

ФАКУЛЬТЕТ АВТОМАТИКИ, ТЕЛЕМЕХАНІКИ ТА ЗВ'ЯЗКУ

Кафедра спеціалізованих комп'ютерних систем

МЕТОДИЧНІ ВКАЗІВКИ

**до лабораторних робіт
з дисциплін**

***«ОРГАНІЗАЦІЯ БАЗ ДАНИХ»,
«АВТОМАТИЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ»,
«ОПЕРАЦІЙНІ СИСТЕМИ ТА БАЗИ ДАНИХ»***

Харків – 2015

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри спеціалізованих комп'ютерних систем 22 грудня 2014 р., протокол № 7.

У методичних вказівках пояснюється розв'язання типових задач, що розглядаються на лабораторних заняттях з навчальних дисциплін «Організація баз даних», «Автоматизація бізнес-процесів» та «Операційні системи та бази даних». Також наведені варіанти завдань для самостійного рішення. Рекомендуються для студентів усіх форм навчання факультету АТЗ з напрямів підготовки «Автоматизація та автоматизація на транспорті» та «Спеціалізовані комп'ютерні системи» і слухачів ННППК.

Укладачі:

доц. О.Ю. Гуль,
асистенти Т.О. Чуян,
М.С. Курцев

Рецензент

доц. К.А. Грубчанінова

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт
з дисциплін

*«ОРГАНІЗАЦІЯ БАЗ ДАНИХ»,
«АВТОМАТИЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ»,
«ОПЕРАЦІЙНІ СИСТЕМИ ТА БАЗИ ДАНИХ»*

Відповідальний за випуск Чуян Т.О.

Редактор Еткало О.О.

Підписано до друку 23.02.15 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 4,0. Тираж 100. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

ЗМІСТ

Вступ.....	4
1 Введення в бази даних (БД).....	4
1.1 Основні поняття БД.....	4
1.2 Обробка інформації за допомогою СУБД.....	6
1.3 Реляційна модель БД.....	7
1.4 Особливості СУБД Access.....	9
1.5 Структура СУБД Access.....	10
2 Створення бази даних і основні операції з таблицями...	11
2.1 Створення БД.....	11
2.2 Створення таблиць.....	13
2.3 Робота з полями.....	23
2.4 Зв'язки між таблицями.....	28
2.5 Використання індексів.....	35
2.6 Операції з таблицями.....	38
2.7 Стиск БД.....	40
3 Обробка даних у таблицях.....	41
3.1 Можливості обробки даних.....	41
3.2 Форматування таблиці.....	52
3.3 Використання підтаблиць.....	55
3.4 Застосування підстановки.....	58
Завдання на виконання лабораторних робіт.....	62
Список літератури.....	66

ВСТУП

У методичних вказівках розглядається розв'язання типових задач автоматизації, що вирішуються на лабораторних заняттях та в години самостійної роботи з дисциплін «Організація баз даних», «Автоматизація бізнес-процесів» та «Операційні системи та бази даних». Для кожної задачі додаються варіанти завдань, які повинні бути вирішені студентами в години самостійної роботи.

Задачі автоматизації, що наведені в методичних вказівках, також можуть бути використані при проведенні поточного контролю знань студентів, модульного контролю та на екзамені.

1 Введення в бази даних (БД)

1.1 Основні поняття БД

У наш час життя людини настільки насичене різною інформацією, що для її обробки необхідне створення величезної кількості сховищ і банків даних різного призначення. Зараз практично будь-яке завдання пов'язано з маніпулюванням інформацією та даними. З цієї причини останніми роками з'явилася безліч різних комп'ютерних систем, які називаються системами управління базами даних і призначені саме для цього.

По суті, БД – це деяка подоба електронної картотеки, електронного сховища даних, що зберігається в комп'ютері у вигляді одного або декількох файлів. При цьому виникає необхідність у виконанні ряду операцій з БД, зокрема:

- додавання нової інформації в існуючі файли БД;
- додавання нових порожніх файлів у БД;
- зміна (модифікація) інформації в існуючих файлах БД;
- пошук інформації в БД;
- видалення інформації з існуючих файлів БД;
- видалення файлів із БД.

Комп'ютеризована інформаційна система являє собою програмний комплекс, завдання якого полягає в підтримці надійного зберігання БД у комп'ютері, виконанні перетворень інформації і відповідних обчислень, наданні користувачам зручного і легкого в освоєнні інтерфейсу. Традиційно обсяги інформації, з якими доводиться мати справу таким системам, досить великі, а самі БД мають досить складну структуру. Прикладами інформаційних систем є системи замовлення залізничних або авіаційних квитків, банківські системи та інші.

Як ілюстрацію можна розглянути невелику БД (рисунок 1), у якій зберігається інформація про стипендії студентів.

KOD	NAME	GROUP	BALL	STIP
525122	Батракова И.Н.	5-4-УПП	4.5	22.50
525123	Баулин А.В.	5-4-УПП	4.0	18.75
525124	Белянин Д.А.	5-4-УПП	5.0	25.00
525125	Бондаренко О.В.	5-4-УПП	4.5	22.50
525126	Зеленский И.В.	5-4-УПП	3.0	0.00
525127	Климова В.В.	5-4-УПП	4.5	22.50
525128	Кричун О.В.	5-4-УПП	4.0	18.75
525129	Кузнецова В.В.	5-4-УПП	4.5	22.50
525130	Нестреляй А.Н.	5-4-УПП	3.0	0.00
525131	Салова О.П.	5-4-УПП	4.5	22.50

Запис →

↑
Поле

Рисунок 1 – Приклад таблиці Стипендія

Дані про студентів записуються в таблицю, що має чітку структуру даних. Інформація всередині таблиці складається з *полів*, кожне з яких має власне ім'я (таблиця 1).

Таблиця 1– Опис полів таблиці Стипендія

Поле	Опис
KOD	номер студентського квитка
NAME	прізвище й ініціали студента
GROUP	найменування навчальної групи студента

BALL	середній бал успішності студента
------	----------------------------------

STIP	розмір нарахованої стипендії
------	------------------------------

Кожен рядок таблиці потрібно розглядати як *запис*, при цьому інформація в таблицю заноситься у відповідні поля. Усі записи складаються з однакових полів, а характер інформації для одного поля у всіх записах однаковий. У той же час різні поля мають різний характер збережених даних.

Основним призначенням БД у першу чергу є швидкий пошук інформації, що зберігається в ній. При значному розмірі БД ручний пошук, а також модифікація інформації, що в ній зберігається, забирає значний час. Використання комп'ютера для обробки БД усуває перераховані вище проблеми при пошуці і вибірці інформації. У цьому випадку її модифікація здійснюється досить швидко й ефективно, а сама БД легко розміщується на будь-якому диску і за необхідності може бути перенесена на інший комп'ютер.

1.2 Обробка інформації за допомогою СУБД

Існує велика кількість програм, які призначені для структурування інформації, розміщення її в таблицях і маніпулювання наявними даними – такі програми й одержали назву СУБД. Основна особливість СУБД – це наявність засобів для введення й зберігання не тільки самих даних, але й описів їхньої структури.

Якщо говорити більш детально, то до функцій СУБД відносять такі:

керування даними безпосередньо в БД – функція, що забезпечує зберігання даних, що безпосередньо входять у БД, і службової інформації, що забезпечує роботу СУБД;

керування даними в пам'яті комп'ютера – функція, пов'язана в першу чергу з тим, що СУБД працюють з БД великого розміру. З метою прискорення роботи СУБД використовується буферизація даних в оперативній пам'яті комп'ютера. При цьому користувач СУБД використовує тільки необхідну для його конкретного завдання частину БД, а за необхідності одержує нову "порцію" даних;

керування транзакціями – функція СУБД, що робить ряд операцій над БД як над єдиним цілим. Як правило, такі операції виконуються в пам'яті комп'ютера. У першу чергу транзакції необхідні для підтримки логічної цілісності БД у багатокористувальницьких системах. Якщо транзакція (маніпуляція над даними) успішно виконується, то СУБД вносить відповідні зміни в БД. В іншому випадку жодна зі зроблених змін ніяк не впливає на стан БД;

підтримка мов БД – для роботи з БД використовуються спеціальні мови, які мають спільну назву *мови баз даних*. У СУБД звичайно підтримується єдина мова, що містить усі необхідні засоби – від створення БД до забезпечення користувальницького інтерфейсу при роботі з даними. Найпоширенішою на цей час мовою СУБД є мова SQL (Structured Query Language).

1.3 Реляційна модель БД

Останніми роками в більшості БД використовуються реляційні моделі даних, і практично всі сучасні СУБД орієнтовані саме на таке подання інформації.

Реляційну модель можна подати як особливий метод розгляду даних, що включає як власне дані (у вигляді таблиць), так і способи роботи і маніпуляції з ними (у вигляді зв'язків). Інакше кажучи, у реляційній БД використовується кілька таблиць, між якими встановлюються зв'язки. Таким чином, інформація, введена в одну таблицю, може бути зв'язана з одним або кількома записами з іншої таблиці.

Між записами двох таблиць (наприклад, таблиць А і В) можуть існувати такі види зв'язку:

- “один до одного” (кожному запису з А відповідає один певний запис із В, наприклад, у кожного студента є лише одна залікова книжка);
- “один до багатьох” (кожному запису з А відповідає кілька записів з В, наприклад, у будинку проживає багато мешканців);
- “багато до одного” (безлічі записів з А відповідає один певний запис із В, наприклад, декілька студентів навчаються в одній групі);
- “багато до багатьох ” (безлічі записів з А відповідає безліч записів з В, наприклад, у різних студентів заняття ведуть різні викладачі).

Важливим поняттям у теорії реляційних БД є *нормалізація*, принципи якої можна сформулювати у вигляді нижченаведених основних правил, що використовуються при розробленні структури БД.

1 У кожному полі таблиці повинен перебувати унікальний вид інформації, тобто в одній і тій же таблиці не повинні перебувати поля, що повторюються.

2 У кожній таблиці повинен бути *первинний ключ* або унікальний ідентифікатор, що однозначно визначає даний запис серед безлічі записів таблиці.

3 Кожному значенню первинного ключа повинна відповідати вичерпна інформація про об'єкт таблиці (зокрема, об'єктом таблиці може бути студент, повна інформація про якого подана в таблиці БД Стипендія).

4 Зміна значення кожного поля таблиці, що не входить до складу первинного ключа, не повинна впливати на інформацію в інших її полях.

1.4 Особливості СУБД Access

Додаток Access є реляційною СУБД, яка підтримує всі засоби і можливості з обробки даних, властиві реляційним моделям. При цьому інформація, яку необхідно зберігати у відповідних БД, може бути подана в практично будь-якому форматі, зокрема, текстовому, графічному, числовому, грошовому, дата або час тощо.

Серед засобів, які пропонує СУБД Access, не можна не відзначити можливість динамічного обміну даними (DDE) між Access та іншими додатками, які підтримують цю технологію. Також є можливість застосування технології ActiveX, що дає змогу використовувати розробнику у своєму програмному продукті не тільки ті об'єкти, які властиві даному додатку (зокрема, Access), але й об'єкти інших додатків (наприклад, Excel або Word).

Досить зручною можливістю є той факт, що користувач при обробці даних може працювати не тільки з БД оброблюваного в Access формату, але й експортувати дані інших СУБД, що мають зовсім інший формат подання, зокрема, формат FoxPro, Paradox та ін.

Крім цього, за допомогою Access користувач може обробляти БД, що підтримують відкритий доступ до даних (стандарт ODBC), зокрема, досить популярних останнім часом серверів баз даних Oracle та SQL Server.

При обробці даних в Access використовується структурована мова запитів SQL, яку без перебільшення можна назвати стандартною мовою БД. З її допомогою можна виконувати найрізноманітнішу обробку наявних даних, зокрема, створювати вибірки необхідної структури, вносити необхідні зміни в наявні БД, перетворювати або видаляти таблиці, формувати дані для звітів і багато чого іншого.

Важливою перевагою СУБД Access є те, що з її допомогою можна розробляти системи, які обробляють БД як на окремому

комп'ютері, так і у локальній мережі підприємства або в Інтернеті, використовуючи режим обробки даних “клієнт-сервер”.

Необхідно також відзначити, що Access надає широкі можливості зі створення додатків, пов'язаних з обробкою БД. При цьому розробнику не обов'язково бути програмістом високого класу, а цілком достатньо мати уявлення про створення подібних додатків у середовищі Windows, а також мати деякі навички програмування мовою Visual Basic. У цьому випадку розробник досить швидко зможе опанувати навички зі створення додатків в Access, що дасть змогу виконувати автоматизацію як простих, так і досить складних завдань, пов'язаних з обробкою даних.

1.5 Структура СУБД Access

СУБД Access при обробці інформації розглядає БД як набір декількох структурних елементів, кожний з яких може включати один або кілька об'єктів. Серед основних складових БД, з погляду Access, можна виділити нижченаведені об'єкти:

1 *Таблиці*. Являють собою об'єкти, які створюються користувачем для зберігання інформації про предмети або суб'єкти у певній структурі. Будь-яка таблиця складається з *полів* (стовпчиків) і *записів* (рядків) (рисунок 1).

2 *Запити*. Є об'єктами, які призначені для одержання необхідних даних з наявних у БД таблиць. Як правило, при створенні запитів використовується мова SQL. За допомогою запитів можна створювати вибірки даних, додавати або видаляти інформацію в певній таблиці. Крім цього, за допомогою запиту можливе також створення нових таблиць на підставі однієї або декількох наявних у БД таблиць.

3 *Форми*. Являють собою об'єкти, які використовуються для розроблення інтерфейсу, за допомогою якого відбувається введення даних користувачем, а також відображення наявної в БД інформації на екрані. Крім цього, форми застосовуються для керування розробленим додатком, наприклад, для виконання якоїсь дії при виникненні певної події.

4 *Звіти*. Є об'єктами, які використовуються для підведення будь-яких підсумків на підставі наявних даних, і виводу цих підсумків у певному форматі на друк.

5 *Сторінки*. Являють собою об'єкти, які забезпечують доступ до інформації, наявної в БД, з мережі Internet за допомогою браузера. Кожна сторінка, як правило, являє собою HTML-файл, за допомогою якого користувачі Internet одержують доступ до наявної БД.

6 *Макроси*. Є об'єктами, призначеними для виконання певних дій при виникненні тієї або іншої події. Наприклад, за допомогою макросів можна створювати запити, формувати звіти, відкривати таблиці, обробляти форми тощо.

7 *Модулі*. Являють собою об'єкти, які містять одну або кілька процедур, написаних мовою Visual Basic, у системі розроблення додатків Visual Basic for Applications. За допомогою модулів можна реалізовувати широкий спектр можливостей з обробки даних, які недоступні розробнику у випадку використання макросів.

2 Створення бази даних і основні операції з таблицями

2.1 Створення БД

Після того, як додаток Access запущений, на екрані з'явиться діалогове вікно початку роботи з даною СУБД (рисунок 2). За його допомогою можна створити нову БД (*Нова база даних*) або відкрити вже наявну, яка недавно використовувалася (*Відкрити базу даних*). Якщо необхідної БД немає в списку, то необхідно викликати діалогове вікно *Відкриття файлу бази даних* за допомогою команди *Інші файли.....* При цьому необхідно враховувати, що файли БД мають розширення *.mdb.

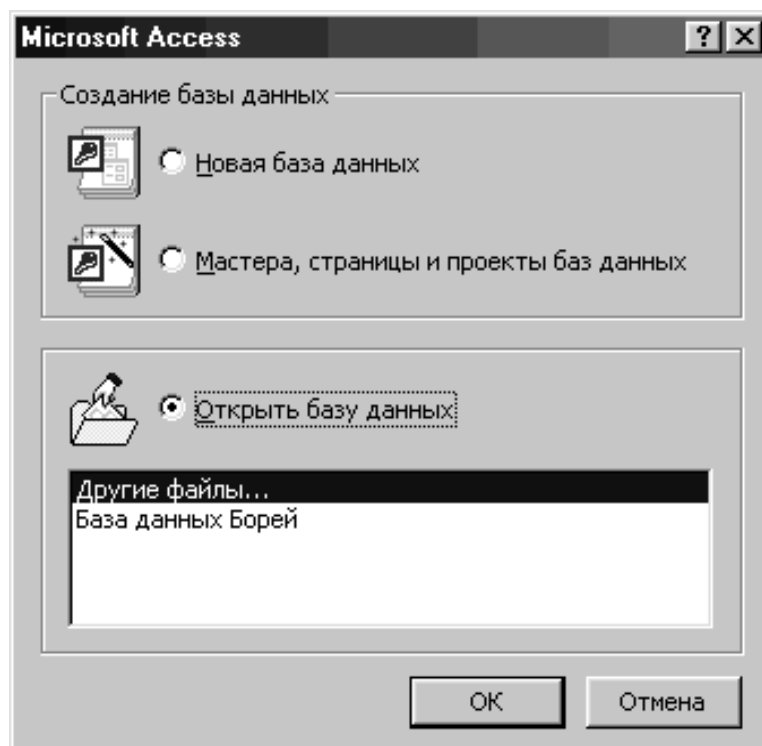


Рисунок 2 – Початок роботи з Access

При виборі команди *Нова база даних* на екрані буде відображено діалогове вікно створення нової БД, у якому необхідно в полі *Ім'я файлу* вказати ім'я БД, що створюється. Як приклад буде розглянуто створення навчальної БД *Освіта.mdb*, що містить інформацію про студентів, предмети, які вони вивчають, викладачів, а також успішність студентів.

