

# Keine Angst vor Microsoft Access!

## - Was ist wichtig? -

Die Seitenangaben beziehen sich auf die sechste Auflage des Buches. Leser älterer Auflagen müssten bitte etwas hin und her blättern, um die richtige Seite zu finden.

### Kapitel 1: „Ein Blick hinter die Kulissen“

1. Stellen Sie die Verweise auf die benötigten Bibliotheken ein (siehe Abbildung 1.2 auf Seite 21).
2. Eine Datenbankanwendung besteht immer aus den Komponenten GUI und DBMS, die miteinander SQL »reden« (siehe Abbildung 1.4 auf Seite 24).
3. Stellen Sie die Makrosicherheitsstufe und die vertrauenswürdigen Speicherorte ein (siehe Anweisungen im ersten Fenster, das nach dem Öffnen der Beispieldatenbank angezeigt wird).
4. Drucken Sie sich die Dateien Namenskonventionen.pdf, WichtigeStandardaktionen.pdf und WasIstWichtig.pdf aus dem Ordner \Buch\KapA (Link s. Einleitung!) aus.
5. Ein Primärschlüssel ist eine Eigenschaft eines Objekts, die es für diesen Objekttyp (d.h. in dieser Tabelle) nur ein einziges Mal gibt; er hat den Datentyp Autowert (siehe Abschnitt »Tabellen, Felder und Primärschlüssel« ab Seite 40).
6. Ein Fremdschlüssel ist eine Eigenschaft eines Objekts, die in einer anderen Tabelle Primärschlüssel ist; er hat den Datentyp Zahl/LongInteger (siehe Abschnitt »Beziehungen und Fremdschlüssel« ab Seite 48).
7. Lesen Sie die Beziehungen immer in Form von zwei Sätzen, die jeweils mit »Ein ...« beginnen (siehe Abschnitt »Beziehungen und Fremdschlüssel« ab Seite 48).
8. Ein Formular »kennt« nur diejenigen Daten, die ihm über die Eigenschaft »Datensatzquelle bekannt gemacht wurden (siehe Abschnitt »Datensatzquelle« ab Seite 54).
9. Ein Unterformular ist mit dem übergeordneten Formular über die Eigenschaften Verknüpfen von« und »Verknüpfen nach« verbunden (siehe Abschnitt »Unterformulare« ab Seite 56).
10. Kombinations- und Listenfelder beziehen ihren Inhalt aus der Datensatzherkunft und schreiben den ausgewählten Wert in den Steuerelementinhalt (siehe Abschnitt »Datensatzherkunft und Steuerelementinhalt« ab Seite 59).
11. Windows ist ein ereignisgesteuertes System, das auf die vom Benutzer ausgelösten Ereignisse (z.B. Mausklick) mit bestimmten Aktionen reagiert (z. B. Kombinationsfeld öffnen) (siehe Abschnitt »VBA-Prozeduren« ab Seite 63).

