# МЕТОДИЧНІ ВКАЗІВКИ ЧАСТИНА №3 ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

# з дисципліни: «Організація баз даних»

Для студентів 3 курсу навчання Спеціальності– Комп'ютерні науки

Затвердженно на методичній комісії

Факультету комп'ютерних наук

Протокол№\_\_\_\_ від \_\_\_\_\_

Декан \_\_\_\_\_Коваленко Л.Б.

Затверджено на засіданні кафедри

Інформаційних технологій

Протокол№\_\_\_\_ від \_\_\_\_\_

Зав. Кафедрою \_\_\_\_\_Кузніченко С.Д.

Одеса 2018

Методичні вказівки частина №3 для виконання лабораторних робіт з дисципліни «Організація баз даних» для студентів III курсу денної форми навчання. Спеціальність – «Комп'ютерні науки» / Козловська В.П.. Штефан Н.З.– Одеса, ОДЕКУ, 2018 – с. 56

### 3MICT

ВСТУП
Лабораторна робота №7 «Підключення БД MS SQL до клієнтського додатку
y Visual Studio 2010»
Теоретичні відомості
Практична частина6
Завдання
Контрольні питання
Лабораторная работа№8 «Створення додаткових форм для редагування
даних таблиць» 40
Практична частина
Завдання
Контрольні питання 54
ЛІТЕРАТУРА

#### ВСТУП

Методичні вказівки призначені для виконання лабораторних робіт №№7-8 з дисципліни «Організація баз даних».

Завдання основані на результатах попередніх лабораторних робіт, а саме створена база даних згідно з варіантом студента.

Оцінювання здійснюється згідно робочої програми (з урахуванням виконаної програмної частини роботи та кількість часу, за який студент реалізував поставлену перед ним задачу).

У процесі самостійного вивчення курсу студент повинен керуватися його програмою і вивчити за конспектом лекцій та літературою, що рекомендована викладачем, відповідний теоретичний матеріал.

# Лабораторна робота №7 «Підключення БД MS SQL до клієнтського додатку у Visual Studio 2010»

#### Мета роботи:

- 1. Знайомство з інтерфейсом Visual Studio 2010.
- 2. Отримання практичних навичок підключення бази даних MS SQL за допомогою компонентів Visual Studio 2010.

#### Теоретичні відомості

Система Microsoft Visual Studio має в наявності потужні вбудовані засоби створення і управління базами даних. Передбачена можливість створити новий проект бази даних і імпортувати схему бази даних з існуючої бази даних, SQL-файлу скрипта.

Потім можна викликати ті ж кошти програми проектування з графічним інтерфейсом (редактор Transact-SQL, конструктор таблиць), які доступні для розробки підключеної бази даних, щоб внести зміни в проект бази даних поза мережею і опублікувати зміни у виробничій базі даних.

Для створення клієнтського додатку до бази даних по перше необхідно створити проект в Visual Studio 2010.

Проект - це основна одиниця, з якої працює програміст. Він вибирає тип проекту - a Visual Studio створює шаблон проекту відповідно до обраного типу. У Visual Studio рідко створюється порожній проект. Замість цього ми повинні вказати середовищі тип проекту, який хочемо створити, а середовище розробки генерує файли і програмний код, які служать основою для обраного типу проекту. Після цього можна працювати над проектом, додаючи ваш код до створеного шаблоном проекту, що значно спрощує роботу.

#### Практична частина

У нашому випадку необхідно створити проект, заснований на Windows Forms, для чого Visual Studio згенерує проект з порожньою формою. Вибраний тип проекту також говорить компілятору які зовнішні бібліотеки необхідно підключити.

Перейдемо безпосередньо до реалізації завдання лабораторної роботи. Після запуску Microsoft Visual Studio вибираємо на стартовій сторінці пункт New Project – створити новій проект (див. рисунок 1):



Рисунок 1 – Створення нового проекту у Visual Studio 2010

У діалоговому вікні New Project зі списку встановлених шаблонів Visual C # обраний за замовчуванням. Кількома за додатком Windows Forms Application Visual C #.



Рисунок 2 – Вибір шаблону проектування

В поле Name потрібно вказати ім'я для майбутньої програми ApplicationPersonal (згідно з завданням на лабораторну роботу, студент пише своє прізвище), в поле Location можна змінити заданий за замовчуванням шлях до проекту.

У разі роботи над прикладом простого проекту, буде зручно все його файли зберігати в одній папці, необхідно скинути прапорець Create Directory For Solution. Тим не менше, більшість додатків зазвичай складаються з декількох проектів, тому скидати даний прапорець в загальному випадку не рекомендується - ця опція допомагає систематизувати структуру папок і уникнути плутанини і протиріч.

У разі установки прапорця Add To Source Control, Visual Studio відкриє нове вікно, в якому можна настроїти репозиторій вихідного коду (source

control). Він являє собою сховище для реєстрації коду. Ця можливість особливо корисна для колективної роботи, коли кожен розробник зможе реєструвати свій код в загальному сховищі вихідних кодовпрі роботі над масштабним рішенням.

Для завершення підтверджуємо, натиснувши кнопку ОК.



Рисунок 3 – Завдання імені та повного шляху до головної папки проекту

Після підтвердження імені та повного шляху до проекту відкривається основна форма програми, з якою і будемо працювати далі (рисунок 4).

Visual Studio надає безліч вікон, який відображають інформацію, необхідну для створення додатків. Деякий вікна, такі як Solution Explorer, відображені за замовчуванням, інші, наприклад налагоджувальні, з'являється при запуску програми в режимі налагодження. Повний список доступних

вікон розташований в головному меню Visual Studio під опцією View (рисунок 5).



Рисунок 4 - Головна форма проекту

View	Refactor Project Build	Debug	Team	Data	To	ols Architecto	ure Test	Analyze	Window H
29	Code	F7			Deb	ug 🔹 🚺	29		
2	Solution Explorer	Ctrl+\	N, S		35		Q = I 🖡	11 (21	0 0 SI
	Team Explorer	Ctrl+\	N, M		8	101 - 101 -			
-	Server Explorer	Ctrl+\	W, L						
	Architecture Explorer	Ctrl+\	N, N		•				
<b>.</b>	Call Hierarchy	Ctrl+\	N, K						
	Class View	Ctrl+V	N, C						
1	Code Definition Window	Ctrl+V	N, D						
	Object Browser	Ctrl+\	N, J						
3	Error List	Ctrl+\	N, E						
	Output	Ctrl+\	N, O						
	Start Page								
2	Task List	Ctrl+\	N, T						
R	Toolbox	Ctrl+\	N, X						
	Find Results								
	Other Windows			•	•	Command Wir	ndow	Ctrl	+W, A
	Toolbars					Web Browser		Ctrl	+W, W
	Full Screen	Shift+	Alt+Ente	er	-	Layer Explorer			
	Navigate Backward	Ctrl+-			de l	Macro Explore	r	Alt+	F8
国,	Navigate Forward	Ctrl+5	ihift+-			Source Contro	Explorer		
	Next Task				3	UML Model Ex	plorer	Ctrl	+ Ctrl+U
	Previous Task				6	Bookmark Win	dow	Ctri	+W, B
3	Properties Window	Ctrl+1	N, P			Document Out	tline	Ctrl	+W, U
	Property Pages	Shift+	F4		0	History			
_					38	Pending Chan	ges		
					3	Property Mana	iger		
						Resource View		Ctrl	+W, R
					F#	F# Interactive		Ctrl	+Alt+F

Рисунок 5 – Список доступних вікон Visual Studio 2010

Наступним кроком необхідно підключити нашу базу даних Personal. Для цього нпотрібно зайти в пункт меню Vew-> Server Explorer. У вікні клацаємо правим кліком по Data Connections та обираємо Add Connection ... (додати з'єднання).



Правой кнопкой





Рисунок 7 - Створення нового з'єднання

Після вибору з'єднання необхідно вибрати джерело даних Data Source. За замовчуванням джерелом є Microsoft SQL Server (SQL Client) (рисунок 8).

За допомогою кнопки Change... робимо виклик вікна Change Data Source (рисунок 9), де можно побачити список доступних джерел, у нашому випадку треба залишити Microsoft SQL Server (SqlClient), а також Data provider – вибираємо .Net Framework Data Provider for OLE DB і підтверджуємо вибір кнопкою OK.

Add Connection	Ľ	? ×	
Enter information to connect to the selected data source "Change" to choose a different data source and/or provid	or click ler.		
Data <u>s</u> ource:			
Microsoft SQL Server (SqlClient)	hange		
Server name:		_	
	<u>R</u> efresh		
Log on to the server			
Use <u>W</u> indows Authentication			
C Use SQL Server Authentication		_	
User name:			
Password:		$\mathbf{n}$	
🗖 Save my password		Если	1 SQL
Connect to a database		Serv	er 2000
Select or enter a database name:			
	7	]	
C Attach a database file:			
	Browse		
Logical name:		_	
	Ad <u>v</u> anced.		
Test Connection OK	Cancel		

Рисунок 8 – Параметри з'єднання з сервером.



Рисунок 9 – Вікно Change Data Source.



Рисунок 10 – Вікно Change Data Source (параметр Data provider).

Вікно Add Connection дозволяє вибрати ім'я доступного сервера і ім'я підключається бази даних (наприклад Personal). Для параметра Log on to the server залишаємо Use Windows Authentication. Після цього запускаємо тестування з'єднання кнопкою Test Connection.

