

---

Birgit Swoboda, Sabine Buhlert

1. Ausgabe, Februar 2020

ISBN 978-3-86249-867-3

**Access 2019**  
**Fortgeschrittene Techniken**  
**für Datenbankentwickler**

ACC2019DF



**HERDT**

<b>Bevor Sie beginnen ...</b>	<b>4</b>	4.5 Besonderheiten bei Listen- und Kombinationsfeldern	64
		4.6 Übung	66
<b>Daten auswerten</b>			
<b>1 Abfragen mit Funktionen</b>	<b>6</b>	<b>5 Erweiterte Formulartechnik</b>	<b>67</b>
1.1 Basiswissen Formeln und Funktionen	6	5.1 Formulare optisch aufwerten	67
1.2 Mit Funktionen arbeiten	9	5.2 Mit Optionsgruppen arbeiten	70
1.3 Datums- und Uhrzeitfunktionen	11	5.3 Mit Unterformularen arbeiten	72
1.4 Text-Datenfelder auswerten	14	5.4 Mit Registersteuerelementen arbeiten	75
1.5 Umwandlungsfunktionen	16	5.5 Mit Webbrowsersteuerelementen arbeiten	77
1.6 Einsatz (finanz-)mathematischer Funktionen	17	5.6 Übung	79
1.7 Besonderheiten bei NULL-Werten und leeren Zeichenfolgen	18	<b>6 Berichte in der Entwurfsansicht bearbeiten</b>	<b>81</b>
1.8 Übungen	21	6.1 Die Entwurfsansicht	81
<b>2 Abfragen mit erweiterten Funktionen</b>	<b>23</b>	6.2 Spezielle Eigenschaften von Berichten	84
2.1 Bedingte Berechnungen mit der Wenn-Funktion	23	6.3 Gruppierungen in Berichten	86
2.2 Aggregatfunktionen	27	6.4 Berechnungen und Auswertungen	90
2.3 Kreuztabellenabfragen erstellen	30	6.5 Unterberichte erstellen	93
2.4 Übungen	34	6.6 Mit Etiketten und Vordrucken arbeiten	94
		6.7 Übung	97
<b>Formulare und Berichte entwerfen</b>			
<b>3 Formulare in der Entwurfsansicht erstellen</b>	<b>36</b>	<b>Makros</b>	
3.1 Die Entwurfsansicht von Formularen	36	<b>7 Makros erstellen und bearbeiten</b>	<b>99</b>
3.2 Basiswissen zu Steuerelementen	38	7.1 Basiswissen zu Makros	99
3.3 Steuerelemente in Formulare einfügen	42	7.2 Makros erstellen	99
3.4 Steuerelemente markieren und entfernen	46	7.3 Argumente für Makrofunktionen	102
3.5 Größe und Position von Steuerelementen ändern	48	7.4 Makros speichern und ausführen	103
3.6 Das Aussehen von Steuerelementen ändern	50	7.5 Aktionen bearbeiten	104
3.7 Bedingte Formatierung für Steuerelemente	54	7.6 Fehlersuche in Makros durchführen	105
3.8 Übungen	55	7.7 Kurzerläuterung für wichtige Aktionen	107
		7.8 Übung	110
<b>4 Eigenschaften von Formularen und Steuerelementen</b>	<b>57</b>	<b>8 Erweiterte Makrotechnik</b>	<b>111</b>
4.1 Basiswissen zu Eigenschaften	57	8.1 Makros durch Ereignisse ausführen	111
4.2 Formulareigenschaften	58	8.2 Makros in Formulare und Berichte einbinden	112
4.3 Besondere Eigenschaften von Steuerelementen	61	8.3 Schaltflächen mit Makros verbinden	114
4.4 Besonderheiten bei Options- bzw. Kontrollfeldern und Umschaltflächen	62	8.4 Makros bedingt ausführen	115
		8.5 Makroblöcke bearbeiten	118
		8.6 <i>AutoKeys</i> - und <i>AutoExec</i> -Makros	118
		8.7 Datenmakros	120
		8.8 Übungen	124

