, називаються параметричними.

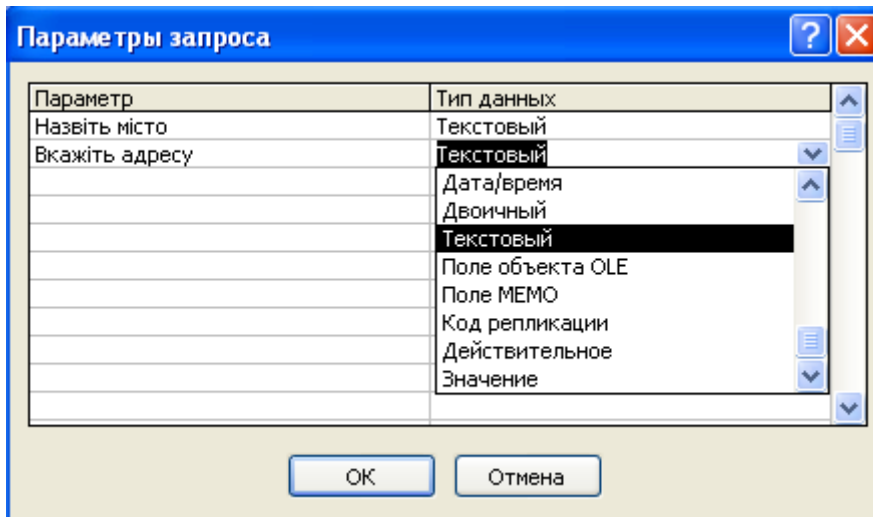
Розглянемо створення параметричного запиту для відбору з таблиці Адреси прізвища осіб, які проживають в зазначеному місті (поле Населений пункт ) і по визначеній адресі (поле Адреса ).

Насамперед розробимо запит на вибірку даних. Включимо в нього поля Прізвище, Населений пункт і Адреса. Поля рядка **Критерии** заповніть не конкретними критеріями відбору, а звертанням до користувача для запровадження критерію. На основі цього критерію (з урахуванням інших критеріїв, заданих у специфікації запиту) буде здійснюватися добір записів. Звертання необхідно узяти в квадратні дужки, наприклад: [Назвіть місто], [Зазначте адресу].

Активізуйте команду **Параметры** вкладка Конструктор. У поле **Параметры** діалогового вікна **Параметри запроса** введіть текст, що знаходиться в полі **Критерии** специфікації запиту. У даному випадку дужки не потрібні. Після визначення типу даних у полі **Тип данных** і натискання кнопки **ОК** запит буде готовий. Тепер можете приступити до вводу параметрів.

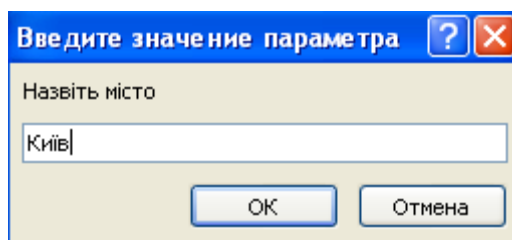


Малюнок 7



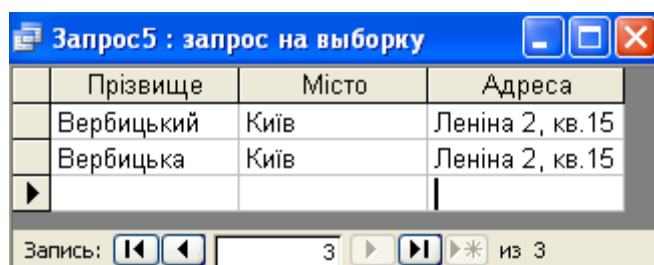
Малюнок 8

Для перевірки працездатності запиту перейдіть у режим таблиці або !. При цьому Access звернеться до вас із проханням задати значення критерію у вікні **Введіть значення параметра**.



Малюнок 9

Привласніть першому параметру значення *Київ*, а другому- *Леніна 2, кв.15*. Після запровадження параметрів буде отриманий результат виконання параметричного запиту (мал.10).



Малюнок 10

### Перехресні запити

Перехресні запити дозволяють компактно відобразити відібрані дані й об'єднувати одночасну інформацію. Таблиці, у яких відображаються результати виконання перехресного запиту, зручні для створення діаграм й графіків за допомогою майстра.

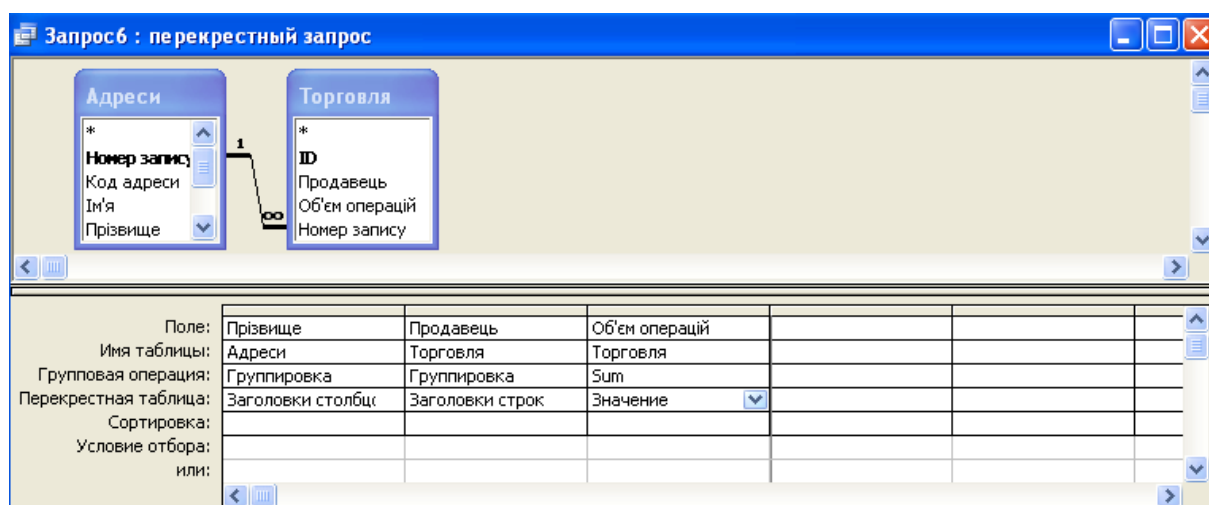
Визначимо об'єм виторгу від продажу різноманітних товарів за умови, що відповідна інформація зберігається в базі даних окремо для кожного місяця. При використанні перехресного запиту найменування товару з'являється в результуючому наборі даних тільки один раз, Access об'єднує дванадцять записів по місяцях. Цим перехресний запит відрізняється від запиту на вибірку, при виконанні якого в результуючий набір даних включається окремо кожен з дванадцяти місячних записів по даному товару.

Створимо перехресний запит *Продавець - Покупець* на основі таблиці *Торгівля*, використовуючи для вибору прізвищ покупців таблицю *Адреси*.

На вкладці Конструктор активізуйте команду **Перекрестный**. Access відобразить у бланку запиту рядки **Total** і **Перекрестная таблица**. Включіть в бланк запиту поля *Продавець* і *Обсяг операцій* із таблиці *Торгівля* і поле *Прізвище* з таблиці *Адреси*. Щоб значення поля (наприклад, *Продавець*) з'явилося в заголовках рядків, натисніть на рядку **Перекрестная таблица** і виберіть у списку рядок **Заголовки строк**. У рядку **Total** для цього поля повинна зберегтися стандартна установка **Группировать по**.

Щоб значення поля *Прізвище* з'явилися в заголовку колонки, натисніть на рядку **Перекрестная таблица** і виберіть рядок **Заголовки столбцов**. У цьому випадку в рядку **Total** повинна залишитися стандартна установка **Группировать по**.

Для відображення змісту поля в перехресній таблиці в якості значення (Обсяг операцій) клацніть на рядку **Перекрестная таблица** і виберіть рядок **Значение**. У рядку **Total** для цього поля введіть необхідну функцію- Sum.



Малюнок 11

При визначенні критеріїв відбору в рядку **Total** для бажаного поля слід вибрати значення **Условие** і ввести вираз в рядок **Критерии**. Для цього поля рядок **Перекрестная таблица** залиште порожнім. Критерії відбору ми вводити не будемо.

Щоб відобразити на екрані результуючий набір записів створеного запиту, варто перейти в режим таблиці.



Малюнок 12

### Створення перехресного запиту за допомогою майстра

Процес розробки перехресного запиту з попереднього приклада за допомогою спеціального майстра. Цей майстер дозволяє створити перехресний запит на основі тільки однієї таблиці чи одного запиту.

У вікні БД перейдіть на вкладку **Создание** і клацніть на кнопці **Мастер запросив**. **Перекрестный** запит вибираємо і слідуючи підказкам виберіть таблицю або простий запит на основі якого майстер створить перехресний запит та інше поки не отримаємо потрібний результат.

#### Контрольні питання до лабораторної роботи 4:

1. Яке поле називають обчислювальним полем?
2. Що являє собою вираз?
3. Перелічіть правила побудови виразів.
4. Перелічіть оператори, які використовують в виразах.
5. Які функції називають вбудованими функціями?
6. Яким чином можна обробляти порожні записи у запитах?
7. Для чого призначений запит на вибірку?
8. Що змінюється при визначенні властивості «Підпись» обчислювального поля?
9. В яких випадках застосовують функцію IsNull()?
10. Як застосувати функцію ПФ в обчислювальних полях?
11. Запити яких типів ви знаєте?
12. Які дії можна робити за допомогою запитів на зміну?
13. Чим відрізняються запити на зміну від запитів на вибірку?
14. Що представляють собою параметричні запити в чому їх зручність?
15. Яким чином будують перехресні запити?
16. Чи можна додавати інформацію в таблиці БД з інших БД?
17. Чи створює запит на зміну динамічний набір записів Recordset в результаті?
18. Як перевірити дію запитів на зміну?
19. Які запити використовують для зберігання резервних копій таблиць БД?
20. Які можливості розкриває запит на поновлення?
21. Запити якого типу служать для видалення з таблиць груп записів, що відповідають деякому критерію відбору?

## Лабораторна робота № 5.

**Тема: Побудова інтерфейсу користувача. Майстер форм. Режим конструктора форм.**

**Мета: Навчитися створювати та редагувати форми для бази даних. Вивчити всі засоби роботи з формою у режимі конструктора.**

### Зміст роботи за варіантом індивідуального завдання:

1. Створити форму на основі одної таблиці за допомогою майстера форм.
2. Швидким способом побудувати форму на основі запиту.
3. Створити «Сводную таблицу».
4. Побудувати «Сводную діаграму».
5. Відкрийте створену за допомогою майстра форму у режимі конструктора, застосовуючи основні засоби для роботи з формою. Додайте елементи за допомогою панелі елементів, списку полів.
6. Додавши до своєї форми області Заголовка, Приміток, верхнього та нижнього колонтитулів, заповніть всі області необхідними даними.
7. Змініть вигляд форми за допомогою панелі властивостей форми (застосувати властивість Підпись, Рисунок, Ширина та ін.).
8. Створіть форму, в якій забороніть користувачу змінювати інформацію.

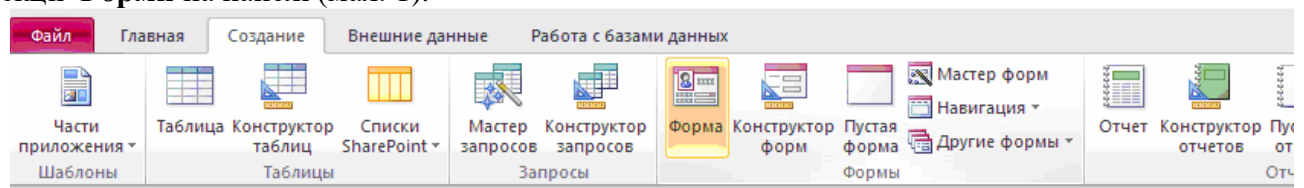
### Теоретичні відомості:

#### Створення та використання форм.

Система Access дозволяє розташовувати на екрані дані так, як того бажає користувач, тобто у вигляді різноманітних форм. Форма може використовуватись також для введення, редагування, пошуку даних, управління ходом виконання програми, виведення різних повідомлень, друку інформації.

Для створення форми можливо застосування майстра Access. Але, точне формування макета форми в відповідності з вимогами, які запланував користувач в процесі її проектування, забезпечується засобами конструювання форм (режим **Конструктор форм**).

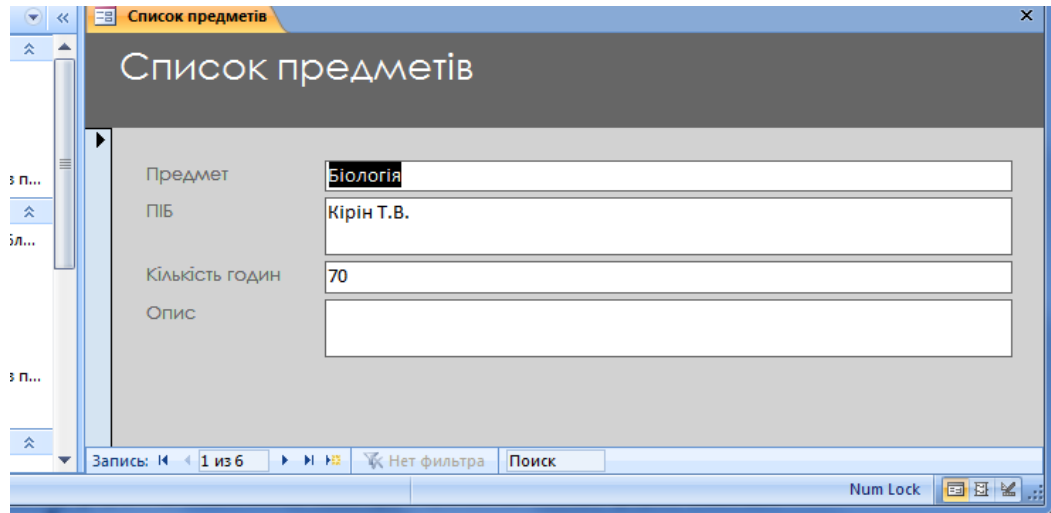
Для створення необхідно вибрати вкладку **Создать** і натиснути необхідний елемент з секції **Форми** на панелі (мал. 1).