Add Connection			<u>? ×</u>			
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.						
Data <u>s</u> ource:						
Microsoft SQL Server	(OLE DB)		⊆hange			
S <u>e</u> rver name:						
VALYA-F3CE527D7		•	<u>R</u> efresh			
Log on to the serve	r					
Use Windows A	uthentication					
C Use SQL Server	Authentication					
User name:						
Password						
	Save my password					
Connect to a datab	ase					
Select or enter	a <u>d</u> atabase name:					
Personal						
C Northwind Perepis			<b>_</b>			
Personal						
pubs						
tempdb TestNew						
TimeTable						
			Advanced			
Test Connection		ок	Cancel			

Рисунок 11– Вікно Add Connection.

При успішному з'єднанні висвічується повідомлення «Test connection succeeded».

Add Connection	<u>? ×</u>
Enter information to connect to the selected data source "Change" to choose a different data source and/or provid	or click er.
Data <u>s</u> ource:	
Microsoft SQL Server (OLE DB)	<u>⊂</u> hange
Server name:	
VALYA-F3CE527D7	<u>R</u> efresh
Log on to the server	
• Use Windows Authentication	
O Use SQL Server Authentication	
User nam	
Eassword Test connection succeeded.	
Connect to a OK OK	
Personal	<b>_</b>
C Attach a database file:	
	Browse,
Logical name;	
	Ad <u>v</u> anced
Test Connection OK	Cancel

Рисунок 12 – Успішне підтвердження з'єднання.

Після чого в Server Explorer відображається ім'я джерела даних (рисунок 13).



Рисунок 13 – Вкладка Server Explorer.

Переходимо до форми додатку.

Елементи управління на формі називається об'єктами. Кожен об'єкт має свій набір властивостей, подій і методів:

- Властивості об'єкта це його характеристики (висота, ширина і т.д.);
- Події об'єкта це події операційних систем або події ініціюються користувачем, на які може реагувати об'єкт (натискання кнопки);
- Методи об'єкта дії, які можна виробляти з об'єктом в ході виконання програм.

В БД всі об'єкти форм діляться на два класи:

- об'єкти управління об'єкти, які здійснюють управління БД (Наприклад: Кнопка або Список, що випадає);
- об'єкти для відображення інформації елементи, що відображають вміст таблиць, запитів або фільтрів, що дозволяють додавати змінювати і видаляти інформацію, і проводити її аналіз.

Всі форми в клієнтському додатку діляться на три групи:

- Форми для роботи з даними форми, що містять як об'єкти управління, так і об'єкти перегляду даних. Такі форми призначені для відображення, зміни, видалення та аналізу даних;
- Кнопкові форми форми, що містять тільки об'єкти управління, призначаються для відкриття всіх інших форм.
   Зауваження: Кнопкова форма, яка з'являється першою після запуску програми, називається, головною кнопковою формою.
- 3. Інформаційні та службові форми форми, що містять тільки елементи управління, призначені для відображення службової інформації (довідки), незв'язаної з таблицями, запитами і фільтрами, або для виконання службових операцій не пов'язаних з даними (Наприклад: форма з калькулятором)

Існує два види дизайну форм:

1. Стрічкові форми - форми, що виводять інформацію по одному запису;

2. Табличні форми - форми виводять інформацію у вигляді таблиці.

Найбільш часто в БД використовуються наступні об'єкти для відображення інформації:

1. Текстове поле (TextBox)

2. Напис (Label)

3. Напис з посиланням (LinkLabel)

4. Календар (DataPicker)

5. Перемикач (CheckBox)

6. Таблиця (DataGridView)

7. Список (ListBox)

8. Список, що випадає (ComboBox)

9. Текстове поле з маскою введення (MaskedTextBox)

<u>ТехtВох</u> - відображає текст і числові поля, це найбільш часто вживається об'єкт для відображення даних. Його можна створювати або перетягуванням з вікна "Data Sources", або підключити вручну. Створення цього об'єкта, перетягуванням можливо майже у полів будь-яких типів даних.

<u>Label</u> - повністю аналогічний об'єкту TextBox, але не дозволяє змінити дані. Цей об'єкт використовується для відображення заблокованих незмінних полів.

<u>LinkLabel</u> - спеціальний об'єкт для відображення посилань на адреси в Інтернеті. Його використовують для відображення текстових полів, якщо в них зберігаються адреси Інтернету або якийсь комп'ютерної мережі. Це новий об'єкт, йому не було аналога в Visual Basic 6.0.

<u>DataPicker</u> - спеціальний об'єкт, призначений для відображення полів типу даних "Дата / Час" у вигляді календаря.

<u>CheckBox</u> - об'єкт використовується для відображення логічних полів, може бути створений перетягуванням тільки для логічних полів.

<u>DataGridView</u> - об'єкт, що відображає джерело даних (таблицю, запит або фільтр) у вигляді таблиці.

<u>ListBox</u> - список відображає значення полів і дозволяє вибирати значення полів зі списку. Більш того, пункти списку можна задавати, використовуючи інше джерело даних.

<u>ComboBox</u> - об'єкт подібний об'єкту ListBox, проте інформація відображається не в списку, а випадаючому списку.

Для початку на головній формі проекту необхідно розмістити компонент TabControl з палітри Toolbox Visual Studio (рисунок 14).



Рисунок 14 – Головна форма проекту.

TabControl: це компонент, який складається з вкладок. Кожна вкладка є контейнером, тобто - там можна розташовувати інші компоненти Visual Studio. Навіть можна ще один TabControl туди помістити. Між вкладками можна перемикатися за допомогою миші (всіх закладках задається своє ім'я).

За замовчуванням з'являється дві вкладки. Але - це кількість можна збільшити. Для цього треба виділити сам елемент TabControl (саме елемент, а не вкладку; для цього треба виділяти саму верхню частину компонента TabControl). Після цього справа зверху з'явиться трикутник. Натиснувши на нього з'явиться дві кнопки: AddTab (додавання вкладки) і Remove Tab (видалити вкладку):

Також - управляти вкладками можна за допомогою властивості TabControl під назвою TabPages. Навпаки цього властивості буде кнопка (з трьома кнопками), натиснувши на неї відкриється вікно, в якому вже можна працювати з кожною конкретною вкладкою (видаляти, додавати вкладки, і задавати різні властивості для вкладок):

Отже, правим кліком по формі на TabControl вибираємо Add Tab (додати нову закладку - для кожної таблиці з нашої бази даних буде своя закладка для відображення її даних). Таким чином маємо 5 закладок.



Рисунок 15 – Робота з компонентом TabControl.



Рисунок 16 – Додавання нових закладок.

Тепер необхідно зайти до властивостей компонента TabControl на нашій формі за допомогою правого кліка. Вікно Properties дає можливість їх керуванням.



Рисунок 16- Вибір Properties компонента TabControl.

Потрібно зайти у властивість TabControl->TabPages... де для кожної сторінки змінюємо текстовий напис TabPage-> на ім'я таблиці з БД (як показано на рисунку 17).