## Kapitel 2: „Datenmodellierung“

1. Erstellen Sie eine strukturierte verbale Beschreibung Ihrer Problematik in Form von Substantiv – Verb – Substantiv – Sätzen wie »Kunden erteilen Aufträge«, »Mitglieder sind in Mannschaften« und »Leser leihen Bücher aus«! (siehe Abschnitt »Die strukturierte Szenario-Beschreibung« ab Seite 86)
2. Erstellen Sie nach dem Grundsatz »Substantive werden zu Entitäten – Verben werden zu Beziehungen« einen ersten Entwurf Ihres Datenmodells! (siehe Abschnitt »... und seine Erstellung« ab Seite 74)
3. Überlegen Sie sorgfältig, ob zu EINER Entität A keine, eine oder mehrere Entitäten B gehören (Kardinalitäten)! Lösen Sie sich dabei vom Alltagsdenken. Jede unnötige (1,1)- oder (1,n)-Kardinalität schränkt später die Arbeitsmöglichkeiten mit der Datenbank ein! (siehe Abschnitt »Kunden ohne Aufträge?« ab Seite 77)
4. Beachten Sie den Unterschied zwischen Entität und Entitätsart (z.B. Material und Materialart)! (siehe Abschnitt »Material vs. Materialart« ab Seite 82)
5. Überlegen Sie für jede Beziehung, ob Sie dafür Zeiten bzw. Mengen in Zwischentabellen festhalten wollen! (siehe Abschnitt »Mengen und Zeiten in Zwischentabellen« ab Seite 83)
6. Überführen Sie das logische Datenmodell ins physische Datenmodell entsprechend den beiden Regeln für 1:n- und m:n-Beziehungen! (siehe Abschnitt »Die Überführung in das physische Modell« ab Seite 87)
7. Überlegen Sie genau, ob eine 1:1-Beziehung wirklich sein muss (Im Allgemeinen ist das NICHT der Fall!). Wenn ja, modellieren Sie sie richtig! (siehe Abschnitt »1:1-Beziehungen« ab Seite 99)
8. Modellieren Sie gegebenenfalls Dreifachbeziehungen! (siehe Abschnitt »Dreifachbeziehungen« ab Seite 102)
9. Sehen Sie spezielle Tabellen für Typ, Art und Status vor! (siehe Abschnitt »Typ, Art, Status« ab Seite 105)
10. Vergewissern Sie ich immer mal wieder, wie alles zusammenhängt! (S. 111)
11. Benutzen Sie ADaMo zur Erstellung Ihrer Datenbank-Anwendung! (siehe Abschnitt »ADaMo« ab Seite 120)

## Kapitel 3: „Erste Formulare“

1. Aktivieren Sie die Optionen „Überlappende Fenster“ und „Beim Schließen komprimieren“! (siehe Abschnitt »Tabellen anlegen« ab Seite 124)
2. Arbeiten Sie zunächst mit dem 2003'er Datenbankformat (\*.mdb)! (siehe Abschnitt »Tabellen anlegen« ab Seite 124)
3. Primärschlüssel haben den Datentyp AutoWert, Fremdschlüssel Zahl/LongInteger! Entfernen Sie den Standardwert 0 bei Fremdschlüsseln! (siehe Abschnitt »Tabellen anlegen« ab Seite 124)
4. Schalten Sie in allen Beziehungen die referentielle Integrität ein! Seien Sie vorsichtig mit der Löschweitergabe! (siehe Abschnitt »Referentielle Integrität« ab Seite 128)
5. (1,1)- und (1,n)-Kardinalitäten werden in Access realisiert, indem man beim Fremdschlüssel einstellt Eingabe erforderlich = Ja! (siehe Abschnitt »Kardinalitäten« ab Seite 130)
6. Arbeiten Sie mit Versionen Ihrer Access-Anwendung! (siehe Abschnitt »Versionen« ab Seite 131)
7. Beachten Sie die erforderliche Reihenfolge bei der Eingabe von Testdaten! (siehe Abschnitt »Auf die Reihenfolge achten« ab Seite 132)
8. Benutzen Sie den Formular-Assistenten für den ersten Formularentwurf! Beachten Sie bei dessen weiterer Bearbeitung den Grundsatz: Erst richtig, dann schön! (siehe Abschnitt »Schritt 1: Mit dem Formular-Assistenten Formulare erzeugen« ab Seite 135)
9. Verwenden Sie keine Nachschlagefelder! (siehe Abschnitt »Nachschlagefelder« ab Seite 134)
10. Klicken Sie beim Listefeld-Assistenten die dritte Option an! (siehe Abschnitt »Schritt 3: Listefeld hinzufügen« ab Seite 138)
11. Der Formular-Assistent verbindet automatisch das Unterformular mit dem übergeordneten Formular über den Primärschlüssel-Fremdschlüssel-Mechanismus! (siehe Abschnitt »Ein Formular mit Unterformular« ab Seite 146)
12. Benutzen Sie Kombinationsfelder zur Realisierung der Einzelzuordnung! (siehe Abschnitt »Schritt 4: Aktive Bedienelemente hinzufügen« ab Seite 154)
13. Vorsicht bei der nachträglichen Umbenennung von Steuerelementen, die mit dem Assistenten angelegt wurden! (siehe Abschnitt »Schritt 3: Listefeld hinzufügen« ab Seite 138)
14. Es gibt nur zwei Grundbausteine, aus denen letzten Endes alle Formulare aufgebaut sind! (siehe Abschnitt »Die beiden Grundbausteine« ab S. 157 und insbesondere die GANZ WICHTIGE Abbildung 3.20!!)
15. Benutzen Sie die Checkliste-Formulare zur Perfektionierung Ihrer Formulare! (im Ordner \Buch\KapA – Link s. Einleitung)

