

1. Einführung in die Arbeit mit Datenbanken

1.1. Allgemeines

Eine Datenbank ist eine Sammlung von vielen Daten, auf die man zentral zugreifen und die man zentral verwalten kann. Die Daten werden einmalig gespeichert und können vom Nutzer auf verschiedene Weise ausgewertet werden.

Das Datenbankverwaltungsprogramm ACCESS aus dem OFFICE Paket der Firma MICROSOFT ist eines der leistungsfähigsten Hilfsmittel der elektronischen Datenverarbeitung, mit dem für die Menschen nahezu unvorstellbare Informationsmengen schnell und effektiv gespeichert, in Bezug gesetzt, bearbeitet und wieder abgerufen werden können. Daten können nach verschiedenen Kriterien geordnet und nach wieder anderen Kriterien für die Anzeige am Bildschirm ausgewählt werden.

Vor der Zeit der Datenverwaltungsprogramme mussten ganze Karteikästen mit der Hand und mit unglaublicher Geduld nach bestimmten Daten durchsucht werden und eventuell mit Radiergummi und Bleistift geändert werden.

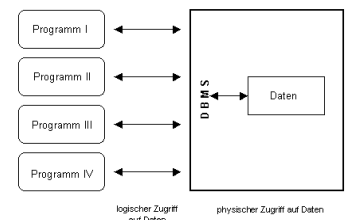
ACCESS unterstützt Sie bei der Bewältigung dieser Aufgaben sowie bei der Verwaltung von Kundendateien, Rechnungen, Statistiken, bei der Versendung von Serienbriefen oder bei der Erarbeitung von Berichten.

Dieser Lehrbrief führt Sie an die Schwerpunkte dieses Programms heran und zeigt Ihnen, wie Sie eine Datei anlegen, verwalten und auswerten.

Wenn Sie die Grundfertigkeiten beherrschen, sind Sie ganz schnell in der Lage mit der Nutzerdokumentation, d.h. mit Hilfe der Handbücher und der Online-Hilfe, weiterführende Befehle und Anweisungen zu erfassen, bis hin zur eigenen Programmierung.

1.2. Datenbank/ Datenbankmanagementsystem

Ein Datenbanksystem (DBS) setzt sich aus einer Datenbasis und einem Datenbank-Managementsystem (DBMS) zusammen. Das DBMS ist ein Softwaresystem zur Definition, Administration, Manipulation und Abfrage von Daten. Es stellt die Schnittstelle zwischen Benutzer und Datenbasis dar und dient der effizienten Speicherung und Abfrage der strukturierten Daten, z.B. in Tabellen erfasste Informationen, dazugehörige Abfragen, Formulare, Berichte zur Anzeige und Ausgabe der Informationen **.accdb (alt .mdb)**

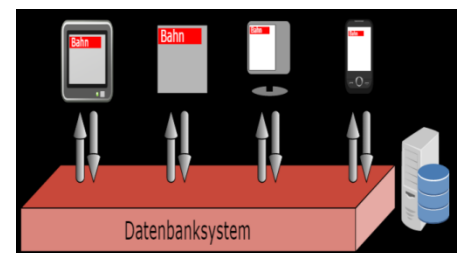


Echte Datenbanksysteme erlauben daher den **gleichzeitigen vollen Zugriff** mehrerer Benutzer auf die gleichen Daten.

Denkt man z.B. an das Informationssystem der Bahn, dann wird klar, dass unterschiedliche Programme auf die gleichen Daten zugreifen müssen:

- Die Fahrplanauskunft im Internet
- Das Buchungsprogramm im Reisebüro
- Das Smartphone eines Schaffners, wenn er mögliche Anschlusszüge durchgibt
- Vielleicht sogar die Infotafel im Bahnhof
- u.v.m.

Typisch für ein Datenbanksystem ist es, dass es zentral auf einem **Server** läuft und dass viele unterschiedliche Programme (**Clients**) auf die gleichen Daten zugreifen können.



