

پایگاه داده

برای اینکه بتوانیم داده‌های برنامه خود را ذخیره کنیم باید آن را در جایی از حافظه جانبی، نگهداری کنیم. یکی از راه‌های ذخیره، استفاده از فایل‌هایی است که برنامه می‌سازد و داده‌ها را در آن ذخیره می‌کند. اما این روش زمانی که اطلاعات افزایش یافت و نیاز به مدیریت بیشتر اطلاعات و سرعت بیشتر باشد، ناکارآمد است. بنابراین ابزار دیگری نیاز است تا داده‌ها را در حجم زیاد بتواند ذخیره کند و مدیریت قدرتمند و متمرکزی را به همراه سرعت در کار ذخیره و بازیابی، فراهم نماید. راه حل این مهم، استفاده از پایگاه داده است.

در این فصل آنچه اهمیت دارد این است که چگونه بین برنامه خود، با پایگاه داده ارتباط برقرار

نماییم؟

پس از پایان این فصل انتظار می‌رود که فراگیر بتواند :

- ۱- از طریق برنامه به زبان C # اتصال بین پایگاه داده و برنامه ایجاد نماید.
- ۲- فایل پایگاه داده را با استفاده از SQL Server در محیط VS ایجاد کند.
- ۳- از ابزارهای نمایش داده‌ها بر روی فرم استفاده کند.
- ۴- برنامه‌های کاربردی مبتنی بر پایگاه داده بنویسد.

۱-۶- پایگاه داده چیست

پایگاه داده 'حجم انبوهی از اطلاعات است که به طور منظم و سازماندهی شده بر روی حافظه‌های جانبی به صورت فایل نگهداری می‌شود. اطلاعات درون یک پایگاه داده به سادگی و با سرعت، قابل دستیابی است. معمولاً در یک پایگاه داده اطلاعات در یک یا چند جدول قرار می‌گیرند^۲. هر جدول شامل اطلاعاتی در مورد یک موضوع است. مثلاً پایگاه داده یک مدرسه که اطلاعات درسی دانش‌آموزان را نگهداری می‌کند، می‌تواند شامل چندین جدول باشد. اطلاعات شناسنامه‌ای دانش‌آموزان در یک جدول و نمرات آنها در جدول دیگری نگهداری می‌شود. در جدول دانش‌آموزان، اطلاعاتی نظیر نام، نام خانوادگی، شماره ملی و آدرس مربوط هر یک از دانش‌آموزان مدرسه نگهداری می‌شود. هر سطر از این جدول یک رکورد دانش‌آموزی است و هر یک از اجزای تشکیل‌دهنده این رکورد، فیلد^۳ نامیده می‌شود (شکل ۱-۶).

جدول دانش‌آموزان

نام	نام خانوادگی	شماره ملی	آدرس
امین	رحیمی	0051234321	خیابان اشرقی ...
...
علی	محتشم	0051234567	خیابان مطهری ...

یک نمونه رکورد

مقدار یک فیلد

شکل ۱-۶- جدول اطلاعات دانش‌آموز

سؤال! با توجه به شکل ۱-۶، رکورد دانش‌آموز، از چند فیلد تشکیل شده است؟

بنابراین یک پایگاه داده کاربردی، از یک یا چند جدول تشکیل شده است و هر جدول نیز از

۱ - Data Base

۲ - Table

۳ - Field

تعدادی رکورد و هر رکورد از تعدادی فیلد.

سؤال: جدول معرفی دروس یک رشته از چه فیلدهایی تشکیل می‌شود؟

تمرین: به جدول شکل ۱-۶ فیلد و دو رکورد دیگر اضافه کنید.

برنامه‌ای که اطلاعات درون یک پایگاه داده را سازماندهی و مدیریت می‌نماید، سیستم مدیریت پایگاه داده^۱ یا به اختصار DBMS نامیده می‌شود. مزیت استفاده از DBMS برای نگهداری و دسترسی به اطلاعات را در درس بانک اطلاعاتی خوانده‌اید. برنامه ACCESS یک نمونه از DBMS است که همراه مجموعه Microsoft Office عرضه می‌شود. برنامه Access برای نگهداری اطلاعات و داده‌های شرکت‌ها و مؤسسات کوچک مناسب است.

برای مطالعه

چنانچه تعداد رکوردها و حجم اطلاعات زیاد باشد، باید از DBMS پیشرفته‌تری مانند Microsoft SQL Server و یا اوراکل^۲ که بسیار پیشرفته و گران قیمت هستند، استفاده کرد. شرکت مایکروسافت نرم‌افزار SQL Server را در نسخه‌های مختلفی با سطوح قابلیت و توانایی‌های متفاوتی عرضه کرده است. در بین آنها، نسخه Express به صورت رایگان از سایت مایکروسافت قابل دانلود است. همچنین نسخه کم حجمی از Express با نام LocalDB منتشر شده است که نیاز به تنظیم‌های خاص Express را ندارد^۳.

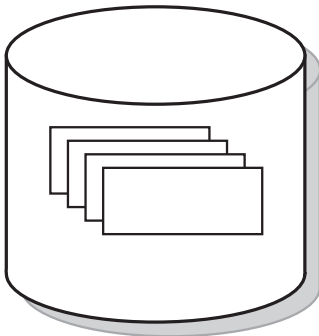
نسخه Sql Express به همراه نصب ویزوال استودیو می‌تواند نصب شود. همچنین در صورت نیاز می‌توان آن را به راحتی از وبسایت مایکروسافت دانلود نمود و نصب کرد. ادامه این فصل با استفاده از امکان فراهم شده توسط این نرم‌افزار به ایجاد یک پایگاه داده خواهیم پرداخت.

۱- Data Base Management System

۲- Oracle. www.oracle.com

۳- نسخه Sql Compact Edition نیز از طرف مایکروسافت ارائه شده است اما بعد از نسخه ۲۰۱۲ سایر نسخه‌های ویزوال

استودیو از آن پشتیبانی نمی‌کنند.

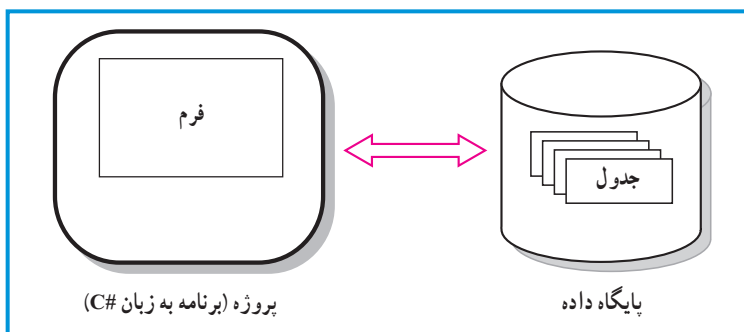


شکل ۲-۶ - پایگاه داده

شکل نمادین ۲-۶ یک پایگاه داده را نشان می‌دهد. هر پایگاه داده دارای یک نام است که از تعدادی جدول به نام‌های مختلف تشکیل شده است.

۲-۶- ایجاد پایگاه داده توسط Visual studio و استفاده از آن

در این بخش می‌خواهیم از طریق یک پروژه WFA که در داخل VS ایجاد می‌کنیم به یک پایگاه داده متصل شویم و اطلاعاتی را درون آن قرار داده و یا اطلاعات موجود آن را مشاهده نماییم. در این فصل ابتدا یک نمونه پایگاه داده برای نگهداری اطلاعات دفتر تلفن ایجاد می‌کنیم. شکل ۳-۶ به‌طور کلی نشان می‌دهد که می‌توان اطلاعات درون یک پایگاه داده را در پروژه نرم‌افزار کاربردی استفاده کرد و یا ویرایش کرد. به عبارت دیگر از طریق کنترل‌های موجود در یک فرم، می‌توان داده‌های درون پایگاه داده را مشاهده کرد. همچنین داده‌هایی که کاربر در روی فرم وارد می‌کند و یا اطلاعات دستکاری^۱ شده، می‌تواند در پایگاه داده، ذخیره شود (شکل ۳-۶).

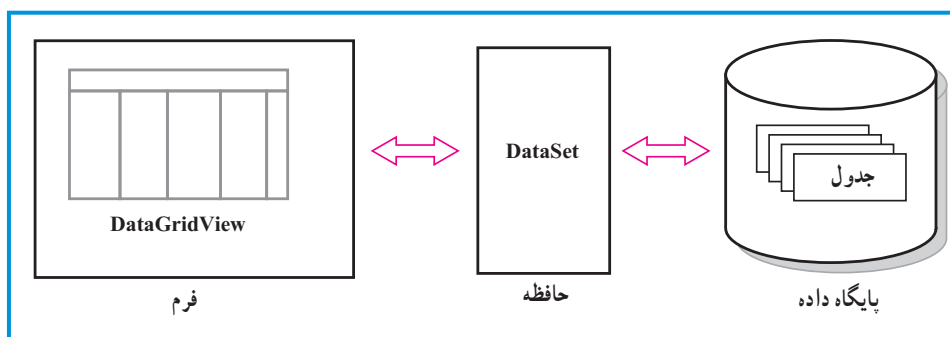


شکل ۳-۶ - ارتباط پروژه نرم‌افزار با پایگاه داده

مشکل بزرگی که معمولاً در کار با پایگاه داده وجود دارد این است که تاکنون همه عملیات بر روی اطلاعات درون متغیرها انجام می‌شد که آنها در حافظه Ram کامپیوتر قرار داشتند و سرعت دسترسی به آنها بسیار بالا بود. اما هنگامی که داده‌ها در پایگاه داده قرار دارند و در حافظه‌های جانبی ذخیره شده‌اند، با وجود سرعت بالای کار آن، در مقایسه با داده‌های موجود در Ram، کار با داده‌ها سرعت کمتری خواهد داشت. برای بالا بردن سرعت عملیات و افزایش کارایی سیستم، معمولاً بخشی از اطلاعات درون یک پایگاه داده، به داخل حافظه RAM آورده شده و سپس در درون فرم نمایش داده و یا دستکاری می‌شود.

بنابراین عملیات لازم برای دسترسی به اطلاعات درون یک پایگاه داده، به دو مرحله تقسیم می‌شود:

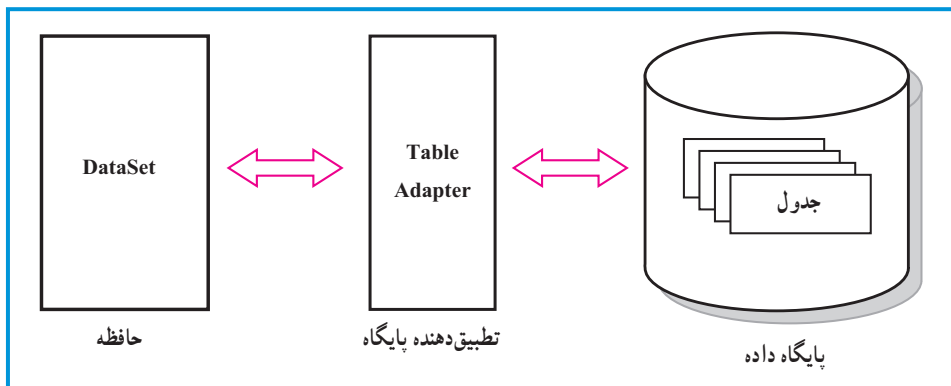
- ۱- ابتدا اطلاعات موردنظر از داخل پایگاه داده، به حافظه RAM کامپیوتر منتقل می‌شود به طوری که قابل مدیریت باشد. این عمل به وسیلهٔ ابزاری به نام DataSet انجام می‌شود.
- ۲- سپس باید از ابزاری برای نمایش اطلاعات در روی فرم مورد استفاده قرار گیرد. هنگامی که بخواهیم اطلاعات یک جدول را در روی فرم نمایش دهیم از ابزار DataGridView استفاده می‌کنیم. اما برای نمایش اطلاعات یک یا چند فیلد از ابزارهای دیگری که تاکنون با آنها آشنا شده‌اید، نظیر کنترل‌های برچسب یا جعبه متنی می‌توان استفاده کرد (شکل ۴-۶).



