

Teil 2 der Anwendungserstellung mit Access

Öffnen Sie das Formular frmRechnung in der Entwurfsansicht.

Text verschieben

Klicken Sie auf das größere Quadrat über dem Text „frmRechnung“. Ziehen Sie den Text bei gedrückter Maustaste an die linke obere Ecke des Formulars.

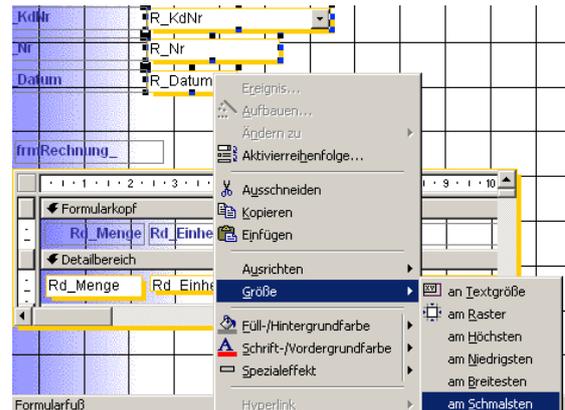


Größe der Textfelder ändern

Markieren Sie die Felder, deren Größe angepasst werden soll.

Klicken Sie mit der rechten Maustaste auf die Markierung.

Wählen Sie „Größe“ „am Schmalsten“, um die Größe der übrigen Objekte an das schmalste Objekt anzupassen.



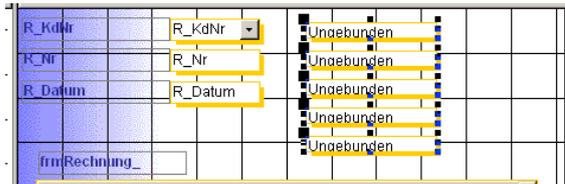
Neue Felder erstellen

Klicken Sie mit der rechten Maustaste auf die markierten Felder, wählen Sie „Ausrichten“ / „linksbündig“.

Klicken Sie auf die Schaltfläche  in der Toolbox.

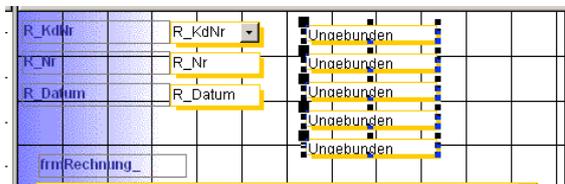
Falls die Toolbox nicht eingeschaltet ist, klicken Sie auf die Schaltfläche:

Fügen Sie 5 Felder für die Adresse ein.



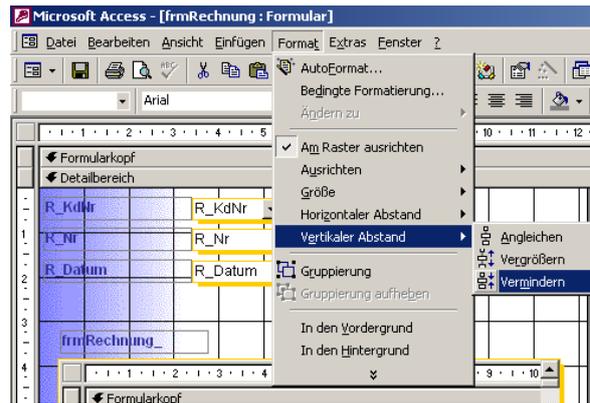
Objekte ausrichten

Klicken Sie mit der rechten Maustaste auf die markierten Felder, wählen Sie „Ausrichten“ / „linksbündig“.



Abstand vermindern / angleichen

Klicken Sie mit der rechten Maustaste auf die markierten Felder, wählen Sie im Menü Format „Vertikaler Abstand“ / Vermindern.

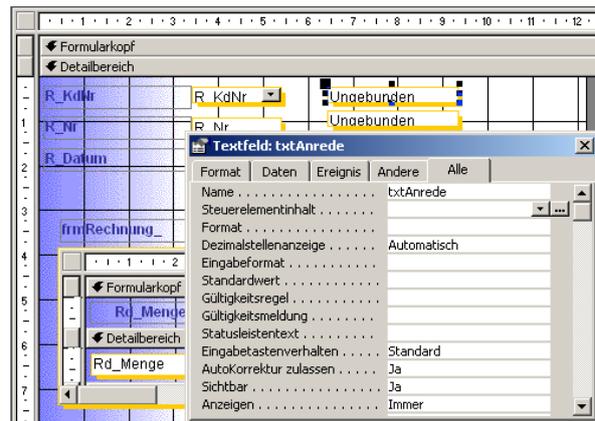


Felder benennen

Doppelklicken Sie außerhalb des Formulars in den grauen Bereich oder wählen Sie im Kontextmenü (RMT) „Eigenschaften“.

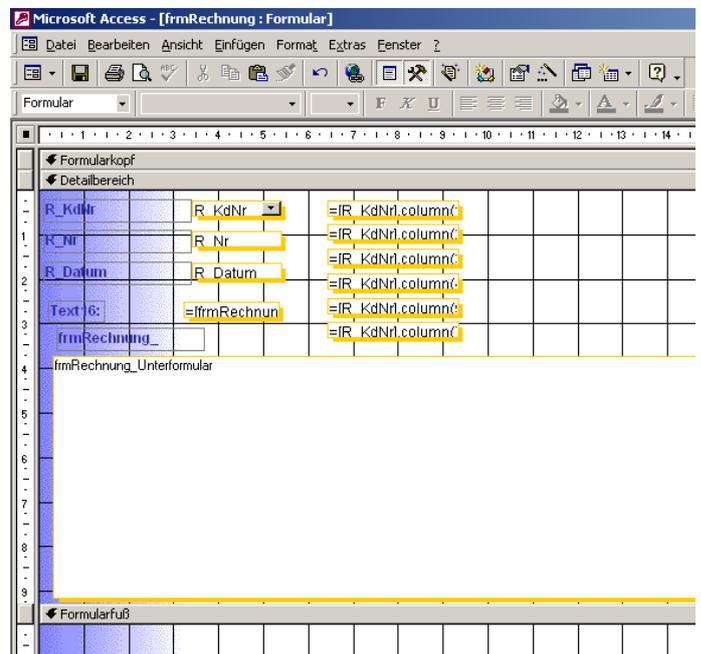
Klicken Sie auf das erste Feld.

Geben Sie den Namen „txtAnrede“ ein.



Anzeigefehler

Wenn das Unterformular im Entwurf nicht komplett angezeigt wird, schalten Sie zurück in die Formularansicht und noch Mal in die Entwurfsansicht.



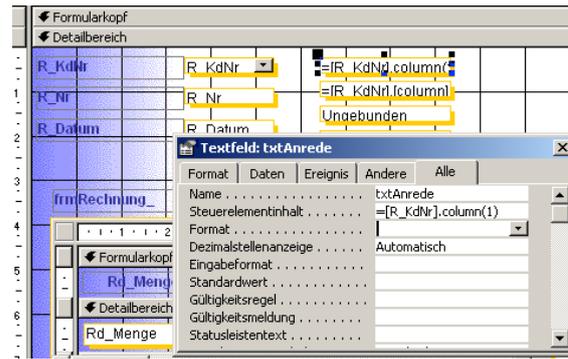
Berechnete Felder im Formular

Geben Sie Steuerelementinhalt des Feldes folgenden Wert ein:

=r_kdnr.column(1)

Sie weisen dem Feld den Inhalt des Nachschlagefeldes r_kdnr zu.

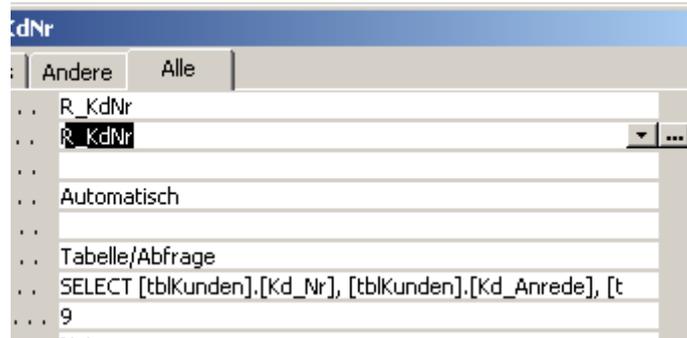
Angezeigt soll der Wert aus der 2. Spalte (column(0) ist die erste Spalte) des Feldes



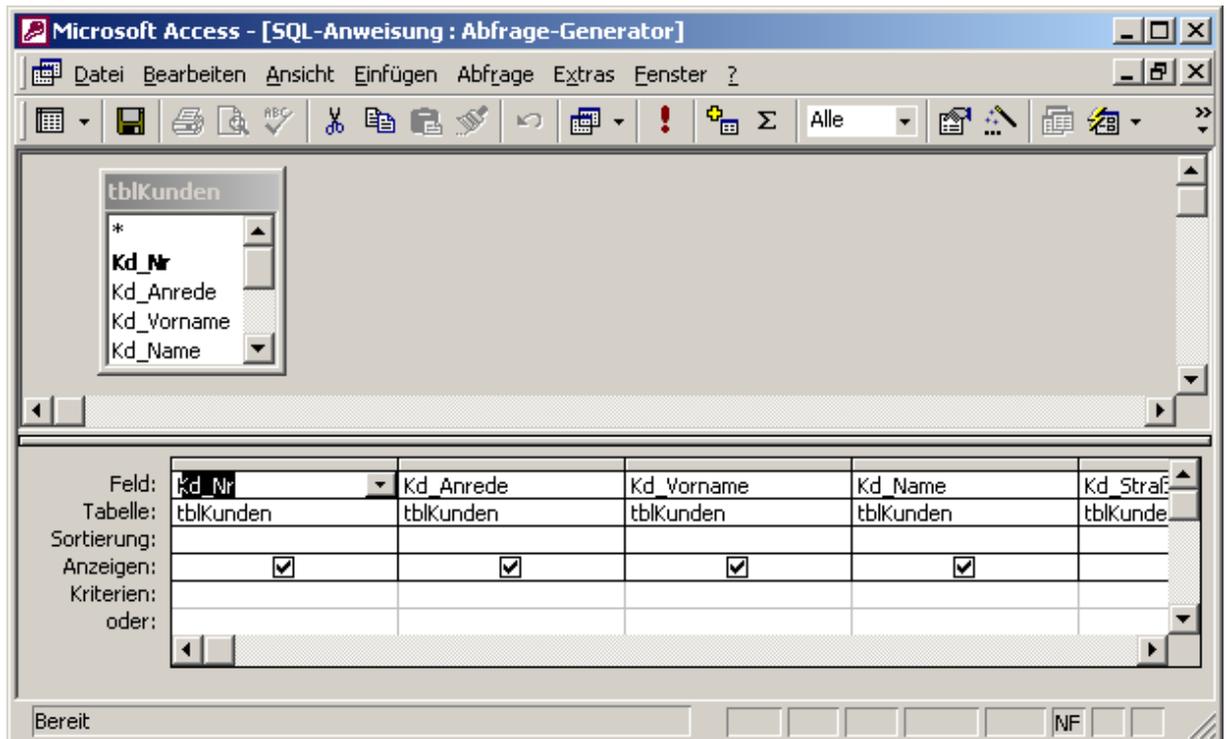
Inhalt des Textfeldes „Anrede“



Inhalt des Textfeldes „R_KdNr“



Diese Abfrage steht hinter dem Nachschlagefeld. Hier kann z.B. die Sortierung eingestellt werden.

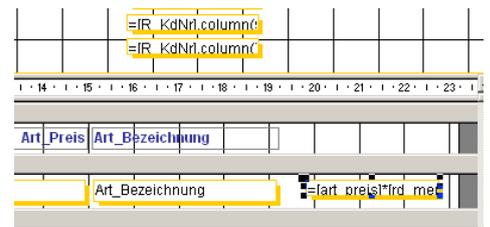


Schließen Sie nun die Abfrage.

Gesamtpreis im Formular anzeigen

Doppelklicken Sie in der Entwurfsansicht auf das Unterformular.

Erstellen Sie ein neues Textfeld im Detailbereich.



Berechnetes Feld im Formular

Sie können den Steuerelementinhalt alternativ auch im Eigenschaftsfenster eintippen.



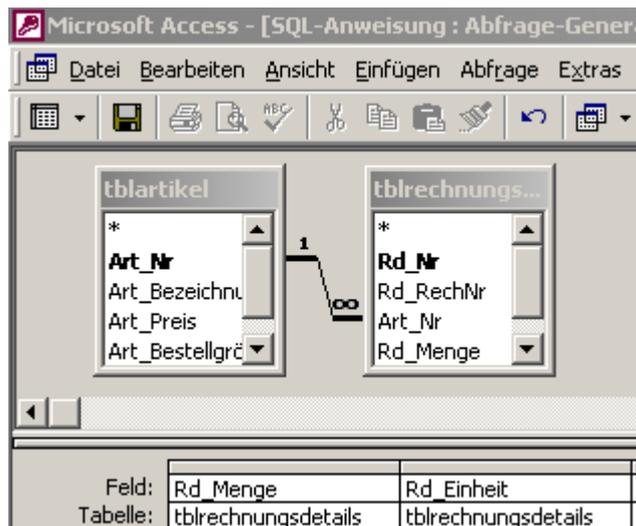
Hinweis

Die Berechnung kann auf diese Weise durchgeführt werden, ist aber nicht sehr sinnvoll.

Für die Summenbildung bzw. für die Weiterverwendung der Abfrage in einem Bericht ist es besser, die Berechnung in der zugrunde liegenden Abfrage durchzuführen.

Doppelklicken Sie im Unterformular auf den „grauen Bereich“ außerhalb des Formulars, um das Eigenschaftenfenster einzublenden. Klicken Sie neben „Datenherkunft“ auf das Symbol „...“, um die Abfrage zu öffnen, die hinter dem Unterformular für die Daten sorgt.

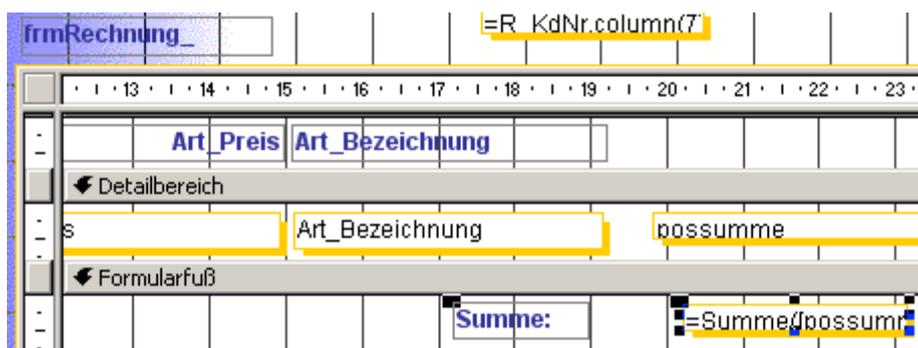
Fügen Sie ein neues Feld ein oder wählen Sie ein leeres Feld rechts neben den bestehenden Feldern.



Drücken Sie Umschalttaste + F2, um das Zoomfenster zu aktivieren. Tippen Sie die Formel wie rechts dargestellt ein. Das Ergebnis der Berechnung kann nun im Formular verwendet werden. Beenden Sie die Abfrage.



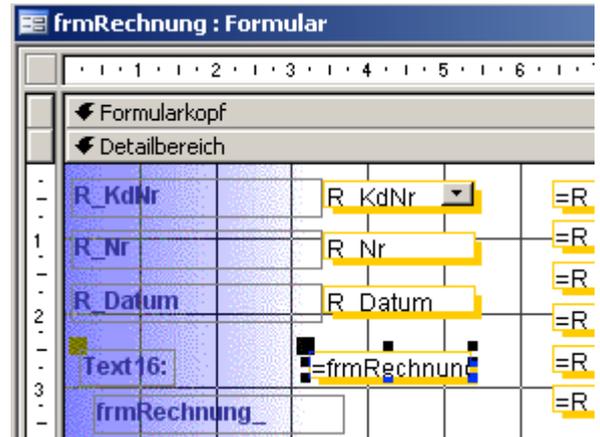
Fügen Sie im Unterformular das neu erstellte Feld „PosSumme“ ein. Erstellen Sie im Formularfuß ein neues Feld „txtSumme“ mit dem Steuerelementinhalt: =Summe(possumme)



Ansicht im Eigenschaftenfenster.



Klicken Sie auf „Speichern“, um den Zwischenstand zu sichern.
 Klicken Sie im Hauptformular an eine beliebige Position, um das Hauptformular zu aktivieren.
 Positionieren Sie ein neues Textfeld unterhalb des Datums. Dort soll der Gesamtbetrag eingefügt werden.
 Aktivieren Sie das Eigenschaftenfenster.
 Klicken Sie auf das neue Feld.