## Kapitel 4: „Daten für die Datenbank“

1. Daten in Textdateien bzw. in Excel-Listen haben eine ganz andere logische Struktur als die Daten in der Access-Datenbank. (siehe Abschnitt »Die Ausgangssituation« ab Seite 164)
2. Bereiten Sie die zu importierenden Daten in Excel so auf, dass sie 1-zu-1 in die Zieltabellen passen, d.h. erzeugen Sie in Excel genau dieselben Spalten in genau der-selben Reihenfolge wie in den Datenbanktabellen! (siehe Abschnitt »Generierung von Testdaten« ab Seite 171)
3. Setzen Sie dabei Excel-Textbearbeitungsfunktionen wie LINKS, RECHTS, LÄNGE und FINDEN ein! (siehe Abschnitt »Excel-Funktionen zur Textbearbeitung« ab Seite 169)
4. Machen Sie sich mit den Techniken zum schnellen Markieren, Kopieren und Formel erstellen in sehr langen Spalten vertraut! (siehe Abschnitt »Datenorganisation: Datenmenge« ab Seite 177)
5. Benutzen Sie meine Excel-Datei „Excel-Vorlage-AS.xlsm“ mit den darin enthaltenen Formularen (siehe Abschnitt »Datenarten« ab Seite 166)
  - a. zur Generierung von Zufallsdaten,
  - b. zur Generierung von zufälligen Buchstabenfolgen,
  - c. zur Bereinigung von Listen mit Leerzellen  
bzw. (siehe Abschnitt »Datenarten« ab Seite 169)
  - d. das in allen Beispieldatenbanken enthaltene Formular frmSpaltenFuellen, um einzelne Tabellenspalten mit Daten zu füllen!
6. Setzen Sie die Excel-Funktionen BEREICH.VERSCHIEBEN und SVERWEIS ein, um Daten von einem Excel-Blatt auf ein anderes zu kopieren bzw. um Bezüge zu Daten auf anderen Excel-Blättern herzustellen (Generierung von Fremdschlüsseln!). (siehe Abschnitt »Generierung von Testdaten« ab Seite 171)
7. Überprüfen Sie Echtdaten vor ihrem Import in die Datenbank auf ihre Datenqualität! (siehe Abschnitt »Datenqualität« ab Seite 179)