شکل ۴-۶- پروژه (برنامه به زبان C#)

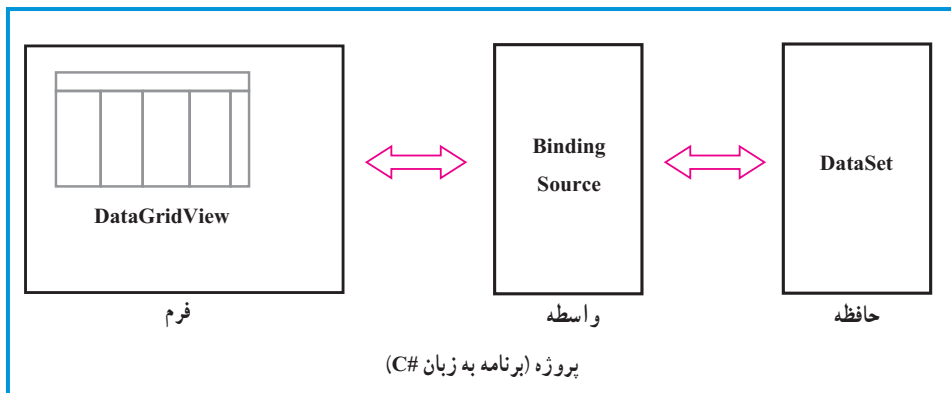
با توجه به اینکه انواع مختلفی از پایگاه داده وجود دارد و سیستم مدیریت هر یک متفاوت است، برای اینکه نیاز به انواع مختلف DataSet نباشد و برنامه کاربردی مستقل از نوع پایگاه داده عمل نماید از یک واسطه به نام تطبیق‌دهنده جدول TableAdapter استفاده می‌شود. بنابراین برای هر

پایگاه داده، یک تطبیق دهنده مخصوص طراحی می‌شود که از آن برای انتقال اطلاعات به DataSet استفاده شود (شکل ۵-۶).



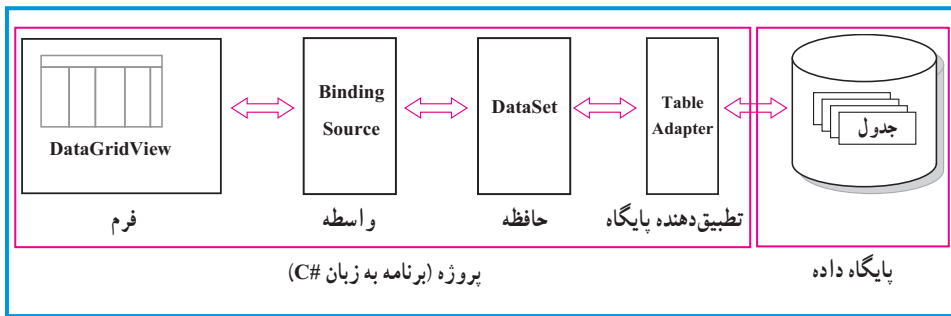
شکل ۵-۶- ارتباط DataSet و حافظه RAM

اکنون به سراغ مرحله دوم می‌رویم:
 برای انجام قدم دوم یعنی نمایش داده‌ها در روی فرم، از یک واسطه دیگری به نام BindingSource استفاده می‌شود. به وسیله این واسطه، کنترل‌های نمایشی در روی فرم به راحتی می‌توانند به بخشی از اطلاعات موجود در حافظه (DataSet) دسترسی داشته باشند (شکل ۶-۶).



شکل ۶-۶- ارتباط فرم با DataSet

برای هر یک از عملیات بالا کلاس و یا ابزاری تهیه شده است که از آنها در برنامه استفاده می‌کنیم (شکل ۷-۶).



شکل ۷-۶- ارتباط پروژه با پایگاه داده

آن طور که به نظر می رسد از دیدن تصویر و توضیحات ارائه شده، کمی ناامید و مأیوس شده اید. ناامید نباشید. در کتابخانه NET. مجموعه ای از کلاس های قدرتمندی برای اتصال با پایگاه داده و انجام عملیات مختلف بر روی داده، تحت عنوان ADO.NET تعریف شده است. در محیط VS با استفاده از ابزارهای تصویری و مرحله به مرحله می توانید تمام اشیای لازم برای اتصال به یک پایگاه داده را به دست آورید بدون اینکه لازم شود دستوراتی را بنویسید.

۳-۶- انواع کلاس ها و ابزارهای کار با پایگاه داده

۱-۳-۶- ابزار **DataGridview**: یک ابزار واسطه گرافیکی است که برای نمایش و به روز رسانی رکوردهای اطلاعاتی استفاده می شود.

۲-۳-۶- ابزار **BindingSource**: کلاسی است که داده های نمایش داده شده بر روی فرم را به منبع آن اتصال می دهد.

۳-۳-۶- ابزار **DataSet**: کلاسی است برای ذخیره داده های پایگاه داده در حافظه. شیء ساخته شده از این کلاس می تواند اطلاعات یک یا چند جدول را ذخیره نماید.

۴-۳-۶- ابزار **TableAdapter**: کلاسی است که ارتباط بین پایگاه داده با نرم افزار کاربردی را مهیا می کند.

اکنون برای آشنایی با ابزارهای توضیح داده شده، مثالی را انجام می دهیم:

کار در کارگاه: سافت یک دفتر تلفن

مثال ۱: می‌خواهیم یک دفتر تلفن بسازیم که بتوان اطلاعات دوستان و آشنایان را در آن ذخیره

کنیم.

برای آنکه یک نرم‌افزار متصل به پایگاه داده بسازیم مراحل زیر را باید دنبال کنیم:

۱- ایجاد الگوریتم و نقشه کار مناسب

۲- طراحی واسط کاربری مناسب

۳- طراحی پایگاه داده مورد نیاز

۴- ایجاد رابطه بین نرم‌افزار و پایگاه داده

۵- تکمیل کدها و برنامه

۶- آزمایش و رفع اشکال برنامه

توجه شود که برخی از این مراحل می‌تواند بنابر مقتضیات کار، جابه‌جا شود. به‌عنوان مثال

گاهی ابتدا پایگاه داده ساخته می‌شود و بعد کنترل‌های مربوط به واسط کاری اضافه می‌شوند.

الگوریتم و روش انجام کار: ابتدا یک واسط کاربری مناسب برای نمایش و درج افراد و

شماره تماس آنها طرح ریزی می‌کنیم. نمونه کار به صورت شکل ۸-۶ است:



نام و نام خانوادگی	شهر	شماره منزل	شماره همراه
حسن لائری	تهران	11111111	09123456789
حسن نوروزیان			
محمد قاسمی	شیراز	2222222	09121111111

شکل ۸-۶

برای نگهداری از این اطلاعات از یک پایگاه داده استفاده می‌کنیم. می‌توانید با برنامه Access پایگاه را ایجاد کنید و یا مانند مثال از sql server استفاده نمایید. پس از ایجاد پایگاه، به سراغ ابزارهای اتصال به پایگاه داده رفته و پس از آن، کنترل‌های نمایش اطلاعات را روی فرم قرار داده و یک واسط کاربری مناسبی را طراحی می‌نماییم. همان‌طور که دیدید ابتدا پایگاه داده ساخته شد و سپس کنترل مربوط اضافه شد. دلیل این تصمیم‌گیری را در ادامه خواهید دید.

اکنون برای ساخت اولین پروژه بانک اطلاعاتی به شرح زیر عمل می‌کنیم:

۱- ایجاد واسط کاربری مورد نیاز: یک پروژه ویندوزی WFA به نام AddressBook در مسیر دلخواه خود ایجاد کنید. با توجه به طرح ریزی انجام شده فعلاً کنترل دیگری نیاز نیست که اضافه شود.

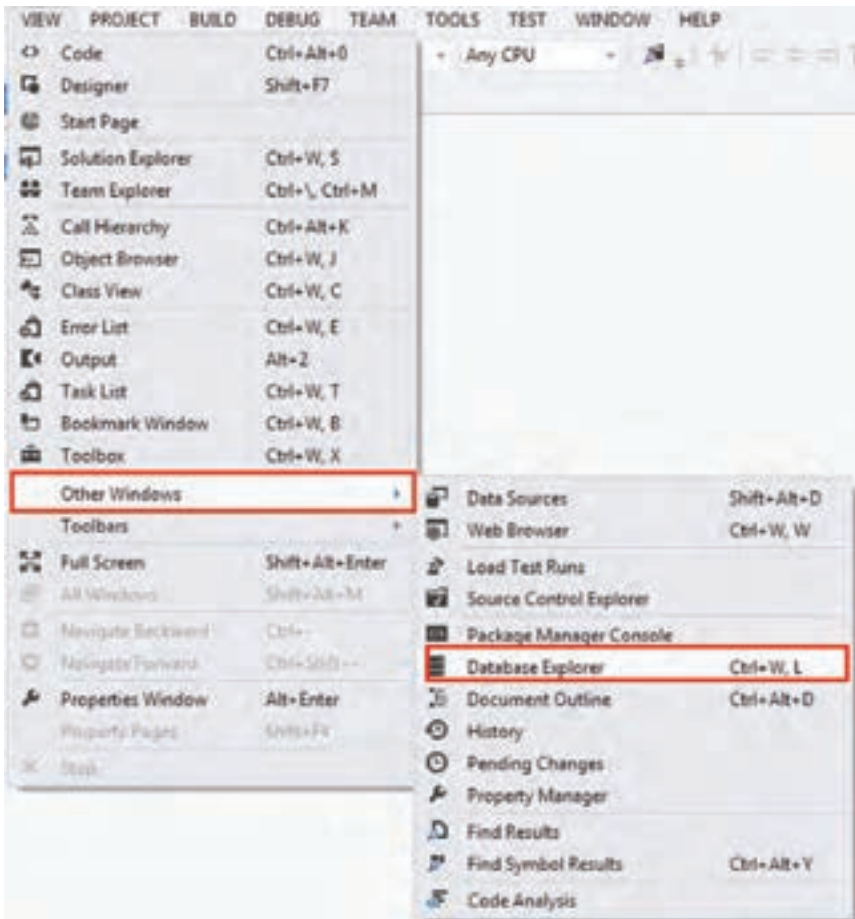
۲- طراحی و ایجاد پایگاه داده مورد نیاز: یک پایگاه داده‌ها برای موضوع مسئله، طراحی می‌کنیم. البته در این مثال به جهت سادگی، یک پایگاه داده به نام myDb ایجاد کرده که تنها شامل یک جدول اطلاعاتی به نام numbers است (جدول ۶-۱).

