



Kurzanleitung für Umsteiger von DataEase

www.datacool.net

1 Umstieg leicht gemacht

1.1 Vorwort

DataCool ist ein Entwicklungswerkzeug für Client-/Server-Anwendungen. Mit Hilfe der intuitiven Benutzeroberfläche können Sie innerhalb kurzer Zeit eigene Anwendungen erstellen. Diese Kurzreferenz richtet sich speziell an Entwickler, die bisher mit dem Datenbanksystem **DataEase** gearbeitet haben.

Dies sind einige wichtige Vorteile von DataCool:

- Speicherung der Daten in SQL-Tabellen
- Hohe Stabilität auch in Mehrplatzumgebungen
- Offene Systemarchitektur durch Microsoft SQL Server
- nie wieder inkonsistente Daten oder defekte Indexdateien
- Transaktionsverarbeitung
- moderne 32 Bit-Anwendung
- keine Reorganisation bei Formularänderungen mehr nötig
- Änderung von Formularen im laufenden Betrieb
- Integrierte PDF-Unterstützung
- TAPI-Telefonwählfunktion
- Schnittstellen zu Word, Excel, Outlook
- direkter E-Mail-Versand aus der Anwendung

1.2 Zweck dieser Anleitung

Diese Anleitung ist kein vollständiges Nachschlagewerk. Vielmehr werden einige grundlegende **Unterschiede** beider Systeme aufgezeigt. Als DataEase Entwickler wird Ihnen der Umstieg aufgrund Ihrer Vorkenntnisse leicht fallen. Weitere Einzelheiten entnehmen Sie bei Bedarf dem **Entwicklerhandbuch** von DataCool. Mit dem **Tutorial** steht Ihnen außerdem eine fertige Beispielanwendung zur Verfügung.

1.3 Migration von DataEase 4.53 Anwendungen

Im Hauptmenü finden Sie unter **Tools** ⇒ **Formular installieren** ein Programm zur Installation einzelner DOS-Formulare. Sämtliche Feldeigenschaften (Feldtyp, Eingabebzwang, Index, etc.) werden hierbei übernommen. Lediglich Verknüpfungen und Ableitungen müssen Sie neu erstellen. Darüberhinaus besteht beim **Datenimport** die Möglichkeit **DBM-Dateien** dieser Version direkt einzulesen. Prozeduren können nicht übernommen werden, da die Abfragesprache DQL von SQL zu stark abweicht.

2 Gegenüberstellung DQL und SQL

Die nachfolgende Aufstellung zeigt die wichtigsten Unterschiede zwischen **DQL** (DataEase Query Language) und **SQL** (Structured Query Language) auf.

Datenzugriff per Script

```
for Formular;  
end .
```

```
select * from Formular;  
next
```

Datenselektion nach Textfeld

```
for Formular with Feld="ABC";  
end .
```

```
select * from Formular where Feld="ABC";  
next
```

Datenselektion mit Wildcard-Suche

⇒ Vergleichsoperator „like“ und Platzhalter „%“ verwenden

```
for Formular with Feld="A*";  
end .
```

```
select * from Formular where Feld like "A%";  
next
```

Datenselektion mit Nummernsuche

⇒ Nummern in Anführungszeichen setzen

```
for Formular with Feld=00001;  
end .
```

```
select * from Formular where Feld="00001";  
next
```

Datenselektion nach Zahlenfeld

⇒ im Script stets den Punkt als Dezimaltrennzeichen verwenden

```
for Formular with Feld=1,19;  
end .
```

```
select * from Formular where Feld=1.19;  
next
```

Datenselektion nach Datum

⇒ Datumswerte in Hochkommas setzen und im Format TT.MM.JJJJ angeben

```
for Formular with Datum>=01/01/09;  
end .
```

```
select * from Formular where Datum>="01.01.2009";  
next
```

Datenselektion nach Datumsbereich

⇒ „between ... to ...“ ersetzen durch „between ... and ...“

```
for Formular with Datum between 01/01/09 to 31/01/09;  
end .
```

```
select * from Formular where Datum between "01.01.2009"  
and "31.12.2009";  
next
```

Datenselektion nach ja/nein-Feld

⇒ Selektion nach der Auswahlnummer: 1=nein, 2=ja

```
for Formular with Feld=nein;  
end .
```

```
select * from Formular where Feld=1;  
next
```

Datenselektion nach Auswahlfeld

⇒ Selektion nach der Auswahlnummer

```
for Formular with Feld=Auswahltext;
end .
```

```
select * from Formular where Feld=1;
next
```

Datenselektion nach Formulareingabe

⇒ „data-entry Feld“ ersetzen durch „Form.Feld“

⇒ in where-Klauseln das @-Zeichen auf der rechten Seite verwenden (Ersetzung)

```
for Formular with Datum>=data-entry Stichtag;
end .
```

```
select * from Formular where Datum>=@Form.Stichtag;
next
```

Abfrage auf leere Felder

```
for Formular with Feld=blank;
end .
```

```
select * from Formular where Feld is null;
next
```

Sortierung, aufsteigend

```
for Formular with Feld="ABC";
  list records
    Feld1 in order ;
    Feld2 in order .
end .
```

```
select * from Formular where Feld="ABC"
order by Feld1, Feld2;
next
```

Sortierung, absteigend

```
for Formular with Feld="ABC";  
  list records Feld1 in reverse .  
end .
```

```
select * from Formular where Feld="ABC" order by Feld1 desc;  
next
```

Datensätze einfügen

```
enter a record in Formular  
  Feld1:="ABC" ;  
  Feld2:=01/01/09 .
```

```
insert into Formular set  
  Feld="ABC",  
  Feld2="01.01.2009";
```

Aktuellen Datensatz ändern

```
for Formular with Feld="ABC";  
  modify records  
    Feld1:="ABC" ;  
    Feld2:=01/01/09 .  
end .
```

```
select * from Formular where Feld="ABC";  
  update record set  
    Feld1="ABC",  
    Feld2="01.01.2009";  
next
```

Ausgewählte Datensätze ändern

```
modify records in Formular with (Feld=data-entry Eingabe)
  Feld1:="ABC" ;
  Feld2:=01/01/09 .
```

```
update Formular set
  Feld1="ABC",
  Feld2="01.01.2009"
where Feld=@Form.Eingabe;
```

Aktuellen Datensatz löschen

```
for Formular with Feld="ABC";
  delete records .
end .
```

```
select * from Formular where Feld="ABC";
  delete record;
next
```

Ausgewählte Datensätze löschen

```
delete records in Formular with (Feld=data-entry Eingabe) .
```

```
delete from Formular where Feld=@Form.Eingabe;
```

Geschachtelte Schleifen (Haupt- und Subformular)

⇒ Verknüpfungen greifen nicht auf Scriptebene und müssen daher explizit genannt werden

```
for Hauptformular;
  for Subformular
  end .
end .
```

```
select * from Hauptformular;
  select * from Subformular where Feld=@Hauptformular.Feld;
  next
next
```