Малюнок 1

#### 1. Майстер форм.

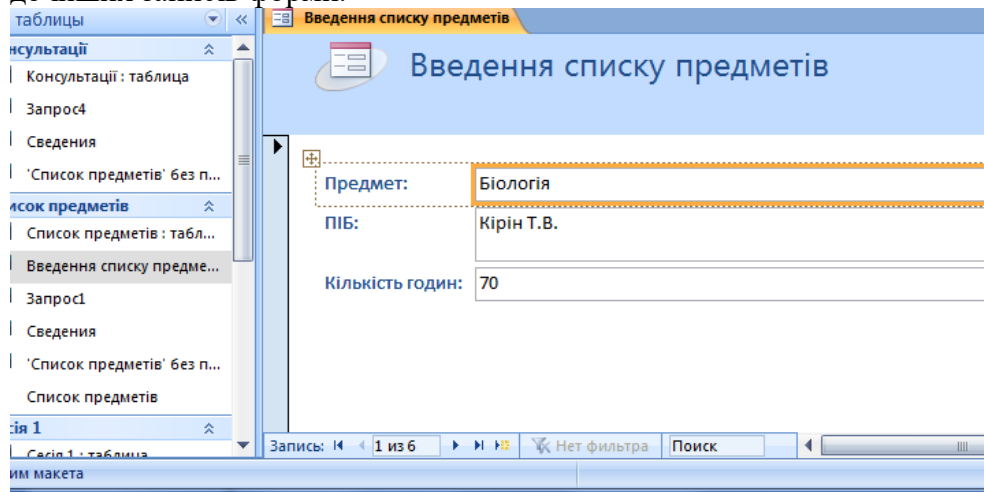
Для створення форми за допомогою майстра форм необхідно: **Создать, Другие формы, Мастер форм** і за підказками майстра, вибравши джерело даних, зовнішній вигляд стиль форми, ім'я та ін. отримаємо готову форму (мал.2)



Малюнок 2

### 2 Форма.

Для швидкого створення форми для введення даних у БД по одному запису за раз можливо скористатись методом Форма. Для цього необхідно: вибрати об'єкт, на основі якого потрібно створювати форму у списку всіх об'єктів (в даному випадку це запит Введення списку предметів), потім вкладка **Создание**, вибираємо **Форма**. Готова форма, яку отримуємо на мал. 3. У нижній частині форми за допомогою кнопок прокрутки можливо добратись до інших записів форми.

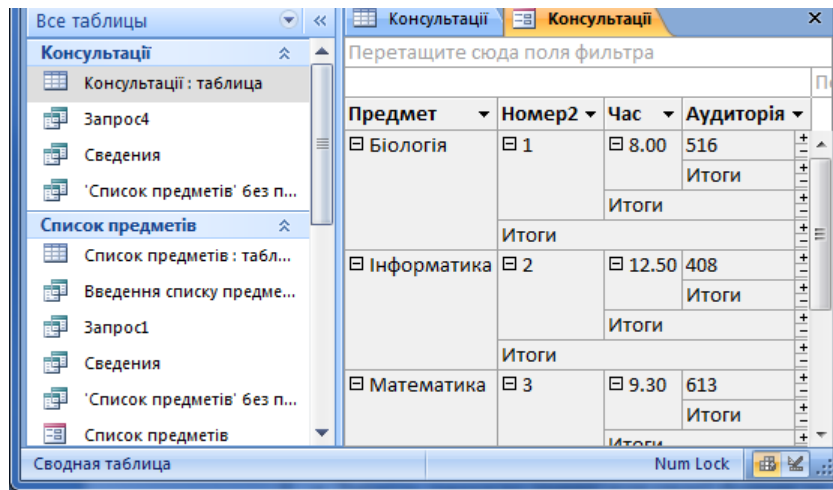


Малюнок 3

### 3 Сводная таблица.

Необхідно вибрати з панелі вкладку **Создание, Другие формы, Сводная таблица**.

Потім додаються необхідні дані (поля) до 4 областей форми таблиці в якості джерел назв рядків, стовпців, фільтру, ітогів чи деталей. В результаті отримуємо готову форму у вигляді Сводная таблица (мал. 4). Кнопки з зображенням знаку + та знаку -, призначні **Показати/приховати** деталі.



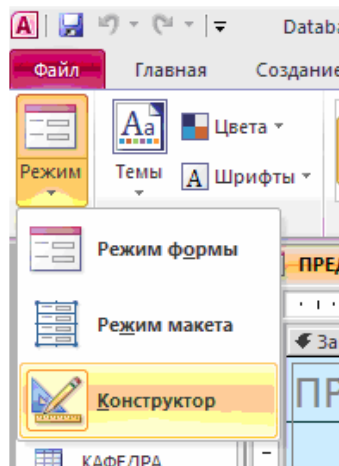
Малюнок 4

#### 4. Сводная диаграмма.

Необхідно вибрати джерело діаграми з списку об'єктів ліворуч, потім з вкладки **Создание, Сводная діаграмма**. Для належного вигляду діаграми потрібно в необхідному місці вибрати потрібні поля.

#### 5. Режим конструктора.

Після створення форми швидкими методами є можливість виправити її у режимі конструктора. Наприклад, для уточнення тексту надписів, місця положення, розміру, шрифту та інших параметрів відображення елементів форми. Під час відкритої форми перехід в режим конструктора можливо здійснити натиском на вкладці **Режимы** кнопки **Режим** (мал. 5).



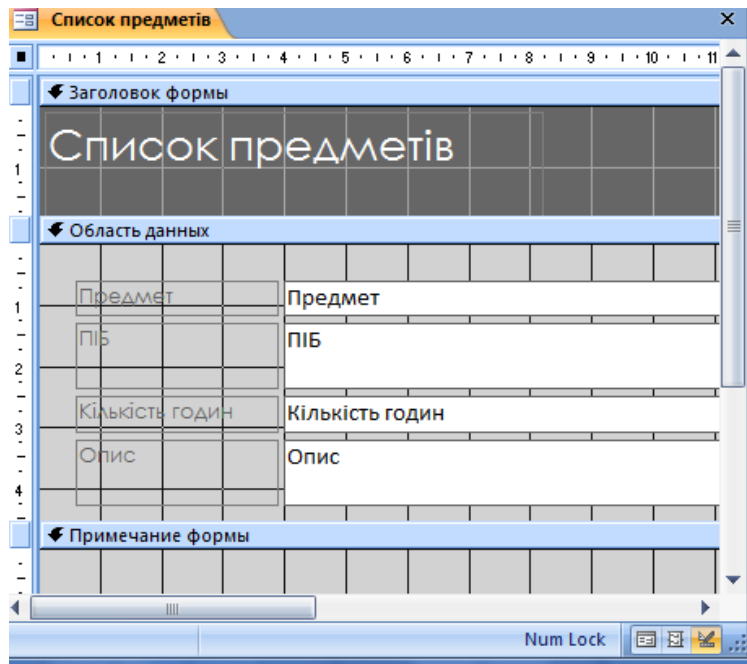
Малюнок 5

Перехід в режим конструктора можливо здійснити також за допомогою **Контекстного меню** потрібної форми. Форма в режимі конструктора виглядає так (мал.6). В режимі конструктора форма містить область заголовка, область даних і розділ приміток та засоби для її редагування.

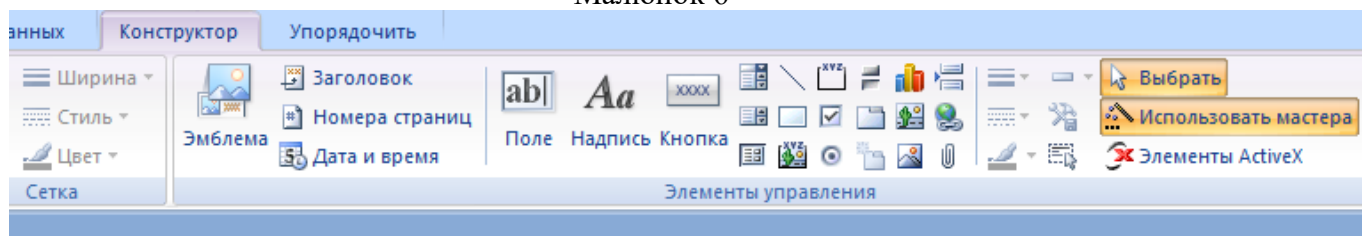
\* **Елементи управління** на вкладці Конструктор (мал. 7). Містить всі елементи керування, що використовуються в формах для введення і відображення інформації (в залежності від типу інформації).

\* Можливість додавати поля за допомогою **Добавить существующие поля**. Список полів, подібний списку полів для роботи в режимі конструктора запитів. Цей список залежить від джерела (таблиці або запити), на якому заснована форма. За допомогою миші ви можете перемістити поля з списку полів в форму, подібно тому, як це робиться в запитах.

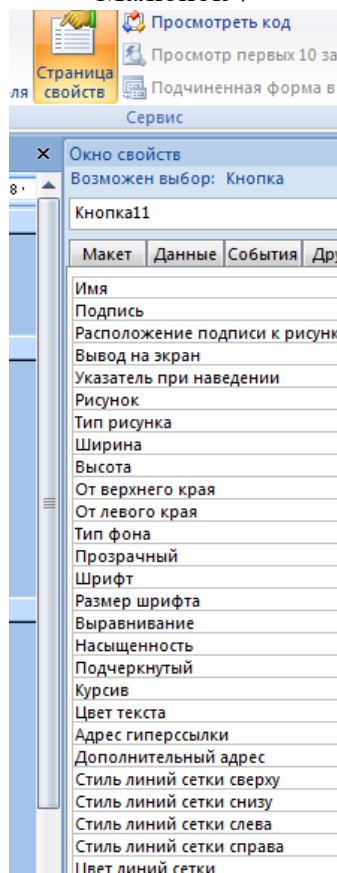
\* Панель властивостей (**Страница свойств** - мал. 8). На панелях властивостей ви знайдете властивості того об'єкту, якому вони належать. Прикладами властивостей є **Подпись, Источник записей** і т.д.



Малюнок 6



Малюнок 7



Малюнок 8

### Категорії властивостей

Подвійним натиском миші на області виділення форми можна викликати на екрані панель властивостей форми, що містить вкладки, що призначені для того, щоб поділити

властивості на категорії. Для різноманітних об'єктів Access властивості будуть однакові, а число властивостей в кожній категорії буде змінюватись. Ось вид цих категорій:

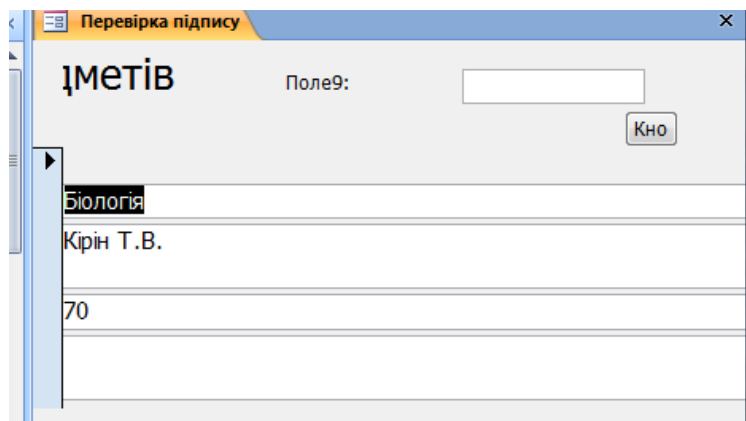
- **Макет.** Визначає, як повинні виглядати об'єкти. Прикладами є властивості **Ширина**, **Кнопки розміров окна**, **Кнопки розміров і Рисунок**.
- **Данные.** Визначає, яким чином Access використовує з об'єктом дані. Властивість **Источник записей** форми говорить про те, що таблиця або запит використовуються в формі, тут же ви бачите властивості **Фильтр**, **Порядок сортировки** і т. д.
- **События.** Дозволяють вам приписати певні команди різноманітним подіям, що відбуваються з об'єктом. Прикладом такої події може служити подія On-Click для будь-якої кнопки керування. Це означає, що коли ви натиснете на цій кнопці керування, Access виконає ту задачу, яку ви забажали виконати.
- **Другие.** Всі інші властивості, які не потрапили в раніше перераховані категорії.

Будь-яка форма будується на основі Access-таблиці або запиту. Імена полів беруться зі специфікації таблиці, а поля у формі можна розташувати за своїм розсудом. При цьому варто визначити послідовність розташування полів у формі, об'єднавши логічно взаємозалежні поля в групи, а також внести у форму елементи керування. На основі однієї таблиці користувач може побудувати декілька форм: першу - для керівника, другу - для бухгалтера, третю - для робітника складу і т.д.