ACCESS ist zugleich Datenbankverwaltungsprogramm und Entwicklungsumgebung mit einer Vielzahl menügesteuerter Werkzeuge (tools) unter Microsoft WINDOWS. Die vollständige Kompatibilität zu früheren Versionen ist gewährleistet.

Die Bedienung (Steuerung) erfolgt entweder durch das umfangreiche **Menüsystem** oder durch die direkte Eingabe von **Befehlen** (SQL Sprache).

Für jeden dieser Bereiche gibt es speziell zugeschnittene Bedienelemente wie Menüs, Dialogfenster oder Schalter.

1.3. Anforderungen an ein DBMS

Aus dem Aufbau eines Datenbanksystems ergeben sich vorteilhafte Eigenschaften, die gleichzeitig Kriterien für ein gutes DBS sind:

Integritätssicherung, Daten werden auf Korrektheit (bereits während der Eingabe) überprüft und Fehlmanipulationen verhindert

Redundanzarmut, es gibt keine ungeordnete Mehrfachspeicherung von Datenwerten

Datensicherheit, ungewollter Datenverlust wird durch interne Backup- und Prüfmechanismen verhindert

Datenschutz, Zugriffskontrolle und spezifische Sichten sorgen für einen Zugang gemäß der Rechte des Nutzers

Mehrbenutzerbetrieb, viele Benutzer können parallel auf der Datenbank arbeiten

Datenunabhängigkeit, das DBMS ist nicht an die Daten der Datenbank gekoppelt, es kann unabhängig von den Daten weiterentwickelt werden

zentrale Kontrolle, ein Administrator ist in der Lage, das gesamte System von einem Rechner aus zu verwalten

1.4. Grundlegende Begriffe

Datenbank/ Datenbanksystem:

Datenansammlung z.B. in Tabellen erfasste Informationen, dazugehörige Abfragen, Formulare, Berichte zur Anzeige und Ausgabe der Informationen **.accdb (alt .mdb)**

Tabelle: Anordnung der Daten in Spalten und Zeilen

Datenfeld: Spalte, gekennzeichnet durch Name, Feldtyp, Länge und andere Eigenschaften

Datensatz: Zeile, gekennzeichnet durch gleiche Struktur (Attributwerte)

Datenbankfenster: Anzeige aller Datenbankobjekte

Abfrage: Logische Definition einer Datensicht (*Auswahl*)

Formular: Bildschirmmaske zur Anzeige oder *Eingabe* von Daten

Bericht: Gestaltung des Erscheinungsbildes der Daten für die *Ausgabe*

ERD (Entity-Relationship-Diagramm)

Graphische Methode zur Darstellung komplexer Beziehungsgebilde der Realität

Grundlegende Komponenten

Entität (Entity): individuell identifizierbares Objekt der Wirklichkeit; z. B. Mitarbeiter

Beziehung (Relationship): Verknüpfung / Zusammenhang zwischen zwei oder mehreren Entitäten; z. B. Angestellte leiten Projekte

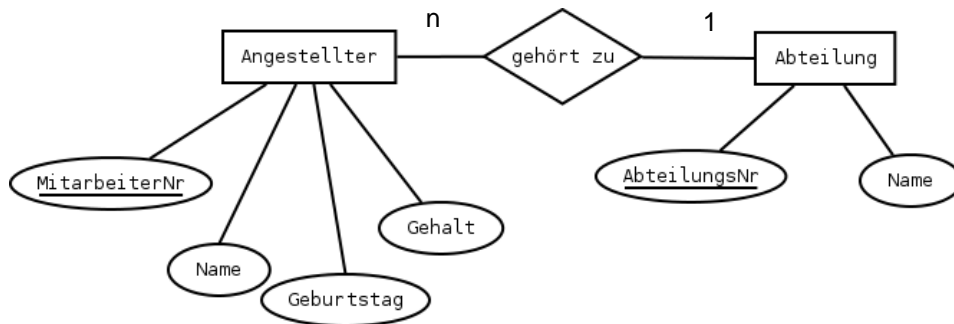
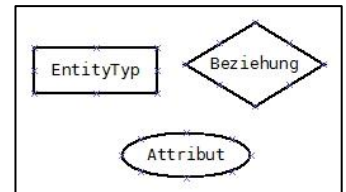
Eigenschaft (engl. attribute): Was über eine Entität (im Kontext) von Interesse ist; z. B. das Eintrittsdatum