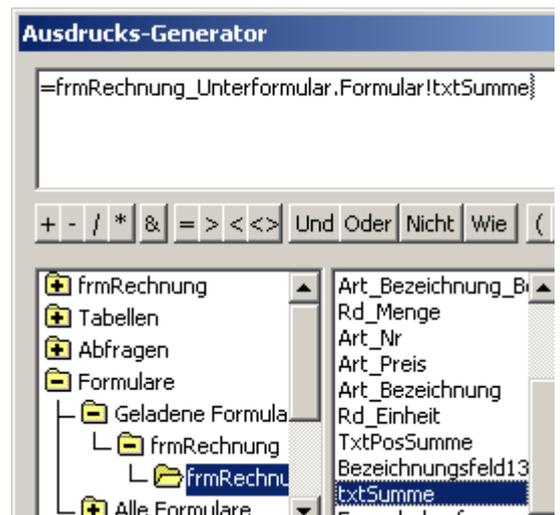


Rechnungsbetrag

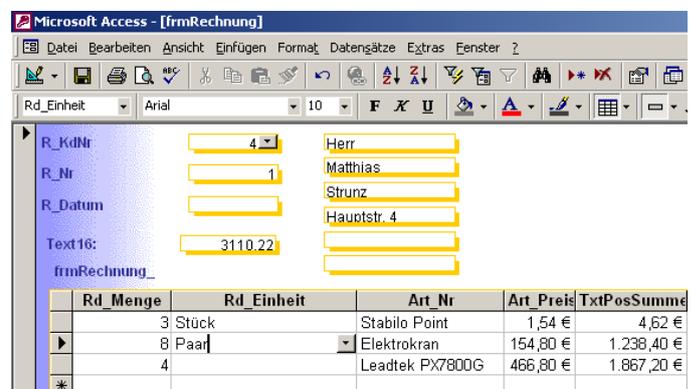
Klicken Sie im Eigenschaftenfenster neben den Eintrag „Steuerelementinhalt“.
 Klicken Sie auf die Schaltfläche (3 Punkte), die neben dem Eintrag sichtbar wird:



Doppelklicken Sie auf „Formulare/Geladene Formulare/frmRechnung/frmRechnungUnterformular“.
 Doppelklicken Sie im rechten Fenster das Formularfeld „TxtSumme“.
 Bestätigen Sie mit OK.



Jede Position, die nun neu erfasst wird, wird im Formular aufaddiert.



Berichte

Die Berichte werden zur Datenausgabe in formatierter Form verwendet. Sie können beliebige Abfragen hinterlegen.

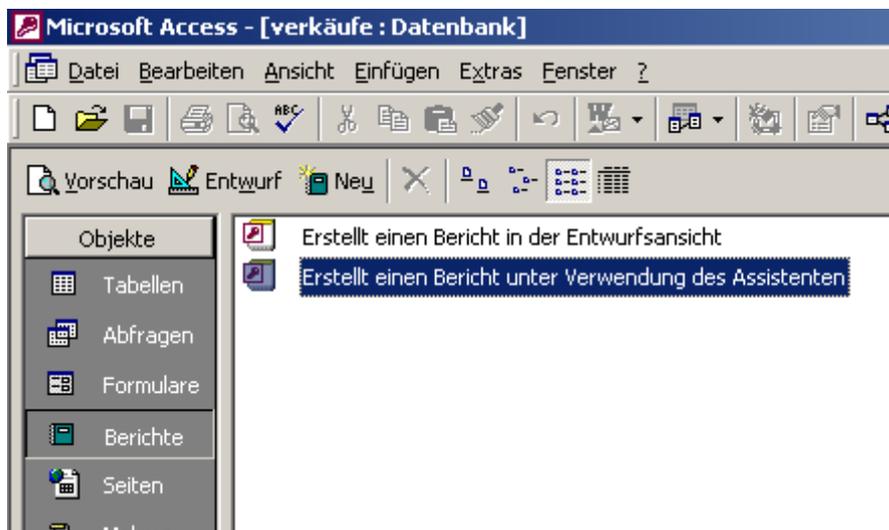
Sinnvoll ist die Verwendung der Abfragen, die auch in den Berichten verwendet werden, um die doppelte Erstellung zu vermeiden.

Bericht erstellen

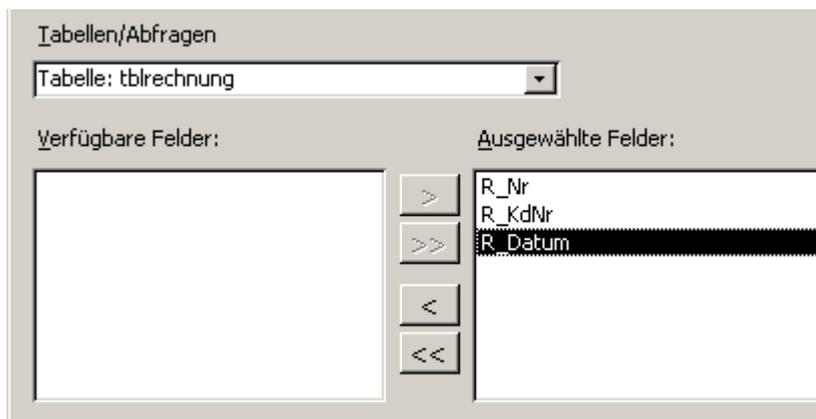
Klicken Sie auf das Objekt „Berichte“.

Wählen Sie „Erstellt einen Bericht unter Verwendung des Assistenten“.

In den nachfolgenden Schritten werden die Daten für den Bericht aus mehreren Tabellen gewählt.



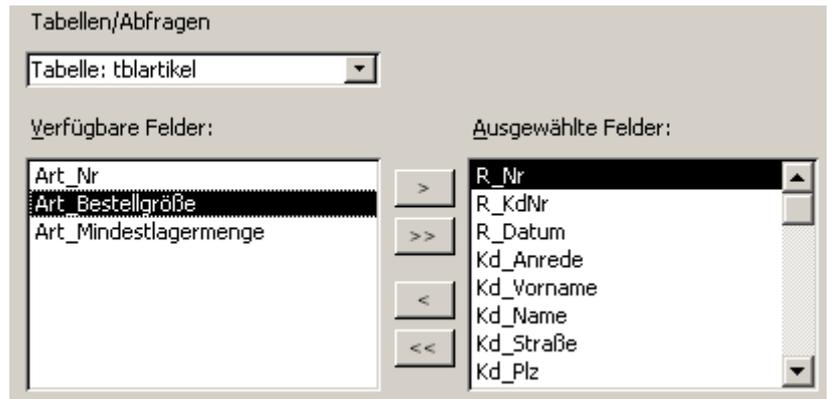
Wählen Sie unter „Tabellen/Abfragen“ die jeweilige Datenherkunft. Hier können Sie mehrere Tabellen verwenden. Voraussetzung ist, dass diese vorher eine Beziehung zwischen den Tabellen/Abfragen erstellt wurde. Wählen Sie die Tabelle „tblRechnung“. Wählen Sie alle Felder durch Klick auf den Doppelpfeil.



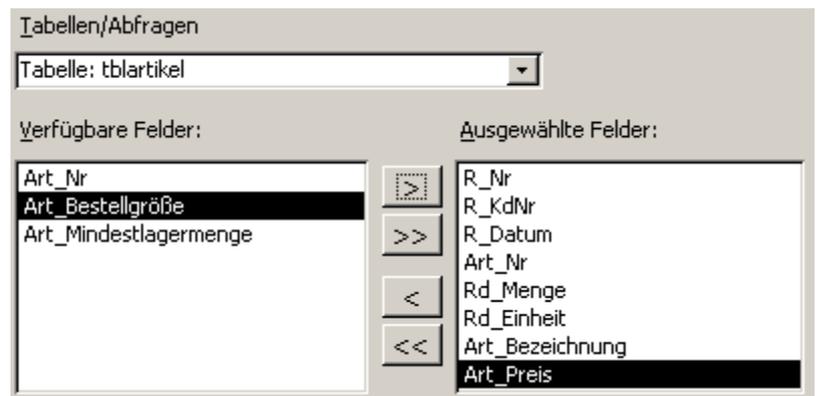
Wählen Sie aus der Tabelle „tblRechnungsDetails“ die Felder „Rd_Einheit“, „Rd_Menge“ und „Rd_Einheit“.



Wählen Sie aus der Tabelle „tblKunden“ alle Felder, die für eine korrekte Anrede/Anschrift nötig sind.



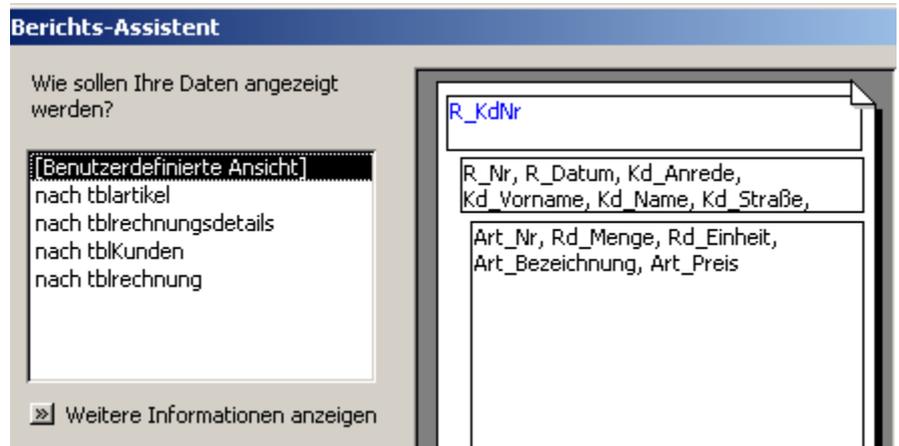
Wählen Sie aus der Tabelle „tblArtikel“ die Felder „Art_Preis“ und „Art_Bezeichnung“. Die Artikelnummer kommt aus der Tabelle „tblRechnungsDetails“.



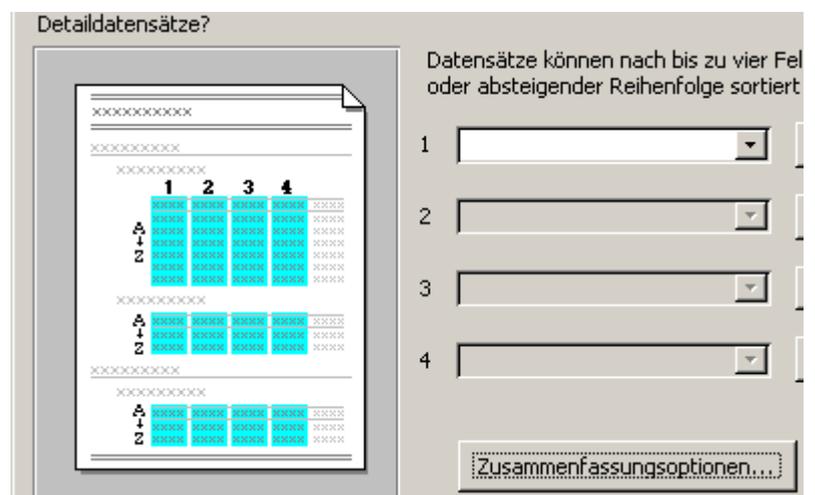
Gruppierung

Die Gruppierung dient zur Summenbildung u.a. Operationen.

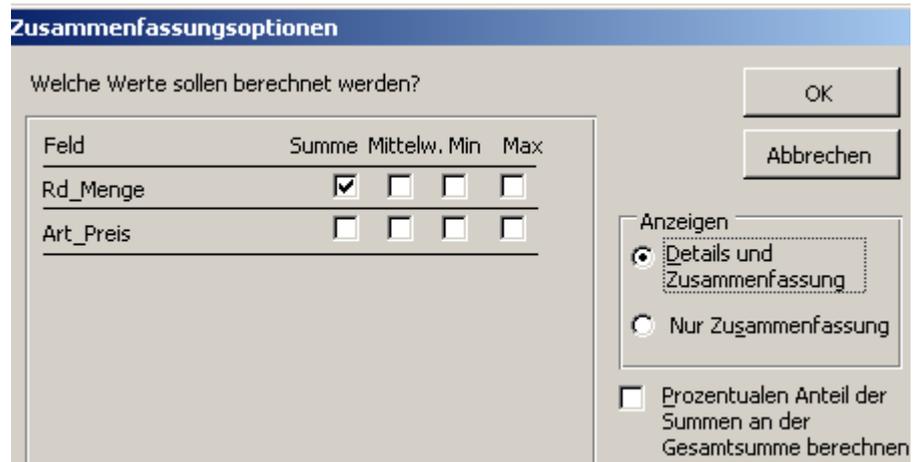
Die Summe soll in diesem Fall über die Rechnungsnummer gebildet werden.



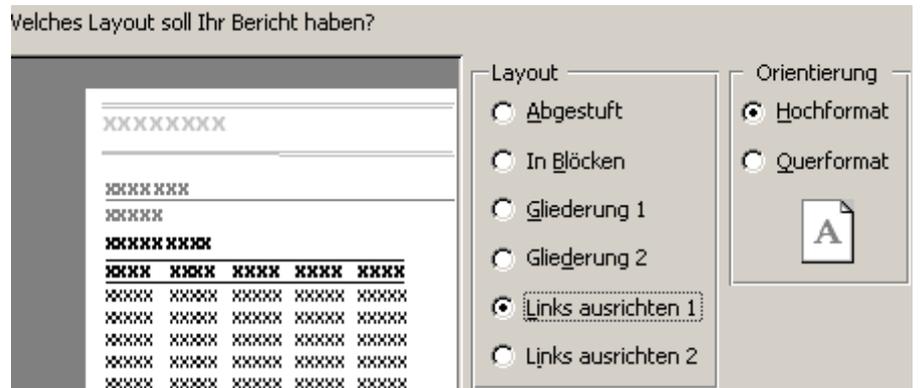
Wählen Sie im nächsten Schritt die Zusammenfassungsoptionen.



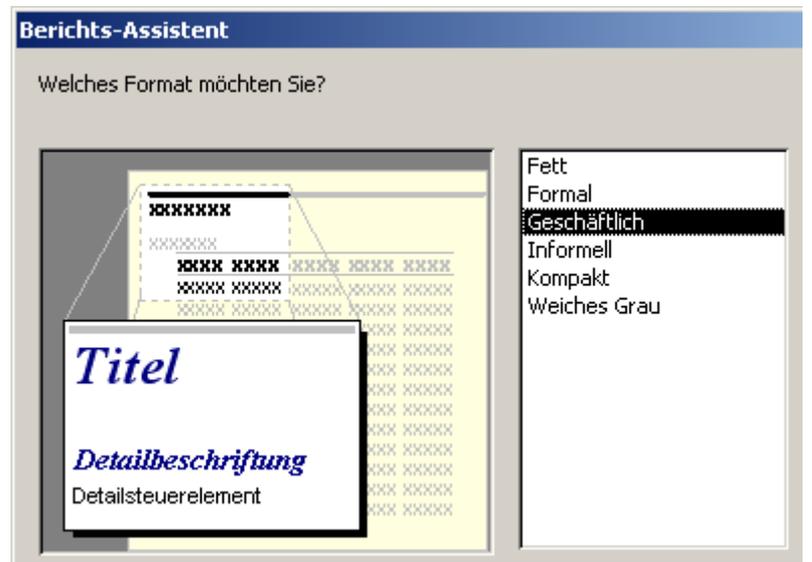
Wählen Sie in einem Feld z.B. Summe. Die Summe über die einzelnen Positionen gibt in diesem Fall je nach Sortiment Aufschluss über den Verpackungsaufwand.



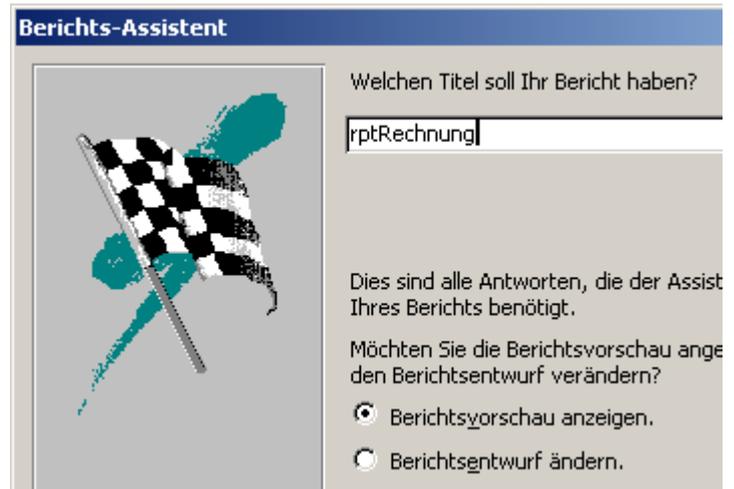
Wählen Sie die Art der Darstellung. Bericht müssen aber trotz der Auswahl in den meisten Fällen manuell nachformatiert werden.



Wählen Sie das „Format“.



Geben Sie im nächsten Schritt den Berichtsnamen ein.
„rptRechnung“ steht für Report (Bericht) Rechnung.