Після вводу імені БД необхідно натиснути кнопку *Створити*, після чого на екрані з'явиться вікно створеної БД *Освіта* (рисунок 3), яке необхідно при роботі з будь-якою БД. Щоб з його допомогою створити новий об'єкт або змінити вже існуючий, наприклад таблицю, необхідно натиснути кнопку *Таблиці* в списку *Об'єкти*. Після цього в правій частині вікна будуть відображені всі можливі режими створення обраного об'єкта (у цьому випадку таблиці), а також список усіх наявних у цій БД таблиць. Крім цього, у верхній частині вікна є кілька кнопок для роботи з обраним об'єктом (таблиця 2).

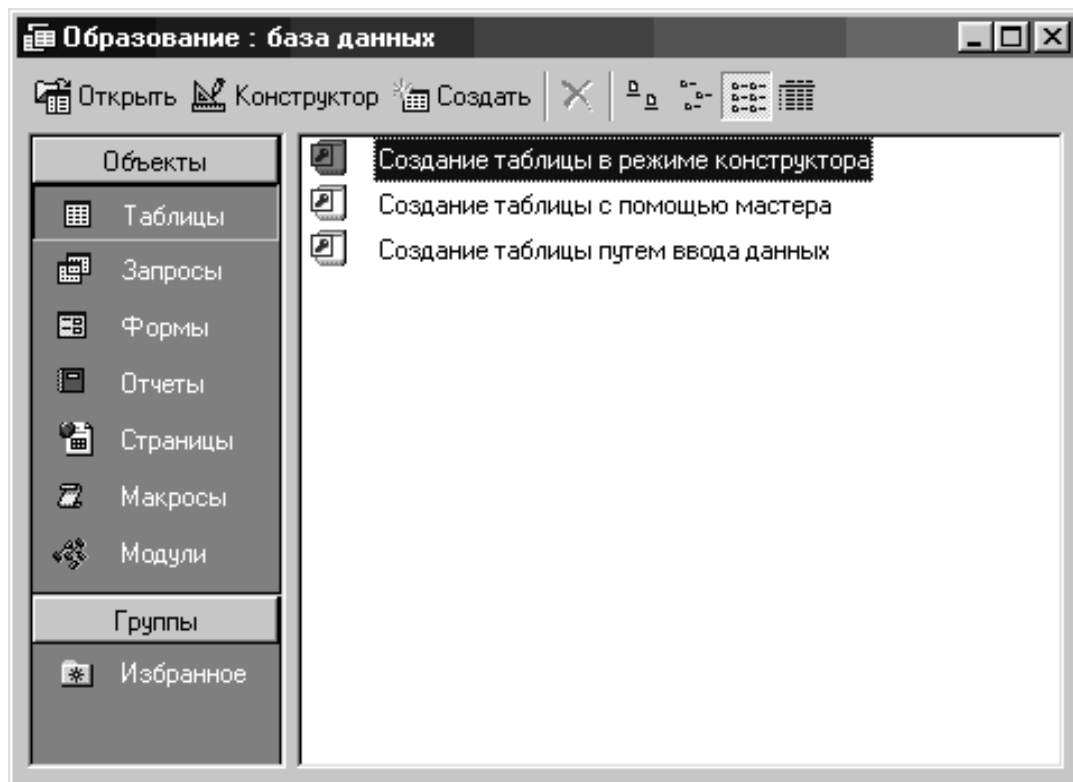

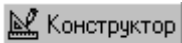




Рисунок 3 – Нова БД Освіта

Таблиця 2– Кнопки роботи з таблицею

Кнопка	Опис
	відкрити таблицю в <i>режимі таблиці</i>
	відкрити таблицю в <i>режимі конструктора</i>
	створити нову таблицю

2.2 Створення таблиць

Щоб створити нову таблицю, необхідно у вікні відкритої БД вибрати у списку *Об'єкти* кнопку *Таблиці* і скористатися кнопкою  (рисунок 3). Після цього відобразиться діалогове вікно

Нова таблиця (рисунок 4), у якому необхідно вибрати один із запропонованих режимів створення.

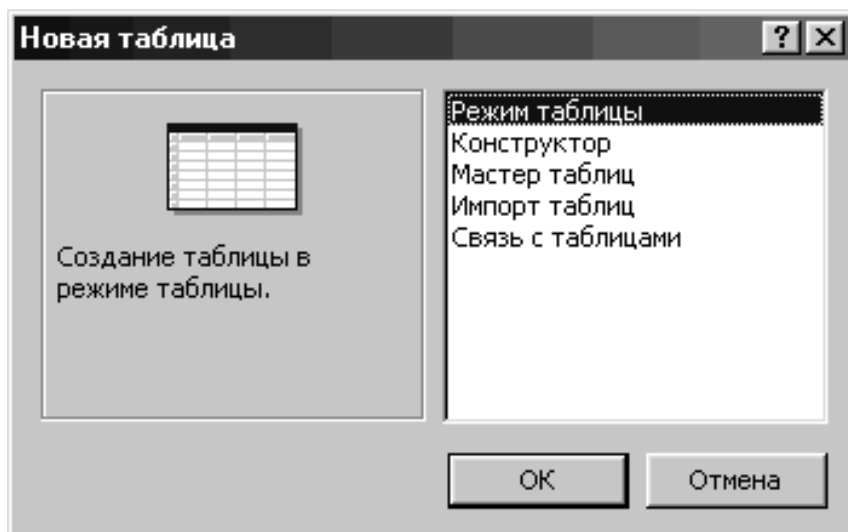


Рисунок 4 – Діалогове вікно Нова таблиця

Ще один варіант створення таблиці полягає у виборі бажаного варіанта створення таблиці (Створення таблиці в режимі конструктора, Створення таблиці з використанням майстра або Створення таблиці шляхом безпосереднього вводу даних).

Режим таблиці. Дає змогу вводити дані безпосередньо в таблицю (зручно використати для редагування наявної таблиці).

Майстер таблиць. Дає змогу створити нову таблицю, використовуючи наявні шаблони таблиць. При цьому у створювану таблицю можна додавати поля з різних шаблонів.

Конструктор. Надає можливість перед введенням даних самостійно визначити структуру нової таблиці, тобто які саме поля в ній будуть і які типи даних будуть у цих полях використовуватися.

Імпорт таблиць. Застосовується для створення нової таблиці БД на підставі якого-небудь файла даних, наявного на диску, причому, можливо, навіть в іншому форматі (не Access).

Зв'язок з таблицями. Дає змогу створити в БД таблицю, що буде пов'язана з таблицею з іншого файла даних, інакше кажучи, зовнішню таблицю. Дуже зручно використовувати цей режим для підключення до будь-якого стороннього сервера БД, наприклад, Microsoft SQL Server.

Далі будуть розглянуті перші три режими створення нових таблиць у БД.

Режим таблиці. У випадку вибору даного режиму буде відображене вікно з новою таблицею, що має стандартне ім'я Таблиця1 (рисунок 5). Назви стовпчиків також мають стандартні назви: Поле1, Поле2 і т.д.







	Поле1	Поле2	Поле3
▶			



Рисунок 5 – Частина вікна Таблиця1

При заповненні нового запису в таблиці дані необхідно вводити в комірки, які розміщені на перетинанні відповідного рядка і стовпчика. Область таблиці, розташована ліворуч від чарунок, називається *областю виділення рядка*, а область над чарунками – *областю виділення стовпчика*. При переміщенні покажчика миші в одну з цих областей він набуде вигляду чорної стрілки, і натискання лівою кнопкою миші призведе до виділення відповідного рядка або стовпчика. З іншого боку, натискання в цьому ж положенні правої кнопки миші призведе до відображення списку можливих дій, які можна виконати для виділеного рядка (стовпчика).

В області виділення рядка може міститися один з нижченаведених символів, що наочно відображає поточний стан поля таблиці, на якому у цей момент перебуває курсор (таблиця 3).

Таблиця 3 – Кнопки в області виділення рядка

Кнопка	Опис
	поточний запис, уже збережений
	новий запис, у який можна вводити дані
	запис, що змінюється, ще не збережений
	запис у цей момент заблокований

Після того, як у будь-яке поле порожнього запису введений перший символ, в області виділення рядка замість зображення поточного збереженого запису  з'явиться зображення запису, що редагується . Після закінчення введення значення в поточне поле запису можна скористатися клавішами **Tab**, **Enter** або клавішею управління курсором → для переходу до наступного поля. Інакше кажучи, введення і редагування даних у комірках таблиці Access виконується так само, як і в Excel. Після завершення вводу в останнє поле поточного запису (рядка) виконується автоматичний перехід до першого поля наступного запису (наступного рядка).

Щоб змінити назву стовпчика в таблиці, необхідно виділити необхідний стовпчик, натиснувши ліву кнопку миші на його області виділення, і вибрати команду *Перейменувати стовпчик* у меню *Формат* (або в контекстному меню для даного стовпчика). Після цього в області виділення стовпчика можна задати для нього нове ім'я. Слід зазначити, що в попередніх версіях Access при зміні назви поля в одній з таблиць БД доводилося змінювати

вручну відповідні назви, пов'язані з цим полем, в інших таблицях, запитах, звітах, формах і т.д. У нових версіях Access існує налаштування, що дає змогу виконувати подібні зміни автоматично. Щоб увімкнути режим автоматичної заміни імен для полів таблиць, необхідно виконати команду *Сервіс | Параметри*, після чого на екрані відобразиться діалогове вікно *Параметри*. Потім необхідно вибрати вкладку *Загальні* (рисунок 6). На режим автоматичної заміни імен впливають прапорці категорії *Автозаміна імен*:

відслідковувати автозаміну імен – зміни відслідковуються, фіксуються, але при цьому заміна імен не виконується автоматично для всіх об'єктів БД – таблиць, запитів, форм і т.д.;

виконувати автозаміну імен – при зміні імені поля в таблиці для всіх інших об'єктів БД виконується автоматична заміна цього імені;

журнал змін автозаміни імен – у таблиці “Журнал автозаміни” будуть фіксуватися всі зміни в БД при виконанні автоматичної заміни імен.

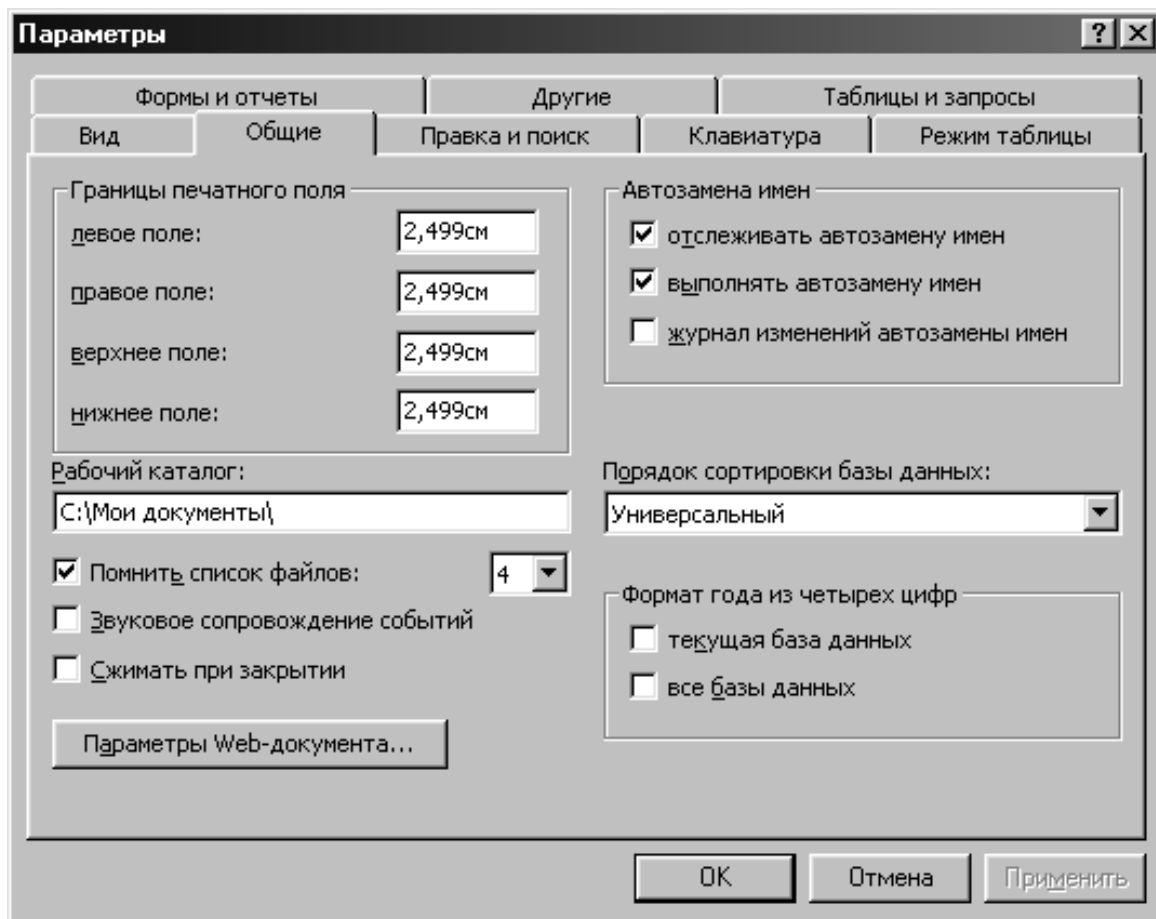


Рисунок 6 – Діалогове вікно Параметри. Вкладка Загальні

Після встановлення необхідних прапорців потрібно натиснути кнопку *ОК* (як правило, перші два прапорці в категорії *Автозаміна імен* установлені за замовчуванням). При цьому необхідно мати на увазі, що параметри, які встановлюються для автоматичної заміни імен, впливають тільки на поточну відкриту БД. Отже, при роботі з кожною БД необхідно перевіряти, чи встановлені для неї параметри автоматичної заміни.






Щоб додати в таблицю новий стовпчик, необхідно спочатку виділити той стовпчик, перед яким необхідно здійснити вставку, після чого виконати команду *Стовпчик* меню *Вставка* (або в контекстному меню для даного стовпчика вибрати *Додати стовпчик*).

Для видалення стовпчика треба спочатку його виділити, після цього в меню *Виправлення* або в контекстному меню для даного стовпчика виконати команду *Видалити стовпчик* і у вікні попередження, що з'явилося, вибрати кнопку *Так*.

За необхідності зміни порядку проходження стовпчиків у таблиці можна виконати перетягування якого-небудь стовпчика в інше місце таблиці. Для цього необхідно спочатку виділити стовпчик або групу стовпчиків, які необхідно перемістити, після цього встановити курсор на виділеному заголовку і перетягнути стовпчики на нове місце в таблиці, утримуючи натиснутою кнопку миші.

Слід також зазначити, що в даному режимі для переходу до того або іншого запису в нижній частині вікна таблиці є кнопки керування курсором, тобто кнопки, що дають змогу встановити курсор на той або інший запис (таблиця 4).

Таблиця 4– Кнопки керування курсором у таблиці

Кнопка	Опис
	перший запис
	попередній запис
	наступний запис
	останній запис
	новий запис

Крім цього, можна також скористатися *полем номера запису*, у якому відображається порядковий номер поточного запису. Для того, щоб перейти до запису з певним порядковим номером, необхідно ввести цей номер у поле номера запису і натиснути **Enter**.

Майстер таблиць. Даний режим переважно використовується при створенні нових таблиць на підставі наявних в Access шаблонів. При виборі режиму майстра буде відображене вікно першого етапу створення таблиці (рисунок 7), у якому необхідно вибрати поля для нової таблиці на підставі запропонованих шаблонів. Наприклад, потрібно створити таблицю Студенти, у якій будуть перебувати поля, що містять код студента, його прізвище, ім'я та по батькові, дату народження, номер групи тощо. Для цього необхідно встановити категорію *Ділові* і у списку *Зразки таблиць* вибрати таблицю Студенти. У результаті цих дій у списку *Зразки полів* буде поданий набір полів, що відповідають обраному зразку таблиці.

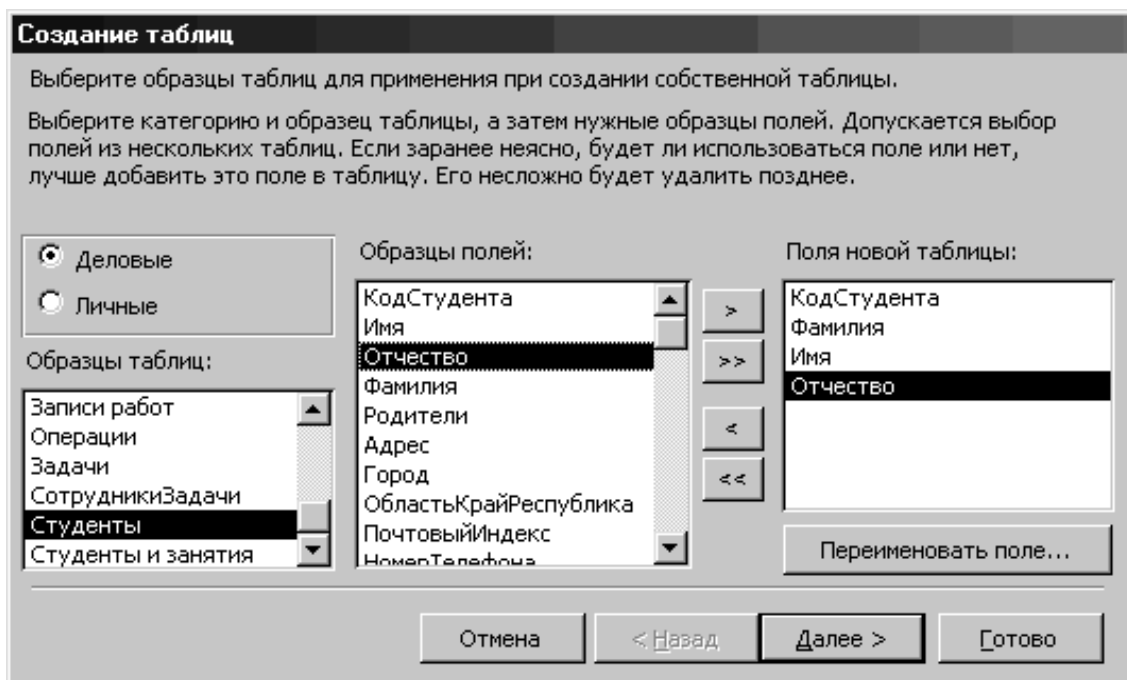







Рисунок 7 – Майстер таблиць. Створення таблиці (крок 1)

Потім потрібно зі списку *Зразки полів* послідовно вибрати поля **Код Студента**, **Прізвище**, **Ім'я**, **По батькові** і за допомогою кнопки переносу назви  додати ці поля в список *Поля нової таблиці*. Для додавання або видалення поля зі створюваної таблиці можна використовувати нижченаведені кнопки у вікні майстра (таблиця 5).

