



Alle Access-Objekte

Tabellen

- tblAbteilung
- tblFahrtenbuch
- tblFahrzeug
- tblFahrzeugModell
- tblFlug

tblFlug

flugID
 flugDatum
 flugZeit
 ankunftsZeit
 fluggesellschaftID
 flugnummer



```

Access
Option Compare Database
Option Explicit

'Deben Sie in Direktbereich (F10-F) AutorenfeldFormnameFormalisieren ein.
und Spalten auf.
'Diese Funktion spaltet den Inhalt der Felder "Autoren" der Hilfetabelle in Vor-
und Nachnamen auf.
Public Funktion AutorenfeldFormnameFormalisieren() As Boolean
    Dim strID As String
    Dim strVorname As String
    Dim strNachname As String
    Dim strVorname As Long
    Dim strNachname As Long
    Dim strVorname As String
    Dim strNachname As String

    'Zähler: Öffnen wir DAO-Recordsets zum Ändern der Daten per Programmcode
    Dim dbCursor As DAO.Recordset
    Dim strCursor As DAO.Recordset

    'Der strCursor ist ein Cursor, geöffnet.
    Dim strCursor As DAO.Recordset
    Dim strCursor As DAO.Recordset

    'Die strCursor ist ein Cursor, geöffnet.
    Dim strCursor As DAO.Recordset
    Dim strCursor As DAO.Recordset

    'Die strCursor ist ein Cursor, geöffnet.
    Dim strCursor As DAO.Recordset
    Dim strCursor As DAO.Recordset
  
```



Wolfram Langer

Access

Das umfassende Handbuch

- ▶ Tabellen, Formulare, Berichte und Abfragen erstellen
- ▶ Datenbankdesign optimieren; Daten filtern, sortieren und auswerten
- ▶ Inkl. Datenimport und -export, SQL, VBA- und DAO-Programmierung, Desktopanwendungen entwickeln, SQL Server als Backend

Alle Codebeispiele und Beispieldatenbanken zum Download

Kapitel 2

Access als Datenbank: Tabellen

*In einer relationalen Datenbank sind die Daten in Tabellen gespeichert.
Zwischen den Tabellen bestehen Beziehungen.*

Wie der Name schon sagt, dreht sich bei einer Datenbank alles um Daten. Access ist ein Datenbanksystem für relationale Datenbanken. Das bedeutet: Die Daten stehen in Tabellen, die miteinander in Beziehung stehen. In diesem Kapitel dreht sich alles um das Thema »Tabellen«:

- ▶ Tabellen erstellen und ändern
- ▶ Daten eingeben, ändern und löschen

Das ist quasi die Grundausstattung zum Arbeiten mit Access. Darüber hinaus werde ich in diesem Kapitel ausführlich auf das *Datenbankdesign* und auf die *Normalformen* eingehen. Beide Themen beschreiben, wie Sie Daten möglichst optimal in Tabellen ablegen. Dazu kann man sehr abstrakt und theoretisch vorgehen. Ich habe mich bemüht, möglichst nahe an der Praxis zu bleiben. Der eine oder andere Profi in Sachen Datenbanktheorie möge es mir bitte verzeihen, wenn meine Ausführungen an einigen Stellen nicht präzise genug sind. Mir ist das Praktische wichtiger: Wie gelangen Sie mit einem guten Datenbankdesign und den Normalformen zu einer strukturierten Sammlung von Daten?

Ich habe diesbezüglich schon sehr viele Diskussionen zwischen Entwicklern erlebt und viele Datenbanken mit gravierenden Fehlern im Datenbankdesign gesehen. Beides zeigt, dass Datenbankdesign und Normalformen keine einfache Sache sind. Aber keine Angst, ich habe das Kapitel mit vielen Beispielen, einigen Geschichten aus der Praxis und schließlich mit den Wahrheiten gespickt, die sich innerhalb der Datenbank-Community bewährt haben (Best Practices). Damit werden Sie einen guten Eindruck davon gewinnen, in welche Richtung der Weg zu einer Datenbank führt, die Ihnen bei der Nutzung keine Sorgen bereitet.

2.1 Ein Schnelleinstieg in relationale Datenbanken

Als ersten Einstieg werde ich Ihnen zeigen, wie Sie in einer Desktop-Datenbank eine leere Tabelle erstellen und mit Daten befüllen können. Das Ergebnis ähnelt einer Excel-Tabelle: Es ist eine *unstrukturierte Sammlung von Daten*.

Eine einfache Tabelle (egal, ob in Excel oder Access) – das ist die intuitive Herangehensweise an eine Datenbank. Sie kennen das bestimmt aus der Praxis: die Bestellliste, die Liste mit den Lagerbeständen usw.

Vorteil gegenüber einer Excel-Tabelle

Selbst mit einer einfachen Access-Tabelle erreichen Sie schon einen Mehrwert gegenüber Excel. Eine Access-Datenbank kann gleichzeitig von mehreren Benutzern zum Lesen und Schreiben geöffnet sein (*Mehrbenutzerfähigkeit*). Damit entfallen viele der lästigen Telefonate wie: »Kannst du bitte mal aus der Excel-Tabelle gehen?«

In den Abschnitten zum Datenbankdesign (Abschnitt 2.2 und Abschnitt 2.7) und den Normalformen (Abschnitt 2.8) werde ich Ihnen zeigen, wie Sie systematisch zu einer strukturierten Sammlung von Daten gelangen und welcher Mehrwert damit verbunden ist.

2.1.1 Eine neue Desktop-Datenbank mit Access erstellen

Um zu starten, müssen wir zunächst einmal eine leere Desktop-Datenbank erstellen:

1. Starten Sie Microsoft Access.
2. Klicken Sie auf die Schaltfläche NEUE LEERE DATENBANK (siehe Abbildung 2.1).

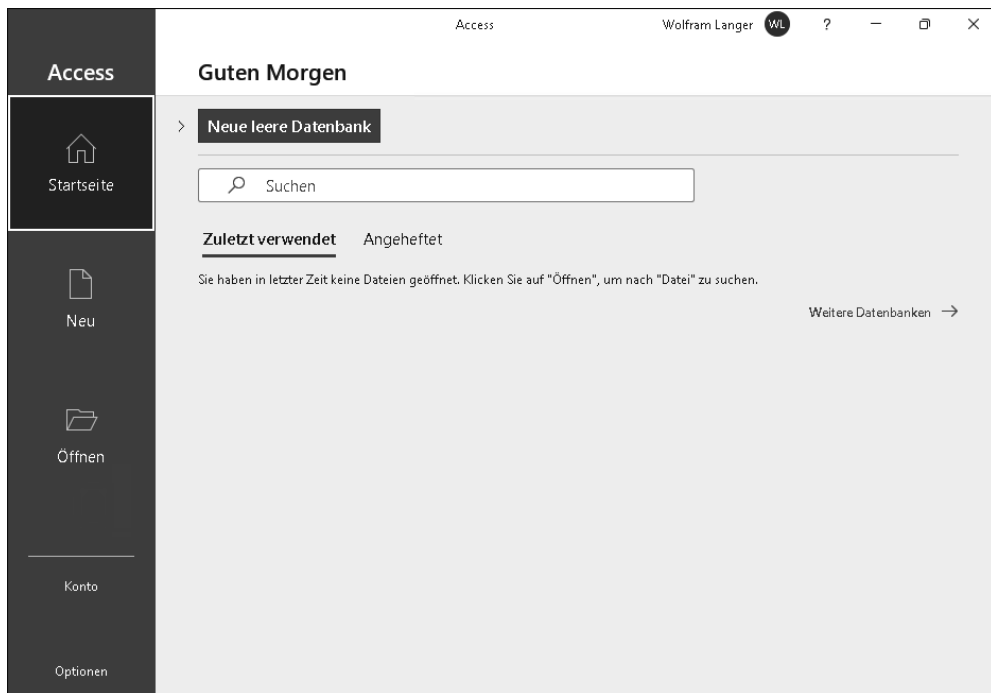


Abbildung 2.1 Die Schaltfläche für eine neue Datenbank finden Sie ganz oben.

3. Wählen Sie den Speicherort aus, geben Sie einen Namen für die Datenbankdatei an, und klicken Sie auf OK (siehe Abbildung 2.2).

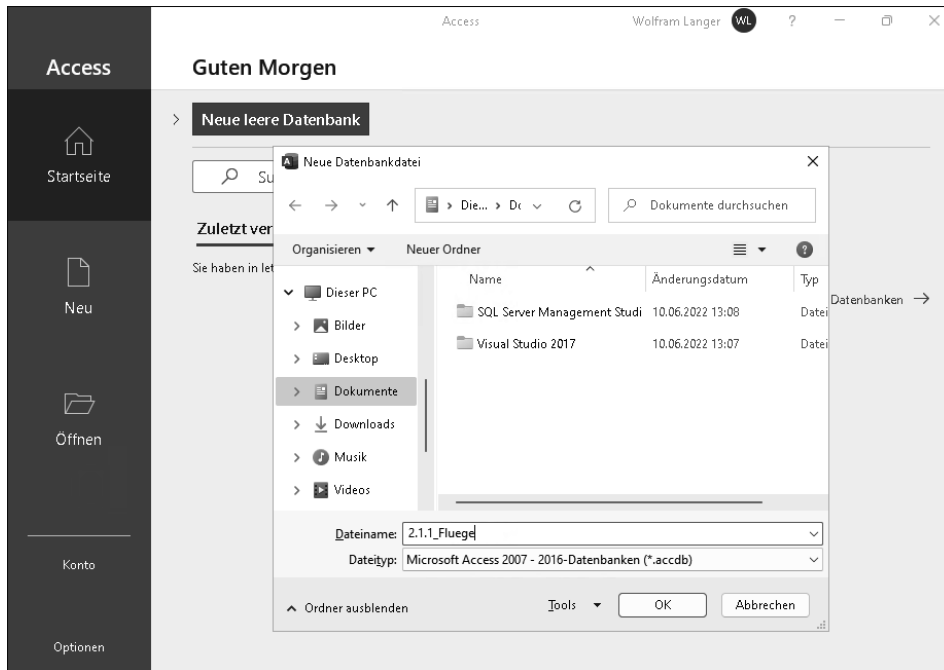


Abbildung 2.2 Geben Sie den Namen und den Speicherort für die ».accdb«-Datei an.

Eine Desktop-Datenbank benötigt immer eine Datei auf der Festplatte

Sie müssen an dieser Stelle einen Dateinamen angeben. Eine Datenbank, die es nur im Arbeitsspeicher gibt (ähnlich einer noch nicht gespeicherten Word- oder Excel-Datei), gibt es in Access nicht. Neben den *.accdb*-Dateien werden übrigens auch noch die älteren *.mdb*-Dateien unterstützt.

Sie gelangen zum Hauptbildschirm von Access (siehe Abbildung 2.3). Auf der linken Seite sehen Sie im *Navigationsbereich* ❶, welche *Datenbankobjekte* in Ihrer Datenbank enthalten sind. Abgesehen von der einen Tabelle ❷ ist die Datenbank erst einmal leer. Später werden Sie im Navigationsbereich die verschiedenen Arten von Datenbankobjekten wiederfinden:

- ▶ Tabellen
- ▶ Abfragen
- ▶ Formulare
- ▶ Berichte
- ▶ Makros
- ▶ VBA-Module

Wir befassen uns in diesem Kapitel nur mit Tabellen.



Abbildung 2.3 Access hat in der neuen Datenbank gleich eine leere Tabelle erstellt.

2.1.2 Eine Tabelle in Access erstellen

In der Excel-Tabelle *O2_Access_als_Datenbank_Tabellen\2.1.1_Fluege.xlsx* (siehe Abbildung 2.4) habe ich einige Datensätze mit Flugbuchungen vorbereitet.

Datum	Abflug	Ankunft	Fluggesellschaft	Flugnummer	Flug von	nach	Passagiere
03.08.2022	15:00	16:20	easyJet	EJU5846	Wien	Berlin-Brandenburg	
03.08.2022	15:10	16:40	Austrian Airlines	OS131	Wien	Frankfurt am Main	
03.08.2022	15:40	17:20	Air France	AF2334	Paris-Charles-de-Gaulle	Berlin-Brandenburg	
03.08.2022	15:50	17:35	Air France	AF2035	Berlin-Brandenburg	Paris-Charles-de-Gaulle	
03.08.2022	15:55	17:15	Air France	AF1919	Frankfurt am Main	Paris-Charles-de-Gaulle	
03.08.2022	15:55	05:35	Lufthansa	LH401	New York John F. Kennedy	Frankfurt am Main	Margot Wilke (Abt. Verkauf)
03.08.2022	16:00	17:10	Lufthansa	LH191	Berlin-Brandenburg	Frankfurt am Main	Ramona Rathke (Projekt GH)
03.08.2022	16:10	17:15	Lufthansa	LH2890	Berlin-Brandenburg	Köln/Bonn	Eva Hachmann (Geschäftsführer)
03.08.2022	16:15	16:55	Lufthansa	LH914	Frankfurt am Main	London Heathrow	

Abbildung 2.4 Eine Liste von 50 Flügen mit mehreren Buchungen – das ist eine kleine »Datenbank«, die mit Excel nachempfunden ist.

In ähnlicher Weise können Sie eine Tabelle in Access erstellen:

1. Falls Sie die von Access vorbereitete Tabelle versehentlich geschlossen haben: Klicken Sie auf **ERSTELLEN • TABELLEN • TABELLE**.
2. Klicken Sie auf **ZUM HINZUFÜGEN KLICKEN**, um ein neues *Feld* (Spalte) einzufügen.

3. Im Kontextmenü zeigt Access den *Felddatentyp* an (Abbildung 2.5). Wählen Sie der Einfachheit halber den Eintrag KURZER TEXT aus.

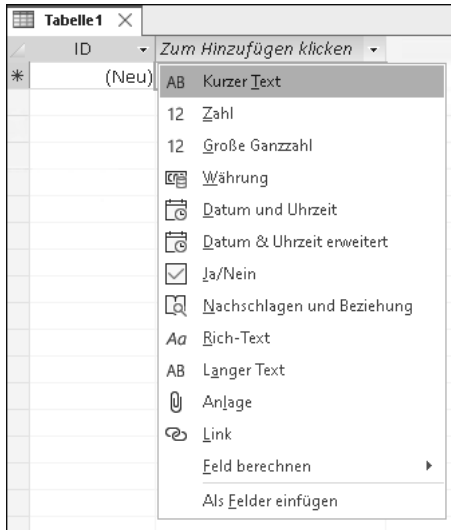


Abbildung 2.5 Jedes Feld hat einen Felddatentyp. Details dazu kommen später; wählen Sie erst einmal »Kurzer Text« aus.

4. Geben Sie den Namen des Feldes an, und wiederholen Sie die Schritte 2–4, um diese Felder zu erstellen:
- »AbflugDatum«
 - »AbflugZeit«
 - »AnkunftZeit«
 - »Fluggesellschaft«
 - »Flugnummer«
 - »AbflugFlughafen«
 - »AnkunftFlughafen«
 - »Passagiere«

Übrigens würde ich bei der Benennung von Feldnamen immer auf Leer- und Sonderzeichen verzichten, weil dies später Probleme bereiten könnte. Also verwenden Sie lieber »Flugnummer«, nicht jedoch »Flug-Nr.«.

5. Speichern Sie die Tabelle unter dem Namen *tblFlug* ab, indem Sie in der Symbolleiste für den Schnellzugriff auf **SPERICHERN** **1** klicken (oder **[Strg] + [S]** drücken).

Im Navigationsbereich links erscheint die neu erstellte Tabelle *tblFlug* **2** wie in Abbildung 2.6. Sie können das Datenblatt schließen **3** und das Datenbankobjekt per Doppelklick jederzeit wieder öffnen.

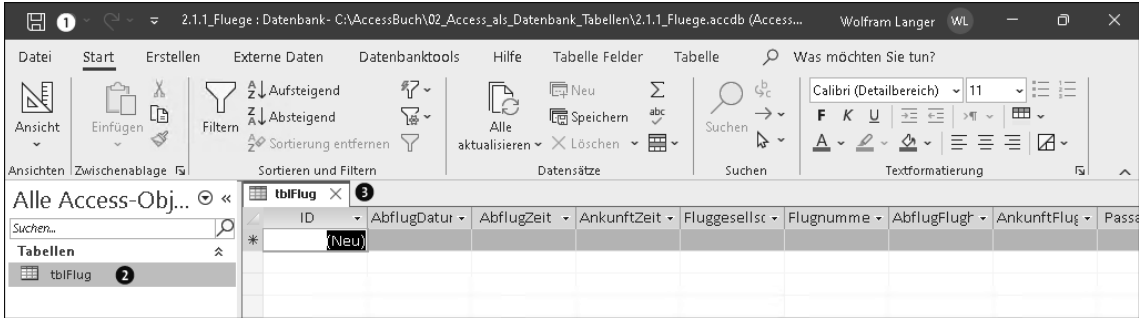


Abbildung 2.6 So sieht die leere Access-Tabelle mit neun Feldern aus.

2.1.3 Die Tabelle mit Inhalt füllen

Noch ist die Tabelle leer – das soll sich jetzt ändern. Tippen Sie einfach einmal eine Flugbuchung in die erste Zeile.

ID	AbflugDatum	AbflugZeit	AnkomstZeit	Fluggesellsc	Flugnummer	AbflugFlug	AnkomstFlug	Passagier
1	03.08.2022	15:00:00	16:					
(Neu)								

Abbildung 2.7 Jede Zeile der Tabelle ist ein Datensatz. Sobald Sie mit dem Tippen beginnen, erstellt Access den neuen Datensatz.

Am linken Rand der Zeile erscheint ein kleiner Stift, wie auch in Abbildung 2.7. Access zeigt damit an, dass der Datensatz im *Editiermodus* (*Bearbeitungsmodus*) ist. Sobald Sie in die nächste Zeile gehen, verschwindet der Stift. Erst jetzt wird der Datensatz wirklich in der *.accdb*-Datei gespeichert.

Gemeinsam mit einer Access-Datenbank arbeiten

Sobald der Stift verschwunden ist, ist der Datensatz gespeichert. Die Datenbank selbst muss nicht noch einmal gespeichert werden. Falls andere Personen die Datenbank gleichzeitig geöffnet haben, werden die Änderungen frühestens jetzt für andere sichtbar. Access aktualisiert die Anzeige nach einiger Zeit automatisch (oder aktualisieren Sie sie mit $\square + [F9]$ manuell).

Leider werden neue Datensätze nicht immer automatisch angezeigt. Aber immerhin funktioniert Folgendes zuverlässig:

- ▶ Änderungen an Datensätzen werden sichtbar.
- ▶ Löschungen von Datensätzen werden sichtbar.
- ▶ Access warnt, bevor Änderungen einer anderen Person überschrieben werden.

Damit lässt sich sehr gut gemeinsam an einer Datenbank arbeiten.

Um Tipparbeit zu sparen, können Sie die Excel-Daten bequem über die Zwischenablage in eine Access-Tabelle importieren.

1. Öffnen Sie die Excel-Tabelle *O2_Access_als_Datenbank_Tabellen\2.1.1_Fluege.xlsx*.
2. Markieren Sie den Bereich mit den Daten, jedoch ohne die Überschriften (Bereich A2:H51).
3. Drücken Sie **[Strg] + [C]**.
4. Wechseln Sie zu Access, und öffnen Sie die Tabelle *tblFlug*, falls nicht schon geschehen.
5. Markieren Sie in der Kopfzeile die Spalten »AbflugDatum« bis »Passagiere«, und fügen Sie die Datensätze mit **[Strg] + [V]** ein.
6. Access teilt Ihnen mit, dass 50 Datensätze eingefügt werden (siehe Abbildung 2.8). Bestätigen Sie den Import, indem Sie auf JA klicken.

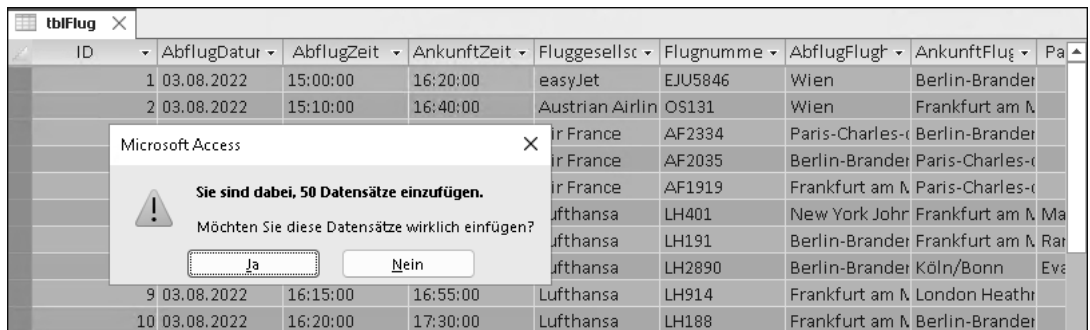


Abbildung 2.8 Über die Zwischenablage können Sie bequem mehrere Datensätze gleichzeitig aus der Excel-Tabelle importieren.

Noch ein paar Worte zur ersten Spalte, »ID«: Sie ist ein *AutoWert-Feld* und gleichzeitig *Primärschlüssel* der Tabelle *tblFlug*. Access vergibt eine Identifikationsnummer automatisch; Sie können diese Nummer nicht selbst festlegen oder ändern. Wie ich Ihnen noch zeigen werde, ist der Primärschlüssel in einer relationalen Datenbank elementar wichtig.

2.2 Datenbankdesign

Erfahrungsgemäß entstehen neue Datenbankprojekte häufig wie in Abschnitt 2.1, »Ein Schnelleinstieg in relationale Datenbanken«, beschrieben. Und vielleicht kennen Sie das aus dem Arbeitsalltag: Zunächst beginnt alles mit einer Excel-Tabelle, diese wird im Laufe der Zeit immer größer, und irgendwann wird das Ganze recht unübersichtlich. Spätestens wenn mehrere Benutzer mit derselben Excel-Tabelle arbeiten, klingelt permanent das Telefon: »Könntest du bitte gerade mal die Excel-Tabelle schließen, damit ich die neuen Daten eintragen kann?« oder: »Kannst Du die Datei für alle zur Bearbeitung freischalten?«. Der Grund für diese Anrufe ist die gegenseitige Sperrung der Datei. Spätestens zu diesem Zeitpunkt stellt sich heraus, dass die Datenbankmöglichkeiten von Excel ausgereizt sind.

Unabhängig davon, ob bereits die eine oder andere Excel-Tabelle als Datenbankkrücke vorhanden ist, am Anfang eines erfolgreichen Datenbankprojekts steht immer intensive Kopfarbeit. Dies ist die in meinen Augen ehrlich gesagt schwierige und anstrengende Arbeit des sogenannten *Datenbankdesigns*. Beim Prozess des Datenbankdesigns müssen Sie die folgenden Fragen klären:

- ▶ Welche Tabellen sind notwendig?
- ▶ Welche Daten gehören in genau welche Tabelle? (Normalisierung der Datenbank)
- ▶ Können Einschränkungen und Regeln definiert werden, um Ordnung in die Daten zu bringen?
- ▶ In welchen Beziehungen stehen die Tabellen untereinander?

Die Antworten zu diesen Fragen leiten sich aus Ihrem *Geschäftsmodell* ab. Dieser Begriff klingt vielleicht etwas hochtrabend, gerade wenn Sie eine Datenbank für Ihre private Musiksammlung erstellen möchten. Gemeint ist damit die Beschreibung der logischen Funktionsweise in Ihrem Projekt, d. h. wer oder was auf welche Weise miteinander agiert. Einfach gesagt: »Worum geht's?« Und genau da helfen Ihnen die oben aufgeführten Fragen weiter.

In einem Datenbankprojekt wird die Arbeit des Datenbankdesigns nie vollständig abgeschlossen sein. Für diese wichtige Arbeit nehme ich mir aber gerade beim Start eines neuen Projekts ausreichend Zeit – und genau das empfehle ich Ihnen ebenfalls. Denn Sie werden sehen: Der Aufwand für ein gutes Datenbankdesign zahlt sich im weiteren Projektverlauf für Sie fortwährend aus.

Grundsätzliches zum Datenbankdesign

Diese beiden folgenden Punkte sind so wichtig, dass sie einen eigenen Hinweiskasten verdient haben:

- ▶ Stellen Sie sich vor jedem Datenbankprojekt die Fragen aus dem Abschnitt vor dem Kasten, und beantworten Sie sie gewissenhaft.
- ▶ Nehmen Sie sich dazu bitte Zeit. Je sorgfältiger Sie planen, desto mehr Zeit und Nerven sparen Sie im Verlauf des Projekts.

Vertrauen Sie wenigstens in diesen beiden Punkten meinem (aus eigener Erfahrung abgeleiteten) Rat. Nicht jede Erfahrung müssen Sie zwangsläufig selbst machen, es sei denn, auch Sie wollen der Kaffeerösterei Ihres Vertrauens durch häufige Nachtschichten zu deutlichen Umsatzsteigerungen verhelfen.

In diesem Abschnitt lassen wir die Excel-Welt hinter uns. Ich zeige Ihnen, wie Sie eine Datenbank mit mehreren Tabellen erstellen, dabei die Übersicht behalten und durch richtiges Datenbankdesign Ordnung in Ihre Datenbank bringen. Sobald Sie den Dreh raus haben, werden Sie feststellen, dass Datenbankdesign richtig Spaß macht und dass Sie gar nicht mehr aufhören wollen!

2.2.1 Die Welt in Tabellen abbilden

In unserem Beispiel geht es darum, Flugbuchungen in Tabellen festzuhalten. Während der ersten Besprechungen eines neuen Datenbankprojekts frage ich gerne: »Was genau soll denn gespeichert werden?«, »Was ist wichtig?« und »Was ist unwichtig oder soll erst einmal nicht abgebildet werden?«.

Eine Antwort, die ich dann immer wieder höre, lautet: »Alles ist wichtig.« Aber wenn Sie sich diese Antwort zu eigen machen, dann besteht die Gefahr, dass Sie sich verzetteln.

In unserer Beispieldatenbank sollen nur die folgenden Daten gespeichert werden:

1. Flüge mit Flugnummer, Datum sowie dem Zeitpunkt von Abflug und Ankunft
2. Flughäfen
3. Mitarbeiter, die auf einen oder mehrere Flüge gebucht sind
4. Abteilungen, in denen die Mitarbeiter beschäftigt sind
5. Projekte, an denen die Mitarbeiter beteiligt sind

Dies ist schon eine ganze Menge unterschiedlicher Daten. Grundsätzlich empfehle ich Ihnen, lieber erst einmal mit einem überschaubaren Rahmen von Daten zu starten und Ihre Datenbank danach schrittweise zu erweitern.