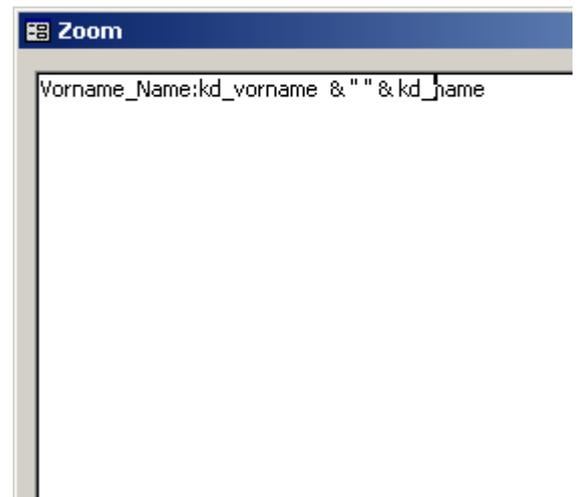
Die Berichtsansicht muss erheblich modifiziert werden.
Schalten Sie dazu in die Entwurfsansicht.

rptRechnung

R_KdNr				4
R_Nr	1	Kd_Name	Struz	
R_Datum		Kd_Straße	Hauptstr. 4	
Kd_Anrede	Herr	Kd_Plz		
Kd_Vorname	Matthias	Kd_Ort		

Rd_ArtNr	Rd_Menge	Rd_Einheit	Art_Bezeichnung	Art_Preis
Leadtek PX7800GTX T	4		Leadtek PX7800GTX T	466,80 €
Elektrokran	8	Paar	Elektrokran	154,80 €
Stabilo Point	3	Stück	Stabilo Point	1,54 €
<i>Zusammenfassung für 'R_Nr' = 1 (3 Detaildatensätze)</i>				
Summe				623,14 €
<i>Zusammenfassung für 'R_KdNr' = 4 (3 Detaildatensätze)</i>				
Summe				623,14 €
Gesamtsumme				623,14 €

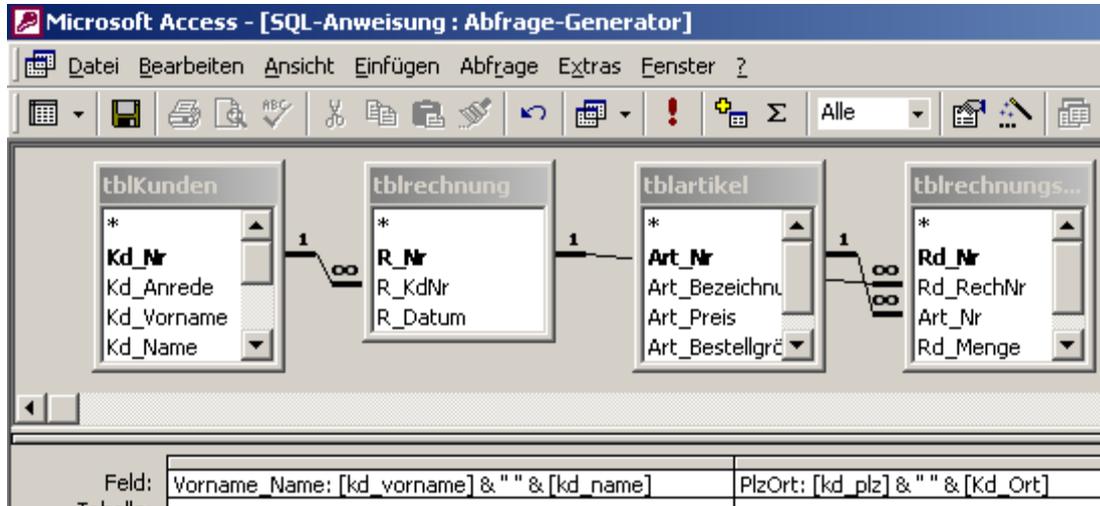
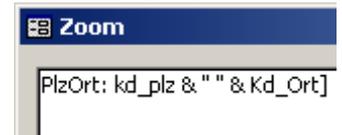
Erstellen Sie in der Datenherkunft des Berichts ein neues Textfeld Vorname_Name, in welchem Sie die beiden Felder Kd_Vorname und Kd_Name über die Textverkettung zu einem neuen Feld zusammenfügen.



Falls rechts keine Spalte mehr „frei“ ist, fügen Sie über das Menü „Einfügen“ eine neue Spalte ein.



Erstellen Sie in der Abfrage auch ein Feld für Plz und Ort wie rechts dargestellt.



Erstellen Sie ein Feld für die Positionssumme.
Die Abfrage des Formulars beinhaltet dieses Feld bereits. Durch gleichzeitige Verwendung dieser Abfrage könnte man sich Tipparbeit sparen und späteren Änderungsaufwand verringern.



Erstellen Sie auch ein Feld „MwSt“ wie rechts dargestellt.

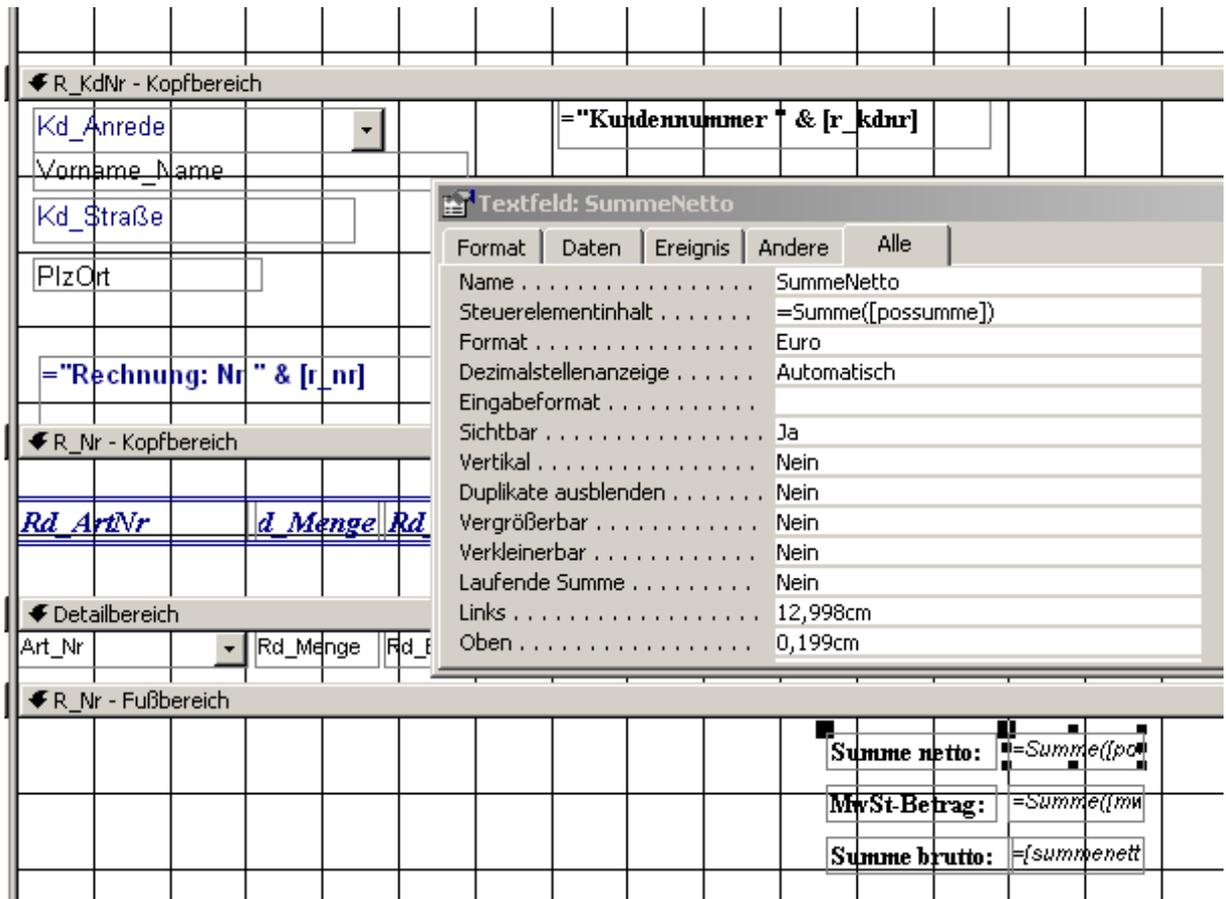


Schließen Sie die Abfrage.
Bestätigen Sie die Frage, ob gespeichert werden soll.
Fügen Sie die neuen Felder „PlzOrt“ und den Kundennamen im Bericht ein.
Löschen Sie die nicht benötigten Bezeichnungsfelder.



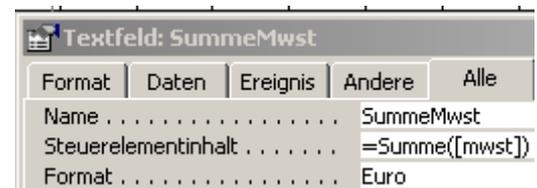
Erstellen Sie im Fußbereich der Rechnung ein neues Feld „SummeNetto“. Geben Sie die unten dargestellte Formel ein.

Auf diese Weise werden alle Datensätze dieser Rechnung aufsummiert. Klicken Sie dann auf das Feld und stellen Sie das Format „Währung“ ein.



Mwst-Summe

Erstellen Sie ein weiteres Feld „SummeMwst“: „=summe(mwst)“



Erstellen Sie das Feld „summeBrutto“. Geben Sie im Steuerelement die rechts dargestellte Formel ein.



Formatierungsvorschlag für den Bericht

Berichtskopf										
Meine Firma										
Meine Straße										
Mein Ort										
Seitenkopf										
R_KdNr - Kopfbereich										
Kd_Anrede						="Kundennummer" & [r_kdnr]				
Vorname_Name								R_Datum		
Kd_Straße										
PlzOrt										
="Rechnung: Nr" & [r_nr]										
R_Nr - Kopfbereich										
ArtNr		Menge	Einheit		Bezeichnung		Preis	Pos-Summe:		
Detailbereich										
Art_Nr		Rd_Menge	Rd_Einheit		Art_Bezeichnung		Art_Preis	FosSumme		
R_Nr - Fußbereich										
							Summe netto:	=Summe([po.		
							MwSt-Betrag:	=Summe([mw		
							Summe brutto:	=[summenett		

Rechnung in der Druckvorschau

Meine Firma
Meine Straße
Mein Ort



Herr
 Matthias Strunz
 Hauptstr. 4

Kundennummer 4

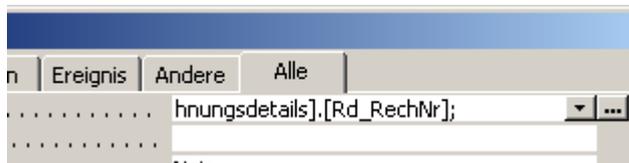
Rechnung: Nr 1

ArtNr	Menge Einheit	Bezeichnung	Preis	Pos-Summe:
Leadtek PX7800GTX T	4	Leadtek PX7800GTX T	466,80 €	1.867,20 €
Elektrokran	8 Paar	Elektrokran	154,80 €	1.238,40 €
Stabilo Point	3 Stück	Stabilo Point	1,54 €	4,62 €
			Summe netto:	3.110,22 €
			Mw St-Betrag	590,94 €
			Summe brutto:	3.701,16 €

Aktuelle Rechnungsnummer anzeigen

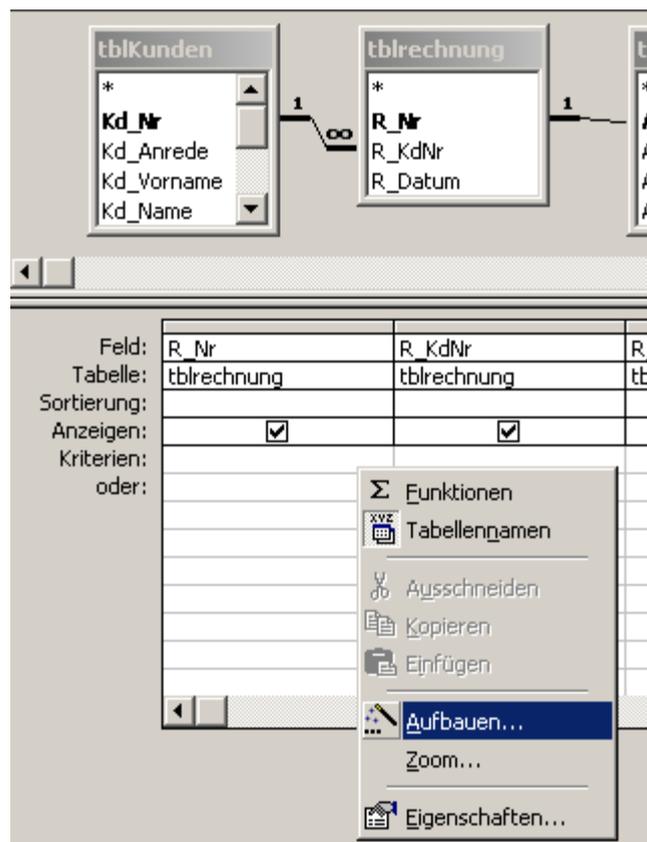
Beim Öffnen des Berichtes soll die Rechnungsnummer angezeigt werden, die derzeit im Formular angezeigt wird.

Klicken Sie im Eigenschaftsfenster des Berichts in der Datenherkunft auf die 3 Punkte.



Drücken Sie in der Abfrage in der Kriterienzeile die rechte Maustaste.

Wählen Sie „Aufbauen“.

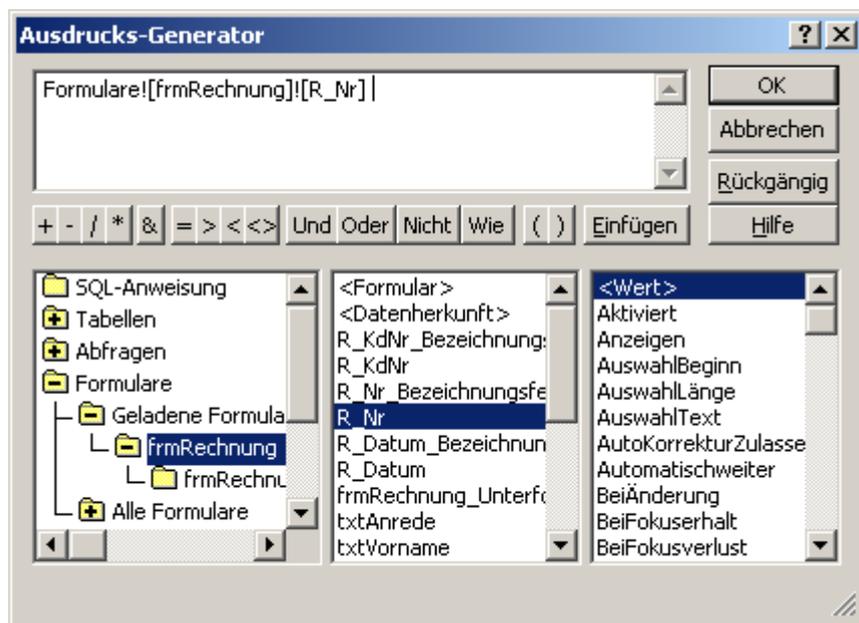


Klicken Sie auf

- „Formulare“
- „Geladene Formulare“ (dazu muss das Formular geöffnet sein, sonst auf „Alle Formulare“).
- „frmRechnung“
- In der mittleren Liste auf „R_Nr“.

Sie bewirken damit, dass beim Öffnen des Berichtes im geöffneten Formular nachgeschaut wird, welche Rechnungsnummer derzeit geöffnet ist.

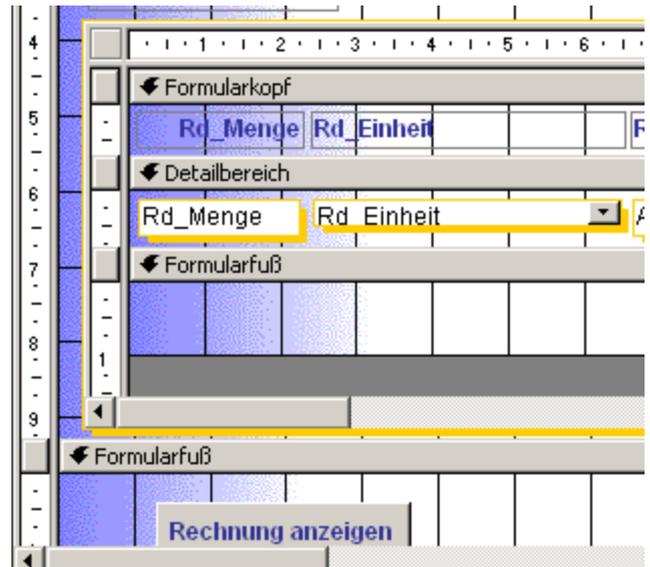
- Schließen Sie die Abfrage Speichern mit JA bestätigen“.
- Schließen Sie den Bericht. Speichern mit JA bestätigen“.



Schaltfläche zum Öffnen der Rechnung

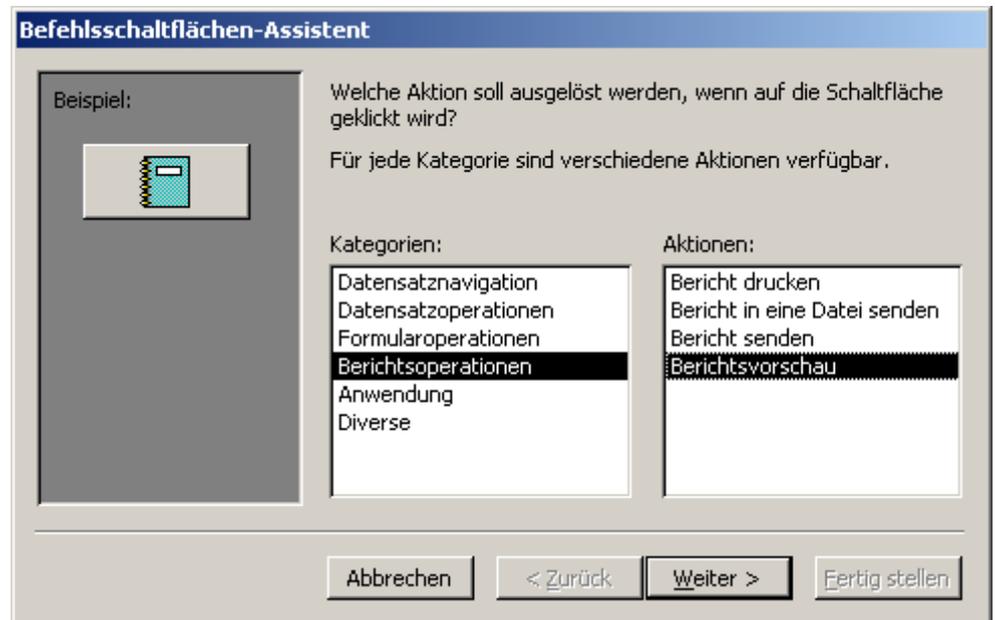
Öffnen Sie das Formular „frmRechnung“.