## Import, Export und spezielle Datentypen

<b>9 Kommunikation mit anderen Anwendungen</b>	<b>126</b>
9.1 Basiswissen Datenaustausch	126
9.2 Access-Objekte importieren oder verknüpfen	127
9.3 Fremddaten importieren oder verknüpfen	129
9.4 Access-Daten exportieren	131
9.5 Access-Daten in eine HTML-Datei exportieren	132
9.6 Access-Daten im XML-Format exportieren	133
9.7 Gespeicherte Importe und Exporte	134
9.8 Übungen	135

<b>10 Datentypen <i>Link</i>, <i>OLE-Objekt</i> und <i>Anlage</i></b>	<b>137</b>
10.1 Links in Tabellen verwenden	137
10.2 Mit OLE-Objekten arbeiten	139
10.3 OLE-Objekte hinzufügen	141
10.4 Mit Anlagen arbeiten	142
10.5 Übungen	144

## Datenbanken verwalten und einstellen

<b>11 Datenbankverwaltung</b>	<b>146</b>
11.1 Datenbanken komprimieren und reparieren	146
11.2 Leistungsanalyse für eine Datenbank durchführen	148
11.3 Den Datenbankdokumentierer verwenden	149
11.4 Datenbankdatei konvertieren	152
11.5 Übung	153

<b>12 Einstellungen für Datenbanken</b>	<b>154</b>
12.1 Ansicht des Navigationsbereichs personalisieren	154
12.2 Optionen für die Datenbank festlegen	157
12.3 Datenbanken schützen	158
12.4 Datenbank als ACCDE-Datei weitergeben	160
12.5 Übung	161

<b>Stichwortverzeichnis</b>	<b>162</b>
-----------------------------	------------

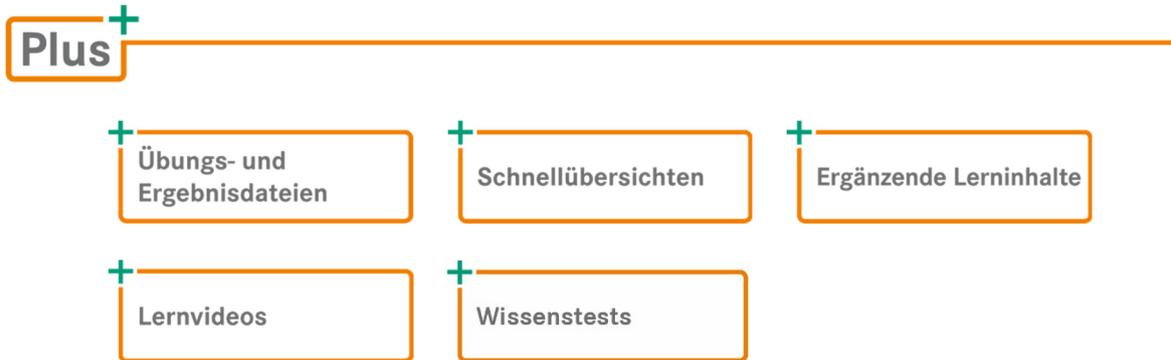
# Bevor Sie beginnen ...

## HERDT BuchPlus – unser Konzept:

Problemlos einsteigen – Effizient lernen – Zielgerichtet nachschlagen

(weitere Infos unter [www.herdt.com/BuchPlus](http://www.herdt.com/BuchPlus))

Nutzen Sie dabei unsere maßgeschneiderten, im Internet frei verfügbaren Medien:



So können Sie schnell auf die BuchPlus-Medien zugreifen:

- ▶ Rufen Sie im Browser die Internetadresse [www.herdt.com](http://www.herdt.com) auf.



Für einen optimalen Lernerfolg verfügen Sie über folgende Kompetenzen:

- ✓ Tabellen, Formulare und Standardberichte erstellen und bearbeiten,
- ✓ mit Filtern und Abfragen arbeiten,
- ✓ Datenbankobjekte bearbeiten.

Die Funktionsbeschreibungen des Buchs basieren auf:

- ✓ Windows 10 (64-Bit-Version) mit einem **lokalen** angemeldeten Microsoft-Benutzerkonto,
- ✓ einer Erstinstallation von Access 2019 (64-Bit-Version) mit den empfohlenen Einstellungen,
- ✓ einem Access-Update mit dem Stand von Oktober 2019,
- ✓ einer ständig verfügbaren Verbindung zum Internet.

**!** Je nach gewählter Bildschirmauflösung kann das Aussehen des Menübands und des Access-Fensters von den Abbildungen im Buch abweichen.