Der **Primärschlüssel** muss die Entität eindeutig identifizieren und wird im ERD unterstrichen dargestellt.

Der **Fremdschlüssel** wird bei 1:n und n:m Beziehungen notwendig (siehe unten)

Kardinalität

<http://www.datenbank-grundlagen.de/entity-relationship-modell.html>
[https://de.wikipedia.org/wiki/Kardinalit%C3%A4t_\(Datenbankmodellierung\)](https://de.wikipedia.org/wiki/Kardinalit%C3%A4t_(Datenbankmodellierung))



Fremdschlüssel

ERM (Entity-Relationship-Modell)

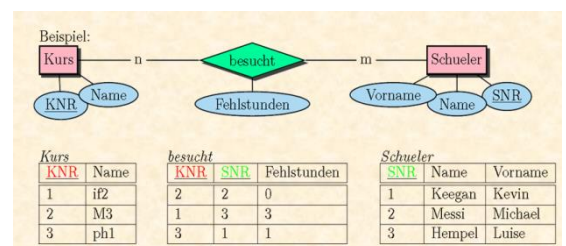
Angestellter(MitarbeiterNr, Name, Geburtstag, Gehalt, AbteilungsNr)
 Abteilung(AbteilungsNr, Name)

Kardinalität	Beschreibung	Darstellung im Diagramm
1:1-Beziehung	Jeder Entität der Entitymenge A wird genau eine Entität der Entitymenge B zugeordnet und umgekehrt.	A — 1 —◇— 1 — B
1:n-Beziehung	Jeder Entität der Entitymenge A können mehrere Entitäten der Entitymenge B zugeordnet werden, umgekehrt kann jeder Entität der Menge B nur genau eine Entität der Menge A zugeordnet werden.	A — 1 —◇— n — B
m:n-Beziehung	Jeder Entität der Entitymenge A können mehrere Entitäten der Entitymenge B zugeordnet werden und umgekehrt.	A — m —◇— n — B

Transformationsregeln

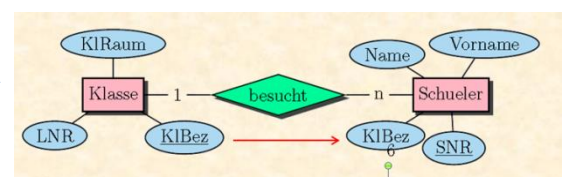
Jeder Entitätstyp wird als Tabelle dargestellt. Jede Tabelle benötigt einen Primärschlüssel.

1. Jede n:m Beziehung wird durch eine eigene Tabelle dargestellt. Die Beziehungstabelle enthält die Primärschlüssel der verknüpften Entities und die eigenen Attribute.



2. Jede 1:n und 1:1 Beziehung **mit** eigenen Attributen wird durch eine eigene Tabelle dargestellt. Die Beziehungstabelle enthält die Primärschlüssel der verknüpften Entities und die eigenen Attribute.

3. Jede 1:n Beziehung **ohne** eigene Attribute wird so dargestellt, dass die Primärschlüssel der 1-Entität Fremdschlüssel der n-Entität wird.



Jede 1:1 Beziehung **ohne** eigene Attribute wird so dargestellt, dass der Primärschlüssel der ersten Entität bei der 2. Entität Primär- und Fremdschlüssel zugleich sind.