Erstellen Sie eine Befehlsschaltfläche: 



Platzieren Sie die Schaltfläche im Formularfuß.

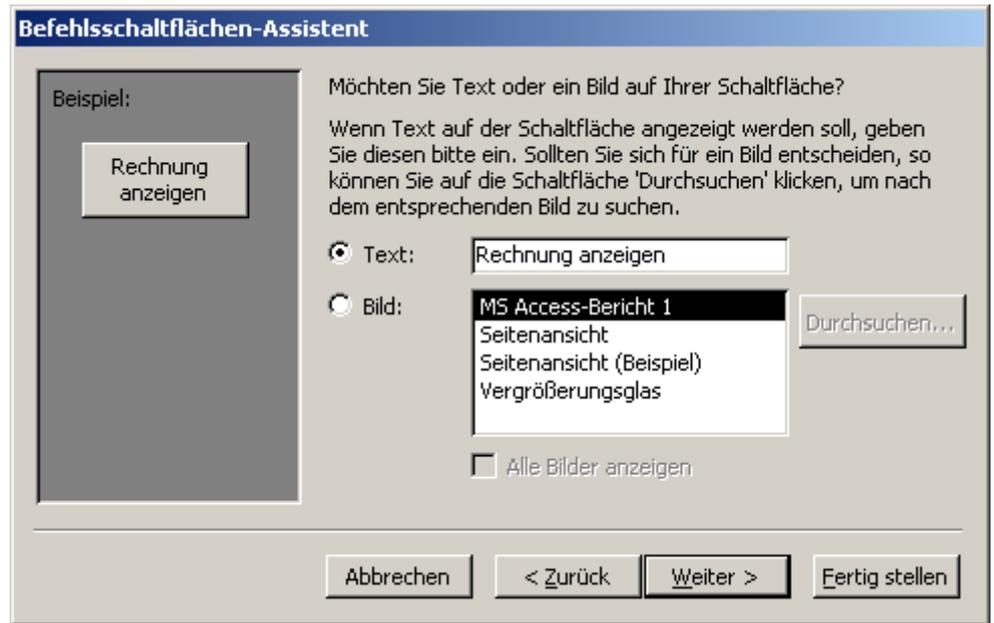
Wählen Sie „Berichtsoperationen/ Berichtsvorschau“.



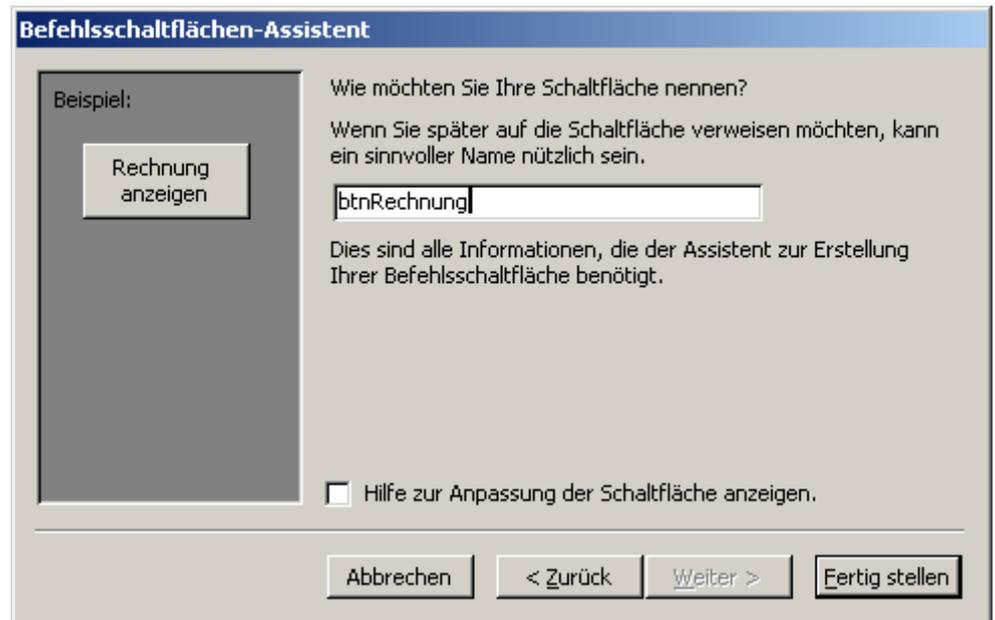
Wählen Sie den Bericht „rptRechnung“.



Geben Sie den Text „Rechnung anzeigen“ als Schaltflächenbeschriftung ein.



Benennen Sie die Schaltfläche als „btnRechnung“. (Button Rechnung)



Klicken Sie auf das Feld „R_KdNr“.
 Nach der Auswahl des Kunden soll geprüft werden, ob bereits ein Rechnungsdatum eingegeben wurde.
 Falls dies nicht der Fall ist, wird nun das Tagesdatum erfasst.

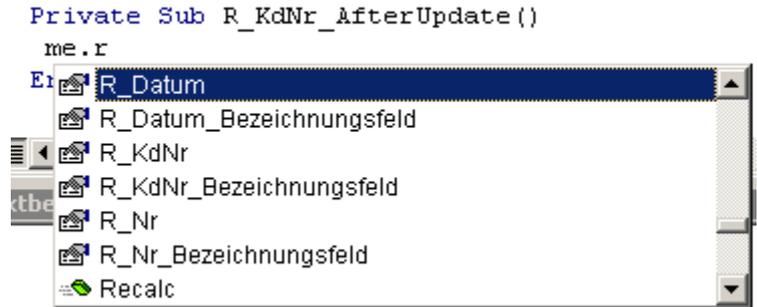


Ereignis: Nach Aktualisierung

Tagesdatum eintragen

Geben Sie den Text „me.r“ ein. Nun sollte automatisch eine Liste der im Formular (me) vorhandenen Felder angezeigt werden, die mit „R“ beginnen.

Wenn das Feld „R_Datum“ hinterlegt dargestellt wird, drücken Sie die Leertaste, um das Feld einzufügen.



Diese Routine trägt das Tagesdatum ein – ohne Rücksicht darauf, welcher Wert vorher im Feld stand.

Unten sehen Sie den verbesserten Code, der prüft, ob ein Wert im Feld R_Datum steht.

```
Private Sub R_KdNr_AfterUpdate()
    me.R_Datum = date()
End Sub
```

If ohne End If

Im Regelfall wird die Syntax If..then..End if verwendet. Wenn man nur eine Bedingung hat, kann diese in eine Zeile geschrieben werden – ohne End If.

```
Private Sub R_KdNr_AfterUpdate()
    'If ohne End if
    If IsNull(Me.R_Datum) Then Me.R_Datum = Date
End Sub
```

Kundenformular erstellen

Zur Eingabe von Kundendaten benötigen Sie ein Formular.

Erstellen Sie ein Formular unter Verwendung des Assistenten.

Wählen Sie als Datenbasis die Tabelle „tblKunden“. Nennen Sie das Formular „frmKunden“.

The screenshot shows the 'tblKunden' data entry form. It contains the following fields and values:

Kd_Nr	<input type="text"/>
Kd_Anrede	Frau
Kd_Vorname	Heike
Kd_Name	Ludwig
Kd_Straße	Hirschvogelstr. 13
Kd_Plz	85570
Kd_Ort	<input type="text"/>
Kd_Telefon	<input type="text"/>
Kd_Mail	<input type="text"/>

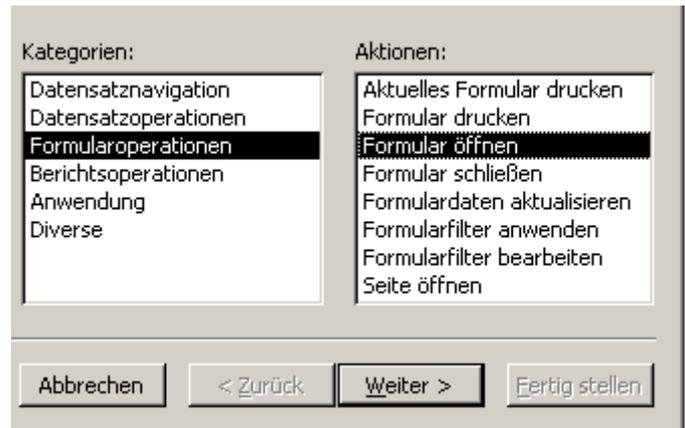
At the bottom, the status bar shows 'Datensatz: 1 von 4'.

Schalter zum Öffnen des Kundenformulars

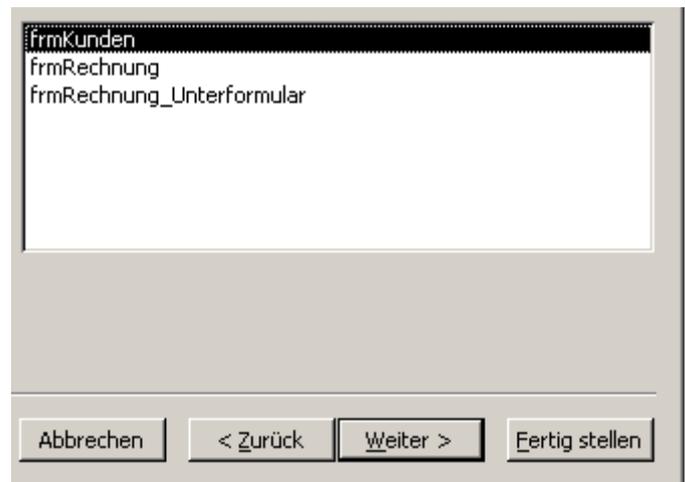
Öffnen Sie das Rechnungsformular in der Entwurfsansicht.

Fügen Sie eine neue Schaltfläche im Formularfuß ein.

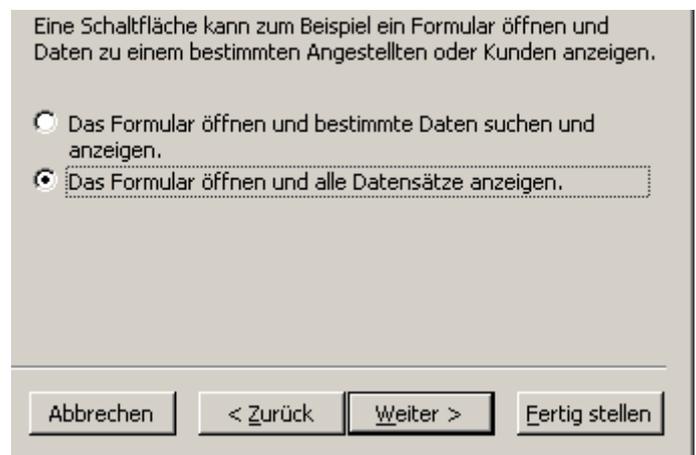
„Formularoperationen / Formular öffnen“.



Wählen Sie das Kundenformular.



Die Datensätze sollen angezeigt werden.



Wählen Sie „Text“.

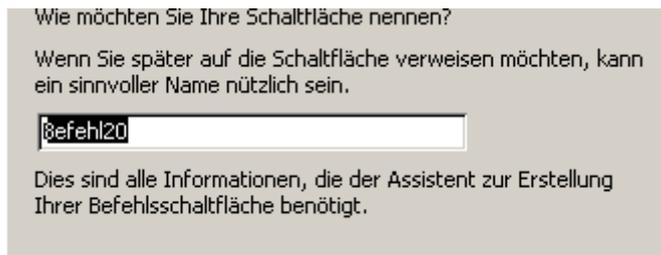
Geben Sie die Bezeichnung „Formular öffnen ein“.



Bestätigen Sie den Namen.

Anmerkung:

Im richtigen Leben wäre es sinnvoller, den Namen auf „btnKunden“ zu ändern.



Klicken Sie mit der rechten Maustaste auf die neue Schaltfläche.
Wählen Sie „Ereignis“.
Hier der Programmcode, der automatisch erzeugt wurde:

```
Private Sub Befehl20_Click()
On Error GoTo Err_Befehl20_Click

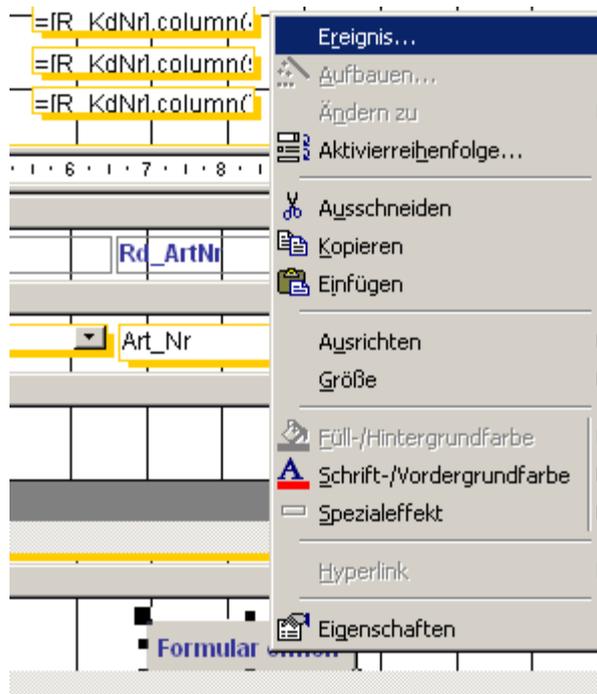
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmKunden"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Befehl20_Click:
Exit Sub

Err_Befehl20_Click:
MsgBox Err.Description
Resume Exit_Befehl20_Click

End Sub
```



Ereignis: Beim Doppelklicken

Beim Doppelklicken auf das Feld „R_KdNr“ soll ebenfalls das Kundenformular geöffnet werden. Stellen Sie dazu im Ereignis des Feldes R_KdNr „Beim Doppelklicken“ den Wert „Ereignisprozedur“ ein. Nun wird, wenn Sie auf das Feld doppelklicken ein Ereignis aufgerufen.



Ereignis: Beim Öffnen

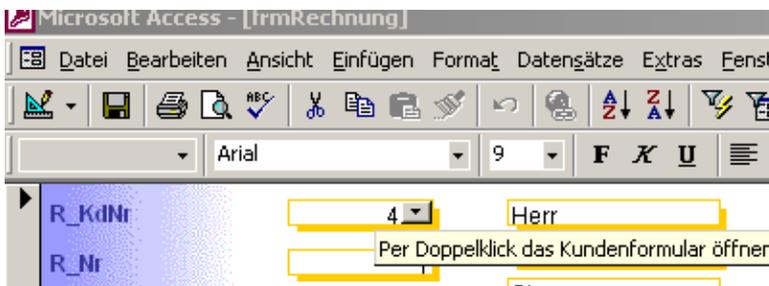
Die Funktion „DoCmd.OpenForm“ öffnet das Formular, das dahinter in Anführungszeichen angegeben wird. Hier: „frmKunden“.

```
Private Sub R_KdNr_DblClick(Cancel As Integer)
DoCmd.OpenForm "frmKunden"
End Sub
```

SteuerelementTip-Text

Der in dieser Zeile eingegebene Text wird angezeigt, wenn man den Mauszeiger auf das Feld bewegt.

Bewegen Sie den Mauszeiger in der Formularansicht auf das Feld „R_KdNr“.



Versehentliches Ändern unterbinden

Die Gefahr, dass eine abgeschlossene Rechnung geändert wird, ist in diesem Zustand der Programmierung noch sehr groß. Das Formular soll nun für die Eingabe gesperrt werden.

Wählen Sie aus den Ereignissen des Formulars „Beim Anzeigen“.



Ereignis: Beim Anzeigen

Dieses Ereignis tritt immer ein, wenn auf einen Datensatz geblättert wird, bzw. wenn das Formular geöffnet wird. Geben Sie den rechts dargestellten Code ein.

```
Private Sub Form_Current()
    me.Form.AllowEdits = False
End Sub
```

Unterformular sperren

Doppelklicken Sie auf das Unterformular. Geben Sie im Ereignis des Unterformulars „Beim Anzeigen“ den rechts dargestellten Code ein. Das Unterformular prüft beim Anzeigen, ob das Hauptformular gesperrt ist und wird in diesem Fall auch gesperrt bzw. entsperrt. Die Eigenschaft „Parent“ verweist auf das übergeordnete Formular „frmRechnung“.

```
Private Sub Form_Current()

    If Me.Parent.AllowEdits = True Then
        Me.AllowEdits = True
    Else
        Me.AllowEdits = False
    End If

End Sub
```

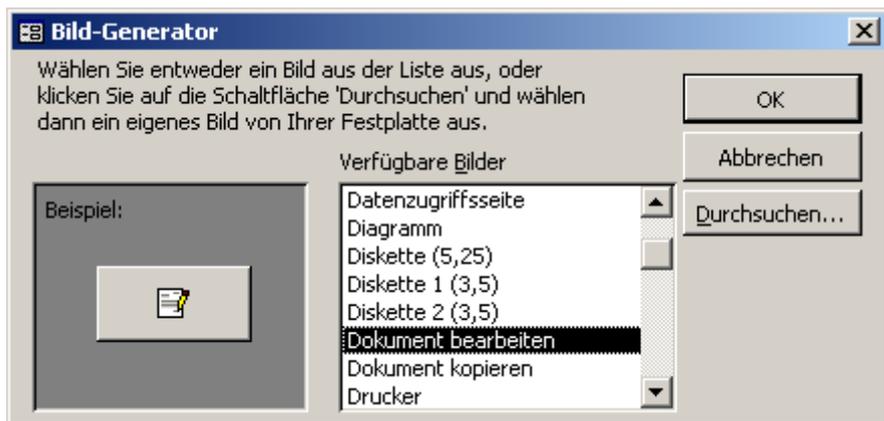
Formular entsperren

Nun ist das Formular eine Spur zu sicher: Wir können nichts mehr eingeben.