Durch Update-Funktionen des Herstellers Microsoft können sich Menübefehle und Aussehen der Oberfläche zu einem späteren Zeitpunkt von den im Buch verwendeten Abbildungen unterscheiden.

Für die Erstellung des Buchs wurden die standardmäßig bestehenden Sicherheitseinstellungen für Makros ausgeschaltet (Register *Datei - Optionen*, Kategorie *Trust Center*, Schaltfläche *Einstellungen für das Trust Center*, Kategorie *Makroeinstellungen*, Optionsfeld *Alle Makros aktivieren*), um die entsprechenden Sicherheitswarnungen zu vermeiden. In der Praxis sollten die Standardeinstellungen beibehalten werden, um die Maßnahmen zum Schutz vor Viren nicht einzuschränken.

## 1

# Abfragen mit Funktionen



Übungsdatei: *Buero01.accdb*

Ergebnisdatei: *Buero01-E.accdb*

Ergänzende Lerninhalte: *Datenbanken-Theorie.pdf*

## 1.1 Basiswissen Formeln und Funktionen

### Datenauswertung in Abfragen

Von den Daten einer Datenbank werden häufig Zusammenstellungen oder Auswertungen selektiver Daten benötigt. Dazu erzeugen Sie Abfragen. Beim Erstellen einer Abfrage werden unter anderem:

- ✓ Datenfelder ausgewählt, die für den aktuellen Zweck benötigt werden. Dieser Vorgang wird als **Projektion** (Auswahl von Spalten) bezeichnet.
- ✓ nur diejenigen Datensätze angezeigt, die bestimmten Kriterien entsprechen. Dieser Vorgang wird als **Selektion** (Auswahl von Zeilen) bezeichnet.
- ✓ Berechnungen zur Datenauswertung durchgeführt, z. B. mit Formeln und Funktionen.

### Formeln und Funktionen in Abfragen

Um Daten auszuwerten, setzen Sie in Abfragen Formeln und Funktionen ein. Access bietet eine Vielzahl von Funktionen an. Sie haben auch die Möglichkeit, eigene, benutzerdefinierte Funktionen zu erstellen.

Funktionen können in einer Abfrage in den Zeilen *Feld*, *Funktion*, *Kriterien* und *oder* definiert werden.

Die Zeile *Funktion* ① können Sie im Register *Entwurf* (*Abfragetools*) in der Gruppe *Einblenden/Ausblenden* über *Summen* einblenden.

Berechnetes Feld, neuer Feldname <i>Umsatz</i>			
②	Feld:	ArtikelID	Umsatz: Summe([Menge]*[Einzelpreis])
	Tabelle:	Artikel	
①	Funktion:	Gruppierung	Formel
	Sortierung:		
	Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
③	Kriterien:		
	oder:		

### Mögliche Berechnungen

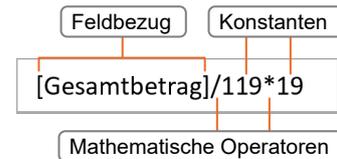
<b>Berechnete Felder</b>	In der Zeile <i>Feld</i> ② können Sie berechnete Felder definieren. Berechnete Felder sind neue Felder mit Rechenergebnissen, die sich aus Berechnungen mit bestehenden Feldern ergeben. Sie können sowohl arithmetische Operationen durchführen als auch Access-Funktionen einsetzen.
<b>Aggregatfunktionen</b>	Aggregatfunktionen beziehen sich auf mehrere Datensätze, entweder auf das gesamte Abfrageergebnis oder auf das Ergebnis einer Gruppierung. In der Zeile <i>Funktion</i> ① können Sie Aggregatfunktionen auswählen.
<b>Kriterien/ Bedingungen</b>	Um aus allen Datensätzen bestimmte Datensätze zu selektieren, können Bedingungen in den Zeilen <i>Kriterien</i> ③ bzw. <i>oder</i> formuliert werden.

### Aufbau von Formeln (Ausdrücke)

Formeln für berechnete Felder oder Kriterien werden in Access als Ausdrücke bezeichnet. Ein Ausdruck besteht in der Regel aus der Kombination von mindestens einem Operator und wenigstens einem weiteren Element, wie z. B. Konstanten, Funktionen, Feldbezügen oder Eigenschaften. Das Ergebnis eines Ausdrucks ist ein Wert.