جدول ۶-۱- طرح جدول اطلاعاتی برای دفتر تلفن

نام و نام خانوادگی	شهر محل اقامت	شماره تلفن ثابت	شماره تلفن همراه

۳- پنجره مرورگر پایگاه داده (Database Explorer) را از منوی View بیاورید (شکل ۶-۹).

۱- در نسخه‌های غیر از express می‌توانید از پنجره server explorer نیز استفاده کنید.



شکل ۹-۶- اجرای فرمان نمایش پنجره مرورگر پایگاه داده

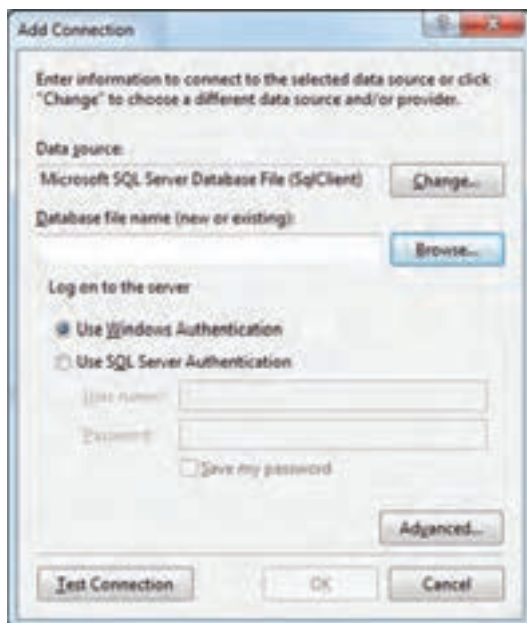
۴- در پنجره مرورگر پایگاه داده‌ها داده (Database Explorer)، کلیک راست کرده و گزینه Add Connection... را انتخاب کنید (شکل ۱۰-۶).



شکل ۱۰-۶- اتصال اضافه کردن

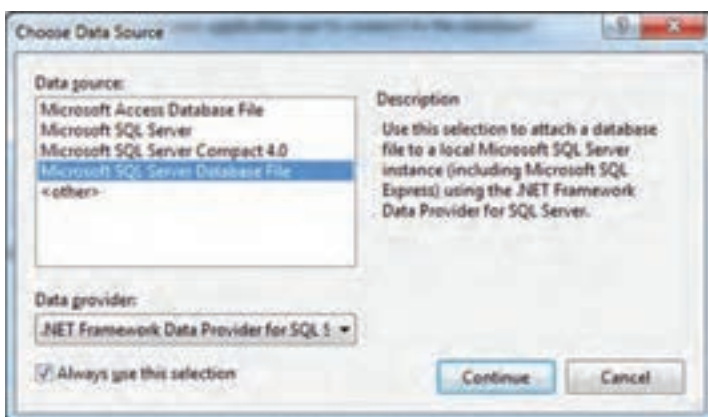
۵- در پنجره Add connection باید مشخصات پایگاه داده‌ها را مشخص کنیم (شکل ۶-۱۱).

۶- در قسمت Data Source نوع پایگاه داده را معین می‌کنیم. با کلیک بر روی دکمه Change در شکل ۶-۱۱ می‌توانید نوع پایگاه داده مورد نظر خود را انتخاب کنید.



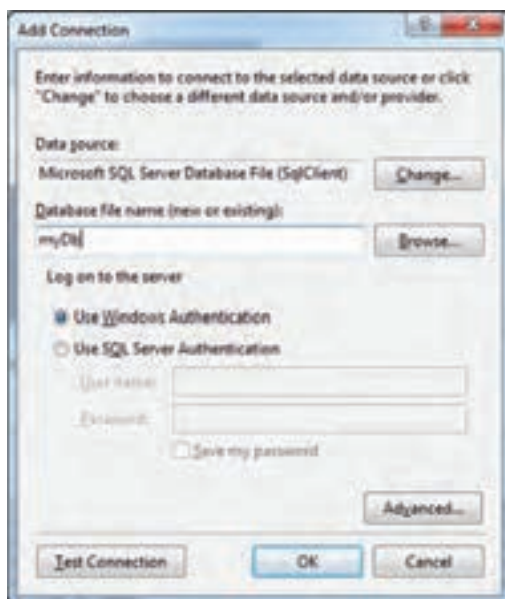
شکل ۶-۱۱- کادر اضافه کردن اتصال

۷- در این مثال پایگاه داده‌ای از نوع Microsoft Sql Server DataBase File را معین می‌کنیم (شکل ۶-۱۲).



شکل ۶-۱۲- انتخاب منبع داده

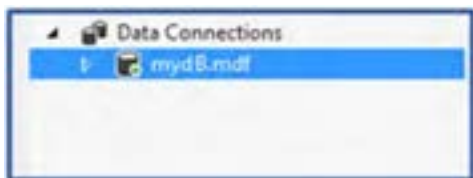
۸- پس از تعیین نوع پایگاه داده و زدن دکمه continue، باید مسیر و نام پایگاه را مشخص کنیم. برای آنکه فایل در محل پیش فرض (پوشه documents) ایجاد شود در کادر Database file name نام دلخواه را برای پایگاه داده می نویسیم، اما اگر می خواهید در مکان دیگری فایل را ذخیره نمایید، برای این منظور می توانید بر روی دکمه Browse کلیک کنید. در اینجا فایل در محل پیش فرض ایجاد شده است (شکل ۱۳-۶).



شکل ۱۳-۶- نام گذاری فایل پایگاه داده

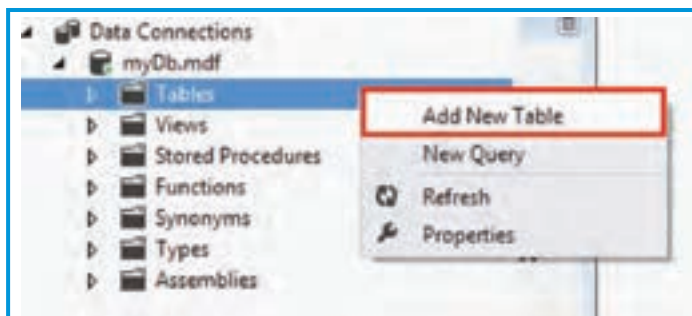
همچنین اگر فایل پایگاه داده را قبلاً ساخته اید می توانید در این مرحله، آن فایل را انتخاب کنید. در این صورت فایل جدید ایجاد نخواهد شد و از فایل انتخاب شده، استفاده خواهد شد. ۹- با زدن دکمه ok، پیامی نمایش داده خواهد شد که می گوید در این فایل پایگاه داده وجود ندارد. آیا می خواهید آن را ایجاد کنید؟ شما با زدن دکمه Yes فایل را خواهید ساخت.

۱۰- پس از ساخته شدن فایل در پنجره مرورگر پایگاه داده خواهیم دید یک اتصال به پایگاه داده جدید، ایجاد شده است (شکل ۱۴-۶).



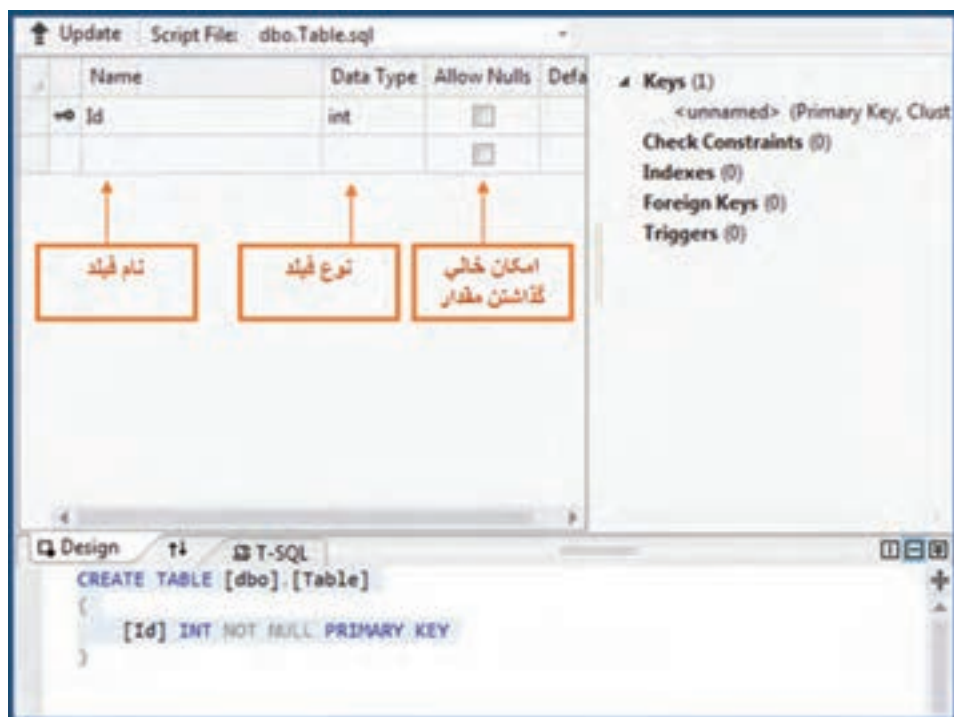
شکل ۱۴-۶- نمایش اتصال به پایگاه داده

۱۱- در مرحله بعد روی منلث کنار نام فایل کلیک کرده تا زیر شاخه‌های آن دیده شود. سپس بر روی شاخه Tables کلیک راست کرده و مانند شکل ۱۵-۶ گزینه Add New Table را انتخاب کنید.



شکل ۱۵-۶- اضافه کردن یک جدول جدید

۱۲- در پنجره‌ای که ظاهر می‌گردد می‌توانید نام و مشخصات جدول جدید را وارد کنید (شکل ۱۶-۶).



شکل ۱۶-۶- طراحی جدول

برای آنکه نوع یک فیلد را مشخص کنید باید اطلاعاتی را دربارهٔ انواع داده‌ای در Sql server بدانید. از آنجا که در اینجا هدف آشنایی اولیه شما با Sql server می‌باشد. تعداد محدودی از این انواع را در جدول ۲-۶ مرور می‌کنیم.

جدول ۲-۶

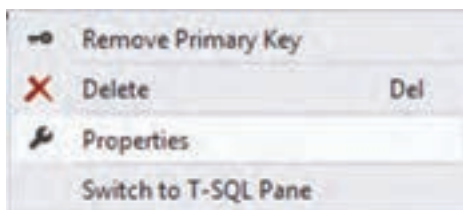
نوع	توضیح
char	برای فیلدهای متنی غیر یونیکد در نظر گرفته می‌شود. بنابراین اگر متنی که می‌خواهید وارد کنید انگلیسی است از این نوع استفاده کنید. همچنین این نوع مناسب برای رشته‌هایی با طول معین می‌باشد.
nchar	برای فیلدهای متنی یونیکد استفاده می‌شود. بنابراین برای وارد کردن متنی که حروف فارسی دارد این نوع استفاده می‌شود. این نوع نیز برای رشته‌های با طول معین، مناسب است.
nvarchar	برای فیلدهای متنی یونیکد این نوع در نظر گرفته می‌شود. مقدار بایت در نظر گرفته شده برای هر فیلد بستگی به تعداد حرف متن ورودی دارد. این نوع برای رشته‌های متنی با طول نامعین مناسب است.
int	برای مقادیر عددی صحیح، از این نوع استفاده می‌شود.