File Edit Verw Project Build Debug Team Duta Tools Architecture Test Analyze Window Help         Image: State S	😒 Winda	owsFormsApplication16 - Microsoft Vi	sual Studio		_ 8 ×
Image: Set and Set	File Edi	t View Project Build Debug Team	Data Tools Architecture Test Analyze Window Help		
Image: Image	- 📑	🖮 • 📂 🛃 🦪   🐰 🖬 🛝   🄊 🔸	🗠 - 💭 - 🖳 🕨 Debug 🔹 🧭 connection 🔹 💀 😭 🖸	3 🔆 🗈 🖳 🖘	,
ToolSoot <ul> <li>Poperties</li> <li>Popertie</li></ul>	<b>#</b>	는 후 녜   ㅠ ゕ 프   클 萴 昭 :	ᆥ│┉깛;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		
Image: Construction of the second of the	Toolbox	<del>▼</del> ₽ ×	Form1.cs [Design]* ×	Properties	<b>-</b> □ ×
PageSetupDalog       ImageData         Panel       ImageData         PerformanceCounter       ImageData         PetureBox       ImageData         PrintDalog       ImageData         PrintDalog       ImageData         PrintDalog       ImageData         PrintDalog       ImageData         PrintDalog       ImageData         PrintDecoment       ImageData         PrintPreviewControl       ImageData         Process       ImageData         PropertyGrid       ImageData         SevelEcontroller       ImageData         SevelEcontroller       ImageData         SevelEcontroller       ImageData         StatusStrip       ImageData         TableLayoutPreve       ImageData         Prodesar       ImageData         Protesar       ImageData         Protesar       ImageData         SeveData       ImageData </th <th><u>*</u></th> <th>OpenFileDialog</th> <th></th> <th>tabControl1 System</th> <th>Windows.Forms.TabC 🕶</th>	<u>*</u>	OpenFileDialog		tabControl1 System	Windows.Forms.TabC 🕶
Panel       Alkohorp       False         PerformanceCounter       Alkohorp       False         Phintbrakes       Printbrakes       Andro       Top, Left         Printbrakes       Printbrakes       CauseNales       Andro       Top, Left         Printbrakes       Printbrakes       CauseNales       None       CauseNales       CauseNales       CauseNales       None         Printbrakes       Printbrakes       Printbrakes       None       CauseNales       None       None       Dock       None       No		PageSetupDialog	💀 Form1 📃	2 21 9 4 1	8
PerformanceCounter       Anchor       Top, Left         Appearance       Normal         PhitDalog       ContextMem.Strip       ContextMem.Strip         PrintDacument       PrintPreviewControl       Dock       Normal         ProgressBar       ProgressBar       Bendlag       Michaeter         ProgressBar       ProgressBar       Michaeter       Michaeter         SaveFielDialog       ProgressBar       Michaeter       Michaeter         SaveFielDialog       ProgressBar       Michaeter       Michaeter         SaveFielDialog       Michaeter       Michaeter       Michaeter         Sav		Panel		AllowDrop	False 🔺
PictureBox       Appearance       Normal         Pintbalog       CussesValidation       True         PintbreviewControl       PintPreviewOialog       CussesValidation       CussesValidation         Process       ProfessBar       PintBreviewOialog       Diskington         ProgressBar       ProgressBar       ProfessBar       PintPreviewOialog         ProgressBar       ProgressBar       ProfessBar       PintPreviewOialog         ProgressBar       ProfessBar       PintPreviewOialog       PintPreviewOialog         ProgressBar       ProfessBar       PintPreviewOialog       PintPreviewOialog         ProfessBar       PintPreviewOialog       PintPreviewOialog       PintPreviewOialog         ProfessBar       PintPreviewOialog       PintPreviewOialog       PintPreviewOialog         ProfessBar       PintPreviewOialog       PintPreviewOialog       PintPreviewOialog         PintPreviewOialog       PintPreviewOialog       PintPreviewOialog       PintPreviewOialog         PintPreviewOialog       PintPreviewOialog       PintPreviewOialog       PintPreviewOialog         PintPreviewOialog       PintPreviewOialog       PintPreviewOialog       PintPreviewOialog         PintPreviewOialog       PintPreviewOialog       PintPreviewOialog       PintPreviewOialog	~	PerformanceCounter	TabPage1   tabPage2   tabPage3   tabPage4   tabPage5	Anchor	Top, Left
PrintDialog       CauseValidation         PrintDrocument       ContextMemuStrip         PrintPreviewColor       False         PrintPreviewColor       False         PrintPreviewColor       False         PrintPreviewColor       False         SeviceControl       EntenSize         SeviceControl       BasinumSize         SeviceControl       MinimumSize         SeviceControl       MinimumSize         SeviceControl       MinimumSize         SeviceControl       MinimumSize         SeviceControl       MinimumSize         SeviceControl       Prading         SeviceControl       MinimumSize         TableLayoutPanel       Sevicontrip <th>2</th> <th>PictureBox</th> <th></th> <th>Appearance</th> <th>Normal</th>	2	PictureBox		Appearance	Normal
PrintDocument       ContextMen.Strip       SignMen.Strip       SignMen.Strip       SignMen.Strip       SignMen.Str	3	PrintDialog		CausesValidation	True
PrintPreviewControl       Cursor       Default         PrintPreviewControl       Dock       None         Process       Enabled       True         PropertyGrid       Enabled       True         PropertyGrid       B Font       Microsoft Sans Serf;         GenerateMember       True       Horsoft Sans Serf;         SerieFleDialog       B Font       Microsoft Sans Serf;         SerieFleDialog       B Font       Microsoft Sans Serf;         SerieFleDialog       B Font       B Font         SerieFleDialog       B Font       B Font         SerieFont       B Font       B Font	24	PrintDocument		ContextMenuStrip	(none)
PrintPreviewDialog       None         Process       None         ProgessBar       ProgessBar         PropertyGrid       ProdewtyGrid         RabeButton       ProdewtyGrid         RabeButton       ProdewtyGrid         RabeButton       ProdewtyGrid         SeverileDialog       ProdewtyGrid	3	PrintPreviewControl		Cursor	Default
Process       Enabled       True         ProgressBar       Enabled       True         PropertyGrid       GenerateMember       True         RadioButton       Hornsold       True         RichTextBox       ImageList       (rone)         ServielCohroller       E SoverlieDialog       Bendied       True         ServieCohroller       E Coction       9; 20       Location       9; 20         SetusStrip       SetusStrip       MaximumSize       0; 0       Bedied       Frivate         Multine       False       Bedied       Frivate       Bedied       Filse       Bedied       Filse       Bedied       Filse       Bedied       Filse       Bedied       Filse       Filse <th></th> <th>PrintPreviewDialog</th> <th></th> <th>DrawMode</th> <th>Normal</th>		PrintPreviewDialog		DrawMode	Normal
ProgressBar       ProgressBar         PropertyGrid       GenerateMember       True         RadioButton       H       File         RadioButton       ImageList       (none)         RichTextBox       ImageList       (none)         ServiceController       ImageList       (none)         ServiceController       ImageList       (none)         SplitContainer       ImageList       (none)         StatusTrip       StatusTrip       ImageList       (none)         TabControl       ImageList       (none)       ImageList       (none)         ImageList       (none)       ImageList       (none)       ImageList       (none)         ServiceController       ImageList       (none)       ImageList       (none)       ImageList       (none)         SplitContainer       ImageList       (none)       ImageList       ImageL		Process		Enabled	True
PropertyGrid       GenerateMember       Tue         PropertyGrid       HotTrack       False         Maine       NonControl       BitemSize       Sig         SerieFleDialog       BitemSize       Sig       BitemSize       Sig         SerieFleDialog       BitemSize       Sig       BitemSize       Sig       BitemSize       Sig         SerieFleDialog       BitemSize       Sig       Sig       BitemSize       Sig       Sig <td< th=""><th>_</th><th>ProgressBar</th><th></th><th>Font</th><th>Microsoft Sans Serif;</th></td<>	_	ProgressBar		Font	Microsoft Sans Serif;
■ Roberty data       ■ Hotrack       False         ● Radbeatton       ■ Hotrack       False         ■ RichTextBox       ■ DemSize       Size         ■ SaveFileDialog       ■ DemSize       Size         ● SeriaPort       ■ DemSize       Size         ■ SplitContainer       ■ MaximumSize       0;0         ■ SatusStrip       ■ MaximumSize       0;0         ■ TableLayoutPanel       ■ DemSize       0;0         ■ ToolStrip       ■ ToolStrip       ■ Size		Property/Grid		GenerateMember	True
<ul> <li>Radiudutoni</li> <li>Radiudutoni</li> <li>RekrextBox</li> <li>ServiceController</li> <li>SplitContainer</li> <li>SplitContainer</li> <li>SplitControl</li> <li>TableLayoutPanel</li> <li>TextBox</li> <li>ToolStripContainer</li> <li>ToolStripContai</li></ul>		Property and		HotTrack	False
Rch ExtBox       ImeMode       No.Cohrol         SaveFileDialog       Entendode       No.Cohrol         ServiceController       Entendode       No.Cohrol         SplitContainer       Entendode       No.Cohrol         StatusTrip       TableLayoutPanel       Entendode       No.Cohrol         TextBox       ImeMode       No.Cohrol       Entendode       No.Cohrol         TableLayoutPanel       Imemode       No.Cohrol       Entendode       No.Cohrol         ToolStrip       ToolStrip       StatusTrip       StatusTrip       StatusTrip       StatusTrip       RightToleft No.         ToolStrip       ToolStrip       ToolStrip       StatusTrip       StatusTrip       StatusTrip         ToolStrip       ToolStrip       ToolStrip       Normai       Tables       Normai         ToolStrip       ToolStrip       ToolStrip       StatusTrip       Tables       Normai         ToolStrip       ToolStrip       ToolStrip       ToolStrip       Normai       Tables         ToolStrip       ToolStrip       ToolStrip       ToolStrip       ToolStrip       ToolStrip         ToolStrip       ToolStrip       ToolStrip       ToolStrip       ToolStrip       ToolStrip         ToolS		Radiobutton		ImageList	(none)
SaveFileDialog       Bit Alliable       30,10         SerialPort       Dication       9,20         SerialPort       Locked       False         SplitContainer       Bit MaximumSize       0;0         StatuStrip       Bit MaximumSize       0;0         TabControl       Bit Padding       6;3         TabLetyoutPanel       Bit Padding       6;3         ToolStrip       ToolStrip       ShowToolTips         ToolStripContainer       Haffinder       Alse         ToolStripContainer       Collection       StatusStrip         ToolStripContainer       ToolStripContainer       ToolStripContainer	A	RichTextBox		ImeMode Imemoire	NoControl
SerialPort       Imagin ServiceController         ServiceController       Imagin Signature         Splitcontainer       Imagin Signature         StatusStrip       Imagin Signature         TabControl       Imagin Signature         Imagin TableLayoutPanel       Imagin Signature         Imagin ToolStrip       Imagin Signature         Imagin Signature       Imagin Signature         Imagin ToolStrip       Imagin Signature         Imagin	31	SaveFileDialog			9:20
ServiceController          B Margin       3; 3; 3; 3          SplitContainer          B Margin       3; 3; 3; 3          SplitContainer          B Margin       3; 3; 3; 3          StatusStrip          B Margin       3; 3; 3; 3          TableLayoutPanel          B Padding          Bile         ToolStrip          ToolStrip          Bale         ToolStripContainer          SizeMode          Normal         TableLayoutPanel          SizeMode          Solution         ToolStrip          ToolStrip          SizeMode          Normal         ToolStrip          ToolStrip          ToolStrip          SizeMode          Normal         Tablear          ToolStrip          ToolStrip          SizeMode          Normal         Tablear          ToolStrip          ToolStrip          SizeMode          Normal         Tablear          ToolStrip          ToolStrip          ToolStrip          ToolStrip         ToolStrip          ToolStrip          ToolStrip          ToolStrip          ToolStrip          ToolStrip         ToolStrip          ToolStri	1	SerialPort		Locked	False
SplitContainer		ServiceController		🖽 Margin	3; 3; 3; 3
Image: Splitter       Image: Splitter         StatusStrip       Modifiers         TabControl       Modifiers         TabLoontrol       Modifiers         TabLoontrol       Modifiers         TableLayoutPanel       Image: Splitter         TableLayoutPanel       Image: Splitter         TableLayoutPanel       Image: Splitter         TableControl       Image: Splitter         T		SplitContainer		MaximumSize     ■	0; 0
StatusStrip       Modifiers       Private         TabControl       B       Pading       6;3         TableLayoutPanel       B       Pading       6;3         RightToLeft       No       RightToLeft       No         Image: StatusStrip       RightToLeft       No       RightToLeft         Image: StatusStrip       StatusStrip       StatusStrip       StatusStrip         Image: StatusStrip       Image: StatusStrip       RightToLeft       No         Image: StatusStrip       StatusStrip       StatusStrip       StatusStrip         Image: StatusStrip       Image: StatusStrip       StatusStrip       StatusStrip         Image: StatusStrip       Image: StatusStrip       StatusStrip       StatusStrip         Image: StatusStrip       Image: StatusStrip       StatusStrip       StatusStrip         Image: StatusStrip       StatusStrip       StatusStrip       StatusStrip	۰ŀ	Splitter		MinimumSize	0; 0
TabControl     Multime     Fase       TableLayoutPanel     D     Pading     6; 3       Image: TextBox     No     RightToLeft     No       Image: ToolStrip     Fase     Stop     Fase       ToolStripContainer     SteP Mode     Normal       ToolStripContainer     ToolStripContainer     Collection       ToolStripContainer     ToolStripContainer     ToolStripContainer		StatusStrip		Modifiers	Private
Image: TableLayoutPanel     Image: TableLayoutPanel     RightToLeft No       Image: ToolStrip     ShowToolTips     False       ToolStripContainer     SizeMode     Normal       ToolStripContainer     ToolStrip     TableAgoutPanel       ToolStripContainer     Collection     TableAgoutPanel       ToolStripContainer     TableAgoutPanel     TableAgoutPanel		TabControl		Multiine Radding	Faise
Bell TextBox     RightToLeftLayout False       Imer     Size       ToolStrip     Size       ToolStripContainer     SizeMode       ToolTip     TabltAges       ToolStripContainer     TabltAges       ToolStripContainer     TabltAges       ToolStripContainer     TabltAges       ToolStripContainer     TabltAges		TableLayoutPanel		RightToLeft	No
Immer     ShowToolTips     False       Immer     El Size     486; 382       SizeMode     Normal       ToolStripContainer     TabIndex     0       ToolTip     TabIndex     0       ToolStripContainer     TabIndex     0       ToolTip     TabStop     True	abl	TextBox		RightToLeftLayout	False
B     Size     486;382       SizeMode     Normal       ToolStripContainer     TabIndex:       D     TabIndex:       ToolTip     TabStop       TrackBar     TabStop	12	Timer		ShowToolTips	False
ToolStripContainer SizeMode Normal TabIndex 0 TabIndex		ToolStrip		E Size	486; 382
Tabladex 0 Tablages (Collection) T TabPages (Collection) T TabStop True	77	ToolStripContainer		SizeMode	Normal
Tadrages (Collector)		TaalTia		TabPages	U (Collection)
V Traubar	<u>~</u>	Tued Dev		TabStop	True
		Trackbar			