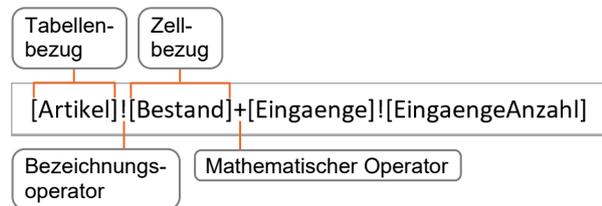
#### Einfache Formel mit Konstanten

Die abgebildete Formel ermittelt die Mehrwertsteuer eines Brutto-Rechnungsbetrages.



#### Formel mit Werten aus verschiedenen Tabellen

Diese Formel addiert zwei Werte aus unterschiedlichen Tabellen.



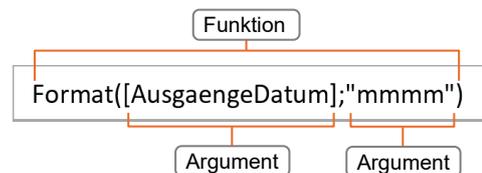
#### Berechnung mit Eigenschaftswerten

Hier wird auf einen Eigenschaftswert eines Formulars Bezug genommen.



#### Formel mit Funktion

Die Formel zeigt mit der Funktion `Format` den Monat aus einem Datum an. Der Feldbezug ist in diesem Beispiel das Datenfeld `[AusgaengeDatum]`.



## Elemente von Ausdrücken

Ausdrücke können folgende Elemente enthalten:

<b>Konstante Werte</b>	Werte, die sich innerhalb einer Formel nicht ändern, werden als konstant bezeichnet. Es wird zwischen Zahlen (z. B. 500) und Text (z. B. "kein Eintrag", "Heidelberg") unterschieden.
<b>Feldbezüge (auch Bezeichner genannt)</b>	Um mit Inhalten von Datenfeldern rechnen zu können, müssen Sie den Feldbezug angeben. Dazu wird das Datenfeld in eckige Klammern geschrieben, beispielsweise [Bestelldatum].
<b>Funktionen</b>	Mithilfe von Funktionen lassen sich viele Standardberechnungen einfach und schnell ausführen, wie Summe, Mittelwert, Anzahl, Zeitwerte.
<b>Bezeichnungsoperatoren</b>	<ul style="list-style-type: none"> <li>✓ <b>! (Ausrufezeichen)</b> Das Ausrufezeichen wird zur Bestimmung von Datenbankobjekten verwendet. Es trennt die verschiedenen Bezeichner, beispielsweise Tabellename und Datenfeldname. Im Beispiel wird das Datenfeld <i>Ausgangsdatum</i> der Tabelle <i>AusgangVorjahr</i> verwendet. [AusgangVorjahr] ! [Ausgangsdatum].</li> <li>✓ <b>. (Punkt)</b> Der Punkt-Operator trennt die Objektbezeichnung eines Objektes von der Objekteigenschaft, im Beispiel die Eigenschaft <i>Standardwert</i> des Feldes <i>Mindestbestellmenge</i>. [Mindestbestellmenge] .Standardwert</li> </ul>
<b>Mathematische Operatoren</b>	Mit mathematischen Operatoren wie: + - * / können Sie unterschiedliche Berechnungen durchführen. Es gilt die Punkt-vor-Strich-Regelung.
<b>Runde Klammern</b>	Mithilfe von Klammern kann die Reihenfolge der Berechnung verändert werden.

## Wie sind Funktionen aufgebaut?

Alle Funktionen haben eine genau festgelegte Schreibweise (Syntax), die exakt eingehalten werden muss.

<b>Syntax</b>	Funktionsname (Argument1;Argument2;...)
<b>Funktionsname</b>	Die Art der Berechnung bzw. Auswertung wird durch den Funktionsnamen festgelegt. Access stellt eine Vielzahl verschiedener Funktionen zur Verfügung, beispielsweise: Datum(), Summe(Argumente), Min(Argumente), Anzahl(Argumente).
<b>Argumente</b>	<p>Funktionen brauchen zum Berechnen oft weitere Informationen, die als Argumente bezeichnet werden. Argumente in eckigen Klammern [ ] sind sogenannte <b>optionale</b> Argumente. Sie können bei Bedarf angegeben werden. Die Funktion Wochentag(#Datumsangabe#; [ErsterWochentag]) liefert aus einem Datum den Wochentag (Montag, Dienstag...); als Argument benötigt die Funktion ein Datum. Das Argument <i>ErsterWochentag</i> ist optional. Wird es nicht angegeben, wird automatisch Sonntag als erster Wochentag verwendet.</p> <p>Funktionen <b>ohne</b> Argumente bestehen nur aus dem Funktionsnamen und dem runden Klammernpaar, z. B. Datum().</p>