در اکثر مواقع برای فیلدهای متنی از نوع nvarchar استفاده می‌شود. زیرا ما از نوشتار فارسی استفاده می‌کنیم و همچنین در بیشتر موارد رشته متنی ورودی، طول ثابتی ندارد. همچنین مقدار حداکثر کاراکتر در نظر گرفته شده در فیلدهای متنی، جلوی نوع و در درون پراتز نوشته می‌شود. به عنوان مثال اگر برای یک فیلد متنی حداکثر ۲۰ حرف در نظر گرفته‌اید نوع آن به صورت nvarchar(20) در نظر گرفته می‌شود.

در لیست شماره تلفن‌ها، فیلدهای جدول ۶-۳ را برای جدول در نظر بگیرید.

جدول ۶-۳- تعیین نام و نوع فیلدها

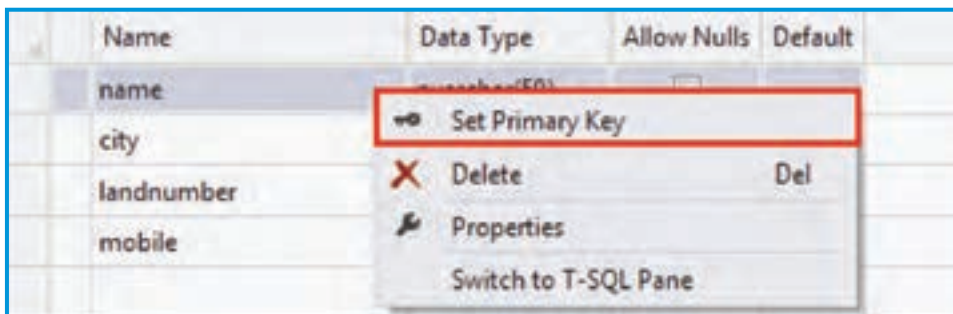
نام فیلد	نوع	کاربرد
name	nvarchar(50)	نام شخص
city	nvarchar(30)	شهر
landnumber	char(8)	شماره منزل
mobile	char(11)	شماره همراه



پس از وارد کردن فیلدها می‌توانید فیلد را که در ابتدا ایجاد شده بود با کلیک راست روی آن و زدن delete حذف کنید (شکل ۶-۱۷).

شکل ۶-۱۷- منوی زمینه فیلدها

سپس می‌توانید با کلیک راست روی فیلد name، گزینه Set Primary Key را بزنید. با این کار این فیلد به‌عنوان کلید اصلی شناخته خواهد شد. البته شما می‌توانستید فیلد id را از ابتدا تبدیل به name کنید (شکل ۶-۱۸).



شکل ۶-۱۸

سؤال: آیا فیلدهای مجزایی را می‌توان به عنوان کلید اصلی ایجاد و معرفی کرد؟

سؤال: آیا از ترکیب فیلدها می‌توان به عنوان کلید اصلی استفاده کرد؟

فیلدهای وارد شده را در شکل ۱۹-۶ می‌توانید مشاهده کنید.

Name	Data Type	Allow Nulls
name	nvarchar(50)	<input type="checkbox"/>
city	nvarchar(30)	<input checked="" type="checkbox"/>
landnumber	char(8)	<input checked="" type="checkbox"/>
mobile	char(11)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

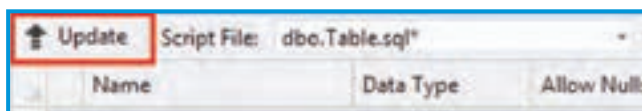
شکل ۱۹-۶- فیلدهای ایجادشده در محیط طراحی

۱۳- پس از تعیین فیلدها نام جدول را مشخص می‌کنیم. برای این کار در کد پایین پنجره طراحی، نام جدول را numbers تعیین کنید (شکل ۲۰-۶).

```
CREATE TABLE [dbo].[numbers] (  
    [name] NVARCHAR (50) NOT NULL,  
    [city] NVARCHAR (30) NULL,  
    [landnumber] CHAR (8) NULL,  
    [mobile] CHAR (11) NULL,  
    CONSTRAINT [PK_numbers] PRIMARY KEY ([name])  
);
```

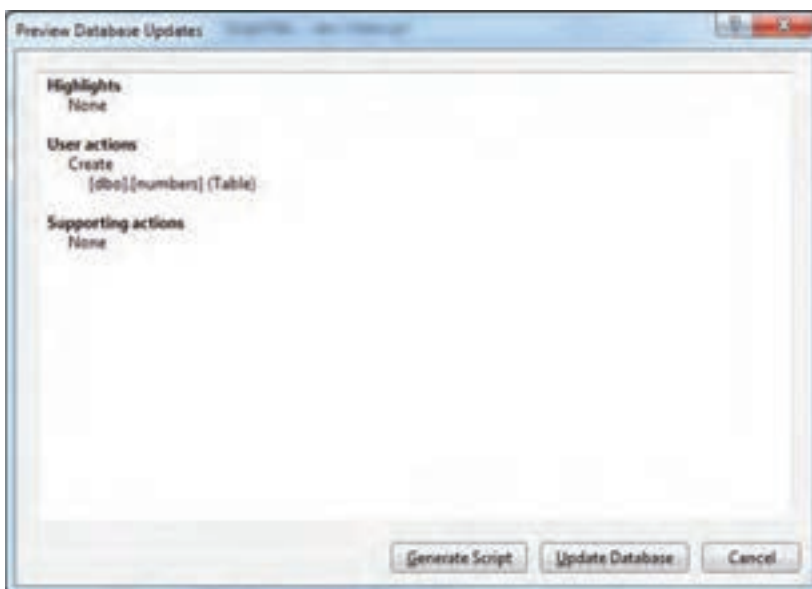
شکل ۲۰-۶- کد SQL برای ایجاد جدول

۱۴- اکنون نوبت بروزرسانی پایگاه داده و ثبت تغییرهای انجام شده است. ابتدا دکمه update در بالای کادر طراحی جدول را کلیک کنید (شکل ۲۱-۶).



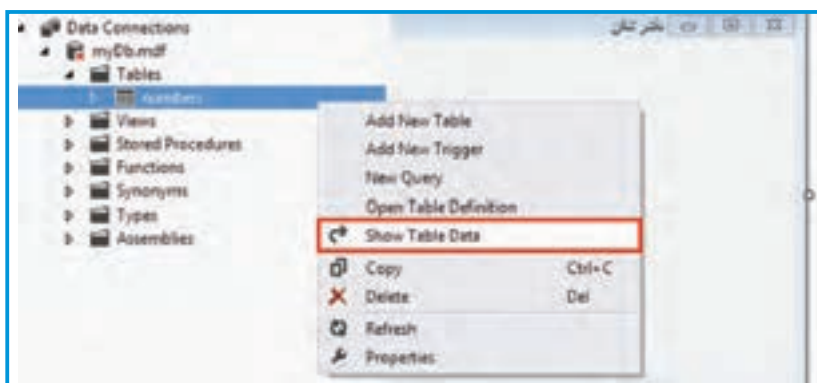
شکل ۲۱-۶- دکمه update

سپس با زدن دکمه Update Database در پنجره باز شده، جدول در پایگاه ثبت خواهد شد (شکل ۶-۲۲).



شکل ۶-۲۲- کادر پیش‌نمایش به‌روزرسانی پایگاه داده

۱۵- پس از ثبت تغییرات، برای اینکه بتوان تعدادی رکورد در جدول وارد کرد، به پنجره مرورگر پایگاه داده باز گردید و روی نام جدول numbers کلیک راست کرده و گزینه Show Table Data را انتخاب کنید (شکل ۶-۲۳).



شکل ۶-۲۳- اجرای فرمان نمایش داده‌های جدول

۱۶- با کلیک بر روی گزینه Show Table Data، می‌توانید عمل ورود اطلاعات را انجام دهید. علامت * در ابتدای سطر، نشانه رکورد جدید است. از سطر جدید، شروع کرده و اطلاعات خود و چند تن از دوستان خود را بنویسید. لزومی ندارد اطلاعات تمام افراد را وارد کنید، بعداً نیز می‌توانید اطلاعاتی را مانند شکل ۲۴-۶ به جدول اضافه کنید. آنچه در اینجا آورده شده فقط برای نمونه آورده شده است اما شما سعی کنید اطلاعات با معنی وارد کنید.

	name	city	landnumber	mobile
	حسن نادری	تهران	11111111	09123456789
	حسن نوروزیان	NULL	NULL	NULL
	محمد قاسمی	شیراز	2222222	09121111111
**	NULL	NULL	NULL	NULL

شکل ۲۴-۶- رکوردهای واردشده در جدول

۱۷- طراحی واسط کاربری و اتصال به پایگاه داده: پس از ایجاد پایگاه و جدول اطلاعاتی می‌توانیم به سراغ طراحی واسطه کاربری (فرم) برویم. بنابراین پنجره طراحی فرم را باز کرده تا با اضافه کردن واسطه‌ها اتصال به پایگاه داده را برقرار کرده و اطلاعات مورد نظر را روی فرم نمایش دهیم.

۱۸- ویژگی‌های فرم را طبق جدول ۴-۶ تنظیم کنید:

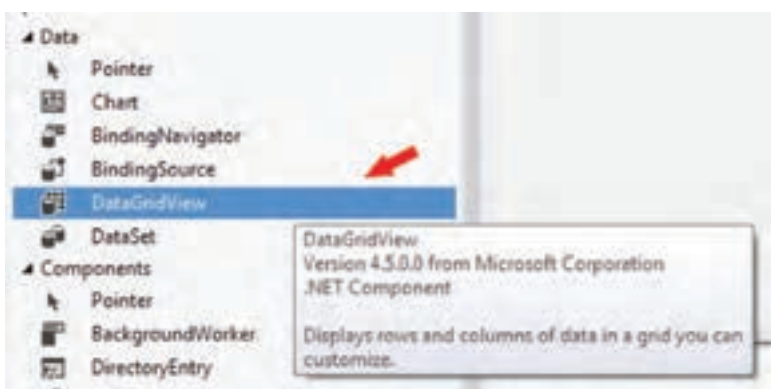
جدول ۴-۶- ویژگی‌های فرم

Form		شیء
ویژگی		مقدار
Name		Form1
Text		دفتر تلفن
RightToLeft		Yes
Size	Width	600
	Height	300

۱۹- برای اینکه بتوان اطلاعات موجود در یک جدول پایگاه داده را، به صورت یک جدول نمایشی در روی فرم مشاهده کرد از ابزار DataGridView استفاده می‌کنیم. برای این منظور به جعبه ابزار Data مراجعه کرده و کنترل DataGridView را با ویژگی‌های جدول ۵-۶ به فرم اضافه کنید.