In einer relationalen Datenbank versuchen wir, die reale Welt in Tabellen abzubilden. Für jede Art von Gegenstand der realen Welt wird es eine eigene Tabelle geben. In unserem Beispiel werden wir also mindestens fünf Tabellen erstellen:

1. Tabelle *tblFlug*
2. Tabelle *tblFlughafen*
3. Tabelle *tblMitarbeiter*
4. Tabelle *tblAbteilung*
5. Tabelle *tblProjekt*

Anders gesagt: Packen Sie nicht alles in eine Tabelle. Das ist nämlich so ähnlich, als ob Sie in Ihrem Schreibtisch nur eine einzige Schublade nutzen und dort alle möglichen Sachen hineinwerfen. Gut, ich muss zugeben, dass in meinem Regal auch eine »allgemeine Ablage« (eine schöne Umschreibung für »Chaos-Box«) steht. Also eine Zauberkiste, in die ich alle möglichen Schreiben und Notizen werfe und in der ich diese Dinge manchmal auch wiederfinde (eben herauszaubere). Aber ganz ehrlich: Mir persönlich reicht eine dieser Zauberkisten!

Datenbanken sollten das genaue Gegenteil sein: Unterschiedliche Dinge gehören in unterschiedliche Tabellen, das schafft von vornherein Ordnung. Damit Sie und Ihre Kollegen in Bezug auf die Elemente der Datenbank die gleiche Sprache sprechen, kann es an dieser Stelle hilfreich sein, ein Glossar wie in Tabelle 2.1 zu erstellen. Dazu reicht eine einfache Word-Datei, in der jeder Begriff kurz definiert wird. Übrigens finde ich, dass ein Glossar auch dann sehr

hilfreich sein kann, wenn man eine Datenbank allein, ohne Projektteam erstellt. Ich habe mich schon häufig dabei ertappt, dass ich Unterschiedliches in eine Tabelle packen wollte. Beim Aktualisieren des Glossars habe ich dann gemerkt, dass ich in Wirklichkeit eine neue Tabelle benötigte.

Begriff	Definition
Flug	Transport von Personen per Flugzeug von einem <i>Flughafen</i> zu einem anderen. Jeder Flug hat eine <i>Flugnummer</i> , die zusammen mit dem <i>Datum</i> eindeutig ist.
Flughafen	Örtlicher Start- und Endpunkt eines <i>Fluges</i> . In einer Stadt kann es auch mehrere Flughäfen geben. Jeder Flughafen hat einen eindeutigen Namen und einen eindeutigen <i>IATA-Code</i> . Die Liste der Flughäfen wird durch die IATA gepflegt (http://en.wikipedia.org/wiki/List_of_airports).
Flugnummer	Umgangssprachliche Bezeichnung für <i>flight code</i> gemäß der IATA-Definition. Besteht aus dem IATA-Code für die Fluggesellschaft (http://en.wikipedia.org/wiki/List_of_airlines) und einer Zahl (<i>flight number</i>). Beispiel: LH401
IATA	<i>International Air Transport Association</i> , der internationale Dachverband der Fluggesellschaften
...	...

Tabelle 2.1 Wichtige Begriffe sollten Sie in einem Glossar definieren, damit alle im Projektteam die gleiche Sprache sprechen.

In Abschnitt 2.8, »Durch die Normalformen das Datenbankdesign verbessern«, werde ich Ihnen zeigen, wie Sie Daten systematisch in unterschiedliche Tabellen trennen und welche Vorteile Sie dadurch erzielen. In Abschnitt 2.8.8, »Zweckmäßiges Datenbankdesign«, zeige ich Ihnen, wie Sie dabei möglichst nahe an der Realität bleiben und sich dadurch gleichzeitig Dokumentationsarbeit ersparen können.

Best Practice beim Abbilden der Welt in Tabellen

- ▶ Definieren Sie wichtige Begriffe (Glossar).
- ▶ Aktualisieren Sie das Glossar im Projektverlauf.
- ▶ Unterschiedliche Sachen gehören auch in unterschiedliche Tabellen.
- ▶ Bleiben Sie möglichst nahe an der Realität.

2.2.2 Namenskonventionen

Bevor wir die Tabellen erstellen, sollten wir eine scheinbar ganz banale Frage klären: Wie sollen die Tabellen überhaupt heißen? Sobald Sie mehr als eine Tabelle haben, ist diese Frage überhaupt nicht mehr trivial. Ich mache mir im Gegenteil jedes Mal sehr viele Gedanken darüber, wie ich eine Tabelle oder ein Feld treffend bezeichne und ob die Benennung den Inhalt der Tabelle auf den Punkt bringt.

Das lässt Access bei der Benennung technisch zu

In Access gibt es von technischer Seite nur sehr wenige Einschränkungen bei der Benennung von Tabellen und Feldern:

- ▶ maximal 64 Zeichen
- ▶ alle Zeichen mit Ausnahme von eckigen Klammern (»[« und »]«), Punkt (».«) und Ausrufezeichen (»!«)
- ▶ Leerzeichen nicht am Anfang oder am Ende

Rein technisch betrachtet, haben Sie also recht freie Wahl. Sie dürfen auch Wörter verwenden, die bei der Programmierung als reservierte Schlüsselwörter gelten (beispielsweise in der Abfragesprache SQL oder in der Programmiersprache Visual Basic for Applications).

Mit diesen Regeln der Benennung behalten Sie den Überblick

Trotz aller Freiheiten möchte ich Ihnen gerne einige freiwillige Einschränkungen bei der Benennung ans Herz legen, die sich in der Praxis bewährt haben:

1. Verwenden Sie im Namen von Tabellen (und anderen Datenbankobjekten) oder Feldern *keine Umlaute, Sonderzeichen oder Leerzeichen*.

Access erscheint mir recht fehlerfrei beim Umgang mit exotischen Zeichen zu sein. Das gilt aber nicht unbedingt für Erweiterungen, Programmbibliotheken, Werkzeuge etc. Solche Software ist mitunter nur mit einem englischen Windows und Access entwickelt oder getestet worden, sodass es mit deutschen Umlauten und Sonderzeichen Probleme geben kann. Zu dieser Art von Fehlern werden Sie im Internet auch nur begrenzt Informationen finden, denn die große Anzahl der englischsprachigen Access-Anwender wird nie auf diese Fehler stoßen.

Diese ärgerliche und, wie ich selbst erleben musste, auch ziemlich frustrierende Fehlersuche möchte ich Ihnen gerne ersparen. Verwenden Sie daher ausschließlich die Groß- und Kleinbuchstaben von A bis Z, Zahlen und wenn notwendig den Unterstrich (»_«).

2. Sie erleichtern sich die Arbeit bei der Benennung, indem Sie *stets die Einzahl verwenden*. Also »Flug« als Tabellename, nicht »Flüge«. Zum einen vermeiden Sie von vornherein die Gefahr, einen Umlaut einzutippen, denn viele Pluralwörter haben in der deutschen Sprache mindestens einen Umlaut! Zweitens steht in einer Datenbanktabelle in den seltensten Fällen wirklich nur ein einziger Datensatz. Daher ist es nicht sinnvoll, im Tabellen-

namen nach der Einzahl oder der Mehrzahl zu unterscheiden. Verwenden Sie stattdessen die kürzere Singularform.

3. Manchmal reicht ein einziges Wort nicht aus, um den Inhalt einer Tabelle oder eines Feldes treffend zu beschreiben. Denken Sie einmal an eine Tabelle, in der eingetragen wird, welcher Mitarbeiter von wann bis wann in einer bestimmten Abteilung des Unternehmens tätig war. Bitte verwenden Sie *bei zusammengesetzten Wörtern* keine Leerzeichen, auch wenn das technisch möglich wäre. Besser ist es, wenn Sie entweder *den Unterstrich* (»Mitarbeiter_taeig_in_Abteilung«) oder die sogenannte *CamelCase-Schreibweise* verwenden, also immer den ersten Buchstaben eines Wortes großschreiben (»MitarbeiterTaetig-InAbteilung«). Da ich ein Befürworter von wenig Schreiarbeit und damit von wenigen Zeichen bin, verwende ich den CamelCase-Stil.
4. In der Access-Welt hat sich die Benennung nach der sogenannten *Leszynski Naming Convention* (LNC) – einer Variante der sogenannten *ungarischen Notation* – eingebürgert. Nach dieser Konvention bekommt jeder Name ein *Präfix*, das sich aus dem Typ ableitet. Eine Tabelle wird nach der LNC beispielsweise *tblFlug* genannt. Der Vorteil liegt auf der Hand: Sie erkennen anhand des Präfixes sofort, ob Sie eine Tabelle (*tbl*), eine Abfrage (*qry*) oder etwas ganz anderes vor sich haben. Das erhöht die Verständlichkeit Ihrer Datenbank ungemein! Tabelle 2.2 gibt Ihnen einen Überblick über die Präfixe für die gängigsten Datenbankobjekte.

Datenbankobjekt	englische Bezeichnung	LNC-Präfix
Tabelle	<i>table</i>	<i>tbl</i>
Abfrage	<i>query</i>	<i>qry</i>
Formular	<i>form</i>	<i>frm</i>
Bericht	<i>report</i>	<i>rpt</i>
Makro	<i>macro</i>	<i>mcr</i>
Modul	<i>module</i>	<i>bas</i> (angelehnt an »basic module«)
Klassenmodul	<i>class module</i>	<i>cls</i>

Tabelle 2.2 Präfixe für Datenbankobjekte nach der »Leszynski Naming Convention« (LNC)

Neben diesen sieben Präfixen gibt es weitere, unter anderem für verschiedene Arten von Tabellen oder Abfragen. Ob eine solche feinere Unterteilung nützlich ist, mag Geschmackssache sein. Ich begnüge mich mit *tbl* für Tabellen und *qry* für Abfragen.

Auch für die Bezeichnung von Feldern nach Felddatentyp wurden Präfixe in der LNC festgelegt. Mit Felddatentypen werden wir uns ausführlich in Abschnitt 2.3, »Erstellen von Tabellen«, beschäftigen. In Bezug auf eine Namenskonvention gibt es das Problem, dass sich

Felddatentypen zwischen Access, SQL und anderen Datenbanksystemen allein vom Namen her schon unterscheiden und es daher leicht zu Verwirrung kommt. Von LNC-Präfixen für Feldnamen rate ich Ihnen daher ab.

Felddatentyp	englische Bezeichnung	Jet-SQL- Datentyp	LNC-Präfix
AUTOWERT INKREMENT	<i>AutoNumber Increment</i>	COUNTER	<i>ids</i>
AUTOWERT REPLIKATIONS-ID	<i>AutoNumber Replication ID</i>	–	<i>idr</i>
AUTOWERT ZUFALL	<i>AutoNumber Random</i>	–	<i>idn</i>
BINÄR	<i>Binary</i>	BINARY	<i>bin</i>
BYTE	<i>Byte</i>	TINYINT	<i>byt</i>
DATUM/UHRZEIT	<i>Date/Time</i>	DATETIME	<i>dtm</i>
DATUM/UHRZEIT ERWEITERT	<i>Date/Time Extended</i>	–	–
DOUBLE	<i>Double</i>	FLOAT	<i>dbl</i>
GROSSE GANZZAHL	<i>Large Number</i>	–	–
INTEGER	<i>Integer</i>	SMALLINT	<i>int</i>
JA/NEIN	<i>Yes/No (Boolean)</i>	BIT	<i>ysn</i>
KURZER TEXT	<i>Short Text</i>	CHAR VARCHAR	<i>chr</i> (angelehnt an »character«)
LANGER TEXT	<i>Long Text</i>	TEXT	<i>mem</i> (angelehnt an »memo«)
LINK	<i>Hyperlink</i>	–	<i>hlk</i>
LONG INTEGER	<i>Long</i>	INTEGER	<i>lng</i>
OLE-OBJEKT	<i>OLE Object</i>	IMAGE	<i>ole</i>
SINGLE	<i>Single</i>	REAL	<i>sng</i>
WÄHRUNG	<i>Currency</i>	MONEY	<i>cur</i>

Tabelle 2.3 Falls Sie wirklich LNC-Präfixe für Feldnamen verwenden wollen, beachten Sie bitte, dass z. B. das Präfix »int« nicht dem SQL-Datentyp »INTEGER« entspricht!

Eine andere Herangehensweise ist, die Felder mit einem thematischen Präfix zu versehen. Gemeint ist eine *eindeutige* Abkürzung wie beispielsweise »flg« für die Tabelle *tblFlug*. Die Felder heißen dann »flgID«, »flgAbflugDatum«, »flgAbflugZeit« ... Ein wesentlicher Vorteil liegt auf der Hand: Jeder Feldname ist nun eindeutig in der gesamten Datenbank, Sie können schon anhand des Präfixes die Tabelle erkennen oder erraten. Ich könnte mir vorstellen, dass diese Konvention für kleinere Datenbanken praktikabel ist. Am besten fügen Sie eine zusätzliche Spalte in Ihrer Word-Datei des Glossars ein, dann haben Sie das Präfix immer im Blick. Achten Sie in jedem Fall darauf, dass jedes Präfix eindeutig für eine Tabelle ist, denn sonst ist das Konzept der thematischen Präfixe für Feldnamen sinnlos. Sobald Ihre Datenbank mehrere Tabellen enthält, kann es durchaus schwierig werden, geeignete und gleichzeitig eindeutige Präfixe zu finden. Ich verwende thematische Präfixe aus genau diesem Grund nicht.

Sie haben sicherlich gemerkt, dass es bei den Namenskonventionen unterschiedliche Meinungen und Herangehensweisen gibt. Ich empfinde Präfixe bei der Bezeichnung von Feldern als wenig sinnvoll, jedoch *tbl* für Tabellen und *qry* für Abfragen als sehr hilfreich. Wie Sie sich auch immer entscheiden: Legen Sie sich zu Projektbeginn auf eine einheitliche Richtlinie zur Benennung von Tabellen und Feldern fest. Das ist umso wichtiger, wenn Sie in einem Team arbeiten.

Best Practice bei der Benennung von Tabellen und Feldern

- ▶ *tbl* als Präfix für den Tabellennamen
- ▶ keine Umlaute
- ▶ keine Sonderzeichen
- ▶ keine Leerzeichen
- ▶ maximal 64 Zeichen
- ▶ einheitlich CamelCase oder den Unterstrich verwenden
- ▶ stets die Einzahl verwenden

Unter Berücksichtigung dieser Namenskonventionen sieht die Tabelle *tblFlug* nun so aus wie in Abbildung 2.9:

tblFlug
FlugID
AbflugDatum
AbflugZeit
AnkunftsZeit
Fluggesellschaft
Flugnummer
AbflugFlughafen
AnkunftFlughafen
Passagiere

Abbildung 2.9 Die Tabelle »tblFlug« unter Einhaltung der Namenskonventionen

Ihnen ist vielleicht aufgefallen, dass ich das erste Feld in »FlugID« umbenannt habe. Dieses Feld ist der sogenannte *Primärschlüssel*, ein äußerst wichtiges Element, das ich Ihnen im folgenden Abschnitt vorstellen möchte.

2.2.3 Der Primärschlüssel

Jede Tabelle sollte einen *Primärschlüssel* (engl. *primary key*) haben. Er ist eindeutig für einen Datensatz. Das heißt, wenn Sie den Primärschlüssel kennen, finden Sie auch den Datensatz wieder. Hinzu kommt, dass die Reihenfolge der Datensätze in einer Datenbanktabelle nicht festgelegt ist! Der Primärschlüssel ist entweder genau ein Feld der Tabelle oder die Kombination mehrerer Felder. Wichtig ist nur:

- ▶ Der Primärschlüssel ist *eindeutig*.
- ▶ Der Primärschlüssel *ändert sich nie*.

AutoWert-Feld als Primärschlüssel

Das Feld »FlugID« in Abbildung 2.9 ist der Primärschlüssel von *tblFlug*. Hierbei handelt es sich um ein *AutoWert-Feld*, das heißt um einen Zähler, mit dem Access automatisch alle Datensätze nummeriert. »FlugID« ist also eine künstlich vergebene Identifikationsnummer (ID), die keinen Bezug zur Realität hat.

Felder mit realen Inhalten als Primärschlüssel

Aber auch Felder mit realen Inhalten können Primärschlüssel sein. In den meisten Tabellen werden Sie ein oder mehrere Felder finden, die eindeutige Werte haben und sich daher prinzipiell als Primärschlüssel eignen. Man nennt sie *Schlüsselkandidaten*. Beispielsweise ist in unserer Tabelle *tblFlug* die Kombination der Felder »AbflugDatum« und »Flugnummer« eindeutig und daher ein Schlüsselkandidat (siehe Tabelle 2.4).

Es ist nicht ungewöhnlich, wenn Sie in einer Tabelle mehrere Schlüsselkandidaten finden. Ich empfehle Ihnen, möglichst jeden Schlüsselkandidaten (wie in Abschnitt 2.9.2, »Eindeutiger Schlüssel (Alternativschlüssel)«, erläutert wird) als *Alternativschlüssel* zu definieren.

Feld oder Kombination von Feldern	Schlüsselkandidat	Primärschlüssel	Alternativschlüssel	stellvertretender Schlüssel (»surrogate key«)	natürlicher Schlüssel (»natural key«)
»FlugID«	•	•		•	
»AbflugDatum« »Flugnummer«	•		•		•

Tabelle 2.4 Schlüsselkandidaten in der Tabelle »tblFlug«

Feld oder Kombination von Feldern	Schlüsselkandidat	Primärschlüssel	Alternativschlüssel	stellvertretender Schlüssel (»surrogate key«)	natürlicher Schlüssel (»natural key«)
»AbflugDatum« »AbflugZeit« »Fluggesellschaft« »AbflugFlughafen« »AnkunftFlughafen«	•		•		•
»AbflugDatum« »AnkunftZeit« »Fluggesellschaft« »AbflugFlughafen« »AnkunftFlughafen«	•		•		•
»AbflugDatum« »AbflugZeit«	<i>kein</i> Schlüsselkandidat, denn zwei Flüge können zeitgleich an unterschiedlichen Flughäfen starten				
»AbflugDatum« »AbflugFlughafen« »AbflugZeit«	ebenfalls <i>kein</i> Schlüsselkandidat, denn ein Flughafen kann zwei Startbahnen haben, auf denen zeitgleich Flugzeuge starten				

Tabelle 2.4 Schlüsselkandidaten in der Tabelle »tblFlug« (Forts.)

Eignet sich nun jeder Schlüsselkandidat als Primärschlüssel? Nein, denn es gibt noch eine weitere Eigenschaft des Primärschlüssels, die gerne vergessen wird: Der Primärschlüssel eines Datensatzes darf sich *nie* ändern. An dieser Einschränkung scheitern leider die meisten Schlüsselkandidaten. Irgendwann meldet sich beispielsweise eine Sachbearbeiterin bei Ihnen und sagt, sie habe sich bei der Flugnummer vertippt. Und schon haben Sie das Dilemma.

Künstlich vergebene IDs als Primärschlüssel helfen, Probleme zu vermeiden!

Mit künstlich vergebenen IDs als Primärschlüssel können Sie das Problem von vornherein umgehen. Diese Art von Primärschlüsseln hat sich in der Praxis derart bewährt, dass sie sogar einen eigenen Namen bekommen haben: *stellvertretende Schlüssel* (engl. *surrogate keys*). In Abgrenzung zu den künstlich vergebenen IDs bezeichnet man die anderen Schlüsselkandidaten übrigens als *natürliche Schlüssel* (engl. *natural keys*).

Best Practice zum Primärschlüssel

- ▶ Jede Tabelle bekommt einen Primärschlüssel.
- ▶ Legen Sie ein AUTOWERT-Feld mit künstlich vergebenen IDs als Primärschlüssel fest.
- ▶ Lehnen Sie die Benennung an den Tabellennamen an (»tblFlug« und Primärschlüssel »FlugID«).

2.2.4 Die Tabellenstruktur festlegen

In Abschnitt 2.2.1, »Die Welt in Tabellen abbilden«, haben wir uns ausführlich angesehen, welche Tabellen für unser Beispiel, die Flug-Datenbank, notwendig sind und wie wir sie benennen werden. In diesem Abschnitt werden wir die *Tabellenstruktur* festlegen. Tabellen in einer relationalen Datenbank (zu denen auch Access-Datenbanken gehören) unterscheiden sich in ein paar Merkmalen wesentlich von Excel-Tabellen:

Merkmal	Excel (ab Version 2007)	relationale Datenbank (Access)
Anzahl der Spalten	maximal 16.384	maximal 255
Bezeichnung der Felder (Spalten)	A, B, C ... (Spaltennamen werden üblicherweise in Zeile 1 eingetragen.)	Muss festgelegt werden.
Felddatentyp	variabel (kann individuell für jede Zelle festgelegt werden)	Muss für jede Spalte festgelegt werden.
Anzahl der Datensätze (Zeilen)	maximal 1.048.576	Unbegrenzt! (für eine Access-Datenbank limitiert durch maximal 2 GB für die Größe einer Datenbankdatei)
Sortierung der Datensätze (Zeilen)	Die Sortierung der Zeilen bleibt unverändert. Über SORTIEREN können Sie neu sortieren.	Die Sortierung der Zeilen kann sich ändern! Sortierte Datensätze können Sie aber jederzeit durch eine Abfrage erhalten.
Regeln zur Einschränkung	Datenüberprüfung (Gültigkeitskriterien können individuell für jede Zelle festgelegt werden.)	► Feldeinschränkungen ► Tabelleneinschränkungen
Mehrere Anwender können gleichzeitig mit einer Tabelle arbeiten.	nein	ja

Tabelle 2.5 Wenn Sie Flexibilität benötigen, ist Excel Ihr Werkzeug der Wahl. Mit relationalen Datenbanken schaffen Sie Ordnung – selbst bei extrem großen Datenmengen.

Bevor Sie Ihre Tabelle mit Daten füllen können, müssen Sie die Tabellenstruktur festlegen. Erst strukturieren, dann mit Daten füllen – das klingt zunächst unnötig kompliziert. Schließlich geht es doch auch anders, wie Sie es vielleicht von Excel gewohnt sind. An dieser Stelle möchte ich Sie noch einmal an Abschnitt 2.2.1, »Die Welt in Tabellen abbilden«, erinnern: Es ist *die* grundlegende Idee von Datenbanken, von vornherein für Ordnung zu sorgen – das heißt, Daten zu trennen und klare Regeln für die Daten (»Was kommt wohin?«) aufzustellen.

Mit diesen Schritten legen Sie die Tabellenstruktur fest

Nachdem Sie festgelegt haben, in welchen Tabellen Ihre Daten gespeichert werden sollen, müssen Sie für jede einzelne Tabelle die Tabellenstruktur festlegen. Dies umfasst jeweils die folgenden drei Schritte:

1. die Felder anlegen
2. für jedes Feld den Felddatentyp festlegen
3. Regeln zur Einschränkung festlegen

Auch diese Schritte sind sehr wichtig und gehören zum Prozess des Datenbankdesigns.

2.3 Erstellen von Tabellen

Beim Datenbankdesign habe ich Ihnen gezeigt, vor welchen Aufgaben wir beim Erstellen einer relationalen Datenbank stehen:

- ▶ Tabellen festlegen
- ▶ Tabellenstruktur festlegen (Felder mit Felddatentyp und Einschränkungen anlegen)
- ▶ Tabellen und Felder benennen

All das lässt sich auf dem Papier erledigen. Und vielleicht werden Sie staunen: Bei komplizierten Datenbankprojekten setze ich mich wirklich erst einmal mit Papier, Bleistift und Radiergummi (sehr wichtig!) hin.

In den nächsten Abschnitten werde ich Ihnen zeigen, wie Sie das Ergebnis, das Datenbankdesign, in Access übertragen. Und selbstverständlich dürfen Sie auch direkt in Access beginnen und das Datenbankdesign am Bildschirm aufbauen!

2.3.1 Die Entwurfsansicht

In Access können Sie eine Tabelle in zwei Ansichten öffnen:

1. in der *Datenblattansicht*
2. in der *Entwurfsansicht*

Durch Doppelklick auf eine bestehende Tabelle wird immer die Datenblattansicht geöffnet. Mit der Datenblattansicht werden wir uns ausführlich in Abschnitt 2.5, »Daten in Tabellen«, beschäftigen. Wenn Sie die Tabellenstruktur ändern möchten, müssen Sie die Tabelle in der Entwurfsansicht öffnen.

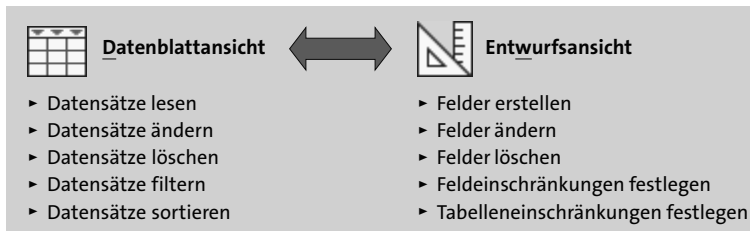


Abbildung 2.10 Verwenden Sie die Datenblattansicht, um die Daten in einer Tabelle zu ändern. Wechseln Sie in die Entwurfsansicht, um die Tabellenstruktur zu verändern.

So erstellen Sie eine neue Tabelle in der Entwurfsansicht

Eine neue Tabelle ohne Felder erstellen Sie wie folgt (siehe Abbildung 2.11):

1. Gehen Sie zu **ERSTELLEN 1 • TABELLEN**.
2. Klicken Sie auf **TABELLENENTWURF 2**.

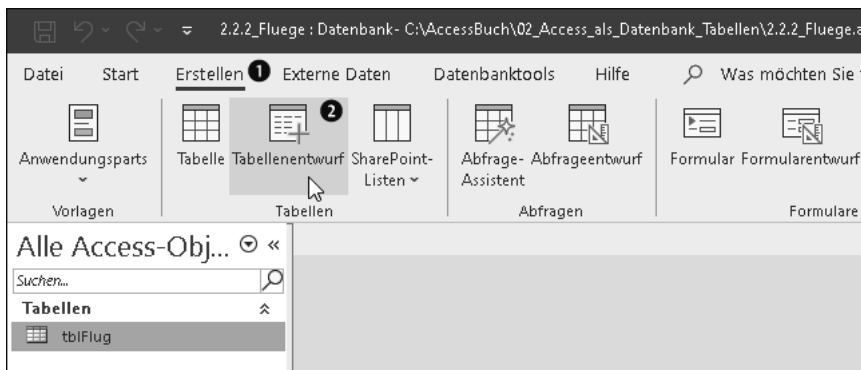


Abbildung 2.11 Eine neue Tabelle in der Entwurfsansicht erstellen

Access öffnet die leere Tabelle in der Entwurfsansicht.