1.5. Die Arbeit mit bestehenden Datenbanken und Tabellen

- Datenbank/ Tabelle öffnen
- Navigation im Datenblatt
- Datenblattlayout
 - Spaltenbreite/ Zeilenhöhe
 - Spalten ein-/ausblenden
 - Spalten fixieren
 - Spaltenanordnung umstellen
 - Schriftart, Schriftgestaltung
 - Zellformat (Entwurfsansicht)
- Daten ergänzen
- Daten speichern
- Daten bearbeiten
- Daten löschen
- Daten kopieren, verschieben, einfügen
- Daten sortieren
 - * Sortiert wird die Spalte, die entweder markiert ist oder in welcher der Cursor steht
 - * Es wird nur das Layout verändert
 - * Bei zwei nebeneinander markierten Spalten wird zuerst nach dem 1. Feld geordnet und innerhalb dieses Feldes nach dem anderen Kriterium (gleichzeitig entweder auf- oder absteigend!)
- Daten drucken
- Selektion (siehe Auswahlabfragen)

1.6. Datenbank oder Tabellenentwurf

1. Erarbeitung eines „Planes“ (Objekte/ Entities, Beziehungen)
2. Fixierung der Relationen im ERD
3. Ergänzung der Attribute (Eigenschaften der Entities)
4. Festlegung der Primärschlüssel und der Beziehungen (1:1, 1:n, n:m)
5. Übertragung des ERD in das ERM
6. Übertragung des ERM in das Programm (ACCESS) mit Festlegung der Feldnamen, Datentypen, Gültigkeitsregeln, Eingabeformaten etc.
7. Anlegen der Tabelle über die Objektliste
8. Eingabe der Daten (in der Tabellenansicht)
9. Speichern der Tabelle

Feldname	Felldatentyp
Klasse	Text
	Memo
	Zahl
	Datum/Uhrzeit
	Währung
	AutoWert
	Ja/Nein
	OLE-Objekt
	Hyperlink
	Anlage
	Berechnet
	Nachschlage-Assistent

Datentyp	Beschreibung	Beispiel
Text	Zeichenfolge ohne numerischen Wert	Müller 23-123-66
Auto Wert	Eindeutiger Wert, der von ACCESS für jeden neuen Datensatz erstellt wird	1 ↓ 2 ↓
Zahl (numerische Werte)	Byte – ganze Zahlen von 0-255 Integer – ganze Zahlen von -32768 -32767 Single – Dezimalzahlen von -3,4x10 ³⁸ -3,4x10 ³⁸ Double – Dezimalzahlen von -1,797x10 ³⁰⁸ – 1,797x10 ³⁰⁸	127 -16354 3466,853 2,7x10 ²⁰⁷
Währung	Geldwerte mit einer Genauigkeit von 2 Dezimalstellen	1003,65 €
Ja/Nein	Boolean Daten (Ja/Nein, Wahr/Falsch)	<input checked="" type="checkbox"/>
Datum/ Zeit	Datums- und/oder Uhrzeitangaben	11.01.2019, 2:05:11
Berechnetes Feld	Ergebnis einer Berechnung aus Feldern derselben Tabelle (evtl. Generator benutzen)	[brutto]*1,19 [menge]/10

1.7. Auswahlabfragen

Unter einer Abfrage versteht man Anweisungen, die es dem Programm ermöglichen, eine bestimmte Teilmenge der Daten aus einer Bestandsdatei darzustellen.

Abfragen werden im **Abfragefenster** erstellt.

Feld:	NACHNAME	GESCHLECHT	WOHNORT	GEBURT
Tabelle:	SCHÜLER	SCHÜLER	SCHÜLER	SCHÜLER
Sortierung:				
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:				
oder:				

1.7.1. Grundlagen

Das Erstellen einer Sicht erfolgt entweder in **der Entwurfsansicht/ Entwurfswenster** oder mit dem Assistenten für Auswahlabfragen.