Wir benötigen eine Schaltfläche, mit welcher wir die Eingabe erlauben.

Erstellen Sie eine neue Schaltfläche im Formular.

Wählen Sie das Bild „Dokument bearbeiten“.

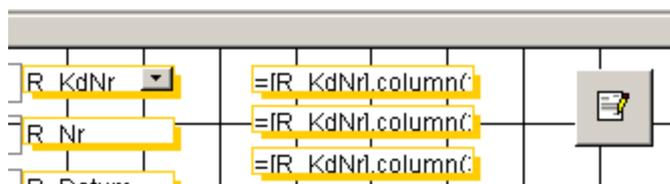


Ereignis: Beim Klicken

Klicken Sie mit der rechten Maustaste auf die neue Schaltfläche. Wählen Sie „Ereignis“.

Geben Sie den folgenden Programmcode ein:

Me.AllowEdits = True



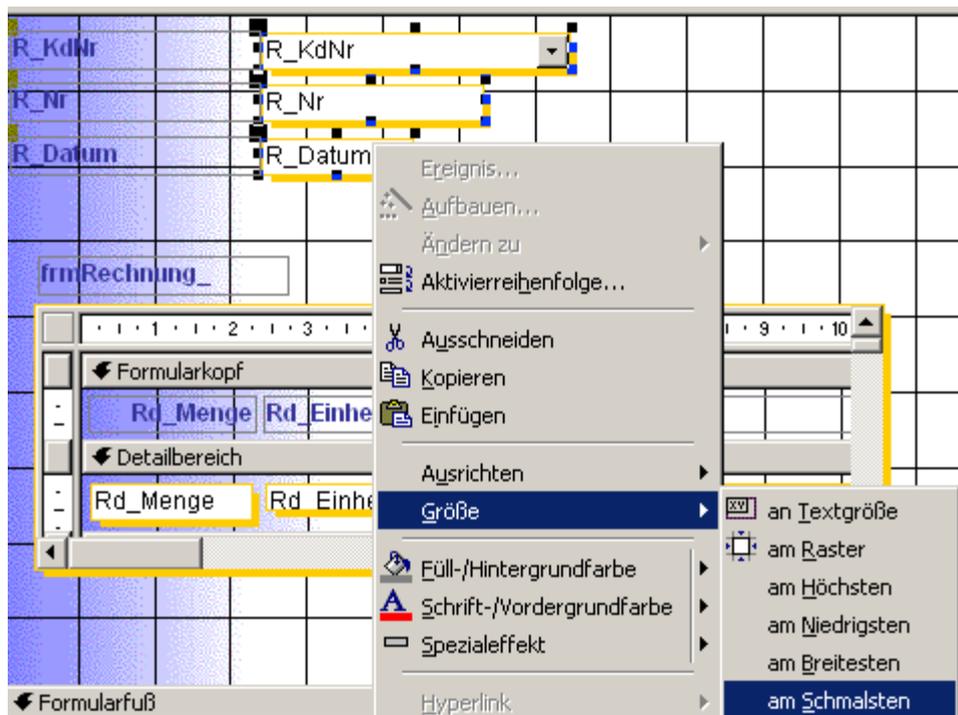
Größe angleichen

Markieren Sie die Felder z.B. über Fenster.

Wählen Sie im Kontextmenü „Größe“.

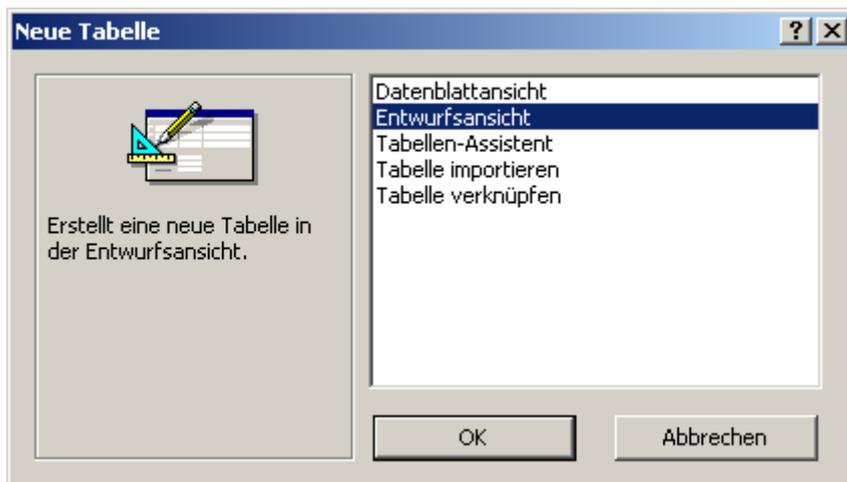
Wählen Sie „am Schmalsten“.

Speichern und schließen Sie das Formular.

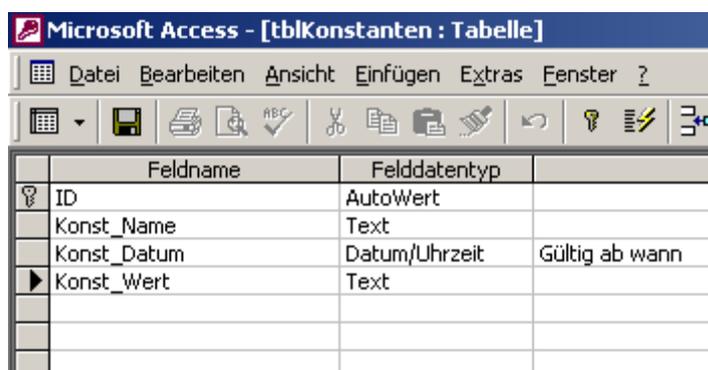


Konstantentabelle

Erstellen Sie eine neue Tabelle in der Entwurfsansicht.
 In dieser Tabelle sollen Konstanten hinterlegt werden.



Die Konstanten können bei Bedarf einfach über die Tabelle geändert werden.
 Konstanten können auch direkt im Programmcode eingefügt werden.
 Das ist z.B. bei „dauerhaften“ Konstanten wie der Kreiszahl Pi sinnvoll. Die Mehrwertsteuer ändert sich von Zeit zu Zeit und wird deshalb so angelegt, dass der Benutzer sie – wenn erforderlich - ergänzen kann.



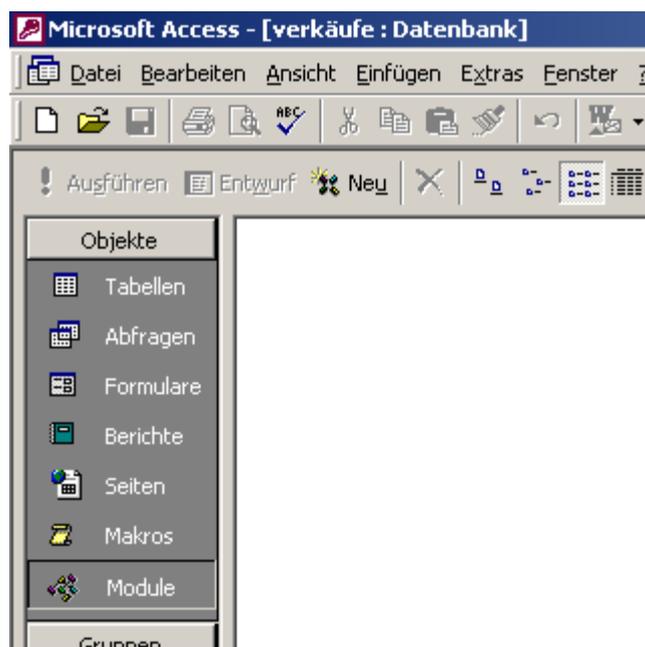
In der Tabelle wird die Bezeichnung der Konstanten und das Änderungsdatum, ab wann sie gilt, eingetragen.



Modul erstellen

Erstellen Sie ein neues Standardmodul.
 Die Funktionen/Prozeduren in diesem Modul sind grundsätzlich aus allen Objekten der Datenbank aufrufbar. (Z.B.: Schlüsselwort PUBLIC).

Wechseln Sie dazu zum Eintrag „Module“.
 Klicken Sie auf „NEU“ in der darüber angezeigten Symbolleiste.



Domänenfunktion: Dlookup

Mit der Funktion „Dlookup“ können Sie Werte aus Tabellen einlesen. Sie ist ähnlich aufgebaut, wie eine Abfrage, benötigt aber erheblich weniger Zeilen.

Variable = Dlookup(„Feldname“, „Tabellenname“, „Bedingung“)

```
Option Compare Database
Option Explicit
```

```
Function MwStHolen(datRechnungsDatum As Date)
    MwStHolen = Dlookup("Konst_Wert", "tblKonstanten", "Konst_Name='mwst'")
End Function
```

In der oben dargestellten Form wird nicht berücksichtigt, dass sich die Mehrwertsteuer abhängig vom Datum ändert – leider.

Deshalb wird in der folgenden Funktion auch das Rechnungsdatum berücksichtigt:

```
Function MwStHolen(datRD As Date)
    MwStHolen = Dlookup("Konst_Wert", "tblKonstanten", "Konst_Name='mwst' and Konst_datum <#" & sqlDatum(datRD) & "#")
End Function
```

Datum in Abfragen

Das Datum in der Abfrage muss so umgewandelt werden, dass es richtig interpretiert werden kann.

Dazu ist eine neue Funktion erforderlich.

Die Funktion zerlegt das Datum in seine Bestandteile Tag, Monat und Jahr und fügt es dann in der amerikanischen Schreibweise neu zusammen.

Dann wird es an den Funktionsnamen „sqlDatum“ zurückgegeben und kann so in Berechnungen oder in Berichten und Abfragen verwendet werden.

```
Function sqlDatum(Datum As Date)
    Dim strJ As String, strT As String, strM As String
    strJ = Year(Datum)
    strT = Day(Datum)
    strM = Month(Datum)

    sqlDatum = strM & "/" & strT & "/" & strJ
End Function
```

Erstellen Sie zum Testen der Funktion eine neue Prozedur. Stellen Sie den Cursor in die Zeile MwStHolen... und drücken Sie die Taste F8, um die Funktion zu durchlaufen.

```
Sub MwstProbieren()
    MwStHolen ("10.02.07")
End Sub
```

Rufen Sie den Bericht „rptRechnung“ auf. Schalten Sie in die Entwurfsansicht. Klicken Sie im Eigenschaftenfenster neben „Datenherkunft“ die 3 Punkte, um die Abfrage aufzurufen. Wählen Sie das Feld „MwSt“. Ändern Sie den Wert wie rechts dargestellt. Nun wird die Mehrwertsteuer mit dem Wert aus der Konstantentabelle berechnet.



Format-Funktion

Schließen Sie die Abfrage.

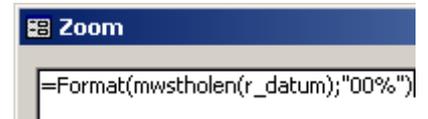
Fügen Sie im Berichtsentwurf ein neues Feld „txtSteuerSatz“ ein. Geben Sie als Steuerelementinhalt den rechts dargestellten Code ein.

Die Funktion Format stellt den Mehrwertsteuersatz in der gewünschten Form dar:

Syntax: =Format(Wert;"00%").

2-stellig im Prozentformat. Im Modul kann die Funktion ebenfalls verwendet werden, dort muss aber anstatt des Strichpunktes (Semikolons) ein Komma als Listentrennzeichen verwendet werden.

		Summe netto:	=Summe([po.
=Format(mwst:		MwSt-Betrag:	=Summe([mw
		Summe brutto:	=[summenett



Hier die Übersicht über die 3 Prozeduren:

```
Option Compare Database
```

```
Option Explicit
```

```
Sub MwstProbieren()  
    MwStHolen ("10.02.06")  
End Sub
```

```
Function MwStHolen(datRD As Date)
```

```
    MwStHolen = DLookup("Konst_Wert", "tblKonstanten", "Konst_Name='mwst' and Konst_datum <#" & sqlDatum(datRD) & "#")  
End Function
```

```
Function sqlDatum(Datum As Date)
```

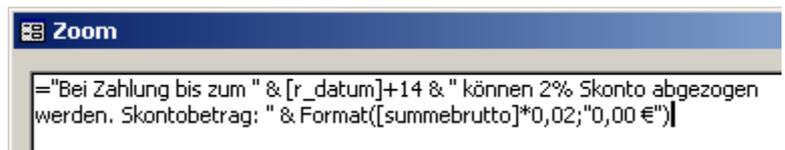
```
    Dim strJ As String, strT As String, strM As String  
    strJ = Year(Datum)  
    strT = Day(Datum)  
    strM = Month(Datum)
```

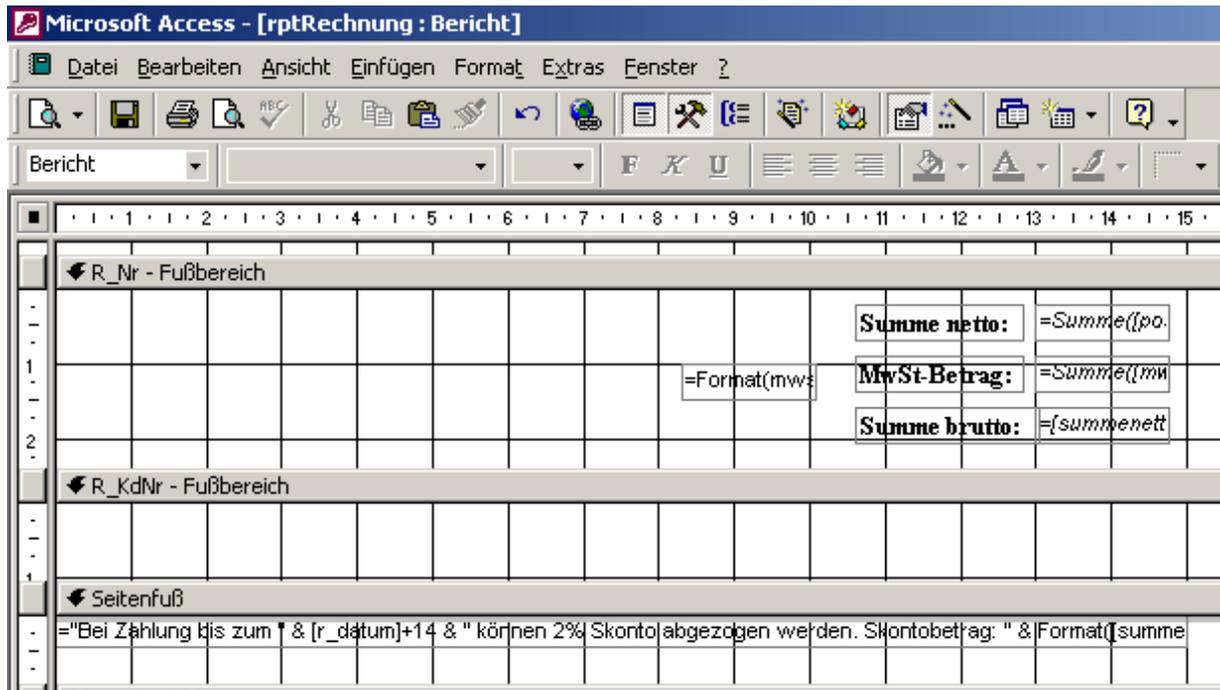
```
    sqlDatum = strM & "/" & strT & "/" & strJ
```

```
End Function
```

Skontobetrag berechnen

Geben Sie im Seitenfuß des Berichts die Funktion für den Skontobetrag ein.





Bei Zahlung bis zum 15.12.2007 können 2% Skonto abgezogen werden. Skontobetrag: 74,02 €

ABC-Analyse der Lagerartikel

Die Lagerartikel sollen nun nach Zusammensetzung des Wertes dargestellt werden. Basis hierfür sind die Werte der Einzelartikel.

Erstellen Sie eine neue Abfrage in der Entwurfsansicht.

Wählen Sie die Tabelle „tblArtikel“ als Basis.

Die Bereiche wurden willkürlich auf die Stufen bis 100, bis 1000, über 1000 Euro Einzelwert festgelegt.



Fügen Sie das Feld „Art_Preis“ 2 Mal ein.

Klicken Sie auf „Funktionen“.

Die Abfrage wird nun gruppiert.

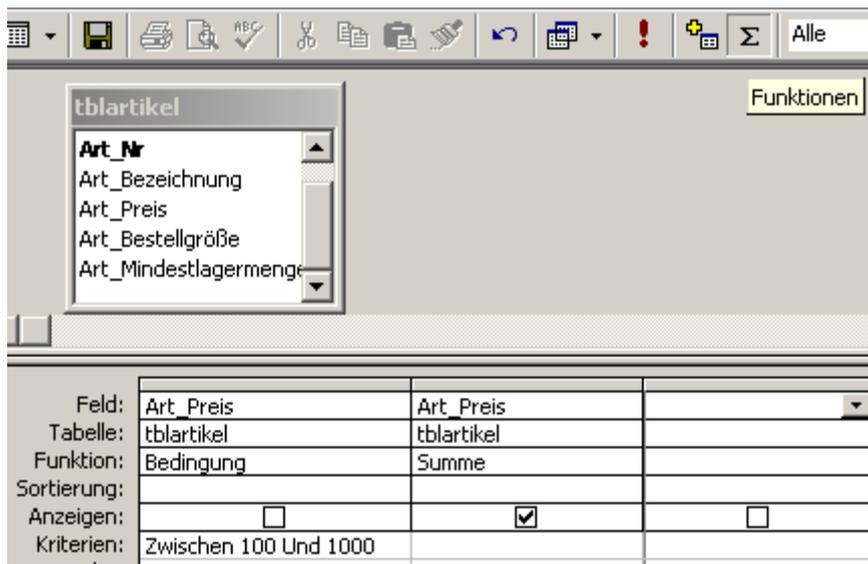
Geben Sie die Kriterien ein, die angezeigt werden sollen:

Art_Preis: Funktion „Bedingung“:
Dieses Feld wird nicht angezeigt, sondern dient zum Festlegen der Bedingung.