Рисунок 17 – Вибір властивості TabPages.

TabPage Collection Editor				<u>?</u> ×
<u>M</u> embers:	tab	Page1 properties:		
0 tabPage1	+ 🔋	<b>₽</b> 2↓   ⊂		
2 tabPage3	+	AccessibleDescript		<b></b>
3 tabPage4	<u> </u>	AccessibleName		
4 tabPage5		AccessibleRole	Default	
		Appearance		
		BackColor	Control	
		BackgroundImage	(none)	
		BackgroundImagel	Tile	
		BorderStyle	None	
		Cursor	Default	
	Ð	Font	Microsoft Sans	s Serif;
		ForeColor	ControlT	ext
		RightToLeft	No	
Add Remove		Text	Отдел	
	L	HeaViewalStulaBac	THUS	<b>_</b>
		0	K C	Cancel

Рисунок 18 – Робота з Text TabControl. Так для tabPage1-> висвітиться напис Відділ, tabPage2-> висвітиться напис Посада, tabPage3-> висвітиться напис Особовий склад, tabPage4-> висвітиться напис Відділ, tabPage5-> висвітиться напис Відпустка.

В результаті головна форма проекту прийме наступний вигляд:



Рисунок 19 – Перейменовані закладки TabControl.

Задайте властивості форми наступним чином:

- FormBorderStyle (Стиль кордону форми): Fixed3D;
- MaximizeBox (Кнопка розгортання форми на весь екран): False;
- MinimizeBox (Кнопка згортання форми на панель задач): False;
- Text (Текст напису в заголовку форми): призвище студента.

Тепер необхідно вибрати елемент керування DataGridView з палітри Toolbox. Він дозволяє відображати і редагувати табличні дані з різних типів джерел данних. Розглянемо його детальніше.

Об'єкт DataGridView призначений для відображення всієї інформації з таблиць, запитів або фільтрів на формі у вигляді таблиці. Цей об'єкт може бути створений як вручну (з подальшим його підключенням), так і перетягуванням за все джерела даних з вікна "Data Sources".

Однак найбільш часто його створюють перетягуванням всієї таблиці, запиту або фільтра з вікна "Data Sources" на форму.

При перетягуванні цього об'єкта на форму, як і у випадку з іншими об'єктами з'являється панель навігації. Вона виконує функції: переміщення по записах, додавання, видалення і збереження записів. Після створення об'єкта DataGridView можна налаштовувати як властивості всього об'єкта, так і властивості окремих стовпців.

Почнемо з настройки властивостей всього об'єкта. Налаштування даних властивостей здійснюється в основному через меню дій.

Можливі наступні варіанти ввімкнення:

- Choose Data Source джерело даних, що відображається в таблиці;
- Enable Adding додавати записи;
- Enable Deleting дозволяється користувачам видаляти записи;
- Enable Editing дозволяється користувачам змінювати значення полів таблиці;
- Enable Column Reordering дозволяється користувачам змінювати порядок стовпців, просто перетягуючи їх мишею.

Також в меню дій можливі наступне дії з таблицею:

- Dock in parent container вписати об'єкт в форму;
- Preview Data з'являється вікно з попереднім переглядом таблиці;
- Add Query додає SQL запит, який виконується на стороні клієнта;
- Add Column додавання нового стовпця в таблицю;
- Edit Columns настройка властивостей окремих стовпців таблиці.

Тепер перейдемо до налаштування окремих стовпців таблиці.

Якщо в меню дій вибрати пункт "Edit Columns", то з'являється вікно, де можна додавати, видаляти і редагувати стовпці. Для цього в списку стовпців лівій частині вікна вибираємо стовпець, а в правій - налаштовуємо його властивості. Найбільш часто настроюються наступні властивості:

1. Name - ім'я стовпця;

2. AutoSizeMode - підгонка ширини шпальти по його вмісту;

3. ColumnType - визначає зовнішній вигляд комірок стовпчика (який об'єкт для відображення інформації знаходиться в комірках стовпчика);

4. DataPropertyName - ім'я, що відображає в стовпці поля;

5. Frozen - фіксація шпальти (стовпець не пересувається при прокручуванні таблиці);

6. HeaderText - текст заголовка стовпця;

7. Width - ширина поля;

8. MaxInputLength - максимально вводиться довжина тексту;

9. MinimumWidth - мінімальна ширина стовпця;

10. ReadOnly - блокування стовпчика для редагування даних;

11. Resizable - дозволяє змінювати ширину стовпця;

12. SortMode - сортування даних в таблиці з цього стовпцю;

13. ToolTipText - підказка для стовпця;

14. Visible - робить стовпець невидимим.



Рисунок 20 – Вибір компонента DataGridView.

У вікні завдань DataGridView Tasks знімаємо галочки з Enable Adding (можливість додавання), Enable Eding (можливість редагування), Enable Deleting (можливість видалення) (рисунок 21).

Клацаємо по Choose Data Source (вибір джерела) і заходимо в Add Prijects Data Source... (рисунок 22).





Рисунок 21 – Робота з вікном DataGridView Tasks.



Рисунок 22 – Вибір джерела даних.

Використовуючи вікно Data Source Configuration Wizard як джерело даних вибираємо базу даних (DataBase) і переходимо далі по кнопці Next..

Data Source	e Configuration W	/izard				<u>?</u> ×
þ	Choose a Dat	a Source Ty	/pe			
<u>W</u> here w	ill the application	get data froi	m?			
Databa	se Service	Object	SharePoint			
Lets you (	connect to a databa	se and choose t	the database of	ojects for your a	pplication.	
			< <u>P</u> revious	<u>N</u> ext >	Einish	Cancel

Рисунок 23 – Choose a data source type.