Таблиця 5 – Кнопки переносу полів у вікні майстра

Кнопка	Описание
	Перенести обране поле зразка в нову таблицю
	Перенести всі поля зразка в нову таблицю
	Видалити обране поле з нової таблиці
	Видалити всі поля з нової таблиці

Після натискання кнопки *Далі >* на екрані з'явиться вікно наступного етапу створення таблиці (рисунок 8), у якому необхідно задати ім'я нової таблиці, а також визначити ключове поле, яке необхідне для однозначної ідентифікації кожного запису в таблиці. Для цього в таблицю додано поле **Код Студента**, у якому буде зберігатися унікальний номер для кожного студента. При виборі режиму автоматичного визначення Access як ключ вибере саме це поле. Потім необхідно знову натиснути кнопку *Далі >*.

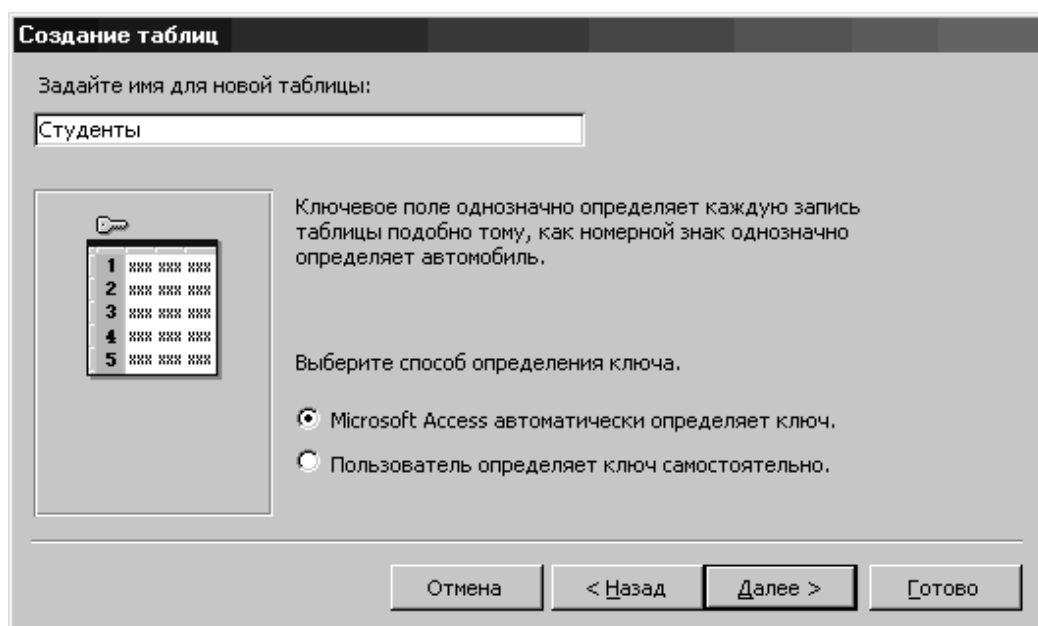


Рисунок 8 – Мастер таблиц. Створення таблиці (крок 2)

На останньому етапі роботи майстра пропонується вибрати режим роботи, який необхідно встановити після створення таблиці. У цьому випадку можна вибрати пункт *Ввести дані безпосередньо в таблицю* і натиснути кнопку *Готово*.

Після виконання зазначених вище дій буде відображено вікно таблиці Студенти в режимі редагування даних (рисунок 9).

	Код студента	Фамилия	Имя	Отчество
▶	(Счетчик)			

Рисунок 9 – Редагування таблиці Студенти

Конструктор. Даний режим застосовується для створення структури нової таблиці. При виборі режиму конструктора буде відображено вікно (рисунок 10), у якому необхідно визначити структуру нової таблиці, тобто набір полів і їх формат. Щоб описати чергове поле в структурі таблиці, необхідно спочатку вказати назву поля (*Имя поля*) і після цього визначити в списку, що розкривається, тип даних, які будуть у ньому зберігатися (*Тип даних*). Крім цього, можна також описати призначення інформації, що буде вводиться в це поле (*Опис*). Слід зазначити, що вміст параметра *Опис* буде відображатися в рядку стану вікна Access у момент редагування даного поля таблиці.

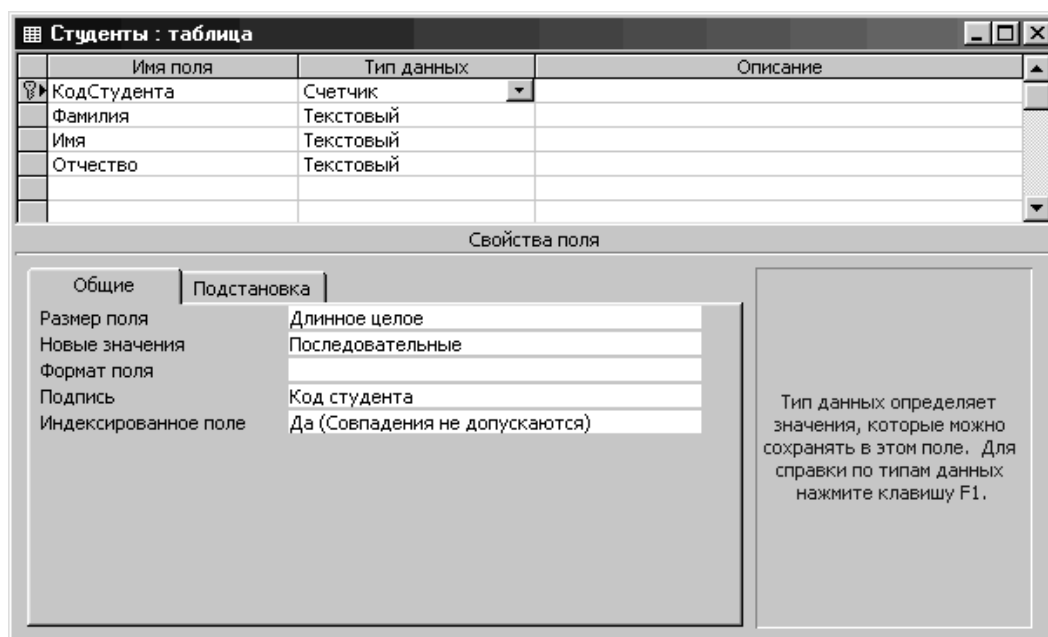





Рисунок 10 – Режим конструктора таблиці


Після того, як визначено структуру таблиці, можна перейти до безпосереднього вводу даних. Для перемикання між режимами конструктора і таблиці можна скористатися командами меню *Вид: Конструктор* і *Режим таблиці*. Крім цього, з тією ж метою зручно користуватися кнопкою *Вид* панелі інструментів (таблиця 6).

Таблиця 6– Кнопка Вид панелі інструментів

Кнопка	Опис
	перехід до режиму конструктора
	перехід до режиму таблиці

Для будь-якої таблиці в реляційній БД повинен бути заданий так званий *первинний ключ* або *ключове поле*, що дає змогу однозначно визначити той або інший запис у таблиці. Інакше кажучи, ключ необхідний для дотримання унікальності наявних у таблиці записів. Насправді таблиця може існувати і без ключового поля, однак у цьому випадку можливі проблеми з підтримкою цілісності даних.

Щоб створити в таблиці первинний ключ, необхідно відкрити таблицю в режимі *Конструктор*, потім установити курсор у поле, яке необхідно призначити ключовим, натиснувши ліву кнопку миші на області виділення рядка, що відповідає даному полю. Після цього необхідно виконати команду *Виправлення | Ключове поле* або натиснути кнопку *Ключове поле*  на панелі інструментів, що призведе до появи відповідного зображення в області виділення рядка. Щоб скасувати призначення первинного ключа для поля, необхідно ще раз виконати описані вище дії.

Для збереження змін, внесених у таблицю, необхідно виконати команду *Файл | Зберегти* або натиснути кнопку *Зберегти*  на панелі інструментів.

2.3 Робота з полями

При створенні структури таблиці необхідно визначати типи даних, які будуть у ній використовуватися. Щоб задати тип даних, які будуть зберігатися в певному полі, необхідно натиснути в режимі *Конструктора* таблиць ліву кнопку миші в стовпчику *Тип даних* відповідного рядка, після чого вибрати у списку необхідний тип. Наприклад, щоб визначити тип даних для поля **Код Студента** таблиці *Студенти*, необхідно виконати натискання в стовпчику *Тип даних* (рисунок 11), після чого скористатися стрілкою, що з'явилася, у даному рядку. Потім необхідно вибрати у списку необхідний тип даних, і задати параметри обраного типу в розділі *Властивості поля* вікна конструктора.

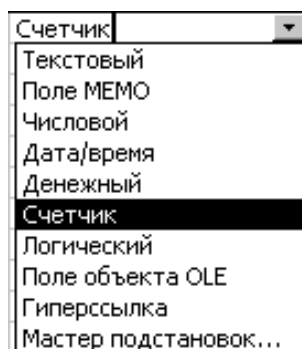


Рисунок 11 – Типи даних

У таблиці 7 наведені всі наявні в Access типи даних.

Для кожного типу даних можна встановити значення властивостей, які будуть визначати особливості обробки відповідного поля. Серед усіх властивостей можна виділити загальні, які можна задати для більшості типів (таблиця 8).

Слід також зазначити, що в Access існує можливість вводу даних у певному форматі. Для цього відповідному полю необхідно задати *маску вводу*, що визначає формат вводу інформації, що, наприклад, може обмежити діапазон можливих

символів, які можуть бути введені в поле, тільки цифрами або тільки буквами, установити автоматичне перетворення введеного тексту до нижнього або верхнього регістра (тобто змінювати всі введені букви на маленькі або на великі) тощо.

Таблиця 7 – Типи даних СУБД Access

Типу	Опис	Обсяг пам'яті
Текстовий	Текст – один рядок	До 255 байт
Поле МЕМО	Текст – кілька рядків	До 65535 байт
Числовий	Число	1, 2, 4 або 8 байт
Дата/час	Дата й/або час	8 байт
Грошовий	Грошові суми (числа із чотирма десятковими знаками)	8 байт
Лічильник	Унікальне ціле число, що позначає порядковий номер запису в таблиці	4 байт
Логічний	Логічні значення	1 біт
Поле об'єкта OLE	Об'єкти інших додатків Windows	До 1 Гбайт
Гіперпосилання	Посилання на файл (у даному комп'ютері, у локальній мережі, в Internet)	До 2048 символів
Майстер підстановок	Вибирає значення полів з іншої таблиці або запиту або ж дає змогу ввести фіксований набір варіантів	Залежить від обраного варіанта підстановок

		и
--	--	---

Таблиця 8 – Основні властивості більшості типів даних

Властивість	Опис
Підпис	Установлюється інформативна назва поля, яка автоматично буде використовуватися при створенні форм і звітів
Значення за замовчуванням	Для поля задається значення, що буде використовуватися за замовчуванням, тобто в тому випадку, якщо в дане поле не буде введена інша інформація
Умова на значення	Установлює обмеження на введені дані, інакше кажучи, не дає змоги вводити в поле такі дані, які не відповідають зазначеній умові
Повідомлення про помилку	Задає текст повідомлення, що буде відобразитися в тому випадку, якщо дані, введені в поле, не відповідають типу даних або обмеженням, зазначеним у властивостях <i>Умова на значення</i> , <i>Обов'язкове поле</i> , <i>індексоване поле</i> , тощо
Обов'язкове поле	Визначає режим обов'язкового вводу інформації в дане поле
Індексоване поле	Установлює режим використання індексу для даного поля, що дає змогу прискорити процедуру доступу до інформації в поле, а також задати режим, при якому в поле не можна вводити значення, які повторюються

Нижче наведено опис основних типів даних, які використовуються у таблицях СУБД Access.

Текстовий. Використовується для зберігання строкових даних з довжиною до 255 символів. Властивість *Розмір* дає змогу задати максимальну кількість символів (до 255), які будуть утримуватися в рядку (за замовчуванням встановлюється *Розмір* = 50).

Поле МЕМО. Застосовується для зберігання тексту, що перевищує 255 символів і/або складається з декількох рядків (тобто тексту, у якому зустрічається символ *повернення каретки*).

Основні властивості цих типів даних, які встановлюються в нижній частині вікна Конструктора наведено на рисунку 12.

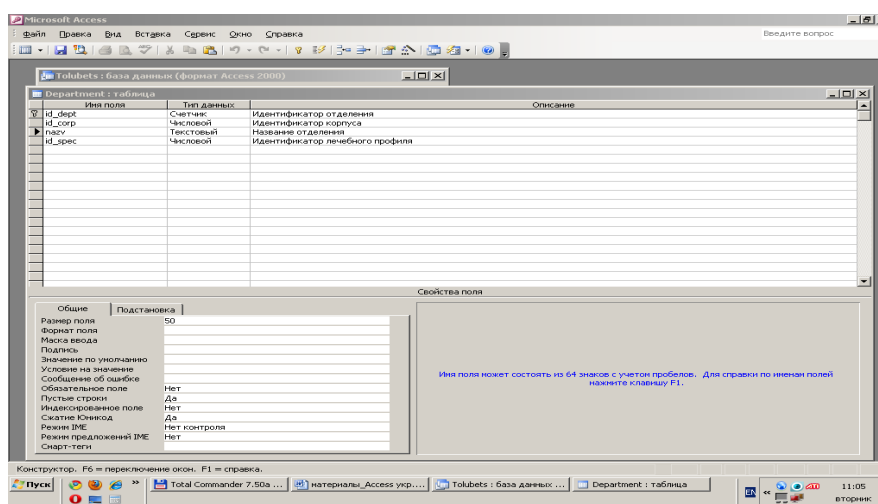


Рисунок 12 – Закладка Властивості поля

Формат поля – задається спеціальний формат, у якому дані будуть подані на екрані, наприклад, текст буде виведений із заданими кольорами (наприклад, синім) і вирівняний по лівому краю.

Порожні рядки – встановлюється режим, при якому можна вводити в поле порожній рядок (тобто значення “”).

Порожнім значенням в Access називається спеціальне значення Null, що має ряд особливостей. Зокрема не можна поставити знак рівності між значенням Null і будь-яким іншим значенням, у тому числі ще одним Null. Крім цього, значення Null не обробляються при використанні підсумкових функцій, зокрема функції *Sum* (підсумовування), наприклад, при створенні звітів.

Однак полям формату *Текстовий* і *Поле МЕМО* можна присвоїти порожні рядки, тобто значення “”, які потім можна використовувати при обчисленнях.

Числовий. Використовується для обробки чисел різного розміру і формату. Розмір визначається за допомогою властивості *Розмір поля* (таблиця 9), формат числа задається властивістю *Формат поля* (таблиця 10), який також використовується для типів даних *Грошовий* і *Лічильник*. Необхідно зазначити, що для форматів *Числовий* і *Грошовий* за замовчуванням встановлюється значення 0.

Таблиця 9– Основні значення властивості *Розмір поля* типу даних *Числовий*

Значення	Зміст	Діапазон	Обсяг
Байт	Цілі невід’ємні числа	$[0; 2^8-1]$	1 байт
Ціле	Цілі числа	$[-2^{15}; 2^{15}-1]$	2 байти
Довге ціле	Цілі числа	$[-2^{31}; 2^{31}-1]$	4 байти
Одинарне з плаваючою крапкою	Дійсні числа з точністю до 7 знаків після коми	$[-2^{128}; -2^{-149}]$ і $[2^{-149}; 2^{128}-1]$	4 байти
Подвійне із плаваючою крапкою	Дійсні числа з точністю до 15 знаків після коми	$[-2^{1024}; -2^{-1074}]$ і $[2^{-1074}; 2^{1024}-1]$	8 байт
Дійсне	Дійсні числа із заданою точністю (за замовчуванням	$[-10^{28}; 10^{28}]$	12 байт

	точність = 18 знаків)		
--	-----------------------	--	--

Таблиця 10 – Властивість *Формат* поля типів даних
Числовий, Грошовий і Лічильник

Формат	Приклад
Основний	123,456
Грошовий	123,45Грн
Євро	1234,56€
Фіксований	1234,56
З роздільниками розрядів	1 234,56
Процентний	12,34%
Експонентний	12,3E+02

Дата/час. Застосовується при обробці значень календарних дат і часу. Цей тип даних дає змогу зберігати значення в декількох форматах (таблиця 11). За замовчуванням встановлюється *Повний формат дати*.