So öffnen Sie eine bestehende Tabelle in der Entwurfsansicht

Wenn Sie die Struktur einer bestehenden Tabelle verändern möchten, öffnen Sie die Tabelle in der Entwurfsansicht (siehe Abbildung 2.12):

1. Gehen Sie im **Navigationsbereich 1** auf die Tabelle **2**.
2. Klicken Sie mit der rechten Maustaste.
3. Klicken Sie im Kontextmenü auf **ENTWURFSANSICHT 3** (oder drücken Sie **[Strg] + [↵]**).

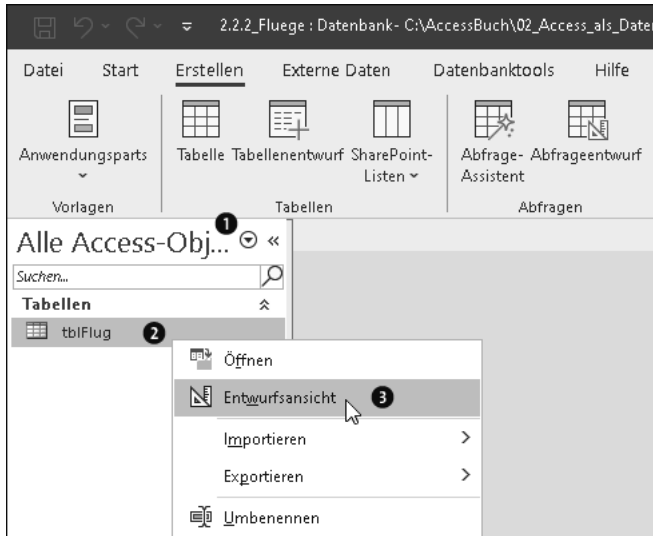


Abbildung 2.12 Eine bestehende Tabelle in der Entwurfsansicht öffnen

Sie können das einmal an der fertigen Beispieldatenbank ausprobieren, die Sie in den Materialien zum Buch unter *O2_Access_als_Datenbank_Tabellen\2.7.5_Fluege_umfassend.accdb* finden.

So schalten Sie eine geöffnete Tabelle von der Datenblattansicht in die Entwurfsansicht um

Ebenso können Sie von der Datenblattansicht ausgehend in die Entwurfsansicht umschalten:

1. Gehen Sie wie in Abbildung 2.13 zu **START** ① • **ANSICHTEN** • **ANSICHT** ②.
2. Klicken Sie auf **ENTWURFSANSICHT** ③ (oder drücken Sie **[Strg] + [.]**).



Abbildung 2.13 Aus der Datenblattansicht in die Entwurfsansicht umschalten

So speichern Sie Änderungen an der Tabellenstruktur ab

In der Entwurfsansicht können Sie Änderungen an der Tabellenstruktur durchführen. Die Entwurfsansicht (siehe Abbildung 2.14) ist der einzige Ort in Access, an dem Änderungen nicht direkt in der *.accdb*-Datei gespeichert werden. Erst wenn Sie in der Symbolleiste für den Schnellzugriff auf **SPEICHERN** klicken (oder **[Strg] + [S]** drücken), wird die neue Tabellenstruktur wirklich gespeichert.

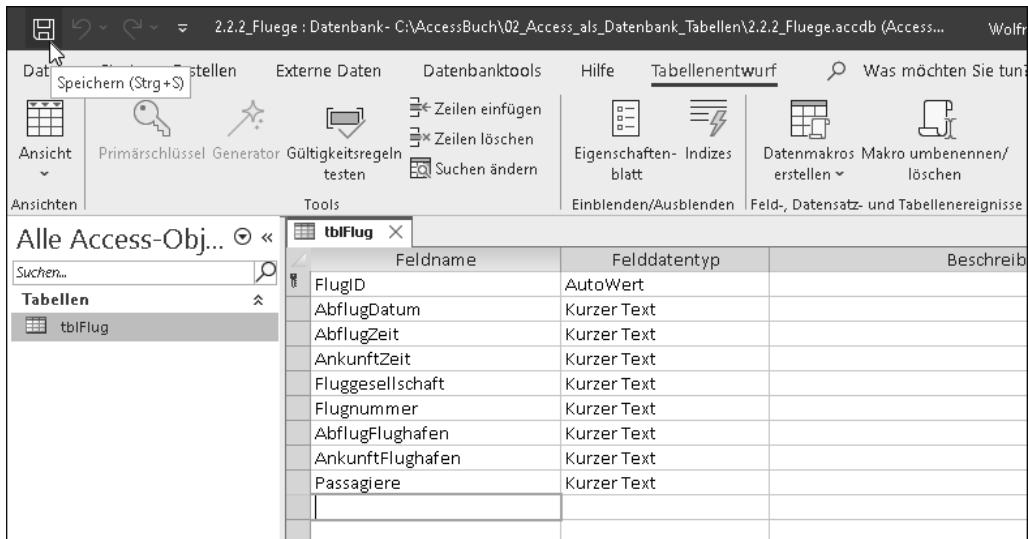


Abbildung 2.14 Speichern Sie Änderungen an der Tabellenstruktur ab, indem Sie auf das Diskettensymbol klicken.

Änderung einer Tabelle mit Datensätzen: Automatische Datenüberprüfung und -anpassung

Sollte Ihre Tabelle Datensätze enthalten, wird Access diese vor dem Speichern überprüfen. Auf dieses Thema – die automatische Überprüfung und Anpassung von Daten – werde ich in Abschnitt 2.4, »Ändern von Tabellen«, genauer eingehen. Sie wissen jetzt aber bereits, dass es so etwas wie eine automatische Datenüberprüfung und -anpassung in Access überhaupt gibt.

Am besten fangen Sie erst einmal ohne Datensätze an. Dann lässt sich die Tabellenstruktur ohne Rücksicht auf Inhalte jederzeit verändern.

Ein Feld in der Entwurfsansicht erstellen

In der Entwurfsansicht sind die Felder einer Tabelle untereinander dargestellt, und so tragen Sie ein neues Feld ein (siehe Abbildung 2.15):

1. Tippen Sie zunächst den Namen des Feldes unter FELDNAME ❶ ein.
2. Anschließend wählen Sie rechts daneben unter FELDDATENTYP ❷ den passenden Felddatentyp aus.
3. Weitere Eigenschaften zum Felddatentyp, das FORMAT, das EINGABEFORMAT und einige weitere Feldeinschränkungen und -eigenschaften können Sie im Registerblatt ALLGEMEIN ❸ einstellen.

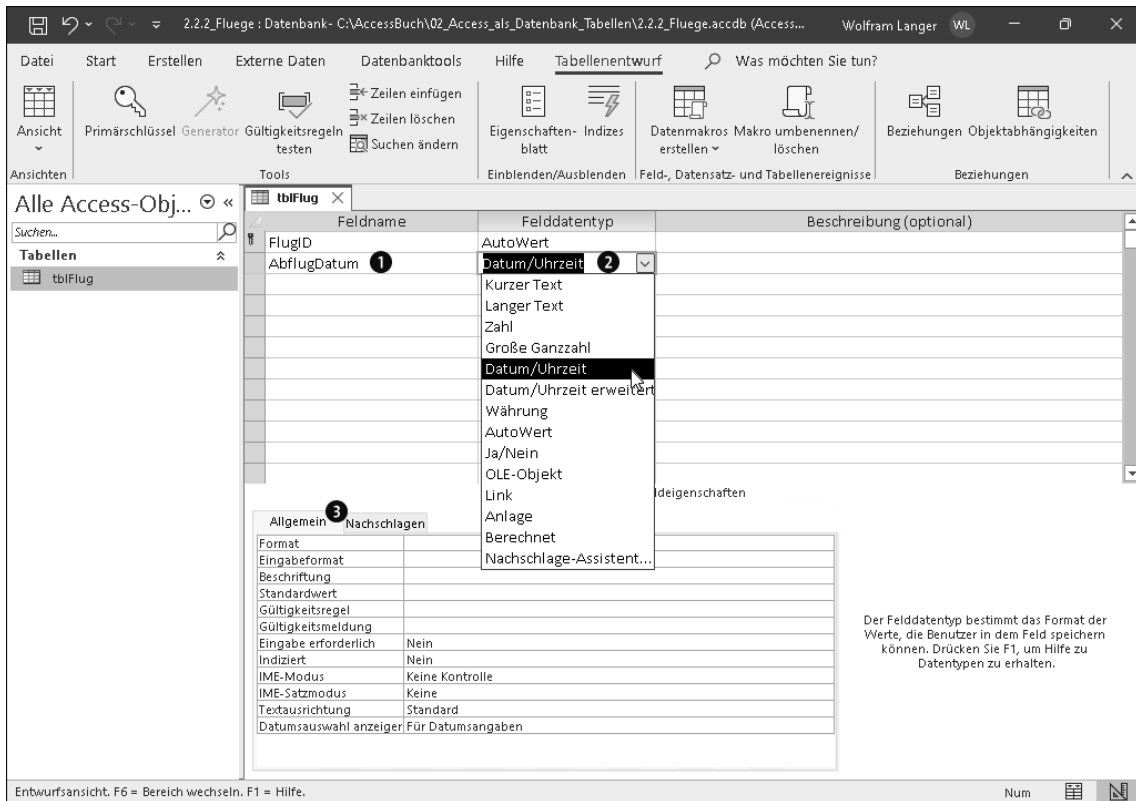


Abbildung 2.15 In der Entwurfsansicht legen Sie die Felder einer Tabelle mit ihren Felddatentypen fest.

Warum ist es wichtig, den passenden Felddatentyp zu wählen?

Mit dem Felddatentyp legen Sie fest, welche Art von Werten Access für ein Feld in der Tabelle überhaupt zulässt. Sie haben mittlerweile bestimmt gemerkt, dass ich Ordnung sehr liebe ... Genau die erreichen Sie mit Felddatentypen: Wenn Sie als Felddatentyp beispielsweise DATUM/UHRZEIT wählen, sorgt Access dafür, dass im Feld »AbflugDatum« wirklich nur ein Datum steht. Im Fachjargon sagt man, dass Access die *Datenkonsistenz* sicherstellt.

Es gibt aber weitere Gründe, warum es lohnt, nach dem passenden Felddatentyp zu suchen:

- ▶ Mit Felddatentypen können Sie Speicherplatz sparen.
- ▶ Sie können Inhalte richtig filtern und sortieren.
- ▶ Sie können Inhalte unterschiedlich formatieren.

Speicherplatz ist bei den heutigen Festplattenkapazitäten und -preisen eher selten ein Problem. Bei Datenbanken dürfen Sie aber nicht vergessen: Die Masse (d. h. die große Anzahl der Datensätze) macht's!

Das *Filtern und Sortieren* funktioniert überhaupt nur richtig, wenn Sie die passenden Felddatentypen verwenden. Sie kennen das: Es ist ein großer Unterschied, ob Sie Werte als Text, als Zahl oder als Datum sortieren.

Unterschiedliche *Formatierungen* sind besonders bei Datum und Uhrzeit interessant.

Alle drei Punkte sind also sehr wichtig. Daher empfehle ich Ihnen:

- ▶ Machen Sie sich mit den einzelnen Felddatentypen vertraut.
- ▶ Nehmen Sie sich Zeit für die Suche nach dem passenden Felddatentyp.
- ▶ Verwenden Sie denjenigen Felddatentyp, der die Realität bestmöglich abbildet.

In den nächsten Abschnitten werde ich Ihnen die einzelnen Felddatentypen ausführlich vorstellen. Auch die anderen Einstellungen im Registerblatt ALLGEMEIN werde ich Ihnen im Detail erläutern.

Wenn Sie nach dem Abspeichern der Tabellenstruktur in die Datenblattansicht umschalten, werden alle Felder als Spalten dargestellt. Access dreht also die Anordnung der Felder zwischen beiden Ansichten um 90°. Um Verwirrung zu vermeiden, werde ich mich bemühen, nicht von Zeilen oder Spalten, sondern nur von *Datensätzen* und *Feldern* zu sprechen.

Änderungen der Tabellenstruktur in der Datenblattansicht

Seit der Version Access 2003 können Sie die Tabellenstruktur auch in der Datenblattansicht ändern (siehe Abbildung 2.16). Diese Möglichkeit hatte ich Ihnen bereits eingangs in Abschnitt 2.1.2, »Eine Tabelle in Access erstellen«, gezeigt.

Allerdings offenbart das Kontextmenü nicht die ganze Wahrheit. Den vollen Zugriff auf alle Optionen finden Sie nur in der Entwurfsansicht. Genau deshalb empfehle ich Ihnen, die Tabellenstruktur *ausschließlich* in der Entwurfsansicht zu ändern.

Für die aktuelle Datenbank können Sie die Funktion zum Verändern der Tabellenstruktur in der Datenblattansicht deaktivieren. Gehen Sie dazu auf DATEI • OPTIONEN, und entfernen Sie dann im Fenster ACCESS-OPTIONEN den Haken unter AKTUELLE DATENBANK • ANWENDUNGSOPTIONEN • ENTWURFSÄNDERUNGEN FÜR TABELLEN IN DER DATENBLATTANSICHT AKTIVIEREN. Diese Änderung wirkt sich allerdings erst aus, nachdem Sie Ihre Datenbank geschlossen und wieder geöffnet haben.

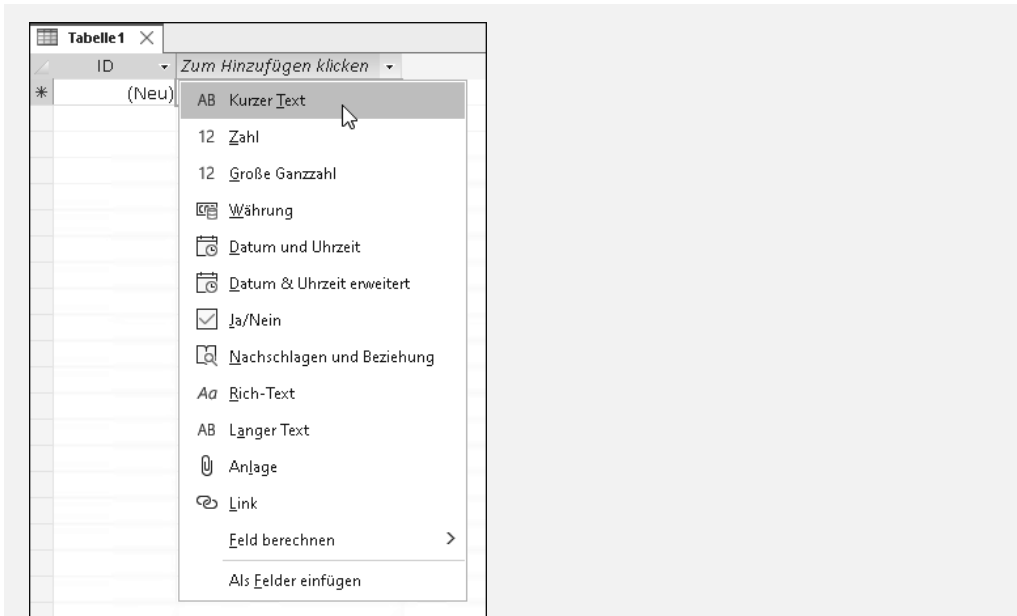


Abbildung 2.16 »Zum Hinzufügen klicken« – über dieses Menü erweitern Sie die Tabellenstruktur in der Datenblattansicht.

Es ist jetzt an der Zeit, dass wir uns die einzelnen Felddatentypen ansehen. Anschließend zeige ich Ihnen weitere Regeln und Einschränkungen, die Sie ebenfalls in der Entwurfsansicht festlegen können.

2.3.2 Der Felddatentyp »Kurzer Text«

Verwenden Sie den Felddatentyp KURZER TEXT, um jegliche Art von *Zeichenfolgen* (engl. *strings*) zu speichern. Im Registerblatt ALLGEMEIN unter FELDGRÖSSE können (und sollten!) Sie die maximal zulässige Anzahl von Zeichen für ein Feld festlegen. Wenn Sie eine große Anzahl von Zeichen zulassen möchten, denken Sie bitte daran, dass die Textfelder in Formularen und Berichten auch breit genug sein müssen. Ein Feld mit dem Felddatentyp KURZER TEXT darf maximal 255 Zeichen enthalten.

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
KURZER TEXT	<i>Short Text</i>	CHAR VARCHAR	maximal 255 Zeichen	<ul style="list-style-type: none"> ▶ 2 Bytes je Zeichen ▶ mit aktivierter Unicode-Kompression auch weniger

Tabelle 2.6 Der Felddatentyp »Kurzer Text« eignet sich für Zeichenfolgen bis zu 255 Zeichen.

ID	FeldKurzerText
1	Video bietet eine leistungsstarke Möglich
2	Wenn Sie auf "Onlinevideo" klicken, könn
3	Кириллица
4	עברית
5	日本
6	Jeg købte hot wings med pommes frites o

Abbildung 2.17 Exotische Zeichen aus dem Unicode-Bereich sind für Access kein Problem.

Unicode-Zeichen

Alle Zeichenfolgen speichert Access in *Unicode* ab. Sie können daher mehr als nur die 127 ASCII-Zeichen verwenden. Wichtig wird Unicode, wenn Sie eine andere Schrift statt der lateinischen verwenden möchten, beispielsweise Kyrrillisch, Hebräisch oder Schriften aus Fernost, oder für spezielle Sonderzeichen. All dies unterstützt Access; Sie können also auch exotische Zeichen und Wörter in einer Access-Datenbank speichern.

Ein anderes Thema ist, ob alle Zeichen am Bildschirm richtig angezeigt werden. Nur die wenigsten Schriftarten unterstützen sämtliche Unicode-Zeichen. Im Office-Paket enthalten ist die Schriftart »Arial Unicode MS« mit knapp 40.000 Zeichen (und einer Größe von fast 22 MB); reicht Ihnen dieser Umfang fürs Erste? Die Anzeigeschriftart können Sie ändern, indem Sie die Tabelle in der Datenblattansicht öffnen und dann unter **START • TEXTFORMATIERUNG** die gewünschte Schriftart auswählen.

2.3.3 Der Felddatentyp »Langer Text«

Wenn Ihnen 255 Zeichen zu wenig sind, können Sie auf den Felddatentyp **LANGER TEXT** ausweichen. Ich sage »ausweichen« deshalb, weil Access nicht alle Funktionen für den Felddatentyp **LANGER TEXT** unterstützt. An diese Einschränkungen sollten Sie sich immer erinnern, wenn Sie den Felddatentyp **LANGER TEXT** verwenden:

- ▶ Die maximal zulässige Anzahl von Zeichen können Sie nicht festlegen.
- ▶ Sortieren in der Datenblattansicht ist gar nicht möglich.
- ▶ Beim Sortieren per Abfrage werden nur die ersten 255 Zeichen berücksichtigt.
- ▶ Beim Filtern werden nur die ersten 255 Zeichen berücksichtigt.

Diese Einschränkungen können zu bösen Überraschungen führen. Dafür können Sie aber wirklich jede Menge Zeichen speichern: bis zu 65.535 Zeichen über die Benutzeroberfläche von Access, per Programmierung sogar bis zu 2 GB!

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
LANGER TEXT	<i>Long Text</i>	TEXT	maximal 2 GB (über die Benutzeroberfläche von Access: maximal 65.535 Zeichen)	<ul style="list-style-type: none"> ▶ 2 Bytes je Zeichen ▶ mit aktivierter Unicode-Kompression auch weniger ▶ Die Unicode-Kompression greift nur, wenn der Inhalt größer als 4 kB ist.

Tabelle 2.7 Platz für Zeichen ohne Ende: der Felddatentyp »Langer Text«

»Langer Text« und »Memo«

Vor der Access Version 2013 hieß dieser Felddatentyp *Memo*, während *Kurzer Text* einfach nur als *Text* bezeichnet wurde. Ich finde es nur zu ärgerlich (und für den Neueinsteiger leider unnötig schwierig), dass die Bezeichnungen in der Sprache SQL wieder anders sind, nämlich CHAR für kurzen Text sowie TEXT für langen Text.

Der Felddatentyp LANGER TEXT bietet noch zwei Extras, die Sie im Registerblatt ALLGEMEIN aktivieren können:

1. **Formatierter Text:** Felder mit dem Felddatentyp KURZER TEXT werden immer ohne Formatierungen abgespeichert. Üblicherweise ist das auch für Felder mit dem Felddatentyp LANGER TEXT so.

Wenn Sie das TEXTFORMAT auf RICH-TEXT umstellen, können Sie Formatierungen einsetzen (siehe Abbildung 2.18). Access speichert die Formatierungen automatisch als HTML-Code ab – nicht jedoch im Rich Text Format (RTF), wie der Name vermuten lässt.

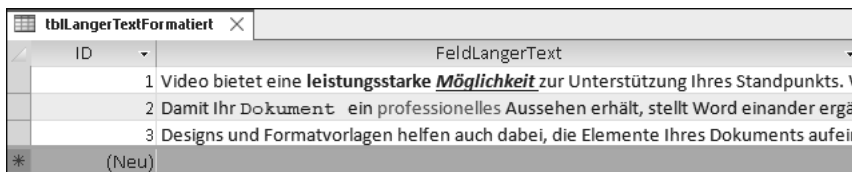


Abbildung 2.18 In einem Feld mit dem Felddatentyp »Langer Text« können Sie Formatierungen aktivieren.

2. **Protokollierung:** Wenn Sie die Option NUR ANFÜGEN auf JA setzen, protokolliert Access alle Änderungen an einem Feld mit dem Felddatentyp LANGER TEXT. Sie können sich die alten Versionen anzeigen lassen (siehe Abbildung 2.19), indem Sie in der Datenblatt-

ansicht mit der rechten Maustaste auf das Feld klicken und anschließend im Kontextmenü SPALTENVERLAUF ANZEIGEN ... auswählen.

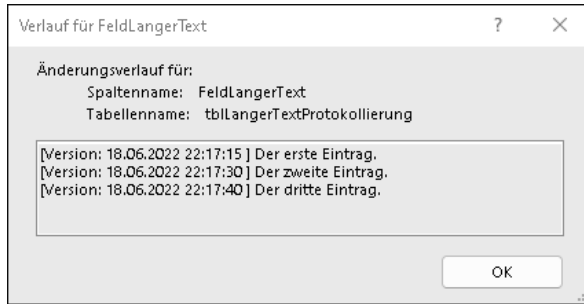


Abbildung 2.19 Nach aktivierter Protokollierung können Sie die alten Versionen des Feldinhalts aufrufen.

Protokolliert werden allerdings nur die Änderungen an dem einen Feld, nicht jedoch die unterschiedlichen Versionen des gesamten Datensatzes (für diesen Zweck können Sie die *Änderungentabelle* verwenden, die ich Ihnen in Abschnitt 2.7.6, »Typische Muster von Tabellenbeziehungen«, vorstellen werde).

2.3.4 Der Felddatentyp »Zahl«

Zum Speichern von Zahlen bietet Access eine ganze Reihe von Felddatentypen an, die entweder unter dem Felddatentyp ZAHL oder unter GROSSE GANZZAHL zu finden sind. Und beim Felddatentyp ZAHL legen Sie den eigentlichen Felddatentyp im Registerblatt ALLGEMEIN unter FELDGRÖSSE fest. Warum sind große Ganzzahlen separat und nicht beim Felddatentyp ZAHL aufgeführt, wie es eigentlich zu erwarten wäre? Das hat historische Gründe, doch dazu gleich mehr. Zunächst ein paar Grundlagen: Zahlen werden binär abgespeichert.

Binäre Speicherung von Zahlen

Normalerweise brauchen Sie sich keine Gedanken darüber zu machen, wie Access die Zahlen binär ablegt. Für die Ganzzahlen habe ich Ihnen trotzdem einmal grafisch dargestellt, wie eine Zahl intern in binärer Form gespeichert wird (siehe Abbildung 2.20). Diese Informationen benötigen Sie, wenn Sie mit VBA-Programmierung auf die einzelnen Bits zugreifen möchten. Ein Beispielprogramm finden Sie in den Materialien zum Buch unter *02_Access_als_Datenbank_Tabellen\2.3.4_BinaerManipulation.accdb*.

1. Zu der Gruppe der *Ganzzahlen* (engl. *integer number*) gehören die Felddatentypen BYTE, INTEGER und LONG INTEGER. Der Felddatentyp BYTE kann nur positive Ganzzahlen aufnehmen und umfasst – wie der Name schon sagt – den Speicherbereich von genau einem Byte.

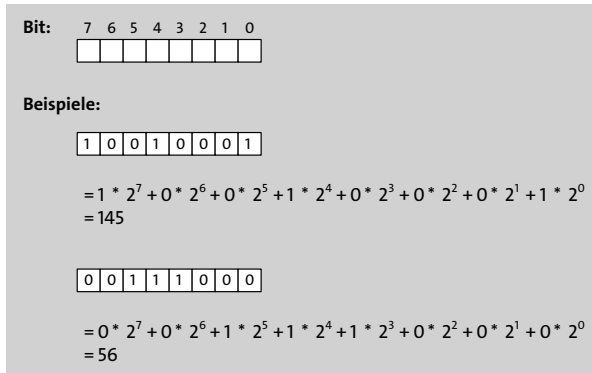


Abbildung 2.20 Ganzzahlen des Felddatentyps »Byte« werden in einem Byte (mit 8 Bits) ohne Vorzeichen gespeichert.

INTEGER und LONG INTEGER umfassen zwei bzw. vier Byte. Beide dieser Felddatentypen nehmen Zahlen mit Vorzeichen auf. Das Vorzeichen wird immer im höchsten Bit gespeichert (siehe Abbildung 2.21).