Вся інформація форми міститься в її елементах керування, деякі з них безпосередньо пов'язані з полями базової таблиці. У таких елементах можна показати зміст відповідних полів таблиці і внести в них зміни. Інші елементи форми, наприклад назви полів, служать для її оформлення.

### Зміна властивості Подпись

Щоб змінити значення властивості Подпись, натисніть на вкладці **Макет** панелі властивостей. Змініть значення властивості **Подпись** на Перевірка підпису. Переходимо в режим форми, в рядку заголовку форми ви побачите те, що надрукували в полі властивості **Подпись** (мал. 9).



Малюнок 9

Аналогічно змінюють інші властивості форми. Є і інші властивості, що впливають на зовнішній вигляд форми, наприклад, це **Тип рисунка**, **Масштабы рисунка**, **Выравнивание рисунка** і **Мозаичное заполнение** і т.д.

### Властивості з вкладки Данные

Деякі з перерахованих в таблиці властивостей **Данные** допомагають при роботі з формами.

Властивість	Опис
<b>Источник записей</b>	Властивість, яка визначає те, на чому ґрунтується форма

<b>Разрешить изменение</b>	Виберіть <b>Да</b> або <b>Нет</b> , щоб дозволити або заборонити користувачу змінювати інформацію в формі
<b>Разрешить удаление</b>	Виберіть <b>Да</b> або <b>Нет</b> , щоб дозволити або заборонити користувачу видаляти записи
<b>Разрешить добавление</b>	Виберіть <b>Да</b> або <b>Нет</b> , щоб дозволити або заборонити користувачу додавати нові записи

Ми розглянули найпростіші типи форм. Microsoft Access дозволяє створювати досить складні форми. Існують так звані підлегли, зв'язані, багато сторінкові, смугові та інші форми розробка яких не набагато складніша за ті форми, які ми розглядали.

У формі, як і в запиті, також можна здійснити пошук необхідних даних. При цьому методика роботи щодо пошуку даних близька до аналогічної методики у запитах.

Проектування екранних форм – це захоплююча творча робота, яка є подякою від СКБД, за те що ви нею користуєтесь.

Будь-яка форма будується на основі Access-таблиці або запиту. Імена полів беруться зі специфікації таблиці, а поля у формі можна розташувати за своїм розсудом. При цьому варто визначити послідовність розташування полів у формі, об'єднавши логічно взаємозалежні поля в групи, а також внести у форму елементи керування. На основі однієї таблиці користувач може побудувати декілька форм : першу - для керівника, другу - для бухгалтера, третю - для робітника складу і т.д.

Вся інформація форми міститься в її елементах керування, деякі з них безпосередньо пов'язані з полями базової таблиці. У таких елементах можна показати зміст відповідних полів таблиці і внести в них зміни. Інші елементи форми, наприклад назви полів, служать для її оформлення.

### **Контрольні питання до лабораторної роботи 5:**

- 1. Що таке форми? Де їх застосовують?**
- 2. На основі яких даних створюють форми?**
- 3. Які способи створення форм існують?**
- 4. Який з способів створення найлегший? Чому?**
- 5. Перелічіть режими відображення форм. Як переключатися між режимами?**
- 6. Як зробити різними зовнішній вигляд та стилі ваших форм?**
- 7. Що представляє собою «Сводная таблица»?**
- 8. Якими способами можна вводити та редагувати інформацію у базі даних?**
- 9. У якому випадку використовують форми для роботи з даними?**
- 10. Які засоби для роботи з формою існують?**
- 11. Які властивості впливають на зовнішній вигляд форми?**
- 12. Що змінилось у вашій формі після зміни властивості Підпис форми?**
- 13. Який засіб для роботи з формою містить всі елементи керування?**
- 14. Яким чином можна викликати панель властивостей форми?**
- 15. Яким чином можна додати на форму елемент поле?**

## Лабораторна робота №6

**Тема: Побудова інтерфейсу. Робота з елементами керування в формах бази даних. Різні види форм. Побудова меню роботи з БД.**

**Мета: Навчитися використовувати елементи керування в формах, та застосовувати їх для розширення можливостей форм. Навчитися працювати з багатосторінковими формами, кнопковими формами, головною з підлеглою формами та ін.**

### **Зміст роботи за варіантом індивідуального завдання:**

При побудові нових форм:

1. Додайте змістовний заголовок до області заголовку форм, поля та надписи до області даних ваших форм, елементи до області приміток форм.
2. Створіть кнопки для виходу та відкриття іншої форми.
3. Створіть прапорці або перемикачі, які працюють.
4. Додайте малюнок (фірмовий знак, логотип), графічні елементи.
5. Для кожного елемента форми за допомогою панелі властивостей визначте властивості: ім'я, малюнок, прозорість, відображення на екрані та ін.
6. Створіть форму з підлеглою формою за допомогою майстра форм.
7. Змініть вигляд підлеглої форми (складна структура, режим відображення).
8. Створіть початкову кнопку форму, яка повинна відкривати інші форми вашої бази даних, у тому числі і вашу головну з підлеглою.
9. Створіть головну кнопку форму за допомогою «Диспетчера кнопкових форм».
10. Створити Модальне діалогове вікно-форму для привітання перед запуском головного вікна системи.
11. Зробіть так, щоб ваша програма розпочиналась з запуску головної кнопкової форми з робочого стола комп'ютера.

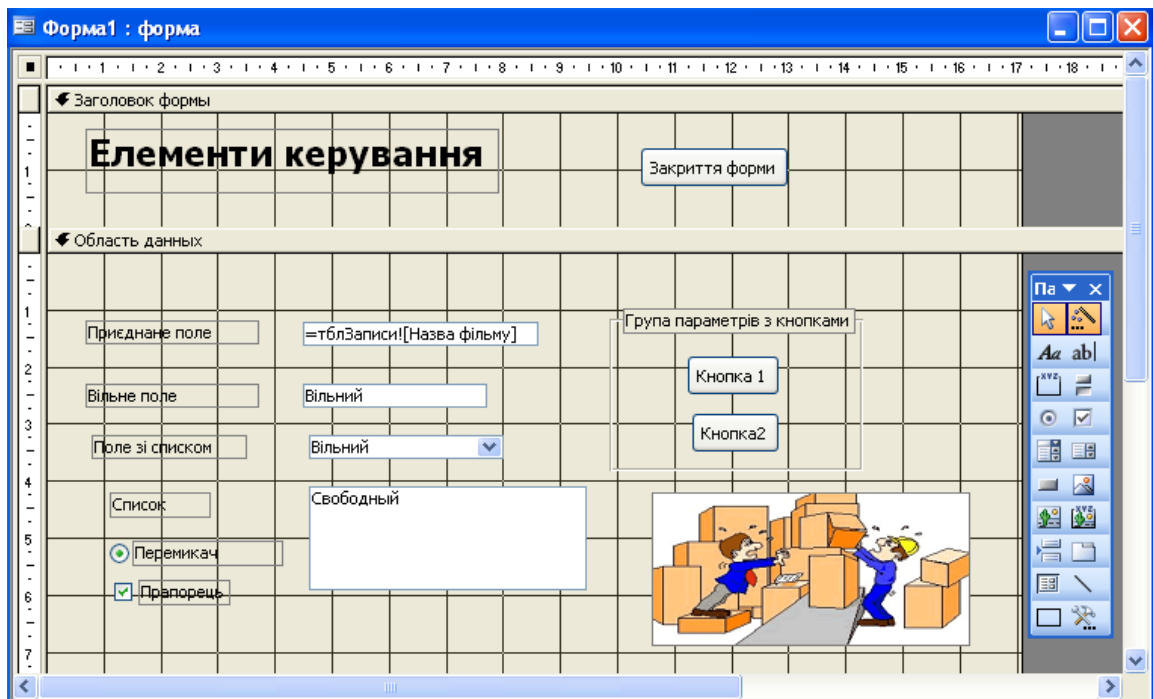
### **Теоретичні відомості:**

#### **Елементи керування Access**

Елементи керування представляють собою об'єкти в формах, звітах або на сторінках доступу до даних, які використовують для відображення даних, виконання дій або покращення інтерфейсу. Наприклад, можна використати поле у формі для відображення даних, кнопку в формі для відкриття іншої форми, лінію або прямокутник для розділу та групування елементів керування, які роблять більш зрозумілим їх призначення.

Елементи керування використовуються, щоб полегшити роботу користувача під час пошуку інформації. Крім того, вони підвищують швидкість введення даних. Прикладами елементів керування є кнопки команд, перемикачі, прапорці опцій, поля зі списками і текстові поля (мал. 1).

Елементи керування бувають зв'язаними, вільними або такими, які обчислюються. Зв'язаний елемент керування прив'язаний до поля базової таблиці або запита. Такі елементи керування використовуються для відображення, введення або відновлення значень з полів бази даних. Для вільного елемента керування джерела даних не існує. Вільні елементи керування використовуються для виводу на екран даних, ліній, прямокутників та малюнків. Для елемента керування, що обчислюється в якості джерела даних використовуються вирази. В виразах можливо використовувати дані з полів базової таблиці або запиту для форми, а також дані іншого елемента керування форми.



Малюнок 1

Якщо в вікні властивостей в якості джерела даних для елемента керування (властивість **Данные**) вказане поле , наприклад, **Название фильма**. Це те, з чим пов'язаний елемент керування. Тому в режимі форми в полі елемента керування з'явиться інформація з поля **Название фильма**. Якщо для властивості **Данные** значення не вказане, елемент керування є вільним.

Буває, що хоча елемент керування і виглядає як два окремих елемента керування, але це дві частини одного цілого: зв'язане поле разом з приєднаним до нього підписом. Підписи приєднуються до більшості елементів керування, призначених для введення даних.

### Переміщення елементів керування незалежно один від одного

Щоб пересунути елемент керування незалежно від приєданого до нього підпису або навпаки пересунути окремо підпис, дійте так:

1. Розмістіть вказівник миші на маркері переміщення тієї частини складеного елемента керування, яку вимагається перемістити (або на підписі, або на текстовому полі). Показчик миші прийме вигляд руки, що вказує вгору.
2. Побачивши цей вказівник, натисніть і не відпускайте ліву кнопку миші.
3. Пересуньте елемент керування в нову позицію в формі, відпустити кнопку миші.

### Одночасне переміщення декількох елементів керування

Для того щоб ви могли перемістити декілька елементів керування водночас, спочатку ви повинні помістити в форму пару нових полів. Щоб захопити більше одного поля відразу, натисніть в списку полів на першому з них. Після цього, утримуючи клавішу <Ctrl> натисніть на наступному полі. Пересуньте обидва поля в форму, помістивши де потрібно, і відпустіть кнопку миші. В результаті поля виявляться в формі.

Щоб водночас пересунути декілька полів в формі, виконайте наступне:

1. Помістіть вказівник миші в формі, під нижнім правим кутом останнього з елементів керування, що необхідно пересунути.
2. Натисніть і не відпускайте ліву кнопку миші.
3. Пересуньте вказівник миші вгору і ліворуч над всіма елементами керування, що необхідно виділити. Ви повинні побачити рамку, захоплюючу ці елементи керування.

4. Захопивши всі елементи керування, що ви хотіли б пересунути, відпустіть кнопку миші.

### Вирівнювання елементів керування

В більшості випадків при розміщенні елементів керування в формі бажано їх чітко вирівнювати. В Access є можливість легко зробити це.

1. Виділити елементи керування, що необхідно вирівняти. При виборі декількох елементів керування для вирівнювання або зміни розмірів вибирайте тільки ту частину складених елементів керування, яку ви хотіли б вирівняти (або розміри якої ви хотіли б змінити).
2. Виберіть пункт **Формат, Вировняйте** з меню конструктора форми. В підміню, що з'явиться ви побачите наступні варіанти вирівнювання елементів керування: **по лівому краю, по правому краю, по верхньому краю, по нижньому краю** або **по узлам сетки**.
3. Виберіть пункт по правому краю з меню **Вировняйте**. Ви побачите, як текстові вікна вирівнюються за самим правим елементом керування.
4. Повторивши пункти 1-3, вирівняйте підписи елементів керування, виділивши і підрівнявши їх по лівому краю.

### Зміна розмірів елементів керування

Існує два засоби зміни розмірів елементів керування, в залежності від їхнього числа.

Щоб змінити розмір одного елементу керування, зробіть наступне:

1. Натисніть на елементі керування, щоб виділити його.
2. Протягніть покажчик миші через маркер розміру. Коли покажчик миші опиниться над маркером розміру, він перетвориться в пару направлених в протилежні сторони стрілок, що вказують в різноманітних напрямках.
3. Утримуючи ліву кнопку миші, пересуньте маркер розміру елементу керування. Ви побачите, як він звужується або розширюється, в залежності від того, в якому напрямку ви його рухаєте.

При зміні розміру декількох елементів керування можна виділити елементи керування, а потім:

-використовувати засіб, описаний вище,  
або

-вибрати пункт **Размер**. Після цього в підменю, яке з'явилося вибрати один з можливих варіантів: **по размеру данных, по узлам сетки, по самому высокому, по самому низкому, по самому широкому і по самому узкому**.

### Робота з властивостями елементу керування

Можна змінити властивості елементу керування так само як і змінити властивості форми.

Щоб побачити, як змінюються властивості елементу керування, давайте змінимо властивість **Текст строки состояния** в вікні властивостей поля Назва предмету. Для цього треба виконати наступне:

-Помістіть курсор в рядок властивості **Другие, Всплывающая подсказка** в вікні властивостей.

-Замініть поточний текст в рядку цієї властивості на *“Це назва дисципліни”*.

-Перейдіть в режим форми.