## 1.2 Mit Funktionen arbeiten

### Basiswissen zum Arbeiten mit Funktionen

- ✓ Die Argumente werden in runden Klammern direkt im Anschluss an den Funktionsnamen eingetragen.
- ✓ Anzahl, Reihenfolge und Art der benötigten Argumente sind für jede Funktion durch die Syntax festgelegt.
- ✓ Die einzelnen Argumente werden durch ein Semikolon voneinander getrennt.
- ✓ Funktionen, die keine Argumente besitzen, benötigen trotzdem im Anschluss an den Funktionsnamen die beiden runden Klammern, beispielsweise die Funktion `Datum()`, die das aktuelle Datum einfügt.
- ✓ Funktionen werden innerhalb einer Formel an die Stelle gesetzt, an der das Ergebnis der Funktion bei der Auswertung stehen soll. Falls erforderlich, kann ein neues Datenfeld in der Abfrage erzeugt werden.
- ✓ Funktionen lassen sich in Formeln einbinden und ineinander verschachteln. Wichtig ist, dass die Datentypen (z. B. Text, Zahl) immer zu den benötigten Argumenten passen.

Als Argumente sind beispielsweise folgende Typen möglich:

Argumenttypen	Erläuterung	Beispiele
Zahlen	Zahlen werden als konstante Werte eingegeben. Sie können nachträglich nur direkt in der Formel verändert werden.	7013, 3211
Datenfeldname	Datenfeldnamen weisen auf alle Datenfeldinhalte hin. Gleiche Datenfeldnamen, bei Verwendung mehrerer Tabellen, müssen durch den Tabellennamen unterschieden werden.	[Bestand] [Artikel]! [ArtikelID]
Formeln	Das Ergebnis einer Formel kann als Argument eingesetzt werden.	( [Bestand] +50) /2
Funktionen	Das Argument kann aus einem Funktionsergebnis resultieren.	Monat (Datum())
Text	Texte werden in Anführungszeichen als konstante Werte eingegeben.	"Umsatz" "kein Eintrag"
Wahrheitswerte	Wahrheitswerte geben an, ob ein Ergebnis wahr oder falsch ist.	WAHR      FALSCH
Datums- bzw. Uhrzeitangaben	Möchten Sie ein Datum oder eine Uhrzeit als Argument einer Funktion angeben, schreiben Sie Datum und Uhrzeit in #-Zeichen eingeschlossen.	Monat (#01.12.20#)

## Funktionen mithilfe des Ausdrucks-Generators in Abfragen einfügen

Der Ausdrucks-Generator kann in der Regel von jedem Feld aus geöffnet werden, in dem Ausdrücke formuliert werden können.

- ▶ Setzen Sie den Cursor in die Zelle des Entwurfsbereichs, in der die Formel stehen soll.
- ▶ Klicken Sie im Register *Entwurf*, Gruppe *Abfragesetup*, auf *Generator*. 
- oder Rufen Sie den Kontextmenüpunkt *Aufbauen* auf.

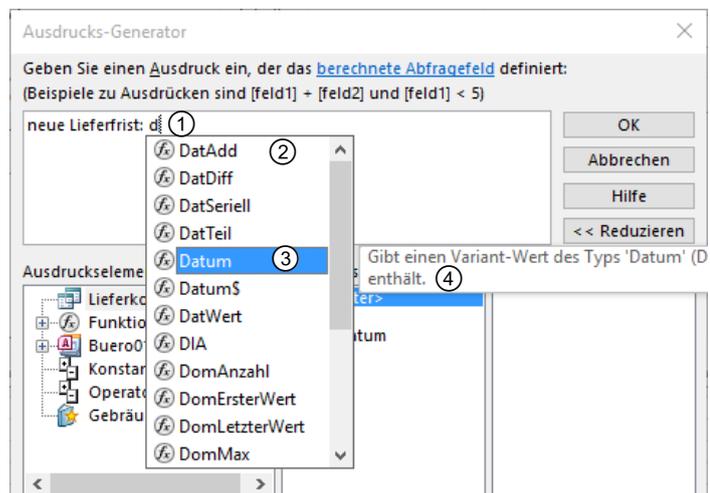
Access öffnet das Dialogfenster *Ausdrucks-Generator*.

