
Birgit Swoboda, Sabine Buhlert

1. Ausgabe, November 2025

ISBN 978-3-98569-246-0

Access 2024

**Datenbankentwicklung –
Fortgeschrittene Techniken**

ACC2024DF



HERDT

Bevor Sie beginnen ...	4	5.3 Mit Unterformularen arbeiten	73
		5.4 Mit Registerstueerelementen arbeiten	76
		5.5 Übung	79
Daten auswerten			
1 Abfragen mit Funktionen	5	6 Berichte in der Entwurfsansicht bearbeiten	81
1.1 Basiswissen Formeln und Funktionen	5	6.1 Die Entwurfsansicht	81
1.2 Mit Funktionen arbeiten	8	6.2 Spezielle Eigenschaften von Berichten	84
1.3 Datums- und Uhrzeitfunktionen	10	6.3 Gruppierungen in Berichten	86
1.4 Text-Datenfelder auswerten	13	6.4 Berechnungen und Auswertungen	91
1.5 Umwandlungsfunktionen	15	6.5 Unterberichte erstellen	94
1.6 Einsatz (finanz-)mathematischer Funktionen	16	6.6 Mit Etiketten und Vordrucken arbeiten	95
1.7 Besonderheiten bei NULL-Werten und leeren Zeichenfolgen	18	6.7 Übung	98
1.8 Übungen	21	Makros	
2 Abfragen mit erweiterten Funktionen	23	7 Makros erstellen und bearbeiten	100
2.1 Bedingte Berechnungen mit der Wenn-Funktion	23	7.1 Basiswissen zu Makros	100
2.2 Aggregatfunktionen	27	7.2 Makros erstellen	100
2.3 Kreuztabellenabfragen erstellen	30	7.3 Argumente für Makrofunktionen	103
2.4 Übungen	34	7.4 Makros speichern und ausführen	104
		7.5 Aktionen bearbeiten	105
		7.6 Fehlersuche in Makros durchführen	106
		7.7 Kurzerläuterung für wichtige Aktionen	108
		7.8 Übung	111
Formulare und Berichte entwerfen			
3 Formulare in der Entwurfsansicht erstellen	36	8 Erweiterte Makrotechnik	112
3.1 Die Entwurfsansicht von Formularen	36	8.1 Makros durch Ereignisse ausführen	112
3.2 Basiswissen zu Steuerelementen	38	8.2 Makros in Formulare und Berichte einbinden	113
3.3 Steuerelemente in Formulare einfügen	42	8.3 Schaltflächen mit Makros verbinden	115
3.4 Steuerelemente markieren und entfernen	46	8.4 Makros bedingt ausführen	116
3.5 Größe und Position von Steuerelementen ändern	48	8.5 Makroblöcke bearbeiten	119
3.6 Das Aussehen von Steuerelementen ändern	50	8.6 <i>AutoKeys</i> - und <i>AutoExec</i> -Makros	119
3.7 Bedingte Formatierung für Steuerelemente	54	8.7 Datenmakros	121
3.8 Übungen	56	8.8 Übungen	125
4 Eigenschaften von Formularen und Steuerelementen	58	Import, Export und spezielle Datentypen	
4.1 Basiswissen zu Eigenschaften	58	9 Kommunikation mit anderen Anwendungen	127
4.2 Formulareigenschaften	59	9.1 Basiswissen Datenaustausch	127
4.3 Besondere Eigenschaften von Steuerelementen	62	9.2 Access-Objekte importieren oder verknüpfen	128
4.4 Besonderheiten bei Options- bzw. Kontrollfeldern und Umschaltflächen	63	9.3 Fremddaten importieren oder verknüpfen	131
4.5 Besonderheiten bei Listen- und Kombinationsfeldern	64	9.4 Access-Daten exportieren	133
4.6 Übung	67	9.5 Access-Daten in eine HTML-Datei exportieren	134
5 Erweiterte Formulartechnik	68	9.6 Access-Daten im XML-Format exportieren	135
5.1 Formulare optisch aufwerten	68	9.7 Gespeicherte Importe und Exporte	136
5.2 Mit Optionsgruppen arbeiten	70	9.8 Übungen	137