Тепер, при наведенні курсору на поле Назва предмету, текст *“Це назва дисципліни”* буде спливати поряд як підказка.

Панель інструментів форматування дозволяє змінювати зовнішній вигляд елементів керування.

### Інші типи форм

Ми розглянули найпростіші типи форм. Microsoft Access дозволяє створювати досить

складні форми. Як вже відомо існують так звані підлеглі, зв'язані, багатосторінкові, смугові та інші форми, розробка яких не набагато складніша за ті форми, які ми розглядали.

Багатосторінкова форма – це форма, яка використовується при необхідності вивести на екран записи, кожен з яких містить великий об'єм інформації.

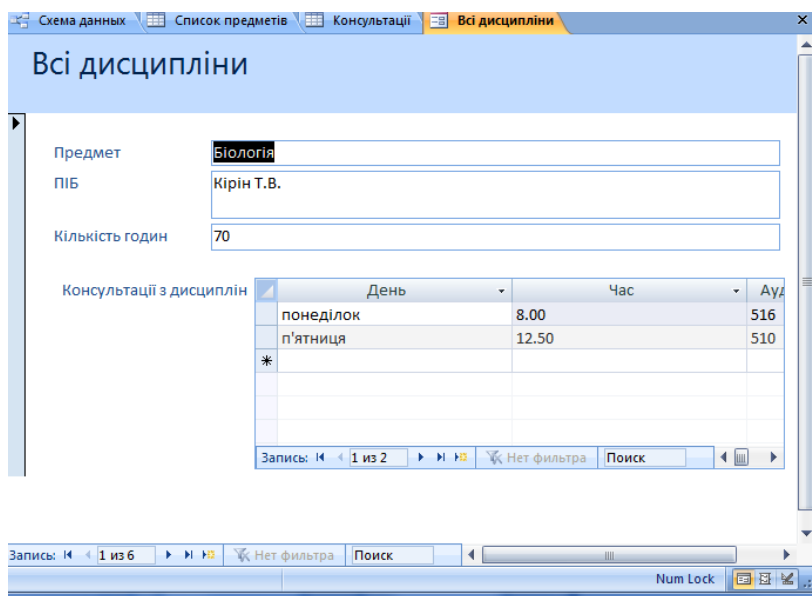
Підлегла форма - це форма, яка знаходиться всередині іншої форми. Первинна форма називається головною формою, а форма всередині форми називається підлеглою формою.

### Зв'язок головної та підлеглої форм в Microsoft Access

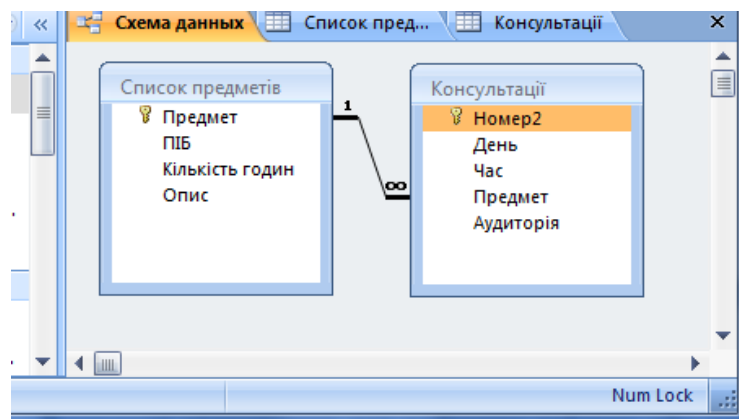
Підлегла форма звичайно використовується для виведення даних з таблиць або запитів, з відношенням «один-до-багатьох». Наприклад, таблиця «Список предметів» та таблиця «Консультації» зв'язані відношенням «один-до-багатьох». Ми розглянемо, наприклад, форму «Список предметів» (мал. 2), яка вміщує підлеглу «Консультації».

Зі створенням форми та підлеглої форми, які ґрунтуються на таблицях, зв'язаних відношенням «один-до-багатьох», головна форма представляє сторону «один» відношення, а підлегла форма представляє сторону «багато» (мал.3).

Головна форма синхронізується з підлеглою формою таким чином, що в підлеглий формі виводяться тільки записи, зв'язані з записом в головній формі. Якщо підлегла форма створюється за допомогою майстра або шляхом перетягування форми, таблиці або запиту з вікна бази даних в іншу форму, Microsoft Access автоматично синхронізує головну форму з підлеглою формою із виконанням двох наступних умов.



Малюнок 2



Малюнок 3

- Визначені відношення для вибраних таблиць або для базових таблиць вибраних запитів.
- Головна форма ґрунтується на таблиці з первинним ключем, а підлегла форма ґрунтується на таблиці, яка зберігає поле з таким же іменем, як і в цього первинного ключа, і з таким же або сумісним типом даних. Наприклад, якщо первинний ключ базової таблиці головної форми має тип даних лічильник, а його властивість **Размер поля (FieldSize)** має значення **Длинное целое**, то відповідне поле в базовій таблиці підлеглої форми повинно мати тип даних **Числовой** з властивістю **Размер поля (FieldSize)**, яка має значення **Длинное целое**.

Головна з підлеглою формою створюються так:

1. Вибираємо **Мастер форм** з **Другие формы**.
2. Вибираємо з списку таблицю або запит. Наприклад, таблицю «Список предметів» (сторона «один» відношення «один-до-багатьох»).
3. Виберіть поля таблиці або запиту, які потрібно включати в форму.
4. В тому ж діалоговому вікні майстра виберіть з списку іншу таблицю або запит, «Консультації» (сторона «багато» відношення «один-до-багатьох»). Послідовність вибору таблиць або запитів не має значення.
5. Виберіть поля таблиці або запита, які потрібно включити в форму.
6. Якщо до запуску майстра зв'язки були встановлені коректно, то виберіть перемикач **Подчиненные формы**.
7. Слідуючи інструкціям інших діалогових вікон майстер Microsoft Access, у результаті створить дві форми.

З використанням форми з підлеглою формою для вводу нових записів поточний запис в головній формі зберігається з переходом до підлеглої форми. Це гарантує, що записи з таблиці на стороні «багато» будуть мати зв'язаний запис в таблиці на стороні «один». Це також автоматично зберігає кожен запис, який додається в підлеглу форму. Підлегла форма може відображатись у вигляді таблиці, або у вигляді простої чи стрічкової форми. Головна форма може відображатись тільки як проста форма.

Головна форма може мати будь-яке число підлеглих форм, якщо кожна підлегла форма поміщається в головну форму. Є також можливість створювати підлеглі форми до сьомого рівня вкладення. Це позначає, що можна мати підлеглу форму в головній формі, а іншу підлеглу форму в цій підлеглий формі и т.д. Наприклад, можна створити головну форму, яка відображає інформацію про клієнтів, підлеглу форму, яка відображає інформацію про замовлення, та ще одну підлеглу форму, яка відображає інформацію про замовлення товарів.

Під час створення підлеглої форми користувач має можливість вказати її відображення в наступних режимах: тільки в режимі таблиці, тільки як проста або стрічкова форма в режимі форми або в будь-якому вигляді. Підлеглі форми, відображені як прості або стрічкові форми, мають бути простими або мати достатньо складну структуру, включаючи заголовки, області приміток і т.д.

### **Визначимо режим відображення підлеглої форми**

Під час створення підлеглої форми за допомогою майстра форм або майстра підлеглих форм, Microsoft Access встановлює значення властивості **Режим по умовчанию (DefaultView)** підлеглої форми в відповідності з тим, що вибрав користувач в діалогових вікнах майстра. Наступні дії можуть бути корисними у створенні підлеглої форми без допомоги майстра, або для зміни режиму, вибраного в майстрі.

1. Відкрийте головну форму в режимі конструктора. Підлегла форма буде виведена у режимі конструктора в елементі керування **Подчиненная форма**.
2. Виберіть елемент керування «підлегла форма». Тепер можна натиснути його ще раз та встановити властивість форми як і у випадку відкриття підлеглої форми в режимі конструктора з вікна бази даних.
3. Двічі натисніть область виділення форми, щоб відкрити вікно властивостей підлеглої форми.

4. В **Режим по умолчанию (DefaultView)** виберіть один з режимів: **Простая форма**, **Ленточная форма** або **Таблица**.

### Створення кнопкової форми

Якщо створювати базу даних за допомогою майстра автоматично створюється кнопкова форма, яка допомагає пересуватись базою даних. На кнопковій панелі містяться кнопки, при натисканні яких відкриваються форми або звіти (або відкриваються інші кнопкові форми, за допомогою яких відкриваються додаткові форми та звіти), виконується вихід з Microsoft Access або налагоджується сама кнопкова форма. Диспетчер кнопкових форм дозволяє створювати кнопкові форми, які подібні тим, що створює майстер баз даних.

### Створення кнопкової форми за допомогою «Диспетчера кнопкових форм»

Створюючи базу даних за допомогою майстра автоматично створюється кнопкова форма, яка дозволяє легко переміщуватись формами та звітами бази даних. В базі даних, що вже існує створити кнопку форму можна за допомогою «Диспетчера кнопкових форм»:

1. Вкладка **Работа с базами данных**, натиснути **Диспетчер кнопковых форм**.
2. В діалоговому вікні **Диспетчер кнопковых форм** натисніть кнопку **Изменить**.
3. В діалоговому вікні **Изменение страницы кнопочной формы** наберіть ім'я кнопочної форми в полі **Название кнопочной формы** та натисніть кнопку **Создать**.
4. В діалоговому вікні **Изменение элемента кнопочной формы** наберіть у полі **Текст** текст для першого елемента кнопкової форми та виберіть команду в полі із списком **Команда**.
5. Для більшості команд під полем **Команда** відчиняється нове поле зі списком. За необхідністю, виберіть потрібний елемент в цьому полі.
6. Таким чином створюють інші кнопки кнопкової форми. Щоб змінити або видалити яку-небудь з створених кнопок, виберіть її ім'я з списку **Элементы данной кнопочной формы** та натисніть кнопку **Изменить** або **Удалить**. Якщо потрібно змінити порядок елементів кнопкової форми, виберіть елемент з списку та скористайтесь кнопками **Вверх** або **Вниз**.
7. Натисніть кнопку **Закреть**.

Диспетчер кнопкових форм дозволяє створювати кнопкові форми, які відкривають інші кнопкові форми, щоб назначити кнопку форму **стартовою** (тобто, яка автоматично відкривається з відкриттям бази даних), виберіть ім'я кнопкової форми в діалоговому вікні **Диспетчер кнопковых форм** та натисніть кнопку **По умолчанию**. Щоб із натиском кнопки цієї кнопкової форми відчинялась інша кнопкова форма, на кроці 5 описаної вище процедури виберіть команду **Перейти к кнопочной форме** у полі **Команда** та вкажіть кнопку форму, на яку потрібно перейти.

При створенні кнопкової форми за допомогою диспетчера кнопкових форм Microsoft Access створює таблицю «Switchboard Items», яка описує текст і дії кнопок форми. Якщо потім змінити кнопку форму в режимі конструктора форми, програма може перестати працювати. Якщо потрібно буде значно змінити кнопку форму, краще всього створити її заново і назначити стартовою.

Якщо перетягнути форму на робочий стіл вашого комп'ютера, то користувачу буде зручніше працювати з базою даних, запускаючи її з ярлика.

Щоб створити модальне діалогове вікно-форму привітання у своїй БД, необхідно: вибираємо вкладку **Создание, Другие формы, Модальное диалоговое окно**. Потім за допомогою потрібних елементів скласти текст привітання, додати графічні елементи для приємного зовнішнього вигляду та визначити дії кнопок. Якщо необхідно по натисненню на ОК, викликати головну кнопку форму, необхідно це визначити у властивостях кнопки.

## **Контрольні питання до лабораторної роботи 6:**

- 1. Що таке елемент керування?**
- 2. Яких типів елементи керування бувають?**
- 3. Які елементи називають вільними?**
- 4. Які елементи називають зв'язаними?**
- 5. Які елементи називають обчислювальними?**
- 6. Перелічіть елементи керування, які присутні на панелі елементів.**
- 7. Якими способами можливо змінювати розташування та розмір елементів керування?**
- 8. Перелічіть типи форм, які ви знаєте.**
- 9. Яку форму називають головною формою?**
- 10. Яку форму називають підлеглою?**
- 11. В яких випадках використовують підлеглі форми і яким чином повинні бути зв'язані таблиці на основі яких створюють такі форми?**
- 12. Зі створенням форми та підлеглої форми, які ґрунтуються на таблицях, зв'язаних відношенням «один-до-багатьох», яку сторону відношення представляє підлегла форма? Продемонструйте на схемі даних.**
- 13. Якими двома способами можна створити підлеглу форму?**
- 14. Із виконанням яких двох умов Microsoft Access автоматично синхронізує головну форму з підлеглою формою?**
- 15. У якому вигляді може відобразитись підлегла форма?**
- 16. У якому вигляді може відобразитись головна форма?**
- 17. Яке число підлеглих форм може мати головна форма, якщо кожна підлегла форма поміщається в головну форму?**
- 18. До якого рівня вкладення є можливість створювати підлеглі форми?**
- 19. Що представляє собою головна кнопкова форма?**

## Лабораторна робота №7

Тема: Побудова інтерфейсу. Макроси. Звіти.

**Мета:** Навчитися застосовувати макроси та швидко будувати звіти за допомогою майстра звітів. Ознайомитися з можливостями конструктора звітів. Навчитися створювати групи та сортувати дані у звітах.