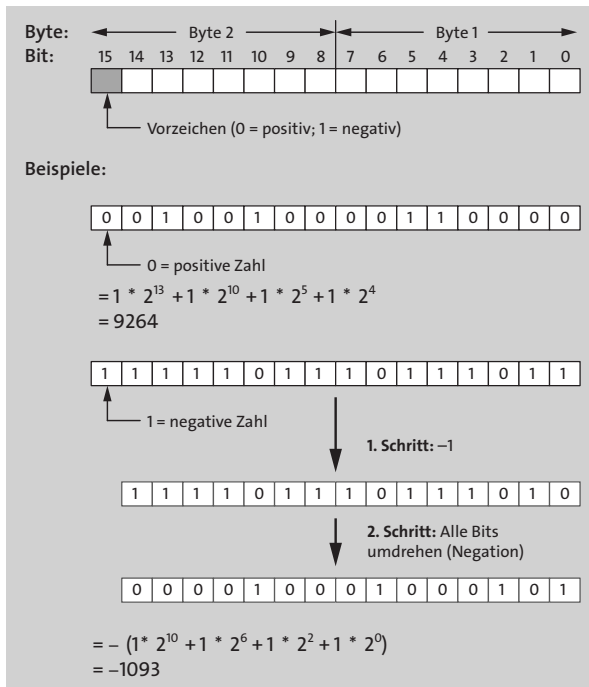


Abbildung 2.21 Ganzzahlen des Felddatentyps »Integer« werden in zwei Bytes mit Vorzeichen gespeichert. Um eine negative Zahl von der Binärdarstellung in die Dezimaldarstellung umzurechnen, sind zwei Schritte notwendig (negative Zahlen werden in der sogenannten Two's-Complement-Repräsentation gespeichert). Ganzzahlen des Felddatentyps »Long Integer« werden in vier Bytes gespeichert, alles andere ist identisch mit dem Felddatentyp »Integer«.

Mit dem Felddatentyp REPLIKATIONS-ID können Sie *GUIDs* speichern. Eine GUID ist eine 16 Byte große Ganzzahl, die in einem besonderen Format angezeigt wird.

Globally Unique Identifier (GUID)

GUIDs sind zunächst nichts anderes als positive Ganzzahlen, die sehr groß werden dürfen (Wertebereich: 0 ... 340.282.366.920.938.463.463.374.607.431.768.211.455). Praktischerweise werden GUIDs als Hexadezimalzahlen (siehe nächster Kasten) dargestellt.

Beispiele:

```
{9D5759B3-0EBF-4957-837D-09B1E330E998}
{C57DC5D5-B647-4966-87A6-C585264AC85C}
{C5C64BD8-A75C-48DA-98D2-FAA00F13A962}
```

Durch den großen Wertebereich ist es möglich, dass (nach festgelegten Algorithmen) generierte GUIDs praktisch weltweit eindeutig sind. Auch in Access können Sie GUIDs automatisch generieren lassen (mehr dazu folgt in Abschnitt 2.3.5, »Der Felddatentyp ›AutoWert«). In einer Datenbank sind GUIDs als Primärschlüssel interessant, wenn Datensätze an getrennten Standorten ohne Netzwerkverbindung erfasst und später zusammengetragen werden sollen. Solche *Replikationsszenarien* gehören zu den anspruchsvollsten Aufgaben der Datenbankprogrammierung.

Das Hexadezimalsystem

Das Hexadezimalsystem ist das Zahlensystem mit der Basis 16 und den 16 Ziffern 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Zur Unterscheidung von Zahlen im Dezimalsystem wird als Präfix &H vorangestellt (gelegentlich werden auch andere Präfixe verwendet, z. B. 0x).

Beispiele:

```
&HC94B = 12 * 16^3 + 9 * 16^2 + 4 * 16^1 + 11 * 16^0 = 51531
&H16 = 1 * 16^1 + 6 * 16^0 = 22
&H0 = 0 * 16^0 = 0
```

Die Hexadezimaldarstellung ist in der Informatik besonders verbreitet, da zwei Stellen genau einem Byte (8 Bit) entsprechen.

2. **GROSSE GANZZAHLEN** sind als eigener Felddatentyp aufgeführt, obwohl sie im Grunde genommen genauso wie die anderen Ganzzahlen abgespeichert werden. Dies ist der Felddatentyp für Ganzzahlen der Größe 64 Bit, was 8 Bytes entspricht. Diesen Felddatentyp gibt es erst seit der Access Version 2016, und das dürfte der einzige Grund dafür sein, warum große Ganzzahlen nicht unterhalb des Felddatentyps ZAHL aufgelistet sind.

Sobald Sie den Felddatentyp GROSSE GANZZAHL das erste Mal in seiner Datenbank nutzen möchten, zeigt Access eine Warnung an (siehe Abbildung 2.22). Denn eine solche Datenbank kann nur mit Access 2016 oder höher geöffnet werden.

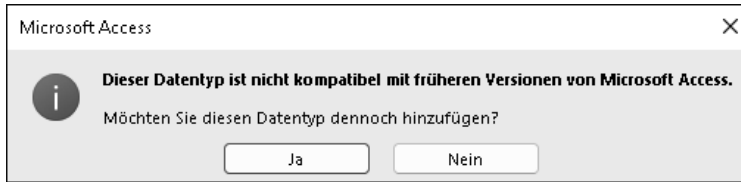


Abbildung 2.22 Datenbanken mit dem Felddatentyp »Große Ganzzahl« können nur mit Access 2016 oder höher geöffnet werden.

Sollten Sie eine Version älter als Access 2016 nutzen, können Sie unbesorgt sein: Abgesehen von einer Beispieldatenbank unter *O2_Access_als_Datenbank_Tabellen\2.3.4_Große_Ganzzahl.accdb* werde ich in diesem Buch den Felddatentyp GROSSE GANZZAHL nicht verwenden, sodass Sie alle Datenbanken in den Materialien zum Buch problemlos nutzen können.

Version der Access-Datenbank

Falls Sie es genau wissen wollen: Es gibt unterschiedliche Versionen einer Access-Datenbank. Einige Anhaltspunkte dazu finden Sie in der Benutzeroberfläche, wenn Sie eine Datenbank neu erstellen (siehe Abbildung 2.2, Auswahl des DATEITYPs) oder unter einem anderen Format abspeichern möchten (DATEI • SPEICHERN UNTER):

- ▶ das alte *.mdb*-Format (Hierzu gibt es zwei Versionen, sowohl Access 2000 als auch Access 2002 – 2003.)
- ▶ das neue *.accdb*-Format (Es wird mit Access 2007 – 2016 bezeichnet.)

Mit den großen Ganzzahlen kommt für das *.accdb*-Format noch eine weitere Version hinzu. Es wird aktiviert, nachdem Sie die Warnmeldung aus Abbildung 2.22 mit JA bestätigt haben.

Prüfen können Sie die Version der aktuellen Datenbank mit diesem VBA-Befehl in der Direkt-eingabe (`[Strg] + [G]`):

```
Debug.Print CurrentDb().Version
```

Es wäre nur logisch, wenn das vierte Format in der Benutzeroberfläche neben den anderen drei oben genannten erscheinen würde. Leider ist Microsoft einen anderen Weg gegangen.

Zu erwähnen ist noch ein weiterer Weg, der ebenfalls zum neuesten Format der Datenbank führt: Rufen Sie die Access-Optionen unter DATEI • OPTIONEN auf, und setzen Sie dann eines der Häkchen bei AKTUELLE DATENBANK • DATENTYPUNTERSTÜTZUNGS-OPTION. Allerdings führt der eigentlich harmlose Befehl DATENBANKTOOLS • DATENBANK KOMPRIMIEREN UND REPARIEREN wieder zur vorherigen Version. Mich würde es nicht wundern, wenn es noch die eine oder andere Unzulänglichkeit mit dem neuen Felddatentyp GROSSE GANZZAHL und der neuen Version des Datenbank-Formats gibt.

3. *Gleitkommazahlen* (engl. *floating-point numbers*) sind Zahlen mit Nachkommastellen und Exponent. In Access können Sie sich entweder für den Felddatentyp SINGLE oder für DOU-

BLE entscheiden. Beide Felddatentypen unterscheiden sich in ihrer Genauigkeit, d. h. in der Anzahl der Ziffern, die gespeichert werden.

Es gibt bei Gleitkommazahlen aber einen großen Haken: Nicht jede Zahl kann im Binärsystem exakt dargestellt werden. Selbst eine Zahl wie z. B. 0,1 (im Dezimalsystem exakt darstellbar) entspricht im Binärsystem dem Wert 0,0001100110011... Diese Zahl ist mit begrenzt vielen Ziffern nicht exakt zu speichern, denn sie hat eine nicht abbrechende Periode hinter dem Komma. Beim Umgang mit Gleitkommazahlen kann es also zu *Artefakten* und dadurch zu Überraschungen kommen!

Artefakte beim Umgang mit der Gleitkommaarithmetik

Gleitkommazahlen und Gleitkommaarithmetik sind ein eigenes Thema für sich, das über den Umfang dieses Buches hinausgeht. Die folgenden Artefakte können Ihnen beispielsweise beim Umgang mit Gleitkommazahlen begegnen:

- ▶ $0,1 - 1/10 = 5.5E-18$ (statt 0)
- ▶ $0,0001234567890123 + 1 = 1,0001234567890100$ (statt 1,0001234567890123)
- ▶ $1E20 + 1 = 1E20$ (statt 100000000000000000001)

Der erste Fall kommt dadurch zustande, dass unterschiedliche Genauigkeiten (SINGLE und DOUBLE) gemischt werden. Wenn in einer Rechnung gleichzeitig sehr große und sehr kleine Zahlen auftreten, kommt es zu den letzten beiden Artefakten. Diese Artefakte treten übrigens auch in Excel auf.

Am besten eignen sich Gleitkommazahlen für Messwerte oder wissenschaftliche Zahlen, die mit einem Messfehler behaftet sind. Sie können Gleitkommazahlen auch für andere Zwecke nutzen, beispielsweise um einen Prozentwert (30 % = 0,30) in einer Tabelle abzuspeichern. Nur dürfen Sie nie vergessen, dass es bei nachfolgenden Berechnungen zu Artefakten kommen kann. Achten Sie besonders darauf, Rechenergebnisse nicht auf exakte Gleichheit zu überprüfen. Noch einmal im Klartext: Verwenden Sie nicht das Gleichheitszeichen (=)!

In manchen Fällen sind Artefakte wirklich unverzeihlich, insbesondere dann, wenn es um das liebe Geld geht. Verwenden Sie für Geldbeträge *niemals* den Felddatentyp SINGLE oder DOUBLE! Für Geldbeträge gibt es in Access den Datentyp WÄHRUNG, den ich Ihnen in Kürze vorstellen werde; und nur dort gehören Geldbeträge hinein.

4. Neben den Gleitkommazahlen gibt es in Access einen Datentyp für *exakte Dezimalzahlen* (*Festkommazahlen*, engl. *fixed-point numbers*), den Felddatentyp DEZIMAL. Im Registerblatt ALLGEMEIN können Sie unter GENAUIGKEIT die Anzahl der Stellen angeben (maximal 28). Unter DEZIMALSTELLEN legen Sie fest, wie viele der Stellen hinter dem Komma stehen (maximal der Wert von GENAUIGKEIT). Im Gegensatz zu Gleitkommazahlen bleiben die Stellen hinter dem Komma exakt so, wie Sie es festgelegt haben, auch bei Berechnungen. Sie können unter DEZIMALSTELLEN auch den Wert 0 eintragen und auf diese

Weise sehr große Ganzzahlen speichern (größere als mit dem Felddatentyp LONG INTEGER).

Der Felddatentyp DEZIMAL wird vor allem in Server-Datenbanken verwendet (beispielsweise in Microsoft SQL Server oder Oracle Database; dort aber unter einem anderen Namen als in Access, nämlich »Decimal« bzw. »Number«). Der DEZIMAL-Felddatentyp der Server-Datenbanken kann noch einiges mehr, unter anderem können Sie bis zu 38 Stellen speichern.

In Access wurde dieser Felddatentyp erst relativ spät mit der Version Access 2000 eingeführt. Am Anfang gab es in Access eine Reihe von Fehlern und anderen Unzulänglichkeiten. Dazu gehört, dass es bis heute keinen entsprechenden DEZIMAL-Datentyp in Visual Basic for Applications (VBA) gibt. Wegen dieser schlechten Erfahrungen wird der Felddatentyp DEZIMAL in Access immer noch selten eingesetzt. Vielleicht liegt das auch daran, dass der Wertebereich (nicht jedoch die Genauigkeit!) der Gleitkommazahlen so verlockend aussieht. Mein Eindruck ist, dass die meisten Kinderkrankheiten des Felddatentyps DEZIMAL mittlerweile behoben sind. Für exakte Dezimalzahlen bevorzuge ich daher den Felddatentyp DEZIMAL gegenüber SINGLE oder DOUBLE.

Als Zusammenfassung fasst Tabelle 2.8 noch einmal alle Felddatentypen zusammen, die Sie für Zahlen einsetzen können:

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
BYTE	<i>Byte</i>	TINYINT	0 ... 255 &H0 ... &HFF	1 Byte
INTEGER	<i>Integer</i>	SMALLINT	-32.768 ... +32.767 -&H8000 ... +&H7FFF	2 Byte
LONG INTEGER	<i>Long Integer</i>	INTEGER	-2.147.483.648 ... + 2.147.483.647 -&H80000000 ... +&H7FFFFFFF	4 Byte
REPLIKATIONS-ID	<i>Replication ID</i>	UNIQUE-IDENTIFIER	&H0 ... &HFFFFFFFFFFFFFFF FFFFFFFFFFFFFFF	16 Byte

Tabelle 2.8 Bei den Felddatentypen für Zahlen sind Ganzzahlen (»Byte«, »Integer«, »Long Integer«, »Replikations-ID«), Gleitkommazahlen (»Single«, »Double«) und Festkommazahlen (»Dezimal«) zu unterscheiden. Bei den Ganzzahlen habe ich den Wertebereich auch in Hexadezimalzahlen angegeben. Sehr wichtig für die Gleitkommazahlen (»Single« und »Double«) ist die unterschiedliche Genauigkeit (7 Stellen bzw. 15 Stellen). Trotzdem kann mit Gleitkommazahlen nicht jede rationale Zahl exakt abgebildet werden. Mathematisch ausgedrückt: Es gibt Lücken im Wertebereich.

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
GROSSE GANZZAHL	<i>Large Number</i>	-	-9.223.372.036.854.775.808 ... +9.223.372.036.854.775.807 -&H8000000000000000 ... +&H7FFFFFFFFFFFFFFF	8 Byte
SINGLE	<i>Single</i>	REAL	-3,402823E38 ... -1,401298E-45 0 +1,401298E-45 ... +3,402823E38	4 Byte
DOUBLE	<i>Double</i>	FLOAT	-1,79769313486232E308 ... -4,94065645841247E-324 0 +4,94065645841247E-324 ... +1,79769313486232E308	8 Byte
DEZIMAL	<i>Decimal</i>	DECIMAL	-1E-28 ... +1E+28	17 Byte

Tabelle 2.8 Bei den Felddatentypen für Zahlen sind Ganzzahlen («Byte«, »Integer«, »Long Integer«, »Replikations-ID«), Gleitkommazahlen («Single«, »Double«) und Festkommazahlen («Dezimal«) zu unterscheiden. Bei den Ganzzahlen habe ich den Wertebereich auch in Hexadezimalzahlen angegeben. Sehr wichtig für die Gleitkommazahlen («Single« und »Double«) ist die unterschiedliche Genauigkeit (7 Stellen bzw. 15 Stellen). Trotzdem kann mit Gleitkommazahlen nicht jede rationale Zahl exakt abgebildet werden. Mathematisch ausgedrückt: Es gibt Lücken im Wertebereich. (Forts.)

2.3.5 Der Felddatentyp »AutoWert«

Beim Felddatentyp AUTOWERT vergibt Access die Werte – wie der Name schon sagt – automatisch. Ein AUTOWERT eignet sich daher hervorragend für den Primärschlüssel. Unter ALLGEMEIN • FELDGRÖSSE können Sie zwischen LONG INTEGER und REPLIKATIONS-ID wählen. Sie kennen diese beiden Felddatentypen bereits aus dem vorhergehenden Abschnitt, denn eigentlich ist der AUTOWERT nur eine besondere Form des Felddatentyps ZAHL. Der einzige Unterschied zum Felddatentyp ZAHL ist, dass Access die Werte automatisch vergibt. In ein Feld mit dem Felddatentyp AUTOWERT können Sie also weder einen Wert selbst eintragen noch können Sie einen von Access vergebenen Wert ändern.

1. Für LONG INTEGER können Sie unter ALLGEMEIN • NEUE WERTE auswählen, ob Access aufsteigende Werte beginnend mit der Zahl 1 (INKREMENT) oder zufällige Zahlen erzeugt (ZUFALL).

Den Zähler für »AutoWert«-Felder zurücksetzen

Übrigens setzt Access den Zähler für INKREMENT nicht zurück, wenn Sie Datensätze löschen. Wenn Sie alle Datensätze einer Tabelle löschen und dann wieder mit dem Wert 1 beginnen möchten, klicken Sie auf DATENBANKTOOLS • DATENBANK KOMPRIMIEREN UND REPARIEREN. Dadurch werden die Zähler aller Tabellen zurückgesetzt.

2. Im Fall der REPLIKATIONS-ID erzeugt Access GUIDs (siehe den Kasten »Globally Unique Identifier (GUID)« weiter oben), die mit an Sicherheit grenzender Wahrscheinlichkeit weltweit eindeutig sind.

In Abschnitt 2.7, »Beziehungen zwischen Tabellen«, werde ich Ihnen *Fremdschlüssel* vorstellen, mit denen Sie zwei Tabellen verbinden können. Ohne viel vorgreifen zu wollen, möchte ich Sie jetzt schon darauf hinweisen, dass die Feldgröße von Primär- und Fremdschlüssel identisch sein muss. Am besten entscheiden Sie sich bei allen ID-Feldern einer Datenbank entweder für LONG INTEGER oder für REPLIKATIONS-ID. Ich empfehle Ihnen LONG INTEGER, denn spätestens bei der Programmierung ist dies viel angenehmer als der Umgang mit den großen GUIDs (Felddatentyp REPLIKATIONS-ID).

Es gibt keinen AutoWert für den neuen Felddatentyp »Große Ganzzahl«

Sie möchten bei ID-Feldern den neuen Felddatentyp GROSSE GANZZAHL nutzen? Dieser Felddatentyp ist erst mit Access 2016 hinzugekommen. Bei neuen Features bin ich immer etwas vorsichtig. Nach dem, was ich bisher gesehen habe, sollte es funktionieren. Allerdings gibt es keinen AUTOWERT für diesen Felddatentyp, sodass in der Praxis doch nur LONG INTEGER (oder REPLIKATIONS-ID) bleibt.

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
AUWERT LONG INTEGER	<i>AutoNumber</i> <i>Long Integer</i>	COUNTER	-2.147.483.648 ... + 2.147.483.647 -&H80000000 ... +&H7FFFFFFF	4 Byte
AUWERT REPLIKA- TIONS-ID	<i>AutoNumber</i> <i>Replication</i> <i>ID</i>	–	&H0 ... &HFFFFFFFFFFFFFFF FFFFFFFFFFFFFFF	16 Byte

Tabelle 2.9 Die beiden Felddatentypen der Gruppe »AutoWert« sind identisch mit den beiden Felddatentypen »Long Integer« und »Replikations-ID«. Einziger Unterschied: Access vergibt die Werte automatisch.

Ein AUWERT-Feld ist *nicht* automatisch der Primärschlüssel. Dazu sind noch zwei weitere Handgriffe notwendig:

1. Wählen Sie das AUTOWERT-Feld in der Entwurfsansicht aus.
2. Klicken Sie auf TABELLENENTWURF • TOOLS • PRIMÄRSCHLÜSSEL.

Feldname	Felddatentyp
FlugID	AutoWert
AbflugDatum	Kurzer Text
AbflugZeit	Kurzer Text
AnkunftZeit	Kurzer Text
Fluggesellschaft	Kurzer Text
Flugnummer	Kurzer Text
AbflugFlughafen	Kurzer Text
AnkunftFlughafen	Kurzer Text
Passagiere	Kurzer Text

Abbildung 2.23 Der Primärschlüssel wird durch ein kleines Schlüsselssymbol vor dem Feldnamen gekennzeichnet.

2.3.6 Der Felddatentyp »Datum/Uhrzeit« und »Datum/Uhrzeit erweitert«

Für alles, was irgendwie mit Zeitangaben zu tun hat, ist der Felddatentyp DATUM/UHRZEIT vorgesehen. Mit Access 2021 ist der Felddatentyp DATUM/UHRZEIT ERWEITERT hinzugekommen, den Sie nur mit großer Vorsicht verwenden sollten. Ich werde erst am Schluss dieses Abschnitts auf ihn eingehen und erst einmal den traditionellen Felddatentyp DATUM/UHRZEIT erläutern.

Intern speichert Access einen Zeitpunkt als Zahl ab, nämlich als Anzahl der Tage, die seit dem 30.12.1899 um 0:00 Uhr vergangen sind. Wie geht dann aus der Zahl die Uhrzeit hervor? Genauer gesagt speichert Access ein Datum als Gleitkommazahl vom Felddatentyp DOUBLE ab. Die Stellen vor dem Komma ergeben den Tag, während mit den Nachkommastellen die Zeit berechnet werden kann. In Tabelle 2.10 sehen Sie ein paar Beispiele.

In der Praxis werden Sie nicht den internen DOUBLE-Wert, sondern nur die Anzeige als Datum und Uhrzeit zu Gesicht bekommen. In Abschnitt 2.6, »Formatierungen in Tabellen«, werde ich Ihnen ausführlich zeigen, wie Sie das Anzeigeformat anpassen können.

»Double«-Wert	Tage seit dem 30.12.1899	Datum	Zeitangabe	Uhrzeit
1,0	1	31.12.1899	,0	00:00:00
22194,5	22194	05.10.1960	,5	12:00:00
41180,6875	41180	28.09.2012	,6875	16:30:00
-54655,2988425926	-54655	10.05.1750	,2988425926	07:10:20

Tabelle 2.10 Intern speichert Access Zeitangaben als Zahlen des Felddatentyps »Double« ab. Die Festlegung auf das Bezugsdatum 30.12.1899 stammt noch aus den Urzeiten von Microsoft Excel und Lotus 1-2-3.

Da Sie jetzt wissen, dass Access ein Datum mit Uhrzeit intern als Zahlenwert speichert, können Sie dies ausnutzen zum:

1. Sortieren nach Datum
2. Vergleichen von Zeitangaben
3. Filtern von Zeitangaben

Epoche, Wertebereich und Genauigkeit von Zeitangaben

Es ist in der Informatik weit verbreitet, Zeitangaben als Zahlenwerte zu speichern. Die Form ist jedoch nicht einheitlich und unterscheidet sich von Programm zu Programm. Mit »Form« meine ich hier zum einen den Zeitpunkt 0, den man *Epoche* nennt. Zum anderen sind das der Wertebereich und die Auflösung. Für Access gilt Folgendes für die Form:

- ▶ Die Epoche ist der 30.12.1899 um 0:00 Uhr.
- ▶ Das kleinste zulässige Datum ist der 01.01.0100 (das Jahr 100 n. Chr.).
- ▶ Das größte zulässige Datum ist der 31.12.9999 (das Jahr 9999 n. Chr.).
- ▶ Die Auflösung beträgt 1 s, das heißt, Zeitangaben werden auf die Sekunde genau gespeichert.

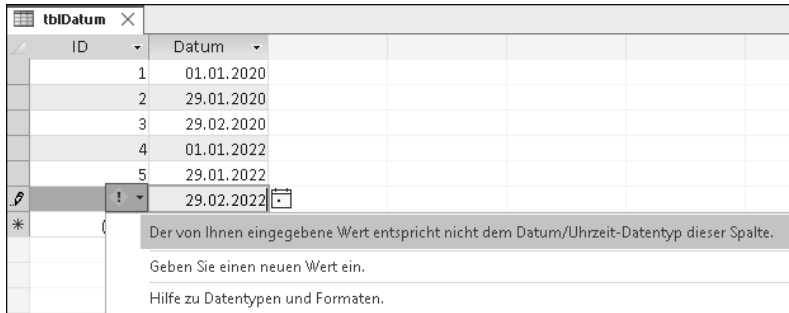
Die Auflösung könnte sogar noch höher sein, denn der `DOUBLE`-Wert unterstützt 15 Stellen. Mit den üblichen Bordmitteln von Access können Sie Millisekunden oder noch kleinere Sekundenbruchteile jedoch nicht erfassen oder bearbeiten. In sehr vielen Fällen reicht die Genauigkeit von einer Sekunde aber aus.

Vergessen Sie bitte nicht, dass Epoche, Wertebereich und Genauigkeit in anderen Programmen unterschiedlich sein können. Beispielsweise speichert der Microsoft SQL Server Zeitangaben auf die Millisekunde genau, mit einem anderen Wertebereich und mit einer anderen Epoche!

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
DATUM/UHRZEIT	<i>Date/Time</i>	DATETIME	01.01.0100 – 31.12.9999	8 Byte
DATUM/UHRZEIT ERWEITERT	<i>Date/Time Extended</i>	–	01.01.0001 – 31.12.9999	42 Bytes

Tabelle 2.11 Zeitangaben in Access können auf die Sekunde genau genutzt werden.

Der Felddatentyp `DATUM/UHRZEIT` geht korrekt mit Schaltjahren um. Wenn Sie versuchen, den 29.02.2022 einzugeben, erhalten Sie eine Fehlermeldung wie in Abbildung 2.24.



ID	Datum
1	01.01.2020
2	29.01.2020
3	29.02.2020
4	01.01.2022
5	29.01.2022
6	29.02.2022

Abbildung 2.24 Die Fehlermeldung ist vielleicht etwas unverständlich, aber richtig: Der 29.02.2022 ist ein ungültiges Datum, denn das Jahr 2022 ist kein Schaltjahr.

Bei der Zeitrechnung gibt es aber noch ein paar fiese Gemeinheiten, die Access nicht beherrscht. Für den Felddatentyp DATUM/UHRZEIT unbekannt sind:

1. Schaltsekunden

Bisher gab es 37 Schaltsekunden, um Unregelmäßigkeiten und die Verlangsamung der Erdrotation auszugleichen.

2. Andere Kalendertypen

Verschiedene Kalendertypen sind insofern relevant, als der heute weltweit verwendete Kalender (der *Gregorianische Kalender*) erst im Jahr 1582 eingeführt wurde. Davor wurde im Mittelalter der *Julianische Kalender* verwendet, der ein paar Probleme mit den Schaltjahren hatte. Bis 1582 war der Julianische Kalender schon etwas aus dem Ruder gelaufen, sodass bei der Umstellung einfach 10 Tage übersprungen wurden. Die Tage vom 05.10.1582 bis einschließlich 14.10.1582 gab es also nie ...