- ▶ Öffnen Sie für dieses Beispiel die Abfrage *Lieferkontrolle* der Übungsdatei *Buero01* in der Entwurfsansicht.

Ermitteln Sie die Lieferfrist für alle Datensätze:

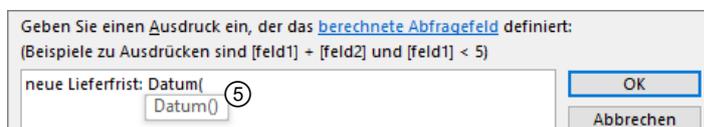
Starten Sie dazu den Ausdrucks-Generator in einem neuen Feld und geben Sie den neuen Feldnamen ein, in diesem Beispiel: *neue Lieferfrist*:. Nach dem Doppelpunkt geben Sie einem Leerschritt und den Buchstaben *d* ① ein.

Bereits nach der Eingabe des ersten Buchstabens werden Sie von IntelliSense unterstützt. Die eingeblendete IntelliSense-Liste ② liefert Ihnen, passend zur begonnenen Eingabe, Vorschläge, im Beispiel Funktionen. Sie können die Liste durch weitere Eingaben reduzieren oder einen Vorschlag markieren. Zum markierten Eintrag ③ wird eine Beschreibung (QuickInfo) ④ eingeblendet.



- ▶ Drücken Sie  (oder klicken Sie doppelt), um den markierten Eintrag ins Ausdrucksfeld zu übernehmen.

Um die weitere Eingabe zu erleichtern, wird die Syntax der Funktion eingeblendet ⑤.



- ▶ Geben Sie die entsprechenden Werte ein.



## Weitere Einsatzmöglichkeiten von Funktionen in Tabellen

Sie können in der Tabellenentwurfsansicht den Datenfeldeigenschaften *Standardwert* und *Gültigkeitsregel* ebenfalls Formeln und Funktionen zuweisen. Der Ausdrucks-Generator zeigt eine Auswahl der Funktionen an, die für diese speziellen Eigenschaften zur Verfügung stehen.

Im Beispiel wird ein Tabellenfeld vom Datentyp *Datum Uhrzeit* mit dem aktuellen Datum vorbelegt.

- ▶ Klicken Sie in das Feld des Eigenschaftsfeldes *Standardwert*.
- ▶ Klicken Sie auf  ① und fügen Sie mit dem Ausdrucks-Generator die gewünschte Funktion ein, im Beispiel die Funktion `Datum()`, die als Standardwert das aktuelle Datum anzeigt.

Allgemein	Nachschlagen
Format	tt\mm\jjjj
Eingabeformat	
Beschriftung	
Standardwert	=Datum() ① 
Gültigkeitsregel	

## 1.3 Datums- und Uhrzeitfunktionen

### Datums- und Uhrzeitwerte verwenden

Datums- und Uhrzeitwerte werden häufig für Anwendungen benötigt, beispielsweise für Projektplanungen oder für das Rechnungswesen. Zur Ermittlung des Datums und seiner Bestandteile existiert eine Vielzahl von Funktionen, mit denen sich Auswertungen vornehmen lassen.

Die Funktion `Datum()` gibt das aktuelle Datum aus. Dieses wird als Wert gespeichert.

Beispiel: `Ausleihfrist = Ausleihdatum + 30 Tage`.

Die Überziehungszeit können Sie mit der Funktion `Datum()` automatisieren, die Leihfristüberschreitung berechnet sich dann aus `Datum() - Ausleihfrist`.