## Kapitel 5: „VBA – Teil 1“

1. Finden Sie Syntaxfehler mit dem Debugger, beugen Sie Laufzeitfehlern durch entsprechende Programmierung vor und testen Sie Ihren Code mit Beispieldaten auf logische Fehler! (siehe Abschnitt »Fehler finden und korrigieren« und Abschnitt »Laufzeitfehler verhindern« ab Seite 224)
2. Wenn Ihr Programm nicht mehr auf Mausklicks reagiert, könnte es sein, dass der Debugger läuft! (siehe Abschnitt »Laufzeitfehler« ab Seite 198)
3. Tippen Sie die erste Zeile einer Prozedur niemals selber ein, sondern lassen Sie sie von Access bzw. VBA automatisch generieren! (siehe Abschnitt »Das Drumherum« ab Seite 209)
4. Finden Sie hartnäckige Programmierfehler durch schrittweise Programmausführung bzw. durch das Setzen von Haltepunkten! (siehe Abschnitt »Der Debugger« ab Seite 207)
5. Passen Sie die VBA-Symbolleiste an, indem sie ihr Symbole für häufig benötigte Befehle hinzufügen! (siehe Abschnitt »Symbolleiste anpassen« ab Seite 208)
6. Zwingen Sie sich durch die Einstellung Option Explicit immer selber zur expliziten Variablendeklaration! (siehe Abschnitt »Das Drumherum« ab Seite 209)
7. Benutzen Sie beim Testen Ihres Programms die Messagebox für die Ausgabe von Zwischenergebnissen! (siehe Abschnitt »MsgBox« ab Seite 220)
8. Sehen Sie mit OnError auch Reaktionen auf nicht vorhersehbare Fehler vor und schalten Sie diese Fehlerbehandlung erst nach der Übergabe der Access-Anwendung an den Benutzer ein! (siehe Abschnitt »OnError« ab Seite 227)
9. Benutzen Sie meine Standardlösungen aus Checkliste-Formulare.xlsx (im Ordner \Buch\KapA – Link in der Einleitung)

## Kapitel 6: „Steuerelemente“

1. Aktivieren Sie den Steuerelement-Assistenten! (siehe Abschnitt »Eigenschaften von Steuerelementen« ab Seite 232)
2. Die wichtigsten Steuerelemente sind Bezeichnungsfelder, Textfelder, Schaltflächen, Listenfelder und Kombinationsfelder. Zusätzlich nützlich sind auch Optionsgruppen und Register.
3. Ein Formular kennt nicht automatisch alle Tabellen und ihre Spalten. Sie müssen sie ihm über die Eigenschaft Datensatzquelle bekannt machen! (siehe Abschnitt »Textfeld« ab Seite 234)
4. Beachten Sie die Unterschiede in der Programmierung von gebundenen und ungebundenen Formularen und Steuerelementen! (siehe Abschnitt »Verwenden der Assistenten ...« ab Seite 245)
5. Halten Sie sich bei der Namensgebung für Steuerelemente an die Namenskonvention! (siehe Namenskonvention.pdf im Ordner \Buch\KapA – Link in der Einleitung)
6. Benutzen Sie für Ihre ersten Übungen das 2003er Datenbankformat (.mdb), weil die Steuerelement-Assistenten dann VBA-Prozeduren statt eingebettete Makros erzeugen! (siehe Abschnitt »Programmierung der Schaltfläche« ab Seite 241)
7. Wenn eines Ihrer Steuerelemente absolut nicht tut, was es soll, dann ist oft eine nachträgliche Namensänderung der Grund dafür! (siehe Abschnitt »Nachträgliche Namensänderung« ab Seite 241)
8. Mit den Einstellungen Enabled, Visible und Locked können Sie einstellen, dass der Benutzer bestimmte Steuerelemente in bestimmten Situationen nicht bedienen darf! (siehe Abschnitt »Enabled, Visible und Locked« ab Seite 252)
9. Access bietet Ihnen eine Reihe von Hilfestellungen zur exakten Platzierung von Steuerelementen auf dem Formular! (siehe Abschnitt »Exakte Platzierung« ab Seite 253)
10. Machen Sie sich mit den Grundlagen von XML vertraut und programmieren Sie damit ein eigenes Menüband! (siehe Abschnitt »Das eigene Menüband« ab Seite 255)