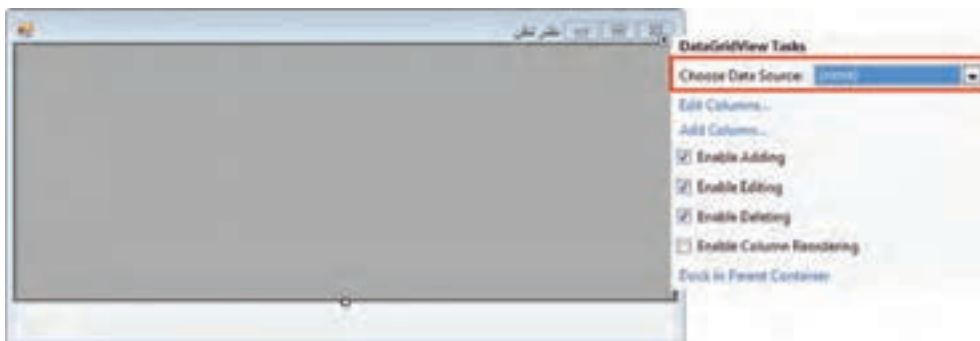
جدول ۵-۶- ویژگی‌های Data Gridview

DataGridView		شیء
ویژگی		مقدار
Name		dataGridView1
RightToLeft		Yes
Dock		top
Size	Height	227



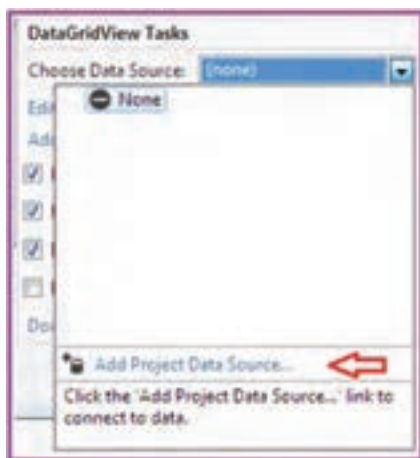
شکل ۲۴-۶

۲۰- با اضافه کردن کنترل DataGridView، کادری در کنار آن اضافه می‌شود که در آن باید منبع اطلاعاتی این جدول نمایشی را معین کنید. همان طور که در ابتدای این فصل بیان شد یک جدول نمایشی نمی‌تواند مستقیماً به پایگاه داده متصل شود بلکه باید از طریق واسطه‌های مختلف که توضیح داده شد، عمل اتصال به داده‌ها انجام شود (شکل ۲۵-۶).



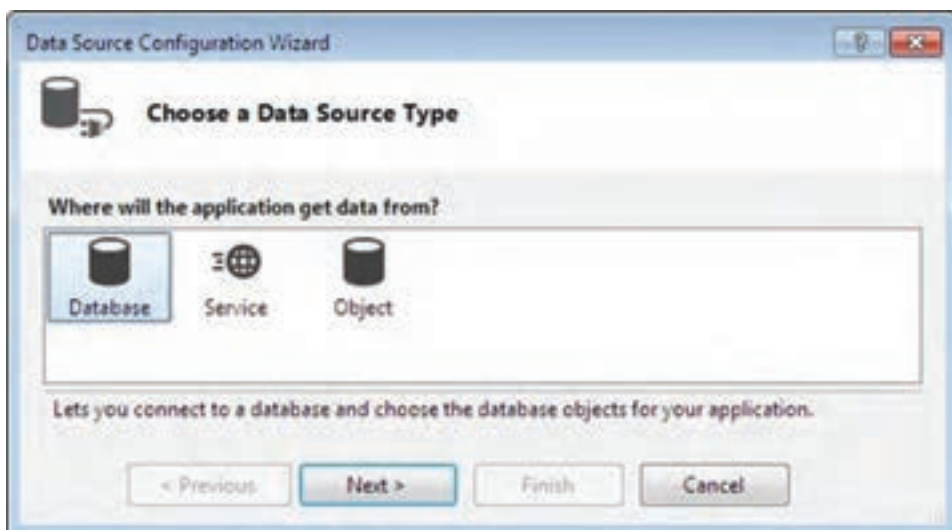
شکل ۲۵-۶

۲۱- برای ایجاد واسطه‌های مورد نیاز، بر روی گزینه Choose Data Source کلیک کنید و سپس بر روی گزینه Add Project Data Source... کلیک کنید (شکل ۶-۲۶).



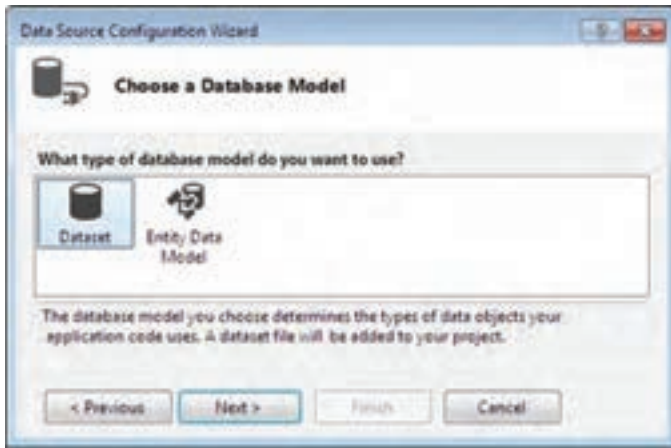
شکل ۶-۲۶- اضافه کردن منبع داده به پروژه

۲۲- چند مرحله را باید به ترتیب دنبال کنیم تا واسطه‌ها ساخته شوند. ابتدا گزینه Database را به عنوان نوع منبع اطلاعات در شکل ۶-۲۷ مشخص می‌کنیم.



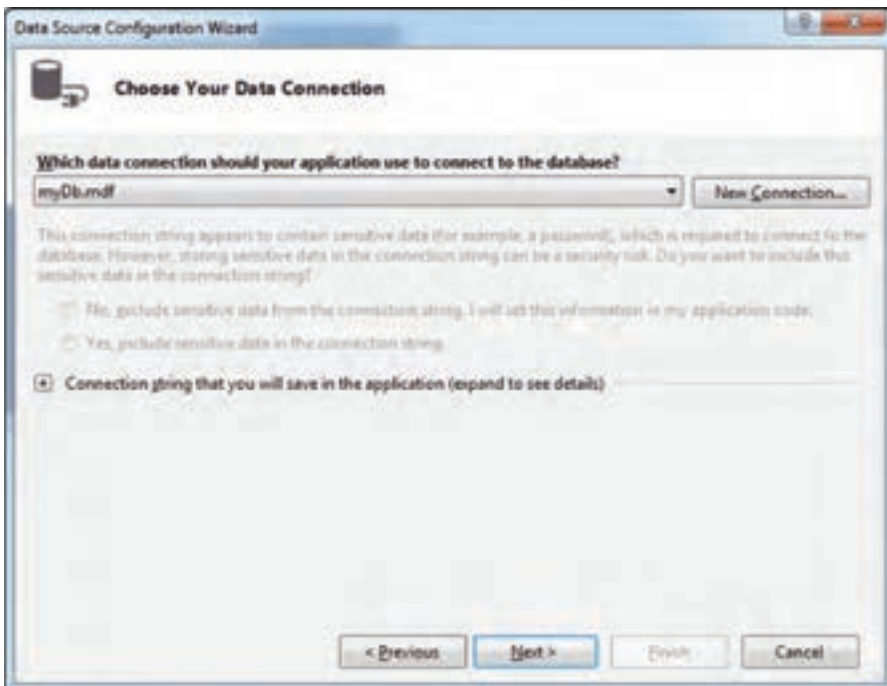
شکل ۶-۲۷- انتخاب نوع منبع داده

۲۳- بر روی دکمه Next کلیک کنید. در شکل ۶-۲۸، گزینه Dataset را به عنوان نمایش دهنده پایگاه داده در حافظه تعیین می کنیم.



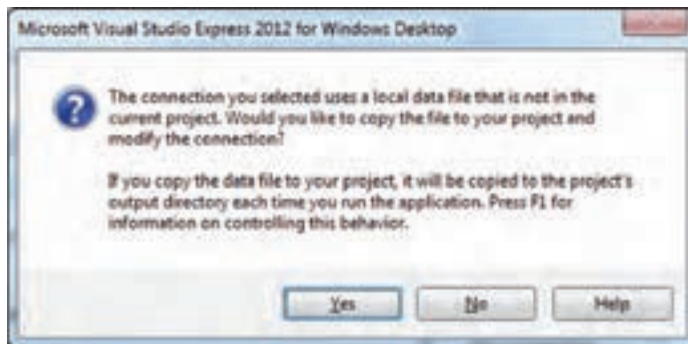
شکل ۶-۲۸- انتخاب مدل استفاده از پایگاه داده

۲۴- به مرحله بعدی می رویم و نام اتصال به پایگاه داده را مشخص می کنیم (شکل ۶-۲۹).



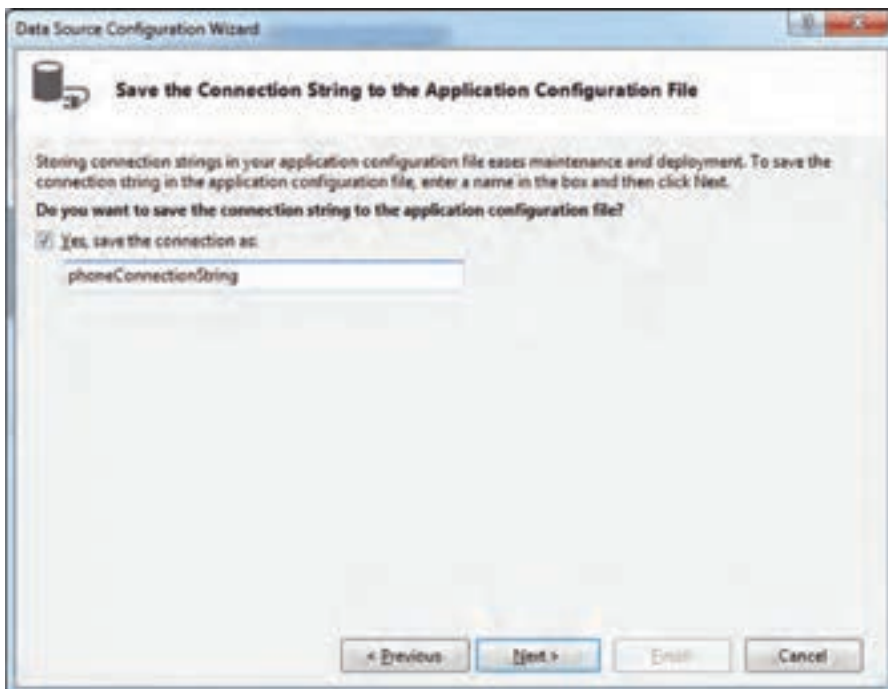
شکل ۶-۲۹- انتخاب اتصال

با زدن دکمه Next کادری نمایش داده می‌شود. دکمه Yes را بزنید. با این کار فایل پایگاه داده در پروژه کپی می‌شود و جزئی از پروژه محسوب می‌شود. بنابراین به صورت خودکار با ساختن فایل خروجی، در مسیر پوشه خروجی، فایل پایگاه داده نیز کپی می‌شود (شکل ۶-۳۰).