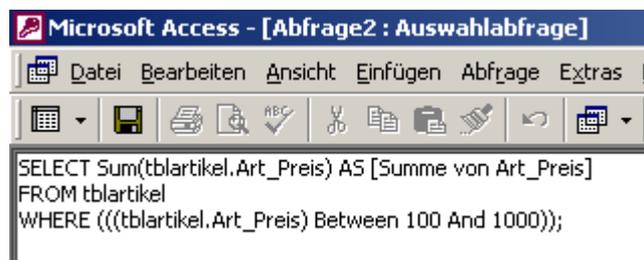
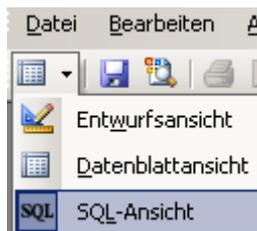
Bedingung in der Kriterienzeile.
Der Preis soll zwischen 100 und 1000 Euro liegen.

Sum

Art_Preis: Summe bilden.
Dieses Feld wird angezeigt.

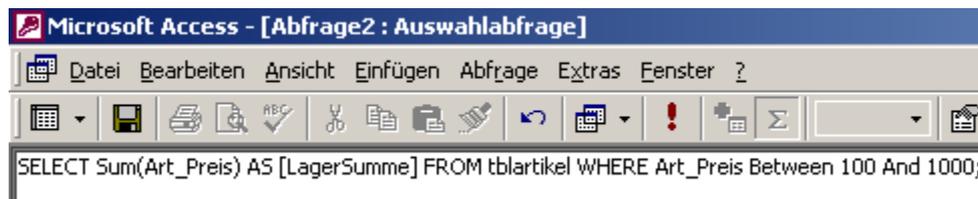


Schalten Sie um in die SQL-Ansicht.
Menü Ansicht oder durch Klicken auf die Schaltfläche.

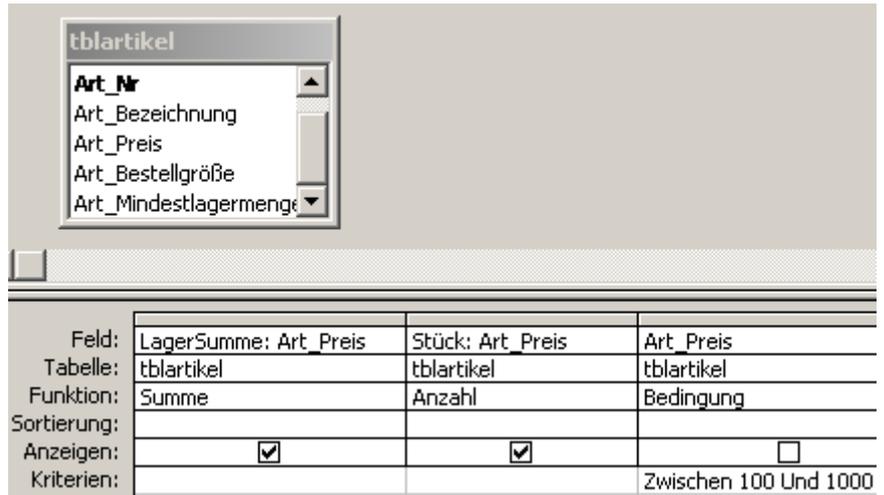


Sie sehen die SQL-Abfrage, die in Access automatisch erstellt wurde. Diese kann z.B. im Programmcode weiterverwendet werden.

Der Programmcode rechts wurde eine wenig gestutzt: Der Bezug „tblartikel“ kann entfernt werden, weil es sich nur um eine Tabelle handelt. Der Bezug wäre nötig bei mehreren Tabellen mit gleichlautenden Feldern.



Die Anzeige im Abfrageentwurfsbereichs (QBE) = Query by Example.

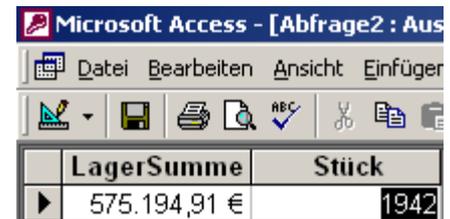


Count

Die Count-Funktion liefert die Anzahl der Datensätze, in diesem Fall die Anzahl der Artikel.

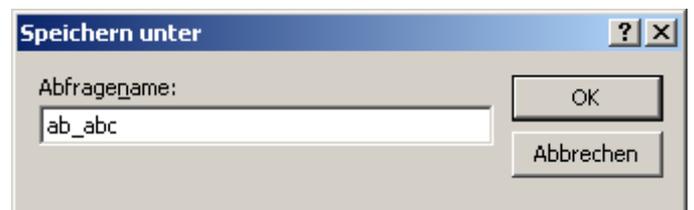


Hier die Abfrage in der Datenblattansicht. Alle Artikel mit einem Wert zwischen 100 und 1000 Euro repräsentieren einen Lagerwert von 575.194,91 Euro. Die Anzahl der Artikel beträgt 1942. Von jedem Artikel ist ein Stück auf Lager.



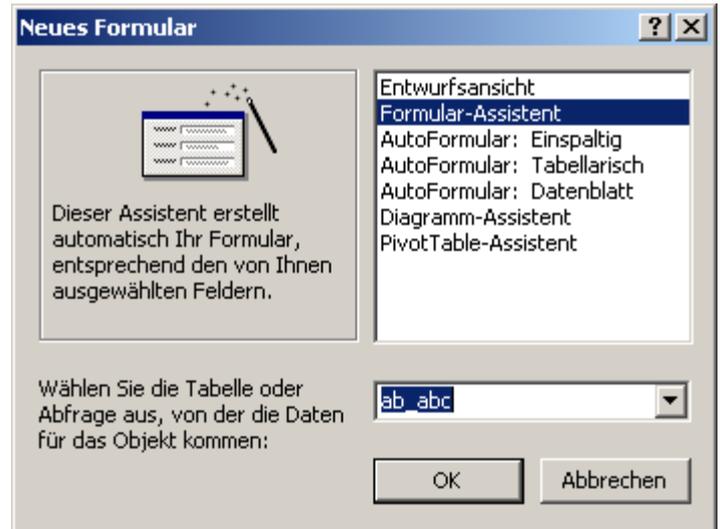
```
SELECT Sum(tblartikel.Art_Preis) AS LagerSumme,
Count(tblartikel.Art_Preis) AS [Stück]
FROM tblartikel
WHERE Art_Preis Between 100 And 1000;
```

Speichern Sie die Abfrage unter "ab_ABC".

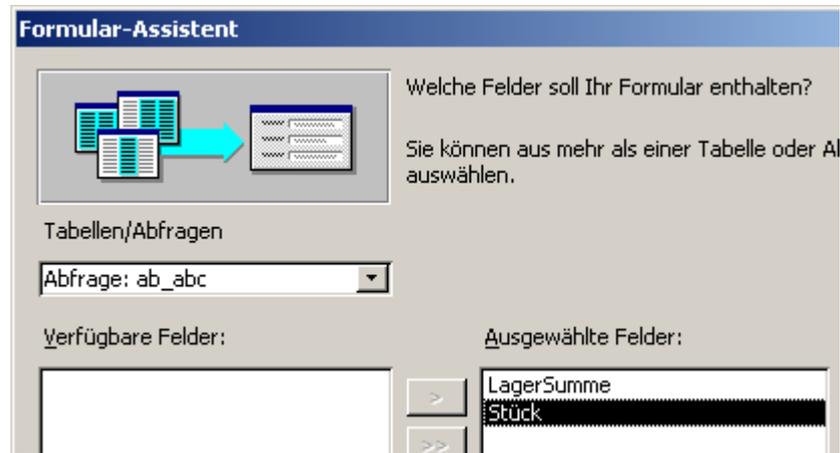


Formular erstellen

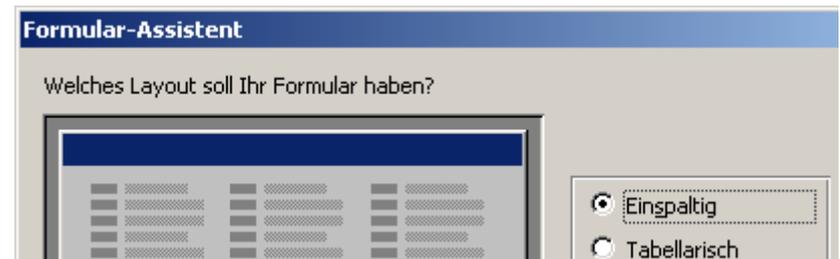
Erstellen Sie das Formular frmStatistik über den Formularassistenten.



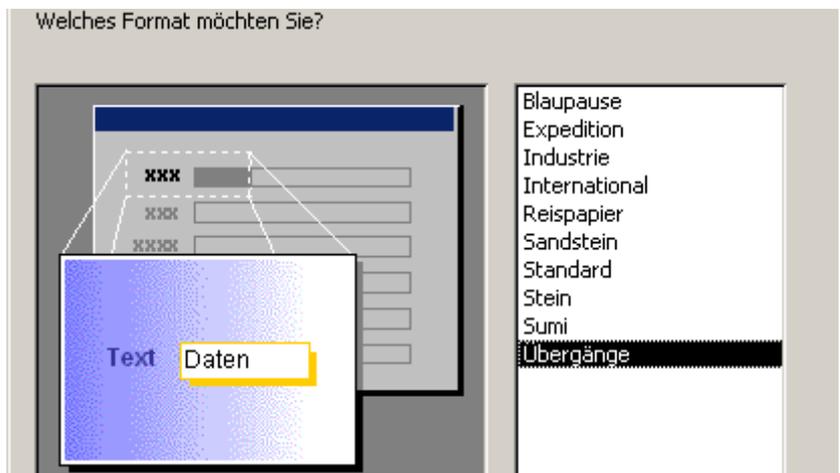
Wählen Sie die Werte LagerSumme und Stück aus der Abfrage ab_abc.



Bestätigen Sie „Einspaltig“.



Wählen Sie „Übergänge“.



Benennen Sie das Formular „frmStatistik“.



Markieren Sie die beiden Felder.

Drücken Sie die Tastenkombination STRG + C (Kopieren) und 2 Mal STRG + V (Einfügen).



Benennen Sie die Felder von rechts nach links:

LagerSummeA, StückA,

LagerSummeB, StückB

LagerSummeC, StückC.

Sie sollten anders als hier im Beispiel die Felder folgendermaßen benennen:

txtLagersummeA und txtStueckA. Vermeiden Sie Umlaute.



Verweise

Schalten Sie in die Codeansicht.

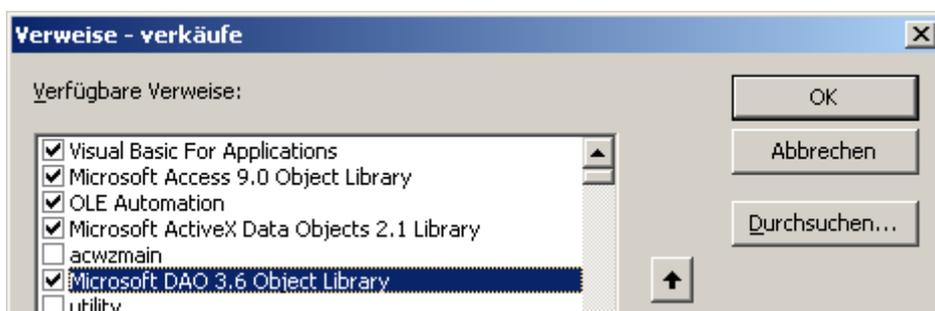
Klicken Sie auf

„Extras/Verweise“.

Wählen Sie die Bibliothek

für die Datenanbindung

„Microsoft DAO 3.6“.



Domänenfunktionen

Die Domänenfunktionen können nur in Access verwendet werden.

Vor und nach Texten steht ein „Hochkomma“.

DSum: MsgBox = DSum(„Rückgabewert“, „Tabelle“, „Bedingung“)

Dsum: Ermittelt die Summe aus einem Datenbestand. (s.u.)

Dcount: Ermittelt die Anzahl aus einem Datenbestand (s.u.)

Dlookup: Sucht einen Eintrag in einem Datenbestand:

Dlookup(„Art_Preis“, „Stammdaten“, „Art_Nr=“ & art_nr &“““).

Davg: Ermittelt den Durchschnitt über einen Datenbestand.

Geben Sie im Formularentwurf im Ereignis „Beim Öffnen“ die folgende Routine ein:

```
Option Compare Database
Option Explicit
```

```
Private Sub Form_Open(Cancel As Integer)

    ' Summe
    Me.LagerSummeA = DSum("Art_Preis", "tblArtikel", "art_preis >1000")
    Me.LagerSummeB = DSum("Art_Preis", "tblArtikel", "art_preis >100 and art_preis <=1000")
    Me.LagerSummeC = DSum("Art_Preis", "tblArtikel", "art_preis <100")

    ' Anzahl
    Me.StückA = DCount("Art_Preis", "tblArtikel", "art_preis >1000")
    Me.StückB = DCount("Art_Preis", "tblArtikel", "art_preis >100 and art_preis <=1000")
    Me.StückC = DCount("Art_Preis", "tblArtikel", "art_preis <100")

Exit Sub
```

In die Felder LagerSummeA – C werden die Summen der Lagerwerte über DSum-Funktionen eingetragen.

In die Felder StückA – C wird über DCount die Anzahl der jeweiligen Artikel eingetragen.

Alternativ hätte man über DAO ebenfalls die erforderlichen Werte erhalten:

Auch über ADO, eine weitere Zugriffsmöglichkeit auf Datenbestände.

```
' Alternativ
Dim db As DAO.Database, r As DAO.Recordset
Set db = CurrentDb

Set r = db.OpenRecordset("SELECT Sum(tblartikel.Art_Preis) AS LagerSumme, Count(tblartikel.Art_Preis) AS [Stück] " _
& " FROM tblartikel WHERE Art_Preis Between 100 And 1000;")

Me.LagerSummeA = r!LagerSumme

Set r = db.OpenRecordset("SELECT Sum(tblartikel.Art_Preis) AS LagerSumme, Count(tblartikel.Art_Preis) AS [Stück] " _
& " FROM tblartikel WHERE Art_Preis > 1000;")

Me.LagerSummeB = r!LagerSumme

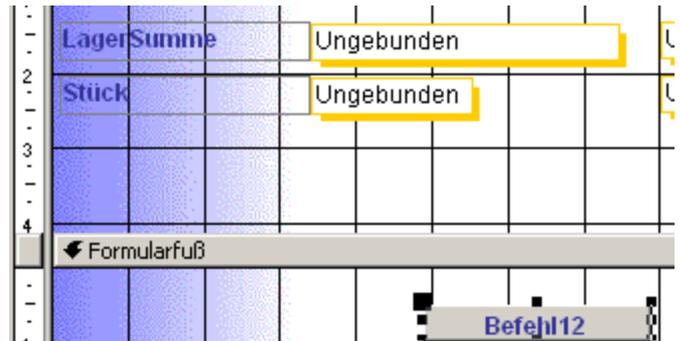
Set r = db.OpenRecordset("SELECT Sum(tblartikel.Art_Preis) AS LagerSumme, Count(tblartikel.Art_Preis) AS [Stück] " _
& " FROM tblartikel WHERE Art_Preis <100;")

Me.LagerSummeC = r!LagerSumme

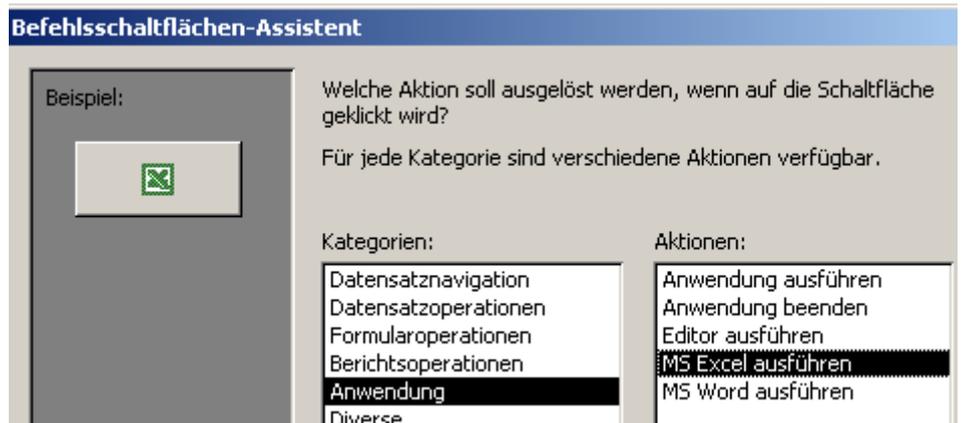
End Sub
```

Daten nach Excel

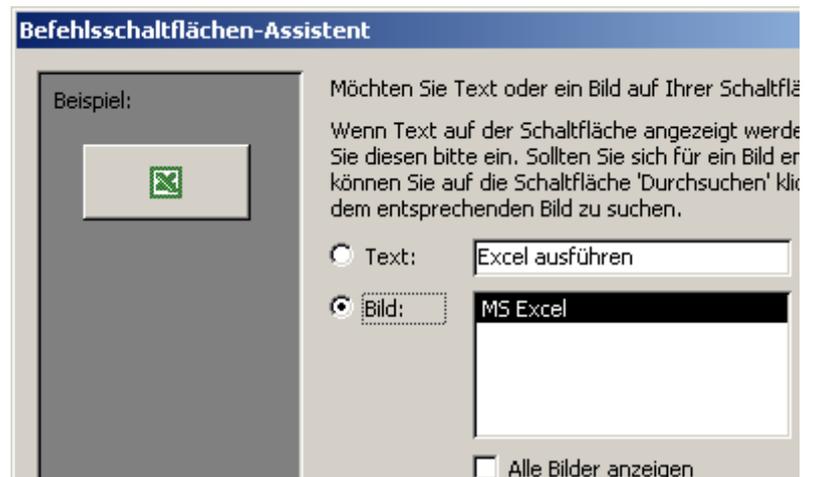
Die Formulardaten sollen nun nach Excel übertragen werden. Dort soll automatisiert ein Diagramm erstellt werden. Erstellen Sie eine neue Befehlsschaltfläche im Formularfuß.