Грошовий. Використовується для зберігання грошових значень і передбачає точність до 4 знаків після десяткової коми і до 15 знаків перед комою.

Таблиця 11 – Властивість *Формат поля* типу даних *Дата/Час*

Формат	Приклад
Повний формат дати	15.11.99 14:11:30
Довгий формат дати	15 листопада 1999 р.
Середній формат дати	15-ноя-99
Короткий формат дати	15.11.99
Довгий формат часу	14:11:30
Середній формат часу	2:11
Короткий формат часу	14:11

Лічильник. Застосовується для опису поля, яке використовується як первинний ключ, і генерує для кожного нового запису в таблиці унікальне значення формату *Довге ціле*.

Логічний. Призначений для зберігання й обробки логічних значень, тобто *Неправда* або *Істина*. Цей тип даних передбачає використання одного з таких форматів, як властивість *Формат поля*: *Істина/Неправда*, *Так/Ні*, *Вкл/Викл*. (За замовчуванням встановлюється формат *Так/Ні*).

У БД Освіта, що буде використовуватися як приклад, таблиця Студенти має поля таких форматів:

Код Студента - Лічильник (первинний ключ);

Прізвище Студента - Текстовий (розмір поля = 20 символів);

Ім'я Студента - Текстовий (розмір поля = 10 символів);

По батькові Студента - Текстовий (розмір поля = 15 символів).

2.4 Зв'язки між таблицями

Структура реляційної БД завжди розробляється таким чином, щоб кожна таблиця, що в ній перебуває, не містила надлишкової інформації. Наприклад, у БД **Освіта** необхідно зберігати дані про оцінки, які одержав кожен студент з того чи іншого предмета. Як наслідок, потрібно також якимось чином зберігати і назви предметів, за які ці оцінки ставляться. Крім цього, необхідно зберігати прізвище, ім'я та по батькові викладачів із вказівкою предмета, що веде кожний з них. Якщо для цього буде використовуватися одна таблиця, то стане очевидним нераціональне використання пам'яті комп'ютера, тому що для кожного студента доведеться зберігати у відповідному записі назви предметів і прізвища викладачів, які будуть повторюватися багато разів. Тому зазначену інформацію необхідно розбити на кілька таблиць, які будуть між собою взаємозалежні, а саме:

Студенти – інформація про студента (унікальний код студента, прізвище, ім'я та по-батькові, номер групи, дата народження тощо);

Викладачі – інформація про викладача (унікальний код викладача, прізвище, ім'я та по- батькові, кафедра, вчений ступінь, наукове звання тощо);

Дисципліни – інформація про дисципліни (унікальний код дисципліни, код викладача, що веде цю дисципліну, назва дисципліни і кількість годин);

Оцінки – інформація про оцінки студентів (унікальний код для кожного запису з оцінкою, код студента, код предмета та оцінка, що отримана даним студентом за цей предмет).

Таким чином, структура таблиць, що використовуються у БД **Освіта**, буде виглядати таким чином (таблиці 12 – 15).

Таблиця 12 – Структура таблиці Студенти

Поле	Опис
Код Студента	Лічильник (первинний ключ)
Прізвище Студента	Текстовий (20 символів)
Ім'я Студента	Текстовий (10 символів)
По батькові Студента	Текстовий (15 символів)
Дата Народження Студента	Дата/час
Група Студента	Текстовий (15 символів)
...	...

Таблиця 13 – Структура таблиці Викладачі

Поле	Опис
Код Викладача	Лічильник (первинний ключ)
Прізвище Викладача	Текстовий (20 символів)

Ім'я Викладача	Текстовий (10 символів)
По батькові Викладача	Текстовий (15 символів)
Кафедра Викладача	Текстовий (10 символів)
...	...

Таблиця 14 – Структура таблиці Предмети

Поле	Опис
Код Предмета	Лічильник (первинний ключ)
Код Викладача	Числовий (довге ціле)
Назва Предмета	Текстовий (40 символів)
Години	Числовий (ціле)

Таблиця 15 – Структура таблиці **Оцінки**

Поле	Опис
Код Предмета	Числовий (довге ціле)

Код Студента	Числовий (довге ціле)
Оцінка	Числовий (байт)
Дата Складання	Дата/час (короткий формат дати)



Варто зазначити, що при визначенні імен можна використовувати назви, що складаються з декількох слів.

Для створених таблиць необхідно встановити зв'язки, щоб, наприклад, для кожного студента (запису з таблиці **Студенти**) визначити всі його оцінки (записи у таблиці **Оцінки**). Зв'язки між таблицями в БД використовуються при формуванні запитів, створенні звітів або розробленні форм. Як видно з таблиць 12 – 15, у розроблюваній структурі даних існують такі взаємозв'язки:

1) таблиця **Студенти**, поле **Код Студента** ↔ таблиця **Оцінки**, поле **Код Студента**;

2) таблиця **Викладачі**, поле **Код Викладача** ↔ таблиця **Предмети**, поле **Код Викладача**;

3) таблиця **Предмети**, поле **Код Предмета** ↔ таблиця **Оцінки**, поле **Код Предмета**.

Для встановлення зв'язків можна скористатися режимом *Схема даних*, що встановлюється командою *Сервіс | Схема даних* або кнопкою *Схема даних*  на панелі інструментів. При виконанні даної команди на екрані з'явиться порожнє вікно *Схема даних*. Для додавання таблиць у це вікно необхідно вибрати команду *Зв'язки | Додати таблицю (Вид | Відобразити таблицю)* або скористатися кнопкою *Відобразити таблицю* . Після цього на екрані з'явиться діалогове вікно *Додавання таблиці* (рисунок 13), у якому на вкладці *Таблиці* потрібно по черзі додати всі

наявні таблиці в схему даних за допомогою кнопки *Додати* або двічі натиснути на кнопку миші по кожній таблиці. Потім необхідно закрити вікно *Додавання таблиці*.

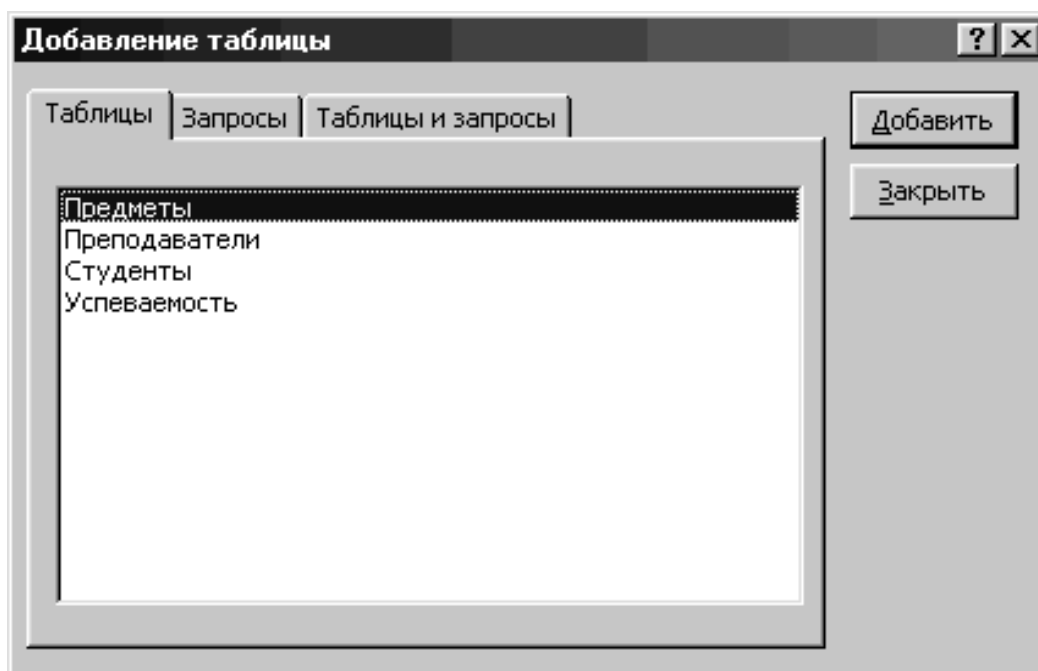


Рисунок 13 – Діалогове вікно Додавання таблиці

Після виконання зазначених дій на схемі даних будуть розташовані всі таблиці БД Освіта, розміщення яких можна змінити за своїм розсудом (рисунок 14).

Наступним етапом у розробленні схеми даних буде встановлення зв'язків між таблицями. Зокрема необхідно зв'язати таблиці Студенти і Оцінки за полем **Код Студента** таким чином, щоб кожному запису з таблиці Студенти відповідало декілька записів таблиці Оцінки (тому що в кожного студента може бути кілька оцінок). Інакше кажучи, між зазначеними таблицями потрібно встановити зв'язок виду “один-до-багатьох”. Аналогічно потрібно встановити зв'язок “один-до-багатьох” за полем **Код Предмета** між таблицями Предмети і Оцінки (тому що по кожному предмету може бути кілька оцінок, виставлених різним студентам), а також зв'язок “один до багатьох” за полем **Код Викладача** між таблицями Предмети і Викладачі (виходячи із припущення, що кожен викладач веде заняття не тільки з одного предмета).

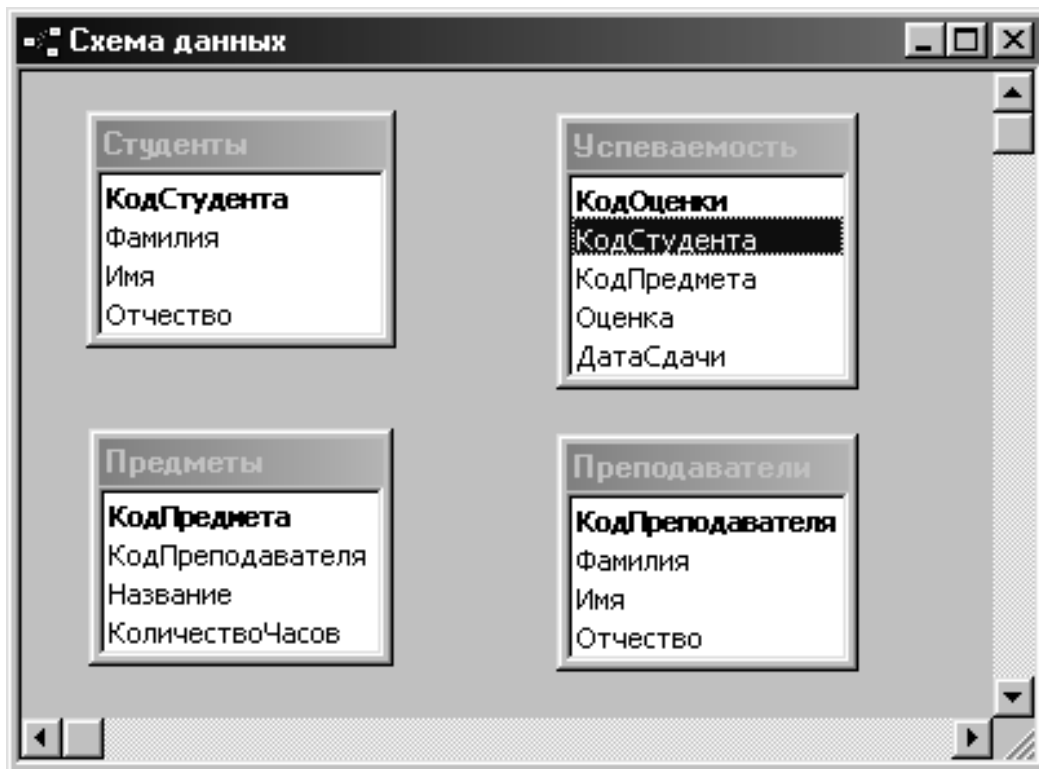


Рисунок 14 – Таблицы БД Освіта на схемі даних

Щоб установити зв'язок, наприклад, по полю **Код Студента** між таблицями Студенти і Оцінки, необхідно виділити у вікні *Схема даних* поле **Код Студента** в таблиці Студенти, за допомогою миші “перетягнути” його зображення на поле **Код Студента** таблиці Оцінки, і потім відпустити кнопку миші. При цьому з'явиться діалогове вікно *Зміна зв'язків*, у якому для обох таблиць повинне бути зазначене поле **Код Студента** (рисунок 15).

The dialog box "Изменение связей" (Change Relationships) contains the following information:

- Таблица/запрос:** Студенты
- Связанная таблица/запрос:** Успеваемость
- Field 1:** КодСтудента
- Field 2:** КодСтудента
- Обеспечение целостности данных
- каскадное обновление связанных полей
- каскадное удаление связанных записей
- Тип отношения:** один-ко-многим

Рисунок 15 – Діалогове вікно Зміна зв'язків

Якщо кнопка миші була відпущена в той момент, коли зображення перебувало над іншим полем, то в стовпчику *Зв'язана таблиця/запит* (таблиця Оцінки) буде відображена назва цього поля. У такому разі для зміни поля, за яким необхідно зв'язати таблиці, можна вибрати назву поля **Код Студента** для таблиці Оцінки у списку полів, що розкривається, яке можна відобразити, попередньо натиснувши в *поле Зв'язана таблиця/запит вікна Зміна зв'язків*. Крім цього, можна закрити вікно *Зміна зв'язків*, і знову виконати “перетягування” поля.

Для того, щоб у таблиці Оцінки не зберігалися оцінки неіснуючих студентів (тобто запису, для яких значення поля **Код Студента** не збігається з жодним зі значень поля **Код Студента** в таблиці Студенти), треба у вікні *Зміна зв'язків* установити прапорець *Забезпечення цілісності даних*. Після цього в таблицю Оцінки неможливо буде ввести запис із таким значенням поля **Код Студента**, для якого не можна буде знайти відповідний запис у таблиці Студенти.

Якщо встановлено прапорець *Забезпечення цілісності даних*, то у вікні *Зміна зв'язків* автоматично стають доступні ще два прапорці:

каскадне відновлення зв'язаних полів. При зміні значення ключового поля в “головній” таблиці Студенти автоматично змінюються значення зв'язаного поля для відповідних записів у “підлеглій” таблиці Оцінки. У розглянутому випадку ключове поле **Код Студента** є лічильником, отже, значення цього поля не можуть бути змінені користувачем. Тому немає необхідності встановлювати даний прапорець. А от якщо для зв'язаного поля передбачається його подальше редагування, прапорець рекомендується встановити;

каскадне видалення зв'язаних полів. При видаленні запису з “головної” таблиці Студенти автоматично видаляються ті записи з “підлеглої” таблиці Оцінки, для яких значення поля **Код Студента** збігається із значенням у вилученому записі з таблиці Студенти. Якщо в таблиці Оцінки не потрібно зберігати

оцінки відрахованих студентів, то прапорець *каскадне видалення зв'язаних полів* необхідно встановити. Якщо ж ми передбачаємо можливість подальшого використання цих даних або ж для запобігання помилковому видаленню даних про студента, у якого є виставлені оцінки в таблиці Студенти, тоді цей прапорець встановлювати не потрібно.

Після того, як обрано поля для встановлення зв'язку між таблицями, у вікні *Зміна зв'язків* необхідно натиснути кнопку *Створити*, після чого на схемі даних у вигляді лінії буде відображений створений зв'язок.

Необхідно відзначити, що тип відносин між зв'язаними таблицями Access визначає автоматично. У цьому випадку, якщо був встановлений прапорець *Забезпечення цілісності даних* (рисунок 15), то поруч із полем **Код Студента** таблиці Студенти над лінією зв'язку відображається символ “1”, а поруч з таблицею Оцінки – символ нескінченності “∞”, тобто встановлено зв'язок типу “один до багатьох” (рисунок 16).

Щоб змінити встановлений зв'язок, необхідно виділити відповідну лінію, натиснувши на ній кнопкою миші. Після цього зв'язок можна видалити командою *Виправлення | Видалити* або за допомогою клавіші **Delete**, а також змінити, викликавши вікно *Зміна зв'язку* командою *Зв'язки | Змінити зв'язок* або подвійним натисканням лівої кнопки миші на лінії зв'язку.

Після того, як усі зв'язки між таблицями встановлені, схема даних для БД Освіта матиме вигляд як на рисунку 16.

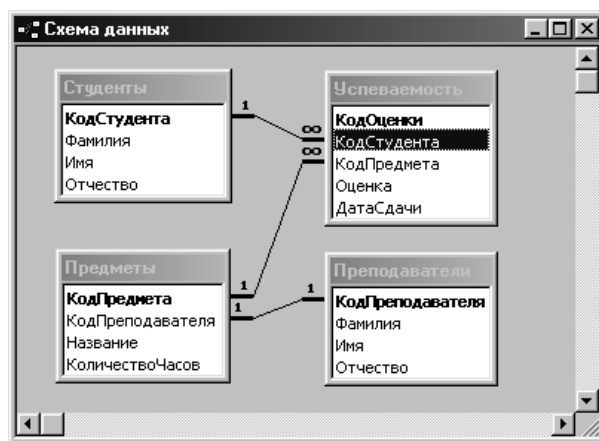






Рисунок 16 – Підсумкова схема даних БД Освіта

Отриману схему даних необхідно зберегти. Для цього потрібно закрити вікно *Схема даних* і відповісти ствердно на питання про збереження схеми даних у системному повідомленні про зміну макета.