10 Datentypen <i>Link</i>, <i>OLE-Objekt</i> und <i>Anlage</i>	139
10.1 Links in Tabellen verwenden	139
10.2 Mit OLE-Objekten arbeiten	141
10.3 OLE-Objekte hinzufügen	143
10.4 Mit Anlagen arbeiten	144
10.5 Übungen	147
Datenbanken verwalten und einstellen	
11 Datenbankverwaltung	149
11.1 Datenbanken komprimieren und reparieren	149
11.2 Leistungsanalyse für eine Datenbank durchführen	151
11.3 Den Datenbankdokumentierer verwenden	152
11.4 Übung	155
12 Einstellungen für Datenbanken	156
12.1 Ansicht des Navigationsbereichs personalisieren	156
12.2 Optionen für die Datenbank festlegen	159
12.3 Datenbanken schützen	161
12.4 Datenbank als ACCDE-Datei weitergeben	163
12.5 Übung	164
Stichwortverzeichnis	165
Nützliche Tastenkombinationen	168

Bevor Sie beginnen ...

HERDT BuchPlus – unser Konzept:

Problemlos einsteigen – Effizient lernen – Zielgerichtet nachschlagen

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



Wie Sie schnell auf diese BuchPlus-Medien zugreifen können, erfahren Sie unter:

www.herdt.com/BuchPlus

Um die Lerninhalte des Buches praktisch nachzuvollziehen, benötigen Sie:

- ✓ Windows 10/11
- ✓ Access 2024
- ✓ gegebenenfalls ein Microsoft- und ein OneDrive-Konto

Für einen optimalen Lernerfolg verfügen Sie über folgende Kompetenzen:

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

Die behandelten Beispiele und Themen setzen den Kenntnisstand des Buches „Access 2024 – Datenbankentwicklung – Grundlagen“ voraus.

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

Für die Erstellung des Buches 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 darüber hinaus die Möglichkeit, eigene, benutzerdefinierte Formeln (auch in Kombination mit 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 *Abfrageentwurf* in der Gruppe *Einblenden/Ausblenden* über *Summen* einblenden bzw. ausblenden.

Zeilen mit möglichen Berechnungen

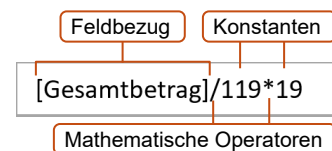
Berechnete Felder	In der Zeile <i>Feld</i> ② können Sie berechnete Felder definieren. Berechnete Felder sind neue Felder mit Rechenergebnissen, die sich aus Berechnungen mit bestehenden Feldern ergeben. Sie können sowohl arithmetische Operationen durchführen als auch Access-Funktionen einsetzen.
Aggregatfunktionen	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.
Kriterien/Bedingungen	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.

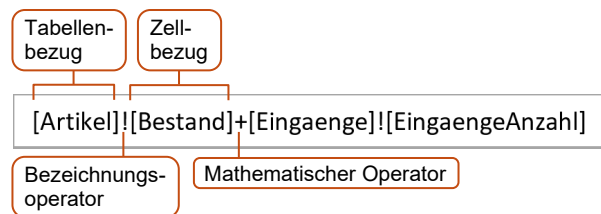
Einfache Formel mit Konstanten

Die abgebildete Formel ermittelt die Mehrwertsteuer eines Brutto-Rechnungsbetrages.



Formel mit Werten aus verschiedenen Tabellen

Diese Formel addiert zwei Werte aus unterschiedlichen Tabellen.



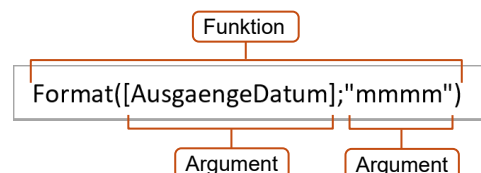
Berechnung mit Eigenschaftswerten

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:

Konstante Werte	Werte, die sich innerhalb einer Formel nicht ändern, werden als konstant bezeichnet. Bei Konstanten wird zwischen Zahlen (z. B. 500) und Text (z. B. "kein Eintrag", "Heidelberg") unterschieden. Text wird in Anführungszeichen gesetzt.
Feldbezüge (auch Bezeichner genannt)	Um mit Inhalten von Datenfeldern rechnen zu können, müssen Sie den Feldbezug angeben. Dazu wird das Datenfeld in eckige Klammern geschrieben, beispielsweise [Bestelldatum].
Funktionen	Mithilfe von Funktionen lassen sich viele Standardberechnungen einfach und schnell ausführen, wie Summe, Mittelwert, Anzahl, Zeitwerte.
Bezeichnungsoperatoren	<ul style="list-style-type: none"> ✓ ! (Ausrufezeichen) 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]. ✓ . (Punkt) 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
Mathematische Operatoren	Mit mathematischen Operatoren wie: + - * / können Sie unterschiedliche Berechnungen durchführen. Es gilt die Punkt-vor-Strich-Regelung.
Runde Klammern	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.

Syntax	Funktionsname (Argument1;Argument2;...)
Funktionsname	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).
Argumente	<p>Funktionen brauchen zum Berechnen oft weitere Informationen, die als Argumente bezeichnet werden. Argumente in eckigen Klammern [] sind sogenannte optionale Argumente. Sie können bei Bedarf angegeben werden.</p> <p>Die Funktion <code>Wochentag (#Datumsangabe#; [ErsterWochentag])</code> liefert aus einem Datum den Wochentag (Montag, Dienstag ...); als Argument benötigt die Funktion eine Datumsangabe. Das Argument <code>ErsterWochentag</code> ist optional. Wird es nicht angegeben, wird automatisch Sonntag als erster Wochentag verwendet.</p> <p>Funktionen ohne Argumente bestehen nur aus dem Funktionsnamen und dem runden Klammernpaar, z. B. <code>Datum()</code>.</p>

1.2 Mit Funktionen arbeiten

Basiswissen zum Arbeiten mit Funktionen

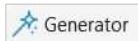
- ✓ Die Argumente werden in runden Klammern direkt im Anschluss an den Funktionsnamen eingetragen.
- ✓ Anzahl, Reihenfolge und Art der benötigten Argumente sind für jede Funktion durch die Syntax festgelegt.
- ✓ Die einzelnen Argumente werden durch ein Semikolon voneinander getrennt.
- ✓ Funktionen, 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 in der Formel verändert werden.	7013,3211
Datenfeldname	Datenfeldnamen weisen auf 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 der zu überprüfende Wert zutrifft oder nicht zutrifft, also 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 *Abfrageentwurf*, 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.

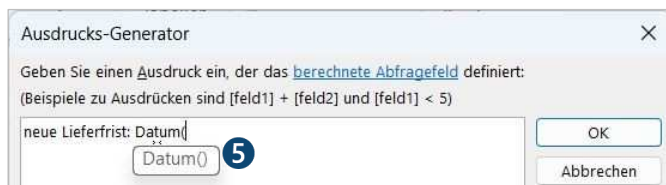
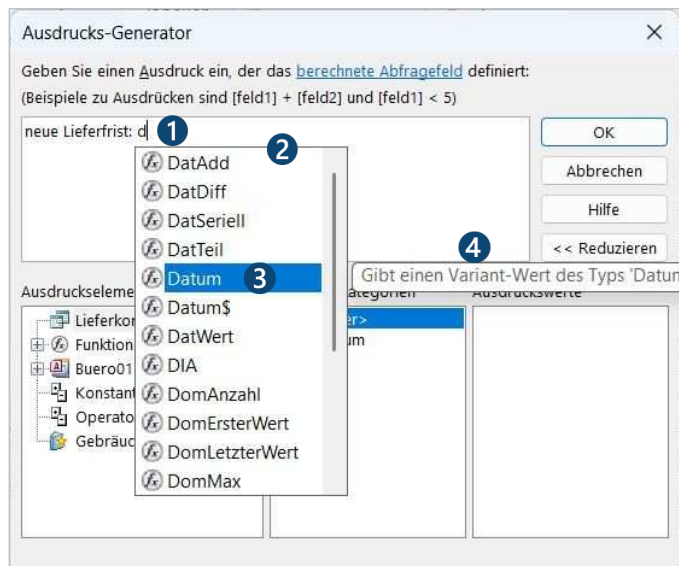
Ermitteln Sie die Lieferfrist für alle Datensätze:

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*  ein.

