



Was ist eine Datenbank?

In einer Datenbank (wie z. B. Access) werden Daten in Form von **Tabellen** organisiert. Jede Tabelle besteht aus Zeilen und Spalten, wobei jede Zeile einen **Datensatz** repräsentiert und jede Spalte ein **Feld** bzw. Attribut darstellt.

Beispiel: Eine Datenbanktabelle für Schüler

Die Tabelle könnte folgende Spalten enthalten:

Schüler-ID: Eine **eindeutige** Nummer, die jeden Schüler identifiziert.

Vorname: Der Vorname des Schülers.

Nachname: Der Nachname des Schülers.

Geburtsdatum: Das Geburtsdatum des Schülers.

Geschlecht: Das Geschlecht des Schülers (männlich oder weiblich).

Klasse: Die aktuelle Klasse, in der der Schüler sich befindet.

Adresse: Die Adresse des Schülers.

Jede Zeile in der Tabelle repräsentiert einen einzelnen Schüler:

Schüler-ID	Vorname	Nachname	Geburtsdatum	Geschlecht	Klasse	Adresse
1	Max	Mustermann	15.03.2015	männlich	4a	Musterstraße 123
2	Lisa	Müller	07.02.2016	weiblich	4b	Beispielweg 456
3	Alex	Schmidt	19.11.2014	männlich	4a	Testgasse 789

In dieser Tabelle werden die Daten von drei Schülern gespeichert.

Jeder Schüler hat eine eindeutige Schüler-ID. Die zugehörigen Informationen wie Vorname, Nachname, Geburtsdatum, Geschlecht, Klasse und Adresse werden in den entsprechenden Spalten aufgeführt.

Diese Tabelle kann in Access erstellt und verwaltet werden, um Schülerdaten effizient zu speichern und darauf zuzugreifen.

Datenbanken bestehen immer aus mehreren Tabellen!

Eine Datenbank enthält im Normalfall viele Tabellen, die untereinander in Beziehung stehen bzw. miteinander verknüpft sind: Eine Datenbank für eine Schule wird in der Praxis zusätzlich Tabellen enthalten wie Klassen, LehrerInnen, Räume, Schulstunden, Schulfächer, Zeugnisse ...

Warum werden die Daten in mehreren Tabellen gespeichert?

- Redundanz (mehrfach gespeicherte Informationen) wird vermieden.
- Inkonsistenzen (Fehlerhafte und widersprüchliche Daten) werden vermieden.
- Speicherplatz wird gespart.

Ein einfaches Beispiel soll zeigen, warum die Speicherung von Daten in mehreren Tabellen sinnvoll ist:

Nehmen wir eine Schulbibliothek. Es existiert bereits eine Büchertabelle und man möchte jetzt Bücher an SchülerInnen verleihen.

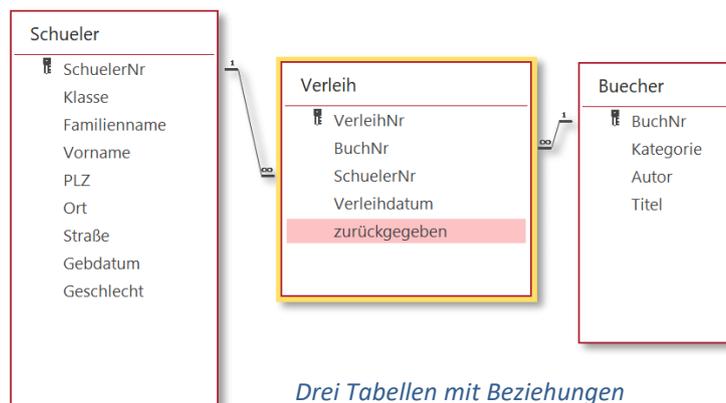
Dazu könnte man auf die (schlechte!) Idee kommen, die Büchertabelle um die Felder Schülername und Verleihdatum zu erweitern. Das hätte aber große Nachteile:



- Bei jedem verliehenen Buch müsste arbeitsaufwendig ein Schülernamen eingetragen werden. Bei mehreren verliehenen Büchern also mehrfach der gleiche Name (Redundanz!).
- Jeder Tippfehler im Namen erzeugt eine **Inkonsistenz**: Ein Schüler Max Mayer ist nicht identisch mit einem Schüler Max Mayr!
- Bei der Rückgabe eines Buches müsste der Schüler mühsam aus dem Feld gelöscht werden.

Viel besser ist die Aufteilung der Daten in drei in Beziehung (**Relation**) stehende Tabellen:

- Eine Tabelle **Bücher** mit eindeutiger Buchnummer und Titel und Autor.
- Eine Tabelle **Schüler** mit einer eindeutigen Schülernummer und Schülerdaten.
- Eine Tabelle **Verleih**, in der jeder Datensatz einen Verleihvorgang repräsentiert.



Bei einem Verleih muss nur noch zum Buch der Schüler gewählt werden. Das Verleihdatum kann automatisch eingetragen werden. Bei der Rückgabe eines Buches wird ein Häkchen in das Feld **zurückgegeben** eingetragen: Keine Redundanz, fehlerhafte Eintragungen (Inkonsistenzen) werden weitgehend vermieden!