При роботі із БД, що містять велику кількість таблиць, буває зручно відображати на схемі даних тільки певну групу зв'язаних таблиць. Наприклад, для БД Освіта необхідно на схемі даних відобразити тільки таблицю Студенти і таблицю, що пов'язана з нею, Оцінки. Для цього необхідно очистити макет схеми даних кнопкою , додати за допомогою кнопки  в схему даних таблицю Студенти і після цього відобразити на схемі всі таблиці, з якими вона зв'язана, за допомогою команди *Зв'язки | Відобразити прями зв'язки* або за допомогою кнопки *Відобразити прями зв'язки* . При цьому на макеті додається таблиця Оцінки. З іншого боку, щоб відобразити на схемі даних усі таблиці, які мають хоча б один зв'язок, можна скористатися командою *Зв'язки | Відобразити всі* або кнопкою *Відобразити всі зв'язки* . Для БД Освіта в цьому випадку на схемі даних відобразяться всі таблиці, тому що кожна з них має хоча б один зв'язок (рисунок 16).

Якщо потрібно сховати якусь таблицю на схемі даних, то для цього потрібно натиснути кнопку на таблиці і вибрати команду *Зв'язки | Сховати таблицю*.

Варто зазначити, що використання майстра підстановок, про який буде розказано далі, при виборі типу поля, дає змогу автоматично встановлювати необхідні зв'язки між таблицями.

2.5 Використання індексів

Кожен запис, що додається в таблицю, розташовується в її кінці, отже, записи в таблиці ніяк не впорядковані. Однак найчастіше буває необхідно виконувати їхнє сортування. Це можна зробити, відсортувавши всі записи таблиці, і перезаписати їх у новому порядку. Але дану операцію довелося б робити щоразу при додаванні і зміні записів у таблиці, що значно

сповільнило б обробку інформації. Тому для прискорення процесу сортування був розроблений метод, пов'язаний з перезаписом окремої таблиці, яку називають *індексом*, що складається тільки із двох полів вихідної таблиці.


Будь-який індекс являє собою автоматично створювану й обновлювану внутрішню таблицю Access, у якій є два стовпчики. У першому перебувають значення виразів, що являють собою вміст індексованого поля або декількох полів, що входять в індекс. У другому стовпчику зберігається фізичний номер кожного запису таблиці, для якої обчислено значення індексу. Таким чином, при використанні індексів записи в таблиці розглядаються не в тому порядку, у якому вони в ній зберігаються (тобто в порядку, у якому вони вводилися в таблицю), а з урахуванням їх розташування в індексі. Наприклад, для прискорення пошуку прізвища студента в таблиці Студенти можна встановити індекс для поля **Прізвище**.

Щоб створити індекс для одного поля (**Прізвище**), необхідно відкрити таблицю Студенти в режимі конструктора, вибрати поле, яке ми плануємо зробити індексованим (у нашому випадку **Прізвище**), і в розділі *Властивості поля* встановити у властивості *Індексоване поле* значення *Так*. Оскільки декілька студентів можуть мати те саме прізвище, то необхідно вибрати значення *Так (Збіги допускаються)*. З іншого боку, якщо індекс створюється для первинного ключа або будь-якого іншого унікального параметра (наприклад, Номер залікової книжки), то потрібно вибрати *Так (Збіги не допускаються)*, тому що значення первинного ключа для кожного запису є унікальним.

Після того, як буде створений індекс для поля **Прізвище**, пошук того або іншого студента за **Прізвищем** у таблиці Студенти буде виконуватися, виходячи з того міркування, що всі записи прізвищ студентів відсортовані за абеткою.

Часто для прискорення процесу пошуку того або іншого запису необхідно створювати індекс на підставі не одного, а декількох полів таблиці одночасно. У такому випадку індекс називається *складовим*. Наприклад, у таблиці Студенти необхідно

відсортувати студентів за абеткою їхніх прізвищ, імен та по батькові. Для цього потрібно створити індекс, що включає в себе поля **Прізвище**, **Ім'я** та **По батькові**.

Щоб побудувати складовий індекс, необхідно відкрити таблицю **Студенти** в режимі конструктора, і потім виконати команду *Вид | Індеси* або скористатися кнопкою *Індеси* . Після цього на екрані з'явиться вікно індесів для таблиці **Студенти – Індеси: Студенти** (рисунок 17).

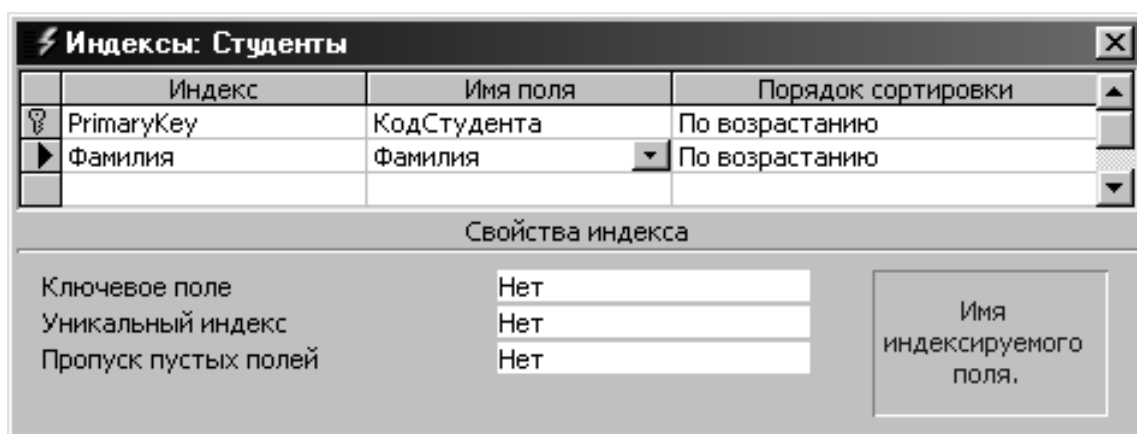


Рисунок 17 – Вікно Індеси: Студенти. Індекс Прізвище

Тут стовпчик *Індекс* містить назву індексу; у стовпчику *Ім'я поля* вибирається поле, за яким будується індекс; стовпчик *Порядок сортування* визначає напрямок сортування в індексі: за зростанням або за зменшенням.

У першому рядку розташований індекс за ключовим полем **Код Студента**, яке не можна змінювати або видаляти. У другому рядку замість індексу **Прізвище** буде створений складовий індекс **ПБ**. Для цього спочатку необхідно ввести нову назву індексу в стовпчику Індекс (**ПБ** замість **Прізвище**). Потім потрібно, залишивши назву поля **Прізвище**, у двох наступних комірках стовпчика *Ім'я поля* по черзі вибрати зі списку, що розкривається, назви полів **Ім'я** та **По батькові** (рисунок 18). Після цього можна закрити вікно індесів і зберегти зміни в структурі таблиці **Студенти**.

Щоб видалити той або інший індекс, наприклад, з індексу **ПБ** виключити поле **По батькові**, необхідно у вікні індесів

відобразити для необхідного поля (**По батькові**) правою кнопкою миші контекстне меню й вибрати в ньому команду *Видалити рядок* або виділити поле, натиснувши на його області виділення, і натиснути клавішу **Delete**. З іншого боку, аналогічно можна вставити порожній рядок для індексу. Для цього необхідно виділити рядок, перед яким потрібно виконати вставку, і натиснути клавішу **Insert**, або виконати в контекстному меню цього ж рядка команду *Додати рядок*.

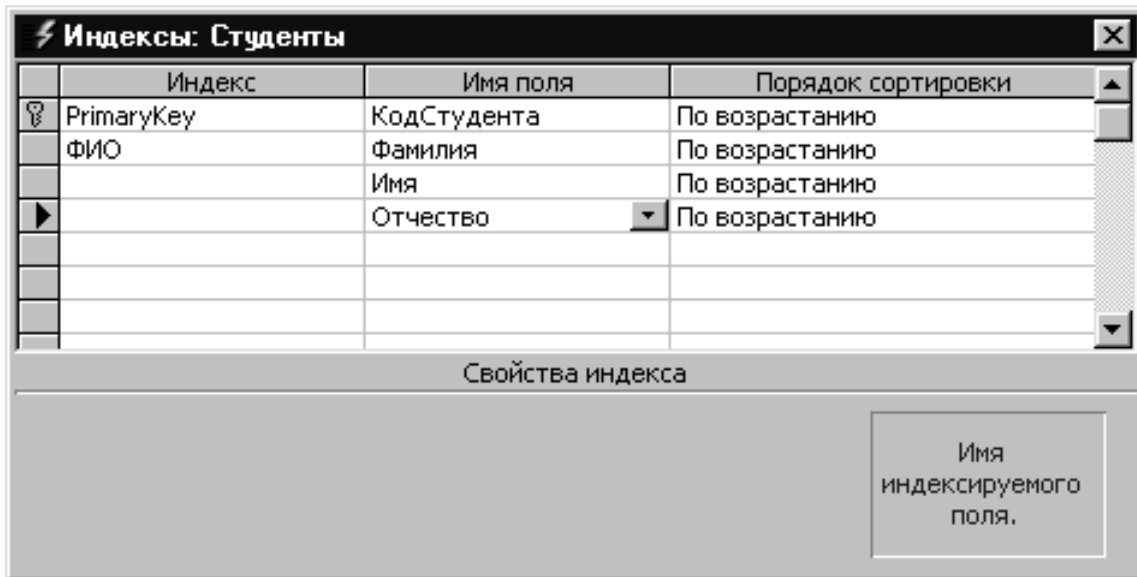


Рисунок 18 – Вікно Індекси: Студенти. Індекс ПІБ

2.6 Операції з таблицями

При обробці БД часто виникає необхідність у виконанні операцій над створеними таблицями, наприклад, перейменувати, створити копію або видалити таблицю. Нижче буде докладно розглянута реалізація цих дій.

Перейменування. Щоб перейменувати створену таблицю, необхідно виділити її у вікні БД і виконати команду *Виправлення | Перейменувати* або просто натиснути на виділеній назві. Після цього буде увімкнений режим перейменування таблиці, у якому їй можна задати нове ім'я (рисунок 19).

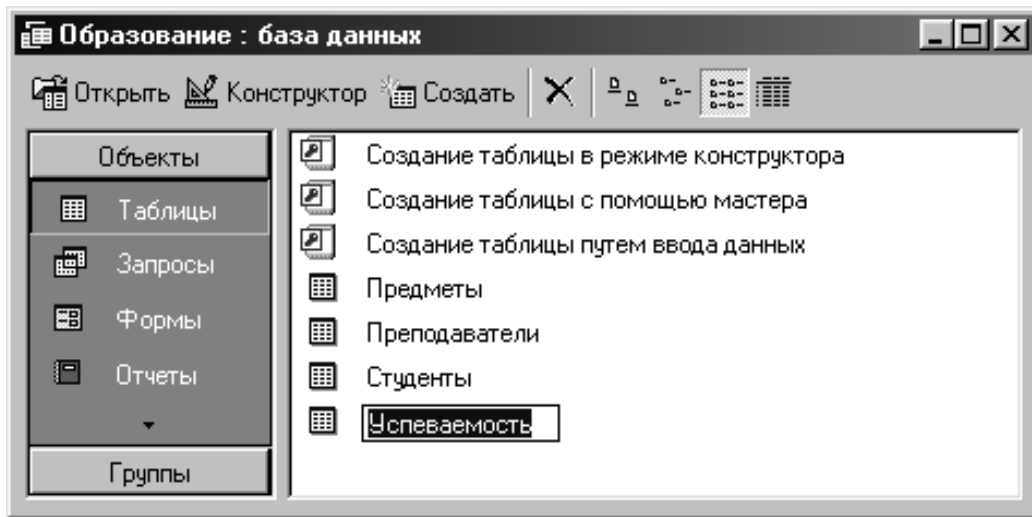




Рисунок 19 – Переименовання таблиці Оцінки у БД Освіта

Копіювання. Якщо необхідно внести важливі корективи в наявну таблицю, то можна попередньо створити її резервну копію з іншим ім'ям, і потім, якщо буде потреба, відновити цю копію під її старим ім'ям. Для цього потрібно видалити невдалий варіант старої таблиці, після чого перейменувати створену резервну копію, відновивши її початкове ім'я.

Наприклад, щоб створити копію таблиці Оцінки, необхідно виділити її у вікні БД Освіта, після чого виконати операцію копіювання в буфер кнопкою *Копіювати* , потім здійснити вставку з буфера кнопкою *Вставити* . Після виконання вставки на екрані з'явиться діалогове вікно *Вставка таблиці* (рисунок 20), у якому необхідно вказати нове ім'я таблиці (наприклад, Копія Успішність), вибрати як *Параметр вставки* пункт *структура і дані* і натиснути кнопку *ОК*. У результаті у вікні БД Освіта з'явиться нова таблиця за назвою Копія Успішність, у яку будуть скопійовані структура і дані таблиці Оцінки.

Якщо в БД уже існує таблиця із зазначеним для копії ім'ям, то в цьому випадку відобразиться вікно повідомлення, у якому буде запропоновано замінити наявну таблицю. Однак у подібній ситуації необхідно перервати операцію копіювання таблиці, тобто натиснути кнопку *Ні*, і задати у вікні *Вставка таблиці* інше ім'я для копії.

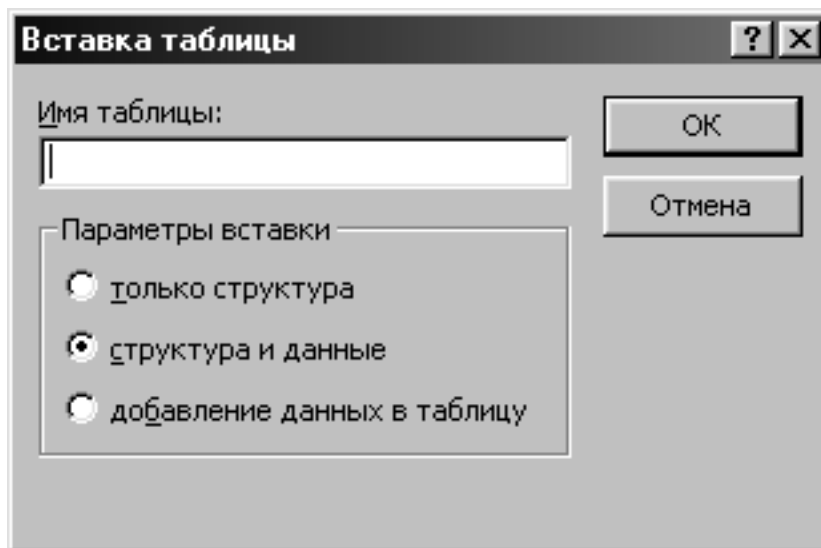


Рисунок 20 – Діалогове вікно Вставка таблиці

Видалення. Щоб видалити таблицю із БД, її спочатку необхідно виділити у вікні БД, після чого скористатися клавішею **Delete** або виконати команду *Виправлення | Видалити*. Потім з'явиться вікно попередження, у якому потрібно підтвердити видалення таблиці, вибравши кнопку *Так*. У тому випадку, якщо видаляється таблиця, яка мала зв'язки з іншими таблицями, Access відобразить ще одне повідомлення про необхідність видалення всіх зв'язків цієї таблиці. Щоб підтвердити видалення зв'язків, необхідно вибрати кнопку *Так*.

2.7 Стиск БД

При інтенсивній обробці інформації в БД, наприклад, при видаленні наявних і створенні нових об'єктів (таблиць, запитів і т.д.), файл БД рано або пізно стане зайве фрагментованим, тобто в ньому будуть виникати “порожнечі”. Оскільки наявність таких порожнеч у файлі призводить до збільшення його обсягу, то час від часу БД необхідно “стискати”, тобто видаляти з файла БД ділянки пам'яті, що не використовуються.

Щоб стиснути БД, необхідно закрити всі відкриті БД і виконати команду *Сервіс | Службові програми | Стиснути і відновити базу даних*. Після цього з'явиться діалогове вікно *База даних для стиску*, у якому необхідно вибрати БД (наприклад Освіта) і натиснути кнопку *Стиснути*.

Необхідно зазначити, що описану команду для стиску БД можна виконати й при відкритій БД. У цьому випадку операція стиску буде виконана для БД, що була відкрита.

3 Обробка даних у таблицях

3.1 Можливості обробки даних

Як приклад у цьому розділі буде розглянута навчальна БД **Освіта**, структура якої була докладно описана в попередньому розділі. Дані, які зберігаються в її таблицях, наведені в таблицях 16 - 19.

Таблиця 16 – Таблиця Студенти

Код Студента	Прізвище	Ім'я	По батькові
1	Жукова	Марина	Володимирівна
2	Морозов	Ігор	Олегович
3	Петров	Михайло	Олександрович
4	Ігнатов	Максим	Валерійович
5	Волков	Юрій	Олексійович
6	Волошина	Вікторія	Іванівна
7	Литвинов	Євгеній	Валерійович

8	Гущина	Олена	Валеріївна
9	Кузнецов	Руслан	Олексійович
10	Семененко	Світлана	Станіславівна

Таблиця 17 – Таблиця Викладачі

Код Викладача	Прізвище	Ім'я	По батькові
1	Вікуліна	Валентина	Іванівна
2	Костиркін	Олег	Володимирович
3	Казанко	Віталій	Володимирович
4	Позднякова	Любов	Олексіївна
5	Загарійчук	Ігор	Дмитрович

Таблиця 18 – Таблиця Предмети

Код Предмета	Код Викладача	Назва	Кількість Годин
1	1	Фізика	34
2	2	Хімія	68

3	3	Математика	68
4	4	Економіка	17
5	5	Філософія	17

Таблиця 19– Таблиця Оцінки

Код Оцінки	Код Студента	Код Предмета	Оцінка	Дата складання
1	1	1	5	15.01.99
2	1	2	5	20.01.99
3	2	1	4	15.01.99
4	2	2	4	20.01.99
5	3	5	3	10.02.99
6	3	4	4	25.01.00
7	3	3	5	22.01.99
8	4	1	5	15.01.99

9	4	2	4	20.01.99
---	---	---	---	----------

Ключові поля в таблицях (наприклад, Код Студента в таблиці Студенти) заповнюються автоматично, у процесі вводу нових даних.