## Kapitel 7: „SQL“

1. Die GUI (Graphical User Interface) spricht SQL mit dem DBMS (Data Base Management System). (siehe Abschnitt »Wofür SQL?« ab Seite 266)
2. Sie als Access-Entwickler brauchen den SQL-Befehl SELECT FROM für die Definition von Datensatzquellen in Formularen und für die Formulierung von Abfragen (siehe Abschnitt »SELECT« ab Seite 272)
3. Eine Sonderform des SELECT-Befehls ist der OUTER JOIN, den Sie immer dann benötigen, wenn den Zeilen einer Tabelle auch keine Zeilen einer anderen Tabelle zugeordnet sein können (siehe Abschnitt »Outer Join« ab Seite 278)
4. Auf moderner Hardware kann Access Tabellen mit ca. 50.000 Datensätzen (Stand Anfang 2019!) auch noch mit sehr großen Joins relativ zügig bearbeiten (siehe Abschnitt »Schnell soll es gehen!« ab Seite 284)
5. Der SELECT-Befehl kann noch sehr viel mehr als hier dargestellt wurde. Sie sollten sich daher selbst mit dieser Thematik beschäftigen und auch Ihre Kunden dazu anhalten! (siehe Abschnitt »Abfragen!« ab Seite 289)
6. Betrachten Sie die im Abfrage-Generator dargestellten Beziehungen nur als ein Angebot von Access. Gegebenenfalls müssen Sie einige davon für die Erstellung einer Abfrage im Abfrage-Generator löschen! (siehe Abschnitt »Beziehungen in Abfragen« ab Seite 290)

## Kapitel 8: „VBA – Teil 2“

1. In ungebundenen Formularen müssen Sie durch geeignete VBA-Programmierung selber dafür sorgen, dass die richtigen Daten angezeigt und bearbeitet werden. (siehe Abschnitt »Gebundene und ungebundene Formulare« ab Seite 297)
2. Für die Kommunikation zwischen GUI und DBMS können Sie Recordsets oder SQL benutzen. (siehe Abschnitt »»Handgeschnitzte« Kommunikation« ab Seite 300)
3. Ein Recordset (»Datensatzgruppe«) ist eine temporäre Kopie eines Teils der Daten aus der Datenbank. (siehe Abschnitt »Recordsets« ab Seite 300)
4. Benutzen Sie den Debug.Print-Befehl, um SELECT-Befehle von VBA aus in das Direktfenster zu schreiben! Von dort kann der SELECT-Befehl in den Abfrageentwurf kopiert und auf richtige Funktion getestet werden. (siehe Abschnitt »Direktzugriff mit SQL« ab Seite 302)
5. Ersetzen Sie die von Access automatisch erzeugten eingebetteten Makros durch VBA-Code! (siehe Abschnitt »Eingebettetes Makro vs. Ereignisprozedur« ab Seite 303)
6. Bedenken Sie beim Programmieren des Speicherns immer, dass Access auch implizit speichert. Das kann zu ungewollten Nebenwirkungen führen! (siehe Abschnitt »Implizites Speichern« ab Seite 305)
7. Alle Objekte (Steuerelemente), die Sie auf Formularen platzieren können, haben eine Fülle an Eigenschaften und Methoden. Darüber können Sie sich im Objektkatalog einen Überblick verschaffen. (siehe Abschnitt »Exkurs: Das Objektmodell« ab Seite 309)
8. Bedenken Sie beim Löschen von Daten, dass es fast immer erforderlich ist, zuerst noch andere Daten zu löschen, die mit den zu löschenden Daten über Fremdschlüssel verbunden sind! (siehe Abschnitt »Löschen von Werten in Tabellen« ab Seite 314)
9. Benutzen Sie in Ihrer eigenen Anwendung ausgiebig die hier von mir vorgestellten Standardlösungen! (siehe Abschnitt »Standardlösungen« ab Seite 317)