**Зміст роботи за варіантом індивідуального завдання:**

1. Побудуйте простий макрос, який відкриває запит, таблицю, форму, але завантажується автоматично під час запуску БД.
2. Побудуйте простий звіт швидким методом.
3. Створити звіт за допомогою майстра звітів.
4. Розглянути звіт у режимі конструктора. Виконайте вказівки удосконалення звітів з пункту “Покращення звіту”(дивись теоретичні відомості).
5. Різними способами зробіть групування й сортування інформації. Роздрукуйте.
6. Побудуйте поштову наклейку за допомогою майстра поштових наклеюк.
7. Використайте майстра діаграм для зображення інформації у своєму звіті.
8. Створіть обчислювальні поля. Додайте в звіт номер сторінки, дату й час. Роздрукуйте отриманий вдосконалений звіт.
9. Створіть на формі кнопку, при натисненні якої викликався б макрос з командою попереднього перегляду звіту, якщо сьогодні 29 число, інакше відкривалась би таблиця для введення даних.

### Теоретичні відомості:

#### Макроси

Мова макросів - це засіб автоматизації БД, дозволяє перетворити виснажливе повторення послідовностей операцій в просту процедуру, що складається лише в натисканні спеціальної кнопки або комбінації клавіш, або в активізації меню.

Для відкриття вікна макросів вибираєте *Создание*, в секції Другие, Макросо включає чотири стовпчики (рисунок 1): *Имя макроса*, *Условие*, *Макрокоманда* і *Примечание*.

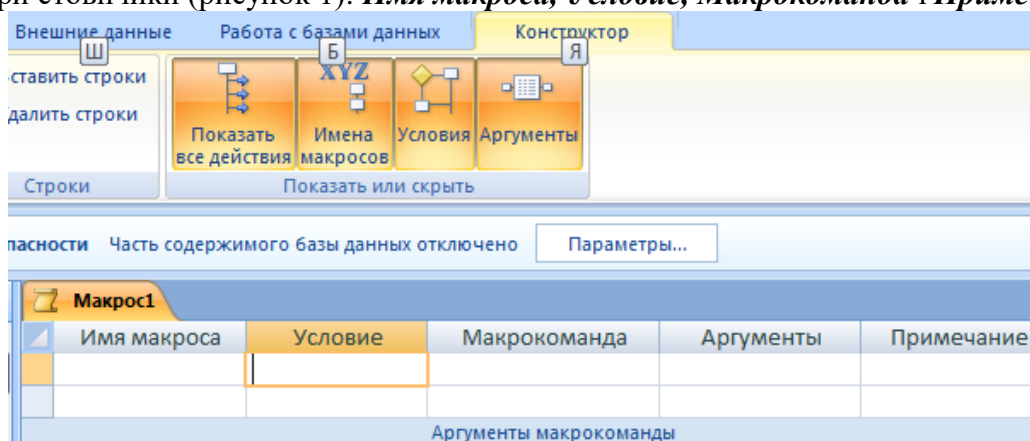


Рисунок 1 – Нове вікно макросів

При створенні нового макроса по замовчуванню відображаються тільки стовпчики *Макрокоманда* і *Примечание*. Показ інших установлюється за допомогою вибору відповідних опцій на вкладці Конструктор.

У стовпчику *Имя макроса* вказується ім'я макроса, що слід задавати, якщо вікно містить декілька макросів. У стовпчику *Условие* використовують умовний вираз. Це такий вираз, значення якого Microsoft Access перевіряє та порівнює з вказаним значенням,

наприклад, в інструкціях If...Then та Select Case. Якщо умова порівняння задовільняється, то виконується одна або декілька операцій. Якщо умова не задовільняється, то включені в умовну конструкцію операції пропускаються, та виконується перехід на наступну інструкцію.

У стовпчику **Макрокоманда** перераховуються дії (макрокоманди), що підлягають виконанню у потрібній послідовності (необхідно вибрати з списку потрібну).

Зручно розробляти макроси для автоматизації нескладних процесів, таких як відкриття і закриття декількох форм або звітів, виведення на екран або друк декількох документів і т.д.

Виконання кожної макрокоманди залежить від значень її аргументів. Всі аргументи вводяться в спеціально відведені для цього поля, розташовані в нижній частині вікна макросу. Аргументи рекомендується задавати в тому порядку, у якому вони розташовані в нижній частині вікна макросу.

Аргумент макрокоманди відкриття об'єктів, введений у поле **Вид**, визначає режим відображення об'єкта на екрані. Припустимі значення цього аргументу відповідають опціям із меню **Вид**.

За допомогою макрокоманди **ВыполнитьКоманду** можна задати виконання більшості команд із меню. Ім'я виконуваної команди вказується в якості аргументу в полі **Команда**.

Редагування макросу здійснюється в режимі конструктора.

### Зв'язування макросів із подіями

У базі даних зберігається інформація про стан її об'єктів. Будь-яка зміна стану форми або звіту називається подією. Кожний із цих об'єктів має свій набір подій. З подіями зручно зв'язувати макроси.

Для того щоб зв'язати макрос із відкриттям бази даних виконайте такі дії:

- Виберіть макрос у вікні бази даних.
- Змініть ім'я макросу на - Autoexec (макрос із таким ім'ям автоматично виконується при завантаженні бази даних).

Тепер макрос пов'язаний із відкриттям бази даних. Таким засобом автоматизуються операції, що повинні завжди виконуватися в момент завантаження бази. При кожному відкритті бази даних Access перевіряє, чи є присутнім в ній макрос з ім'ям Autoexec, і, якщо знаходить, виконує його.

### Автоматизація виводу на друк

Для виводу на друк в залежності від версії передбачені різні макрокоманди. Якщо необхідно роздрукувати звіт:

- Вибираємо макрокоманду ОткрытьОтчет, вказуємо який саме звіт.
- Вибираємо макрокоманду Выполнить команду, вказуємо команду Печать, якщо необхідно, можливо вибрати спочатку попередній перегляд з списку команд.

**Приклад** демонструє використання макросу для виведення звіту на друк. Макрос перевіряє, чи відповідає поточне число місяця, якщо так, то друкує звіт (рисунок 2).

Друк1			
Условие	Макрокоманда	Аргументы	Примечание
Day(Now())=21	ОткрытьОтчет	Наклейки Список предметів;	якщо сьогодні 21-ше число відкриваємо звіт
	ВыполнитьКоманду	Печать	

Рисунок 2

### Використання майстра звітів

Звіт – це такий компонент бази даних, що використовується для того, щоб вибрати дані з однієї чи кількох таблиць, здійснити над ними обчислення, підвести підсумки і вивести їх на екран чи на принтер. Звіти є близькими до форми, але формати **звітів** відповідають стандартам паперу. Для роботи зі звітами найчастіше використовують режим конструктора.

Можливості звітів. З їх допомогою можна групувати інформацію, що виводиться, здійснювати проміжні обчислення. Для кожної групи можна задати окремі заголовки і

примітки. Більша частина даних у звіті отримується з базової таблиці, запиту або інструкції SQL, яка є джерелом даних для звіту. Інша інформація про звіт зберігається у його структурі.

Access пропонує різні засоби для створення звітів. Існує можливість швидко створити найпростіший звіт за допомогою функції **Отчет**. Для цього виберіть джерело даних для звіту серед об'єктів ліворуч, натисніть на вкладці **Создать і Отчет**. Наприклад, малюнок 3.

Предмет	ПІБ	Кількість годин	Опис
Математика	Костенко А.М.	140	
Фізика	Носенко С.К.	70	
Хімія	Борисов Н.П.	35	
Біологія	Кірін Т.В.	70	
Література	Стойко Б.С.	35	
Інформатика	Абрамов К.К.	70	

Малюнок 3

### Побудова звітів за допомогою майстра звітів

**Создание**, на області **Отчеты** вибираємо **Мастер отчетов**. В діалогових вікнах вибираємо джерела даних для звіту, рівні груп, порядки сортування та обчислень, вид макету, стиль та ін. потрібні для звіту дані.

Обравши інформацію, яку ви хотіли б бачити у звіті, ви здатні організувати її з врахуванням її різних аспектів.

Угруповуючи дані, ви надаєте вказівку Access, які поля необхідно показати "як групу" і як підраховувати підсумки по даним. За допомогою сортування можна розташувати дані у звіті у потрібному порядку.

Щоб змінити групування, використовуйте кнопки вибору, які знаходяться на екрані майстра звітів. Для зразка, погодьтесь з групуванням, яке запропонує Access, і клацніть на кнопці **Далее**.

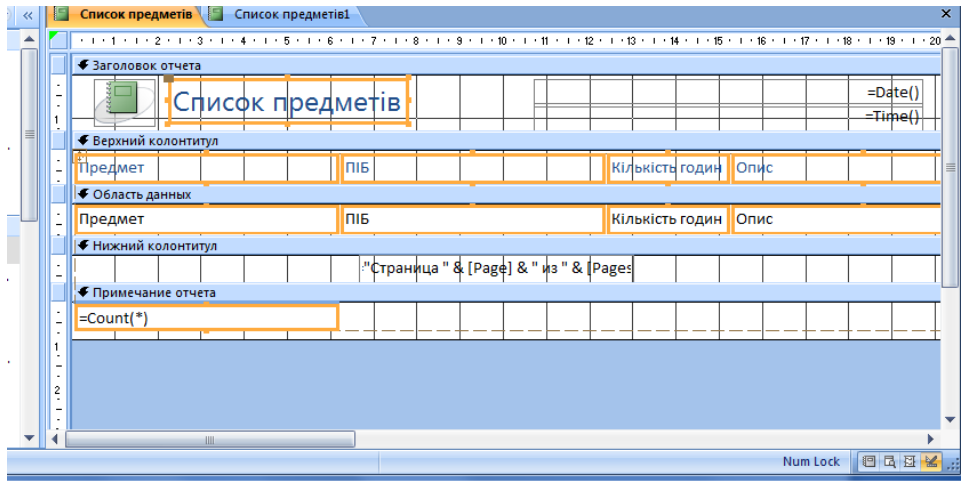
Можливо вибирати макет звіту (ліворуч на екрані, обраний макет буде відображено у графічному вигляді). Окрім того можна вибрати орієнтацію аркушів звіту на екрані - книжкову або альбомну.

На цьому ж екрані можна зробити так, щоб всі поля були на одній сторінці. Інакше, Access надрукує їх на двох сторінках. Обравши потрібне, натисніть на **Далее**.

Ви можете вибрати стиль зі списку різних стилів. **Стиль** звіту надає інформації потрібний зовнішній вигляд, в залежності від аудиторії, на яку він розрахований. **Далі** в полі для імені звіту надрукуйте заголовок звіту. Цей текст буде використано у режимі попереднього перегляду у рядку заголовку і в назві звіту.

Відредагувати швидко створений звіт можливо у режимі конструктора звітів.

Клацніть на кнопці **Вид**, щоб перейти у режим конструктора звітів. Ознайомимось з деякими інструментами в режимі конструктора звітів (мал. 4).

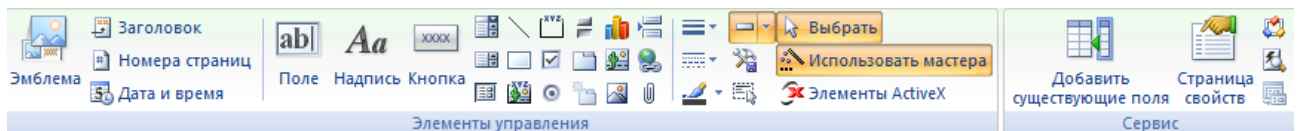


Малюнок 4

Дані у звіті можуть бути розбиті на розділи. Кожен розділ має конкретне призначення та друкується на сторінці та у звіті у заданому порядку.

Нижче приведено список деяких інструментів мал. 5:

- **Елементи керування.** Тут ви знайдете всі елементи, які використовуються у звіті.
  - **Список полів.** Складається з усіх полів джерела записів для звітів.
  - **Сторінка властивостей.** Показує всі властивості усіх об'єктів, які використовуються для звіту, такі як елементи керування, розділи звіту і сам звіт.
- та розділів звіту:
- **Область заголовка звіту.** Включається інформація, зазвичай що міститься на обкладинці: назва звіту та дата. Друкується перед верхнім колонтитулом на першій сторінці звіту.
  - **Область верхнього колонтитула.** Тут взагалі розташовуються назви стовпців, або назва звіту, якщо треба, щоб воно повторювалось на кожній сторінці.
  - **Область нижнього колонтитула.** Використовується для вказівки номера сторінки та посторінкової інформації.
  - **Область даних.** Призначена для відображення даних окремих записів з джерела записів та звіту. Основна складова частина звіту.
  - **Область примечаний.** Призначається для приміток, інструкцій користувача а також ітогових значень та ін.