Далі будуть розглянуті основні можливості обробки даних у режимі таблиці. При цьому як приклад буде використовуватися таблиця Студенти (рисунок 21).

Студенты : таблица					
		КодСтудента	Фамилия	Имя	Отчество
▶	+	1	Жукова	Марина	Владимировна
	+	2	Морозов	Игорь	Олегович
	+	3	Петров	Михаил	Александрович
	+	4	Игнатов	Максим	Валерьевич
	+	5	Волков	Юрий	Алексеевич
	+	6	Волошина	Виктория	Ивановна
	+	7	Литвинов	Евгений	Валерьевич
	+	8	Гущина	Елена	Валерьевна
	+	9	Кузнецов	Руслан	Алексеевич
	+	10	Семенов	Светлана	Станиславовна
*		(Счетчик)			


Рисунок 21 – Таблица Студенти

Сортування даних. Коли в Access відкривається будь-яка таблиця, то записи в ній подані в тому порядку, у якому вони були введені. Щоб змінити порядок відображення записів, можна скористатися можливостями *сортування*. Щоб відсортувати записи в таблиці за зростанням або зменшенням значень певного стовпчика, цей стовпчик спочатку необхідно виділити, потім вибрати пункт *Записи | Сортування* і вибрати одну з команд:

Сортування за зростанням (кнопка ) або

Сортування за зменшенням (кнопка )

При цьому необхідно мати на увазі, що встановлюване в такий спосіб сортування впливає тільки на відображення записів, але не впливає на їх фізичне розташування в таблиці або на наявні індекси.

Пошук і заміна даних. Щоб знайти або замінити в таблиці дані певного виду, наприклад, у всіх датах, які зберігаються в таблиці, замінити рік 99 на 1999, необхідно виконати команду *Виправлення | Знайти* (кнопка ) , після чого буде відображене діалогове вікно *Пошук і заміна* (рисунок 22). У цьому вікні є такі поля для настроювання параметрів пошуку:

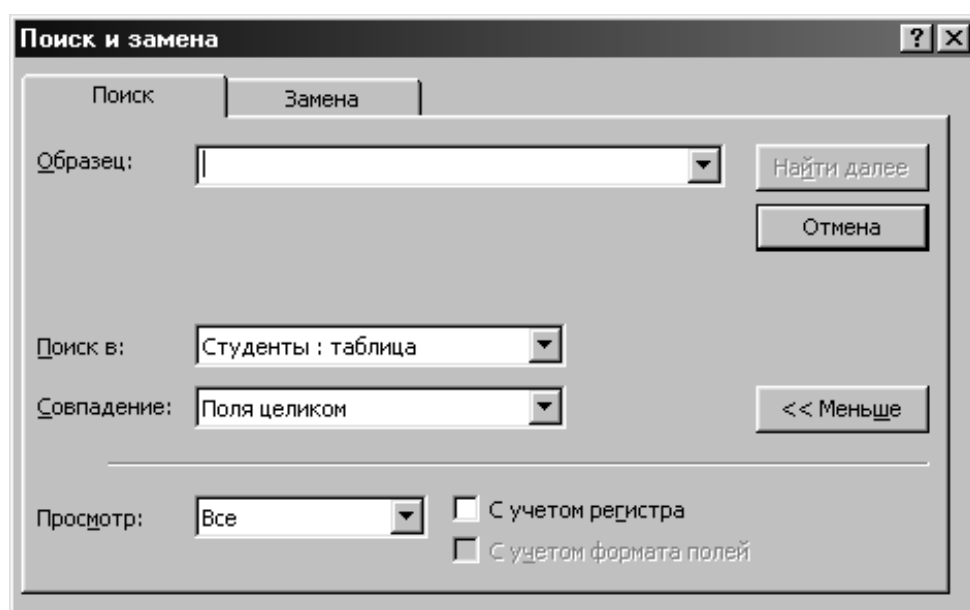


Рисунок 22 – Діалогове вікно Пошук і заміна

Зразок – рядок для пошуку. Наприклад, щоб знайти всі сполучення символів “99”, потрібно в це поле ввести значення 99. Можливе також використання спеціальних символів:

* - заміняє рядок будь-якої довжини, що складається з довільних символів;

? - заміняє один будь-який символ.

Зокрема, щоб знайти всі рядки, у яких є текст, що починається на “а”, можна в поле **Зразок** ввести значення **а***.

Пошук в – визначає режим пошуку або в поточному полі активної таблиці, або у всій таблиці.

Збіг – визначає тип пошуку. Наприклад, якщо в поле **Зразок** введено 99, то для значень 99, 1998 й 991 вибір того або іншого режиму приведе до таких результатів:

- 0 *З будь-якою частиною поля* – знайде значення і **99**, і **1998**, і **991**;
- 1 *Поля цілком* – знайде тільки значення **99**;
- 2 *З початку поля* – знайде значення **99** і **991**.

Перегляд – визначає напрямок пошуку від поточного запису:

- *Нагору* – до першого запису;
- *Вниз* – до останнього запису;
- *Всі* – пошук у всій таблиці.


З урахуванням регістра – при пошуці враховується регістр (тобто малі і великі букви), що використався при вказівці рядка в поле **Зразок**.

З урахуванням формату полів – при пошуці враховується формат відображення, встановлений для того або іншого поля. Наприклад, дата, збережена у форматі 10.02.99, відображається в таблиці як 10-фев-99.

Замінити на – рядок для заміни знайденого значення (тільки на вкладці *Заміна*).


Після того, як знайдено чергове значення, для пошуку наступного використовується кнопка *Знайти далі*. З іншого боку, у режимі заміни використовуються ще дві кнопки – *Замінити* (заміна чергового знайденого значення на зазначене) і *Замінити всі* (заміна всіх знайдених значень на зазначене).

Фільтрація даних. При роботі з таблицями часто буває необхідно відображати не всі записи, а тільки ті, які відповідають певним умовам. Для цього використовуються нижченаведені режими фільтрації.

1 *Фільтр по виділеному*. Щоб задати режим, при якому виводилися б тільки ті записи, у яких у певному полі наявне певне значення, необхідно виділити цю частину поля і скористатися режимом *фільтр по виділеному*. Нехай необхідно встановити такий фільтр, щоб у таблиці відображались тільки ті студенти, прізвища яких закінчуються на букву “а”. Для цього у полі **Прізвище** необхідно виділити останню букву “а” у прізвищі, наприклад, Жукова. Потім потрібно виконати команду *Записи | Фільтр | Фільтр по виділеному* (кнопка ). Після цього будуть відображені записи для студентів: Жукова, Волошина і Гущина (рисунок 23).


Студенты : таблица					
		КодСтудента	Фамилия	Имя	Отчество
▶	+	1	Жукова	Марина	Владимировна
	+	6	Волошина	Виктория	Ивановна
	+	8	Гущина	Елена	Валерьевна
*		(Счетчик)			

Рисунок 23 –Фільтрація за останньою буквою “а”

Найчастіше буває потрібно встановити режим фільтрації не за однією, а за декількома умовами. Наприклад, щоб і у прізвищі, і в імені останньою була буква “а”. У цьому випадку необхідно спочатку задати режим *Фільтр по виділеному* для прізвища (рисунок 23), а в отриманій таблиці задати той же режим для імені, тобто виділити останню “а” у будь-якому імені і натиснути кнопку  (рисунок 24).

Студенты : таблица					
		КодСтудента	Фамилия	Имя	Отчество
	+	1	Жукова	Марина	Владимировна
	+	8	Гущина	Елена	Валерьевна
▶		(Счетчик)			

Рисунок 24 –Фільтрація за ім’ям та прізвищем

Щоб скасувати режим фільтрації, можна скористатися кнопкою *Видалити фільтр* . Необхідно відзначити, що включення цієї кнопки виконує установку останнього застосованого фільтра (*Застосування фільтра*), а її відключення – скасування фільтрації (*Видалити фільтр*).


2 *Звичайний фільтр*. Режим фільтрації по виділеному використовується в тому випадку, коли потрібно відобразити тільки ті записи, які задовольняють всі зазначені обмеження (наприклад, остання буква “а” в прізвищі). Якщо ж необхідно відобразити всі записи, які задовольняють хоча б одну із зазначених умов (наприклад, остання буква “а” в імені АБО прізвищі), то в цьому випадку необхідно використати звичайний фільтр, який можна встановити командою *Записи | Фільтр | Змінити фільтр* (кнопка ). У лівому нижньому кутку вікна фільтрації є вкладки *Знайти* і *АБО*, при виборі яких у полях відображаються встановлені обмеження. Наприклад, після установаження розглянутих вище обмежень на ім'я і прізвище (остання буква “а”) на вкладці *Знайти* в полях **Прізвище** і **Ім'я** будуть встановлені фільтри такого виду: Like “*а” (рисунок 25).



Рисунок 25 –Зміна фільтра

Для задавання будь-яких умов в Access використовуються декілька операторів порівняння (таблиця 20), а також логічні оператори (таблиця 21).

Таблиця 20– Оператори порівняння

Оператор	Опис
=	Дорівнює
<	Менше

>	Більше
<=	Менше або дорівнює
>=	Більше або дорівнює
<>	Не дорівнює
IN	Перевірка на рівність значення якому-небудь елементу списку, що вказується в дужках оператора IN
BETWEEN	Перевірка на знаходження значення в зазначеному діапазоні (границі діапазону вказуються за допомогою логічного оператора AND)
LIKE	Перевірка значення на відповідність зазначеному шаблону

Таблиця 21– Логічні оператори

Оператор	Опис
AND	Перевіряє відповідність значення обом зазначеним умовам
OR	Перевіряє відповідність значення одній із

	зазначених умов
--	-----------------

Як приклад застосування операторів будуть розглянуті кілька умов (таблиця 22). Необхідно зазначити, що назви операторів прийнято писати або великими буквами (LIKE), або починати з великих (Like).

Наприклад, необхідно встановити єдине обмеження, яке повинно полягати в тому, що перша буква поля **Прізвище** повинна перебувати в діапазоні від “А” до “М”. У цьому випадку можна в режимі зміни фільтра задати для поля **Прізвище** обмеження `>= “А*” AND <=“М*”` або обмеження `BETWEEN “А*” AND “М*”` (рисунок 26).

Таблиця 22 – Приклади використання операторів

Приклад	Опис
LIKE “*а”	Вміст текстового поля закінчується на “а”
LIKE “*а” OR “*в”	Вміст текстового поля кінчається на “а” або на “в”
IN (“Волков”; “Морозов”)	Вмістом текстового поля є або рядок “Волков”, або рядок “Морозов”
BETWEEN 1 AND 10	Вміст числового поля перебуває в діапазоні від 1 до 10
BETWEEN “А*” AND “М*”	Перша буква текстового поля перебуває в діапазоні від “А” до “М”

>= "А*" AND <="М*"	Перша буква текстового поля перебуває в діапазоні від "А" до "М"
--------------------	--

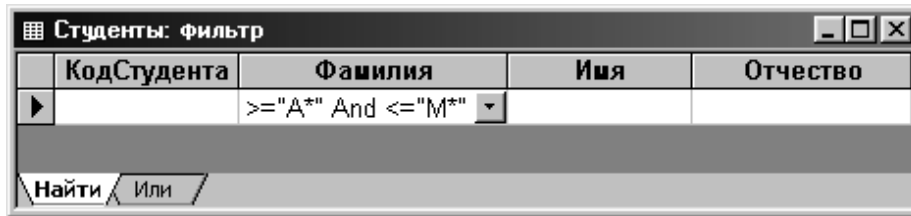



Рисунок 26 – Завдання умови фільтрації

Після застосування цього фільтра (кнопка ) у таблиці **Студенти** будуть відображені тільки ті записи, у яких значення поля **Прізвище** починається на букву, що перебуває в діапазоні від А до М (рисунок 27).

КодСтудента	Фамилия	Имя	Отчество
1	Жукова	Марина	Владимировна
4	Игнатов	Максим	Валерьевич
5	Волков	Юрий	Алексеевич
6	Волошина	Виктория	Ивановна
7	Литвинов	Евгений	Валерьевич
8	Гущина	Елена	Валерьевна
9	Кузнецов	Руслан	Алексеевич

Рисунок 27 – Результат застосування фільтра

Щоб задати обмеження на кілька полів, при яких відображалися б усі записи, які задовольняють хоча б одне із зазначених обмежень, можна скористатися вкладкою *Або* вікна фільтра. Наприклад, необхідно відобразити тільки тих студентів, у яких ім'я або прізвище закінчуються на букву "а". Для цього спочатку необхідно на вкладці *Знайти* вікна фільтра вказати для поля **Прізвище** умова LIKE "*а" (рисунок 28). Після цього необхідно вибрати вкладку *Або* і вказати для поля **Ім'я** умову LIKE "*а" (рисунок 29).

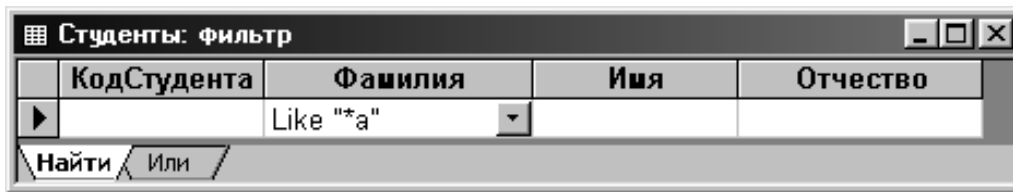


Рисунок 28 – Вкладка Найти вікна фільтра

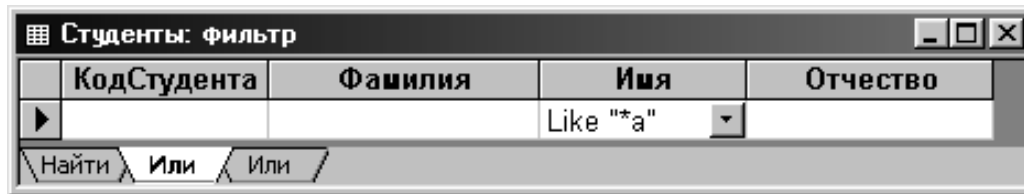




Рисунок29 – Вкладка Або вікна фільтра

Необхідно також відзначити, що при виборі вкладки *Або* автоматично з'являється наступна вкладка *Або*, тобто можна вказати кілька умов, хоча б одна з яких повинна виконуватися. Якщо потрібно видалити всі встановлені обмеження, то при відкритому вікні фільтра необхідно скористатися кнопкою *Очистити бланк* .

Після встановлення обмежень потрібно застосувати фільтр до таблиці (кнопка ), у результаті чого будуть відображені всі записи, для яких виконується хоча б одна із зазначених умов (рисунок 30).

КодСтудента	Фамилия	Имя	Отчество
+	1 Жукова	Марина	Владимировна
+	6 Волошина	Виктория	Ивановна
+	8 Гущина	Елена	Валерьевна
+	10 Семеновко	Светлана	Станиславовна
*	(Счетчик)		

Запись: 1 из 4 (Фильтр)

Рисунок30 – Результат застосування фільтра

3 Розширений фільтр. Якщо можливостей звичайної фільтрації не вистачає, наприклад, необхідно не тільки відображати певні записи, що задовольняють ті або інші умови, але й виконувати сортування даних по декількох стовпчиках, то в цьому випадку можна використати розширений фільтр. Для цього необхідно виконати команду *Записи | Фільтр | Розширений*

фільтр, після чого відобразиться вікно розширеного фільтра (рисунок 31). Потім необхідно вибрати поля, на які будуть накладатися обмеження (*Поле*), а також напрямок їхнього сортування (*Сортування*). Після цього вказати обмеження на вміст обраного поля (*Умова відбору*).

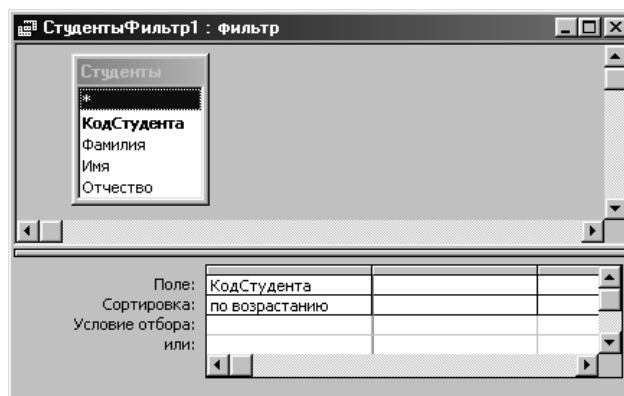



Рисунок 31 – Таблица Студенты. Розширений фільтр

Наприклад, необхідно відображати тільки тих студентів, імена яких починаються на букву “М”, а по батькові – на букву “В”, причому потрібно їх відсортувати за абеткою (прізвище, ім'я і по батькові). Для цього треба в першу комірку параметра *Поле* вибрати зі списку поле **Прізвище** (або перетягнути його в цю комірку зі структури таблиці у верхній частині вікна розширеного фільтра). Потім потрібно в другу комірку вибрати поле **Ім'я** і встановити умову відбору Like “М*”. Після цього необхідно в третю комірку параметра *Поле* вибрати **По батькові** й вказати умова відбору Like “В*”. На закінчення потрібно вказати для всіх полів режим сортування за зростанням (рисунок 32) і скористатися кнопкою . Після цього в таблиці будуть відображені тільки ті записи, які задовольняють зазначені умови, відсортовані в зазначеному напрямку (спочатку – сортування за прізвищем, потім – за ім'ям, і після цього – по батькові) (рисунок 33).