Diese Feinheiten sind Access jedoch unbekannt. Es ist daher wenig sinnvoll, Kalenderangaben vor dem 15.10.1582 exakt anzugeben. In anderen Ländern fand die Umstellung auf den Gregorianischen Kalender noch später statt (z. B. im Britischen Weltreich einschließlich der amerikanischen Kolonien erst im Jahr 1752). Exakte Zeitangaben in der frühen Neuzeit sind also durchaus problematisch.

3. Zeitzonen

Angaben zur Zeitzone sollten Sie in einem zusätzlichen Feld abspeichern, wenn dies notwendig ist. In diesem Fall empfehle ich Ihnen, Zeitangaben grundsätzlich in der *Universalzeit* (UTC) abzuspeichern (mehr dazu folgt im Kasten »Coordinated Universal Time (UTC) – die Universalzeit«).

4. Winterzeit und Sommerzeit

Üblicherweise wird bei der Angabe der Zeitzone vermerkt, ob die Winterzeit (z. B. *mitteleuropäische Zeit*, MEZ; engl. *Central European Time*, CET) oder die Sommerzeit (z. B. *mitteleuropäische Sommerzeit*, MESZ; engl. *Central European Summer Time*, CEST) gilt. Sollte es also relevant sein, so empfehle ich Ihnen auch hier wärmstens die Universalzeit UTC.

Für diese Sonderfälle in der Zeitrechnung müssen Sie leider selbst Sorge tragen, beispielsweise durch ein zusätzliches Feld »Zeitzone«.

Coordinated Universal Time (UTC) – die Universalzeit

Ich erinnere mich noch gut an eine Projektarbeit während meines Studiums, an der vier Studenten beteiligt waren. Jede Woche mussten wir bis freitags um 14:00 Uhr Ergebnisse bei unserem Dozenten per E-Mail einreichen. Etwas hinderlich war, dass wir in diesem Semester Praktika an verschiedenen Orten hatten: Der Erste hatte sicherlich das abenteuerlichste Praktikum von uns allen, denn er befand sich mitten in Sibirien in Irkutsk. Ein anderer Kommilitone war in Philadelphia, ich selbst in Boston, und der vierte aus dem Team war in Hannover geblieben. Die verschiedenen Zeitzonen führten schnell zu Missverständnissen: »Meinst du jetzt 10 Uhr deine oder meine Zeit?«

Immer dann, wenn verschiedene Zeitzonen oder die Sommerzeit stören, hilft die universelle Zeit UTC. Die universelle Zeit läuft nämlich immer normal weiter, sie kennt keine Sprünge wie z. B. unterschiedliche Zeitzonen. Wenn es in Deutschland 10 Uhr UTC ist, dann ist das an jedem anderen Ort in der Welt ebenso. Bezogen auf meine Studienarbeit: Mit UTC hätten wir es einfach gehabt, denn ein Treffen um 10 Uhr UTC lässt sich einfacher planen.

UTC entspricht der Ortszeit von Großbritannien und anderen Ländern der Zeitzone 0 – allerdings nur in der Winterzeit. Für Deutschland gilt eine Stunde mehr, daher während der Winterzeit UTC +1 h bzw. während der Sommerzeit UTC +2 h. Für andere Zeitzonen gibt es entsprechende Angaben (z. B. UTC–5 für die Ostküste der USA, Winterzeit).

Wenn Sie Zeitpunkte in verschiedenen Zeitzonen erfassen möchten, sollten Sie alle Zeitangaben als UTC ablegen. In einem zusätzlichen Feld können Sie die Zeitzone (*UTC-Offset*) angeben. Nur so können Sie die Zeitpunkte richtig miteinander vergleichen. Hier ein paar Beispiele dafür, wann das von Belang sein könnte:

- ▶ Termine, die über verschiedene Zeitzonen hinweg koordiniert werden sollen
- ▶ Flüge über verschiedene Zeitzonen hinweg
- ▶ Laufzeiten und Messwerte einer Windturbine, die auch während der Umstellung von Winterzeit auf Sommerzeit bzw. umgekehrt in Betrieb ist

Übrigens läuft die interne Uhr Ihres Computers auch mit UTC. Die universelle Zeit UTC ist also nichts Exotisches, sondern etwas sehr Praktisches.

Der Felddatentyp »Datum/Uhrzeit erweitert«

Wie versprochen kommen zum Ende dieses Abschnitts noch die Informationen zum Felddatentyp DATUM/UHRZEIT ERWEITERT, den es erst seit Access 2021 gibt. Sobald Sie diesen Felddatentyp nutzen, können Sie die Datenbank nicht mehr mit einer Version vor Access 2021 öffnen. Sie kennen das bereits vom Felddatentyp GROSSE GANZZAHL, der auch relativ neu ist. Das Versionsproblem dürfte einer der Hauptgründe dafür sein, dass man eher die Finger von

DATUM/UHRZEIT ERWEITERT lässt, wenn es nicht unbedingt sein muss. So werde ich es auch handhaben, damit Sie die Materialien zu diesem Buch problemlos mit einer älteren Access-Version öffnen können.

Andererseits gibt es durchaus eine Daseinsberechtigung für DATUM/UHRZEIT ERWEITERT, und zwar im Zusammenspiel mit einer Datenbank auf einem SQL Server. Wie das funktioniert, werde ich in Abschnitt 4.8, »Workshops zum SQL Server«, vorstellen. Es ist nämlich so, dass der SQL Server seine eigenen Felddatentypen hat und es für den Access-Felddatentyp DATUM/UHRZEIT lange Zeit kein exaktes Gegenstück auf dem SQL Server gab bzw. umgekehrt für `datetime`, was mitunter zu Schwierigkeiten führen konnte. Im Detail geht es um die unterschiedliche Genauigkeit der Zeitangaben:

- ▶ DATUM/UHRZEIT: 0,001 s = 1 ms (1 Millisekunde)
- ▶ `smalldatetime`: 1 min
- ▶ `datetime`: 0,00333 s = 3,33 ms
- ▶ DATUM/UHRZEIT ERWEITERT, `datetime2`: 0,0000001 s = 100 ns (100 Nanosekunden)

`smalldatetime`, `datetime` und `datetime2` sind Felddatentypen des SQL Servers und jetzt wird es klar: Erst mit DATUM/UHRZEIT ERWEITERT gibt es in Access den lange ersehnten Felddatentyp, der exakt zum SQL Server passt. Allerdings sprechen wir hier von Unterschieden im Millisekundenbereich oder kürzer, was nur in wenigen Fällen relevant sein dürfte. In Access ist es ohnehin sehr schwierig, Zeitangaben unterhalb einer Sekunde richtig zu formatieren und zu verarbeiten. Somit wird der traditionelle Felddatentyp DATUM/UHRZEIT fast immer die richtige Wahl sein.

2.3.7 Der Felddatentyp »Währung«

Eine der wichtigsten Sachen der Welt ist bekanntlich das liebe Geld. Dafür gibt es in Access einen eigenen Felddatentyp, nämlich WÄHRUNG. Dies ist eine Festkommazahl mit 15 Stellen vor dem Komma und vier Nachkommastellen. Im Gegensatz zu Gleitkommazahlen können Sie hiermit exakte Berechnungen durchführen, was bei Geldbeträgen besonders wichtig ist.

Auch wenn ich es schon mehrfach erwähnt habe, möchte ich an dieser Stelle noch einmal daran erinnern: Verwenden Sie *niemals* Gleitkommazahlen (Felddatentyp SINGLE oder DOUBLE), um Geldbeträge zu speichern! Die Artefakte der Gleitkommaarithmetik sind für ein so sensibles Thema wie Geld einfach nicht akzeptabel. Denken Sie auch daran, dass Ihre Kollegen einer Datenbank, die Sie erstellt haben, vertrauen und diese als Werkzeug akzeptieren sollen. Ein Killerargument wie »Ihre Datenbank kann ja nicht einmal richtig rechnen!« wäre doch eine wahre Katastrophe, oder?

Mit dem Felddatentyp WÄHRUNG speichert Access nicht das Währungssymbol. Wenn Sie nur mit einer Währung arbeiten, ist dies nicht weiter schlimm. Angezeigt wird das standardmäßige Währungssymbol Ihrer Windows-Einstellungen. Wenn gewünscht, können Sie über

Formatierung ein anderes Währungssymbol festlegen (siehe Abschnitt 2.6, »Formatierungen in Tabellen«).

Wenn Sie in einer Tabelle Geldbeträge in unterschiedlichen Währungen ablegen wollen, gibt es nur einen sauberen Weg: ein separates Feld, in dem die Währung gespeichert wird. Dafür eignet sich beispielsweise der Währungscode nach dem ISO-Standard 4217, der immer aus drei Buchstaben besteht:

Währungscode	Währung
EUR	Euro
GBP	Britische Pfund Sterling
USD	US-Dollar
JPY	Japanische Yen

Tabelle 2.12 Einige Währungscode nach dem ISO-Standard 4217

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
WÄHRUNG	<i>Currency</i>	MONEY	-922.337.203.685.477,5808 – 922.337.203.685.477,5807	8 Byte

Tabelle 2.13 Der Felddatentyp »Währung« wird intern als Festkommazahl mit vier Nachkommastellen gespeichert.

Der Felddatentyp »Währung« wird manchmal auch zweckentfremdet

Bevor in Access 2000 der Felddatentyp DEZIMAL eingeführt wurde, war WÄHRUNG der einzige Felddatentyp für Festkommazahlen. Deshalb wurde und wird er gerne zweckentfremdet, nämlich für exakte Dezimalzahlen mit bis zu vier Nachkommastellen. Mit einer geeigneten Formatierung (siehe Abschnitt 2.6, »Formatierungen in Tabellen«) können Sie in solchen Fällen das Währungssymbol unterdrücken.

Ich bin eigentlich immer ein Befürworter davon, so nahe wie möglich an der Realität zu bleiben. Konkret würde das bedeuten, dass dort, wo Währung draufsteht, wirklich nur Geld drinsteckt. Ich finde den Felddatentyp WÄHRUNG aber derart praktisch (vor allem, was die Programmierung angeht), dass ich hier meinen Prinzipien untreu werde.

Wie auch immer Sie sich entscheiden, Sie sollten hierzu eine einheitliche Regelung zu Projektbeginn treffen. Das gilt umso mehr, wenn Sie in einem Team arbeiten. Achten Sie bitte auf aussagekräftige Feldnamen (z. B. »MehrwertsteuerProzent«), damit jeder sofort erkennt, welche Werte tatsächlich abgespeichert werden (in diesem Beispiel: Prozentsatz statt Währung).

2.3.8 Der Felddatentyp »Ja/Nein«

Für die einfachste Art von Information gibt es den Felddatentyp JA/NEIN. Eine andere Bezeichnung ist *boolesche Variable* (engl. *boolean*) oder *Flag*. Praktischerweise zeigt Access in der Datenblattansicht ein Ja/Nein-Feld als *Kontrollkästchen* (engl. *check box*) an, in der Sie ein Häkchen setzen und wieder entfernen können.

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
JA/NEIN	Yes/No (<i>Boolean</i>)	BIT	»Ja« oder »Nein«	1 Byte

Tabelle 2.14 In einem »Ja/Nein«-Feld sind nur die Werte »Ja« (»wahr«, engl. »true«) oder »Nein« (»falsch«, engl. »false«) zulässig.

2.3.9 Der Felddatentyp »OLE-Objekt«

Für große Binärdaten ist der Felddatentyp OLE-OBJEKT geeignet. Dieser Felddatentyp ist verwandt mit dem Felddatentyp LANGER TEXT, den ich in Abschnitt 2.3.3, »Der Felddatentyp »Langer Text«, besprochen habe. Auch für OLE-Objekte gelten nämlich ein paar Besonderheiten:

- ▶ Maximal 2 GB Binärdaten können pro Feld gespeichert werden.
- ▶ Die maximal zulässige Anzahl von Bytes können Sie nicht festlegen.
- ▶ Das Sortieren und Filtern von Binärdaten ist nicht möglich (und auch nicht sinnvoll).

Access nutzt für Binärdaten den Mechanismus der *Objektverknüpfung und -einbettung* (engl. *Object Linking and Embedding, OLE*), woraus sich auch der Name dieses Felddatentyps ableitet. Sie können eine beliebige Datei in das Feld speichern (oder verknüpfen). Access speichert zusätzlich zum Dateinhalt weitere Informationen darüber ab, mit welchem Programm die Datei später einmal wieder geöffnet werden kann.

Diesen OLE-basierten Ansatz können Sie auch verwenden, um Bilder in Ihrer Datenbank abzuspeichern. In Formularen und Berichten können die Bilder später auch dargestellt werden. Dies ist sicherlich der häufigste Einsatzzweck des Felddatentyps OLE-OBJEKT.

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
OLE-OBJEKT	<i>OLE Object</i>	IMAGE	maximal 2 GB Daten	Größe der Binärdaten + zusätzliche OLE-Informationen

Tabelle 2.15 Platz für Binärdaten ohne Ende: der Felddatentyp »OLE-Objekt«

2.3.10 Andere Felddatentypen

In den letzten Abschnitten haben Sie alle wichtigen Felddatentypen kennengelernt. Es gibt noch ein paar Sonderlinge:

1. den Felddatentyp »Binär«

Neben dem OLE-OBJEKT gibt es den Felddatentyp BINÄR zum Speichern von Binärdaten. Mit diesem Felddatentyp können Sie maximal 255 Byte speichern, was verschwindend wenig ist im Vergleich zu 2 GB für ein OLE-OBJEKT. Daher wird der Felddatentyp BINÄR heute so gut wie überhaupt nicht mehr verwendet. Sie können ihn nicht einmal in der Entwurfsansicht einer Tabelle auswählen! Wenn Sie diesen exotischen Felddatentyp unbedingt einsetzen wollen, hilft nur die Programmierung – aber es gibt ihn noch.

2. den Felddatentyp »Anlage«

Mit dem Felddatentyp ANLAGE bietet Access Ihnen eine weitere Möglichkeit, Binärdaten in Ihrer Datenbank zu speichern. Im Unterschied zum OLE-OBJEKT kann ANLAGE mehr als nur eine Datei pro Feld speichern. ANLAGE ist der modernere Weg, Bilder oder Binärdateien innerhalb einer Datenbank zu speichern.

Felddatentyp	englische Bezeichnung	Jet-SQL-Datentyp	Wertebereich	Speicherbedarf
BINÄR	<i>Binary</i>	BINARY	maximal 255 Byte Daten	Anzahl der Byte
ANLAGE	<i>Attachment</i>	–	maximal 256 MB pro Datei	Größe der Binärdaten
LINK	<i>Hyperlink</i>	–	maximal 2.048 Zeichen	2 Bytes je Zeichen

Tabelle 2.16 Mit dem Felddatentyp »Anlage« können Sie auch mehrere Dateien in einem Feld abspeichern.

Welcher Felddatentyp ist besser: »Anlage« oder »OLE-Objekt«?

Ich empfehle Ihnen, üblicherweise den Felddatentyp ANLAGE einzusetzen. Das ist der modernere Weg und erspart Ihnen ein paar Altlasten von OLE (*Object Linking & Embedding* = Mechanismus der Objektverknüpfung und -einbettung).

Wenn Sie jedoch sicherstellen möchten, dass in einem Feld wirklich nur eine einzige Datei abgespeichert werden kann, dann müssen Sie nach wie vor den Felddatentyp OLE-OBJEKT verwenden. In meinen Augen ist dies eine wichtige Überlegung in Bezug auf das Datenbankdesign. Daneben fallen mir drei technische Einschränkungen ein, die den Felddatentyp OLE-OBJEKT unverzichtbar machen:

- ▶ Das Verlinken zu einer Datei im Dateisystem geht nur über OLE-OBJEKT.
- ▶ Server-Datenbanken (beispielsweise der Microsoft SQL Server) kennen den Felddatentyp ANLAGE nicht.
- ▶ Der Felddatentyp ANLAGE blockiert bestimmte Dateitypen (unter anderem *.exe*-Dateien).

3. den Felddatentyp »Link«

Schließlich gibt es noch den Felddatentyp LINK für Hyperlinks. Das Schöne an diesem Felddatentyp ist, dass Sie einen Hyperlink direkt im Feld anklicken können und dadurch die entsprechende Webseite in Ihrem Standardbrowser geöffnet wird.

2.3.11 Berechnete Felder

Eigentlich sind Tabellen nur zum Erfassen, Ändern und Löschen von Daten da. Die Auswertung von Daten – und dazu zählen auch Berechnungen – hat hier üblicherweise nichts verloren, sondern gehört zum Thema *Abfragen*. In Abschnitt 3.2, »Auswerten von Daten eines Datensatzes: Berechnete Felder«, werde ich Ihnen zeigen, wie Sie Berechnungen mithilfe von Abfragen durchführen können.

Sehr zu meinem Leidwesen gibt es in Access inzwischen den Felddatentyp BERECHNET. Ich mag diesen Felddatentyp deswegen nicht, weil er den Prozess des Datenbankdesigns nur unnötig verkompliziert. Sie werden sich die Arbeit deutlich einfacher machen, wenn Sie Berechnungen dort lassen, wo sie hingehören: bei den Abfragen.

Der Vollständigkeit halber zeige ich Ihnen trotzdem, wie Sie innerhalb einer Tabelle berechnete Felder verwenden können. Wenn Sie den Felddatentyp BERECHNET auswählen, zeigt Access Ihnen automatisch das Fenster AUSDRUCKS-GENERATOR an (siehe Abbildung 2.25).

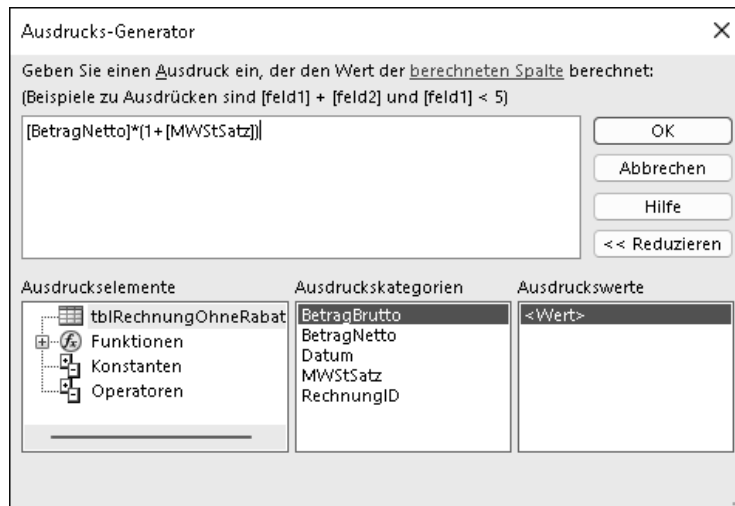


Abbildung 2.25 Mit dem Ausdrucks-Generator können Sie eine Formel zur Berechnung eines berechneten Feldes erstellen.

Tragen Sie hier die *Formel* zur Berechnung ein. Sinnvollerweise greift eine Berechnung Werte aus anderen Feldern auf. Schreiben Sie dazu die Feldnamen in eckigen Klammern ([und]). Hier ein paar Beispiele:

1. Berechnung des Gesamtbetrags mit Mehrwertsteuer:

$$[\text{BetragNetto}] * (1 + \text{MWStSatz})$$

2. Auf einen Betrag einen bestimmten Rabatt gewähren:

$$[\text{BetragNetto}] * (1 - [\text{Rabatt}])$$

3. Erst einen bestimmten Rabatt gewähren, dann die Mehrwertsteuer berechnen:

$$[\text{BetragNetto}] * (1 - [\text{Rabatt}]) * (1 + \text{MWStSatz})$$

Nachdem Sie auf OK geklickt und den Ausdrucks-Generator dadurch geschlossen haben, finden Sie Ihre Formel zur Berechnung im Registerblatt ALLGEMEIN unter AUSDRUCK. Unter ERGEBNISTYP können Sie abschließend den gewünschten Felddatentyp für das Ergebnis der Berechnung auswählen. Ich habe diese Beispiele und ein paar weitere Berechnungen in die Datenbank *O2_Access_als_Datenbank_Tabellen\2.3.11_Berechnete_Felder.accdb* aus den Materialien zum Buch gepackt.

Kompatibilität mit SharePoint

Mein Eindruck ist, dass Microsoft innerhalb der letzten Jahre Access mit viel Aufwand in Richtung SharePoint ausgerichtet hat. Und genau hierfür, nämlich für die Kompatibilität mit Microsoft SharePoint, ist der Felddatentyp `BERECHNET` notwendig.

2.3.12 Feldeinschränkungen

Sie kennen jetzt alle Felddatentypen von Access. Wenn Sie für ein Feld den passenden Felddatentyp auswählen, kann Access damit für konsistente Daten sorgen (und beispielsweise verhindern, dass in einem Zahlenfeld Text abgespeichert wird).

Neben den Felddatentypen können und sollten Sie für ein Feld weitere Einschränkungen festlegen. Access wird auch diese *Feldeinschränkungen* überprüfen, bevor ein neuer Wert abgespeichert wird. Wir werden uns in diesem Abschnitt mit den folgenden Feldeinschränkungen beschäftigen:

1. EINGABE ERFORDERLICH
2. Feld-Gültigkeitsregel
3. STANDARDWERT

Bevor wir uns die einzelnen Feldeinschränkungen ansehen, möchte ich an Sie appellieren, von ihnen rigoros Gebrauch zu machen. Nur so kann Access überhaupt erst Dateneingaben vor dem Abspeichern prüfen und auf diesem Wege die Datenkonsistenz gewährleisten.

Eingabe erforderlich

Eine sehr wichtige Feldeinschränkung ist EINGABE ERFORDERLICH. Beispielsweise ist es wenig sinnvoll, einen Flug ohne Datum, Abflugszeit, Zeit der Landung, Angaben zu den Flughäfen und ohne Flugnummer anzugeben. Alle diese Felder sollten gefüllt sein.

1. Öffnen Sie die Datenbank *O2_Access_als_Datenbank_Tabellen\2.3.12_Fluege.accdb* aus den Materialien zum Buch.
2. Öffnen Sie die Tabelle *tblFlug* in der Entwurfsansicht.
3. Wählen Sie das Feld »AbflugDatum« aus.
4. Wählen Sie im Registerblatt ALLGEMEIN unter EINGABE ERFORDERLICH den Wert JA aus.
5. Setzen Sie auch für alle anderen Felder mit Ausnahme von »FlugID« und »Passagiere« EINGABE ERFORDERLICH auf JA.

Das Ergebnis finden Sie in den Materialien zum Buch unter *O2_Access_als_Datenbank_Tabellen\2.3.12_Fluege_Eingabe_erforderlich.accdb*.

Ein AUTOWERT-Feld wird von Access automatisch eingetragen, daher können Sie für »FlugID« die Feldeinschränkung EINGABE ERFORDERLICH nicht verändern. Jetzt ist nur noch das Feld »Passagiere« optional und darf leer bleiben. Für alle anderen Felder wird Access vor dem Abspeichern des Datensatzes überprüfen, ob wirklich ein Wert eingetragen ist.

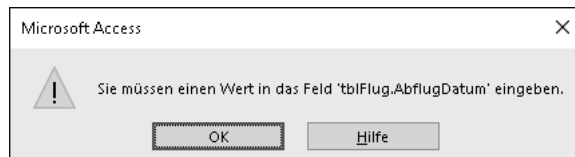


Abbildung 2.26 Eine solche Fehlermeldung erscheint, wenn Sie vergessen haben, ein Feld zu füllen, für das die Eingabe erforderlich ist.

Den Fall »ein Feld bleibt leer« sollten wir uns noch genauer ansehen. In der Welt der Datenbanken hat ein leeres Feld den Inhalt NULL. Am besten passt »nicht definiert« als Bedeutung für NULL. Wenn Sie einen neuen Datensatz erstellen, sind zunächst alle Felder NULL (also nicht definiert; NULL ist nämlich streng genommen kein Wert). Die Feldeinschränkung EINGABE ERFORDERLICH bezeichnet man auch als *NOT NULL* (Abkürzung: NN). Mit anderen Worten: Beim Abspeichern auf keinen Fall NULL akzeptieren.

NULL und die Zahl Null (numerisch 0)

Auch wenn es so ähnlich klingt, hat NULL überhaupt nichts mit der Zahl 0 gemeinsam. Wenn ich 0 Euro auf meinem Konto habe, sind meine Ersparnisse aufgebraucht. NULL bedeutet hingegen, dass ich meinen Kontostand gerade nicht kenne.

Access und VBA verwenden die Schreibweise Null, was ich für Formeln und Listings übernommen habe. Um Missverständnisse zu vermeiden, verwende ich im Text die Schreibweise NULL in Großbuchstaben.

Felder, für die EINGABE ERFORDERLICH auf JA gesetzt ist, werden im Fachjargon als *NOT-NULL-Felder* bezeichnet. Alle anderen Felder werden *optionale Felder*, *NULL-Felder* oder *Nullable-Felder* genannt.

NOT-NULL-Textfelder

Bei Textfeldern (Felddatentyp KURZER TEXT oder LANGER TEXT) gibt es übrigens einen Unterschied zwischen NULL und *leerer Zeichenfolge* (""). Leider werden in Access beide Inhalte in der Tabellenansicht identisch als leerer Kasten angezeigt.

In den meisten Fällen wird eine leere Zeichenfolge nicht aussagekräftig sein. Daher empfehle ich Ihnen, leere Zeichenfolgen für Textfelder zu verbieten. Öffnen Sie dazu die Tabelle in der Entwurfsansicht, und wählen Sie im Registerblatt ALLGEMEIN unter LEERE ZEICHENFOLGE den Wert NEIN aus. Anschließend können Sie über EINGABE ERFORDERLICH festlegen, ob das Textfeld ein NOT-NULL-Feld oder ein optionales Feld ist. Ein Beispiel finden Sie in den Materialien zum Buch unter *02_Access_als_Datenbank_Tabellen\2.3.12_Fluege_leere_Zeichenfolge.accdb*.

FlugID	AbflugDatum	AbflugZeit	AnkunftsZeit	Fluggesellschaft	Flugnummer
1	03.07.2019	15:00:00	16:20:00	easyJet	EJU5846
2	03.07.2019	15:10:00	16:40:00	Austrian Airlines Group	OS131
3	03.07.2019	15:40:00	17:20:00		AF2334
4	03.07.2019	15:50:00	17:35:00	Air France	AF2035
5	03.07.2019	15:55:00	17:15:00	Air France	AF1919

Abbildung 2.27 In diesem Beispiel ist »Fluggesellschaft« ein NOT-NULL-Feld. Im dritten Datensatz enthält das Feld eine leere Zeichenfolge, was keinen Deut besser ist als NULL. Mit der Feldeinschränkung »Leere Zeichenfolge« können Sie das verhindern.