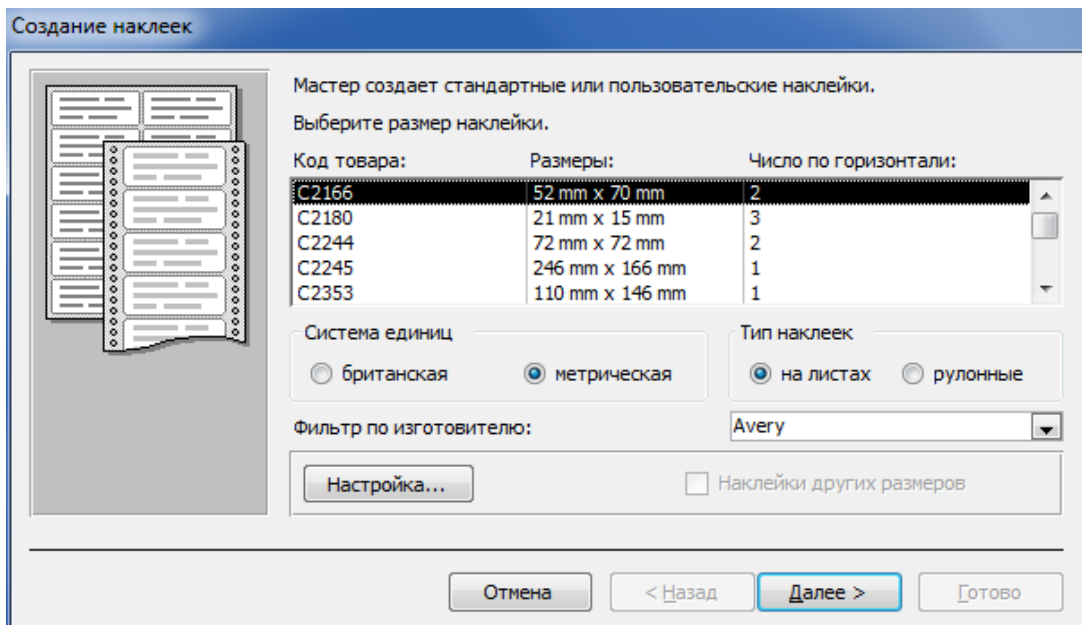
Малюнок 5

### Створення наклейок за допомогою майстра

В Microsoft Access наклейки створюються у вигляді звіту, відформатованого з невеликим розміром сторінки в відповідності з розміром наклейки. Частіше наклейки застосовуються для поштових відправлень, але в форматі наклейок можливо надрукувати будь-які дані.

За допомогою майстра по створенню наклейок можна створити поштові наклейки, бейджи, цінники, візитки та ін. Для цього зробить наступні кроки:

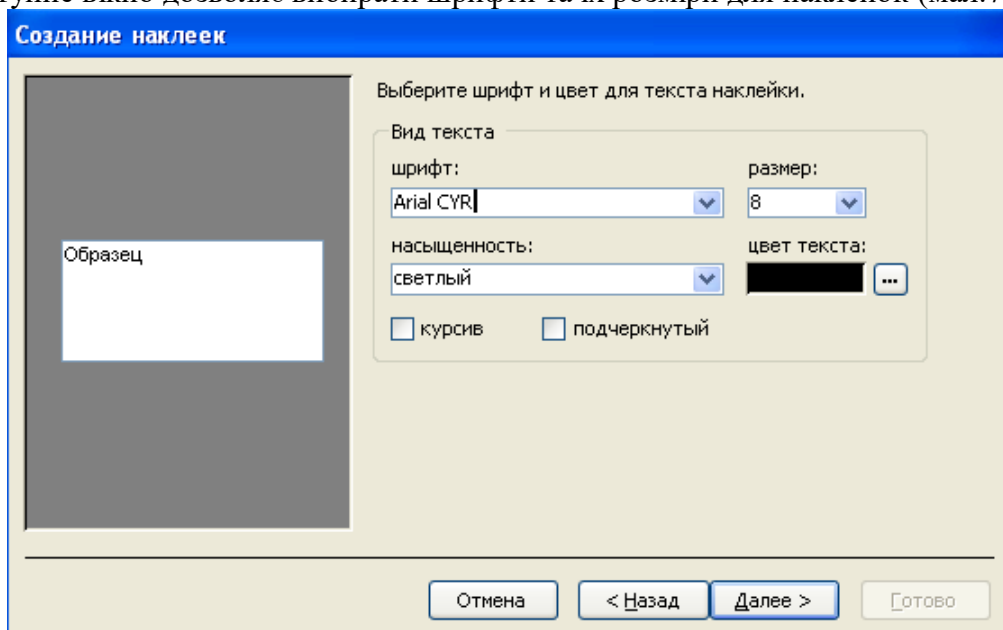
1. Віберіть таблицю або запит, як джерело даних.
2. Вкладка **Создание**. Група **Отчеты**.
3. Потім натисніть **Наклейки** (мал. 6), Access запустить майстра з створення поштових наклейок.



Малюнок 6

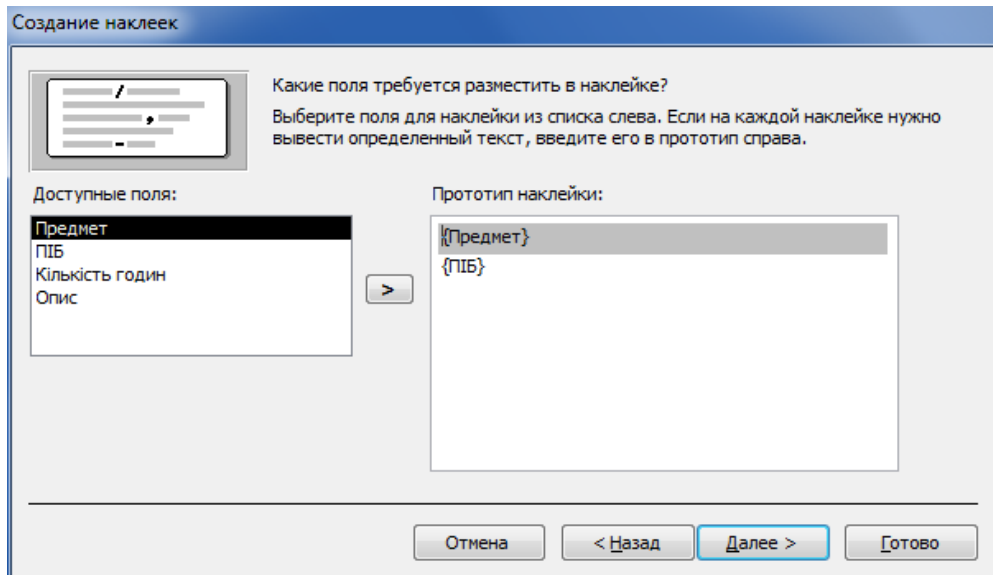
На першому екрані майстра вам потрібно вказати номер Avery, який буде використаний при створенні поштових наклеек. Avery являється стандартом при роздрукуванні поштових наклеек на принтерах. Після закінчення роботи з майстром вже неможливо повернутися назад. **Далее.**

Наступне вікно дозволяє вибирати шрифти та їх розміри для наклеек (мал.7). **Далее.**



Малюнок 7

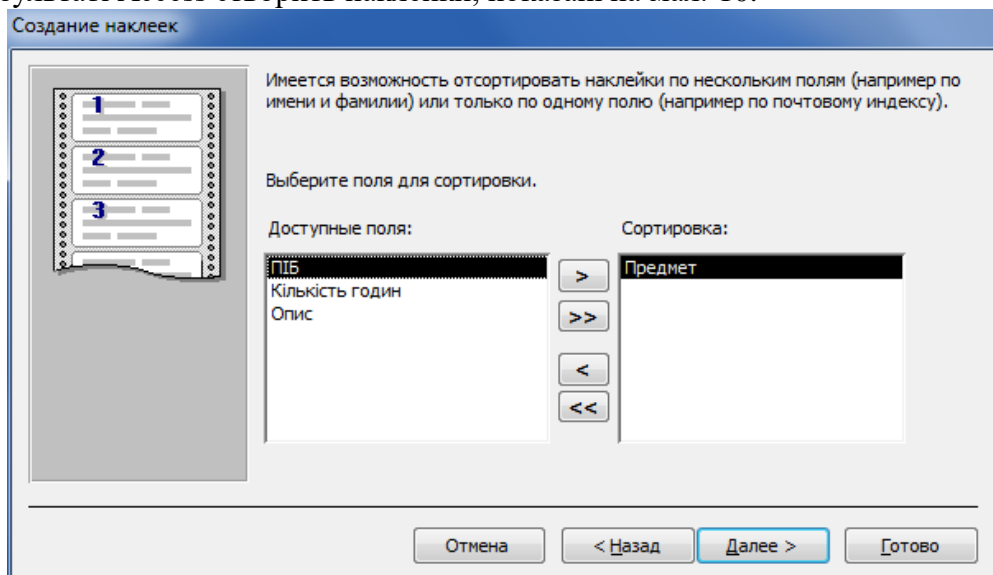
Ви можете оформити наклейку саме так, як вам потрібно, обираючи поля й вводячи відповідний текст. В прототипі необхідно так розміщати поля, як вам буде зручно, вставляючи необхідний текст та розділові знаки (мал. 8).



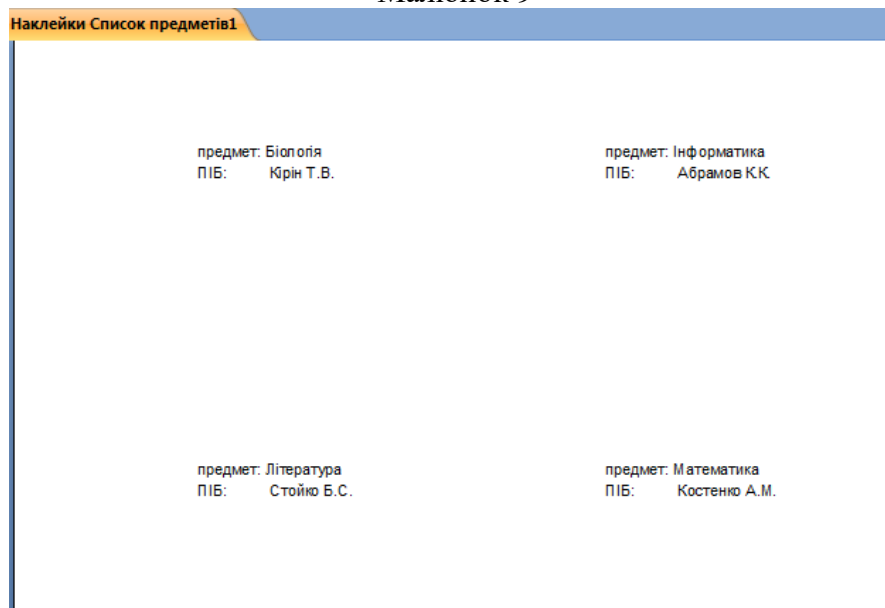
Малюнок 8

Ви можете вибрати поля для сортування в наклейках (мал. 9). **Готово**, щоб завершити роботу з майстром.

В результаті Access створити наклейки, показані на мал. 10.



Малюнок 9



Малюнок 10

В більшості випадках недоліки макету наклейки виправляються внесенням змін у режимі конструктора.

### Друкування звітів

В Access підтримується стандарт Windows, який дозволяє або попередньо продивитися звіт перед друкуванням за допомогою спеціальних опцій, або відразу друкувати звіт на принтері. Також підтримується діалогове вікно, яке дозволяє друкувати звіти на різних принтерах. Розглянемо попередній перегляд.

### Попередній перегляд

Перегляд звіту особливо корисний при частих змінах формату.

Знаходячись в режимі попереднього перегляду, можна використовувати цілий ряд команд з панелі інструментів.

Таблиця :

<b>Команда</b>	<b>Описання</b>
Печать	Дозволяє друкувати звіт з режиму попереднього перегляду.
Масштаб	Дозволяє переключатися між режимом <b>Окно</b> , в якому звіт підганяється під розміри вікна, й розміром звіта у відсотках .
Одна страница	Переключає в режим односторінкового перегляду.
Две страницы	Переключає в режим двосторінкового перегляду. Він зручний, якщо ви бажаєте подивитися, який вигляд буде мати звіт з обох сторін листа. Особливо він зручний при форматуванні, хоча ви можете використовувати вказівник миші, щоб побачити частину сторінки в збільшеному масштабі.
Другие страницы	Ця кнопка дозволяє переглядати на екрані 4,8,12 сторінок звіту.
Word	Експорт виділеного об'єкту у форматі RTF
Текстовый файл	Експорт виділеного об'єкту у текстовий файл
Дополнительно	Додаткові можливості для експорту

Якщо натиснути на кнопці **Печать** панелі інструментів в режимі попереднього перегляду або в режимі конструктора, звіт буде одразу ж надруковано на принтері.

### Огляд інструментів конструктора звітів

Access має ряд інструментів для роботи з звітами, які дуже схожі на інструменти конструктора форм. Відкрийте звіт, побудований за допомогою автозвіту. Потім натисніть на кнопці **Вид**, щоб перейти в режим конструктора звітів.

Деякі інструменти, які використовуються в режимі конструктора звітів:

- **Панель елементов.** Тут ви знайдете всі елементи керування, які використовуються в звітах.
- **Список полей.** Містить всі поля з джерела записів для звіту.
- **Окно свойств.** Показує всі властивості різних об'єктів, які використовуються в звіті, таких як елементи керування, розділи звіту й сам звіт.

- **Область заголовка отчета.** Використовується при побудові титульної сторінки звіту. Друкується з самого початку звіту.
- **Область верхнього колонтитула.** Тут розміщуються заголовки стовпців й назва звіту, якщо потрібно, щоб він повторювався на кожній сторінці.
- **Область нижнього колонтитула.** Використовується для вказування номерів сторінок, дати та інших приміток. Деяким подобається вказувати внизу кожної сторінки загальну кількість сторінок.
- **Область даних.** Використовується для відображення даних окремих записів з джерела записів для звіту. Це головна частина вашого звіту.
- **Примечание отчета.** Використовується для приміток, інструкцій, виразів, кінцевих значень.

### Покращення звіту

Ось перелік способів зміни й удосконалення звітів.