### Interpretation von Datumsangaben mit zweistelliger Jahreszahl

Bei der abgekürzten Schreibweise der Jahreszahl gilt folgende Regel:

01.01.**00** bis 31.12.**29** ⇒ 01.01.**2000** bis 31.12.**2029**

01.01.**30** bis 31.12.**99** ⇒ 01.01.**1930** bis 31.12.**1999**

### Auswahl einfacher Datums- und Zeitfunktionen

Datumsberechnungen können mit den arithmetischen Operatoren (+ - \* /) durchgeführt werden. 1 bedeutet ein Zeitintervall von einem Tag und 0,5 dementsprechend von 12 Stunden. Die folgende Tabelle zeigt Funktionen zur Berechnung mit Datum und Uhrzeit.

Funktion	Beschreibung	Beispiel: 15.04.20 16:09:01
Jetzt ()	Gibt das aktuelle Datum und die aktuelle Zeit des Rechners aus	15.04.20 16:09:01
Datum ()	Gibt das aktuelle Datum des Rechners aus	15.04.20
Zeit ()	Gibt die aktuelle Uhrzeit des Rechners aus	16:09:01
Jahr (#Datumsangabe#)	Ermittelt aus einem Datum das Jahr	2020
Monat (#Datumsangabe#)	Ermittelt aus einem Datum den Monat	4
Tag (#Datumsangabe#)	Ermittelt aus einem Datum den Tag	15
Wochentag (#Datumsangabe#; [ErsterWochentag]; 2)	Ermittelt den Wochentag (Das 2. Argument gibt an, welcher der erste Wochentag ist: im Beispiel 2 = Montag. Bei fehlender Angabe ist Sonntag der erste Wochentag.)	3 (Mittwoch)

Weitere Funktionen zur Ermittlung bzw. Berechnung von Zeiteinheiten sind die Funktionen `Stunde()`, `Minute()`, `Sekunde()`.

Mit den folgenden Funktionen können Sie einzelne Datumselemente herausfiltern und mit diesen weitere Berechnungen durchführen. Maßgebend ist bei diesen Funktionen das Argument `Intervall`:

Funktion	Beschreibung	Beispiel: 15.04.2020 14:09:01
<code>DatTeil (Intervall; #Datumsangabe#)</code> Über die zusätzlichen Argumente <code>ErsterWochentag</code> und <code>ErsteWocheDesJahres</code> kann gesteuert werden, welcher Wochentag als der erste gilt (1 = Sonntag, 2 = Montag) und welche Woche (1 = ab erstem Januar, 2 = erste Woche mit vier Arbeitstagen, 3 = erste Woche mit sieben Arbeitstagen) als die erste des Jahres zählt.	Gibt einen Teil des Datums aus  Intervall = "jjjj" Jahr Intervall = "q" Quartal Intervall = "m" Monat Intervall = "j" Kalendertag Intervall = "t" Tag Intervall = "w" Wochentag Intervall = "ww" Kalenderwoche Intervall = "h" Stunde Intervall = "n" Minute Intervall = "s" Sekunde	<code>DatTeil ("ww"; #15.04.2020 14:09:01#)</code>  Ergebnis: 2020 Ergebnis: 2 Ergebnis: 4 Ergebnis: 106 Ergebnis: 15 Ergebnis: 4 Ergebnis: 16  Ergebnis: 14 Ergebnis: 9 Ergebnis: 1
<code>DatAdd (Intervall; Zahl; #Datumsangabe#)</code>	Gibt ein Datum aus, zu dem ein Wert hinzuaddiert wurde	<code>DatAdd ("t"; +3; #15.04.2020 14:09:01#)</code>  Ergebnis: 18.04.2020 14:09:01
<code>DatDiff (Intervall; #Datumsangabe1#; #Datumsangabe2#; [ErsterWochentag]; [ErsteWocheDesJahres])</code>	Gibt die Differenz zwischen zwei Datumswerten, je nach gesetztem Intervall, aus.	<code>DatDiff ("ww"; #15.04.2020 #; #14.08.2020#)</code>  Ergebnis: 17 (Wochen)



### Anzahl der Tage ermitteln

Für jeden Kunden soll ermittelt werden, wie viele Tage der letzte Verkauf zurückliegt.

- ▶ Öffnen Sie die Abfrage *letzterUmsatz* in der Entwurfsansicht.
- ▶ Geben Sie als berechnetes Feld der Abfrage folgenden Ausdruck ein:  
`Liegtzurueck: DatDiff("t"; [Bestelldatum]; Datum())`  
 Um das Ergebnis in der Form *x Tage* anzuzeigen, können Sie im Eigenschaftentabell in der Zeile *Format* den Eintrag *0 "Tage"* eingeben.
- ▶ Möchten Sie pro Kunde nur die Differenz zum letzten Bestelldatum ermitteln, erweitern Sie die bestehende Funktion mit der Funktion `Min()`:  
`Liegtzurueck: Min(DatDiff("t"; [Bestelldatum]; Datum()))`  
 Die Einstellung in der Zeile *Funktion* müssen von *Gruppierung* auf *Ausdruck* geändert werden.