Ich empfehle Ihnen, NOT-NULL-Felder intensiv zu nutzen. Viele Felder werden natürlich optional bleiben müssen. Aber schauen Sie bitte genau hin, ob Ihr Geschäftsmodell das benötigt. Warum liebe ich NOT-NULL-Felder so sehr? Weil optionale Felder Mehraufwand erzeugen. In Abfragen und spätestens bei der Programmierung müssen Sie für jedes optionale Feld den separaten Fall unterscheiden, dass das Feld leer ist. Dieser Mehraufwand tritt bei jeder Art von Fallunterscheidung auf. NOT-NULL-Felder sind hingegen einfacher: Sie können ganz sicher sein, dass überhaupt ein Wert eingetragen ist. Den NULL-Wert (nicht definiert) brauchen Sie für NOT-NULL-Felder nicht zu berücksichtigen.

Ternäre Logik

Mit NULL-Werten kommen Sie bei Fallunterscheidungen mit der *ternären Logik* in Berührung. In der *klassischen Logik*, der *zweiwertigen Logik*, gibt es nur die Werte »wahr« und »falsch«. In der ternären Logik gibt es die Werte »wahr«, »falsch« und »nicht definiert« (NULL). Der Umgang mit der ternären Logik ist aufwendiger. Wenn alle Felder einer Fallunterscheidung NOT-NULL-Felder sind, vereinfacht sich die Logik zur zweiwertigen Logik.

Feld-Gültigkeitsregel

Mit Feld-Gültigkeitsregeln bietet Access eine elegante Möglichkeit, den Inhalt eines Feldes noch weiter einzuschränken. Als Gültigkeitsregel wird eine Formel eingetragen, die Sie entweder direkt eintippen oder mithilfe des Ausdrucks-Generators erstellen können. Formeln haben Sie bereits in Abschnitt 2.3.11, »Berechnete Felder«, kurz kennengelernt. An dieser Stelle beschäftigen wir uns mit Formeln für den Vergleich.

Hier sehen Sie ein paar Beispiele für Textfelder:

- ▶ Das Feld soll immer den Wert »München« haben:

```
= 'München'
```

- ▶ Alle Werte mit Ausnahme von »München« sind erlaubt:

```
<> 'München'
```

- ▶ Als Wert ist entweder »München« oder »Berlin« erlaubt:

```
'München' Oder 'Berlin'
```

- ▶ Alle gültigen Werte sind in einer Liste aufgeführt:

```
In ('Berlin'; 'Hamburg'; 'München')
```

- ▶ Alle Werte sind erlaubt mit Ausnahme derer aus der Liste:

```
Nicht In ('Berlin'; 'Hamburg'; 'München')
```

- ▶ Alle Werte, die mit »M« beginnen, sind erlaubt:

```
Wie 'M*'
```

- ▶ Nur Werte, die nicht auf »burg« enden, sind erlaubt:

```
Nicht Wie '*burg'
```

Übrigens spielt es keine Rolle, ob Sie Zeichenfolgen mit einfachen Anführungszeichen (') oder doppelten Anführungszeichen (") einrahmen.

Textmustervergleich mit den Operatoren »Wie« und »ALike«

Der Operator **Wie** (engl. *like*) ist ein sehr mächtiges Werkzeug, das einen Text mit einem Muster vergleicht (*Mustervergleich*, engl. *pattern matching*). In Tabelle 2.17 sind alle *Platzhalter* (engl. *wildcard*) mit Beispielen aufgeführt.

ALike ist der zweite Operator zum Textmustervergleich. Die Platzhalter sind jedoch unterschiedlich (siehe Tabelle 2.18). Der Name **ALike** bedeutet »ANSI Like« und lässt vermuten, dass dieser Operator den Vorgaben des ANSI/ISO-SQL-92-Standards entspricht. Wenn Sie den Originaltext dieses Standards sorgfältig lesen, werden Sie aber feststellen, dass das nicht der Fall ist! Zutreffender ist eher, dass **ALike** sehr nahe an den **Like**-Operator des Microsoft SQL Servers angelehnt ist.

Ob Sie sich nun für **Like** oder **ALike** entscheiden, ist letztendlich Geschmackssache. Der einzige Unterschied sind wie gesagt die Platzhalter. Wenn Sie eine Server-Datenbank verwenden, könnte Ihnen **ALike** besser gefallen.

Platzhalter	Bedeutung	Muster (Beispiel)	passende Zeichenfolgen	nicht passende Zeichenfolgen
*	kein, ein oder mehrere beliebige Zeichen	M*	Mainz, München, M	Hamburg, Berlin, H
?	ein beliebiges Zeichen	H?mburg	Hamburg, Homburg	Mainz, Hmburg, Haamburg
#	eine beliebige Ziffer	LH####	LH2890	LH191, LHxxxx, LH12345
[...]	ein Zeichen aus einer Liste	H[ao]mburg	Hamburg, Homburg	Hemburg, Hmburg, Haamburg
[*] [?] [#] [[[]]	die Sonderzeichen selbst	H[*]mburg	H*mburg	Hamburg, Homburg, Hmburg
[!...]	ein Zeichen, aber nicht aus der Liste	H[!ao]mburg	Hemburg	Hamburg, Homburg, Hmburg, Haamburg

Tabelle 2.17 Platzhalter für den Operator »Wie«

Platzhalter	Bedeutung	Muster (Beispiel)	passende Zeichenfolgen	nicht passende Zeichenfolgen
[...!]	ein Zeichen aus einer Liste inklusive Ausrufezeichen	H[ao!]mburg	Hamburg, Homburg, H!mburg	Hemburg, Hmburg
[...-...]	ein Zeichen aus einem Bereich	[a-z]	a, b, c, A, B, ö, ß	aa, 1, 4, %
[!...-...]	ein Zeichen, aber nicht aus einem Bereich	[!0-9]	a, b, c, A, B, %, ö, ß	1, 4, aa

Tabelle 2.17 Platzhalter für den Operator »Wie« (Forts.)

Platzhalter	Bedeutung	Muster (Beispiel)	passende Zeichenfolgen	Nicht passende Zeichenfolgen
%	kein, ein oder mehrere beliebige Zeichen	M%	Mainz, München, M	Hamburg, Berlin, H
_	ein beliebiges Zeichen	H_mburg	Hamburg, Homburg	Mainz, Hmburg, Haamburg
[...]	ein Zeichen aus einer Liste	H[ao]mburg	Hamburg, Homburg	Hemburg, Hmburg, Haamburg
[%] [_] [[[]]	die Sonderzeichen selbst	H[%]mburg	H%mburg	Hamburg, Homburg, Hmburg
[!...]	ein Zeichen, aber nicht aus der Liste	H[!ao]mburg	Hemburg	Hamburg, Homburg, Haamburg
[...!]	ein Zeichen aus einer Liste inkl. Ausrufezeichen	H[ao!]mburg	Hamburg, Homburg, H!mburg	Hemburg, Hmburg
[...-...]	ein Zeichen aus einem Bereich	[a-z]	a, b, c, A, B, ö, ß	aa, 1, 4, %

Tabelle 2.18 Platzhalter für den Operator »ALike«

Platzhalter	Bedeutung	Muster (Beispiel)	passende Zeichenfolgen	Nicht passende Zeichenfolgen
[!...-...]	ein Zeichen, aber nicht aus einem Bereich	[!0-9]	a, b, c, A, B, %, ö, ß	1, 4, aa

Tabelle 2.18 Platzhalter für den Operator »ALike« (Forts.)

Ebenso sind Feld-Gültigkeitsregeln für Zahlenfelder möglich:

- ▶ Nur positive Zahlen:

>0

- ▶ Alle Werte von 1 bis 10 sind erlaubt:

Zwischen 1 Und 10

- ▶ Alle Werte zwischen 1 und 10 (aber 1 und 10 selbst sind nicht dabei):

>1 Und <10

- ▶ Alle Zahlen außer 0:

<>0

- ▶ Eine Liste von Werten zulassen:

In (2;4;8;16)

Aus Abschnitt 2.3.6, »Der Felddatentyp ›Datum/Uhrzeit‹ und ›Datum/Uhrzeit erweitert‹«, wissen Sie, dass eine Datumsangabe in Access als Zahl abgespeichert wird. Die Vergleiche für Felder mit dem Felddatentyp DATUM/UHRZEIT sehen daher genauso wie die der Zahlenfelder aus:

- ▶ Ein Datum ab dem 1. Februar 2020:

>=#01.02.2020#

- ▶ Ein Zeitpunkt im Mai 2022:

>=#01.05.2022# Und <#01.06.2022#

- ▶ Ein Datum in der Vergangenheit:

<Jetzt()

Auf eine kleine Falle möchte ich Sie hinweisen: Wenn Sie Zeitpunkte z. B. bis zum 31.05.2022 zulassen möchten, vergleichen Sie bitte wie im Beispiel oben mit einem Tag später (dem ersten Juni). Wichtig ist das, wenn Sie in Ihrem Feld Datum und Uhrzeit gleichzeitig erfassen wollen. Würden Sie nämlich <=#31.05.2022# festlegen, dann ist für den 31.05.2022 gerade

noch Mitternacht erlaubt. Eine Sekunde später liegt aber schon außerhalb des Gültigkeitsbereichs! Mit dem Vergleich `<#01.06.2022#` (nicht `<=`) sind Sie auf der sicheren Seite, und alle Zeitpunkte am letzten Tag bis 23:59:59 Uhr sind gültig.

Angaben von Datum und Uhrzeit in Formeln

Konkrete Zeitpunkte werden in Formeln in einem besonderen Format – eingerahmt durch zwei Hash-Zeichen (#) – eingetragen:

`#TT.MM.JJJJ hh:mm:ss#`

Möglich sind Angaben zum Datum, zur Uhrzeit oder beides gleichzeitig.

Beispiele:

- ▶ der erste Mai 2022: `#01.05.2022#`
- ▶ Zeit für ein Knoppers: `#09:30:00#`
- ▶ Zeit für ein Knoppers am ersten Mai 2022: `#01.05.2022 09:30:00#`

Sie haben in den Beispielen oben schon gesehen, dass zwei oder mehrere Bedingungen durch die Operatoren Und (engl. And) und Oder (engl. Or) kombiniert werden können. Der Operator Nicht (engl. Not) kehrt die Bedeutung um (*Negation*). Der Vollständigkeit halber erwähne ich noch den Operator ExOder für *exklusives Oder* (engl. *exclusive or*, *Xor*). Das bedeutet, entweder darf die erste Bedingung wahr sein oder die zweite, aber nicht beide gleichzeitig. Mit diesen Operatoren können Sie umfangreiche Gültigkeitsregeln generieren, beispielsweise:

Nicht -5 Und <0 Oder Nicht 5 Und Zwischen 2 Und 8

Reihenfolge der Operatoren und Klammern

Bei umfangreicheren Formeln wie im letzten Beispiel verliert man manchmal den Überblick, welche Teile der Formel mit welchem Nicht, Und, Oder und ExOder verknüpft sind. Für die Operatoren gilt diese Reihenfolge:

1. Zwischen, In, Wie
2. Nicht
3. Und
4. Oder
5. ExOder
6. =, >, <, >=, <=, <>

Ähnlich wie »Punkt- vor Strichrechnung« gilt also: »Nicht vor Und« sowie »Und vor Oder«. Mit runden Klammern können Sie eine abweichende Reihenfolge festlegen. Ich verwende runde Klammern selbst dann, wenn es nicht unbedingt notwendig ist, weil eine Formel nach meinem Empfinden dadurch besser lesbar wird.

Beispiel:

`((Nicht -5) Und <0) Oder ((Nicht 5) Und (Zwischen 2 Und 8))`

In den Materialien zum Buch finden Sie diese Beispiele in der Datenbank *O2_Access_als_Datenbank_Tabellen\2.3.12_Feldgultigkeitsregel.accdb*.

Wenn eine Feld-Gültigkeitsregel eingetragen ist, wird Access vor dem Abspeichern den neuen Wert mithilfe der Formel prüfen. Wenn die Bedingung wahr ist, wird der Wert abgespeichert. Wenn nicht, erscheint eine Fehlermeldung wie in Abbildung 2.28:

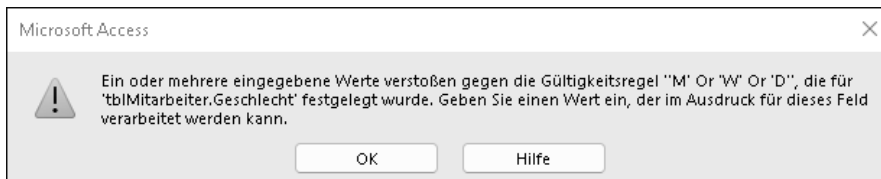


Abbildung 2.28 Ohne eigene Gültigkeitsmeldung erzeugt Access eine Fehlermeldung, in der die Feld-Gültigkeitsregel angezeigt wird. Meiner Erfahrung nach hilft das den wenigsten Anwendern weiter.

Im Registerblatt ALLGEMEIN unter GÜLTIGKEITSMELDUNG können Sie auch eine eigene Fehlermeldung eintragen, die aussagekräftiger ist (siehe Abbildung 2.29):

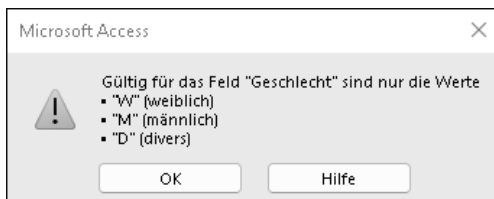


Abbildung 2.29 Hilfreicher und schöner sind eigene Gültigkeitsmeldungen.

Diese Gültigkeitsmeldung finden Sie in den Materialien zum Buch unter *O2_Access_als_Datenbank_Tabellen\2.7.5_Fluege_umfassend.accdb*, dort in der Tabelle *tblMitarbeiter*, Feld »Geschlecht«. Damit Ihre Datenbank richtig schön aussieht, zeige ich Ihnen hier die Schritte, mit denen Sie mehrere Zeilen und die Aufzählungspunkte erzeugen:

1. Öffnen Sie die Tabelle *tblMitarbeiter* in der Entwurfsansicht.
2. Wählen Sie das Feld »Geschlecht« aus.
3. Klicken Sie im Registerblatt ALLGEMEIN auf GÜLTIGKEITSMELDUNG.
4. Mit $\text{☰} + \text{[F2]}$ öffnen Sie das ZOOM-Fenster.
5. Tragen Sie hier den Text ein.
6. Zeilenumbrüche fügen Sie mit $\text{[Strg]} + \text{[↵]}$ ein.
7. Aufzählungspunkte erzeugen Sie mit $\text{[Alt]} + \text{[0]} \text{[1]} \text{[4]} \text{[9]}$ (Geben Sie die Ziffern über den Zahlenblock der Tastatur ein!)

Standardwert

Wenn Ihre Tabelle viele Felder hat, kann es schon sehr aufwendig werden, einen neuen Datensatz einzutragen. Häufig ist es so, dass einige Feldinhalte immer gleich bleiben. An dieser Stelle können Ihnen *Standardwerte* (engl. *default values*) viel Tipparbeit ersparen. Standardwerte spielen überhaupt nur eine Rolle, wenn ein Datensatz neu erstellt wird.

Als Beispiel möchte ich, dass Access in der Tabelle *tblFlug* in »AbflugFlughafen« standardmäßig immer »Frankfurt am Main« einträgt:

1. Öffnen Sie die Datenbank *O2_Access_als_Datenbank_Tabellen\2.3.12_Fluege.accdb* aus den Materialien zum Buch.
2. Öffnen Sie die Tabelle *tblFlug* in der Entwurfsansicht.
3. Wählen Sie das Feld »AbflugFlughafen« aus.
4. Tragen Sie im Registerblatt ALLGEMEIN unter STANDARDWERT

= "Frankfurt am Main"

ein.

5. Klicken Sie in der Symbolleiste für den Schnellzugriff auf SPEICHERN (oder drücken Sie `[Strg] + [S]`).

Das Ergebnis finden Sie in den Materialien zum Buch unter *O2_Access_als_Datenbank_Tabellen\2.3.12_Fluege_Standardwert.accdb*.

Wenn Sie für ein Feld einen Standardwert festgelegt haben, zeigt Ihnen Access in der Datenblattansicht den Standardwert schon an, bevor Sie den neuen Datensatz begonnen haben.

FlugID	AbflugDatum	AbflugZeit	AnkunftZeit	Fluggesellschaft	Flugnummer	AbflugFlughafen
43	03.08.2022	19:20:00	20:35:00	Ryanair	189	Berlin-Schönefeld
44	03.08.2022	19:25:00	20:45:00	easyJet	5847	Berlin-Tegel
45	03.08.2022	19:30:00	20:40:00	Lufthansa	196	Frankfurt am Main
46	03.08.2022	19:50:00	21:15:00	Air France	1819	Frankfurt am Main
47	03.08.2022	20:10:00	21:40:00	Austrian Airlines Group	127	Wien
48	03.08.2022	20:15:00	22:30:00	Eurowings	357	London Stansted
49	03.08.2022	21:55:00	22:40:00	Ryanair	8547	Berlin-Schönefeld
50	04.08.2022	06:05:00	09:40:00	Condor	1478	Frankfurt am Main
* (Neu)						Frankfurt am Main

Abbildung 2.30 Für den neuen Datensatz (in der letzten Zeile) wird für »AbflugFlughafen« bereits der Standardwert »Frankfurt am Main« angezeigt.

Sie können den Standardwert jederzeit mit Ihrem Wunschwert überschreiben. Insofern sind Standardwerte in Wirklichkeit gar keine Einschränkungen. Sie gehören trotzdem zu den Feldeinschränkungen; zumindest wird das bei der Programmierung per SQL deutlich. (Mehr dazu erfahren Sie in Abschnitt 5.6, »Datenbankobjekte verändern: Data Definition Language (DDL)«.)

Ich finde es sehr nützlich, dass auch komplexe Formeln mit Funktionen als Standardwert erlaubt sind. Hier sehen Sie zwei Beispiele für Felder mit dem Felddatentyp DATUM/UHRZEIT:

- ▶ das aktuelle Datum (ohne Uhrzeit):

=Datum()

- ▶ das aktuelle Datum (mit Uhrzeit):

=Jetzt()

An diesen Stellen können Sie Standardwerte in Access festlegen

In Access können Sie Standardwerte an zwei Stellen eintragen:

- ▶ auf Tabellenebene
- ▶ in einem Steuerelement eines Formulars

Um das Thema »Formulare« dreht sich das gesamte Kapitel 6. Das Schöne an Formularen ist, dass Sie für eine Tabelle mehrere verschiedene Formulare erstellen können. Wenn Sie also einen Standardwert für die eine Form der Eingabe benötigen, nicht jedoch in anderen Fällen, dann tragen Sie den Standardwert für das Steuerelement eines Formulars ein.

Noch ein Hinweis: Wenn Sie sowohl auf Tabellenebene als auch im Steuerelement eines Formulars einen Standardwert gesetzt haben, dann hat der Standardwert des Steuerelements im Formular beim Arbeiten im Formular Vorrang.

Zusammenfassung

In diesem Abschnitt haben Sie die Feldeinschränkungen EINGABE ERFORDERLICH, Feld-Gültigkeitsregel und Standardwert kennengelernt. Wenn eine Feldeinschränkung verletzt wird, zeigt Access eine Fehlermeldung an und verhindert, dass der Datensatz gespeichert wird. Mit einer Gültigkeitsmeldung können Sie die Fehlermeldung anpassen. Leider können Sie keine benutzerdefinierte Fehlermeldung für die Verletzung von EINGABE ERFORDERLICH hinterlegen – eigentlich schade. Bei Standardwerten wird keine Regel verletzt, insofern gibt es dafür auch keine Fehlermeldung.

Best Practice für Feldeinschränkungen

- ▶ Setzen Sie EINGABE ERFORDERLICH standardmäßig auf JA.
- ▶ Setzen Sie nur für Felder, die wirklich optional sind, EINGABE ERFORDERLICH auf NEIN.
- ▶ Für Textfelder, für die die Eingabe erforderlich ist, sollten Sie leere Zeichenfolgen nicht zulassen.
- ▶ Formulieren Sie eine Feld-Gültigkeitsregel, wann immer dies möglich ist.
- ▶ Tragen Sie einen Standardwert auf Tabellenebene ein, wenn der Standardwert für alle Formen der Eingabe sinnvoll ist.

- Tragen Sie einen Standardwert für ein Steuerelement eines Formulars ein, wenn der Standardwert nur für eine bestimmte Form der Eingabe (d. h. ein bestimmtes Formular) sinnvoll ist.

2.3.13 Tabelleneinschränkungen

Im letzten Abschnitt haben wir immer nur den Wert eines einzigen Feldes geprüft. Eine *Tabelleneinschränkung* hingegen kann mehrere Felder gleichzeitig prüfen. Es gibt zwei Arten von Tabelleneinschränkungen. Wir beschäftigen uns zunächst nur mit dem ersten Typ, der *Tabellen-Gültigkeitsregel*.

Mehrere Felder in Kombination prüfen

Verwenden Sie eine Tabellen-Gültigkeitsregel immer dann, wenn mehrere Felder in ihrer Kombination wichtig sind.

Ein Beispiel: Die Tabelle *tblMitarbeiter* speichert für jeden Mitarbeiter die beiden Felder »Geburtsdatum« und »Eintrittsdatum«. Es ist offensichtlich, dass ein Mitarbeiter erst nach seiner Geburt in das Unternehmen eintreten kann. »Eintrittsdatum« muss also später als »Geburtsdatum« liegen.

Diese Art von Regeln können Sie als Tabellen-Gültigkeitsregel definieren. Eine Tabellen-Gültigkeitsregel hängt nie an einem einzelnen Feld, sondern gilt übergreifend für den gesamten Datensatz. Und so erstellen Sie eine Tabellen-Gültigkeitsregel:

1. Öffnen Sie die Datenbank *O2_Access_als_Datenbank_Tabellen\2.3.13_Mitarbeiter.accdb* aus den Materialien zum Buch.
2. Öffnen Sie die Tabelle *tblMitarbeiter* in der Entwurfsansicht (siehe Abbildung 2.31).
3. Falls die Tabelleneigenschaften ❶ noch nicht sichtbar sind, klicken Sie auf TABELLENENTWURF • EIGENSCHAFTENBLATT ❷.
4. Tragen Sie im EIGENSCHAFTENBLATT (Tabelleneigenschaften) ❶ unter GÜLTIGKEITSREGEL ❸

[Eintrittsdatum] > [Geburtsdatum]

ein.

5. Klicken Sie in der Symbolleiste für den Schnellzugriff auf SPEICHERN ❹ (oder drücken Sie Strg + S).

Wichtig ist, dass Sie diese Gültigkeitsregel an der richtigen Stelle eintragen, nämlich unter GÜLTIGKEITSREGEL in den Tabelleneigenschaften ❶, nicht jedoch unter ALLGEMEIN • GÜLTIGKEITSREGEL für eines der Datenfelder ❺!

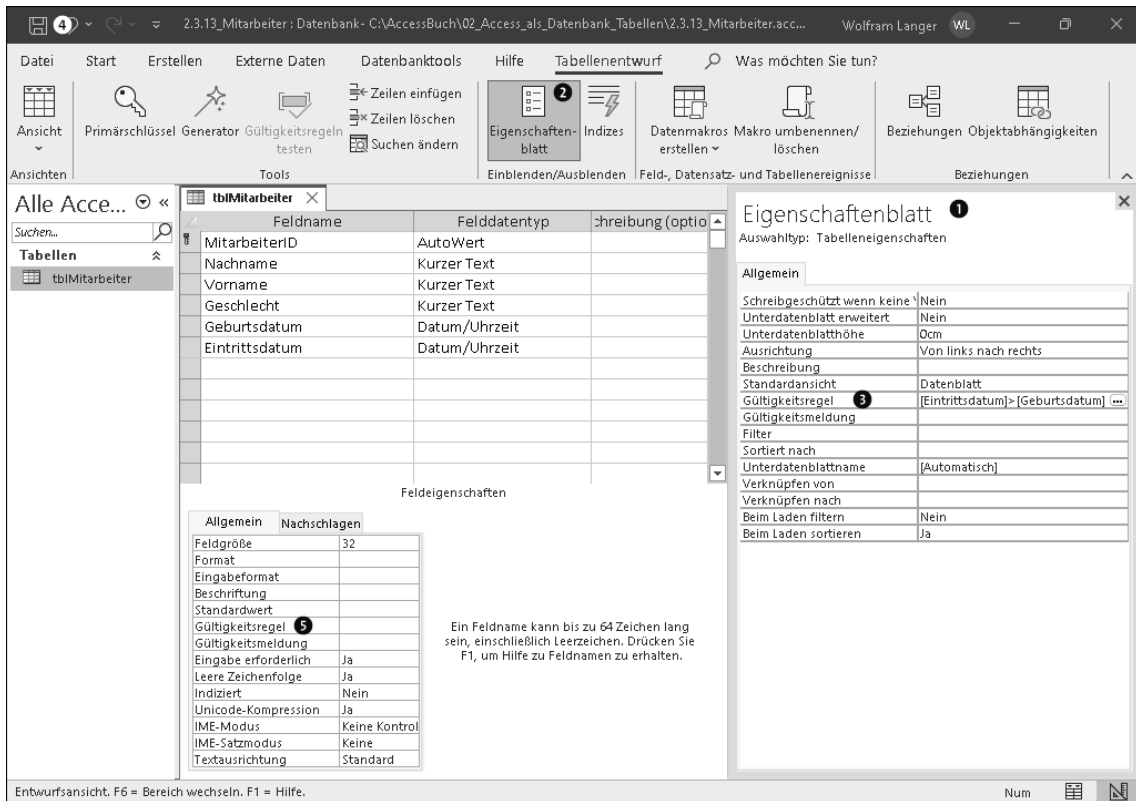


Abbildung 2.31 Eine Tabellen-Gültigkeitsregel können Sie in den Tabelleneigenschaften **1** unter »Gültigkeitsregel« **3** festlegen.