Relationale Datenbanken:

- Enthalten Tabellen, die zueinander in Beziehung (**Relation**) stehen.
- Jeder Datensatz in einer Tabelle sollte eine eindeutige Nummer oder einen eindeutigen Code haben (Primärschlüssel!)
- Beziehungen zwischen Tabellen verhindern Redundanz, weil Daten nur einmal eingetragen werden müssen!
- Grundsätzlich:
Ein Datensatz soll nur die zu einem Objekt gehörigen Daten enthalten: Eine Büchertabelle darf z. B. **keine** Schülerdaten enthalten.



Eine neue leere Datenbank mit Access erstellen

- ⇒ Öffne Access
- ⇒ Wähle Leere Datenbank (1)
- ⇒ Gib einen Namen für die Datenbank ein (2).
- ⇒ Klicke auf Erstellen (3). Damit wird die Datenbank gespeichert.



Es gibt einen wichtigen Unterschied zu anderen Office-Dokumenten: In Word oder Excel kann man auch ungespeicherte Dokumente bearbeiten. Access Datenbanken müssen bei der Neuanlage gespeichert werden.

Alle Änderungen, die in den **Daten** einer Datenbank vorgenommen werden, werden sofort gesichert – ein manuelles Speichern ist nicht erforderlich.

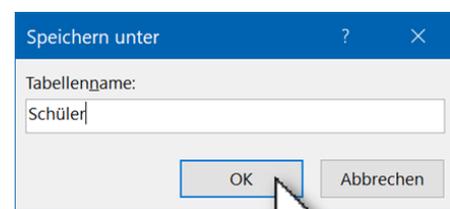
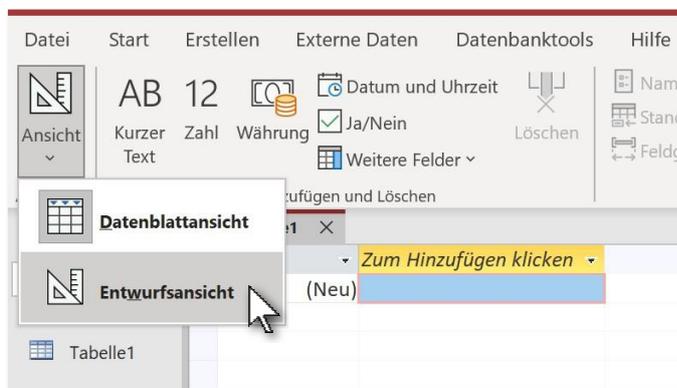
Tabellen erstellen

Eine leere Datenbank enthält noch keine Tabellen zum Speichern der Daten.

Access bietet zuerst an, eine Tabelle in der **Datenblattansicht** zu erstellen.

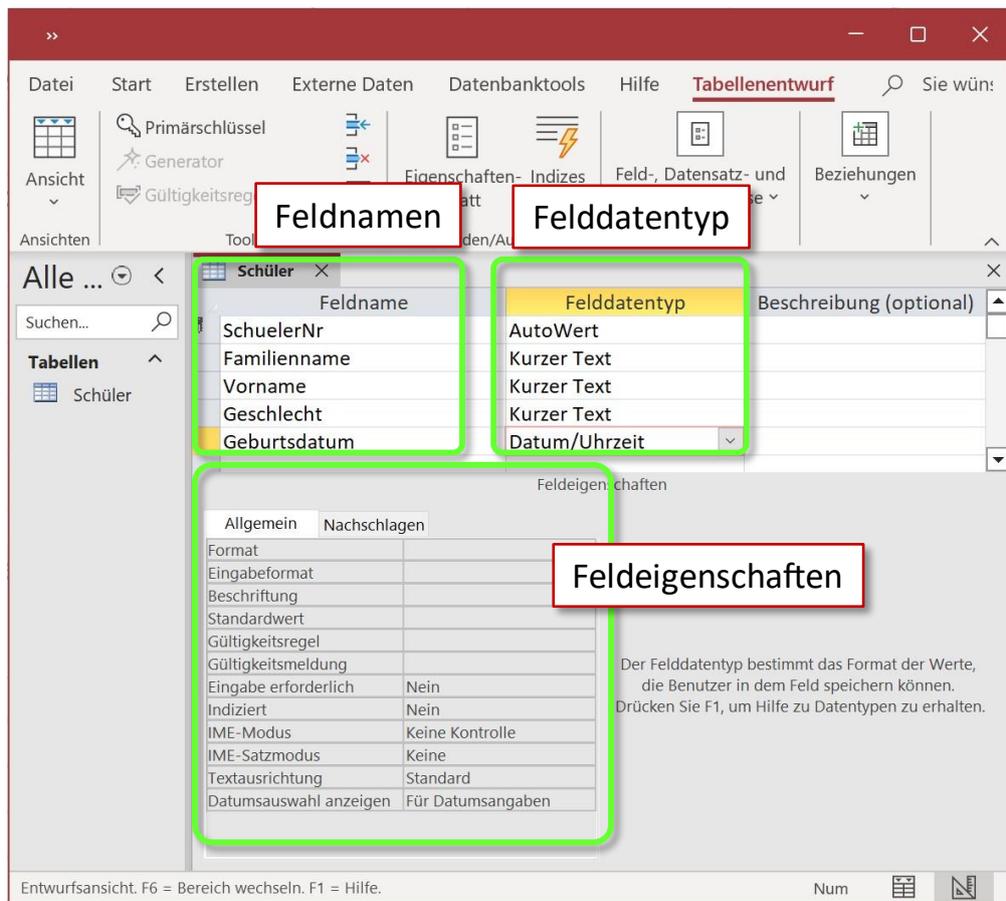
Deutlich mehr Möglichkeiten bietet die **Entwurfsansicht**:

- ⇒ Wechsle in die Entwurfsansicht und gib den Namen für die Tabelle ein: **Schüler**.





Anders als in Excel muss zuerst die Struktur der Tabelle definiert werden. Erst danach können Daten eingegeben werden.



Feldnamen

In unserem Beispiel oben wird eine Tabelle zur Speicherung von Schülerdaten erstellt. Jeder Schüler bekommt eine eindeutige Schülernummer, ein Feld für Familiennname, Vorname, Geburtsdatum etc.

Felddatentyp

Jedem Feld muss ein passender Datentyp zugeordnet werden.

In Access stehen unter anderem diese Felddatentypen zur Verfügung:

Datentyp: Kurzer Text bzw. Langer Text

Typ	Beschreibung
Kurzer Text	Für Texte bis zu 255 Zeichen Geeignet z. B. für Namen, Orte, Buchtitel, Postleitzahlen, Telefonnummern <i>Achtung! Postleitzahlen und Telefonnummern bestehen zwar hauptsächlich aus Ziffern, werden aber als Textfelder definiert. Damit sind führende Nullen, Leerzeichen und Buchstaben wie Länderkennungen möglich.</i>
Langer Text	Für Texte bis zu 65 536 Zeichen. <i>Für längere Texte wie Artikelbeschreibungen, Inhaltsangaben etc.</i>



Datentyp: Zahl

In den Feldeigenschaften (Feldgröße) wird die Art der Zahl festgelegt:

Ganzzahlen: Byte, Integer, Long Integer

Dezimalzahlen: Single, Double

Typ	Beschreibung	
Byte	Zahlen von 0 bis 255	Ganzzahl
Integer	Zahlen von -32 768 bis +32 767	Ganzzahl
Long Integer	Zahlen von -2,1 bis +2,1 Milliarden	Ganzzahl
Single	Zahlen von -10^{38} bis $+10^{38}$ <i>Der Zahlentyp Single muss gewählt werden, wenn man Nachkommastellen benötigt. Durch seinen riesigen Wertebereich ist Single für fast alle numerischen Daten geeignet.</i>	7 Dezimalstellen
Double	Zahlen von -10^{308} bis $+10^{308}$ <i>Für wirklich große Zahlen! Zum Vergleich: Es wird geschätzt, dass das von der Erde aus beobachtbare Universum aus ca. 10^{80} Atomen besteht.</i>	15 Dezimalstellen

Datentyp: Währung

Speziell für Währungsangaben im Bereich von +/- 1 Billiarde (+/- 10^{15})

In den Feldeigenschaften (Format) kann die Darstellung der Zahl definiert werden.

Datentyp: Autowert

Der Datentyp Autowert (entspricht einer Zahl vom Typ Long Integer) wird für eindeutige Nummerierungen verwendet. Jeder neue Datensatz erhält automatisch eine höhere Nummer als der vorhergehende. Dadurch wird sichergestellt, dass eine Nummer nicht mehr als einmal vorkommt. Eine einmal vergebene Nummer wird auch nach der Löschung des Datensatzes nicht mehr neu verwendet.

Datentyp: Datum/Uhrzeit

Das Format der Datumsangabe kann in den Feldeigenschaften passend eingestellt werden:

z.B. *Datum, kurz:* 19.03.13, *Datum, lang:* Dienstag, 19. März 2013

Datentyp: Ja/Nein

Für Felder, die nur die Option Ja oder Nein, Wahr oder Falsch, Ein oder Aus benötigen. Ja/Nein-Felder werden als Kontrollkästchen zum Anhängen dargestellt. Beispiel: In einer Büchertabelle wird der Verleihstatus mit Hilfe des Ja/Nein-Feldes „verliehen“ gespeichert.

Datentyp: Link

Für die Angabe von Hyperlinks

Datentyp: Anlage

Speichert Dateien wie z. B. Bilder



Feldeigenschaften

Je nach Datentyp können verschiedene Feldeigenschaften ausgewählt werden. Beim Datentyp Text kann z.B. die maximale Länge definiert werden, bei Zahlen beispielsweise die Anzahl der Nachkommastellen. Außerdem kann ein Standardwert (z.B. die Vorgabe „Österreich“ bei der Eingabe einer Adresse) festgelegt werden und ob eine Eingabe erforderlich ist (z.B. Familienname oder Geburtsdatum).

