

Regardless of what data access technology or database you use, you need to convert an Image to a Byte first and then save that. On retrieval, you convert the Byte array back to an Image.

To save:

```
Dim connection As New SqlConnection("connection string here")
Dim command As New SqlCommand("UPDATE MyTable SET Picture = @Picture WHERE ID = 1", connection)

'Create an Image object.'
Using picture As Image = Image.FromFile("file path here")
    'Create an empty stream in memory.'
    Using stream As New IO.MemoryStream
        'Fill the stream with the binary data from the Image.'
        picture.Save(stream, Imaging.ImageFormat.Jpeg)

        'Get an array of Bytes from the stream and assign to the parameter.'
        command.Parameters.Add("@Picture", SqlDbType.VarBinary).Value =
stream.GetBuffer()
    End Using
End Using

connection.Open()
command.ExecuteNonQuery()
connection.Close()
```

To retrieve:

```
Dim connection As New SqlConnection("connection string here")
Dim command As New SqlCommand("SELECT Picture FROM MyTable WHERE ID = 1",
connection)

connection.Open()

Dim pictureData As Byte() = DirectCast(command.ExecuteScalar(), Byte())

connection.Close()

Dim picture As Image = Nothing

'Create a stream in memory containing the bytes that comprise the image.'
Using stream As New IO.MemoryStream(pictureData)
    'Read the stream and create an Image object from the data.'
    picture = Image.FromStream(stream)
End Using
```

That example is for ADO.NET and SQL Server but the principle of using a MemoryStream for the conversion is the same regardless.