Im Gegensatz zur Feld-Gültigkeitsregel müssen Sie bei der Tabellen-Gültigkeitsregel immer den Feldnamen angeben. Der Feldname wird dazu in eckige Klammern gesetzt ([und]). Mit den Operatoren Und, Oder, ExOder und Nicht können Sie beliebig komplexe Gültigkeitsregeln erzeugen. Hier ein paar Beispiele:

1. Genau eines von drei JA/NEIN-Feldern soll gesetzt sein:

[GebundeneAusgabe] ExOder [Broschiert] ExOder [Taschenbuch]

2. Nur wenn das JA/NEIN-Feld »IstGestrichen« gesetzt ist, darf und muss in zwei anderen Feldern etwas eingetragen werden:

```
(
  ([IstGestrichen] = Falsch)
  Und ([IstGestrichenWordenDurch] Ist Null)
  Und ([IstGestrichenWordenAm] Ist Null)
)
```

Oder

```
(
  ([IstGestrichen] = Wahr)
  Und ([IstGestrichenWordenDurch] Ist Nicht Null)
  Und ([IstGestrichenWordenAm] Ist Nicht Null)
)
```

So weiß jeder, wann und durch welchen Mitarbeiter ein Flug annulliert wurde.

3. Wenn ein Flug verspätet ist, dann muss der tatsächliche Zeitpunkt von Abflug oder Ankunft später als geplant liegen:

```
([IstVerspaetet]=Falsch)
Oder
(
  ([IstVerspaetet]=Wahr)
  Und ([AbflugDatumTatsaechlich] Ist Nicht Null)
  Und ([AbflugZeitTatsaechlich] Ist Nicht Null)
  Und ([AnkunftZeitTatsaechlich] Ist Nicht Null)
  Und
  (
    ([AbflugDatumTatsaechlich]>[AbflugDatumGeplant])
    Oder ([AbflugZeitTatsaechlich]>[AbflugZeitGeplant])
    Oder ([AnkunftZeitTatsaechlich]>[AnkunftZeitGeplant])
  )
)
```

Alle diese Beispiele von Tabellen-Gültigkeitsregeln finden Sie in den Materialien zum Buch in der Datenbank *O2_Access_als_Datenbank_Tabellen\2.3.13_Tabellengueltigkeitsregel.accdb*.

Sie merken schon, dass Tabellen-Gültigkeitsregeln sehr umfangreich werden können. Um überhaupt einen Überblick zu erhalten, welches `Und` und `Oder` zu welcher Klammer gehört, habe ich die Tabellen-Gültigkeitsregeln in mehreren Zeilen und mit Einrückungen formatiert. Meistens erledige ich das in einem Texteditor, denn nach dem Abspeichern verwirft Access die Formatierungen und packt alles wieder in eine einzige Zeile.

Mehr als eine Tabellen-Gültigkeitsregel pro Tabelle

Sie können auch mehr als eine Tabellen-Gültigkeitsregel eintragen. Dazu trennen Sie die einzelnen Gültigkeitsregeln mit dem Operator `Und`:

```
(<Tabellen-Gültigkeitsregel 1>)
Und
(<Tabellen-Gültigkeitsregel 2>)
Und
...
```

Dieses konkrete Beispiel finden Sie in den Materialien zum Buch unter *02_Access_als_Datenbank_Tabellen\2.3.13_Tabellengültigkeitsregel.accdb* in der Tabelle *tblFlugMehrereTabellengültigkeitsregeln*:

```
(
  ([IstVerspaetet]=Falsch)
  Oder
  (
    ([IstVerspaetet]=Wahr)
    Und ([AbflugDatumTatsaechlich] Ist Nicht Null)
    Und ([AbflugZeitTatsaechlich] Ist Nicht Null)
    Und ([AnkunftZeitTatsaechlich] Ist Nicht Null)
    Und
    (
      ([AbflugDatumTatsaechlich]>[AbflugDatumGeplant])
      Oder ([AbflugZeitTatsaechlich]>[AbflugZeitGeplant])
      Oder ([AnkunftZeitTatsaechlich]>[AnkunftZeitGeplant])
    )
  )
)
Und
(
  (
    ([IstGestrichen] = Falsch)
    Und ([IstGestrichenWordenDurch] Ist Null)
    Und ([IstGestrichenWordenAm] Ist Null)
  )
  Oder
  (
    ([IstGestrichen] = Wahr)
    Und ([IstGestrichenWordenDurch] Ist Nicht Null)
    Und ([IstGestrichenWordenAm] Ist Nicht Null)
  )
)
```

Eine Tabellen-Gültigkeitsregel prüfen

Bei umfangreichen Tabellengültigkeitsregeln passieren schnell kleinere Fehler, die nur schwer zu finden sind. Ich empfehle Ihnen, die Tabellen-Gültigkeitsregel entweder ganz oder in Teilen mit Beispieldaten zu testen. Hier ist noch ein Tipp für Profis, die schon Erfahrung im Umgang mit Abfragen haben:

1. Erstellen Sie die Tabelle ohne Tabellen-Gültigkeitsregel.
2. Tragen Sie Testdaten in die Tabelle ein (sowohl konsistente als auch inkonsistente Datensätze!).
3. Erstellen Sie eine neue Abfrage in der SQL-Ansicht.
4. So finden Sie alle konsistenten Datensätze:

```
SELECT *
FROM <Tabellenname>
WHERE (<Tabellen-Gültigkeitsregel>);
```

5. Und so finden Sie alle inkonsistenten Datensätze:

```
SELECT *
FROM <Tabellenname>
WHERE NOT (<Tabellen-Gültigkeitsregel>);
```

Auf diese Weise können Sie die Tabellen-Gültigkeitsregel schrittweise erstellen und überprüfen. Wenn alles richtig ist, löschen Sie die inkonsistenten Datensätze aus der Tabelle und tragen die fertige Formel als Tabellen-Gültigkeitsregel in der Entwurfsansicht der Tabelle ein.

Eine Tabellen-Gültigkeitsregel bezieht sich immer nur auf einen Datensatz

Mit einer Tabellen-Gültigkeitsregel können Sie immer nur jeden einzelnen Datensatz für sich überprüfen. Oder anders gesagt: Sie können in der Formel für die Tabellen-Gültigkeitsregel nicht auf die Werte in anderen Datensätzen zugreifen.

Hier ist ein Beispiel, bei dem eine Tabellen-Gültigkeitsregel *nicht* weiterhilft: Jeder Mitarbeiter in der Tabelle *tblMitarbeiter* bekommt eine Personalnummer, die eindeutig pro Mitarbeiter sein soll. In sämtlichen Datensätzen von *tblMitarbeiter* darf jede Personalnummer höchstens einmal auftreten.

Mit einer Tabellen-Gültigkeitsregel kommen Sie hier nicht weit! Denn schließlich müssten Sie alle bisher vergebenen Personalnummern überprüfen, also alle anderen Datensätze in *tblMitarbeiter*. Ich hatte eingangs erwähnt, dass es zwei Arten von Tabelleneinschränkungen gibt. Für diesen Fall könnten Sie den zweiten Typ, nämlich den *eindeutigen Schlüssel*, einsetzen, den ich Ihnen in Abschnitt 2.9.2, »Eindeutiger Schlüssel (Alternativschlüssel)«, vorstellen werde.

Zusammenfassung

Je nach Art der Einschränkung können Sie also nur auf bestimmte Felder einer Tabelle zugreifen:

Einschränkung	Bezieht sich auf ein einziges Feld	Kann auf alle Felder in einem Datensatz zugreifen	Kann alle Werte der gesamten Tabelle überprüfen
Felddatentyp	•		
EINGABE ERFORDERLICH	•		
Feld-Gültigkeitsregel	•		
Standardwert	•		
Tabellen-Gültigkeitsregel		•	
eindeutiger Schlüssel			•

Tabelle 2.19 Eine Übersicht der verschiedenen Einschränkungen mit Zugriffsbereich

Sie können jede Feld-Gültigkeitsregel als Tabellen-Gültigkeitsregel darstellen. Die Tabellen-Gültigkeitsregel ist somit die allgemeinere Einschränkung verglichen mit einer Feld-Gültigkeitsregel. Beispielsweise wäre für das Feld »Geschlecht« in der Tabelle *tblMitarbeiter* die Feld-Gültigkeitsregel

'M' Oder 'W' Oder 'D'

sinngemäß identisch mit der Tabellen-Gültigkeitsregel:

[Geschlecht]='M' Oder [Geschlecht]='W' Oder [Geschlecht]='D'

Gleichwohl rate ich Ihnen davon ab, denn die Feld-Gültigkeitsregel ist schneller eingetippt. Vor allem aber hängt sie an dem Feld, auf das sie sich bezieht, was für meinen Geschmack die Lesbarkeit und Verständlichkeit ungemein erhöht.

Best Practice für Tabellen-Gültigkeitsregeln

- ▶ Tragen Sie eine Gültigkeitsregel, die sich nur auf ein Feld bezieht, als Feld-Gültigkeitsregel ein.
- ▶ Definieren Sie eine Gültigkeitsregel, die sich auf mehrere Felder bezieht, als Tabellen-Gültigkeitsregel.
- ▶ Tabellen-Gültigkeitsregeln werden recht schnell umfangreich und komplex. Hier gilt es abzuwägen, was unbedingt notwendig ist und welche Regeln nicht so wichtig sind.
- ▶ Bauen Sie umfangreiche Tabellen-Gültigkeitsregeln schrittweise auf.
- ▶ Prüfen Sie Tabellen-Gültigkeitsregeln mit Beispieldaten.
- ▶ Wenn Sie sehr viele Regeln prüfen müssen, kann Programmcode in einem Formular übersichtlicher sein.

2.3.14 Der Nachschlage-Assistent

Neben den besprochenen Felddatentypen führt Access im Dropdown-Menü FELDDATENTYP zusätzlich die Option NACHSCHLAGE-ASSISTENT auf. Dies ist *kein* eigener Felddatentyp, sondern erzeugt ein Textfeld mithilfe eines Assistenten. Der Assistent befüllt letztendlich die Angaben im Registerblatt NACHSCHLAGEN, auf das ich in Abschnitt 2.6.4, »Nachschlagen«, eingehen werde. Da der Assistent weitgehend selbsterklärend ist, werde ich ihn – wie üblich in diesem Buch – nicht näher erläutern.

2.4 Ändern von Tabellen

Im vorherigen Abschnitt habe ich Ihnen gezeigt, wie Sie eine Tabelle in der Entwurfsansicht öffnen und dort die Felder festlegen. Damit kennen Sie bereits das Werkzeug, mit dem Sie eine Tabelle ändern, recht gut: die *Entwurfsansicht*. In diesem Abschnitt werde ich Ihnen zeigen, wie Sie die Felder einer Tabelle (die *Tabellenstruktur*) ändern können, wenn bereits Datensätze in der Tabelle vorhanden sind.

2.4.1 Hinzufügen von Feldern

Ein neues Feld zu einer bestehenden Tabelle hinzuzufügen ist eigentlich eine langweilige Sache:

1. Öffnen Sie die Tabelle in der Entwurfsansicht.
2. Gehen Sie in die leere Zeile nach dem letzten Feld.
3. Legen Sie den Namen des Feldes unter FELDDATENTYP fest.
4. Anschließend wählen Sie unter FELDDATENTYP den passenden Felddatentyp aus.
5. Optional stellen Sie weitere Eigenschaften zum Felddatentyp, zu Feldeinschränkungen und zu Formatierungen im Registerblatt ALLGEMEIN ein.
6. Klicken Sie in der Symbolleiste für den Schnellzugriff auf SPEICHERN (oder drücken Sie Strg + S).

Wenn in Ihrer Tabelle Datensätze vorhanden sind, dann hat es der letzte Schritt aber in sich! In der Datenbankdatei (.accdb- oder .mdb-Datei) werden die Datensätze hintereinander weggespeichert, so wie sie einmal eingegeben wurden. Ein neues Feld bedeutet, dass Access Platz schaffen muss. Und das betrifft alle schon vorhandenen Datensätze (siehe Abbildung 2.32). Gerade bei umfangreichen Tabellen kann das Umkopieren der Daten sehr lange dauern.

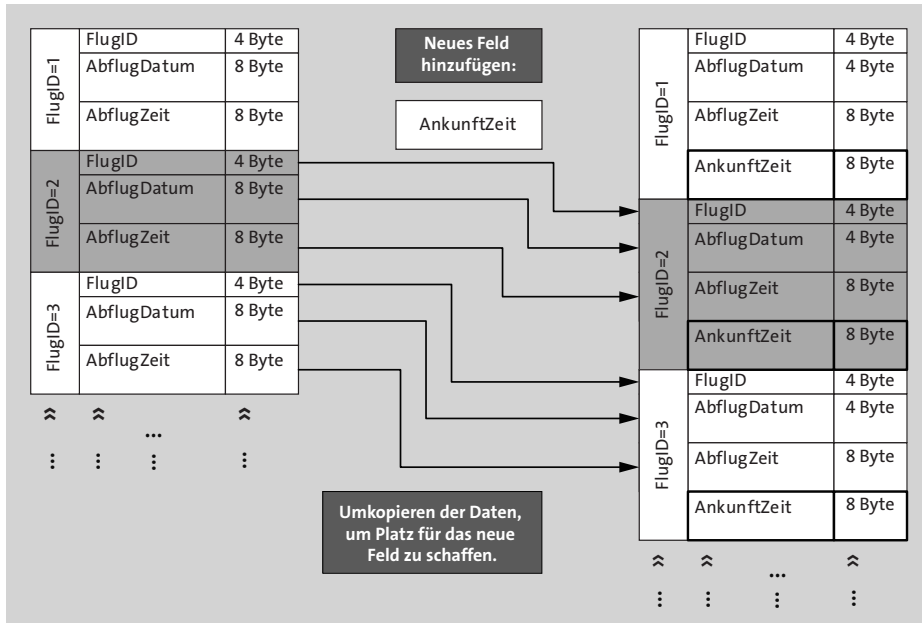


Abbildung 2.32 Wenn Sie ein neues Feld in die Tabelle »tblFlug« einfügen, müssen die bestehenden Datensätze umkopiert werden, um Platz für das neue Feld zu schaffen.

Für bestehende Datensätze bekommt das neue Feld den Wert NULL. Meiner Erfahrung nach erledigt Access das Umkopieren der Daten recht zuverlässig. Ich empfehle Ihnen außerdem, in regelmäßigen Abständen die Datenbank zu komprimieren (DATENBANKTOOLS • DATENBANK KOMPRIMIEREN UND REPARIEREN). Diese Funktion bewirkt, dass alle Tabellen (und andere Datenbankobjekte) umkopiert werden. Dabei werden die Datensätze optimal angeordnet.

2.4.2 Verschieben von Feldern

In der Entwurfsansicht einer Tabelle können Sie die Reihenfolge der Felder verändern.

tblFlug		Feldname	Felddatentyp
		FlugID	AutoWert
		AbflugDatum	Datum/Uhrzeit
		AbflugZeit	Datum/Uhrzeit
		AnkunftZeit	Datum/Uhrzeit
		Fluggesellschaft	Kurzer Text
		Flugnummer	Kurzer Text
1		AbflugFlughafen	Kurzer Text
		AnkunftFlughafen	Kurzer Text
		Passagiere	Kurzer Text

Abbildung 2.33 Felder können Sie markieren, indem Sie auf das leere Feld 1 vor dem Feldnamen klicken.

1. Klicken Sie auf das leere Feld vor dem Feldnamen ❶. Das Feld wird dadurch ausgewählt (siehe Abbildung 2.33).
2. Ziehen Sie das Feld per Drag-and-drop an die gewünschte Position.
3. Klicken Sie in der Symbolleiste für den Schnellzugriff auf **SPEICHERN** (oder drücken Sie **Strg** + **S**).

Auch hier gilt wieder: Sobald Datensätze in der Tabelle vorhanden sind, beginnt Access beim Speichern mit dem Umkopieren der Datensätze. Bei umfangreichen Tabellen kann also eine kleine Drag-and-drop-Operation eine ganze Menge Kopierarbeit erzeugen.

2.4.3 Löschen von Feldern

So entfernen Sie ein Feld, das Sie nicht mehr benötigen:

1. Öffnen Sie die Tabelle in der Entwurfsansicht.
2. Wählen Sie das Feld aus, das entfernt werden soll (klicken Sie dazu das leere Feld vor dem Feldnamen an).
3. Drücken Sie **Entf**.
4. Klicken Sie in der Symbolleiste für den Schnellzugriff auf **SPEICHERN** (oder drücken Sie **Strg** + **S**).

Wenn Datensätze vorhanden sind, gehen die gelöschten Felder unwiederbringlich verloren. Darauf weist die Meldung aus Abbildung 2.34 hin:

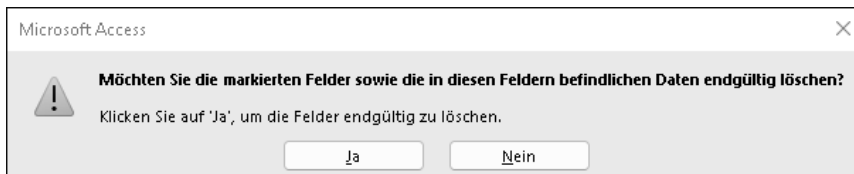


Abbildung 2.34 Wenn Sie die Taste »Entf« drücken, warnt Access, dass die Daten in den gelöschten Feldern verloren gehen werden. Wirklich gelöscht wird das Feld jedoch erst, wenn Sie die Änderungen abspeichern.

Beginnt Access beim Löschen von Feldern auch mit dem Umkopieren bestehender Datensätze, ähnlich wie beim Hinzufügen oder Verschieben von Feldern? Interessanterweise kopiert Access in diesem Fall *nicht* um! Ich habe einmal eine Miniversion unserer Tabelle *tblFlug* mit diesen Feldern eingerichtet:

- ▶ »FlugID«
- ▶ »Flugnummer«
- ▶ »AbflugFlughafen«
- ▶ »AnkunftFlughafen«

Anschließend habe ich drei Datensätze eingetragen und dann das letzte Feld »AnkunftFlughafen« gelöscht. In der Datenblattansicht sieht *tblFlug* jetzt so aus wie in Abbildung 2.35.

FlugID	Flugnummer	AbflugFlughafen
1	LH401	New York
2	AF1919	Frankfurt
3	LH191	Berlin Tegel
*	(Neu)	

Abbildung 2.35 Die Tabelle »tblFlug« mit drei Datensätzen, nachdem ich das letzte Feld »AnkunftFlughafen« gelöscht habe

Lassen Sie uns einen Blick in die Datenbankdatei werfen. Abbildung 2.36 ist ein Ausschnitt aus der Datenbankdatei in hexadezimaler Darstellung und als Text:

```

00053f70 00 00 00 00 00 00 00 00 00 04 00 03 00 00 00 ff .....ÿ
00053f80 fe 4c 48 31 39 31 ff fe 42 65 72 6c 69 6e 20 54 pLH191ÿpBerlin T
00053f90 65 67 65 6c ff fe 4d fc 6e 63 68 65 6e 2400 1b egelypMünchen$.
00053fa0 00 0d 00 06 00 03 00 0f 04 00 02 00 00 00 ff fe .....ÿp
00053fb0 41 46 31 39 31 39 ff fe 46 72 61 6e 6b 66 75 72 AF1919ÿpFrankfur
00053fc0 74 ff fe 50 61 72 69 73 20 00 19 00 0e 00 06 00 tÿpParis .....
00053fd0 03 00 0f 04 00 01 00 00 00 ff fe 4c 48 34 30 31 .....ÿpLH401
00053fe0 ff fe 4e 65 77 20 59 6f 72 6b ff fe 46 72 61 6e ÿpNew YorkÿpFran
00053ff0 6b 66 75 72 74 22 00 17 00 0d 00 06 00 03 00 0f kfurtÿp.....
00054000 04 01 05 0e 51 00 00 00 00 00 00 00 00 00 00 .....Q.....
    
```

Abbildung 2.36 So ist die Tabelle »tblFlug« in der Datenbankdatei gespeichert: Die Inhalte des vierten, mittlerweile gelöschten Feldes »AnkunftFlughafen« sind immer noch vorhanden!

Die etwas kryptischen Zeichen verwendet Access für interne Zwecke, beispielsweise um die Felder zu trennen. Aber Sie sehen sofort: Alle drei Ankunftsflughäfen stehen immer noch in der Datei! Leider gibt es in Access die Funktion »Feld löschen rückgängig machen« nicht; diese Felddaten sind also unbrauchbar.

Beim Löschen eines Feldes geht Access sparsam vor: Es wird kein neuer Platz gebraucht, also wird auch nicht umkopiert. Verlassen würde ich mich darauf aber ehrlich gesagt nicht. Auch an dieser Stelle rate ich Ihnen: Komprimieren Sie Ihre Datenbank immer mal wieder (DATENBANKTOOLS • DATENBANK KOMPRIMIEREN UND REPARIEREN). Diese Funktion entfernt dann auch wirklich die Altlasten aus gelöschten Feldern.

2.4.4 Ändern des Felddatentyps

Sie können den Felddatentyp eines bestehenden Feldes nachträglich ändern:

1. Öffnen Sie die Tabelle in der Entwurfsansicht.
2. Gehen Sie in das Feld, das geändert werden soll.

3. Wählen Sie unter FELDDATENTYP den neuen Felddatentyp aus.
4. Klicken Sie in der Symbolleiste für den Schnellzugriff auf SPEICHERN (oder drücken Sie **Strg** + **S**).

Wie Sie sicherlich schon ahnen, wird Access auch hier mit dem Umkopieren bestehender Datensätze beginnen, nachdem Sie die Änderungen gespeichert haben. Zusätzlich wird Access versuchen, die bestehenden Daten in den neuen Felddatentyp umzuwandeln. Das wird aber nicht immer gelingen! Eine Konvertierung von DATUM/UHRZEIT in Text wird fehlerfrei ablaufen, wenn das Textfeld entsprechend lang ist. Aber umgekehrt sieht es ganz anders aus. Wenn in einem Textfeld kein gültiges Datum steht, schlägt die Konvertierung fehl, und NULL wird als neuer Wert eingetragen (siehe Abbildung 2.37). Ähnliches kann Ihnen beim Konvertieren von Text in ZAHL oder WÄHRUNG passieren.

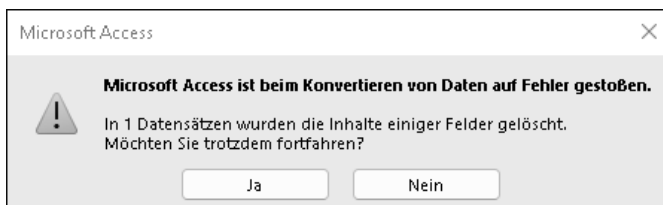


Abbildung 2.37 Beim Ändern des Felddatentyps versucht Access, die Inhalte zu konvertieren. Wenn dies nicht gelingt, wird NULL eingetragen.

2.4.5 Ändern von Feld- und Tabelleneinschränkungen

Auch Feld- und Tabelleneinschränkungen können Sie nachträglich ändern:

1. Öffnen Sie die Tabelle in der Entwurfsansicht.
2. Tragen Sie die neue Tabelleneinschränkung im EIGENSCHAFTENBLATT (Tabelleneigenschaften) unter GÜLTIGKEITSREGEL ein.
3. Gehen Sie in das Feld, das geändert werden soll.
4. Tragen Sie die neue Feldeinschränkung im Registerblatt ALLGEMEIN unter GÜLTIGKEITSREGEL ein.
5. Optional können Sie im Registerblatt ALLGEMEIN weitere Anpassungen vornehmen, beispielsweise EINGABE ERFORDERLICH ändern oder den STANDARDWERT anpassen.
6. Klicken Sie in der Symbolleiste für den Schnellzugriff auf SPEICHERN (oder drücken Sie **Strg** + **S**).

Wenn die Tabelle bereits Daten enthält, können Sie entscheiden, ob Access die Konsistenz der Daten überprüfen soll (siehe Abbildung 2.38).

Ich empfehle Ihnen, diese Überprüfung durchzuführen. Nur so erfahren Sie, ob in der Tabelle inkonsistente Daten vorhanden sind (siehe Abbildung 2.39).

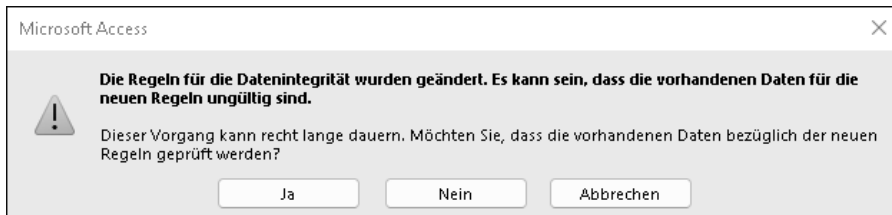


Abbildung 2.38 Nach dem Ändern einer Feld- oder Tabelleneinschränkung sollten Sie überprüfen lassen, ob die bestehenden Daten immer noch konsistent sind.

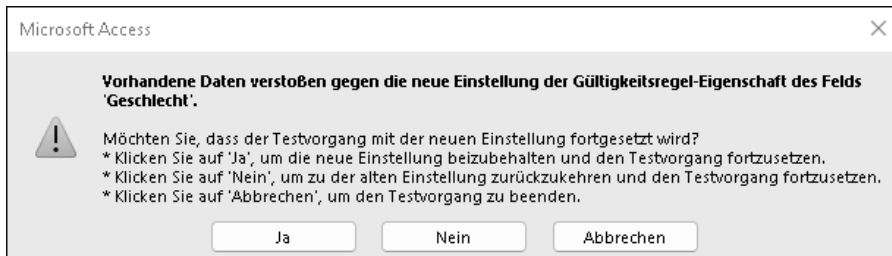


Abbildung 2.39 Access hat einige Datensätze gefunden, die gegen die neue Gültigkeitsregel verstoßen.

Inkonsistente Datensätze werden bei dieser Prüfung von Access weder gelöscht noch verändert. Wenn Sie die Fehlermeldung aus Abbildung 2.39 sehen, müssen Sie also selbst Hand anlegen und die fehlerhaften Datensätze anpassen!

Best Practice beim Ändern von Feld- und Tabelleneinschränkungen

Ich empfehle Ihnen die folgende Vorgehensweise, um inkonsistente Datensätze zu vermeiden:

1. Erstellen Sie eine neue Gültigkeitsregel (z. B. in einem Texteditor).
2. Tragen Sie die neue Gültigkeitsregel in die Entwurfsansicht der Tabelle ein.
3. Prüfen Sie die Datenintegrität.
4. Falls eine Inkonsistenz gemeldet wurde, klicken Sie auf NEIN und verwerfen die neue Gültigkeitsregel zunächst einmal.
5. Suchen Sie jetzt die inkonsistenten Datensätze (entweder manuell oder über eine Abfrage).
6. Passen Sie die inkonsistenten Datensätze so an, dass sie der neuen Gültigkeitsregel entsprechen.
7. Gehen Sie zurück zu Schritt 2.

