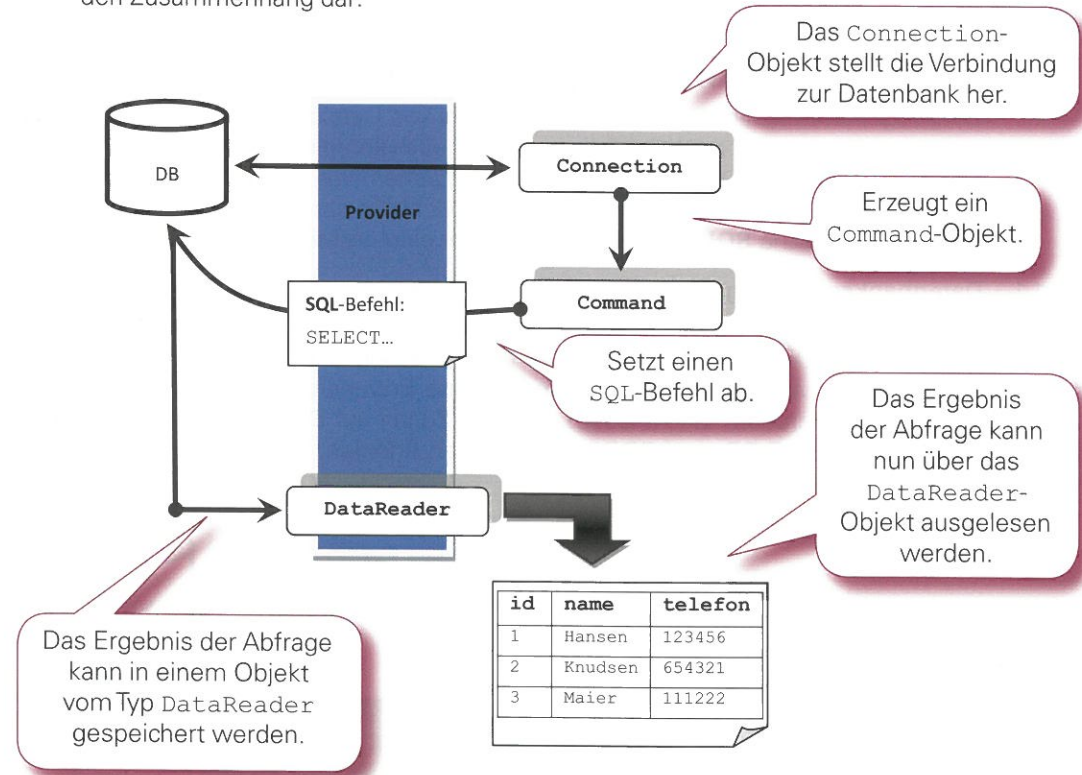


10.1.2 Provider nutzen und eine Verbindung aufbauen

Um eine Verbindung zu einer Datenbank aufzubauen, muss der entsprechende Datenbank-provider vorliegen. Einige Treiber sind bereits in der Standard-Installation von **Visual C#** vorhanden (wie der **OLE DB-Provider**). Andere Provider müssen von den jeweiligen Herstellern bezogen und installiert werden. Die eigentliche Verbindung wird dann mit einem Objekt der Klasse **Connection** hergestellt. Je nach Datenbank sind Nutzernamen und Passwörter anzugeben. Über ein **Command**-Objekt kann dann eine Abfrage gestartet und mit einem **DataReader**-Objekt ausgelesen werden. Die folgende Abbildung stellt den Zusammenhang dar:



10.1.3 Beispiel eines Zugriffs auf eine ACCESS-Datenbank

- Im Folgenden wird der Zugriff auf eine ACCESS-Datenbank mit dem **OLE DB-Provider** vorgestellt. Das Prinzip ist übertragbar auf andere relationale Datenbanksysteme wie beispielsweise den Microsoft-SQL-Server oder MySQL/MariaDB. Die Verbindung zur Datenbank wird mit einem **OleDbConnection**-Objekt aufgebaut, ein SQL-Befehl mit einem **OleDbCommand**-Objekt abgesetzt und das Ergebnis mit einem Objekt vom Typ **OleDbDataReader** ausgelesen.
- Als Entwicklungsumgebung wird in den folgenden Beispielen die kostenfreie Edition **Visual Studio Community** von Microsoft verwendet.

Der folgende Quellcode zeigt den Verbindungsaufbau zu einer ACCESS-Datenbank „Kunden.accdb“, die in einem Ordner (hier: C:\temp) zur Verfügung steht. Sie verfügt über eine Beispieltabelle Kunden mit den Attributen id (Typ Zahl) und name, strasse, ort und telefon (jeweils Text):

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

Datensatz: 1 von 3 | Kein Filter | Suchen

```
using System;
using System.Data;
using System.Data.OleDb;

namespace DB_Zugriff_CSharp
{
    class CDBZugriff
    {
        static void Main(string[] args)
        {
            string verbindungsstring =
                "Provider=Microsoft.ACE.OLEDB.12.0;
                Data Source=C:\\Temp\\Kunden.accdb";

            OleDbConnection dbVerbindung = null;
            OleDbCommand befehl = null;
            OleDbDataReader datenleser = null;
            bool offen = false;

            try
            {
                dbVerbindung =
                    new OleDbConnection(verbindungsstring);
                dbVerbindung.Open();
                offen = true;

                befehl = dbVerbindung.CreateCommand();
                befehl.CommandText = "SELECT * FROM Kunden";

                datenleser = befehl.ExecuteReader();

                while (datenleser.Read())
                {
                    Console.WriteLine("Name: "
                        + datenleser.GetString(1));
                }
            }
        }
    }
}
```

Einbinden der benötigten Namensräume!

Den Verbindungsstring mit der Provider-Angabe und der Datenquelle festlegen.

Ein Verweis auf eine OLEDB-Verbindung.

Ein Verweis auf ein OLE DB-Kommando.

Ein Verweis auf einen OLE DB-Datenleser.

Eine Verbindungs-Instanz

WICHTIG: Fehlerbehandlung

Flag setzen

Datenbank öffnen

Ein Befehlsobjekt erstellen lassen.

SQL-Befehl (alles aus der Tabelle auswählen) zuweisen.

Eine Datenleser-Instanz auf der Grundlage des SQL-Befehls erstellen lassen.

Sequenzielles Auslesen des Datenlesers

Die Methode GetString() liefert den Wert der aktuellen Zeile und vom übergebenen Spaltenindex.