- Переміщення надписів з області заголовка звіту в область верхнього колонтитулу й створення кращої назви.
- Зміна властивості **Подпись отчета** на таку, яка краще відображає зміст звіту.
- Видалення із звіту зайвих полів.
- Зміна розмірів тих полів, для яких це необхідно (щоб всі назви, прізвища вміщувались в відповідні поля).
- Додавання графіки до звіту.

### Зміна висоти розділів звіту й переміщення елементів керування

Бувають звіти, для яких потрібно щоб назва звіту з'являлась на кожній сторінці. Для цього треба перемістити назву з області заголовка звіту в область верхнього колонтитула. Для цього потрібно спочатку збільшити висоту колонтитула, а потім зсунути униз поля, які вже знаходяться в цьому розділі звіту й звільнити місце для назви звіту.

### Створення груп й сортування

Access дозволяє вам згрупувати дані в звіті. При цьому ви можете створювати верхні й нижні колонтитули, які відображають сумарні дані по кожній групі.

1. Відкрийте звіт в режимі конструктора.
2. Натисніть на кнопки **Группировка и сортировка**. З'явиться діалогове вікно **Группировка, сортировка и итоги**.
3. В першому стовпці діалогового вікна надрукуйте ім'я поля з вашої таблиці. Ви одразу ж побачите, що в нижній частині з'являться властивості групи.
4. Встановіть значення **Да** для властивості **Заголовок группы**. В звіті з'явиться новий розділ – область заголовка групи “ім'я вашого поля”.
5. Клацніть на властивості **Примечание группы** й встановіть для нього значення **Да**. Закрийте діалогове вікно.

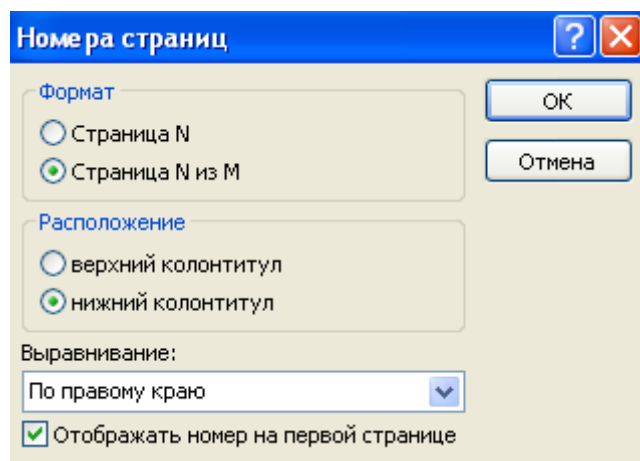
### Створення обчислювальних полів

Обчислювальне поле складається з функцій Access й полів. Щоб додати обчислювальне поле, зробіть наступні кроки:

1. Помістіть елемент керування **поле** в область примітки групи.
2. Двічі клацніть на приєднаному до нього підпису й надрукуйте в ньому текст **Кількість записів**.
3. Щоб встановити властивість **Данные поля**, двічі клацніть в цьому полі й надрукуйте **=Count(\*)**.

Крім функції **Count()**, маються й інші корисні функції. Ви можете, наприклад, використовувати функції **Sum(ім'я поля)** (дає в результаті суму значень поля) й **Avg(ім'я поля)** (середнє арифметичне значення).

Додавання в звіт номера сторінки або дата/час.



Малюнок 11

Щоб додати до звіту номер сторінки, зробіть наступне:

1. В режимі конструктора звіту виберіть команду **Номера страниц**.
2. З'явиться діалогове вікно **Номера страниц** (мал. 11), в якому ви можете вибрати тип й положення номеру на сторінці. Виберіть наступні установки:
  - **Формат Страница N из M**
  - **Расположение нижний колонтитул**
  - **Выравнивание По правому краю**
3. Access розмістить відповідне поле в область нижнього колонтитула у нижньому правому куті звіту.  
Те ж саме можна зробити з часом й датою, вибравши команду **Дата и время**.  
Сітка полегшує процес конструювання.  
Якщо лінійки відсутні, то для їх появи необхідно на вкладці **Упорядочить** в розділі **Размер і порядок** і там вибрати необхідне.

### Контрольні питання до лабораторної роботи 7:

1. Що представляє собою об'єкт Макросы?
2. Які дії можна автоматизувати за допомогою макросів?
3. Що таке звіти? Для чого їх застосовують?
4. На основі яких даних створюють звіти?
5. Які способи створення звітів існують?
6. Що створюють за допомогою майстра звітів?
7. Для чого призначені поштові наклейки?
8. Які можливості дає попередній перегляд звіту?
9. Перелічіть головні інструменти конструктора звітів?
10. Для яких даних призначена область заголовку звіту?
11. Для яких даних призначена область верхнього колонтитулу звіту?
12. Для яких даних призначена область нижнього колонтитулу звіту?
13. Для яких даних призначена область даних звіту?
14. Перелічіть засоби, за допомогою яких покращують звіти.
15. Як додавати до звіту обчислювальні поля?
16. Яким чином можна додати дату і час, номер сторінок у звіт?

## Використана література

1. Пасічник В.В., Резніченко В.А. Організація баз даних та знань. – К.:Видавнича група ВХВ, 2006, - 348 с.
2. Гайна Г.А. Г12 Основи проектування баз даних: Навчальний посібник. – К.: КНУБА, 2005. – 204 с.
3. Access 2000 Шаг за шагом издательство Эком Москва, 2002.
4. С.В. Глушаков Д.В. Ломотько Базы данных Учебный курс Харьков «Фолио» Москва «АСТ» 2001.
5. І.Т. Зарецька Б.Г. Колодяжний А.М. Гуржій О.Ю Соколов Інформатика, Київ, “Навчальна книга”2002.
6. Боб Виллариал Программирование Access 2002 в примерах перевод с английского Москва 2003. - 496
7. А.Ю. Берко О.Н. Верес Організація баз даних Практичний курс. Навчальний посібник. Львів, 2003.
8. Гурвиц Геннадий Александрович Microsoft Access 2010. Разработка приложений на реальном примере.-СПб.: БХВ-Петербург, 2010.- 496 с.

Додаток 1

Приклад звіту з виконаної лабораторної роботи

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Кафедра кібербезпеки та програмного забезпечення

Дисципліна : Бази даних

**ЗВІТ  
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №1**

Тема: Загальні поняття ІС. Архітектура ІС. Робота з таблицями та їх редагування.

**ВИКОНАВ:** студент групи КІ-55

Іванов Я. П.

**ПЕРЕВІРИВ:** викладач

Константинова Л.В.

Кропивницький 2020

МЕТА: Ознайомитись з типами та властивостями полів в Access. Навчитись створювати таблиці. Навчитись редагувати їх та оптимально використовувати Access.

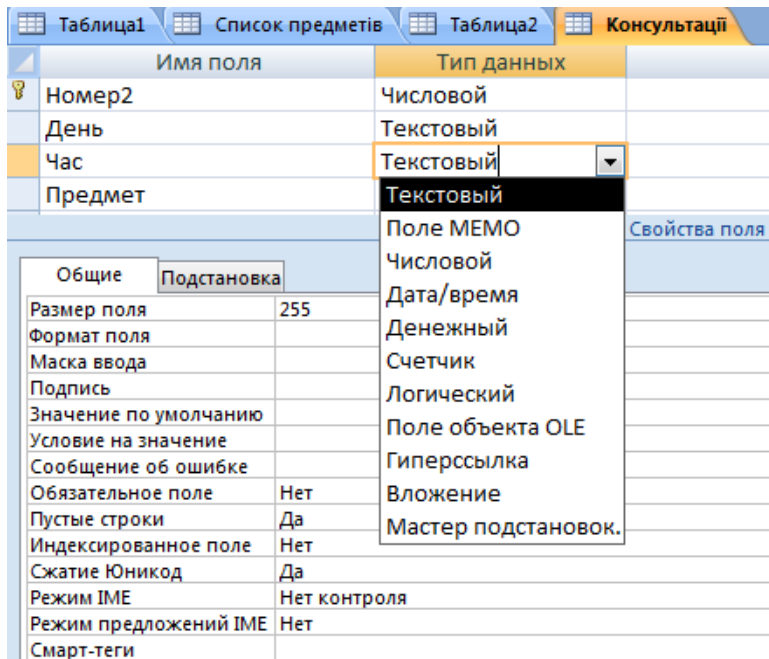
Тема індивідуального завдання: “База даних для навчального закладу”.

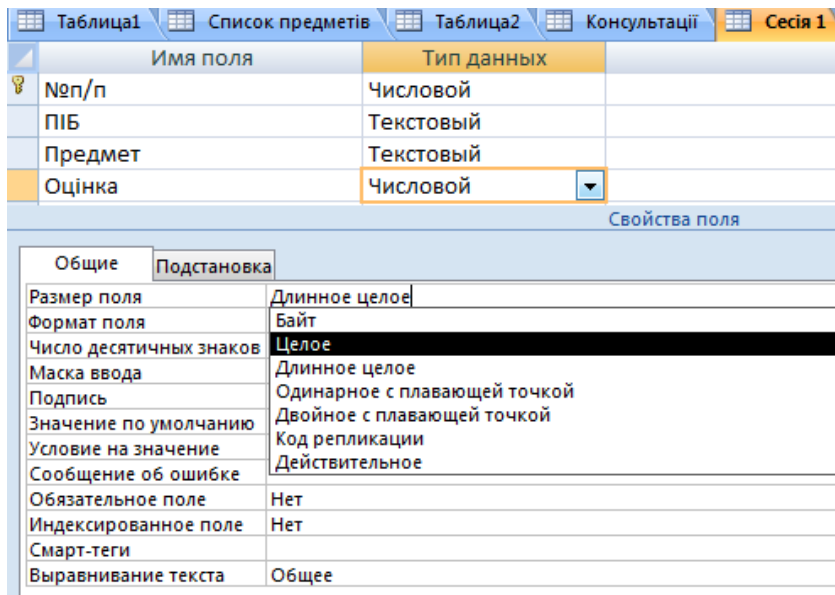
### Завдання:

1. За вибраною предметною областю побудуйте таблиці (дві і більше) з полями різних типів та визначте їх ключові поля.
2. Відредагувати таблиці своєї бази даних за допомогою перелічених контекстних меню.
3. Зробити пошук.
4. Відсортувати записи.
5. Створіть фільтр.
6. Створіть ярлик таблиці та розмістіть його на робочому столі Windows.
7. Приховуйте таблицю вашої БД.

### Хід роботи:

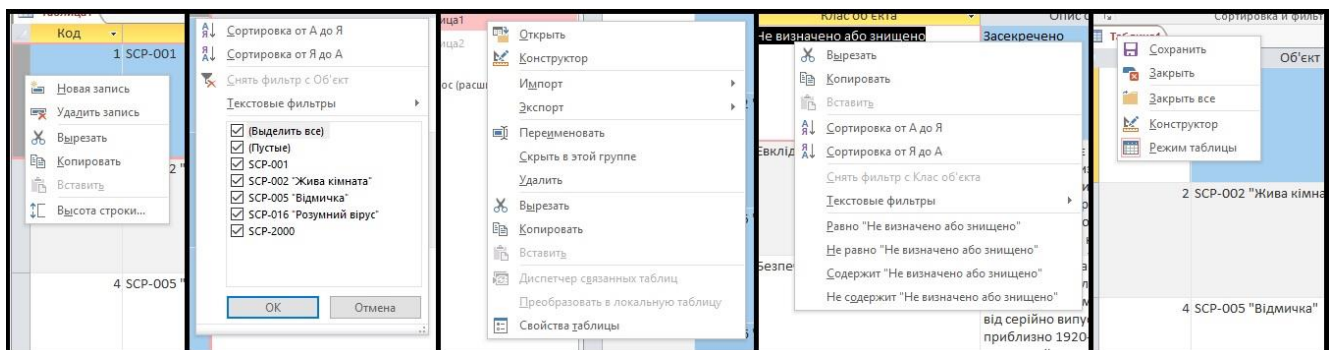
1. По заданій предметній області будує таблиці (дві і більше) з полями різних типів та визначаю їх ключові поля.





2. Відредагувати таблиці своєї бази даних за допомогою перелічених контекстних меню.

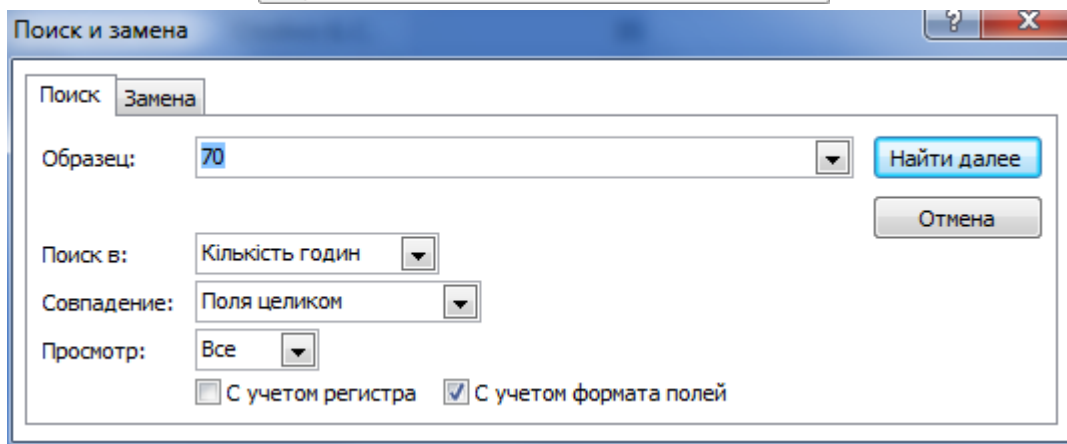
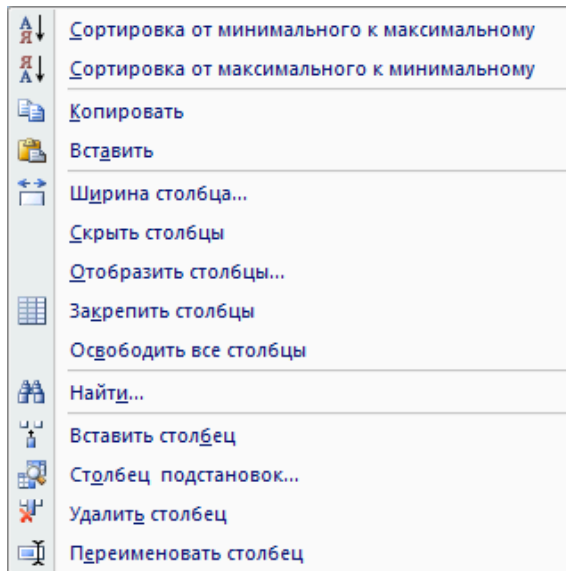
- \* контекстне меню стовпців;
- \* контекстне меню рядків;
- \* контекстне меню таблиці;
- \* контекстне меню окремої комірки таблиці;
- \* контекстне меню назви таблиці.



