

22.4.8 Klasse Index (gb.db)

Die Klasse *Index* repräsentiert einen Tabellen-Index. Die Eigenschaften können für eine existierende DB-Tabelle ausgelesen werden. Die Klasse besitzt keine Methoden. Ein Index dient u.a. dazu, Datensätze mit bestimmten Eigenschaften schnell zu finden.

Hinweise auf die Eigenschaft *Indexes* vom Datentyp *Table.Indexes* im Zusammenhang mit der Definition eines Indexes finden Sie in der Klasse *Table* im Kapitel 22.4.6.

22.4.8.1 Eigenschaften

Die Klasse *Index* verfügt über die folgenden Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Fields	String[]	Liefert die DB-Felder, aus denen der Index besteht, als String-Array.
Name	String	Liefert den Namen des Indexes.
Primary	Boolean	Gibt True zurück, wenn ein Index der Primär-Index der Tabelle ist.
Table	Table	Liefert das Tabellenobjekt, das diesen Index besitzt.
Unique	Boolean	Liefert die Information, ob der Index als eindeutig deklariert wurde oder nicht.

Tabelle 22.4.8.1.1 : Eigenschaften der Klasse Index

Beispiel

Zuerst wird eine DBTabelle ``members`` in der aktuellen Datenbank ``contact.sqlite`` erzeugt, wenn sie *nicht* existiert und dann ein Index auf das Feld ``lastname`` erzeugt. Anschließend werden die Index-Eigenschaften für den DBIndex mit dem Index-Namen ``index_lastname_members`` ausgelesen:

```
Dim hDBTable As Table
Dim iCount As Integer = 1

If Not DBCS.DBConnection.Tables.Exist("members") Then

'-- DB-Tabellen-Objekt erzeugen - Tabellename: `members`
hDBTable = DBCS.DBConnection.Tables.Add("members")

'-- DB-Felder definieren
With hDBTable.Fields
.Add("m_id", db.Serial) ' gb.Serial => INTEGER PRIMARY KEY AUTOINCREMENT
.Add("lastname", db.String)
.Add("date", db.Date)
.Add("description", db.String)
End With

'-- Primärschlüssel definieren » Feldname: `m_id`
hDBTable.PrimaryKey = ["m_id"]

'-- DB-Tabelle `members` in der Datenbank `contacts.sqlite` erzeugen
hDBTable.Update()

'-- DBIndex definieren » Indexname: `index_lastname_members`, Feldname: `lastname`
If Not hDBTable.Indexes.Exist("index_lastname_members") Then
hDBTable.Indexes.Add("index_lastname_members", ["lastname"])
EndIf
Else
'-- Die existierende DBTabelle `members` wird aktuelle DBTabelle
hDBTable = DBCS.DBConnection.Tables["members"]

EndIf
```

Die folgenden Anweisungen lesen die Eigenschaften eines bestimmten Datenbank-Indexes aus und zeigen sie an:

```
Dim hIndex As Index
Dim sTableIndexName As String = "index_lastname_members"
Dim vElement As Variant

If hDBTable.Indexes.Exist(sTableIndexName) Then
hIndex = hDBTable.Indexes[sTableIndexName]
Print "Index-Name: "; hIndex.Name
Print "Ist der Index ein Primär-Index: "; hIndex.Primary
```

```

Print "Tabellen-Name zu diesem Index: "; hIndex.Table.Name
Print "Ist der Index als 'unique' deklaliert: "; hIndex.Unique
Print "Der Index wird über "; hIndex.Fields.Count; IIf(1, " DBFeld", " DBFelder"); " gebildet."
For Each vElement In hIndex.Fields
    Print "Index-FeldName "; iCount; ": "; vElement
    Inc iCount
Next
Else
    Print "Die Tabelle "; hDBTable.Name; " besitzt keinen Index: "; sTableName
Endif

```

Das sind die Ausgaben der Index-Inspektion in der Konsole der IDE für eine DBTabelle:

```

Index-Name: index_lastname_members
Ist der Index ein Primär-Index: False
Tabellen-Name zu diesem Index: members
Ist der Index als 'unique' deklaliert: False
Der Index wird über 1 DBFeld gebildet.
Index-FeldName 1: lastname

```

So löschen Sie einen Datenbank-Index:

```

If DBCS.DBConnection.Tables.Exist("members") Then

    If hDBTable.Indexes.Exist("index_lastname_members") Then
        Message.Info("Der Index `index_lastname_members` existiert.\nDer Index wird gelöscht!")
        hDBTable.Indexes.Remove("index_lastname_members")
        hDBTable.Indexes.Refresh()
    Else
        Message.Info("Die Tabelle " & hDBTable.Name & " besitzt keinen Index: " & sTableName)
    Endif

Endif

Catch
    Message.Error("Fehler!" & gb.NewLine & Error.Text & gb.NewLine & Error.Where)
    DBCS.DBConnection.Rollback()

```

Beachten Sie:

- Sie können in den o.a. Quelltext noch eine Index-Löschen-Abfrage einfügen, um Überraschungen auszuschließen.
- Wird die Datenbank-Tabelle gelöscht, zu der ein Index gehört, dann wird automatisch auch der Index gelöscht!
- Wenn Sie zum Beispiel in einer SQLite3-Datenbank zwei DB-Tabellen table1 und table2 angelegt haben, dann muss der Index-Name einmalig sein. Das bedeutet, dass Sie der Tabelle *table2* nicht einen Index mit dem gleichen Index-Namen wie in der Tabelle *table1* geben dürfen.