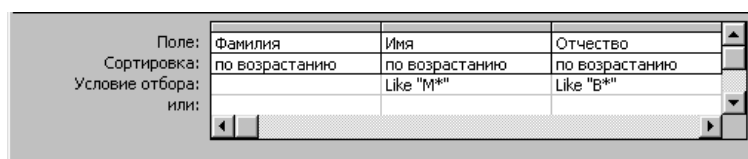


Рисунок 32 – Таблица Студенты. Параметри розширеного фільтра

	КодСтудента	Фамилия	Имя	Отчество
▶ +	1	Жукова	Марина	Владимировна
+	4	Игнатов	Максим	Валерьевич
*	(Счетчик)			

Запись: 1 из 2 (Фильтр)

Рисунок 33 –Результат застосування розширеного фільтра

3.2 Форматування таблиці

Як правило, далеко не завжди ті налаштування для подання таблиці, які встановлені за замовчуванням, повністю влаштовують користувача. Перебуваючи в режимі перегляду інформації таблиці, за допомогою команд меню *Формат* можна відформатувати відкриту таблицю, наприклад, встановити нові параметри шрифту, сховати або відобразити стовпчики, тощо. У меню формат є нижченаведені основні команди.

Шрифт. Можна встановити параметри шрифту, що буде використовуватися при відображенні даних у таблиці. Після виконання команди *Формат | Шрифт* на екрані буде відображено діалогове вікно *Шрифт* (рисунок 34), у якому можна встановити необхідні параметри шрифту: тип, розмір, вид або кольори. Після вибору всіх параметрів потрібно натиснути кнопку *ОК*.

Режим таблиці. Щоб установити параметри відображення таблиці, зокрема оформлення або лінії сітки, необхідно виконати команду *Формат | Режим таблиці*. Потім у діалоговому вікні *Формат таблиці* (рисунок 35) можна вибрати вид оформлення таблиці, відобразити або сховати лінії сітки, а також встановити для звичайного типу оформлення кольори фону і ліній сітки або вид границь і ліній у таблиці.

Висота рядка. Щоб змінити висоту рядків у таблиці, можна скористатися командою *Формат | Висота рядка*, після чого задати в діалоговому вікні *Висота рядка* (рисунок 36) необхідну висоту і натиснути кнопку *ОК*.

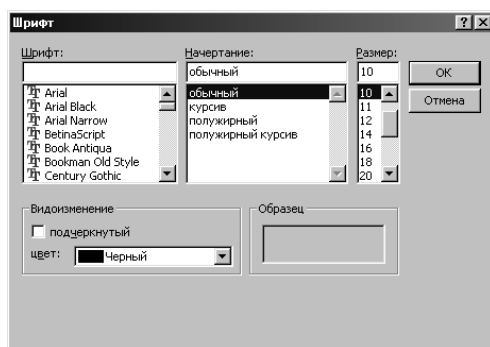


Рисунок 34 – Діалогове вікно Шрифт

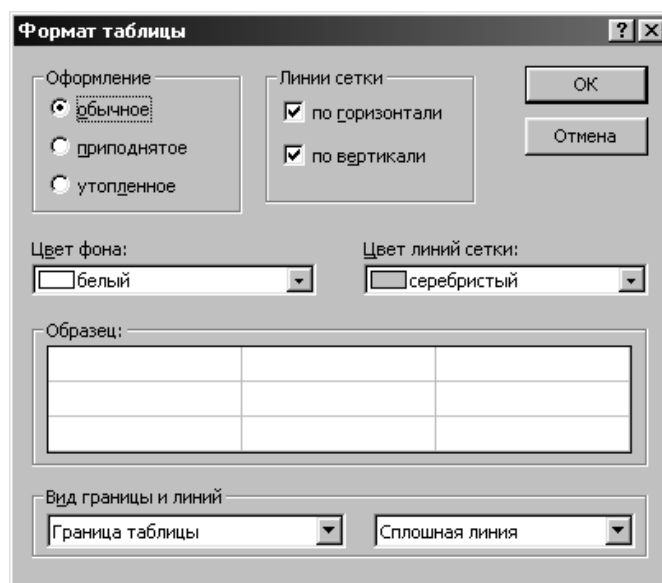


Рисунок 35 – Діалогове вікно Формат таблиці

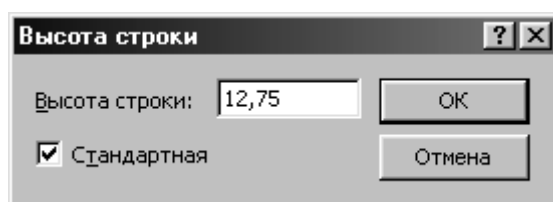


Рисунок 36 – Діалогове вікно Висота рядка

Ширина стовпчика. Для зміни ширини стовпчика необхідно його виділити (або встановити в ньому курсор) і виконати команду *Формат | Ширина стовпчика*. Після цього в діалоговому вікні *Ширина стовпчика* (рисунок 37) встановити необхідну ширину і натиснути кнопку *ОК*. Можна також скористатися кнопкою *По ширині даних*, однак необхідно мати на увазі, що при цьому ширина буде встановлена за максимальним

значенням цього стовпчика з тих його значень, які в цей момент відображені на екрані.

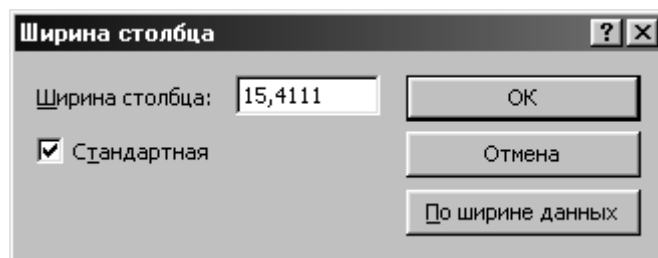


Рисунок 37 – Діалогове вікно Ширина стовпчика

Якщо жодне із установлюваних значень висоти рядка або ширини стовпчика не підходить, можна скористатися прапорцем *Стандартна*, щоб повернути стандартне значення ширини стовпчика або висоти рядка.

Сховати стовпчики. Щоб не відображати в таблиці деякі стовпчики (один або декілька), спочатку їх необхідно виділити, і потім виконати команду *Формат | Сховати стовпчики*. Для виділення декількох стовпців, розташованих поруч, тобто діапазону стовпчиків, необхідно виділити один із крайніх стовпчиків діапазону (наприклад лівий) і потім протягти виділення до правого стовпчика. З іншого боку, щоб виділити довільно розташовані стовпці, необхідно при виділенні чергового стовпчика утримувати натиснутою клавішу **Ctrl**. Можна також зменшити ширину стовпчика таким чином, щоб його не було видно, за допомогою миші або встановити для нього ширину 0.

Відобразити стовпчики. Якщо необхідно показати заховані раніше стовпчики, то в цьому випадку необхідно виконати команду *Формат | Відобразити стовпчики*, потім у діалоговому вікні *Відображення стовпчиків* відзначити стовпчики, які необхідно показати (рисунок 38), і натиснути кнопку *Закрити*.

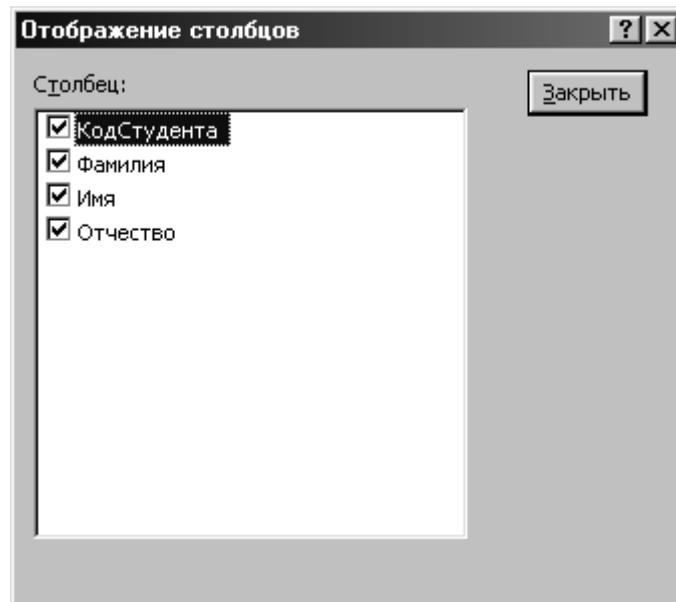


Рисунок 38 – Діалогове вікно *Відображення стовпчиків*

Закріпити стовпчики. Часто всі стовпчики таблиці не вміщаються у вікні, і тому доводиться виконувати прокручування вмісту полів таблиці (ліворуч-праворуч). У цьому випадку може виникнути необхідність у тому, щоб певний стовпчик (або кілька стовпчиків) не прокручувалися, а залишалися на місці. Щоб виконати закріплення стовпчика, необхідно його виділити або встановити на ньому курсор, і потім виконати команду *Формат | Закріпити стовпчики*.

Звільнити всі стовпчики. Якщо у встановленому раніше закріпленні стовпчиків уже немає потреби, можна скасувати цей режим за допомогою команди *Формат | Звільнити всі стовпчики*.

3.3 Використання підтаблиць

Структура реляційної БД, як правило, створюється таким чином, що в одних таблицях, поряд з інформативними даними, перебувають коди, за якими можна знайти зв'язані дані з інших таблиць. Наприклад, у БД Освіта таблиця Студенти зв'язана за полем **Код Студента** з таблицею Оцінки, у якій перебувають оцінки студентів. Очевидно, працювати з таблицею Студенти

було б зручніше, якби в ній для кожного студента можна було б відобразити його оцінки, тобто записи з таблиці Оцінки, які відповідають його індивідуальному коду.

В Access є можливість відображення в одній таблиці даних з іншої таблиці, якщо між цими таблицями є зв'язок. Одержувана в такий спосіб структура “таблиця в таблиці” називається *підтаблицею*. Загалом кажучи, підтаблиці автоматично створюються відразу після визначення зв'язків між таблицями. При цьому поруч із кожним записом у таблиці з'являється зображення хрестика, при натисканні на який буде розгорнута структура підтаблиці. Наприклад, для таблиці Студенти така структура буде створена на основі таблиці Оцінки (рисунок 39). Необхідно відзначити, що дані в підтаблицях можна редагувати так само, як і в звичайних таблицях (наприклад, додати нову оцінку, змінити або видалити введену раніше).

Студенты : таблица								
	КодСтудента	Фамилия	Имя	Отчество				
▶	1	Жукова	Марина	Владимировна				
					КодОценки	КодПредмета	Оценка	ДатаСдачи
▶					1	1	5	15.01.99
▶					2	2	5	20.01.99
*					(Счетчик)	0	0	
+	2	Морозов	Игорь	Олегович				
+	3	Петров	Михаил	Александрович				
+	4	Игнатов	Максим	Валерьевич				
+	5	Волков	Юрий	Алексеевич				
+	6	Волошина	Виктория	Ивановна				
+	7	Литвинов	Евгений	Валерьевич				
+	8	Гущина	Елена	Валерьевна				
+	9	Кузнецов	Руслан	Алексеевич				
+	10	Семенов	Светлана	Станиславовна				
*					(Счетчик)			

Рисунок 39 – Таблица Студенты з підтаблицею Оцінки

У меню *Формат* є команди для роботи з підтаблицями поточної таблиці. Наприклад, щоб для кожного запису відкрити підтаблиці, можна скористатися командою *Формат | Підтаблиця | Розгорнути всі*. З іншого боку, щоб згорнути всі розкриті підтаблиці, необхідно виконати команду *Формат | Підтаблиця | Згорнути всі*.

Однак далеко не у всіх випадках автоматично створювані підтаблиці будуть відповідати вимогам користувача. Щоб видалити наявну підтаблицю, необхідно скористатися командою *Формат | Підтаблиця | Видалити*. У цьому випадку підтаблиця буде вилучена, і користувач для даної таблиці може створювати підтаблицю за своїм розсудом.

Щоб створити нову підтаблицю, необхідно у відкритій таблиці виконати команду *Вставка | Підтаблиця*. Потім у діалоговому вікні *Вставка підтаблиці* (рисунок 40) можна вибрати таблицю (або запит), дані з якої будуть відображатися у підтаблиці. Для таблиці Студенти як підтаблицю необхідно на вкладці *Таблиці* вибрати *Оцінки*. Потім потрібно як параметри *Підлеглі поля* і *Основні поля* вказати поля, за якими виконується зв'язок (у цьому випадку це **Код Студента**), і натиснути кнопку *ОК*. У результаті буде сформована підтаблиця, що була раніше створена автоматично (рисунок 39).

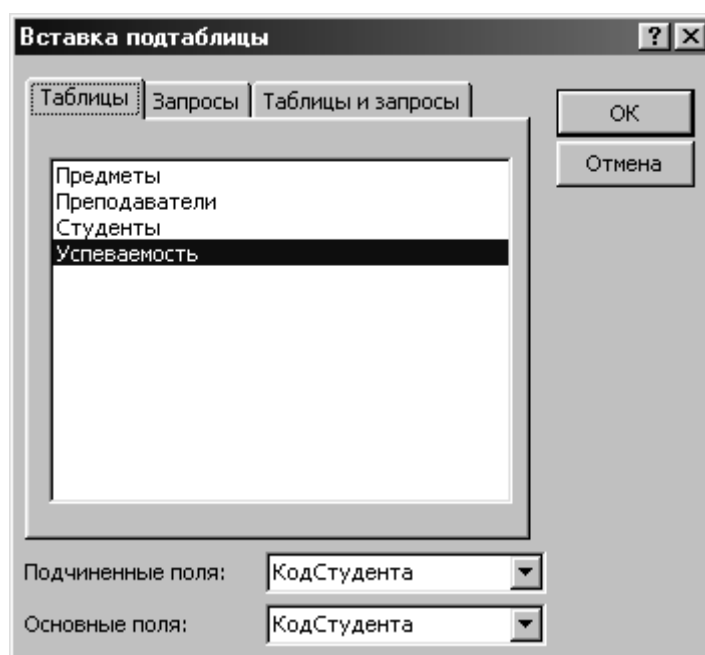


Рисунок 40 – Діалогове вікно Вставка підтаблиці

Можна також створити складну структуру, що складається з декількох підтаблиць. Наприклад, було б зручно відображати для кожної оцінки назву предмета, за який вона отримана. Для цього необхідно розгорнути підтаблицю для одного із записів (рисунок 39) і встановити в ній курсор. Потім потрібно виконати

команду *Вставка | Підтаблиця*, вибрати для зв'язку таблицю Предмети і вказати як сполучне поле **Код Предмета**. У результаті буде отримана структура “підтаблиця у підтаблиці”, тобто для кожної оцінки студента, що цікавить, можна відобразити відповідний предмет. При цьому, якщо в отриманій підтаблиці “другого рівня” автоматично була створена власна підтаблиця “третього рівня”, її необхідно видалити, тому що в результаті може вийти, що підтаблиці посилаються одна на одну. Зокрема, для оцінок відображаються предмети, для яких відображаються оцінки і т.д.

Можна створити підтаблицю третього рівня, у якій для кожного предмета буде відображатися прізвище викладача. Для цього треба в таблиці Студенти розгорнути підтаблицю Оцінки, у якій для будь-якого запису відобразити підтаблицю Предмети. Потім установити в останню підтаблицю курсор, знову виконати команду *Вставка | Підтаблиця*. Після цього вибрати таблицю Викладачі і вказати поле для зв'язування **Код Викладача**. У результаті вийде структура, що складається з чотирьох рівнів: основна таблиця Студенти і три вкладені підтаблиці Оцінки→Предмети→Викладачі (рисунок 41).

Студенти : таблиця				
	КодСтудента	Фамилия	Имя	Отчество
▶	1	Жукова	Марина	Владимировна
		КодОценки	КодПредмета	Оценка
▶	1	1	5	15.01.99
		КодПреподавателя	Название	КоличествоЧасов
▶	1	Физика		34
		Фамилия	Имя	Отчество
▶		Викулина	Валентина	Ивановна
*			0	0
+	2		2	5
*		(Счетчик)	0	0
+	2	Морозов	Игорь	Олегович
+	3	Петров	Михаил	Александрович
+	4	Игнатов	Максим	Валерьевич
+	5	Волков	Юрий	Алексеевич
+	6	Волошина	Виктория	Ивановна
+	7	Литвинов	Евгений	Валерьевич
+	8	Гуцина	Елена	Валерьевна
+	9	Кузнецов	Руслан	Алексеевич
+	10	Семенов	Светлана	Станиславовна
*		(Счетчик)		

Рисунок 41 – Таблица Студенти з декількома підтаблицями

3.4 Застосування підстановки

У попередньому розділі було розглянуто використання підтаблиць, які звичайно застосовуються в тому випадку, коли для запису в таблиці необхідно відображати більш повну інформацію, що міститься в іншій таблиці. Зокрема, для кожного студента можна відображати отримані ним оцінки.

У зв'язаних таблицях реляційних БД є поля, за якими встановлюється зв'язок. Наприклад, щоб у БД Освіта зв'язати таблиці Оцінки і Предмети, використовується поле **Код Предмета**. При цьому в таблиці Оцінки відображаються тільки коди предметів, а не їх назви. Щоб у цьому випадку відображати інформацію зі зв'язаної таблиці, тобто Предмети, можна створити підтаблицю, як було розглянуто раніше, але простіше і наочніше в цьому випадку використати *підстановку*.

