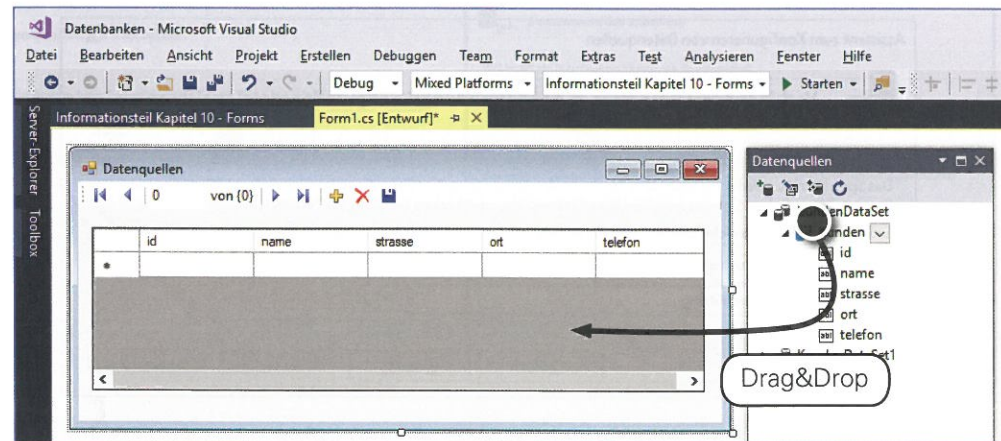
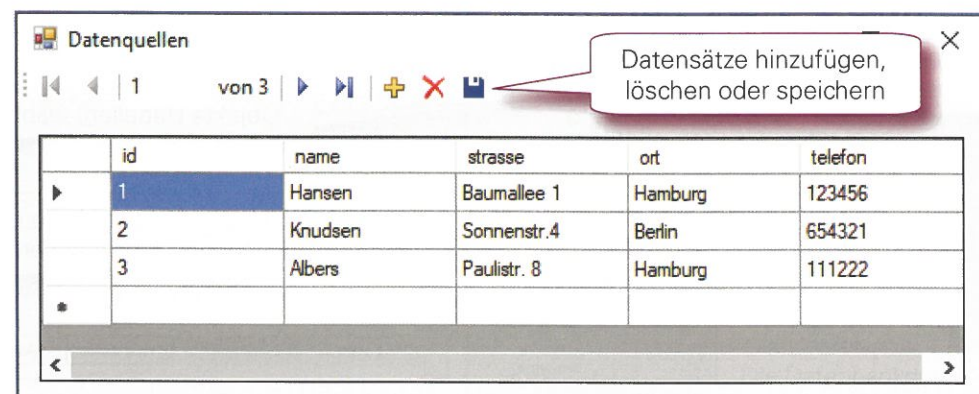


Per *Drag&Drop* kann die Tabelle Kunden nun in der gewünschten Ansicht (hier *DataGridView*) auf die Form gezogen werden. Der Assistent legt automatisch das entsprechende Steuerelement und die Verbindung zur Datenbank und der Tabelle an:



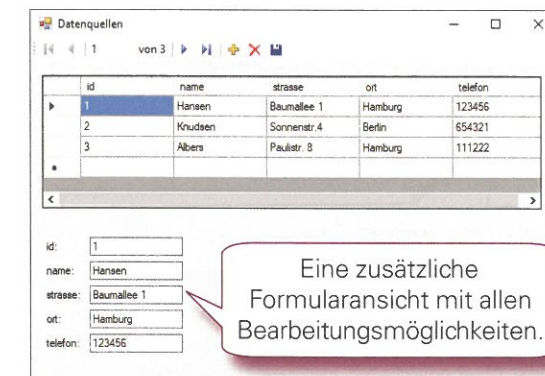
Nach dem Starten steht eine funktionstüchtige Ansicht der Datenbanktabelle zur Verfügung.