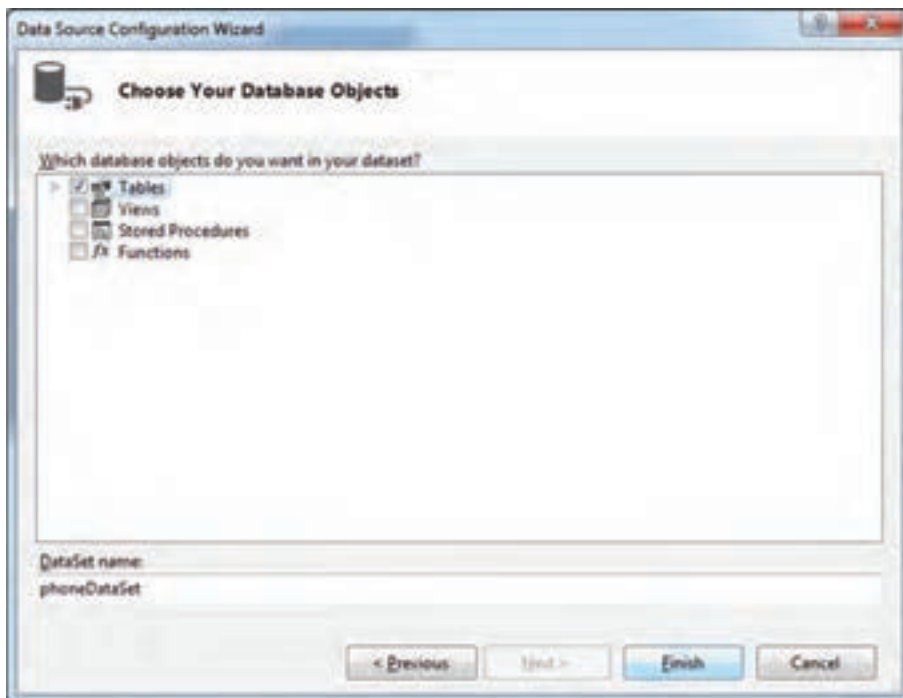
شکل ۶-۳۰

۲۵- در مرحله بعدی تنظیم‌های انجام شده در برنامه ذخیره می‌شود. می‌توانیم نام اتصال را به صورت دلخواه تعیین نماییم. نام را `phoneConnectionString` انتخاب می‌کنیم (شکل ۶-۳۱).



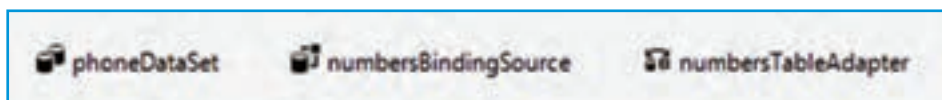
شکل ۶-۳۱- تعیین نام اتصال

۲۶- در مرحله بعد، جدول یا جدول‌هایی را که می‌خواهیم از طریق Dataset به آن دسترسی داشته باشیم (تمام جداول (Tables)) با فعال کردن کادر تیک مشخص می‌کنیم. همچنین نام dataset را نیز می‌توانیم به صورت دلخواه phoneDataSet تعیین کنیم. پس از انجام کارهای لازم، دکمه Finish را به منزله پایان عملیات کلیک کنید (شکل ۶-۳۲).



شکل ۶-۳۲- انتخاب جدول‌های موجود در پایگاه داده برای ورود در Dataset

۲۷- به این ترتیب تمام واسطه‌های مورد نیاز به طور خودکار و بدون برنامه‌نویسی ساخته می‌شود. اگر قسمت پایین صفحه در پنجره طراحی فرم را مشاهده کنید، واسطه‌های مورد نیاز ساخته شده‌اند (شکل ۶-۳۳). phoneDataSet نمایش پایگاه داده در حافظه RAM کامپیوتر است. numbersBindingSource واسطه بین DataGridview و Dataset است و numbersTableAdapter نیز واسطه بین پایگاه داده فیزیکی (ذخیره شده بر روی هارد دیسک) و نمایش پایگاه در حافظه (Dataset) است.



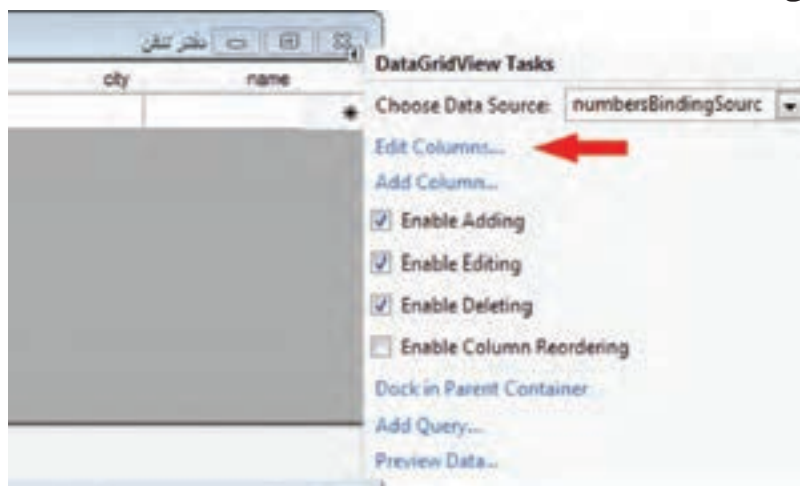
شکل ۶-۳۳- واسطه‌های اضافه‌شده به فرم

۲۸- اکنون می‌توانید برنامه را اجرا کنید. همان طور که در شکل ۶-۳۴ مشاهده می‌کنید، اطلاعات وارد شده در جدول numbers پایگاه داده در داخل جدول نمایشی بر روی فرم دیده می‌شود.

mobile	landnumber	city	name
09123456789	11111111	تهران	حسن نادری
09121111111	2222222		حسن نوروزیان
			محمد قاسمی

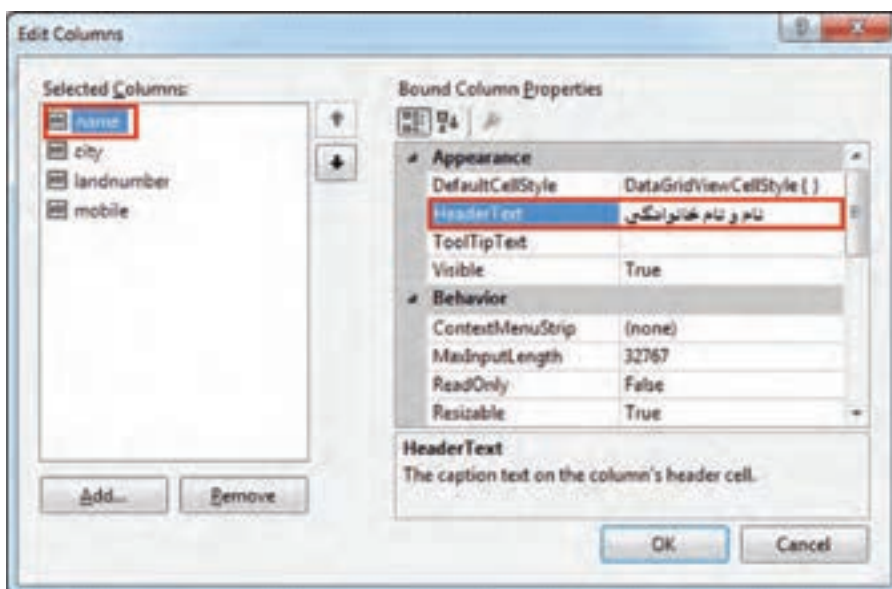
شکل ۶-۳۴

۲۹- همانطور که در شکل ۶-۳۴ ملاحظه می‌کنید، آنچه که در نمایش دفتر تلفن چندان دلچسب نیست عناوین ستون‌ها است که به زبان انگلیسی است و همچنین جهت نمایش اطلاعات است که از چپ به راست است و در زبان فارسی چندان مناسب نیست. برای تغییر در ستون‌ها، با استفاده از گزینه Edit Columns در منوی شکل ۶-۳۵ می‌توانید ستون‌ها را مطابق با میل خود تغییر داده و به اصطلاح سفارشی کنید.



شکل ۶-۳۵- فرمان نمایش کادر ویرایش ستون‌ها

۳۰- در شکل ۶-۳۶ ملاحظه می کنید که چگونه می توانید عنوان ستون ها را فارسی کنید.



شکل ۶-۳۶- تغییر متن عنوان ستون

آنچنان که در شکل ۶-۳۶ نمایش داده شد عنوان ستون name تغییر می کند. این عملیات را برای سایر ستون ها نیز انجام دهید.

همچنین می توانید در صورت لزوم، عرض ستون را توسط ویژگی Width تغییر دهید.

۳۱- اکنون برنامه را اجرا کنید و دفتر تلفن را مشاهده کنید. اگر بر روی عنوان ستون ها کلیک کنید، می توانید لیست را براساس آن مرتب شده مشاهده کنید.

۳۲- به سطر آخر جدول که با علامت * مشخص شده است مراجعه کرده و اطلاعات یکی از دوستان خود را در آن وارد کنید. در واقع یک رکورد جدید ایجاد کنید. سپس بین رکوردها حرکت کنید، به اولین رکورد بروید و به آخرین رکورد بازگردید.

۳۳- اکنون از برنامه خارج شوید و دوباره برنامه را اجرا کنید. به دفتر تلفن توجه کنید. آیا رکوردی که اضافه کردید، وجود دارد؟ چرا در پایگاه داده ذخیره نشده است؟

۳۴- هر گاه اطلاعات درون یک Dataset را تغییر دادید، برای اعمال تغییرات بر روی پایگاه داده فیزیکی، باید متد Update شیء TableAdapter را فراخوانی کنید.