Як модель бази даних потрібно вказати «Набір даних» (Dataset). Натискаємо далі на кнопку Next та переходимо у наступне вікно.



Рисунок 24 – Choose a Database Model.

Потім у вікні Вибору підключення бази даних (Data Source Configuration Wizard) необхідно ввести параметри підключення до бази MS SQL сервера. Переходимо на наступний крок кнопкою Next..

Data Source Configuration Wizard	? ×
Choose Your Data Connection	
Which data connection should your application use to connect to the dat	abase?
SQLOLEDB.VALYA-F3CE527D7.Personal	New Connection
This connection string appears to contain sensitive data (for example, a password), connect to the database. However, storing sensitive data in the connection string ca you want to include this sensitive data in the connection string?	which is required to in be a security risk. Do
m O No, gxdude sensitive data from the connection string. I will set this informati	on in my application code.
f C Yes, include sensitive data in the connection string,	
+ Connection string	
< <u>P</u> revious <u>N</u> ext >	inish Cancel

Рисунок 25 – Вибір з'єднання

Ставимо позначку на угоді збереження рядка підключення в файлі конфігурації. натискаємо Next і переходимо на наступний крок.

Клацнувши на кнопку Next, майстер Data Source Configuration відобразить діалогове вікно, в якому він поцікавиться, чи зберігати рядок підключення у файлі конфігурації програми. І хоча в даному підході є багато за і проти, одне з ключових його переваг полягає в тому, що він являє собою простий спосіб відокремити інформацію про підключення від іншої частини коду.

Якщо в діалоговому вікні встановити прапорець Yes, save the connection string as, то середовище Visual Studio збереже рядок підключення і ім'я, яке ви введете, в файлі конфігурації програми для виконуваного проекту. Крім того, вона додасть логічні оператори в код проекту, щоб отримувати і використовувати цей рядок підключення для взаємодії з БД. Файли конфігурації додатків - стандартне місце зберігання інформації про рядку підключення.

Data Source	Configuration Wizard				? ×
Ļ	Save the Connection \$	String to the App	lication Cor	nfiguration f	file
Storing con	nection strings in your applicati string in the application configu	on configuration file eas iration file, epter a pam	es maintenanc e in the hox an	e and deployme d then click Nex	ent. To save the
Do you wa	int to save the connection	string to the applica	tion configu	ration file?	
🔽 Yes, sa	ve the connection as:				
Conne	ctionString		1		
,					
		< Previous	Next >	Einish	Cancel

#### Рисунок 26 – Save the Connection String to the Application File

Тепер у вікні Data Source Configuration Wizard необхідно указати об'єктиджерела даних (рисунок 27).

Можна вибирати також окремі стовпці в таблицях та подання. У нижній частині діалогового вікна є поле, яке дозволяє вказати ім'я для класу DataSet. За замовчуванням Visual Studio використовує ім'я підключеної в даний момент БД і додає до цього імені слово DataSet.

Data Source Configuration Wizard	<u>?</u> ×
Choose Your Database Objects	
Which database objects do you want in your dataset?	
DataSet name: DataSet1	
< Previous Mext > Finish	Cancel

Після натискання на кнопку Finish робота майстра Data Source Configuration буде завершена і у вікні Solution Explorer (команда View-Solution Explorer) в проекті з'явиться новий запис. Елемент нового проекту має ім'я, вказане в діалоговому вікні та містить визначення класу, відомого як DataSet зі строгим контролем типів.

Об'ектDataSet - це набір даних, в нього входять колекції DataTables i DataRelations. Колекція Data Table містить набори рядків і стовпців з даними,

клас Data Relation дозволяє пов'язувати дані елементів Data Tables. Дані в об'єкті Data Set від'єднані від БД. Після збирання результатів запиту в об'єкті Data Set за допомогою об'єкта Sql Data Adapter з'єднання між БД і об'єктом Data Set припиняється. Об'єкт Sql Data Adapter призначений для роботи з від'єднаними даними.

Можливо, найкраще підтвердження такої його структури - метод Fill. Для виклику цього методу не потрібно «живе» підключення до БД. Коли викликаний метод Fill об'єкта Sql Data Adapter, хто ж не має відкритого підключення до БД, об'єкт відкриває з'єднання, виконує запит до БД, вибирає і заносить результати запиту в об'єкт Data Set і потім закриває з'єднання з БД.

Data Source Configuration Wizard	?×
Choose Your Database Objects	
Which database objects do you want in your dataset?   Image: Tables   Image: Tables	
DataSet name: DataSet1	
< Previous Mext > Finish Car	ncel

Рисунок 27 – Вибір об'єктів бази даних

Відзначаємо Tables, Views, Stored Procedures (Таблиці, Уявлення і Збережені процедури). В поле DataSet name вказуємо DataSet1.



Рисунок 28 – Choose Your database Objects

Тепер в списку Project Data Source -> Data Set відображаються всі наші об'єкти бази даних Personal, як показано на рисунку 29.

Далі кожну сторінку на формі необхідно прив'язати до свого джерела для відображення даних (рисунок 30).



Рисунок 29 – Відображення списку Project data Sources



Рисунок 30 – Робота з Choose data Sources



Рисунок 31

За прикладом сторінки Відділ заходимо в властивості правим кліком. У вікні Properties вибираємо властивість Columns (Collections ...).



Рисунок 32



Рисунок 33 – Робота з властивостями dataGridView1

У вікні Edit Columns вибираємо поле таблиці Otdel-> id\_otd. Значення поля Name - є ідентифікатором для даного об'єкта в коді програми.

Edit Columns				<u>? ×</u>
Selected <u>C</u> olumns:		Bou	und Column <u>P</u> roperties	
abl id_otd	+	•	2↓ □	
abl name_otd	+		MaxInputLength	32767
			ReadOnly	False
			Resizable	True
			SortMode	Automatic
		Ξ	Data	
			DataPropertyName	id_otd
		Ξ	Design	
			(Name)	idotdDataGridViewTextBo>
			ColumnType	DataGridViewTextBoxColu 💌
Add <u>R</u> emove		(N In ob	l <b>ame)</b> dicates the name used ject.	in code to identify the
	L. L.			OK Cancel

Рисунок 34 – Работа з вікном Edit Columns

У списку Bound Column Properties потрібно змінити властивість Visible-

>True.

Edit Columns				<u>? ×</u>
Selected Columns:		Вс	ound Column Properties	
abl id_otd	+	•	<u>₽</u> 2↓ 🖻	
abl name_otd	+		Appearance	▲
			DefaultCellStyle	DataGridViewCellStyle { }
			HeaderText	id_otd
			ToolTipText	
			Visible	True 🗾
			Behavior	True
			ContextMenuStrip	False
			MaxInputLength	32767
			ReadOnly	False 💌
		Y	isible	
	1	Ir	ndicates whether the co	blumn is visible.
Add Remove				
				OK Cancel

Рисунок 35 – Властивість Visible для поля id\_otd

Для зручності користувача заголовок поля name\_otd позначимо як «Назва відділу».

Edit Columns		<u>? ×</u>
Selected Columns:	Bound Column Propertie	IS
abl name_otd	Appearance	▲
	DefaultCellStyle	DataGridViewCellStyle { }
	HeaderText	name_otd
	ToolTipText	
	Visible	True
	🗆 Behavior	
	ContextMenuStrip	(none)
	MaxInputLength	32767
	ReadOnly	False 🗾 💌
	HeaderText The caption text on the	column's header cell.
Add <u>R</u> emove		
	·	OK Cancel

Рисунок 36 – Робота с властивістю HeaderText

😋 Appli	cationPersonal - Microsol	t Visual Studio	_ B ×
<u>File E</u> d	it <u>V</u> iew <u>P</u> roject <u>B</u> uild	ebug Team Data Tools Architecture Test Analyze Window Help	
i 🛅 -	🛅 • 📂 🚽 🖉 🕹 🛍	🛗 🤊 🕶 🖓 📲 🖳 🕨 Debug 🔹 🧭 🚮 🖸	🧟 🎌 🛃 🖳 🚬 🖕
単	을 후 릐   ㅠ 아 프	当到 寇 幸   ~ ~ 햐 햐 햐 ㅎ   우 한 햐 햐 ! 印 臣   관 하   등 후 :	
Toolbox		→ ╀ × DataSet1.xsd DataSet1.Designer.cs Form1.cs* Form1.cs [Design]* ×	Properties 🔹 🗖 🗙
- 🔅	DataSet1		dataGrid¥iew1 System.Windows.Forms.D≀ ◄
- ÷	Otdel_PostTableAdapter	Form1	
- 49	OtdelTableAdapter		AutoSizeColumpsMo None
<b>\$</b>	PersonaTableAdapter	Отдел Должность Личный состав Сотрудники Отпуск	AutoSizeRowsMode None
6	PostTableAdapter		BackgroundColor 📃 AppWorkspace
-	TableAdapterManager	Название отдела	BorderStyle FixedSingle
- 63	VacationTableAdapter		CausesValidation True
	ViewVacTableAdapter		CellBorderStyle Single
	ViewWerkerTable&daeter		ClipboardCopyMode EnablewithAutoHead
345	viewworkerrabieAuapter		ColumnHeadersDefz DataGridViewCellStyle
	workerTableAdapter		ColumnHeadersHeig 31
E Exce	d Controls		ColumnHeadersHeig AutoSize
t Wind	lows Workflow v3.0		ColumnHeadersVisib True
E Stan	idard		Columns (Collection)
E Cont	rol Flow		ContextMenuStrip (none)
🗄 Offic	e Ribbon Controls		Cursor Default
🗄 Com	mon WPF Controls		DataSource otdelBindingSourc
🗄 Shar	ePoint Controls		
₩ Wind	lows Workflow v3.5		Dock None
🗉 Data	) adv. mate		EditMode EditOnKeystrokeOrF2
	indows Forms		Enabled True
	Pointer		EnableHeadersVisua True
- <u>-</u>	RackgroundWorker		GenerateMember True
	Diackyrodnid worker		GridColor ControlDark
<b>1</b>	Bindingivavigator		Inerviode NoControl
<u> </u>	BindingSource		Locked False
ab	Button		
<b>~</b>	CheckBox	🖉 dataSet1 📅 otdelBindingSource 🖏 otdelTableAdapter 🛱 postBindingSource	MaximumSize 0; 0
80	CheckedListBox		MinimumSize 0; 0
36	ColorDialog	🔄 🔯 postTableAdapter 📅 personaBindingSource 🛛 🖾 personaTableAdapter	Modifiers Private
	ComboBox		
Ready			
		🕅 SOL Query Ana 🛛 🕐 Pervontratium 🔛 Lect 1 C# doc 🖉 Application Re 🏠 Берьмянный -	🔲 🗖 🚜 🖓 🕾 🗐 🗖 🚳 10:10