- Schritte:
1. Öffnen der Bezugstabelle
 2. Eintragen der Suchbedingung(en)
 - Festlegung der Felder, die angezeigt werden sollen
 - Festlegung der Feldreihenfolge
 - Festlegung der Sortierreihenfolge (BEACHTE Schachtelung)
 - Festlegung von Kriterien
 3. Datenblattansicht (evtl. Speichern und drucken)

Durch die Eingabe von Bedingungen in die jeweilige Spalte unterhalb des Feldnamens oder in ein Bedingungsfenster können Sie festlegen, welche Daten die neue Sicht enthalten soll.

Selektion- Zeilen werden aus einer Tabelle ausgewählt, die bestimmten Eigenschaften genügen
Projektion - Spalten werden aus einer Tabelle ausgewählt, die bestimmten Eigenschaften genügen

Operatoren	Rangfolge
Relationale O.	= , < > , < , <= , > , >=
Arithmetische O.	* , / , + , -
Logische O.	.NOT. , .AND. , .OR.

Hinweise zum Umgang mit Filterbedingungen:

- Platzhalter möglich (siehe unten)
- Feldspezifik: Text „...“, Zahl blank, Datum #...#, logisch ja/nein
- kein Eintrag → ist null/ irgendein Eintrag → ist nicht null
- Datumsabfragen: monat([...])=4, tag([gebdat])=15, jahr([.....])=2015
Datum() =Aktuelles Datum

• Verwendung von Platzhaltern:

- * beliebige Anzahl von Zeichen, z.B. Annab*
- ? genau 1 Zeichen, z.B. Me?er
- # genau 1 numerisches Zeichen, z.B. 1#3
- [] genau 1 Zeichen innerhalb der Klammer, z.B. w[a,e]

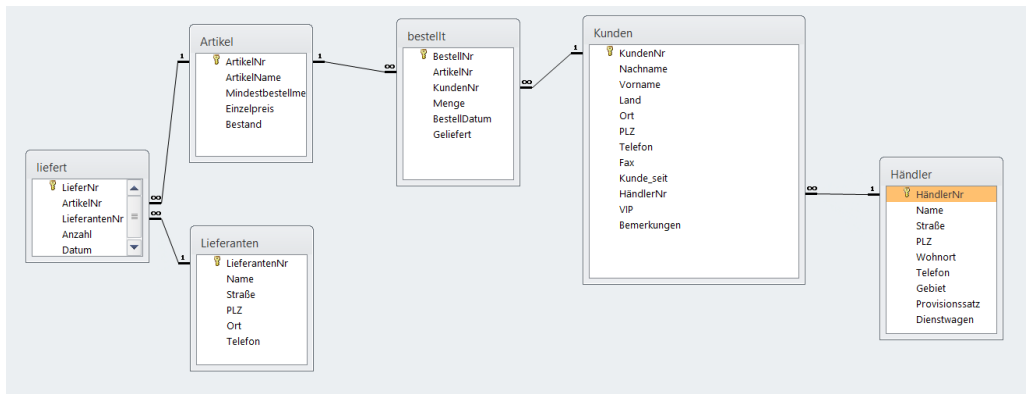
1.7.2. Verknüpfungen

Beim Kombinieren von **UND**- und **ODER**-Bedingungen im Abfrage-Designer werden die Bedingungen vom Programm zeilenweise und nicht Feld für Feld ausgewertet. Das heißt, es werden zunächst alle in der ersten Zeile der Bedingungsfelder angegebenen Bedingungen ausgewertet, anschließend die in der zweiten Zeile, dann die in der dritten Zeile usw.

- >=1000 **und** <=2000 (zwischen...beide Werte eingeschlossen)
- „Berlin“ **oder** „Leipzig“
- <55 **und nicht** 20
- >#1.1.2001# **und** <#31.12.2010# (zwischen...)