۳۵- بدین منظور یک دکمه در جای مناسبی از فرم قرار دهید و ویژگی‌های آن را به صورت جدول ۶-۶ تنظیم کنید :

جدول ۶-۶- ویژگی‌های دکمه

دکمه	Button
مقدار	ویژگی
save	Name
ثبت تغییرات	Text

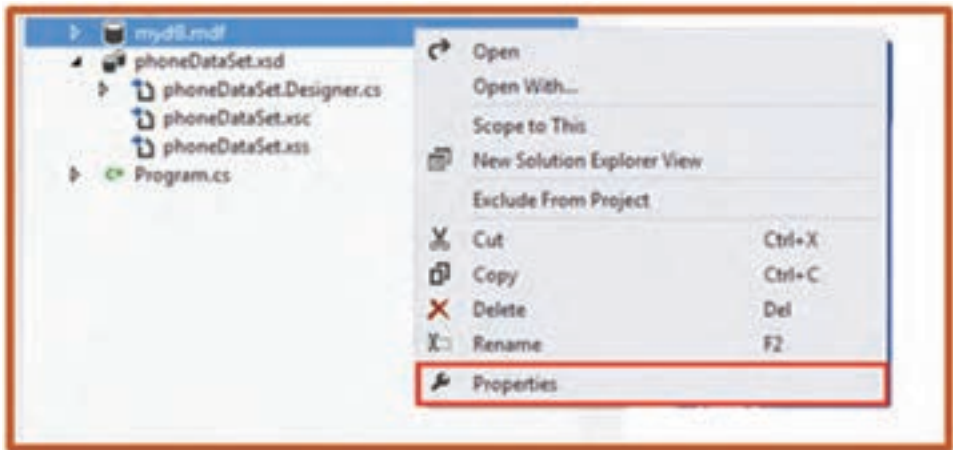
۳۶- در متد EH مربوط به رویداد کلیک دکمه ، دستور زیر را بنویسید.
این کد تغییرات داده شده در جدول numbers از جدول phoneDataSet را به جدول numbers در پایگاه داده منتقل می کند.

```
private void save_Click(object sender, EventArgs e)
{
    numbersTableAdapter.Update(phoneDataSet.numbers);
}
```

۳۷- پروژه را اجرا کنید و رکوردهای جدیدی به دفترچه تلفن اضافه کنید و آن را ذخیره نمایید.
اکنون برنامه را ببندید و دوباره پروژه را اجرا نمایید. آنچه مشاهده خواهید کرد همان رکوردهای اولیه می باشد. دلیل این اتفاق آن است که فایل پایگاه داده با اجرا و Debug توسط ویژوال استودیو فایل پایگاه داده، در محل فایل اجرایی کپی می شود و در واقع نرم افزار ایجاد شده، با این فایل پایگاه داده کار می کند. انجام عمل کپی فایل پایگاه داده به محل فایل اجرایی (پوشه Debug یا Release) در هر بار اجرا صورت می گیرد، بنابراین فایل قبلی که حاوی آخرین داده ها بوده است حذف شده و فایل قبلی جایگزین می شود.

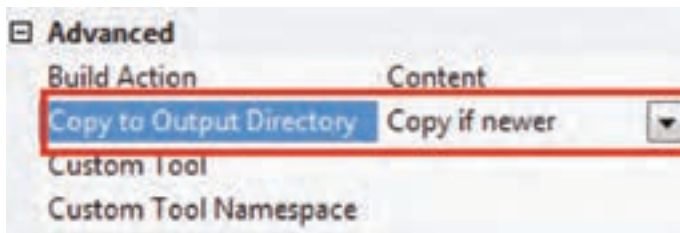
برای آنکه عملیات کپی فقط در صورتی انجام شود که فایل پایگاه داده ساخته نشده باشد یا از فایل ابتدایی قدیمی تر باشد به صورت زیر عمل کنید :

روی فایل پایگاه داده در پنجره Solution Explorer کلیک راست کنید و گزینه properties را بزنید (شکل ۶-۳۷).



شکل ۶-۳۷- گزینه Properties برای پایگاه داده

سپس مقدار گزینه Copy to Output Directory را به Copy if newer تغییر دهید (شکل ۶-۳۸).



شکل ۶-۳۸- تنظیم ویژگی Copy to output Directory

نکته

در برنامه دفتر تلفن، در پنجره Solution Explorer روی Form1 کلیک راست و پنجره کدنویسی را فعال کنید. به متد Form1_Load() توجه کنید. دستور فراخوانی متد Fill شیء numbersTableAdapter را مشاهده می کنید. به وسیله فراخوانی متد Fill() است که

اطلاعات درون پایگاه داده به داخل حافظه و در شیء DataSet قرار می‌گیرد. (واژه Fill به چه معنا است؟ آیا با عملکرد آن متناسب است؟)

```
private void Form1_Load (object sender, EventArgs e)
{
    // TODO: This line of code loads data into the phoneDataSet.
    numbers table.
    //          You can move, or remove it, as needed.
    this.numbersTableAdapter.Fill (this.phoneDataSet.numbers);
}
```

کار در کارگاه ۲

مثال ۲-۶ : می‌خواهیم فرمی داشته باشیم که اطلاعات وارد شده در دفتر تلفن را برای نام مورد جستجوی ما نمایش دهد.

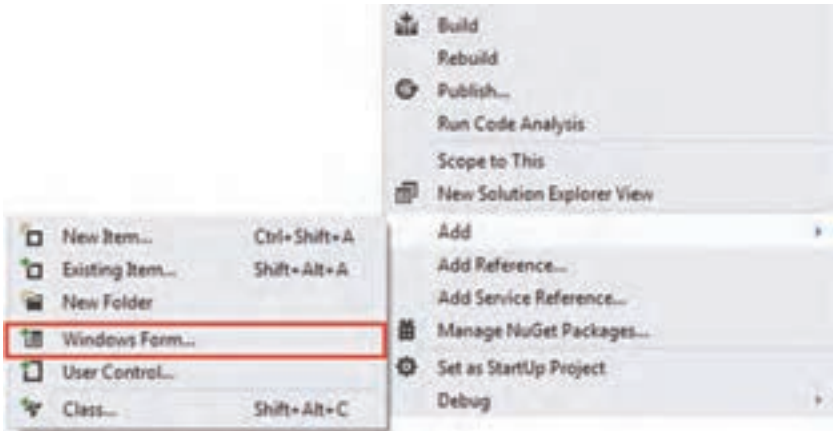
الگوریتم و روش انجام کار : فرم جستجو را ایجاد می‌کنیم تا با زدن یک دکمه در فرم اصلی، نمایش داده شود. در فرم جستجو، نام شخص در کادر متنی نوشته می‌شود و با زدن دکمه جستجو، اگر رکوردی برای آن فرد ثبت شده باشد اطلاعات تماس در چند برجسب نمایش داده می‌شود (شکل ۳۹-۶).



شکل ۳۹-۶ فرم جستجو در پروژه دفتر تلفن

طراحی واسط کاربری

۱- برای افزودن فرم جدید باید ابتدا کلاس فرم جدیدی تعریف کرد. این کار با استفاده از امکانات ویژوال استودیو بسیار آسان است. روی نام پروژه کلیک راست نمایید تا منوی آن نمایش داده شود. گزینه Windows Form را از منوی Add کلیک نمایید (شکل ۶-۴۰).



شکل ۶-۴۰ افزودن فرم جدید

۲- با کلیک این گزینه، کادر افزودن بخش جدید باز می‌شود که به صورت خودکار گزینه Windows Form انتخاب شده است.

۳- با وارد کردن نام فرم جدید و زدن دکمه Add فرم جدید ایجاد خواهد شد.

۴- در پروژه دفتر تلفن، فرمی با مشخصات جدول ۶-۷ اضافه می‌کنیم (شکل ۶-۴۰).

جدول ۶-۷- ویژگی‌های فرم

فرم		Form
ویژگی	مقدار	
Name	Form2	
Text	جستجو	
FormBorderStyle	FixedToolWindow	
Size	Width	324
	Height	317

۵- در این فرم یک کادر متنی، یک دکمه و سه برچسب ایجاد می‌کنیم و در جای مناسب قرار می‌دهیم:

جدول ۶-۸- ویژگی‌های دکمه

دکمه Button	
ویژگی	مقدار
Name	search
Text	جستجو

جدول ۶-۹- ویژگی‌های کادر متنی

کادر متنی TextBox	
ویژگی	مقدار
Name	nameBox
Text	
RightToLeft	Yes

جدول ۶-۱۰- ویژگی‌های برچسب‌ها

برچسب Label			
ویژگی	مقدار	مقدار	مقدار
Name	cityLb	landNumberLb	mobileLb
RightToLeft	Yes	Yes	Yes
AutoSize	False	False	False
BackColor	white	white	white
TextAlign	MiddleCenter	MiddleCenter	MiddleCenter
Size	Width	206	206
	Height	33	33

جدول ۶-۱۱- ویژگی‌های دکمه

دکمه Button	
ویژگی	مقدار
Name	showSearchForm
Text	جستجوی اطلاعات

همچنین چهار برچسب دیگر برای برچسب‌های نام، شهر، شماره منزل و شماره همراه ایجاد کرده و در جای مناسب قرار می‌دهیم.

۶- در فرم اصلی (Form1) نیز یک دکمه ایجاد می‌کنیم تا با زدن آن، فرم جستجو باز شود (شکل ۶-۴۱).

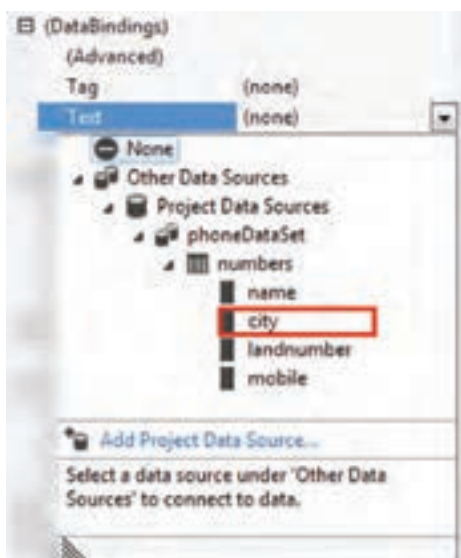
نام و نام خانوادگی	شهر	شماره منزل	شماره همراه
حسن نادری	تهران	11111111	09123456789
حسن نوروزیان			
محمد قاسمی	شیراز	2222222	09121111111

شکل ۴۱-۶- دکمه جستجوی اطلاعات در فرم اصلی

اتصال کنترل‌ها به پایگاه داده

۷- اکنون طراحی واسط کاربری به پایان

رسیده است و نوبت اتصال به پایگاه داده است. برچسب cityLbl را انتخاب نمایید. در پنجره properties و در گروه Data با زدن علامت + گزینه (DataBindings) آن را باز نمایید. سپس مطابق شکل ۴۲-۶ با کلیک روی علامت مثلث شکل، کادر تنظیم آن را باز کنید و برای گزینه Text، مسیر را مطابق شکل طی کرده و فیلد city را انتخاب نمایید.



شکل ۴۲-۶

با این عملیات ما از phoneDataSet

موجود در پروژه استفاده می‌کنیم و ویژگی متن (Text) برچسب cityLbl را به فیلد city در

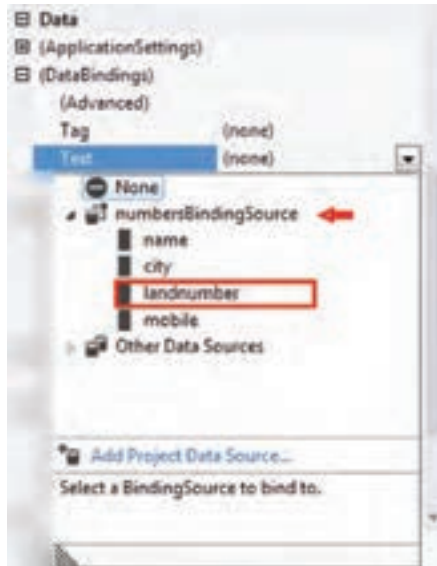
جدول numbers مقید می‌کنیم. با انجام این کار، یک شیء BindingSource و یک TableAdapter و همچنین خود DataSet به صورت خودکار اضافه می‌شود (شکل ۴۳-۶).