Рисунок 37 – Змінений заголовок поля «Название отдела».

В результаті вміст таблиці буде відображатися в такий спосіб:

🔡 F	orm1					- 🗆 🗵
0	тдел ј	1олжность   Личныі	й состав   С	отрудники	Отпуск	1
		Название отдела				
	•	Администрация				
		Бухгалтерия				
		Маркетинга				
		Производствен				
		Технический				

Рисунок 38 – Відображення змісту таблиці Відділ.

Edit Columns			<u>? ×</u>
Selected <u>Columns:</u> abl FIO         abl name_otd         abl name_post         abl MainWork         ✓ main_work         abl date_work         abl date_work         abl id_post         abl id_otd	<b>↑</b> <b>↓</b>	Bound Column Properties ColumnType Layout AutoSizeMode DividerWidth FillWeight Frozen MinimumWidth Width	DataGridViewTextBoxColu NotSet 0 100 False 5 5 5)
<u>A</u> dd <u>R</u> emove		Width The current width of the	Column.

Рисунок 39 – Робота з властивістю Width для атрибуту stavka.

Для атрибуту таблиці Відділ – stavka за допомогою параметру Width вкажемо максимальну кількість введених знаків.

Аналогічно все вищесказане виконуємо щодо інших сторінок (назви стовпців міняємо на російську мову для зручного використання клієнтського застосування).

ee App	licationP	Personal - Microso	oft Visual Studio									<u>- 8 ×</u>
File 6	Edit Viev	v Project Build	Debug Team Da	ita Tools Archite	cture Test 4	Analyze W	indow Help					
( i 🗇 ·	- 🛅 - 🛛	🎽 🛃 🎒   🐰 🖣	🖻 🛍 🕑 • (°	- 📮 - 📑 🕨	Debug	- 🌁 co	nnection	-	🔩 🖀 🌆 🛙	🖄 🎌 🛃 🔜 T 🚽	-	
草	후 휘	희 파 아 파	雪田昭尊		움 붉‡ 음‡ 음‡		14 18 18	₩.				
DataS	et1.xsd	DataSet1.Desig	ner.cs Form1.c	s* Form1.cs [D	esign]* ×					Properties		▼ □ ×
										dataGridView4 Syste	em.Windows.For	rms.Da 🕶
	Form1								_ 🗆 🗙	2↓ 🗉 🖋	3	
	-	- 1-										<b></b>
	Отдел	Должность   Личн	ый состав Сотруд	чики   Отпуск	<b>.</b>				in the second second			
		Фамилия	0	0	Осноовная		D			(Name)	dataGrid∀iew	14
		И.О.	Отдел	должность	работа	Ставка	дата приема			AccessibleDescriptio		- 1
										AccessibleName		- 1
										AccessibleRole	Default	- 1
										AllowDrop	False	- 11
										AllowUserToAddRov	False	- 11
										AllowUserToDeleteR	owsse	- 1
										AllowUserToOrderC	False	- 1
										AllowUserToResizeC	True	- 1
	ģ								þ	AllowUserToResizeR	True	lim 1
										AlternatingRowsDer	DataGridviewCe	зіібсук
										Anchor Auto Cine Columna Ma	Top, Lert	
										AutoSizeColumnismo AutoSizeDourdMode	None	
										AutobizeRowsMode	None Applitation	
										Backgroundcolor	EivedSingle	pace
										CausesValidation	True	
										CellBorderStyle	Single	
										ClinboardConvMode	EnableWithAutr	Head
					_					ColumnHeadersBorc	Raised	- read
					00					ColumnHeadersDefa	DataGridViewCr	ellStyle
										ColumnHeadersHeig	31	
	L									ColumnHeadersHeig	AutoSize	
										ColumnHeadersVisib	True	
										Columns	(Collection)	
										ContextMenuStrip	(none)	
	dataSet:	1 😽 otdelBind	dingSource 🏾 🏷	otdelTableAdapter	🚏 postBi	ndingSource				Cursor	Default	
					<u> </u>					DataMember		
•	nostTabl	leAdapter 🙄	nersonaBindingSour	re 🚯 person	aTableAdapter					DataSource	viewWorkerB	lindin 🗾
			per contact namigood r									

Рисунок 40 – Поля таблиці «Сотрудники».

	Фамилия	Отлел	Полжность	Осноовная	Ставка	Пата приема	
<u> </u>	И.О.	A		работа	1.00	01.02.2005	
·	иванов и.и.	Администрация	директор		1,00	01.03.2005	
	Петрова О.И.	Администрация	Зам.директора		1,00	01.02.2006	
	Кузнецова О.П.	Производствен	Рабочий		1,00	01.03.2010	
	Зимина А.Л.	Администрация	Секретарь	•	1,00	03.09.2012	
	Потапов И.И.	Производствен	Начальник отде	~	1,00	01.03.2010	
	Сидорова И.С.	Бухгалтерия	Секретарь		0,50	09.01.2014	
	Сидорова И.С.	Бухгалтерия	Бухгалтер	~	1,00	01.03.2010	
	Иванченко А.И.	Производствен	Инженер	<b>v</b>	1,00	01.04.2010	
	Петренко И.П.	Технический	Секретарь		0,50	01.09.2014	Ē
	Петренко И.П.	Технический	Инженер	~	1,00	04.02.2006	
	Коваль О.И.	Производствен	Рабочий	•	1,00	01.03.2010	
	Зимин Л.Л.	Технический	Водитель	<b>N</b>	1,00	01.03.2010	
	Соколов С.И.	Бухгалтерия	Бухгалтер		0,50	01.03.2010	
	Сидоренко Н.С.	Мариатинга	Нацальных отле	5	1.00	21 11 2011	

Рисунок 41 – Відображення даних таблиці на формі.

#### Завдання

Кожен студент повинен підключить свою базу даних (з попередньої лабораторної роботи) до клієнтського додатку у Visual Studio 2010 згідно з наведеним вище послідовним алгоритмом. Результат повинен бути аналогічним рисунку 41.

#### Контрольні питання

- 1. Дайте визначення властивостям об'єкта. Наведіть приклади.
- 2. Що таке події та методи об'єкту?
- 3. На які класи поділяють об'єкти?
- 4. Які групи клієнтських форм ви знаєте?
- 5. Які види дизайну форм ви знаєте?

## Лабораторная работа№8 «Створення додаткових форм для редагування даних таблиць»

**Мета:** навчитися поширювати клієнтський додаток додатковими формами до таблиць для редагування даних.

#### Практична частина

У 7-й лабораторній роботі розробляється додаток для перегляду вмісту таблиць бази даних. При цьому студенти виконують підключення до бази даних (Data Connection), а потім на всіх закладках TabControl встановлюють інструмент DataGridView для перегляду даних і вибирають для нього джерело даних – таблицю або уявлення.

Вибрані джерела даних можна побачити в програмному вікні - під формою розробляється, як видно на рисунку 1.

Тут:

- 1. dataSet1 набір даних (у вигляді об'єкта класу DataSet), лічених безпосередньо з сервера;
- viewPosBindingSource i viewTeachBindingSource об'єкти класу BindingSource, призначені для зв'язку елементів форми з набором даних (джерела даних для DataGridView);
- 3. viewPosTableAdapter i viewTeachTableAdapter об'єкти (адаптери таблиць), класи яких описані у файлі DataSet1.xsd: вони використовуються для завантаження даних в DataSet з сервера і маніпуляції ними (можливе використання адаптерів для поновлення даних на сервері).