Наприклад, потрібно в таблиці Оцінки відображати не коди предметів, а їх назви. Для цього необхідно відкрити таблицю Оцінки у режимі конструктора і вибрати для поля **Код Предмета** у списку *Тип даних* пункт *Майстер підстановок*. Дана дія запустить майстер підстановок, на першому етапі роботи якого (рисунок 42) необхідно встановити, з якого джерела стовпчик підстановки буде одержувати дані. Оскільки назва предмета міститься у таблиці Предмети, то необхідно вибрати перший пункт (*з таблиці або запиту*) і натиснути кнопку *Далі*>.

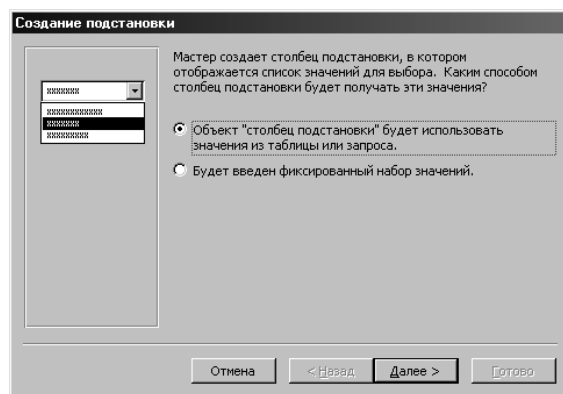


Рисунок 42 – Майстер підстановок. Крок 1

На другому етапі потрібно вибрати таблицю (або запит), що буде використовуватися як джерело даних, тобто таблицю Предмети (рисунок 43) і натиснути кнопку *Далі*>.

Наступний етап припускає вибір поля (або полів), значення якого (яких) будуть використовуватися у стовпчику підстановки. У цьому випадку потрібно в список *Обрані поля* перемістити поле **Назва** (рисунок 44) і натиснути *Далі*>.

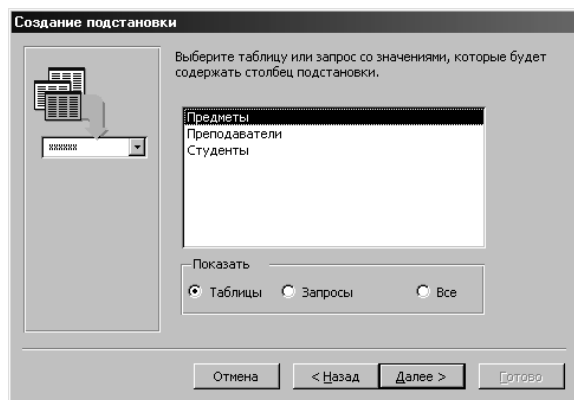


Рисунок 43 – Майстер підстановок. Крок 2

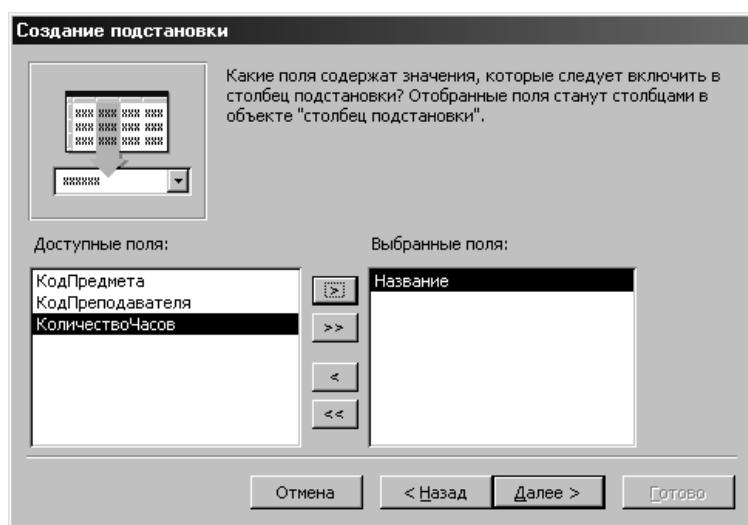


Рисунок 44 – Майстер підстановок. Крок 3

На четвертому етапі роботи майстра можна встановити ширину полів, які будуть відображатися у стовпчику підстановки, а також установити прапорець *Сховати ключовий стовпчик*, для того, щоб не відображалися коди предметів (рисунок 45), і потім натиснути *Далі*>.

На останньому етапі створення підстановки можна визначити підпис, що буде відображатися для стовпчика підстановки. Наприклад, можна задати підпис *Предмет* (рисунок 46), і потім, для завершення роботи майстра, потрібно натиснути кнопку *Готово*. Після цього необхідно зберегти таблицю, як це буде запропоновано зробити у вікні повідомлення.

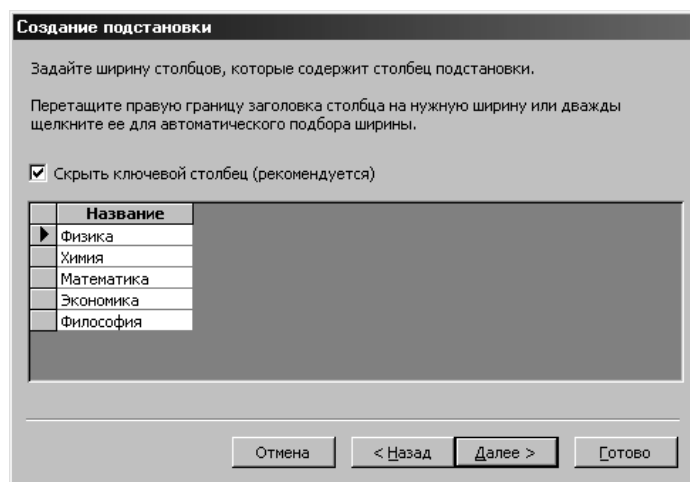


Рисунок 45– Майстер підстановок. Крок 4

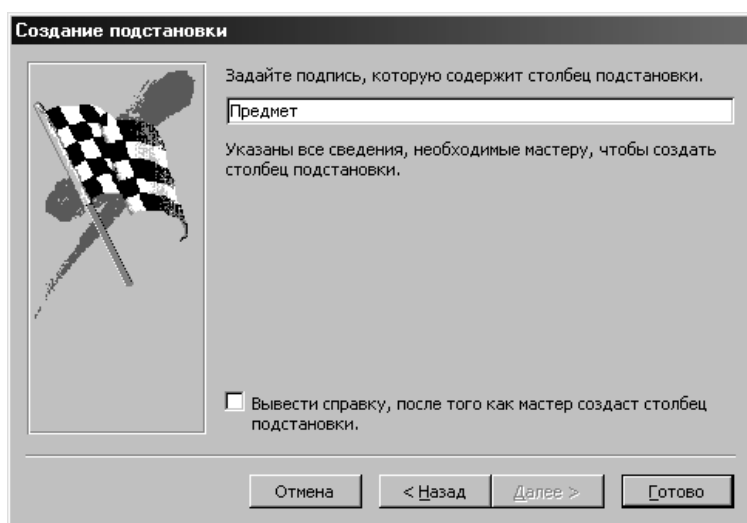


Рисунок 46 – Майстер підстановок. Крок 5

У результаті при перегляді вмісту таблиці Оцінки замість стовпчика **Код Предмета** буде відображатися стовпчик **Предмет**, у якому перебуває не код предмета, а його назва. Щоб змінити вміст стовпчика підстановки для якого-небудь запису, наприклад для першого, потрібно встановити курсор у стовпчик **Предмет**, розгорнути за допомогою стрілки список наявних предметів

(рисунок 47) і вибрати необхідну назву предмета. Однак це не означає, що в полі підстановки будуть записуватися текстові значення назв предметів. Як сполучне поле, так само використовується поле **Код Предмета**, просто в цьому випадку змінюється спосіб відображення інформації. У цьому легко можна переконатися, якщо змінити назву предмета в таблиці Предмети. Дана дія приведе до того, що внесена зміна автоматично відобразиться в полі підстановки таблиці Оцінки.

Успеваемость : таблица					
	КодОценки	КодСтудента	Предмет	Оценка	ДатаСдачи
▶	1	1	Физика	5	15.01.99
	2	1	Физика	5	20.01.99
	3	2	Химия	4	15.01.99
	4	2	Математика	4	20.01.99
	5	3	Экономика	3	10.02.99
	6	3	Философия	4	25.01.00
	7	3	Математика	5	22.01.99
	8	4	Физика	5	15.01.99
	9	4	Химия	4	20.01.99
*	(Счетчик)	0		0	

Рисунок 47 – Таблица Оцінки зі стовпцем підстановки Предмет

Завдання на виконання лабораторних робіт

Лабораторна робота 1. Створення незв'язаних таблиць

1 Знайомство з інтерфейсом програми Access.


2 Створення таблиць у режимі конструктора і заповнення даними.

Порядок виконання роботи

1 Напередодні ознайомитись з теоретичним матеріалом (с. 4-10).

2 За допомогою програми Access створити нову базу даних Освіта.

3 У режимі конструктора *під керівництвом викладача* створити таблицю тблСтуденти з полями:

- **Код Студента.** *Тип даних* – Лічильник (Счетчик). Це поле буде нашим первинним ключем. Щоб визначити зазначене поле ключовим, необхідно натиснути на іконку *Ключове поле* . На вкладці *Загальне (Общие)* у розмірі поля залишаємо за замовчуванням *Довге ціле (Длинное целое)*. Індексоване поле (*Індексоване поле*) – Так (Збіги не допускаються).

- **Прізвище Студента.** *Тип даних* – Текстовий (20 символів). *Обов'язкове поле* – Так. *Індексоване поле* – Так (Збіги допускаються).

- **Ім'я Студента.** *Тип даних* – Текстовий (10 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги - допускаються).

- **По батькові Студента:** *Тип даних:* Текстовий (15 символів); *обов'язкове поле:* Ні; *Індексоване поле:* Ні.

- **Група Студента:** *Тип даних:* Текстовий (10 символів); *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги допускаються).

- **Дата Народження:** *Тип даних:* Дата/час; *Формат поля:* Короткий формат дати; *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги допускаються).

- **Адреса Студента:** *Тип даних:* Текстовий (20 символів); *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги допускаються).

4 Зберігаємо створену таблицю під іменем тбл Студенти.

Самостійна робота

5 З режиму конструктора необхідно перейти в режим таблиці і ввести дані про якнайменше 8 студентів:

- 3 студенти з вашої групи;
- 3 студенти з інших груп вашого курсу (поток);
- 2 студенти не з вашого курсу (поток).

6 Створити нову таблицю (тбл Викладачі) в режимі конструктора з полями:

- **Код Викладача:** *Тип даних* – Лічильник; *Первинний ключ;* *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги не допускаються).

- **Прізвище Викладача:** *Тип даних* – Текстовий (20 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги допускаються).

- **Ім'я Викладача:** *Тип даних* – Текстовий (10 символів); *Обов'язкове поле* – Так; *Індексоване поле* – Так (Збіги допускаються).

- **По батькові Викладача:** *Тип даних* – Текстовий (15 символів); *обов'язкове поле* – Ні; *Індексоване поле* – Ні.

- **Кафедра Викладача:** *Тип даних* – Текстовий (15 символів); *Обов'язкове поле* – Так; *Індексоване поле:* Так (Збіги допускаються).

- **Вчений Ступінь:** *Тип даних* – Майстер підстановок. Обираємо “*Буде введено фіксований набір значень*”. Далі. У комірці (кожне слово у новому рядку) вписуємо: доцент, професор, академік. Далі. У наступному вікні залишаємо все за замовчуванням. *Готово*. *Обов’язкове поле* – Ні; *Індексоване поле* – Так (Збіги допускаються).

7 Зберігаємо таблицю з ім’ям тбл Викладачі.

8 У таблицю тбл Викладачі потрібно ввести не менше 6 викладачів з різних кафедр. Щонайменше одного викладача вказати без ученого ступеня і хоча б 2 – 3 викладачі (але не всі) мають бути з однієї кафедри.

Лабораторна робота 2. Створення зв’язаних таблиць

1 Встановлення зв’язків між таблицями з використанням майстра підстановок і заповнення даними.

2 Сорткування та фільтрація даних. Ознайомлення з елементом **Схема даних**.

Порядок виконання роботи.

1 Напередодні ознайомитись з теоретичним матеріалом 3-го розділу (с. 39 – 59).

2 У режимі конструктора (під керівництвом викладача) створити таблицю тбл Дисципліни з полями:

- **Код Дисципліни:** *Тип даних:* Лічильник (Счетчик). Первинний ключ; *Обов’язкове поле:* Так; *Індексоване поле:* Так (Збіги не допускаються).

- **Код Викладача:** *Тип даних:* Майстер підстановок. З двох можливих варіантів обираємо той, у якому пропонується використовувати значення з таблиці або запиту (*Объект столбец подстановки будет использовать значения из таблицы или запроса*). Тиснемо *Далі*. Із запропонованого списку вибираємо таблицю тбл Викладачі. Тиснемо *Далі*. У списку *Доступні поля* вибираємо **Код Викладача** і **Прізвище Викладача**. Тиснемо *Далі*.

Обираємо порядок сортування списку – **Прізвище Викладача** (можливі варіанти сортування в алфавітному порядку або ж навпаки). Тиснемо *Далі*. За необхідності можна змінити ширину стовпчика, який містить стовпчик підстановки. Тиснемо *Далі*. У наступному вікні залишаємо все за замовчуванням. Тиснемо *Готово*. У вікні, що випадає, погоджуємось з пропозицією зберегти таблицю і вводимо її назву (тбл Дисципліни). В описанні параметрів цього поля визначаємо: *Обов'язкове поле*: Так; *Індексоване поле*: Так (Збіги не допускаються).

- **Назва Дисципліни**: *Тип даних*: Текстовий (20 символів); *Обов'язкове поле*: Так; *Індексоване поле*: Так (Збіги допускаються).

- **Кількість Годин**: *Тип даних*: Числове (ціле); *Обов'язкове поле*: Так; *Індексоване поле*: Ні.

3 Зберігаємо таблицю (назву уже не вводимо, оскільки ми ввели її під час виконання попереднього пункту).

4 У режимі таблиці необхідно ввести дані про 5-7 дисциплін (у таблиці має бути щонайменше 1 викладач, що викладає більше 1 дисципліни).


5 У режимі конструктора створити таблицю **тбл Оцінки** з полями:

- **Код Дисципліни**: *Тип даних*: Майстер підстановок (усі дії у Майстрі підстановок виконуються самостійно аналогічно з попереднім завданням, ключове поле не задається). Для підстановки вибираються поля **Код Дисципліни** і **Назва Дисципліни**. *Обов'язкове поле*: Так; *Індексоване поле*: Так (Збіги допускаються).

- **Код Студента**: *Тип даних*: Майстер підстановок (усі дії у Майстрі підстановок виконуються самостійно аналогічно з попереднім завданням). Для підстановки вибираються поля **Код Студента**, **Прізвище Студента**, **Ім'я Студента** і **Група Студента**. *Обов'язкове поле*: Так; *Індексоване поле*: Так (Збіги допускаються).

- **Оцінка**: *Тип даних*: Числовий (ціле); *Обов'язкове поле*: Так; *Індексоване поле*: Так (Збіги допускаються).

- **Дата Здачі:** *Тип даних:* Дата/час (Короткий формат дати); *Обов'язкове поле:* Так; *Індексоване поле:* Так (Збіги допускаються).

6 Для таблиці тбл Оцінки ключовими є пара полів **Код Дисципліни** і **Код Студента**. Необхідно виділити за допомогою миші обидва поля і натиснути на панелі інструментів кнопку *Ключове поле* .

7 У режимі таблиці необхідно ввести інформацію про 2 - 3 оцінки, виставлені кожному студенту з різних дисциплін, використовуючи підстановку в полях **Код Студента** і **Код Дисципліни**.

8 Відкрийте вікно *Схема даних*. Перевірте коректність структури даних і встановіть цілісність даних у вікні *Зміна зв'язків*. Покажіть результат викладачеві.

9 Самостійно, використовуючи теоретичний матеріал 3-го розділу, застосувати різні варіанти сортування і фільтрації (за прізвищем, датою народження, номером групи і т. ін.).

Список літератури

1 Бекаревич, Ю.Б. Самоучитель Microsoft Access 2009 [Текст] / Ю.Б. Бекаревич, Н.В. Пушкина. – СПб.: БХВ – Петербург, 2009. – 720 с.

2 Дейт, К.Дж. Введение в системы баз данных [Текст] / Дейт К. Дж. – 6-е изд. – К.; М.; СПб.: Издательский дом "Вильямс", 2008. – 848 с.

3 Калянов, Г.Н. CASE – технологии: Консалтинг в автоматизации бизнес-процессов [Текст] / Г.Н. Калянов. – 3-е изд. - М.: Горячая линия-Телеком, 2008. - 320 с.

4 Маклаков, С.В. BPWin, ERWin. CASE – средства разработки информационных систем [Текст] / С.В. Маклаков. – М.: Диалог-МИФИ, 2007. – 300 с.

5 Мандрыкин, А.В. Информационные технологии в экономике [Текст]: Учеб. пособие / А.В. Мандрыкин, А.В. Непышневский. – Воронеж: ВГТУ, 2008. – 235 с.

6 Михеева, В.Д. Microsoft Access 2002 [Текст] / В.Д. Михеева, И.А. Харитоновна. – СПб.: БХВ – Петербург, 2007. – 1040 с.