Beispiel Namensliste

Feldnamen	Felddatentyp	Feldeigenschaften
SchülerNr	Autowert	Feldgröße: Long Integer
Nachname	Text	Feldgröße: 25 Eingabe erforderlich: ja
Vorname	Text	Feldgröße: 25 Eingabe erforderlich: ja
Geschlecht	Text	Feldgröße: 1 Eingabe erforderlich: ja Gültigkeitsregel: "m" Oder "w" Oder "d"
Geburtsdatum	Datum/Uhrzeit	Format: Datum, kurz
Straße	Text	Feldgröße: 50
PLZ	Text	Feldgröße: 10
Ort	Text	Feldgröße: 30
Staat	Text	Standardwert: "Österreich"

Gültigkeitsregel

Wird in der Entwurfsansicht in den Feldeigenschaften definiert.

Durch eine Gültigkeitsregel werden die Eingabemöglichkeiten eingeschränkt. Wird gegen diese Regel bei der Dateneingabe verstoßen, kommt eine Fehlermeldung.

Beispiel: Beim Feld Geschlecht wird nur „m“ oder „w“ oder „d“ (für divers) zugelassen.

Gültigkeitsregel	"m" Oder "w" Oder "d"
------------------	-----------------------

Eingabe erforderlich

Wird in der Entwurfsansicht in den Feldeigenschaften definiert

Wird z. B. im Feld Geburtsdatum bei „Eingabe erforderlich“ **Ja** ausgewählt, so muss bei der Dateneingabe ein Geburtsdatum eingetragen werden.

Standardwert

In den Feldeigenschaften kann ein Standardwert (z.B. die Vorgabe „Österreich“ bei der Eingabe einer Adresse) festgelegt werden. Damit erspart man sich die Eingabe von „Österreich“ bei den Adressangaben.



Primärschlüssel

Jede Tabelle in einer Datenbank braucht ein Feld, das einen Datensatz eindeutig identifiziert. Diesem Feld wird der Primärschlüssel zugewiesen. Doppelte Einträge (Duplikate) sind dadurch nicht mehr möglich.

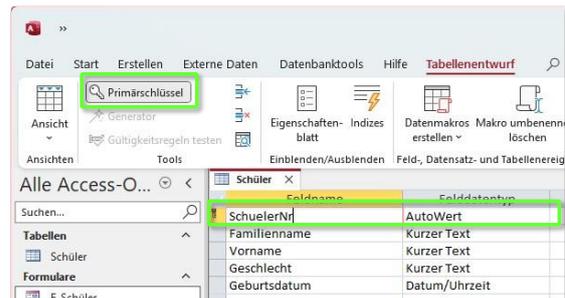
Beispiel 1: Eine Bibliothek hat mehrere Exemplare eines bestimmten Buches. Man kann sie nur mit Hilfe der Buchnummer unterscheiden.

Beispiel 2: Die Sozialversicherungsnummer ist auch bei Personen mit gleichem Namen und gleichem Geburtsdatum unterschiedlich.

Der Primärschlüssel ist oft eine Nummer, kann aber auch ein Textfeld sein. Ein Primärschlüsselfeld darf nicht leer sein und ändert sich fast nie.

Primärschlüssel festlegen:

Zur Entwurfsansicht wechseln > in das Feld klicken > Tabellenentwurf > Tools > Primärschlüssel



Ein Index beschleunigt Suchen!

Felder mit Index werden vorsortiert. Suchvorgänge werden durch einen Index viel schneller! Ein Index sollte für die Felder erstellt werden, in denen oft Werte gesucht oder sortiert werden. Eine wichtige Rolle spielen Indizes bei der Verknüpfung von Tabellen in Abfragen. Die richtige Vergabe von Indizes ist entscheidend für die Suchgeschwindigkeit einer Abfrage.

Beispiel:

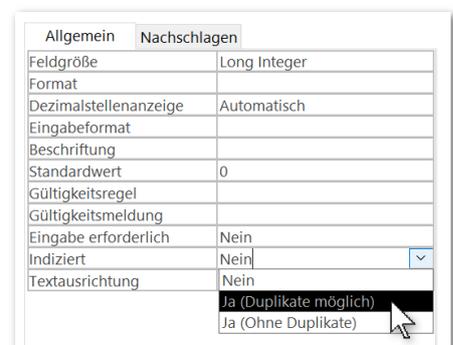
Das Telefonbuch von Wien ist nach Namen geordnet: Ein Herr Max Mustermann ist auch händisch innerhalb kurzer Zeit gefunden. Wenn du aber den Teilnehmer zu einer bestimmten Telefonnummer wissen möchtest, so würde das Durchsuchen der Einträge Tage brauchen. Hättest du ein Telefonbuch, das nach Nummern geordnet wäre, bräuchtest du nur Sekunden!

In einer Datenbank wird die Sortierung durch einen Index bestimmt. Im Unterschied zu einem Telefonbuch kann es mehrere Sortierungen (Indizes) zugleich geben.

Ohne Index muss der Computer jeden Datensatz einer Tabelle bei einer Suche lesen – das dauert bei einer großen Datenbank sehr lange. Mit einem passenden Index sind die Daten in Sekundenbruchteilen gefunden.