TeachFo	송 긬   과 쇼 프   orm.cs TeachForm.c Работа с БД "Пр <u>епод</u>	忌 迎 密 俳 s [Design] Ро аватели"	000 200 000 002   bs.cs Pos.cs [De	불	Main.cs [Design] X		Solution Explorer
	Волжности Преподава Название должности	гели   Читает лекции	Дате	Время	Harpyska (vac)	Добавить Изменить Удалить	Control Contro Control Control Control Control Control Control Control Control Co
•	dataSet1 🚏 viewP	osBindingSource	집 viewPosTabl	eAdapter S	<sup>24</sup> viewTeachBindingSource	िं viewTeachTableAdapter	

Рисунок 1 – Видиме відображення джерел даних програми

Також генерується метод, який оновлює вміст джерел даних:

```
private void Form1_Load(object sender, EventArgs e)
        {
            // TOD0: This line of code loads data into the
            'dataSet1.ViewTeach' table. You can move, or remove it, as needed.
            this.viewTeachTableAdapter.Fill(this.dataSet1.ViewTeach);
            // TOD0: This line of code loads data into the 'dataSet1.ViewPos'
table. You can move, or remove it, as needed.
            this.viewPosTableAdapter.Fill(this.dataSet1.ViewPos);
        }
```

Окремі рядки цього методу потрібно використовувати після додавання, зміни або видалення записів у відповідних таблицях. Наприклад, команда:

```
this.viewTeachTableAdapter.Fill(this.dataSet1.ViewTeach);
```

оновлює (перечитує) дані в джерелі даних, пов'язаному з поданням ViewTeach, створеному для перегляду таблиці Teachers. Цю команду в програмі потрібно застосовувати після додавання, зміни або видалення записів в таблиці Teachers.

Під формою додатка відображаються не всі джерела даних, а тільки ті, які використані в інструментах DataGridView. Всі підключення джерела даних (які можна використовувати в програмі) можна побачити, якщо клацнути по імені файлу DataSet1.xsd (якщо його не перейменовували) в правій частині екрана в віконці "Solution Explorer" (рис.2).



Рисунок 2 – Перегляд джерел даних додатка

На малюнку 2 видно, що в якості джерел даних вказані 3 таблиці: Teachers, Assistent, Teach\_position, - 2 подання: - ViewTeach, ViewPos, - і 6 процедур: - Pos\_add, Pos\_del, Pos\_edit, Teacher\_add, Teacher\_del, Teacher\_edit. При установці інструментів DataGridView потрібно було вибирати режим роботи цих інструментів без можливості додавання, зміни і видалення записів прямо в цих табличних компонентах.

#### Завдання

Для можливості додавання і зміни записів в таблицях потрібно використовувати додаткові форми, на яких користувач повинен писати або вибирати дані. Також користувач повинен мати можливість видалити обрану їм запис в таблиці. Зазначені дії можна виконувати або за допомогою пунктів меню, або за допомогою кнопок. Розглянемо спосіб з кнопками.

На кожну вкладку компонента TabControl необхідно встановити 3 кнопки (Button). Після установки кнопок потрібно змінити у них властивості (Properties): (Name) і Text. У властивості (Name) вказується ім'я компонента, як змінної програми, тому слід давати компонентам осмислені імена, щоб легше писати і читати програму.

Наприклад, кнопка, при натисканні якої викликається форма для додавання запису в таблицю Teachers, може називатися addTeachersButton (AddTeacherButt, AddTeachButt, AddTeachButton i т.п.).

Як властивості Text кнопок можна використовувати слова «Додати», «Змінити», «Видалити», а можна їх уточнити: «Додати викладача», «Змінити дані викладача», «Видалити викладача» (рис.3).

	Работа с БД "	Преподава	атели"					<u>-   ×</u>
, T	Цолжности   П <sub>Р</sub> Назва	еподавател ние	и Читает	<b>D</b>	Partic	Нагрузка	 	
	должн	ости	лекции	дага	Бремя	(час)	Добавить	
							Manager	
							Изменить	
							Удалить	

Рисунок 3 – Кнопки для маніпулювання даними в таблиці Teachers

Для маніпулювання даними потрібно використовувати створені раніше (лабораторна робота №3) збережені процедури. У середовищі Visual Studio можна не тільки переглянути перелік використовуваних процедур зі списком параметрів (рис.2), але також можна визначити їх типи, і уточнити, які параметри є вихідними.

Для цього потрібно в контекстному меню збереженої процедури вибрати пункт Properties (Властивості), серед властивостей вибрати Parameters (Параметри), і натиснути на три крапки біля слова (Collection) (рис.4).

	Pro	perties	<b>▼</b> □ ×
🔯 QueriesTableAdapter	Po	os_del Query	•
💷 Pos_add (@name_pos, @DateBeg, @lect, @loadHour, @k, @msg)		<b>₽</b> ↓ 🖻	
🧮 Pos_del (@id_pos, @m, @k, @msg)		Commond Tout	dha Dag, dal
🔟 Pos_edit (@id_pos, @name_pos, @DateBeg, @lect, @loadHour, @k, @msg)			abo.Pos_aei
🔳 Teacher_Add (@id_dep, @id_pos, @surname, @name1, @name2, @id_teacher, (		CommandType	StoredProcedure
📧 Teacher Del (@id teacher, @m, @k, @n, @msg)	E.	Connection	ConnectionString (Setti
Teacher, Edit (@id. den, @id. nos, @surname, @name1, @name2, @id. teacher, (		ExecuteMode	NonQuery
redener[care(@ra[cap)@ra[bos)@samame)@namer)@ramer)@ra[cadener)@		MethodsParameterTyp	CLR
		Modifier	Public
🖬 Assistent 🔿		Name	Pos_del
id_teacher		Parameters	(Collection)

Рисунок 4 – Перегляд властивостей збереженої процедури

Відкриється вікно редактора колекції параметрів, в якому зліва відображаються всі параметри процедури, що, а праворуч - властивості виділеного параметра. Для всіх типів параметрів основною властивістю є властивість Provider Type - це тип, який вказується для параметра при ініціалізації в програмі збереженої процедури.

Для цілих типів даних важливою властивістю є також властивість DbType - це тип, до якого потрібно конвертувати значення, що привласнюються параметру, наприклад, при введенні значення за допомогою компонента TextBox. Правда, для типу SQL Server varchar чомусь вказується тип Char як Provider Type (рис.7), хоча при ініціалізації процедури, що потрібно вказувати тип VarChar (рис.10).

Для символьних типів також важливо властивість Size - довжина рядка (рис.7).



Рисунок 5 – Властивості параметра, що має тип smallint в SQL Server (вхідного)

За замовчуванням вважається, що параметри процедури є вхідними параметрами - для них властивість Direction = Input. Для вихідних параметрів властивість Direction = InputOutput (рис.6,7). Це властивість обов'язково потрібно вказувати при ініціалізації збереженої процедури.



Рисунок 6 – Властивості параметра, що має тип tinylint в SQL Server

Parameters Collection Editor			?	2
<u>M</u> embers:	0	msg properties:		
0 @id_pos	+	<b>2↓</b> 🖻		
1 @m		Data		•
		AllowDbNull	True	٦
o Jennog		ColumnName		
		DbType	AnsiStringFixedLe	
		Direction	InputOutput	
		Precision	255	
		ProviderType	Char	
		Scale	255	
		Size	100	
		SourceColumn		
		SourceColumnNull	False	
J		SourceVersion	Current	
Add Remove	E	Misc		
	L	MonHiororchuIdTo	Falco	-
		0	K Cancel	

(вихідного)

Рисунок 7 – Властивості параметра, що має тип varchar (100) в SQL Server (вихідного)

<u>M</u> embers:		@D	ateBeg properties:		
0 @id_pos	+	•	<b>! 2↓</b>   🖻		
2 @DateBeg			Data		
3 @lect			AllowDbNull	True	
4 @loadHour			ColumnName		
5 @k			DbType	Time	
6 @msg			Direction	Input	
			Precision	255	
			ProviderType	DBTimeStamp	
			Scale	255	
			Size	2147483647	
			SourceColumn		
			SourceColumnNull	False	
			SourceVersion	Current	
Add Remove			Misc		_
			ManHiorarchuIdTo	Ealco	

Рисунок 8 – Властивості параметра, що має тип datetime в SQL Server (вхідного)

Members:		@le	ct properties:		<u> </u>
	+	•	<b>2</b> ↓ □		
2 @name_pos 3 @DateBeg	+		Data AllowDbNull	True	<b>^</b>
4 @lect			ColumnName DbTvpe	Boolean	
6 @k			Direction	Input	
(@msg			Precision ProviderType	255 Boolean	
			Scale Size	255 2147483647	
			SourceColumn	- Enlag	
			SourceVersion	Current	
Add <u>R</u> emove	]	Ξ	MapHiorarchuIdTe	Esleo	-
			0	K Cano	el

Рисунок 9 – Властивості параметра, що має тип bit в SQL Server (вхідного)

Оскільки при підключенні бази даних MS SQL Server 2000 використовується Data Provider for OLE DB, то для виклику збережених процедур використовуються об'єкти класу OleDbCommand. Типи параметрів вказуються через перерахування OleDbType (рис.10).

Об'єкти класу OleDbCommand (для простоти будемо називати їх просто командами), які використовуються для виклику збережених процедур - описуються як приватні (private) змінні класу форми перед конструктором форми (puc.10).

У конструкторі виконується ініціалізація цих команд. Можна весь код ініціалізації всіх команд помістити прямо в конструктор, але тоді він стане погано читаються. Тому краще ініціалізацію кожної команди описати як окремий метод, і в конструкторі викликати ці методи для всіх команд даного класу форми.