شکل ۶-۴۳- واسطه‌های اضافه‌شده به فرم جستجو

۸- اکنون چون یک BindingSource ایجاد شده است برای متصل کردن ویژگی متن برچسب شماره منزل (landNumberLbl) به صورت زیر عمل می‌کنیم:

در کادر تنظیم گزینه Text از بخش (DataBindings)، گزینه numbersBindingSource را باز کنید و فیلد landnumber را انتخاب نمایید.



شکل ۶-۴۴

برای شماره همراه (mobileLbl) نیز به همین روش عمل کرده و آن را به فیلد mobile مقید کنید.

کد نویسی:

۹- اکنون کد متد EH رویداد کلیک دکمه showSearchForm در فرم اصلی (Form1) را

به صورت زیر می نویسیم :

```
private void showSearchForm_Click(object sender, EventArgs e)
{
    Form2 f2 = new Form2();
    f2.ShowDialog();
}
```



۱۰- برنامه را اجرا کنید و با دکمه «جستجوی اطلاعات» در فرم اصلی حاصل کار را بررسی کنید.

شکل ۴۵-۶

همان طور که می بینید مشخصات اولین رکورد در برچسب‌ها آمده است. این موضوع به دلیل آن است که ما برچسب‌ها را به طریقی که پیش‌تر آمد به فیلدها مقید کرده‌ایم. اما می‌خواهیم فقط با عمل جستجو، اطلاعات فرد نمایش داده شود. بنابراین لازم است که مقیدسازی این برچسب‌ها در ابتدای کار به حالت تعلیق در بیاید. برای این کار در متد سازنده Form2 کد زیر را اضافه می‌نماییم :

```
public Form2()
{
    InitializeComponent();
    numbersBindingSource.SuspendBinding();
}
```

متد `SuspendBinding` در کد بالا، مقیدسازی را برای `numbersBindingSource` به حالت تعلیق در می‌آورد. اگر اکنون برنامه را اجرا کنیم خواهید دید که متن برجسب‌ها خالی است. **۸-** برای عمل جستجو، متد `EH` رویداد کلیک دکمه را به صورت زیر بنویسید.

```
private void search_Click(object sender, EventArgs e)
{
    int find = numbersBindingSource.Find("name", nameBox.Text);
    if (find >= 0)
    {
        numbersBindingSource.ResumeBinding();
        numbersBindingSource.Position = find;
    }
    else
        numbersBindingSource.SuspendBinding();
}
```

دستور `numbersBindingSource.Find("name", nameBox.Text)` عمل جستجو را در رکوردها می‌پردازد و شماره اولین رکوردی که فیلد `name` آن برابر با متن درون کادر متنی `nameBox` باشد برگشت می‌دهد. این شماره از صفر شروع می‌شود. یعنی اگر نام و نام‌خانوادگی اولین رکورد همان چیزی باشد که در کادر متنی `nameBox` وارد شده است، آن‌گاه مقدار صفر را برمی‌گرداند. این مقدار برگشتی را درون یک متغیر به نام `find` قرار می‌دهیم تا بعد از آن استفاده کنیم. اکنون سؤال اینجاست که اگر رکوردی پیدا نشد که مطابق جستجوی ما باشد چه عددی برگشت داده می‌شود؟ جواب این است که در این صورت عدد برگشتی عدد `۱-` می‌باشد. از همین نکته می‌توان استفاده کرد تا عملیات مورد نظر را در صورت پیدا شدن رکورد یا پیدا نشدن آن، طرح ریزی کرد. شرط `if (find >= 0)` بررسی می‌کند که اگر نتیجه جستجو، بزرگتر یا مساوی `0` باشد (رکوردی مطابق با جستجو پیدا شود) دستورهای مناسب را انجام دهد. این دستورها به صورت زیر می‌باشند:

```
numbersBindingSource.ResumeBinding();
numbersBindingSource.Position = find;
```

دستور اول حالت مقید سازی داده‌ها را از سر می‌گیرد و به عبارتی آن را دوباره فعال می‌کند. چرا که پیش از این ممکن است در حالت تعلیقی بوده باشد. دستور دوم، موقعیت رکورد جاری در لیست رکوردها را مشخص می‌کند. در هنگام مقاردهی، اطلاعات رکورد جاری در کنترل‌های متصل استفاده می‌شود. در انتها کد

else

```
numbersBindingSource.SuspendBinding();
```

مشخص می‌کند که در صورت درست نبودن شرط یعنی پیدا نشدن هیچ رکوردی، مقیدسازی داده‌ها به حالت تعلیق درآید تا بدین ترتیب متن برچسب‌های مورد نظر، خالی شود. اکنون برنامه را اجرا کنید و با وارد کردن نام شخص، حاصل کار را بررسی کنید (شکل ۴۶-۶).

شکل ۴۶-۶

خودآزمایی فصل ششم

الف) درستی یا نادرستی هر عبارت را تعیین کنید.

- ۱- اطلاعات درون یک پایگاه داده را در برنامه‌های ویژوال استودیو نمی‌توان ویرایش کرد.
- ۲- در جستجو توسط متد find از واسط BindingSource، اگر رکوردی پیدا نشود مقدار صفر برگردانده می‌شود.
- ب) جاهای خالی را با عبارت مناسب پر کنید.
- ۳- هر یک از اجزای تشکیل دهنده یک رکورد را می‌نامند.
- ۴- یک کپی از داده‌های پایگاه داده در حافظه RAM کامپیوتر، به وسیله شیء ایجاد می‌شود.
- ۵- ورود اطلاعات درون پایگاه داده به داخل حافظه و در شیء DataSet، به وسیله فراخوانی متد انجام می‌گیرد.
- ۶- برای از سر گرفتن مقید سازی داده‌ها از متد استفاده می‌شود.
- ۷- برای تعیین موقعیت رکورد جاری در لیست رکوردها در واسط BindingSource از متد استفاده می‌شود.
- ج) به سؤالات زیر پاسخ دهید.
- ۸- مزیت‌های استفاده از پایگاه داده به جای فایل برای ذخیره داده‌ها در حافظه جانبی را بنویسید.
- ۹- جدول زیر را تکمیل نمایید.

کاربرد	ابزار پایگاه داده
	DataSet
	DataGridView
	TableAdapter
	BindingSource

- ۱۰- به چه دلیل در تنظیمات پایگاه داده مقدار ویژگی Copy to Output Directory را به Copy if newer تغییر می‌دهید؟
- ۱۱- در ارتباط با پایگاه داده به چه دلیل متد SuspendBinding فراخوانی می‌شود؟

تمرینات برنامه‌نویسی فصل ششم

۱- یک برنامه برای نگهداری اطلاعات گوشی‌های موبایل ایجاد کنید. این برنامه از DataGridView برای نمایش و ثبت و ویرایش در جدول پایگاه داده استفاده می‌کند. فیلدهای اطلاعاتی که در مورد هر گوشی اهمیت دارد و باید نگهداری شود را پیدا کنید و در جدول به کار بگیرید.

۲- پروژه‌ای ایجاد نمایید که شامل پایگاه داده‌ای برای نگهداری مشخصات دانش‌آموزان شامل کد دانش‌آموزی، نام کامل (نام و نام خانوادگی) و سه نمره درسی باشد. توسط DataGridView اطلاعات دانش‌آموزان را وارد نمایید. در فرم جستجو کد دانش‌آموز وارد شود و در صورت پیدا شدن دانش‌آموز، نام و میانگین نمرات وی نمایش داده شود.

پایان پروژه

در بخش پایانی پروژه، ضمن عرض خدا قوت به شما هنرجویان و هنرآموزان عزیز، پایگاه داده‌ها را برای واژگان تشکیل می‌دهیم.

مرحله ششم:

در مرحله پایانی تکمیل پروژه، راهکار استفاده از پایگاه داده برای نگهداری اطلاعات واژگان را تمرین می‌کنیم.

۱- آرایه‌های words و meaning را حذف کنید. همچنین عملیات مربوط به جستجو نیز باید حذف شود.

۲- یک فایل پایگاه داده ایجاد کنید.

۳- جدولی به نام words به صورت زیر درون پایگاه داده جدید ایجاد نمایید.

Name	Data Type	Allow Nulls
word	nvarchar(150)	<input type="checkbox"/>
description	nvarchar(1000)	<input type="checkbox"/>

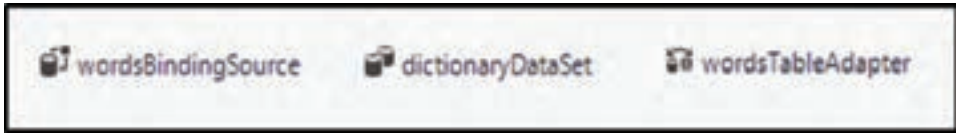
۴- یک کنترل DataGridView در فرم RegisterAndEditForm اضافه کنید و آن را به جدول words مقید کنید.

۵- یک دکمه به نام save و با متن «ذخیره» به فرم RegisterAndEditForm اضافه نمایید.

همچنین ستون‌های آن را به «واژه» و «معنی و توضیح» تغییر دهید.



۶- در فرم اصلی، کادر معنی و توضیح (description) را به فیلد description در جدول words مقید کنید. در حین عمل اشیاء DataSet، BindingSource و TableAdapter به صورت زیر ساخته می‌شود.



۷- برای اینکه مقید شدن کنترل در ابتدا به تعلیق در آید و مقداری در کادر متنی description نمایش داده نشود کد زیر را به متد سازنده Form1 اضافه کنید.

```
public form1()
{
    InitializeComponent ();
    wordBindingSource. SuspendBinding (); ←
}
```

کدی که در متد EH رویداد کلیک دکمه جستجو نوشته اید را حذف کنید و کد زیر را جایگزین کنید :

```
private void search_Click (object sender, EventArgs e)
{
    Int find = wordsBindingSource. Find ("word",searchword.Text);
    if (find > 0)
    {
        wordsBindingSource. ResumeBinding ();
        wordsBindingSource. position = find;
    }
    else
    {
        wordsBindingSource. SuspendBinding ();
        MessageBox.Show ("پیدا نشد");
    }
}
```

در این کد ابتدا نتیجه جستجو توسط متد Find برای واسط wordsBindingSource در متغیر find ذخیره می‌شود. سپس این نتیجه مورد بررسی قرار می‌گیرد. اگر رکوردی پیدا شده باشد، مقیدسازی ادامه پیدا می‌کند و کادر متنی description، فیلد توضیح (description) مربوط به رکورد پیدا شده را نمایش می‌دهد. در غیر این صورت مقیدسازی به تعلیق در می‌آید و پیام متناسب نمایش داده می‌شود.

واژگان و اصطلاحات انگلیسی فصل ششم

ردیف	واژه انگلیسی	معنی به فارسی
۱	Adapter	
۲	Allow	
۳	Binding	
۴	Compact	
۵	Connection	
۶	Continue	
۷	Data base	
۸	Data type	
۹	Edition	
۱۰	Express	
۱۱	Fill	
۱۲	Find	
۱۳	Grid	
۱۴	Primary	
۱۵	Set	
۱۶	Source	
۱۷	Table	
۱۸	Update	
۱۹	View	