Index erstellen

Einen Index erstellt man in der Entwurfsansicht der Tabelle. Man kann wählen ob Duplikate (mehrfach vorkommende Werte wie z.B. Vornamen) zugelassen sind oder nicht.



Datensätze in die Tabelle eingeben

Grundsätzlich können Daten direkt in die Tabelle eingegeben werden. Mit einem Doppelklick auf den Tabellennamen gelangt man in die Datenblattansicht der Tabelle. Hier können Daten eingegeben, gelöscht, gesucht, sortiert und gefiltert angezeigt werden.

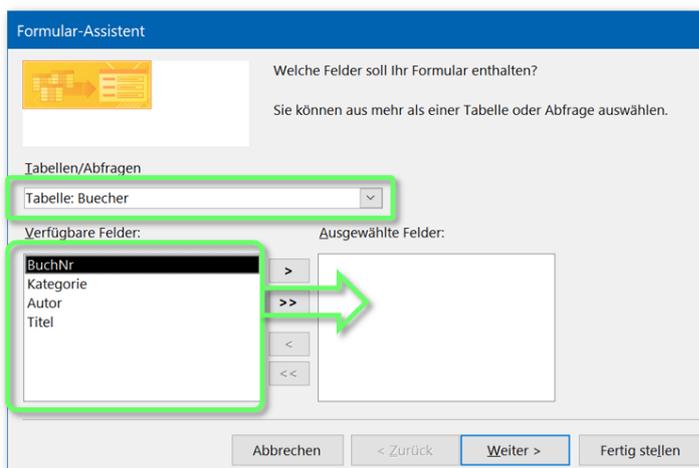


Formulare

So gut wie immer wird für die Eingabe von Daten in eine Tabelle ein Formular verwendet. Formulare beschränken den Zugriff auf die notwendigen Datenbankfelder und können mit Hilfe von Unterformularen mehr als eine Tabelle darstellen (z.B. Kunden mit dazugehörigen Rechnungen)

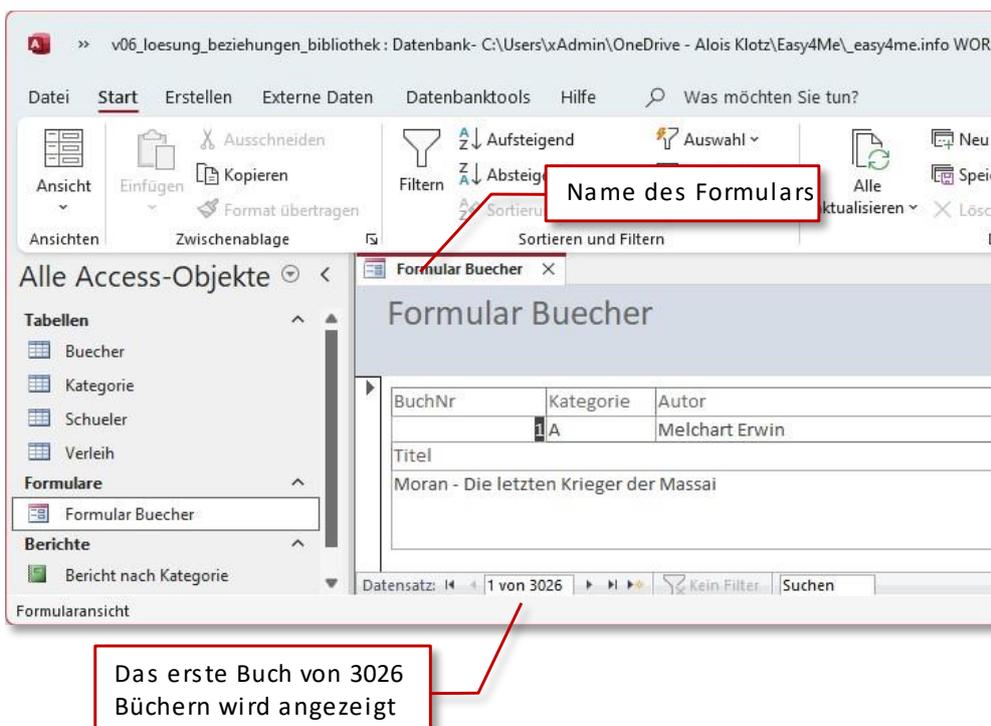
Ein Formular erstellen

In diesem Beispiel ist die Datenbank **Bibliothek** geöffnet. Formulare erstellt man einfach mit dem Formularassistenten: *Erstellen > Formulare > Formular-Assistent*. Nach der Auswahl der Tabelle bzw. einer Abfrage werden die Felder gewählt.