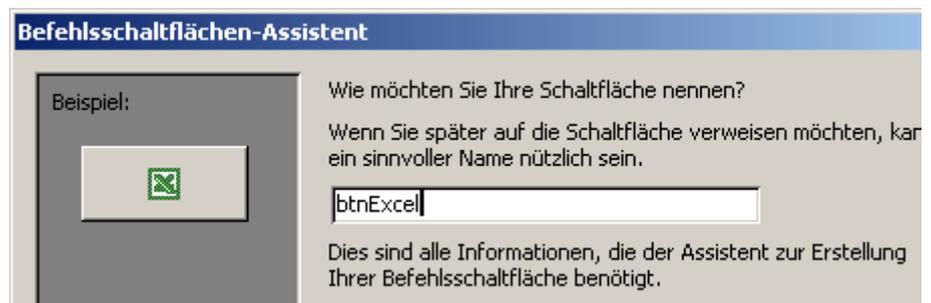
Wählen Sie im Assistenten „Anwendung“. „Ms-Excel ausführen“.



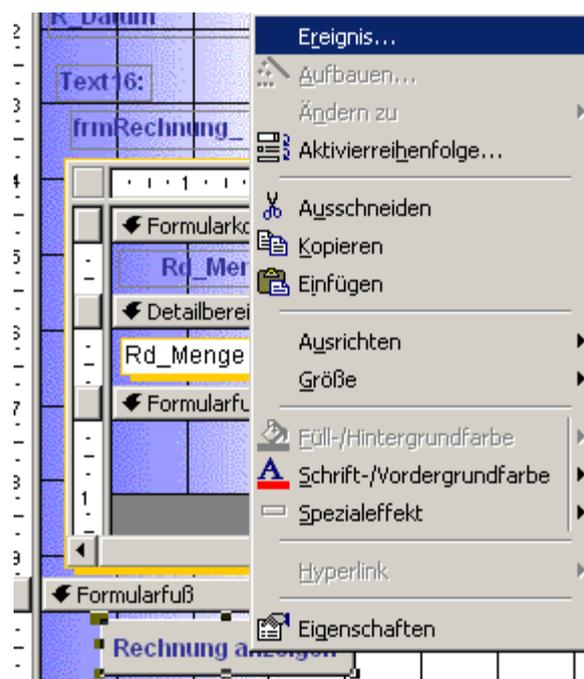
Wählen Sie die Schaltfläche.



Benennen Sie die Schaltfläche „btnExcel“.



Klicken Sie mit der rechten Maustaste auf die Schaltfläche und wählen Sie „Ereignis“ oder klicken Sie im Eigenschaftenfenster auf die 3 Punkte neben dem Eintrag „Beim Klicken“.



Objekt Excel

Die Variable oApp repräsentiert das Programm Excel.

Bevor man mit Excel arbeiten kann, muss es geöffnet werden.

Der Variablen oApp muss eine Excelanwendung zugeordnet werden:

Set oApp = CreateObject....

Excel muss sichtbar geschaltet werden.

oApp.Visible = True

Nun sollen die Werte aus dem Formular nach Excel übertragen werden.

Verwenden Sie am besten die Funktion „Range“ zum Adressieren von Zellen.

oApp.Range(„A3“) = 10.

Schreibt den Wert 10 in die Zelle A3.

```
Private Sub btnExcel_Click()
    On Error GoTo Err_btnExcel_Click

    Dim oApp As Object

    Set oApp = CreateObject("Excel.Application")
    oApp.Visible = True
    'Only XL 97 supports UserControl Property
    On Error Resume Next
    oApp.UserControl = True

    oApp.range("A3").Value = "Lagersumme: A"
    oApp.range("A4").Value = "Lagersumme: B"
    oApp.range("A5").Value = "Lagersumme: C"

    oApp.range("B3").Value = Me.LagerSummeA
    oApp.range("B4").Value = Me.LagerSummeB
    oApp.range("B5").Value = Me.LagerSummeC

    oApp.range("C3").Value = Me.StückA
    oApp.range("C4").Value = Me.StückB
    oApp.range("C5").Value = Me.StückC

Exit_btnExcel_Click:
    Exit Sub

Err_btnExcel_Click:
    MsgBox Err.Description
    Resume Exit_btnExcel_Click

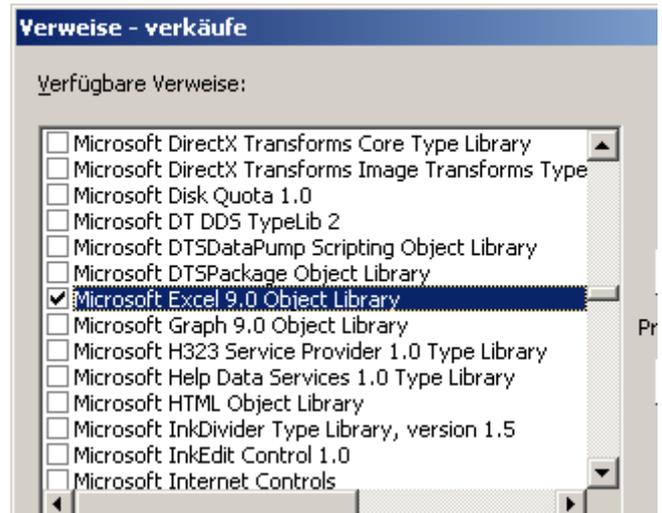
End Sub
```

Verweis auf Excel

Damit die Objektbibliothek von Excel adressiert werden kann, muss zuerst ein Verweis gesetzt werden:

Extras/Verweise im Code-Fenster.

Excel9 verweist dabei auf eine ältere Excel-Version. Die Bibliothek muss umgestellt werden, wenn Sie eine neuere Version verwenden (Excel 10, 11 oder 12 = 2007).

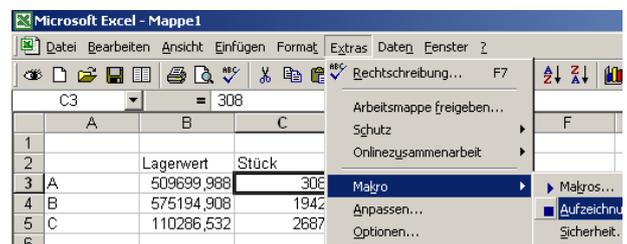


Excel: Makro Aufzeichnen

In Excel soll nun ein Diagramm erstellt werden.

Zeichnen Sie diesen Vorgang auf. So sparen Sie sich viel Tipparbeit.

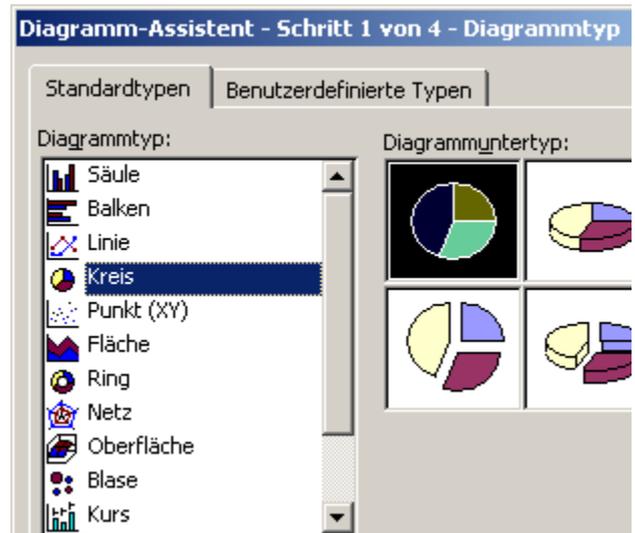
Extras / Makro / Aufzeichnen.



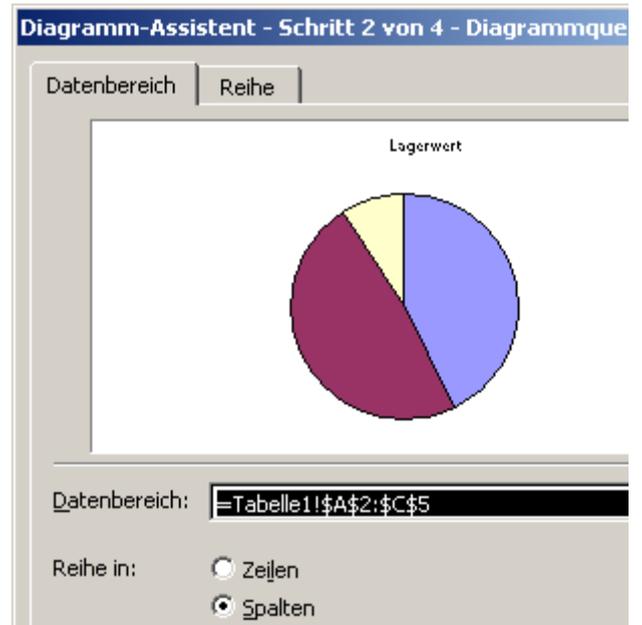
Markieren Sie den Bereich, der im Diagramm dargestellt werden soll: A2 bis C5.

Wählen Sie das Kreisdiagramm.

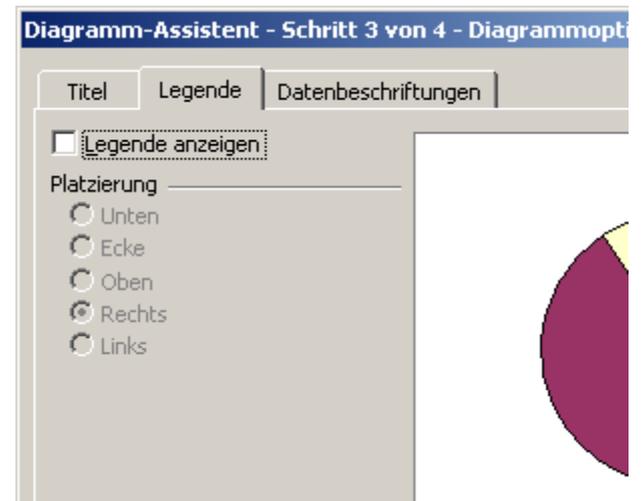
Klicken Sie „Weiter“.



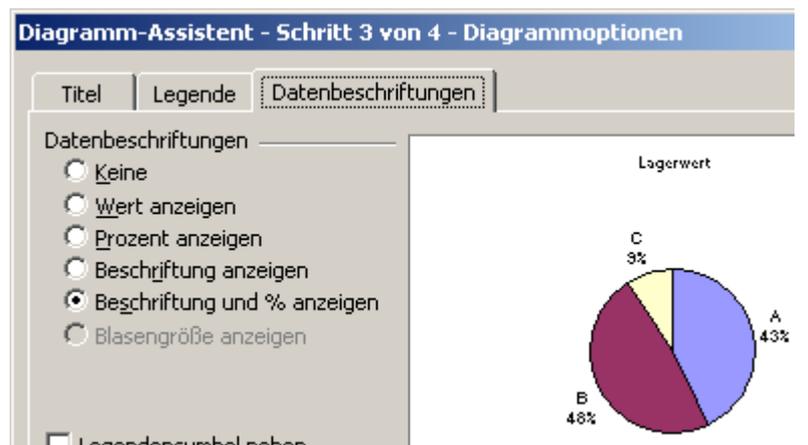
Klicken Sie „Weiter“.



Blenden Sie die Legende aus.



Wählen Sie das Register „Datenbeschriftung“.
Wählen Sie „Beschriftung und % anzeigen“.
Klicken Sie auf „Weiter“.



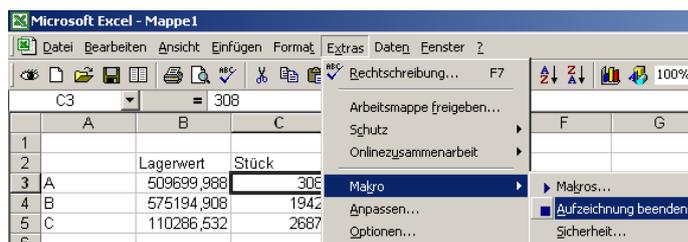
Klicken Sie auf „Fertigstellen“.



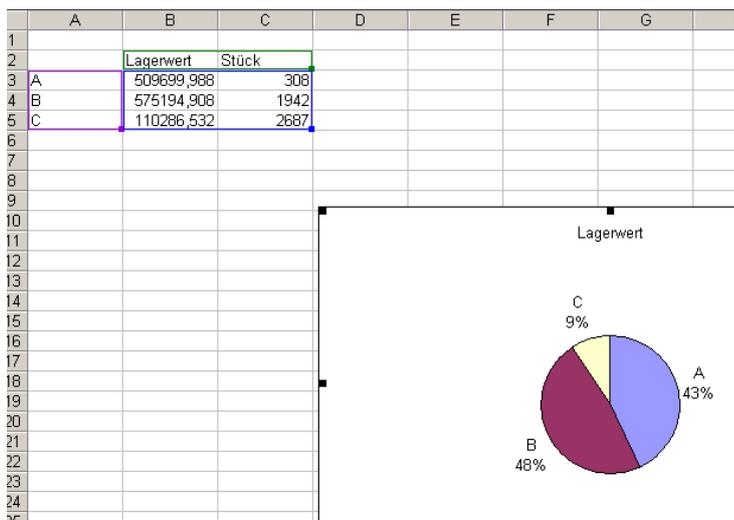
Klicken Sie auf „Aufzeichnung beenden“:



Beachten Sie, dass Sie die Aufzeichnung beenden, sobald der gewünschte Ablauf fertig ist. Sonst kann es zu lustigen Ergebnissen kommen. Berechnungen sollten aber nicht lustig, sondern korrekt sein.



Das Diagramm in Excel.



Klicken Sie auf „Extras / Makro / Makros“.

Wählen Sie das Makro, wählen Sie „Bearbeiten“.

Kopieren Sie den erstellten Programmcode in Ihr Formularmodul (frmStatistik) in Access.

Fügen Sie es unterhalb der bestehenden Zeilen an.

Mit der Anweisung „With oApp...End With“ erreichen Sie, dass nicht vor jedem Bezug zu Excel „oApp“ stehen muss.

Es reicht, wenn Sie einen Punkt vor die Funktion aus der Excelbibliothek schreiben.

```

oApp.Range("C4").Value = Me.StückB
oApp.Range("C5").Value = Me.StückC

'Diagramm
With oApp
    .Charts.Add
    .ActiveChart.ChartType = xlPie
    .ActiveChart.SetSourceData Source:=Sheets("Tabelle1").Range("A2:C5"), PlotBy _
        :=xlColumns
    .ActiveChart.Location Where:=xlLocationAsObject, Name:="Tabelle1"
    .ActiveChart.HasLegend = False
    .ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowLabelAndPercent, LegendKey _
        :=False, HasLeaderLines:=True
End With

End Sub

```

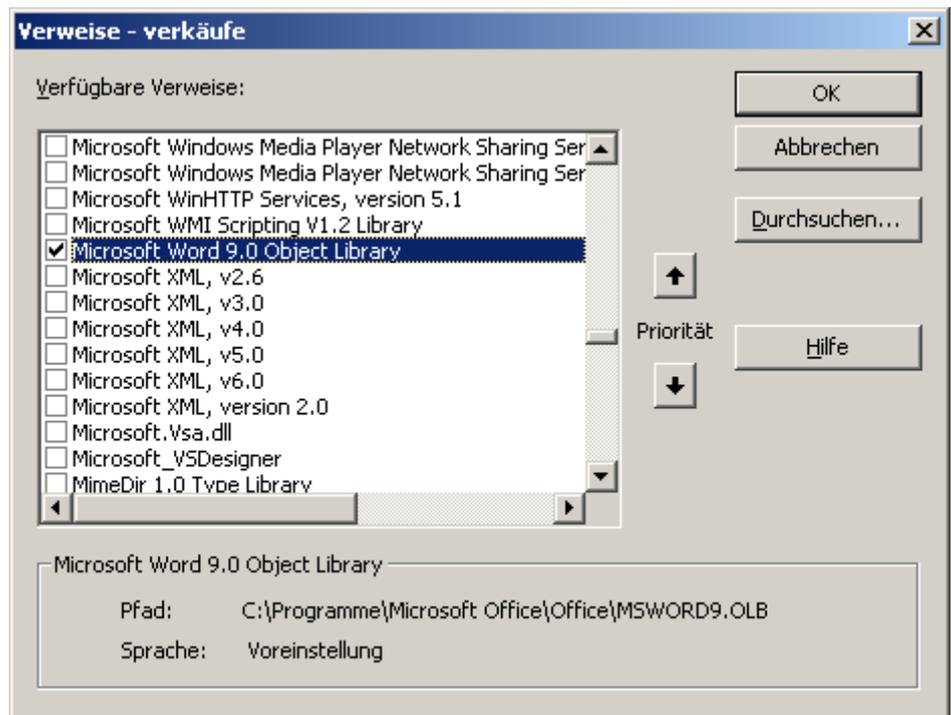
In einem weiteren Schritt soll nun das Diagramm und die einzelnen Werte aus Access bzw. Excel nach Word übertragen werden.