## Kapitel 9: „Formulargrundtypen“

1. Die Tabellen, deren Daten im Formular dargestellt werden sollen, können auf mehr oder weniger komplexe Weise über Beziehungen zusammenhängen. Der Grundbaustein ist aber immer eine 1:n-Beziehung zwischen zwei Tabellen.
2. Benutzen Sie das physische Datenmodell, um sich dieses Beziehungsgeflecht klarzumachen und basierend darauf das Formularlayout zu entwerfen.
3. Bei der Überlegung, welche Steuerelemente auf dem Formular eingesetzt werden sollen, ist immer zu unterscheiden, aus welcher Richtung man auf eine einzelne 1:n-Beziehung blickt. (siehe auch »Die beiden Grundbausteine« auf den Seiten 157/158!!)
4. Verfahren Sie bei der Umsetzung Ihres Entwurfs nach dem Grundsatz »Erst richtig, dann schön!«, sonst haben Sie nachher viel Mühe ins schöne Design gesteckt, aber es funktioniert nicht.
5. Widerstehen Sie der Versuchung, ein Superformular zu entwickeln, das alle Daten Ihrer Datenbank enthält! Sinnvoller sind separate Formulare für einzelne abgegrenzte Arbeitsaufgaben.
6. Stellen Sie einen Zusammenhang zwischen all Ihren Formularen her über Buttons, die direkt von einem Formular aus ein anderes Formular öffnen oder über ein zentrales Startformular oder über ein selbst programmiertes Menüband!
7. Werfen Sie ein Formular, das nicht richtig funktioniert, komplett weg und entwickeln Sie es neu. Das ist oft weniger aufwendig, als etwas Verpfushtes zu reparieren!
8. Benutzen Sie die in diesem Kapitel beschriebenen Formularlayouts als Denkanstoß für Ihre eigene Entwicklung!



## Kapitel 10: „Datenbankanwendungen“

1. Orientieren Sie sich bei Ihrer eigenen Datenbankentwicklung an der von mir beschriebenen Vorgehensweise! (siehe Abschnitt »Vorgehensweise« ab Seite 358)
2. Arbeiten Sie intensiv mit Versionen Ihrer Anwendung – fortlaufende Nummern für Versionen, die Sie an Andere herausgeben (»01«, »02«, ...) und Kleinbuchstaben dahinter für Ihre eigenen Entwicklerversionen (»02a«, »02b«). (siehe Abschnitt »Versionen« ab Seite 359)
3. Teilen Sie Ihre Anwendung vor der ersten Übergabe an den Auftraggeber (unter Umständen Sie selbst!) in Front- und Backend auf! (siehe Abschnitt »Aufteilung in Frontend und Backend« ab Seite 361)
4. Nutzen Sie die von mir beschriebene Technik des frontendgesteuerten Backend-Updates über eine Tabelle mit den dafür erforderlichen SQL-Befehlen! (siehe Abschnitt »Frontendgesteuertes Backend-Update« ab Seite 369)
5. Legen Sie in jeder Ihrer Datenbanken eine INFO-Tabelle an, in der Sie z.B. die Versionsnummer speichern! (siehe Abschnitt »Welche Version ist es denn?« ab Seite 367)
6. Schützen Sie Ihre Anwendungen mit den von mir beschriebenen Techniken vor unbefugtem Zugriff auf die Daten bzw. den Code! (siehe Abschnitt »Schutz der Datenbank« ab Seite 371)
7. Geben Sie dem Benutzer, der kein Access auf seinem Computer installiert hat, die Access-Runtime-Version, damit er mit Ihrer Anwendung arbeiten kann! (siehe Abschnitt »Access ohne Access?« ab Seite 374)
8. Informieren Sie sich über dieses Buch hinaus zu den Themen »Abfragen«, »Berichte«, »Makros«, »Schreiben und lesen von Textdateien«, »Benutzerberechtigungen« und »Mehrbenutzer-Zugriff«! (siehe Abschnitt »Wie geht's weiter?« ab Seite 377)