Die Datensätze können bearbeitet sowie gelöscht werden. Ebenso können auch neue Datensätze hinzugefügt werden. Natürlich stehen dem Entwickler eine Vielzahl von Ereignissen und Eigenschaften zur Verfügung, mit denen das Element programmiert werden kann. Beispielsweise kann durch den Doppelklick auf eine Zelle die Ereignisbehandlungsmethode *CellContentClick* erzeugt werden, die auf das Klicken auf einen Zelleninhalt reagiert.

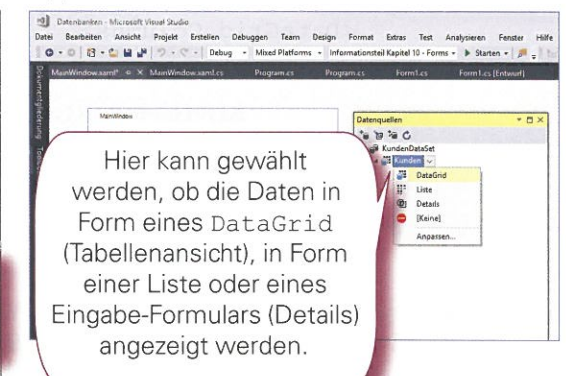
```
private void kundenDataGridView_CellContentClick(object sender,
                                         DataGridViewCellEventArgs e)
{
    MessageBox.Show("Auf einen Zelleninhalt geklickt!");
}
```

Zusätzlich zu der *GridView*-Ansicht kann durch-aus auch die Detailsansicht per *Drag&Drop* auf die Form gezogen werden. Damit stehen dem Benutzer nicht nur die Tabellenansicht, sondern auch eine Formularansicht zur Verfügung. Die Anzeige der Daten wird automatisch synchronisiert:

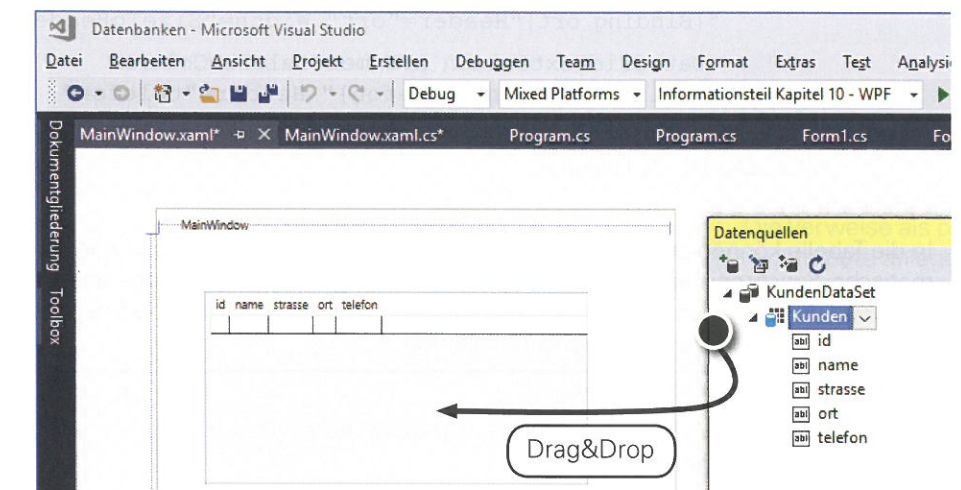


### 10.2.3 WPF-Steuerelemente automatisch anbinden

Mithilfe des Menüpunkts „Ansicht → Weitere Fenster → *Datenquellen*“ wird die Datenquelle angezeigt und kann anschließend für eine WPF-App verwendet werden.



Per *Drag&Drop* kann die Tabelle Kunden nun in der gewünschten Ansicht (hier *DataGrid*) auf das Fenster gezogen werden. Der Assistent legt automatisch das entsprechende Steuerelement und die Verbindung zur Datenbank und der Tabelle an:



Nach dem Starten steht eine Tabellen-Ansicht der Datenbanktabelle zur Verfügung.

id	name	strasse	ort	telefon
1	Hansen	Baumallee 1	Hamburg	123456
2	Knudsen	Sonnenstr.4	Berlin	654321
3	Albers	Paulistr. 8	Hamburg	111222