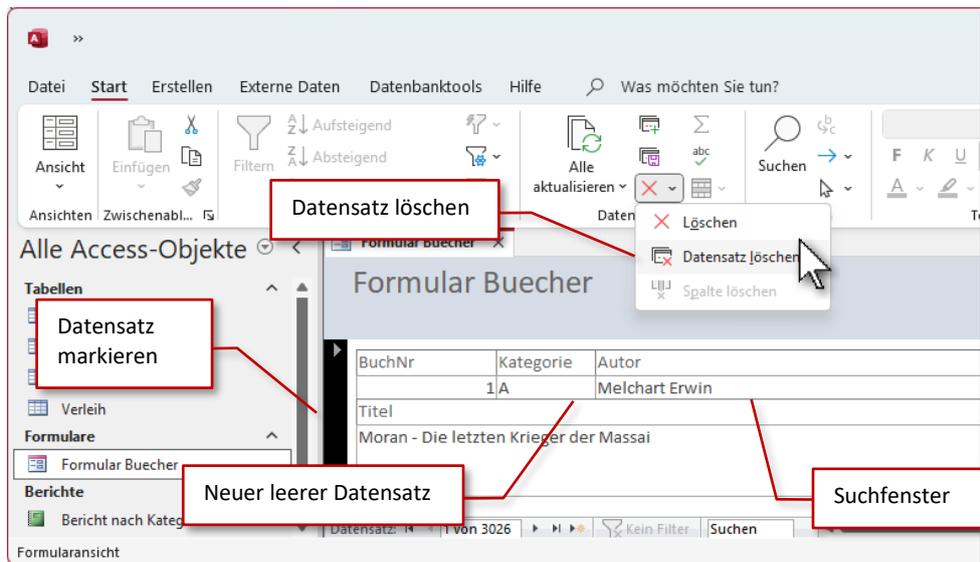
Nach der Fertigstellung kann das Formular sofort zur Eingabe von Büchern verwendet werden.

Elemente eines Formulars





Datensätze können neu angelegt, gesucht, bearbeitet und gelöscht werden:



Beziehungen

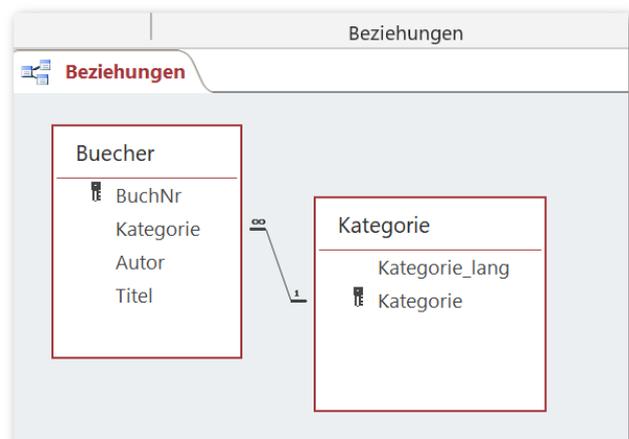
Tabellen werden durch Beziehungen miteinander verknüpft.

Datenbanktools > Beziehungen

Beispiel: Die Tabelle **Buecher** wird mit der Tabelle **Kategorie** (mit Einträgen wie Abenteuer, Wissen, Dokumentation, Reisen etc.) verknüpft.

Vorteil: Man muss die Kategorien nur einmal definieren und kann sie dann den Büchern zuordnen.

Im Beispiel rechts kann jedem Buch **eine** Kategorie zugeordnet werden. Jeder Kategorie hingegen können beliebig viele Bücher zugeordnet sein.



Felder von Tabellen, die durch eine Beziehung verknüpft werden, müssen **den gleichen Feldtyp und die gleiche Feldgröße** haben.

Bei der Erstellung von Beziehungen kann die **referentielle Integrität** aktiviert werden. Das hat in unserem Beispiel folgende Auswirkungen:

- ⇒ Ein Datensatz aus der Tabelle Kategorie kann nur gelöscht werden, wenn kein einziges Buch auf diese Kategorie verweist.
- ⇒ Will man einen Datensatz aus der Tabelle Kategorie löschen, müssen alle Verweise auf diesen Datensatz in der Tabelle Buecher entfernt werden.

Beziehungen zwischen Tabellen sind eine grundlegende Eigenschaft von Datenbanken. Durch Beziehungen werden Wiederholungen (Redundanzen) vermieden.



Abfragen

Mit Abfragen kann man Daten aus einer oder mehreren Tabellen abrufen, bearbeiten und analysieren. Abfragen können gespeichert und immer wieder verwendet werden.

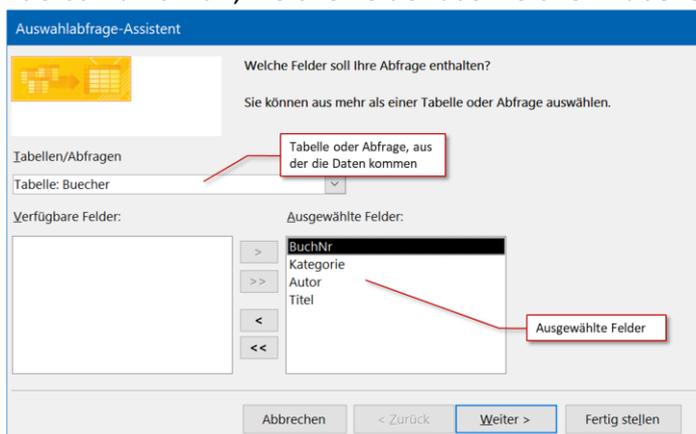
Hier einige Beispiele für Abfragen:

- Schulbibliothek:
 - Welche Bücher wurden seit drei Jahren nie verliehen?
 - Wer hat sich die meisten Bücher im Schuljahr ausgeliehen?
 - Bei welchen Büchern wurde die Verleihfrist überschritten?
- Firma:
 - Eine Liste der Umsätze im letzten Monat.
 - Die zehn Kunden mit dem größten Umsatz.
 - Welche Rechnungen wurden noch nicht bezahlt?
 - Alle Kunden nach Name sortiert.