## 1.4 Text-Datenfelder auswerten

### Textfelder zusammenfassen

Textfelder lassen sich auf verschiedene Arten auswerten. Sie können beispielsweise mithilfe des `&`-Operators mehrere Textfelder zu einem Ausgabefeld zusammenfassen.

Beispiel: Sie möchten die Adressangaben der Tabelle *Vertrieb* in einem einzigen Datenfeld der Abfrage *VertriebKontakt* zusammenfassen:

- ▶ Erstellen Sie eine neue Abfrage *VertriebKontakt*, basierend auf den Tabellen *Vertrieb* und *Mitarbeiter*.
- ▶ Geben Sie den Ausdruck für das berechnete Feld *Kontakt* folgendermaßen ein:  
`Kontakt: [Nachname] &" , "& [Vorname] &" , "& [Straße] &" , "& [PLZ] &" , "& [Wohnort]`

VertriebID	Kontakt
1	Dannemann, Georg, Domgasse 45, 65193, Wiesbaden
2	Andel, Linus, Breite Str. 4, 51183, Mainz
3	Berger, Sybille, Hauptstr. 7, 55174, Mainz
4	Hellmeister, Joachim, Berliner Str. 4, 51174, Leverkusen
5	Braun, Helge, Augustusplatz 64, 55139, Mainz
6	Kaufmann, Sonja, Leipziger Str. 45, 51132, Leverkusen

Sie können zusammengesetzte Datenfelder auch direkt in einem Feld vom Datentyp *Berechnet* in der Tabelle anzeigen. Geben Sie dazu im Ausdrucks-Generator für die Feldeigenschaft *Ausdruck* den entsprechenden Ausdruck ein.

## Beispiele für Funktionen mit Zeichenfolgen

Funktion	Beschreibung	Beispiel ="70ABC-8844"
Links (Zeichenfolge; Länge)	Liefert die angegebene Anzahl Zeichen der übergebenen Zeichenfolge von links	Links ([Beispiel]; 5) Ergebnis: 70ABC
Rechts (Zeichenfolge; Länge)	Gibt die angegebene Anzahl Zeichen der übergebenen Zeichenfolge von rechts zurück	Rechts ([Beispiel]; 4) Ergebnis: 8844
Teil (Zeichenfolge; Startwert; [Länge])	Eine Teilzeichenkette aus einer Zeichenfolge wird ermittelt. Die Position des ersten Zeichens (Startwert) muss angegeben werden. Die Anzahl der auszugebenden Zeichen ist optional.	Teil ([Beispiel]; 3; 3) Ergebnis: ABC
Länge (Zeichenfolge)	Ermittelt die Länge der Zeichenfolge	Länge ([Beispiel]) Ergebnis: 10
Ersetzen (Zeichenfolge; Suchen; ErsetzenDurch, [Startwert]; [Anzahl]; [Zeichenvergleich])	Liefert eine Zeichenfolge, in der die Suchzeichenfolge durch die Ersatzzeichenfolge ausgetauscht ist	Ersetzen ([Beispiel]; 8844; "INFO") Ergebnis: 70ABC-INFO

## Zeichen aus Text-Datenfeldern herausfiltern

Aus den Tabellen *Vertrieb* und *Mitarbeiter* sollen alle Datensätze des Postleitzahlenbereichs 51 angezeigt werden.

- ▶ Geben Sie im Abfrageentwurf in der Zeile *Kriterien* für das Textfeld *PLZ* folgenden Ausdruck ein:  
`Links ([PLZ]; 2) = "51"`



VertriebID	Nachname	Vorname	PLZ
2	Andel	Linus	51183
4	Hellmeister	Joachim	51174
6	Kaufmann	Sonja	51132

Anstelle der beschriebenen Funktion (Zeichenkettenfunktion) könnten Sie in diesem Fall auch folgendes Kriterium verwenden: `Wie "51*"`