3. Виконую пошук.

Для таблиці Список предметів викликаю контекстне меню поля Кількість годин і виконую команду Найти:

Предмет	ПІБ	Кількість годин	Опис	Дс
Біологія	Кірін Т.В.	70		
Інформатика	Абрамов К.К.	70		
Література	Стойко Б.С.	35		
Математика	Костенко А.М.	140		
Фізика	Носенко С.К.	70		
Хімія	Борисов Н.П.	35		
*				

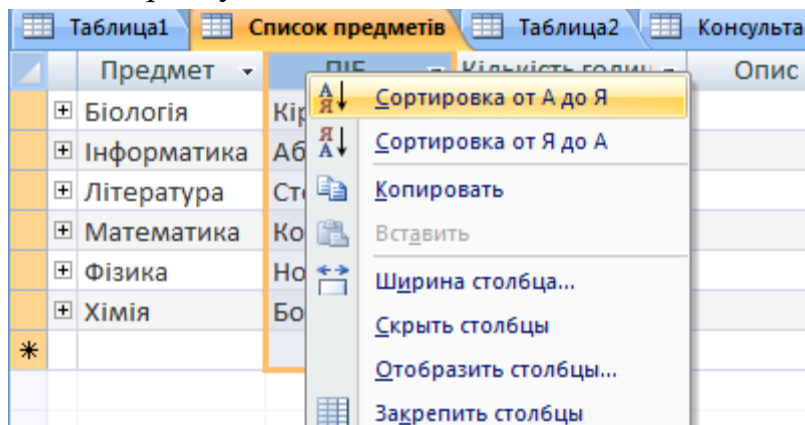


Команда Найти далее

Пошук виконано:

	Предмет	ПІБ	Кількість годин	Опис
+	Біологія	Кірін Т.В.	70	
+	Інформатика	Абрамов К.К.	70	
+	Література	Стойко Б.С.	35	
+	Математика	Костюк А.М.	140	

#### 4. Відсортовую записи.



Відсортована таблиця:

Предмет	ПІБ	Кількість годин
Інформатика	Абрамов К.К.	70
Хімія	Борисов Н.П.	35
Біологія	Кірін Т.В.	70
Математика	Костенко А.М.	140
Фізика	Носенко С.К.	70
Література	Стойко Б.С.	35

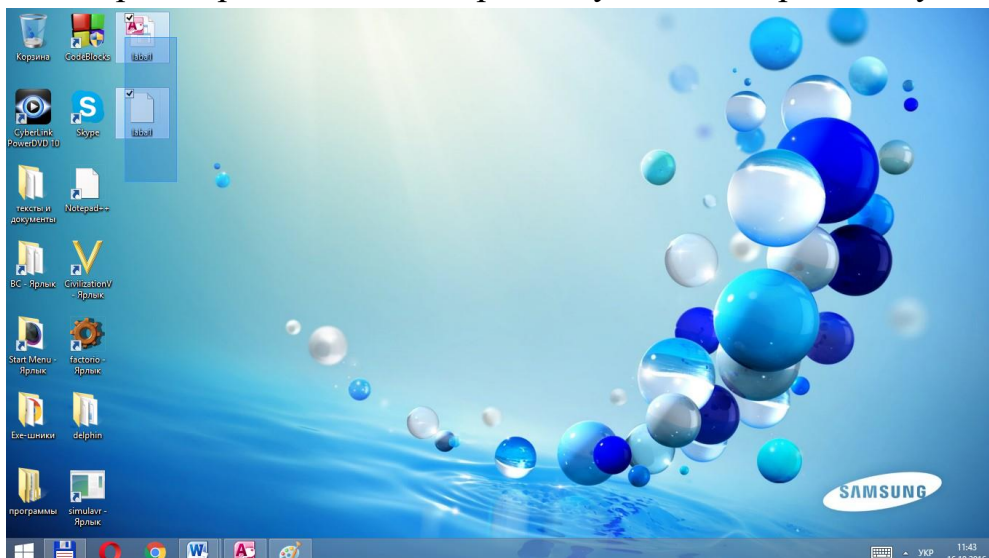
## 5. Створюю фільтр.

The screenshot shows the Excel interface with a table containing subject names, teacher names, and hours. A filter dialog box is open over the 'Кількість годин' column. The dialog box has options for sorting and filtering. Under 'Числовые фильтры', the values 35, 70, and 140 are listed. The value 70 is checked, indicating it is the selected filter criteria.

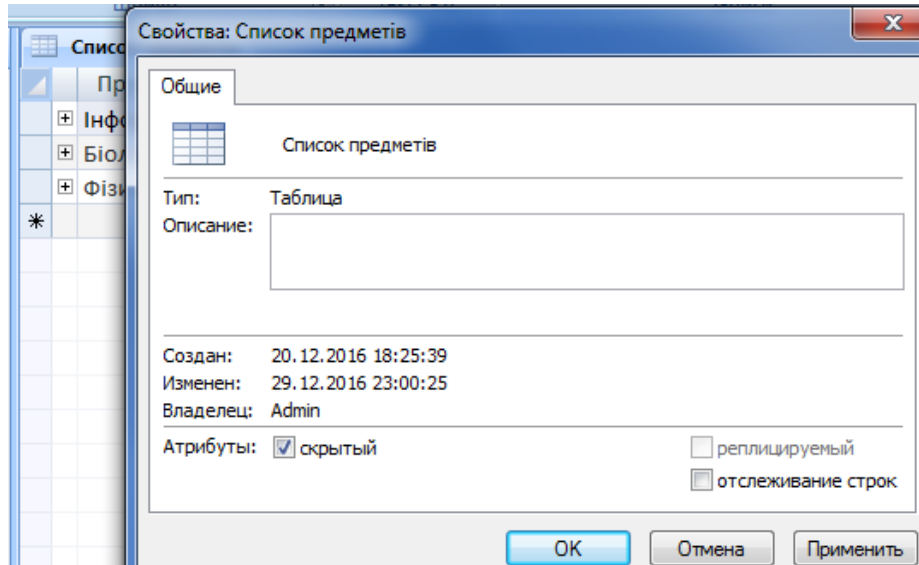
## Результат:

Предмет	ПІБ	Кількість годин	Опис
Інформатика	Абрамов К.К.	70	
Біологія	Кірін Т.В.	70	
Фізика	Носенко С.К.	70	

## 6. Створюю ярлик таблиці та розміщую його на робочому столі Windows.



7. Роблю недоступною одну з таблиць бази даних для інших користувачів.



### Контрольні питання і відповіді на них

1. Що таке СКБД?

Система керування базами даних (СКБД)— комплекс програмного забезпечення, що надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

2. Перелічіть головні об'єкти баз даних.

Таблиця, запит, форма, звіт, макрос, програмний модуль.

3. Які типи полів використовують в Access?

Текстове поле, числове поле, поле дата/час, логічне поле, поле типу лічильник, поле типу memo, поле об'єкту OLE.

4. Коротко розкажіть про суть властивостей текстового поля.

**Розмір поля.** Він має межі від 1 до 255 символів.

**Формат поля.** За допомогою спеціальних символів можна задавати вигляд та розмір рядків, що вводяться.

**Маска введення.** За допомогою спеціальних символів можна визначити тип символу (літера, цифра, пропуск та ін.) у кожній позиції введення.

**Підпис.** По суті це друге ім'я поля.

**Значення за замовчанням.** Дозволяє автоматично вставляти рядки, які часто повторюються, у всі новостворені поля.

**Умова на значення.** Дозволяє створювати фільтр, який забезпечує введення у дане поле тільки тих значень, які відповідають заданій умові.

**Обов'язкове поле.** Набуває тільки двох значень: "так" або "ні". "Так" означає, що у дане поле обов'язково мають бути введені дані, які відповідають раніше визначеним властивостям.

**Порожні строки.** Набувають двох значень: "так" або "ні". "Так" означає, що порожні рядки дозволені

**Індексоване поле.** Набуває двох значень: "так" або "ні". "Так" означає, що поле індексоване.

5. Перелічіть значення які може набирати властивість Розмір поля числових полів.

**Байт** - цілі числа у межах від 0 до 255.

**Ціле** - цілі числа від - 32 768 до 32 767.

**Довге ціле** - цілі числа від -2 147 483 648 до 2 147 483 647.

**З плаваючою комою (4 байта)** - числа у межах від -3.402823 E38 до 3.402823 E38.

**З плаваючою комою (8 байт)** - числа у межах від - 1.79769313486 E308 до 1.79769313486 E308.

**Код реплікації** – глобально унікальний ідентифікатор (GUID) довжиною 16 байт.

6. Перелічіть значення які може набирати властивість **Формат поля полів Дата/время**.

**Повний формат дати** (наприклад, 12.05.97 15:20:21).

**Довгий формат дати** (наприклад, 13 серпня 1997 р.).

**Середній формат дати** (наприклад, 13 серп. 97 р.).

**Короткий формат дати** (наприклад, 13.08.97).

**Довгий формат часу** (наприклад, 15:20:21).

**Середній формат часу** (наприклад, 03:21 PM).

**Короткий формат часу** (наприклад, 15:20).

7. Перелічіть способи створення таблиць.

Режим таблиці, Конструктор, Імпорт таблиць

8. Що таке ключове поле.

Ключове поле - це стовпець таблиці, що дозволяє встановити зв'язок із записами в іншій таблиці. Значення ключового поля однозначно визначає кожен запис у таблиці.

9. Перелічіть типи ключових полів.

Поле лічильника, Простий ключ, Складений ключ.

10. Які операції можна виконувати над даними в таблицях?

Можна додавати інформацію, видаляти, змінювати. Якщо потрібно змінювати структуру таблиці, можна змінювати імена полів та типи даних, додавати нові поля та ін., сортувати, виконувати пошук і т.д.

11. Перелічіть об'єкти в таблицях які можна редагувати за допомогою контекстних меню.

Запис, поле, значення поля, таблиця.

12. В яких режимах можна працювати з таблицями?

Режим таблиці, Режим конструктора.

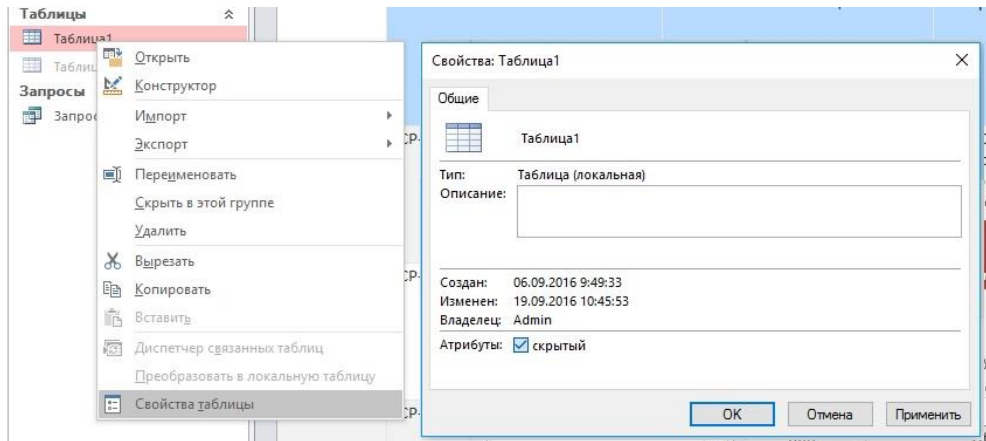
13. Що таке фільтр?

Фільтр – інструмент для відображення лише необхідних даних.

14. В результаті фільтрації і сортування що змінюється, а що залишається незмінним?

При фільтрації в таблиці залишається лише та інформація, яка задовольняє умови фільтрації. При сортуванні, інформація, що знаходиться в таблиці «вибудовується» у певним чином, наприклад «за алфавітом від А до Я».

15-16. За допомогою якої команди можна приховати (відновити) потрібні стовпці вашої таблиці?



ПКМ>Таблица 1>Властивості таблиці>«Атрибути: скритий» >ОК

### Висновок:

У процесі роботи я ознайомився на практиці з типами та властивостями полів в Access. Навчився створювати таблиці різними способами. Навчився працювати з даними у таблицях БД, редагувати їх, виконувати пошук, фільтр та оптимально використовувати можливості СКБД.