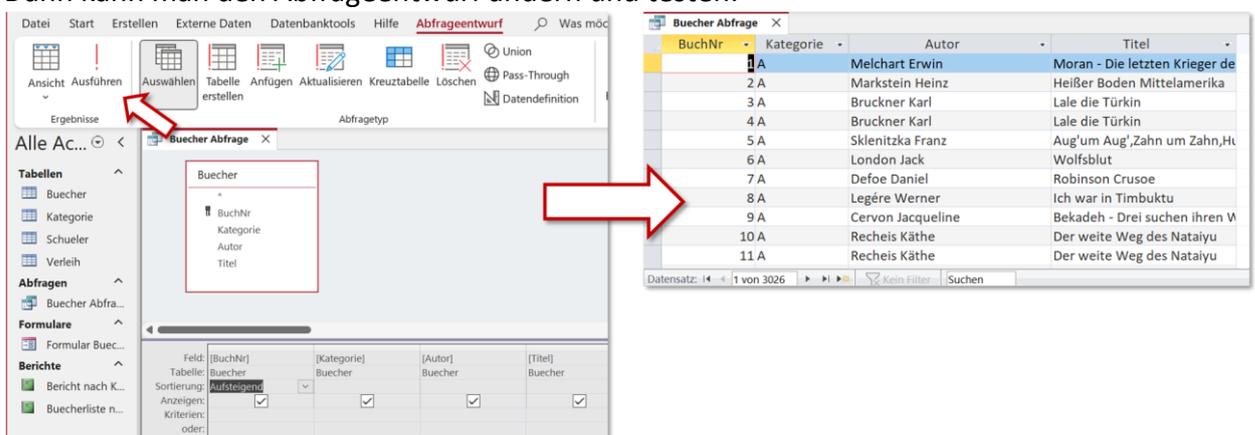
Eine Abfrage erstellen

Am einfachsten verwendet man den Abfrage-Assistent (*Erstellen > Abfragen > Abfrageassistent*) und wählt Auswahlabfrage-Assistent.

- Zuerst wählt man, welche Felder aus welchen Tabellen angezeigt werden sollen:



- Dann kann man den Abfrageentwurf ändern und testen:



- Zuletzt speichert man die Abfrage unter einem passenden Namen. So kann sie später wieder verwendet und/oder geändert werden.



Berichte

Berichte werden erstellt, um Daten aus einer oder mehreren Tabellen übersichtlich darzustellen, sodass sie leicht gelesen und analysiert werden können. Berichte können einfach ausgedruckt und weitergegeben werden.