1.7.3. Verbundene Tabellen

Wenn die Beziehungen zwischen den einzelnen Tabellen richtig fixiert worden sind, kann durch die Definition einer referenziellen Integrität zwischen den Tabellen sichergestellt werden, dass bei jeder Veränderung von Inhalten der Schlüsselfelder (Primärschlüsselfeld und Fremdschlüsselfelder) quer geprüft wird, ob die in Beziehung stehenden Tabellen durch diese Veränderung betroffen sind und wie damit umgegangen werden soll.



1.7.4. Kalkulationsabfragen

In einer Sicht haben Sie die Möglichkeit, Kalkulationsfelder zu definieren. Dies ist dann sinnvoll, wenn Sie z.B. die beiden Felder PREIS und RABATT (in %) in die Sicht aufgenommen haben und über diese Daten den tatsächlichen Nachlass in € berechnen und ausgeben wollen.

Um die Berechnung für das Kalkulationsfeld zu definieren geben Sie den zu berechnenden Ausdruck in das Eingabefeld ein.

Funktionen und Ausdrücke

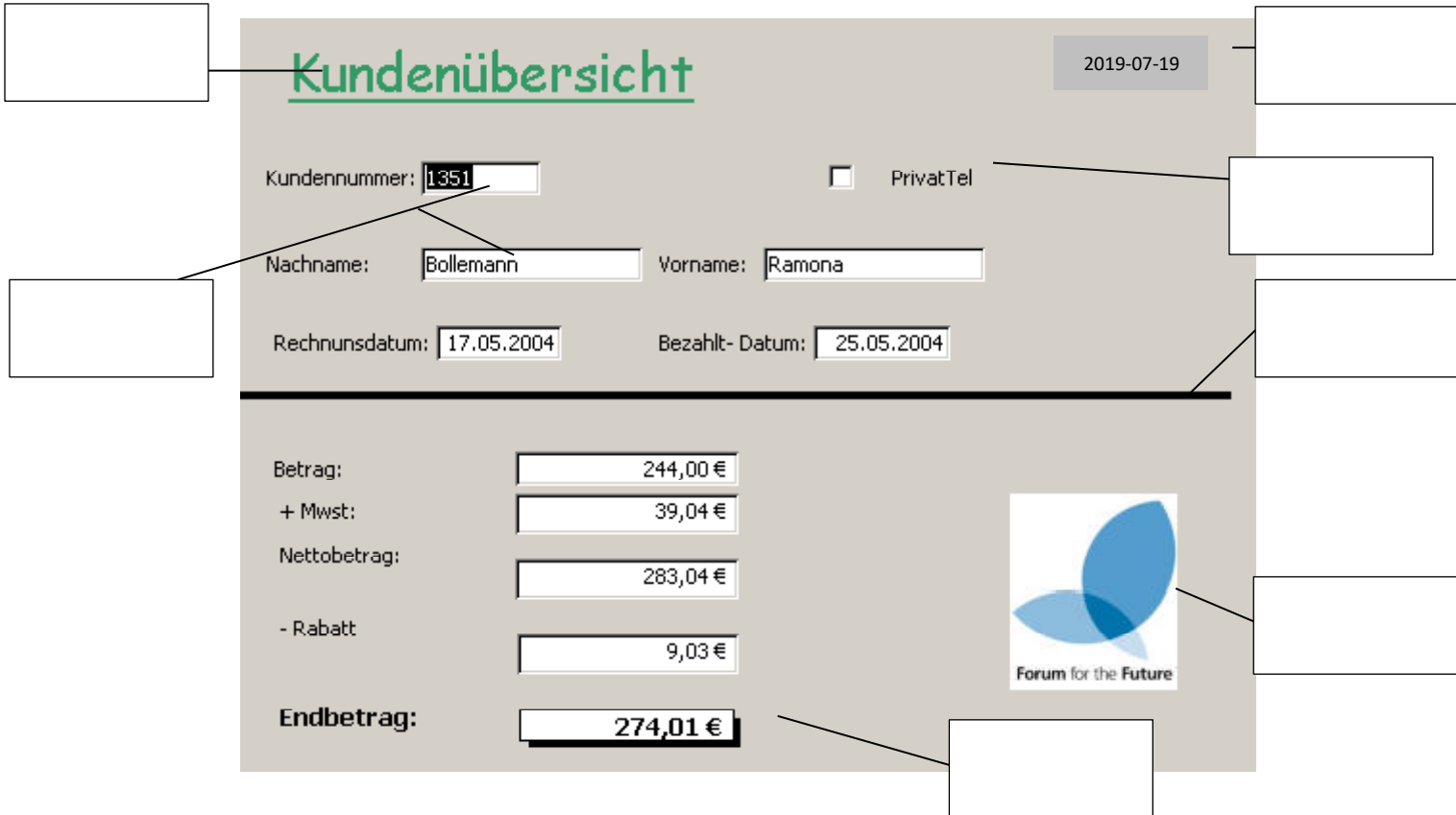
- Mwst:[Preis]*[steuer]/100
- Kosten:[sparen]+[gehalt]-[beitrag]
- Freunde: [vname] & „ „ & [Nachname]