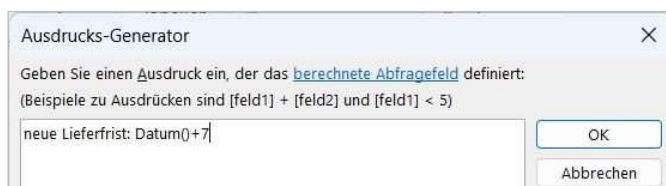
Bereits nach der Eingabe des ersten Buchstabens werden Sie von IntelliSense unterstützt. Die eingblendete IntelliSense-Liste  liefert Ihnen, passend zur begonnenen Eingabe, Vorschläge, im Beispiel Funktionen. Sie können die Liste durch weitere Eingaben reduzieren oder einen Vorschlag markieren. Zum markierten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

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

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




- ▶ Geben Sie die entsprechende Formel 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 die Schaltfläche mit den drei Punkten  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	Datum, kurz
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: $\text{Ausleihfrist} = \text{Ausleihdatum} + 30 \text{ Tage}$.

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

Interpretation von Datumsangaben mit zweistelliger Jahreszahl

Bei der zweistelligen Schreibweise der Jahreszahl gilt folgende Regel:

01.01.**00** bis 31.12.**49** ⇒ 01.01.**2000** bis 31.12.**2049**

01.01.**50** bis 31.12.**99** ⇒ 01.01.**1950** 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: 15.04.24 16:09:01
Jetzt ()	Gibt das aktuelle Datum und die aktuelle Zeit des Rechners aus	15.04.25 16:09:01
Datum ()	Gibt das aktuelle Datum des Rechners aus	15.04.25
Zeit ()	Gibt die aktuelle Uhrzeit des Rechners aus	16:09:01
Jahr (#Datumsangabe#)	Ermittelt aus einem Datum das Jahr	2025
Monat (#Datumsangabe#)	Ermittelt aus einem Datum den Monat	4
Tag (#Datumsangabe#)	Ermittelt aus einem Datum den Tag	15
Wochentag (#Datumsangabe#; [ErsterWochentag])	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.)	2 (Montag)

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: 15.04.2024 14:09:01
<code>DatTeil (Intervall; #Datumsangabe#)</code>	Gibt einen Teil des Datums aus	<code>DatTeil ("ww"; #15.04.2025 14:09:01#)</code> Ergebnis: 16 (16. Kalenderwoche)
<code>DatAdd (Intervall; Zahl; #Datumsangabe#)</code>	Gibt ein Datum aus, zu dem ein Wert hinzuaddiert wurde	<code>DatAdd ("t"; +3; #15.04.2025 14:09:01#)</code> Ergebnis: 18.04.2025 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"; #15.04.2025#; #14.08.2025#)</code> Ergebnis: 17 (Wochen)

Über die zusätzlichen Argumente `ErsterWochentag` und `ErsteWocheDesJahres` 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.

Intervall	Beschreibung	Ergebnis
"jjjj"	Jahr	2025
"q"	Quartal	2
"m"	Monat	4
"j"	Kalendertag	105
"t"	Tag	15

Intervall	Beschreibung	Ergebnis
"w"	Wochentag	3
"ww"	Kalenderwoche	16
"h"	Stunde	14
"n"	Minute	9
"s"	Sekunde	1

Formatierung von Datum und Uhrzeit

Um eine benutzerfreundliche Darstellung der Ergebnisse zu erhalten, können Sie die Datums- und Uhrzeitwerte formatieren.