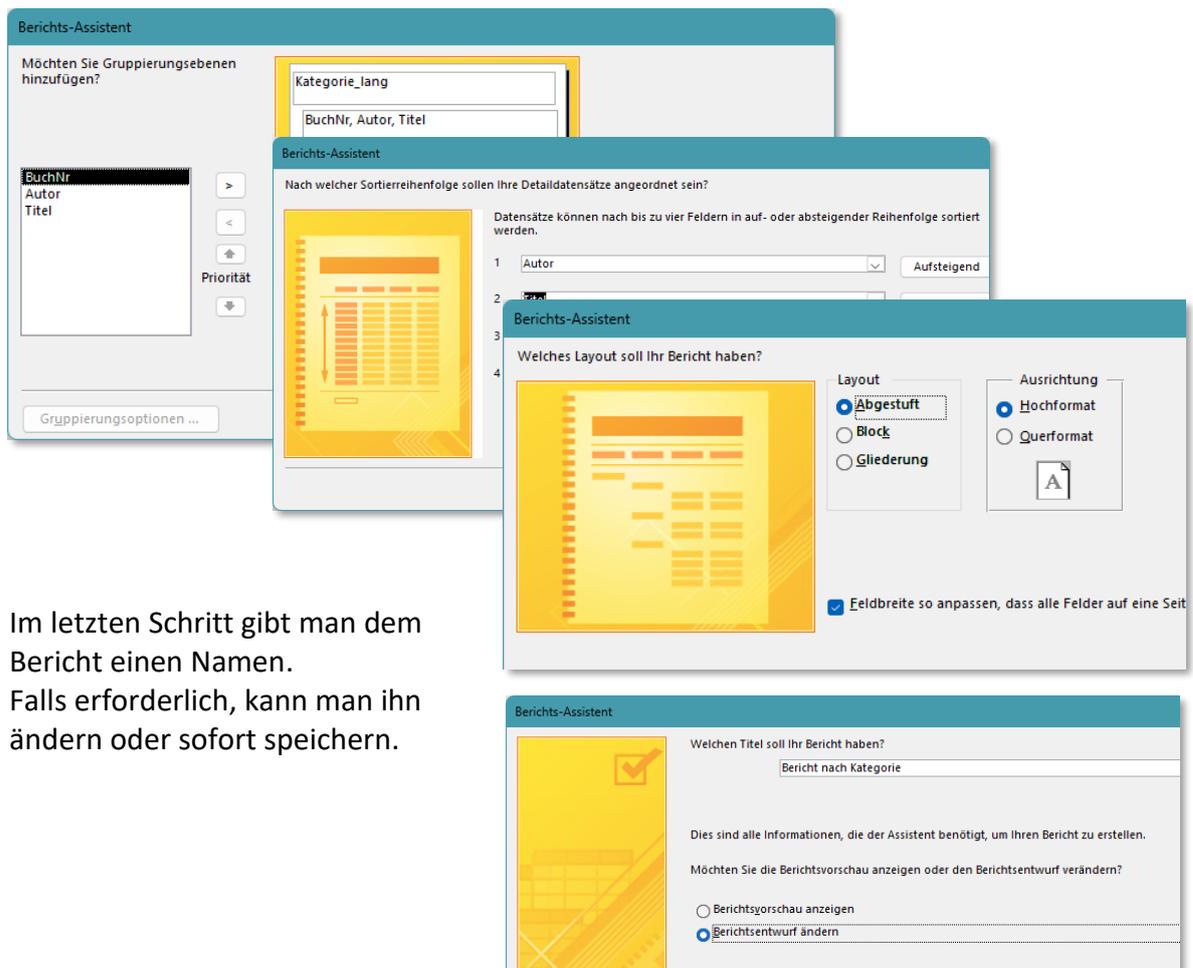
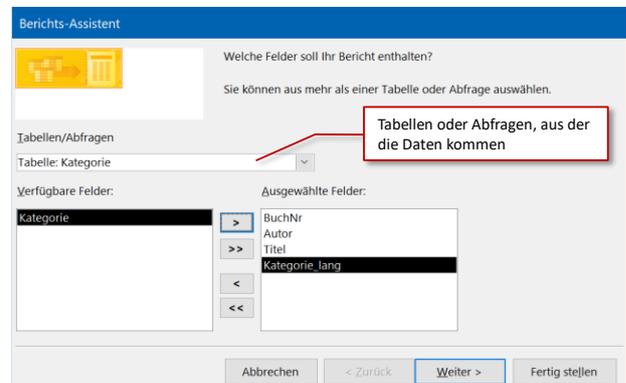
Bericht erstellen

Hier wird als Beispiel eine Bücherliste einer Schulbibliothek erstellt.

Am einfachsten erstellt man einen Bericht mit dem Berichtsassistenten:

Erstellen > Berichte > Berichts-Assistent

- Zuerst wählt man die Tabellen oder Abfragen, deren Daten man darstellen möchte.
- Mit einem Klick auf **>** oder **>>** wählt man einzelne bzw. alle Felder, die im Bericht dargestellt werden sollen. Hier wurden aus den Tabellen **Buecher** und **Kategorie** Felder gewählt.
- In den nächsten Schritten wählt man, wie die Daten gruppiert, sortiert und angeordnet werden sollen.



- Im letzten Schritt gibt man dem Bericht einen Namen. Falls erforderlich, kann man ihn ändern oder sofort speichern.



Begriffe kurz erklärt

Abfrage	
Bericht	Berichte zeigen Daten aus einer Datenbank. Beispiel: Schülerliste mit Adressen
Beziehungen	Beziehungen sind Verknüpfungen zwischen Tabellen, die es ermöglichen, Daten aus verschiedenen Tabellen zu verbinden und abzufragen.
Daten	Daten sind Fakten ohne weitere Bedeutung. <i>Beispiel: gemessene Temperaturdaten: 15 °C, 12 °C, 11 °C, 5 °C, -3 °C, ...</i>
Datensatz	Entspricht einer Zeile in einer Tabelle. <i>Beispiel: Ein Schüler</i>
Feld	Ein Feld einer Datenbank ist eine spezifische Eigenschaft, die in einem Datensatz gespeichert wird. <i>Beispiel: Geburtsdatum</i>
Formular	Dienen zur Eingabe von Daten
Gültigkeitsregel	Schränkt die Eingabe von Daten ein. Beispiel Ein Schüler darf zum Beispiel kein Geburtsdatum in der Zukunft haben!
Index	Beschleunigt Suchen in der Datenbank
Information	Grundlage von Informationen sind Daten, die in einen sinnvollen Zusammenhang gebracht werden.
Inkonsistenz	Fehlerhafte Eintragung in einer Datenbank. <i>Beispiel: Ein Schüler ist ein einer Datenbank irrtümlich ein zweites Mal eingetragen worden.</i>
Primärschlüssel	Ein Primärschlüssel ist eine eindeutige Kennzeichnung für jeden Datensatz in einer Tabelle.
Redundanz	Dieselben Daten werden mehrfach gespeichert.
Relationale Datenbank	Enthält Tabellen, die miteinander in Beziehung stehen