- Wenn(Bedingung; Ja- Teil; Nein- Teil)
- Aussage:wenn([gehalt]>3000;„gut“;„schlecht“)
- Tipp:wenn([wert]>=1 oder [wert]<7; „HURRA“;0)

ACHNAME	Tipp: Wenn([sparen]> 100 Und [sparen]<300;"HURRA";0)
UFTRAG2	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

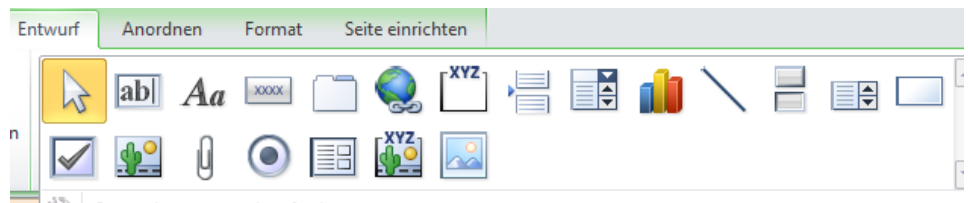
4. Formulare/ Masken

Formulare sind Fenster, in denen Meldungen und Daten angezeigt sowie Reaktionen des Programms auf Benutzereingaben ausgegeben werden. Ausgangspunkt für die Erstellung eines Formulars ist der Formular Designer, d.h. ein leeres Formularfenster in welchem man die erforderlichen Objekte platziert. Sobald die Oberfläche im Wesentlichen erstellt ist, werden den Objekten Eigenschaften zugewiesen, um ihre statischen und dynamischen Merkmale zu definieren.

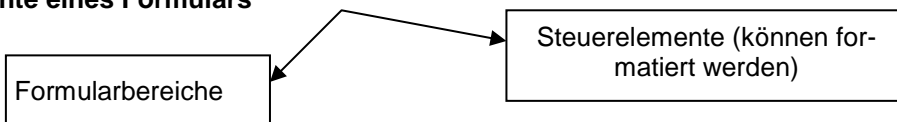


Gestaltungselemente:

Toolbox
Feldliste



Elemente eines Formulars



- Formulkopf
- Detailbereich
- Formularfuß

- Gebundene Steuerelemente (Verknüpfung mit einem vorhandenen Datenfeld)
- Ungebundene Steuerungselemente (für beschreibenden Text, Bilder, Rechtecke)
- Berechnende Steuerelemente (sind mit einem Ausdruck verbunden z.B. =Datum())
- Listenfelder
- Kombinationsfelder