Funktion	Beispiel: 03.02.2024
Format (Ausdruck; [Format]; [ErsterWochentag]; [ErsteWocheDesJahres]):	Format (#03.02.2024#; "jjjj"; 2)

Varianten des Datumsformats:

Format	Beschreibung	Ausgabe
"t"	Tag ohne führende Null	3
"tt"	Tag mit führender Null	03
"ttt"	Abgekürzter Name des Wochentages	Mo
"tttt"	Vollständiger Name des Wochentages	Montag
"m"	Monat ohne führende Null	2
"mm"	Monat mit führender Null	02
"mmm"	Abgekürzter Name des Monats	Feb
"mmmm"	Vollständiger Name des Monats	Februar
"jj"	Jahreszahl zweistellig	24
"jjjj"	Jahreszahl vierstellig	2024

Weitere Beispiele zum Rechnen mit Datum und Uhrzeit

Mit Datumswerten rechnen

Aufgrund des Datums der Bestellung als Auftragsdatum sollen Liefertermin und Zahlungseingang ermittelt werden.

- ▶ Erstellen Sie in der Abfrage *Lieferdatum* (Übungsdatei *Buero01*) ein neues Feld *LieferDatum* und berechnen Sie den Wert mit $[BestellDatum]+10$.
- ▶ Erstellen Sie ein weiteres Feld *Rechnungsdatum* und berechnen Sie dort $[LieferDatum]+30$.

Monatsnamen aus Datumswerten bestimmen

Für die Auswertung der monatlichen Verkäufe wird das Verkaufsdatum als Monatsname angezeigt.

- ▶ Geben Sie folgenden Ausdruck als neues, berechnetes Feld der Abfrage *Ausgaenge* ein:

Verkaufsmonat: `Format([Ausgangsdatum]; "mmmm")`

Ausgaenge		AusgaengeMonate	
ArtikelID	Ausgangsdatum	ArtikelID	Verkaufsmonat
ALG-003	07.01.2024	ALG-003	Januar
ALG-005	14.01.2024	ALG-005	Januar
ALG-007	22.01.2024	ALG-007	Januar
ALG-007	24.01.2024	ALG-007	Januar
EDV-001	09.02.2024	EDV-001	Februar
ALG-006	15.02.2024	ALG-006	Februar
EDV-005	18.02.2024	EDV-005	Februar
ALG-004	24.02.2024	ALG-004	Februar

Anzahl der Tage ermitteln

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

- ▶ 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 Eigenschaftenblatt 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
7	Hoffmann, Theresa, Dorfstr. 4, 55142, Mausbach
8	Klinker, Ralf-Rüdiger, Buchenweg 23, 55134, Mainz

Abfrage mit zusammengesetzten Datenfeldern

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) y	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"`

VertriebsID	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*"`

Nach dem Speichern der Abfrage wird die Funktion und das Kriterium eventuell in eine neue Spalte verschoben.

1.5 Umwandlungsfunktionen

Unverträglichkeiten von Datentypen

Die Datenfelder einer Tabelle haben oft verschiedene Datentypen. Wird jedoch mit Daten unterschiedlicher Datentypen gerechnet, können Fehlermeldungen auftreten, da die Datentypen zum Rechnen kompatibel sein müssen. Möchten Sie mit nicht kompatiblen Daten rechnen, können Sie die Umwandlungsfunktionen verwenden, um einen Datentyp anzupassen.

Andere Anwendungen für Umwandlungsfunktionen

- ✓ Für wissenschaftliche Berechnungen benötigen Sie Ergebnisse mit hoher Genauigkeit und größeren Wertebereichen. Dafür können Sie beispielsweise mit einer Umwandlungsfunktion den Datentyp *Integer* in *Double* umwandeln.
- ✓ Zahlenwerte sollen als Währung auf zwei Nachkommastellen gerundet und mit dem €-Symbol dargestellt werden.

Sie können mit Textfeldern, die ausschließlich Ziffern und Dezimaltrennzeichen enthalten, Berechnungen wie mit Zahlenfeldern vornehmen. Umgekehrt können Sie Funktionen mit Zeichenfolgen auf Zahlen- und Datumsfelder anwenden. Access nimmt die notwendigen Umwandlungen automatisch vor.

Beispiele für Umwandlungsfunktionen

Funktion	Beschreibung	Beispiel ="70ABC-8844"
<code>Str (Zahl)</code>	Eine Zahl wird in Text umgewandelt. Eine Stelle wird für das Vorzeichen reserviert, die Ausrichtung und Sortierreihenfolge ändern sich, (da z. B. 10 als Text vor 2 als Text kommt).	<code>Str (33, 67)</code> Ergebnis: "33.67"