Inhalt

Vorwort	17
1 Einleitung	21
1.1 Was ist Access?	21
1.1.1 Relationale Datenbank	23
1.1.2 Formulare zum Bearbeiten von Daten	24
1.1.3 Berichte zur Auswertung und zum übersichtlichen Ausdrucken von Daten	25
1.1.4 Programmierung: Optional, zum Beispiel zum Lösen komplexerer Probleme	26
1.1.5 Schnelle Entwicklung kompletter Anwendungsprogramme (»Rapid Development«)	26
1.2 Wie kann eine Datenbank meine Arbeitsprozesse unterstützen?	27
1.2.1 Beispiel einer einfachen Desktop-Datenbank: Mitgliederverwaltung eines Vereins	27
1.2.2 Beispiel einer komplexen Access-Desktop-Datenbank: Erfassung von Mitarbeitern, Flügen und Flugbuchungen	33
1.2.3 Access Web Apps in SharePoint gibt es nicht mehr	37
2 Access als Datenbank: Tabellen	39
2.1 Ein Schnelleinstieg in relationale Datenbanken	39
2.1.1 Eine neue Desktop-Datenbank mit Access erstellen	40
2.1.2 Eine Tabelle in Access erstellen	42
2.1.3 Die Tabelle mit Inhalt füllen	44
2.2 Datenbankdesign	45
2.2.1 Die Welt in Tabellen abbilden	47
2.2.2 Namenskonventionen	49
2.2.3 Der Primärschlüssel	53
2.2.4 Die Tabellenstruktur festlegen	55
2.3 Erstellen von Tabellen	56
2.3.1 Die Entwurfsansicht	56
2.3.2 Der Felddatentyp »Kurzer Text«	62

2.3.3	Der Felddatentyp »Langer Text«	63
2.3.4	Der Felddatentyp »Zahl«	65
2.3.5	Der Felddatentyp »AutoWert«	71
2.3.6	Der Felddatentyp »Datum/Uhrzeit« und »Datum/Uhrzeit erweitert«	73
2.3.7	Der Felddatentyp »Währung«	77
2.3.8	Der Felddatentyp »Ja/Nein«	79
2.3.9	Der Felddatentyp »OLE-Objekt«	79
2.3.10	Andere Felddatentypen	80
2.3.11	Berechnete Felder	81
2.3.12	Feldeinschränkungen	82
2.3.13	Tabelleneinschränkungen	93
2.3.14	Der Nachschlage-Assistent	99
2.4	Ändern von Tabellen	99
2.4.1	Hinzufügen von Feldern	99
2.4.2	Verschieben von Feldern	100
2.4.3	Löschen von Feldern	101
2.4.4	Ändern des Felddatentyps	102
2.4.5	Ändern von Feld- und Tabelleneinschränkungen	103
2.5	Daten in Tabellen	105
2.5.1	Daten eingeben, ändern und löschen	105
2.5.2	Navigationsschaltflächen	109
2.5.3	Die Spaltenbreite eines Feldes anpassen	110
2.5.4	Felder sortieren	110
2.5.5	Felder ein- und ausblenden	110
2.5.6	Felder fixieren	112
2.5.7	Datensätze sortieren	112
2.5.8	Suchfunktionen	116
2.5.9	Datensätze filtern	119
2.6	Formatierungen in Tabellen	128
2.6.1	Format	128
2.6.2	Eingabeformat	143
2.6.3	Beschreibung und Beschriftung	152
2.6.4	Nachschlagen	153
2.7	Beziehungen zwischen Tabellen	155
2.7.1	Die 1:n-Beziehung	155
2.7.2	Die 1:1-Beziehung	159
2.7.3	Die m:n-Beziehung	162
2.7.4	Referenzielle Integrität	165
2.7.5	Übersicht schaffen mit Datenbankdiagrammen	171
2.7.6	Typische Muster von Tabellenbeziehungen	178

2.8	Durch die Normalformen das Datenbankdesign verbessern	186
2.8.1	Die nicht normalisierte Form	187
2.8.2	Die erste Normalform (1NF)	187
2.8.3	Die zweite Normalform (2NF)	190
2.8.4	Die dritte Normalform (3NF)	195
2.8.5	Die Boyce-Codd-Normalform (BCNF)	199
2.8.6	Die vierte Normalform (4NF)	203
2.8.7	Weitere Normalformen	205
2.8.8	Zweckmäßiges Datenbankdesign	207
2.9	Indizes	211
2.9.1	Automatisch generierte Tabellenindizes	211
2.9.2	Eindeutiger Schlüssel (Alternativschlüssel)	214
2.9.3	Erstellen eines Indexes	217
2.9.4	Löschen eines Indexes	220
2.10	Besondere Tabellen	221
2.10.1	Versteckte Datenbankobjekte	221
2.10.2	Systemtabellen (»MSys«) und Usertabellen (»USys«)	222
2.10.3	Das Verzeichnis aller Datenbankobjekte: »MSysObjects«	223

3 Daten filtern, sortieren und zusammenfassen: Abfragen

225

3.1	Einfache Abfragen	225
3.1.1	Vertikales Filtern – Felder auswählen	227
3.1.2	Horizontales Filtern – Datensätze auswählen	232
3.1.3	Der Umgang mit NULL-Werten	241
3.1.4	Mehrere Filterkriterien verwenden	242
3.1.5	Datensätze sortieren	247
3.2	Auswerten von Daten eines Datensatzes: Berechnete Felder	251
3.2.1	Einfache Operationen für Zahlenfelder	252
3.2.2	Aneinanderreihen von Textfeldern	255
3.2.3	Grundsätzliches zu Funktionen	255
3.2.4	Funktionen für Textfelder	257
3.2.5	Funktionen für Zahlenfelder	258
3.2.6	Funktionen für Datumsfelder	260
3.2.7	Konvertieren des Felddatentyps	262
3.2.8	Einfache Fallunterscheidungen	265
3.2.9	Parameter in Abfragen	268

3.3	Zusammenfassen von Datensätzen: Gruppierung und Aggregieren	272
3.3.1	Datensätze zu Gruppen zusammenfassen	273
3.3.2	Die Anzahl der Datensätze pro Gruppe	275
3.3.3	Alle Zahlen einer Gruppe summieren	277
3.3.4	Den Minimal- und Maximalwert aller Zahlen einer Gruppe ermitteln	278
3.3.5	Mittelwert, Varianz und Standardabweichung aller Zahlen einer Gruppe	279
3.3.6	Datensätze vor dem Zusammenfassen filtern – Filter auf die Rohdaten	280
3.3.7	Datensätze nach dem Zusammenfassen filtern – Filter auf das Ergebnis	282
3.3.8	Kreuztabellenabfrage	283
3.4	Abfragen von Daten aus mehreren Tabellen	288
3.4.1	Unterdatenblätter	289
3.4.2	Passende Datensätze zusammenbringen: Abfragen mit INNER-JOIN- Verknüpfung	291
3.4.3	Jeder Datensatz mit jedem gepaart: Das kartesische Produkt	296
3.4.4	Wenn vorhanden, dann nur passende Datensätze zusammen: Abfragen mit OUTER-JOIN-Verknüpfung	297
3.4.5	Abfragen mit mehr als zwei Tabellen	301
3.5	Aktionsabfragen	305
3.5.1	Anfügeabfragen	307
3.5.2	Löschabfragen	311
3.5.3	Aktualisierungsabfragen	314
3.6	Workshops zum Normalisieren von Daten	320
3.6.1	Die Haupttabelle einer 1:n-Beziehung füllen	321
3.6.2	Die Suche nach Inkonsistenzen	325
3.6.3	Inkonsistenzen beheben	329
3.6.4	Dublettensuche	332
4	Access und die Verbindungen zur Außenwelt	333
4.1	Wann braucht man Datenverbindungen?	333
4.1.1	Eine Access-Datenbank als Insel-Lösung verwenden	335
4.1.2	Import und Export von Daten	337
4.1.3	Verbundene Datenquellen	339
4.1.4	Verteilte Datenquellen – »Datawarehouse« und »Datamining«	342
4.2	Export von Daten	343
4.2.1	Export über die Zwischenablage	343
4.2.2	Export in eine andere Access-Datenbank	346
4.2.3	Export nach Excel	347

4.2.4	Export nach Word	349
4.2.5	Export als PDF- oder XPS-Datei	351
4.2.6	Export in eine Textdatei	353
4.2.7	Export in eine HTML-Datei	359
4.2.8	Export in eine dBASE-Datei	361
4.2.9	Export in eine XML-Datei	364
4.2.10	Daten als E-Mail versenden	369
4.2.11	Export als SharePoint-Liste	370
4.2.12	Gespeicherte Exporte	372
4.3	Zugriff auf eine Access-Desktop-Datenbank aus anderen Programmen	376
4.3.1	Öffnen einer Access-Tabelle in Excel	376
4.3.2	Serienbriefe in Word erstellen	381
4.3.3	Zugriff auf eine Access-Datenbank vom Microsoft SQL Server aus	386
4.4	Import von Daten	393
4.4.1	Import aus einer anderen Access-Datenbank	394
4.4.2	Importfehler	396
4.4.3	Import eines Outlook-Ordners	400
4.4.4	Import aus einer XML-Datei	403
4.4.5	Import aus einer dBASE-Datei	407
4.4.6	Daten per E-Mail sammeln	408
4.4.7	Import einer SharePoint-Liste	409
4.4.8	Datendienste nutzen	410
4.4.9	Gespeicherte Importe	411
4.5	Workshops zum Import von Daten	411
4.5.1	Workshop: Einmaliger Import von Daten	412
4.5.2	Workshop: Regelmäßiger Datenabgleich per Import	417
4.5.3	Workshop: Import einer Tabelle mit fehlenden Daten	429
4.6	Anbindung externer Datenquellen: Verknüpfen von Tabellen	461
4.6.1	Eine Tabelle aus einer anderen Access-Datenbank verknüpfen	461
4.6.2	Trennung von Daten und Anwendung	463
4.6.3	Aktualisieren von Tabellenverknüpfungen	466
4.6.4	Andere Dateiformate verknüpfen	468
4.6.5	Eine SharePoint-Liste verknüpfen	470
4.7	Was sind Server-Datenbanken?	471
4.7.1	Vor- und Nachteile gegenüber Access	471
4.7.2	Häufig eingesetzte Datenbank-Server	474
4.7.3	Eine Tabelle über die ODBC-Schnittstelle verknüpfen	475
4.7.4	Optimierung der Client-Server-Anwendung	487
4.7.5	Access-Projekte	488

4.8	Workshops zum SQL Server	489
4.8.1	Workshop: Tabellen im SQL Server erstellen und in Access nutzen	489
4.8.2	Workshop: Datenbankdesign auf dem SQL Server	499
4.8.3	Workshop: Eine bestehende Access-Datenbank auf den SQL Server migrieren	505
5	SQL – die Programmiersprache für Datenbanken	511
5.1	Warum SQL lernen?	511
5.1.1	Voraussetzungen für dieses Kapitel	511
5.1.2	Der grafische Abfrage-Editor und SQL	512
5.2	Daten abfragen: SELECT	512
5.2.1	Abfragen von Daten aus einer Tabelle	514
5.2.2	Datensätze filtern	518
5.2.3	Datensätze sortieren	521
5.2.4	Parameter in Abfragen	522
5.2.5	Berechnete Felder	523
5.2.6	Abfragen von Daten aus mehreren Tabellen	524
5.3	Gruppierung und Aggregieren	531
5.3.1	GROUP BY	531
5.3.2	Aggregatfunktionen	532
5.3.3	HAVING	533
5.4	Unterabfragen	535
5.4.1	In der SELECT-Klausel anstelle eines Tabellenfeldes	535
5.4.2	In der FROM-Klausel anstelle einer Tabelle	538
5.4.3	In der WHERE-Klausel	540
5.4.4	In der ORDER-BY-Klausel anstelle eines Tabellenfeldes	545
5.5	Aktionsabfragen: Data Manipulation Language (DML)	545
5.5.1	INSERT INTO	546
5.5.2	SELECT ... INTO	549
5.5.3	DELETE	550
5.5.4	UPDATE	551
5.6	Datenbankobjekte verändern: Data Definition Language (DDL)	553
5.6.1	CREATE TABLE	554
5.6.2	ALTER TABLE	555
5.6.3	DROP TABLE	555
5.6.4	CREATE INDEX	556
5.6.5	DROP INDEX	556

5.7	Besondere Abfragen	556
5.7.1	UNION-Abfragen	556
5.7.2	Pass-through-Abfragen	559
6	Formulare	563
<hr/>		
6.1	Formulare entwerfen	563
6.1.1	Ein einfaches Formular in der Entwurfsansicht erstellen	563
6.1.2	Datensatzquelle	564
6.1.3	Einen Datensatz in der Formularansicht bearbeiten	566
6.1.4	Formulare als Registerblätter oder als Fenster	566
6.1.5	Welche Ansichten von Formularen gibt es?	568
6.1.6	Navigationsschaltflächen	569
6.1.7	Datensatzmarkierer	573
6.2	Arbeiten mit Steuerelementen	574
6.2.1	Hinzufügen und bearbeiten	574
6.2.2	Anpassen von Größe und Abstand	584
6.2.3	Ausrichten	586
6.2.4	Visuelles Erscheinungsbild	588
6.2.5	Aktivieren und deaktivieren	590
6.2.6	Formatierungen	594
6.2.7	Bedingte Formatierungen	596
6.2.8	Datenbalken	598
6.2.9	Reihenfolge der Aktivierung	599
6.2.10	Gruppierung von Steuerelementen	600
6.3	Die verschiedenen Steuerelemente	601
6.3.1	Welche Steuerelemente gibt es?	601
6.3.2	Steuerelemente für Text	603
6.3.3	Auswahl Ja/Nein	607
6.3.4	Auswahl aus verschiedenen Optionen	610
6.3.5	Auswahl aus einer Liste	622
6.3.6	Grafik-Steuerelemente	629
6.3.7	Diagramme generieren	650
6.3.8	Internetfunktionen	658
6.3.9	Andere Steuerelemente	662
6.4	Mit Layouts in Formularen arbeiten	668
6.4.1	Layout- und Entwurfsansicht	669
6.4.2	Layout hinzufügen	670
6.4.3	Den Layoutbereich anpassen	672

6.4.4	Layout entfernen	674
6.4.5	Layout ähnlich einer Tabelle: Formulkopf und Formulfuß	675
6.4.6	Abstand und Gitternetzlinien	677
6.4.7	Steuerelemente verankern	678
6.5	Formulartypen	681
6.5.1	Das gebundene Standardformular	682
6.5.2	Das ungebundene Standardformular	683
6.5.3	Das Endlosformular	684
6.5.4	Das geteilte Formular	687
6.5.5	Das einfache Popup-Formular	689
6.5.6	Das modale Popup-Formular	690
6.5.7	Nicht mehr unterstützte Formulartypen	692
6.6	Arbeiten mit mehreren Formularen	692
6.6.1	Zugriff auf Daten in anderen Formularen	693
6.6.2	Unterformulare	704
6.6.3	Synchronisieren von Unterformularen	707
6.6.4	Das Navigationssteuerelement	712
6.7	Formulare aus einer Vorlage erstellen	715
6.7.1	Formularvorlagen	715
6.7.2	Anwendungsparts	716
6.8	Workshops zu Formularen	718
6.8.1	Workshop: Formulare und Unterformulare im Zusammenspiel	718
6.8.2	Workshop: Eine individuelle Eingabe-Dialogbox	726
6.8.3	Workshop: Platzreservierung	739
7	Berichte	749
7.1	Berichte entwerfen	749
7.1.1	Einen einfachen Bericht in der Entwurfsansicht erstellen	749
7.1.2	Die Datensatzquelle	751
7.1.3	Steuerelemente in Berichten	753
7.1.4	Unterberichte	758
7.2	Arbeiten mit Layouts in Berichten	763
7.2.1	Layout hinzufügen	763
7.2.2	Unterschiede zu Formularen	764
7.3	Ansichten eines Berichts	764
7.3.1	Die Seitenansicht	766
7.3.2	Direktes Drucken eines Berichts	768

7.3.3	Die Berichtsansicht	769
7.3.4	Einen Bericht als PDF- oder XPS-Datei exportieren	770
7.4	Bereiche in Berichten	771
7.4.1	Kopf- und Fußbereich	771
7.4.2	Ebenen zur Gruppierung	775
7.4.3	Seitenumbrüche setzen	789
7.4.4	Gruppen zusammenhalten und Abstände richtig setzen	790
7.4.5	Sich wiederholende Einträge ausblenden	792
7.4.6	Laufende Summen und Gesamtsummen	794
7.5	Berichte mit mehreren Spalten	796
7.5.1	Etiketten mit dem Assistenten erstellen	796
7.5.2	Mehrere Spalten einrichten	797
7.5.3	Der Seitenkopf für mehrere Spalten	801
7.6	Workshops zu Berichten	804
7.6.1	Workshop: Summen und Prozentsätze	804
7.6.2	Workshop: Filterformulare für Berichte	811
7.6.3	Workshop: Bericht in eine HTML-Datei exportieren	815
8	Einfache Programmierung mit Makros	819
8.1	Makros	819
8.1.1	Aktionen	820
8.1.2	Unsichere Aktionen und vertrauenswürdige Datenbanken	823
8.1.3	Kommentare im Makro	831
8.1.4	Fallunterscheidungen	831
8.1.5	Gruppieren und Untermakros	836
8.2	Auf Ereignisse reagieren	837
8.2.1	Ereignisse von Formularen	838
8.2.2	Ereignisse von Steuerelementen	842
8.2.3	Mit dem Zeitgeber arbeiten (Timer)	844
8.2.4	Ereignisse eines Berichts	844
8.2.5	Eingebettete Makros	846
8.3	Besondere Makros	847
8.3.1	Datenmakros (Trigger)	847
8.3.2	»AutoExec« startet beim Öffnen der Datenbank	852
8.3.3	Tastaturkürzel in »AutoKeys«	852

9	Visual Basic for Applications (VBA), die Programmiersprache für Microsoft-Office-Anwendungen	855
9.1	Warum VBA lernen?	855
9.1.1	Voraussetzungen für dieses Kapitel	856
9.1.2	Die Zukunft von VBA	856
9.2	Formulare bereichern	857
9.2.1	Auf Ereignisse reagieren	857
9.2.2	Kommentare im Code	860
9.2.3	Berechnungen und Formeln	860
9.2.4	Fallunterscheidungen	861
9.2.5	Meldungen am Bildschirm anzeigen: »MsgBox«	864
9.2.6	Befehle, die über mehrere Zeilen reichen	866
9.3	Die integrierte Entwicklungsumgebung	867
9.3.1	Ansicht im Codefenster	867
9.3.2	Mit dem Objektkatalog und dem Projekt-Explorer die Übersicht behalten	870
9.3.3	VBA-Code im Direktbereich ausführen	873
9.3.4	Programme starten	874
9.3.5	Haltepunkte und Debug-Modus	875
9.4	Daten im Speicher verarbeiten	877
9.4.1	Variablen	878
9.4.2	Konstanten	885
9.4.3	Eingaben vom Anwender abfragen: »InputDialog()«	887
9.4.4	Statische Variablen	889
9.4.5	Arrays	889
9.4.6	Dynamische Arrays	890
9.5	Kontrollstrukturen: Verzweigungen und Wiederholungen	891
9.5.1	Bedingungen und Fallunterscheidungen	892
9.5.2	Fallauswahl	893
9.5.3	Schleifen	893
9.5.4	Iteration	898
9.5.5	Sprungbefehle und Sprungmarken	898
9.6	Module	899
9.6.1	Ein Modul erstellen und bearbeiten	900
9.6.2	Prozeduren	900
9.6.3	Funktionen	903
9.6.4	Parameter	911
9.6.5	Gültigkeitsbereiche	921
9.6.6	Variablen auf Modulebene und globale Variablen	925

9.7	Klassenmodule	927
9.7.1	Ein eigenständiges Klassenmodul erstellen und bearbeiten	929
9.7.2	Objektvariablen	933
9.7.3	Eigenschaften (Propertyts)	940
9.7.4	Der Befehl »With«	947
9.7.5	Auflistungen (Collections)	949
9.8	Fehlerbehandlung	956
9.8.1	Auf Laufzeitfehler reagieren	956
9.8.2	Die Behandlung von Fehlern	961
9.8.3	Eigene Laufzeitfehler auslösen	962
9.8.4	Zentrale Fehlerbehandlung	963
10	Die Data-Access-Objects-(DAO-)Klassenbibliothek	965
<hr/>		
10.1	Daten abfragen: »Recordset«-Objekte	966
10.1.1	Lesend auf eine Tabelle zugreifen	966
10.1.2	Einen Datensatz suchen	971
10.1.3	Domänenfunktionen – Datenabfrage in einer Zeile	972
10.1.4	Datensätze verändern	974
10.1.5	Abfragen erstellen und als Datensatzquelle nutzen	975
10.2	Aktionsabfragen ausführen	978
10.2.1	Bestehende Aktionsabfragen aufrufen	978
10.2.2	SQL-Code direkt ausführen	980
10.3	Transaktionen	980
10.3.1	Eine Transaktion starten und abschließen	981
10.3.2	Änderungen doch nicht durchführen: Rollback	983
10.4	Andere Klassenbibliotheken	984
10.4.1	Referenzen definieren	985
10.4.2	Word-Dokumente bearbeiten	986
10.4.3	Daten aus einer Excel-Tabelle gezielt auslesen	988
10.4.4	ActiveX Data Objects (ADO)	989
10.4.5	Zugriff auf das Windows-Betriebssystem	990
11	Anwendungsprogramme mit Access erstellen	993
<hr/>		
11.1	Wichtige Elemente eines Anwendungsprogramms	994
11.1.1	Start der Anwendung	994

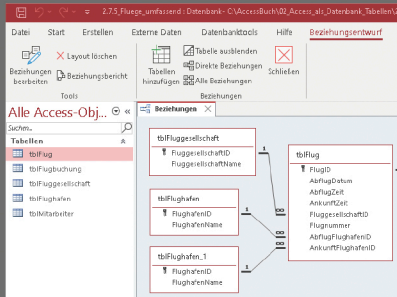
11.1.2	Formen des Hauptmenüs	995
11.1.3	Das Menüband	998
11.2	Zugriffsberechtigungen	1005
11.2.1	Benutzer, Gruppen und Zugriffsberechtigungen in ».mdb«-Dateien	1005
11.2.2	Das Verschlüsseln der Datenbank	1014
11.2.3	Erstellen einer ».accde«-Datei	1015
11.3	Ohne Access mit Datenbank-Dateien arbeiten: Die Access Runtime	1016
11.3.1	Installation der Access Runtime	1017
11.3.2	Workshop: Eine Datenbank Runtime-fähig machen	1019
11.3.3	Workshop: Anwendungsprogramm mit Backend-Datenbank	1025
11.4	Die Grenzen der Möglichkeiten in Access – Beyond the Limits	1030

Anhang 1035

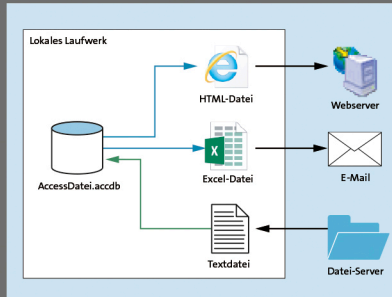
A.1	Microsoft SQL Server	1035
A.1.1	Installationsdateien herunterladen	1035
A.1.2	Die SQL Server Express Edition installieren	1037
A.1.3	Arbeiten nach der Installation	1046
A.1.4	SQL Server Management Studio: Das universelle Werkzeug zur Verwaltung des SQL Servers	1052
A.1.5	Eine Datenbank im SQL Server erstellen	1058
A.1.6	Datenbank trennen und wieder anfügen	1062
A.1.7	Datensicherung einer Datenbank des SQL Servers	1065
A.1.8	Authentifizierung und Berechtigungen	1066
Index		1077

So planen und entwickeln Sie Datenbanken richtig!

Erfahren Sie, wie Sie in Access Datenbanken konzipieren und schrittweise aufbauen. Nach einer Einführung in gutes Datenbankdesign macht Sie Wolfram Langer mit Tabellen, Abfragen, Formularen und Berichten vertraut. Sie lernen alle wichtigen Techniken kennen: Von der Abfrageoptimierung über die VBA- und DAO-Programmierung bis zur Entwicklung von Datenbankanwendungen.



Datenbanken richtig planen



Effizient mit Access arbeiten



Eigene Formulare erstellen

Grundlagen kennen, Access optimal einrichten

Was ist eine relationale Datenbank? Welche Einstellungen müssen Sie nach der Installation in Access vornehmen? Was steckt hinter Begriffen wie Abfrage, Formular oder 1:n? Das Buch beantwortet Ihnen alle Fragen!

Access von A bis Z verstehen

Lernen Sie die Grundlagen für gutes Datenbankdesign kennen und nutzen Sie zahlreiche Praxistipps für Ihre eigenen Access-Projekte. Schnell wenden Sie das neue Wissen über Tabellen und deren Beziehungen, Abfragen, Datenimport und -export, Benutzereingaben, Formulare und Berichte sicher an.

Programmierung und fortgeschrittene Techniken

SQL, Makros, VBA-Programmierung: Nutzen Sie Access voll aus und automatisieren Sie Ihre Arbeitsprozesse. Mit den Expertentipps realisieren Sie Ihre individuellen Anforderungen und erstellen selbst Datenbankanwendungen mit Anbindung an einen SQL-Server. Dieses Buch macht Sie zum Access-Profi!



Alle Codebeispiele und Beispieldatenbanken zum Download



Wolfram Langer ist Geschäftsführer der Launhardt Langer IT-Services GmbH. Microsoft Access setzt er in vielen Kundenprojekten produktiv ein. Er hat umfassende Datenbankprojekte für die Automobil-, Pharma- und Luftfahrtbranche realisiert.

Aus dem Inhalt

Wichtige Grundlagen

- Datenbanktypen
- Datenbankdesign
- Formulare erstellen
- Tabellen erstellen und ändern
- Feldtypen und Beziehungen

Access im Praxiseinsatz

- Abfragen erstellen
- Datenverbindungen
- Import und Export von Daten
- SharePoint-Listen einbinden
- Workshops und Best Practices

Fortgeschrittene Techniken

- Server-Datenbanken
- SQL lernen
- Makro- und VBA-Programmierung
- DAO-Klassenbibliothek
- Access-Anwendungsprogramme