Word aus Access steuern

Aus den Access-Daten und dem Diagramm aus Excel soll nun ein Worddokument erstellt werden. Suchen Sie im VBA-Editor in Extras / Verweise den Eintrag „Microsoft Word 9.0 (oder 10.0, 11.0, 12.0) je nach Word-Version, die Sie installiert haben. Der folgende Programmcode wurde für Word 9 erstellt.

Daten nach Word übertragen

Öffnen Sie in Access ein Modul. Schalten Sie z.B. im Formular in die Codeansicht. Klicken Sie auf „Extras / Verweise“. Wählen Sie die Wordbibliothek. Achten Sie darauf, dass es sich in der angezeigten Version um Word 2000 handelt (= Word 9.0).



Wordvorlage

Erstellen Sie das Schreiben an die Geschäftsleitung wie rechts dargestellt.

Trans-IT GmbH
Edith Müller Geschäftsleitung
Bahnhofsweg 2
77234 Lüthen

Dann fügen Sie Textmarken an den Stellen im Text ein, wo die Werte aus Access bzw. das Diagramm aus Excel abgelegt werden sollen. Setzen Sie den Cursor hinter den Doppelpunkt nach „100 Euro“. Geben Sie ein Leerzeichen ein.

Lagerkennzahlen

Sehr geehrte Frau Müller,

hier erhalten Sie die aktuellen Zahlen des Lagerwertes:

Lagerwert im Bereich A kleiner 100 Euro:
Lagerwert im Bereich A ab 100 bis 1000 Euro:
Lagerwert im Bereich A größer 1000 Euro:

Mit freundlichen Grüßen
Walter Eder

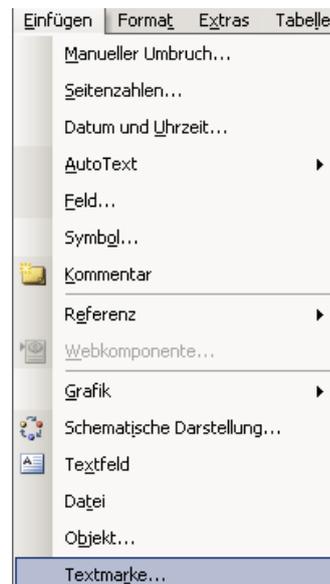
Rufen Sie im Menü Einfügen die Option Textmarke auf.

Geben Sie die Textmarken Kleiner100 ein.

Setzen Sie den Mauszeiger hinter den Doppelpunkt ...bis 1000 Euro: Rufen Sie die Textmarke auf: Benennen Sie sie Bis1000.

Setzen Sie den Mauszeiger hinter den Doppelpunkt ...größer als 1000 Euro:

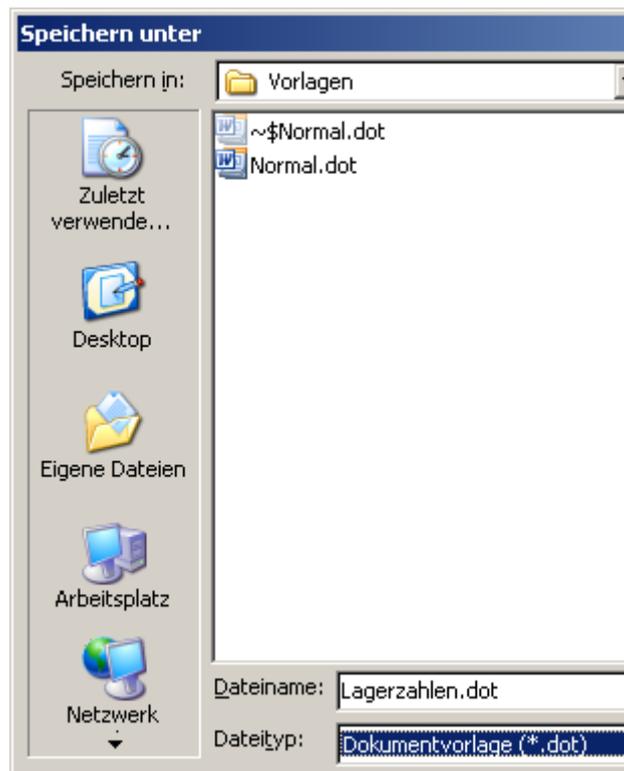
Rufen Sie die Textmarke auf: Benennen Sie sie Über1000.



Setzen Sie den Mauszeiger eine Zeile weiter nach unten und fügen Sie die Textmarke „Diagramm“ ein.

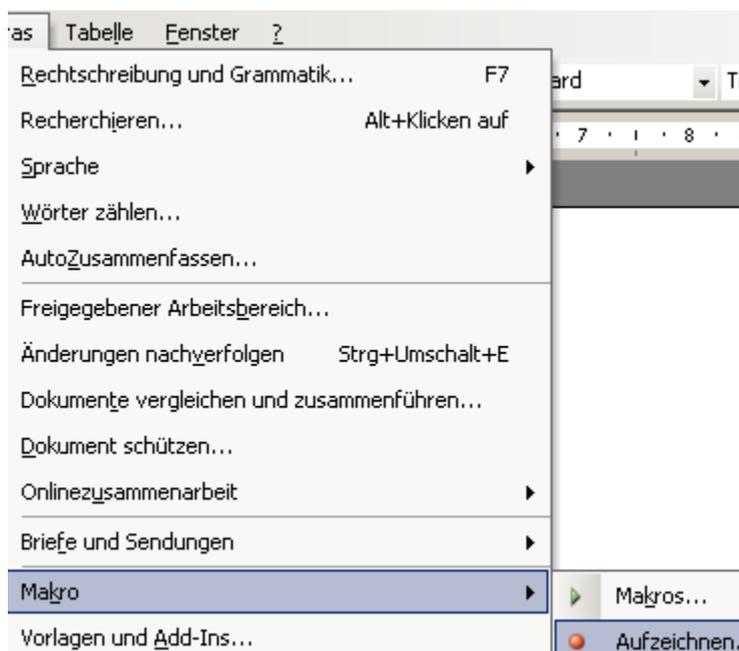
Vorlage speichern

Speichern Sie das Dokument als Vorlage (Dateityp Dokumentvorlage wählen) „Lagerzahlen.dot“. Schließen Sie das Dokument.



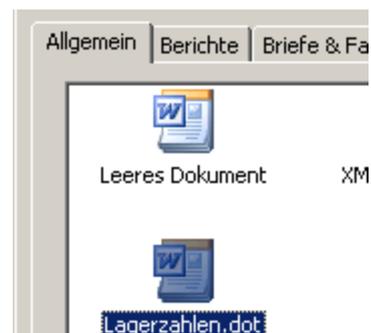
Makro aufzeichnen

Rufen Sie im Menü Extras / Makro / Aufzeichnen auf. Bestätigen Sie den Makronamen.



Die Aufzeichnung beginnt

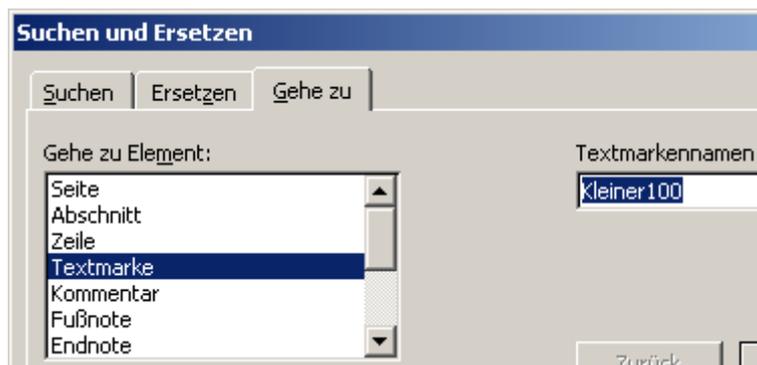
Erstellen Sie ein neues Dokument aus der Vorlage „Lagerzahlen.dot“.



Textmarken

Rufen Sie den Menüpunkt Bearbeiten / Gehe zu auf.

Wählen Sie das Element: Textmarke.
Wählen Sie in der Auswahl der Textmarken „Kleiner100“, wählen Sie dann Bis1000, dann Über1000, dann Diagramm.
Diese Auswahl wird jeweils in das Makro geschrieben. So ersparen Sie sich das spätere Heraussuchen, wenn Sie die Textmarken per VBA ansteuern.



Beenden Sie die Aufzeichnung.

Rufen Sie im Menü Extras / Makro / Makros auf. Wählen Sie bearbeiten.

Kürzen Sie das automatisch aufgezeichnete Makro auf die dargestellten Zeilen. Das Vorlageverzeichnis sieht auf Ihrem PC anders aus als hier dargestellt.

Kopieren Sie die dargestellten Zeilen (ohne Sub Makro1() und End Sub) in den Zwischenspeicher: (Markieren, STRG + C).

Sub Makro1()

Documents.Add Template:= _

"C:\Dokumente und Einstellungen\Franz Ertl\Anwendungsdaten\Microsoft\Vorlagen\Lagerzahlen.dot"

Selection.GoTo What:=wdGoToBookmark, Name:="Kleiner100"

Selection.GoTo What:=wdGoToBookmark, Name:="Bis1000"

Selection.GoTo What:=wdGoToBookmark, Name:="Über1000"

Selection.GoTo What:=wdGoToBookmark, Name:="Diagramm"

Selection.Find.ClearFormatting

End Sub

Programmcode aus Word in das Formular übertragen

Öffnen Sie das Formular Statistik.
Schalten Sie in die Entwurfsansicht.
Schalten Sie in die Codeansicht.

Erzeugen Sie die Variable: Dim oW as Object.
Weisen Sie der Variablen den Wert zu:
Schalten Sie das Word-Dokument sichtbar:

```
.ActiveChart.SetSourceData Source:=Sheet
    :=xlColumns
.ActiveChart.HasLegend = False
.ActiveChart.ApplyDataLabels Type:=xlDat
    :=False, HasLeaderLines:=True
End With
```

```
Dim oW As Object
Set oW = CreateObject("word.application")
oW.Visible = True
```

```
Set oW = CreateObject(„Word.Application“)
oW.visible = True
```

Fügen Sie den Programmcode im Modul des Formulars ein und ergänzen Sie ihn wie unten dargestellt. Aus dem Formular werden die Inhalte der Textfelder in das Worddokument übertragen und an der jeweiligen Textmarke positioniert.

```
.ActiveChart.ChartArea.Copy

End With

'Word starten
Dim oW As Object
Set oW = CreateObject("word.application")
oW.Visible = True

oW.Documents.Add Template:= _
    "C:\Dokumente und Einstellungen\Franz Ertl\Anwendungsdaten\Microsoft\Vorlagen\Lagerzahlen.dot" _
    , NewTemplate:=False, DocumentType:=0

oW.Selection.GoTo What:=wdGoToBookmark, Name:="kleiner100"
oW.Selection.Find.ClearFormatting
oW.Selection.TypeText Text:="Stück: " & Me.StückA & " Wert: " & Me.LagerSummeA

oW.Selection.GoTo What:=wdGoToBookmark, Name:="Bis1000"
oW.Selection.Find.ClearFormatting
oW.Selection.TypeText Text:="Stück: " & Me.StückB & " Wert: " & Me.LagerSummeB

oW.Selection.GoTo What:=wdGoToBookmark, Name:="Über1000"
oW.Selection.Find.ClearFormatting
oW.Selection.TypeText Text:="Stück: " & Me.StückC & " Wert: " & Me.LagerSummeC

'Diagramm an der Textmarke "Diagramm" einfügen
oW.Selection.GoTo What:=wdGoToBookmark, Name:="diagramm"
oW.Selection.Paste
```

Schalten Sie zurück in die Formularansicht.
Testen Sie den Ablauf.

Dokument speichern und schließen

```
oW.ActiveDocument.SaveAs Filename:="Lagerwert_" & Day(Date) & "_" & Month(Date) & "_" & Year(Date)
& ".doc"
oW.ActiveDocument.Close
```

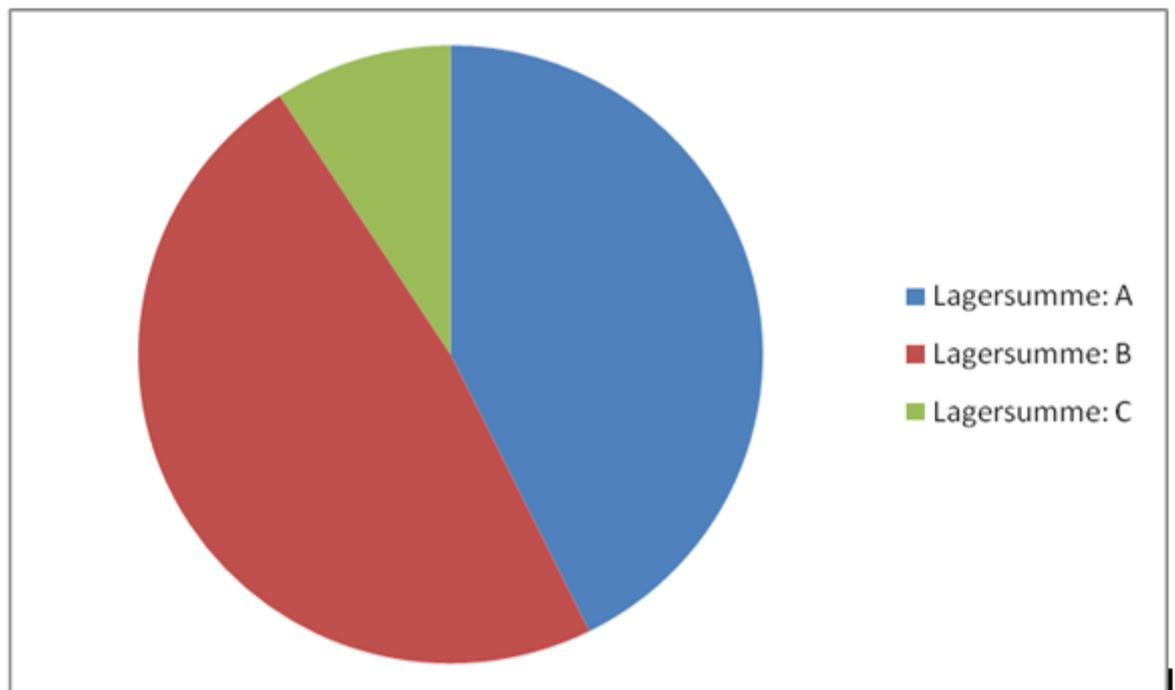
Sehr geehrte Frau Müller,

hier erhalten Sie die aktuellen Zahlen des Lagerwertes:

Lagerwert im Bereich A kleiner 100 Euro: Stück: 308 Wert: 509699,988

Lagerwert im Bereich B ab 100 bis 1000 Euro: Stück: 1942 Wert: 575194,908

Lagerwert im Bereich C größer 1000 Euro: Stück: 2687 Wert: 110286,532



Mit freundlichen Grüßen

Walter Eder

ABC-Analyse	27	If ohne End If.....	19
Abfrage ... 3, 4, 5, 11, 15, 25, 26, 27, 28, 29, 30		Konstanten	24
Abstand vermindern	2	Konstantentabelle	24, 26
ADO	32	Makro aufzeichnen.....	40
Aktuelle Rechnungsnummer anzeigen.....	15	Makro Aufzeichnen	35
am Schmalsten	1, 23	Modul	24, 26, 38
Ändern unterbinden	22	MwstHolen	25
Anwendungserstellung	1	Mwst-Summe	12
Anzeigefehler	2	Neue Felder erstellen.....	1
Aufzeichnung	37, 41	Objekt Excel.....	34
Aufzeichnung beenden.....	37	Objekte ausrichten	1
Befehlsschaltfläche.....	16, 33	Programcode aus Word in das Formular übertragen	42
Beim Anzeigen	22	QBE.....	29
Beim Klicken.....	23, 34	Query by Example	29
Berechnete Felder	3	Range	34
Bericht erstellen.....	7	Rechnungsbetrag.....	6
Berichte.....	7	Schaltfläche zum Öffnen der Rechnung	16
Bibliothek	31	schließen.....	42
column	3	SELECT	29
Count	29	Skontobetrag.....	26
DAO	32	speichern	42
Daten nach Excel	33	Steuerelementinhalt.....	6
Datenanbindung	31	SteuerelementTip-Text	21
Datenbeschriftung	36	Sum.....	28, 29
Datum	18, 25	SummeNetto	11
Davg	32	Tagesdatum	18
Dcount	32	Text verschieben.....	1
Diagramm	35	Textfeld	3, 4, 6, 10
Dlookup	32	Textmarke	39, 41
Domänenfunktionen	32	Umschalttaste + F2.....	5
Dsum	32	Unterformular	2, 4, 5, 22
Ereignis.....	18, 21, 22, 23, 32, 34	Vertikaler Abstand.....	2
Excel	33	Verweis	35
F8.....	25	Verweise	31
Felder benennen	2	Visible	34
Format-Funktion	26	Vorlage speichern	40
Formular 1, 3, 4, 5, 6, 15, 16, 18, 19, 20, 21, 22, 23, 30, 31, 34, 38		With.....	37
Formularfuß	5	Word.....	38
Größe angleichen	23	Word aus Access steuern	38
Größe der Textfelder	1	Wordvorlage.....	39
Gruppierung.....	8	Zoomfenster.....	5