У прикладі на рис.10 на формі MainForm стоять дві кнопки для видалення записів з таблиць: кнопка DelPosButton і кнопка DelTeachButton; кожен з обробників події натискання на кнопку викликає свою збережену процедуру: Pos\_del (команда DelPosCmd), Teacher\_del (команда DelTeachCmd). Метод init\_command\_pos () ініціалізує команду DelPosCmd, а метод init\_command\_teach () - команду DelTeachCmd.

```
🔧 Teacher . MainForm
                                                            DelPosButton_Click(object sender, EventArgs e)
    using System.ComponentModel;
    using System.Data;
    using System.Drawing;
    using System.Linq;
    using System.Text;
    using System.Windows.Forms;
  ⊡namespace Teacher
    {
        public partial class MainForm : Form
  Ė
            private System.Data.OleDb.OleDbCommand DelPosCmd;
            private System.Data.OleDb.OleDbCommand DelTeachCmd;
            public MainForm()
                InitializeComponent();
                init_commands_pos();
                init commands teach();
            }
            private void init_commands_pos()
                DelPosCmd = new System.Data.OleDb.OleDbCommand();
                DelPosCmd.Connection = this.viewPosTableAdapter.Connection;
                DelPosCmd.CommandText = "Pos del";
                DelPosCmd.CommandType = System.Data.CommandType.StoredProcedure;
                DelPosCmd.Parameters.Add("@id_pos", System.Data.OleDb.OleDbType.SmallInt);
                DelPosCmd.Parameters.Add("@m", System.Data.OleDb.OleDbType.UnsignedTinyInt);
                DelPosCmd.Parameters.Add("@k", System.Data.OleDb.OleDbType.UnsignedTinyInt).Direction =
                                                             System.Data.ParameterDirection.InputOutput;
                DelPosCmd.Parameters.Add("@msg", System.Data.OleDb.OleDbType.VarChar, 100).Direction =
                                                             System.Data.ParameterDirection.InputOutput;
            }
```

#### Рисунок 10 – Опис і иініціалізація зберігаємої процедури

Розглянемо докладніше код програмі на малюнку 10. У перших рядках опису класу MainForm описуються його локальні змінні (приватні властивості). Такими в даному випадку є об'єкти класу OleDbCommand, який призначений для виконання команд SQL (запитів) і збережених процедур - DelPosCmd i DelTeachCmd:.

private System.Data.OleDb.OleDbCommand DelPosCmd;

У конструкторі викликається стандартний метод InitializeComponent (), а потім - методи для ініціалізації команд (використовуваних для виклику збережених процедур):

Далі описуються власне методи ініціалізації команд. Наприклад, метод init\_commands\_pos():

```
private void init_commands_pos()
{
          DelPosCmd = new System.Data.OleDb.OleDbCommand();
          DelPosCmd.Connection =
    this.viewPosTableAdapter.Connection;
          DelPosCmd.CommandText = "Pos_del";
```

DelPosCmd.CommandType =

System.Data.CommandType.StoredProcedure;

DelPosCmd.Parameters.Add("@id\_pos",

System.Data.OleDb.OleDbType.SmallInt);

DelPosCmd.Parameters.Add("@m",

System.Data.OleDb.OleDbType.UnsignedTinyInt);

DelPosCmd.Parameters.Add("@k",

System.Data.OleDb.OleDbType.UnsignedTinyInt).Direction =
System.Data.ParameterDirection.InputOutput;

DelPosCmd.Parameters.Add("@msg",

System.Data.OleDb.OleDbType.VarChar, 100).Direction =
System.Data.ParameterDirection.InputOutput;

}

У першому рядку методу створюється об'єкт DelPosCmd типу <u>OleDbCommand.</u> У другому рядку инициализируется його властивість Connection - так визначається використовується цією командою підключення до сервера. Для ініціалізації використовується властивість Connection відповідного табличного адаптера (адаптера змінною таблиці).

У третьому рядку властивості CommandText присвоюється назва збереженої процедури. В 4-му рядку вказується, що команда є збереженої процедурою.

У рядках з 5-го по 8-ю додаються параметри команди (збереженої процедури). Параметри додаються в тому порядку, як вони перераховані при описі збереженої процедури!

У процедурі Pos\_del параметри @k i @msg є вихідними параметрами. Це зазначено в методі за допомогою властивості Direction.

Власне виклик збереженій процедурі відбувається в обробнику події натискання на кнопку DelPosButton - методі DelPosButton\_Click (рис.11).



Рисунок 11 – Видалення запису з таблиці Teach position за допомогою

#### хранимой процедури

Работа с БД "Преп Должности Препода	одаватели" ватели	_		
название должности	Читает лекции Дата	Нагрузка Время (час)	Добавить	
•	Selected Columns: id_pos Название должности DateBeg Читает лекции LoadHour TimeBeg Дата Всема	Bound Column Properties	Изменить Удалить	
	ареняя Нагрузка (час) Add Remove	(Name) idposData( ColumnType DataGridVid (Name) Indicates the name used in code to ic object.	SridViewTextBo: ewTextBoxColu ▼ Jentify the Cancel	

Рисунок 12 - Столбці преставлення ViewPos

У першому рядку методу DelPosButton\_Click викликається діалогове вікно з заголовком «Видалення посади» і текстом всередині «Хочете видалити посаду назву». При цьому замість назву підставляється назва посади в виділеному рядку DataViewGrid. Атрибут Name\_pos знаходиться в другому стовпці уявлення ViewPos (рис.12), але нумерація починається з нуля.

Якщо користувач відповів «ОК» на питання в діалоговому вікні (вибрав кнопку «ОК») - System.Windows.Forms.DialogResult.OK в програмі то виконується решті код методу DelPosButton\_Click.

Спочатку присвоюються значення параметрам процедури, що Pos\_del; вхідних параметрів присвоюються певні значення, а вихідним параметрам порожні (System.DBNull.Value).

Потім в змінну previousConnectionState записується поточний стан з'єднання команди DelPosCmd з базою даних. Далі перевіряється, якщо поточний стан не є станом з відкритим з'єднанням, то з'єднання відкривається.

В операторі try .. finally виконується процедура, що зберігається. Якщо параметр @k повертає значення більше нуля (код помилки - наявність записів в пов'язаних таблицях), то виводиться діалогове вікно «Видалення з пов'язаних таблиць» і вас запитають на видалення виділеної (на DataViewGrid) записи з таблиці Teach\_position і всіх пов'язаних з нею записів в підлеглих таблицях.

Якщо користувач вибере «ОК», то збережена процедура викликається знову з параметром @m = 1, що означає видалення записів в пов'язаних таблицях. У розділі finally закривається з'єднання команди з базою даних. В останньому рядку коду методу DelPosButton\_Click оновлюється вміст DataViewGrid.

{

```
if (MessageBox.Show("Хотите удалить должность " +
              (string)(dataGridView1.CurrentRow.Cells[1].Value) + "?",
                  "Удаление должности", MessageBoxButtons.OKCancel) ==
               System.Windows.Forms.DialogResult.OK)
            {
                DelPosCmd.Parameters["@id_pos"].Value =
                     (short)(dataGridView1.CurrentRow.Cells[0].Value);
                DelPosCmd.Parameters["@m"].Value = 0;
                DelPosCmd.Parameters["@k"].Value =
System.DBNull.Value;
                DelPosCmd.Parameters["@msg"].Value =
                   System.DBNull.Value;
                System.Data.ConnectionState previousConnectionState =
                   DelPosCmd.Connection.State;
                if (((DelPosCmd.Connection.State &
                       System.Data.ConnectionState.Open)
                        != System.Data.ConnectionState.Open))
                {
                    DelPosCmd.Connection.Open();
                }
                try
                {
                    DelPosCmd.ExecuteNonQuery();
                    if ((byte)(DelPosCmd.Parameters["@k"].Value) > 0)
                    {
                        if
                      (MessageBox.Show((string)(DelPosCmd.Parameters["@
                      msg"].Value) + "\nВсе равно удалить?",
                            "Удаление из связанных таблиц",
                         MessageBoxButtons.OKCancel) ==
                         System.Windows.Forms.DialogResult.OK)
                        {
                            DelPosCmd.Parameters["@m"].Value = 1;
                            DelPosCmd.ExecuteNonQuery();
```

53

```
}
}
}
finally
{
    if ((previousConnectionState ==
        System.Data.ConnectionState.Closed))
        DelPosCmd.Connection.Close();
    }
    this.viewPosTableAdapter.Fill(dataSet1.ViewPos);
}
```

#### Контрольні питання

- 1. Як елементи форми можна зв'язати з набором даних?
- 2. Як переглянути джерела даних проекту у Visual Studio?
- 3. Як переглянути властивості збереженої процедури?

#### ЛІТЕРАТУРА

- 1. Дейт К. Дж. Введение в системы баз данных, 6-е издание: Пер. с англ. К.; М.; СПб.: Издательский дом «Вильямс», 2000. 848 с.: ил.
- Дет Л.Дж. Введение в системы баз данных 6-е изд. К.: Диалектика, 1998.
- 3. Мейер С.М. Проектирование баз данных М.: Мир, 1987.
- 4. Мамаев Е.В. Micrjsoft® SQL Sever 2000. СПб.: БХВ-Петербург, 2004. 1280 с.: ил.
- 5. Дэвидсон Л. Проектирование баз даннях на SQL Server 2000.: Пер. с англ. М.:: БИНОМ. Лаборатория знаний, 2003. 680 с., ил.
- Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1440 с.: ил.